A place-aware redundancy methodology for multi-cell upsets mitigation in NoC

Jiajia Jiao^{a)}, JiaBin Wang, and Dezhi Han

College of Information Engineering, Shanghai Maritime University, 201306

a) jiaojiajia@shmtu.edu.cn

Abstract: With scaling technology node, increasing Multi-Cell Upsets (MCU) is dramatically challenging the reliable system design. Network on Chip (NoC) as the communication infrastructure in a many-core processor, is also suffering the serious MCU impacts. Therefore, a place-aware redundancy methodology is proposed to alleviate the MCU impacts on NoC via exploiting MCU correlation. The simulation results demonstrate that, compared with 50% error recovery of latest works, the proposed approach achieves up to 95.8% error recovery with even only 6.91% extra area cost.

Keywords: soft error, MCU, redundancy, place-aware, NoC

Classification: Electron devices, circuits and modules

References

- [1] R. C. Baumann: "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Rel.* **5** (2005) 305 (DOI: [10.1109/TDMR.2005.853449](https://doi.org/10.1109/TDMR.2005.853449)).
- [2] K. Constantinides, *et al.*: "Bulletproof: A defect-tolerant CMP switch architecture," *High-Performance Computer Architecture* (2006) 5 (DOI: [10.1109/HPCA.2006.1598108](https://doi.org/10.1109/HPCA.2006.1598108)).
- [3] F. T. Bortolon, *et al.*: "Exploring the impact of soft errors on NoC-based multiprocessor systems," *Circuits and Systems (ISCAS)* (2018) 1 (DOI: [10.1109/ISCAS.2018.8351391](https://doi.org/10.1109/ISCAS.2018.8351391)).
- [4] D. Alexandrescu: "A comprehensive soft error analysis methodology for SoCs/ASICs memory instances," *On-Line Testing Symposium* (2011) 175 (DOI: [10.1109/IOLTS.2011.5993833](https://doi.org/10.1109/IOLTS.2011.5993833)).
- [5] J. Jiao, *et al.*: "Accelerated assessment of fine-grain AVF in NoC using a multi-cell upsets considered fault injection," *Microelectron. Reliab.* **54** (2014) 2629 (DOI: [10.1016/j.microrel.2014.06.008](https://doi.org/10.1016/j.microrel.2014.06.008)).
- [6] J.-S. Shen, *et al.*: "PRESSNoC: Power-aware and reliable encoding schemes supported reconfigurable network-on-chip architecture," *IEEE Embedded and Multimedia Computing* (2009) 1 (DOI: [10.1109/EM-COM.2009.5402995](https://doi.org/10.1109/EM-COM.2009.5402995)).
- [7] M. Ebrahimi, *et al.*: "Layout-based modeling and mitigation of multiple event transients," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **35** (2016) 367 (DOI: [10.1109/TCAD.2015.2459053](https://doi.org/10.1109/TCAD.2015.2459053)).
- [8] Nirgam: <http://nirgam.ecs.soton.ac.uk/>.
- [9] A. B. Kahng, *et al.*: "ORION 2.0: A fast and accurate NoC power & area model for early-stage design space exploration," *DATE* (2009) 423 (DOI: [10.1109/DATE.2009.5090700](https://doi.org/10.1109/DATE.2009.5090700)).

- [10] B. S. L. K. Anusha and K. S. N. Raju: “Design of high performance shared buffer NoC router,” *J. Eng. Comput. Sci.* **4** (2015).

1 Introduction

With the scaling technology of integrated circuit, Network on Chip (NoC) has become the compromising communication component in the dominate many-core system. However, Radiation-induced soft errors from high-energy neutrons and alpha particles typically results in bits upset in NoC, which leads to NoC status and even system corruption. Therefore, NoC is also suffering the soft error impacts seriously. Due to lower supply voltages and higher integration density, the soft error rate dramatically increases as the technology node scales down [1]. The router buffers of NoC, also called virtual channels (VCs), are implemented as SRAM cells for their speed and density considerations. They take up half of area in a router [2] and vulnerable to soft errors [3]. Consequently, the soft error of VCs is urgent for the reliable NoC design.

Especially, as the cell size and critical charge is getting smaller and smaller resulted from technology scaling, the MCU (Multi-Cell Upset) can no longer be ignored. The MCU ratio of neutral radiation increases dramatically to 40% after 45 nm technology [4]. The typical SECDEC (Single Error Correction Double Errors Check) for NoC VCs is proven to recover 50% errors under MCU fault injection [5]. But it is not enough for high reliability applications like economic computing, aerospace or automobile fields. Thus, the redundancy policy is picked up again. A novel NoC architecture is also proposed via enabling the use of the four proposed different redundancy encoding strategies [6]. For example, DUal Cycle Encoding (DUCE) requires the duplicate buffer space with same bandwidth while Single Additional Flit Encoding and Dual Additional Flit Encoding (SAFE/DAFE) demands extra buffer space with double bandwidth. However, this design doesn't consider MCU correlation so that the reliability improvement is limited under the MCU case. Therefore, how to use the redundancy appropriately for effective MCU mitigation is our focus.

2 Observations

Considering regularity and scalability, typical Mesh topology and XY distributed routing algorithm is adopted for the following analysis and evaluation. The router uses the effective wormhole flow control mechanism. It consists of virtual channels (VCs) for each input port, a central routing computing unit, a virtual channel allocator, a crossbar to connect five input ports and five output ports. VCs based NoC architecture is used to configure the redundancy policy.

Redundancy is often classified into two categories: temporal and spatial. The former denotes the available resource is used in time division multiplexing. It is often adopted in the case of enough latency slack. For NoC reliability issue, this temporal redundancy seems low efficiency because of pipeline-style packets transmission. It requires large buffer to store the first coming copies and long wait

Fig. 1. Soft error test results for neutrons on 45 nm SRAM used in [5] and provided by IROC: (a) bits upset distribution; (b) physical MCU error maps for 2 and 3-bit upset.

latency for multi-copy synchronous for arbitration. Instead, the spatial redundancy behaves space division multiplexing and matches with rich inherent VCs in NoC. Consequently, all protected VCs are based on spatial redundancy in this paper.

However, we observe that the raw redundancy for MCU errors in NoC has low efficiency. E.g., if MCU_3 (three bits upset) occurs in the adjacent redundant VCs, the error recovery rate of TMR (Triple Modular Redundancy) in the case can be zero. Instead of configuring a higher redundancy degree, our idea is to exploit the MCU correlation for high reliability improvement of VCs redundancy. This method is inspired by the observation, where different particle environments determine different MCU patterns as the MCU distribution used in Fig. 1 [5] from iROC Corp. The related adjacent cells upset of MCU causes more serious impacts than SBU due to the multi-bits upset correlation. In other words, if the correlation of MCU patterns can be degraded, the MCU impacts can be mitigated. In a NoC router, MCUs mainly occur in VCs. The smallest element of VC is for storing a flit. The MCU patterns correlation in NoC are divided into two types: 1) one is self-correlation where bits of MCU occur in a flit; 2) the other one is cross-correlation where bits of MCU occur in different flits. The former can be mitigated well by redundancy policy. Instead, the latter of MCU cross-correlation can be exploited via scheduling the VC elements utilization order so that redundancy can run better. Especially, the cross-correlation ratio is up to 90% of total MCUs for the patterns in [4]. Therefore, it is possible to degrade the MCU cross-correlation for the potential

Fig. 2. Variable TMR VC elements scheduling for ERR evaluation

error recovery rate (ERR) improvement in NoC from the intuitive analysis perspective.

On the other hand, we do a simple test for the quantitative verification. As Fig. 2 shows, an example of 4-depth VC under variable scheduling policy has a variable ERR value. The MCU_2 patterns including nine cases are normalized into three types (I), (II) and (III), whose percentages are $p_1(45.97\%)$, $p_2(34.33\%)$ and $p_3(19.7\%)$ respectively. The faults are mapped in regular VCs structure while a row represents a flit element, e.g., flit width is 32 bits. And we use three scheduling policies a), b), c) with different priorities of VC elements for transmitting flits (R0, R1 and R2 are three copies for TMR, F means free VC with no flit occupation). The case of a) is the normal and default case with sequential element utilization, while b), c) have different skipping intervals. For error type (I), there are no difference between three policies of a), b), c), where two bits upset occurring in a flit and TMR recovers all this kind of error. For error type(II), two bits upset occurring in R1 and R2 respectively for policy a), b) causes TMR failure while only bit upset in R2 (bit upset occurring in F has no impact on normal execution) maintains TMR to work well for policy c). Similar to the error type(II), error type(III) have the same impacts on NoCs reliability. The ERR values of three cases a) b) c) are $\{0.6726, 0.6726, 0.7871\}$. This difference also leads to different

reliability. Therefore, the case study demonstrates that the effective architectural VC scheduling policy is feasible for mitigating MCU impacts on NoCs.

3 Proposed place-aware redundancy method in NoC

Unlike the layout-based mitigation for MCU at circuit-level [7], the proposed place-aware policy is designed for NoC reliability mitigation at architecture-level. It exploits the MCU patterns correlation in NoC router for reliability improvement in a cost-effective way. Using the new VC elements scheduling is a good approach so that the MCU mapped place-aware VC elements are interleaving. The key technique is to select the appropriate skipping intervals for the maximum reliability improvement.

Firstly, the skipping interval distribution should be determined. To keep the minimal design complexity, we choose the uniform distribution. The interval is assumed to be a constant value, k . However, not all intervals are identical. For example of ‘b’) in Fig. 2, the interval is 2 for both $\langle \text{elem_0}, \text{elem_2} \rangle$ and $\langle \text{elem_1}, \text{elem_3} \rangle$. However, that of $\langle \text{elem_2}, \text{elem_1} \rangle$ is 1. Such a case results from finite linear structure (not ring) of VCs. For the problem, we use a new start point technique to maintain the maximum degree of uniform distribution. For a VC, the elements are numbered $\{\text{elem_0}, \text{elem_1} \dots \text{elem_n-1}\}$. The algorithm in Fig. 3 defines new scheduling order of the elements.

The function of VC elements $F(\text{elem_i})$ is defined as the sequential number. We also make the whole set for initial sequential number Q an ascending set $\{0, 1, 2 \dots n-1\}$ in the line 1. And the line 2 to line 7 is the main part to calculate the VC elements scheduling order. The first element elem_0 is assumed as the first to schedule in the line 3 and line 4. The rest uses a circular modulo operation if the assigned value is legal in Q ; otherwise the smallest number in Q is available in the line 6. To avoid the duplicated assignment, the specified value should be deleted from the whole set Q . Finally, the complete order is available in the line 8.

Then, the interval number k should be determined. The legal k is an integer in the range of $[0, n-1]$. If k equal to 0, it represents the default case without skipping in VCs scheduling. We try a greedy-based iterative technique to select k

Algorithm for place-aware VC scheduling order

Input: k, Q ;

Output: $\{F(\text{elem_i})\}$

1.	Initialize $Q: \{0, 1, 2 \dots n-1\}$ in order, i is zero;
2.	For (i from 0 to $n-1$)
3.	If ($i==0$)
4.	$F(\text{elem_i})=0$; /*define elem_0 is the first*/
5.	Else /* handle the non- elem_0 cases */
6.	If $((F(\text{elem_i-1})+k) \% n \text{ locates in } Q)$
	$F(\text{elem_i}) = (F(\text{elem_i-1})+k) \% n$;
	Else
	$F(\text{elem_i}) = \text{the smallest number of } Q$
	Delete $F(\text{elem_i})$ from Q .
7.	Endfor /* loop*/
8.	Return $\{F(\text{elem_i})\}$;

Fig. 3. Place-aware algorithm to configure VC elements scheduling order

so that the MCU induced correlation of redundant VC elements is degraded. As is well known, the closer the VC elements are, the higher the correlation of elements is. Therefore, the best k would make the scheduling order of adjacent VC elements as far as possible for the highest EER. $F^{-1}(i)$ is the inverse function of $F(elem_i)$ and d denotes the physical distance is of MCU patterns. m represents the total number of MCU patterns. And $w(q)$ is the percentage of specified MCU pattern q . $s(q)$ is the set of the distances of any two bits upset for MCU pattern q . rd is the given redundancy degree, e.g., rd is set to 3 for TMR policy. $Coef(q, i, j)$ is the coefficient for MCU pattern q between VC element i and element j . The overall logical distance (scheduling order difference) of all correlated VC elements is D . Eq. (1) shows that the target is to maximize D by calculating all possible k values in the range of $[0, n - 1]$.

$$\begin{aligned} \text{Maximize } D &= \sum_{j=0}^{n-1} \sum_{i=j+1}^{n-1} \sum_{q=0}^{m-1} |F_k^{-1}(i) - F_k^{-1}(j)| \times (1 - coef(q, i, j) \times w(q)) \\ \text{Subject to } \begin{cases} coef(q, i, j) = \begin{cases} 0 & (|F_k^{-1}(i) - F_k^{-1}(j)| \notin s(q)) \vee (F_k^{-1}(i)/rd \neq F_k^{-1}(j)/rd) \\ 1 & \text{otherwise} \end{cases} \\ k = 0 \dots n - 1 \\ \sum_{q=0}^{m-1} w(q) = 1 \end{cases} \end{aligned} \quad (1)$$

In a word, we degrade the MCU patterns correlation via the proposed place-aware redundancy for minimizing MCU impacts on NoC.

4 Simulations and analysis

All simulation results are based on the cycle accurate NoC simulator Nirgam [8]. To support various redundancy methods verification and our fault model, we have extended original simulator to setup the complete experimental platform. Table I provides the detailed configuration. We choose typical synthesis traces and realistic applications with different traffic load cases. And each case also does fault injection campaign of 1 K faults respectively. Considering the redundancy cost, this paper just takes the often-used TMR as cases study in the following analysis.

Table I. Simulation parameters configuration

Parameters	Value
Basic topology	$\{3 \times 3, 3 \times 4\}$ Mesh
Routing algorithm	XY
Data/Tail flit payload size	4 bytes
Packet payload size	32 bytes (a cache line)
Packet size	9
Packet occurring strategy	CBR (Constant Bit Rate)
Traffic trace	{ VOPD, MWD, Transpose, Uniform }
VC Buffer size (depth)	4
Load Rate (packet/cycle/node)	{ 0.01, 0.05, 0.1, 0.5, 1 }
Fault list size	{ 1 K }
Fault tolerant configuration	Proposed method, ECC, typical TMR
Each Simulation Time	10000 cycles

Fig. 4. Comparison between different mitigations (a) AVF value; (b) area cost.

To evaluate the proposed method from a comprehensive perspective, the results are compared with ECC in [5] as well as TMR used in [6] under MCU errors. It is noted that: 1) to make sure the consistent configuration of ECC in [5], we set the same parameters in Table I; 2) TMR is one of basic reliable configurations for a flexible NoC design of [6], thus we re-implement a simple redundancy version of TMR structure and set the VC depth to $4 * 3$. 4) Architecture-level Architectural Vulnerability Factor (AVF), a widely-used metric to quantify the reliability of NoC, represents the probability that a single bit upset will result in a user-visible error in the final output [5]. The lower AVF is, the better fault tolerant method is.

The greedy-based iterative algorithm in Section 3 generates the best interval k , equal to 4. And the place-aware scheduling order is $\{0 \rightarrow 4 \rightarrow 8 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 2 \rightarrow 6 \rightarrow 10 \rightarrow 3 \rightarrow 7 \rightarrow 11\}$. To verify the effectiveness of the best interval, we also select the fixed interval as 1. We use the accelerated fault injection method [5] to evaluate the AVF results in Fig. 4(a). It is shown that the place-aware methods perform better than typical ECC and default TMR protections. Default TMR protection even has a little higher AVF value 0.1361 on average than 0.1321 by ECC protection. This case is reasonable and caused by the MCU impacts, e.g., if two SBU errors occur in two flits simultaneously, ECC can correct this kind of errors while TMR is failed. Default TMR and ECC for MCU mitigation have 51.2% and 50.04% error recovery over the no protection configuration. The more important is the proposed place-aware method with the best interval has up to 95.85% error recovery for benchmark VOPD, and 94.84% error recovery for four benchmarks on average. If select the fixed interval as 1, the MCU correlation cannot be considered in the VC scheduling completely so that the error recovery is 75.73% on average. The MCU

Fig. 5. A logical diagram of mixed place-aware method with shared VCs and partial TMR

model includes two bits, three bits ... up to 9 bits upset so that TMR is powerless in some extreme cases, where two or more copies of TMR all have bit upsets. In all, the proposed place-aware method can provide the strong reliability and makes good use of the redundancy.

We also evaluate cost of the place-aware redundancy method using the open source tool Orion [9]. Ignore the voter and extra wire placement in redundant configuration for its low area cost. Fig. 4(b) list the normalized area cost of a 5-port NoC router. ECC has about 5.79% extra overhead while TMR schemes require about 51.8% extra overhead.

To further reduce the area overhead by redundancy, the place-aware method can also easily applied in some existing works. The idea is to pay acceptable performance loss or reliability loss for saving area. At first, combine the place-aware method with the multi-port (four neighbors N, S, W, E and local C to connect IP core) shared VCs architecture as Fig. 5 shows. The VCs for each port are centralized for high utilization in shared VCs architecture. Dynamic VC scheduling policy lets place-aware information directly serve the shared VCs structure. They can be mixed in the virtual channel arbiter design together. The critical work of combination two methods is to reset the VCs depth for selecting the corresponding scheduling interval using eq. (1). The results in [10] demonstrate that for the same area configuration, shared VCs can achieve 1.5x performance speedup. In other words, for the same performance, the shared VCs can save equivalently 33.33% area cost. Therefore, the TMR scheme in the shared VCs brings only 34.53% extra router overhead. If use the partial place-aware TMR in shared VCs structure, the area cost can be further saved, e.g., decoded information of flit tag shows only reliability sensitive 20% data transfer requires TMR while the rest 80% data does not need protection in Fig. 5(b). And the corresponding extra router area cost can be saved 80% over the full TMR configuration, only 6.91%.

From the above results and analysis, we can conclude the proposed place-aware redundancy performs high reliability with acceptable area cost.

5 Conclusions

In this paper, the proposed place-aware redundancy methodology can exploit the inherent MCU correlation and enhance the error recovery of redundant VCs architecture well. The simulation results demonstrate that, compared with 51.2% error recovery of ECC protection and 50.04% error recovery of default redundancy, the proposed efficient method achieves 94.85% error recovery on average. In addition, the place-aware redundancy method brings no extra area cost over default redundancy and can be applied in some shared VCs architecture or partial redundancy for further area reduction. Therefore, the proposed place-aware redundancy is cost-effective to mitigate the MCU impacts on NoC well.

Acknowledgments

This work was supported by National Natural Science Foundation of China (grant numbered 61502298) and Innovation Program of Shanghai Maritime University.