

A Circuit Design of 16×16 -bit Multiplier Using Redundant Binary Arithmetic

Tae-Min Kim

Dept. of Electronic Eng.,
Kumoh National University of Technology
188 Shinpyung, Kumi, Kyungbuk, Korea
Tel :+82-0546-467-4055
Fax : +82-0546-467-4321
E-mail : tmkim@knut.kumoh.ac.kr

Gun-Sun Shin

Dept. of Electronic Eng.,
Kumoh National University of Technology
188 Shinpyung, Kumi, Kyungbuk, Korea
Tel:+82-0546-467-4325
Fax: +82-0546-467-4321
E-mail : gsshin@knut.kumoh.ac.kr

Abstract — In this paper, a 16×16 -bit multiplier is designed using a Redundant Binary Adder (RBA) circuit so that it can make a fast addition of the Redundant Binary Partial Products (RB_PP's) by using Wallace-tree structure. Because a RBA adds two RB numbers, it acts as a 4-2 compressor, which reduces four inputs to two output signals. We propose a method to convert the Redundant Binary (RB) representation into the 2's complement binary representation. Instead of using the conventional full adders, a more efficient RB number to binary number converter can be designed with new conversion method. The method can be applied to both 'serial' and 'selector' modes. The sign extension can be reduced by the modified sign extension with no additional circuit. A 16×16 -bit multiplier is designed with this architecture. The active area size is $1.85 \times 1.28 \text{ mm}^2$ and the number of transistors is 7,410. This is the smallest number for all 16×16 -bit multipliers ever reported. Under the condition with 3.3V supply voltage, the chip achieves 10.6 ns multiplication time.

I. INTRODUCTION

Multipliers are fundamental components of digital hardware. They occupy a relatively large portion of the overall chip area and have often been the limiting factor in terms of speed. A simplicity of design and low transistor count are also required for the multipliers because of the increase in the bit size of the number to be calculated. The Wallace-tree method is commonly used to realize high speed because it is theoretically the fastest method [1]. Generated Redundant Binary Partial Products (RB_PP's) are added up by the Wallace-tree of Redundant Binary Adders (RBA's). Because an RBA adds two RB numbers to make one RB number, four inputs are reduced to two output signals. The RBA acts as a 4-2 compressor. The array of a RBA tree can increase operating speed by use of high speed RBA. Many algorithms and circuits have been reported for RBA [2-3].

The Wallace-tree of a RBA array adds the partial products until the final RB number is obtained. Then the final RB number must be converted to a NB number, that is the product, by an Redundant Binary to Normal Binary (RB-to- NB)

converter. We present a new method for converting the RB on to the 2's complement binary representation. The method requires less VLSI chip area and takes less conversion time than the conventional method [2-4].

The sign extension can be reduced by the modified sign extension with no additional circuit. There is no need to consider sign extension for partial products addition.

In the next section the details of this architecture are explained. To realize higher speed RB multiplier than conventional Normal Binary (NB) multipliers, the architecture must be optimized to CMOS circuit by taking advantage of RB number. In section III, the design and simulation of a 16×16 -bit multiplier employing this architecture are described. In section IV, this paper is concluded.

II. RB ARCHITECTURE

A. Partial Products Generation

RB architecture uses the well known method as shown in reference [5]. This is done according to the following consideration. The addition of A to B is expressed as

$$\begin{aligned} A + B &= A - (-B)_2 \\ &= A - (\bar{B} + 1) \\ &= (A - \bar{B}) - 1 \\ &= (-1)(a_{n-1} - \bar{b}_{n-1}) \cdot 2^{n-1} + \sum_{k=0}^{n-2} (a_k - \bar{b}_k) \cdot 2^k - 1 \\ &= \sum_{k=0}^{n-1} d_k \cdot 2^k - 1 \\ &= \sum_{k=0}^{n-1} d_k \cdot 2^k + (0, 1) \end{aligned} \quad (1)$$

Where \bar{B} is the inversion of B. Then the subtraction $A - \bar{B}$ becomes one of the following four forms whose values are 1, 0, or -1 :

$$(1, 0) = 1 \quad (1, 1) = 0 \quad (0, 0) = 0 \quad \text{and} \quad (0, 1) = -1 \quad (2)$$

Thus, the RB partial product, that is equal to the sum of two NB partial products, is generated by inverting one of the two

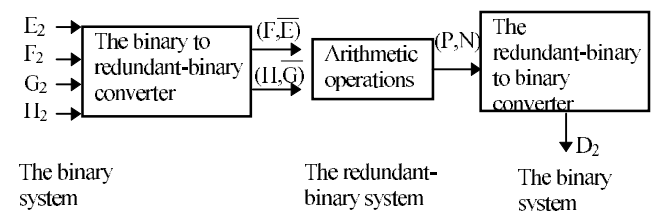


Fig. 1. Conversion process of the Redundant Binary to Normal Binary system

partial products and adding (0,1) to the lowest digit.

The drawback of the Booth algorithm is the sign extension. The use of 2's complement arithmetic to produce negative values of the multiplicand causes the MSB sign bits to be spanned on the entire bit-width when a signal changes sign. Fig. 2 exemplifies the RB_PP's multiplication scheme without the sign extension.

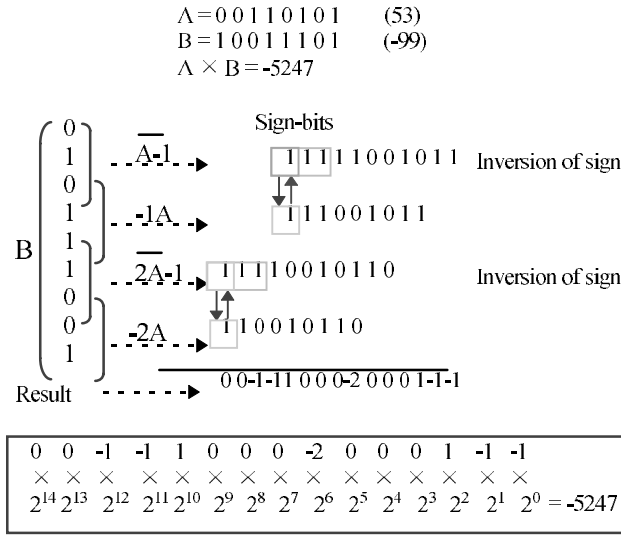


Fig. 2. An example of the RB_PP's multiplication scheme without the sign extension

B. Addition in Redundant Binary Representation

One important property of the redundant binary representation is its carry propagation free addition. This is the reason that RB adders usually are used to realize fast arithmetic operations. Another important property of the RB representation is that it does not use the 2's complement method to handle negative numbers, therefore, multiplication and division operations can be performed easily using the representation. However, for the RB representation, the conversion between the binary system and the RB system has to be performed. This is shown in Fig. 1. In the RB representation, the algebraic value of a n -bit number

$$B = (b_{n-1}, b_{n-2}, \dots, b_1, b_0) \quad (3)$$

case	(F_k, \bar{E}_k) (H_k, \bar{G}_k)	C_{k-1}	Carry Sum (C_k^+, C_k^-) (C_{k-1}^+, S_k^-)
1	$(0,0) (1,1) (1,0)$	1	$(0,0) (C_{k-1}^+, 1)$
	$(0,0) (0,0) (0,1)$	0	$(0,0) (C_{k-1}^+, 0)$
2	$(1,0) (0,0) (1,1) (1,0)$	1	$(0,0) (C_{k-1}^+, 0)$
	$(0,0) (1,0) (1,0) (1,1)$	0	$(1,0) (C_{k-1}^+, 1)$

Fig. 3. Rule for RB Addition. Immediate Carry and sum are shown for the six cases of input (F_k, \bar{E}_k) and (H_k, \bar{G}_k) .

2's complement B may be rewritten as follows :

$$\begin{aligned}
 B &= -b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} b_k \cdot 2^k \\
 &= 2 \times [-b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} b_k \cdot 2^k] - [-b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} b_k \cdot 2^k] \\
 &= -2 \times b_{n-1} \cdot 2^{n-1} + b_{n-1} \cdot 2^{n-1} + [\sum_{k=0}^{n-2} b_k \cdot 2^{k+1} - \sum_{k=0}^{n-2} b_k \cdot 2^k] \\
 &= -b_{n-1} \cdot 2^{n-1} + [\sum_{k=0}^{n-2} b_k \cdot 2^{k+1} - \sum_{k=0}^{n-2} b_k \cdot 2^k] \\
 &= [b_{n-2} - b_{n-1}] \cdot 2^{n-1} + \sum_{k=0}^{n-3} [b_k - b_{k+1}] \cdot 2^{k+1} + [0 - b_0] \\
 &= [b_{n-2} - b_{n-1}] \cdot 2^{n-1} + \sum_{k=0}^{n-2} [b_{k-1} - b_k] \cdot 2^k ; (b_{-1} = 0) \\
 &= d_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} d_k \cdot 2^k \\
 &= \sum_{k=0}^{n-1} d_k \cdot 2^k \quad (4)
 \end{aligned}$$

Where d_k is a binary signed digit ($d_k \in \{1, 0, -1\}$). Therefore, the conversion from an n -bit 2's complement binary representation into its RB representation based on new converter algorithm.

Let us consider four numbers E, F, G and H represented in 2's complement form and their RB product D expressed by (1) and (4).

$$E = -e_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} e_k \cdot 2^k \quad (5.a)$$

$$F = -f_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} f_k \cdot 2^k \quad (5.b)$$

$$G = -g_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} g_k \cdot 2^k \quad (5.c)$$

$$H = -h_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k \quad (5.d)$$

$$\begin{aligned}
 D &= E + F + G + H + (c_0^+ + c_0^-) \\
 &= (F, \bar{E}) + (H, \bar{G}) + (c_0^+ + c_0^-) - 2 \quad (6)
 \end{aligned}$$

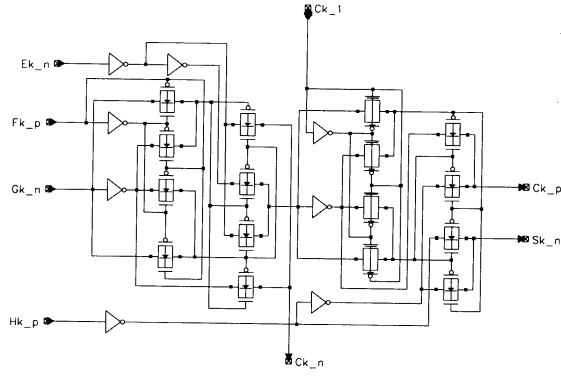


Fig. 4. CMOS circuit diagram of proposed RBA

By substituting (5) into (6) and using 2's complement number characteristics, D becomes

$$\begin{aligned}
 D &= (F - \bar{E}) + (H - \bar{G}) + (c_0^+ + c_0^-) - 2 \\
 &= (-f_{n-1} \cdot 2^{n-1} - \sum_{k=0}^{n-2} \bar{e}_k \cdot 2^k + \bar{e}_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} f_k \cdot 2^k \\
 &\quad - h_{n-1} \cdot 2^{n-1} - \sum_{k=0}^{n-2} \bar{g}_k \cdot 2^k) + \bar{g}_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k + (c_0^+ + c_0^-) - 2 \\
 &= ([2 \cdot c_{n-1}^- + s_{n-1}^+] \cdot 2^{n-1} + \sum_{k=0}^{n-2} [2 \cdot c_k^- + s_k^+] \cdot 2^k) \\
 &\quad + \bar{g}_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} h_k \cdot 2^k + (c_0^+ + c_0^-) - 2 \\
 &= c_{n-1}^- \cdot 2^n + [c_{n-2}^- + s_{n-1}^+ + \bar{g}_{n-1}] \cdot 2^{n-1} \\
 &\quad + \sum_{k=0}^{n-3} [c_k^- + s_{k+1}^+ + h_{k+1}] \cdot 2^{k+1} + s_0^+ + h_0 + (c_0^+ + c_0^-) - 2 \\
 &= c_{n-1}^- \cdot 2^n + [2 \cdot c_{n-1}^- + s_{n-1}^+] \cdot 2^{n-1} \\
 &\quad + \sum_{k=0}^{n-3} [2 \cdot c_{k+1}^- + s_{k+1}^+] \cdot 2^{k+1} + 2 \cdot c_0^- + s_0^+ + (c_0^+ + c_0^-) - 2 \\
 &= [c_{n-1}^+ + c_{n-1}^-] \cdot 2^n + \sum_{k=0}^{n-2} [c_k^+ + s_{k+1}^-] \cdot 2^{k+1} + (c_0^+ + s_0^-) - 2 \\
 &= d_n \cdot 2^n + \sum_{k=0}^{n-1} d_k \cdot 2^k - 2 \\
 &= \sum_{k=0}^n d_k \cdot 2^k - 2
 \end{aligned} \tag{7}$$

Using this Eq.(4), we define four numbers, c^+ , c^- , s^+ and s^- as given in (7).

$$f_k - \bar{e}_k - \bar{g}_k = 2 \cdot c_k^- + s_k^+ \tag{8}$$

By substituting (8) into (6), d becomes

$$d = 2 \cdot c_k^- + s_k^+ + h_k + (c_{k-1}^+ + c_{k-1}^-) - 2 \tag{9.a}$$

$$c_{k-1}^- + s_k^+ + h_k = 2 \cdot c_k^- + s_k^- \tag{9.b}$$

By substituting (9.b) into (9.a), d becomes

$$\begin{aligned}
 d &= 2 \cdot c_k^- + 2 \cdot c_{k-1}^- + c_{k-1}^+ + s_k^- - 2 \\
 &= 2(c_k^- + c_{k-1}^-) + (c_{k-1}^+ + s_k^-) - 2 \\
 &= 2(c_k^- + c_{k-1}^-) + d_k - 2
 \end{aligned} \tag{10}$$

We obtain RB representation as (4) (where, $d_k \in \{-1, 0, 1\}$). Finally, the RB number can be expressed as

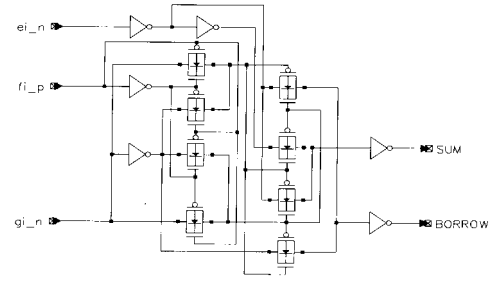


Fig. 5. CMOS circuit diagram of full subtractor (FS).

$$D = \sum_{k=0}^N d_k \cdot 2^k \tag{11}$$

C. Improved RBA

Generated RB partial products are added up by the Wallace tree of RBA's. Because a RBA adds two RB numbers to make one RB number, four inputs are reduced to two output signals. The RBA acts as a 4-2 compressor.

We consider the addition of the k th digit of two redundant binary numbers (F_k, \bar{E}_k) and (H_k, \bar{G}_k) expressed by the definition (7) (10).

$$(F_k, \bar{E}_k) + (H_k, \bar{G}_k) + (c_{k-1}^+ + c_{k-1}^-) = 2(c_k^+ + c_k^-) + (d_k^+, d_k^-) \tag{12}$$

where (d_k^+, d_k^-) is the sum. To simplify the consideration, we assume that both the inputs (F_k, \bar{E}_k) and (H_k, \bar{G}_k) take one of the four states (0,1), (0,0), (1,1) and (1,0). By this assumption, there are sixteen kinds of combination in the sum of (F_k, \bar{E}_k) and (H_k, \bar{G}_k) . They are classified into the six cases by the different results of the addition as shown in Fig. 3. The Fig. shows the intermediate sum (d_k^+, d_k^-) and carry (c_k^+, c_k^-) for each case. The carry is added to the sum of the higher digit.

Fig. 4 shows the CMOS circuit diagram of RBA that realizes the above expressions (7). This consists of inverters and Transmission Gate circuits (TG's) only.

D. RB-to-NB Conversion

The partial products are added in parallel in the Wallace tree of an RBA array until to be a final RB number. The final RB number is converted to an NB number by the final converter part. In the RB representation, the algebraic value of a n -digit number $[d_{n-1}, d_{n-2}, \dots, d_1, d_0]$ ($d_k \in \{-1, 0, 1\}$) is equal to $\sum_{k=0}^{n-1} d_k \cdot 2^k$, which is the same as the unsigned binary representation. Therefore, the conversion from the RB representation into its n -bit 2's complement binary number $[b_{n-1}, b_{n-2}, \dots, b_1, b_0]_2$ ($b_k \in \{0, 1\}$), representation can be realized by using new conversion algorithm Eq.(14). In the 2's complement representation, the algebraic value of an n -digit number $B = ([b_{n-2} - b_{n-1}], \dots, [b_0 - b_1], [0, -b_0])$ written in RB (See Eq. 4).

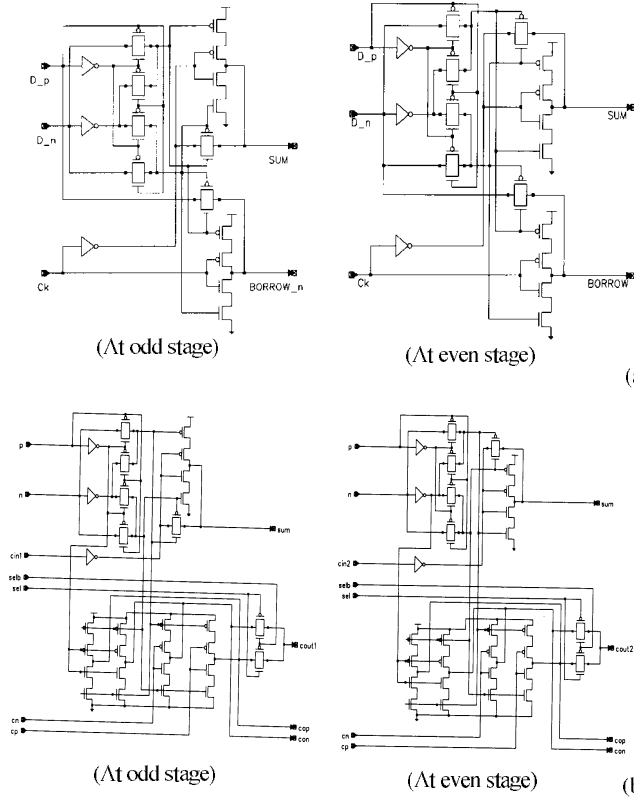


Fig. 6. CMOS circuit diagram of the new redundant-binary to binary con-

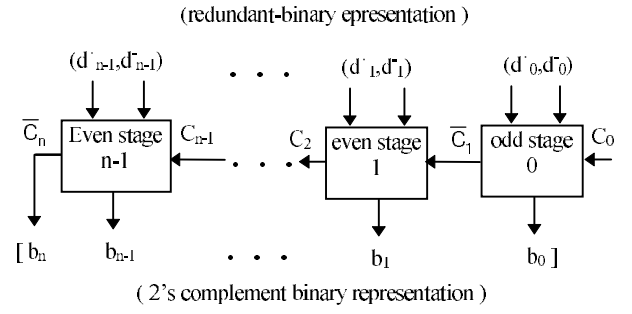
$$B = [b_{n-2} - b_{n-1}] \cdot 2^{n-1} + \left[\sum_{k=0}^{n-3} b_k - \sum_{k=0}^{n-3} b_{k+1} \right] \cdot 2^{k+1} - b_0 \cdot 2^0 \quad (13)$$

B may be rewritten as follows :

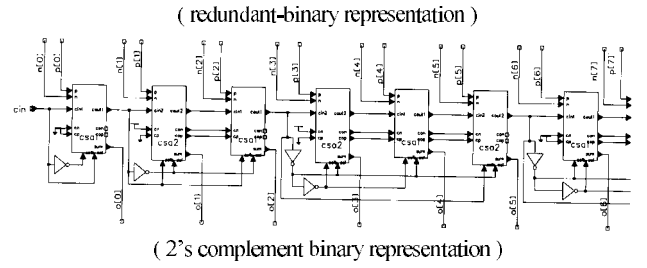
$$\begin{aligned} B &= [b_{n-2} - b_{n-1}] \cdot 2^{n-1} + \sum_{k=0}^{n-3} [b_k - b_{k+1}] \cdot 2^{k+1} + [0 - b_0] \\ &= [b_{n-2} - b_{n-1}] \cdot 2^{n-1} + \sum_{k=0}^{n-3} [b_k - b_{k+1}] \cdot 2^{k+1} + [0 - b_0] \\ &\quad - (b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-3} b_{k+1} \cdot 2^{k+1} + b_0) \\ &\quad + (b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-3} b_{k+1} \cdot 2^{k+1} + b_0) \\ &= [b_{n-2} + b_{n-1}] \cdot 2^{n-1} + \sum_{k=0}^{n-2} [b_{k-1} + b_k] \cdot 2^k + [0 + b_0] \\ &\quad - 2 \cdot (b_{n-1} \cdot 2^{n-1} + \sum_{k=0}^{n-2} b_k \cdot 2^k + b_0) \\ &= -b_{n-1} \cdot 2^n + \sum_{k=0}^{n-1} [b_{k-1} + b_k - b_{k-1}] \cdot 2^k + [0 + b_0] \\ &= -b_{n-1} \cdot 2^n + \sum_{k=0}^{n-1} b_k \cdot 2^k \end{aligned} \quad (14)$$

In this equation, the term in brackets is in the set $\{0, 1\}$. Therefore, the conversion from an n -bit RB representation into its 2's complement binary representation based on new converter algorithm. It is noted that the new converter does not need any extra operation required by the conventional converter. The conversion is carried out by the addition of d_k^+ to d_k^- , that is the addition of two NB numbers, therefore the

carry propagation occurs from lowest to highest digit. The



(a) The serial mode



(b) The select mode

serial-mode converter is shown in Fig. 6(a). For each stage k , given the input redundant digit (d_k^+, d_k^-) and the input variable C_k^- , we obtain the binary output B_k and the output variable C_{k+1} . The conversion rules are shown in Table I. It is noted that the new converter does not need any extra operation required. From the conversion rules shown in Table I, we have the Boolean equations for b_k and C_{k+1} .

TABLE I

The conversion rules in stage k shown in Fig. 6.

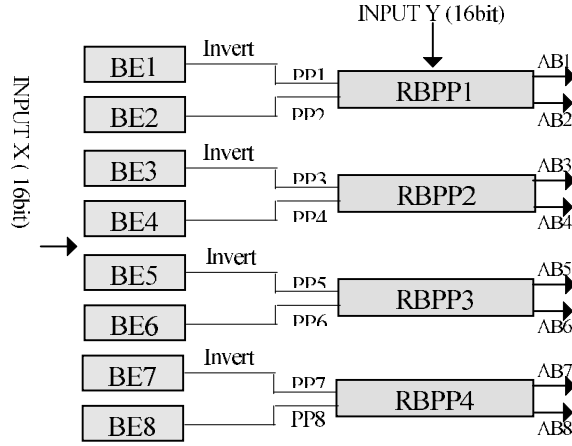
Input Signal			Output signal		
Redundant binary			Input	Sum	Borrow
D_k	(d_k^+, d_k^-)	C_k^-	C_{k+1}	b_k	C_{k+1}
0	0	0	0	0	0
0	0	0	1	1	1
-1	0	1	0	1	1
-1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	1

$$\begin{aligned} b_k &= (d_k^+ \text{ nand } d_k^-) \text{ xor } c_k \\ C_{k+1} &= C_k \text{ or } (c_p \text{ and } c_k) \end{aligned} \quad (15.a)$$

(15.b)

$$c_g = c_s \text{ and } d_k^-$$

(15.c)



An additional approach to increase the speed of a parallel adder that expends area in favor of speed is to use a carry-

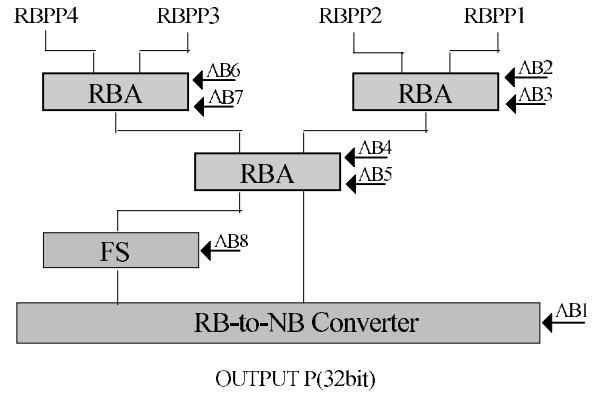
Fig. 7. Block diagram of 16×16 -bit multiplier.

TABLE II

Transistor count and delay time of RB-to-NB converters for different word lengths and different designs. Each value is obtained by Ispice simulation. Simulation parameter of $0.8 \mu\text{m}$ CMOS is used

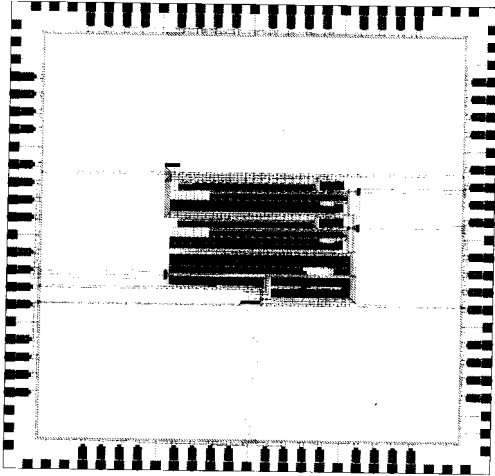
Word length	Transistor count		Delay time	
	serial mode	select mode	serial mode	select mode
8	182	308	3.0n	3.0n
16	390	660	5.8n	3.7n
32	806	1364	11.0n	5.5n
64	1638	2772	21.5.n	7.3n

TABLE III

Logic path and number of transistors of three types of multipliers

	Logic path		Number of transistors	
	16-bit	32-bit	16-bit	32-bit
Our multiplier	23	30	7,410	26,200
Array multiplier	93*	187*	10,700	43,300
Multiplier using a Wallace tree	25*	32*	7,700	31,200

* Full adder:3, Half adder:2

Fig. 8 Layout of 16×16 -bit multiplier

$$c_s = (d_k^+ \text{ xor } d_k^-) \quad (15.d)$$

$$c_p = (d_k^+ \text{ xnor } d_k^-) \quad (15.e)$$

The logic diagram in each stage k using inverter, TG's and tristate-inverter gates are given in Fig. 6(a).

select adder. Usually, two ripple-carry-adder structures are built, one with a zero carry-in and the other with a one carry-in. Fig. 6(b) shows a CMOS circuit diagram of the carry propagation circuit in this architecture. This is a kind of carry select method. Table II show the delay time and transistor count of each methods for word length of 8, 16, 32, and 64 bits. The HSPICE simulation was carried out to calculate the delay time using $0.8 \mu\text{m}$ CMOS parameters. Our converter has the fast speed and the least transistor count. Besides the speed and transistor count, our converter has one more advantage that the layout is quite easy, because our converter has a regular structure with simple interconnection. The easy layout reduces the design cost when the word length becomes large.

III. 16×16 -BIT MULTIPLIER DESIGN

A 16×16 -bit multiplier is designed with this RB Architecture [5]. Fig. 7 shows the block diagram and Fig. 8 shows the layout pattern of 16×16 -bit multiplier.

In the RBPP generation stage, modified Booth's algorithm [7] is used to halve the number of partial products. The pairs of two adjacent partial products, in which one is a non-inverted and the other is an inverted partial products, become RB partial products. The additional bits of all partial products that arise from the inversion of the sign are added by each RBA, FS (Full Subtractor) and RB-to-NB converter. The

additional bit (+1) of RBPP8 is converted to its complement form -1 and added in parallel by the FS's (Fig. 5).

TABLE IV
Features of 16×16 -bit multiplier

Multiplier, Multiplicand	16-bit (two's complement)
Product	32-bit (two's complement)
Supply voltage	3.3V
Multiplication time	10.6ns
Active area size	$1.85 \times 1.28 \text{ mm}^2$
Transistor count	7,410
Density of transistors	3.13k/ mm^2
Process	0.8 μm double metal CMOS

In the RB-to-NB conversion stage, the final RB number with 32 digits is converted to an NB number with the same number of digits. A CMOS circuit diagram in this architecture is shown by Fig 6(b). This is a kind of carry select mode. The conversion speed of this stage depends on the conversion time of 32-digits. The number of stages of the multiplexer circuits is 8 in the critical path of the 32-bit converter.

We compare our multiplier having such features with other high-speed multipliers. Table III shows a comparison of our multiplier, an array multiplier, and a multiplier using a Wallace tree for 16-bit and 32-bit multipliers for the intrinsic logic path.

We evaluated the operating speed of the multiplier using the HSPICE simulation. We used the parameter of 0.8 μm CMOS in the simulation. The supply voltage is 3.3V. The simulation is carried out for the critical path extracted from the whole multiplier circuit. A multiplication time of 10.6ns is obtained by the simulation. The delay time of the RB_PP generation stage is 2.2ns. The delay time in the Wallace-tree stage is 4.1ns, which consists of two stages of RBA's and one FS's. In the RB-to-NB conversion stage, the delay time of 4.3ns is obtained, which is the delay time of 32-bit conversion. The features of the multiplier are shown in table IV.

IV. CONCLUSION

A 16×16 -bit multiplier is designed using the high speed RB architecture. The RB_PP's are added up by an array of the improved RBA's, that is less transistor count than the conventional 4-2 compressors. A simple and high speed RB-to-NB converter is developed. The process technology of 0.8 μm CMOS with double metal will be used to fabricate the multiplier. The supply voltage is 3.3V. The number of transistors is 7,410. These are the smallest number for all 16×16 -bit multipliers ever reported.

ACKNOWLEDGMENT

This work was financially supported by the IC Design Education Center (IDEC).

REFERENCES

- [1] C. S. Wallace, "A Suggestion for Fast Multiplier." *IEEE Trans. Electron. Comput* vol.EC-13, pp.14-17, Feb. 1964.
- [2] Y. Harata, Y. Nakamura, H. Natase, M. Takigawa, and N. Takagi, "A high speed multiplier using a redundant binary adder tree," *IEEE J. Solid-state Circuits*, vol. SSC-22, pp. 28-34, Feb. 1987.
- [3] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation," *Dig. Tech. Papers of 1988 ISSCC*, pp. 152-153, Feb. 1988.
- [4] M. Tonomura, "High-speed digital circuit of discrete cosine transform," *SP97-41, DSP94-66, Tech. Rep. IEICE Japan*, pp. 39-46, 1994.
- [5] MAKINO et al, "An 8.8-ns 54 x 54-bit multiplier with high speed architecture" *IEEE journal of solid-state circuits*. Vol. 31, No 6, June 1996.
- [6] Sung-ming Y, Chi-sung L, Chin-hsing C, and Jau-yien L, "An efficient redundant-binary number to binary number converter." *IEEE Journal of solid-state circuits*. Vol. 27, No. 1, January 1992.
- [7] A. D. Booth: "A Signed Binary Multiplication Technique," *Quart. J. Mech. Appl. Math.*, Vol. 4, Pt. 2, pp. 236-240, 1951.
- [8] L. P. Rubinfeld, "A proof of the modified Booth's algorithm for multiplication," *IEEE Trans. On Computers*, Vol.24, no. 10, pp. 1014-1015, Oct. 1975.
- [9] H. Sam and A. Gupta: A Generalized Multibit Recoding of Two's Complement Binary Numbers and Its Proof with Application in Multiplier Implementations, *IEEE Trans. Comput.*, Vol. 39, No. 8, pp. 1006-1015, Aug. 1990.