Logic Soft Errors in Sub-65nm Technologies Design and CAD Challenges

Subhasish Mitra
Intel Corporation
subhasish.mitra@intel.com

Tanay Karnik
Intel Corporation
tanay.karnik@intel.com

Norbert Seifert
Intel Corporation
norbert.seifert@intel.com

Ming Zhang
Intel Corporation
ming.y.zhang@intel.com

ABSTRACT

Logic soft errors are radiation induced transient errors in sequential elements (flip-flops and latches) and combinational logic. Robust enterprise platforms in sub-65nm technologies require designs with built-in logic soft error protection. Effective logic soft error protection requires solutions to the following three problems: (1) Accurate soft error rate estimation for combinational logic networks; (2) Automated estimation of system effects of logic soft errors, and identification of regions in a design that must be protected; and, (3) New cost-effective techniques for logic soft error protection, because classical fault-tolerance techniques are very expensive.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing and fault-tolerance.

General Terms

Design, Reliability.

Keywords

Architectural Vulnerability Factor, Built-In Soft Error Resilience, derating, error blocking, error detection, recovery, soft error.

1. INTRODUCTION

Logic soft errors affect sequential elements (latches and flip-flops) and combinational logic. Most of these errors do not have any impact on system operation [1, 2]. For example, an error in a flip-flop whose output is AND-ed with another signal with logic value 0 has no effect on the system. As another example, an error in an operand of a speculatively executed instruction which is finally not committed (and becomes a dead instruction) does not impact system operation. However, a significant percentage of logic soft errors can result in data corruption without the system or the user knowing about it. As a result, system data integrity is severely compromised. For example, consider the effect of a 1 → 0 bit flip in the most significant bit of the register storing the amount of money deposited into a bank account. This is referred to as an *undetected error* or silent *data corruption*, and is of great concern.

Logic soft errors are very significant contributors to system-level silent data corruption for designs manufactured in advanced technologies (90nm, 65nm, onward) and targeted for enterprise computing and communications applications [3, 18]. Given the undetected soft error rate requirements of such applications, soft error protection of sequential elements (latches and flip-flops) requires immediate attention.

Design and CAD challenges for effective logic soft error control are discussed below.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

1.1 Automated Estimation of Soft-Error Susceptibility of Combinational Logic

Automated estimation of soft error rates of SRAM cells, latches and flip-flops from pre-layout or post-layout circuit structures is now well-understood [16]. In contrast, more research is required in automating soft error rate estimation of combinational logic.

Radiation can cause a logic hazard at any gate output of a combinational circuit. The hazard may propagate through the combinational logic and errors may or may not get latched by the sequential elements depending on the following factors [14].

Logical masking: The hazard may not propagate because there may not be any sensitized path from the node where the strike happened to any output of the combinational logic circuit.

Temporal masking: As the hazard propagates towards a sequential element, the noise on the data input node of the sequential element may be outside of its latching window. Hence the error will not be latched and there will be no soft error.

Electrical masking: Since all CMOS circuits have limited bandwidths, hazards with bandwidths greater than the cut-off frequency will be attenuated. The amplitude of the hazard pulse may reduce, the rise and fall times increase, and eventually the hazard pulse may disappear. However, since most logic gates are nonlinear circuits with a substantial voltage gain, low-frequency pulses with sufficient initial amplitude will be amplified.

Techniques that account for temporal and electrical masking of soft errors are discussed in [17, 19].

1.2 Automated Estimation of System-level Effects of Logic Soft Errors

Not all soft errors cause silent data corruption. Moreover, as indicated in several publications, not all portions of a design are equally likely to cause silent data corruption when affected by soft errors. Automated techniques are required to estimate the probability that a soft error in a design results in silent data corruption, given that the soft error event has occurred. This problem is also referred to as the *Architectural Vulnerability Factor* (*AVF*) or *logic derating* estimation.

Two major simulation-based AVF estimation approaches that are currently being used in a limited way are fault simulation (also

called fault injection) [2, 4 and several others], and fault-free simulation [1, 5]. There are several open questions and challenges that must be resolved for these techniques to reach their full potential [3]. These are related to the scalability of these techniques for large designs, execution times of these techniques, accuracy of estimation, and applicability to general designs (and not limited to special designs such as microprocessors).

Like any simulation approach, the accuracy of AVF estimation depends on the simulated input stimuli. For microprocessors, benchmarks originally intended for performance evaluation are often used for AVF computation. The absence of such benchmarks for other designs (e.g., network processors and routers) have led the designers to rely on verification traces for AVF estimation. Since the original objectives of all these stimuli are different from system reliability evaluation, it is questionable whether these are sufficient for AVF estimation. New specialized benchmarks for system reliability evaluation are required.

1.3 Effective Logic Soft Error Protection Techniques

We already discussed that sequential elements (latches and flip-flops) require soft error protection for several designs in advanced technologies. It is needless to say that the major factors that determine the effectiveness of any soft error protection technique are: (1) the amount of soft error protection obtained, and, (2) corresponding power, performance and area overheads. Since all regions of a design do not have the same architectural vulnerability factors, CAD tools are required for optimized insertion of protection techniques that maximize the amount of soft error protection while incurring minimal overheads.

Moreover, the recent industry trend to reuse a core design for various applications introduces a new challenge in the domain of soft error protection. For example, the use of a specific protection technique in a core may incur acceptable power overhead for an an application that requires soft error protection; however, the incurred power overhead may be excessive for another application that intends to reuse the same core, but doesn't require soft error protection. One option is to build in two operation modes – an *error resilient mode* in which the protection mechanisms are turned on, and an *economy mode* when the protection mechanisms are turned off reducing the power overhead.

Tables 1 presents quantitative comparisons of various promising soft error mitigation techniques in terms of power, performance and area overheads, and the amount of soft error protection that can be obtained. The focus is on latches and flip-flops since they require immediate attention. The protection techniques include: (1) forward-body biased transistors [6, 7]; (2) selective node engineering technique, which increases the capacitances of selective nodes of a circuit [9]; (3) circuit hardening [8]; (4) a recently developed Built-In-Soft-Error-Resilience (BISER) technique that reuses already existing design for test and debug resources to provide soft error protection through error blocking or error trapping [3]; and, (5) classical fault-tolerance techniques [11, 12, 13].

It is clear from Table 1 that the forward body bias technique is very effective if we require a modest 20% reduction in the undetected soft error rate. The selective node engineering technique, which increases the capacitances of selective nodes of a circuit, is an effective approach for designs requring 30-50% undetected soft error rate reduction. For the circuit hardening and BISER techniques, the power overheads are derived based on the assumption that 25% of the flip-flops require soft error protection [2]. The power and area overheads are significantly lower for the BISER technique because it reuses already existent design-for-testability and debug resources. Moreover, the BISER technique allows insertion of an economy mode which enables reuse of the same core design for various applications with soft error protection and power trade-offs.

Intelligent insertion of BISER designs at the sensitive regions of a design minimizes the system-level power overhead. CAD tools that produce optimized insertion of BISER flip-flops by taking into account AVFs are required.

For the BISER technique, the power overhead is between 3-5%. In comparison, hardware duplication and time redundancy techniques such as multi-threading for error detection and Software Implemented Hardware Fault Tolerance (SIHFT) have very significant power overheads. For chip-level duplication, the power overhead is expected to be greater than 100%. For more fine-grained duplication (e.g., [10]), the power overhead is lower. (We estimated the power overhead to be similar to area overhead in the absence of published data). These numbers are greater than a worst-case scenario where all flip-flops are protected with a BISER based technique resulting in 12-20% power overhead. Moreover, time redundancy techniques have very significant performance overheads (40-200%) [11, 12], and are mainly applicable for designs with well-defined architectures such as microprocessors. This is a significant drawback of a time redundancy technique.

Table 1 implies that the BISER technique is most cost-effective. Of course, this implies that more research is required to develop efficient micro-architectural techniques for soft error detection. One major advantage of the BISER based error blocking technique is that it doesn't require any error recovery mechanisms. However, efficient micro-architectural support is required for self-recovery from detected soft errors for a BISER technique that employs error trapping.

2. CONCLUSIONS

Logic soft errors are of major concern for enterprise designs manufactured in sub-65nm technologies. We need solutions to several design and CAD problems related to soft error rate estimation of combinational logic, understanding the system-level effects of logic soft errors, and effective soft error protection. Classical fault-tolerance techniques for soft error detection are very expensive. In comparison, the BISER technique is very effective for soft error blocking or detection. New architectural techniques are required for efficient soft error recovery.

3. REFERENCES

[1] H.T. Nguyen and Y. Yagil, "A Systematic Approach to SER Estimation and Solutions", *Proc. Intl. Reliability Physics Symp.*, pp. 60 – 70, 2003.

[2] N.J. Wang, *et al.*, "Characterizing the Effects of Transient Faults on a High-Performance Processor Pipeline," *Intl. Conf. Dependable Systems and Networks*, pp. 61-70, 2004.

[3] S. Mitra, N. Seifert, M. Zhang, Q. Shi and K.S. Kim, "Robust System Design with Built-In Soft Error Resilience," *IEEE Computer*, Vol. 38, No. 2, pp. 43-52, Feb. 2005.

[4] K.K. Goswami, R. Iyer and L.Y. Young, "DEPEND: A Simulation-Based Environment for System Level Dependability Analysis," *IEEE Trans. Computers*, Jan. 1997.

[5] S.S. Mukherjee, *et al.*, "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor," *MICRO*, 2003.

[6] P. Hazucha, *et al.*, "Measurements and Analysis of SER-Tolerant Latch in a 90nm Dual Vt CMOS Process," *IEEE Journal Solid State Circuits*, pp. 1536-1543, Sept. 2004.

[7] T. Karnik, *et al*., "Impact of body bias on alpha- and neutron-induced soft error rates of flip-flops," *VLSI Circuits Symp.*, pp. 324-325, 2004

[8] T. Calin, M. Nicolaidis, and R. Velaco, "Upset Hardened Memory Design for Submicron CMOS Technology," *IEEE Trans. Nucl. Sci.,* Vol. 43, pp. 2874-2878, Dec. 1996.

[9] T. Karnik, *et al*., "Selective Node Engineering for Chip-level Soft Error Rate Improvement," *VLSI Circuits Symp.,* pp. 204-205, 2002

[10] L. Spainhower and T. A. Gregg, "S/390 Parallel Enterprise Server G5 Fault Tolerance," IBM Journal Research & Development., pp. 863-873, Sept./Nov. 1999.

[11] S.S. Mukherjee, M. Kontz and S. Reinhardt, "Detailed Design and Evaluation of Redundant Multithreading Alternatives," *Intl. Symp. Computer Architecture*, 2002.

[12] N. Oh, P.P. Shirvani and E.J. McCluskey, "Error Detection by Duplicated Instructions in Super-Scalar Processors," *IEEE Trans. Reliability*, pp. 63-75, March 2002.

[13] N.R. Saxena, *et al.*, "Dependable Computing and On-line Testing in Adaptive and Reconfigurable Systems," *IEEE Design and Test of Computers*, pp. 29-41, Jan-Mar 2000.

[14] T. Karnik, P. Hazucha, and J. Patel, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Trans. Dependable and Secure Computing*, Vol. 1, Issue 2, pp. 128-143, April-June 2004.

[15] S. Narendra, *et al*., "1.1V 1GHz communications router with on-chip body bias in 150nm CMOS," *Proc. IEEE Solid-State Circuits Conference*, Volume 2, pp. 218-482, Feb 2002.

[16] N. Seifert and N. Tam, "Timing Vulnerability Factors of Sequentials", *IEEE Trans. Device and Materials Reliability,* Vol. 4, No. 3, p. 516-522, September 2004.

[17] N. Seifert, *et al.*, "Radiation-Induced Clock Jitter and Race", *Proc. Intl. Reliability Physics Symp.*, 2005.

[18] R. Baumann, "The Impact of Technology Scaling on Soft Error Rate Performance and Limits to the Efficacy of Error Correction," *Proc. Intl. Electron Devices Meeting*, pp. 329 – 332, 2002.

[19] M. Zhang and N.R. Shanbhag, "A Soft Error Rate Analysis Methodology," *Proc. ICCAD*, pp. 111 – 118, 2004.

Table 1. System-level comparison of various soft error protection techniques.

	Forward body bias [6, 7, 15]	Selective node engineering [9]	Circuit hardening [8]	Built-In Soft Error Resilience with scan reuse (BISER) [3]	Hardware duplication	Time redundancy (Multi-threading, SIHFT) [11, 12, 13]
Undetected soft error rate reduction	30% reduction	1.5 times	20 times	Error blocking: 20 times, Error trapping: Minimal	Minimal	Minimal
Power overhead (resilient mode)	20% saving	3%	6.4%	3-5%	35-100%	No published data, similar to duplication
Power overhead (economy mode)	20% saving	3%	6.4%	1.6-2.9%	Minimal	Multi-threading: Minimal, SIHFT: none
Performance overhead	None	None	None	None	Minimal	Multi-threading: 20-40%, SIHFT: 40-200%
Area overhead	3%	None	~ 0.78%	~0.1%	35-100%	Multi-threading: Some, SIHFT: none
Extra effort for recovery	None	None	None	Error blocking: None, Error trapping: Yes	Yes	Yes
Selective insertion	Difficult	Possible	Possible	Possible	Possible	Difficult
Applicability	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	Microprocessors