

Design Of Low Error And Power Fixed Width Multiplier By Using Compensation Function

S. Eswari
PG Scholar

Srinivasan Engineering College, India

Abstract—In this paper, a new error-compensation network for fixed-width multiplier is proposed. The error compensation block is composed of dual trees which are optimally chosen in order to minimize either the mean-square error or the maximum absolute error. The new technique significantly improves error performance with respect to previous approaches. Simulation results show that new fixed-width multipliers exhibit significant improvements both in mean square error and in power dissipation with respect to previous solutions. As compared with the state-of-the-art, the proposed fixed-width multiplier performs not only with lower compensation error but also with lower hardware complexity, especially as multiplier input bits increases.

Index Terms—Digital integrated circuits, Fixed-width multipliers, hardware-efficient, low-error.

I. 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.

Many techniques have been proposed which exploit the “fixed-width” property to reduce hardware complexity with respect to rounded full-

width multiplier [12], [15]–[18]. In order to simplify the review and the comparison of these techniques, let us subdivide the partial products in the three subsets most significant part (MSP), input correction vector (IC), and less significant part (LSP) shown in Fig. 1.

The approximation error of fixed bias correction (13) is investigated in [16] by Lim. It is shown that the error rapidly increases with multiplier size. The error can be reduced by retaining more partial products (for instance the IC partial products) before adding the fixed bias K. Obviously, this results in a tradeoff between precision and hardware complexity.

In [15], Kidambi *et al.* simplify the multiplier by deleting both IC and LSP partial products. A pre-computed constant is added to the final output in order to compensate for the introduced error. The fixed-width multiplication is hence approximated as follows in (1):

$$P_{\text{Kidambi}} = \sum_{j=1}^{n-1} y_j 2^{-j} \sum_{i=1}^{n-j} x_i 2^{-i} + K. \quad (1)$$

This technique provides a hardware complexity about halved with respect to a full multiplier. However, the introduced error is high, reducing practical applications.

A multiplier calculates $P=X.Y$ as weighted sum of partial products $x_i y_j$

$$P = \sum_{k=1}^{2n} p_k \cdot 2^{-k} = \sum_{j=1}^n \sum_{i=1}^n x_i y_j \cdot 2^{-i-j}. \quad (2)$$

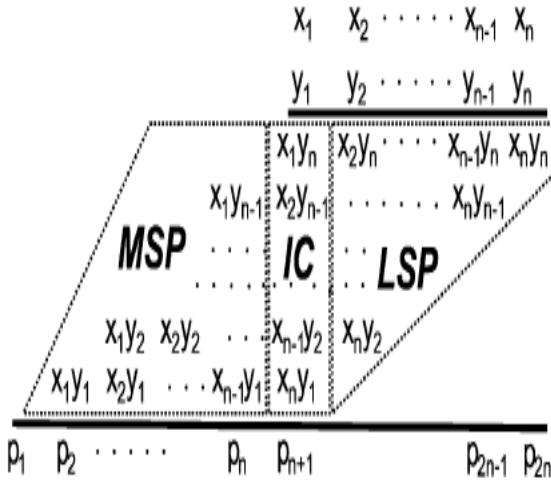


Fig.1. Full multiplier partial product matrix

II. FIXED-WIDTH MULTIPLIERS ERRORS

A. Error Metric

The accuracy of a fixed-width multiplier can be evaluated considering the introduced error with respect to the output of the -bit complete multiplier:

$$\epsilon = P - P_t \tag{4}$$

where P is the output of the complete multiplier given by (2), and P_t is the output of the fixed-width multiplier. As error metric we consider either the normalized maximum absolute error (ϵ_{max}) or the normalized mean-square error (ϵ_{ms}) defined as

$$\epsilon_{max} = \max (|\epsilon| / LSB) \tag{5}$$

$$\epsilon_{ms} = E \{ \epsilon^2 / LSB^2 \} \tag{6}$$

Where $E\{\}$ is the average operator, while $LSB=2^{-n}$ is the weight of the less significant bit at the output of the multiplier. Another parameter useful to characterize fixed-width multipliers accuracy is the normalized mean error (ϵ_m), given by

$$\epsilon_m = E \{ \epsilon / LSB \} \tag{7}$$

An improved fixed-width multiplication algorithm, named partial product “conditional correction”, is also proposed in [16]. This algorithm, basically, exploits the correlation between the IC partial products and the sum of LSP partial products.

Neither algorithm hardware implementation nor circuit performance analysis is given in [16].

The conditional correction algorithm is further developed in [17] by Jou *et al.*. In the Jou architecture, the IC partial products are summed to compute an intermediate quantity S_{IC}

$$S_{IC} = x_1 \cdot y_n + x_2 \cdot y_{n-1} + \dots + x_n \cdot y_1 \tag{3}$$

The sum S_{IC} is then used to calculate a correction factor that estimates the sum of dropped partial products.

In this paper, a new approach to design high performance unsigned fixed-width multipliers is proposed. The multiplier is based on multiple-input error-compensation architecture, like [12], [18]. A new error-compensation function $f(\cdot)$ is developed, that can be optimized in order to minimize either the maximum absolute error or the mean-square error. Our error-compensation function, moreover, can be implemented by using only a few gates, with tree architecture. As a consequence, proposed approach is ideally suited for fast tree-based multipliers [14].

The Results for a circuit implementation in 0.35- μ m technology and a comprehensive comparison with previously proposed techniques are also reported in the paper.

B. Errors in Rounded Full-Width Multipliers

The simplest way to obtain a fixed-width multiplier is through a rounded, full-width multiplier. Rounding introduces a quantization error, that is well known to provide $\epsilon_{max} = 1/2$ and $\epsilon_{ms} = 1/12$ [16]. These values are a lower bound for the errors achievable with any fixed-width multiplier, since full-width multiplier rounding is the most accurate fixed-width technique.

C. Error Bounds for Fixed-Width Multipliers with Multiple-Input Error Compensation

Let us consider a fixed-width multiplier design is given by

$$\epsilon = P - P_t = s(x_1, \dots, x_n; y_1, \dots, y_n) - f(IC) \tag{8}$$

where $s(x_1, \dots, x_n; y_1, \dots, y_n) = s(x; y)$ is the sum of the IC and LSP partial products. The accuracy of fixed-width multipliers with multiple-input error compensation depends on the choice of error-compensation function. The electrical performance depends on implementation of error-compensation function.

TABLE I

PERFORMANCES OF FIXED-WIDTH MULTIPLIERS BY USING DUAL TREE ERROR COMPENSATION

N	Architecture	Error (%)	Area $10^3 \mu\text{m}^2$	Power $\mu\text{m/MHZ}$
12	Rounded	9.098	68.60	112.89
12	Existing fixed width	3.04	45.90	70.90
12	Proposed fixed width	2.11	33.94	55.89
16	Rounded	16.181	120.28	198.64
16	Existing fixed width	2.30	80.80	112.01
16	Proposed fixed width	2.0	59.86	99.90

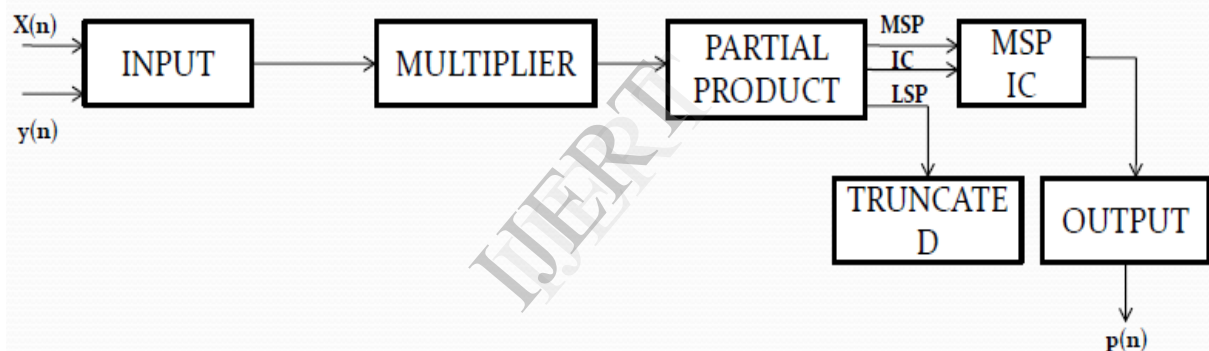


Fig.2. Block diagram for fixed width multiplier

The block diagram show that once multiplication is completed .The partial product is divided into most significant part, input correction and least significant part. The least significant part is truncated from most significant part and input correction vector. 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.Table1 describe the performances of fixed-width multipliers by using dual tree error compensation.

D. Dual Tree Architecture

The architecture of proposed error-compensation block is shown in Fig. 3. To take into account different weights of IC partial products, we divide the

input correction vector in two disjointed sets and use two addition trees to compute the error compensation.

The optimal IC subdivision (between standard and modified summation trees) and the optimal mixing block configuration have been obtained through exhaustive search. We realized two optimizations. In the first one, we assumed as a goal function the absolute error (ϵ_{\max}), whereas the second optimization was carried out to minimize the mean-square error (ϵ_{ms}). This second addition tree uses modified half-adders (mHAs) to take into account the contribution of partial products with higher weights.

The dual-tree architecture has been obtained heuristically, after observing that the error compensation function can be approximated as a weighted sum of input correction vector partial products. In order to introduce our approach with the help of an example, let us consider a 6-bit fixed-

width multiplier, with optimized mean-square error. we can eliminate the modified tree altogether, by sending the partial products originally assigned to the modified tree directly to the carry-save adder, with a weight LSB.

For this type architecture, it can be demonstrated that the final subtraction and the mixing block correspond to the inclusion of a NOR and an AND gate as shown in Fig. 4.

The best accuracy is obtained by designing the error-compensation function according to either (16) or (21). This solution, however, calls for a lookup table to implement either the or functions. Lookup table complexity grows exponentially with, rapidly becoming an impractical solution.

II. CIRCUITS PERFORMANCES

We implemented rounded full-width multipliers, Jou [17], Curticapean [12], and the optimize dual-tree fixed width multipliers proposed in this paper using a three metal 0.35- m technology with 3.3-V supply voltage. In order to have a realistic and accurate indication of the architectures performances, we implemented the carry-save tree of all multipliers using the three-dimensional reduction method (TDM) proposed in [19]. TDM is a state of the art technique to add elements of partial products matrix with a tree based carry-save approach, compensating for different delays in partial products generation, and exploiting delays asymmetries in full-adders to improve overall timing. Silicon area of developed dual-tree multipliers is slightly reduced with respect

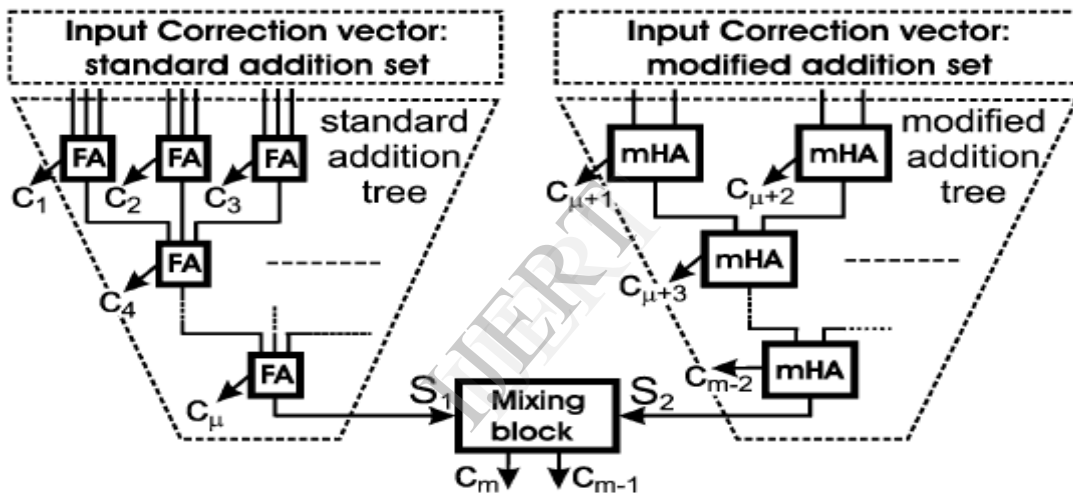


Fig.3. Architecture of dual-tree error compensation block

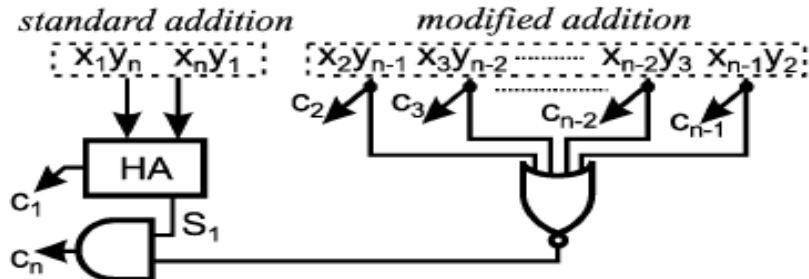


Fig.4. Optimized implementation of dual-tree error-compensation blocks

to Jou and Curicapean solutions, with an area reduction of about 6% for n=16. Obviously, the advantage with respect to complete rounded multiplier is much more evident, with area reduction of about 50%.

According to table 1, due to the reduced glitching in the partial products generation, the proposed circuits exhibit a lower power dissipation with respect to Jou and Curticapean solutions for n>4 . For instance, power saving is about 11% for n equal to 16. Power dissipation is almost halved with respect to the complete rounded multiplier.

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.

IV SIMULATION RESULT AND DISCUSSION

Based on the concept in the previous section, we have designed a FWM for 16-bit with reduced error and low power. The FWM was analyzed using Modelsim simulator at the system level. Modelsim is a simulation and debugging environment created by Mentor Graphics. Modelsim allows you to check the syntax and verify the functionality of VHDL programs.

Modelsim uses libraries in two ways:

- 1) As a local working library that contains the compiled version of your design;
- 2) As are source library. A common example of using both a working library and a resource library is one where your gate-level design and test bench are compiled into the working library and the design references gate-level models in a separate resource library.

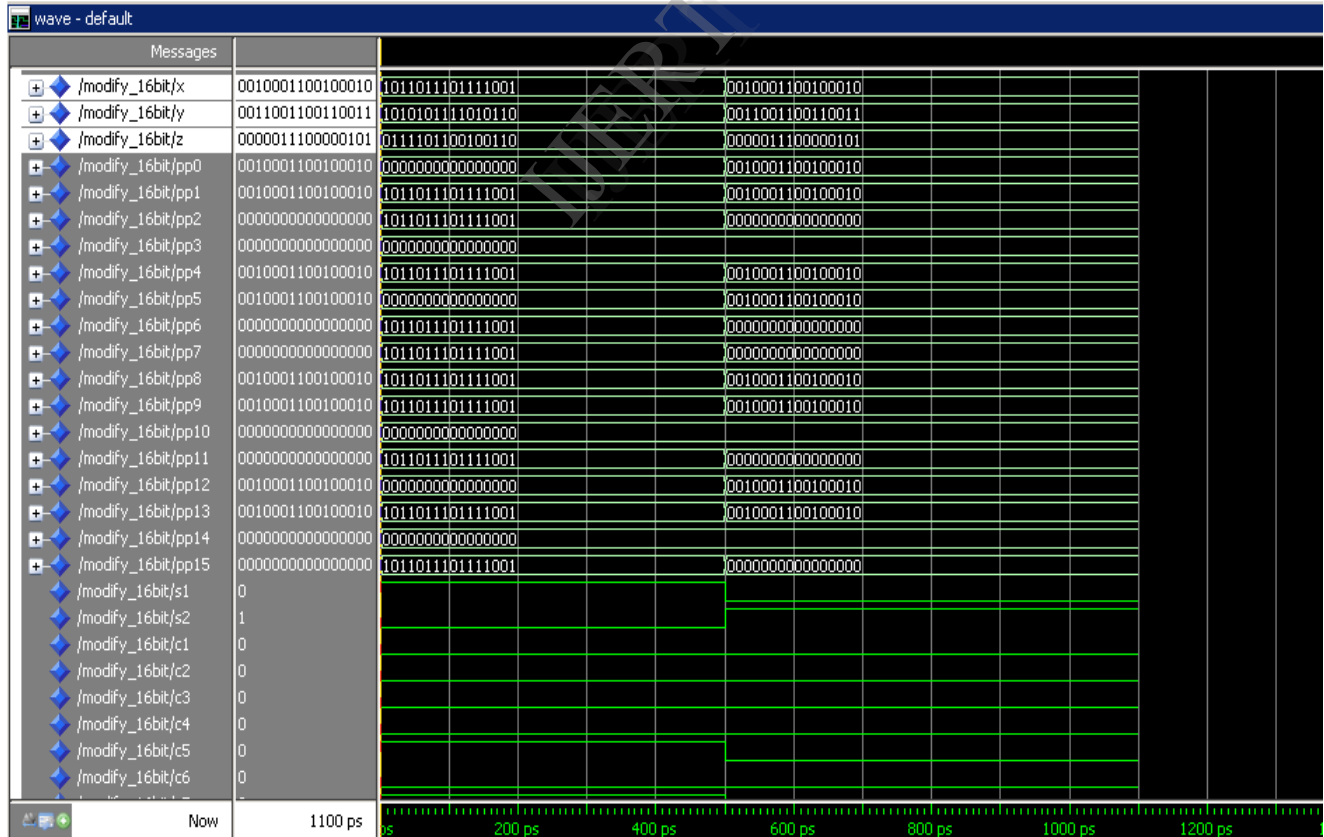


Fig.2. Simulated Result

V CONCLUSION

In this paper, a low-error and area-efficient fixed-width multiplier by using the dual group minor input correction vector is presented. As compared with the state-of-the-art design in [8], the proposed fixed-width multiplier improves accuracy, silicon area, timing performances and power dissipation. Simulation results for a 0.35- μm technology show a decrease of the propagation delay up to 20%, with more than 10% power dissipation reduction.

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 fixedwidth 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. Curticapean 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] Nakamura S. and Chu K. Y. (1988) 'A single chip parallel multiplier by MOSTechnology,' *IEEE Trans. Comput.*, vol. 37, pp. 274–282.
- [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.
- [11] S. R. Kuang, J. M. Jou, and Y. L. Chen, "The design of an adaptive on-line binary arithmetic coding chip," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, pp. 693–706, Jul. 1998.
- [12] F. Curticapean and J. Niittylahti, "A hardware efficient direct digital frequency synthesizer," in *Proc. IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS'01)*, vol. 1, St. Julians, Malta, Sep. 2–5, 2001, pp.51–54.
- [13] A. G. M. Strollo, E. Napoli, and D. De Caro, "Direct digital frequency synthesizers using first-order polynomial Chebyshev approximation," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC'02)*, Florence, Italy, Sep. 24–26, 2002, pp. 527–530.
- [14] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford, U.K.: Oxford Univ. Press, 1999.
- [15] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for signal processing," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 2, pp. 90–95, Feb. 1996.
- [16] Y. C. Lim, "Digital processing applications for Single-precision multiplier with reduced circuit complexity," *IEEE Trans. Comp.*, vol. 41, no. 10, pp. 1333–1336, Oct. 1992.
- [17] Stine, J.E., and O.M. Duverne, "Variations on Truncated Multiplication". In *Proceedings, Euromicro Symposium on Digital System Design*, pp. 112–119, Sep. 2003.
- [18] L. Van, S. Wang, and W. Feng, "Design of the lower error fixed-width multiplier and its application," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 10, pp. 1112–1118, Oct. 2000.
- [19] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comp.*, vol. 45, no.3, pp. 294–306, Mar. 1996.
- [20] K. C. Bickerstaff, E. E. Swartzlander, Jr., and M. J. Schulte, "Analysis of column compression multipliers," in *Proc. 15th IEEE Symp. Computer Arithmetic*, 2001, pp. 33–39.