



**HAL**  
open science

## A study on the reliability improvement factor of fault tolerant mechanisms

Jongwhoa Na, Dongwoo Lee

► **To cite this version:**

Jongwhoa Na, Dongwoo Lee. A study on the reliability improvement factor of fault tolerant mechanisms. Safecom 2013 FastAbstract, Sep 2013, Toulouse, France. pp.NC. hal-00926549

**HAL Id: hal-00926549**

**<https://hal.science/hal-00926549>**

Submitted on 9 Jan 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A study on the reliability improvement factor of fault tolerant mechanisms

Jongwhoa Na, Dongwoo Lee  
Department of Avionics and Electronics Engineering,  
Korea Aerospace University,  
Republic of Korea  
{jwna, dongwoo1}@kau.ac.kr

**Abstract**—We present a study on the reliability improvement factor (RIF) to quantify the reliability of the various fault tolerant mechanisms at the system level. First, we find the system level failure rate using co-simulation models and statistical fault injection (StFi). We built co-simulation targets using SystemC simulation models of baseline single-core ARM7, dual-modular and triple-modular redundant ARM7 processors and Mibench embedded benchmark SW. Since the number of experiments in StFi is large, we utilized simulation kernel-modified simulated fault injection tool. Next, we calculated the RIF using the failure probability functions of the co-simulation targets. In this way, we were able to compare the reliability improvement of the fault tolerant mechanism at the system level.

**Keywords**; statistical fault injection, reliability improvement factor, fault-tolerant processor, fault-tolerant mechanism

## I. INTRODUCTION

In the safety-critical embedded systems (SCES) in aircrafts and automobiles, fault tolerant processors (FTP) became a major components. FTPs increase the reliability of the target using various types of redundancies. However, these redundancies also increase the cost of the target considerably. In order to manage the cost increase in SCES, we need a reliability index to quantify these redundancies. We may use MTTF as a reliability index for the target, which has components with sufficient usage history. However, because of the fast developing speed in VLSI/SoC technology, it is difficult to keep the usage history of the components of the modern SCES. This calls for a reliability index without usage history.

In this paper, we explain the application of the reliability improvement factor (RIF) as a reliability index of the effectiveness of the FT mechanism in the FTP. RIF is defined as the ratio of the probability of failure,  $F(t)$ , of the non-redundant system to that of the redundant system [1,2]. For example, using ARM7 processor as a baseline processor, we may quantify the effectiveness of the TMR mechanism over DMR by finding the  $RIF_{TMR}$  of TMR ARM7 over baseline ARM7 and the  $RIF_{DMR}$  of DMR ARM7 over the same baseline.

We can calculate the  $F(t)$  of RIF by performing the fault injection experiments and finding the failure rate of the FTP and the baseline target. In order to make the failure rate legitimate, we use statistical fault injection (StFi) with confidence level and reasonable targets which can be a real SCES at the final stage or co-simulation target at the early stage of the development life cycle.

We report the case study of RIF for reliability index using StFi and co-simulation target. We built a co-simulation model

of a SystemC hardware simulation model of baseline ARM7, DMR ARM7, TMR ARM7 processors and the cross-compiled executable files of the Mibench embedded benchmark suits [3]. For the statistical fault injection experiment, we calculate the required number of fault injections at a 95% confidence level for the given fault models and the SUT [4]. Because of the complexity of the SUT, the required number of fault injections is very large. Thus, the efficiency of the injection tool is important. In this regard, we use a novel simulated fault injection environment that uses a modified simulation kernel instead of saboteur or mutation technique. A detailed explanation of the kernel-modified simulated fault injection is explained elsewhere [5].

## II. FAULT TOLERANT PROCESSORS

For hardware, we designed a SystemC simulation model of the ARM7 processor that could execute about 40 instructions from the ARM7 architecture, as shown in Fig. 1.

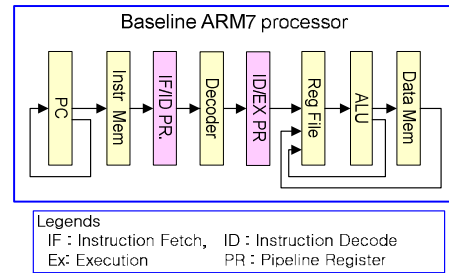


Fig. 1. ARM7 processor model

In order to make the cases of the qualitative comparisons of various FT mechanisms, we designed a SystemC simulation model for the DMR and TMR ARM7 processors. In the case of the DMR ARM7, we duplicated the data path with two ARM processors and added a simple fault recovery controller that could detect faults at the pipeline stages. In the design of the TMR ARM7 in Fig. 2, we implemented the micro-architectural redundancy by triplicating each module and adding a voter. The details of the DMR and TMR architectures can be found in many other studies [4].

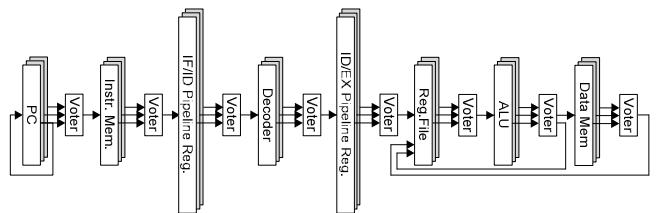


Fig. 2. Triple modular redundant ARM7 processor

### III. STATISTICAL FAULT INJECTION EXPERIMENTS

We performed fault injection experiments using the co-simulation models using the three hardware models (baseline single ARM7, TMR ARM7, and DMR ARM7 processors) and the GSM code from a Mibench embedded benchmark suit, and the four fault models (permanent and transient stuck-at-1/0). For each of the experimental setups, we setup statistical fault injection experiments by calculating the required number of injections for the given confidence level. The results of fault injections are summarized in Table 1. In the case of baseline ARM7, the injection result should be one of the not-active, benign, or silent data corruption (SDC) state. In the case of DMR and TMR ARM7, we have two more states: recovered and detected unrecoverable error (DUE).

TABLE 1 The results of statistical fault injection campaign on co-simulation models and fault models

S/W		GSM benchmark (Mibench)			
H/W	Fault type & model	Transient		Permanent	
	Failure type	stuck-at-1	stuck-at-0	stuck-at-1	stuck-at-0
Baseline ARM	Non Active	28,468	71,191	8	31,244
	Benign	59,035	22,522	12,669	22,500
	Silent Data Corruption	12,497	6,287	87,323	46,256
DMR ARM	Non Active	29,046	71,314	125	32,950
	Benign	33,119	7,926	16,201	11,312
	Recovery	35,217	17,706	44,705	33,312
	Silent Data Corruption	519	769	75	1,008
	Detected Unrecoverable Error	2,099	2,285	38,894	21,418
TMR ARM	Non Active	47,554	74,237	229	5,867
	Benign	2,117	1,279	117	277
	Recovery	89,882	34,260	19,138	13,669
	Silent Data Corruption	447	224	516	187
	Detected Unrecoverable Error	0	0	0	0

Using Table 1, we calculate the failure rate by dividing the sum of the SDC and DUE failure rates by the total number of fault injections for the three types of processors. Using these failure rates and assuming steady state operating condition, we are able to calculate the reliability or failure distribution functions for each case of the baseline ARM, DMR ARM, and TMR ARM co-simulation targets. With the failure function, we can calculate the RIF of the DMR and TMR mechanism as follows:

$$RIF_{FTM} = (1 - R_{baseline}(t)) / (1 - R_{FTM}(t))$$

We presents the  $RIF_{TMR}$  and  $RIF_{DMR}$  over baseline ARM7 processor in Fig. 3. Using the graph, we can compare the effectiveness of the TMR mechanism over DMR mechanism for given time in a quantitative manner. Initially, we can find the effectiveness of the TMR mechanism to be 50~70 times higher than that of the DMR mechanism. Also, we can find that the improvement of  $RIF_{TMR}$  over  $RIF_{DMR}$  decreases over time.

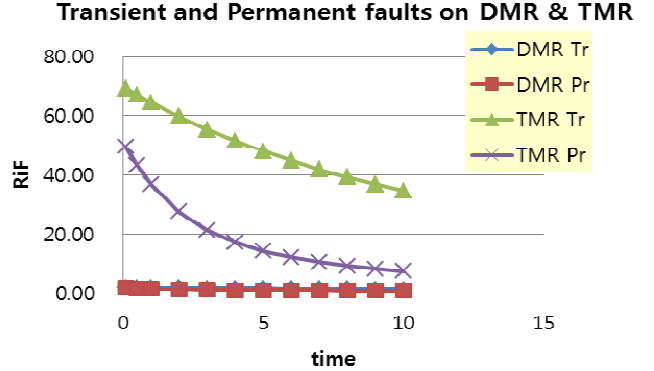


Fig. 3. DMR and TMR reliability improvement factors for the transient and permanent fault models

### IV. CONCLUSION

We have reported on the reliability improvement factor (RIF) of the DMR and TMR mechanisms. Instead of using the static failure rates from the reliability block diagram, we utilized the dynamic failure rates using the co-simulation targets of SystemC hardware and Mibench benchmark software so that the RIF becomes more practical. The experimental results suggested that the TMR mechanism is initially more resilient than DMR. As such, we may compare the reliability or the cost-effectiveness of the FTM at the system level of various types of redundancy mechanisms.

Using these simulation study as a basis, we are planning to extend the experiments using other benchmark software and other FT mechanisms to investigate the applicability of the RIF as a quantitative reliability index of the fault tolerant mechanism for a group of embedded systems.

### REFERENCES

- [1] D.W. Lee and J.W. Na, "A Novel Simulation Fault Injection Method for Dependability Analysis," IEEE Design and Testing, vol. 26, no.6, pp. 50-61, Dec. 2009.
- [2] Krol, Th., A Generalization of Fault-Tolerance Based on Masking, Ph.D. thesis, Eindhoven.
- [3] Mei-Chen Hsueh, Timothy K. Tsai Ravishankar K. Iyer, "Fault Injection Techniques and Tools," IEEE Computer, April 1997.
- [4] Shubu Mukherjee, "Architecture Design for Soft Errors," Morgan Kaufmann Publishers.
- [5] Menkae Jeng and Howard Jay Siegel, "Implementation Approach and Reliability Estimation of Dynamic Redundancy Networks," Real-Time Systems Symposium, pp. 79-88, New Orleans, LA, 1986.