Soft Error Resilient System Design through Error Correction

Subhasish Mitra[*], Ming Zhang[+], Norbert Seifert[+], TM Mak[+], Kee Sup Kim[+]

[*]Stanford University
[+]Intel Corporation

Abstract. This paper presents an overview of the Built-In Soft Error Resilience (BISER) technique for correcting soft errors in latches, flip-flops and combinational logic. The BISER technique enables more than an order of magnitude reduction in chip-level soft error rate with minimal area impact, 7-11% chip-level power impact, and 1-5% performance impact (depending on whether combinational logic error correction is implemented or not). In comparison, several classical error-detection techniques introduce 40-100% power, performance and area overheads, and require significant efforts in designing and validating corresponding recovery mechanisms. Design trade-offs associated with the BISER technique and other existing soft error protection techniques are also analyzed.

1 Who Cares about Soft Errors?

Soft errors are radiation-induced transient errors caused by neutrons generated from cosmic rays and alpha particles from packaging material. Traditionally, soft errors were only a major concern for space applications. That scenario has changed. Terrestrial radiation has been a growing concern, and many designs today implement extensive error detection and correction by way of Error Correcting Codes (ECC) mainly for on-chip SRAMs. However, memory protection alone is not enough for designs in sub-65nm technologies. Most future designs targeting enterprise computing and communication applications require soft error protection of latches and flip-flops, in addition to on-chip SRAMs. While combinational logic protection may not be an immediate necessity, it may eventually be required as more and more transistors are integrated in future technologies. There are multiple ways to minimize system-level soft error rate, applied at various levels of design hierarchy and manufacturing process.

The soft error rate of a design is generally quantified in terms of Failure-in-time, or FIT, where 1 FIT corresponds to one error per billion device hours. According to recent data discussed at the 2006 SELSE workshop (2006 IEEE System Effects of Logic Soft Errors Workshop, www.selse.org), a typical value for latch soft error rate may be assumed to be 10^{-3} FIT. Note that, there is a lot of variance in latch soft error rates depending on

specific latch designs. Assuming that a design contains 1 million flip-flops (and each flip-flop consists of two latches), the contribution of all flip-flops to the overall soft error rate of the design can be conservatively estimated as 1,000 FITs. In this estimate, a 50% latch timing vulnerability factor (TVF) [Ngyuen 03, Seifert 04] is assumed based on the fact that a latch is vulnerable to soft errors when it holds a logic value (i.e., when its clock input is 0).

Soft error rates of 1,000 FITs may not sound too high. However, it is not uncommon for enterprise systems to contain between 500 – 20,000 processors. For the 500 processor system, the system-level soft error rate contribution of the flip-flops will be 500,000 FITs (if our previously discussed design is a processor). This means, roughly once every 3 months some flip-flop in the system will be erroneous. For a system with 20,000 processors, the system-level soft error rate contributions of flip-flops will be 20 Million FITs – i.e., roughly once every 2 days there will be an error in some flip-flop of the system.

Fortunately, some soft errors do not have any impact on system operation. 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 errors in flip-flops can result in data corruption without being detected by the system or the user. As a result, system data integrity is compromised. This situation is referred to as *Silent Data Corruption (SDC)*, and is of great concern. Depending on the design and the application, between 10-40% of soft errors can result in SDC [Mukherjee 03, Nguyen 03, Wang 04]. Imagine the significance of SDC caused by a 1 to 0 bit flip in the most significant bit of the register storing the balance of a bank account.

Suppose that we optimistically assume that only 10% of soft errors cause system-level SDC. Continuing our previous analysis, for a 500-processor system, flip-flops will contribute to SDC roughly once in 30 months. For a 20,000-processor system, the latch contribution to system-level SDC is roughly once every 20 days. These numbers are unacceptable for enterprise system installations such as banks and stock markets. That is why future designs will require adequate protection to prevent such unacceptable situations.

SDC protection in terms of error detection alone is not enough. Suppose that we have a perfect way to detect all the soft errors that can potentially cause SDC. Once an error is detected, the system must recover from the detected error. If there is no user transparent way to recover it, it results into the so called Detected but Uncorrected Errors (DUE). Depending on how recovery is implemented, a part or the entire system may be down. (It is possible to implement efficient recovery in a transparent way without having to bring the entire system down [Spainhower 99].) Downtimes are very expensive in the order of $10K to $10M per hour [Hennessy 02]. Hence, it is not enough to simply employ error detection to prevent silent data corruption – it is absolutely necessary to ensure that system downtime is also minimized.

2 Soft Error Scaling Trend

The importance of soft error protection techniques is best understood by analyzing radiation-induced soft error rate trends for SRAM and logic over technology generations. Figure 1 shows the scaling trend of the soft error rate per SRAM memory cell for Intel designs. Alpha-particle and neutron induced soft error rates both show a clear decrease over the last two generations. This trend is consistent with what TI has also observed [Baumann 05]. Since SRAMs are typically protected by ECC for several reasons (soft errors, infant mortality, etc.), this trend does not have a major impact on most system designs targeting applications requiring high data integrity and availability.

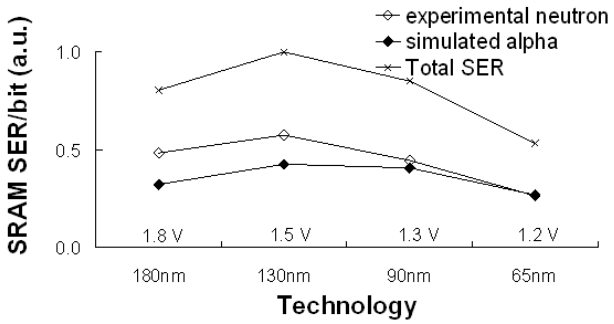


Figure 1. Technology trend of per bit SRAM soft error rates from Intel [Seifert 06].

In contrast to SRAM soft error rates, Baumann of TI [Baumann 05] has observed a steep increase in per latch soft error rate with technology scaling. Intel, on the other hand, has observed a relatively flat trend of per latch soft error rates for the last three generations (Fig. 2). In Fig. 2, the soft error rates of 20-30 most frequently used latches from Intel technology libraries elements are summarized (plotted are the mean and standard deviation). The soft error rate of an actual product depends on the use of specific kinds of latches from the technology library. Soft error rates of various latches in the same technology library can vary by more than an order of magnitude [Seifert 06].

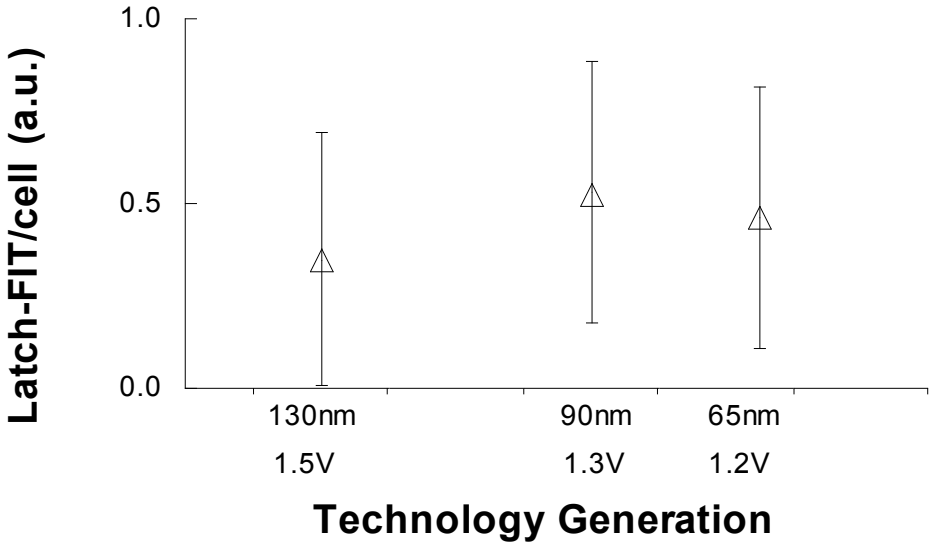


Figure 2. Latch per device relative error rates. Error bars indicate standard deviation within population of 20-30 selected library elements [Seifert 06].

Even if the soft error rate of a single latch or a single SRAM cell stays constant or increases over technology generations, chip-level soft error rates will increase significantly with technology scaling because of increased integration per constant area. We emphasize another soft error rate scaling trend that may be very important for future technology generations. Neutrons do not directly ionize Si but generate electron hole pairs via secondary ions created in neutron – Si spallation reactions. If those secondary ions generate sufficient charge over a region larger than a device, more than one single device maybe affected, creating a so-called multi-bit upsets (MBU). We call this phenomenon charge sharing and it can affect different devices or more than one node in one device. Figure 3 underlines that charge sharing among different SRAM cells is exponentially increasing with process scaling. This trend is expected to grow and we may not be able to ignore the effects of charge sharing in future designs, in particular for radiation hardened designs that are known to be immune to single node upsets only.

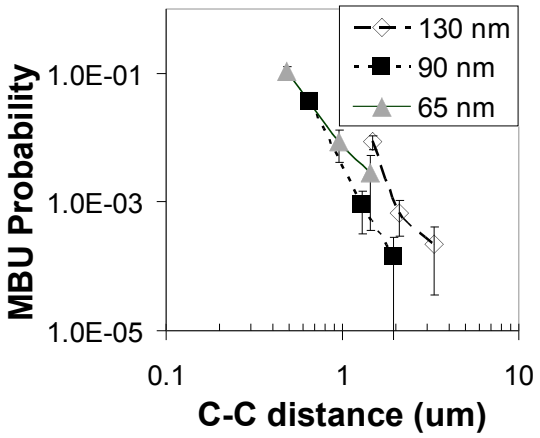


Figure 3. SRAM multiple bit upset probabilities plotted as a function of cell pitch [Seifert 06]. MBU probability is defined as the ratio of the number of MBUs to single bit upsets (SBUs).

3 How to Protect Systems from Soft Errors?

By now the need for logic soft error protection should be clear. The question is how future systems should be protected from soft errors. We will focus on *logic soft error* protection: soft errors in latches, flip-flops and combinational logic. Soft errors in SRAMs are protected using parity or Error Correcting Codes with interleaving (there are some open issues involving efficient error protection of small SRAM arrays and register-files).

Several techniques for logic soft error protection are available in the literature. Each of these techniques has its own advantages and disadvantages. The purpose of this section is to put together a set of metrics that can help distinguish these techniques and understand their pros and cons from an overall system design perspective. We hope that these metrics will help designers understand trade-offs associated with the adoption of one protection technique over another. The metrics are:

- **SDC reduction:** A technique that reduces silent data corruption by a small amount (e.g., by 50%) may be useful in helping a specific design meet its soft error rate goals, but is not scalable with increased integration in future technologies.
- **DUE reduction:** SDC reduction techniques can significantly increase DUEs. Consider a situation where every flip-flop in a design is checked for errors and recovery actions are initiated based upon types of errors detected. All errors that can cause SDC are detected for most practical purposes (double errors may not be

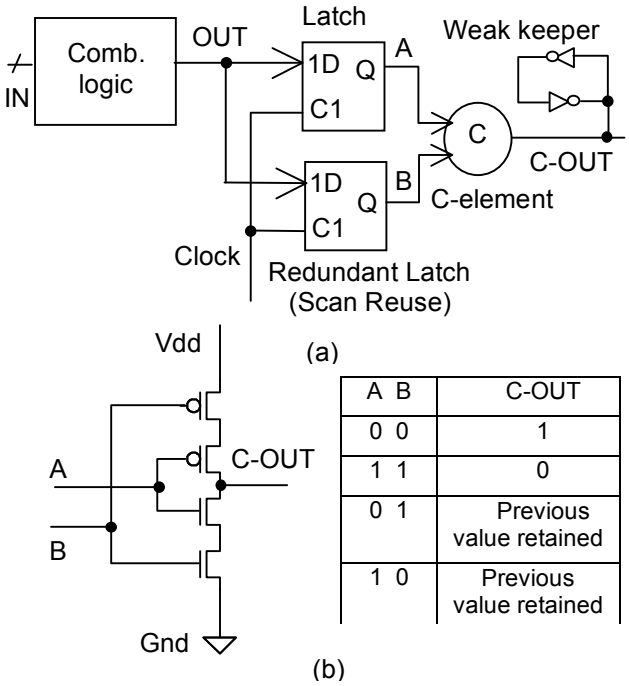
detected). However, this approach can significantly increase DUEs. Any error in any flip-flop manifests as a DUE even though only a portion of these errors will actually cause SDC.

- **Cost:** It is extremely important to understand power, performance and area penalties associated protection techniques.
- **Recovery mechanism design and validation effort:** Designing proper error recovery mechanisms and validating them are non-trivial tasks. Costs associated with the design and validation of recovery mechanisms can limit the advantages associated with soft error protection techniques.
- **Configurability:** Soft error protection in future technologies will be significantly impacted by the industry trend to reuse the same design for multiple applications with a wide range of power, performance and reliability requirements. For example, the use of a specific protection technique may incur acceptable power overhead for 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.
- **Applicability:** Several soft error protection techniques are optimized for specific applications such as processors, signal processing applications, etc. While such techniques are very useful, they have limited applicability for many designs.
- **Flip-flop and combinational logic protection:** It is desirable for protection techniques to address soft errors in both flip-flops and combinational logic using the same soft error protection technique. Otherwise, separate protection techniques for flip-flops and combinational logic introduce additional penalties and design complexity.

4 Built-In Soft Error Resilience (BISER) for Logic Soft Error Correction

We first illustrate the BISER technique for latch-based designs. We will also discuss the use of BISER for flip-flop based designs. Soft errors in latches are corrected using a C-element as shown in Fig. 4 [Mitra 05a, Mitra 05b]. During normal operation, when the clock signal Clock = 1, the latch input is strongly driven by the combinational logic and the latch is not susceptible to soft errors. This is illustrated in the Timing Vulnerability Factor discussion [Nguyen 03, Seifert 04]. When Clock = 0, C-OUT already has the correct value –any soft error in either latch will result in a situation where the logic value

on A will not agree with B. As a result, the error will not propagate to C-OUT and the correct logic value will be held at C-OUT by the keeper. The cost associated with the redundant latch is minimized by the reusing on-chip resources such as scan or scanout for multiple functions at various stages of manufacturing and field use [Mitra 05a, Mitra 05b].



A B	C-OUT
0 0	1
1 1	0
0 1	Previous value retained
1 0	Previous value retained

Figure 4. Latch error correction using C-element: (a): Overall technique; (b) C-element.

Extensive simulations in a sub-90nm process technology using a state-of-the-art simulation tool validated by radiation experiments [Nguyen 03] show that the design in Fig. 4 can achieve more than 20-fold reduction in the soft error rate compared to that of an unprotected latch. Note that, a soft error in the keeper does not have a major effect because the C-element output will be strongly driven by the latch contents assuming single error.

Fault injection simulations have been conducted on an Alpha-like microprocessor to evaluate the system-level effectiveness of the BISER technique for latch error correction. The results show that the BISER technique improves system-level soft error rate by 10 times over an unprotected design with negligible area or performance penalty and 7-11% power penalty [Zhang 06].

Soft errors in combinational logic can be corrected using two techniques – Error Correction using Duplication, and Error Correction using Time-Shifted Outputs. Figure 5 shows the soft error correction technique using duplication. Instead of comparing the contents of the latches storing duplicated outputs, we insert a C-element. This technique results in significant reduction (> 60-fold) in combinational logic soft error rate [Mitra 06]. Moreover, this technique also corrects soft errors in latches when Clock = 0. However, there can be significant cost – power and area costs of combinational logic duplication. The Error Correction using Time Shifted Outputs technique, described next, doesn't require combinational logic duplication, but imposes additional performance penalty.

The Time Shifted Outputs technique for error correction is shown in Fig. 6. This technique takes advantage of the fact that soft errors in combinational logic manifest as glitches. Instead of duplicating combinational logic, we sample the combinational logic output (OUT3), and a delayed version of OUT3 called OUT4. In Fig. 6, OUT3 is delayed by τ time units to obtain OUT4. The clock must be slowed down by τ units compared to Fig. 5. The latch outputs are connected to a C-element. The major advantage of the Error Correction using Time Shifted Outputs technique is that the power and area penalties incurred by the duplication scheme are minimized. Note that, τ is a design parameter that can be tuned based on the reliability requirement. Moreover, this technique also corrects soft errors in latches when Clock = 0. Simulation results in [Mitra 06] show that this technique can reduce combinational logic soft error rate by more than an order of magnitude when τ = 21ps. Note that the incremental power penalty of protecting combinational logic using the Time-shifted outputs technique over latch error correction is very little – less than approximately 7% of the power penalty for latch error correction.

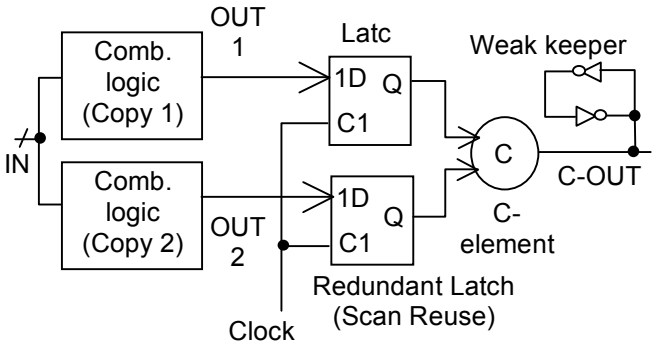


Figure 5. Combinational Logic Soft Error Correction using Duplication.

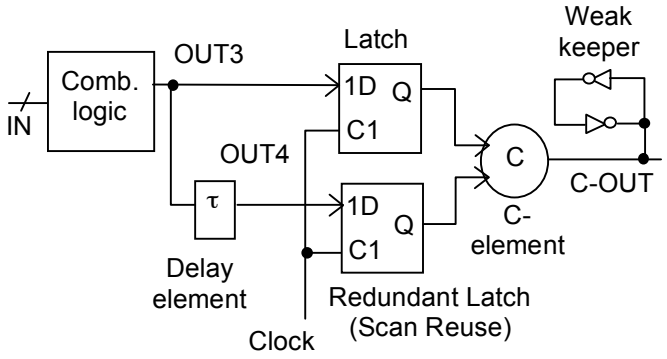


Figure 6. Combinational Logic Soft Error Correction using Time Shifted Outputs.

While the BISER technique has been illustrated for latch-based designs, it is also applicable for flip-flop based designs. Figure 7 shows flip-flop designs for the BISER techniques discussed earlier. Depending on whether duplication or time-shifted-outputs technique is used for combinational logic soft error correction, IN2 in Fig. 4b will be connected to the duplicated logic output (Fig. 5) or the delay element output (Fig. 6), respectively.

5 Comparison of Soft Error Protection Techniques

Table 1 presents a comparative analysis of the trade-offs associated with major soft error protection 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) BISER technique; (2) selective node engineering technique, which increases the capacitances of selective nodes of a circuit [Karnik 02]; (3) transistor sizing technique [Zhou 06]; (4) circuit hardening [Calin 96]; and, (5) classical hardware and time redundancy fault-tolerance techniques [Bartlett 04, Mukherjee 02, Oh 02a, 02b, 02c, Saxena 00].

The circuit-level comparison between BISER and circuit hardening techniques is conducted by a unified timing and power characterization methodology [Zhang 06]. While optimizing the various flip-flop designs, the objective is to match the timing parameter, D-to-Q delay. Several assumptions are made during the power measurement of all flip-flops: (1) the data activity factor (average number of output transitions per clock cycle) is 0.25; (2) low-to-high and high-to-low data transitions are equally likely. The cell layout areas are estimated by an internal tool at Intel, with a worst case error of 5% compared to real layouts. The SERs are obtained from an internal simulator at Intel.

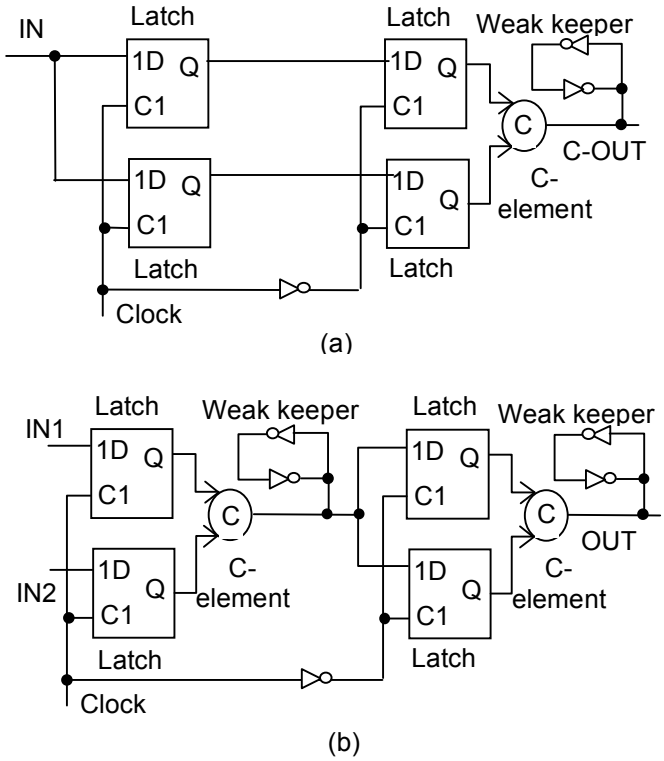


Figure 7. BISER for flip-flop based designs: (a): Flip-flop design for correcting soft errors in flip-flops; (b) Flip-flop design for correcting soft errors in flip-flops and combinational logic.

The selective node engineering technique, which increases the capacitances of selective nodes of a circuit, is an effective approach for designs requiring 30-50% undetected soft error rate reduction. For circuit hardening and BISER techniques, power overheads are derived based on an Alpha processor model with 10-fold chip-level soft error rate reduction [Zhang 06]. 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.

For the BISER technique, the power overhead is between 7-11%. 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., [Spainhower 99]), 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 even a worst-case scenario in which all flip-flops (rather than the subset of important flip-flops) are protected with BISER resulting in 14-22% power overhead. Moreover, time redundancy techniques have very significant performance overheads (40-200%) [Mukherjee 02, Oh 02a], and are mainly applicable for designs with well-defined architectures such as microprocessors.

Tables 1 and 2 imply that the BISER technique is most cost-effective for soft error protection. One major advantage of the BISER based error blocking technique is that it doesn't require any error recovery mechanisms and does not incur significant costs associated with the design and validation of recovery mechanisms.

6 Conclusion

The BISER technique is an efficient and practical way to design systems with built-in soft error correction. Comparative analysis with existing techniques demonstrates that the BISER technique combines the major benefits of circuit-level error correction and architectural techniques such as time redundancy and error detection, while avoiding their drawbacks. This is possible because the characteristics of soft errors are utilized by the BISER technique instead of general error models used by techniques such as duplication. This may limit the use of the BISER technique since all error sources may not have characteristics similar to radiation-induced soft errors.

Acknowledgment

We thank K. Ganesh, V. Zia, P. Shipley, J. Yang, S. Walstra, A. Vo, and J. Maiz from Intel Corporation for discussion and assistance during the course of this research. Prof. Subhasish Mitra is partially supported by DARPA / MARCO Gigascale Systems Research Center (GSRC).

Table 1. Comparative analysis of various soft error protection techniques: (a) Quantitative analysis; (b) Qualitative Analysis.

(a)

	BISER [Mitra 05a, 05b, 06, Zhang 06]	Transistor sizing [Karnik 02, Zhou 06]	Circuit hardening [Calin 96]	Hardware duplication [Bartlett 04]	Time redundancy [Mukherjee 02, Oh 02a, 02b, 02c, Saxena 00]
SDC reduction	**Latch: 20X Comb. Logic: 12-64X**	1.5X	Latch: 20X Comb. Logic: None	Almost all	Almost all
DUE reduction	**Latch: 20X Comb. Logic: 12-64X**	1.5X	Latch: 20X Comb. Logic: None	Increased DUE	Increase SUE
Power penalty (resilient mode)	**7-11%**	3%	12 - 18%	40 – 100%	> 40%
Power penalty (economy mode)	**1.5%**	3%	12 – 18%	Very little	Very little
Speed penalty	**Latch correction: 0 – 1% Comb. Logic correction: ~ 5%**	0-10.4%	0 – 1%	Very small	50%
Area penalty	**Die size increase not expected**	Die size increase not expected	Die size increase not expected	40 – 100%	Die size increase not expected
Recovery design & validation efforts	**None**	None	None	Significant	Significant
Configurability	**Yes**	No	No	Yes	Yes
Applicability	**General**	General	General	General	Processor designs
Flip-flop and comb. Logic protection	**Both**	Both	Latches & flip-flops only	Both	Both

(b)

	BISER [Mitra 05a, 05b, 06, Zhang 06]	Transistor sizing [Karnik 02, Zhou 06]	Circuit hardening [Calin 96]	Hardware duplication [Bartlett 04]	Time redundancy [Mukherjee 02, Oh 02a, 02b, 02c, Saxena 00]
SDC reduction	**Latch: A Comb. Logic: A**	C	Latch: A Comb. Logic: F	A+	A+
DUE reduction	**Latch: A Comb. Logic: A**	C	Latch: A Comb. Logic: F	D	D
Power penalty (resilient mode)	**B**	A	B	D	C
Power overhead (economy mode)	**A-**	B	C	A	A+
Speed penalty	**Latch correction: A Comb. Logic correction: B**	B	A	A	D
Area penalty	**A**	A	A	C	A
Recovery design & validation efforts	**A+**	A+	A+	D	D
Configurability	**A+**	D	D	A+	A+
Applicability	**A+**	A+	A+	A+	C
Flip-flop and comb. Logic protection	**A+**	A+	D	A+	A+

References

[Bartlett 04] Bartlett, W., and L. Spainhower, "Commercial Fault Tolerance: A Tale of Two Systems," *IEEE Trans. Dependable and Secure Computing*, Vol. 1, No. 1, pp. 87-96, 2004.

[Baumann 05] R.C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies", *IEEE Transactions on Device and Materials Reliability*, Vol. 5, No. 3, pp. 305 – 316, 2005.

[Calin 96] Calin, T., M. Nicolaidis, and R. Velaco, "Upset Hardened Memory Design for Submicron CMOS Technology," *IEEE Trans. Nucl. Sci.,* Vol. 43, pp. 2874-2878, Dec. 1996.

[Hazucha 00] P. Hazucha, C. Svensson, "Impact of CMOS technology scaling on the atmospheric neutron soft error rate", *IEEE Transactions on Nuclear Science*, Vol. 47, No. 6, pp. 2586 – 2594, 2000.

[Hennessy 02] Hennessy J., and D. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed., Morgan Kaufmann, 2002.

[Karnik 02] T. Karnik, *et al*., "Selective Node Engineering for Chip-level Soft Error Rate Improvement," *Proc. VLSI Circuits Symp.*, pp. 204-205, 2002.

[Mitra 05a] Mitra, S., 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.

[Mitra 05b] Mitra, S., M. Zhang, T.M. Mak, N. Seifert, V. Zia and K.S. Kim, "Logic Soft Errors: A Major Barrier to Robust Platform Design," *Proc. Intl. Test Conf.*, 2005.

[Mitra 06] Mitra, S., M. Zhang, N. Seifert, B. Gill, S. Waqas and K.S. Kim, "Combinational Logic Soft Error Correction," *Proc. Intl. Test Conf.*, 2006, to appear.

[Mukherjee 02] Mukherjee, S., M. Kontz, S. Reinhardt, "Detailed Design and Evaluation of Redundant Multithreading Alternatives," *Proc. Intl. Symp. Computer Architecture*, 2002.

[Mukherjee 03] Mukherjee S., et al., "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor," MICRO, 2003.

[Nguyen 03] Nguyen, H.T., and Y. Yagil, "A Systematic Approach to SER Estimation and Solutions", *Proc. Intl. Reliability Physics Symp.*, pp. 60 – 70, 2003.

[Oh 02a] Oh, N., P.P. Shirvani and E.J. McCluskey, "Error Detection by Duplicated Instructions in Super-Scalar Processors," *IEEE Trans. Reliability*, Vol. 51, No. 1, pp. 63-75, March 2002.

[Oh 02b] Oh, N., P.P. Shirvani and E.J. McCluskey, "Control-Flow Checking by Software Signatures," *IEEE Trans. Reliability*, Vol. 51, No. 1, pp. 111-122, March 2002.

[Oh 02c] Oh, N., S. Mitra and E.J. McCluskey, "ED4I: Error Detection by Diverse Data and Duplicated Instructions," *IEEE Trans. Computers, Special Issue on Fault-Tolerant Embedded Systems*, Vol. 51, No. 2, pp. 180-199, Feb. 2002.

[Saxena 00] Saxena, N.R., S. Fernandez Gomez, W.J. Huang, S. Mitra, S.Y. Yu and E.J. McCluskey, "Dependable Computing and On-line Testing in Adaptive and Reconfigurable Systems," *IEEE Design and Test of Computers*, pp. 29-41, Jan-Mar 2000.

[Seifert 04] Seifert N., and N. Tam, "Timing Vulnerability Factors of Sequentials", *IEEE Trans. Device and Materials Reliability*, Vol. 4, No. 3, p. 516-522, September 2004.

[Seifert 06] N. Seifert, *et al*., "Radiation-induced Soft Error Rates of Advanced CMOS Bulk Devices," *IEEE International Reliability Physics Symposium*, pp. 217 – 225, 2006.

[Spainhower 99] Spainhower, L., and T.A. Gregg, "S/390 Parallel Enterprise Server G5 Fault Tolerance," *IBM Journal Res. and Dev.*, Vol. 43, pp. 863-873, Sept./Nov., 1999.

[Walstra 05] S.V. Walstra and Changhong Dai, "Circuit-level modeling of soft errors in integrated circuits", *IEEE Transactions on Device and Materials Reliability*, Vol. 5, No. 3, 2005, pp. 358 – 364.

[Wang 04] N. 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.

[Zhang 06] M. Zhang, *et al*., "Sequential Element Design with Built-In Soft Error Resilience," *IEEE Trans. VLSI*, Vol. 14, No. 12, pp. 1368 – 1378, 2006.

[Zhou 06] Q. Zhou, and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Trans. CAD,* Vol. 25, No. 1, pp. 155-166, Jan. 2006.