

FFT implementation using mono-instruction set computer (MISC) architecture

Hiroki Shinba and Minoru Watanabe
Electrical and Electronic Engineering
Shizuoka University
3-5-1 Johoku, Hamamatsu, Shizuoka 432-8561, Japan
Email: tmwatan@ipc.shizuoka.ac.jp

Abstract—Currently, optically reconfigurable gate arrays that can execute high-speed reconfiguration have been under development for high-performance computing. If a high-speed reconfiguration on a VLSI could be achieved, then mono instruction set computer (MISC) architecture, by which each processor has only one instruction, could be applied to VLSI, thereby enhancing the VLSI performance. This paper presents performance analysis results of a Fast Fourier Transform (FFT) implementation using MISC architecture.

I. INTRODUCTION

Recently, soft-core processors implemented on a Field Programmable Gate Array (FPGA) have become widely used for many applications [1]–[4]. However, an important shortcoming is that the soft-core processor performance is never as good as that of hardcore processors or custom processors because the performance of look-up tables of FPGAs on which a soft-core processor is implemented is lower than those of custom cells or standard cells of custom VLSIs and Application Specific Integrated Circuits (ASICs). Moreover, the path delays on switching matrices of FPGAs are much longer than those of custom VLSI and ASICs

To resolve that difficulty, the complexities of current reduced instruction set computer (RISC) architectures must be improved. The processor performance can be increased using a simpler architecture. Here, the ultimately simplest processor architecture is mono-instruction set computer (MISC) architecture, by which each processor has only a single instruction. Under a static implementation, MISC cannot execute various software applications, but if the hardware itself could be reconfigured, then MISC architecture could execute similar software applications to those of RISC processors.

Optically reconfigurable gate arrays (ORGAs) that can execute high-speed reconfiguration have been under development for use in high-performance computing [5]–[10]. Actually, ORGAs have a fine-grained programmable gate array based on a look-up table and switching matrix architecture, similar to those of FPGAs. Therefore, the basic functions of ORGAs are perfectly identical to those of FPGAs. However, ORGAs, are multi-context devices, can support high-speed dynamic reconfiguration of over 100 MHz. To date, an ORGA with 256 reconfiguration contexts has been demonstrated [11]. When using the ORGA architecture, although a MISC processor has only a single instruction, multi-instruction operations

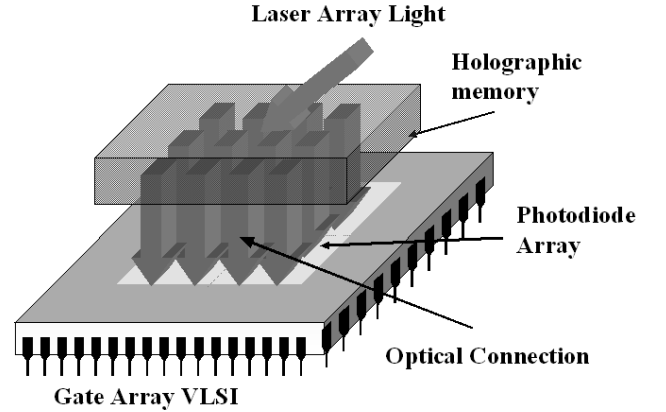


Fig. 1. Overview of an ORGA.

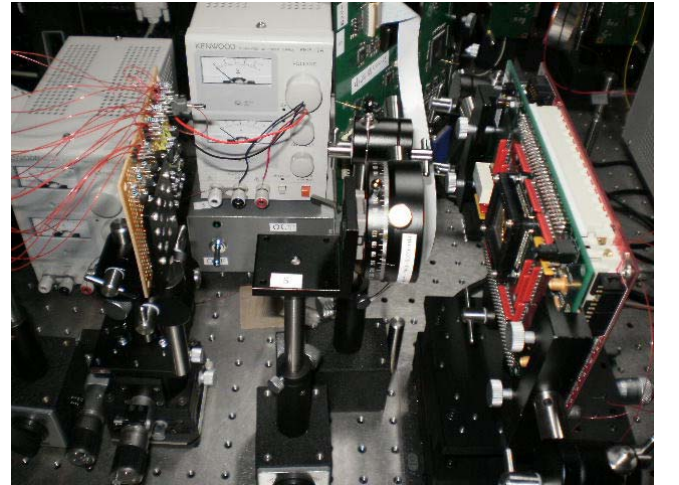


Fig. 2. Optically reconfigurable gate array (ORGA).

identical to those for RISC can be executed by dynamically reconfiguring the programmable gate array on an ORGA. Consequently, the MISC architectures never have a target application restriction.

We have evaluated Fast Fourier Transform (FFT) using such MISC architectures. This paper presents performance analysis results of the FFT implementation.

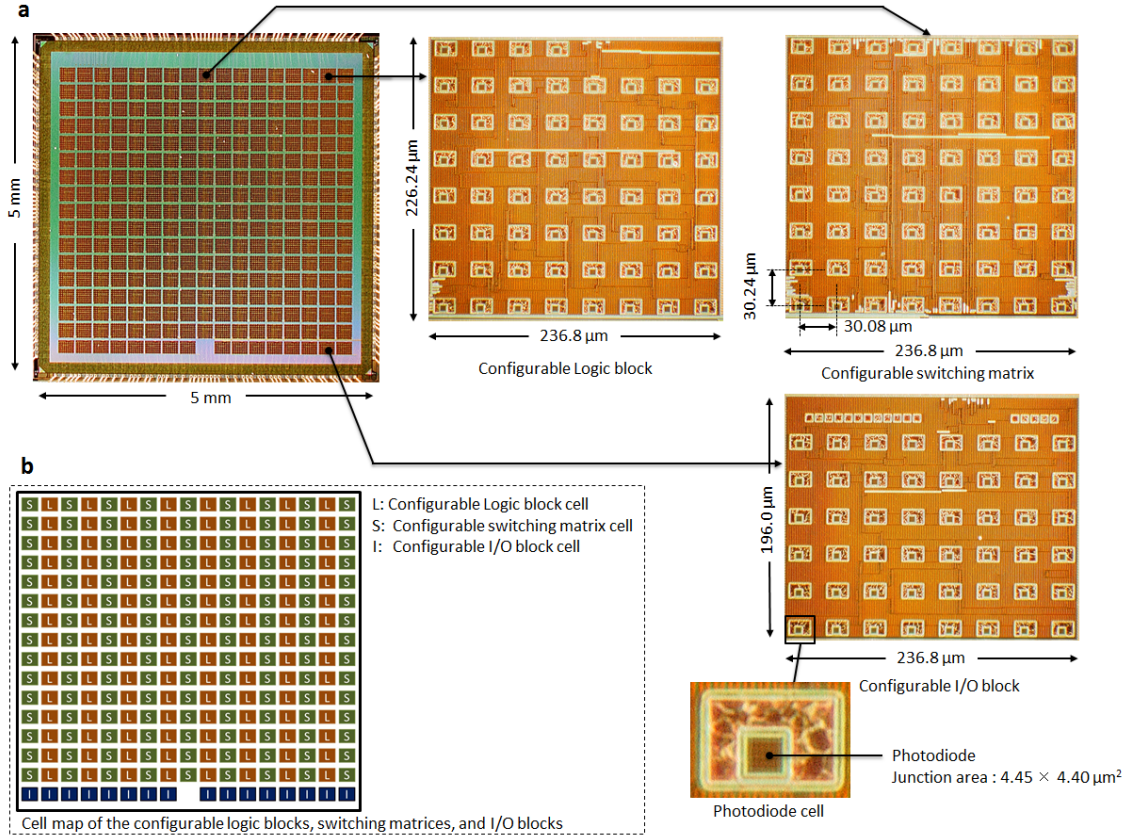


Fig. 3. Optically reconfigurable gate array VLSI chip: a, die photo of the 5 mm \times 5 mm chip and expansion photographs of the configurable logic block (L), configurable switching matrix (S), configurable I/O block (I), and photodiode cell; b, cell map showing locations of the configurable logic blocks, configurable switching matrix, and configurable I/O block.

TABLE I
CHIP CHARACTERISTICS.

Technology	0.18 μm double-poly 5-metal standard CMOS process	
Die size	5.0 \times 5.0 mm^2	
Supply voltage	Core	1.8V
	I/O	3.3V
Photodiode	Junction area	4.40 \times 4.45 μm^2
	Switching energy	2.12 \times 10 ⁻¹⁴ J
	Horizontal interval	30.08 μm
	Vertical interval	30.24 μm
	Num. of photodiodes	17,664
Gate array	Num. of logic blocks	128
	Num. of switching matrices	144
	Num. of Wires in a wiring channel	8
	Num. of I/O blocks	16 (64 bit)
	Gate count	8,704

II. ORGA DESIGN

Figures 1 and 2 respectively present an overview and a photograph of the ORGA prototype system. An ORGA consists of a holographic memory, a laser array, and a programmable gate array VLSI. Since the storage capacities of holographic memories are much higher than those of two-dimensional semiconductor memories, many configuration contexts can be stored on a holographic memory. In addition, the ORGA

architecture supports a parallel configuration exploiting an optical bus between a holographic memory and a programmable gate array VLSI. Therefore, the programmable gate array VLSI has many photodiodes that can receive optically applied configuration contexts. The basic function and performance of the ORGA are the same as those of currently available FPGAs, except for the optical configuration. The prototype system has 16 configuration contexts, with reconfiguration speeds reaching over 100 MHz. In fact, the dynamic reconfiguration procedure can be executed as a background job. Therefore, the dynamic reconfiguration procedure overhead can be removed from consideration.

An ORGA-VLSI chip was fabricated using a 0.18 μm triple-metal standard CMOS process technology as shown in Fig. 3 and in Table I. The ORGA-VLSI chip has a fine-grained gate array consisting of 128 logic blocks, 144 switching matrices, and 64 input-output (I/O) bits, which is similar to FPGAs. A logic block consists of two 16-input look-up tables and two flip-flops. Fundamentally, the VLSI chip functionality is identical to the functionality of typical FPGAs, but each programming element of all logic blocks and switching matrices of the ORGA-VLSI is connected to an optical reconfiguration circuit including a photodiode. For that reason, the gate array can be reconfigured optically. The total number of photodiodes

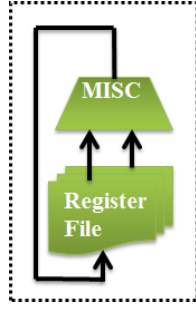


Fig. 4. MISC architecture.

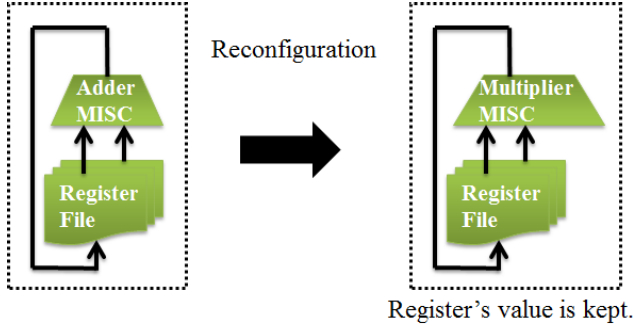


Fig. 5. Instruction change of a MISC processor.

is 17,664. The photodiode was designed to be $4.40 \times 4.45 \mu\text{m}$. The respective horizontal and vertical spaces between photodiodes are $30.08 \mu\text{m}$ and $30.24 \mu\text{m}$. Each photodiode was constructed using a junction between the P-substrate and the N-well. In addition, the outputs of photodiodes are connected to D inputs of configuration flip-flops. The gate count of the programmable gate array is 8,704.

When executing a dynamic reconfiguration, the clock signals of configuration flip-flops connected to the photodiodes are driven by an operating clock signal, which is used for the other flip-flops of a circuit on the programmable gate array. The circuit is operated based on the configuration flip-flops. While an operation is executed on the programmable gate array having a configuration context in the configuration flip-flops, a configuration procedure is executed as a background job. At that time, an addressing laser is activated. Photodiodes receive an optical configuration context generated from a holographic memory addressed by the laser. At the rising edge of the operating clock signal, the configuration flip-flops receive a new configuration context through the photodiode output. Subsequently, a next gate array operation can be executed using information from the configuration flip-flops. Always, the laser array addressing the holographic memory is controlled on the gate array circuit operation.

III. MONO-INSTRUCTION SET COMPUTER (MISC)

Next, the MISC architecture is explained. Fundamentally, a MISC architecture is a set of single-instruction-set-computers. In the MISC implementation, a non-limited number of MISC processors can be used. For example, a MISC implementation

might include an 8-bit multiplier MISC, a 16-bit multiplier MISC, a 32-bit multiplier MISC, and so on. Since the Arithmetic and Logic Unit (ALU) of MISC processor is implemented onto a fine-grained programmable gate array, the number of MISC processor kinds is not limited. For many real applications, MISC processors of more than 1000 kinds might be used. Therefore, a single MISC implementation includes a greater number of instructions or MISC processors than the number of instructions on RISC processors or Complex Instruction set Computer (CISC) processors.

Each MISC processor is implemented onto a fine-grained programmable gate array along with a register file, as shown in Fig. 4. Current RISC processors have many general-purpose registers. As well, a MISC processor uses general-purpose registers. A block diagram is portrayed in Fig. 4. The MISC processor executes its own instruction based on two data on two registers. It writes the calculation result to a register. The programmable gate array region in which MISC processors are implemented is dynamically reconfigured whereas data on register files can be retained at any reconfiguration procedure. In the MISC implementation, MISC processors are changed by reconfiguring a programmable gate array on which the MISC processors are implemented, as presented in Fig. 5. In this case, in the first step, an adder-MISC processor is implemented onto a programmable gate array along with a register file. Then, in the second cycle, the adder-MISC operation is executed by reading two values from the register file. As a background job, the reconfiguration procedure from the adder-MISC processor to a Multiplier-MISC processor is executed. In the third cycle, the multiplier-MISC operation is executed based on the second cycle operation and values on the register file. This is the important point of the MISC processor's operation: it exploits frequent reconfigurations.

IV. MISC PROCESSORS OF 11 KINDS

For this study, MISC processors of 11 kinds were evaluated with 40 nm process technology using the Verilog to Routing (VTR) provided by the University of Toronto. The maximum clock frequency and the resource usage of each MISC processor are shown in Table II. As shown in the table, a 32-bit original soft-core RISC processor with all 11 instructions of the MISC family has been designed as a comparison target. A related block diagram is presented in Fig. 7. The implementation result is also presented at the bottom line of Table II. The soft-core RISC processor can be implemented on a die of 15.95 mm^2 . The operating clock frequency is 4.02 MHz.

For example, an adder-MISC shown in Table II can be implemented onto 0.27 mm^2 . The operating clock frequency is 181.89 MHz. First, the operating clock frequency of the adder-MISC is 45.25 times higher than that of the soft-core RISC processor. Moreover, the implementation areas of the adder-MISC processor can be implemented onto a 59.1-times-smaller region than that of the soft-core RISC processor. If the same implementation area as that of the RISC processor could be used for the adder-MISC processor, then 59 adder

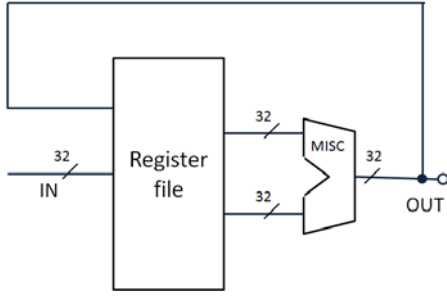


Fig. 6. Block diagram of a MISC processor.

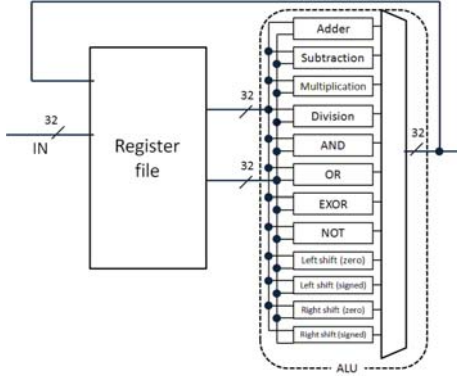


Fig. 7. RISC processor as a comparison target.

MISC processors could be implemented for perfect execution in parallel. The total performance can thereby be enhanced to 2672.2 times faster operation than the soft-core RISC processor.

In the case of the multiplier MISC, as shown in Table II, the multiplier MISC can be implemented onto a 5.33 mm² die. The operating clock frequency is 43.12 MHz. The operating clock frequency of the multiplier-MISC is 10.72 times higher than that of the soft-core RISC processor. Moreover, since the multiplier-MISC processor can be implemented onto a region that is one-third the size of the soft-core RISC processor, three multiplier-MISC processors can be implemented on the same area as the soft-core RISC processor. On those MISC processors, three instructions can be executed in parallel, thereby achieving three-times-better performance. Consequently, the overall performance can be estimated as 32.1 times better than that of the soft-core RISC processor. Table II shows that the MISC performance is 1.9–30,783 times better than that of a conventional soft-core RISC processor taking a static implementation.

The performance of the MISC processor based on the optically reconfigurable gate array architecture is much higher than that of any current soft-core RISC processor.

V. 16-POINT FFT IMPLEMENTATION USING MISC ARCHITECTURE

We have implemented a 16-point Fast Fourier Transform (FFT) onto a 40 nm process technology using the MISC

architecture. For this MISC implementation for the 16-point FFT, a 16-bit fixed point calculation was applied. The number of digits after the decimal point is 8 bits. The basic platform for MISC implementations is presented in Fig. 8. The MISC platform consists of a register file with 32 16-bit registers and MISC processors. The register files can store 16 16-bit real numbers and 16 16-bit imaginary numbers for 16 points. The register files also have 16 outputs with a real number and an imaginary number and 16 write-back inputs with a real number and an imaginary number. First, the calculation data of $X(0)$ to $X(15)$ are set to the register file. Subsequently, MISC operations can be executed based on the register file. The first MISC implementation (MISC-FFT-1) is shown in Fig. 8. The MISC operation is constructed using 16 16-bit adder MISC processors and 16 16-bit subtractor MISC processors. The operation is executed based on current data of the register file. The calculation results of the MISC-FFT-1 are written back to the register file. Then a reconfiguration operation is done as a background operation. For that reason, no overhead exists for this operation. The MISC-FFT-1 operation can be executed within a period of 4.1 ns. The second MISC implementation (MISC-FFT-2) is shown in Fig. 8. The second MISC operation is constructed using 16 16-bit multipliers, 8 adder MISC processors, and 8 16-bit subtractor MISC processors. The second MISC operation is executed based on the first MISC calculation result on the register file. The MISC-FFT-2 operation can be executed within a period of 13.7 ns. When the calculation results are written back to the register file at the clock edge, a reconfiguration operation from MISC-FFT-2 to MISC-FFT-3 is done as a background operation.

In the same manner, each operation is executed along with the reconfiguration. Final calculation results can be read out from the register file. The total operation time can be evaluated as 49.8 ns. The result is presented in Table III.

As a comparison target, we have also implemented the 16-point FFT onto an FPGA with the same 40 nm process technology and no MISC processor. In the case of the full-hardware implementation with no reconfiguration, the required implementation region is 38.53 mm². However, the implementation area for MISC implementations with dynamic reconfigurations is 16.60 mm² at a maximum. Therefore, when the same region as the full-hardware implementation is used for the MISC implementation, at least two FFT units can be implemented onto the VLSI. Consequently, twice the performance of the single MISC implementation could be achieved.

In the MISC implementation, although the operating period was increased slightly from 47.3 ns of the full-hardware implementation to 49.8 ns, since two units can be implemented onto the same region as the full-hardware implementation, the total performance can be estimated as about two times higher than that of the full-hardware implementation. The average time of the MISC operation can therefore be estimated as 23.65 ns. If we were able to use a dynamic reconfigurable hardware platform just as an optically reconfigurable gate array, the circuit performance could be increased drastically.

Additionally, we compared the MISC architecture with

TABLE II

IMPLEMENTATION RESULTS OF MONO-INSTRUCTION SET COMPUTERS (MISCs) OF 11 KINDS. THE BOTTOM LINE SHOWS A CONVENTIONAL RISC SOFT-CORE PROCESSOR INCLUDING ALL INSTRUCTIONS OF THE MISC PROCESSORS DESCRIBED ABOVE, WHICH IS A COMPARISON TARGET UNDER THE SAME CONDITIONS. HERE, THE TARGET PROCESS TECHNOLOGY IS A 40 NM PROCESS ON VTR.

Instruction (32bit)	Area [mm^2]	Num. of units (RISC/MISC)	Operating frequency [MHz]	Frequency ratio (MISC/RISC)	Total Performance
MISC Adder	0.27	59.1	181.89	45.23	2672.2
MISC Subtractor	0.27	59.1	181.25	45.07	2662.7
MISC Multiplier	5.33	3.0	43.12	10.72	32.1
MISC Divider	9.05	1.8	4.23	1.05	1.9
MISC AND	0.11	145.0	795.22	197.76	28675.4
MISC OR	0.11	145.0	795.22	197.76	28675.4
MISC EXOR	0.11	145.0	795.22	197.76	28675.4
MISC NOT	0.11	145.0	853.67	212.30	30783.0
MISC Barrel Shifter(Left,Zero)	1.51	10.6	359.27	89.35	943.8
MISC Barrel Shifter(Left,sign)	1.40	11.4	368.35	91.60	1043.6
MISC Barrel Shifter(Right,Zero)	1.56	10.2	374.24	93.07	951.6
MISC Barrel Shifter(Right,sign)	1.56	10.2	354.94	88.27	902.5
RISC ALU	15.95	1.0	4.02	1.00	1.0

TABLE III

PERFORMANCE COMPARISON OF 16 POINT FFT IN EACH PART BY MISC IMPLEMENTATION.

Instruction	Area [mm^2]	Number FFT/ MISC	Operating Frequency [MHz]	Processing Time [μs]
MISC-FFT-1	3.40	11.33	245.78	0.0041
MISC-FFT-2	16.60	2.32	73.21	0.0137
MISC-FFT-3	3.40	11.33	265.12	0.0038
MISC-FFT-4	7.65	5.04	75.87	0.0132
MISC-FFT-5	3.40	11.33	271.82	0.0037
MISC-FFT-6	1.99	19.36	296.93	0.0034
MISC-FFT-7	3.40	11.33	221.86	0.0045
MISC-FFT-8	2.05	18.80	278.04	0.0036
Total		2.32		0.0498
Full hardware FFT	38.53	1.00	42.29	0.0473
Corei7-4790			3600.00	225.0000

a personal computer with a Core i7-4790 3.6 GHz clock frequency. The processor's calculation is executed on a 32 bit floating point. Results show that the processor operation time was 225 μs . Although the processor's operation was not a fixed-point operation, the processor is produced using a more advanced process technology and a larger die than the MISC implementation. Even so, the MISC operation time is much faster than that of the personal computer: the MISC operation is 9,500 times faster than the personal computer's operation. Therefore, although this comparison is not fair in terms of calculation accuracy, we were able to demonstrate the benefits of MISC implementation.

VI. CONCLUSION

Optically reconfigurable gate arrays that can execute high-speed reconfiguration have been under development for use in high-performance computing. If a high-speed reconfiguration on a VLSI could be achieved, then MISC architecture, by which each processor has only one instruction, would be applicable to VLSI, thereby greatly enhancing VLSI performance. This paper has presented performance analysis result of a Fast Fourier Transform (FFT) implementation using MISC archi-

ture. The total performance of the MISC FFT was estimated as about twice that of the full-hardware implementation on FPGA without using any reconfiguration.

ACKNOWLEDGMENT

This research was partly supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), No. 15H02676.

REFERENCES

- [1] D. Wijesundera, A. Prakash, T. Srikanthan, "Rapid design space exploration for soft core processor customization and selection," International Conference on Field-Programmable Technology, 185 - 188, 2016.
- [2] M.M. Kinage, D.G. Khairnar, "Design and implementation of FPGA soft core processor for low power multicore Embedded system using VHDL," International Conference on Automatic Control and Dynamic Optimization Techniques, pp. 328 - 332, 2016.
- [3] A. Lindoso, L. Entrena, M. Garcia-Valderas, L. Parra, "A Hybrid Fault-Tolerant LEON3 Soft Core Processor Implemented in Low-End SRAM FPGA," IEEE Transactions on Nuclear Science, Vol. 64, Issue 1, pp. 374 - 381, 2017.
- [4] P. G. Salunke, A.M. Sayyed, "Design of embedded web server based on NIOS-II soft core processor," International Conference on Electrical, Electronics, and Optimization Techniques, pp. 488-492, 2016.
- [5] T. Fujimori, M. Watanabe, "Parallel light configuration that increases the radiation tolerance of integrated circuits," Optics Express, Vol. 25, Issue 23, pp. 28136-28145, 2017.
- [6] T. Fujimori, M. Watanabe, "High-speed scrubbing demonstration using an optically reconfigurable gate array," Optics Express, Vol. 25, Issue 7, pp. 7807-7817, 2017.
- [7] D. Seto, M. Watanabe, "Radiation-hardened optically reconfigurable gate array exploiting holographic memory characteristics," Japanese Journal of Applied Physics, vol. 54, no. 9S, pp. 09MA06-1 - 09MA06-5, 2015.
- [8] R. Moriwaki, M. Watanabe, "Optical configuration acceleration on a new optically reconfigurable gate array VLSI using a negative logic implementation," Applied Optics, Vol. 52, No. 9, pp. 1939-1946, 2013.
- [9] M. Nakajima, M. Watanabe, "Fast optical reconfiguration of a nine-context DORGA using a speed adjustment control," ACM Transaction on Reconfigurable Technology and Systems, Vol. 4, Issue 2, 2011.
- [10] T. Mabuchi, M. Watanabe, "A superimposing acceleration and optimization method of optical reconfiguration speed without any increase of laser power," Applied Optics, Vol. 49, No. 22, pp. 4120-4126, 2010.
- [11] Y. Yamaji, M. Watanabe, "A 256-configuration-context MEMS optically reconfigurable gate array," International Conference on Solid State Devices and Materials, pp. 232-233, 2012.

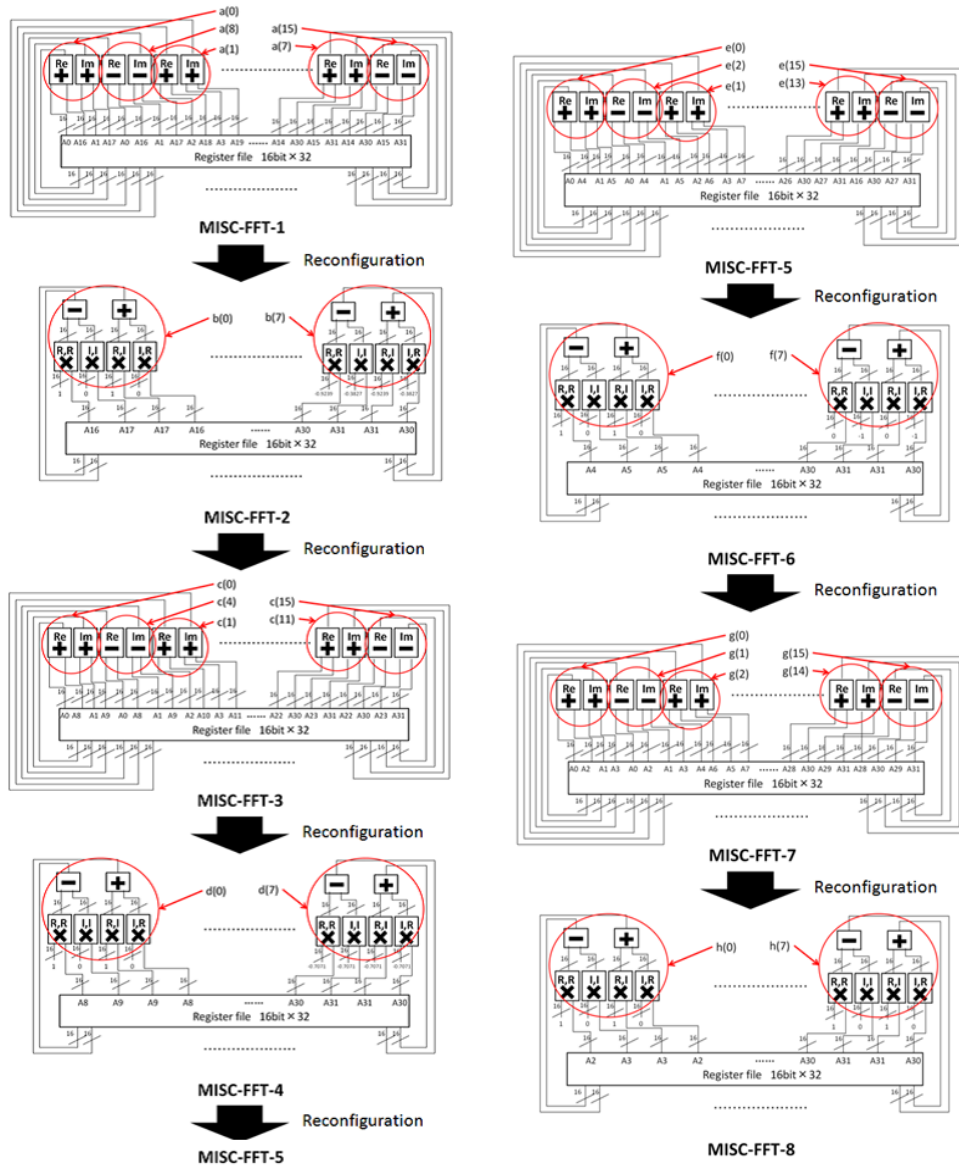


Fig. 8. Reconfiguration procedure of the 16-point FFT operation realized by MISC processors.

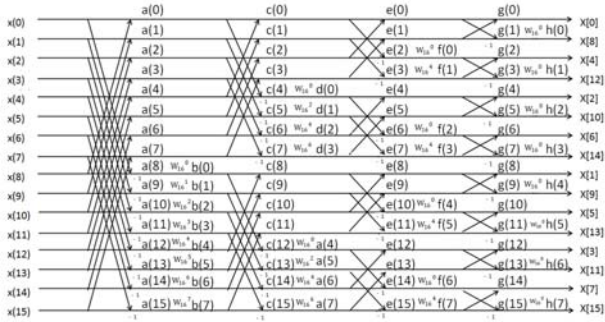


Fig. 9. 16 point FFT operation.