IEICE TRANSACTIONS

on Electronics

VOL. E104-C NO. 6
JUNE 2021

A PUBLICATION OF THE ELECTRONICS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER Special Section on Low-Power and High-Speed Chips

Recovering Faulty Non-Volatile Flip Flops for Coarse-Grained Reconfigurable Architectures

Takeharu IKEZOE[†a)], ***Nonmember***, **Takuya KOJIMA**[†], ***Student Member***, ***and*** **Hideharu AMANO**[†], ***Fellow***

SUMMARY Recent IoT devices require extremely low standby power consumption, while a certain performance is needed during the active time, and Coarse-Grained Reconfigurable Arrays (CGRAs) have received attention because of their high energy efficiency. For further reduction of the standby energy consumption of CGRAs, the leakage power for their configuration memory must be reduced. Although the power gating is a common technique, the lost data in flip-flops and memory must be retrieved after the wake-up. Recovering everything requires numerous state transitions and considerable overhead both on its execution time and energy. To address the problem, Non-volatile Cool Mega Array (NVCMA), a CGRA providing non-volatile flip-flops (NVFFs) with spin transfer torque type non-volatile memory (NVM) technology has been developed. However, in general, non-volatile memory technologies have problems with reliability. Some NVFFs are stacked-at-0/1, and cannot store the data in a certain possibility. To improve the chip yield, we propose a mapping algorithm to avoid faulty processing elements of the CGRA caused by the erroneous configuration data. Next, we also propose a method to add an error-correcting code (ECC) mechanism to NVFFs for the configuration and constant memory. The proposed method was applied to NVCMA to evaluate the availability rate and reduction of write time. By using both methods, the average availability ratio of 94.2% was achieved, while the average availability ratio of the nine applications was 0.056% when the probability of failure of the FF was 0.01. The energy for storing data becomes about 2.3 times because of the hardware overhead of ECC but the proposed method can save 8.6% of the writing power on average.

key words: *CGRA, configuration reduction, non-volatile memory, fault-tolerance*

1. Introduction

Recent emerging Internet of Things (IoT) application requires flexibility, high domain-specific performance, and low power consumption. Reconfigurable devices including Field-Programmable Gate Array (FPGA) and CGRA have been tried to be used because of their flexibility and low Non-Recurring Engineering cost. Widely used fine-grained reconfigurable devices are extremely flexible and may perform better than the CGRA. However, their large leakage power caused by a large configuration data memory and wiring resources make them difficult to be used for applications with long standby time. Although the leakage power of CGRAs is lower than that of FPGAs, it still tends to be large because of their large array of processing elements (PEs) and memory cells to store the configuration data. Power gating reduces such leakage power by shutting off the power grid with high-threshold low leakage transistors for power switches. Some commercial CGRAs have already provided it because it is a promising solution to reduce the standby leakage power of CGRAs [1].

One of the most significant issues of the power gating is the loss of data on memory elements or flip-flops when the power switches shut down the power in the sleeping time. Recovering everything after the wake-up requires a large number of state transitions and considerable overhead both on its execution time and energy. To address this issue, several reconfigurable devices using non-volatile memory technology have been proposed [2], [3]. [2] uses spin-transfer torque magneto-resistive random access memory (STT-MRAM) to reduce power and enable instant restart even when FPGA power gating is applied.

NVCMA, a CGRA that provides spin-gate torque type non-volatile memory technology was the first trial to introduce NVFFs into the CGRA [4]. NVCMA adopts Non-Volatile Flip-Flops (NVFFs) for configuration memory elements to retain data even when power gating is applied. Power gating can be aggressively applied to suppress the leakage power in the idle state without considering the loss of the data and waking up delay. The time to store data into NVFFs can be controlled by the controller embedded in CGRA.

However, NVM technology generally has reliability issues, such as process variations in transistors and MTJ elements during manufacturing. The NVFF yield is much worse than other memory elements, and we must operate NVCMA properly even with some faulty NVFFs. Techniques with redundant rows commonly used for memory modules are hard to be applied because the structure of NVFF is completely different from the common memory cells. Triplication of the NVFFs is not efficient considering its large energy cost.

Fortunately, not all PEs and switching elements (SEs) are used in a CGRA, when its PE array is enough large for the target application. Therefore, by modifying the mapping algorithm to the PE array, the malfunctioning of PEs can be avoided. Next, we also introduce ECC to reduce malfunctions caused by configuration memory faults. In this research, we propose a mapping algorithm to avoid the bad processing elements of CGRA in order to improve the chip yield as well as introducing ECC.

NVCMA is the first CGRA embedding NVFFs for its configuration/constant memory. Compared with common memory cells, NVFFs has a problem of high error ratio re-

Manuscript received July 3, 2020.
Manuscript revised October 6, 2020.
Manuscript publicized December 14, 2020.

[†]The authors are with Dept. of Information and Computer Science, Keio University, Yokohama-shi, 223–8522 Japan.
a) E-mail: wasmii@am.ics.keio.ac.jp

DOI: 10.1587/transele.2020LHP0002

lated to the storing time and energy. As shown in Sect. 3.1, most of previous researches focus on bypassing the faulty PEs or links rather than the configuration/constant memory, because unlike the large amount of configuration memory used in FPGAs, the fault probability of such memory cells for CGRA can be enough small if the common memory cells are used. However, when non-volatile memory is introduced, fault recovery methods must be prepared and evaluated. The novelty and technical contribution of this paper is as follows:

- The influence of the faulty NVFFs is evaluated and analyzed with the practical model based on the implementation.
- Two fault recovery methods: using ECC and bypassing the faulty element by changing the mapping are proposed and the their effects are evaluated.
- The impact of the proposed method to the storing energy is evaluated.

The rest of the paper is organized as follows. Overview of the NVCMA and related work are introduced in Sect. 2 and NVCMA with a fault-tolerant non-volatile configurable memory is proposed in Sect. 3. Evaluation of fault tolerance is shown in Sect. 4. Finally, we conclude this paper in Sect. 5.

2. Background and Related Work

2.1 Non-Volatile Flip-Flops

NVCMA has a spin torque injection type NVFF for all NVM elements. The magnetic tunnel junction (MTJ) stores information by the property that the direction of magnetization can be changed by flowing a certain amount of electric current and STT-RAM uses it. At this time, to store the data safely, the current must be kept for a longer time than required. That is, for storing data, a considerable amount of power is required in total. An improved NVFF circuit [5], [6] has been proposed to reduce the excessive energy for storing data. Before storing data, the NVFF compares the data currently stored with the data to be written. If they are the same, the storage process is skipped, and the stored data are used.

The next improvement is for saving a certain time margin to store the data safely. The time required for storing data depends on each chip with the process variation. Even for each NVFF in the same chip, there is a certain inner-chip variation to cause the difference. For the above mentioned first improvement, a comparator is provided to skip the unnecessary data store. The VR-NVFF shown in Fig. 1 was proposed to use the comparator also for data verification [6]. It consists of a common master latch and a slave latch with two MTJs (MTJ0 and MTJ1). For verification, a small balloon is provided to compare the reading data from the MTJ part with the current writing data. When the data to be written is different from the currently stored data, the writing process starts. After a trial of writing data in a short

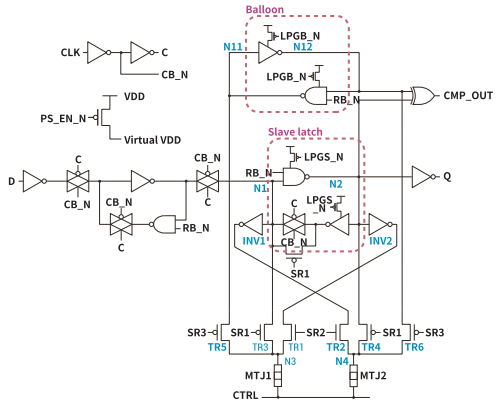


Fig. 1 VR-NVFF

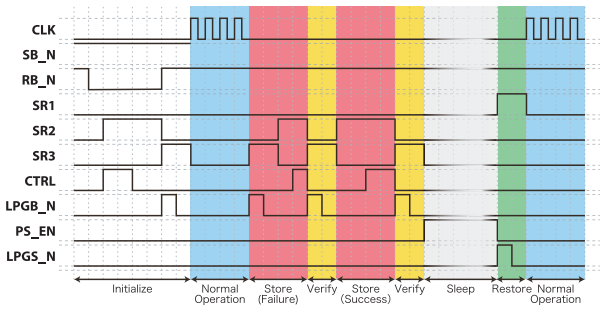


Fig. 2 Timing chart of VR-NVFF

time, the data are read out and compared with the data to be written again. If they are the same, it finishes the writing; otherwise, the next trial is done until the correct data are stored. This adaptive control of storing time can save energy as well as improving reliability.

However, as shown in Fig. 2, the VR-NVFF requires numerous state transitions for verification, storing and restoring the data. Seven control signals are used to manage NVFF (SR1, SR2, SR3, SB_N, RB_N, LPGB_N, and LPGS_N), one is used for power gating (PS_EN), and the other is for clock gating. The comparing result (CMP_OUT) must be checked, and if it is not correct (i.e., CMP_OUT = Hi), the writing must be continuously executed. Although it can be done with a dedicated hardwired logic, such logic often lacks flexibility, and only a fixed flow can be applied. Since the NVFF is used with the power gating mechanism, providing a unified power manager that controls the power shut-down, wake up, data store, restore and verification is advantageous. It means that NVFFs should be used with a programmable controller to manage to store/restore data from/to NVFFs, shut-down the power, and wake-up.

2.2 Overview of NVCMA

2.2.1 Cool Mega Array (CMA)

CMA [7] is a type of straight forward CGRAs for mapping a straight forward dataflow on the PE array as shown in Fig. 3. Piperench [8], Kilo-core [9], S5 engine [10], and EGRA [11]

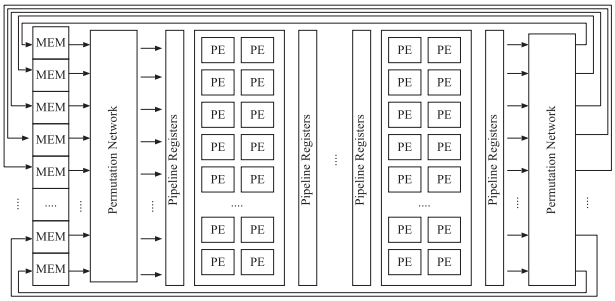


Fig. 3 Straight forward CGRAs

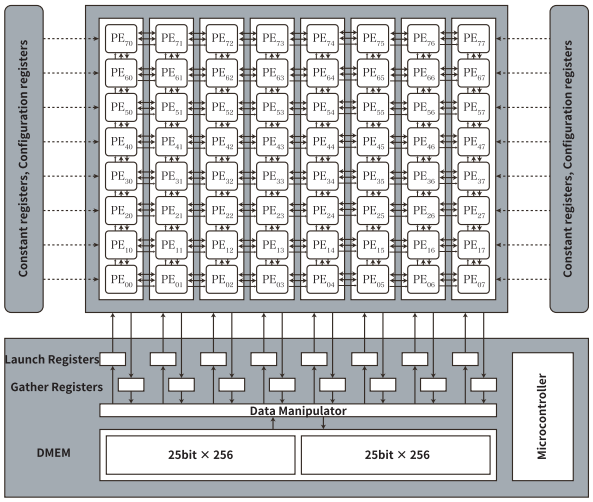


Fig. 4 Block diagram of NVCMA

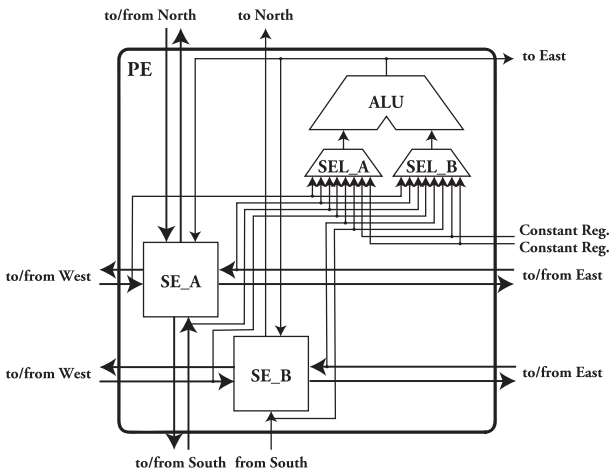


Fig. 5 PE of NVCMA

belong to straight forward CGRAs. Since PEs do not have register files to hold intermediate results, the PE array of the CMA forms large combinational circuits. A clock signal for the PE array is not required, and so dynamic power consumption is saved. Although long critical path delay is a concern, PE arrays are designed to be multi-cycle paths, and the system clock frequency is not degraded. The number of execution cycles is programmable depending on a mapped application. First, the data to be computed are read out from the interleaved dual-port data memory modules, passing through a permutation network and stored in the fetch registers provided at the input of the large PE array on which the application dataflow is directly mapped. After a certain delay time, the results are stored into the gather register, and then transferred into the interleaved dual-port memory through another permutation network. The whole system is controlled by a tiny microcontroller that provides a RISC style simple ISA.

2.2.2 Non-Volatile CMA (NVCMA)

CMA is suitable for embedding NVFFs because of its simple memory structure. Providing configuration registers and some other registers with NVFFs allows quick sleep-down and wake-up. CMA also provides a microcontroller used for the management of the power and storing/restoring NVFFs. Various types of CMA and its extensions have been developed [12], [13]. Some of them use body bias control [12], and others are embedded into a 3D chip stack by using wireless inductive coupled through chip interface [14].

We picked up CMA-SOTB [12] as the basis of NVCMA [4]. Like other CMAs, NVCMA has a large PE array and a tiny microcontroller to manage data transfer between the data memory and PE array (Fig. 4). As shown in the figure, the size of PE array is 8×8. The PE array is connected to the network using a two-channel island-style interconnect (Fig. 5). The SEs transfer input data from south, west, and east of the PE and forwards ALU output data to the PEs in the appropriate direction according to the configuration data. NVCMA only provides registers at input and output of the PE array. Since the leakage current in the PE array can be suppressed with the power gating, it has the possibility to achieve extremely low leakage consumption by introducing NVFFs. We introduced VR-NVFFs for the most memory elements of NVCMA: configuration registers, constant registers, data memory, and microcode memory of the microcontroller.

NVCMA has an external bus and chip interface for connecting to a host CPU or other accelerators. The external bus consists of a 12-bit address bus and a 32-bit data bus. Each module is mapped to the same address space, so the host CPU can access data in the NVCMA memory through the interface and external bus. The PE configuration data consists of 10-bit ALU parts (OPCODE, SEL_A, and SEL_B), 12-bit SE_A parts (NORTH, SOUTH, EAST, and WEST) and 9-bit SE_B parts NORTH, EAST, and WEST).

First of all, memory elements of NVCMA were divided into 10 power domains, each of which has about 2,000 NVFFs. Each power domain can independently execute store, restore, sleep, and wake-up operation. Three of 10 power domains are used for the configuration data, five for data memory, and the remaining two are for the instruction memory of the microcontroller. Verification is successful when all FFs in a domain have been verified. It allows man-

agement of power required for each power domain, and also, a rush current to write all NVFFs in a chip at the same time can be avoided.

2.3 Power Management

Considering the IoT usage, NVCMA sleeps down for the idle time after storing the required data into NVFFs. Here, only memory domains needed are saved. For example, since the configuration data and instruction memory data are not changed while execution a single task, they are needed to be saved only once. Both the self-wake-up mode and the requested wake-up mode are provided. For the self-wake up after a certain time interval, a part of the microcontroller must be an ever-on block that does not use the power gating mechanism. A wake-up request signal from outside the chip also can trigger the wake-up.

NVCMA also supports the quick wake-up from unexpected power down. In this case, the program can start just after the last time when the data were saved into NVFFs. For such situations, the programmer must save the data in a certain interval like common checkpoint methods. However, compared with saving data outside the chip, the checkpoint using NVFFs is much easier. Of course, an unexpected power down is possible during data saving into NVFFs, and the instruction memory might be destroyed. For such a case, NVCMA has a hardwired read-only memory area to keep instructions for fundamental power control.

The microcontroller of NVCMA transfers the data from the data memory to the fetch register in front of the PE array, and after a certain delay time, it writes back the results to the data memory. The data memory is divided into eight banks, and eight words are read out and written back simultaneously. Between the data memory and PE array, a data-manipulator is provided to transfer data to the appropriate position of the PE array inputs/outputs. Two data memory modules form the double buffer to overlap the computation and data input/output. The microcontroller is a tiny microprocessor that manages all data transfer to perform an application. 16-bit micro-instruction consists of 4bit op-code and 10bit operand. Simple numerical operations and branches are provided as well as special instructions for data transfer. Sixteen 16-bit general-purpose registers are used mainly for pointers for data memory. The detail of the microcode of NVCMA is shown in [15].

The power management including the verification control of NVFFs should be included in the application context since the write time and domains to be saved into NVFFs depend on the application execution. So, we propose a power manager embedded into the microcontroller of NVCMA. As shown in Fig. 1, NVFF has 9 signals to store, verify, power down, and restore. That is, CLK_EN for clock gating control signal, ERR_SET for storing CMP_OUT from each NVFF, and eight bits for input signals are shown in Fig. 1. In total, the 10-bit control register is required for each domain, and so 10 control registers are provided as shown in Fig. 6. Note that, except ERR_SET, each bit of a control register

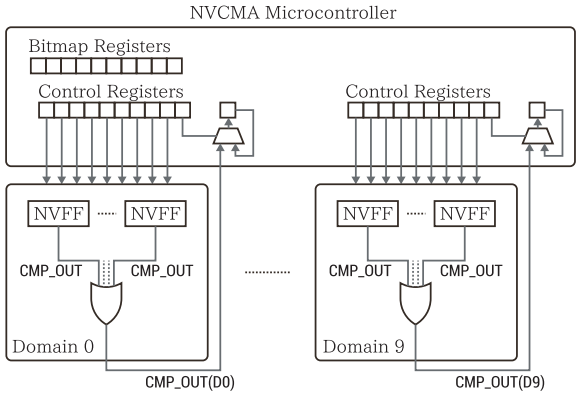


Fig. 6 NVCMA microcontroller

is directly connected to control signals of all NVFFs in the corresponding power domain. The verification results are on the signal CMP_OUT from each NVFF. When ERR_SET is '1', the OR of CMP_OUTs of all NVFFs in that domain is taken, and its value is stored in the corresponding bit of the error result register.

2.4 Storing Data

Since NVCMA uses VR-NVFF, it has parameters "short store time" and "long store time" as the time for writing data. Here, the "long store time" should be set to "time for which store can be reliably performed." [4] uses 18ns as the long store time, assuming that the average NVFF store time is 10ns. The standard deviation σ is 5ns in [4]. It has not been fully considered whether NVCMA operates normally at this store time.

3. Proposed Architecture

3.1 Reliability Problems of NVFFs

NVCMA aims to control the time to write the data into NVFF to reduce energy consumption. However, the spin-transfer torque type NVFF has still much faulty compared with a common CMOS memory cell. It means that some FFs cannot write the data even with long storing time. Thus, especially at the stage of process development, the yield of NVFF is often too small to obtain a chip whose embedded NVFFs are all available.

In several researches on fault tolerant CGRA, [16], [17] and [18] try to bypass the faulty elements by locating faults and bypassing them by changing the mapping. The target of [16] is a CGRA overlaid on an FPGA, and it focuses on saving the soft-error occurs in the configuration memory of the FPGA. Since the CGRA is built on an FPGA, the method for bypassing fault is different. [17] focuses on the faulty elements on PEs and interconnction network. Although the mapping algorithm is proposed, it assumes an ideal configuration memory. Also, the target CGRA allows context switching, thus the flexibility is larger than our case. [18] assumes fault on the configuration memory, but it regards

such faults as a fault of PE. Thus, with this method, faulty configuration memory for interconnect or constant data are not treated. It is also the method for CGRAs with context switching mechanism.

The method proposed here is for faulty configuration memory built with NVFFs whose fault possibility is much higher than the common digital elements. Since the NVCMA is the first trial to embedded NVFFs into CGRA, there are no previous work based on the same assumption.

This paper is based on the conference paper [19] in our earlier stage of the research. The comprehensive evaluation and discussion including resource usage and parallel applications are new contribution as a journal paper.

3.2 Modeling of NVFFs

A hard error and a soft error are caused by the failure in writing data into the NVFF. A hard error is caused by the problem on the process, and the value is eternally stacked-at 0 or 1. Data cannot be written even with any long writing time and any high writing voltage. Soft errors occur when the voltage or retention time required to write for the NVFF is larger than others. The data are sometimes written but unstable because of the chip variation and temperature. Not that, writing disturbances are not needed to care since the NVFF used in NVCMA [6] is not written when the current value is the same as the writing value. As the reading current flows in the writing direction, reading disturbances are not needed to be cared either. In such a case, both a hard error and a soft error show the same phenomenon that remains 1 when 0 is written, or 0 when 1 is written.

Several models of writing time to STT-MRAM have been proposed such as normal distribution [20], skew-normal distribution [20], and gamma distribution [21]. In this study, we use a model of normal distribution. For example, if you estimate enough voltage and write time of 3σ for NVFF without any hard error, 0.27% of FFs are modeled as stacked-at-0 or stacked-at-1 FFs, and the other 99.73% can be assumed as normal. Note that 0.27% of FFs can be written if the storing time is longer.

3.3 Fault Tolerant Mapping

In our method, a host processor examines which FFs operate correctly, and which ones are stacked-at-0 or 1 on the target chip. Here, it is called "failure map." The failure map depends on writing voltage and writing time. To obtain it, the host processor writes 0 and 1 to the configuration memory (conf.) and constant memory (const.). Then, it reads written values from the memories and checks them. In this way, for the given voltage and writing time, two failure maps (i.e., stacked-at-0 map and stacked-1-map) are obtained.

The NVCMA can set the writing time to the MTJs by the program of the microcontroller. Besides, we assume the programmable writing voltage for the NVFF. Thus, we have to find appropriate writing time and writing voltage depending on a target application. If the PE array is large enough

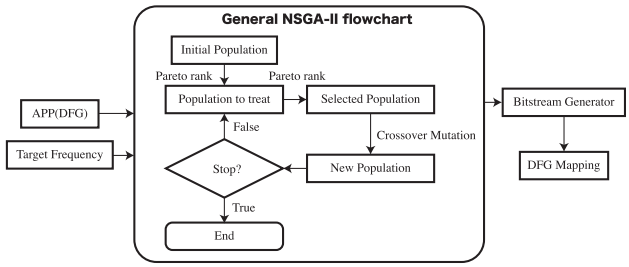


Fig. 7 An overview of mapping flow

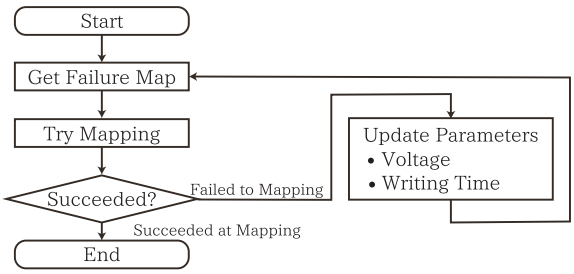


Fig. 8 Flow chart of configuration

for the target application, it is possible to find the application mapping which avoids using PEs associated with faulty FFs. In contrast, if the target application needs many PEs, writing voltage and/or writing time have to be increased to reduce faulty FFs and increase available PEs.

A genetic-algorithm-based application mapper for CMA architectures has been proposed in [22]. It can optimize mapping quality with multiple criteria (e.g., power consumption and throughput, etc.) by using Non-dominated Sorting Genetic Algorithms-II (NSGA-II). The general flow of NSGA-II for application mapping is illustrated in Fig. 7. In this method, the mapped application data-flow-graph (DFG) is represented as an acyclic directed graph. In the DFG, a node is denoted as an operation (e.g., addition and multiplication) and an edge between nodes indicates data dependency between them. Besides, a target frequency is specified to guarantee that the critical path of a generated mapping does not violate the timing constraint. Placement of the operations on PEs, that is, which PEs are utilized for the operations, is coded as a chromosome. Therefore, if the chromosome is different, so are the utilized PEs. For each chromosome, routing between PEs is carried out by using A-star routing algorithm. A population is a set of chromosomes that are candidates of optimal solutions. During the evolution of the population, the NSGA-II tries to improve the solutions. Finally, it generates multiple solutions (if exists) since it provides trade-off possibilities between multiple objectives.

Although the original mapping method brings good mappability and well-optimized mappings, it cannot account for the effects of faulty FFs. To find the application mapping considering it, we propose an extended mapping flow shown in Fig. 8. Assuming a specific writing voltage and write time, it tries to find a valid application mapping

by using the same NSGA-II. A new objective function described as bellow is added.

$$\begin{aligned}\min N_{\text{fault}} = & \sum_{i,j \in UsedPEs} isPE_{\text{fault}}(i, j, Op(i, j)) \\ & + \sum_{i,j \in UsedSEs} isSE_{\text{fault}}(i, j, R(i, j))) \\ & + \sum_{c \in UsedConstRegs} isConst_{\text{fault}}(c, val(c)) \end{aligned} \quad (1)$$

where

- N_{fault} is the number of unavailable resources due to the faulty FFs,
- $Op(i, j)$ is mapped operation for the PE_{ij},
- $isPE_{\text{fault}}(i, j, Op(i, j))$ is equal to 1 if the PE_{ij} cannot work for $Op(i, j)$; otherwise 0,
- $R(i, j)$ is routing configuration of SE_{ij},
- $isSE_{\text{fault}}(i, j, R(i, j))$ is equal to 1 if the SE_{ij} cannot work for $R(i, j)$; otherwise 0,
- $val(c)$ is constant value for the constant register c, and
- $isConst_{\text{fault}}(c, val(c))$ is equal to 1 if the constant register c cannot provide $val(c)$; otherwise 0.

They are calculated by the obtained failure maps. If the writing of the same value as the value which the NVFF stacked at is attempted, it is considered as successful write. For example, a zero write to a stacked-at-0 NVFF is considered as successful. Although the NSGA-II tries to minimize the N_{fault}, if the smallest value in the final population is not 0, it means to fail to generate valid mappings.

To find optimal writing voltage and writing time, our method iterates the extended mapping algorithm as shown in Fig. 8. It starts at a certain writing voltage and writing time. Then, it iteratively tries to find valid mappings while increasing the writing voltage and extending the writing time step by step. When it finds at least one valid mapping for the application, it stops the iteration and generates the configuration for the availability of the application. Note that Fig. 8 shows our final goal. For achieving it, we first analyze the availability with a certain possibility of faults, and then the energy reduction by changing the storing time is discussed.

NVCMA controls the write time at the granularity of the operating frequency. Therefore, it is not realistic to change the minimum write time unless power is prioritized. Considering the target frequency of NVCMA, we are planning to extend the writing time gradually. First, in the case where the target frequency of the NVCMA is 80 MHz, the write time to the MTJ of the NVFF is set to 12.5 ns, then after acquiring failure maps, mapping is performed. If this mapping fails, the mapping would be tried in 25ns. For example, if the writing time is 25 ns in consideration that only soft errors at the time of writing occur, it means that writing has been successfully performed to NVFF of $17/3\sigma$. Here, the standard deviation of σ is 3ns, and the average writing time is 8ns. This indicates that the rate of bad NVFF is less than 10^{-8} in the normal distribution. The NVCMA does not make the operating voltage programmable. Since the operating voltage is not programmable in NVCMA, the supply voltage is constant when operating NVCMA. Changing voltage is limited to the case that the mapping is retried with the normal voltage if the mapping fails for a long time at a low voltage.

3.4 An ECC Mechanism for the Configuration Data

Here, to cope with the problem of Sect. 3.1, we also propose an NVCMA with an ECC mechanism for the configuration data. Fortunately, the NVCMA provides a comparison mechanism for reducing the time of storing data. Although the output CMP_OUT from each NVFF is logically or-ed to generate the signal whether all NVFFs are correctly written or not [4] in the original NVCMA, it can be used for correction with ECC code. Here, we propose to provide ECC modules between each PE and NVFFs storing its configuration data, using a cyclic code, which is a popularly used error correction code, to correct the storing data. We assume to modify it for each word, and $\text{CMP\_OUT(conf.)}_{ij}$ represents CMP_OUT of the configuration memory of the PE / SE in the i-th row j-th column. The cyclic code used here requires a relatively small circuit size as well as computational cost. Thus, the delay for writing to the configuration NVFFs is not so stretched. As an example, in this research, a cyclic code that expands 12 bits to 23 bits and corrects 3 bits are used. This applies to the PE ALU, one SE of two in the PE, 5 bits above constant memory and 5 bits below constant memory. The surplus bits are zero-filled.

For programming the PE array, PE configuration registers, SE configuration registers, and constant registers are provided in NVCMA. Each register is connected to each element directly. Both the method described in Sect. 3.3 and the introduction of ECC can be used together. Evaluation is performed using NVCMA with ECC and NVCMA without ECC.

4. Evaluation

4.1 Target Applications

Five application programs (af, gray, sepia, sf, dft4) used here for CMA evaluation are shown in Table 1. The resources they use in the array of NVCMA are shown in Table 2. We evaluated the writing energy from the data of the logic level simulation (NC-Verilog) based on the NVCMA, and SPICE simulation results. NVCMA and VR-NVFF circuits are developed with a 65nm process technology and the same process is used for the SPICE simulations. Here, the supply voltage VDD is set to be 1.2V. As described in Sect. 3.3, the short storing time is 12.5ns and long storing time is 25ns.

In order to investigate the effect of the usage of resources, we use applications *alpha2, alpha3*, and *alpha4*, which are parallel applications of simple application *alpha*.

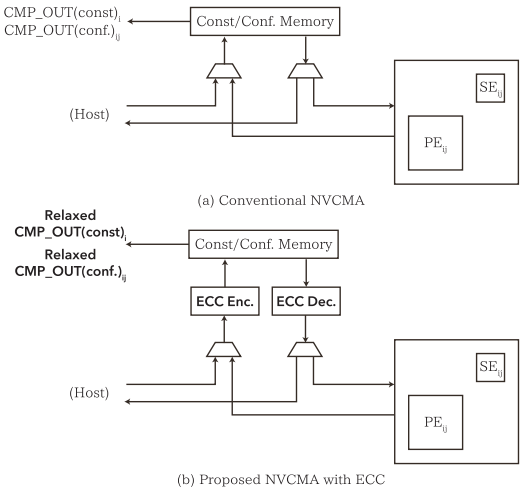


Fig. 9 ECC mechanism for SE / PE

Table 1 Simulated applications

Name	Contents
af	24-bit (RGB) Alpha blender
sepia	24-bit (RGB) Sepia Filter
gray	24-bit (RGB) Grayscale Filter
sf	8-bit Sepia Filter
dft4	4-point Discrete Fourier Transform
alpha	Grayscale Alpha Blender
alpha2	Double Parallel Grayscale Alpha Blender
alpha3	Triple Parallel Grayscale Alpha Blender
alpha4	Quadruple Parallel Grayscale Alpha Blender

Table 2 Resources of applications

Name	Input	Output	Nodes in DFG	Constant Values
af	2	1	24	6
sepia	3	3	12	8
gray	1	1	13	5
sf	1	1	20	11
dft4	4	5	11	2
alpha	2	1	4	3
alpha2	4	2	8	3
alpha3	6	3	12	3
alpha4	8	4	16	3

4.2 Availability Ratio

Figure 10 shows the availability ratio indicating the probability that the application is available when a chip provides a certain ratio of the faulty NVFFs. Here, we assume a number of faulty NVFFs, and evaluate the availability ratio. Here, "Conv." shows the baseline case that no fault recovery method is applied. "Mapping" shows the case where the application mapping is changed for the faulty NVFFs. "ECC" shows the case where the error correction mechanism is provided, yet the mapping is not cared. "Mapping+ECC" indicates the case where the application mapping is changed considering the ECC error correction.

As described in Sect. 3, the model of the NVFF at the time of data writing can be represented by the ratio

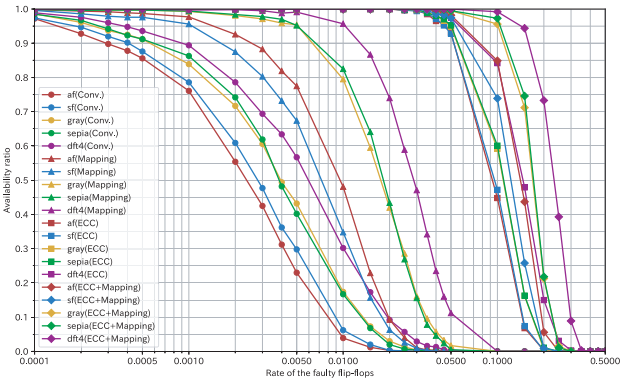


Fig. 10 The availability ratio for the rate of the faulty NVDFF

of stacked-at-0/1 NVFFs number to the fault-free NVFFs number. Whether stacked-at-0 or stacked-at-1 is depending on the manufacturing process and the design of the NVFF. Here, it is assumed to occur at the same possibility (1:1).

We prepared one thousand seed values for generating a model of a faulty NVFFs to evaluate the availability ratio under each condition. That is, one thousand different failure maps are prepared and availability ratio for each method is calculated by the average of one thousand trials. From Fig. 10, when the ratio of faulty FFs is 0.1%, the availability ratio of af is 76.1% in the conventional method in the application af but is improved to 97.7% by the "Mapping" method. This evaluation shows that it is not always necessary to use ECC to increase the total available ratio if the failure rate of the FFs is low to a certain degree. The available ratio of af is 84.9% with "ECC+Mapping", while it is 0.0% with "Mapping" when the failure rate of the FFs is 10%. However, if the ratio of stacked-at-0/1 NVFFs is less than 10%, using non-ECC NVCMA is not realistic. Under the condition that a hard error occurs at 10%, it indicates that the PE array cannot perform the required applications with most seeds in the non-ECC NVCMA.

4.3 Writing Time/Storing Energy

We evaluated the writing energy for each application shown in Table 1. As described before, the NVFF we used here compares the value being written with the value already written, and does not write if the value is the same. In the case of other NVM technologies without such a comparison mechanism, the writing time almost directly influences the writing energy. Here, we assumed the adaptive control of writing proposed in [4]. In [4], first, "short store" is tried to write most of the NVFFs, and "long store" is subsequentially tried to write NVFFs which could not write the data by "short store". Since writing to the successfully written NVFF is not written again, power consumption due to long writing time can be suppressed. Although this sophisticated writing time control works to degrade the effect of introducing ECC and changing the mapping, a certain effect is still existing.

The probability distribution considered here is shown

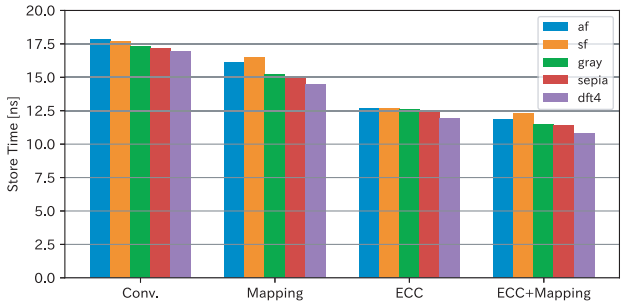


Fig. 11 Write time of "long store" to be required to achieve the availability ratio 85%

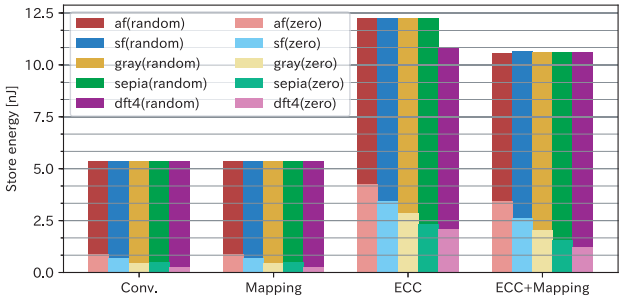


Fig. 12 Store energy at the availability ratio 85% (@80MHz)

as Eq. (2) (normal distribution). Here, t is the writing time, μ is its average time that writes data successfully, and σ is its standard deviation. As our target design, we evaluated the case with $\mu = 6$ns and $\sigma = 3$ns. At this time, the energy W_{minstore} required for the writing into the MTJ can be expressed by Eq. (3). Here, P_{storeall} is the whole power for the writing, t_{short} is the write time at short store time, and t_{long} is the write time at the long store.

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \tag{2}$$

$$W_{\text{minstore}} = 2P_{\text{storeall}} \min\Big\{t_{\text{short}} + t_{\text{long}}(1 - \int_0^{t_{\text{short}}} f(t)dt), \quad 0 < t_{\text{short}} < t_{\text{long}}\Big\} \tag{3}$$

Figure 11 shows the storing time to achieve an availability ratio of 85% which is relaxed so that it can be satisfied even with "Conv.". It is obtained as follows. First, from Fig. 10, the ratio of faulty NVFFs for the target availability ratio is obtained. Since the exact value to achieve the target availability ratio was not always in the graph, we used an interpolated value on the semi-log graph. Then, the storing time is obtained with Eq. (2) assuming a normal distribution.

From the viewpoint of achieving the same availability ratio, as expected, the write time is long in the order of the conventional method, only Mapping, only ECC, and Mapping + ECC. As the NVCMA was designed to work at 80MHz, in practice, the writing time can be controlled at the unit of 12.5 ns. In other words, t_{short} and t_{long} of Eq. (3) are multiples of 12.5 ns. Here, t_{short} is set to 12.5ns and t_{long} is set to 25ns. Figure 12 shows the result of evaluating the writing energy with the granularity of the writing time being 12.5 ns.

The number of bit to be written in configuration data $N_{\text{NVFFs to be written}}$ is shown in Eq. (4).

$$N_{\text{NVFFs to be written}} = \begin{cases} N_0(p_1+p_{s1}) + N_1(p_0+p_{s0}) + N_d(p_1+p_{s1}) \\ \qquad \text{(if failure map is not given)} \\ N_0(p_1+p_{s1}) + N_1(p_0+p_{s0}) + N_d(p_1) \\ \qquad \text{(if failure map is given)} \end{cases} \tag{4}$$

Table 3 Configuration bits of applications

application	N_0	N_1	N_d	Sum
af (Non-ECC)	435	185	1636	2256
sf (Non-ECC)	384	142	1730	2256
gray (Non-ECC)	267	94	1895	2256
sepia (Non-ECC)	265	101	1890	2256
dft4 (Non-ECC)	193	55	2008	2256
alpha (Non-ECC)	82	27	2147	2256
alpha2 (Non-ECC)	135	49	2072	2256
alpha3 (Non-ECC)	175	67	2014	2256
alpha4 (Non-ECC)	234	104	1918	2256
af (ECC)	1254	793	3105	5152
sf (ECC)	1004	606	3542	5152
gray (ECC)	802	463	3887	5152
sepia (ECC)	689	346	4117	5152
dft4 (ECC)	509	273	4370	5152
alpha (ECC)	221	101	4830	5152
alpha2 (ECC)	354	198	4600	5152
alpha3 (ECC)	487	299	4370	5152
alpha4 (ECC)	674	475	4002	5152

where

- N_0 is the number of bits 0 in the application configuration data,
- N_1 is the number of bits 1 in the application configuration data,
- N_d is the number of "don't care" bits in the application configuration data,
- p_0 is the ratio of the fault-free NVFFs in which 0 is written,
- p_1 is the ratio of the fault-free NVFFs in which 1 is written,
- p_{s0} is the ratio of stacked-at-0 NVFFs,
- p_{s1} is the ratio of stacked-at-1 NVFFs,
- e is the ratio of the fault NVFFs

p_{s0} and p_{s1} can be ignored if e is small. When NVFFs are initially 0 or 1 at the same rate, p_0 and p_1 are $(1-e)/2$, and p_{s0} and p_{s1} are $e/2$. When NVFFs are initialized to 0, p_0 is $1-e$, p_1 is 0, and p_{s0} and p_{s1} are $e/2$. Table 3 shows the number of 0, 1 and "don't care" bits in a successfully-mapped configuration data in "Conv." We use the data of Table 3 to evaluate the writing energy.

Introducing ECC increased the writing energy more than double of "Conv." because of its hardware overhead. In this case, 2.28 times the number of NVFFs compared to non-ECC NVCMA is required. However, if shorter write time is

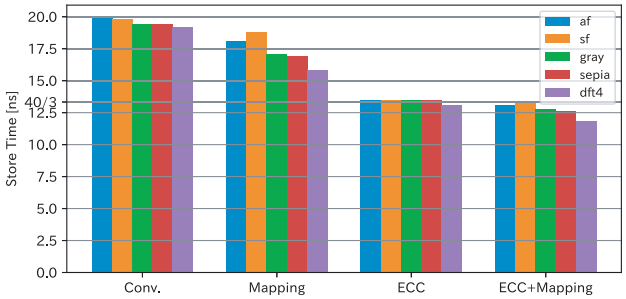


Fig. 13 Write time of "long store" to be required to achieve the availability ratio 99%

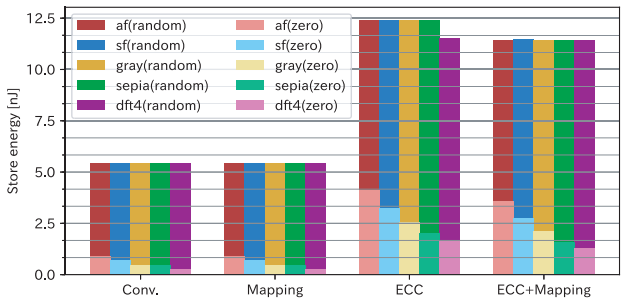


Fig. 14 Store energy at the availability ratio 99% (@75MHz)

achieved, the reduction of write energy can be achievable. Figure 12 shows that the store energy can be reduced by using the mapping even with the ECC. Here, "random" and "zero" in Fig. 12 represent the initial state of NVFF before writing the configuration data. "random" is an initial state where NVFFs are 0 or 1 at the same rate, and "zero" is an initial state where NVFFs are initialized to 0. The store energy with "ECC" and "random" is 2.23 times compared to "Conv." on average. In the case with "ECC" and "zero", the store energy is 5.75 times compared to "Conv." on average. This reason is that the ratio of N_1 of the sum is with "ECC" is higher than "Non-ECC." Using "ECC+Mapping," it can be reduced about 11.1% of the store energy compared to "ECC" with "random". When using "ECC+Mapping" with "zero", it can be reduced about 29.4% of the store energy compared to "ECC". The writing time of dft4 with both "ECC" and "ECC+Mapping" is smaller than 12.5ns so that it does not contribute to the reduction in writing energy.

When the operating frequency is 75 MHz, the power reduction by mapping with an operation rate of 99% is expected. Figure 13 shows the write time of "long store" at the availability ratio 99 % and the write granularity at 75MHz 40/3ns. If it is greater than 40/3ns, the "long store" is executed as 80/3ns, and if it is less than 40/3ns, only the 80/3ns "short store" is executed. Figure 14 shows the store energy at the availability ratio 99 % when the operating frequency is set to 75MHz. The store energy with "ECC" and "random" is 2.25 times as "Conv." on average. In the case with "ECC" and "zero", the store energy is 5.12 times as "Conv." on average. Besides, it can be reduced about 6.5% of the store energy using "ECC+Mapping" compared to "ECC"

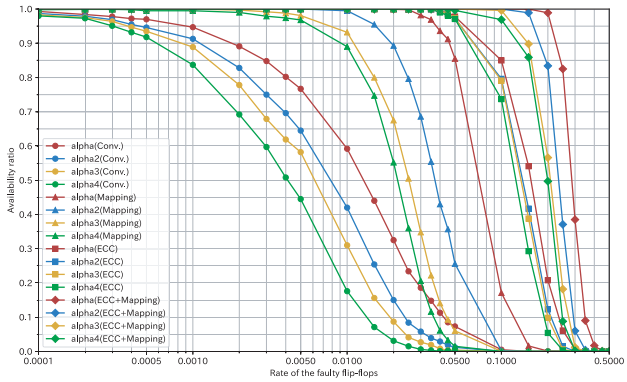


Fig. 15 The availability ratio of *alpha* applications for the rate of the faulty NVDFF

with "random". It can be reduced about 18.3% of the store energy when using "ECC+Mapping" with "zero" compared to "ECC".

4.4 Changes in Operating Rate and Writing Energy Due to Resource Usage

We evaluated the available ratio of *alpha, alpha2, alpha3*, and *alpha4* shown in Table 1 in the same way as other applications. *alpha2, alpha3*, and *alpha4* are double, triple, and quadruple parallel applications of *alpha*. The resources of the *alpha* applications in each NVCMA PE array are shown in Table 2. The resource usage of the applications increases as the degree of parallelism increases. We assumed the possibility of occurrence of 0/1-stacked NVDFF is the same.

We prepared 1,000 seeds to generate a faulty NVFF model and evaluated the available ratio under each condition in *alpha* applications. The result of the evaluation is shown in Fig. 15. The available ratio decreases in the order of *alpha, alpha2, alpha3*, and *alpha4* depending on the resource usage of the applications.

This result indicates that if the resources used by the applications are low, it may not be necessary to use ECC. For example, when the parallel degree of *alpha* application is 1 and the ratio of faulty NVFF is 0.01, the available ratio achieves 59.2% only by "Mapping", while the available ratio achieves 59.2% in the conventional method.

We obtain the write time of the "long store" required to achieve 85% of the available ratio by linearly approximating the result of Fig. 15. Figure 16 shows the result and the write granularity 40/3ns when the operating frequency is set to 75MHz. The results of the Table 3 to evaluate the energy required for writing. We calculated the store energy required to achieve the available ratio of 75% for the *alpha* applications when the operating frequency is 75MHz. Figure 17 shows the store energy of *alpha* applications.

Since the "long store" write time required to achieve an operation rate of 85 % is smaller than 40/3ns in the "Mapping" method, the "Mapping" method reduces the store energy. In the case of "random", store energy reduced by 7.2% and that of 12.8% reduced in the case of "zero".

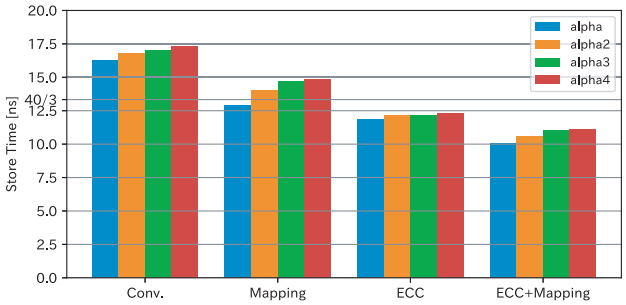


Fig. 16 Write time of "long store" of application *alpha* to be required to achieve the availability ratio 85%

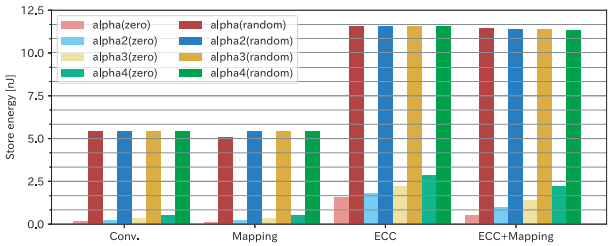


Fig. 17 Store energy of application *alpha* at the availability ratio 85% (@75MHz)

5. Conclusion

We proposed methods to keep the availability of CGRA with faulty NVFFs in its configuration and constant memory. As the target of CGRA, we selected NVCMA, a CGRA with a simple memory structure. First, we proposed the failure model of NVFFs used in NVCMA and a method to obtain the failure state of NVFF in the configuration memory and constant memory of NVCMA. In addition, we proposed a mapping method to avoid un-available PEs or SEs with the faulty NVFFs by using an application mapping tool based on a genetic algorithm. Finally, we proposed a method to add ECC mechanism to the configuration memory and the constant memory.

By using both methods with 9 applications, the 94.2% availability ratio is achieved on average when the failure rate of the FFs is 1.0%, while 0.056% chips are available without using them. "ECC+Mapping" reduced the energy for storing data by about 8.6% compared to "ECC" except for alpha, while it becomes about 2.3 times because of the hardware overhead of ECC assuming an initial state where NVFFs are 0 or 1 at the same rate at 80MHz. The proposed mapping method can be applied to CGRA other than NVCMA.

Here, we did not care about the supply voltage. Changing supply voltage and trying mapping is another possibility to optimize the energy consumption. Although the discussion here is independent of the process technology, the NVCMA with 65nm CMOS technology now straggling on the problem of faulty NVFFs. We will report it based on the real chip evaluation results when the development process of NVFFs will be stable.

Acknowledgements

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

References

[1] C. Kim, M. Chung, Y. Cho, M. Konjnenbug, S. Ryu, and J. Kim, "ULP-SRP: Ultra low power Samsung Reconfigurable Processor for biomedical applications," 2012 International Conference on Field-Programmable Technology, pp.329–334, Dec. 2012.

[2] M. Natsui, D. Suzuki, A. Tamakoshi, T. Watanabe, H. Honjo, H. Koike, T. Nasuno, Y. Ma, T. Tanigawa, Y. Noguchi, M. Yasuhira, H. Sato, S. Ikeda, H. Ohno, T. Endoh, and T. Hanyu, "12.1 An FPGA-Accelerated Fully Nonvolatile Microcontroller Unit for Sensor-Node Applications in 40nm CMOS/MTJ-Hybrid Technology Achieving 47.14μW Operation at 200MHz," 2019 IEEE International Solid- State Circuits Conference - (ISSCC), pp.202–204, Feb. 2019.

[3] L. Xie, H.A. Du Nguyen, M. Taouil, S. Hamdioui, K. Bertels, and M. Alfailakawi, "Non-volatile look-up table based FPGA implementations," 2016 11th International Design Test Symposium (IDT), pp.165–170, Dec. 2016.

[4] T. Ikezoe, H. Amano, J. Akaike, K. Usami, M. Kudo, K. Hiraga, Y. Shuto, and K. Yagami, "A coarse grained-reconfigurable accelerator with energy efficient mtj-based non-volatile flip-flops," 2018 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2018, Cancun, Mexico, pp.1–6, Dec. 2018.

[5] S. Yamamoto, Y. Shuto, and S. Sugahara, "Nonvolatile flip-flop using pseudo-spin-transistor architecture and its power-gating applications," 2012 International Semiconductor Conference Dresden-Grenoble (ISCDG), pp.17–20, Sept. 2012.

[6] J. Akaike and K. Usami, "MTJ-based nonvolatile flip-flop circuit enabling to verify stored data," IEICE technical report, vol.116, no.417, pp.175–180, Jan. 2017.

[7] N. Ozaki, Y. Yasuda, M. Izawa, Y. Saito, D. Ikebuchi, H. Amano, H. Nakamura, K. Usami, M. Namiki, and M. Kondo, "Cool mega-arrays: Ultralow-power reconfigurable accelerator chips," IEEE Micro, vol.31, no.6, pp.6–18, Nov. 2011.

[8] H. Schmit, D. Whelihan, A. Tsai, M. Moe, B. Levine, and R.R. Taylor, "PipeRench: A virtualized programmable datapath in 0.18 micron technology," Proceedings of the IEEE 2002 Custom Integrated Circuits Conference, pp.63–66, IEEE, 2002.

[9] B. Levine, "Kilocore: Scalable, high performance and power efficient coarse grained reconfigurable fabrics," Proc. of International Symposium on Advanced Reconfigurable Systems, pp.129–158, 2005.

[10] J.M. Arnold, "S5: The Architecture and Development Flow of a Software Configurable Processor," Proc. 4th IEEE Int'l Conf. Field Programmable Technology (ICFPT2005), pp.120–128, Dec. 2005.

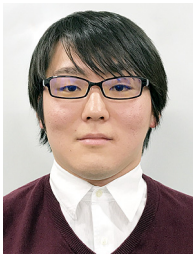
[11] G. Ansaloni, P. Bonzini, and L. Pozzi, "EGRA: A coarse grained reconfigurable architectural template," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.19, no.6, pp.1062–1074, 2011.

[12] K. Masuyama, Y. Fujita, H. Okuhara, and H. Amano, "A 297mops/0.4mw ultra low power coarse-grained reconfigurable accelerator CMA-SOTB-2," 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig), pp.1–6, Dec. 2015.

[13] N. Ando, K. Masuyama, H. Okuhara, and H. Amano, "Variable pipeline structure for coarse grained reconfigurable array cma," Proceedings of the 2016 International Conference on Field-Programmable Technology, FPT 2016, pp.217–220, Institute of Electrical and Electronics Engineers Inc., 5 2017.

[14] Y. Koizumi, E. Sasaki, H. Amano, H. Matsutani, Y. Take, T. Kuroda, R. Sakamoto, M. Namiki, K. Usami, M. Kondo, and H. Nakamura, "Cma-cube: A scalable reconfigurable accelerator with 3-d wireless inductive coupling interconnect," 22nd International Conference on Field Programmable Logic and Applications (FPL), pp.543–546, Aug. 2012.
[15] Y. Matsushita, H. Okuhara, K. Masuyama, Y. Fujita, R. Kawano, and H. Amano, "Body bias grain size exploration for a coarse grained reconfigurable accelerator," FPL 2016 - 26th International Conference on Field-Programmable Logic and Applications, Institute of Electrical and Electronics Engineers Inc., 9 2016.
[16] A. Kourfali, A. Kulkarni, and D. Stroobandt, "SICTA: A superimposed in-circuit fault tolerant architecture for SRAM-based FPGAs," 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), pp.5–8, July 2017.
[17] A.S.B. Lopes, E. Santos, M. Kreutz, and M. Pereira, "A Runtime Mapping Algorithm to Tolerate Permanent Faults in a CGRA," 2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC), pp.63–70, Nov. 2016.
[18] S. Eisenhardt, A. Küster, T. Schweizer, T. Kuhn, and W. Rosenstiel, "Spatial and Temporal Data Path Remapping for Fault-Tolerant Coarse-Grained Reconfigurable Architectures," 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, pp.382–388, Oct. 2011.
[19] T. Ikezoe, T. Kojima, and H. Amano, "A coarse-grained reconfigurable architecture with a fault tolerant non-volatile configurable memory," 2019 International Conference on Field-Programmable Technology (ICFPT), pp.81–89, IEEE, 2019.
[20] R. De Rose, M. Lanuzza, F. Crupi, G. Siracusano, R. Tomasello, G. Finocchio, M. Carpentieri, and M. Alioto, "A variation-aware timing modeling approach for write operation in hybrid cmos/stt-mtj circuits," IEEE Transactions on Circuits and Systems I: Regular Papers, vol.65, no.3, pp.1086–1095, March 2018.
[21] A.F. Vincent, N. Locatelli, J. Klein, W.S. Zhao, S. Galdin-Retailleau, and D. Querlioz, "Analytical Macrospin Modeling of the Stochastic Switching Time of Spin-Transfer Torque Devices," IEEE Transactions on Electron Devices, vol.62, no.1, pp.164–170, Jan. 2015.
[22] T. Kojima, N. Ando, Y. Matshushita, H. Okuhara, N.A.V. Doan, and H. Amano, "Real chip evaluation of a low power cgra with optimized application mapping," Proceedings of the 9th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, p.13, ACM, 2018.

Takeharu Ikezoe received BS degree from Keio University, Yokohama, Japan, in 2018. He is a master student in Keio university in the presence.



Takuya Kojima received his B.E. and M.E. degrees from Keio University, Japan, in 2017 and 2019, respectively. He is currently working towards a Ph.D. degree at Keio University. He is also working as a JSPS research fellow (DC1) from 2019. His research interests include optimization methods, especially for reconfigurable devices, and heterogeneous computing with 3-D stacked LSI. He is a member of IEEE and IEICE.

Hideharu Amano received Ph.D. degree from the Department of Electronic Engineering, Keio University, Japan in 1986. He is currently a professor in the Department of Information and Computer Science, Keio University. His research interests include the area of parallel architectures and reconfigurable systems.