

Low-Error Vector compensation and correction of Dual Group Input Vector and Hardware-Efficiency Using Fixed-Width Multiplier

Rayala Mahesh¹, P. M. Francis², B. Prasad Kumar³

¹M. Tech, GITAS College, Bibbili, A. P, India

²Department of ECE, HOD, Assistant Professor, GITAS, College, Bobbili, A.P, India

³Department of ECE, Assistant Professor, GITAS College, Bobbili, A.P, India

Abstract: *In this paper, we describe a new novel algorithm for design of low power an hardware efficient error compensation circuit by using the dual group minor input correction vector to lower input correction vector compensation error. The on chip soc applications increases the capacity of data transfer that can be utilizing the symmetric property of the minor input correction vector, and complex hardware of the error compensation circuit can be lowered. The error compensation circuit mainly from the “outer” partial products, the hardware complexity only increases slightly as the multiplier input bits increase. By the utilization of LSB techniques In the proposed 16 X 16 bits fixed-width multiplier, the truncation error can be lowered by 87% as compared with the direct-truncated multiplier and the transistor count can be reduced by 47% as compared with the full-length multiplier. With the help of fixed-width multiplier performs not only with lower compensation error but also with lower hardware complexity, especially as multiplier input bits increase.*

Keywords: Fixed-width multiplier, hardware-efficient, low-error.

1. Introduction

In many high-speed digital signal processing (DSP) and multimedia applications, the multiplier plays a very important role because it dominates the chip power consumption and operation speed. In DSP applications, in order to avoid infinite growth of multiplication bit width, we usually have to reduce the number of multiplication products. Cutting off n-bit less significant bit (LSB) output can construct a fixed-width multiplier with n-bit input and n-bit output. However, truncating the LSB part leads to a large number of truncation errors. Many truncation error compensation techniques [1]–[10] have been presented to design an error compensation circuit with less truncation error and less hardware overhead. The compensation methods can be divided into two categories: compensation with constant correction value [1]–[3] and compensation with variable correction value [4]–[10]. The circuit complexity to compensate with constant corrected value can be simpler than that of variable correction value; however, the variable correction approaches usually can be more precise. In the approaches with variable correction value, literature [4] proposed an input-dependent method by using probability, statistics, and linear regression analysis to find the approximate compensation value. The error compensation circuit is constructed by the partial product terms with the most-significant weight in the least-significant segment. The compensation value is dependent on the input number and thus has less truncation error. In [5], the error compensation algorithm made use of binomial distribution instead of uniform distribution used in [4] to model the probability of occurrence of multiplier inputs.

This modification can bring a more precise error compensation result. Moreover, the compensation vector in [5] can directly inject into the fixed-width multiplier as compensation, which does not need extra compensation logic gates. Therefore, the fixed-width multiplier area can be smaller than [4]. In [6], a two-dimensional conditional estimation method was proposed to compensate truncated error based on both the dependency among the partial product terms and multiplication inputs. The error compensation in [6] can be more precise; however, the hardware is too complex. In [7], [8], multiple-input error compensation vector designs were proposed to further enhance the error compensation precision. Unlike [4] or [5] to set the same weight for each partial product terms in the input correction vector, they applied different weights to each input correction vector element. In [8], “inner” partial products were designed to have a higher weight with respect to “outer” partial products. To take into account different weights of input correction (IC) partial products, the IC vector was divided into two disjointed sets with dual addition trees to compute the error compensation value. In this way, the compensation value can be more approximated to the expected results. Hence it performed better results in terms of error compensation. Recently, the design in [8] was further extended in [9] and [10]. In [9], a parallel configurable error-compensation circuit was proposed to perform nearly the same error compensation precision as [8], but with lower computation delay. In [10], a variable correction to include the partial products of LSB part was proposed to trade-off between hardware complexity and error compensation precision. Nowadays [8]–[10] are the state-of-the-art fixed-width multiplier designs that can

perform lower error with efficient hardware. In this paper, we consider the impact of truncated products with the second most significant bits on the error compensation, which is similar to [10] but with lower hardware complexity. We propose a new error compensation circuit by using the dual group minor input correction (MIC) vector to further lower IC vector compensation error in [8]. By utilizing the symmetric property of MIC, fan-in can be reduced to half and hardware in up-MIC and down-MIC can be shared. Therefore, the hardware complexity of error compensation circuit can be lowered. Moreover, the hardware complexity just increases slightly as the multiplier input bits increase because we construct the proposed error compensation circuit mainly by the “outer” partial products. As compared with the state-of-the-art design in [8]–[10], the proposed fixed-width multiplier not only performs with lower compensation error but also with lower hardware complexity, especially as multiplier input bits increase.

2. Proposed Error Compensation Circuit Design by Using the Dual-Group Minor Input

2.1 Correction Vector

Baugh-Wooley array multiplier with two unsigned P-bit inputs of X and Y, which are shown as

$$X = \sum_{i=0}^{n-1} x_i \cdot 2^i, \quad Y = \sum_{j=0}^{n-1} y_j \cdot 2^j$$

The multiplication result P_i is the summation of partial products of $x_i y_j$ which is shown as

$$P_i = \sum_{j=1}^{n-1} y_j 2^j \sum_{i=n-j}^{n-1} x_i 2^i + f(x_0 y_{n-1}, x_1 y_{n-2}, \dots, x_{n-2} y_1, x_{n-1} y_0) = \sum_{j=1}^{n-1} y_j 2^j \sum_{i=n-j}^{n-1} x_i 2^i + f(IC)$$

The full-length: bit unsigned Baugh-Wooley partial product array can be divided into three subsets of most significant part (MSP), IC vector and less significant part (LSP) as shown in Fig. 1. To evaluate the accuracy of a fixed-width multiplier, we can exploit the difference between the n-bit fixed-width multiplier output and the P-bit full-length multiplier output, which is expressed as $E = P - P_t$

where P is the output of the complete multiplier, and P_t is the output of the fixed-width multiplier. P_t can be expressed as

$$f(IC) = \begin{cases} 0, & \text{if } \beta = 0 \\ \beta, & \text{if } (x_{n-1}y_0 + x_{n-2}y_1 + x_1y_{n-2} + x_0y_{n-1} = 0, 1) \\ \beta - 1, & \text{if } (x_{n-1}y_0 + x_{n-2}y_1 + x_1y_{n-2} + x_0y_{n-1} = 2, 3) \\ \beta - 2, & \text{if } (x_{n-1}y_0 + x_{n-2}y_1 + x_1y_{n-2} + x_0y_{n-1} = 4) \end{cases}$$

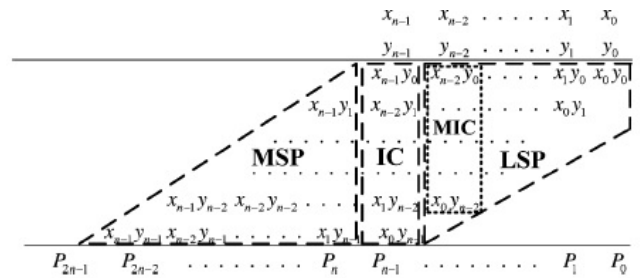


Figure 1 n-bit Baugh-Wooley multiplier partial product array consists of MSP, IC, and LSP, in which MIC is the partial product vector with the most significant bits of LSP. Where $f(IC)$ is the error compensation function. In [8], the error compensation function $f(IC)$ is approximated as the sum of input correction vector with corresponding weight. To realize, $f(IC)$ the error compensation vector is divided into two disjointed sets and uses two addition trees to compute the error compensation. The error compensation algorithm in [8] is developed as

3. Proposed Error Compensation Method

Literature [8] and [10] are the state-of-the-art designs that can perform the most precise error compensation with efficient hardware among the previous published fixed-width multipliers. However, there are still some compensation errors $|E| > 2^{n-1}$ existing in [8]. The compensation errors can be divided into two categories: the first type is caused by insufficient error compensation, in which output P_t is smaller than ideal value p . In this case $|E| = p - p_t > 0$. On the other hand, the second type is due to over error compensation, in which output is larger than ideal value. In this case, $E = P - P_t > 0$ to consider both approximation error and circuit complexity, we mainly aim at dealing with the case of

Table 1: average value of sum of the ic and lsp partial products

Row	IC	Savg
1	1,0,0,0,0,0,0,0,0	0.944
2	0,1,0,0,0,0,0,0,0	0.999
3	0,0,1,0,0,0,0,0,0	1.025
4	0,0,0,1,0,0,0,0,0	1.035
5	0,0,0,0,0,1,0,0,0	1.035
6	0,0,0,0,0,0,1,0,0	1.025
7	0,0,0,0,0,0,0,1,0	0.999
8	0,0,0,0,0,0,0,0,1	0.944

$|E| > 2^{n-1}$ in this paper. The weight of IC compensation circuit is $2n$. We cannot correct all the cases of effectively if we only apply the partial product terms in IC to construct the error compensation function. Therefore, in this paper we adopt IC together with MIC, where MIC is the partial product vector with the most significant bits of LSP, to amend the error compensation value of $F(IC)$. In this way, the cases of can be reduced effectively. In [8], IC compensation circuit is constructed by dual IC compensation trees, which are the “inner” partial products with higher compensation weight and the “outer” partial products with lower compensation weight. According to the relation of IC and

savg(IC) in Table I, we can find out that the average compensation errors in the outer part and inner part are nearly the same, where the average compensation error is 0.0285 in the outer part and it is 0.0300 in the inner part. Here Savg(IC) the average value of sum of the IC and LSP partial products. However, the number of partial product items with higher weight will increase with the number of bits, while the number of partial product items with lower weight is fixed. Therefore, we only analyze the error compensation tree with lower weight to find out the cases of $|E| > 2n-1$. Then we combine IC with MIC to adjust the function of to make the compensation error over than $2n-1$. In this way, the error compensation circuit can be relatively simple and the compensation error can be lowered more efficiently. To find out a precise error correction vector, we analyze the sum of total errors in the cases $|E| > 2n-1$ and $|E| < 2n-1$ under various β values in accordance with the compensation algorithm in (5). In order to achieve an efficient error correction, we only amend the error compensation function $F(IC)$ under the cases that the total error summation value of $|E| > 2n-1$ and $|E| < 2n-1$ it can be observed that some under-compensated errors occur when $\beta = 2$ and $\beta = 4$ a result, we combine IC with MIC to correct the under-compensated situations under the cases of $\beta = 2$ and $\beta = 4$. As for the case of $\beta = 1$ there exists some over compensation errors. However, the total error summation value of $|E| < 2n-1$ is the above the same with that of $|E| > 2n-1$. We combine IC with MIC to correct the over compensation situations under the case of $\beta = 1$ and $Sch \neq 0$ instead of the case $\beta = 1$ only since in such case the error summation value of $|E| > 2n-1$ is much lower. Here Sch is the summation of IC that with higher weight, which can be written as the lower unit with the second most significant bits of truncated partial products, is adopted as minor input correction (MIC) vector to reduce the compensation error, which is defined as

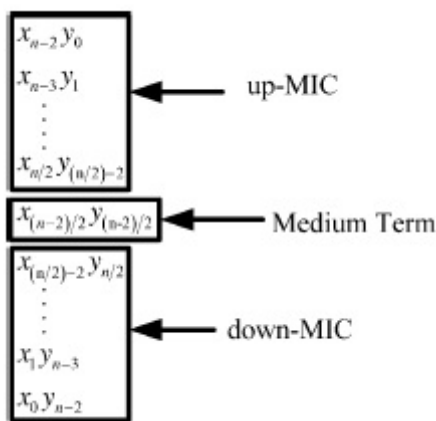


Figure 2: MIC is divided into up-MIC, medium term, and down-MIC.

Its illustration is shown in the Fig 1. There is a systematic relation between IC and Savg (IC) as illustrated In table 1 that is $Savg = (1,0,0,0,0) = Savg (0,0,0,0,0,1)$, $Savg (0,0,1,0,0,0) = Savg (0,0,0,0,0,1,0,0) = Savg (0,0,0,10,0,0) = Savg (0,0,0, 0, 10 ,0,0,0)$ Similarly, this symmetric relation exists between MIC and savg (IC) Therefore, we divide the MIC vector into two groups in order to save hardware cost.

We set the middle item, $X(n-2) / 2Y(n-2)/2$ of MIC as the dividing line. The upper MIC is defined as up-MIC

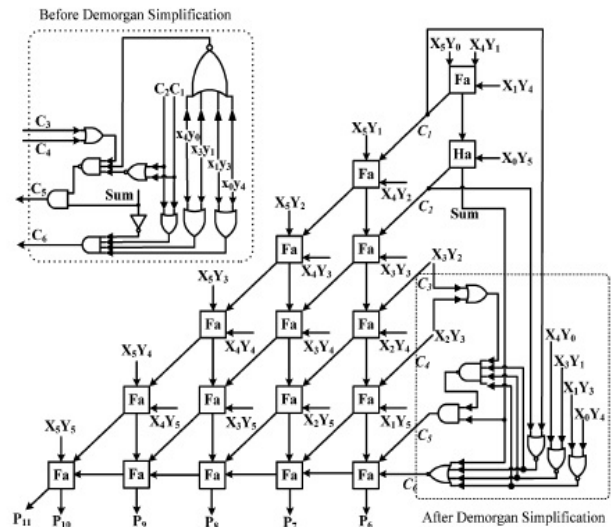


Figure 3: Proposed low- error and area efficient fixed width multiplier

In the under compensation case $\beta = 2$ and $\beta = 4$ we inject one more compensation carry C_n to modify the error correction vector [8] from $\beta = 1$ and $\beta = 2$ to $\beta = 1$ respectively, when both of Sup-MIC and S down-MIC $\neq 0$. On the other hand in the case of the over compensation as $\beta = 1$ and $Sch \neq 0$ all, $E < 2n-1$ Proposed Error Compensation Circuit Design The error compensation circuit we proposed is modified from the dual-tree design [8]. To further reduce the compensation errors.

4. Experiment Result Comparisons

In this section, we compare the proposed fixed-width multiplier with other literature designs [4]–[10] to analyze their approximation error and hardware complexity, respectively. All performance comparisons are evaluated from 8-, 12-, to 16-bit. To analyze the compensation error, we inject all possible input patterns into the fixed-width multiplier. Then we compare the truncated output with their corresponding full-length multiplier output. By exploiting the difference between the n-bit fixed-width multiplier output and the -bit full-length multiplier output, we can obtain each error term. For truncation error comparison, we define the index of mean square error In general, to achieve lower compensation error needs more complex compensation algorithm and more complicated circuit hardware. In this paper, we combine IC with MIC to adjust the function of to lower the compensation error. We also analyze the error compensation tree only with lower weight to find out the cases in our proposed design. Therefore, circuit complexity in the most error compensation circuit is fixed, which will not increase along with input bit number. As a result, the error compensation circuit can be relatively simple, especially as the input bit number increases. As illustrated in Fig. 7, the slope of transistor count increasing as the fixed-width multiplier input number increases is gentler in our proposed design. Though in our proposed design we must spend more transistor count in the 8-bit

fixed-width multiplier, we spend less transistor count in the cases of input bit number are larger than eight. The superiority in area-efficiency in our design is more obvious as input number increases. Finally, we implement the proposed 16-bit low-error, area-efficient fixed-width multiplier in TSMC 0.18- μ m process as illustrated in Fig. 8. The silicon chip area of the proposed fixed-width multiplier circuit is 109.8 m by 106.8 m. As compared with [8], the critical paths in both our design and [8] are located in the path of In both designs the circuit delay are nearly the same under various timing constraints, which all are faster than the conventional ripple designs. The circuit layout area and power consumption in the proposed design is slightly lower than that of [8] since lower transistor count and less wire connection in the error compensation circuit even though our design is more irregular.

5. Conclusion

In this paper, a low-error and area-efficient fixed-width multiplier by using the dual group minor input correction vector is presented. The fixed-width multiplier performs not only with lower compensation error but also with lower hardware complexity, especially as multiplier input bits increase. The proposed 16-bit fixed-width multiplier circuit describes the multilayer with the help of Verilog xilinx12.1i the design is working perfectly and the proposed method is working well.

References

- [1] Y. C. Lim, "Single-precision multiplier with reduced circuit complexity for signal processing applications," *IEEE Trans. Comput.*, vol. 41, no. 10, pp. 1333–1336, Oct. 1992.
- [2] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in *Proc. Workshop VLSI Signal Process.*, 1993, vol. VI, pp. 388–396.
- [3] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 2, pp. 90–95, Feb. 1996.
- [4] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixed width multipliers for DSP applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 46, no. 6, pp. 836–842, Jun. 1999.
- [5] S. J. Jou and H. H. Wang, "Fixed-width multiplier for DSP application," in *Proc. IEEE Int. Symp. Comput. Design*, 2000, pp. 318–322.
- [6] Y. C. Liao, H. C. Chang, and C. W. Liu, "Carry estimation for two's complement fixed-width multipliers," in *Proc. Workshop Signal Process. Syst.*, 2006, pp. 345–350.
- [7] F. Curticeanu and J. Niittylahti, "A hardware efficient direct digital frequency synthesizer," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst.*, 2001, vol. 1, pp. 51–54.
- [8] A. G. M. Strollo, N. Petra, and D. D. Caro, "Dual-tree error compensation for high performance fixed-width multipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 8, pp. 501–507, Aug. 2005.
- [9] S. R. Kuang and J. P. Wang, "Low-error configurable truncated multipliers for multiply-accumulate applications," *Electron. Lett.*, vol. 42, no. 16, pp. 904–905, Aug. 2006.
- [10] N. Petra, D. D. Caro, V. Garofalo, N. Napoli, and A. G. M. Strollo, "Truncated binary multipliers with variable correction and minimum mean square error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.