A 600-MHz 54×54-bit Multiplier with Rectangular-Styled Wallace Tree

Niichi Itoh, Yuka Naemura, Hiroshi Makino, Yasunobu Nakase, Tsutomu Yoshihara, and Yasutaka Horiba

***Abstract*—This paper presents an efficient layout method for a high-speed multiplier. The Wallace-tree method is generally used for high-speed multipliers. In the conventional Wallace tree, however, every partial product is added in a single direction from top to bottom. Therefore, the number of adders increases as the adding stage moves forward. As a result, it generates a dead area when the multiplier is laid out in a rectangle.**

To solve this problem, we propose a rectangular Wallace-tree construction method. In our method, the partial products are divided into two groups and added in the opposite direction. The partial products in the first group are added downward, and the partial products in the second group are added upward. Using this method, we eliminate the dead area. Also, we optimized the carry propagation between the two groups to realize high speed and a simple layout. We applied it to a 54×54-bit multiplier. The 980 μm × 1000 μm area size and the 600-MHz clock speed have been achieved using 0.18-μm CMOS technology.

***Index Terms*—CMOS digital integrated circuits, multiplication, multiplying circuits, redundant binary, Wallace tree.**

I. Introduction

BECAUSE of the high demand of three-dimensional computer graphics (3DCG), high-speed floating-point (FP) processing is required even in low-cost personal computers. Especially, multimedia applications such as games and audiovisual need to handle 3DCG at very low cost. One solution to this problem is to reduce the IC costs. One way to realize the cost reduction of IC is to minimize the area. Among various FP constructions, multiplication is critical for both speed and area. Therefore, the reduction of the area size of a high-speed multiplier is effective for the cost reduction.

The Wallace-tree method [3] is one of the best solutions to construct the high-speed multiplier (MPY), because this method adds the partial products in parallel. However, its physical design is rather difficult because of the complex interconnections. Furthermore, this method results in dead area when the MPY is arranged in a rectangle. For the alleviation of complex interconnections, many studies have been reported, such as the 4–2 compressor [4]–[6], redundant binary architecture [1], [11]–[15], and so on. On the other hand, there have been no effective solutions to solve the dead area problem. About this problem, we propose a construction method of a Wallace tree that reduces the area size with a simple layout. We present the rectangular-styled Wallace-tree construction method. Our application is a 54×54-bit MPY design. In this paper, a conventional Wallace-tree MPY construction method is described in Section II. The problem of the conventional method is discussed in Section III. In Sections IV and V, we show the proposed Wallace-tree construction method and its effect on the MPY design. Then the design of 54×54-bit MPY using the new method is described in Section VI. Section VII shows the result of the design. Finally, Section VIII concludes this paper.

II. Conventional Wallace-Tree MPY Construction Method

The functions of a conventional MPY are divided into three stages. The first stage is the generation stage of the partial products. The next stage is the adding stage of the partial products. The last stage is the final addition.

The first stage generates the partial products that are obtained from the multiplicand multiplied by each bit of the multiplier. In this stage, the MPY generally uses the Booth algorithm to reduce the number of the partial products [2]. By using the second-order Booth algorithm, the number of partial products is reduced almost to half the bit width of the multiplier [4]–[6].

The second stage sums up the partial products. They are added up repeatedly until two partial products are obtained. In the case of low-speed MPYs, the partial products are added serially by using a carry–save adder [16]. This method has the advantage that the layout is easy because of its regular structure. However, it cannot operate at high speed, especially for large bit widths, because the delay time of the carry–save method is in proportion to the number of bits of a multiplier. In the case of high-speed MPYs, the Wallace-tree method is usually used.

The Wallace-tree method adds the partial products in parallel using multiple full-adders. This method is fastest theoretically [3], because its calculation time is in proportion to the logarithm of the number of bits of a multiplier. Although the Wallace tree has a speed advantage, its layout is complicated and the design cost increases. Because the signal connection wires connect the three-input/two-output full-adders, the layout is very complicated and difficult. To solve this problem, it is effective to use the four-input/two-output adder (4–2 compressor) [4]–[6] in place of full-adders. The 4–2 compressor simplifies the wire connection drastically compared with the conventional Wallace tree using full-adders. As a result, the design cost is reduced.

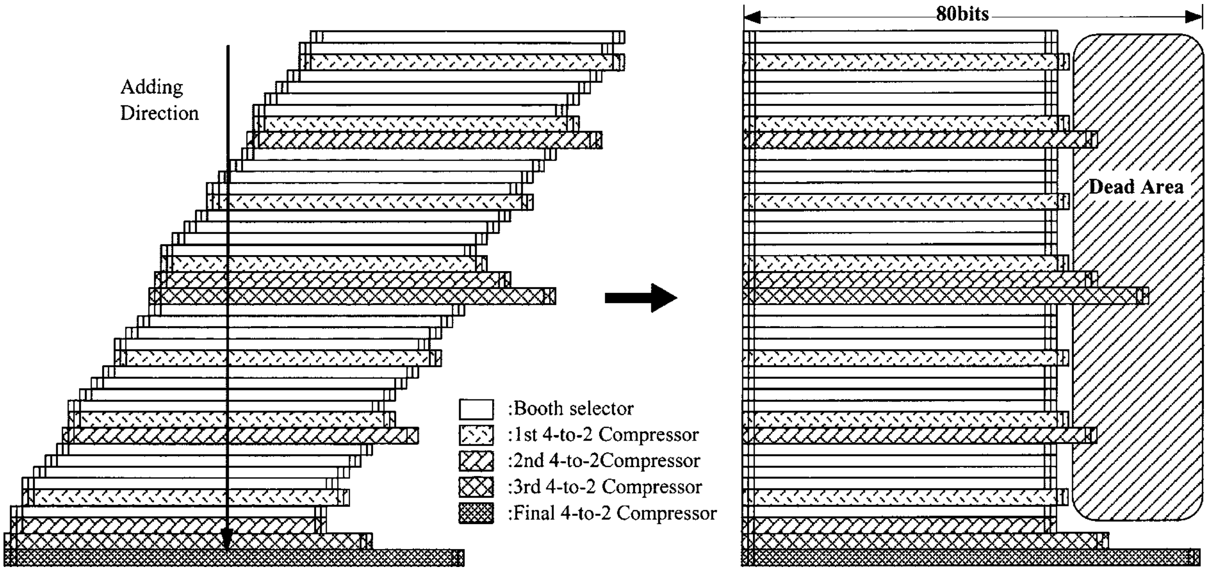
The third stage generates the final multiplication result. In this stage, any kind of adder (ripple-carry adder [7], carry-look-

Manuscript received February 1, 2000; revised October 18, 2000. This paper was presented at the 1999 Symposium on VLSI Circuits, Kyoto, Japan.

N. Itoh, H. Makino, Y. Nakase, T. Yoshihara, and Y. Horiba are with the Advanced Circuit Design Group, System LSI Development Center, Mitsubishi Electric Corporation, Hyogo 664-8641, Japan (e-mail: n_itoh@lsi.melco.co.jp).

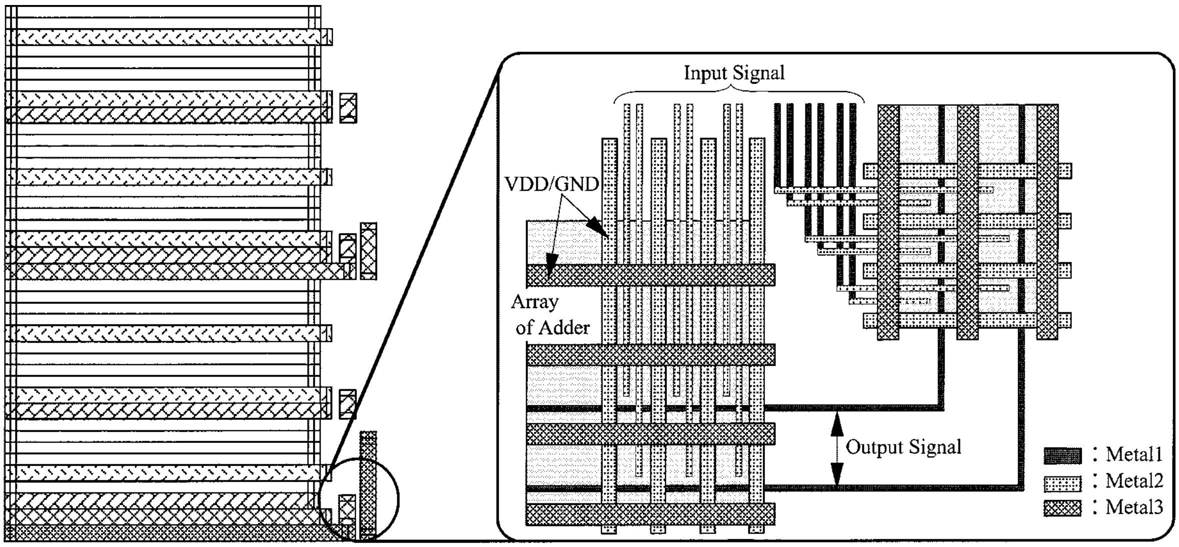
Y. Naemura is with the LSI Design Center, Mitsubishi Electric Engineering Company Limited, Hyogo 664-0004, Japan.

Publisher Item Identifier S 0018-9200(01)00932-5.



(a) (b)

Fig. 1. Conventional layout construction of 54×54-bit multiplier. (a) Before arrangement. (b) After arrangement.



(a) (b)

Fig. 2. Conventional layout optimization of 54×54-bit multiplier. (a) Optimized layout of conventional Wallace tree. (b) Bending point of 4–2 compressor.

ahead adder [8], carry–skip adder [9], carry–select adder [10], etc.) is applicable as a carry propagation adder.

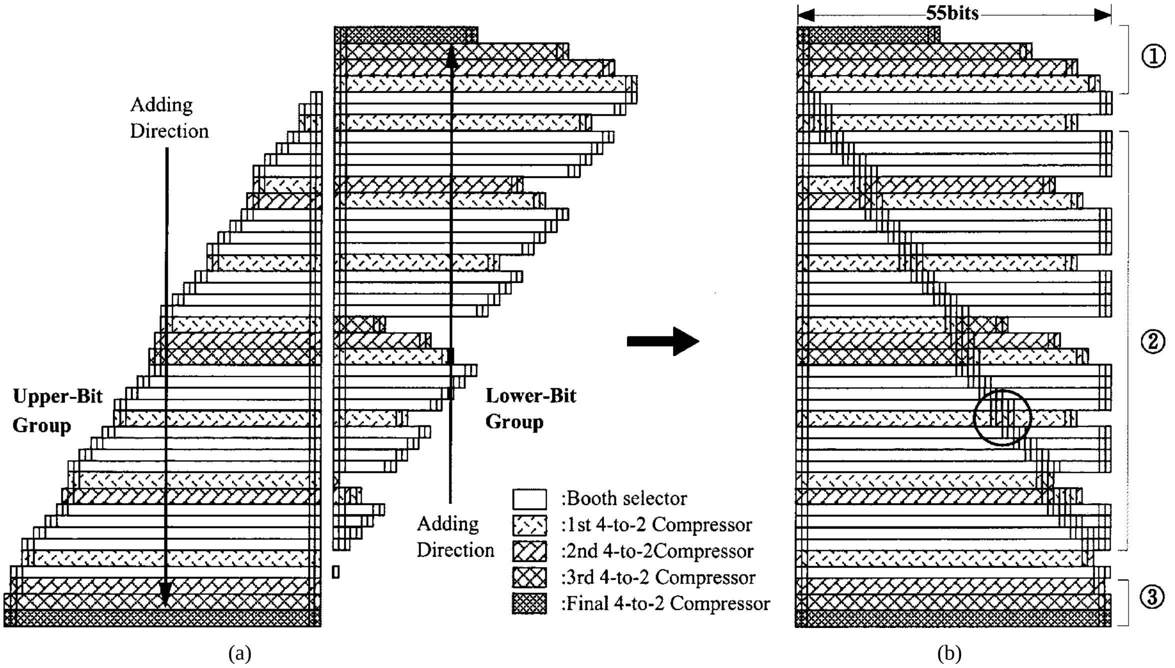
III. Problem of Wallace-Tree Consruction Method

Fig. 1 shows an example of a 54×54-bit MPY implemented with a conventional Wallace-tree structure. The second-order Booth algorithm is assumed to reduce the number of partial products. The 4–2 compressors are also assumed for the fast multiplication. This construction is popular among high-speed multipliers [4]–[6].

As shown in Fig. 1(a), the Wallace tree consists of Booth selectors and 4–2 compressors. In a conventional Wallace tree, every partial product is added in one direction from top to bottom. Therefore, the number of 4–2 compressors increases as the adding stage moves downward. As a result, the number of adders becomes 80 bits at the final stage.

Fig. 1(b) shows the layout of conventional MPY when it is arranged in a rectangle style. As shown in this figure, the increased bits of 4–2 compressors are stuck out at the right side of the MPY. The signal wires that connect the stuck-out 4–2 compressors extend into this area. Therefore, it is difficult to place other circuits near these 4–2 compressors. This area will be empty and it becomes a dead area. It wastes silicon area and increases the chip cost.

To reduce the dead area, conventional MPY optimizes the layout as shown in Fig. 2(a). In this layout, increased 4–2 compressors are manually arranged in the open space of the right side of MPY [4], [6]. Although this layout is effective to reduce the area, it requires a longer time for design than the conventional method in Fig. 1. The metal layer of the interconnection wires of the right side of the MPY needs to change at the bending point of the 4–2 compressors, as shown in Fig. 2(b). Therefore, this approach is complicated and it increases the design cost.



(a) (b)

Fig. 3. Proposed layout construction of 54×54-bit multiplier. (a) Before arrangement. (b) After arrangement.

IV. Rectangular-Styled Wallace-Tree Construction Method

To solve the above problem, we propose a new Wallace-tree construction method. Fig. 3 shows an example of this applied to a 54×54-bit MPY. As shown in Fig. 3(a), the partial products are divided into two groups by an appropriate digit around the center of the Wallace tree. One is Upper-bit group and the other is Lower-bit group. Then, the partial products of each group are added in the opposite directions, respectively. For example, the Upper-bit group is added downward, and the Lower-bit group is added upward. This Wallace tree can be arranged in a rectangle as shown in Fig. 3(b). By using this method, the dead area on the right side of MPY is eliminated. The horizontal bit width, 55 bits, is the same as those of the partial products.

In this method, because the partial products are added in opposite directions, it needs an extra four rows of 4–2 compressors at the top of the MPY [① of Fig. 3(b)]. Although these 4–2 compressors slightly increase the area of MPY, this area consumption is much smaller than the dead area of conventional MPY. The quantitative estimation of the area is described in the next section.

This method has another advantage, namely, the simplicity of the layout. Fig. 4 shows the signal flow of the proposed MPY. This figure is the part of the circled area in Fig. 3(b). As shown in this figure, the layout of every signal from the Booth encoder is the same as a conventional method. These signals are routed in a horizontal direction over the array of the Booth selectors. Also, every signal generated by the Booth selectors, which is the partial product, flows in the same way as in the conventional method. The only difference is the input/output signal direction of the 4–2 compressors. These signals flow in the opposite direction between the Upper-bit group and the Lower-bit group. Although these signals flow in opposite directions, the physical routing topology is the same as the conventional method. Therefore, the layout is as easy as the conventional MPY shown in Fig. 1. It does not require a complicated arrangement of the increased 4–2 compressors.

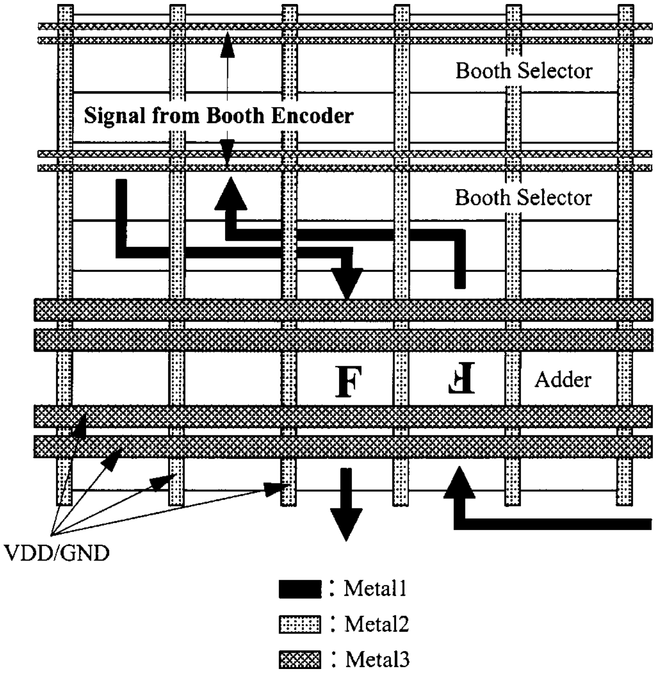


Fig. 4. Signal flow of proposed multiplier.

In this way, our construction method eliminates the dead area without requiring the complicated arrangement of 4–2 compressors. Furthermore, the layout complexity is the same as a conventional method shown in Fig. 1.

V. Effects of New Method

We estimated the effects of the proposed method in the horizontal width and the area size. Fig. 5 shows the width size

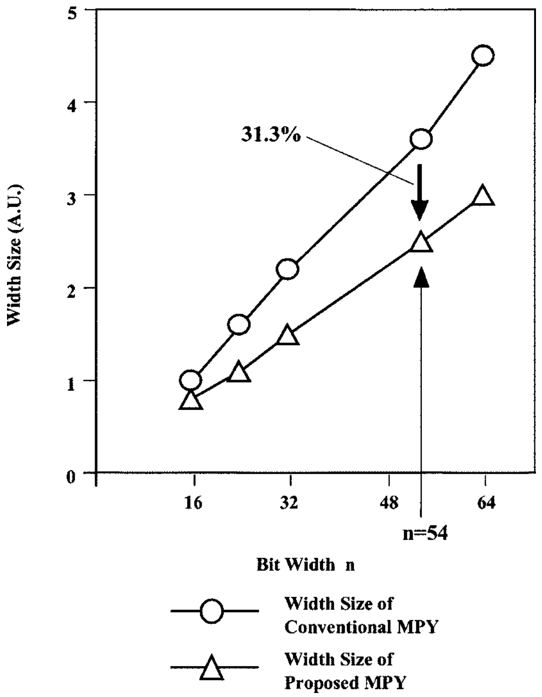


Fig. 5. Comparison of layout width.

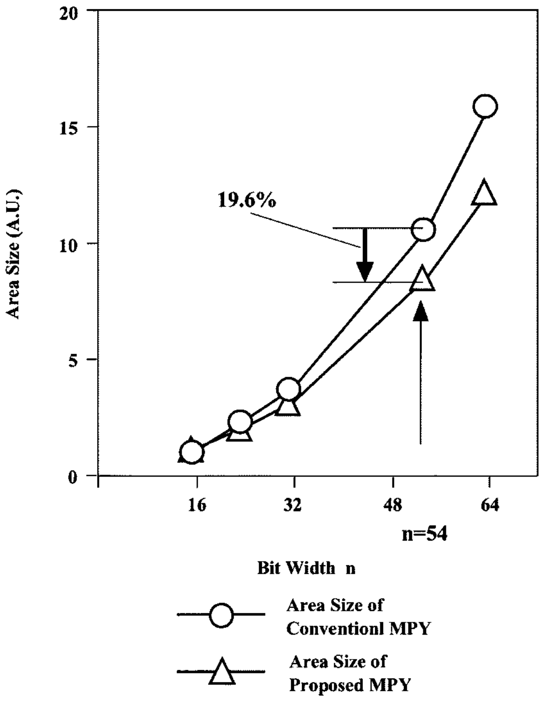


Fig. 6. Comparison of layout area.

comparison between the conventional and the proposed method. Vertical axis indicates the relative size compared with conventional 16×16-bit MPY. The circles show the sizes of conventional MPYs, and the triangles show the sizes of the proposed method. The width sizes of the proposed MPYs are smaller than conventional ones. The difference extends as the bit width increases. In the case of $n = 54$, the difference becomes 31.3% of a conventional MPY.

Fig. 6 shows the area comparison. The vertical axis also indicates the relative area compared with conventional 16×16-bit MPY. The circles show the areas of conventional MPYs, and the triangles show the sizes of the proposed MPYs. As for the area, the proposed MPYs are also smaller than conventional ones. The size difference is somewhat smaller than that of the width size. Because the proposed method needs the additional four rows of the 4–2 compressors at the top of the Wallace tree, as shown in Fig. 3, these adders slightly increase the vertical size. In the case of $n = 54$, the area is 19.6% smaller than the conventional MPY.

From these results, the proposed method enables us to construct a compact MPY compared with a conventional method.

VI. Design of 54×54-bit MPY

We designed a 54×54-bit MPY using the proposed construction method of the Wallace tree. The high-speed redundant binary (RB) architecture [1] was used to realize high-speed operation. By using the RB for the partial product, the high-speed addition is possible because the addition of RB requires no continuous carry propagation. Furthermore, it makes the wiring easy because the RB adder functions as a 4–2 compressor. Although the RB architecture needs the conversion from the redundant binary (RB) to the normal binary (NB) in the final stage, any kind of carry–propagation adder is usable as the RB-to-NB converter. Therefore, there is no area penalty in using the RB architecture.

Fig. 7 shows the block diagram of the 54×54-bit MPY. This MPY operates with two pipeline stages. The first stage includes the Booth encoders, the proposed Wallace tree, and the 28-bit redundant RB-to-NB converter. The RB-to-NB converter consists of a carry–select adder. This converter also works as the final carry–propagation adder of the conventional Wallace tree. The second stage consists of the extra RB adder and the 80-bit RB-to-NB converter. This extra RB adder adds the carry signal from the top of the MPY to the lower partial product. We divided the RB-to-NB converter into two parts, 28 bits in the first stage and 80 bits in the second stage, to minimize the carry-signal delay. Also, we optimized the circuit arrangement and carry-signal flow to minimize the critical path delay. Fig. 8 shows the carry-signal paths of the RB adders before optimization. This figure shows the area ② in Fig. 3. In our construction method, every partial product is added in opposite direction between the Upper-bit group and Lower-bit group. In this scheme, every carry signal generated in each adding stage in the Lower-bit group is passed to the corresponding adding stage in the Upper-bit group. Therefore, if we connect the carry signal normally, carry-signal paths become complicated, as shown in Fig. 8. For example, in the Lower-bit group, the partial products that are generated by the Booth selectors (A) are added in the first RB adders (B). In the Upper-bit group, however, the corresponding partial products that are generated by the Booth selectors (C) are added in the first RB adders (D). Therefore, the carry-signal path is (E), as shown in Fig. 8.

As for the second RB adders, the carry-signal path becomes much longer. In the Lower-bit group, the results of the first RB adders (F) and (G) are added by the second RB adders (H). In the Upper-bit group, however, the results of the first RB adders

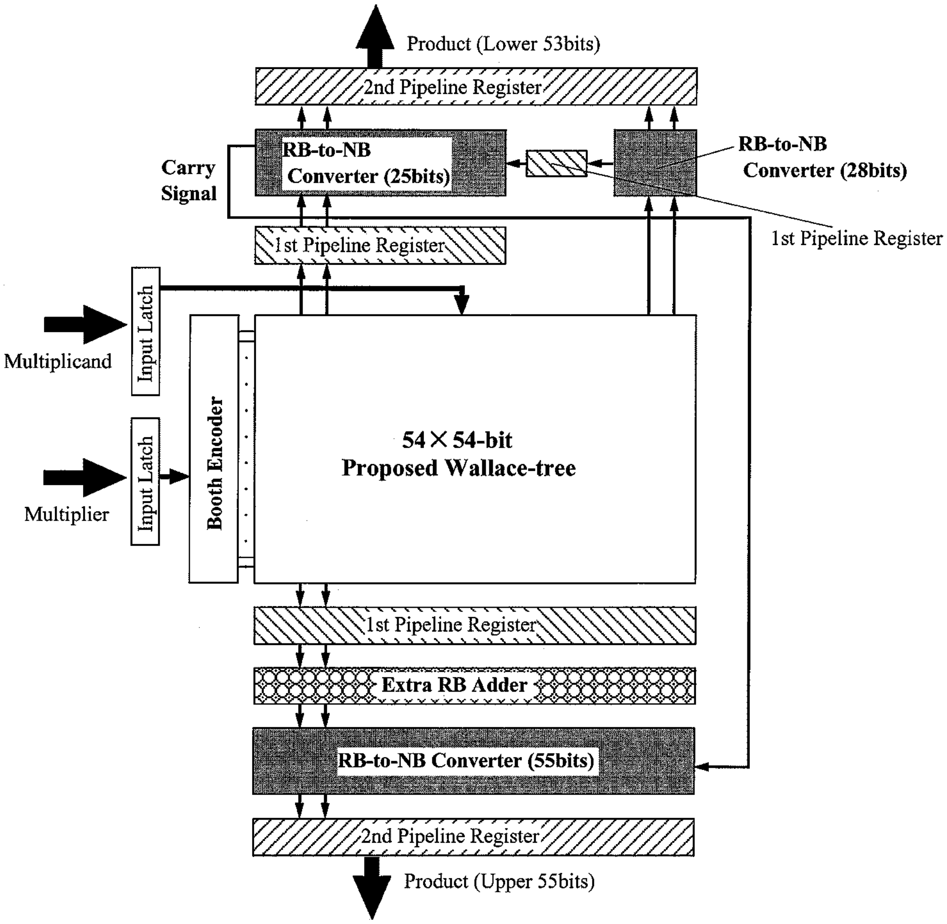


Fig. 7. Block diagram of 54×54-bit multiplier.

(I) and (J) are added by the second RB adders (K). Therefore, the carry-signal path is (L), as shown in Fig. 8.

Due to these carry-signal connections, the layout becomes very complicated. Also, such long carry paths decrease the operating speed of the MPY.

To solve these problems, we connect the carry signals, as shown in Fig. 9. In our method, every carry signal is connected to the nearest RB adder in the same adding stage between the Upper-bit group and the Lower-bit group. For example, the carry signal which is generated by the first RB adder (M) goes to the first RB adder (N), the carry signal from the second RB adder (O) goes to the second RB adder (P), and so on. Although the carry-signal flows are quite different from the conventional MPY, the result of the multiplication is the same. Because all carry signals from the Lower-bit group (such as C1, C2, C3,. . .,C9) are finally added to the lowest digit in the Upper-bit group, only the order of addition is different. In general, the difference of the order of addition does not affect the result because the addition satisfies the commutative and associative low. By using this method, we can avoid the long carry paths. Therefore, the layout becomes simple and the operating speed does not decrease.

In our construction method, we need to solve another problem. Because the partial products are added to upward in the Lower-bit group, the carry signals come from the RB-to-NB converter and four rows of RB adders at the top of MPY. They should be connected to the corresponding ones in the bottom part of the MPY. Therefore, when these signals are connected normally, they become the critical paths and decrease the operating speed. To solve this problem, we optimized the carry-signal propagation. Fig. 10 shows the carry-signal flow from the top RB adder to the bottom one in the Wallace tree. This figure shows the areas ① and ③ in Fig. 3. There are long carry-signal paths from the left-top side to the right-bottom side. To hide the signal delays of these paths, carry signals from the left-top RB adders go to the right-bottom RB adders in the next adding stages. The carry signal from the first RB adder in ① goes to the second RB adder in ③. The carry signal from the second RB adder in ① goes to the third RB adder in ③, and so on. The carry signal from the final RB adder is stored in the pipeline register, and then is added to the result of the final RB adder in the bottom side. This addition is carried out by the extra RB adder in the second pipeline stage.

This connection method enables the carry signals to propagate before the arrival of the adding results of RB adders at the bottom of the Wallace tree. By using this method, these carry paths do not become the critical paths. Therefore, this method prevents the degradation of operating speed. Although this method needs the extra RB adders, the impact on the area is small. The increase of the area is 2.2%.

The carry signal of the RB-to-NB converter (A) in the second pipeline stage is also connected from left-top side to

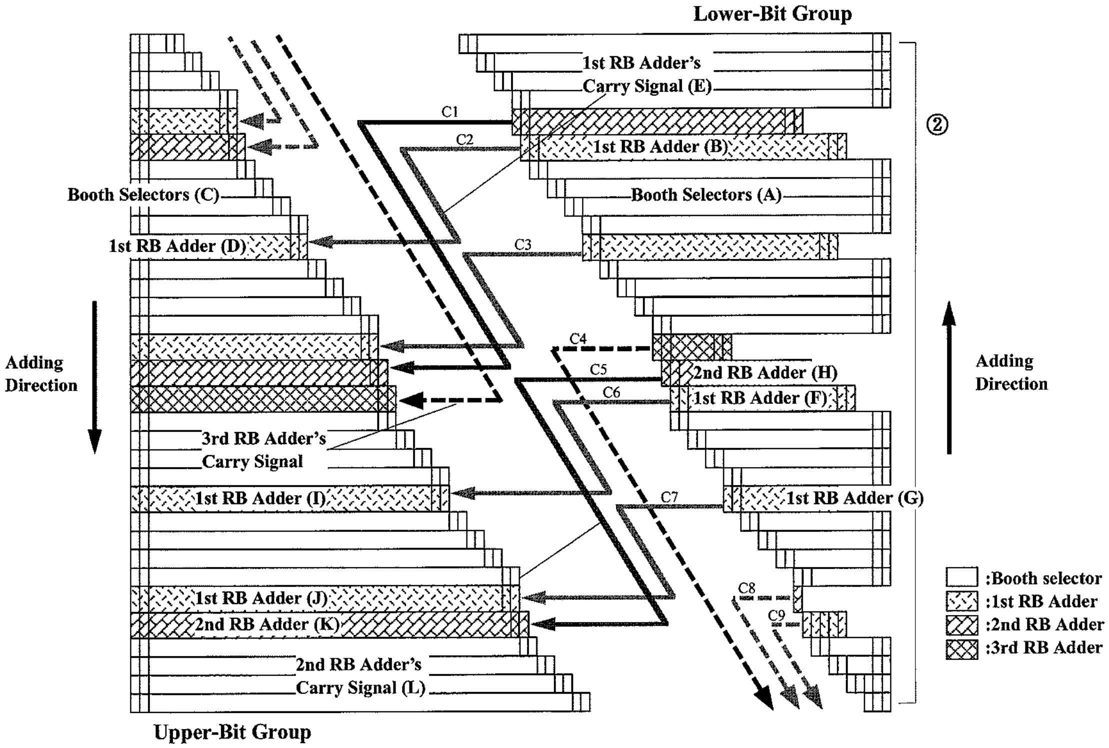


Fig. 8. Carry-signal connection (before optimization).

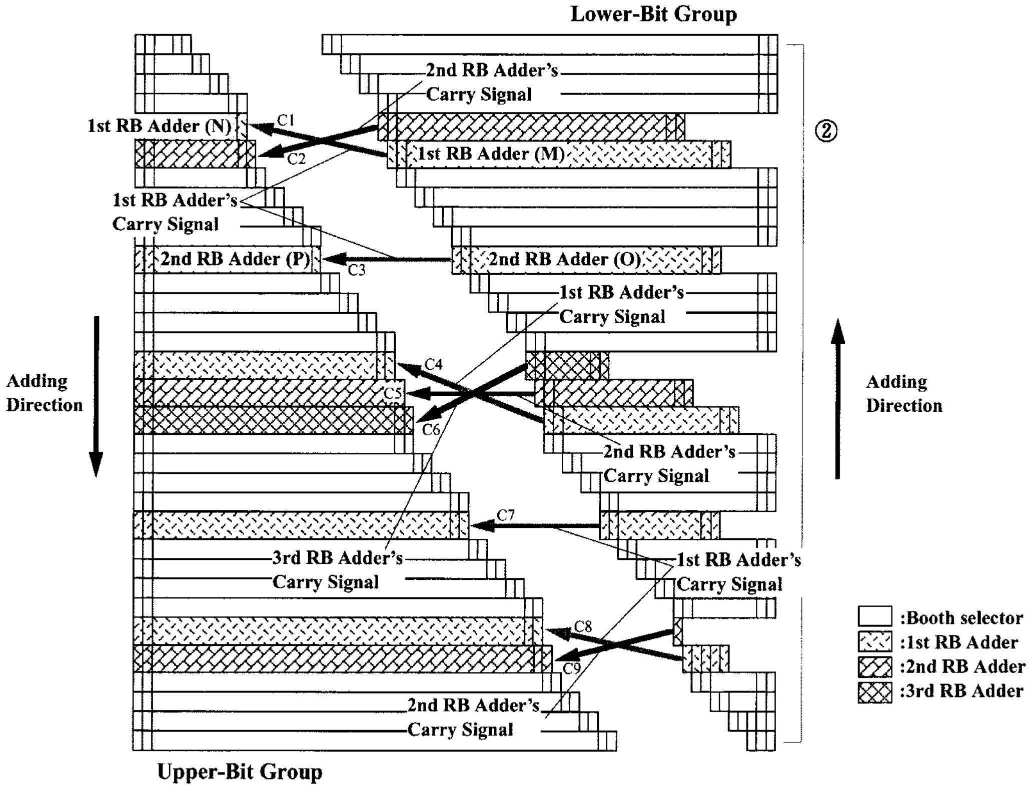


Fig. 9. Carry-signal connection (after optimization).

right-bottom side. This carry signal becomes the critical path in the second pipeline stage. Its delay time, however, has enough margin compared with the delay time of the first pipeline stage, as shown in the next section. Therefore, this signal does not affect the operating speed of MPY.

VII. Result

Fig. 11 shows the layout of the 54×54-bit MPY with a 0.18-μm four-metal-layer CMOS process technology. The metal pitches are 0.8 μm for the first, second, and third metal,

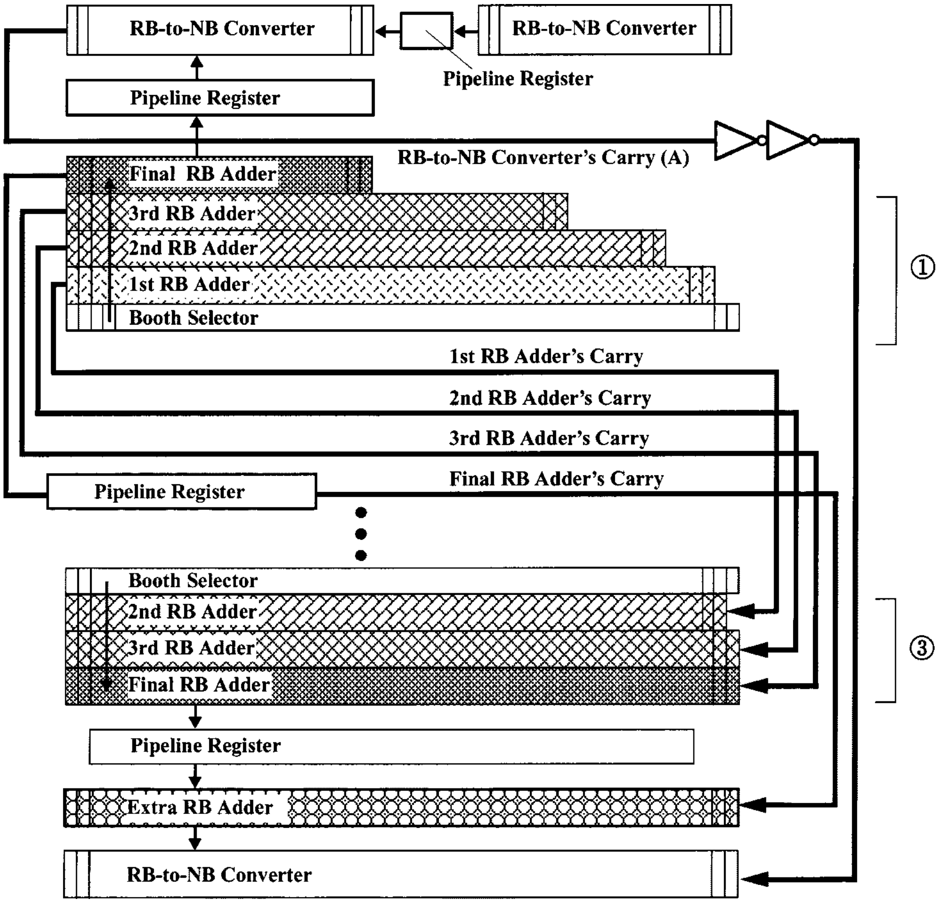


Fig. 10. Carry-signal connection (top and bottom).

TABLE I
CIRCUIT DELAY OF CRITICAL PATH

	1st Stage					2nd Stage			
	Path1	Path2	Path3	Path4	Total	Path5	Path6	Path7	Total
Delay Time (ps)	105	369	76	1030	1580	489	112	599	1200

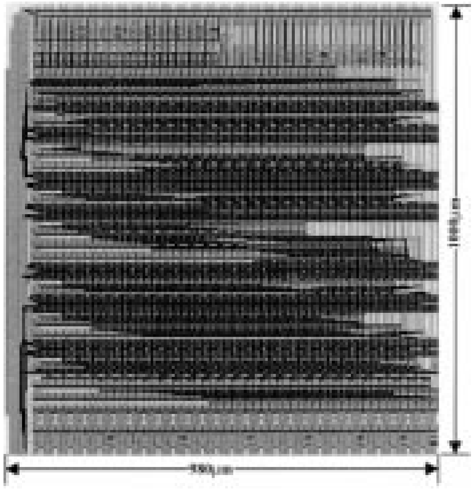
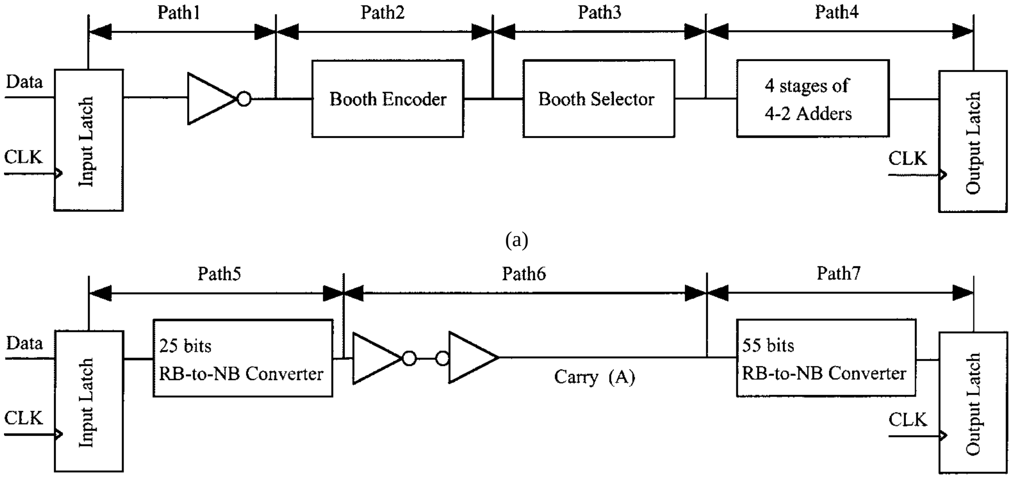


Fig. 11. Layout of 54×54-bit multiplier.

respectively, and 1.4 μm for the fourth metal. The area size is 980 μm × 1000 μm. It is the smallest 54×54-bit multiplier ever reported.

Fig. 12 shows the block diagram of the critical paths of the two pipeline stages. Table I shows the circuit delay time of each path obtained by SPICE simulation. As for the simulation condition, the maximum currents of the nMOS and pMOS transistors are $Idn = 660$ μA/μm and $Idp = -300$ μA/μm, respectively, at the temperature of 27 °C. Supply voltage is 1.8 V. At the first pipeline stage, the critical path consists of the Booth encoder, the Booth selector, and four stages of RB adders. This path's delay is 1.58 ns. At the second pipeline stage, critical path consists of the 25-bit RB-to-NB converter, the long interconnection of carry (A) in Fig. 10, and the 55-bit RB-to-NB converter. This path's delay is 1.20 ns. Although the second pipeline stage has the long carry connection, the worst critical path exits in the first pipeline stage. Therefore, the long carry path in the second pipeline stage does not affect the operating speed. In other words, our method has no disadvantage in speed compared with the conventional method if the two-stage pipeline scheme is used.

Fig. 13 shows the simulated waveform of the input and output signal of the critical path in the first pipeline stage. This path is the worst critical path of the MPY, and its delay time (1.58 ns)



(a)

(b)

Fig. 12. Block diagram of critical path. (a) First stage. (b) Second stage.

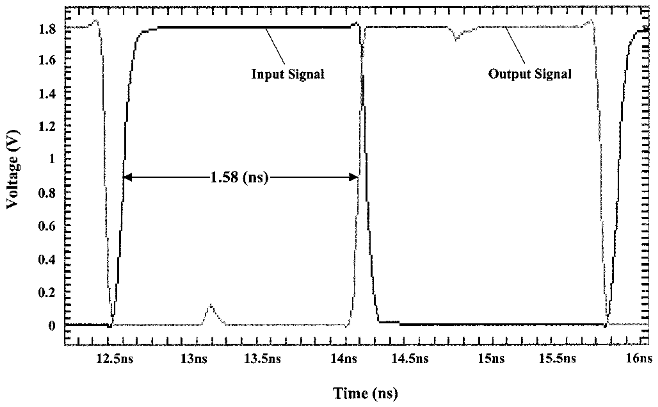


Fig. 13. Simulated waveform of critical path.

defines the MPY's operating speed. Therefore, this multiplier can operate at up to 600 MHz.

VIII. Conclusion

We proposed a compact 54×54-bit multiplier with an improved Wallace-tree structure. This construction method simplifies the layout and eliminates the dead area that is inevitable in conventional methods. The 980 μm × 1000 μm area size and 600-MHz clock cycle have been achieved using a 0.18-μm CMOS process technology. This method is effective for realizing a compact and high-speed multiplier.

Acknowledgment

The authors would like to thank Dr. S. Iwade and Dr. Y. Matsuda for the opportunity of this work. They also thank T. Itoh for CAD support.

References

[1] H. Makino *et al.*, "An 8.8-ns 54×54-bit multiplier with high-speed redundant binary architecture," *IEEE J. Solid-State Circuits*, vol. 31, pp. 773–783, June 1996.

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

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

[4] J. Mori *et al.*, "A 10-ns 54×54-b parallel structured full array multiplier with 0.5-μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 26, pp. 600–605, Apr. 1991.

[5] G. Goto *et al.*, "A 54×54-b regularly structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1229–1235, Sept. 1992.

[6] N. Ohkubo *et al.*, "A 4.4 ns CMOS 54×54-b multiplier using pass-transistor multiplier," *IEEE J. Solid-State Circuits*, vol. 30, pp. 251–257, Apr. 1995.

[7] B. Gilichrist *et al.*, "Fast carry logic for digital computers," *IRE Trans. Electron. Comput.*, vol. EC-4, pp. 133–136, 1995.

[8] A. Winberger *et al.*, "A logic for high-speed addition," *Nat. Bur. Stand. Circ.*, no. 591, p. 312, 1958.

[9] V. G. Oklobdzjia *et al.*, "Some optimal schemes for ALU implementation in VLSI technology," in *Proc. 7th Symp. Computer Arithmetic*, 1985, pp. 2–8.

[10] J. Sklansky, "Conditional-sum addition logic," *IRE Trans. Electron. Comput.*, vol. EC-9, pp. 226–231, 1960.

[11] Y. Harata *et al.*, "A high-speed multiplier using a redundant binary adder tree," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 28–34, Feb. 1987.

[12] S. Kuninobu *et al.*, "Design of high speed MOS multiplier and divider using redundant binary representation," in *IEEE Proc. 8th Symp. Computer Arithmetic (ARITH8)*, May 1987, pp. 80–86.

[13] H. Edomatsu *et al.*, "A 33MFLOPS floating point processor using redundant binary representation," in *ISSCC Dig. Tech. Papers*, Feb. 1988, pp. 152–153.

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

[15] S. Kuninobu *et al.*, "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.

[16] M. Hatamimian and G. L. Cash, "A 70-MHz 8-bit×8-bit parallel pipelined multiplier in 2.5-μm CMOS," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 505–513, Aug. 1986.

Niichi Itoh was born in Aichi, Japan, in 1968. He received the B.S. degree in electrical engineering from Osaka Sangyo University, Osaka, Japan, in 1991.

In 1991, he joined the Kita-Itami Works, Mitsubishi Electric Corporation, Hyogo, Japan, where he worked on the research and development of ASIC LSIs until 1993. Since 1993, he has been engaged in the research and development of CAD tools of ASICs in the ASIC Engineering Center and System LSI Laboratory. From 1996 to 1998, he worked on the development of Alpha 21164pc microprocessor, which was a joint project between Digital Equipment Corporation and Mitsubishi Electric Corporation. His current research interests are high-speed and low-power digital circuits in the System LSI Development Center.

Yuka Naemura received the B.S. degree in mathematics from Osaka Women's University, Osaka, Japan, in 1998.

In 1998, she joined the LSI Design Center, Mitsubishi Electric Engineering Corporation, Hyogo, Japan. Since then, she has been engaged in the research and development of ASICs.

Hiroshi Makino was born in Osaka, Japan, in 1959. He received the B.S. degree in physics from Kyoto University, Kyoto, Japan, in 1983 and the Ph.D. degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1997.

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 LSIs until 1990. Since 1991, he has been engaged in the research and development of Si CMOS high-speed and low-power digital circuits in the System LSI Laboratory and System LSI Development Center.

Dr. Makino received the Best Paper Award at the IEEE International Conference on Computer Design in 1993. He is currently 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, Japan, in 1981 and 1983, respectively, and the Ph.D. degree in electronic engineering from Osaka University, Suita, Japan, in 1999.

In 1983, he joined the LSI Laboratory, Mitsubishi Electric Corporation, Hyogo, Japan. He has been engaged in high-speed bipolar ECL SRAMs from 1K to 16K bits, BiCMOS analog/digital circuit design, CMOS processing units including multiport SRAMs and floating-point arithmetic cores. His current researches are ultrahigh-speed interface technologies and low-power system LSI design techniques.

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

Tsutomu Yoshihara was born in Takamatsu, Japan, on February 2, 1947. He received the B.S. and M.S. degrees in physics and the Ph.D. degree in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1983, respectively.

He joined the Semiconductor Division, Mitsubishi Electric Corporation, Hyogo, Japan, in 1971. Since then, he has been engaged in the research and development of MOS LSI memories at the ULSI Laboratory.

Dr. Yoshihara is a member of the Institute of Electronics, Information and Communication Engineers of Japan.

Yasutaka Horiba was born in 1941. He received the B.S. and M.S. degrees in electronics from Nagoya University, Nagoya, Japan, in 1964 and 1966, respectively, and the Ph.D. degree in electronic engineering from Osaka University, Osaka, Japan, in 1982.

In 1966, he joined Mitsubishi Electric Corporation, with the Semiconductor Research and Development Department, Kita-Itami Works, Itami, Japan, where he was engaged in the research and development of bipolar-analog and CMOS-digital integrated circuits. From 1976 to 1993, he was with the LSI Research and Development Laboratory of the company, where he worked on CMOS and Bipolar gate arrays, and then on digital signal processing LSIs. Since 1993, he has been working on the development of system-level VLSIs.