# New Cost-effective VLSI Implementation of Multiplierless FIR Filter using Common Subexpression Elimination

Yasuhiro Takahashi and Michio Yokoyama
Graduate School of Science and Engineering,
Yamagata University, Yonezawa-shi, 992-8510 Japan.
Tel: +81-238-26-3315, Fax: +81-238-26-3315
Email: yas_ta@m.ieice.org

***Abstract*— In this paper, we propose a novel common subexpresson elimination (CSE) method to be used for VLSI design of multiplierless finite impulse response (FIR) filter with a small number of adders and registers. The proposed method is an efficient way to reduce the function blocks using the horizontal and vertical CSE. The FIR filters were synthesized from Verilog HDL code. Area and critical path values were evaluated for 0.35 $\mu$m standard CMOS library. Compared with the previous CSE techniques, the presented approach can save from a minimum of 0.8% up to a maximum 21.5% of gate area. Also, the critical path of proposed method is an average of 3.2% or 17.6% shorter than those of the other methods.**

## I. Introduction

In the VLSI implementation of a linear phase finite impulse response (FIR) filter with fixed coefficients, multiplications require very large silicon areas compared to additions and subtractions. Since constant multiplications are replaced with an additions/subtractions and shifts, we can reduce the silicon area. The shifts also can be realized by using the hard-wired shifters and hence they are essentially area-free.

The complexity of FIR filters is dominated by the number of additions/subtractions used for coefficient multiplications. To reduce the complexity of FIR filters, the coefficients can be expressed in the canonic signed digit (CSD) form. The CSD codes minimize the number of adders/subtracters required in each coefficient multiplications.

The common subexpression elimination (CSE) methods [1]–[7] have been proposed to make the multiplier block as simple as possible. Recently, Jang et al. [5] and Vinod et al. [6] proposed the methods of further reducing the number of adders by using vertical CSE. However, these methods do not consider the number of registers that result from the use of the vertical CSE. The gate number ratio of adders to registers is $1 : 0.6 - 0.8$; therefore, in the case of a filter with many registers, we can not reduce the area of filter. On the other hand, in a previous paper [7], we have reported on the synthesis of multiplierless FIR filter which has a small number of adders and registers, we however have evaluated only the area of FIR filter.

In this paper, we propose a CSE method for designing the high-speed and small-area FIR filter which has a small number of adders and registers. In particular, our CSE method is aimed at reducing the number of not only adders but also registers in the multiplier block. This is achieved by finding the bit-patterns with the minimum cost function in the coefficient matrix. Finally, we compare the performance in terms of area and critical path time through some examples.

## II. Problem Definition

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

$$Y_i = a_{ij} X_i \quad (i = 0, 1, \cdots, N-1;\ j = 0, 1, \cdots, M-1), \quad (1)$$

where $X_i$ and $Y_i$ are input and output variable vectors, respectively. Also, $a_{ij}$ is a set of constant coefficients, $N$ is the number of coefficients and $M$ is the word length. One typical example is the transposed form FIR filter that one input data is multiplied with the filter coefficients. In this paper, we perform multiple multiplications in Equation (1) using registers and additions/subtractions in order to reduce the area. Then the problem of reducing the costs is stated as the problem of minimizing the weighted sum of the numbers of the registers and adders/subtracters which are needed to perform all of the multiplications. That is, the objective cost function ($CF$) to be minimized is written as:

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

where $N_{reg}$ and $N_{a-s}$ are the number of registers and adders/subtracters, respectively, $\beta(> 0)$ and $\gamma(> 0)$ are weights.

The above problem is called the multiple constant multiplication (MCM) problem. However, the MCM problem is very complex so that it is believed to be NP-hard [8]. Therefore, we have to find out an approximate solution by the heuristic approach, like the CSE technique.

## III. 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. We consider here a 26th order FIR lowpass filter as described in [6].

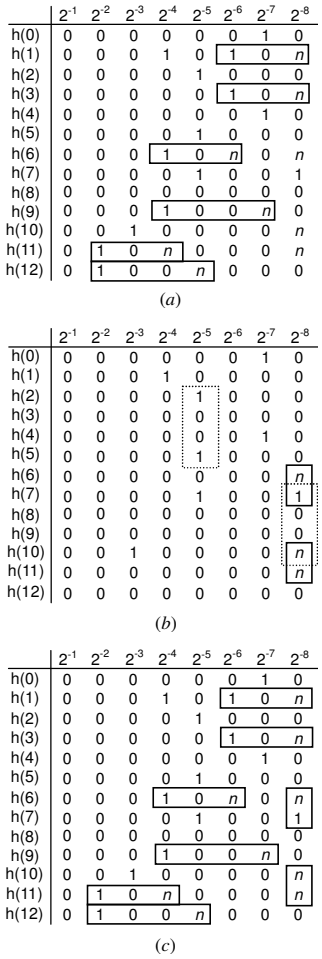


Fig. 1. Proposed horizontal and vertical CSE in 26th order FIR lowpass filter coefficients. (a) Horizontal CSE method. (b) Pattern selection by using vertical CSE method. (c) Final horizontal and vertical CSE method.

### A. 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 I, where $n$ denotes $-1$. This table shows that we should select $10n$ as the most frequency of non-zero bit pattern.

Table II summarizes the frequency of the valid non-zero bit patterns in 26th order FIR lowpass 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/subtracter 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 CSD shown in Fig. 1(a).

TABLE I

NON-ZERO BIT PATTERNS OF COEFFICIENT $010n010n$.

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

TABLE II

FREQUENCY OF BIT PATTERNS IN 26TH ORDER FIR LOWPASS FILTER COEFFICIENTS.

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

### B. Step 2: Vertical CSE Method

The remaining non-zero bits are examined for optimum vertical common subexpression. Pattern identification of vertical CSE is completed through two procedures. At first, we search for the same vertical patterns in order to reduce the number of adders. In this example, same vertical patterns are not included in Fig. 1(b). Secondly, to reduce the number of registers, we search for the similar vertical patterns. Target vertical common subexpressions are surrounded by solid (group-1) and dotted (group-2) line shown in Fig. 1(b). Since the vertical common subexpression increases the number of registers, extra care must be taken to ensure that the FIR filter is constructed to minimize the register produced by the vertical subexpression. Fig. 2(a) explains the structure of group-1 shown in Fig. 1(b). In Fig. 2(a), the cost function $CF_1$ is expressed as:

$$CF_1 = \beta + 2\gamma. \tag{3}$$

On the other hand, Fig. 2(b) represents the structure of group-2 shown in Fig. 1(b). Similarly, the cost function $CF_2$ is as following:

$$CF_2 = 3\beta + 2\gamma. \tag{4}$$

From Equations (3) and (4) we have:

$$CF_2 > CF_1 \quad (\because \beta > 0, \gamma > 0). \tag{5}$$

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

### C. Evaluation of Implementation Cost

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

$$C = A + \alpha D, \tag{6}$$

where $C$ is an implementation cost factor, $A$ is the number of adders, $D$ is the number of registers and $\alpha$ is adder per register ratio (We set the parameter $\alpha$ to 0.6, if we assume that the FIR filters are fabricated in a 0.35 $\mu$m standard CMOS process).

Table III summarizes a comparison of the number of adders and registers for the 26th order FIR lowpass filter and the implementation cost ($\alpha = 0.6$). This table shows that the

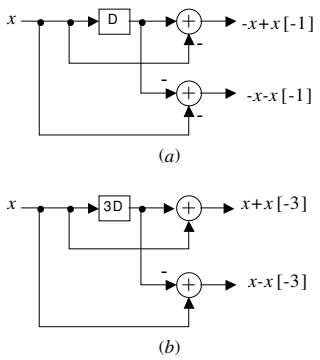


Fig. 2. Implementation by using vertical CSE method. (a) Signal flow of synthesizing $n1$ and $nn$. (b)Signal flow of synthesizing 1001 and $100n$.

TABLE III

COMPARISON OF ADDERS AND REGISTERS REQUIRED TO IMPLEMENT THE 26TH ORDER FIR FILTER.

| | Adder | Register | Cost | Reduction |
|---|---|---|---|---|
| Jang et al. [5] | 32 | 36 | 53.6 | ——— |
| Vinod et al. [6] | 29 | 35 | 50.0 | 6.7% |
| Proposed | 30 | 27 | 46.2 | 13.8% |

TABLE IV

COMPARISON OF LOGICAL DEPTH AND CRITICAL PATH DELAY IN THE MULTIPLIER BLOCK OF 26TH ORDER FIR FILTER.

| | $LD$ | $T_m$ |
|---|---|---|
| Jang et al. [5] | $3+2\delta$ | $11.2T_a$ (100%) |
| Vinod et al. [6] | $2+3\delta$ | $10.3T_a$ (91.9%) |
| Proposed | $2+1\delta$ | $10.1T_a$ (90.1%) |

implementation cost of the proposed method is 13.8% smaller than that of the Jang et al.'s method, and that the implementation cost of the proposed method is $7.1\%$ $(= 13.8 - 6.7)$ smaller than that of the Vinod et al.'s method.

### D. Evaluation of Critical Path in Multiplier Block

The speed of FIR filter is limited by critical path delay and is determined by the logical depth in the multiplier block. In the horizontal CSE method, the logical depth is specified by the number of adder-steps that denotes the maximum number of adders/subtractors allowed to pass though to produce any multiplication. For a set of coefficients, $\{h_0, h_1, \cdots, h_{N-1}\}$, the low bound of adder-steps, $AS$, required in implementing the multiplier block is given by

$$AS = \max\{\lceil \log_2 k_i \rceil\}, \tag{7}$$

where $k_i$ is the number of nonzero digits in the CSD format of $h_i$. In contrast, since the register is required for the multiplier block generated through the vertical CSE method or combining horizontal and vertical CSE method, critical path of register is another important consideration on the logical depth. Therefore, the logical depth in the multiplier block is defined as the following:

$$LD = AS + \delta \times d \tag{8}$$

where $\delta$ is weight and $d$ is the number of register on the critical path. If we assume that the structure of multiplier block uses horizontal and vertical common subexpression blocks, its critical path delay can be written as

$$T_m = \max\{AS \times T_a + d \times T_d + (\log_{1.5} N) \times T_a\}, \tag{9}$$

where $T_a$ is the delay through an adder, $T_d (< T_a)$ is the delay through a register and $N$ is the filter order. The term $(\log_{1.5} N) \times T_a$ is the delay through an adder tree structure (see, e.g. [9] and [10]). This equation includes only arithmetic critical path delay in the architecture. In the actual VLSI design, RC delay due to layout parasitics needs to be taken into consideration.

Table. IV summarizes the maximum logical depth ($LD$) of multiplier block and the critical path delay ($T_m$) given in Equation (9). $T_d$ is estimated to be $T_d = 0.1T_a$ provided that the FIR filter is fabricated in a $0.35$ $\mu$m standard CMOS process. From this results, we found that the proposed method offers a critical path reduction ratio of about 10% over the Jang et al.'s method and of about 2% over the Vinod et al.'s method.

## IV. IMPLEMENTATION RESULTS

In order to validate our proposed method, the examples were synthesized top-down fashion using a CAD system; SYNOPSYS. The filters were fabricated using a Rohm $0.35$ $\mu$m CMOS process. This logic cell has layout parasitic information. Hence, the following results include the RC delay due to circuit parasitics. In the filter structure, we select a bit-parallel implementation. In addition, carry-save adder operators are used both in the multiplier block and the transposed section due to their carry-free properties.

### A. Example 1: 26th order FIR Lowpass filter

Table V summarizes the synthesis results and the performance results obtained by different approaches. As you can see, the implementation result shows that the area of the proposed lowpass filter is 7% smaller than that of the Jang et al.'s method, and that the area of the proposed lowpass filter is 4.7% smaller than that of the Vinod et al.'s method. Also, the proposed filter has the shortest critical path.

### B. Example 2: 15 randomly designed FIR filters

In order to confirm whether the silicon core area and the critical path of FIR filter are dependent on the number of coefficients ($N$) or the size of wordlength ($b$), we use a randomly generated number of coefficients and size of wordlength. The 15 randomly designed FIR filters are included in four types of digital filters: Lowpass, Highpass, Bandpass, and Bandstop.

Table VI compares the silicon core area for the proposed method with those for the conventional method. This table shows that the area of proposed mthod is an average of $6.4\%$ smaller than that of Jang et al.'s method, and is an average of $3.8\%$ smaller than that of the Vinod et al.'s method.

Table VII summarizes the results in term of critical path for 15 randomly designed FIR filters. This table shows that

TABLE V

COMPARISON OF SYNTHESIS RESULTS FOR 26TH ORDER FIR FILTER.

| | Core area $[mm^2]$ | Power dissipation $[mW/\text{MHz}]$ | Path delay $[ns]$ |
|---|---|---|---|
| Jang et al. [5] | 0.72 (100%) | 3.38 (100%) | 39.8 (100%) |
| Vinod et al. [6] | 0.70 (97.2%) | 3.35 (99.1%) | 36.5 (91.7%) |
| Proposed | 0.67 (93.0%) | 3.22 (95.3%) | 34.8 (87.7%) |

TABLE VI

COMPARISON OF AREA RESULTS ON 15 EXAMPLES.

| Design $N \times b$ | Jang et al. $[mm^2]$ | Vinod et al. $[mm^2]$ | Proposed $[mm^2]$ |
|---|---|---|---|
| BPF, 64 | 0.31 (100%) | 0.30 (96.7%) | 0.28 (90.3%) |
| LPF, 99 | 0.47 (100%) | 0.46 (99.1%) | 0.45 (91.7%) |
| BSF, 180 | 0.70 (100%) | 0.61 (87.1%) | 0.55 (78.5%) |
| LPF, 208 | 0.72 (100%) | 0.70 (97.2%) | 0.67 (93.1%) |
| LPF, 225 | 0.92 (100%) | 0.91 (98.9%) | 0.89 (96.7%) |
| HPF, 261 | 0.78 (100%) | 0.78 (100%) | 0.72 (92.3%) |
| BSF, 288 | 0.90 (100%) | 0.92 (102%) | 0.89 (98.8%) |
| LPF, 300 | 1.15 (100%) | 1.17 (102%) | 1.14 (99.1%) |
| BSF, 336 | 2.02 (100%) | 1.86 (92.1%) | 1.84 (91.1%) |
| BPF, 496 | 1.80 (100%) | 1.77 (98.3%) | 1.76 (97.8%) |
| BPF, 605 | 2.50 (100%) | 2.48 (99.2%) | 2.45 (98.0%) |
| LPF, 630 | 1.79 (100%) | 1.72 (96.0%) | 1.69 (94.4%) |
| HPF, 640 | 2.86 (100%) | 2.74 (95.8%) | 2.69 (94.0%) |
| BSF, 649 | 2.60 (100%) | 2.60 (100%) | 2.57 (98.8%) |
| HPF, 720 | 3.54 (100%) | 3.40 (94.4%) | 3.32 (92.2%) |
| Avg. red. | ——— | 2.6% | 6.4% |

TABLE VII

COMPARISON OF CRITICAL PATH RESULTS ON 15 EXAMPLES.

| Design $N \times b$ | Jang et al. $[ns]$ | Vinod et al. $[ns]$ | Proposed $[ns]$ |
|---|---|---|---|
| BPF, 64 | 16.05 (100%) | 15.71 (97.9%) | 15.70 (97.8%) |
| LPF, 99 | 22.72 (100%) | 13.97 (61.5%) | 13.88 (61.1%) |
| BSF, 180 | 11.28 (100%) | 11.21 (99.4%) | 11.16 (98.9%) |
| LPF, 208 | 39.68 (100%) | 36.55 (92.1%) | 34.79 (87.7%) |
| LPF, 225 | 23.50 (100%) | 22.76 (96.9%) | 21.64 (92.1%) |
| HPF, 261 | 16.81 (100%) | 16.19 (96.3%) | 16.04 (95.4%) |
| BSF, 288 | 16.58 (100%) | 15.25 (91.9%) | 12.22 (73.7%) |
| LPF, 300 | 11.03 (100%) | 10.71 (98.0%) | 10.67 (96.7%) |
| BSF, 336 | 43.34 (100%) | 28.47 (65.7%) | 26.58 (61.3%) |
| BPF, 496 | 48.91 (100%) | 34.77 (71.1%) | 34.21 (69.9%) |
| BPF, 605 | 35.72 (100%) | 21.49 (60.2%) | 21.39 (59.9%) |
| LPF, 630 | 10.62 (100%) | 10.60 (99.8%) | 10.18 (95.6%) |
| HPF, 640 | 34.17 (100%) | 33.49 (98.0%) | 31.40 (91.9%) |
| BSF, 649 | 21.71 (100%) | 21.37 (98.4%) | 21.16 (97.5%) |
| HPF, 720 | 41.26 (100%) | 24.62 (59.7%) | 24.27 (58.8%) |
| Avg. red. | ——— | 14.4% | 17.6% |

the proposed method gives the shortest critical path compared to other methods regardless of the coefficient values and the wordlength. Moreover, we found that the critical path proposed method is an average of $17.6\%$ shorter than that of the Jang et al.'s method and is an average of $3.2\%$ shorter than that of the Vinod et al.'s method.

From the results of Example 2 we can conclude the following:

- The area of the proposed FIR filters has the smallest area without dependence on the number of coefficients $N$ and the wordlength $b$.
- The proposed FIR filters has the shortest critical path without dependence on $N$ and $b$.

## V. CONCLUSIONS

We have presented a new technique to be used for VLSI design of CSD FIR filter with a small number of adders and registers. The proposed method has been an efficient way to find the bit-patterns with the minimum cost function in the coefficient matrix. The usefulness of the proposed method has been shown through the examples. The proposed method has given the lowest implementation and the shortest critical path compared to other methods in the examples.

## VI. ACKNOWLEDGEMENT

The custom circuits in this study have been designed with Synopsys CAD tools through the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm Corporation.

## REFERENCES

[1] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," *Proc. IEEE Int. Symp. on 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. on 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. Păsko, P. Schaumout, V. Derudder, S. Vernalde, and D. Ďuračková, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Computer-Aided Design*, 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] Y. Takahashi, K. Takahashi, and M. Yokoyama, "Synthesis of multiplierless FIR filter by efficient sharing of horizontal and vertical common subexpression elimination," *Proc. Int. Tech. Conf. on Circuits/Systems Computers and Communications* (ITC-CSCC 2004), pp.7C2L-4-1–7C2L-4-4, July 2004.

[8] A. Matsuura and A. Nagoya, "Formulation of the addition shift sequence problem and its complexity," *Proc. Int. Symp. on Algorithms and Computation* (ISAAC'97), pp.42–51, Dec. 1997.

[9] P.A. Ruetz, "The architectures and design of a 20-MHz real-time DSP chip set," *IEEE J. Solid-States Circuits*, vol.24, no.2, pp.338–348, Feb. 1989.

[10] R. Jain, P.T. Yang, and T. Yoshino, "FIRGEN: A computer-aided design system for high performance FIR filter integrated circuits," *IEEE Trans. Signal Processing*, vol.39, no.7, pp.1655–1668, July 1991.