PAPER Special Section on Solid-State Circuit Design—Architecture, Circuit, Device and Design Methodology

A Cost-Effective Selective TMR for Coarse-Grained Reconfigurable Architectures Based on DFG-Level Vulnerability Analysis

Takashi IMAGAWA[†a)], ***Nonmember***, **Hiroshi TSUTSUI**[†b)], **Hiroyuki OCHI**[†c)], ***and*** **Takashi SATO**[†d)], ***Members***

SUMMARY This paper proposes a novel method to determine a priority for applying selective triple modular redundancy (selective TMR) against single event upset (SEU) to achieve cost-effective reliable implementation of application circuits onto coarse-grained reconfigurable architectures (CGRAs). The priority is determined by an estimation of the vulnerability of each node in the data flow graph (DFG) of the application circuit. The estimation is based on a weighted sum of the node parameters which characterize impact of the SEU in the node on the output data. This method does not require time-consuming placement-and-routing processes, as well as extensive fault simulations for various triplicating patterns, which allows us to identify the set of nodes to be triplicated for minimizing the vulnerability under given area constraint at the early stage of design flow. Therefore, the proposed method enables us efficient design space exploration of reliability-oriented CGRAs and their applications.

key words: *soft error, single event upset, triple modular redundancy, reliability, simulated annealing*

1. Introduction

As CMOS process technologies enter into the range of a few tens of nanometers, various phenomena that disturb the normal operation of LSI systems have become prominent. In particular, soft-errors, such as a single-event upset (SEU), have received increasing attention in the recent years. SEU has been a major cause of problems in satellite systems [1] and avionics systems [2]. Its impact is expected to become even larger in scaled devices. In the near future, the consideration of soft-error vulnerability will become a common practice even for consumer-oriented system designs, where the trade-off between cost (e.g., chip area, power consumption) and quality (e.g., performance, reliability) should be always considered. To design a soft-error-tolerant system cost-effectively, the impacts of soft-error should be quantitatively evaluated. For example, in the case of video streaming systems, minor error, such as a temporal bit-flip of a pixel in a frame, may be acceptable. Rather, we should allocate extra hardware resources to protect more critical modules, with which we can avoid catastrophic damage over multiple frames.

As discussed in our previous work [3], coarse-grained reconfigurable architectures (CGRAs) are suitable for reliability-oriented LSI systems on account of their reconfigurability and granularity. The reconfigurability reduces non-recurring engineering cost (NRE). A reconfigurable devices can implement various target applications with various constraints including reliability without manufacturing specific chips. In contrast, in case of ASICs, individual chips have to be manufactured for each reliability requirements. Therefore, the reconfigurable devices are superior to ASICs in terms of the cost to satisfy various reliability constraints. Moreover, reconfigurability also extends the life time of LSI systems because we can avoid using the faulty units by reconfiguration. CGRAs are advantageous to their fine-grained counterparts, such as field-programmable gate arrays (FPGAs), in terms of performance and energy efficiency. More noteworthy is the fact that CGRAs have a much smaller amount of configuration memory than FPGAs, and this can reduce the incidence of soft-errors. For example, according to our preliminary experiments, the amount of the configuration information to implement a 1024-point FFT is 312,545 bit when the circuit is synthesized with only 4-input LUTs on an FPGA. In contrast, the amount is 10,458 bit when the circuit is synthesized with only ALUs on a CGRA. Then, the mean time to failure (MTTF) of the circuit composed of ALUs is 29.89 times longer than the circuit composed of LUTs, so that, the CGRA circuit is more reliable than the LUT circuit. The calculation method of MTTF based on the amount of the configuration information is described in [4]. Our final goal is to establish a design methodology for reliability-oriented LSI systems with CGRAs.

For SEU mitigation, triple modular redundancy (TMR) is widely accepted in mission-critical applications. Recently selective triple modular redundancy (selective TMR) is attracting attentions in the areas where both reliability and cost should be considered [5], [6]. In selective TMR, some components in a given circuit are selectively triplicated when a design constraint (e.g., area) does not allow triplicating all the components. On the other hand, when we implement a smaller scale application than the chip size of the target CGRA, we can improve the reliability of the target application as an added value with assigning the remaining design margin for triplication with a low cost. This is expected to be a new superiority in constructing the consumer-oriented systems with the CGRAs to the conventional ASICs. Recently, [7] proposed a unique reliability-oriented CGRA in order to implement these concepts of se-

Manuscript received August 6, 2012.
Manuscript revised November 3, 2012.

[†] The authors are with the Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: imagawa@easter.kuee.kyoto-u.ac.jp
b) E-mail: tsutsui@easter.kuee.kyoto-u.ac.jp
c) E-mail: ochi@kuee.kyoto-u.ac.jp
d) E-mail: takashi@i.kyoto-u.ac.jp

DOI: 10.1587/transele.E96.C.454

lective TMR. This CGRA enables us to triplicate a part of the target circuit easily, but it is still difficult to find which parts should be triplicated for making the best of the design margin.

The components to be triplicated must be carefully determined because the impact observed at the output data greatly depends on where the SEU occurs. We must note that error observed at the output data can take quite different values depending on the processing node where the SEU occurs. As a motivative example, the output images of an edge detection filter implemented on a CGRA are shown in Fig. 1. The left-most image in Fig. 1 is an error-free output, while the other three are erroneous images each of which is obtained by injecting just one SEU to a processing node in the CGRA. As seen from these images, an SEU results in vastly different results — from an output image that is almost indistinguishable from the error-free one, to an output image that is completely different. Therefore, within the given cost constraints, it is important to figure out the components that has greater impact on output stream and triplicate them for minimizing the vulnerability. Although it is important to determine the priority for triplication, it has not been well investigated for CGRAs. In case of FPGAs, an analysis in [5] gives evidence that SEUs in a feedback section cause persistent and unrecoverable upset in the output stream while SEUs in other components cause only temporal errors.

To find an exactly optimal solution for which nodes in a DFG we should triplicate, the exhaustive search over possible SEUs and input data is necessary. However, such an exploration is impractically expensive because they require long simulation time. That is particularly true when designing a CGRA architecture and implementing the target applications on it, since these time-consuming simulations and evaluations are heavily repeated.

This paper proposes a method to determine a priority for applying selective TMR, which achieves cost-effective reliable implementation of an application circuit to a CGRA. The priority is determined by an estimation of the vulnerability of each node in the data flow graph (DFG) of the application circuit. The estimation is based on a weighted sum of the features and parameters of each node in the DFG which characterize impact of the SEU in the node to the output data. This method does not require time-consuming placement-and-routing processes, as well as extensive fault simulations for various triplicating patterns, which allows us to identify the set of nodes to be triplicated for minimizing the vulnerability under given area constraint at the early stage of design flow. Therefore, the proposed method enables us efficient design space exploration of reliability-oriented CGRAs and their applications.

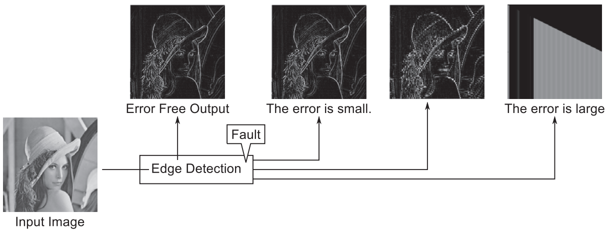


Fig. 1 Different impacts on output image of an edge detection filter caused by a single SEU fault.

The remainder of this paper is organized as follows. Section 2 introduces the target CGRA considered in this paper. Section 3 describes the overview of our framework for designing reliability-oriented LSI systems with CGRAs. Section 4 describes the proposed method. Section 5 demonstrates the effectiveness of our proposed method. Finally, Sect. 6 concludes this paper.

2. Target CGRA

This section describes the target reliability-oriented CGRA [7] considered in this paper. Figure 2 provides an overview of the CGRA. It has a cluster array architecture designed to achieve various levels of reliability. It is a two-dimensional array of clusters, each of which consists of four cells that have an execution module (EM), configuration memories (CFG), voting circuits (VCs), and a configuration switch matrix (CFG-SM).

To realize flexible reliability, the CGRA also introduces a redundancy control unit (RDU) and a comparing-and-voting unit (CVU). The cluster has four operation modes, each with a different redundancy. In this paper TMR mode and single modular with multi-context (SMM) mode are considered. These modes are explained as follows. The details of the other two modes are omitted because these are not used in this paper.

TMR mode In TMR mode, three CFGs in a cell holds an identical configuration, and an SEU occurring in the CFG will be repaired when the next clock is given to the CFGs, since values in these CFGs are voted by VCs and the voted values are rewritten to the CFGs in every clock cycle. In addition, the outputs of three cells are also voted by the CVU in TMR mode so that an SEU at the datapath are also repaired. Thus, both CFG and datapath are protected in TMR mode. Although error correction code (ECC) is widely used for SEU protection

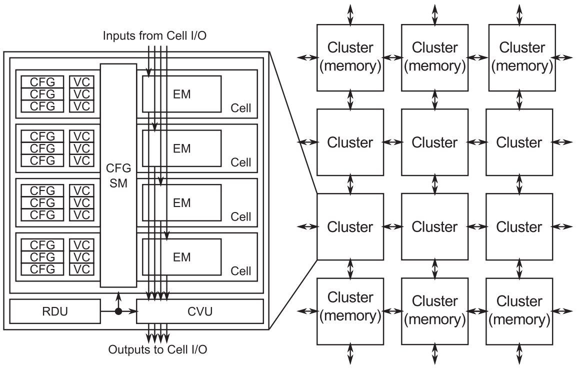


Fig. 2 The target reliability-oriented CGRA.

in SRAMs, TMR results better area-efficiency for protecting SEU in configuration memory in reconfigurable devices [4]; SRAM requires only one ECC encoder/decoder for each port, while configuration memory in reconfigurable devices require ECC encoder/decoder for each configuration word, since all configuration data are referred to every clock cycle.

SMM mode In SMM mode, four cells in a cluster operate independently, and there is no redundancy in either configuration memory or datapath. It is noteworthy that this CGRA supports selective TMR since the operation mode of every cluster can be selected independently, which offers an area-reliability trade-off, i.e., we can increase reliability by selecting more clusters to be in reliable modes at the expense of area usage.

As shown in Fig. 2, on top and bottom edges of the array, there are specialized clusters, which can be used like memories to communicate with external systems. This type of cluster has memory elements instead of the EMs. This paper excludes the memory elements from the vulnerability evaluation because the memory elements which have the similar structure to SRAM are expected to be protected with other techniques such as ECC.

3. Overview of Our Framework

Our framework explores the design space for reliability-oriented CGRAs and applications. This framework has the following features.

- For a design entry, register-transfer-level (RTL) description is allowed, and configuration bitstream of the partly triplicated circuit is generated as a result.
- Time-consuming fault simulation is not necessary to evaluate the vulnerability of each component in the application circuit.

The following is the design flow of our framework to generate a cost-effective reliability-oriented implementation of a given circuit for the target CGRA (Fig. 3).

1. A DFG for the given RTL descriptions of an application circuits is generated, and technology mapping for the target CGRA is applied.
2. The components which are to be triplicated are determined by using the proposed evaluation function described in Sect. 4.
3. The placement and routing tool generates the configuration bitstream to implement the given application circuit onto the target CGRA.
4. (Optional) The simulation scripts to verify the functionality of the target CGRA circuit are generated automatically with the scripts for the RTL descriptions.
5. (Optional) The fault injection scripts are also generated automatically to evaluate the vulnerability of the target circuit and the cost-effectiveness of the applied selective TMR.

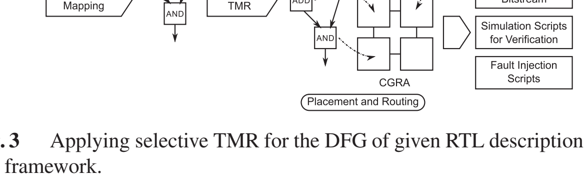


Fig. 3 Applying selective TMR for the DFG of given RTL description in our framework.

4. Estimation of Priority for Selective TMR

In this section, we propose an evaluation function for estimating priority for TMR. The function estimates the vulnerability of each node in a DFG of application circuit. In this context, a "vulnerable node" means that SEUs in the component which implements the function of this node tend to cause a serious damage observable at the output data.

We use the following function f to evaluate the vulnerability of node n,

$$f(n) = \sum_i w_i a_i(n), \tag{1}$$

where i represents a type of key features or parameters of node, whose value at node n is represented by $a_i(n)$. The key features or parameters are expected to indicate the node vulnerability, such as operation type executed in the node, the node distance to primary output nodes, and so forth. The weight of i is represented by w_i. In this paper, $f(n)$, $a_i(n)$, and w_i are called *evaluation function*, *terms* and *coefficients* of evaluation function, respectively.

The following subsections discuss two important points: what kind of features or parameters should be considered as i and how to determine its appropriate weight w_i.

4.1 Terms of the Evaluation Function

The nodes in a DFG can be characterized in various aspects and some of them have a strong relationship to the vulnerability. From our preliminary experiments, we found that some of following values have strong relationship to the vulnerability.

1. Operations of the nodes (Boolean)
   We introduce terms which indicates the operation of the node. The number of these terms is equal to the number of possible operations of the processing element. For example, when the operation of node is OR, the corresponding term takes 1 and the other terms take 0.
2. Utilization of register (Boolean)
   We introduce one term for each node, in order to indicate that the register in the processing elements for the node are enabled or not.
3. Distance from primary input/output (floating-point number)

We introduce two terms for each node, in order to indicate the shortest distance from primary input and primary output, respectively. The value is normalized by dividing with the maximum distance in the DFG, so that the value falls into the range from 0.0 to 1.0.

4. Closeness to primary input/output (floating-point number)
   In addition to distance, we also introduce two terms that represent closeness to primary input and primary output. These terms are defined to be reciprocal of the respective terms in distance. If the value of the term explained in 3. is d, the value of closeness takes $1/d$. This is intended to emphasize the nodes that are very close to the edge of the DFG, while 3. indicates the distance from the edge of the DFG linearly.
5. Output cone group (Boolean)
   We define four categories for outputs, memory write data, memory address, memory write enable, and external done signal, which will be explained later. We introduce four terms for each node, in order to indicate that the transitive fanout of the node is the primary output of one of above categories.

These terms are normalized to take values from 0.0 to 1.0 for simplicity of implementation, so that this also enables us to compare the weights w_i each other directly to know the important terms from the viewpoint of impact for vulnerability.

In the following, we explain how the above terms are related to the vulnerability of the node.

4.1.1 Operations of the Nodes

Generally, the logical operations, such as OR operation, often mask the influence of the erroneous inputs. For example, assuming that an input of a 2-input AND gate is given value 0 with the probability 50%, the erroneous value of the other input will be masked at 50% probability. These are not negligible values especially when many these operations are chained in large-scale circuits. On the other hand, ADD operation never output the correct value in the same case. For example, the done signal generated with logical operations mainly is more robust than others such as write data generated with arithmetic operations mainly.

4.1.2 Utilization of Register

When the incorrect result of an operation is stored in a register in the processing element, it is expected that the error has a persistent impact for several clock cycles. So the utilization of register has an impact on whether the error in the node is temporal or persistent.

4.1.3 Distance/Closeness from/to Primary Input/Output

The nodes near the primary output nodes are considered as vulnerable because the faults in such nodes are rarely masked logically. On the other hand, the errors generated in the nodes near the primary inputs are sometimes masked logically when the erroneous data pass through logical operations. However, such errors can affect a wide portion of the DFG. Therefore, the relationship between the vulnerability and the distance is so complex that the relationship is worth while evaluating including its importance.

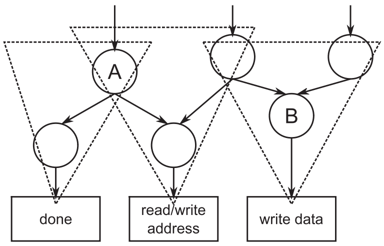


Fig. 4 Each node in a DFG can be categorized based on the output cones.

4.1.4 Output Cone Group

Besides these popular features, this paper focuses on a novel feature of nodes in DFG taking into account that the target framework is based on CGRAs. As described in Sect. 3, in the target framework, the operand and processed data is exchanged with the external system through the memory element. Therefore the primary output of the target application can be categorized into the following groups: write data, read/write address and write enable signal for memory elements, and a done signal for the external system. Therefore, each node can be characterized with above categories of its transitive fanouts. As Fig. 4 illustrates, each node in a DFG belongs to the grouping cones whose tops are the primary output (or the memory port) nodes. In the case of Fig. 4, the node A belongs to two groups, the done signal cone and the read/write address cone, while the node B belongs only to the write data cone. Applicability of this feature is not limited to the target CGRA described in Sect. 2, because other general CGRAs also have memory elements as interfaces with external systems. Moreover, even in the case of FPGAs, the proposed evaluation function is applicable, if the primary outputs can be categorized into the above groups.

It is expected that the nodes belonging to the done signal cone are relatively robust, since the signal is generated by some conditional judgments which are mainly composed of logical functions, and hence high probability of logical masking is expected. If input and output stream has space- or time-correlation which can be often seen in audio and video data, the nodes belonging to the read/write address cone are expected to be relatively robust. When the fault causes only a little error in the read address, the differences of the operand values are also a little. The case of write address is similar.

To show quantitatively that the nodes belonging to address cone are robust, we present a simple example. Assume that a pixel value (pixel0) which is kept in a memory (named mem) at an address (adrs0) is to be copied to another memory at the same address.

- When pixel0 is corrupted by a fault, the value of write data changes randomly to pixel1.
- When adrs0 is corrupted by a fault, the value of read-/write address changes randomly to adrs1 and the write data is mem[adrs1].

The errors in the above two cases are |pixel0 − pixel1| and |pixel0 − mem[adr1]|, respectively. To calculate the averages of errors with faults in data and address, 10,000 above-mentioned trials for every one hundred figures are executed assuming that word length of both address and data of the memory are 8-bit. As a result, the averages of errors with faults in data and address are 96.94 and 39.52, respectively. This means that the faults in the address cone have only 40% impact than the data cone.

4.2 Coefficients of the Evaluation Function

To use the proposed evaluation function, Eq. (1), we need to determine the coefficients w_i. To determine the coefficients w_i, we should define the error metric to measure the impact (or error) observed at the output data. More specifically, for each node, we measure the error which would be observed if the node were damaged. This error indicates the node vulnerability. If the error is small, the node triplication priority should be low; if large, the priority should be high. In this paper, we use mean absolute error (MAE) to measure the error in output data. MAE is so simple that we can apply it to the wide-ranged applications such as the image and audio processing. The definition of MAE is given by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |x_i - \tilde{x}_i|, \tag{2}$$

where N is the number of words in the output data, x_i and $\tilde{x}_i$ are the i-th word of actual output and error-free output, respectively. Note that other error metrics can be used in the same manner. Generally, the persistent effect makes MAE larger than the temporal error. So that, MAE can quantitatively evaluate the persistent effect considered in [5].

To determine the coefficients w_i, we apply a generic simulated annealing (SA) method to some sample circuits to conduct appropriate coefficients w_i as described below.

(1) Finding the ideal priority for sample circuits:
At first, DFGs for the given RTL descriptions of some sample application circuits are generated, and technology mapping for the target CGRA is applied using our framework. For each configuration memory bit of the obtained netlist, we inject an SEU and run simulation to evaluate MAE observed at the output data raised by the SEU. This simulation repeats the following sequence for every error: initializing, injecting an error, providing an input stream, data processing and obtaining an output stream. This evaluation gives the ideal priority for applying selective TMR to the processing elements. For example, assume that there are four processing elements (n_0, n_1, n_2, n_3) in a netlist, and MAE values which are observed when the faults are induced in these processing elements are (e_0, e_1, e_2, e_3), and their magnitude correlation is $e_0 > e_1 > e_2 > e_3$, then the ideal priority is $[n_0, n_1, n_2, n_3]$.

(2) Finding the best/worst case trade-off curve:
By increasing the triplicated nodes one by one according to the above priority, the best case trade-off curve between vulnerability and TMR cost is obtained as described in Fig. 6. Similarly, with increasing the triplicated nodes with reversed order of the priority, the worst case trade-off curve is obtained. These two curves are important for evaluating the quality of the coefficients explored in the following process.

(3) Finding the optimal coefficients w_i of the evaluation function:
To find the generally applicable coefficients w_i, we use a generic SA procedure whose pseudo code is described in Fig. 5. The procedure has four scheduling parameters: “init” is the initial temperature, “end” is the final temperature, “step” is the ratio of temperature decreasing and N is the repeat count in a temperature.

(a) Set 0.5 to all w_i as initial value.
(b) Set “init” to the current temperature.

```
Parameters: init, end, step, $N$

$[\cdots, w_i, \cdots] = [\cdots, 0.5, \cdots]$          $\cdots$ (a)
cur_temp = init ... (b)                                   $\cdots$ (b)
eval = evaluate tradeoff curves                           $\cdots$ (c, d, e)
while cur_temp > end do
    for $n = 0$ to $N$ do
        new $\boldsymbol{w}$ = change one of $w_i$ randomly     $\cdots$ (f)
        new_eval = evaluate tradeoff curves               $\cdots$ (g)
        if accept new $\boldsymbol{w}$ then                       $\cdots$ (h)
            $\boldsymbol{w}$ = new $\boldsymbol{w}$
            eval = new eval
        end if
    end for
    cur_temp $*$ = step                                     $\cdots$ (j)
end while
```

Fig. 5 The pseudo code of the SA procedure.

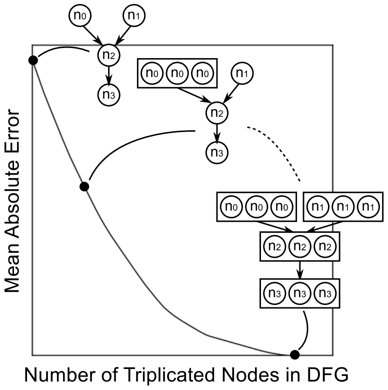


Fig. 6 Incremental selective TMR in order of error for each node.

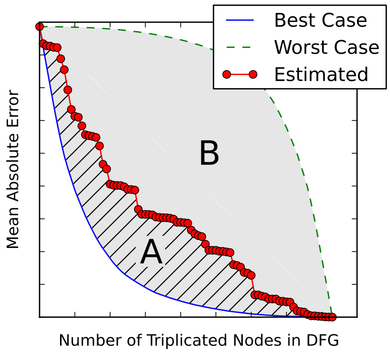


Fig. 7 Quality of trade-off curve is evaluated based on graph areas.

(c) Using the evaluation function, Eq. (1), with the set of w_i, derive the priority of triplication for the nodes.

(d) Using the determined priority, draw a trade-off curve on a similar way as the best case trade-off curve as described in Step (2).

(e) Evaluate the quality of the estimated priority with the ratio of the area enclosed by the estimated and best case curves (A, the hatched portion of Fig. 7) to the one enclosed by the best case and the worst case curves (B, the hatched or shaded portion of Fig. 7). The ratio A/B becomes smaller as the estimated curve is closer to the best curve.

(f) Randomly alter one coefficient.

(g) Re-evaluate the quality of the coefficients with Step (c) to (e)

(h) Accept the change applied at the Step (f) based on the improvement of the ratio A/B and temperature of SA, and reject otherwise.

(i) Repeat Step (f) to (h) processes N times.

(j) Decrease the current temperature with “step”.

(k) Go back to Step (f) while the current temperature is larger than “end”.

Using the above procedure, the coefficients w_i of the evaluation function is obtained. Once we obtain the coefficients w_i, we can find the priority of triplication for other application circuits without time-consuming fault simulation and SA procedure as outlined in Sect. 3. The result of applying the set of w_i optimized with some sample application circuits to other circuits is shown in Sect. 5.3.

More noteworthy, the magnitude of the coefficient w_i suggests the importance of the factor from the vulnerability point of view in designing reliability-oriented LSI architecture. This is one of the main contributions of this paper which has not yet been well investigated.

5. Evaluation

This section demonstrates the effectiveness of the proposed method. The contents of the evaluations are summarized as follows.

Section 5.1 derives set of coefficients for evaluation function using sample applications. The sample applications are a 1024-point FFT and three image filters, that is, a color invert filter, a horizontal-differential filter and an edge detection filter. The lengths of each input stream are 65,536 words for the image filters and 1,024 words for the FFT and the FIR filter. We also confirm that the set of coefficients determined with all the four applications enables us an efficient selective TMR for these applications.

Table 1 Summary of the estimated trade-off curve quality in each subsection.

Application	Sect. 5.1	5.2	5.3	5.4
Color Invert Filter	0.19	0.18	—	0.22
Horizontal-Differential Filter	0.18	0.17	—	0.19
Edge Detection Filter	0.25	0.27	—	0.26
1024-FFT	0.15	0.23	—	0.16
FIR Filter	—	—	0.26	—

Section 5.2 evaluates the dependency on input vectors. Generally, impact of errors at the output data depends on input vectors. So it should be evaluated whether the determined priority is effective for other input vectors.

Section 5.3 evaluates whether the determined priority is effective for the applications which are not used in SA process for determining the coefficients. We estimate the trade-off curve of an FIR filter, as a newcomer application, with the set of coefficients extracted in Sect. 5.1.

Section 5.4 evaluates whether the estimated priority is valid for the post place-and-route circuit on CGRAs. The proposed method does not take into account the impact of the routing element in the CGRA. Then this subsection evaluates the trade-off curves between vulnerability and TMR cost with fault injection simulation for partly triplicated, post place-and-route circuits implemented on the CGRA.

In these subsections, estimated trade-off curves are evaluated in the same way as the case of the SA process described in Sect. 4.2. The results are summarized in Table 1.

Section 5.5 evaluates the proposed method from the perspective of the processing time. This subsection suggests that the proposed method enables us to make use of the chip area margin effectively to improve the reliability of the target circuit at the early stage of design flow in a practical time.

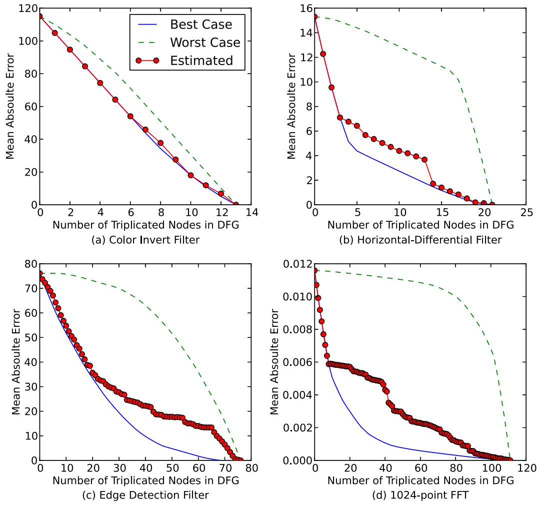
5.1 Deriving Coefficients

The set of coefficients is determined by SA to improve the trade-off curves quality of all the four applications evenly. The scheduling parameters in Fig. 5 are (init, end, step, N) = (100, 0.001, 0.95, 100). All the resulting values of the coefficients are described in Table 2.

Figure 8 shows the estimated trade-off curves for all the applications with the obtained coefficients. The qualities of these curves are evaluated by the ratio of areas enclosed by the estimated, the best case and the worst case curves as explained in Sect. 4.2. If the ratio is zero, the estimated curve matches the best case curve; on the other hand, if the ratio is one, the estimated curve matches the worst case curve.

Table 2 Extracted coefficients by SA.

1. Operation	addition	0.979
	subtract	0.963
	multiplication	0.744
	and	0.055
	or	0.023
	not	0.132
	exor	0.045
	right shift	0.983
	left shift	0.995
	multiplexing	0.066
	less than	0.078
	equal	0.025
	nop	0.697
2. Utilization of registers		0.277
3. Distance	from input	0.366
	from output	0.425
4. Closeness	to input	0.562
	to output	0.227
5. Output cone group	write data	0.288
	read/write address	0.153
	write enable	0.153
	done signal	0.005



(a) Color Invert Filter

(b) Horizontal-Differential Filter

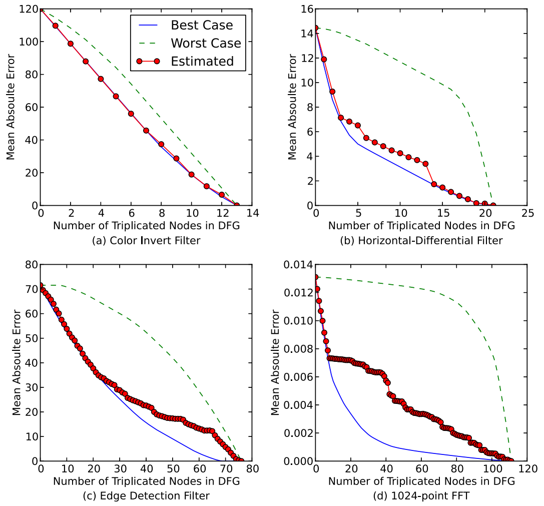
(c) Edge Detection Filter

(d) 1024-point FFT

Fig. 8 Estimated trade-off curves.

The ratios are 0.19 for the color invert filter, 0.18 for the horizontal-differential filter, 0.25 for the edge detection filter and 0.15 for FFT.

The obtained set of coefficients makes all the estimated trade-off curves concave, so it can be said that the set is appropriate. The magnitude correlation and the value of the determined coefficients for output cones is (write data (0.288) > write enable signal (0.153) > done signal (0.013) > read/write address (0.005)). This result suggests that the nodes to calculate the output stream are more sensitive than the nodes to calculate only the other signals. The coefficients for the logical operations tend to be smaller (i.e., less sensitive) than for the arithmetic operations. The most vulnerable arithmetic operation is left shift operation and its coefficient value is 0.995. On the other hand, the values of coefficients for AND and OR operations are as low as 0.055 and 0.023, respectively. These results correspond to the expectations described in Sect. 4.1. Moreover, the extracted coefficients in Table 2 show that the terms of distance from output and closeness to input are important. These results show that the nodes which are near the primary inputs are more vulnerable than the other nodes. These quantitative results have obtained only after these evaluations.



(a) Color Invert Filter

(b) Horizontal-Differential Filter

(c) Edge Detection Filter

(d) 1024-point FFT

Fig. 9 Input vector dependency of estimated order.

There are some points on which the vulnerability drops steeply in the latter of the trade-off curves. This result means that the priority of some vulnerable nodes is estimated low. It suggests that further improvement of the evaluation function is desired.

5.2 Input Vector Dependency

The trade-off curves with different input vectors which are not used in determining the coefficients are evaluated. Figure 9 shows the representative examples for each application. The ratios which indicate the quality of the trade-off curves are 0.18 for the color invert filter, 0.17 for the horizontal-differential filter, 0.27 for the edge detection filter and 0.23 for FFT.

The figure shows that the obtained coefficients are robust for the image filter applications. On the other hand, the quality of FFT's trade-off curve is a little worse, although the shape of the curve is still concave. Especially, when the input vector is artificial (e.g., sine waves or chopping waves), the degradation of quality is notable. These results suggest a necessity for some application-specific customization of the evaluation function.

5.3 Application Dependency

Figure 10 shows the trade-off curves for the FIR filter which is not used in determining the coefficients in Sect. 5.1. The ratio which indicates the quality of the trade-off curves is 0.26. The figure shows that the obtained coefficients are something robust for the newcomer application. The result suggests a necessity for some application-specific cus-

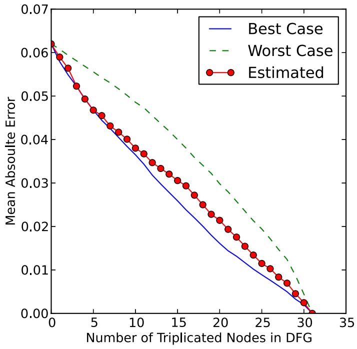


Fig. 10 Estimated trade-off curve of FIR filter.

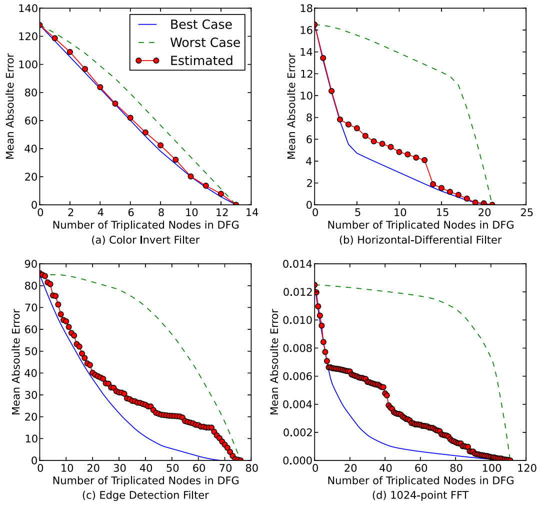


Fig. 11 Trade-off curve of the applications implemented on the CGRA.

tomization of the evaluation function. Regarding the customization, the common features between the FFT and the FIR filter seem promising.

5.4 Vulnerability of Post P&R Circuits

Figure 11 shows the trade-off curves for the target applications implemented on the CGRA after placement and routing. The ratios which indicate the quality of the trade-off curves are 0.22 for the color invert filter, 0.19 for the horizontal-differential filter, 0.26 for the edge detection filter and 0.16 for FFT.

The quality of the trade-off curves entirely becomes a little worse, because the number of vulnerable routing elements increases, when the triplicated nodes disturb the routing with the shortest path. Therefore the evaluation function may be improved by taking into account of the routing efficiency when each triplicating pattern is applied.

However, as Table 1 describes, it is notable that the degradation of the trade-off curves is very small compared with Fig. 8, although the coefficients used in the proposed evaluation function are not derived using any information obtained after placement and routing. Thus, this result suggests that finding an optimal selective TMR at the early stage of the design flow is possible.

5.5 Processing Time

The processing time to find the ideal priority, which corresponds to the Step (1) of the procedure described in Sect. 4.2, is about 7.3 hours for the four sample circuits. In the case of their post P&R circuits, this step required more than 24.0 hours. Therefore, as the results of Sects. 5.2 to 5.4 suggest, once the coefficients of the evaluation function are determined, the proposed method can find a reasonable trade-off point with the required chip area and reliability constraints in a practical time. It is expected that the advantage is more remarkable with more complex applications than the sample applications used in this paper.

6. Conclusion

This paper proposed a method to determine a priority for applying selective TMR to achieve cost-effective reliable implementation of an application circuit to a CGRA. Once the coefficients used in the evaluation function are determined, this method does not require time-consuming processes such as placement-and-routing and extensive fault simulations. The evaluations show that the extracted coefficients are robust for newcomer input vectors and applications. Therefore, the proposed method has generality enough to explore the optimal selective TMR with area constraints for various CGRA systems. This feature allows us to identify the set of nodes to be protected from SEU at the early stage of design flow. This paper also demonstrated the effectiveness and robustness of the proposed method with some sample applications.

A challenging future work is to develop a versatile set of sample applications enough to explore more appropriate terms and coefficients of the evaluation function.

Acknowledgements

This work is part of a JST CREST project and is partly supported by Grain-in-Aid for JSPS Fellows 22·6265, and by VDEC, the University of Tokyo in collaboration with Synopsys, Inc. and Mentor Graphics, Inc.

References

[1] H.C. Koons, J.E. Mazur, R.S. Selesnick, J.B. Blake, J.F. Fennell, J.L. Roeder, and P.C. Anderson, “The impact of the space environment on space systems,” Proc. Spacecraft Charging Conference, pp.7–11, Nov. 1998.

[2] D.C. Matthews and M.J. Dion, “NSEU impact on commercial avionics,” Proc. International Reliability Physics Symposium (IRPS), pp.181–193, April 2009.

[3] T. Imagawa, M. Hiromoto, H. Ochi, and T. Sato, “Reliability evaluation environment for exploring design space of coarse-grained reconfigurable architectures,” IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2524–2532, Dec. 2010.

[4] K. Nakahara, S. Kouyama, T. Izumi, H. Ochi, and Y. Nakamura, “Autonomous repair fault tolerant dynamic reconfigurable device,” IEICE

Trans. Fundamentals, vol.E91-A, no.12, pp.3612–3621, Dec. 2008.

[5] B. Pratt, M. Caffrey, P. Graham, K. Morgan, and M. Wirthlin, "Improving FPGA design robustness with partial TMR," Proc. International Reliability Physics Symposium (IRPS), pp.226–232, March 2006.

[6] X. She and P. Samudrala, "Selective triple modular redundancy for single event upset (SEU) mitigation," Proc. NASA/ESA Conference on Adaptive Hardware and Systems, pp.344–350, 2009.

[7] 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," Proc. International Conference on Field Programmable Logic and Applications (FPL), pp.186–192, Aug. 2009.

Takashi Imagawa received his B.E. degree in Electrical and Electronic Engineering, his master degree in Communications and Computer Engineering, from Kyoto University in 2008 and 2010. Presently, he is a doctor course student at Department of Communications and Computer Engineering, Kyoto University. He is a student member of IPSJ and IEEE.

Hiroshi Tsutsui received his B.E. degree in Electrical and Electronic Engineering and his master and Ph.D. degrees in Communications and Computer Engineering from Kyoto University in 2000, 2002, and 2005, respectively. He is currently an assistant professor in the Department of Communications and Computer Engineering, Kyoto University. His research interests include circuits and systems for image processing and VLSI design methodology. He is a member of IEEE, ACM, IPSJ, IEEJ, and IIEEJ.

Hiroyuki Ochi received the B.E., M.E., and Ph.D. degrees in Engineering from Kyoto University in 1989, 1991, and 1994, respectively. In 1994, he joined Department of Computer Engineering, Hiroshima City University as an associate professor. Since 2004, he has been an associate professor of Department of Communications and Computer Engineering, Kyoto University. His research interests include low-power/reliability-aware VLSI design and reconfigurable architectures. He is a member of IPSJ, IEEE, and ACM.

Takashi Sato received B.E. and M.E. degrees from Waseda University, Tokyo, Japan, and a Ph.D. degree from Kyoto University, Kyoto, Japan. He was with Hitachi, Ltd., Tokyo, Japan, from 1991 to 2003, with Renesas Technology Corp., Tokyo, Japan, from 2003 to 2006, and with the Tokyo Institute of Technology, Yokohama, Japan. In 2009, he joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently a professor. He was a visiting industrial fellow at the University of California, Berkeley, from 1998 to 1999. His research interests include CAD for nanometer-scale LSI design, fabrication-aware design methodology, and performance optimization for variation tolerance. Dr. Sato is a member of the IEEE. He received the Beatrice Winner Award at ISSCC 2000 and the Best Paper Award at ISQED 2003.