Design of a Lower-Error Fixed-Width Multiplier for Speech Processing Application

Lan-Da Van, Shuenn-Shyang Wang*, Shing Tenqchen, Wu-Shiung Feng, and Bor-Shenn Jeng****

Department of Electrical Engineering, Lab 353, National Taiwan University, Taipei, Taiwan, ROC

*Department of Electrical Engineering, Lab 300, Tatung Institute of Technology, Taipei, Taiwan, ROC

**Chunghwa Telecom Telecommunication Labs., 12, Lane 551, Sec. 5, Min-Tsu Rd., Yang-Mei Zien, Tao-Yuan County, Taiwan 326, ROC. E-mail: stc@ms.chttl.com.tw

ABSTRACT

A lower-error and lower-variance $n \times n$ multiplier is suitably proposed for VLSI design. Considering next lower significant stage in P_{n-1} column and useful error-compensation model in the least significant part, and utilizing a near optimized index to classify the error terms are our strategies in order to achieve lower error and variance as compared with previously proposed structure in the subproduct-array of Baugh-Wooley algorithm. This novel structure applied to the fixed-width low-pass digital FIR filter for speech signal processing system has excellent performance in reducing maximum error, average error, and variation of errors as shown in given tables and figures.

1. INTRODUCTION

Low error, high speed, and small area multipliers are always the most important hardware processing element for digital signal processing (DSP) applications [1] such as MPEG (Moving Picture Experts Group), camera recorders, digital filters, and so on. These multipliers based on Baugh-Wooley algorithm [1-2] produce 2*n* bits output with *n*-bit multiplier and *n*-bit multiplicand inputs. However, in practice, one requires *n*-bit output and truncates *n* least-significant bits to preserve the *n* most-significant bits. In this paper, we present an alternative approach to design a lower-error fixed-width $n \times n$ multiplier applying a near optimized index. Thus, we provide a simple solution derived by approaches under considering hardware realization and verified by computer simulation including full search. At last, we successfully apply above structures to the fixed-width low-pass FIR filter for speech processing [3].

2. DESIGN OF A FIXED-WIDTH MULTIPLIER

Considering two 2's complement integer operands, an *n*-bit multiplicand and an *n*-bit multiplier, can be represented by

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i \quad (1)$$

$$B = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j \quad (2)$$

The product P can be usually written as

$$\begin{aligned} P &= A \times B \\ &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j 2^{i+j} \\ &+ 2^{n-1}(-2^{n-1} + \sum_{j=0}^{n-2} \overline{a_{n-1}b_j} 2^j + 1) + \\ &+ 2^{n-1}(-2^{n-1} + \sum_{i=0}^{n-2} \overline{b_{n-1}a_i} 2^i + 1). \end{aligned} \quad (3)$$

Eq. (3) is a Baugh-Wooley array multiplier in which combining partial products with the same weighting factor and placing them in the same column. The algorithm yields the subproduct array as shown in Fig.1 for 8×8 multiplication.

In 1997, Jou and Kuang (*J-K*) [1] provided another way to improve the error compensation. However, they solely improve error but not discuss the model of error compensation and the choice of index, so it is our motivation to improve them. Let

0-7803-5474-5/99/$10.00(C)1999 IEEE

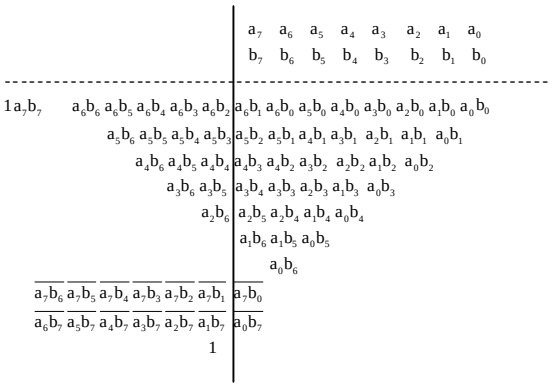

Fig. 1 The subproduct array of 8×8 multiplication.

the product form be as follows:

$$P \cong MP + s_{Temp} \times 2^n ,$$

$$s_{Temp} = \left\lceil \begin{array}{l} \frac{1}{2}(\overline{a_{n-1}b_0} + a_{n-2}b_1 + \cdots + \overline{a_0 b_{n-1}}) \\ + \frac{1}{2^2}(a_{n-2}b_0 + \cdots + a_0 b_{n-2}) + \cdots + \\ \frac{1}{2^{n-1}}(a_1 b_0 + a_0 b_1) + \frac{1}{2^n} a_0 b_0 \end{array} \right\rceil \quad (4)$$

where $\lceil t \rceil$ denotes the minimum integer greater than or equal to t, s_{Temp} is the temporary error-compensation term. Because the variation of the error-compensation terms depends on input signals, we use index to classify the error-compensation terms for hardware realization.

According to index definition as shown below:

$$q_{index} \triangleq < a_{n-1}b_0 >^{q_{n-1}} + < a_{n-2}b_1 >^{q_{n-2}} + \ldots + < a_0 b_{n-1} >^{q_0} \quad (5)$$

where q_0, q_1, …, and q_{n-1} have only two binary values in which 1 and 0 represent the complement of product and the original product without complement, respectively. Then s_{Temp} can be rewritten as

$$s_{Temp} = (< a_{n-2}b_1 >^{q_{n-2}} + \ldots + < a_1 b_{n-2} >^{q_1}) + \left\lceil \begin{array}{l} < a_{n-1}b_0 >^{q_{n-1}} + < a_0 b_{n-1} >^{q_0} - q_{index} \\ + \frac{1}{2}E_{main} + \frac{1}{2}E_{remian} \end{array} \right\rceil \quad (6)$$

where

$$E_{main} \triangleq \overline{a_{n-1}b_0} + a_{n-2}b_1 + \cdots + \overline{a_0 b_{n-1}},$$

$$E_{remain} \triangleq \frac{1}{2}(a_{n-2}b_0 + a_{n-3}b_1 + \cdots + a_0 b_{n-2}) + \cdots + \frac{1}{2^{n-1}} a_0 b_0 .$$

For convenience, we define b as

$$b \triangleq < a_{n-1}b_0 >^{q_1} + < a_0 b_{n-1} >^{q_n} - q_{index} + \frac{1}{2}E_{main} \quad (7)$$

Equation (6) is our first proposed error-compensation model depending on the choice of index. Of course, user can design many kinds of fixed-width multipliers after statistical calculations. We assume the input bits have uniform distribution and then obtain the expected value of $\frac{1}{2}E_{remain}$ to replace $\frac{1}{2}E_{remain}$.

Writing it as following:

$$E\{\frac{1}{2}E_{remain}\} = 1 \times \sum Pb\{a_i b_j = 1 \mid i + j < n-1\}$$

$$\cong \frac{n}{8} - \frac{1}{4} \text{ , if } n \geq 4 \quad (8)$$

Based on full search, we can obtain a near optimized index as:

$$q_{Propose} = \overline{a_{n-1}b_0} + \overline{a_0 b_{n-1}} + \sum_{(i+j)=n-1,\, i=j\neq n-1} a_i b_j \quad (9)$$

Fig. 2 shows that applying the new index has lower variance of compensation than applying J-Ks' index.

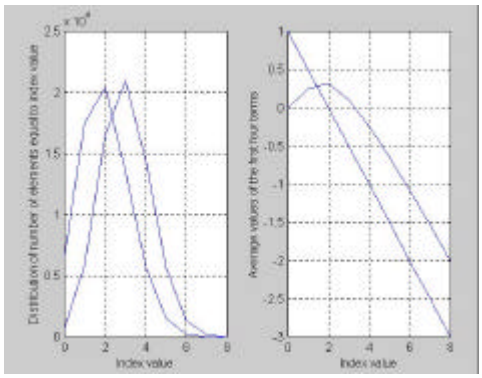

Fig. 2. The comparison results of variation of b between J-Ks' index and proposed index.

Also, we estimate the first four terms in Eq. (7) based on a near optimized index as

$$E\{b\} = 1 \times \sum Pb\{\overline{ab} = 1 \mid \overline{ab} \in P_{n-1}\} + E\{-\frac{1}{2}E_{main}\}$$

0-7803-5474-5/99/$10.00(C)1999 IEEE

$$\cong -\frac{n}{8}+1 \qquad (10)$$

Substituting Eqs. (8), (10) into Eq. (6) and considering the hardware realization, we obtain a new error-compensation term as

$$s_{\Pr opose} = \begin{cases} a_{n-2}b_1 + a_{n-3}b_2 + \cdots + a_1 b_{n-2} + 1, & \text{if } q_{\Pr opose.} < n \\ a_{n-2}b_1 + a_{n-3}b_2 + \cdots + a_1 b_{n-2}, & \text{if } q_{\Pr opose} = n \end{cases} \qquad (11)$$

According to Eq. (11), Fig. 3 is a lower error 8×8 multiplier in which error-compensation circuit is a series of AND-AND gates and three NAND gates.

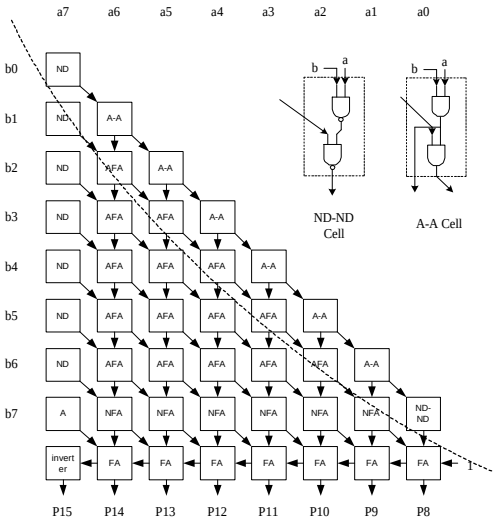

Fig. 3 The proposed lower error 8×8 multiplier.

Utilizing our structure, the maximum error listed in Table 1 can be reduced, and the average error shown in Table 2 obviously decreases compared to Kidambi's and J-Ks' structure. Here, the average error is the sum of the absolute value of error divided by the number of errors. In physical meaning, Table 3 indicates the variation of errors. Especially for 8×8 multiplication, our structure restricts that the maximum error is less than 512, i.e., it guarantees that maximum error doesn't extend to 10-th significant bit.

Table 1: Comparison results of maximum error

Multiplier	n=4	n=8	n=10	n=12
Kidambi's Structure	33	1281	6145	32769
J-K's Structure	21	515	2403	10979
Proposed Structure	17	441	2105	9785

Table 2: Comparison results of average error

Multiplier	n=4	n=8	n=10	n=12
Kidambi's	6.96	188.29	906.40	3842.06
J-K's Structure	7.20	170.46	736.62	3065.25
Proposed	5.17	105.96	456.14	1907.36

Table 3: Comparison results of the number of errors greater than 2^{n+1}

Multiplier	n=4	n=8	n=10	n=12
Kidambi's	1	2717	60970	1570086
J-K's Structure	0	2	1435	78445
Proposed Structure	0	0	8	2254

3. APPLICATION OF FIR FILTER FOR SPEECH PROCESSING APPLICATION

In this section, we apply a new multiplier to the fixed-width 35-tap digital FIR filter [3]. For the consideration of best performance, the maximum coefficient in the FIR filter is normalized to 127 and represented in two's complement. The maximum input speech data is normalized to a maximum integer using 8 bits, that is, its value is 127. The coefficients of 35-point lowpass filter can be found in [3] and voice data, pronounced with "Chicken", are given by 1000 samples as shown in Fig. 4. We use no loss accumulation as standard output as shown in Fig. 5. Using constant bias method [2], Fig. 6 shows much larger variance in consonant part. According to J-Ks' method, it shows better performance in Fig 7 than that in Fig. 6. But as compared to standard output, we find that the output signals in Fig. 7 still have large variance. The smaller variance of the output signals as shown in Fig. 8 is obtained by using a near optimized index especially for consonant part.

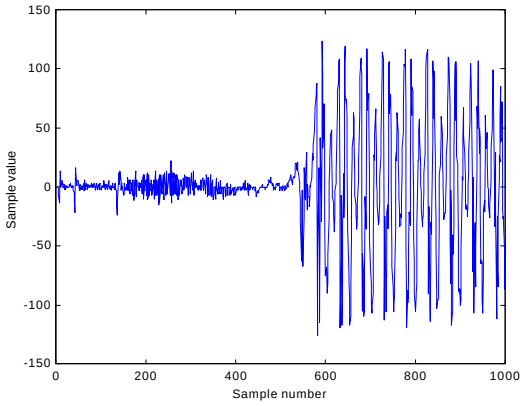

Fig. 4. Original input voice signal.

0-7803-5474-5/99/$10.00(C)1999 IEEE

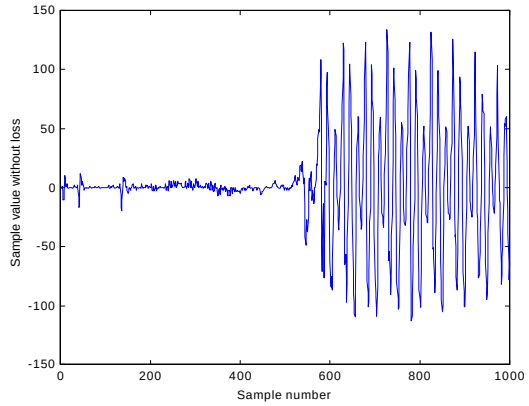

Fig.5. Standard filtering output signals without loss.

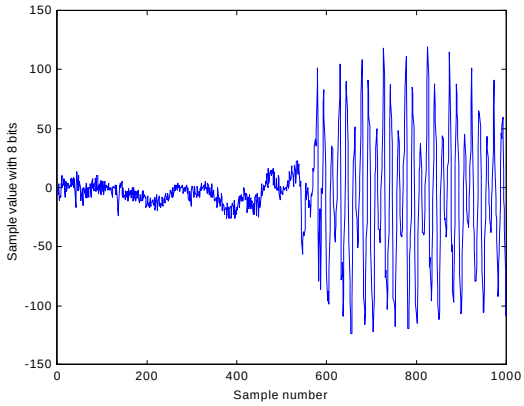

Fig. 6. Output signals using Kidambi's structure.

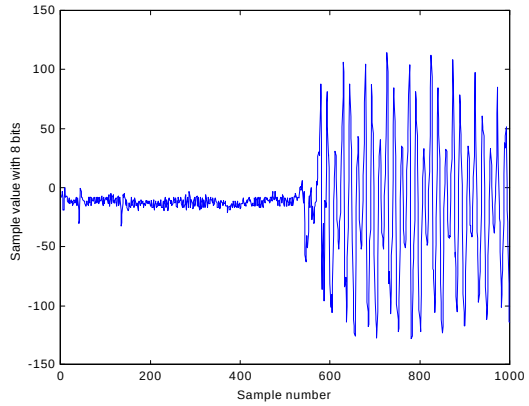

Fig. 7. Output signals using J-Ks' structure.

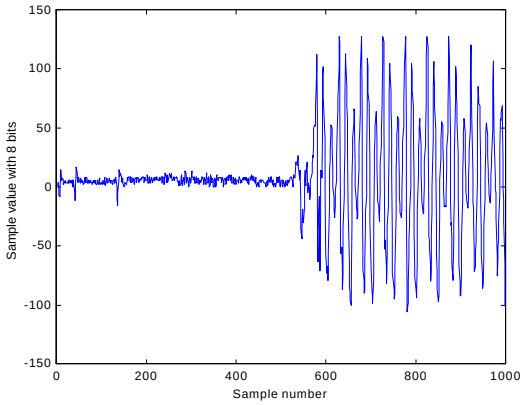

Fig. 8. Output signals using proposed structure

4. CONCLUSIONS

A 2's-complement lower-error fixed-width multiplier that receives two n-bit numbers and produces an n-bit product, has been developed. Our paper proposes an efficient error compensation model in hardware for VLSI design. Based on this model, one can easily design any fixed-width multiplier after statistic calculation Also, we provide a better index choice to allow the error compensation to be near optimized. This novel structure to fixed-width digital FIR filter for speech signal processing has shown with excellent performance in maximum error, average error, and variance compared with the performance of Kidambi's and J-Ks' structures.

References:

[1] J. M. Jou, and S. R. Kuang, 'Design of low-error fixed-width multiplier for DSP applications', *Electron. Lett.*, 1997, **33**, (19), pp. 1597-1598

[2] S. S. Kidambi, F. EL-Guibaly, and A. Antoniou, 'Area-efficient multipliers for digital signal processing applications', *IEEE Trans. Circuits Syst. II*, **43**, (2), pp. 90-94, 1996

[3] M. E. Paul, and K. Bruce, *C Language Algorithms for Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

0-7803-5474-5/99/$10.00(C)1999 IEEE