[3] D. R. Frey, "Log-domain filtering: An approach to current-mode filtering," *Proc. Inst. Elect. Eng., pt. G,* vol. 140, pp. 406–416, Dec. 1993.
[4] ____, "A 3.3V electronically tunable active filter usable to beyond 1 GHz," in *Proc. ISCAS'94,* 1994, pp. 493–496.
[5] ____, "Exponential state space filters: A generic current mode design strategy," *IEEE Trans. Circuits Syst. I,* vol. 43, pp. 34–42, Jan. 1996.
[6] C. Toumazou, J. Ngarmnil, and T. S. Lande, "Micropower log-domain filter for electronic cochlea," *Electron. Lett.,* vol. 30, no. 22, pp. 1839–1841, Oct. 1994.
[7] E. Seevinck, "Companding current-mode integrator: A new circuit principle for continuous-time monolithic filters," *Electron. Lett.,* vol. 26, no. 24, pp. 2046–2047, Nov. 1990.
[8] D. R. Frey, "Current mode class AB second order filter," *Electron. Lett.,* vol. 30, no. 3, pp. 205–206, Feb. 1994.
[9] ____, "Log filtering using gyrators," *Electron. Lett.,* vol. 32, no. 1, pp. 26–28, Jan. 1996.
[10] D. R. Frey and L. Steigerwald, "An adaptive analog notch filter using log filtering," in *Proc. ISCAS'96,* 1996, pp. 297–300.
[11] F. Yang, C. Enz, and G. van Ruymbeke, "Design of low-power and low-voltage log-domain filters," in *Proc. ISCAS'96,* pp. 117–120.
[12] W. A. Serdijn, J. Mulder, A. C. van der Woerd, and A. H. M. van Roermund, "Design of wide-tunable translinear second-order oscillators," in *Proc. ISCAS'97,* pp. 829–832.
[13] A. Thanachayanont, A. Payne, and S. Pookaiyaudom, "A current-mode phase-locked loop using a log-domain oscillator," in *Proc. ISCAS'97,* pp. 277–280.
[14] A. Thanachayanont, S. Pookaiyaudom, and C. Toumazou, "State-space synthesis of log domain oscillators," *Electron. Lett.,* vol. 31, no. 21, pp. 1797–1799, Oct. 1995.
[15] S. Pookaiyaudom and J. Mahattanakul, "A 3.3 volt high-frequency capacitorless electronically-tunable log-domain oscillator," in *Proc. ISCAS'95,* pp. 829–832.
[16] L. P. L. van Dijk, A. C. van der Woerd, J. Mulder, and A. H. M. Roermund, "An ultra-low-power, low-voltage electronic audio delay line for use in hearing aids," *IEEE J. Solid-State Circuits,* vol. 33, pp. 291–294, Feb. 1998.
[17] R. Fox and M. Nagarajan, "A CMOS log-domain filter with multiple operating points," *IEEE Trans. Circuits Syst. II,* to be published.
[18] Y. Tsividis, "Externally linear, time-invariant systems and their application to companding signal processors," *IEEE Trans. Circuits Syst. II,* vol. 44, pp. 65–85, Feb. 1997.

Design of Low-Error Fixed-Width Multipliers for DSP Applications

Jer Min Jou, Shiann Rong Kuang, and Ren Der Chen

***Abstract*— In this brief, two designs of low-error fixed-width sign-magnitude parallel multipliers and two's-complement parallel multipliers for digital signal processing applications are presented. Given two n-bit inputs, the fixed-width multipliers generate n-bit (instead of $2n$-bit) products with low product error, but use only about half the area and less delay when compared with a standard parallel multiplier. In them, cost-effective carry-generating circuits are designed, respectively, to make the products generated more accurately and quickly. Applying the same approach, a low-error reduced-width multiplier with output bit-width between n and $2n$ has also been designed. Experimental results show that the proposed fixed-width and reduced-width multipliers have lower error than all other fixed-width multipliers and are still cost-effective. Due to these properties, they are very suitable for use in many multimedia and digital signal processing applications such as digital filtering, arithmetic coding, wavelet transformation, echo cancellation, etc.**

***Index Terms*—Fixed-width parallel multipliers, multiplying operation.**

I. Introduction

In general, the multiplier is one of the core components in multimedia and digital signal processing (DSP) chips because it dominates these chips' performance and area. To get a higher speed, parallel multipliers are always adopted at the expense of high area complexity. In the past, many parallel multiplication algorithms (architectures) (e.g., [1]–[3]) have been proposed to reduce the chip area and increase the speed of the multipliers. This brief proposes another approach to significantly reduce the chip area of the parallel multipliers without sacrificing performance. The approach is based on the fact that the multiplication operations used in many multimedia and DSP applications [4], [5] usually have the special fixed-width property. That is, their input data and output product have the same bit width.

One can obtain a fixed-width multiplier by directly omitting about half the adder cells of the conventional parallel multiplier but a significant error would be introduced in the resulting product, and this is undesirable for many fixed-width multimedia and DSP applications. Kidambi *et al.* [6] gave a simple method, i.e., added a constant bias to the retained cells to reduce the error of the fixed-width multiplier. However, its product error is still large. In this brief, we propose the low-error fixed-width multipliers for both sign-magnitude and two's-complement formats, respectively. In them, efficient carry-generating circuits, which feed the revising information to the carry inputs of the retained adder cells, are designed to significantly reduce the product error. In addition, the proposed fixed-width multipliers still uses only about half the area of and less delay than the standard parallel multiplier. Our design strategy for fixed-width multipliers has also been applied to design the reduced width multiplier, which is useful in some special applications such as wavelet transformation [7], [9]. Experimental results show that the proposed fixed-width and reduced-width multipliers have lower error than all other fixed-

Manuscript received January 29, 1998; revised March 30, 1999. This work was supported by the National Science Council, R.O.C., under Contract NSC-86-2221-E-006-022. This paper was recommended by Associate Editor N. Ranganathan.

The authors are with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7130(99)04883-1.

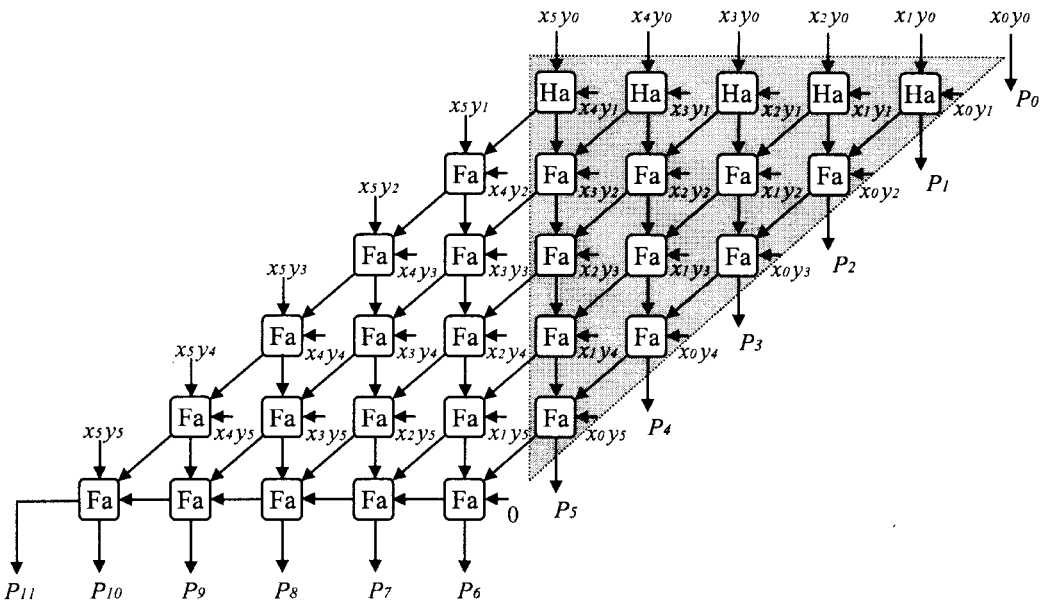

Fig. 1. The architecture of a standard 6×6 sign-magnitude parallel multiplier.

width multipliers and are still cost-effective. The design method and architectures of the proposed fixed-width multipliers will be introduced in Section II. In addition, this design strategy is extended to design the reduced-width multipliers. In Section III, the error and area comparisons of the proposed fixed-width multipliers with other fixed-width multipliers are reported.

II. Design of Fixed-Width Parallel Multipliers

This section first describes the method to design the low-error fixed-width sign-magnitude parallel multiplier and proposes the multiplier's architecture. Then, the similar ideas and approach are applied to designing the fixed-width two's-complement and reduced-width parallel multipliers.

A. Fixed-Width Sign-Magnitude Multiplier

Considering the multiplication of two n-bit inputs X and Y, a standard multiplier performs the following operations to obtain the $2n$-bit product P

$$P = XY = \sum_{i=0}^{2n-1} P_i 2^i = \left(\sum_{i=0}^{n-1} x_i 2^i\right)\left(\sum_{i=0}^{n-1} y_i 2^i\right) \tag{1}$$

where x_i, y_i, and P_i denote the ith bit of X, Y, and P, respectively. Fig. 1 shows the architecture of a standard 6 × 6 sign-magnitude parallel multiplier, where cells Ha and Fa are the half and full adders, respectively. The product in (1) can be expressed by the sum of two segments: the most-significant segment MP and the least-significant segment LP

$$P = MP + LP = \sum_{i=n}^{2n-1} P_i 2^i + \sum_{i=0}^{n-1} P_i 2^i. \tag{2}$$

Fig. 1 also shows the sections generating MP and LP for $n = 6$; in the shaded region of Fig. 1, the circuit producing LP is also denoted as LP, and the column (column circuit) generating P_i is also denoted as P_i. A fixed-width multiplier can be obtained directly by removing the shaded region and feeding each carry input of the residuary part with 0 to form a circuit denoted as MP', which is different from the circuit used to generate MP. This will save half the area of the circuit. However, a significant error will be introduced in the fixed-width product.

To design a low-error fixed-width multiplier, we first analyze the source of errors generated by MP', and then derive a small carry-generating circuit Cg to feed each carry input of MP' to reduce errors effectively. Let ε denote the difference between the two products produced by the standard multiplier and circuit MP', and it is caused by the carries generated from column circuit P_{n-1} (i.e., the column circuit generating P_5 in Fig. 1 for $n = 6$) in circuit LP. Let α_i denote the sum of carries generated from column circuit P_i, then $\varepsilon = \alpha_{n-1} * 2^n$. Since $0 \le \alpha_{n-1} \le n-1$, then $0 \le \varepsilon \le (n-1) * 2^n$. According to (1) and the multiplier architecture shown in Fig. 1, we have

$$\alpha_{n-1} = \left\lfloor \frac{1}{2}(x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_1 y_{n-2} + x_0 y_{n-1} + \alpha_{n-2}) \right\rfloor \tag{3}$$

where the notation $\lfloor \cdot \cdot \rfloor$ denotes the floor function. Recursively, α_{n-2} can be expanded into

$$\alpha_{n-2} = \left\lfloor \frac{1}{2}(x_{n-2}y_0 + x_{n-3}y_1 + \cdots + x_1 y_{n-3} + x_0 y_{n-2} + \alpha_{n-3}) \right\rfloor. \tag{4}$$

Continuing the expansion of α_i for $i = n-3, n-4, \cdots, 1$, then α_{n-1} can be expressed as

$$\begin{aligned} \alpha_{n-1} = \Bigg\lfloor & \frac{1}{2}(x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_1 y_{n-2} + x_0 y_{n-1}) \\ & + \frac{1}{4}(x_{n-2}y_0 + x_{n-3}y_1 + \cdots + x_1 y_{n-3} + x_0 y_{n-2}) \\ & + \frac{1}{8}(x_{n-3}y_0 + x_{n-4}y_1 + \cdots + x_1 y_{n-4} + x_0 y_{n-3}) \\ & + \cdots + \frac{1}{2^{n-1}}(x_1 y_0 + x_0 y_1) \Bigg\rfloor. \end{aligned} \tag{5}$$

Equation (5) shows that α_{n-1} is dominated by the bit-products $x_i y_j$, where $i + j = n - 1$, since they have the largest weight

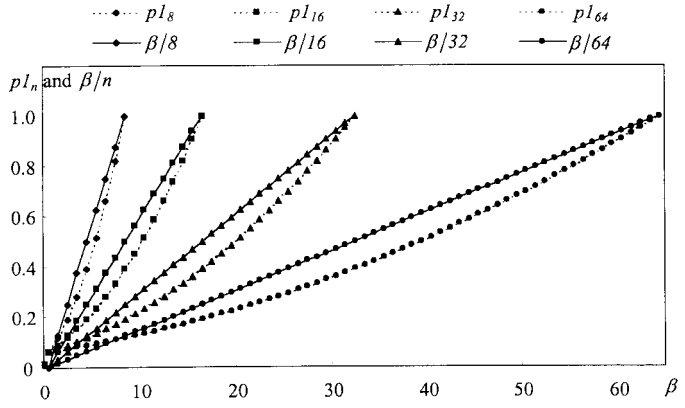

Fig. 2. The curves of $p1_n$ and β/n for $n = 8, 16, 32$, and 64.

1/2. To simplify the expression, we define $\beta = \sum_{i+j=n-1} x_i y_j = x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_1 y_{n-2} + x_0 y_{n-1}$, and rewrite (5) as

$$\alpha_{n-1} = \left\lfloor \frac{1}{2}(x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_0 y_{n-1}) + \frac{1}{4}x_{n-2}y_0 + \cdots + \frac{1}{2^{n-1}}x_0 y_1 \right\rfloor = \left\lfloor \frac{1}{2}\beta + \lambda \right\rfloor \quad (6)$$

where

$$\lambda = \frac{1}{4}x_{n-2}y_0 + \frac{1}{4}x_{n-3}y_1 + \cdots + \frac{1}{2^{n-1}}x_1 y_0 + \frac{1}{2^{n-1}}x_0 y_1.$$

A low-cost Cg can be designed easily if a simple relationship between α_{n-1} and β is found. The following Lemmas and Theorems first find the relationship between β and λ, and then that between α_{n-1} and β.

Let X and Y be the two n-bit input operands of a multiplier, then the bits of X and Y, denoted as x_i and y_i for $0 \le i \le n-1$, are called *input bits*. In the following derivation, we assumed that the probability of each input bit equal to 1 is 0.5, and the probability of bit-product $x_i y_j$ of the multiplier equal to 1 is denoted as $p1(x_i y_j)$.

Lemma 1: For a given β, then there are at most $n+\beta$ and at least 2β bits in X and Y equal to 1.

Proof: See the Appendix. □

Lemma 2: Given a β, the probability of any input bit of the multiplier equal to 1 is $(n + 3\beta)/4n$, and $p1(x_i y_j) = (n + 3\beta)^2/16n^2$.

Proof: See the Appendix. □

The curves of $p1(x_i y_j)$'s for different bit-width n's, denoted as $p1_n$, versus β are drawn in Fig. 2. We can see that $p1_n$ is approximately proportional to β. For simplification, a straight line is found to approximate the curve of $p1_n$ by using linear regression analysis. The straight line found is $\delta = \beta/n$, where δ is the value of the ordinate. The straight lines of β/n versus β for different n's are also drawn in Fig. 2 for comparison. In addition, the average relative errors between β/n and $(n + 3\beta)^2/16n^2$ for some given n's, denoted as $\overline{\phi}$, are listed in Table I. The results show that β/n is indeed approximate to $p1_n$. Therefore, we get

$$p1(x_i y_j) = (n + 3\beta)^2/16n^2 \cong \beta/n. \quad (7)$$

Theorem 1: For a given β, we have that

$$\lambda \cong \frac{\beta}{2} - \frac{\beta}{n}. \quad (8)$$

TABLE I
THE VALUES OF $\overline{\phi}$ FOR DIFFERENT BIT-WIDTH n's

n	8	16	32	64
$\overline{\phi}$	0.11304	0.12775	0.13525	0.13904

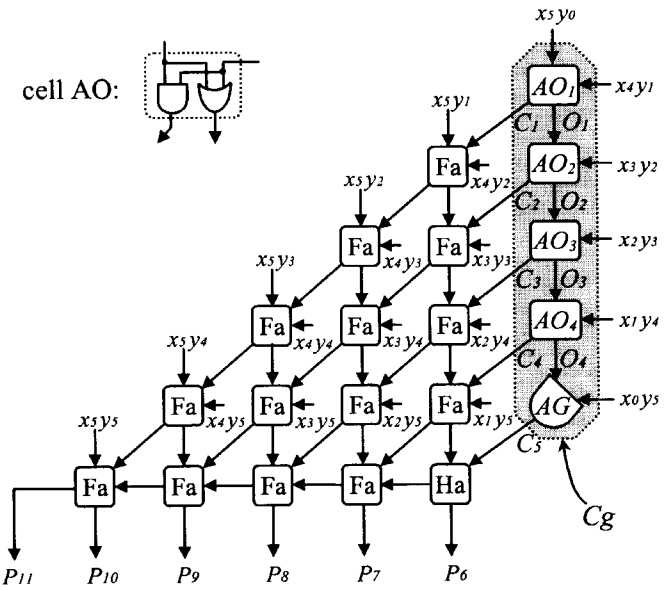

Fig. 3. The low-error fixed-width sign-magnitude multiplier for $n = 6$.

Proof: See the Appendix. □

By Theorem 1, a simple relationship between α_{n-1} and β is summarized as follows.

Theorem 2: Given a β, we have that

$$\alpha_{n-1} \cong \alpha' = \begin{cases} 0, & \text{if } \beta = 0 \\ \beta - 1, & \text{if } \beta > 0. \end{cases} \quad (9)$$

Proof: See the Appendix. □

Theorem 2 gives us a guideline on designing a carry-generating circuit Cg. The designed Cg has n inputs and $n-1$ outputs, and consists of $n-2$ AO cells and one 2-input AND gate. Each AO cell contains one 2-input AND gate and one 2-input OR gate (see Fig. 3). A 6-bit Cg is shown in the shaded area of Fig. 3. Feeding each input of the n-bit Cg with a proper bit-product $x_i y_j$ where $i + j = n - 1$, it will generate α' outputs equal to 1 and the other $n - \alpha' - 1$ outputs equal to 0. That is, it satisfies (9). The following theorem gives a proof.

Theorem 3: Feeding the n-bit Cg with proper bit-product $x_i y_j$'s for $i + j = n - 1$, it will generate α' outputs equal to 1 and the other outputs equal to 0.

Proof: See the Appendix. □

Feeding each carry input of circuit MP' with the outputs of the proposed circuit Cg properly, a new low-error fixed-width multiplier is formed. Fig. 3 shows an example of the low-error fixed-width sign-magnitude multiplier for $n = 6$. The speed of the proposed fixed-width multiplier is faster than that of the standard parallel multiplier, and the analyzes of area and product errors of it are given in Section III.

B. Fixed-Width Two's-Complement Multiplier

Considering the multiplication of two n-bit inputs X and Y, a standard two's-complement multiplier performs the following operations

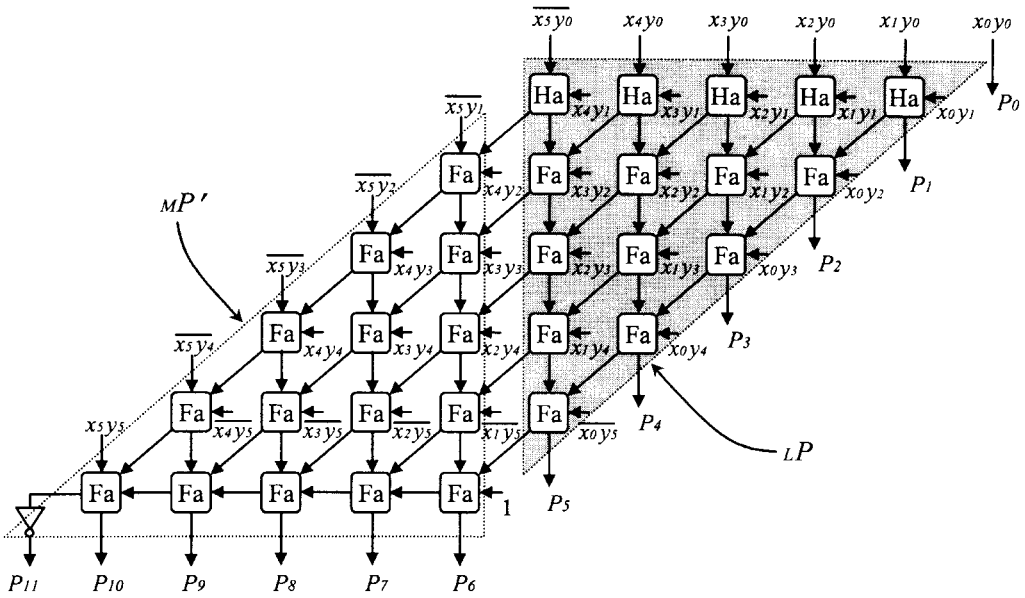

Fig. 4. The architecture of a standard 6×6 two's-complement parallel multiplier.

to obtain a $2n$-bit product P [2]:

$$\begin{aligned} P &= XY \\ &= \left(-x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i\right) \cdot \left(-y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i\right) \\ &= 2^{2n-1} + x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} x_i 2^i \cdot \sum_{i=0}^{n-2} y_i 2^i \\ &\quad + \sum_{i=0}^{n-2} (\overline{x_{n-1}y_i}) 2^{n-1+i} + \sum_{i=0}^{n-2} (\overline{y_{n-1}x_i}) 2^{n-1+i} + 2^{n-1}. \end{aligned} \tag{10}$$

Fig. 4 shows the architecture of a standard two's-complement parallel multiplier for $n = 6$.

To design a low-error fixed-width two's-complement parallel multiplier, we adopt an approach similar to that described in Section II-A. Let the carry-generating circuit of the two's-complement parallel multiplier be denoted as $\overline{Cg}$. According to (10) and the multiplier architecture shown in Fig. 4, the sum of carries generated from column circuit P_{n-1} in circuit LP, also denoted as $\alpha_{n-1,}$ is

$$\begin{aligned} \alpha_{n-1} &= \left\lfloor \frac{1}{2}(\overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1 y_{n-2} + \overline{x_0 y_{n-1}}) \right. \\ &\quad \left. + \alpha_{n-2} \right\rfloor \\ &= \left\lfloor \frac{1}{2}(\overline{x_{n-1}y_0} + \cdots + \overline{x_0 y_{n-1}}) \right. \\ &\quad \left. + \frac{1}{4}(x_{n-2}y_0 + \cdots + x_0 y_{n-2}) + \alpha_{n-3} \right\rfloor = \cdots. \end{aligned} \tag{11}$$

Equation (11) is then rewritten as

$$\alpha_{n-1} = \left\lfloor \frac{1}{2}\overline{x_{n-1}y_0} + \frac{1}{2}x_{n-2}y_1 + \cdots + \frac{1}{2}x_1 y_{n-2} + \frac{1}{2}\overline{x_0 y_{n-1}} + \lambda \right\rfloor \tag{12}$$

where

$$\lambda = \frac{1}{4}x_{n-2}y_0 + \frac{1}{4}x_{n-3}y_1 + \cdots + \frac{1}{2^{n-1}}x_1 y_0 + \frac{1}{2^{n-1}}x_0 y_1.$$

In (12), $\overline{x_{n-1}y_0} = 1 - x_{n-1}y_0$ and $\overline{x_0 y_{n-1}} = 1 - x_0 y_{n-1}$. Thus, (12) can be expressed as

$$\begin{aligned} \alpha_{n-1} &= \left\lfloor \frac{1}{2}(1 - x_{n-1}y_0) + \frac{1}{2}x_{n-2}y_1 + \cdots + \frac{1}{2}x_1 y_{n-2} \right. \\ &\quad \left. + \frac{1}{2}(1 - x_0 y_{n-1}) + \lambda \right\rfloor \\ &= \left\lfloor \frac{1}{2}(-x_{n-1}y_0 + x_{n-2}y_0 + \cdots + x_1 y_{n-2} - x_0 y_{n-1}) \right. \\ &\quad \left. + 1 + \lambda \right\rfloor. \end{aligned} \tag{13}$$

Moreover, Theorem 1 states that $\lambda \cong \beta/2 - \beta/n$. Therefore, (13) can be rewritten as

$$\begin{aligned} \alpha_{n-1} &\cong \left\lfloor \frac{1}{2}(-x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_1 y_{n-2} - x_0 y_{n-1}) \right. \\ &\quad \left. + 1 + \frac{\beta}{2} - \frac{\beta}{n} \right\rfloor \\ &= \left\lfloor \frac{1}{2}(-x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_1 y_{n-2} \right. \\ &\quad \left. + -x_0 y_{n-1} + \beta)1 - \frac{\beta}{n} \right\rfloor \\ &= \left\lfloor (x_{n-2}y_1 + \cdots + x_1 y_{n-2}) + 1 - \frac{\beta}{n} \right\rfloor \\ &= (x_{n-2}y_1 + \cdots + x_1 y_{n-2}) + 1 + \left\lfloor -\frac{\beta}{n} \right\rfloor. \end{aligned} \tag{14}$$

Since $\beta \le n$, we get that $\lfloor -\beta/n \rfloor = 0$ if $\beta = 0$ and $\lfloor -\beta/n \rfloor = -1$ if $\beta > 0$. Then, we get the following approximate relationship:

$$\alpha_{n-1} \cong \overline{\alpha}' = \begin{cases} \displaystyle\sum_{\substack{i+j=n-1 \\ i,\, j \neq n-1}} x_i y_i + 1, & \text{if } \beta = 0 \\ \displaystyle\sum_{\substack{i+j=n-1 \\ i,\, j \neq n-1}} x_i y_i, & \text{if } \beta > 0. \end{cases} \tag{15}$$

According to (15), an n-bit $\overline{Cg}$ that consists of $n-2$ OR cells, each contains a 2-input OR gate, and one 2-input NOR gate is designed. An example of a 6-bit $\overline{Cg}$ is given in the shaded area of Fig. 5. Moreover, Fig. 5 shows the low-error fixed-width two's-complement multiplier for $n = 6$.

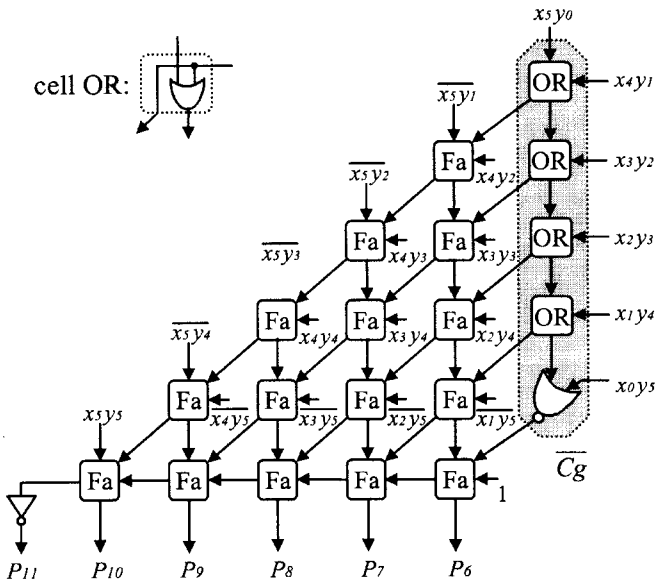

Fig. 5. The low-error fixed-width two's-complement multiplier for $n = 6$.

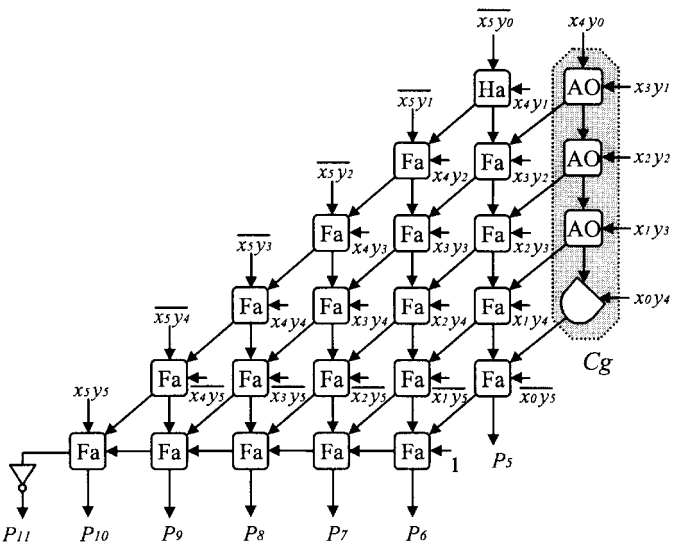

Fig. 6. A 12-to-7 reduced-width two's-complement multiplier.

C. Reduced-Width Multiplier

Given two n-bit inputs, a multiplier which generates an m-bit product where $n < m < 2n$ is called a $2n$-to-m reduced-width multiplier. It is useful in some special applications such as wavelet transform [9] and data coding in which the encoder generates m-bit data so that only the m most-significant bits of a $2n$-bit product are required. In addition, the reduced-width multiplier can further reduce the product error of the fixed-width multiplier described above.

Again, our design strategy for the design of a fixed-width multiplier is applied to the design of a low-error reduced-width multiplier. A $2n$-to-m reduced-width sign-magnitude multiplier can be obtained by eliminating the adder cells needed to produce the $2n - m$ least-significant bits (LSB's) of the product and then by adding a $2n - m$ bits carry-generating circuit Cg to feed revised data to the carry inputs of the retained adder cells. Since the input bit-products of column circuits P_{n-2}, P_{n-3}, $\cdots$, P_1 in the two's-complement and the sign-magnitude multipliers are identical (see Figs. 1 and 4 for $n = 6$), the design of a $2n$-to-m reduced-width two's-complement multiplier is the same as the design of a $2n$-to-m reduced-width sign-magnitude multiplier. An example of a 12-to-7 reduced-width two's-complement multiplier is shown in Fig. 6.

III. Error and Area Comparisons

We take the standard multiplier M_S, the proposed fixed-width multiplier M_F, the fixed-width multiplier MP' defined in Section II-A, the fixed-width multiplier M_1 proposed in [6], the fixed-width multiplier M_2 which consists of circuit MP' and column circuit P_{n-1} with each carry input fed with 0, and the proposed $2n$-to-$(n+1)$ reduced-width multiplier M_R for comparisons.

To compare the accuracy of the different fixed-width multipliers, we calculate and compare the maximal absolute error, the average error, and the relative error of their products. Let FP be the product of an n-bit fixed-width multiplier, then its maximal absolute error ε_M is the maximum value of $|MP - FP|$'s for all input pairs, and its average error $\overline{\varepsilon}$ is $(\sum|MP - FP|)/2^{2n}$, where $\sum|MP - FP|$ is the sum of $|MP - FP|$'s for all input pairs. In addition, the relative error ε_R of a fixed-width multiplier is defined as $|MP - FP|/MP$. Let $P\varepsilon_R$ denote the percentage of input pairs whose ε_R is larger than 0.01, as shown at the bottom of the page. Then, the ratio of $P\varepsilon_R$'s of the other fixed-width multipliers to that of our fixed-width multiplier M_F is denoted as $\Re(0.01)$. The larger $\Re(0.01)$ is, the less accurate the multiplier is. The comparison results of ε_M, $\overline{\varepsilon}$, and $\Re(0.01)$ for different fixed-width sign-magnitude and two's-complement multipliers are listed in Table II. The results show that M_F is more accurate than other fixed-width multipliers. Moreover, the errors of M_R are also listed in Table II. M_R offers another choice for obtaining a higher accuracy.

To make the area comparison, let A_N, A_A, A_H, and A_F be the areas of a NAND (NOR) gate, an AND (OR) gate, a half adder, and a full adder, respectively, and we assume that the multipliers are composed of these modules. Using the standard cells of 0.6-μm 1P3M technology [8], we have $A_H \cong 0.45A_F$, $A_A \cong 0.08A_F$, and $A_N \cong 0.06A_F$. The comparison results of the ratio of areas between the fixed-width and standard multipliers are listed in Table III. The area of the proposed fixed-width multiplier is nearly half the area of a standard multiplier and is less than the area of M_2. In addition, the area of M_R is also listed in Table III. The results show that the area of M_R is also competitive with the standard parallel multiplier.

For a more practical error comparison, the different multipliers are applied to wavelet transformation [7], [9] and then the quality of reconstructed images is compared. Four 512 $\times$ 512 images of Lena, F16, Face, and Bear are picked for this experiment, and quality comparison among different multipliers is done based on root mean square error (RMSE), signal-to-noise ratio (SNR), and peak signal-to-noise ratio (PSNR). The smaller RMSE and the larger SNR and PSNR represent the better quality of the reconstructed images. One resolution level wavelet operations performed by different multipliers and the quality comparison are reported in Table IV. In Table IV, M_{R1} and M_{R2} denote the 18-to-10 and 18-to-11 reduced-width multipliers, respectively. The results show that the proposed fixed-width multiplier M_F obtains better quality than other fixed-width multipliers, and the quality can be further improved by using M_{R1} and M_{R2}.

$$P\varepsilon_R = \frac{\text{the number of } \varepsilon_R\text{'s which are larger than 0.01 for each input pair}}{\text{the number of all input pairs}} * 100\%$$

TABLE II
THE COMPARISON RESULTS OF ERRORS FOR DIFFERENT FIXED-WIDTH MULTIPLIERS

multipliers		errors	n=4	n=8	n=12	n=16
sign-magnitude	$_MP'$	ε_M	48	1792	45056	983040
	M_1		48	1536	36864	786432
	M_2		16	768	20480	458752
	M_F		16	512	8192	196608
	M_R		16	256	4096	131072
	$_MP'$	$\bar{\varepsilon}$	5.750	322.75	9219.75	212996.8
	M_1		5.750	190.88	3959.51	74892.3
	M_2		1.500	130.50	4099.50	98308.5
	M_F		0.938	65.04	1570.32	30403.7
	M_R		0.125	26.76	731.39	14629.6
	$_MP'$	$\Re(0.01)$	5.2	3.3	3.4	3.3
	M_1		5.2	2.4	1.8	2.3
	M_2		1.6	1.7	1.6	1.7
	M_F		1	1	1	1
	M_R		0.1	0.4	0.4	0.5
two's complement	$_MP'$	ε_M	48	1792	45056	983040
	M_1		32	1280	32768	720896
	M_2		16	768	20480	458752
	M_F		16	512	8192	196608
	M_R		16	256	4096	131072
	$_MP'$	$\bar{\varepsilon}$	13.750	450.75	11267.75	245764.7
	M_1		5.375	187.89	3927.92	74497.4
	M_2		1.500	130.50	4099.50	98308.5
	M_F		0.938	65.04	1570.32	30403.7
	M_R		0.125	26.76	731.39	14629.6
	$_MP'$	$\Re(0.01)$	12.9	3.7	5.1	7.1
	M_1		5.6	2.5	2.3	2.8
	M_2		1.6	1.9	2.5	3.2
	M_F		1	1	1	1
	M_R		0.1	0.4	0.5	0.5

TABLE III
AREA RATIO FOR DIFFERENT FIXED-WIDTH MULTIPLIERS

multipliers		n=8	n=16	n=32	n=64
sign-magnitude	$_MP'$	0.4523	0.4790	0.4901	0.4952
	M_1	0.4599	0.4807	0.4905	0.4953
	M_2	0.5837	0.5430	0.5218	0.5109
	M_F	0.5507	0.5274	0.5141	0.5072
	M_R	0.6698	0.5887	0.5451	0.5227
two's complement	$_MP'$	0.4573	0.4791	0.4897	0.4949
	M_1	0.4670	0.4813	0.4902	0.4950
	M_2	0.5906	0.5437	0.5215	0.5106
	M_F	0.5485	0.5231	0.5113	0.5056
	M_R	0.6773	0.5895	0.5448	0.5224

TABLE IV
THE QUALITY COMPARISON OF ONE-LEVEL WAVELET TRANSFORM USING DIFFERENT MULTIPLIERS

image	error	multiplier					
		$_MP'$	M_1	M_F	M_{R1}	M_{R2}	M_S
Lena	RMSE	93.09	65.31	26.75	8.80	1.39	0.91
	SNR	1.63	4.71	12.46	22.12	38.13	41.79
	PSNR	8.75	11.83	19.59	29.24	45.25	48.91
F16	RMSE	80.07	39.84	18.06	12.90	1.83	0.91
	SNR	7.29	13.35	20.22	23.14	40.11	46.18
	PSNR	10.06	16.12	23.00	25.92	42.89	48.96
Face	RMSE	98.67	49.13	32.30	4.80	1.11	0.90
	SNR	-0.50	5.56	9.20	25.75	38.49	40.30
	PSNR	8.25	14.34	17.95	34.50	47.24	49.50
Bear	RMSE	99.14	69.78	40.07	20.87	2.09	1.05
	SNR	2.11	5.17	9.98	15.65	35.62	41.64
	PSNR	8.21	11.26	16.08	21.74	41.71	47.74

IV. CONCLUSION

The design of low-error fixed-width sign-magnitude and two's-complement multipliers have been presented. By using this type of multiplier, the chip area can be significantly reduced and a little performance promotion is also introduced. In addition, the design strategy has also been applied to designing a reduced-width multiplier, which has lower product error than that of a fixed-width multiplier and still maintains low area complexity. They are useful in fixed-width data path architectures for multimedia and DSP applications where a uniform or reduced word width is usually required.

APPENDIX

Proof of Lemma 1: If $x_iy_j = 1$, then both input bits x_i and y_j are 1. If $x_iy_j = 0$, then only one of x_i and y_j is 1, or both are 0. Moreover, if there are β bit-product x_iy_j's for $i+j = n-1$ equal to 1, then there are $n-\beta$ bit-product x_iy_j's for $i+j = n-1$ equal to 0. Therefore, at most $2\beta + (n-\beta) = n+\beta$ bits and at least 2β bits in input bits are 1. □

Proof of Lemma 2: By Lemma 1, $[(n+\beta)+2\beta]/2 = (n+3\beta)/2$ bits of $2n$ input bits are 1 on average. Therefore, the probability of any input bit equal to 1 is $[(n+3\beta)/2]/2n = (n+3\beta)/4n$. Moreover, the probability of the bit-product x_iy_j equal to 1 is $[(n+3\beta)/4n] * [(n+3\beta)/4n] = (n+3\beta)^2/16n^2$. □

Proof of Theorem 1: The number of bit-product terms in λ whose coefficients are equal to $1/2^m$ is $n-m+1$, and (7) states that $p1(x_iy_j) \cong \beta/n$. Thus

$$\begin{aligned}\lambda &= \frac{1}{4}x_{n-2}y_0 + \frac{1}{4}x_{n-3}y_1 + \cdots + \frac{1}{2^{n-1}}x_1y_0 + \frac{1}{2^{n-1}}x_0y_1\\ &= \frac{1}{4}p1(x_{n-2}y_0) + \frac{1}{4}p1(x_{n-3}y_1) + \cdots + \frac{1}{2^{n-1}}p1(x_1y_0)\\ &\quad + \frac{1}{2^{n-1}}p1(x_0y_1)\\ &= p1(x_iy_j)\times\left(\frac{1}{4}+\frac{1}{4}+\cdots+\frac{1}{2^{n-1}}+\frac{1}{2^{n-1}}\right)\\ &\cong \frac{\beta}{n}\times\left(\frac{n-2+1}{2^2}+\frac{n-3+1}{2^3}+\frac{n-4+1}{2^4}\right.\\ &\qquad \left.+\cdots+\frac{n-(n-1)+1}{2^{n-1}}\right)\\ &= \frac{\beta}{n}\times\left(\frac{n-1}{4}+\frac{n-2}{8}+\frac{n-3}{16}+\cdots+\frac{2}{2^{n-1}}\right)\\ &= \frac{\beta}{n}\times\left(\frac{n}{2}-1\right) = \frac{\beta}{2}-\frac{\beta}{n}.\end{aligned}$$

□

Proof of Theorem 2: By Theorem 1, (6) can be reexpressed as

$$\begin{aligned}\alpha_{n-1} &= \left\lfloor\frac{\beta}{2}+\lambda\right\rfloor \cong \left\lfloor\frac{\beta}{2}+\frac{\beta}{2}-\frac{\beta}{n}\right\rfloor = \left\lfloor\frac{\beta-\beta}{n}\right\rfloor\\ &= \beta + \left\lfloor-\frac{\beta}{n}\right\rfloor.\end{aligned}$$

Since $\beta \le n$, we get that $\lfloor-\beta/n\rfloor = 0$, if $\beta = 0$ and $\lfloor-\beta/n\rfloor = -1$, if $\beta > 0$. Thus, we get the following approximate relationship:

$$\alpha_{n-1} \cong \alpha' = \begin{cases}0, & \text{if } \beta = 0\\ \beta-1, & \text{if } \beta > 0.\end{cases}$$

□

Proof of Theorem 3: Referring to Fig. 3, an n-bit Cg consists of $n-2$ AO cells: $AO_1\cdots AO_{n-2}$, and one 2-input AND gate AG. C_i is one output of AO cells and gate AG. Let $\wedge$ and $\vee$ denote logic AND and OR, respectively. The Boolean equation of C_i, $1 \le i \le n-1$, is

$$\begin{aligned}C_i &= O_{i-1}\wedge x_{n-i-1}y_i\\ &= (x_{n-1}y_0 \vee x_{n-2}y_1 \vee \cdots \vee x_{n-i}y_{i-1})\wedge x_{n-i-1}y_i. \quad (16)\end{aligned}$$

The inputs of Cg must satisfy the one of the following two cases.

Case 1—All Inputs are 0: By (16) $C_i = 0$, for $1 \le i \le n-1$. That is, all outputs are 0, thus (9) is satisfied.

Case 2—At Least One Input is not 0: Assume that the first one in the input terms $x_{n-1}y_0,\ x_{n-2}y_1,\ \cdots,\ x_0y_{n-1}$ which is not equal to 0 is $x_{n-j-1}y_j$. That is $x_{n-1}y_0 = x_{n-2}y_1 = \cdots = x_{n-j}y_{j-1} = 0$ and $x_{n-j-1}y_j = 1$. Then, by (16), we have $C_i = 0$, for $1 \le i \le j$ and $C_i = x_{n-j-2}y_{j+1}$, for $j+1 \le i \le n-1$. That is,

$$\sum_{i=1}^{n-1} C_i = x_{n-j-2}y_{j+1} + \cdots + x_0y_{n-1} = \sum_{k=0}^{n-1} x_{n-k-1}y_k - \sum_{k=0}^{j} x_{n-k-1}y_k = \beta - 1.$$

Thus, (9) is also satisfied. □

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers, whose comments were useful in improving the manuscript.

REFERENCES

[1] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 14–17, 1964.
[2] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. Comput.*, vol. C-22, pp. 1045–1047, Dec. 1973.
[3] L. Dadda, "On parallel digital multipliers," *Alta Frequenz.*, vol. 45, pp. 574–580, 1976.
[4] 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*, vol. 45, pp. 693–706, July 1998.
[5] T. J. Terrell, *Introduction to Digital Filters.* New York: Wiley, 1988.
[6] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 90–94, 1996.
[7] G. Strang and T. Nguyen, *Wavelets and Filter Banks.* Wellesley, MA: Wellesley-Cambridge, 1996.
[8] 0.6 μm 1P3M technology manual, Chip Implementation Center, National Science Council, R.O.C., 1996.
[9] S.-L. Kuang, "On the design of data compression and color correction ASIC's with some new high level synthesis approaches," Ph.D. dissertation, Dept. of Electron. Eng., National Cheng Kung Univ., Taiwan, R.O.C., June 1998.

The Impact of Induced Gate Noise When Simultaneously Power and Conjugate Noise-Matching MOS Transistors

Tajinder Manku

***Abstract*—A method for simultaneously power matching and conjugate noise matching a MOS transistor for radio frequency applications is presented in this paper. Experimental results from a 0.6 μm nMOS transistor show that the magnitude of input reactance equals the optimum noise reactance (i.e., $X_{\rm in} = X_{\rm opt}$). Using this result, we provide a method that can be used to match the optimum noise resistance to the source resistance. The described method uses the number of gate fingers as a parameter to conjugately match for noise.**

I. INTRODUCTION

Radio frequency electronics is increasingly taking advantage of the advances in CMOS technology. These advances include the development of a 0.1-μm CMOS technology [1]. This particular CMOS technology has demonstrated active devices with f_t's (i.e., the unity current gain cutoff frequency) exceeding 100 GHz and minimum noise figures less than 0.5 dB at 2 GHz [1]. The more commercially available 0.5-μm CMOS technologies have demonstrated f_t's of 20 GHz and minimum noise figures of 1.6 dB at 2 GHz [2]. There have also been several reports within the literature of front-end RF designs (i.e., low noise amplifiers (LNA's) and mixers) within a CMOS environment (e.g., [3]–[7]). A majority of the LNA designs have demonstrated a relatively high noise figure in comparison to the expected minimum [3]–[6]. We attribute this to the fact that the input MOS device was not conjugately matched for minimum noise. For the LNA design in [7], the minimum noise figure was achieved by using exterior impedance tuners to match for both noise and power. Though this method was effective in matching the device for both noise and power, it would have a relatively low level of integration; i.e., it requires many off-chip components. In [5], a mathematical description of matching for both noise and power for a MOSFET was provided. Here it was shown that if one includes the effects of induced gate noise, conjugate noise matching is not straightforward because the optimum noise reactance does not equal the input reactance of the device. In this paper, we further examine how induced gate noise influences conjugate noise matching. Furthermore, we will outline a simple method that can be used to simultaneously match for both power and noise for a MOS transistor.

II. INDUCED GATE NOISE

Induced gate noise is assumed to be one of the main components of noise within the intrinsic portion of a MOS transistor. The other noise sources include the drain channel noise and the thermal noise due to the resistance of the gate material; Fig. 1(a) illustrates the physical location of these noise sources with respect to an approximate lumped network model of a MOS transistor. The thermal noise resistance due to the resistance of the gate material has been shown to be equal to the total resistance of the gate length divided by three. In the network

Manuscript received June 1, 1998; revised February 10, 1999. This work was supported by the Natural Sciences and Engineering Research Council of Canada and by Mitel Semiconductor. This paper was recommended by Associate Editor P. H. Young.

The author is with RF Technology Group, Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Publisher Item Identifier S 1057-7130(99)04882-X.