An 8.8-ns 54 × 54-Bit Multiplier with High Speed Redundant Binary Architecture

Hiroshi Makino, Yasunobu Nakase, Hiroaki Suzuki, Hiroyuki Morinaka, Hirofumi Shinohara, and Koichiro Mashiko, *Member, IEEE*

Abstract—**A high speed redundant binary (RB) architecture, which is optimized for the fast CMOS parallel multiplier, is developed. This architecture enables one to convert a pair of partial products in normal binary (NB) form to one RB number with no additional circuit. We improved the RB adder (RBA) circuit so that it can make a fast addition of the RB partial products. We also simplified the converter circuit that converts the final RB number into the corresponding NB number. The carry propagation path of the converter circuit is carried out with only multiplexer circuits. A 54 × 54-bit multiplier is designed with this architecture. It is fabricated by 0.5 μm CMOS with triple level metal technology. The active area size is 3.05 × 3.08 mm^2 and the number of transistors is 78,800. This is the smallest number for all 54 × 54-bit multipliers ever reported. Under the condition of 3.3 V supply voltage, the chip achieves 8.8 ns multiplication time. The power dissipation of 540 mW is estimated for the operating frequency of 100 MHz. These are, so far, the fastest speed and the lowest power for 54 × 54-bit multipliers with 0.5-μm CMOS.**

I. Introduction

As THE HIGH speed computing systems become increasingly versatile, which demand not only faster multiplier chips but also more precise calculation within the chips, the bit length and the integration density of the multipliers increase. For example, the bit length of 54 is needed to calculate the double-precision floating-point number, which consists of 53-bit mantissa and 1-bit sign. Therefore, a simple design with a small number of transistors is also a key for high speed multipliers. The Wallace-tree method [1] is commonly used to realize high speed because it is theoretically the fastest method. However, the conventional Wallace-tree has a complicated interconnection between three-input/two-output full adders (3-2 compressors). This makes the layout very difficult and inefficient. The extended layout process increases the design costs. Furthermore, as the multipliers increase in bit size, the interconnection becomes exponentially complicated.

To solve the above problem with conventional Wallace-trees, the following two methods have been proposed. One method is to use four-input/two-output adders (4-2 compressors) instead of the conventional 3-2 compressors. The use of 4-2 compressors simplifies the interconnection drastically because the partial products are added up in the form of a binary tree. The 4-2 compressor circuit was simply realized by two stages of 3-2 compressors that have the total delay of four exclusive-OR (XOR) gates [2]. Recently, the optimized 4-2 compressors, that have the delay of three XOR gates, were reported [3]–[5]. High speed 54 × 54-bit multipliers have been realized with the optimized 4-2 compressors. Another method is to use redundant binary (RB) for the partial products. The use of the RB simplifies the interconnection because the RB partial products can also be added up in the binary tree form by RB adders (RBA's). It not only simplifies the design, but it also improves speed because it requires no continuous carry propagation. Therefore, many studies have been made to apply the RB to multipliers [6]–[10]. However, the conventional RB multipliers have not exceeded the NB multipliers in speed so far, because of the following three reasons

1) Additional circuits are needed to convert NB partial products to RB numbers. Therefore, the hardware increases and some delay arises from the conversion.
2) Several types of the RB adder (RBA), the key component of the multiplier, have been proposed [6]–[13]. However, they have not proved their obvious superiority in operating speed over the NB 4-2 compressors [3]–[5].
3) RB multipliers require conversion from RB to NB in the last stage, where NB multipliers use carry look-ahead adders (CLA's). Several kinds of RB-to-NB converters have been proposed [14]–[16], some of which had the advantage over CLA's. However, there has been no report on applying them to multipliers effectively.

To realize higher speed RB multiplier than conventional NB multipliers, the architecture must be optimized to CMOS circuit by taking advantage of RB number. We have developed an RB architecture that solves the above problems and enables us to obtain an RB multiplier with a higher speed and a lower transistor count than NB multipliers. In the next section the details of this architecture

Manuscript received June 21, 1994; revised August 15, 1995.

H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, and K. Mashiko are with the System LSI Laboratory, Mitsubishi Electric Corporation, 4-1 Mizuhara, Itami 664 Japan.

H. Shinohara is with Mitsubishi Electric Corporation, Headquarters, 2-2-3 Marunouchi Chiyoda-ku Tokyo, 100 Japan.

Publisher Item Identifier S 0018-9200(96)04207-2.

are explained. In Section III, the design and simulation of a 54 × 54-bit multiplier employing this architecture are described. In Section IV, the fabrication technology and the experimental result of the multiplier are presented. In Section V, this paper is concluded.

II. High-Speed RB Architecture

A. RB Partial Products Generation

In the RB architecture proposed here, an RB partial product (RBPP) is generated from two NB partial products as proposed in [7], [8], and [10]. However, in this architecture, the RB partial product generation is carried out more simply than in the previous ones. This is done according to the following consideration.

The addition of A to B is expressed as

$$A + B = A - (-B) \tag{1}$$

where A and B are the NB partial products. Using the two's complement representation, we can obtain $-B$ by inverting all bits of B and then adding a bit "1" to the lowest digit. The procedure is expressed as

$$-B = \overline{B} + 1 \tag{2}$$

where $\overline{B}$ is obtained by inverting all bits of B. Substitution of (2) into (1) produces the following expression

$$A + B = A - \overline{B} - 1. \tag{3}$$

Here we assume a_i and b_i are the ith digits of A and B, respectively. Also we define

$$(a_i, \overline{b_i}) \equiv a_i - \overline{b_i} \tag{4}$$

$$(A, \overline{B}) \equiv A - \overline{B} \tag{5}$$

where $\overline{b_i}$ is the inversion of b_i. Then the subtraction $a_i - \overline{b_i}$ becomes one of the following four forms whose values are 1, 0, or -1:

$$(1, 0) = 1, (1, 1) = (0, 0) = 0 \quad \text{and} \quad (0, 1) = -1. \tag{6}$$

This means that the pair of $(A, \overline{B})$ can be regarded as an RB number having $(a_i, \overline{b_i})$ for each digit. The value of $(A, \overline{B})$ is equal to $A + B + 1$ from the expression (3) and (5). That is

$$A + B = (A, \overline{B}) - 1 = (A, \overline{B}) + (0, 1) \tag{7}$$

because -1 is equal to (0, 1) from (6). Thus, the RB partial product, that is equal to the sum of two NB partial products, is generated by inverting one of the two partial products and adding (0, 1) to the lowest digit.

An example of the conversion is shown in Fig. 1. This figure shows that the conversion of 8-bit two's complement partial products $A = 10100110$ and $B = 01101101$, which are -90 and 109 in decimal expression, respectively. First, the sign of B is inverted. This inversion is carried out by inverting all digits and adding "1" to the lowest digit. Therefore, B is converted to 10010010 and "1." Then, A and the inverted B make a pair. The pair becomes an RB partial product (1,1)(0,0)(1,0)(0,1)(0,0)(1,0)(1,1)(0,0) (= 20) and the additional bit (0, 1) (= −1). The sum of the RB partial product (= 20) and the additional bit (= −1) is equal to the sum of the two NB partial products (−90 + 109 = 20 + (−1) = 19). In this way, the RB partial product is easily generated from two NB partial products.

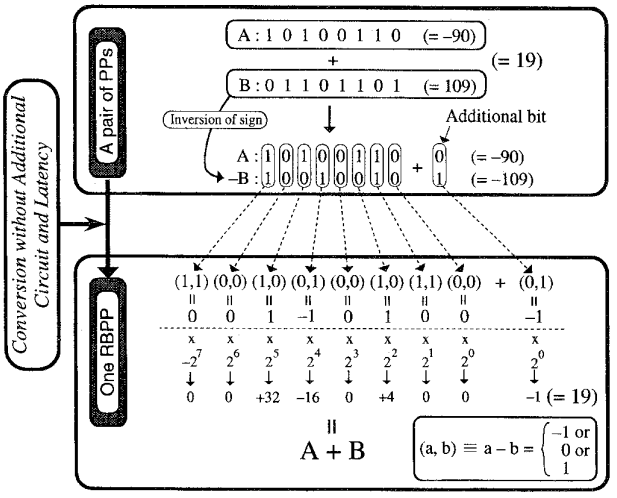


Fig. 1. Example of RBPP generation. Addition of two's complement NB number A (= 10100110) and B (= 01101101) is carried out only by inverting the sign of B.

In the previously proposed RB method [7], [8], [10], the sign and absolute value are used for the RB expression instead of the definition (6). Therefore, each digit must be converted to the sign and absolute value by OR-gate and Exclusive-OR-gate that increase the hardware and delay time. In our RB architecture, an RB partial product can be obtained from two NB partial products only by inverting one of the pairs because the expression (6) is adopted. No additional hardware is needed. This method is applicable to any kind of summation of multiple numbers, although the previous methods are limited for multipliers using Booth algorithm [17].

B. Improved RBA

Generated RB partial products 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 an RBA tree can increase operating speed by use of high speed RBA. Many algorithms and circuits have been reported for RBA [6]–[13]. Also several kinds of 4-2 compressors (NBA's) are proposed for NB multipliers [3]–[5]. Fig. 2 shows the representative conventional RBA's and NBA's that have simple and high speed structures. RBA2 [8] adds two RB's expressed by signs a_i^s and b_i^s, and absolute values a_i^v and b_i^v to make the sum d_i^s and d_i^v. RBA3 [13] adds the RB's with the plus-values a_i^+ and b_i^+ and minus-values a_i^- and

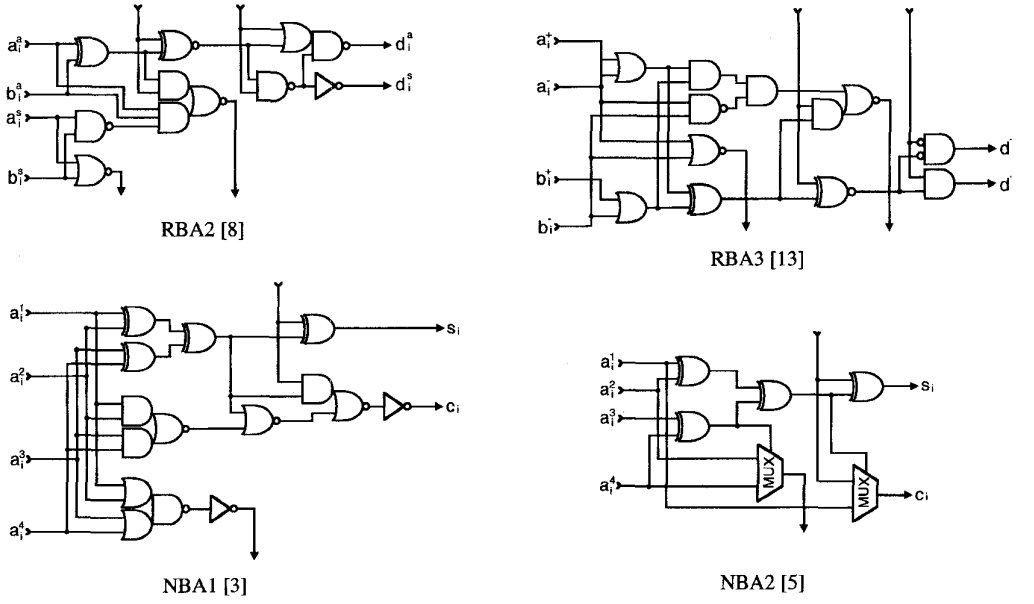


Fig. 2. Schematic diagram of conventional 4-2 compressors. RBA2 adds two RB's in sign and absolute expression. RBA3 also adds two RB's in plus and minus values. NBA1 and NBA2 add four NB's to make sum and carry.

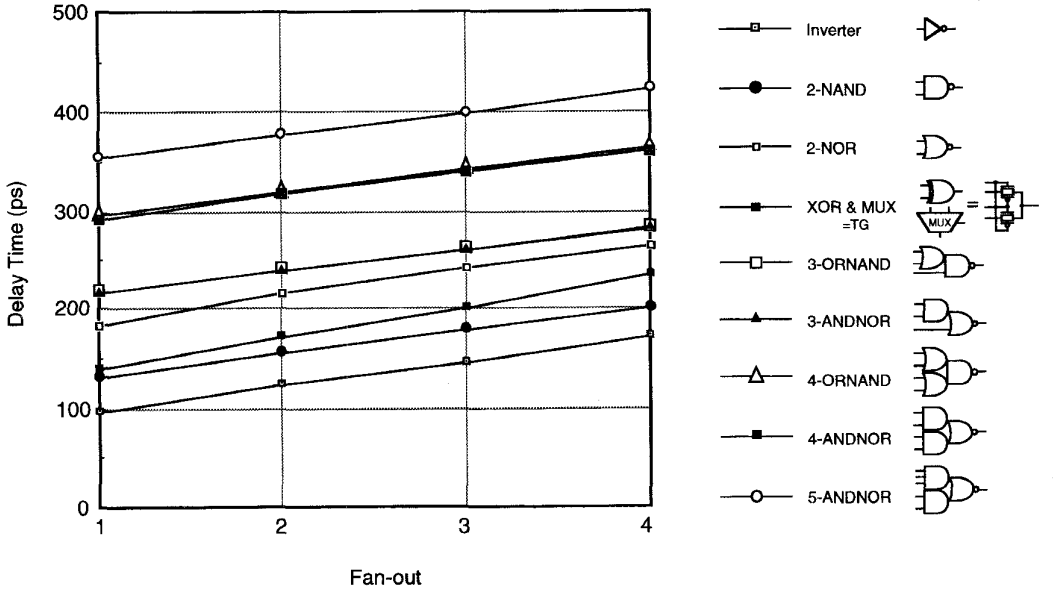


Fig. 3. Fan-out dependence of delay time for various gates by SPICE2 simulation. The parameter of 0.5-μm CMOS is used. Inverter, 2-NAND and TG are faster than other gates.

b_i^- to make the sum d_i^+ and d_i^-. This expression is the same as this paper. NBA1 [3] and NBA2 [5] are the conventional 4-2 compressors. Both NBA's add four NB numbers a_i^1, a_i^2, a_i^3, and a_i^4 to make sum s_i and carry c_i. To analyze the operating speed of each adder, we simulated the delay time of basic gates constructing them. The simulator is SPICE2. The simulation parameter of 0.5-μm CMOS is used. Fig. 3 shows the delay time of nine kinds of gates as a function of fanout. For exclusive-OR (XOR) circuits and multiplexer circuits, transmission gate circuits (TG's) are used as in the figure, because TG is faster than other types of XOR gate circuits proposed in [3] and [10]. From Fig. 3, inverter, 2-NAND and TG are faster than other six gates. NOR gates and complex gates with more than three inputs are not suitable for high speed operation. Therefore, it is desirable to use only inverters, 2-NAND's, and TG's for the RBA.

In the RB architecture proposed here, we have improved the RBA to fit to the CMOS circuit by eliminating OR gates and multi-input complex gates. We refined the conventional algorithm for RBA as in the following. Now we consider the addition of the ith digit of two redundant

TABLE I
RULE FOR RB ADDITION. IMMEDIATE CARRY (C_i^+, C_i^-) AND SUM (S_i^+, S_i^-) ARE SHOWN FOR THE FIVE CASES OF INPUT (a_i^+, a_i^-) AND (b_i^+, b_i^-)

case	(a_i^+, a_i) (b_i^+, b_i)	h_{i-1}	Immediate carry (c_i^+, c_i)	Immediate sum (s_i^+, s_i)
1	(0, 0) (0, 0) (1, 0) (0, 1) (0, 1) (1, 0)	any	(0, 0)	(0, 0)
2	(0, 1) (0, 0) (0, 0) (0, 1)	0	(0, 0)	(0, 1)
		1	(0, 1)	(1, 0)
3	(1, 0) (0, 0) (0, 0) (1, 0)	0	(1, 0)	(0, 1)
		1	(0, 0)	(1, 0)
4	(0, 1) (0, 1)	any	(0, 1)	(0, 0)
5	(1, 0) (1, 0)	any	(1, 0)	(0, 0)

binary numbers (a_i^+, a_i^-) and (b_i^+, b_i^-) expressed by the definition (4)

$$(a_i^+, a_i^-) + (b_i^+, b_i^-) \rightarrow (d_i^+, d_i^-) \tag{8}$$

where (d_i^+, d_i^-) is the sum. To simplify the consideration, we assume that both the inputs (a_i^+, a_i^-) and (b_i^+, b_i^-) take one of the three states (0, 1), (0, 0), and (1, 0), and no (1, 1). We assume that (1, 1) has already been transformed to (0, 0). By this assumption there are nine kinds of combination in the sum of (a_i^+, a^-) and (b_i^+, b_i^-). They are classified into the five cases by the different results of the addition as shown in Table I. The table shows the intermediate sum (s_i^+, s_i^-) and carry (c_i^+, c_i^-) for each case. The carry is added to the sum of the higher digit. The cases 2 and 3 are further divided into two cases by the value of h_i from the next lower digit, where h_i is defined as follows

$$h_i = 1 \text{ if } \begin{matrix}(a_i^+, a_i^-)\\(b_i^+, b_i^-)\end{matrix} \in \left\{\begin{matrix}(0,1) & (1,0) & (0,1) & (0,0) & (0,1)\\(1,0) & (0,1) & (0,0) & (0,1) & (0,1)\end{matrix}\right\},$$

$$h_i = 0 \text{ if } \begin{matrix}(a_i^+, a_i^-)\\(b_i^+, b_i^-)\end{matrix} \in \left\{\begin{matrix}(0,0) & (1,0) & (0,0) & (1,0)\\(0,0) & (0,0) & (1,0) & (1,0)\end{matrix}\right\}. \tag{9}$$

The h_i is introduced to prevent the continuous carry propagation by eliminating the collision of the sum and the carry from the lower digit. Therefore, the addition can be carried out without the carry propagation as long as it follows the rule shown in the Table I. The final sum (d_i^+, d_i^-) can be expressed as follows:

$$d_i^+ = (s_i^+ + c_{i-1}^+) \cdot \overline{(s_i^- + c_{i-1}^-)}, \tag{10}$$

$$d_i^- = \overline{(s_i^+ + c_{i-1}^+)} \cdot (s_i^- + c_{i-1}^-). \tag{11}$$

This expression eliminates the state of (1, 1) for (d_i^+, d_i^-), then the (d_i^+, d_i^-) takes either of the three states of (0, 1), (0, 0) and (1, 0), just like the input cases of (a_i^+, a_i^-) and (b_i^+, b_i^-). From (8)–(11) and the rule of Table I, we obtain the following seven equations for the RBA

$$d_i^+ = \overline{\alpha_i} \cdot \beta_{i-1}, \tag{12}$$

$$d_i^- = \alpha_i \cdot \overline{\beta_{i-1}}, \tag{13}$$

$$\alpha_i = l_i \oplus h_{i-1}, \tag{14}$$

$$\beta_i = \overline{l_i} \cdot k_i + l_i \cdot \overline{h_{i-1}}, \tag{15}$$

$$l_i = (a_i^+ + a_i^-) \oplus (b_i^+ + b_i^-), \tag{16}$$

$$h_i = a_i^- + b_i^-, \tag{17}$$

$$k_i = a_i^+ + b_i^+. \tag{18}$$

Fig. 4 shows the schematic and the circuit diagram of our RBA (RBA1) that realizes the above expressions (12)–(18). Negative logic are used for inputs and output signals to simplify the circuits. This consists of inverters, 2-NAND's and TG's only. No complex gate or NOR gate are used.

To compare our RBA1 with the other RBA's and NBA's, we simulated the RBA1-3 and NBA1-2. We investigated the delay time of each adder for the practical load condition of fan-out 1 and the interconnection length of 0.5 mm. Because the output of the adder travels through some wiring into the next adder. Table II shows the delay time of the adders. Our RBA1 is more than 150 ps faster than the other adders. RBA2, RBA3, and NBA1 have complex gates that degrade the operating speed RBA3 is the slowest because it has the long critical paths as a complex gate. NBA2 has a simple structure without complex gate. However, it has series TG's, in which the output of TG is connected to the source node of next TG. Such series connection degrades the propagating signal seriously and consequently leads to a slower speed. To avoid the degradation, the output of the TG's are buffered by inverters in the critical path. Thus, the number of gates in the critical path increase and the operating speed is slower than RBA1. The same problem also decreases the operating speed of NBA1. Our RBA1 does not have complex gates and such series TG's as NBA1-2. Thus, RBA1 is the fastest.

It should be noted that the use of RBA1 requires the conversion from (1, 1) to (0, 0) before the Wallace-tree stage. However this overhead is small, because the conversion is carried out by a stage of 2-NAND gate with 150 ps delay. The high speed of RBA1 hides the overhead. Furthermore, the same kind of overhead also exists in the NBA, because the buffering circuits are needed to drive the TG's in the first stage of NBA. Therefore, this overhead is not a disadvantage of this architecture.

C. RB-to-NB Conversion

The Wallace-tree of an RBA array adds the partial products until the final RB number (F^+, F^-) is obtained. Then the final RB number must be converted to an NB number Z, that is the product, by an RB-to-NB converter. The conversion is carried out by the addition of F^+ to

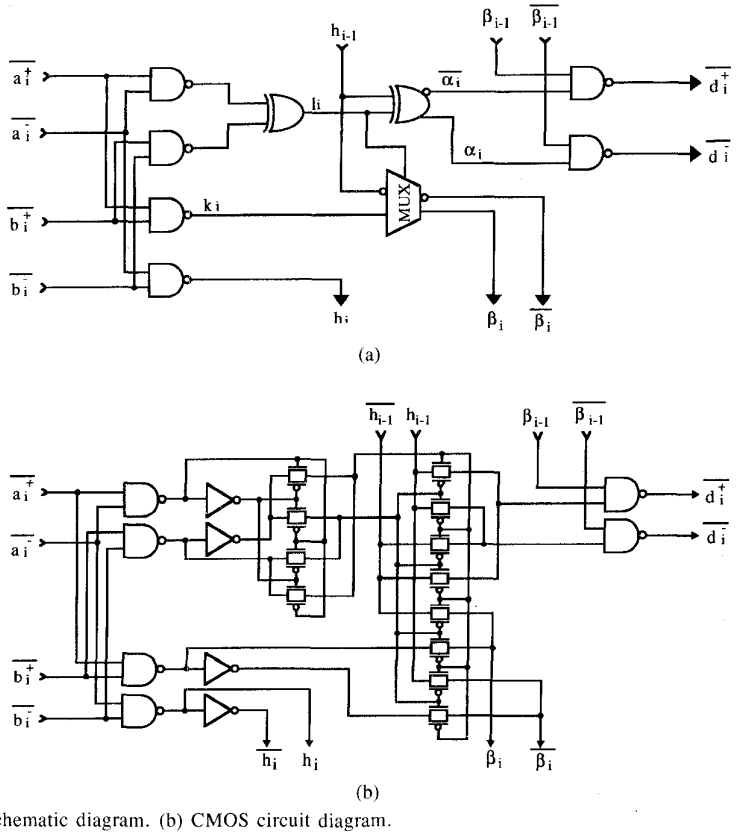


Fig. 4. Proposed RBA1. (a) Schematic diagram. (b) CMOS circuit diagram.

TABLE II
COMPARISON OF DELAY TIME FOR DEFFERENT 4.2 COMPRESSORS. EACH VALUE IS OBTAINED BY SPICE2 SIMULATION FOR THE LOAD CONDITION OF FAN-OUT 1 AND 0.5-mm INTERCONNECTION. SIMULATION PARAMETER OF 0.5-μm CMOS IS USED

	Present Work				
Adder Type	RBA1	RBA2	RBA3	NBA1	NBA2
Delay Time (ns)	0.89	1.11	1.36	1.20	1.04

$-F^-$ from the definition (5). That is

$$Z = F^+ - F^- = F^+ + (-F^-) = F^+ + \overline{F}^- + 1. \tag{19}$$

Because there are two NB numbers, the continuous carry propagation occurs from the lowest to the highest digit. The CLA is used in the conventional RB-to-NB converters [7], [9] to speedup the conversion. CLA adders are also used in the conventional NB multipliers [3]–[5]. There has been several reports on special circuits for RB-to-NB conversion [14]–[16]. The RB-to-NB converter has an advantage that it does not need the circuits to generate the "generate signal G" nor "propagate signal P" [15]. Therefore, an RB-to-NB converter is possible to realize higher speed and less transistor count than conventional CLA's. However, these special circuits have not exceeded the conventional CLA adders because the CLA in NB multipliers have also been optimized to realize a high speed.

In the architecture proposed here, we optimized the RB-to-NB conversion circuit so that the carry propagation circuit comprises simple and high speed circuits. We took into consideration the following criteria in designing the new RB-to-NB converter. If each digit of the final RB partial product (F^+, F^-) is (f_i^+, f_i^-) and the carry-in from the next lower digit is c_{i-1}, the carry-out signal c_i is expressed from (19) as follows:

$$c_i = f_i^+ \cdot \overline{f_i^-} + f_i^+ \cdot c_{i-1} + \overline{f_i^-} \cdot c_{i-1}, \tag{20}$$

$$c_0 = 1. \tag{21}$$

Because, as mentioned above, the output of the RBA takes one of the three states, (1, 0), (0, 0) or (0, 1) and never takes the state of (1, 1), f_i^+ and f_i^- never become "1" simultaneously. Then we can rewrite (20) as

$$c_i = f_i^+ \cdot \overline{c_{i-1}} + \overline{f_i^-} \cdot c_{i-1}. \tag{22}$$

This implies the carry propagation can be carried out by multiplexer circuits. The carry-out c_i is selected from the two signals f_i^+ and $\overline{f_i^-}$ by a carry-in signal c_{i-1}. As shown in the above section, the multiplexer circuits consists of the TG circuits. Fig. 5(a) shows a schematic diagram of the carry propagation circuit in this architecture. Each square indicates a multiplexer. This is a kind of carry select method. The carry-in to the first (lowest) digit is al-

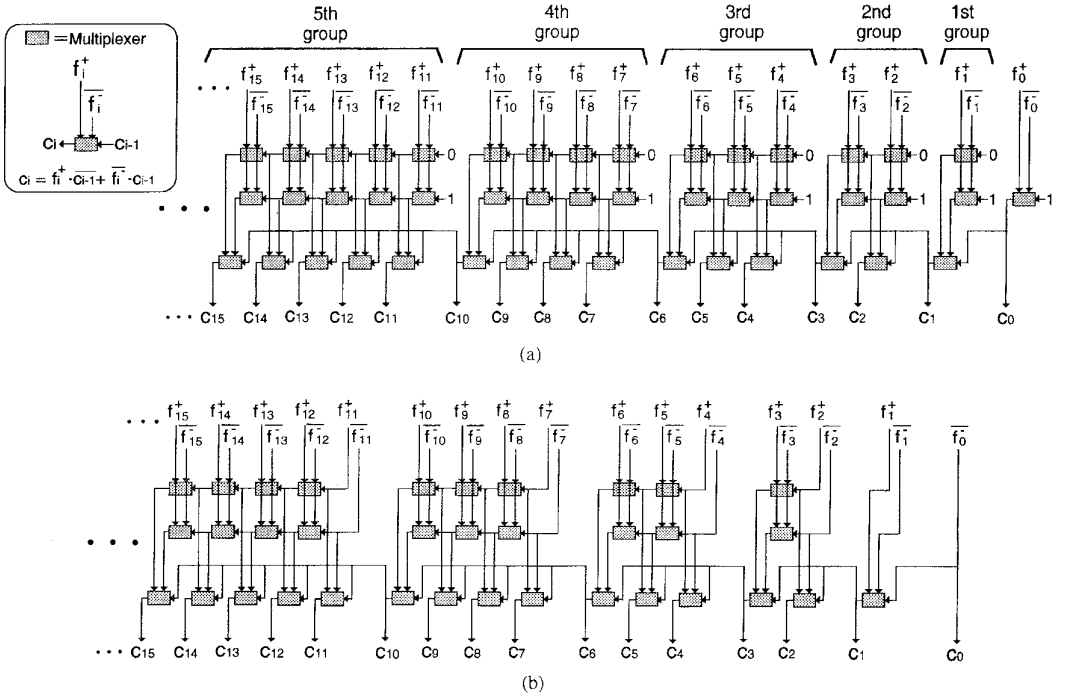


Fig. 5. Schematic diagram of carry propagation circuit of proposed RB-to-NB to converter. Basic circuit (a) is simplified to reduced circuit (b) by eliminating the multiplexers with predetermined input.

ways "1" from (21). The digits beyond the second digit are grouped in such a way that the number of the digits in the group increases by one starting from one. For example, the number of digits in the first group is one, the second group, two, and so on. In each group, two carry propagation circuits are prepared to propagate the carry-out signals for the two cases of carry-in "0" and "1". The carry-out signal from the highest digit in the lower group alternately selects the generated two carry-out signals. This carry propagation circuit is still optimized as shown in Fig. 5(b) because the multiplexers having the predetermined input can be eliminated. The number of the multiplexers in all critical paths are the same by increasing the number of digits in the groups by one. Therefore, the speed is minimized.

The product Z can be easily calculated using the carry-out signal from the lower digit and the following relation

$$z_i = f_i^+ \oplus \overline{f_i^-} \oplus c_{i-1} \tag{23}$$

where z_i is the ith digit of the product Z. This relation can be simplified by using the fact that f_i^+ and f_i^- do not become "1" simultaneously. Then we get

$$z_i = (\overline{f_i^+ + f_i^-}) \oplus c_{i-1}. \tag{24}$$

This is carried out by a 2-NAND and a TG circuit.

In this conversion scheme, the output of TG is connected to the gate node of the next TG from the relation (22) and (24). There is no series TG's with more than two-stages, in which the output of the TG is connected to the source node of next TG. As mentioned in the previous section, the absence of such series TG's is advantageous for high speed because the insertion of the buffer circuits is not needed into the output of TG's. Therefore, this conversion scheme is suitable for high speed operation.

Our RB-to-NB converter (CONV1) is compared with other methods. To make a fair comparison, we chose good conventional architecture that realize a high speed, and then optimized them for different word length. In the conventional CLA adder, we chose two kinds of adders CLA1 [3] and CLA2 [5], CLA1 uses improved carry look-ahead circuits and carry select method. CLA2 uses a kind of binary look-ahead method and carry select method. Both CLA's were used in the multipliers and realize a high speed. In the conventional RB-to-NB converter, we also chose two kinds of converter CONV2 [15] and CONV3 [16]. CONV2 is a simplified CLA by changing the generating circuits of the "generate signal G" and the "propagate signal P" to inverters according to the consideration in [15]. CLA2-type circuit is used for the CLA part of CONV2. CONV3 is a specialized circuit for RB-to-NB converter [16] that uses series TG's for the carry propagation circuit. Because series TG's degrades the speed as mentioned above, we inserted inverters by every two stages of series TG's. Tables III and IV show the delay time and transistor count of each method for word length of 8, 16, 32, and 64 bits. The SPICE2 simulation was carried out to calculate the delay time using 0.5-μm CMOS parameters. CLA1 and CLA2 make little differences in both the speed and transistor count. CLA2 has slightly faster speed and less transistors than CLA1 for 64-bit word. CONV2 still realizes faster speed and less transistor count than CLA2 because of the simplified cir-

TABLE III
COMPARISON OF DELAY TIME OF RB-to-NB CONVERTERS AND CLA ADDERS FOR DEFFERENT DESIGN AND DIFFERENT WORD LENGTH

Word Length (W)	Our Converter	Conventional Converter		Conventional CLA adder	
	CONV1	CONV2 [15]	CONV3 [16]	CLA1 [3]	CLA2 [5]
8	1.21ns	1.18ns	1.90ns	1.06ns	1.27ns
16	1.56	1.63	2.08	1.72	1.72
32	1.90	1.95	2.80	2.22	2.04
64	2.42	2.58	2.94	2.70	2.67

TABLE IV
COMPARISON OF TRANSISTOR COUNT OF RB-to-NB CONVERTERS AND CLA ADDERS FOR DEFFERENT DESIGN AND DIFFERENT WORD LENGTH

Word Length (W)	Our Converter	Conventional Converter		Conventional CLA adder	
	CONV1	CONV2	CONV3	CLA1	CLA2
8	264	228	262	274	276
16	536	564	604	548	660
32	1048	1268	1312	1424	1460
64	1840	2676	2936	3192	3060

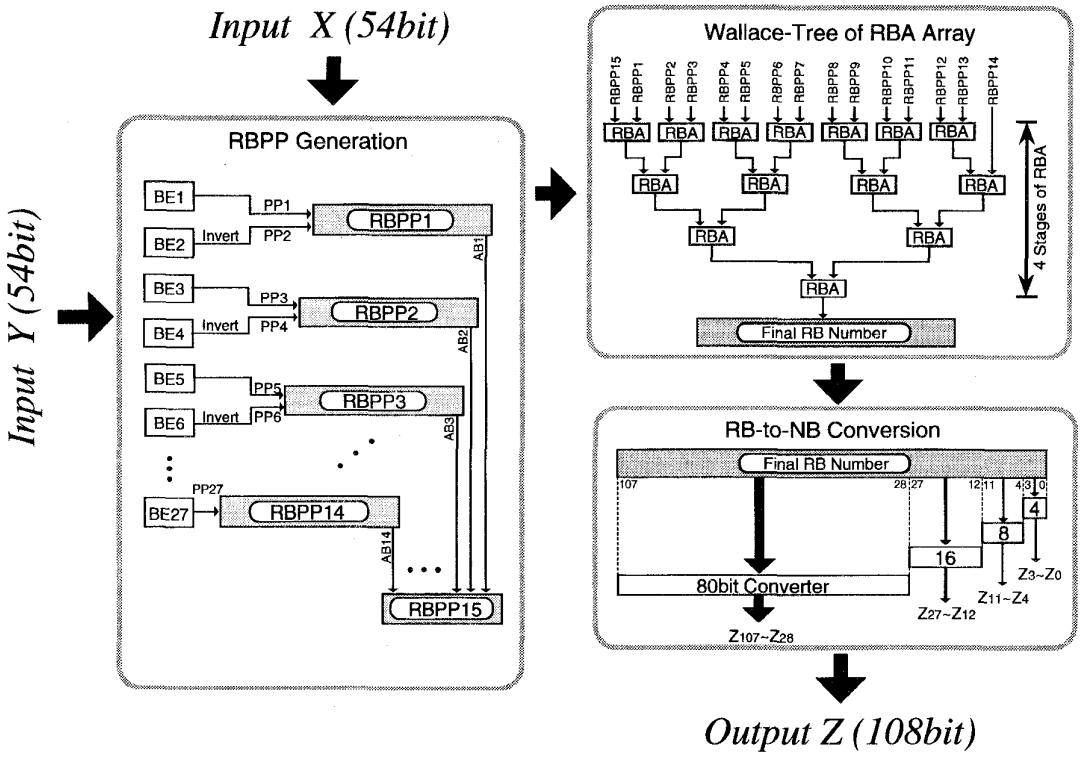


Fig. 6. Block diagram of 54 × 54-bit multiplier. Input multiplicand X and multiplier Y are multiplied to make product Z by high speed RB architecture.

cuits. CONV3 is slower than other methods because the inserted inverters to series TG's causes the increase in circuit stages. Our converter CONV1 has the fastest speed and the least transistor count for more than 16-bit word length. Besides the speed and transistor count, CONV1 has one more advantage that the layout is quite easy, because CONV1 has a regular structure with simple interconnection, while the CLA circuit requires the complicated structure. The easy layout reduces the design cost drastically when the word length becomes large.

III. 54 × 54-Bit Multiplier Design

A 54 × 54-bit multiplier is designed with this RB architecture. Fig. 6 shows the block diagram. There are three main stages. The first is the RB partial product (RBPP) generation stage, the second, the Wallace-tree stage with the improved RBA's, and the third, the RB-to-NB converter stage. In the RBPP generation stage, the second-order Booth's algorithm [17] is used to halve the number of partial products. There are 27 (= 54/2) Booth encoder/selector circuits BEL–27 to generate partial products PPI–PP27. The 13 Booth encoder/selector circuits whose orders are even numbers (BS2, 4, · · · , 26) generate the inverted partial products. The pairs of two adjacent partial products (PP1, PP2), (PP3, PP4), · · · , in which one is a noninverted and the other is an inverted partial product, become RB partial products. The RBPP14 is the exception. It is generated from a single partial product of PP27. An additional RB partial product, RBPP15, is generated from the additional bits of all partial products (AB1–14) that arise from the lowest digits from the inversion of the sign. Thus the number of the RB partial products is 15 (RBPP1–15) that is about 1/4 of the input bit length (= 54).

In the Wallace-tree stage, the 15 RBPP's are added in parallel by the improved RBA's (RBA1's). It should be noted that the additional RB partial product, RBPP15, is firstly added to the RBPP1. This eliminates the undesir-

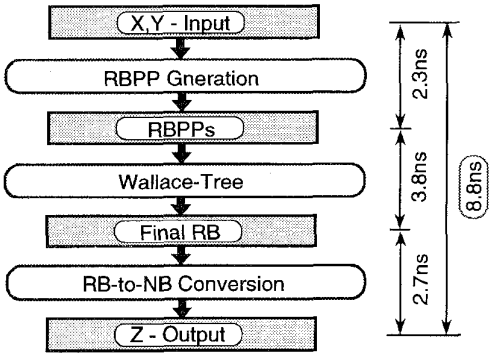


Fig. 7. Simulated delay time of each stage. Total delay of 8.8 ns is obtained.

able increase in hardware, because the position of the lowest digit of RBPP15 is the same as that of RBPP1. The number of RBA stages is four.

In the RB-to-NB conversion stage, the final RB number with 108 digits is converted to an NB number with the same number of digits. This is the product to be calculated. This conversion is carried out by the newly developed converter circuit explained in the above section. The 108 digits of the final RB number are divided into 4, 8, 16, and 80 digits from the lowest digit. Then the conversion is carried out successively in this order because the final RB number is settled in the same order by each stage RBA addition in the Wallace-tree stage. The first two conversions of four and eight digits and the carry generation of 16 digits are finished while the addition is carried out in the Wallace-tree stage. Therefore, the conversion speed of this stage depends on the conversion time of 80 digits. The number of stages of the multiplexer circuits is 12 in the critical path of the 80 bit converter.

We evaluated the operating speed of the multiplier using the SPICE2 simulation. We used the parameter of 0.5-μm CMOS in the simulation. The supply voltage is 3.3 V. The simulation is carried out for the critical path extracted from the whole multiplier circuit. A multiplication time of 8.8 ns is obtained by the simulation. Fig. 7 shows the delay time of each stage in the multiplier. The delay time of the RB partial product generation stage is 2.3 ns including the delays of the input buffer, Booth's encoder, and selector. The delay time in the Wallace-tree stage is 3.8 ns, which consists of four stages of RBA's. In the RB-to-NB conversion stage, the delay time of 2.7 ns is obtained, which is the delay time of 80-bit conversion.

IV. Fabrication and Test

The 54 × 54-bit multiplier is fabricated using 0.5 μm-CMOS technology with triple level metal interconnection. Fig. 8 shows the photograph of the chip. The width/space of first, second, and third metal are 1.2/0.8 μm, 1.2/1.3 μm, and 2.0/1.0 μm, respectively. The first metal is used for the horizontal interconnection and the second metal is used for the vertical interconnection. The third metal is used for the horizontal power supply lines. The

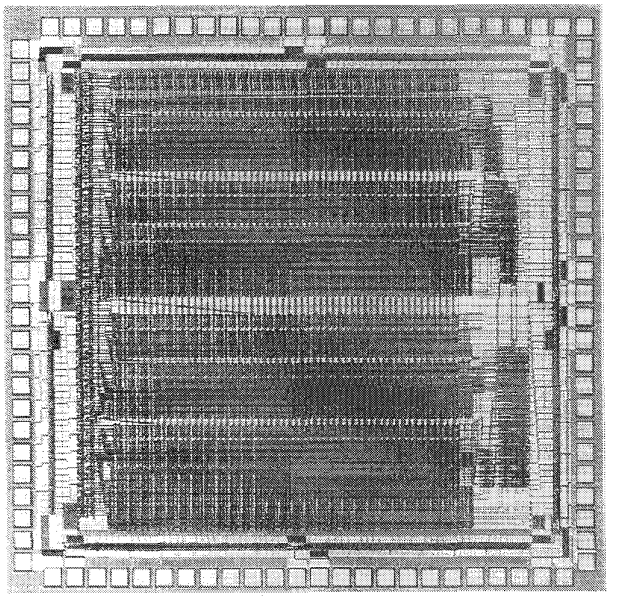


Fig. 8. Photograph of 54 × 54-bit multiplier chip. Active area size is 3.05 × 3.08 mm^2. Transistor count is 78 800.

TABLE V
Comparison of Transistor Count for Defferent 54 × 54-bit Multipliers

MultiplierType	Present Work	Conventional		
		[3]	[4]	[5]
Transistor Count	78,800	81,600	82,500	100,200

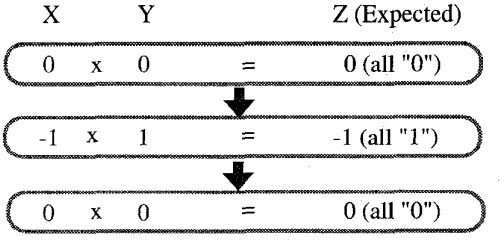


Fig. 9. Critical test pattern.

active area size is 3.05 × 3.08 mm^2. The transistor count is 78,800. Thus the density of transistors is 8.4 k/mm^2. Table V compares this chip with the conventional 54 × 54-bit multipliers [3]–[5] for the transistor count. In this work, the reduction of the transistor count is realized by the simple structure of the architecture.

We measured the performance of the fabricated 54 × 54-bit multiplier. Fig. 9 shows the critical pattern used for the test. First, the X and Y are "0." Then they become "−1" and "1," respectively. Then they change to "0." By this input pattern, the expected output changes from all "0" to all "1," and then to all "0" again because the two's complement number is used. This is the critical pattern. More than ten thousand patterns including the critical pattern are tested. Fig. 10 shows the schmoo plot of the multiplication time and the supply voltage. The

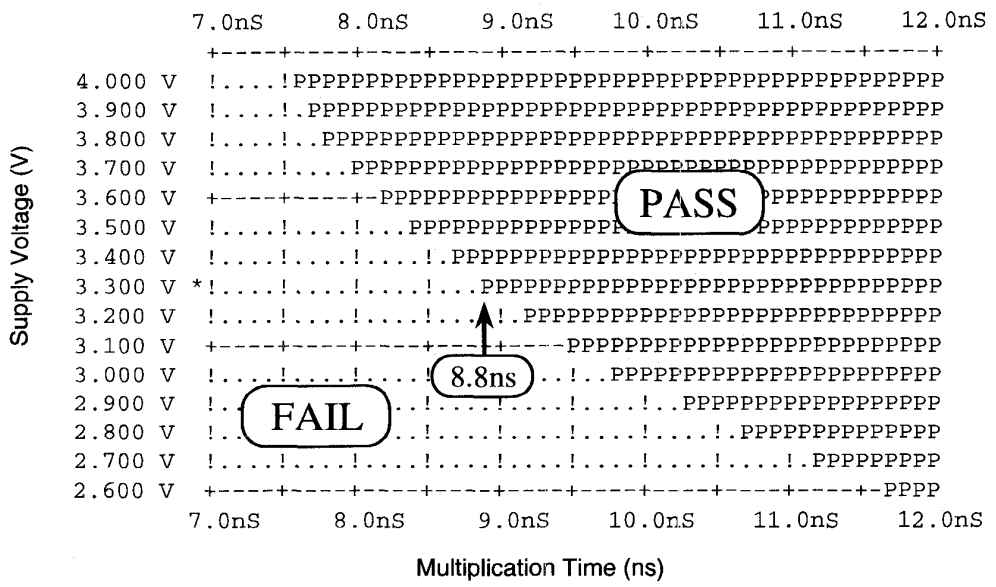


Fig. 10. Schmoo plot of supply voltage vs multiplication time. "P" region indicates that the multiplier pass the test. Multiplication time of 8.8 ns is obtained for 3.3 V supply voltage.

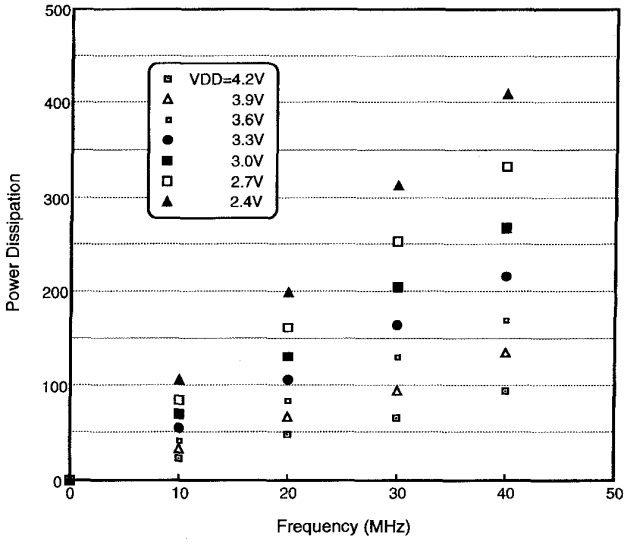


Fig. 11. Measured power dissipation as a function of frequency. Supply voltage is 2.4, 2.7, 3.0, 3.3, 3.6, 3.9, and 4.2 V.

"P" region implies the pass region. Multiplication time for the supply voltage of 3.3 V is 8.8 ns. This is the fastest value ever reported for 0.5-μm CMOS technology. Fig. 11 shows the measured dependence of the power dissipation of the multiplier on the operating frequency. We measured the power dissipation at the supply voltages of VDD = 4.2 V, 3.9 V, 3.6 V, 3.3 V, 3.0 V, 2.7 V, and 2.4 V as a function of operating frequencies of 10 MHz, 20 MHz, 30 MHz, and 40 MHz. The power dissipation increases linearly in accordance with the increase in the frequency. We obtained 216-mW power dissipation for 40 MHz and 3.3 V. The estimated value of the power dissipation for the frequency of 100 MHz and VDD = 3.3 V is 540 mW. It is 38% reduction from the conventional 54 × 54-bit multiplier [3]. This is because the transistor count is reduced by this architecture and wiring capacitance is also reduced by use of triple level metal interconnection. The features of the multiplier are shown in Table VI.

TABLE VI
FEATURES OF 54 × 54-BIT MULTIPLIER

Multiplier, Multiplicand	54b (two's complement)
Product	108b (two's complement)
Supply voltage	3.3V
Multiplication time	8.8ns
Power dissipation (estimated)	540mW (at 100MHz)
Active area size	3.05 x 3.08 mm^2
Transistor count	78,800
Density of transistors (without pad area)	8.4k/mm^2
Process	0.5-μm CMOS Triple metal

V. CONCLUSION

A 54 × 54-bit multiplier is designed and fabricated using the high speed RB architecture. This architecture requires no additional circuit to make RB partial products. The RB partial products are efficiently added up by an array of the improved RBA's, that is faster than the conventional 4-2 compressors. A simple and high speed RB-to-NB converter is developed. The carry propagation circuit in the converter is constructed with only multiplexer circuits. The process technology of 0.5-μm CMOS with triple level metal is used to fabricate the multiplier. The

supply voltage is 3.3 V. The active area size is 3.05 × 3.08 mm^2. The number of transistors is 78 800. These are the smallest figures reported so far for 54 × 54-bit multipliers. The measured multiplication time is 8.8 ns. The power dissipation is 216 mW at the frequency of 40 MHz. 540-mW power dissipation is estimated for 100 MHz. These figures are more than 12% faster speed and 38% lower power than the conventional one with 0.5-μm CMOS. These results prove this RB architecture is effective in the design of a multiplier having high speed and a lower power.

Acknowledgment

The authors would like to acknowledge H. Komiya and Y. Horiba for giving them the opportunity of this work. They also thank S. Kishida and E. Torii for CAD support, A. Maeda for fabricating the chip, and H. Kakiuchi for supporting the test issues.

References

[1] C. S. Wallace, "A suggestion for fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 14–17, Feb. 1964.

[2] M. R. Santoro and M. A. Horowitz, "A pipelined 64 × 64-bit iterative multiplier," *IEEE J. Solid-State Circuits*, vol. 24, pp. 487–493, Apr. 1989.

[3] J. Mori, M. Nagamatsu, M. Hirano, S. Tanaka, M. Noda, Y. Toyoshima, K. Hashimoto, H. Hayashida and K. Maeguchi, "A 10-ns 54 × 54-b parallel structured full array mutliplier with 0.5-μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 26, pp. 600–605, Apr. 1991.

[4] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54 × 54-b regularly structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1229–1235, Sept. 1992.

[5] N. Okubo, M. Suzuki, T. Shinbo, T. Yamanaka, N. Shimizu, K. Sasaki, Y. Nakagome, "A 4.4-ns CMOS 54 × 54-bit multiplier using pass-transistor multiplier," *IEEE Proc. 1994 CICC*, pp. 599-602, May 1994.

[6] Y. Harata, Y. Nakamura, H. Nagase, 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.

[7] S. Kuninobu, T. Nishiyama, H. Edamatsu, T. Taniguchi, and N. Takagi, "Design of high speed MOS multiplier and divider using redundant binary representation," *IEEE Proc. of the 8th Symp. on Computer Arithmetic (ARITH8)*, pp. 80–86, May 1987.

[8] 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.

[9] T. N. Rajashekhara and O. Kal, "Fast multiplier design using redundant signed-digit number," *Int. J. Electronics*, vol. 69, pp. 359–368, 1990.

[10] S. Kuninobu, T. Nishiyama, and T. Taniguchi, "High speed MOS multiplier and divider using redundant binary representation and their implementation in a microprocessor," *IEICE Trans. Electron*, vol. E76-C, pp. 436–445, Mar. 1993.

[11] C. Y. Chow and J. E. Robertson, "Logical design of a redundant binary adder," in *Proc. 4th Symp. on Computer Arithmetic (ARITH4)*, Oct. 1978, pp. 109–115.

[12] T. Nakanishi, H. Yamauchi, and H. Yoshimura, "CMOS radix-2 signed-digit adder by binary code representation," *Trans. IECE Japan*, vol. E69, Apr. 1986.

[13] M. Tonomura, "High-speed digital circuit of discrete cosine transform," SP94-41, DSP94-66, Tech. Rep. IEICE Japan, pp. 39–46, 1994.

[14] T. N. Rajashekhara and A. S. Nale, "Conversion from signed-digit to radix compliment representation," *Int. J. Electronics*, vol. 69, pp. 717–721, 1990.

[15] S. M. Yen, C. S. Laih, C. H. Chen and J. Y. Lee, "An efficient redundant-binary number to binary number converter," *IEEE J. Solid-State Circuits*, vol. 27, pp. 109–112, Jan. 1992.

[16] H. R. Srinivas and K. K. Parhi, "A fast VLSI adder architecture," *IEEE J. Solid-State Circuits*, vol. 27, pp. 761–767, May. 1992.

[17] A. D. Booth, "A signed binary multiplication technique," *Q. J. Mech. Appl. Math. 4*, pp. 236–240, 1951.

Hiroshi Makino was born in Osaka, Japan, in 1959. He received the B.S. degree in physics from Kyoto University, Kyoto, Japan, in 1983.

In 1983 he joined the LSI Research and Development Laboratory, Mitsubishi Electric Corporation, Itami, Japan, where he worked on the research and development of GaAs digital LSI's until 1990. Since 1991, he has been engaged in the research of Si high speed digital circuits in System LSI Laboratory. He received the Best Paper Award at IEEE International Conference on Computer Design in 1993.

Mr. Makino is a member of the Institute of Electronics, Information, and Communication Engineers of Japan.

Yasunobu Nakase received the B.S. and M.S. degrees in physics from Ritsumeikan University, Kyoto, in 1981 and 1983, respectively.

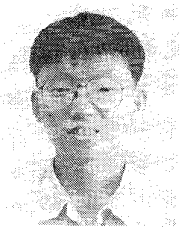
He joined the System LSI Laboratory, Mitsubishi Electric Corporation, Hyogo, Japan, in 1983. From 1983 to 1990 he worked on the development of high-speed bipolar ECL static RAM's from 1 K-b to 16 K-b. Since 1990 he has been involved in the research and development of high-speed CMOS/BiCMOS logic LSI's and CMOS embedded SRAM's. He received the Best Paper Award at IEEE International Conference on Computer Design in 1993.

Mr. Nakase is a member of the Institute of Electronics, Information, and Communication Engineers of Japan.

Hiroaki Suzuki was born in Wakayama, Japan, in 1967. He received the B.S. and M.S. degrees in electrical engineering from Osaka Institute of Technology, Osaka, Japan, in 1989 and 1991, respectively.

He joined the LSI Laboratory, Mitsubishi Electric Corporation, Hyogo, Japan, in 1991. He has been engaged in the research and development of high-speed CMOS/BiCMOS logic LSI's. He currently belongs to the System LSI Laboratory, Mitsubishi Electric Corporation, Hyogo, Japan.

Mr. Suzuki is a member of the Institute of Electrical Engineers of Japan.



Hiroyuki Morinaka received the B.S. degree in physics from Osaka University, Osaka, Japan, in 1991.

In 1991 he joined the LSI Laboratory of Mitsubishi Electric Corporation, Itami, Japan, where he was engaged in the research and development of the design of high speed CMOS digital circuits. He currently belongs to the System LSI Laboratory.

Hirofumi Shinohara was born in Hyogo, Japan, on January 2, 1954. He received the B.S. and M.S. degrees in electronic engineering from Kyoto University, Kyoto, Japan, in 1976 and 1978, respectively.

In 1978 he joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan. From 1978 to 1987 he worked on the development of high-density and high-speed MOS static RAM's. From 1987 to 1993 he was involved in the areas of ASIC including module generators, datapath compilers, neural network LSI's and high-speed data processing LSI's. He is currently a staff of corporate R&D planning and administration department, Tokyo, Japan.

He received the Best Paper Award at the IEEE International Conference on Computer Design in 1993.

Koichiro Mashiko (M'88) was born in Shizuoka, Japan, in 1952. He received the B.S. and M.S. degrees in physics from the University of Tokyo, Tokyo, Japan, in 1975 and 1977, respectively. In 1988 he received the Ph.D. degree in electrical engineering from Osaka University, Osaka, Japan.

In 1977 he joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Hyoto, Japan. There he was engaged in the research and development of dynamic RAM's from 64 k-b to 4 M-b 256 K-b dual-port Video RAM, Cache Controller Chips, Neural Network Chips, and so on. In 1990 he transferred to the Headquarters R&D, Mitsubishi Electric Corporation, Tokyo, Japan. There he worked on a scheme of establishing a laboratory of computer science in the USA. In 1991 the laboratory of Mitsubishi Electric Research Laboratories, Inc., was established in Cambridge, MA. He worked on the operation there from 1991 to 1993. In 1993 he transferred to System LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan. Currently he is managing the research and development of LSI circuit technologies for high-speed, low-power, low voltage applications.