Implementing Flexible Reliability in a Coarse-Grained Reconfigurable Architecture

Dawood Alnajjar, *Student Member, IEEE*, Hiroaki Konoura, *Student Member, IEEE*, Younghun Ko, Yukio Mitsuyama, *Member, IEEE*, Masanori Hashimoto, *Senior Member, IEEE*, and Takao Onoye, *Senior Member, IEEE*

***Abstract*— This paper proposes a coarse-grained dynamically reconfigurable architecture that offers flexible reliability to deal with soft errors and aging. The notion of a cluster is introduced as a basic architectural element; each cluster can select four operation modes with different levels of spatial redundancy and area efficiency. We evaluate the aging effect due to negative bias temperature instability and illustrate that periodically alternating active cells with resting ones will greatly mitigate the effects of the aging process with a negligible power overhead. The area of circuits that are added for immunity to soft errors and for mitigating aging effects is 29.3% of the proposed reconfigurable device. A fault-tolerance evaluation of a Viterbi decoder mapped on the architecture suggests that there is a considerable tradeoff between reliability and area overhead. Finally, we design and fabricate a test chip that contains a 4 × 8 cluster array in a 65-nm process and demonstrate its immunity to soft errors. Accelerated tests using an alpha particle foil showed that the mean time to failure and failure in time are well characterized with the number of sensitive bits and that our architecture can trade off soft error immunity with the area of implementation.**

***Index Terms*— Aging, coarse-grained architecture, reconfigurability, reliability, soft error.**

I. Introduction

WITH aggressive process scaling, sustaining reliability has become a major concern in VLSI design. As devices get smaller, the critical charge, which is the minimum charge to cause a bit flip, becomes smaller, and the functional correctness becomes more susceptible to single-event effects, also known as soft errors. In addition, negative bias temperature instability (NBTI) has become prominent, and designers are encountering complications in the guaranteeing of device life time due to the aging process. Reliability requirements depend on the application and operating environment, and, hence, there is a growing demand for a design scheme that can flexibly choose countermeasures to prevent reliability degradation.

Manuscript received January 31, 2012, revised June 22, 2012, and accepted October 20, 2012. Date of publication December 19, 2012; date of current version October 14, 2013.

D. Alnajjar, H. Konoura, M. Hashimoto, and T. Onoye are with the Department of Information Systems Engineering, Osaka University, Suita-shi 565-0871, Japan, and also with JST, CREST, Tokyo 102-0075, Japan (e-mail: dawood.alnajjar@ist.osakau.ac.jp; konoura.hiroaki@ist.osaka-u.ac.jp; hasimoto@ist.osaka-u.ac.jp; onoye@ist.osaka-u.ac.jp).

Y. Ko is with the R&D Department, Sometech Inc., Seoul 152-775, Korea (e-mail: jazzbasskyh@sometech.com).

Y. Mitsuyama is with the School of Systems Engineering, Kochi University of Technology, Kami-shi 782-8502, Japan, and also with JST, CREST, Tokyo 102-0075, Japan (e-mail: mitsuyama.yukio@kochi-tech.ac.jp).

Digital Object Identifier 10.1109/TVLSI.2012.2228015

In order to mitigate NBTI, the dependence of NBTI on the design and operating parameters has been studied. pMOS transistors age when a negative gate-to-source voltage (V_{gs}) is applied, which is referred to as stress [1], causing a shift in the threshold voltage ($|V_{th}|$) and, consequently, the circuit delay increases. Once the stress is removed, $|V_{th}|$ partially recovers. Consequently, the ratio of the stress and recovery phases and their frequency are important factors for characterizing NBTI [2], [3]. In particular, long-term stress, e.g., for 10 s or more, heavily degrades the performance. Therefore, avoiding such stressful operation helps to extend the device lifetime. The replacement of active functional units with resting spare units is a promising way for dealing with NBTI in reconfigurable devices. In addition, replacement without downtime, which is referred to as "hot-swapping", would be desirable from the perspective of an application's execution. In fact, swapping techniques such as alternating active lookup tables (LUTs) and wires with spare elements have already been devised for mitigating NBTI in field-programmable gate arrays (FPGAs) [4].

Soft-error-tolerant designs have been created especially for aerospace applications. Time redundancy, spatial redundancy, error correction coding (ECC), and other techniques have been widely studied and utilized to detect soft errors and avoid failures [5]–[7]. In particular, a reconfigurable device is suitable for applications that have spatial redundancy, since redundant hardware, e.g., triple modular redundancy (TMR), can be easily realized with a regular array structure. Voters and ECC circuits are essential elements for making circuits immune to soft errors. In the case of fine-grained reconfigurable devices such as FPGAs, such circuits can be implemented with LUTs in any part of the device, and many techniques for making fault-tolerant FPGAs have been proposed [8]–[12], [23].

Unlike fine-grained architectures, which have huge routing area overheads and poor routability [13], coarse-grained architectures offer word-level data paths with better performance, reduction in placement and routing efforts, less routing area overhead, and dramatic reduction in the configuration data by two or more orders of magnitude. Especially, the last advantage is highly desirable from a reliability point of view, since smaller configuration data inherently bring better robustness against soft errors and less effort for implementing hot-swapping. However, typical coarse-grained architectures are inefficient in implementing voters since conventional coarse-grained reconfigurable architectures have no reliability consideration and do not have such functionalities in their basic elements.

For this reason, researchers have tried to develop coarse-grained architectures with reliability considerations [14]–[19]. In particular, [14], [15] describe a reconfigurable architecture that uses residue modulo 3 codes for mitigating soft errors instead of TMR and duplication with comparison. In [16], a fault-tolerant architecture that incurs less overhead through implementing the voters and comparators associated with redundancy on the processing element array itself was proposed. Moreover, [17] describes a radiation-hardened field-programmable object array for space applications. However, details regarding the mitigation mechanism of the radiation effect are not disclosed, as far as we are aware. In [18], we reported our preliminary work on a coarse-grained dynamically reconfigurable architecture with flexible reliability enhancements, wherein the reliability level could be selected for each processing element in order to minimize the area and power overhead. Several other fault-tolerant coarse-grained architectures have been proposed in the literature [19], but as far as we can see, none has actually been verified to operate as expected in a high radiation environment. Acceleration tests under alpha particle and neutron beam radiation are very important, since they increase the level of hardware test coverage including soft error occurrence, and they are useful in uncovering unexpected architecture- and circuit-level vulnerabilities. Furthermore, the level of reliability provided by these architectures has not been quantitatively investigated, and it seems impossible to know whether there is an excessive consideration for reliability, which imposes a large area and power penalty, or a deficient consideration for reliability causing unpredictable results and system failures.

This paper discusses a new reconfigurable architecture that offers flexible reliability for soft errors and aging. Its main contributions are as follows.

1) A coarse-grained dynamically reconfigurable architecture in which the reliability level for each processing element can be chosen flexibly is presented, focusing on the additional mechanism to change reliability levels depending on applications and environments. For reliability-oriented applications, the reconfigurable architecture achieves a sufficient level of reliability at the cost of an area and power overhead, while for cost-oriented applications it provides area/power efficiency. Note that reliability requirements vary even among circuit modules within an application: e.g., the control parts of an application may require higher reliability level than that of data-path parts. Circuits on timing-critical paths with a shorter delay margin are more sensitive to delay increase due to aging and need NBTI recovery by hot-swapping. We devised a scheme where the reliability level can be individually selected for each basic element simply by selecting the operation mode of that basic element to reduce the area and power overhead by avoiding excessive reliability as much as possible. This enables device users to systematically trade off area for improving the reliability without having a deep knowledge of reliability enhancement techniques.
2) The robustness of the architecture to alpha particle radiation is demonstrated. Radiation tests on a test chip fabricated in the 65-nm process are carried out, and the experimental results of accelerated tests show that the mean time to failure (MTTF) and the failure in time (FIT) rate are well characterized with the number of sensitive bits, which is a variable estimated solely through simulation. Moreover, test results also demonstrate that this architecture can vary the level of soft-error immunity through reliability-aware mapping. This is very important since the obtained hardware and model correlation allows us to identify the minimum resources necessary to achieve the required FIT rate.

The remainder of this paper is organized as follows. Section II describes how our architecture achieves flexible reliability. Section III introduces the coarse-grained reconfigurable architecture with flexible reliability. Section IV shows the area, soft error, and aging evaluations. Section V explains the setup, procedure, analysis, and discussion of the accelerated tests. Conclusions are finally drawn in Section VI.

II. Flexible Reliability in Architecture Design

In order to achieve an appropriate level of reliability together with area efficiency, each basic element of the reconfigurable device should be able to change its own reliability by referring to the sensitivity to soft errors and circuit aging. Reliability classifications to soft errors and aging must be taken into account in the architecture design stage of the basic elements. This section reviews the reliability classifications and defines the operation modes that have different levels of reliability versus soft errors and aging. Throughout this paper, a single soft error in a memory element and a soft error in combinational logic will be referred to as a single event upset (SEU) and a single event transient (SET), respectively.

A. Reliability Classification to Soft Errors

In reconfigurable systems, the reliability of the configuration memory is often considered to be more critical than that of the computed data, since an SEU in the configuration memory damages the functionality until the configuration data is reloaded again; we will refer to this as a permanent error. Focusing on soft errors, we suppose the following four conditions for CS1–CS4, ordered from the most demanding to the least demanding, that we aim to satisfy with the basic elements of the reconfigurable architecture in this paper:

1) CS1: the functionality must be correct, and the computed data must be correct as well;
2) CS2: the functionality must be correct, and the errors in the computed data can be detected; however, only some of them can be corrected;
3) CS3: the functionality must be correct, but errors in the computed data are not considered;
4) CS4: error detection and recovery is unnecessary.

B. Reliability Classification to Aging

In a coarse-grained reconfigurable device, aging of the data processing elements, especially those that may compose critical paths, must be mitigated in order to avoid failures and

TABLE I
RELIABILITY LEVELS TOGETHER WITH SOFT ERROR AND AGING CONDITIONS

Reliability Levels	Conditions	
	Soft Error	Aging
RL1	CS1	CA1
RL2	CS2	CA1
RL3	CS3	CA2
RL4	CS4	CA2

prolong the device lifetime. On the other hand, the circuits related to the configuration memory never form speed-limiting paths. Accordingly, we should set the following aging-oriented conditions for CA1 and CA2:

1) CA1: the effect of aging of an execution module must be considered;
2) CA2: aging mitigation is unnecessary.

C. Definition of Four Reliability Levels

Taking conditions mentioned in Sections II-A and II-B into consideration, in order to achieve different levels of reliability to soft errors and to circuit aging, we define four reliability levels RL1–RL4 as summarized in Table I. In the conditions of CS1 and CS2, we assume that the circuit aging mitigation in data processing elements should be considered, while in CS3 and CS4 the circuit aging mitigation is compromised. The following sections describe architecture for selecting among these four reliability levels with a small area overhead.

III. ARCHITECTURE FOR FLEXIBLE RELIABILITY

Here we describe the reconfigurable architecture with flexible reliability for soft errors and circuit aging.

A. Reconfigurable Architecture for Soft Error Tolerance

Fig. 1 illustrates the architecture. Having designed an architecture independent of the data-path width, we represent the width by using the variable n. Clusters, which are basic elements of our architecture, are placed in a 2-D array. A cluster core has a switch (ConfSM: configuration memory switching matrix) and four cells, each of which consists of an execution module (EM) in the case of the ALU and the multiplier cluster or a register module (RM) in the case of the register cluster, three configuration memories (ConfMem) for dynamically configuring the EM/RM, and voters (VCs), all in a reconfigurable cell unit. For flexible reliability, the architecture has a redundancy control unit (RDU) and a comparing and voting unit (CVU).

The cluster interconnect is composed of inner and outer layer connections and a swap Mux layer. The outer layer connection has four tracks (Track 0–3), and each cell inside a cluster is placed on one of them. Thus, each cell in a cluster can be connected to the cells in adjacent clusters on the same track. The outer layer connection also has a diagonal track connecting adjacent cells in one cluster. The inner layer connection is composed of multiplexers and demultiplexers to select the inputs and direct the outputs of the cells. The swap Mux layer contains the multiplexers necessary for hot-swapping. The cluster swap Mux layer and inner/outer layer connections will be explained in Sections III-B and III-C, respectively. This overall interconnect enables application mapping in all four operation modes, as summarized in Table II, and these modes correspond to the four conditions RL1–RL4 mentioned in Table I. TMR for satisfying RL1, double modular redundancy (DMR) for satisfying RL2, single modular with single context (SMS) for satisfying RL3, and single modular with multicontext (SMM) for satisfying RL4 offer different capabilities of dynamic reconfigurability (#contexts) and throughput per cluster. In the case of TMR, DMR, and SMS, as shown in Fig. 2, each cell has three redundant ConfMems which contain one context, three VCs, a selector (part of the ConfSM), and the EM/RM. An error occurring in the ConfMem caused by SEU will be corrected when the next clock is given to the ConfMem, since the voted value is rewritten to the ConfMem every clock cycle. This configuration data is stored with bitwise TMR and, therefore, bit flips caused by multiple SEUs in different bits will also be corrected when the next clock is given. Note that the power overhead resulting from the continuous overwriting depends on the required reliability since the reliability depends on the raw soft error rate of ConfMem and the overwriting frequency.

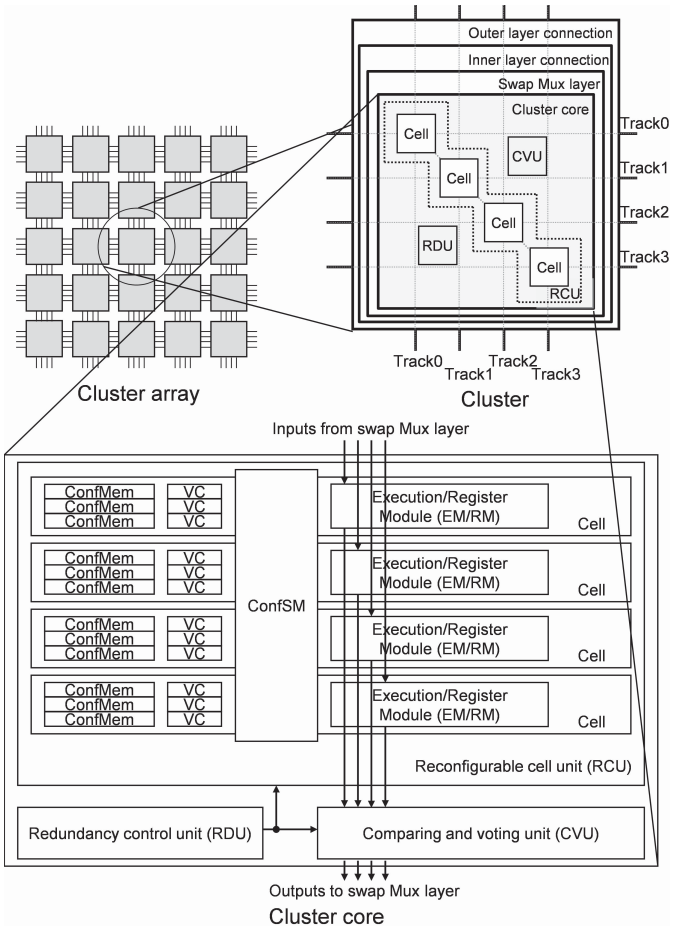


Fig. 1. Cluster and cluster interconnect.

TABLE II
REDUNDANCY AND RELIABILITY IN FOUR OPERATION MODES

Operation Mode	Reliability to Soft Errors							Reliability to Aging		
	Redundancy		SEU	SEU	SET	Utilization		Mitigation	#cells	
	ConfMem	EM/RM	in ConfMem	in EM/RM	in EM/RM	#contexts	#cells		Active	Resting
TMR (RL1)	3	3	Detect & Recover	Detect & Recover	Detect & Recover	3	3	Y	3	1
DMR (RL2)	3	2	Detect & Recover	Detect & Recover	Detect	2	2	Y	2	2
SMS (RL3)	3	1	Detect & Recover	Detect	Cannot detect	1	1	N	4	0
SMM (RL4)	1	1	Cannot detect	Detect	Cannot detect	3	1	N	4	0

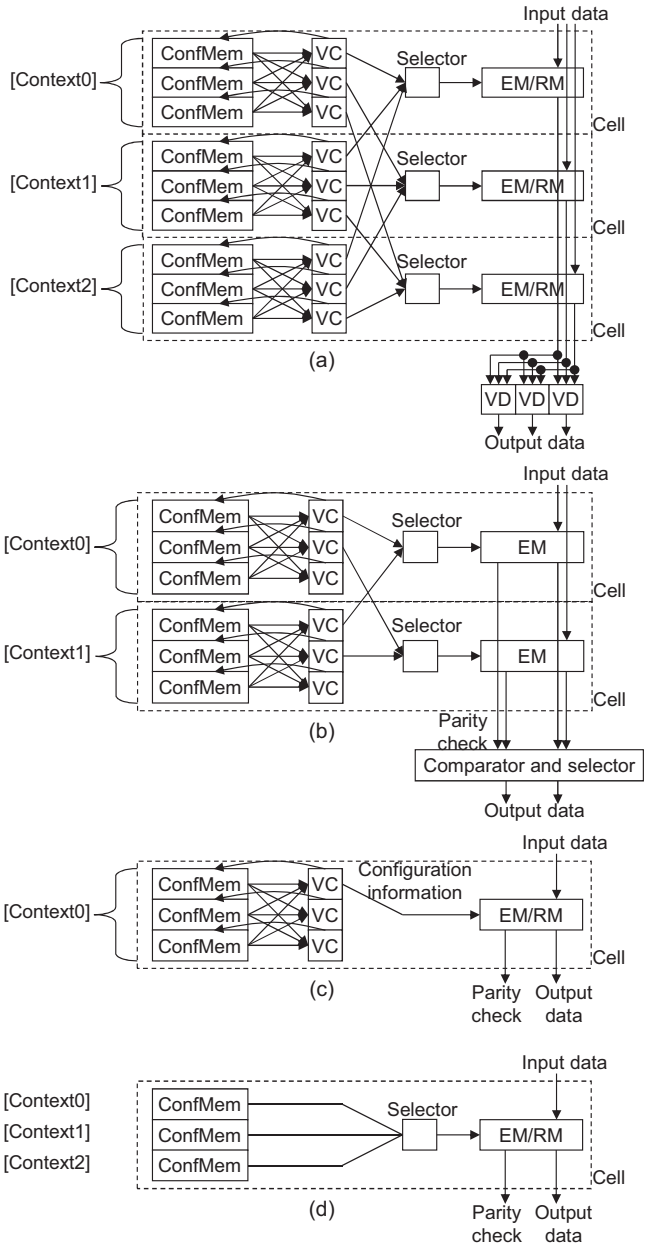


Fig. 2. Operations in (a) TMR, (b) DMR, (c) SMS, and (d) SMM modes.

On the other hand, in SMM mode, the voters are disabled, and three contexts are stored in the ConfMem of each cell. With this implementation, the user can choose the operation modes, depending on the importance of different types of soft errors. For example, when SET occurrence is rare, SMS mode could be used. In contrast, when SETs occur frequently, TMR mode must be used. Thus, our architecture can be flexibly adapted to the user requirements.

TABLE III
ELIMINATION OF SOFT ERRORS IN DIFFERENT BLOCKS IN TMR MODE

Block	Error Recovered By
ConfMem	VC
VC	VD
Selector	VD
EM (SEU)	VD
EM (SET)	VD
VD	VD at successive clusters
Swap Mux layer	VD at successive clusters
Inner/Outer layer conn.	VD at successive clusters

In the TMR mode shown in Fig. 2, the outputs of three EMs/RMs pass through the three voters (VD), which are parts of the CVU, while the fourth cell is reserved as a spare cell for the hot-swapping operation shown in Section III-B. A SET or SEU occurring in the VC, selector, or EM will be recovered in the VDs. A SET occurring in the VDs (i.e., in the CVU) will propagate to successive clusters. Through the prohibition of data feedback inside a cluster and enforcement of voting at every output of the cells, the architecture can prevent errors from accumulating in the EM/RM without the need for a rollback mechanism. In DMR mode, on the other hand, the outputs of each EM along with the parity bits are directed to the Comparator and selector, which is also part of the CVU. SEUs occurring in the registers of the EM are detectable using parity bits, and they can be recovered in the Comparator and selector by selecting the correct output. However, SETs in the VC, Selector, or EM can only be detected in the Comparator and selector. A SET occurring in the Comparator and selector can be detected as well by consecutive clusters in DMR mode. With external support, such as executing an interrupt service routine on a processor, cycle re-execution can be performed. In the case of SMS and SMM modes, only SEUs in the registers of the EM can be detected.

The (RDU) configures the operation mode of the cluster and the context selection stored in the ConfMem. The RDU holds six-bit configuration data; two bits for the operation mode selection, two bits for cell usage selection for TMR and DMR in the case of hot-swapping, and two bits for the context selection. This configuration data is stored with bitwise TMR, and, hence, the RDU can tolerate SEUs. The dynamic context selection can be carried out simply by changing the

TABLE IV
ELIMINATION OF SOFT ERRORS IN DIFFERENT BLOCKS IN DMR MODE

Block	Error Recovered By
ConfMem	VC
VC	Comparator and selector
Selector	Comparator and selector
EM (SEU)	Comparator and selector
EM (SET)	(re-execution)
Comparator and selector	(re-execution)
Swap Mux layer	(re-execution)
Inner/Outer layer conn.	(re-execution)

two bits in the RDU. This paper does not describe the dynamic context control circuit. It will be described in future works. SETs in the swap Mux layer and inner and outer layer connections propagate to successive clusters and are corrected in TMR mode and detected in DMR mode. Tables III and IV summarize the soft error mitigation in different blocks.

The ConfMem size varies between the ALU, multiplier, and register clusters and depends on n. The ConfMem configures the interconnect and the EM/RM of the cell, which will be explained in this section. The ConfSM is responsible for selecting a context for each cell and enabling dynamic reconfiguration and the four operation modes.

B. Reconfigurable Architecture for Mitigating Aging

The hot-swapping operation, which can replace active functional units with resting spare units during runtime, is performed in TMR and DMR modes in order to mitigate circuit aging. Fig. 3(a) depicts the swap Mux layer. Hot-swapping multiplexers are labeled with the letter H. In TMR mode, there are three redundant cells and one spare cell, as explained in Section III-A, and three redundant cells are in the stress phase, while one spare cell is in the recovery phase. Consequently, the EMs rest alternately through time; one rests while the others perform data processing on the input data 0, 1, and 2; e.g., EM0, EM1, and EM2 can be replaced with EM0, EM1, and EM3, respectively, after one hot-swapping cycle. As shown in Fig. 3(b), a cell-swapping operation can alternate each active cell and a spare cell. On the other hand, in DMR mode, there is a pair of redundant cells and a pair of spare cells, or two pairs of redundant cells. In the same manner as in TMR mode, a cell-swapping operation can alternate an active cell and a spare cell; e.g., EM0 and EM2 are replaced with EM1 and EM3 after one hot-swapping cycle, as shown in Fig. 3(c).

C. Functionality and Interconnect

The configuration data for each cell in a cluster is composed of the information for functionality and the information for interconnect. Table V lists the number of configuration bits for each type of cluster for $n = 8$. For $n = 16$ and $n = 32$, the number of configuration bits increases by 8 and 24, respectively, for all three cluster types. This increase is due to the initial/constant values stored in configuration bits. The initial/constant values are necessary when mapping equations with fixed values (e.g., $y = x + 5$).

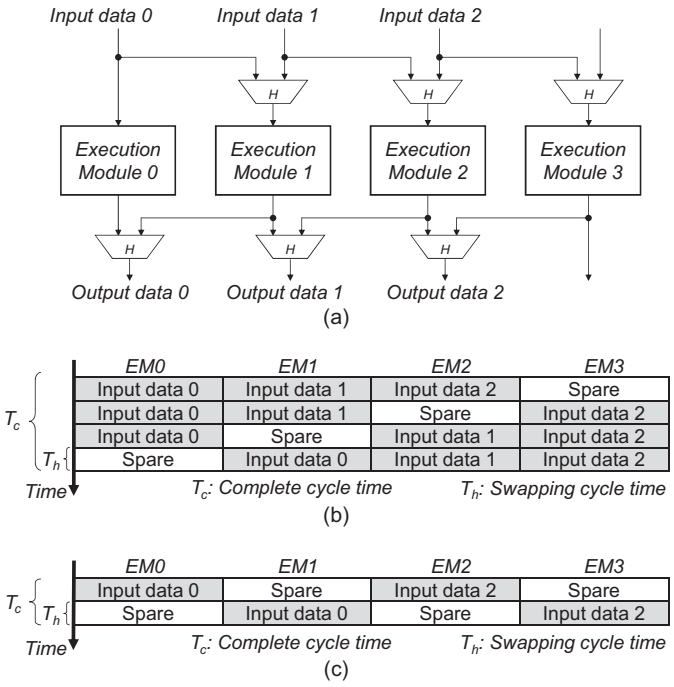


Fig. 3. (a) Swap Mux layer. (b) Hot-swapping operations in TMR. (c) Hot-swapping operations in DMR.

TABLE V
CONFIGURATION DATA DETAILS ($n = 8$)

Cluster Type	# of ConfMem bits		Total no. of of Bits
	Functionality	Interconnect	
ALU	26	37	63
Multiplier	28	32	60
Register	15	36	51

In the ALU cluster, each EM in the cell has an n-bit ALU, a shifter, and a parity checker, as depicted in Fig. 4. According to the configuration memory, the EM can be configured to perform addition and subtraction operations with or without cooperation of the neighboring cells in the same cluster. It also can be configured to perform logical operations such as logical AND and OR, multiplexing, and fixed or variable shifting. In SMS or SMM mode, each cell in a cluster can cooperate with the neighboring cells to perform a multibyte operation. For example, four cells can work collaboratively to perform a $4n$-bit operation.

The EM in the multiplier cluster contains a multiplier, an adder, and a parity checker. Similarly, corresponding to the bit string stored in ConfMem memory, the multiplier can be configured to perform $n \times n$ bit signed/unsigned multiplication. However, unlike in the ALU cluster, the maximum bit operation is n in all operation modes.

The ALU EM has two n-bit registers for input (A, B), an output register (Y), one 1-bit parity register for each input/output register, and 1-bit register for the flag output. The Multiplier EM is similar to that of the ALU, while the input registers are four instead of two. The ConfMem also controls the configuration of all registers. They can be configured to be either pipeline registers, which store new data every cycle, or disabled or fixed to store a constant value.

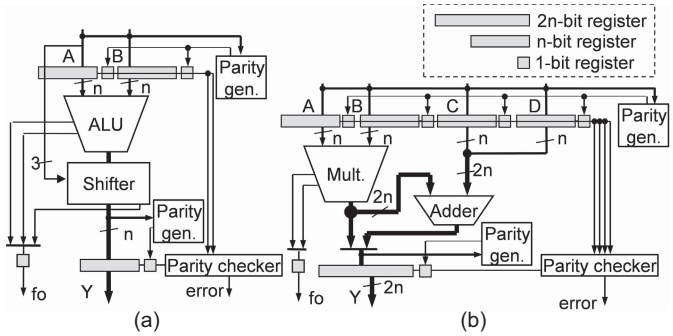


Fig. 4. EM of ALU and multiplier clusters. (a) ALU EM. (b) Multipler EM.

On the other hand, RMs in register clusters contain a 16-word register file with an n-bit word size. The register file works not only as a register file but also as a delay unit, which outputs the input data after 1 to 16 cycles, or as an LUT. In the case of TMR, simple voting at the output of three cells is not an adequate mitigation technique, since the soft error will remain in the register file. Consequently, we perform a successive data write of the voted value after every data read in the TMR mode to ensure that soft errors are eliminated. As for DMR mode, it is not implemented in the register cluster.

The architecture of the interconnect is rather consistent for the different types of clusters. As mentioned in Section III-A, the cluster interconnect contains two layers, the outer layer shown in Fig. 5, which enables the routing between different clusters, and the inner layer shown in Fig. 6, which contributes to the routing inside one cluster. Cells of one cluster are connected to the cells in the neighboring clusters using horizontal and vertical tracks, while cells in one cluster are connected to each other using a diagonal track, as shown in Fig. 1. Each track contains two 8-bit interconnect wires, e.g., A and B, for $n = 8$. Furthermore, each cell also contains 18 switches controlled by the ConfMem of the cell. It allows each EM/RM of cell X to select the necessary inputs from cluster ports AX0∼AX3 and BX0∼BX3, and it also allows the output to be demultiplexed to one of the ports AX0–AX3 and BX0–BX3.

Special routing is necessary between the clusters of the data source and data sink if they are configured in different operation modes. Let us suppose a situation in which cluster 1 configured with SMS mode needs to deliver data to cluster 2 configured in TMR mode (Fig. 7). In this case, the output of cluster 1 must be distributed to three cells in cluster 2, and the output can be distributed using the diagonal and horizontal tracks of the outer layer connection.

IV. Architecture Evaluation

This section evaluates the proposed architecture in terms of permanent soft error rates, circuit aging, and area overhead.

A. Soft Error Tolerance Evaluation

Let us evaluate the tradeoff between area and reliability of the cluster array on which an example application is implemented in TMR and SMM modes. Reliability is evaluated in a way similar to that in [21] and [22], that is, by counting the number of "sensitive bits" in each cell. Here, a configuration memory element that impacts the primary output of a particular design is defined as a "sensitive bit."

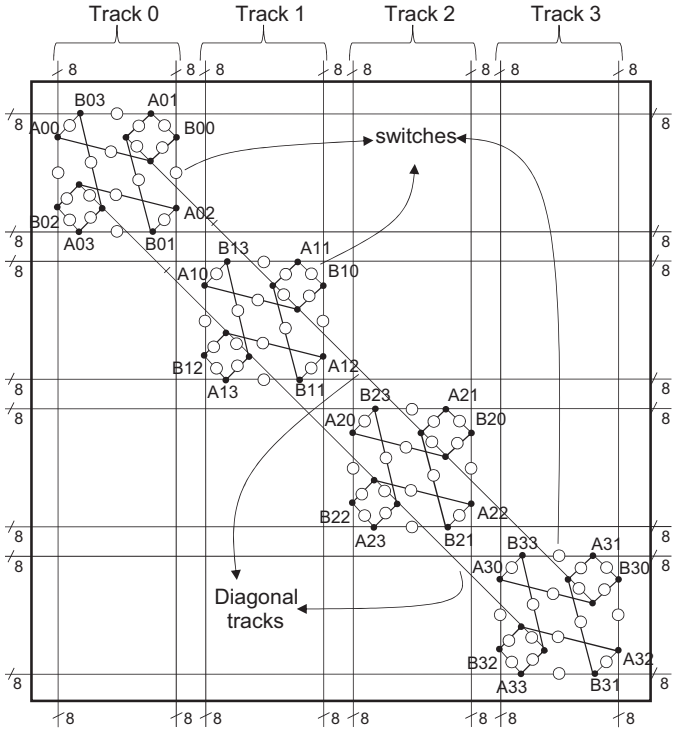


Fig. 5. Outer layer connection of one cluster ($n = 8$).

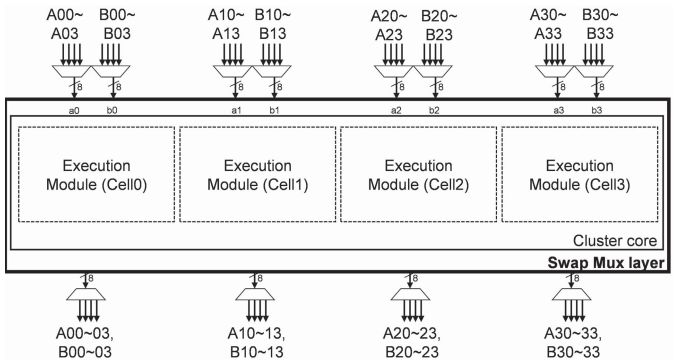


Fig. 6. Inner layer connection of one cluster ($n = 8$).

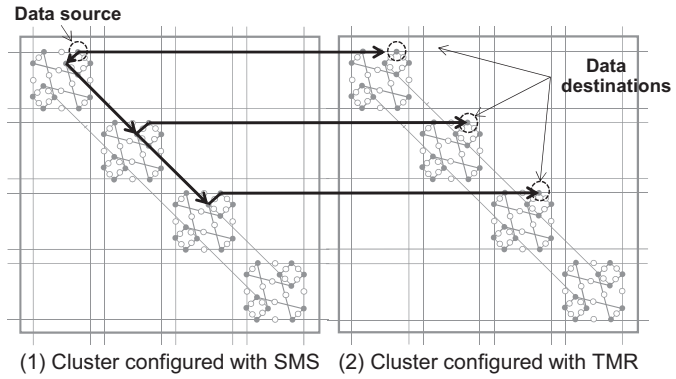


Fig. 7. Example of routing from a cluster configured in SMS mode to another configured in TMR mode.

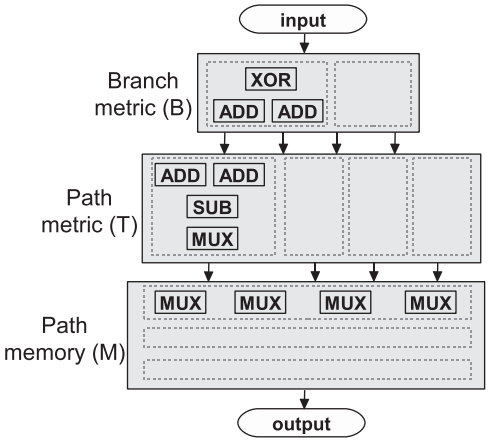


Fig. 8. Overview of Viterbi decoder.

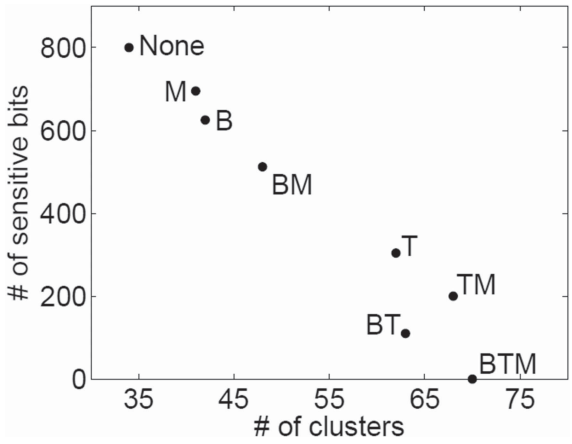


Fig. 9. Area–reliability tradeoff of the Viterbi decoder. The notation represents TMRed units in the implementation, where B, T, and M stand for branch metric, path metric and path memory units, respectively. For example, TM implies that the path metric (T) and the path memory (M) are TMRed while the branch metric is not.

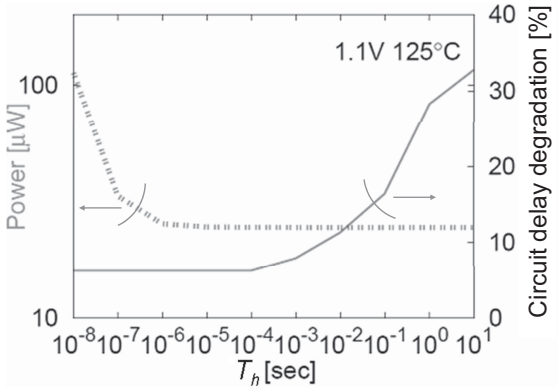


Fig. 10. Tradeoff between aging mitigation and power consumption (after 10 years).

To count the number of sensitive bits, a cycle-based simulation is performed on implementations with each single bit in configuration flipped reflecting an SEU. If the primary output becomes erroneous, the bit is classified as a sensitive bit. SETs are not considered in this tolerance evaluation methodology since SEUs are the main consequence of a single event [23]. SETs will be evaluated in future works.

The example application is a Viterbi decoder (with constraint length 3) manually mapped to the cluster array. As illustrated in Fig. 8, Viterbi decoding is divided into three parts: the branch metric unit (B), the path metric unit (T), and the path memory unit (M). When it is mapped on the architecture, each unit is implemented in either TMR mode or SMM mode. Consequently, eight different combinations of mapping the Viterbi decoder are obtained. Each combination is denoted by the TMRed group; e.g., BT implies that B and T are mapped in TMR mode and the rest of the units (M) are mapped in SMM mode.

Fig. 9 shows the number of required clusters and the number of sensitive bits in each configuration pattern. In "None," the number of sensitive bits is 790 with 34 clusters, while in "BTM" the number of sensitive bits is 0 with 70 clusters. The other configurations reveal that different numbers of sensitive bits are obtained with different mapping areas, and thus the reliability levels achieved are different. Therefore, it can be concluded that there is a considerable tradeoff between area and reliability. Note that the mapping has a large effect on the number of sensitive bits. It can be seen that case "T" has almost the same number of clusters as "BT," but there is a difference of approximately 200 sensitive bits. The relation between MTTF and the number of sensitive bits will be investigated in Section V. This evaluation suggests that careful selection of the operation mode of each cluster can improve the area–reliability tradeoff.

B. Aging Process

Mitigating aging by hot-swapping heavily depends on the swapping cycle time T_h. Moreover, power consumed due to power gating for recovery also depends on T_h. We evaluated the worst V_{th} shift using the NBTI long-term prediction model [2] assuming that all pMOSs in a cell are under stress when the cell is active. In TMR mode, as illustrated in Fig. 3(b), the stress phase lasts for $3T_h$ and it is followed by a recovery phase of T_h. In DMR mode, as illustrated in Fig. 3(c), the stress and recovery phases change alternately in T_h. We gave the estimated V_{th} shift to a commercial transistor-level static timing analyzer and evaluated the increase in circuit delay. We also evaluated the power dissipation due to power gating by circuit simulation with Synopsys Nano Time. Fig. 10 illustrates the circuit delay degradation after 10 years and the power consumption as a function of T_h. By assigning T_h to values from 10^{-5} to 10^{-4} s, the delay degradation is reduced to 6.2% within 2% power overhead, whereas the delay degradation would reach up to 35% without hot-swapping. Fig. 11 illustrates the circuit delay degradation as $T_h = 10^{-4}$ s. As we assume that the lifetime ends when the degradation of circuit delay reaches 10%, the lifetime of a cluster without hot-swapping is less than 1 year. However, those of TMR and DMR mode with hot-swapping are more than 10 years. Thus, the hot-swapping operation can effectively mitigate circuit aging.

We next study the impact of hot-swapping on the application level by projecting the results obtained from investigating the

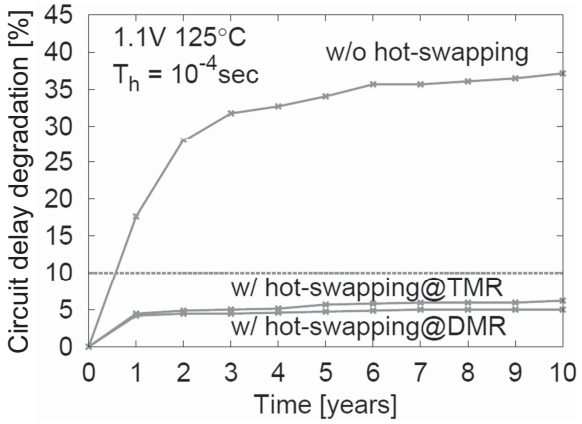


Fig. 11. Comparison of circuit delay degradations.

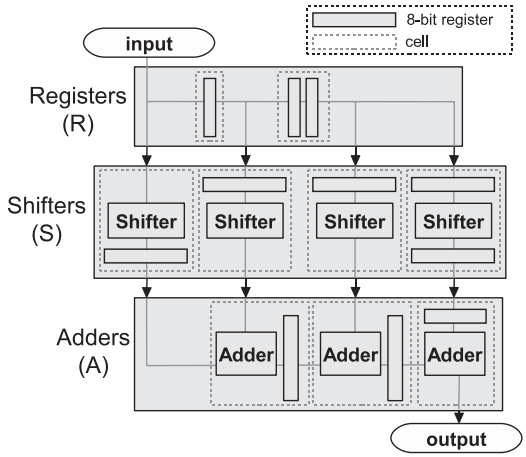


Fig. 12. Implementation A of a four-tap FIR filter.

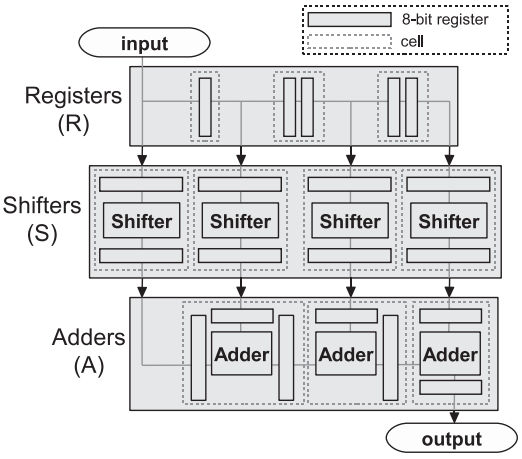


Fig. 13. Implementation B of a four-tap FIR filter.

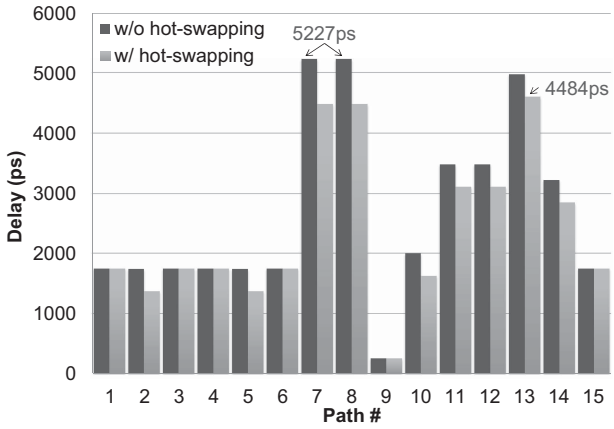


Fig. 14. Delay of each path in implementation A.

hot-swapping effect on a single execution module shown in Fig. 11. NBTI degradation affects all clusters, and it may seem reasonable to perform hot-swapping on all clusters to mitigate degradation and extend the life span of the device. However, the failures in digital logic due to NBTI are mainly timing failures in the critical path. Therefore, to minimize the number of clusters that are configured in TMR and DMR modes with hot-swapping, hot-swapping should be performed only on the clusters that are involved in the critical path of the application. Motivated by this, we further evaluate the impact of the NBTI mitigation technique on an application circuit that is mapped in two different ways in TMR mode. In this evaluation, an FIR filter is selected as the application circuit. The two circuit mappings A and B, respectively consisting of 9 and 10 clusters, are shown in Figs. 12 and 13. Note that this evaluation assumes that all the pMOS in the circuit are under stress when the cell is active. A more precise evaluation considering the stochastic property of pMOS transistor activation will be addressed in future works.

To identify which clusters require hot-swapping and which do not, there is a need to estimate the delay of all the paths of the circuit and determine the longest paths. Therefore, we shall estimate the delay of each input-to-register, register-to-register, and register-to-output path after 10 years with and without hot-swapping for implementations A and B. The delay degradation with and without hot-swapping is assumed to be 6.2% and 35%, respectively, as estimated above.

The estimated delays are shown in Figs. 14 and 15. Here, it should be noted that not all the degradation in the critical path can be mitigated by hot-swapping, since the critical path is composed of execution modules (swappable) and other data routing logic (not swappable). Implementation A contains 15 paths, and there are 3 paths with large delays: i.e., paths 7, 8, and 13. Paths 7 and 8 are composed of two clusters, while path 13 is composed of one cluster. That means, in order to get the maximum frequency results in this circuit, five clusters out of nine need to have hot-swapping in order to reduce the max delay from 5227 ps (191 MHz) to 4484 ps (217 MHz), limited by path 13. As for implementation B, 10 clusters are used containing 24 paths, and paths 23 and 24 have large delays. Knowing that both paths are composed of only one cluster, it can be said that, to achieve the maximum frequency in this circuit, 2 out of 10 clusters need to have hot-swapping in order to reduce the max delay from 3487 ps (287 MHz) to 3115 ps (321 MHz). These results indicate that hot-swapping has a significant impact on the maximum frequency of the application. In summary, the number of clusters that perform hot-swapping is not only application-dependent but also quite limited.

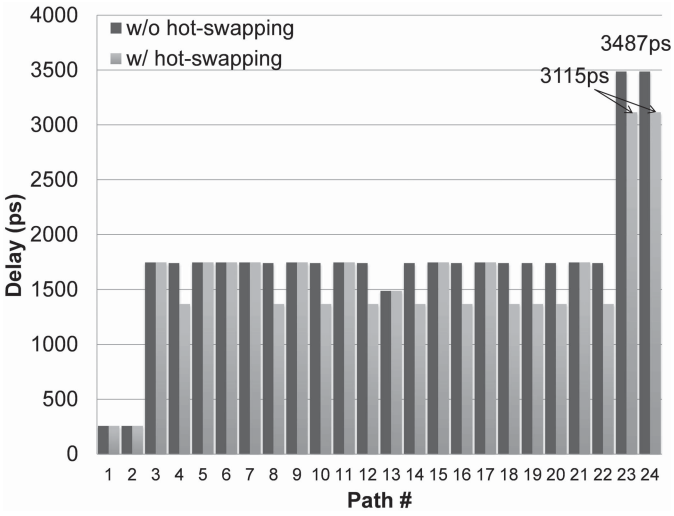


Fig. 15. Delay of each path in implementation B.

TABLE VI
AREA OVERHEAD ($n = 8$)

	ALU Cluster		Mult. Cluster		Reg. Cluster	
Block Name	Area	Overhead	Area	Overhead	Area	Overhead
CVU	412	412	749	749	255	255
RDU	194	194	194	194	194	194
ConfSM	2234	974	2264	1064	1852	832
EMs/RMs	3110	220	6484	484	8894	1220
ConfMem	6532	–	6230	–	5296	–
VCs	3570	3570	3400	3400	2890	2890
Swap Mux layer	235	235	290	290	255	255
Inner/Outer Layer connection	1250	–	1481	–	1343	–
	17 537	5605	21 092	6181	20 979	5676

C. Area Overhead and Performance

Here, we show the area overhead needed to ensure flexible reliability to soft errors and NBTI and to realize the four operation modes. To analyze the overhead quantitatively, we compare the number of gates in the proposed architecture with that of a baseline architecture containing minimum hardware to perform dynamic reconfiguration similar to the proposed architecture, but which is not immune to SEUs or SETs. The baseline architecture would have all ConfMems, since they are necessary for dynamic reconfiguration, and some of the ConfSM, EMs/RMs, and interconnects. In contrast, the proposed architecture includes ConfMems, VCs, EMs/RMs, RDU, CVU, and interconnects. The gate counts of the proposed architecture and the baseline architecture are estimated in the RTL design with an industrial 65-nm cell library and the Synopsys Design Compiler.

The gate counts of units in each cluster are listed in Table VI, where n is 8-bit. The area of additional circuits to provide flexible dependability occupies 27–31.9% of the total area, and the average is 29.3%. Most of the area overhead arises from the voters for the configuration memory (VCs), and overhead for the remaining part is limited. On the other hand, the overhead varies depending on the data width of the architecture. The area overhead of the ALU cluster is reduced to 24.7% and 20.1%, respectively, for $n = 16$ and 32. Furthermore, when comparing the proposed architecture to an architecture with a single reliability option that performs only TMR and hot-swapping, the area overhead for achieving flexible reliability is equal to 14.9%.

TABLE VII
ESTIMATED MAXIMUM OPERATION FREQUENCY ($n = 8$)

Cluster	Max. Freq. (MHz)
ALU	378
Multiplier	240
Register	499

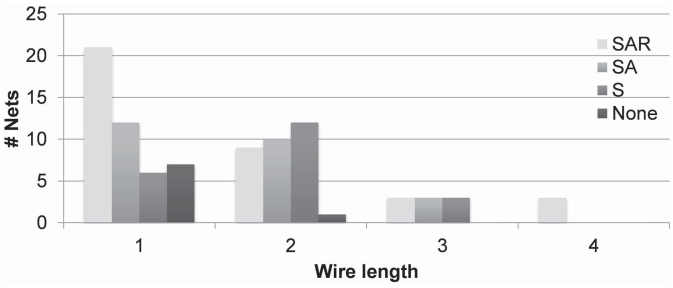


Fig. 16. Wire length distributions of a four-tap FIR filter. The notation represents TMRed units in the implementation, where S, A, and R stand for shifters, adders, and registers, respectively. For example, SA implies that the shifters (S) and adders (A) are TMRed while the register's group is configured in SMM mode.

Table VII summarizes the maximum operation frequency of the three types of clusters. Because the longest critical path is of the multiplier clusters, in applications where multiplier clusters are necessary, the maximum operation frequency is limited to 240 MHz. If multiplier clusters are not used, the operation frequency can be increased to 378 MHz, which is the maximum operation frequency of the ALU cluster. Note that the evaluation of maximum operation frequency assumes that all the input and output registers in EMs are enabled. As for performance, it should be noted that clusters configured in the TMR and DMR modes have a three-fold and two-fold decrease in throughput per area, respectively, when compared with SMS and SMM because they use the available resources in a redundant manner.

The interconnect resource usage is evaluated for four application mappings having different mixtures of TMR and SMM configured clusters. The application is the four-tap FIR filter shown in Fig. 12, and the mappings are identical to ones that will appear in Section V. The interconnect Track usage ranges from 28% to 37% of the total number of available Tracks in the used clusters. The wire lengths of different mappings are shown in Fig. 16. Note that the number of used tracks is not necessarily equal to the number of wires for two reasons: a track that is occupied is not necessarily a wire, and when track A from cluster X is used to pass data to track B in cluster Y, two tracks are used, but they correspond to only one wire. The proposed architecture accommodates the wires even for the mappings consisting of clusters in different operation modes.

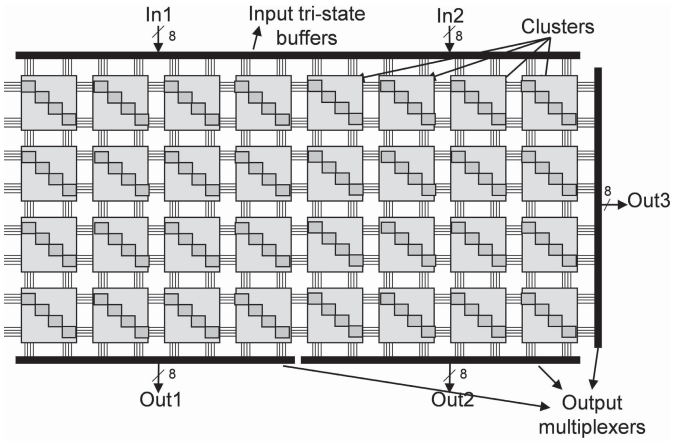


Fig. 17. Structure of the fabricated chip.

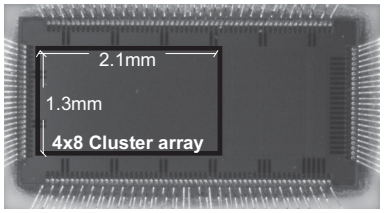


Fig. 18. Micrograph of the fabricated chip.

V. Experiment

A test chip was designed and fabricated to confirm the reliability of the proposed architecture against soft errors, and the MTTF was measured in accelerated radiation tests with various mapping configurations.

A. Design and Specifications

A 4×8 ALU cluster array was fabricated in a 65-nm CMOS process. The design is depicted in Fig. 17, and its micrograph is shown in Fig. 18. In the layout, each cluster occupied an area of 210 μm $\times$ 210 μm. The chip has two 8-bit inputs, In1 and In2, and three 8-bit outputs, Out1 to Out3. The input tri-state buffers are in charge of passing input In1/In2 to selected inputs of the clusters on the top side of the cluster array. The output multiplexers select three 8-bit outputs from the outputs of clusters on the bottom and right side. The configuration data is manually generated for each measurement. During the configuration, a byte of configuration data is stored every cycle in the configuration memories. This cluster array implementation requires 3104 clock cycles for distributing and storing the configuration data. The probability of nonrepairable errors occurring during the configuration is estimated to be sufficiently small since the configuration data sent to the chip in TMR, DMR, and SMS mode is also redundant and errors can be eliminated at the VC by bitwise TMR after the distribution.

B. Reliability Evaluation Setup

This section explains the setup of the radiation experiment for evaluating the MTTF of several mapped circuits having different numbers of sensitive bits and different occupied areas in the cluster array. Fig. 19 shows the measurement setup. The inputs and outputs of the fabricated test chip are controlled by a Stratix II FPGA. In addition, the FPGA is used to implement the golden circuit that produces the correct output necessary for comparison. The comparator unit on the FPGA compares the output of the mapped circuit on the fabricated chip with the output of the golden circuit on the FPGA. Input data is given constantly from the PC through the FPGA until a single inconsistency is detected by the comparator, which constitutes a failure. When a failure is observed, the time to failure (TTF) is recorded in the local memory on the FPGA, and, at the same time, the configuration control block is triggered to reconfigure the fabricated chip and restart the TTF evaluation. The reconfigurable array was operating on a 4-MHz clock, which was determined by the constraints of the evaluation board and the packaging of the chip, not by the operation of the cluster array.

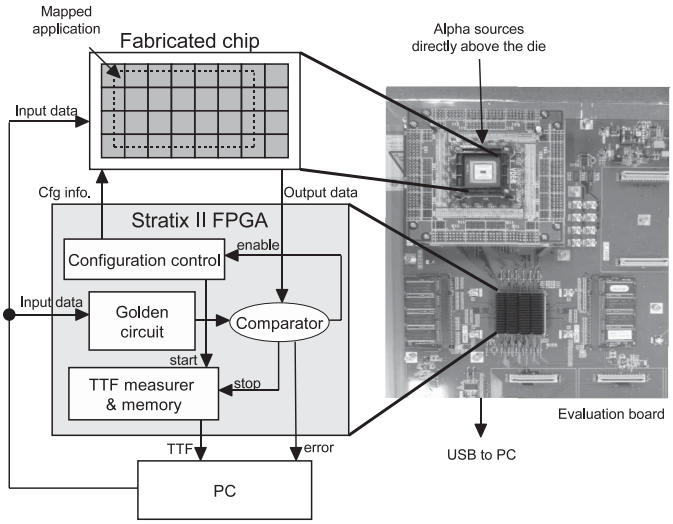


Fig. 19. Experimental setup.

The alpha particle source was Am-241 foil with a flux of 9×10^9 cm^{-2}h^{-1} . The main peak energy of the alpha particles is 5.49 MeV. During the TTF measurement, the foil was placed immediately above the fabricated die in the manner described in [24] and [25]. Note that the FPGA was not irradiated with alpha particles in this setup.

C. Mapped Circuits and Results

Several circuits were selected for the mapping and MTTF evaluation. The operation modes used for mapping applications on the fabricated chip are SMM and TMR modes. Each circuit has a unique number of sensitive bits, which we expected would be an indicator of the vulnerability of the circuit to soft errors. We selected two circuits for this reliability evaluation: a one-stage pipeline and a four-tap FIR filter.

In both circuit mappings, the clusters that were not used for mapping were configured in TMR mode so that the output of the cell of interest was delivered to the array output.

1) One-Stage Pipeline: A One-stage pipeline can be implemented using one cell only in SMM mode, and therefore it is

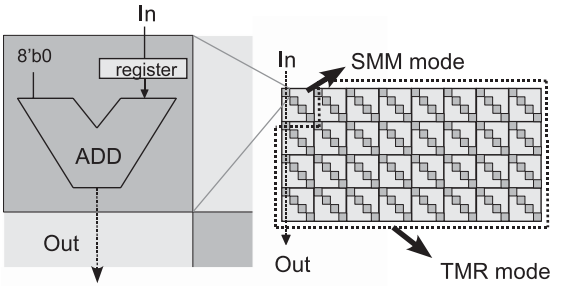


Fig. 20. Technological mapping of a one-stage pipeline.

TABLE VIII
MTTF FOR DIFFERENT MAPPING CONFIGURATIONS OF FOUR-TAP FIR

TMRed gp.	*#T.B.	#S.B.	Trials	Meas. MTTF	No. of Clusters
None	1701	230	231	6.17 s	3
S	3969	124	232	9.82 s	8
SA	5670	53	230	20.1 s	9
SAR	9072	0	1	$\geq$ 80 min	12

*Total number of configuration bits.

suitable for evaluating MTTF for a circuit with few sensitive bits. Fig. 20 illustrates one of the possible mappings of a one-stage pipeline using one cell in one cluster. The number of sensitive bits derived using a simulation is 19, whereas the number of total configuration bits in that cell is equal to 189. The measured MTTF from the experiment was 52.5 s over 84 trials of 73-min duration at a clock frequency of 4 MHz.

2) Four-Tap FIR Filter: In order to evaluate the tradeoff between the MTTF and area, a 4four-tap FIR filter on the fabricated chip was selected for demonstration purposes and not for maximizing the usage of the chip. Fig. 12 illustrates one of the possible implementations of the FIR filter with three main groups of elements distinguished: registers (R), shifters (S), and adders (A), where the main element of each group was represented by registers, shifters, and adders, respectively. To shorten the critical path, several registers were added to the data paths. The constants of the mapped FIR filter were all identical and equal to 0.25, which was represented by a 2-bit shift to the right. The number of sensitive bits for each cell was evaluated using a simulation.

The number of sensitive bits for cells that are used for both functionality and interconnect (20–33) is much larger than those of the cells only used for interconnect (at most 10). On the other hand, the variation in the number of sensitive bits used for functionality and interconnect is not so large. For example, in the adders group, the number of sensitive bits in each cell ranges from 20 to 26, and from 24 to 29 in the shifters group, and, finally, in the register group, it ranges from 20 to 33. In contrast, the total number of configuration bits in an ALU cluster is 63. Note that the number of sensitive bits is smaller than the number of configuration bits because some configuration bits are "don't cares" in their mappings. We thus regarded that the variation in the number of sensitive bits in each group is small, and we assigned the operation modes group by group rather than cluster by cluster for simplicity.

The experimental results are shown in Table VIII. In the table, the notation SAR implies that shifters, adders, and registers were TMRed, while SA means that only shifters and adders were TMRed, and the remainder was implemented in SMM mode. The table also shows that the area varies from 3 to 12 clusters depending on the selection of TMRed groups. The output of the implementation with full TMR remained correct for over 80 min, which means that the MTTF is longer than 80 min.[1] These results show that TMR offers high reliability. Fig. 21(a) and (b), respectively, depict examples of the cell utilization of S- and SA-TMRed mapping cases on a 4×3 cluster array. The number of cells used for data processing in S and SA are 17 and 23, respectively, whereas the number of clusters are 8 and 9, respectively.

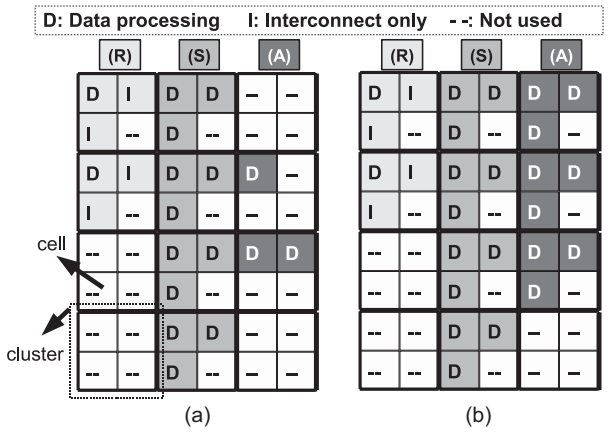


Fig. 21. Cell utilization of A- and SA-TMRed mapping cases. (a) S-TMRed mapping. (b) SA-TMRed mapping.

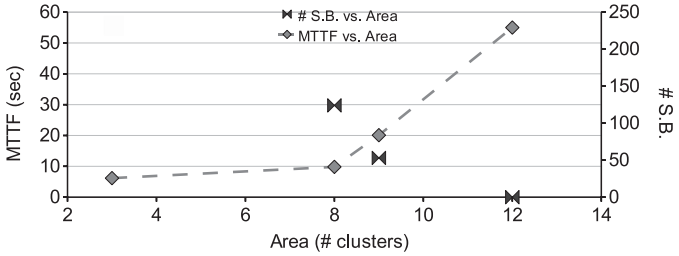


Fig. 22. Area versus number of sensitive bits and MTTF.

The plot of the area versus MTTF and the number of sensitive bits, illustrated in Fig. 22, demonstrates that the tradeoff between area and reliability can be exploited in our architecture as expected. Furthermore, the plot of the number of sensitive bits and MTTF, illustrated in Fig. 23, shows that MTTF increases as the number of sensitive bits decreases. The correlation between MTTF and the number of sensitivity bits is discussed in the next section.

D. Discussion

The experiments presented in Sections V-C.1 and V-C.2 were accelerated by using an alpha particle source, and the MTTF in this harsh environment was measured. On the other hand, it is necessary to evaluate MTTF under actual conditions in space or any other environment where some level of

[1]We stopped the radiation after 80 min. in order to avoid a significant total dose effect [26].

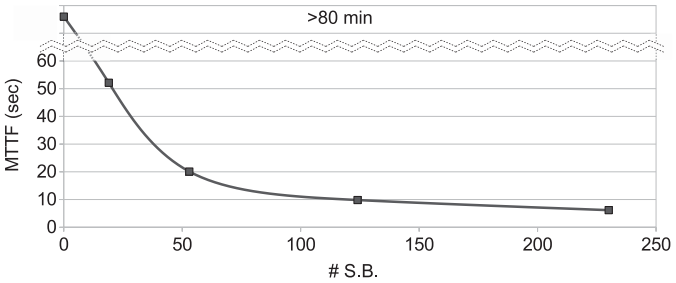


Fig. 23. Sensitive bits (no. of S.B.) versus MTTF.

TABLE IX
MTTF ESTIMATION IN TERRESTRIAL ENVIRONMENT

Application	No. of S.B.	MTTF (years)
FIR-none	230	154
FIR-S	124	245
FIR-SA	53	502
One-stage pipeline	19	1312
FIR-SAR	0	$\geq 1.2 \times 10^5$

reliability is required. As an example, we chose to evaluate MTTF in the terrestrial environment [27], [28].

We assume standard packaging whose emission rate of alpha particles is 10 $cm^{-2}h^{-1}$. In this case, a 1-h test conducted under accelerated conditions using Am-241 foil with an emission flux equal to 9×10^9 $cm^{-2}h^{-1}$ is equivalent to 9×10^4 years in the actual environment. Table IX lists the estimated MTTF values in the terrestrial environment for a one-stage pipeline and four-tap FIR filter circuits.

The MTTF values shown in Table IX were estimated as a function of the number of sensitive bits by using regression analysis. The resulting equation is

$$\text{MTTF} = \frac{M}{\#S.B.} + C \tag{1}$$

where #S.B. is the number of estimated sensitive bits for the mapped application, MTTF is the mean time to failure in years, M is equal to 2.397×10^4, and C is equal to 51. The mean sum of square error of (1) is as low as 0.71, and the coefficient of determination is as high as 99%, which means that the number of sensitive bits explains 99% of the MTTF value, while the remaining 1% is explained by other factors such as SEUs and SETs in the data path. These results support our hypothesis that the vulnerability of the circuit can be characterized mostly by the number of sensitive bits.

An important observation here is that, in comparing (1) and the theoretical equation MTTF $= 1/\lambda N$ [29], λN is proportional to the number of sensitive bits, where λ is a constant hazard rate, and N is the number of memory bits. This means that the MTTF, hazard rate, and the number of memory bits of the application mapped on the proposed architecture are well characterized with the number of sensitive bits.

Furthermore, the FIT rate, which is another popular metric for characterizing device reliability and is defined as the number of errors per 10^9 h [30], can be estimated from the derived MTTF using a simple transformation, i.e., FIT $= 10^9$/MTTF.

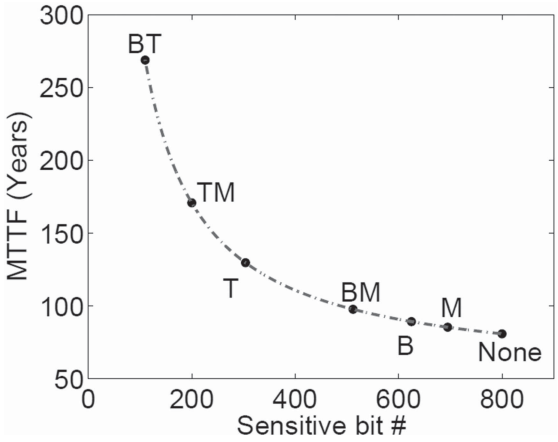


Fig. 24. Estimation of MTTF for different mappings of a Viterbi decoder with different numbers of sensitive bits.

As an example, we estimated the MTTF for different mappings of Viterbi decoders in Section IV-A using (1); the results are shown in Fig. 24. The dashed line based on (1) represents the achievable tradeoff through mapping with different reliability configurations.

The MTTF shown in Table IX and Fig. 24 may seem larger than expected, for which there are various explanations. This evaluation assumes a terrestrial (test) environment. On the other hand, at airplane altitudes, the flux of particles is 1000 times that of the incident flux, and in space it is much higher [28] because of the different types of SEU-inducing particles that exist there. Another reason is that the number of configuration bits is considerably small especially compared with that of FPGAs, which means coarse-grained reconfigurable devices are inherently more robust to soft errors than FPGAs. Moreover, the mapped logic was relatively small in this experiment. A circuit that is 100 times larger will have a corresponding 100 times decrease in MTTF, as indicated by the estimated equation.

VI. CONCLUSION

We proposed a coarse-grained dynamically reconfigurable architecture in which four operation modes with different reliability levels and area efficiencies could be selected for each cluster. TMR, DMR, and SMS modes provide error correction in the configuration information. In data paths, TMR mode offers error correction, while DMR offers error detection. SMM mode has no reliability consideration, and is selected for achieving the highest performance per area. The evaluation of the aging process shows that the circuit delay degradation can be mitigated by a hot-swapping operation with a power increase of 2%. The area overhead to attain considerable mitigation effect and provide flexible reliability accounts for 29.3% of the proposed coarse-grained dynamically reconfigurable device. Fault-tolerance evaluation based on sensitive bits of a Viterbi decoder suggests that the variation in the number of sensitive bits in each cluster could be utilized to improve the tradeoff between reliability and area overhead.

Furthermore, accelerated radiation tests demonstrated that our coarse-grained dynamically reconfigurable architecture

with flexible reliability could trade off soft error immunity and the area of implementation, as expected. In addition, the MTTF and FIT rates of the mapped applications were characterized using the number of sensitive bits, and models for estimating both metrics were constructed.

ACKNOWLEDGMENT

The authors would like to thank the project members of JST CREST of Kyoto University, the Kyoto Institute of Technology, Sakyo-Ku, Kyoto, the Nara Institute of Science and Technology, Nara, Japan, and ASTEM RI for their helpful discussions.

REFERENCES

[1] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2006, pp. 182–192.

[2] W. Wang, V. Reddy, A. T. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "Compact modeling and simulation of circuit reliability for 65 nm CMOS technology," *IEEE Trans. Device Mater. Rel.*, vol. 7, no. 4, pp. 509–517, Dec. 2007.

[3] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," in *Proc. Asia South Pacif. Des. Autom. Conf.*, Jan. 2006, pp. 1047–1052.

[4] E. Stott, J. S. J. Wong, and P. Y. K. Cheung, "Degradation analysis and mitigation in FPGAs," in *Proc. Field Programm. Logic Appl. Conf.*, Sep. 2010, pp. 428–433.

[5] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. IEEE Very Large Scale Integr. Test Symp. Conf.*, Apr. 1999, pp. 86–94.

[6] L. Anghel, D. Alexandrescu, and M. Nicolaidis, "Evaluation of a soft error tolerance technique based on time and/or space redundancy," in *Proc. Symp. Integr. Circuits Syst. Des. Conf.*, Sep. 2000, pp. 237–242.

[7] A. D. Houghton, *The Engineer'S Error Coding Handbook*. London, U.K.: Chapman & Hall, 1997.

[8] "Single-event effect mitigation in RTAX-DSP spaceflight FPGAs," *Actel*. (2007) [Online]. Available: http://www.actel.com/documents/RTAXDSP_SEE_WP.pdf

[9] E. Stott, P. Sedcole, and P. Cheung, "Fault tolerant methods for reliability in FPGAs," in *Proc. Field Programm. Logic Appl. Conf.*, Sep. 2008, pp. 415–420.

[10] F. Lima, L. Carro, and R. Reis, "Designing fault tolerant systems into SRAM-based FPGAs," in *Proc. Asia South Pacif. Des. Autom. Conf.*, Jan. 2003, pp. 650–655.

[11] E. Fuller, M. Caffrey, A. Salazar, C. Carmichael, and J. Fabula, "Radiation testing update, SEU mitigation, and availability analysis of the virtex fpga for space reconfigurable computing," in *Proc. Milit. Aerosp. Programm. Logic Devices Conf.*, Sep. 2001, pp. 1–8.

[12] C. Carmichael, E. Fuller, J. Fabula, and F. Lima, "Proton testing of SEU mitigation methods for the virtex FPGA," in *Proc. Milit. Aerosp. Appl. Programm. Logic Devices Conf.*, Sep. 2001, pp. 1–7.

[13] R. Hartenstein, "Coarse grain reconfigurable architectures," in *Proc. Asia South Pacif. Des. Autom. Conf.*, Feb. 2001, pp. 564–569.

[14] S. M. A. H. Jafri, S. J. Piestrak, O. Sentieys, and S. Pillement, "Design of a fault-tolerant coarse-grained reconfigurable architecture: A case study," in *Proc. Int. Symp. Qual. Electron. Des. Conf.*, Mar. 2010, pp. 845–852.

[15] M. M. Azeem, S. J. Piestrak, O. Sentieys, and S. Pillement, "Error recovery technique for coarse-grained reconfigurable architectures," in *Proc. 14th IEEE Symp. Des. Diagnost. Electron. Circuits Syst.*, Apr. 2011, pp. 441–446.

[16] G. Lee and K. Choi, "Thermal-aware fault-tolerant system design with coarse-grained reconfigurable array architecture," in *Proc. NASA/ESA Conf. Adaptiv. Hardware Syst.*, Jun. 2010, pp. 265–272.

[17] D. Wick, S. Riley, and D. Lupia, "Radiation hardened field programmable object array (FPOA) for space processing," *Military and Aerospace FPGA and Applications*. (2007) [Online]. Available: http://nepp.nasa.gov/mafa/talks/MAFA07_12_Lupia.pdf

[18] D. Alnajjar, Y. Ko, T. Imagawa, H. Konoura, M. Hiromoto, Y. Mitsuyama, M. Hashimoto, H. Ochi, and T. Onoye, "Coarse-grained dynamically reconfigurable architecture with flexible reliability," in *Proc. Field Programm. Logic Appl. Conf.*, Sep. 2009, pp. 186–192.

[19] K. Nakahara, S. Kouyama, T. Izumi, H. Ochi, and Y. Nakamura, "Fault tolerant dynamic reconfigurable device based on EDAC with rollback," *IEICE Trans. Fundamen.*, vol. 89, no. 12, pp. 3652–3658, Dec. 2006.

[20] Y. Mitsuyama, K. Takahashi, R. Imai, M. Hashimoto, T. Onoye, and I. Shirakawa, "Area-efficient reconfigurable architecture for media processing," *IEICE Trans. Fundamen.*, vol. 91, no. 12, pp. 3651–3662, Dec. 2008.

[21] B. Pratt, M. Caffrey, P. Graham, K. Morgan, and M. J. Wirthlin, "Improving FPGA design robustness with partial TMR," in *Proc. IEEE Int. Rel. Phys. Symp. Conf.*, Mar. 2006, pp. 226–232.

[22] T. Imagawa, M. Hiromoto, H. Ochi, and T. Sato, "A routing architecture exploration for coarse-grained reconfigurable architecture with automated SEU-tolerance evaluation," in *Proc. IEEE Int. Conf.*, Sep. 2010, pp. 248–253.

[23] F. L. Kastensmidt, L. Carro, and R. Reis, *Fault-Tolerance Techniques for SRAM-Based FPGAs*. Berlin, Germany: Springer-Verlag, 2006.

[24] "Measurement and reporting of alpha particles and terrestrial cosmic ray-induced soft-errors in semiconductor devices." *JEDEC Standard JESD89*. (2009) [Online]. Available: http://www.jedec.org/sites/default/files/docs/jesd89a.pdf

[25] T. Karnik, P. Hazucha, and J. Patel, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Trans. Depend. Secure Comput.*, vol. 1, no. 2, pp. 128–143, Apr. 2004.

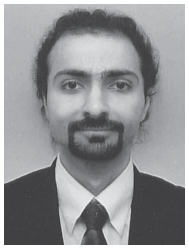
[26] K. Iniewski, *Radiation Effect in Semiconductors*. Boca Raton, FL: CRC Press, 2010.

[27] H. Kobayashi, N. Kawamoto, J. Kase, and K. Shiraishi, "Alpha particle and neutron induced soft error rates and scaling trends in SRAM," in *Proc. IEEE Int. Rel. Phys. Symp. Conf.*, Apr. 2009, pp. 206–211.

[28] H. Zhu, "Logic SER characterization," in *Proc. IEEE Int. Rel. Phys. Symp. Conf.*, Apr. 2008, pp. 1–10.

[29] B. S. Blanchard, "Logistic Engineering and Management," Englewood Cliffs, NJ: Prentice Hall, 1998.

[30] S. Mukherjee, *Architecture Design for Soft Errors*. San Mateo, CA: Morgan Kaufmann, 2008.



Dawood Alnajjar (S'08) received the B.E. degree from the University of Jordan, Amman, Jordan, in 2006, and the M.E. degree in information systems engineering from Osaka University, Osaka, Japan, in 2010, where he is currently pursuing the Ph.D. degree with the Department of Information Systems Engineering.

His current research interests include high-performance reconfigurable computing, computer architectures, and reliable systems.

Hiroaki Konoura (S'09) received the B.E. and M.E. degrees in information systems engineering from Osaka University, Osaka, Japan, in 2009 and 2011, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information Systems Engineering.

His current research interests include mitigation and evaluation of circuit degradation.

Mr. Konoura is a Student Member of the Institute of Electronics, Information and Communication Engineers.

Younghun Ko received the B.E. degree from Nagasaki University, Nagasaki, Japan, and the M.E. degree from the Department of Information Systems Engineering, Osaka University, Osaka, Japan, in 2007 and 2009, respectively.

He is currently with the Department of R&D, Sometech Inc., Seoul, Korea. His current research interests include reconfigurable computing and reliable systems.

Yukio Mitsuyama (S'97–M'02) received the B.E., M.E., and Ph.D. degrees in information systems engineering from Osaka University, Osaka, Japan, in 1998, 2000, and 2010, respectively.

He was an Assistant Professor with the Graduate School of Engineering, Osaka University. Since 2011, he has been an Assistant Professor with the School of Engineering, Kochi University of Technology, Kami-shi, Japan. His current research interests include reconfigurable architecture and its VLSI design.

Dr. Mitsuyama was a recipient of the Best Paper Award at IEEE ISCE 2004. He is a member of the Institute of Electronics, Information and Communication Engineers and the IPSJ.

Masanori Hashimoto (S'00–A'01–M'03–SM'11) received the B.E., M.E., and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively.

He has been an Associate Professor with the Department of Information Systems Engineering, Osaka University, Osaka, Japan, since 2004. His current research interests include computer-aided-design for digital integrated circuits and high-speed and low-power circuit design.

Dr. Hashimoto was a recipient of the Best Paper Award at ASP-DAC 2004. He is a member of the Institute of Electronics, Information and Communication Engineers and the IPSJ. He was on the technical program committees of international conferences including DAC, ICCAD, ASP-DAC, DATE, ICCD, and ISQED.

Takao Onoye (S'93–M'95–SM'07) received the B.E. and M.E. degrees in electronic engineering and the Dr.Eng. degree in information systems engineering from Osaka University, Osaka, Japan, in 1991, 1993, and 1997, respectively.

He was an Associate Professor with the Department of Communications and Computer Engineering, Kyoto University, Kyoto, Japan. Since 2003, he has been a Professor with the Department of Information Systems Engineering, Osaka University. He has authored or co-authored over 200 research papers in reputed journals and proceedings of international conferences on VLSI design and multimedia signal processing. His current research interests include media-centric low-power architecture and its system-on-a-chip implementation.

Dr. Onoye was a member of the CAS Society Board of Governors from 2008 to 2010. He is a member of the Institute of Electronics, Information and Communication Engineers, the IPSJ, and the ITE-J.