Low-cost Two's Complement Multipliers Using Signed Binary Digits for High-speed Digital Systems

Shiann-Rong Kuang, Shao-Hean Hsu, Shu-You Liu, and Kuo-Chin Huang

Department of Computer Science Engineering
National Sun Yat-Sen University

srkuang@cse.nsysu.edu.tw

Abstract

Multiplication is the most important operation in many high-speed digital systems. Redundant binary number system has been used to design fast multipliers, but whose area is probably larger than other kind of multipliers. In this paper, area-efficient two's complement multipliers using binary signed-digit number system are designed for digital systems with constant data size by truncating the 2*n*-bit product into *n*-bit. Based on the variable correction value scheme, a novel carry compensation formulation and corresponding circuit are developed to largely degrade the product error. Simulation results show that the proposed truncated multipliers are more accurate than other truncated architectures while maintaining high speed and small area. When applying to discrete cosine transform (DCT), the proposed multiplier can significantly reduce the area and power of DCT circuit and still obtain good image quality.

Keywords: binary signed-digit number system, two's complement multiplier, discrete cosine transform

1. Introduction

Numerous multiplication schemes have been introduced to enhance the performance of multipliers. An efficient method to design a fast multiplier is to represent the partial products as redundant binary (RB) numbers and accumulate them by a RB adder tree. The RB multiplier not only improves speed because it requires no continuous carry propagation, but also simplifies the interconnection. The literature [1] has reported that the RB multiplier is more suitable for VLSI design due to its regular layout and results in high-speed circuit implementations.

Although the RB multiplier is very fast, its area is probably large due to the redundant binary number representation and number system conversion. Fortunately, the property of constant data size appeared in many real applications, which requires that the 2*n*-bit product of multiplication operation is truncated into *n* bits, can be applied to significantly reduce the area and power of RB multiplier. The simplest method to obtain a truncated multiplier is directly eliminating about half the adder cells of the standard multiplier, but a large product error would be introduced. Many papers [3]-[8] have proposed efficient methods and circuits to reduce the product error. However, most of them design the low-error truncated multiplier from the Baugh-Wooley multiplier, and none pays attention to the redundant binary signed-digit (RBSD) multiplier [1, 2]. This paper focuses on the design of low-cost and low-error truncated RBSD multiplier to reduce the area and power of multiplier.

In the past, fixed constant [3]-[5] or variable correction value [6]-[8] is used to reduce the product error of truncated multiplier. The former adds a fixed constant obtained based on statistic average to the remaining adder cells of the truncated multiplier. The latter adds an input-data dependent correction value to the remaining adder cells so that it usually works better than the former. Therefore, we develop low-cost truncated RBSD multipliers based on the scheme of variable correction value. Simulation results show that the proposed truncated RBSD multipliers have lower product error than other architectures while maintaining high speed and small area.

The remainder of this paper is organized as follows. Section 2 briefly introduces the RBSD number representation and multiplier. The product error correction for truncated RBSD

multipliers is described in Section 3. Section 4 provides error comparisons with previous designs and some applications. Finally, the conclusion is given.

2. RBSD Number and Multiplier

The redundant binary signed-digit number system uses the digit set $\{\bar{1}, 0, 1\}$ to represent numbers, where $\bar{1}$ denotes the digits value –1. Each digit in the RBSD representation can be encoded by using two bits if the positive-and-negative encoding is employed. The value of each digit is calculated by $x_i = (x_i^+, x_i^-) = (x_i^+) - (x_i^-)$, where (x_i^+, x_i^-) is one of the four forms (0, 0), (0, 1), (1, 0), and (1, 1), whose value is 0, $\bar{1}$, 1, and 0, respectively.

Fig. 1 shows the block diagram of an $n \times n$-bit RBSD multiplier for n=8. The multiplier essentially consists of RBSD Booth's encoders, an RB adder tree, and an RBSD-to-NB converter. Given two n-bit binary numbers X and Y in two's complement form, the RBSD Booth-2 (radix-4) encoders [2] generate multiples and $\lceil n/2 \rceil$ rows of RBSD partial products PP_i, where $\lceil x \rceil$ denotes the smallest integer that is larger than or equal to the real number x. The RBSD Booth's encoder uses the same encoding table as the modified Booth's encoding to generate RBSD partial products without any additional time delay and with almost no extra hardware. Then the RBSD partial products are added up by using the redundant binary adders (RBA) tree. The array of an RBA tree can increase operating speed by use of high speed RBA. For example, the RBA presented in [1] (shown in Fig. 2) is optimized for speed and area efficiency by employing transmission gates. In [1], both the inputs (a_i^+, a_i^-) and (b_i^+, b_i^-) of a RBA cell are assumed to take one of the three states (0, 1), (0, 0), (1, 0), and no (1, 1) to simplify the consideration. The (b_i, h_i) denotes the carry and h_i is defined to prevent the continuous carry propagation by eliminating the collision of the sum and the carry from the lower digit. The final RBSD product R must be converted to a normal binary (NB) product N by an RBSD-to-NB converter [1].

3. Design of Truncated RBSD Multiplier

In a $n \times n$ standard RBSD multiplier, $\lceil n/2 \rceil$ rows of RBSD partial products are added up to generate the final RB product $R[2n-1]$ to $R[0]$. Fig. 3 shows the case of n=8. The $2n$-bit RB product can be truncated to n-bit by eliminating the n least significant columns (column 0 to column n-1) to form a truncated RBSD multiplier TRM_H. In TRM_H, the complexity of RBSD Booth's encoders, RBA tree, and RBSD-to-NB converter is reduced by almost half, but large error is introduced into the product.

Let s_{n-1} denote the sum of carries from the column $n-1$, a good estimation of s_{n-1} can be used as a correction value to degrade the product error of TRM_H. By Fig. 3, we have

$$\begin{aligned} s_{n-1} &= \lfloor 2^{-1}(p_{0,n-1} + p_{1,n-3} + \ldots + p_{\lceil n/2 \rceil - 1,1}) \\ &\quad + 2^{-2}(p_{0,n-2} + \ldots + p_{\lceil n/2 \rceil - 1,0}) \\ &\quad + \ldots + 2^{-(n-1)} p_{0,1} + 2^{-n} p_{0,0} \rfloor \\ &= \lfloor 2^{-1}(q + s_{n-2}) \rfloor. \end{aligned} \tag{1}$$

where q is the sum of partial products in column $n-1$, and $\lfloor x \rfloor$ denotes the integer part of the real number x. When round-off is considered, the sum of carries from the column n-1, denoted as d_{n-1}, becomes

$$\begin{aligned} d_{n-1} &= s_{n-1} + (q + s_{n-2}) \bmod 2 \\ &= \lfloor 2^{-1}(q + s_{n-2}) \rfloor + (q + s_{n-2}) \bmod 2. \end{aligned} \tag{2}$$

The following goal is to find a good estimation of d_{n-1} to obtain a compensation value for reducing the product error of TRM_H.

Using the fixed constant correction scheme to reduce the product error of TRM_H doesn't work well since the partial products of RBSD multiplier may be 0, 1, or $\bar{1}$ so that the average value of d_{n-1} always approximates to zero. It means that this scheme will use constant 0 as the compensation value and no improvement of the product error can be achieved. Therefore, we adopt the variable correction value scheme and find two possible adaptive compensation formulations, and then the better one is selected by simulation results. The first candidate is q. q has been used in other kind of truncated multipliers as the compensation value. For example, the truncated array multiplier in [6] and the truncated Booth's multiplier in [8] use q to degrade the product error. Therefore, q is a possible approximation of d_{n-1}. The second candidate, denoted as l, is derived from Eq. (2). As mentioned above, the partial products of RBSD multiplier may be 0, 1, or $\bar{1}$ so that the average value of s_{n-2} also

approximates to zero. Replacing s_{n-2} by 0, Eq. (2) is rewritten as

$$d_{n-1} \cong l = \lfloor 2^{-1} \cdot q \rfloor + (q \bmod 2). \quad (3)$$

We apply all possible input combinations to standard RBSD multiplier to inspect that q or l is more approximate to d_{n-1}. Let $a_1 = d_{n-1} - q$ and $a_2 = d_{n-1} - l$, the probability distribution of a_1 and a_2 for the cases of n from 8 to 14 are shown in Table 1 and Table 2, respectively. It is obvious that d_{n-1} is equals to l (i.e. a_2 =0) for most input combinations. Thus, l is the better choice.

The subsequent challenge is to design a fast and simple circuit to perform Eq. (3). Since the partial products are represented as RBSD numbers, it is very difficult to design a general circuit that completely match the behavior of Eq. (3). In our design, the compensation circuit consists of three kinds of cells *RHA1*, *RHA2*, and *RHA3* as shown in Fig. 4. Taking the partial products of column n-1 as inputs, these cells generate the compensation value of Eq. (3) in the similar form of adder tree. In the compensation tree, *RHA1* and *RHA3* are used in the first (top) level and last (bottom) level, respectively. *RHA2* is used in other levels between the first and last levels. The carries generated by these cells then are applied to the RB full adders of TRM_H to from a low-error truncated multiplier TRM_C as shown in Fig. 5(a) and Fig. 5(b) for n=8 and 12.

Table 3 shows the transistor ratio of truncated RBSD multiplier TRM_H and TRM_C versus the standard RBSD multiplier M_S. Comparing with M_S, the proposed truncated multiplier TRM_C saves about 32% area. Moreover, TRM_C needs about 8% area overhead but has very low product error than TRM_H.

4. Experimental Results

To appreciate the accuracy of the proposed truncated multiplier TRM_C, we take the K-G-As' structure ($M_{K\text{-}G\text{-}A}$) [5], the J-K-Cs' structure ($M_{J\text{-}K\text{-}C}$) [7], the multiplier M_{Booth} proposed in [8], TRM_H, and TRM_S (truncate the n LSBs of 2n-bit product of M_S to obtain its n-bit product) for comparison. Let e, e_M, $\bar{e}$, and u denote the absolute error, the maximal absolute error, the average error, and the variance of error, respectively. That is,

$$e \equiv |M_S - F_P|, \quad (4)$$

$$\bar{e} \equiv E\{e\}, \quad (5)$$

$$u \equiv E\{(e - \bar{e})^2\}, \quad (6)$$

where F_P represents the output value for different truncated multiplier, respectively, and $E\{\bullet\}$ is the expectation operator. The comparison results of e, e_M and u for different truncated multipliers are shown in Table 4 to Table 6. The results show that TRM_C is more accurate than other truncated multipliers.

The proposed multiplier is applied to the design of a discrete cosine transform (DCT) circuit for image processing. We use the different 11×11-to-15 truncated RBSD multipliers to test the quality of reconstructed images. Four 256×256 images are picked for this experiment, and quality comparison among different multipliers is based on PSNR and RMSE. The larger PSNR and smaller RMSE represent the better quality of the reconstructed images. The quality comparison reported in Table 7 shows that the proposed truncated multipliers can obtain very good image quality.

5. Conclusion

This paper has proposed low-cost and low-error RBSD multipliers to save hardware area and power dissipation. The correction value for product error was dependent upon input data and has been verified by simulation. Experimental results shown that the product error of the proposed RBSD truncated multipliers was lower than that of other truncated multipliers.

Acknowledgment

This work was supported in part by the National Science Council, R.O.C., under Grant NSC-91-2215-E-110-027.

References

[1] H. Makino, Y. Nakase, H. Suzuki, H. Moronika, H. Shinohara, and K. Mashiko, "An 8.8-ns 54×54-Bit Multiplier with High Speed Redundant Binary Architecture," *IEEE Joural of Solid-State Circuit*s, vol. 31, no. 6, pp.773-783, June 1996.

[2] N. Besli and R. G. Deshmukh, "A novel redundant binary signed-digit (RBSD) Booth's encoding," *IEEE SoutheastCon*, pp. 426-431, 2002.

[3] Y. C. Lim, "Single-recision multiplier with

reduced circuit complexity for signal processing applications," *IEEE Trans. on Computers*, vol. 41, no. 10, pp. 1333-1336, 1992.

[4] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," *Workshop on VLSI Signal Processing*, VI, pp. 388-396, 1993.

[5] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. on Circuits & Systems II*, vol. 43, no.2, pp. 90-95, Feb. 1996.

[6] E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," *31st Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1178-1182, 1997.

[7] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixed-width multipliers for DSP applications," *IEEE Trans. on Circuits & Systems II*, vol. 46, no. 6, pp. 836-842, June 1999.

[8] S. J. Jou and H. H. Wang, "Fixed-Width Multiplier for DSP Application," *IEEE International Conference on Computer Design*, pp. 318-322, 2000.

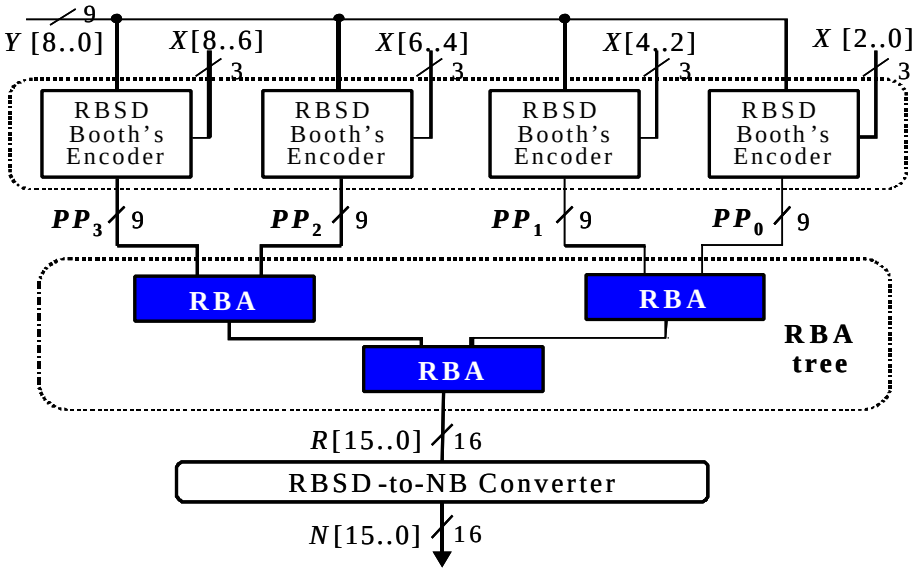

Fig. 1. The block diagram of an 8×8-bit RBSD multiplier

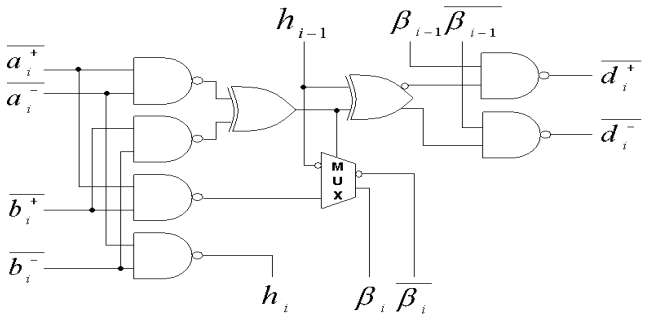

Fig. 2. Redundant Binary Adder schematic diagram

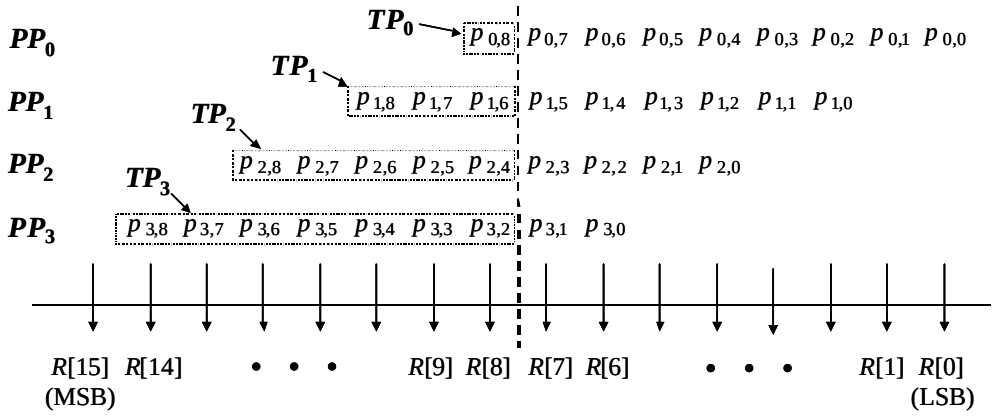

Fig. 3. Partial products of an 8×8-bit RBSD multiplier

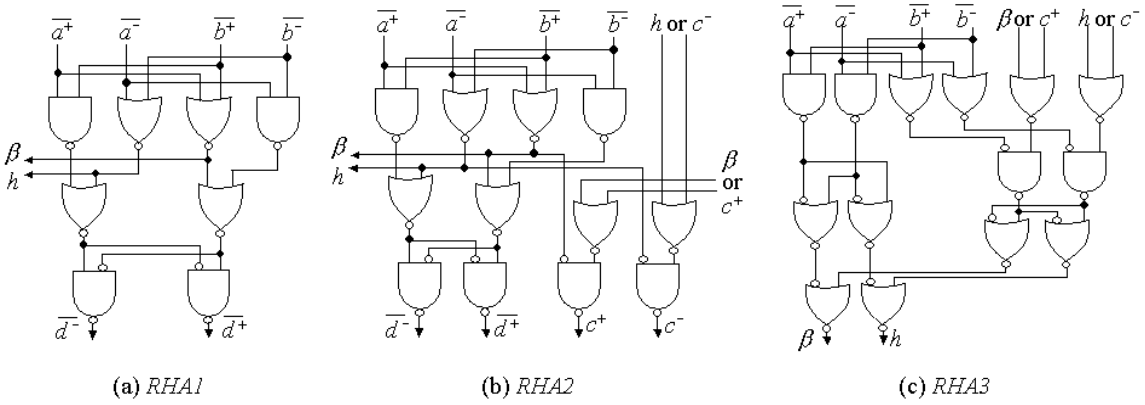

Fig. 4. The cells for generating compensation value

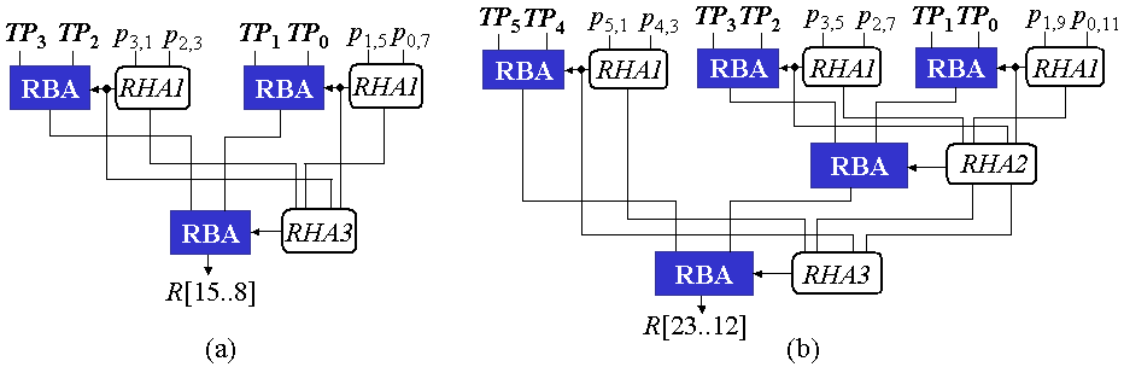

Fig. 5. Apply compensation value to TRM_H

Table 1. Probability distribution of a_1 with different n

n \ a_1	-3	-2	-1	0	1	2	3
8	-	0.12%	12%	75%	12%	0.12%	-
10	-	0.31%	14%	69%	15%	0.31%	-
12	-	0.61%	17%	64%	17%	0.61%	-
14	0.04%	0.98%	19%	60%	19%	0.99%	0.04%

Table 2. Probability distribution of a_2 with different n

n \ a_2	-3	-2	-1	0	1	2	3
8	-	0.04%	8.2%	86%	5.3%	0.04%	-
10	-	0.13%	10.6%	82%	7.2%	0.10%	-
12	-	0.26%	12.6%	78%	8.8%	0.19%	-
14	-	0.45%	14.1%	75%	10.2%	0.32%	-

Table 3. Transistor ratio for different multipliers

Multiplier	Transistor ratio			
	n=8	n=10	n=12	n=14
M_S	1	1	1	1
TRM_H	0.62	0.61	0.60	0.59
TRM_C	0.71	0.69	0.68	0.67

Table 4. Comparison results of average error

Multipliers	Error	n=8	n=10	n=12	n=14
$M_{K\text{-}G\text{-}A}$	$\bar{e}$	188.3	906.4	3842.1	17752.0
$M_{J\text{-}K\text{-}C}$		170.5	736.6	3094.2	12805.5
M_{Booth}		106.2	455.7	1912.1	11488.3
TRM_H		149.1	675.4	2982.9	12961.2
TRM_S		124.6	507.7	2043.1	8185.6
TRM_C		101.1	425.5	1786.0	7476.0

Table 5. Comparison results of the maximal absolute error

Multipliers	Error	n=8	n=10	n=12	n=14
$M_{K\text{-}G\text{-}A}$	e_M	1281	6145	32769	163841
$M_{J\text{-}K\text{-}C}$		515	2403	10979	49379
M_{Booth}		441	2105	9785	44601
TRM_H		938	4778	23210	109226
TRM_S		255	1023	4095	16383
TRM_C		459	2219	10923	51883

Table 6. Comparison results of variance of error

Multipliers	Error	n=8	n=10	n=12	n=14
$M_{K\text{-}G\text{-}A}$	u	22959	416043	9204493	377915712
$M_{J\text{-}K\text{-}C}$		10159	190805	3417020	63473866
M_{Booth}		6247	125055	2341510	41516237
TRM_H		14400	289841	5576828	104164651
TRM_S		5470	87463	1398529	22371591
TRM_C		5470	97922	1747885.28	30981360

Table 7. Quality comparison of reconstructed images for different multipliers

Image	Error	Multiplier		
		TRM_H	TRM_S	TRM_C
Lena	PSNR	37.35	40.98	42.47
	RMSE	11.96	5.18	3.68
Baboon	PSNR	37.10	40.20	43.51
	RMSE	12.68	6.21	2.90
Bear	PSNR	36.14	41.12	43.55
	RMSE	15.83	5.04	2.87
F16	PSNR	35.26	39.93	42.42
	RMSE	19.35	6.61	3.72