1

# Fault tolerance of programmable devices

Minoru Watanabe
*Shizuoka University*
*Japan*

## 1. Introduction

Currently, we are frequently facing demands for automation of many systems. In particular, demands for cars and robots are increasing daily. For such applications, high-performance embedded systems are necessary to execute real-time operations. For example, image processing and image recognition are heavy operations that tax current microprocessor units. Parallel computation on high-capacity hardware is expected to be one means to alleviate the burdens imposed by such heavy operations.

To implement such large-scale parallel computation onto a VLSI chip, the demand for a large-die VLSI chip is increasing daily. However, considering the ratio of non-defective chips under current fabrications, die sizes cannot be increased (1),(2). If a large system must be integrated onto a large die VLSI chip or as an extreme case, a wafer-size VLSI, the use of a VLSI including defective parts must be accomplished.

In the earliest use of field programmable gate arrays (FPGAs) (3)–(5), FPGAs were anticipated as defect-tolerant devices that accommodate inclusion of defective areas on the gate array because of their programmable capability. However, that hope was partly shattered because defects of a serial configuration line caused severe impairments that prevented programming of the entire gate array. Of course, a spare row method such as that used for memories (DRAMs) reduces the ratio of discarded chips (6),(7), in which spare rows of a gate array are used instead of defective rows by swapping them with a laser beam machine. However, such methods require hardware redundancy. Moreover, they are not perfect. To use a gate array perfectly and not produce any discarded VLSI chips, a perfectly parallel programmable capability is necessary: one which uses no serial transfer.

Currently, optically reconfigurable gate arrays (ORGAs) that support parallel programming capability and which never use any serial transfer have been developed (8)–(15). An ORGA comprises a holographic memory, a laser array, and a gate-array VLSI. Although the ORGA construction is slightly more complex than that of currently available FPGAs, the parallel programmable gate array VLSI supports perfect avoidance of its faulty areas; it instead uses the remaining area. Therefore, the architecture enables the use of a large-die VLSI chip and even entire wafers, including fault areas. As a result, the architecture can realize extremely high-gate-count VLSIs and can support large-scale parallel computation.

This chapter introduces an ORGA architecture as a high defect tolerance device, describes how to use an optically reconfigurable gate array including defective areas, and clarifies its high fault tolerance. The ORGA architecture has some weak points in making a large VLSI, as

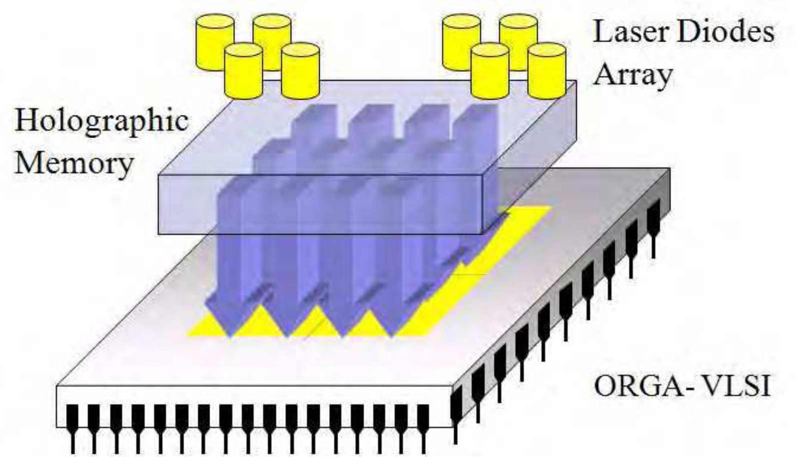


Fig. 1. Overview of an ORGA.

do FPGAs. Therefore, this chapter also presents discussion of more reliable design methods to avoid weak points.

## 2. Optically Reconfigurable Gate Array (ORGA)

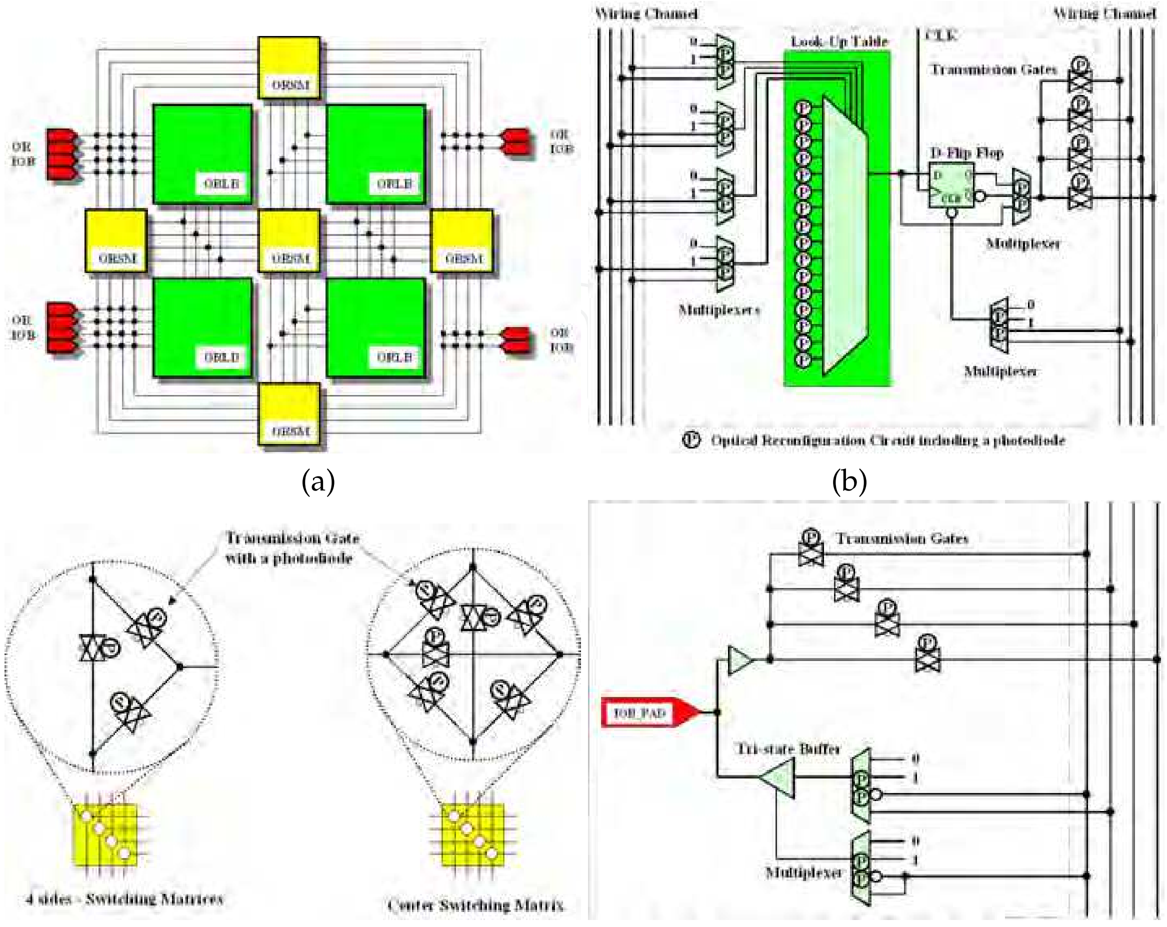
The ORGA architecture has the following features: numerous reconfiguration contexts, rapid reconfiguration, and large die size VLSIs or wafer-scale VLSIs. A large die size VLSI can produce large physical gates that increase the performance of large parallel computation. Furthermore, numerous reconfiguration contexts achieve huge virtual gates with contexts several times more numerous than those of the physical gates. For that reason, such huge virtual gates can be reconfigured dynamically on the physical gates so that huge operations can be integrated onto a single ORGA-VLSI. The following sections describe the ORGA architecture, which presents such advantages.

### 2.1 Overall construction

An overview of an Optically Reconfigurable Gate Array (ORGA) is portrayed in Fig. 1. An ORGA comprises a gate-array VLSI (ORGA-VLSI), a holographic memory, and a laser diode array. The holographic memory stores reconfiguration contexts. A laser array is mounted on the top of the holographic memory for use in addressing the reconfiguration contexts in the holographic memory. One laser corresponds to a configuration context. Turning one laser on, the laser beam propagates into a certain corresponding area on the holographic memory at a certain angle so that the holographic memory generates a certain diffraction pattern. A photodiode-array of a programmable gate array on an ORGA-VLSI can receive it as a reconfiguration context. Then, the ORGA-VLSI functions as the circuit of the configuration context. The reconfiguration time of such ORGA architecture reaches nanosecond-order (14),(15). Therefore, very-high-speed context switching is possible. Since the storage capacity of a holographic memory is extremely high, numerous configuration contexts can be used with a holographic memory. Therefore, the ORGA architecture can dynamically treat huge virtual gate counts that are larger than the physical gate count on an ORGA-VLSI.

### 2.2 Gate array structure

This section introduces a design example of a fabricated ORGA-VLSI chip. Based on it, a generalized gate array structure of ORGA-VLSIs is discussed.



(a) (b)

(c) (d)

Fig. 2. Gate-array structure of a fabricated ORGA. Panels (a), (b), (c), and (d) respectively depict block diagrams of a gate array, an optically reconfigurable logic block, an optically reconfigurable switching matrix, and an optically reconfigurable I/O bit.

### 2.2.1 Prototype ORGA-VLSI chip

The basic functionality of an ORGA-VLSI is fundamentally identical to that of currently available field programmable gate arrays (FPGAs). Therefore, ORGA-VLSI takes an island-style gate array or a fine-grain gate array. Figure 2 depicts the gate array structure of a first prototype ORGA-VLSI chip. The ORGA-VLSI chip was fabricated using a 0.35 $\mu$m triple-metal CMOS process (8). The photograph of a board is portrayed in Fig. 3. Table 1 presents the specifications. The ORGA-VLSI chip consists of 4 optically reconfigurable logic blocks (ORLB), 5 optically reconfigurable switching matrices (ORSM), and 12 optically reconfigurable I/O bits (ORIOB) portrayed in Fig. 2(a). Each optically reconfigurable logic block is surrounded by wiring channels. In this chip, one wiring channel has four connections. Switching matrices are located on the corners of optically reconfigurable logic blocks. Each connection of the switching matrices is connected to a wiring channel. The ORGA-VLSI has 340 photodiodes to program its gate array. The ORGA-VLSI can be reconfigured perfectly in parallel. In this fabrication, the distance between each photodiode was designed as 90 $\mu$m. The photodiode size was set as $25.5 \times 25.5$ $\mu$m$^2$ to ease the optical alignment. The photodiode was constructed between the N-well layer and P-substrate. The gate array's gate count is 68. It was confirmed experimentally that the ORGA-VLSI itself is reconfigurable within a nanosecond-order period

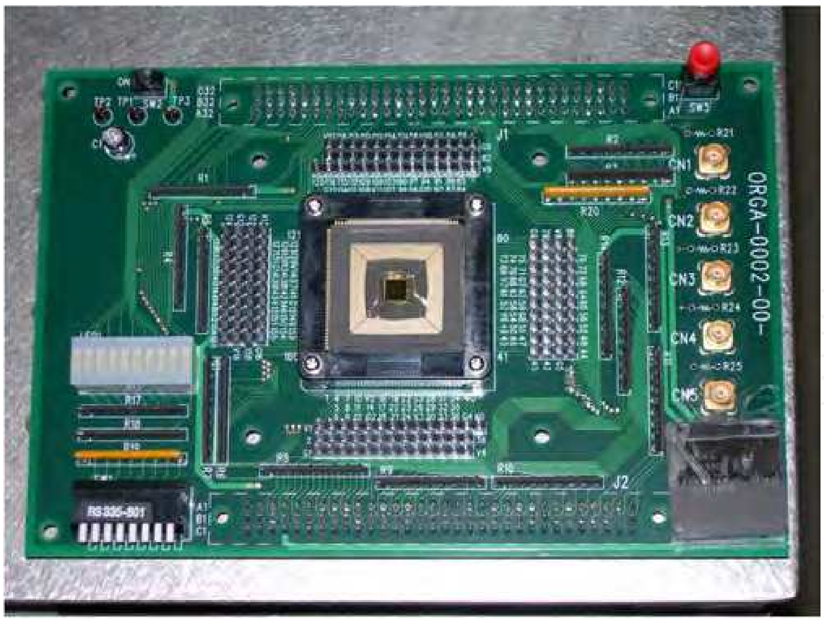


Fig. 3. Photograph of an ORGA-VLSI board with a fabricated ORGA-VLSI chip. The ORGA-VLSI was fabricated using a 0.35 $\mu$m three-metal 4.9 × 4.9 $mm^2$ CMOS process chip. The gate count of a gate array on the chip is 68. In all, 340 photodiodes are used for optical configurations.

(14),(15). Although the gate count of the chip is too small, the gate count of future ORGAs was already estimated (12). Future ORGAs will achieve gate counts of over a million, which is similar to gate counts of FPGAs.

### 2.2.2 Optically reconfigurable logic block

The block diagram of an optically reconfigurable logic block of the prototype ORGA-VLSI chip is presented in Fig. 2(b). Each optically reconfigurable logic block consists of a four-input one-output look-up table (LUT), six multiplexers, four transmission gates, and a delay type flip-flop with a reset function. The input signals from the wiring channel, which are applied through some switching matrices and wiring channels from optically reconfigurable I/O blocks, are transferred to a look-up table through four multiplexers. The look-up table is used for implementing Boolean functions. The outputs of the look-up table and of a delay type flip-flop connected to the look-up table are connected to a multiplexer. A combinational circuit and sequential circuit can be chosen by changing the multiplexer, as in FPGAs. Finally, an output of the multiplexer is connected to the wiring channel again through transmission gates. The last multiplexer controls the reset function of the delay-type flip-flop. Such a four-input one-output look-up table, each multiplexer, and each transmission gate respectively have 16 photodiodes, 2 photodiodes, and 1 photodiode. In all, 32 photodiodes are used for programming an optically reconfigurable logic block. Therefore, the optically reconfigurable logic block can be reconfigured perfectly in parallel. In this prototype chip, since the gate array is too small, a CLK for each flip-flop is provided through a single CLK buffer tree. However, for a large gate array, CLKs of flip-flops are applied through multiple CLK buffer trees as programmable CLKs, as well as that of FPGAs.

| | |
|---|---|
| Technology | 0.35$\mu$m double-poly<br>triple-metal CMOS process |
| Chip size | 4.9 mm × 4.9 mm |
| Photodiode size | 25.5 $\mu m$ × 25.5 $\mu m$ |
| Distance between photodiodes | 90 $\mu m$ |
| Number of photodiodes | 340 |
| Gate count | 68 |

Table 1. ORGA-VLSI Specifications.

### 2.2.3 Optically reconfigurable switching matrix

Similarly, optically reconfigurable switching matrices are optically reconfigurable. The block diagram of the optically reconfigurable switching matrix is portrayed in Fig. 2(c). The basic construction is the same as that used by Xilinx Inc. One four-directional with 24 transmission gates and 4 three-directional switching matrices with 12 transmission gates were implemented in the gate array. Each transmission gate can be considered as a bi-directional switch. A photodiode is connected to each transmission gate; it controls whether the transmission gate is closed or not. Based on that capability, four-direction and three-direction switching matrices can be programmed, respectively, as 24 and 12 optical connections.

### 2.2.4 Optically reconfigurable I/O block

Optically reconfigurable gate arrays are assumed to be reconfigured frequently. For that reason, an optical reconfiguration capability must be implemented for optically reconfigurable logic blocks and optically reconfigurable switching matrices. However, the I/O block might not always be reconfigured under such dynamic reconfiguration applications because such a dynamic reconfiguration arises inside the device and each mode of Input, Output, or Input/Output, and each pin location of the I/O block must always be fixed due to limitations of the external environment. However, the ORGA-VLSI supports optical reconfiguration for I/O blocks because reconfiguration information is provided optically from a holographic memory in ORGA. Consequently, electrically configurable I/O blocks are unsuitable for ORGAs. Here, each I/O block is also controlled using nine optical connections. Always, the optically reconfigurable I/O block configuration is executed only initially.

## 3. Defect tolerance design of the ORGA architecture

### 3.1 Holographic memory part

Holographic memories are well known to have a high defect tolerance. Since each bit of a reconfiguration context can be generated from the entire holographic memory, the damage of some fraction rarely affects its diffraction pattern or a reconfiguration context. Even though a holographic memory device includes small defect areas, holographic memories can correctly record configuration contexts and can correctly generate configuration contexts. Such mechanisms can be considered as those for which majority voting is executed from an infinite number of diffraction beams for each configuration bit. For a semiconductor memory, single-bit information is stored in a single-bit memory circuit. In contrast, in holographic memory, a single bit of a reconfiguration context is stored in the entire holographic memory. Therefore, the holographic memory's information is robust while, in the semiconductor memory, the defect of a transistor always erases information of a single bit or multiple bits. Earlier studies have shown experimentally that a holographic memory is robust (13). In the experiments, 1000 impulse noises and 10% Gaussian noise were applied to a holographic memory. Then the holographic memory was assembled to an ORGA architecture. All configuration experiments were successful. Therefore, defects of a holographic memory device on the ORGA are beyond consideration.

### 3.2 Laser array part

In an ORGA, a laser array is a basic component for addressing a configuration memory or a holographic memory. Although configuration context information stored on a holographic memory is robust, if the laser array becomes defective, then the execution of each configuration becomes impossible. Therefore, the defect modes arising on a laser array must be analyzed. In an ORGA, many discrete semiconductor lasers are used for switching configuration contexts. Each laser corresponds to one holographic area including one configuration context. One laser addresses one configuration context. The defect modes of a certain laser are categorizable as a turn-ON defect mode and a full-time turn-ON defect mode or a turn-OFF defect mode. The turn-ON defect mode means that a certain laser cannot be turned on. The full-time turn-ON defect mode means the state in which a certain laser is constantly turned ON and cannot be turned OFF.

#### 3.2.1 Turn-ON defect mode

A laser might have a Turn-ON defect. However, laser source defects can be avoided easily by not using the defective lasers, and not using holographic memory areas corresponding to the lasers. An ORGA has numerous reconfiguration contexts. A slight reduction of reconfiguration contexts is therefore negligible. Programmers need only to avoid the defective parts when programming reconfiguration contexts for a holographic memory. Therefore, the ORGA architecture allows Turn-ON defect mode for lasers.

#### 3.2.2 Turn-OFF defect mode

Furthermore, a laser might have a Turn-OFF defect mode. This trouble level is slightly higher than that of the Turn-ON defect mode. The corresponding holographic memory information is constantly superimposed to the other configuration context under normal reconfiguration procedure if one laser has Turn-OFF defect mode and turns on constantly. Therefore, the Turn-OFF defect mode of lasers presents the possibility that all normal configuration procedures are impossible. Therefore, if such Turn-OFF defect mode arises on an ORGA, a physical action to cut the corresponding wires or driver units is required. The action is easy and can perfectly remove the defect mode.

#### 3.2.3 Defect mode for matrix addressing

Such laser arrays are always arranged in the form of a two-dimensional matrix and addressed as the matrix. In such matrix implementation, the defect of one driver causes all lasers on the addressing line to be defective. To avoid simultaneous defects of many lasers, a spare row method like that used for memories (DRAMs) is useful (6)(7). By introducing the spare row method, the defect mode can be removed perfectly.

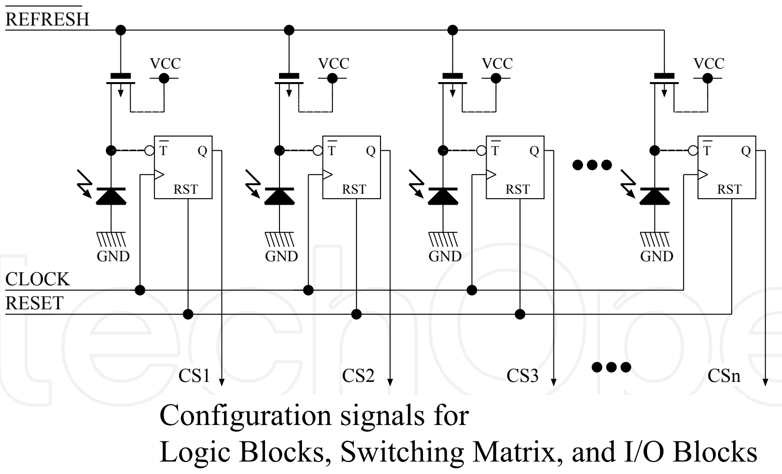

REFRESH
VCC
VCC
VCC
VCC
T
Q
RST
GND
GND
GND
GND
CLOCK
RESET
CS1
CS2
CS3
CSn
Configuration signals for
Logic Blocks, Switching Matrix, and I/O Blocks

Fig. 4. Circuit diagram of reconfiguration circuit.

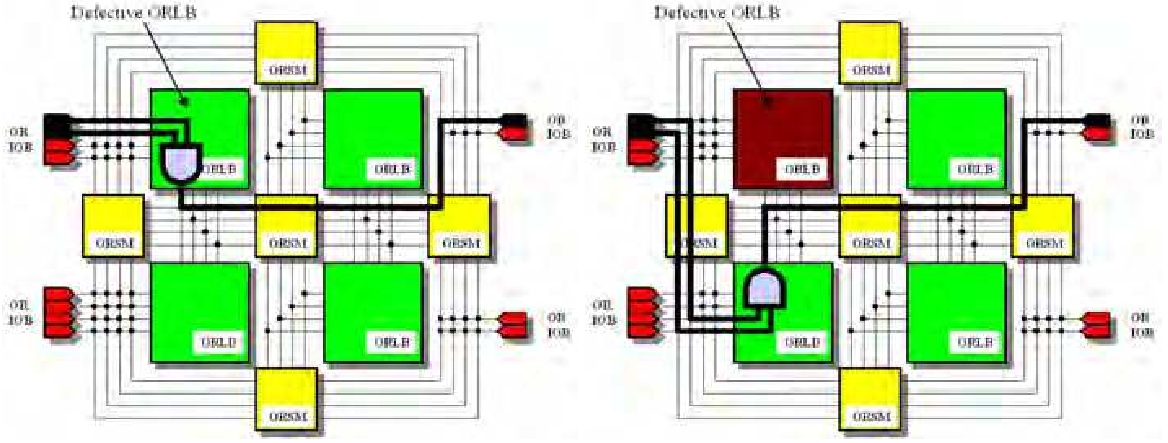


Fig. 5. Defective area avoidance method on a gate array. Here, it is assumed that a defective optically reconfigurable logic block (ORLB) exists, as portrayed in the upper area of the figure. In this case, the defective area is avoided perfectly using parallel programming with the other components, as presented in the lower area of the figure.

### 3.3 ORGA-VLSI part

In the ORGA-VLSIs, serial transfers were perfectly removed and optical reconfiguration circuits including static memory functions and photodiodes were placed near and directly connected to programming elements of a programmable gate array VLSI. Figure 4 shows that the toggle flip-flops are used for temporarily storing one context and realizing a bit-by-bit configuration. Using this architecture, the optical configuration procedure for a gate array can be executed perfectly in parallel. Thereby, the VLSI part can achieve a perfectly parallel bit-by-bit configuration.

#### 3.3.1 Simple method to avoid defective areas

Using configuration, a damaged gate array can be restored as shown in Fig. 5. The structure and function of an optically reconfigurable logic block and optically reconfigurable switching matrices on a gate array are mutually similar. If a part is defective or fails, the same function can be implemented onto the other part. Here, the upper part of Fig. 5 shows that it is assumed that a defective optically reconfigurable logic block (ORLB) exists in a gate array. In that case, the lower part of Fig. 5 shows that another implementation is available. By reconfiguring the gate array VLSI, the defective area can be avoided perfectly and its functions can be realized using other blocks. For this example, we assumed a defective area of only one optically reconfigurable logic block. For the other cells, for optically reconfigurable switching matrices, and for optically reconfigurable I/O blocks, a similar avoidance method can be adopted. Such a replacement method can be adopted onto FPGAs; however, such a replacement method is based on the condition that the configuration is possible. Regarding FPGAs, the defect or failure probability of configuration circuits is very high because of the serial configuration. On the other hand, the ORGA architecture configuration is very robust because of the parallel configuration. For that reason, the ORGA architecture has high defect and fault tolerance.

### 3.3.2 Weak point

However, a weak point exists on the ORGA-VLSI design. It is a common clock signal line. When using a single common clock signal line to distribute a clock for all delay-type flip-flops, damage to one clock tree renders all delay-type flip-flops useless. Therefore, the clock line must be programmable with many buffer trees when a large gate count VLSI or a wafer scale VLSI is made. In currently available FPGAs, each clock line of delay-type flip-flops has already been programmable with several clock trees. To reduce the probability of the clock death trouble, sufficient programmable clock trees should be prepared. If so, along with FPGA, defects for clock trees in ORGA architecture can be beyond consideration.

### 3.3.3 Critical weak points

Figure 4 shows that more critical weak points in the ORGA-VLSIs are a refresh signal, a reset signal, and a configuration CLK signal of configuration circuits to support optical configuration procedures. These signals are common signals on VLSI chip and cannot be programmable since the signals are necessary for programming itself. Therefore, along with the laser array, a physical action or a spare method is required in addition to enforcing the wire and buffer trees for defects so that critical weak points can be removed.

## 3.4 Possibility of greater than tera-gate capacity

In ORGA architecture, a holographic memory is a very robust device. For that reason, defect analysis is done only for an ORGA-VLSI and a laser array. In ORGA-VLSI part, even if defect parts are included on the ORGA-VLSI chip, almost all defect parts can be avoided using parallel programming capability. The only remaining concern is the common signals used for controlling configuration circuits. For those common signals, spare hardware or redundant hardware must be used. On the other hand, in a laser array part, only a spare row method must be applied to matrix driver circuits. The other defects are negligible.

Therefore, exploiting the defect tolerance and using methods of ORGA architecture described above, a very large die size VLSI is possible. At that time, according to an earlier paper (12), if it is assumed that an ORGA-VLSI is built on a 0.18 $\mu m$ process 8 inch wafer and that 1 million configuration contexts are stored on a corresponding holographic memory, then greater than 10-tera-gate VLSIs will be realized. Currently, although this remains only a distant objective, optoelectronic devices might present a new VLSI paradigm.

## 4. Conclusion

Optically reconfigurable gate arrays have perfectly parallel programmable capability. Even if a gate array VLSI and a laser array include defective parts, their perfectly parallel programmable capability enables perfect avoidance of defective areas. Instead, it uses the remaining area of a gate array VLSI, remaining laser resources, and remaining holographic memory resources. Therefore, the architecture enables fabrication of large-die VLSI chips and wafer-scale integrations using the latest processes, even those chips with a high defect fraction. Finally, we conclude that the architecture has a high defect tolerance. In the future, optically reconfigurable gate arrays will be a type of next-generation three-dimensional (3D) VLSI chip with an extremely high gate count and with a high manufacturing-defect tolerance.

## 5. References

[1] C. Hess, L. H. Weiland, ”Wafer level defect density distribution using checkerboard test structures,” International Conference on Microelectronic Test Structures, pp. 101–106, 1998.

[2] C. Hess, L. H. Weiland, ”Extraction of wafer-level defect density distributions to improve yield prediction,” IEEE Transactions on Semiconductor Manufacturing, Vol. 12, Issue 2, pp. 175-183, 1999.

[3] Altera Corporation, ”Altera Devices,” http://www. altera.com.

[4] Xilinx Inc., ”Xilinx Product Data Sheets,” http://www. xilinx.com.

[5] Lattice Semiconductor Corporation, ”LatticeECP and EC Family Data Sheet,” http://www. latticesemi.co.jp/products, 2005.

[6] A. J. Yu, G. G. Lemieux, ”FPGA Defect Tolerance: Impact of Granularity,” IEEE International Conference on Field-Programmable Technology,pp. 189–196, 2005.

[7] A. Doumar, H. Ito, ”Detecting, diagnosing, and tolerating faults in SRAM-based field programmable gate arrays: a survey,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 11, Issue 3, pp. 386 – 405, 2003.

[8] M. Watanabe, F. Kobayashi, ”Dynamic Optically Reconfigurable Gate Array,” Japanese Journal of Applied Physics, Vol. 45, No. 4B, pp. 3510-3515, 2006.

[9] N. Yamaguchi, M. Watanabe, ”Liquid crystal holographic configurations for ORGAs,” Applied Optics, Vol. 47, No. 28, pp. 4692-4700, 2008.

[10] D. Seto, M. Watanabe, ”A dynamic optically reconfigurable gate array - perfect emulation,” IEEE Journal of Quantum Electronics, Vol. 44, Issue 5, pp. 493-500, 2008.

[11] M. Watanabe, M. Nakajima, S. Kato, ”An inversion/non-inversion dynamic optically reconfigurable gate array VLSI,” World Scientific and Engineering Academy and Society Transactions on Circuits and Systems, Issue 1, Vol. 8, pp. 11- 20, 2009.

[12] M. Watanabe, T. Shiki, F. Kobayashi, ”Scaling prospect of optically differential reconfigurable gate array VLSIs,” Analog Integrated Circuits and Signal Processing, Vol. 60, pp. 137 - 143, 2009.

[13] M. Watanabe, F. Kobayashi, ”Manufacturing-defect tolerance analysis of optically reconfigurable gate arrays,” World Scientific and Engineering Academy and Society Transactions on Signal Processing, Issue 11, Vol. 2, pp. 1457- 1464, 2006.

[14] M. Miyano, M. Watanabe, F. Kobayashi, ”Optically Differential Reconfigurable Gate Array,” Electronics and Computers in Japan, Part II, Issue 11, vol. 90, pp. 132-139, 2007.

[15] M. Nakajima, M. Watanabe, ”A four-context optically differential reconfigurable gate array,” IEEE/OSA Journal of Lightwave Technology, Vol. 27, No. 24, 2009.

ISBN 978-953-307-057-5
Hard cover, 290 pages
**Publisher** InTech
**Published online** 01, January, 2010
**Published in print edition** January, 2010

The 14 chapters presented in this book cover a wide variety of representative works ranging from hardware design to application development. Particularly, the topics that are addressed are programmable and reconfigurable devices and systems, dependability of GPUs (General Purpose Units), network topologies, cache coherence protocols, resource allocation, scheduling algorithms, peertopeer networks, largescale network simulation, and parallel routines and algorithms. In this way, the articles included in this book constitute an excellent reference for engineers and researchers who have particular interests in each of these topics in parallel and distributed computing.

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821