No Free Lunch in Soft Error Protection?

Ilia Polian[1], Sudhakar M. Reddy[2], Irith Pomeranz[3], Xun Tang[2], Bernd Becker[1]

[1]Institute for Computer Science
Albert-Ludwigs-University
Freiburg, Germany

[2]ECE Department
University of Iowa
Iowa City, USA

[3]School of ECE
Purdue University
West Lafayette, USA

Extended Abstract

It is generally expected that nanoelectronic circuits will have to be protected against soft errors [Shivakumar 02]. Recent publications suggest that a significant fraction of soft errors do not have any effect [Wang 04, Li 07]. As a consequence, it may be sufficient to apply hardening techniques to selected parts of a circuit [Mohanram 03, Hayes 07]. We argue that employing massive redundancy for full protection of the circuit cannot be avoided unless the specification is loosened.

A soft error may be masked by the circuit, architecture or application.[Shivakumar 02, Wang 04, Li 07]. A spot in the circuit, e.g., a flip-flop or a combinational logic signal, is called *one-cycle-redundant* if a soft error on this spot, defined as a bit flip lasting for one cycle, is masked irrespective of the system state and the input sequence being processed by the system. A spot is called *redundant* if a permanent fault on this spot, i.e., a stuck-at fault, does not have an effect on the system behavior. One-cycle-redundant faults represent transient faults that are present for one cycle of operation in this discussion. Intuitively, one-cycle redundancy is a weaker requirement than redundancy. Hence, one may be tempted to believe that there are more one-cycle-redundant spots than redundant spots.

We determined the numbers of single-cycle-redundant and redundant spots in ISCAS-89 sequential benchmark circuits based on the observation that a (single-cycle-) redundant spot correspond to a (single-cycle-) redundant, i.e., untestable, fault on this spot. We used a modified version of a tool that identifies permanent redundant stuck-at faults in sequential circuits [Reddy 99]. This tool identified almost all redundant spots in ISCAS-89 circuits. A spot is determined to be redundant if the fault-free circuit and the faulty circuits are synchronizable and there is no test to detect the fault [Pomeranz 96]. The results are given in Table 1. We used the set of uncollapsed single stuck-at faults, which corresponds to all signals in the combinational logic. In Table 1, after the circuit name we give the number of redundant permanent stuck-at faults followed by the number of single-cycle-redundant faults. It can be seen that the numbers of stuck-faults that are redundant and the number of single cycle redundant faults are the same in many of the circuits. In thirteen circuits (shown by entries in bold) where the numbers of single cycle redundant faults are higher than the number of redundant stuck-at faults the difference is small with few exceptions. For all the circuits in our experiment the set of single-cycle-redundant faults contains the set of redundant stuck-at-faults. However, we have not been able to prove that this observation will hold in general.

In the last two columns of Table 1 we give the number of redundant faults considering only the outputs of the flip-flops in the circuit. Inside parentheses we show the number of flip-flop outputs at which both the stuck-at-0 and stuck-at-1 faults are redundant or both the single cycle faults are redundant. It can be seen that the numbers of redundant stuck-at faults and the numbers of redundant single cycle redundant faults are identical when we consider faults at flip-flop outputs. Additionally the numbers in parentheses show that the numbers of flip-flops at which both the faults are redundant are also identical. These are the flip-flops that need not be augmented to be fault-tolerant to transient faults as suggested in [Seshia 07].

Empirical data given in Table 1 shows that the sets of single-cycle-redundant and redundant spots are very similar for combinational logic, and always identical for flip-flops. Does this mean that no low-cost soft error protection is possible? Not necessarily. In [Seshia 07], the properties which are used to formally verify a telecommunication chip are proven to hold even under errors in some of the flip-flops. In [Polian 06], over 70% of the logic signals in an MPEG motion estimator were shown to be non-critical; soft errors on these signals led to limited effects which were guaranteed to disappear after a few clock cycles.

This seems to suggest that the analyzed systems were over-specified and, consequently, over-designed. Even though no output bits were totally correct, the essential properties of the systems were satisfied. In our opinion, the key to the design of dependable systems is the understanding of critical versus non-critical errors with respect to the specification. For instance, soft errors in performance-enhancing modules of a microprocessor, such as branch predictors, are typically non-critical, except in application with hard real-time constraints. Tolerating occasional pixel deviations in imaging application such as video processing may be preferred to spending additional hardware and power consumption for massive redundancy needed to prevent the errors from happening [Nowroth 08]. A range of applications, e.g., in communication, can handle certain classes of errors at the system level. Additional fields in which this concept appears to be easily applicable include recognition, mining, synthesis, tracking and control.

In our opinion, design and implementation of cost-aware nanoelectronic systems will have to incorporate dependability aspects with respect to soft error resilience as part of specification. We need novel methods of *specification-aware synthesis*, which satisfy the dependability requirements by a carefully chosen mix of techniques such as hardware redundancy, commit-rollback recovery, error-resilient information coding and algorithmic fault tolerance in software. This is in contrast to today's approach where a circuit is designed first and then its dependability is enhanced without modifying the circuit's Boolean function. The future specification formalisms must distinguish between functionality considered essential, and functions that are allowed to fail occasionally due to soft errors.

References

[Hayes 07] J. Hayes, I. Polian, and B. Becker, "An analysis framework for transient-error tolerance", Proceedings IEEE VLSI Test Symposium, Berkeley, CA, USA, 2007.

[Li 07] X. Li and D. Yeung. Application-level correctness and its impact on fault tolerance. Proceedings Int'l Symp. on High Performance Computer Architecture, pp. 181-192, 2007.

[Mohanram 03] N. Mohanram and N. Touba. Partial error masking to reduce soft error failure rate in logic circuits. Proceedings Int'l Symp. on Defect and Fault Tolerance, pp. 443-440, 2003.

[Nowroth 08] D. Nowroth, I. Polian and B. Becker. A study of cognitive resilience in a JPEG compressor. Proceedings Int'l Conference on Dependable Systems and Networks, 2008 (in press).

[Polian 06] I. Polian, B. Becker, M. Nakasato, S. Ohtake, and H. Fujiwara, "Low-cost hardening of image processing applications against soft errors", Proceedings International Symposium on Defect and Fault Tolerance, pp. 274-279, Arlington, VA, USA, 2006.

[Pomeranz 96] I. Pomeranz and S.M. Reddy, "On Removing redundancies from Synchronous Sequential Circuits with Synchronizing Sequences", IEEE Trans. on Computers, pp. 20-32, Jan. 1996.

[Reddy 99] S.M. Reddy, I. Pomeranz, X. Lin and N.Z. Basturkmen, "New Procedures for Identifying Undetectable and Redundant Faults In Synchronous Sequential Circuits," Proc. VLSI Test Symposium, pp. 275-281, April 1999.

[Seshia 07] S.A. Seshia, W. Li, and S. Mitra. Verification-guided soft error resilience. Proceedings Design Automation and Test in Europe Conf., 2007.

[Shivakumar 02] P. Shivakumar, M. Kistler, W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic", Proceedings Int'l Conference on Dependable Systems and Networks, pp. 389-398, 2002.

[Wang 04] N.J. Wang, J. Quek, T.M. Rafacz, and S.J. Patel. Characterizing the effects of transient faults on a high-performace processor pipeline. Proceedings Int'l Conference on Dependable Systems and Networks, 2004.

Table 1: Experimental results

circuit	permanent redundant	single cycle redundant	permanent redundant at DFF	single cycle redundant at DFF
S208	**121**	**124**	**39 (14)**	**39 (14)**
S298	60	60	22 (3)	22 (3)
S344	18	18	5 (0)	5 (0)
S349	22	22	6 (0)	6 (0)
S382	8	8	4 (0)	4 (0)
S386	**76**	**81**	**0 (0)**	**0 (0)**
S400	20	20	7 (0)	7 (0)
S420	**438**	**447**	**123 (44)**	**123 (44)**
S444	30	30	9 (1)	9 (1)
S499	163	163	62 (0)	62 (0)
S526	56	56	12 (0)	12 (0)
S641	**140**	**152**	**8 (0)**	**8 (0)**
S713	**218**	**230**	**8 (0)**	**8 (0)**
S820	**60**	**71**	**18 (0)**	**18 (0)**
S832	**79**	**96**	**26 (1)**	**26 (1)**
S838	**1070**	**1107**	**291 (104)**	**291 (104)**
S953	**10**	**14**	**5 (0)**	**5 (0)**
S967	**21**	**24**	**7 (0)**	**7 (0)**
S1196	**3**	**5**	**0 (0)**	**0 (0)**
S1238	83	83	1 (0)	1 (0)
S1269	**3**	**4**	**1 (0)**	**1 (0)**
S1423	40	40	5 (0)	5 (0)
S1488	68	68	14 (0)	14 (0)
S1494	94	94	16 (0)	16 (0)
S3384	**1**	**17**	**0 (0)**	**0 (0)**
S4863	210	210	0 (0)	0 (0)
S5378	2112	2112	203 (56)	203 (56)
S35932	7344	7344	1280 (640)	1280 (640)