PAPER *Special Section on Digital Signal Processing*

Design of a Field-Programmable Digital Filter Chip Using Multiple-Valued Current-Mode Logic

Katsuhiko DEGAWA[†a)], ***Student Member***, **Takafumi AOKI**[†], ***Regular Member***, ***and*** **Tatsuo HIGUCHI**[††], ***Fellow***

SUMMARY This paper presents a Field-Programmable Digital Filter (FPDF) IC that employs carry-propagation-free redundant arithmetic algorithms for faster computation and multiple-valued current-mode circuit technology for high-density low-power implementation. The original contribution of this paper is to evaluate, through actual chip fabrication, the potential impact of multiple-valued current-mode circuit technology on the reduction of hardware complexity required for DSP-oriented programmable ICs. The prototype FPDF fabrication with 0.6 μm CMOS technology demonstrates that the chip area and power consumption can be reduced to 41% and 71%, respectively, compared with the standard binary logic implementation.

key words: *multiple-valued logic, signal processor, FPGAs, FIR filters*

1. Introduction

In many real-time Digital Signal Processing (DSP) applications, traditional microprocessor-based architectures are often inadequate to meet the requirements for intensive computation. Field-Programmable Gate Arrays (FPGAs) provide an alternative that maintains a rapid prototyping capability while providing performance levels significantly beyond that of programmable processors. If we introduce domain-specific FPGA architectures, further performance improvement is expected in specific DSP applications. Recently, several DSP-oriented FPGA architectures have been reported [1]–[3]. The key to success in designing specialized FPGA architectures for DSP relies on hardware algorithms that make possible not only higher performance but also lower circuit complexity. The problem of interconnection complexity has been recognized as a basic limitation in applying the reconfigurable computing technique for DSP tasks.

Addressing this problem, this paper presents a technique for implementing high-performance DSP-oriented FPGAs exhibiting both low power consumption and reduced circuit complexity. The proposed technique employs (i) binary SD (Signed-Digit) arithmetic [4]–[6] for high-speed multiply-add operations, and (ii) Multiple-Valued Current-Mode Logic (MV-CML) for significant reduction in wiring complexity. Figure 1 summarizes the impacts of the proposed technique for realizing a high-performance configurable signal processing architecture.

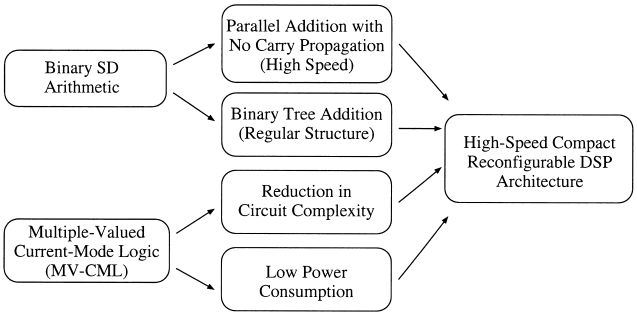


Fig. 1 Design technique overview.

In this paper, we demonstrate the potential of the proposed technique in a typical application example—design of a specialized FPGA architecture for high-speed FIR filtering. The reference [7] has already proposed a high-speed programmable filter IC, called a Field-Programmable Digital Filter (FPDF)*, dedicated for signal processing and communication applications. The architecture was designed on the basis of ordinary binary logic circuits. In this paper, on the other hand, we propose a new design of FPDF based on the combination of MV-CML circuit technology and redundant arithmetic algorithms. The goal of this paper is to provide a case study to evaluate the impact of MV-CML on the reduction of hardware complexity required for DSP-oriented programmable ICs. A prototype chip of FPDF using MV-CML has been successfully designed and fabricated using 0.6 μm CMOS technology. The chip area and power consumption (@40 MHz operation) can be reduced to 41% and 71%, respectively, compared with the standard binary logic implementation described in [7].

Manuscript received November 22, 2002.
Manuscript revised March 3, 2003.
Final manuscript received April 14, 2003.

[†]The authors are with the Department of Computer and Mathmatical Sciences, Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980-8579 Japan.
[††]The author is with the Department of Electronics, Tohoku Institute of Technology, Sendai-shi, 982-8577 Japan.
a) E-mail: degawa@aoki.ecei.tohoku.ac.jp

*The FPDF architecture is dedicated to the implementation of FIR filters, and hence only the number of filter taps, the filter coefficients, datapath word length and mapping structure are programmable. To verify the potential of the proposed technique, further performance analysis is required assuming various different DSP tasks for practical applications.

Table 1 Generation of W_i and C_i in Step 2.

Z_i		-2	-1	0	$+1$	$+2$
$Z_{i-1} > 0$	W_i	0	-1	0	-1	0
	C_i	-1	0	0	$+1$	$+1$
$Z_{i-1} \leqq 0$	W_i	0	$+1$	0	$+1$	0
	C_i	-1	-1	0	0	$+1$

2. Binary SD Arithmetic

This section describes binary Signed-Digit (SD) number representation, which allows high-speed arithmetic operations without carry propagation and is suitable for the implementation with Multiple-Valued Current-Mode Logic (MV-CML). The SD number representation is a redundant representation using a symmetrical digit set $\{-1, 0, 1\}$. Any integer X can be represented as a sequence of radix-2 signed digits X_i as follows:

$$\begin{aligned} X &= [X_{n-1}X_{n-2}\cdots X_1X_0]_{SD2} \\ &= \sum_{i=0}^{n-1} X_i \cdot 2^i, \end{aligned} \tag{1}$$

where $X_i \in \{-1, 0, 1\}$.

Consider two binary SD numbers:

$$X = [X_{n-1}\cdots X_i \cdots X_0]_{SD2}, \tag{2}$$

$$Y = [Y_{n-1}\cdots Y_i \cdots Y_0]_{SD2}. \tag{3}$$

The addition of these two numbers is performed by the following three steps in each digit:

$$\begin{aligned} \text{Step 1:} \quad Z_i &= X_i + Y_i, \\ \text{Step 2:} \quad 2C_i + W_i &= Z_i, \\ \text{Step 3:} \quad S_i &= W_i + C_{i-1}, \end{aligned}$$

where Z_i, W_i, C_i and S_i are the *linear sum*, the *intermediate sum*, the *carry* and the *final sum*, respectively. Each variable takes the following range:

$$\begin{aligned} X_i, Y_i, S_i, C_i &\in \{-1, 0, 1\}, \\ Z_i &\in \{-2, -1, 0, 1, 2\}, \\ W_i &\in \{-1, 0\} \text{ (if } Z_{i-1} > 0), \\ W_i &\in \{0, 1\} \text{ (if } Z_{i-1} \leqq 0). \end{aligned}$$

Step 2 decomposes the linear sum Z_i into the intermediate sum W_i and the carry C_i so that the dynamic range of the final sum S_i created in Step 3 may fit within the range from -1 to 1. Table 1 and Fig. 2 show how to generate W_i and C_i from Z_i in Step 2. Since the carry C_i can be computed without referring the lower-digit carry C_{i-1}, carry-propagation-free addition can be achieved. Figure 3 shows an example of binary SD addition, where $\bar{1}$ denotes -1. Figure 4 illustrates the structure of a parallel adder using binary SD arithmetic, where "Binary SDFA" denotes a one-digit binary SD full adder cell realizing the operations of Step 1, Step 2 and Step 3.

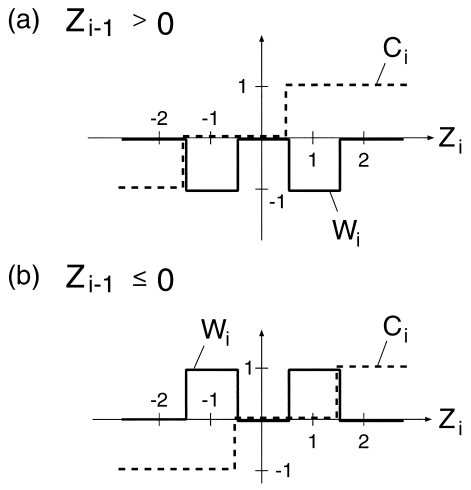


Fig. 2 Transfer characteristic for Step 2.

Weight	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
X	1	$\bar{1}$	0	1	1	$\bar{1}$	$\bar{1}$	1	= 83
Y +)	1	0	1	1	0	0	$\bar{1}$	0	= 174
Z	2	$\bar{1}$	1	2	1	$\bar{1}$	$\bar{2}$	1	
W	0	$\bar{1}$	$\bar{1}$	0	1	1	0	1	
C	1	0	1	1	0	1	1	0	
S	1	0	0	0	0	0	0	0	1 = 257

Fig. 3 Example of binary SD addition.

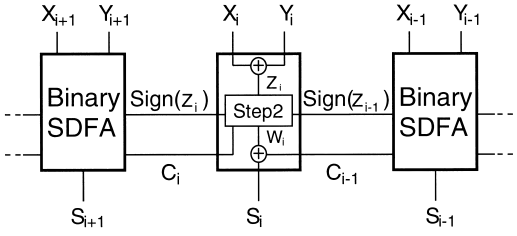


Fig. 4 Parallel implementation of a binary SD adder.

3. Binary SD Adder Using MV-CML

This section describes the implementation of the binary SD adder using Multiple-Valued Current-Mode Logic (MV-CML) technology [5], [8]. In MV-CML, each digit $\{-1, 0, 1\}$ for binary SD arithmetic can be represented by bidirectional current on a wire; the sign (+ or −) is represented by current direction (positive or negative) and the digit magnitude is represented by current level (amount).

Figure 5 shows the basic building blocks for MV-CML used in our design. Threshold detectors, which compare the amount of input current and that of reference current, are the most characteristic components in MV-CML circuits. In the following, we explain the operation of the pMOS Type-B threshold detector (Type-B pTD) in detail as an example (see the portion marked

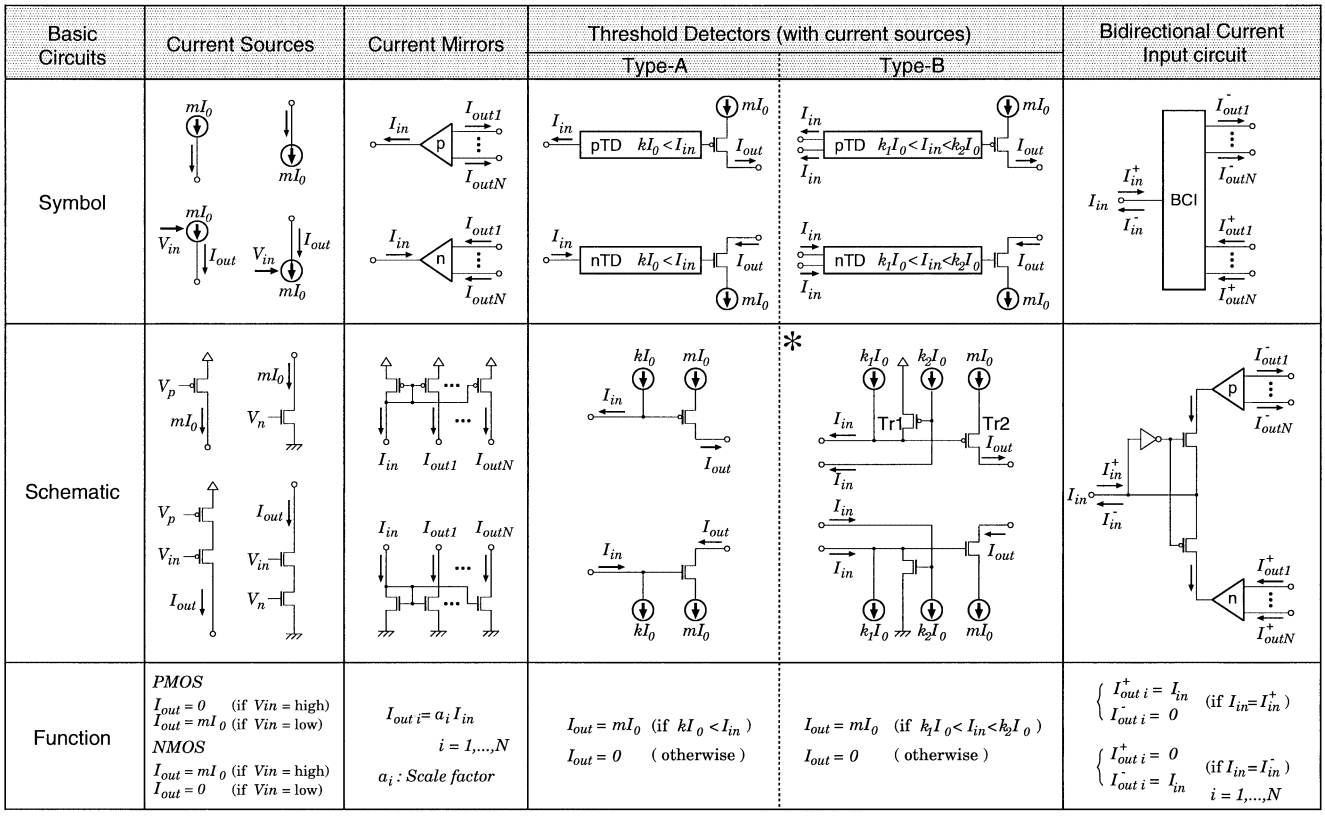


Fig. 5 Basic MV-CML circuits, where I_0 is the unit current ($I_0 = 10\,\mu$A in our design).

with "$*$" in Fig. 5). We assume that the two inputs indicated by I_{in} are identical current inputs produced by current mirrors in the Bidirectional Current Input (BCI) circuit. When $I_{in} < k_1 I_0$, the gate voltages of Tr1 and Tr2 become HIGH and hence $I_{out} = 0$. When $k_1 I_0 < I_{in} < k_2 I_0$, the gate voltage of Tr1 is HIGH and that of Tr2 is LOW resulting in $I_{out} = mI_0$. When $k_2 I_0 < I_{in}$, the gate voltage of Tr1 is LOW and that of Tr2 is HIGH resulting in $I_{out} = 0$. Thus, Type-B pTD produces the current $I_{out} = mI_0$ if and only if the condition $k_1 I_0 < I_{in} < k_2 I_0$ is satisfied.

One of the most important features of MV-CML is that linear summation can be performed by wiring without any active devices. This makes possible drastic reduction in the number of transistors in arithmetic circuits. Figure 6 shows the MV-CML implementation of the binary SD full adder (SDFA) described in the last section. The linear summation operations for Step 1 and Step 3 are implemented by simple wiring points. The most of the circuit resources are devoted for Step2, which accepts the bidirectional current sum Z_i and generates W_i and C_i. The bidirectional current Z_i is copied to unidirectional currents $I_1 \sim I_2$ or to $I_3 \sim I_5$ depending on the current direction. The threshold detectors produce the binary voltage signals $V_1 \sim V_5$, which control the switching of 10 pass transistors. These pass transistors form a pair of series-parallel logic circuits to implement the transfer characteristics for W_i and C_i as

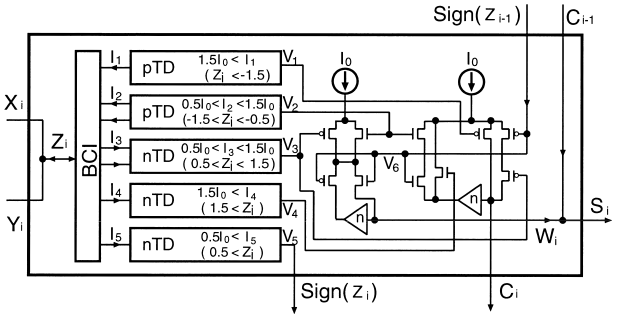


Fig. 6 Binary SD Full Adder (SDFA) using MV-CML.

illustrated in Fig. 2. For example, $W_i = -1$ (i.e., $-I_0$) when $\{(V_2$ OR $\overline{V_3})$ AND $\overline{V_6}\}$ = HIGH.

Figure 7 shows the transfer characteristics for the binary SDFA obtained by HSPICE simulation, where standard $0.6\,\mu$m CMOS technology is employed. The unit current I_0 corresponding to the digit value 1 is $10\,\mu$A. The unit current determines the maximum operating frequency of MV-CML circuits. We derive the condition $I_0 = 10\,\mu$A from our target frequency 50 MHz. The unit current $10\,\mu$A is about 1/20 of short-circuit current in a CMOS inverter.

The current sources are most important components in MV-CML circuits since they are used to produce the output signals of functional modules as well as the reference signals for threshold detectors. In our

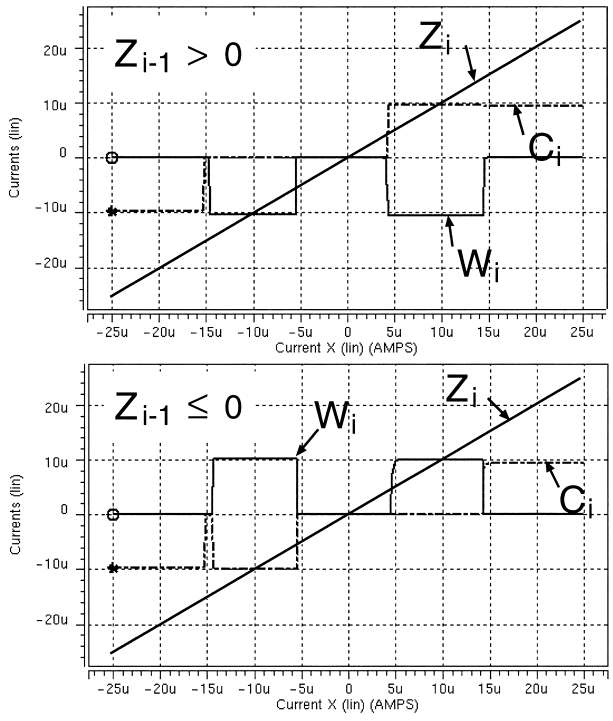


Fig. 7 HSPICE simulation for the binary SDFA.

design, the unit current I_0 is programmed as $10\,\mu$A by controlling the transistor sizes (W and L) of current sources, and their gate bias voltages V_p and V_n. We use transistors with the channel length $L = 1.5 \times L_{min} \sim 2 \times L_{min}$ as current sources, where L_{min} denotes the minimum channel length of a transistor. Thus, we can use fairly small transistors in MV-CML circuits compared with ordinary analog circuits. The operation of MV-CML circuits is not so sensitive to V_t variation, since a designer must guarantee the correct circuit operation only for a limited range of discrete operating points: $-20\,\mu$A, $-10\,\mu$A, $0\,\mu$A, $10\,\mu$A and $20\,\mu$A.

Note that MV-CML can carry a single digit ($\in \{-1, 0, 1\}$) of binary SD representation on a single wire. On the other hand, ordinary binary logic implementation requires a pair of wires to convey a single digit of SD representation since the digit set $\{-1, 0, 1\}$ must be encoded into binary vectors, such as $\{10, 00, 01\}$. Thus, 50% reduction in the number of I/O interconnections is expected for the MV-CML SDFA compared with the binary logic implementation.

Also, the current summation technique of MV-CML, which does not require any active devices, makes possible further reduction in the number of interconnections. For example, assume that we would like to send a pair of SD operands X_i and Y_i ($\in \{-1, 0, 1\}$) that should eventually be added at a specific destination. In voltage-mode binary logic implementation, we need 4 wires to transmit the SD operands X_i and Y_i independently. In MV-CML circuits, on the other hand, the two SD operands are represented by a pair of bidirectional current signals. Assuming the two numbers will be summed up at the receiver end, it may be a good idea to pack X_i and Y_i by current summation before transmission; thus, we transmit a 5-level current signal ($\in \{-2, -1, 0, 1, 2\}$) on a single interconnection. (Note that we need an SDFA to reduce the signal range from $\{-2, -1, 0, 1, 2\}$ to $\{-1, 0, 1\}$ at the receiver end.) This technique can reduce the number of interconnects in MV-CML circuits to 25% (=1/4) compared with the binary logic implementation. This property is particularly useful for DSP-oriented FPGA devices, in which efficient implementation of arithmetic operations is required.

4. Designing FPDF Using MV-CML

4.1 Architecture

This section describes the design and implementation of a Field-Programmable Digital Filter (FPDF)—a programmable filter IC for high-speed FIR filtering dedicated for signal processing and communication applications [9], [10]. We choose FIR filter application in order to demonstrate the potential of the proposed technique—the combination of redundant arithmetic algorithms and MV-CML circuit technology—for high-speed, low-power and compact implementation of DSP-oriented programmable ICs. This technique will also be effective in other applications, in which efficient implementation of multiply-add operation is required.

Figure 8 illustrates the overall architecture of the FPDF, which consists of Configurable Arithmetic Blocks (CABs), Connection Blocks (CBs), Switch Blocks (SBs) and Product Generators (PGs). The basic structure of FPDF is similar to those of conventional FPGAs—building blocks are connected by programmable interconnects. Major difference is that FPDF employs an 8-bit latched SD adder as a functional block (called the CAB) instead of Configurable Logic Blocks (CLBs) used in typical FPGAs. These CABs are connected by programmable current-mode interconnections according to configuration data downloaded into FPDF in advance. Each current-mode interconnection can convey a 3-valued signal $\{-1, 0, 1\}$ or a 5-valued signal $\{-2, -1, 0, 1, 2\}$ as described in the last section. The routing elements, CBs and SBs, are used to guide the bidirectional current signals from source CABs to destination CABs.

Figure 9 illustrates how to map an FIR filter structure onto basic components of FPDF, where the word length of the filter is 8 bits in this example. The filter coefficients are encoded by Canonical Signed-Digit (CSD) representation [11]. The multiplications between the input signal and CSD coefficients are performed by PGs followed by CABs. The pipelined accumulation of generated products is performed by CABs to produce the output of the FIR filter. For pipelining,

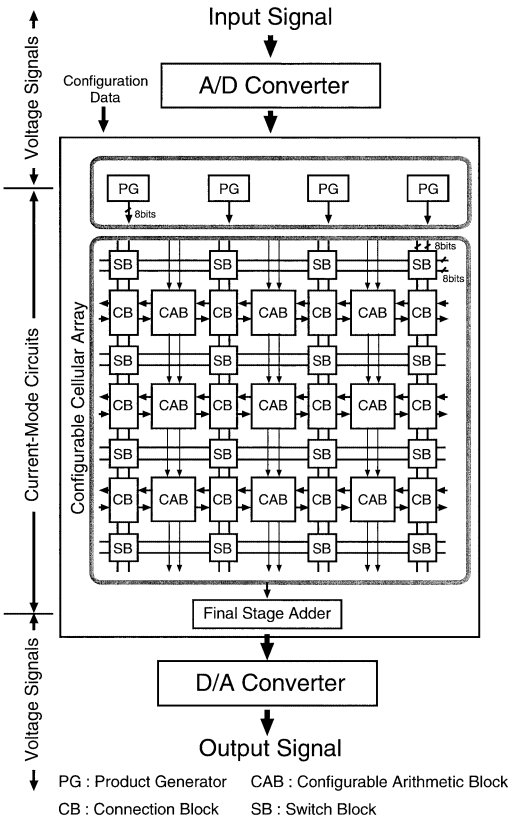


Fig. 8 MV-CML FPDF architecture.

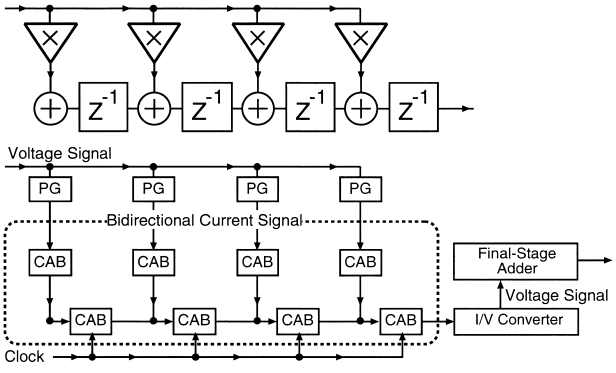


Fig. 9 FIR filter mapping.

a CAB contains an 8-bit register, which could be bypassed when it is used for CSD multiplication. Bidirectional current summation is fully utilized for pipelined accumulation leading to drastic reduction in interconnection complexity.

4.2 Functional Blocks for FPDF

This subsection describes the design of basic functional blocks for FPDF, where every functional block is newly designed using MV-CML circuit technology. The use of MV-CML makes possible significant reduction in the number of programmable interconnects, which leads to

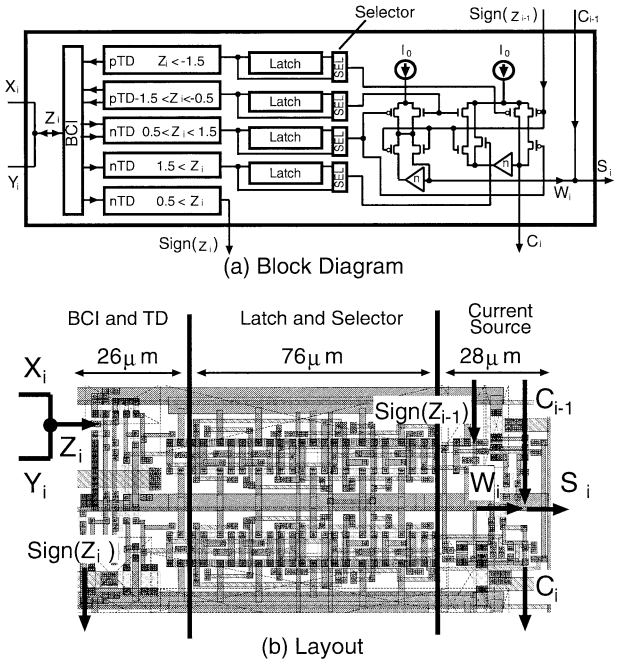


Fig. 10 Latched SDFA for CAB (1-bit circuit is shown).

compact implementation of CBs and SBs. Also, direct implementation of SD arithmetic using MV-CML without binary encoding allows high-speed low-power implementation of CABs. In the following, we describe the newly designed MV-CML functional blocks in detail.

CAB (Configurable Arithmetic Block) is an 8-bit latched binary SD adder with no carry propagation chain. Figure 10(a) shows the 1-bit circuit for CAB. MV-CML circuit technology is used to implement the CAB circuit except for latches for pipelining; the latches employ conventional voltage-mode circuits and could be bypassed by selectors when combinational operation is required. Figure 10(b) shows the layout of CAB using 0.6 μm CMOS triple-metal technology. Note that this circuit implements Step 2 and Step 3 of binary SD addition, while Step 1 is realized by simple current summation on a input wire.

There are two routing elements in FPDF, that is, CB (Connection Block) and SB (Switch Block). Figure 11 shows the structure of CB, which consists of nMOS switches for wire routing and SRAM cells for controlling the nMOS switches. CBs are used not only for programmable routing, but also for current summation by wiring (Step 1 of binary SD addition); if two current signals are connected to a common interconnection, these signals are added together to form a 5-level current signal ($\in \{-2, -1, 0, 1, 2\}$), resulting in 25% reduction in the number of interconnects. On the other hand, SB shown in Fig. 12 is a switch matrix placed at an intersection of vertical and horizontal data lines.

PG (Product Generator) shown in Fig. 13, which

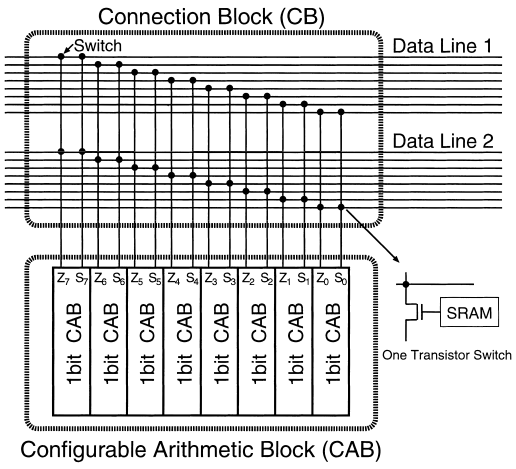


Fig. 11 Connection Block (CB) and Configurable Arithmetic Block (CAB).

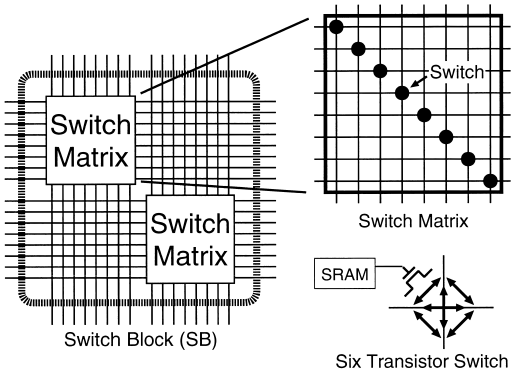


Fig. 12 Switch Block (SB).

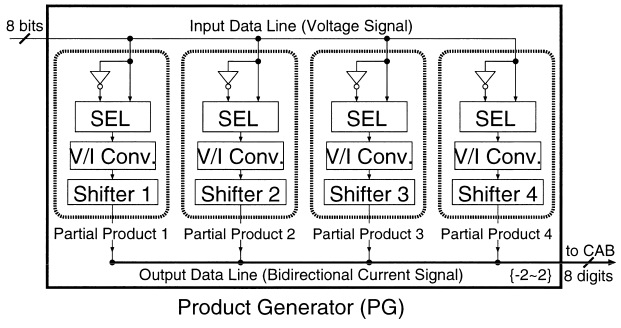


Fig. 13 Product Generator (PG).

is placed at the first stage of FPDF architecture, generates a product between an 8-bit input signal and an 8-digit CSD coefficient (that contains four nonzero digits (-1 or 1) at most). The sign-vector conversion technique of Signed-Weight (SW) arithmetic [7] is used to control the sign of partial products so that we can pack every four partial products into a single 8-digit string of 5-valued digit ($\{-2, -1, 0, 1, 2\}$).

This product generation algorithm is explained in Fig. 14. Let assume that the coefficient of multiplication is $0.1011011_2 (= 0.7109375)$ in two's complement binary number system. This coefficient can be converted into CSD coefficient as $1.0\bar{1}00\bar{1}0\bar{1}_{CSD2}$, which contains four nonzero digits. Assume also that the input to the FIR filter is 0.1011010_2 in two's complement binary number system. Multiplication between the input and the CSD coefficient produces the four partial products (corresponding to the four nonzero digits in the CSD coefficient). Note that the first partial product has positive sign and the other three partial products have negative signs as shown in Fig. 14. For making efficient use of sign-symmetric number representation in SD notation, we need to balance the number of positive and negative partial products. Thus, in Fig. 14, the sign of the 4th partial product is flipped into positive by introducing negative bias quantity as a side effect (the bias thus introduced should be canceled at the final-stage adder in the FPDF architecture). As a result, we have two positive partial products and two negative partial products. By adding every pair of positive and negative partial products, we can convert the four partial products into two partial products in binary SD number representation. Note that these two binary SD products must be added at CAB in the succeeding stage as shown in Fig. 9. For this purpose, PG generates a single 5-valued signal by packing two binary SD products with current summation in advance. The CAB at the next stage receives this packed signal and converts it into binary SD notation. Figure 15 illustrates another example of product generation for the CSD coefficient $0.0101011_{CSD2} (= 0.3359375)$. As

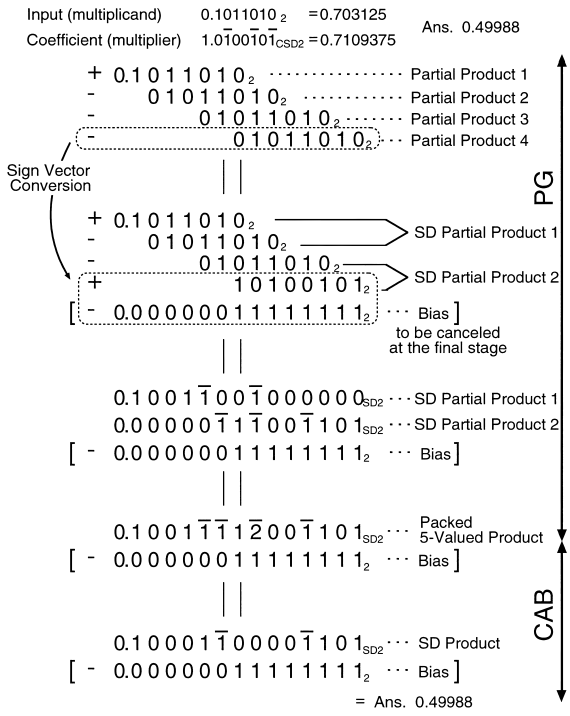


Fig. 14 Example of CSD-coefficient multiplication.

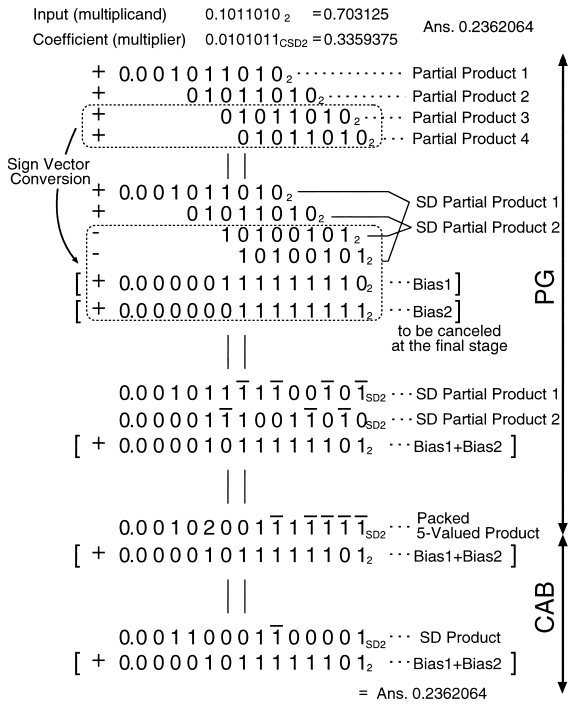


Fig. 15 Example of CSD-coefficient multiplication.

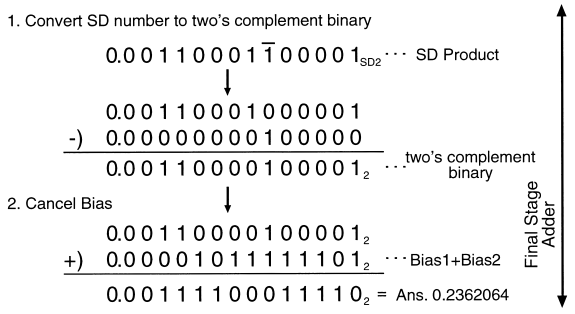


Fig. 16 Example of final stage additions.

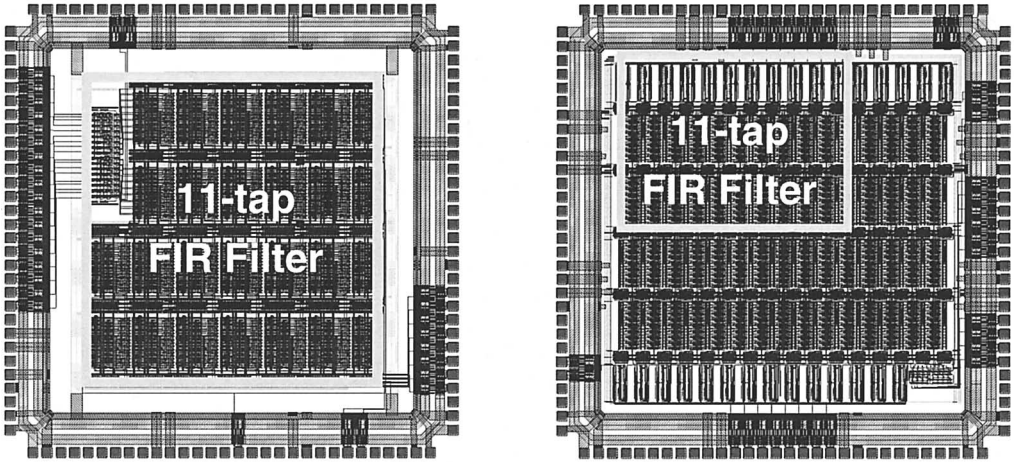
shown in this example, the sign vector conversion technique makes possible to generate a pair of SD partial products for arbitrary CSD coefficients. The side effect of this process—the bias quantity—must be canceled at the final stage adder. This bias cancelling process is illustrated in Fig. 16.

The above example illustrates 8-bit CSD coefficient multiplication. The wordlength of internal datapath can be extended to any multiple of 8 bits by cascading the carry-out/carry-in signals of CABs. Input/output data length, on the other hand, cannot be extended in our present design due to the limited number of I/O pins.

4.3 Performance Evaluation

This section compares the proposed MV-CML FPDF with the corresponding voltage-mode binary logic implementation (using SW arithmetic) [7]. The binary logic FPDF considered here is based on the design described in [7], where each CAB employ a 4-2 counter with no carry propagation chain. Also, the binary logic FPDF does not employs PGs, since the partial product generation is performed by CBs through simple shift operations. Both FPDFs (binary logic and MV-CML) are dedicated to high-speed FIR filter realization and have the same function. The major design parameters, such as wordlength and data line width, are almost same for both designs.

For comparison, we designed both FPDFs using the same 0.6 μm CMOS triple-metal technology. Figure 17 compares the layout of MV-CML FPDF with that of binary-logic FPDF on a 4.4 mm × 4.4 mm chip. Binary-logic FPDF integrates 22 CABs on a chip and can implement an 11-tap FIR filter at most. On the other hand, MV-CML FPDF integrates 60 CABs on a chip and can implement a 30-tap FIR filter at most. When mapping an 11-tap FIR filter, binary-logic FPDF



(a) Binary-logic FPDF (b) MV-CML FPDF

Fig. 17 Comparison of FPDF chips: binary logic vs. MV-CML (0.6 μm CMOS triple metal technology).

Table 2 Comparison of FPDF chip features.

		Binary-Logic FPDF	MV-CML FPDF
Input data/Output data		8 bit/16 bit (two'complement)	8 bit/8 bit (two'complement)
Maximum filter tap		11-tap	30-tap
Transistor count		109,704 Transistors	150,247 Transistors
Chip size		4.4 mm × 4.4 mm	
Effective size		3.0 mm × 3.0 mm	3.4 mm × 3.5 mm
Process		0.6 μm CMOS triple metal layers	
Active area size for an 11-tap FIR filter		9.0 mm^2 [109,704 Transistors] (3.0 mm × 3.0 mm)	3.7 mm^2 [51,891 Transistors] (2.2 mm × 1.7 mm)
Power consumption for an 11-tap FIR filter (HSPICE simulation)	10 MHz	24 mW	34 mW
	20 MHz	48 mW	45 mW
	30 MHz	73 mW	56 mW
	40 MHz	96 mW	68 mW

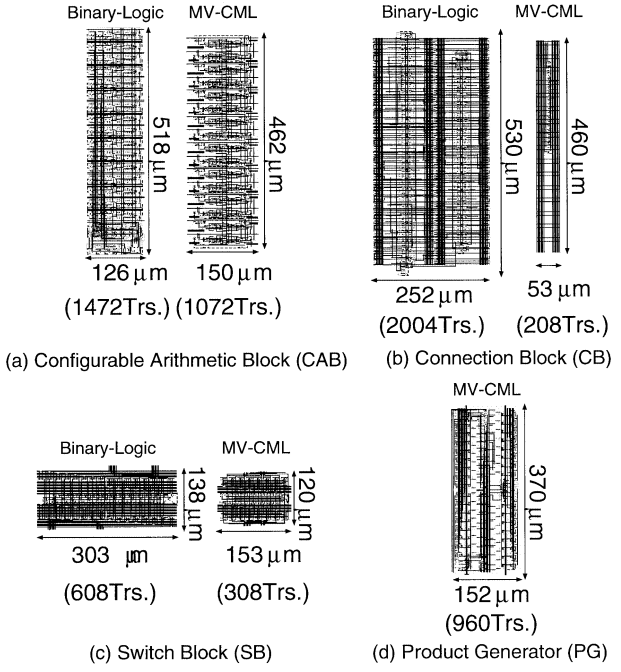


Fig. 18 Layouts of basic building blocks.

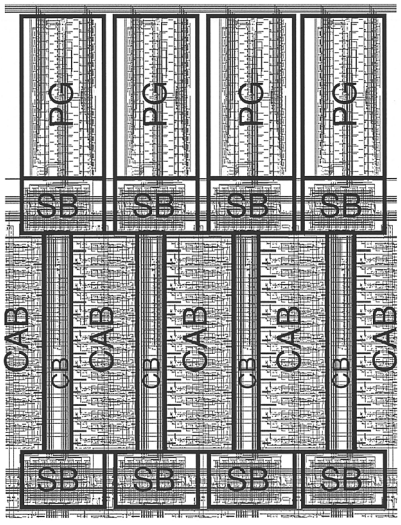


Fig. 19 Enlarged layout of MV-CML FPDF.

requires active area of 9.0 mm^2, while MV-CML FPDF requires only 3.7 mm^2. Thus, the use of MV-CML technology makes possible to reduce the active area to 41% in comparison with binary logic implementation. The transistor count and power consumption (@40 MHz operation) are also reduced to 47% and 71%, respectively.

Table 2 summarizes the result of comparison. The output data length of MV-CML FPDF is 8 bits simply because the wordlength of the final stage adder on a fabricated chip is 8 bits. We can easily extend the wordlength of the final stage adder to 16 bits with only 1% increase in transistor count. Note that there are no essential differences in the functionality between binary logic FPDF and MV-CML FPDF. The wordlength of internal datapath for both designs can be extended to any multiple of 8 bits by changing configuration data.

Figure 18 compares the layouts of basic building blocks for binary-logic FPDF with those for MV-CML FPDF. The size of CAB is almost same for both designs. As for routing elements, such as CBs and SBs, MV-CML FPDF requires much less area on a chip due to the reduction in the number of programmable interconnections by using multi-level current-mode signaling. Figure 19 shows the enlarged view of the layout of MV-CML FPDF.

A major disadvantage of MV-CML circuit technology is its limited range of operating frequency. In our design, the unit current (current for digit value 1) is 10 μA. In this case, the maximum operating frequency is limited to 40 MHz, while the binary-logic implementation reaches 90 MHz. In order to achieve higher performance, we need to increase the unit current of MV-CML circuits. For example, if we introduce higher unit current, say 30 μA, we can achieve 85 MHz operation[†]. In this case, however, relatively higher power dissipation in low-frequency region may be a major problem. In order to resolve the tradeoff between speed and

[†]HSPICE simulation shows that we can change the unit current within the range 10–40 μA simply by changing the bias voltages V_p and V_n for current sources. We must change the transistor sizes (W and L) in order to change the unit current beyond this range.

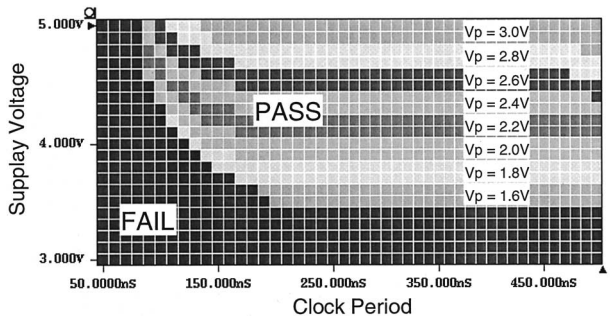


Fig. 20 Chip measurement result of the MV-CML FPDF chip ($V_n = 2\,\mathrm{V}$).

power, application specific consideration is required for selecting optimal value of unit current. Variable-unit-current approach should also be explored for future applications.

We have designed and fabricated an MV-CML FPDF chip shown in Fig. 17(b). The valid operation is successfully confirmed by mapping various FIR filters having a tap count up to 30. Figure 20 shows an example of chip measurement using an LSI tester (Advantest T6671E). The critical delay is measured by connecting the basic components: PG, SB, CB, CAB, CB, SB, and the final stage adder, in series (bypassing all the latches in CAB). In this particular case, the measured critical delay is 90ns, which significantly depends on the circuit configuration mapped onto the FPDF.

5. Conclusion

In this paper, we proposed a technique for implementing high-performance DSP-oriented FPGAs exhibiting low power consumption and reduced circuit complexity. The technique employs (i) binary SD (Signed-Digit) arithmetic for high-speed multiply-add operations, and (ii) multiple-valued current-mode logic (MV-CML) for reducing wiring complexity. We need to verify the potential of the proposed technique in various different DSP tasks for practical application.

Acknowledgments

The VLSI chip in this study has been fabricated under the chip fabrication program of the VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation.

References

[1] O.T.C. Chen and W. Liu, "An FIR processor with programmable dynamic data ranges," IEEE Trans. VLSI Syst., vol.8, no.4, pp.440–446, Aug. 2000.
[2] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: A survey," J. VLSI Signal Processing, vol.28, pp.7–27, Aug. 2001.
[3] T. Sueyoshi and M. Iida, "Configurable and reconfigurable computing for digital signal processing," IEICE Trans. Fundamentals, vol.E85-A, no.3, pp.591–599, March 2002.
[4] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," IRE Trans. Electronic Computers, vol.EC-10, pp.389–400, Sept. 1961.
[5] S. Kawahito, M. Kameyama, and T. Higuchi, "Multiple-valued radix-2 signed-digit arithmetic circuits for high-performance VLSI systems," IEEE J. Solid-State Circuits, vol.25, no.1, pp.125–131, Feb. 1990.
[6] N. Takagi, H. Yasuura, and S. Yajima,"High-speed VLSI multiplication algorithm with a redundant binary addition tree," IEEE Trans. Comput., vol.C-34, no.9, pp.789–796, Sept. 1985.
[7] T. Aoki, Y. Sawada, and T. Higuchi, "Signed-weight arithmetic and its application to a field-programmable digital filter architecture," IEICE Trans. Electron., vol.E82-C, no.9, pp.1687–1698, Sept. 1999.
[8] S. Kawahito, M. Kameyama, T. Higuchi, and H. Yamada, "A 32 × 32-bit multiplier using multiple-valued MOS current-mode circuits," IEEE J. Solid-State Circuits, vol.23, no.1, pp.124–132, Feb. 1988.
[9] B.C. Wong and H. Samueli, "A 200-MHz all-digital QAM modulator and demodulator in 1.2-μm CMOS for digital radio application," IEEE J. Solid-State Circuits, vol.26, no.12, pp.1970–1980, Dec. 1991.
[10] C. Henning, R. Schwann, V. Gierenz, and T.G. Noll, "A low power reconfigurable 12-tap FIR interpolation filter with fixed coefficient sets," Proc. 26th European Solid-State Circuits Conference, Sept. 2000.
[11] I. Koren, Computer Arithmetic Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 1993.

Katsuhiko Degawa received the B.E. degree in electronic engineering from Tohoku University, Sendai, Japan, in 2002. He is currently working toward the M.E. degree. His research interest includes hardware algorithms and VLSI architecture. Mr. Degawa received the Tohoku Section Presentation Award from the Institute of Electrical Engineers of Japan in 2002.

Takafumi Aoki received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1988, 1990, and 1992, respectively. He is currently a Professor of the Graduate School of Information Sciences at Tohoku University. For 1997–1999, he also joined the PRESTO project, Japan Science and Technology Corporation (JST). His research interests include theoretical aspects of computation, VLSI computing structures for signal and image processing, multiple-valued logic, and biomolecular computing. Dr. Aoki received the Outstanding Paper Award at the 1990, 2000 and 2001 IEEE International Symposiums on Multiple-Valued Logic, the Outstanding Transactions Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan in 1989 and 1997, the IEE Ambrose Fleming Premium Award in 1994, the IEICE Inose Award in 1997, the IEE Mountbatten Premium Award in 1999, and the Best Paper Award at the 1999 IEEE International Symposium on Intelligent Signal Processing and Communication Systems.

Tatsuo Higuchi received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1962, 1964, and 1969, respectively. He is currently a Professor at Tohoku Institute of Technology. From 1980 to 1993, he was a Professor in the Department of Electronic Engineering at Tohoku University. He was a Professor from 1994 to 2003, and was Dean from 1994 to 1998 in the Graduate School of Information Sciences at Tohoku University. His general research interests include the design of 1-D and multi-D digital filters, linear time-varying system theory, fractals and chaos in digital signal processing, VLSI computing structures for signal and image processing, multiple-valued ICs, multiwave opto-electronic ICs, and biomolecular computing. Dr. Higuchi received the Outstanding Paper Awards at the 1985, 1986, 1988, 1990, 2000 and 2001 IEEE International Symposiums on Multiple-Valued Logic, the Outstanding Transactions Paper Award from the Society of Instrument and Control Engineers (SICE) of Japan in 1984, the Technically Excellent Award from SICE in 1986, and the Outstanding Book Award from SICE in 1996, the Outstanding Transactions Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan in 1990 and 1997, the Inose Award from IEICE in 1997, the Technically Excellent Award from the Robotics Society of Japan in 1990, the IEE Ambrose Fleming Premium Award in 1994, the Outstanding Book Award from the Japanese Society for Engineering Education in 1997, the Award for Persons of scientific and technological merits (Commendation by the minister of state for Science and Technology), the IEE Mountbatten Premium Award in 1999 and the Best Paper Award at the 1999 IEEE International Symposium on Intelligent Signal Processing and Communication Systems. He also received the IEEE Third Millennium Medal in 2000. He received the fellow grade from IEEE, IEICE, and SICE.