A Four-Context Optically Differential Reconfigurable Gate Array

Mao Nakajima, *Student Member, IEEE*, and Minoru Watanabe, *Member, IEEE*

Abstract—Reconfiguration applications based on reconfigurable devices present new computational paradigms since, by increasing the reconfiguration frequency of reconfigurable devices, their activity and performance can be improved dramatically. Recently, Optically Reconfigurable Gate Arrays (ORGAs) with a holographic memory were developed. They realize rapid reconfigurations and numerous reconfiguration contexts. Furthermore, Optically Differential Reconfigurable Gate Arrays (ODRGAs) have been developed to accelerate optical reconfigurations of conventional ORGAs. However, fast configuration experiments under multiple contexts exploiting the ODRGA architecture have never been reported. Therefore, this paper presents a four-context ODRGA system and experimental results demonstrating its fast reconfiguration. The advantage of the ODRGA architecture is discussed based on those results.

Index Terms—Field programmable gate arrays, holographic memories, optical data processing, programmable logic devices, semiconductor laser arrays.

I. INTRODUCTION

FIELD Programmable Gate Arrays (FPGAs) have been used widely in recent years because of their flexible reconfiguration capabilities [1]–[3]. Moreover, demand for high-speed reconfigurable devices has been increasing. If circuit information can be programmed rapidly from a memory to a gate array, idle circuits can be removed; only necessary circuits need be programmed into the gate array at the time, thereby increasing the gate-array activity. Moreover, such devices offer the possibility of providing a virtual gate count that is much larger than those of currently available VLSIs.

For those reasons, reconfiguration applications based on such reconfigurable devices can introduce new computational paradigms.

Nevertheless, because FPGA reconfiguration requires more than several milliseconds and because its gate array can not be used during reconfiguration, FPGAs are unsuitable devices for dynamic reconfiguration.

Manuscript received September 10, 2008; revised March 21, 2009. First published June 02, 2009; current version published August 28, 2009. This work was supported in part by the Ministry of Internal Affairs and Communications of Japan under the Strategic Information and Communications R&D Promotion Programme (SCOPE) and in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research on Innovative Areas, No. 20200027, 2009.

The authors are with the Faculty of Engineering, Shizuoka University, Hamamatsu 432-8561, Japan.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JLT.2009.2024173

On the other hand, high-speed reconfigurable devices have been developed, e.g., DAP/DNA chips, DRP chips, and multi-context FPGAs [4]–[9]. Such devices package reconfiguration memories and a microprocessor array or a gate array onto a chip. The internal reconfiguration memory stores reconfiguration contexts of 4–16 banks, which can be changed from one to another during a clock cycle. Thereby, an arithmetic logic unit or a gate array of such devices can be reconfigured on every clock cycle of a few nanoseconds. However, increasing the amount of internal reconfiguration memory while simultaneously maintaining the gate density is extremely difficult because the gate density and the amount of internal reconfiguration memory present a tradeoff relation.

As another rapid configuration approach, optically programmable gate arrays (OPGAs) with a holographic memory have been developed [10]–[12]. In the OPGAs, many reconfiguration contexts can be stored in a holographic memory. Those contexts are subsequently read out by a laser array in a very short time. Therefore, the architecture allows both realization of fast reconfigurations and numerous reconfiguration contexts. However, the OPGA has an issue in the OPGA-VLSI part. The OPGA-VLSI consists of an optical receiver array and a gate array with configuration SRAM. The construction of a gate array with a configuration SRAM is the same as that of FPGAs. Because the configuration data are transferred serially between the optical receiver array and the configuration SRAM of the gate array, the reconfiguration period is limited to 16–20 μ s. Moreover, the architecture prohibits the use of the gate array during reconfiguration. Consequently, this device is also unsuitable for dynamic reconfigurations.

To realize fast reconfiguration, an Optically Differential Reconfigurable Gate Array (ODRGA) has been developed [13], [14]. The ODRGA architecture facilitates both rapid configuration and the use of a gate array during reconfiguration. Moreover, the ODRGA architecture has a much higher reconfiguration frequency than those of either conventional ORGAs or OPGAs.

To date, optical configuration records have been reported as follows. A dynamic ORGA has achieved a 1.59 ms holographic reconfiguration [15]. An OPGA has demonstrated a 16 μ s holographic configuration [10]–[12]. In addition, as a method not using holographic memory, a two-context configuration system has demonstrated 220 ns reconfiguration using image formulation [16]. Nevertheless, this latter method had an important shortcoming: its reconfiguration contexts were limited to two. On the other hand, the configuration contexts are not limited in the holographic configuration. Therefore, its multicontext implementation must be a holographic configuration.

Fig. 1. Relationship between photodiode response frequency and its incoming light intensity.

This paper presents a four-context holographic configuration system and the experimental results of fast reconfigurations. The advantages of the ODRGA architecture are discussed based on those results.

II. ODRGA ARCHITECTURE

A. Holographic Configuration Property

Since an ORGA receives optical configuration contexts on photodiodes and because the photodiode response frequency is typically proportional to the light intensity, as shown in Fig. 1, high-intensity light reduces the period of optical reconfiguration. Therefore, the easiest way to reduce reconfiguration periods is to use high-powered lasers. However, the uses of such high-power lasers increase the unit's power consumption and might require the use of a cooling system. Such a cooling system would greatly increase the package size; for that reason, high-power lasers should be used only as a last resort.

In an ORGA, a holographic memory generates optical configuration contexts. A holographic memory has a property by which the light intensity of each bit diffracted from a holographic memory is inversely proportional to the number of bright bits included in a configuration context. Summarizing those points very simply, if the number of bright bits in a certain configuration context can be decreased, the reconfiguration speed of the context can be accelerated with no increase of laser power. The differential reconfiguration architecture is an architecture that reduces the number of such bright bits included in a configuration context.

Fig. 2 presents the experimental results of light intensity of each bit diffracted from a holographic memory. Fig. 2(a) and

(d) shows examples of a configuration context including only one bright bit. In addition, the light intensities of bright bits of configuration contexts including 12 bright bits. Then 23 bright bits are shown respectively in Fig. 2(b) and (e) and in Fig. 2(c) and (f). It can be confirmed from the experimental results that the light intensity of each bright bit of a context including fewer bright bits is higher than that of a context including a greater number of bright bits. Consequently, with fewer bright bits, the reconfiguration frequency can be radically increased. The relation between the reconfiguration period and the number of bright bits in a configuration context is presented in Fig. 3. The result was measured using a fabricated prototype ORGA.

B. Differential Reconfiguration Architecture

A differential reconfiguration architecture of ODRGAs enables bit-by-bit reconfiguration based on the difference between a previous reconfiguration context and the subsequent reconfiguration context. That difference is programmed in a holographic memory as a reconfiguration context. A programmable gate array in an ODRGA-VLSI can store a context. A reconfiguration is executed using the previous context stored in ODRGA-VLSI and a different context in a holographic memory. The following equation describes operation of an optical differential reconfiguration using bit-length N generalized from the length of configuration bits on an ODRGA. The i -th configuration pattern is represented in the form of N -dimensional vectors as

$$\alpha_i = (\alpha_1, \alpha_2, \dots, \alpha_N) \quad (1)$$

where each element of the vectors takes a binary value $\{0, 1\}$. It is assumed ex-ante that the prior configuration pattern α_i has

Fig. 2. Light intensity of each bit of optical configuration context diffracted from a holographic memory. Panels (a) and (d) show the light intensity of a configuration context including a single bright bit. Panels (b) and (e) show the light intensity of a configuration context including 12 bright bits. Panels (c) and (f) show the light intensity of a configuration context including 23 bright bits.

already been stored on a VLSI chip and that a subsequent configuration pattern α_{i+1} has already been determined. The i -th difference vector γ_i is defined in the following equation as

$$\gamma_i = \alpha_{i+1} \oplus \alpha_i \quad (2)$$

where $\oplus$ represents an exclusive-or operation between binary vectors; also, γ_i expresses the i -th reconfiguration pattern, which is written to an optical holographic memory. A differential reconfiguration procedure is performed using an exclusive-or operation between the difference vector γ_i and the prior configuration vector α_i as follows:

$$\alpha_{i+1} = \gamma_i \oplus \alpha_i. \quad (3)$$

This fact indicates that the subsequent configuration vector α_{i+1} is generated based on the previous configuration vector α_i , which is stored on the VLSI chip, and that an arbitrary location of the gate array on the VLSI chip can be reconfigured where elements of the difference vector γ_i equal 1, corresponding to the bright state of light. Therefore, that differential reconfiguration strategy enables bit-by-bit reconfiguration of an arbitrary area. Since dynamic reconfiguration is always executed partially and not entirely, the bit-by-bit reconfiguration capability can reduce the number of bright bits included in a reconfiguration context drastically, so that the reconfiguration frequency can be accelerated.

C. Inversion Technique

To realize a higher-speed reconfiguration in addition to the effect of the differential reconfiguration architecture of a

bit-by-bit reconfiguration, an inversion technique is introduced. In this strategy, each configuration datum for logic blocks, switching matrices, and I/O blocks are divided into small segments. For small segments, the inversion technique is applied. The following discussion uses bit-length N_s , which represents the number of bits included in a small segment. Similarly to the definition of the differential reconfiguration architecture, the i -th configuration vector α_i and the i -th difference vector γ_i are indicated in the forms of N_s -dimensional vector as

$$\alpha_i = (\alpha_1, \alpha_2, \dots, \alpha_{N_s}) \quad (4)$$

$$\gamma_i = (\gamma_1, \gamma_2, \dots, \gamma_{N_s}) \quad (5)$$

where each element of the vectors takes a binary value $\{0, 1\}$. In addition, the i -th difference vector γ_i is defined in the following equation as

$$\begin{aligned} \gamma_i &= \alpha_{i+1} \oplus \alpha_i \oplus C_i \\ I_{i+1} &= C_i \oplus I_i \end{aligned} \quad (6)$$

where I_i , I_{i+1} , and C_i mean the current i -th inverse variable, the successive $i + 1$ -th inverse variable, and the condition variable, which also take a binary value $\{0, 1\}$, the role of which is to decide whether the exclusive OR operator takes α_{i+1} and the current configuration vector α_i or α_{i+1} and its inversion vector $\bar{\alpha}_i$. Condition variable C_i is given as shown below

$$C_i = \begin{cases} 1 : \sum_{j=1}^{N_s} (\alpha_{i+1}(j) \oplus \alpha_i(j)) \geq \left\lceil \frac{N_s}{2} + 1 \right\rceil \\ 0 : \text{otherwise.} \end{cases} \quad (7)$$

In that equation, $\lceil N_s/2 + 1 \rceil$ denotes that $N_s/2 + 1$ is rounded down to the nearest whole number. A subsequent configuration vector α_{i+1} in the same segment is generated by calculating

Fig. 3. Relation between the reconfiguration period and the number of bright bits in a configuration context.

exclusive OR operations between the difference vector γ_i and the current configuration vector α_i , as follows:

$$\alpha_{i+1} = \gamma_i \oplus \alpha_i. \quad (8)$$

Therefore, the next configuration vector α_{i+1} is generated using the current configuration vector α_i when the next configuration vector α_{i+1} is more similar to the current configuration vector α_i than its inversion vector $\bar{\alpha}_i$. In contrast, the vector α_{i+1} is generated using vector α_i when an adverse condition obtains. Here, the difference vector γ_i and the condition variable C_i are programmed for a holographic memory.

The reduction efficiency of the number of bright bits of the inversion configuration method is discussed. Here, it is assumed that configuration contexts are given continuously for an ORGA-VLSI and that they uniformly include all possible patterns. For example, regarding 4-bit configuration, all possible patterns means 16 patterns of “0000”, “0001”..., and “1111”.

Under such a condition, first, the reduction efficiency of the number of bright bits in a configuration context of conventional ORGAs is estimated. The average number of 1s corresponding to laser irradiation is calculated by counting bit 1 of all possible vectors and dividing that quantity by $N_s 2^{N_s}$ of the summation of bits of all possible vectors, as in the following equation:

$$\kappa_{\text{ORGA}} = \frac{\sum_{r=1}^{N_s} r \cdot N_s C_r}{2^{N_s} N_s} = \frac{1}{2} \quad (9)$$

where $N C_r$ is a combination. Similarly, the reduction of the number of bright bits included in a configuration context of the inversion configuration method can be estimated as presented in Table I. The average number of ‘1’ writings corresponding to

TABLE I
EFFECTIVENESS OF THE INVERSION TECHNIQUE

α_4	α_3	α_2	α_1	γ_5	γ_4	γ_3	γ_2	γ_1	Num. of Bright Bits
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	0	1
0	0	1	1	0	0	0	1	1	2
0	1	0	0	0	0	1	0	0	1
0	1	0	1	0	0	1	0	1	2
0	1	1	0	0	0	1	1	0	2
0	1	1	1	1	1	0	0	0	2
1	0	0	0	0	1	0	0	0	1
1	0	0	1	0	1	0	0	1	2
1	0	1	0	0	1	0	1	0	2
1	0	1	1	1	0	1	0	0	2
1	1	0	0	0	1	1	0	0	2
1	1	0	1	1	0	0	1	0	2
1	1	1	0	1	0	0	0	1	2
1	1	1	1	1	0	0	0	0	1
Average =									1.526

laser irradiation is calculated by counting the ‘1’ bits of all possible vectors and dividing that number by $N 2^N$ of the summation of bits of all possible vectors, as in the following equation:

$$\kappa_{\text{new}} = \frac{\sum_{r=1}^{[N_s/2]} r \cdot N_s C_r}{2^{N_s} N_s} + \frac{\sum_{r=[N_s/2+1]}^{N_s} (N_s - r + 1) \cdot N_s C_r}{2^{N_s} N_s}. \quad (10)$$

The first term in the right side of the upper (10) is identical to (9). In this case, an inversion bit is equal to 0. In addition, the second term in the right side of the upper (10) represents the case in which the inversion bit is equal to 1. Table I portrays a calculation example of the number of bright bits in a configuration context including four configuration bits. In this case, the average number of bright bits can be decreased from 2.0 of conventional ORGAs to 1.526. Using this inversion method, in the case of four bits, about 22% of bright bits are removable. Consequently, the reconfiguration frequency can be increased.

Fig. 4. Schematic diagram of a configuration circuit including an inversion circuit. A toggle flip-flop at the left side of this figure generates an inversion signal I_i . The other flip-flops and EXORs generate configuration signals α_i for a gate array.

D. Differential Reconfiguration Circuit

Fig. 4 portrays a circuit diagram of a single-segment differential reconfiguration circuit. The configuration circuit consists of refresh transistors, photodiodes, toggle flip-flops (T-FFs), and exclusive-OR gates (EXORs). The configuration procedure is begun by activating refresh transistors to charge the junction capacitances of photodiodes. After charging is completed, laser light is irradiated onto the photodiodes. The states of photodiodes that have received light become '0' and the states of others become '1'. Then, the states are latched by toggle-flip-flops. The toggle-flip-flops connected to photodiodes that have received light flip and the other toggle-flip-flops retain their previous states. The execution means a differential reconfiguration strategy. In addition, a toggle flip-flop with role of the inverse variable is the left-most side of toggle-flip-flops. The output of the toggle flip-flop determines whether the outputs of the other toggle-flip-flops are reversed or not.

III. FOUR-CONTEXT ODRGA SYSTEM I

A. Holographic Memory Calculation

Here, the calculation method of a holographic memory is described. A hologram for ODRGAs is assumed as a thin holographic medium. A laser aperture plane, a holographic plane, and an ODRGA-VLSI plane are parallelized. The laser beam is expanded and is assumed for which the aperture size is fully wide for the holographic medium. Therefore, the laser beam can be considered as a plane wave. The reference wave from the laser propagates into the holographic plane. The holographic medium comprises rectangular pixels on the $x_1 - y_1$ holographic plane. The pixels are assumed as analog values. On the other hand, the input object is made up of rectangular pixels on the $x_2 - y_2$ object plane. The pixels can be modulated to be either on or off. The intensity distribution of a holographic medium is calculable using the following equation:

$$H(x_1, y_1) \propto \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} O(x_2, y_2) \sin(kr) dx_2 dy_2, \\ r = \sqrt{Z_L^2 + (x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (11)$$

In that equation, $O(x_2, y_2)$ is a binary value of a reconfiguration context, k is the wave number, and Z_L represents the distance between the holographic plane and the object plane. The value $H(x_1, y_1)$ is normalized as 0–1 for the minimum intensity $H_{\min}$ and maximum intensity $H_{\max}$ as the following:

$$H'(x_1, y_1) = \frac{H(x_1, y_1) - H_{\min}}{H_{\max} - H_{\min}}. \quad (12)$$

Finally, the normalized image H' is used for implementing the holographic memory. The other areas on the holographic plane are opaque to the illumination.

B. Experimental System

For confirmation of the ODRGA architecture's fast-configuration advantages, first, an ODRGA holographic memory system not using the differential reconfiguration strategy and the ODRGA architecture's inversion technique were demonstrated. The ODRGA holographic memory system has four configuration contexts. It includes liquid crystal spatial light modulator (LC-SLM) as a holographic memory and a 532 nm – 300 mW laser as laser light sources, as portrayed in Fig. 5. The photograph is shown in Fig. 6. In this experiment, one laser was shared for four configurations because of experimental resources. The laser beam was collimated; the beam was incident to a holographic memory on the LC-SLM. The holographic memory was calculated using (11) and (12), as shown in Fig. 7(e). The LC-SLM's recording area is divided to four areas, each consisting of 300×300 pixels and each including configuration information of a XNOR circuit, an AND circuit, a XOR circuit, or a NOR circuit. The LC-SLM is a projection TV panel (L3P07X-31G0; Seiko Epson Corp.). It is a 90° twisted nematic device with a thin film transistor. The panel consists of $1,024 \times 768$ pixels, each of which is $14 \times 14 \mu\text{m}^2$. The LC-SLM is connected to an evaluation board (L3B07X-E10A; Seiko Epson Corp.). The video input of the board is connected to the external display terminal of a personal computer. Programming for the LC-SLM is executed by displaying a holographic memory pattern of green with 256 gradation levels on the personal computer display. A

Fig. 5. Experimental system of an entire ODRGA multicontext holographic reconfiguration architecture.

Fig. 6. Photograph of the experimental system of Fig. 5. In this figure, panel (a) presents a photograph of the entire experimental system. Panel (b) shows an expanded photograph of the experimental system around the ODRGA-VLSI. Panel (c) portrays an ODRGA board. Panel (d) shows an ODRGA-VLSI chip photograph.

holographic pattern in Fig. 7(e) is displayed on the LC-SLM. The ODRGA-VLSI was placed at a distance of 100 mm from the LC-SLM.

In this experiment, a 0.35 μm triple-metal CMOS process-fabricated ORGA-VLSI chip like that shown in Fig. 6(c) was used. The photodiodes were constructed between the N-Well layer and the P-substrate. The photodiode size and distance between photodiodes were designed as $25.5 \times 25.5 \mu\text{m}^2$ and as 90 μm to facilitate the optical alignment. The gate array structure is fundamentally identical to that of typical FPGAs. The ORGA-VLSI chip includes 4 logic blocks, 5 switching matrices, and 12 I/O bits. The number of photodiodes used to program the gate array is 340.

C. Experimental Results

Using the previously described experimental system setup, the configurations were executed. Four configuration contexts of a XNOR circuit, an AND circuit, a XOR circuit, and a NOR circuit were programmed onto four areas of the LC-SLM. Their

Fig. 7. Configuration context patterns and a multicontext holographic memory. Panels (a), (b), (c), and (d), respectively, present context patterns of an XNOR circuit, an AND circuit, an XOR circuit, and a NOR circuit. Panel (e) portrays a calculated multicontext holographic memory pattern storing four configuration context information of the XNOR circuit, the AND circuit, the XOR circuit, and the NOR circuit.

CCD-captured images at the position of ODRGA-VLSI are depicted in Fig. 8. The reconfiguration periods of the XNOR circuit, the AND circuit, the XOR circuit, and the NOR circuit were measured respectively as 812 ns, 1,000 ns, 854 ns, and 917 ns. The results are much faster than those obtained using

Fig. 8. CCD-captured images of the configuration contexts of the XNOR circuit, the AND circuit, the XOR circuit and the NOR circuit at the ODRGA-VLSI position.

currently available FPGAs. However, these experimental results were obtained using a normal configuration method; the experiments were not executed with the differential reconfiguration strategy and the inversion technique of the ODRGA architecture. Section IV presents a description of the improvements obtained when using the differential reconfiguration strategy and the inversion technique.

IV. FOUR CONTEXT EXPERIMENTS EXPLOITING DIFFERENTIAL RECONFIGURATION AND INVERSION TECHNIQUE

A. Experimental Results

Successive configurations were executed using the experimental system setup described previously. Here, only the collimated beam size differs from that used in the previous experiment. A configuration context of an AND circuit and a reconfiguration context from the AND circuit to an OR circuit exploiting the differential reconfiguration architecture were programmed onto the LC-SLM. Fig. 9(a) and (b), respectively, portrays the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the OR circuit. Fig. 9(c) portrays a calculated multicontext holographic memory pattern storage of the AND circuit and the OR circuit. First, configuration of the AND circuit was executed, as portrayed in Fig. 9(f). A CCD-captured image of the configuration context of the AND circuit at the position of the ODRGA-VLSI is depicted in Fig. 9(d). The optical configuration context was programmed onto the ODRGA-VLSI. Results confirmed that the AND circuit works correctly. Then, reconfiguration from the AND circuit to an OR circuit was executed as presented in Fig. 9(f) using the differential reconfiguration strategy. A CCD-captured image of the reconfiguration context from the AND circuit to the OR circuit at the position of the ODRGA-VLSI is depicted in Fig. 9(e). After the reconfiguration, it was confirmed that the OR circuit works correctly. Finally, the number of bright bits for such a reconfiguration was reduced to two bits. The reconfiguration time was confirmed as 83.3 ns by exploiting the ODRGA architecture.

Fig. 9. Reconfiguration experiment from an AND circuit to an OR circuit. Panels (a) and (b) show the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the OR circuit, respectively. Panel (c) portrays a calculated multicontext holographic memory pattern storing the AND circuit and the OR circuit. Panel (d) and (e) show CCD-captured images of the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the OR circuit, respectively. Panel (f) shows the waveform of the 83.3 ns reconfiguration from the AND circuit to the OR circuit.

Moreover, a configuration context of an XOR circuit and a reconfiguration context from the XOR circuit to a NOR circuit

exploiting the differential reconfiguration architecture were programmed onto the LC-SLM. Fig. 10(a) and (b), respectively, show the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the NOR circuit. Fig. 10(c) portrays a calculated multicontext holographic memory pattern storing the XOR circuit and the NOR circuit. First, configuration of the XOR circuit was executed, as portrayed in Fig. 10(f). A CCD-captured image of the configuration context of the XOR circuit at the position of the ODRGA-VLSI is depicted in Fig. 10(d). The optical configuration context was programmed onto the ODRGA-VLSI. Results therefore confirmed that the XOR circuit works correctly. Then, reconfiguration from the XOR circuit to a NOR circuit was executed using the differential reconfiguration strategy, as presented in Fig. 10(f). A CCD-captured image of the reconfiguration context from the XOR circuit to the NOR circuit at the position of the ODRGA-VLSI is depicted in Fig. 10(e). After reconfiguration, it was confirmed that the NOR circuit functions correctly. Finally, the number of bright bits for such a reconfiguration was reduced to two bits. The reconfiguration time was confirmed as 83.3 ns by exploiting the ODRGA architecture.

Next, the inversion technique advantage was demonstrated. A configuration context of an AND circuit and a reconfiguration context from the AND circuit to a NAND circuit exploiting the differential reconfiguration architecture and the inversion technique were programmed onto the LC-SLM. Fig. 11(a) and (b), respectively, depicts the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the NAND circuit. Fig. 11(c) portrays a calculated multicontext holographic memory pattern storing the AND circuit and the NAND circuit. First, a configuration of the AND circuit was executed, as portrayed in Fig. 11(f). A CCD-captured image of the configuration context of the AND circuit at the position of the ODRGA-VLSI is depicted in Fig. 11(d). The optical configuration context was programmed onto the ODRGA-VLSI. Results confirmed that the AND circuit works correctly. Then, a reconfiguration from the AND circuit to a NAND circuit was executed as presented in Fig. 11(f) using the differential reconfiguration strategy and the inversion technique. A CCD-captured image of the reconfiguration context from the AND circuit to the NAND circuit at the position of the ODRGA-VLSI is depicted in Fig. 11(e). After reconfiguration, it was confirmed that the NAND circuit works correctly. Finally, the number of bright bits for such reconfiguration was reduced to a bit. The reconfiguration time was confirmed as 41.7 ns by exploiting the ODRGA architecture.

Moreover, a configuration context of an XOR circuit and a reconfiguration context from the XOR circuit to an XNOR circuit exploiting the differential reconfiguration architecture and the inversion technique were programmed onto the LC-SLM. Fig. 12(a) and (b), respectively, presents the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the XNOR circuit. Fig. 12(c) portrays a calculated multicontext holographic memory pattern storing the XOR circuit and the XNOR circuit. First, a configuration of the XOR circuit was executed, as portrayed in Fig. 12(f). A CCD-captured image of the configuration context

Fig. 10. Reconfiguration experiment from an XOR circuit to a NOR circuit. Panels (a) and (b) show the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the NOR circuit, respectively. Panel (c) portrays a calculated multicontext holographic memory pattern storing the XOR circuit and the NOR circuit. Panel (d) and (e) show CCD-captured images of the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the NOR circuit, respectively. Panel (f) shows the waveform of the 83.3 ns reconfiguration from the XOR circuit to the NOR circuit.

of the XOR circuit at the position of the ODRGA-VLSI is depicted in Fig. 12(d). The optical configuration context was

Fig. 11. Reconfiguration experiment from an AND circuit to a NAND circuit. Panels (a) and (b) show the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the NAND circuit, respectively. Panel (c) portrays a calculated multicontext holographic memory pattern storing the AND circuit and the NAND circuit. Panel (d) and (e) show CCD-captured images of the configuration context pattern of the AND circuit and the reconfiguration context pattern from the AND circuit to the NAND circuit, respectively. Panel (f) shows the waveform of the 41.7 ns reconfiguration from the AND circuit to the NAND circuit.

programmed onto the ODRGA-VLSI. Consequently, the XOR circuit was confirmed to have worked correctly. Then, recon-

Fig. 12. Reconfiguration experiment from an XOR circuit to an XNOR circuit. Panels (a) and (b) show the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the XNOR circuit, respectively. Panel (c) portrays a calculated multicontext holographic memory pattern storing the XOR circuit and the XNOR circuit. Panel (d) and (e) show CCD-captured images of the configuration context pattern of the XOR circuit and the reconfiguration context pattern from the XOR circuit to the XNOR circuit, respectively. Panel (f) shows the waveform of the 41.7 ns reconfiguration from the XOR circuit to the XNOR circuit.

reconfiguration of an XOR circuit to an XNOR circuit was executed as presented in Fig. 12(f) using the differential reconfi-

uration strategy and the inversion technique. A CCD-captured image of the reconfiguration context from the AND circuit to the XNOR circuit at the position of the ODRGA-VLSI is depicted in Fig. 12(e). After the reconfiguration, the XNOR circuit was verified as functioning correctly. Finally, the number of bright bits for such reconfiguration was reduced to a bit. The reconfiguration time was confirmed as 41.7 ns by exploiting the ODRGA architecture.

Finally, the number of bright bits for such reconfigurations was reduced to a bit. The reconfiguration time was confirmed as 41.7 ns by exploiting the ODRGA architecture. It must be emphasized that the 41.7 ns optical reconfiguration is the world's fastest result. Moreover, the result shows that the circuit functions during the reconfiguration process. For this reason, in the four-context ODRGA, fast configuration was confirmed with no idling of the gate array. The ODRGA architecture is a useful technique to achieve rapid reconfiguration.

V. DISCUSSION FOR PRACTICAL APPLICATIONS

This section presents discussion of some issues related to ODRGAs' future practical applications and necessary actions.

A. Gate Count

In ODRGAs, although a large virtual gate count can be realized using the large storage capacity of a holographic memory, an actual gate count, which is the gate count of a programmable gate array VLSI, is important to increase the instantaneous performance. Although the real gate count of the first prototype 68-gate ODRGA-VLSI chip was too small for practical applications, the future prospect of ODRGA-VLSIs were already reported to be as large as FPGAs [18]. In the case of 45 nm process, the gate count was estimated as 7 k gates/mm². Therefore, 2.8 million gates can be implemented on a 20 mm × 20 mm die chip. Precise comparison reveals that the ODRGA gates are slightly fewer than those of FPGAs. However, given 1,000,000 reconfiguration contexts in a holographic memory and a 1 million gate count ODRGA-VLSI, then a 1 Tera gate count VLSI is possible. Therefore, the ODRGA presents the possibility of realizing huge virtual gate-count VLSIs. It therefore presents a gate count advantage compared to FPGAs.

B. Alignment Precision

The ODRGA consists of a holographic memory, a laser diode array, and a gate array VLSI. Of course, although the ODRGAs are extremely advanced devices, the ODRGA requires high-precision alignment between a holographic memory and an ODRGA-VLSI and between a laser array and the holographic memory. To date, such an alignment has been analyzed theoretically and experimentally [17]. Results show that 1 μm precision alignment is sufficient to assemble ODRGAs. Therefore, ODRGA packages can be assembled using currently available manufacturing technologies.

C. Power Consumption

The power consumption issue is a salient point of difficulty related to dynamic reconfiguration devices. For ODRGAs, optical power consumption might become the main component of power consumption. For example, as an ODRGA-VLSI

gate count increases, necessary optical power consumption is also increased, as shown in Fig. 3. Here, programming for a 1 million fine grain gate array is assumed. In this case, about 3,000,000 photodiodes are necessary to program it. Each photodiode of the current prototype ODRGA-VLSI can respond to 2.1×10^6 photons ($\lambda = 532$ nm). On the other hand, the diffraction efficiency of the liquid crystal holographic memory was about 0.05%. In this case, a 10 W laser is required for a 231 μs configuration. However, such high power consumption can be lessened by increasing the efficiency of the holographic memory, and the photodiodes. For example, the diffraction efficiency of a current amplitude-modulation-type holographic memory is not good. However, the use of phase-modulation-type holographic memory will increase its efficiency dramatically. Moreover, the photodiode sensitivity of current prototype ODRGA-VLSI chip is not good because the ODRGA-VLSI chip was not fabricated using the photodiode process. According to results presented in one paper [10], if the ODRGA-VLSI chip can use the photodiode process, the number of photons to which a photodiode can respond can be decreased to 800. In addition, the power consumption will be improved because a 100 mW laser will achieve a less than 1 μs configuration for 1 million gates if the system uses a phase-modulation-type holographic memory.

D. Reconfiguration Advantage for FPGAs

In an earlier discussion of power consumption, the reconfiguration laser power for a future ODRGA with 1 million gates was estimated. In fact, less than 1 μs reconfiguration can be achieved using reasonable laser power. Here, an FPGA with the same 1 million gates is assumed. The FPGA is assumed to be reconfigured at 100 MHz. At that time, 10 ms is required to program it. Therefore, comparing ODRGAs with FPGAs, ODRGA presents the important advantage of extremely rapid reconfiguration.

E. Dynamic Reconfiguration Advantage

Recently, almost all computer systems have come to use a reduced instruction set computer (RISC) architecture. That architecture presents advantages in terms of higher clock frequency, smaller implementation area, and lower power consumption than conventional complex instruction set computer (CISC) architecture. The success is based on the fact that, in the circuit implementation, a simple circuit is the best means to realize high-clock frequency or high performance. That principle can be adapted also to programmable devices. Currently, a general purpose unit is always implemented onto a programmable device. Reconfiguration is never used. However, if clock-by-clock reconfigurable devices can be used, a single function or a single instruction set computer can be implemented onto it. Of course, the function will be an unused circuit after a clock cycle. However, at that time, the unused circuit need only be reconfigured to another necessary circuit if we can use clock-by-clock reconfigurable devices. Such a single circuit or a single-instruction-set computer can function at an extremely high clock frequency and provide extremely high performance. The performance of a gate array can therefore be improved compared with current general purpose unit

implementation or multifunction implementation. Moreover, such a single circuit or a single-instruction-set computer can be implemented in a small implementation area or on a small die. Consequently, using the same implementation area, large parallel computation is possible. For the reasons described above, the overall performance can be increased dramatically. This is a future use of programmable devices. For that reason, a fast reconfigurable device with many reconfiguration contexts is necessary for such realization. The FPGAs' reconfiguration capability is insufficient even if FPGAs can be partially reconfigured. Therefore, ODRGAs can open a new computation paradigm.

VI. CONCLUSION

To realize faster reconfigurations than those afforded by conventional ORGAs, an Optically Differential Reconfigurable Gate Array (ODRGA) has been developed. Nevertheless, rapid configuration experiments performed under multiple contexts exploiting the ODRGA architecture have never been reported. This paper has introduced and described a demonstration of a four-context ODRGA system. First, normal configuration periods not using ODRGA architecture were measured as 812 ns, 1,000 ns, 854 ns, and 917 ns, which are faster than that of FPGAs. Moreover, the second reconfiguration time was confirmed as 41.7 ns by exploiting the ODRGA architecture. In fact, the 41.7 ns optical reconfiguration is the world's fastest result. In addition, the result shows that circuits work during the reconfiguration processes. Using this architecture, the period of time necessary for the reconfiguration procedure is negligible. Finally, in the four-context ODRGA, fast configuration with no idleness of a gate array has been confirmed, which demonstrates that the ODRGA architecture is an extremely useful technique for rapid reconfiguration.

ACKNOWLEDGMENT

The VLSI chip in this study was fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Co., Ltd., and Toppan Printing Co., Ltd.

REFERENCES

- [1] Altera Devices Altera Corporation [Online]. Available: <http://www.altera.com>
- [2] I. Xilinx, Xilinx Product Data Sheets [Online]. Available: <http://www.xilinx.com>
- [3] LatticeECP and EC Family Data Sheet Lattice Semiconductor Corporation, 2005 [Online]. Available: <http://www.latticesemi.co.jp/products>
- [4] [Online]. Available: <http://www.ipflex.co.jp>
- [5] H. Nakano, T. Shindo, T. Kazami, and M. Motomura, "Development of dynamically reconfigurable processor LSI," *NEC Tech. J. (Japan)*, vol. 56, no. 4, pp. 99–102, 2003.
- [6] A. Dehon, "Dynamically programmable gate arrays: A step toward increased computational density," in *Proc. 4th Canadian Workshop on Field Programmable Devices*, 1996, pp. 47–54.
- [7] S. M. Scalera and J. R. Vazquez, "The design and implementation of a context switching FPGA," in *Proc. IEEE Symp. FPGAs Custom Computing Machines*, 1998, pp. 78–85.

- [8] S. Trimberger *et al.*, "A time-multiplexed FPGA," *FCCM*, pp. 22–28, 1997.
- [9] D. Jones and D. M. Lewis, "A time-multiplexed FPGA architecture for logic emulation," in *Proc. Custom Integrated Circuits Conf.*, 1995, pp. 495–498.
- [10] J. Mumburu, G. Panotopoulos, D. Psaltis, X. An, F. Mok, S. Ay, S. Barna, and E. Fossum, "Optically programmable gate array," in *Proc. SPIE Optics Computing*, 2000, vol. 4089, pp. 763–771.
- [11] J. Mumburu, G. Zhou, X. An, W. Liu, G. Panotopoulos, F. Mok, and D. Psaltis, "Optical memory for computing and information processing," in *Proc. SPIE Algorithms, Devices, and Systems for Optical Information Processing III*, 1999, vol. 3804, pp. 14–24.
- [12] J. Mumburu, G. Zhou, S. Ay, X. An, G. Panotopoulos, F. Mok, and D. Psaltis, "Optically reconfigurable processors," in *Proc. SPIE Critical Rev. Euro-American Workshop on Optoelectronic Information Processing*, 1999, vol. 74, pp. 265–288.
- [13] M. Watanabe and F. Kobayashi, "An optically differential reconfigurable gate array and its power consumption estimation," in *Proc. IEEE Int. Conf. Field-Programmable Technology*, 2002, pp. 197–202.
- [14] M. Watanabe and F. Kobayashi, "An optically reconfigurable gate array VLSI with partial reconfiguration capability," *IEICE Trans. Electron.*, vol. J86-C, no. 8, pp. 869–877, 2003.
- [15] D. Seto and M. Watanabe, "A dynamic optically reconfigurable gate array—Perfect emulation," *IEEE J. Quantum. Electron.*, vol. 44, pp. 493–500, 2008.
- [16] M. Miyano, M. Watanabe, and F. Kobayashi, "Optically differential reconfigurable gate array," *Electron. Comput. Jpn. II*, vol. 90, no. 11, pp. 132–139, Nov. 2007.
- [17] H. Morita and M. Watanabe, "Allowable alignment errors of components in an optically reconfigurable gate array," in *Proc. Int. Top. Meet. Information Photonics*, 2008, pp. 50–51.
- [18] M. Watanabe, T. Shiki, and F. Kobayashi, "Scaling prospect of optically differential reconfigurable gate array VLSIs," presented at the Analog Integrated Circuits and Signal Processing, Sep. 2008.

Mao Nakajima (S'07) was born in Shizuoka, Japan, in March, 1986. He is currently a graduate student in the Department of Electrical and Electronic Engineering at Shizuoka University, Hamamatsu, Japan.

His current research interests are optically reconfigurable gate arrays.

Minoru Watanabe (M'00) received B.S. and M.S. degrees in opto-electronic and mechanical engineering from Shizuoka University, Hamamatsu, Japan, in 1992 and 1994, respectively, and the Ph.D. degree in computer science and systems engineering from the Kyushu Institute of Technology, Iizuka, Japan, in 2005.

He joined Nissan Motor Co., Ltd., Zama, Japan, in 1994, where he was a development engineer of embedded systems until December 1999. He was a Research Associate in the Department of Control Engineering and Science during January 2000–2003 and in the Department of Systems Innovation and Informatics during 2004–2006, Kyushu Institute of Technology, Iizuka, Japan. He is currently an Associate Professor in the Department of Electrical and Electronic Engineering, Shizuoka University. His current research interests are programmable devices and optically reconfigurable gate arrays.

Dr. Watanabe is a member of IEEE, the Institute of Electronics, Information, and Communication Engineers of Engineers of Japan, the Japan Society of Applied Physics, and the Society of Instrument and Control Engineers.