Robust System Design with Built-In Soft-Error Resilience

Transient errors caused by terrestrial radiation pose a major barrier to robust system design. A system's susceptibility to such errors increases in advanced technologies, making the incorporation of effective protection mechanisms into chip designs essential. A new design paradigm reuses design-for-testability and debug resources to eliminate such errors.

Subhasish Mitra
Norbert Seifert
Ming Zhang
Quan Shi
Kee Sup Kim
Intel

Soft errors, also called single-event upsets (SEUs), are radiation-induced transient errors caused by neutrons from cosmic rays and alpha particles from packaging material.

Traditionally, soft errors were regarded as a major concern only for space applications. Yet, for designs manufactured at advanced technology nodes—such as 90 nm, 65 nm, and onward—system-level soft errors are much more frequent than in the previous generations.

Further, customers demand stringent limits on soft-error rates for enterprise servers and networking hardware. All these chips, sometimes hundreds or thousands of them, must operate correctly, with very high system data integrity and availability. An IT executive quoted in *Forbes Magazine*[1] expressed how customers feel when the hardware fails to meet expectations: "It's ridiculous. I've got a $300,000 server that doesn't work. The thing should be bulletproof." That is why digital-system soft errors have received significant attention.[1,2]

The soft-error rate of a system generally is measured in units of Failures in Time, or FIT. A soft-error rate of 1 FIT means that the mean time before an error occurs is a billion device hours. IBM sets its target for undetected errors caused by SEUs at 114 FITs,[3] which would require a mean time before an SEU causes an undetected error of roughly 1,000 years.

The high data-integrity and availability requirements for servers and networks[4] make soft errors an extremely important design aspect for microprocessors, network processors, high-end routers, and network storage components. Thus, soft-error protection is just as important as other product characteristics such as performance, power consumption, yield, and test quality.

Chip designers must address soft errors very early, starting from the product definition phase and continuing through the architecture planning, circuit design, logic design, and postlayout phases.

Designers routinely use well-known techniques such as error detection and correction to cope with soft errors in static random access memory. Protecting SRAMs isn't enough, however, given the soft-error rates and customer expectations. Designers must evaluate the effects of soft errors in flip-flops, latches, and combinational logic, and effective protection mechanisms must be incorporated into the design.

SYSTEM-LEVEL SOFT-ERROR-RATE ESTIMATION

The soft-error rate (SER) of a design can be expressed in terms of the nominal soft-error rates of individual elements such as SRAMs, sequential ele-

Soft-Error Testing: Key Points

Michael Nicolaidis and Damien Chardonnereau, iRoC Technologies

Following a strategy similar to traditional burn-in for general-reliability purposes, soft-error testing seeks to reproduce and then accelerate the die's real-life environment. Researchers use a neutron beam accelerator and alpha foils to conduct this testing.

Because each neutron beam has a specific and complex set of neutron properties, the beams must be carefully qualified to correlate the resulting data with real-time results. Beam qualification includes factors such as energy, spectrum, fluency, and tail-effect correction.

Likewise, the actual die tester also must be specifically designed for portability, ruggedness, flexibility, and dynamic testing.

These issues and the effort required to access a neutron beam facility have prompted many companies to outsource this work to a soft-error test consolidator. Doing so gives companies more test-schedule flexibility, lowers the total costs of soft-error testing, and strengthens their SER data value through test independence.

Environmental acceleration

Real-time testing offers another means for accurate soft-error rate detection. However, given that neither single-event upsets nor soft-error-induced latch-ups occur frequently, testers employ environmental acceleration, such as testing at high altitudes where the neutrons' flux is stronger while the spectrum remains equal to that at ground level. Table A shows the advantages of accelerated testing over real-time testing.

Consider, for example, the Jungfraujoch lab in Switzerland. Located at 11,000 feet, the facility can accelerate sea-level test times by a factor of 11. In testing conducted at this lab, iRoC Technologies obtained a statistically significant number of soft errors on different devices over a period of 4 to 6 months. This test for soft-error rates covers all different phenomena, including multibit upsets.

SER trends

Predicting the soft-error rate and its impact on a specific die has always challenged physics experts. Many parameters influence SER, which is statistical in nature.

As processes migrate to nanometer scale, the reduction in activation energies and the increased amount of embedded memory will cause soft errors to become an issue that designers must deal with. Even as the per-unit FIT rate stabilizes with advanced processes, system-level soft errors have been increasing.

iRoC Technologies has performed more than 1,000 SER analyses on different process nodes and devices. This work has revealed a clear trend for SRAM/CAM: The average FIT per megabyte slightly decreases at each process node, through to 130 nm. From that point down to 90 nm, the FIT per megabyte begins to stabilize.

Even with stabilization, however, researchers must consider three additional trends:

- Several neutron-induced latch-ups have been observed in nanometer memory devices.
- Multibit upsets have been observed more frequently.
- SEU-rate dispersion becomes more significant at 90 nm than at 130 nm, indicating that SER is both a fixed element driven by a process and an element affected by design methodology.

Silicon test results show that the average soft-error rate hovers around 1,000 FIT per megabit (neutron + alpha). The small expected FIT-per-megabit decrease per process node will not counteract the significant amount of memories designers expect to embed in future SoCs. In addition, as designs move to newer nodes, the logic elements in the design will become more sensitive.

Techniques must be put into place that will ensure developers take this new sensitivity into consideration.

Michael Nicolaidis *is a cofounder of iRoC Technologies and the company's chief technology officer.*

Damien Chardonnereau *is a project leader and product manager for iRoC Technologies.*

Table A. Accelerated testing versus real-time testing.

Test type	Logistics	Time	Accuracy	Devices under test
Accelerated	Complex: Requires qualified beams access; expert team required	Average: 2 to 3 months	Good	Memories, SoC, FPGA systems level
Real-time	Reasonable	Average: 4 to 6 months	Excellent	All types

ments such as flip-flops and latches, combinational logic, and factors that depend on the circuit design and the microarchitecture,[5,6] as follows:

$$SER^{design} = \sum_i SER_i^{\text{nominal}} \times$$

(Probability error in i^{th} circuit element produces system-level error)

In this expression, SER_i^{nominal} refers to the soft-error rate of the ith circuit element—for example, an SRAM cell, flip-flop, or latch—under static conditions when all inputs and outputs of the element are constant, independent of the system that uses the element.

The SER_i^{nominal} term is generally estimated using radiation testing and circuit simulation tools. The

"Soft-Error Testing: Key Points" sidebar provides more details about these calibrations. The *timing vulnerability factor* (TVF_i) and *architectural vulnerability factor* (AVF_i) of circuit element *i* determine the probability component in the preceding expression, as follows:

Probability error in i^{th} element produces system-level error = $TVF_i \times AVF_i$

The TVF of circuit element *i*, TVF_i, also called the timing derating,[5] is defined as the fraction of time the element is susceptible to SEUs that will cause an error in that element.

For example, consider the simple D-latch in Figure 1. When the clock input of the D-latch is 1, the upstream combinational logic drives the latch's D-input and writes the corresponding logic value into the latch. During this time, any SEU that affects the transistors inside the latch has a negligible effect because the correct value is being driven at the D-input.

However, when the clock input of the D-latch is 0, an SEU affecting transistors, such as those with drains connected to nodes S and F, can flip the latch content. Thus, the latch is susceptible to an SEU that can cause an error during the fraction of the total clock period for which the clock signal is 0, which is the TVF of this latch.

If the clock duty cycle is 50 percent for a flip-flop-based design, the TVF of an individual D-latch inside a flip-flop is 50 percent. A latch's TVF can be less than 50 percent, however.[6] The TVFs of SRAMs are very close to 1.

A glitch induced in the static combinational logic by an SEU must arrive at the destination sequential element within its setup and hold time window to create an error in that sequential element. The TVF of combinational logic is impacted by the clock speed and number of gates located between the node where the glitch is induced and the destination sequential element. Since the setup time and hold times of a sequential element are independent of the clock speed, the TVF of static combinational logic increases with increasing clock frequency.

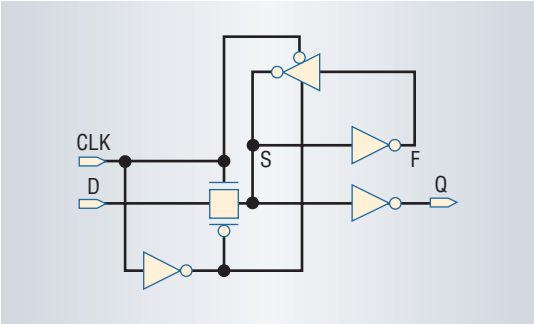

Figure 1. A D-latch. When the clock (CLK) input is 0, a single-event upset affecting transistors, such as those with drains connected to nodes S and F, can flip the D-latch's content, causing an error.

The architectural vulnerability factor of the *i*th circuit element, AVF_i, also called logic derating,[5] is the probability that an error in an element results in a system-level undetected error. AVF values depend on the design's architecture and input stimulus. Consider the following two simple examples.

First, suppose that a flip-flop's content is erroneous. However, if the flip-flop output is ANDed with another signal whose logic value is 0, the error will have no effect.

Second, suppose that an error affects a register holding the operand of an instruction in a microprocessor with speculative execution. If this instruction is executed speculatively and becomes a dead instruction later, this error will not affect the results produced by the program the microprocessor executes. Table 1 summarizes various AVF estimation approaches.

Table 1. Architectural-vulnerability-factor (AVF) estimation approaches.

Approach	Description	Major issues	Advantages	Disadvantages
Manual	–	• No systematic analysis	–	• Subjective, error-prone, time-consuming, difficult quantitative justification
Fault injection[7,8]	Inject error(s) and simulate to see if injected error(s) causes system-level error(s) by comparing the system response with simulated fault-free response	• What inputs to simulate • How many errors to inject • Which signals to inject errors to • Which signals to use for comparison	• Applicable to any design • Easy automation	• Long simulation time (several days or weeks) for statistically significant results • Dependence on chosen stimuli
Fault-free simulation[5,9]	Perform architectural or logic simulation and identify situations that do not contribute to system-level errors, such as unused variables and dead instructions	• What inputs to simulate • How to identify situations that do not contribute to system-level errors	• Much faster compared to fault injection • Easy automation	• Applicable to very specific designs and not general enough • Dependence on chosen stimuli

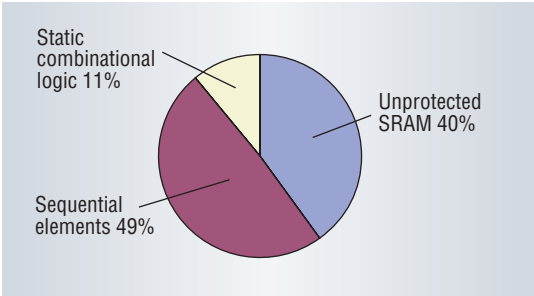

Figure 2. Contributions to the overall soft-error rate for a design manufactured using state-of-the-art technology.

Figure 2 shows the estimated soft-error-rate contributions of various elements for typical designs such as microprocessors, network processors, and network storage controllers. This analysis includes both the TVFs and AVFs of the individual elements.

The soft-error-rate contribution of combinational logic for state-of-the-art processes is still considerably smaller compared to the contributions of unprotected SRAMs and sequential elements such as latches and flip-flops.

Designers routinely use parity or error-correcting codes (ECC) to protect large memories and register files. For applications requiring high data integrity and availability, the unprotected memories usually represent a small percentage of total memory bits. These memories are composed of small memory arrays for which parity or ECC is useful, but expensive.

For the design used in Figure 2, the combined soft-error-rate contribution of sequential elements and combinational logic exceeds that of the unprotected SRAMs. Hence, special attention is required to develop techniques for protecting nonSRAM portions of a design from soft errors.

TECHNOLOGY TRENDS

Several experimental and theoretical studies have demonstrated that the nominal soft-error rate of an SRAM bit, built with state-of-the-art processes, has been saturating or even decreasing for both bulk CMOS and SOI technologies.[10,11] For latches and flip-flops, available data in the literature shows less consistency than that for SRAMs. Robert Baumann[11] observed that the nominal soft-error rates of sequential elements increase with technology scaling. At Intel, however, we have observed a different trend for some of our latches. The nominal soft-error rates for some latches are fairly constant or even decreasing slightly for the 130-nm to 65-nm technologies.[10]

The AVFs and TVFs do not change significantly with technology generations.[6] As Figure 2 shows,

Soft-Error Protection: Test Results

Michael Nicolaidis and Damien Chardonnereau, iRoC Technologies

iRoC Technologies has optimized, designed, and manufactured different test chips and processor cores to characterize the tradeoffs between various soft-error protection design schemes.

The company designed 32-bit and 8-bit RISC cores implementing memory-protection and logic-time redundancy techniques. These two silicon test cases validated that logic is sensitive to soft errors and that the design process can detect, isolate, and eliminate soft errors.

SPARC efforts

iRoC Technologies has optimized RoC-S81, an example of soft-error detection based on time redundancy, by inserting fault-tolerant mechanisms into the European Space Agency's LEON SPARC processor design.[1] In addition to code-correction techniques implemented in its memory blocks, the processor includes a time-redundancy detection technique for logic blocks (no correction).

Using radiation testing to compare the RoC-S81 with the original LEON design showed the RoC-S81's integrated fault-tolerant mechanisms to be efficient, although its logic parts proved to be sensitive to strikes and propagated transients.

The developers used a dedicated design scheme to estimate a transient on-chip pulse width versus the particle's energy, validating the ability to, detect within logic blocks, transient pulse width. Figure A shows this process in action, as an ion striking a transistor causes a transient fault to become a soft error.

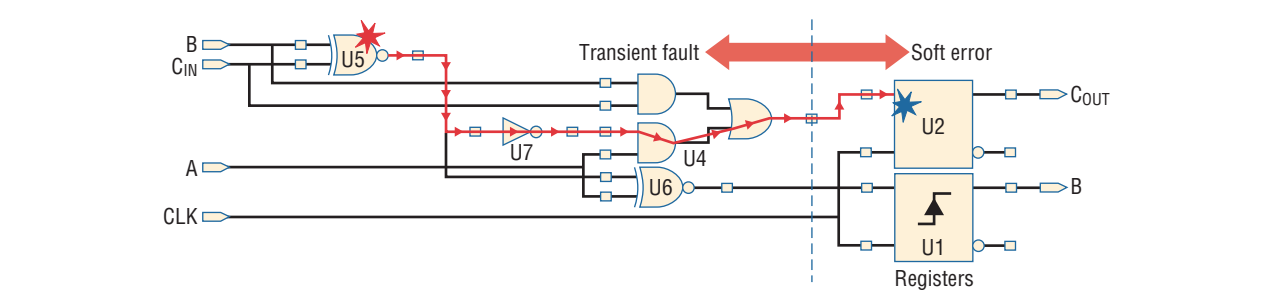

Figure A. Soft-error chain. An ion striking a transistor causes a transient fault to become a soft error.

Given that transient pulse propagation depends on the technology node and pulse width, understanding what energies atmospheric neutrons can generate when colliding with a silicon atom becomes essential. Neutrons striking silicon can generate any of more than 100 different nuclear reactions. Complete knowledge of the various combinations is necessary to identify the pertinent pulse characteristics and allow accurate fault injection, making an SER logic contribution possible.

Even if protecting the chip's memories brings a significant improvement in fault tolerance, time-out or application errors could still occur in the nonprotected logic blocks, whose contribution to the overall SoC soft-error rate ties directly to the particle's energy. An average of 10 calculation errors per test cycle have been observed in both chips without logic block correction, only detection.

CoolRISC

Based on the CoolRISC core from CSEM (the Swiss Center for Electronics and Microtechnology; www.csem.ch), iRoC Technologies developed and manufactured, for the French Space Agency (CNES), the RoC-CR11 in 180-nm silicon, implementing soft-error detection and correction on both the logic block and memory blocks. The company also manufactured a nonprotected version of CoolRISC.

Both chips integrate an 8-bit logic core block, a memory controller for external and internal memory, embedded program and data memory blocks, and some external interfaces. After manufacturing, these two chips were radiation tested to assess the nonprotected CoolRISC's sensitivity and the efficiency of the protection implemented in the RoC-CR11.

Memory blocks protection and test

The CoolRISC and the RoC-CR11 contain 200 Kbits of embedded SRAM. The protection techniques implemented on the RoC-CR11, based on iRoC's specific methodology for error-corrected code, share the correction code among the different 8-bit memory words to save area.

The RoC-CR11 also implemented an error-detection signal to monitor the error-correction mechanisms. Protecting 100 percent of the memory required a total area overhead of 29 percent; an ECC solution would have required an overhead of 50 percent.

Both chips underwent static and dynamic tests to measure the efficiency of iRoC's soft-error protection techniques. Among the different tests performed, the RoC-CR11 detected and corrected all 80 single-bit errors in its memories, while the unprotected CoolRISC incurred 90 single-bit errors.

Logic blocks protection and test

The CoolRISC and RoC-CR11's logic blocks are latch-based designs. This implies that all the registers are implemented as latches, not flip-flops. This means that the design works by using two nonoverlapping gated clocks, which provides a power-efficient implementation.

Developers designed the RoC-CR11's soft-error detection based on iRoC's patented time redundancy schemes. Heavy ion radiation testing (more stressful than neutron beams) demonstrated that the implemented protection technique provided 100 percent protection.

During the radiation testing, both the nonprotected CoolRISC and the protected RoC-CR11 underwent beam radiation at the same time. For a given application test and a fluency of 1.1e7, the CoolRISC's chip output showed 60 errors. For the same application test and a fluency of 1.5e6—10 times more fluency—the RoC-CR11's chip output showed no errors. The RoC-CR11 also implemented error detection and uncovered 148 errors in its memories and 9 errors in the logic—all of which were corrected.

Developers created different applications to run on the two processors to test both the memory and logic blocks. All tests showed the same results: The nonprotected CoolRISC showed a significant number of errors, whereas the RoC-CR11 showed no die output errors.

The time-redundancy implementation resulted in a 90 percent area overhead for achieving both error detection and correction in the logic elements. This compared to a projected 200 percent overhead area penalty using a more traditional time-redundancy approach. Using optimized ECC protection for memories and time redundancy for logic blocks showed no visible performance penalty.

Designers must consider this significant overhead for logic protection within the overall logic-to-memory ratio in modern chips, where logic might represent only 20 percent of the die and the final application—networking, telecom, or consumer application—doesn't need 100 percent protection.

Simulating soft errors and pinpointing design hotspots will optimize soft-error protection to meet end-user reliability requirements.

Moving forward

Soft errors now form part of the design challenge because, like any other design constraint, there is a tradeoff between this variable and application requirements. At 90 nm and beyond, all parts of a SoC are soft-error sensitive. Reaching the 100 FIT per device target will require an in-depth understanding of the soft-error chain.

As with all other design variables, optimization is essential. A 100 percent soft-error protection rate is not truly needed and is too expensive for most ground-level applications.

Making the most efficient tradeoff choices early in the design phase requires a predictive methodology. An SER prototyping and optimization tool well integrated in the current design flow will help designers and business unit managers make strategic decisions such as library and memory choices or even process or foundry choices.

Reference

1. D. Chardonnereau et al., "32-Bit RISC Processor Implementing Transient Fault-Tolerant Mechanisms and its Radiation Test Campaign Results," *Single-Event Effects Symp.*, NASA, Apr. 2002.

Michael Nicolaidis *is a cofounder of iRoC Technologies and the company's chief technology officer.*

Damien Chardonnereau *is a project leader and product manager for iRoC Technologies.*

Table 2. Comparison of various soft-error protection techniques.

			Time redundancy		
Parameters	**Circuit-level hardening**	**Hardware redundancy**	**Software-implemented hardware fault tolerance**	**Multithreading techniques**	**Multistrobe**
Technique description	Special circuit-level design techniques to decrease implemented circuits' inherent vulnerability to soft errors.[12]	Classical techniques such as triple modular redundancy (TMR) and concurrent error detection, such as duplication, parity prediction, low-cost techniques for matrix operations, and lossless data compression[13]	Program instructions executed twice and results compared to detect errors; program control-flow errors detected using special control-flow checking techniques[14,15]	Same instruction sequence executed using two threads, then results compared to detect any errors[16,17]	Errors detected and corrected by strobing outputs of the same combinational logic block multiple times by delayed clocks[18]
Undetected errors	Yes	Minimal	Minimal	Minimal	Yes
Errors logged	No	Yes	Yes	Yes	Yes
Technology dependence	Yes	Very little	Very little	Very little	Yes
Extra effort for recovery	No	Yes, unless TMR used	Yes, unless TMR used	Yes, unless TMR used	Yes, unless TMR used
Integration with design flow	Simple	Complex, recovery required	Complex, recovery required	Complex, recovery required	Complex, recovery required
Area overhead	Yes	Yes	None	Some	Yes
Performance overhead	Minimal	Minimal	Yes, 40 to 200 percent	Yes, about 20 to 40 percent	Minimal for error detection, can be significant for error correction
Power overhead	Yes	Yes	Yes	Yes	Yes
Selective insertion	Possible	Possible	Difficult	Difficult	Possible
Areas protected	Mainly sequential elements	Sequential elements and combinational logic	Sequential elements and combinational logic	Sequential elements and combinational logic	Sequential elements and combinational logic
Architectural impact	Minimal	Yes	None	Yes	Yes
Applicability	Unlimited	Unlimited	Mainly microprocessors	Mainly microprocessors	Unlimited

the SER contribution of combinational logic for state-of-the-art processes is still considerably smaller compared to contributions of unprotected SRAMs and sequential elements. Hence, the chip-level SER trend is dominated by the SER trends of SRAMs and sequential elements such as latches and flip-flops.

Even if the SER per SRAM bit or latch remains constant over technology generations, integration of more devices in advanced technologies results in higher chip-level SER. In contrast, customer expectations for SERs will either remain constant or become more stringent in advanced technologies.

SOFT-ERROR PROTECTION TECHNIQUES

Designers can use several strategies to provide soft-error protection. These include circuit-level hardening, classical hardware redundancy, and time redundancy techniques.

The "Soft-Error Protection: Test Results" sidebar discusses radiation testing of some soft-error protection techniques. Table 2 shows a comparative analysis of these techniques with respect to several system-level metrics, exploring some variables and factors that determine their applicability to actual designs.

REUSE PARADIGM FOR BUILT-IN SOFT-ERROR RESILIENCE

A new paradigm that leverages the reuse of on-chip resources for multiple functions at various stages of manufacturing and field use can overcome the drawbacks of existing soft-error protection techniques. For example, designers can reuse

on-chip scan design-for-testability resources for soft-error protection during normal operation.

Scan design for testability has become a de facto test standard because it provides an automated solution to high-quality production testing. In addition, scan is extremely valuable for postsilicon debug activities[19,20] because it provides access to an integrated circuit's internal nodes.

Figure 3 shows a microprocessor scan flip-flop design[20] that comprises two distinct circuits: a system flip-flop and a scan portion. All scan flip-flops in a design are connected together as one or more shift registers. The SI input of a scan flip-flop is connected to the SO output of the preceding scan flip-flop in the shift register. The SO output of a scan flip-flop is connected to the SI input of the following scan flip-flop in the shift register. The structure of the scan portion of Figure 3 is similar to the system flip-flop, with the addition of interface circuits to move data between the system flip-flop and the scan portion, as well as shifting the test pattern and test response, as required by the specific scan architecture.

This design has two operation modes: normal-system operation and test. In the test mode, clocks SCA and SCB are applied alternately to shift a test pattern into latches LA and LB. Next, the UPDATE clock is applied to move the contents of LB to PH1. Thus a test pattern is written into the system flip-flop.

Next, functional clock CLK is applied, which captures the system response to the test pattern. Finally, the CAPTURE signal is applied to move the contents of PH1 to LA. The system response is then shifted out by alternately applying clocks SCA and SCB. During normal system operation, the scan portion is shut off by asserting logic-0 values to the scan signals (SCA, SCB, UPDATE, and CAPTURE).

There are three basic reasons for using the scan style of Figure 3: structural testing using automated test pattern generation tools, functional testing using signature analysis, and efficient postsilicon debug.[18]

The opportunity for scan reuse for soft-error protection arises from the redundant scan resources—latches LA and LB in Figure 3—that are unused during normal operation, but add to the occupied area of the chip and the leakage power during normal operation.

Figure 4 shows how reusing the scan flip-flop design can reduce the impact of soft errors that affect latches. The flip-flop design's test mode operation is identical to the design in Figure 3. In normal system operation mode, the scan clocks SCA, SCB, UPDATE, and TEST are forced low, while the

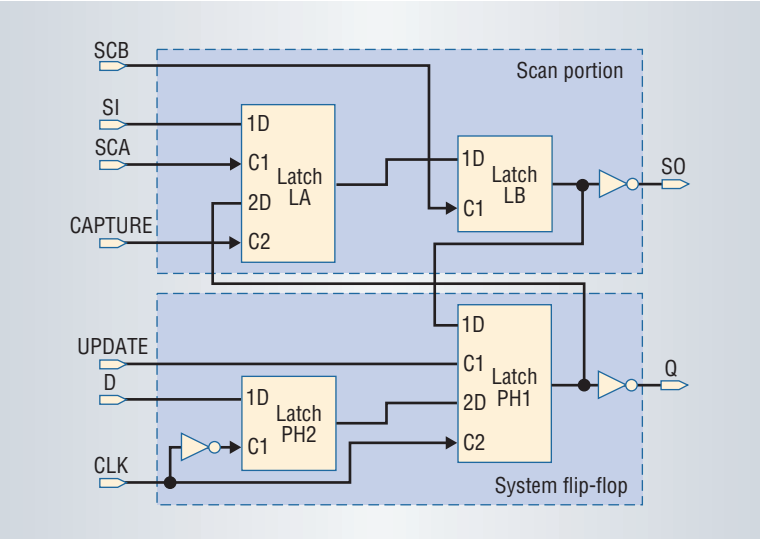

Figure 3. Microprocessor scan cell design. The design has two operation modes: normal-system operation and test.

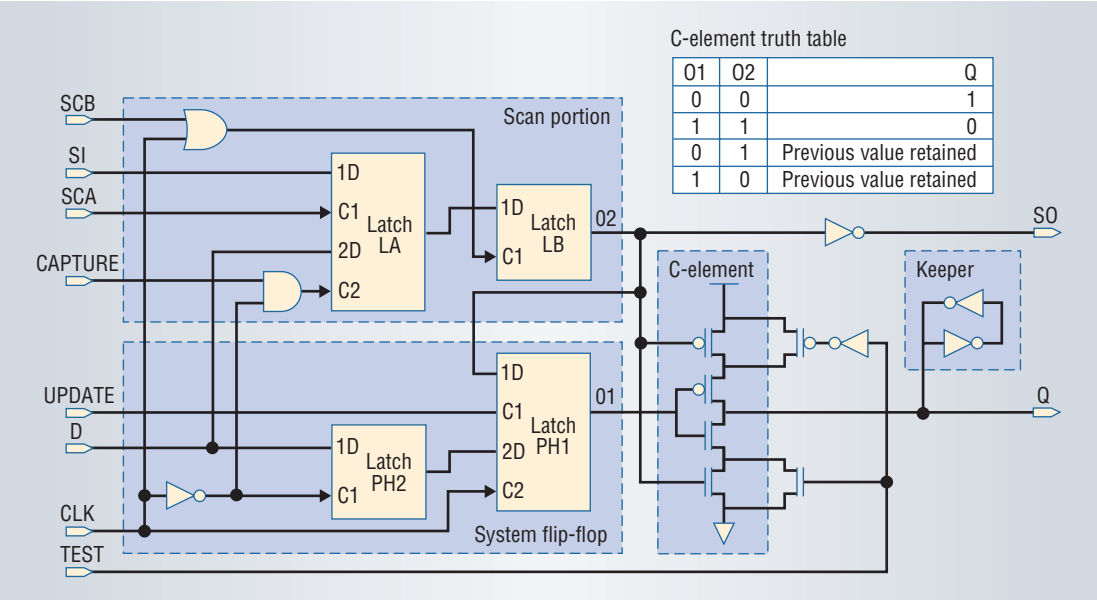

Figure 4. Scan reuse. Soft-error-blocking flip-flop design with a C-element. Reusing the scan flip-flop reduces the impact of soft errors that affect the latches by more than 20 times.

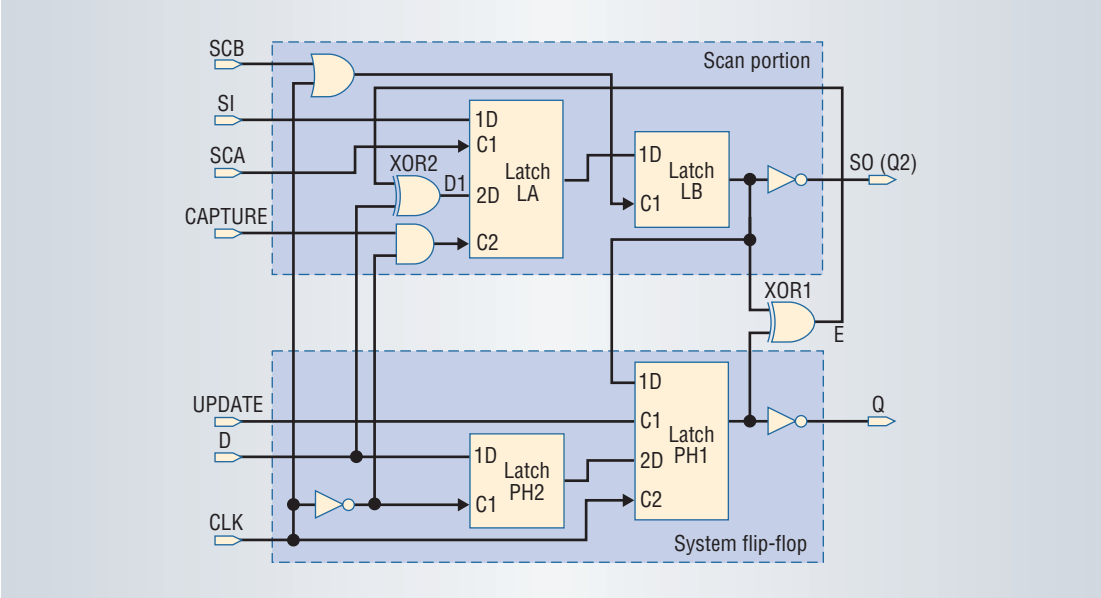

Figure 5. Error-trapping scan cell design. Latches LA and LB store redundant copies of PH2's and PH1's contents, respectively, during normal operation. A soft error in any latch causes the error signal (E) to be 1. Once E is 1, the logic values stored in LA and LB become complements of the contents of PH2 and PH1, respectively, and E continues to be 1, trapping the error until another soft error affects one of the latches, which rarely occurs.

CAPTURE signal is forced high. This converts the scan portion into a master-slave flip-flop that operates as a shadow of the system flip-flop.

During normal operation, when the clock signal CLK is 0, the C-element output drives flip-flop output Q, and the chip transfers the logic value at input D into latches LA and PH2. During this time, latches PH1 and LB are susceptible to soft errors because their clock inputs are 0 and they are holding logic values. If a soft error occurs in PH1 or LB, the logic value on O1 will not agree with O2. As a result, the error will not propagate to output Q, and the keeper will hold the correct logic value at Q. A soft error in PH2 or LA when CLK = 1 produces similar results. Depending on the system's speed and the leakage current, the keeper in Figure 4 might not be necessary.

Extensive SER simulations on an advanced process technology using an internal tool[5] show that this design can reduce the SER by more than 20 times compared to the error rate for an unprotected flip-flop.

Any soft error affecting a single latch inside a flip-flop is guaranteed to be detected by a self-checking scan flip-flop that is obtained by removing the C-element and the associated keeper structure from the design in Figure 4. Various self-checking scan cells choices are possible.

During normal operation, at least one copy of correct data exists, under the assumption of a single error in a latch. To perform self-checking, the approach implements error-detection circuits such as equality checkers that compare the Q and Q2 outputs of all such flip-flops in a design and indicate an error each time a mismatch occurs.

A major drawback of such a self-checking approach is the significant amount of area occupied by the logic network that accumulates the error signals generated by individual flip-flops and produces one or more global error signals.

The error-trapping scan cell shown in Figure 5 eliminates this problem. Latches LA and LB store redundant copies of the PH2 and PH1 content, respectively, during normal operation. A soft error in any latch causes the error signal (E) to be 1. This signal drives the top input of the exclusive-or gate XOR2 so that when E equals 1, the output of XOR2 (D1) becomes the complement of D.

Once the error signal E is 1, the logic values stored in LA and LB become complements of the contents of PH2 and PH1, respectively, and E continues to be 1. Thus, the error is trapped until another soft error affects one of the latches of this flip-flop, which is a rare event.

After a prespecified number of clock cycles, at a recovery point the system shifts out this trapped error signal using the existing scan path, which eliminates the need for global routing of error signals at the cost of error-detection latency. Re-execution then achieves error correction.[13]

Table 3 shows the results generated by performing circuit simulations on a typical process corner for an advanced technology to compare the soft-error-resilient scan flip-flops and a conventional scanned flip-flop.

To evaluate the system-level impact of soft-error-resilient scan cell designs, we estimated the chip-level area and power overheads of new soft-error resilient scan flip-flop designs in Table 4, assuming that 25 percent of the flip-flops are protected from soft errors.[8] The results showed that the overall power and area overheads for all proposed designs are less than 5 percent and 0.3 percent, respectively. Such relatively low overheads, combined with the expected high gain in soft-error resilience, justify the use of proposed designs in various applications. Several optimizations are possible to further reduce the system-level power overhead to 3 percent or less.

Table 3. Relative cell-level timing, power, area, and soft-error rate comparisons.

Approach	Scannable	D-to-Q	C-to-Q	Power	Area	Global interconnect	Undetected soft-error rate
Master/slave flip-flop	Yes	1.00	1.00	1.00	1.00	None	1.00
Error-blocking design	Yes	1.00	1.08	2.13	1.08	None	< 0.05
Self-checking design	Yes	0.99	0.99	2.02	0.95	Several for error accumulation	0
Error-trapping design	Yes	0.97	0.99	2.26	1.24	Reused from existing scan path	0

Table 4. Chip-level power area and performance overhead, by percent.

Approach	Power overhead	Area overhead	Performance overhead
Error-blocking design	4.5	0.10	0
Self-checking design	4.0	-0.06	0
Error-trapping design	5.0	0.30	0

The reuse paradigm for built-in soft-error resilience offers the following unique advantages over existing soft-error protection techniques:

- minimal area overhead because resources already available for test and debug can be reused for soft-error resilience;
- minimal routing overhead;
- no major architectural changes required;
- applicability to any design—microprocessors, network processors, and ASICs; and
- a broad spectrum of design choices with several area, power, performance, and soft-error rate tradeoffs. For example, the design shown in Figure 4 can be redesigned to achieve a 50 percent rather than a 20 times reduction in the SER, with a 30 percent reduction in the cell-level power overhead.

Soft-error rates are getting worse for systems manufactured in advanced technologies with very high levels of integration. Stringent data integrity and the availability requirements of enterprise and networking applications demand special attention to soft errors not only in SRAMs but also in sequential elements and combinational logic from the very early phases of product development forward.

Applying the reuse paradigm for built-in soft-error protection significantly reduces the system-level soft-error rate and introduces minimal overhead. Automated techniques for architectural-vulnerability-factor estimation are required to further reduce the system-level power, performance, and area overheads of these techniques. ■

Acknowledgments

For their help with this article, we thank R. Fuller, J. Maiz, and T.M. Mak of Intel, and E.J. McCluskey of Stanford University.

References

1. D. Lyons, "Sun Screen," *Forbes Magazine*, 2000; www.forbes.com/forbes/2000/1113/6613068a.html.
2. R. Wilson and D. Lammers, "Soft Errors Become Hard Truth for Logic," *EE Times*, 3 May 2004; www.eetimes.com/semi/news/showArticle.jhtml?articleID=19400052.
3. D.C. Bossen, "CMOS Soft Errors and Server Design," *Workshop on Radiation Induced Soft Errors, Proc. IEEE Int'l Reliability Physics Symp.*, IEEE Press, 2002.
4. "Increasing Network Availability"; www.cisco.com.
5. H.T. Nguyen and Y. Yagil, "A Systematic Approach to SER Estimation and Solutions," *Proc. IEEE Int'l Reliability Physics Symp.*, IEEE Press, 2003, pp. 60-70.
6. N. Seifert and N. Tam, "Timing Vulnerability Factors of Sequentials," *IEEE Trans. Device and Materials Reliability*, Sept. 2004, pp. 516-522.
7. K.K. Goswami, R. Iyer, and L.Y. Young, "DEPEND: A Simulation-Based Environment for System-Level Dependability Analysis," *IEEE Trans. Computers*, Jan. 1997, pp. 60-74.
8. N.J. Wang et al., "Characterizing the Effects of Transient Faults on a High-Performance Processor Pipeline," *Proc. Int'l Conf. Dependable Systems and Networks*, IEEE Press, 2004, pp. 61-70.
9. S.S. Mukherjee et al., "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor," *Proc. Int'l*

Symp. Microarchitecture, IEEE CS Press, 2003, pp. 29-40.
10. P. Hazucha et al., "Neutron Soft Error Rate Measurements in a 90-nm CMOS Process and Scaling Trends in SRAM from 0.25- m to 90-nm Generation," *Proc. Int'l Electron Devices Meeting*, 2003, pp. 21.5.1-21.5.4.
11. R. Baumann, "The Impact of Technology Scaling on Soft-Error Rate Performance and Limits to the Efficacy of Error Correction," *Proc. IEEE Int'l Electron Devices Meeting* (IEDM02), IEEE Press, 2002, pp. 329-332.
12. P. Hazucha et al., "Measurements and Analysis of SER-Tolerant Latch in a 90-nm Dual Vt CMOS Process," *IEEE J. Solid State Circuits*, Sept. 2004, pp. 1536-1543.
13. D.P. Siewiorek and R.S. Swarz, *Reliable Computer Systems Design and Evaluation,* 3rd ed., A.K. Peters, 1998.
14. N. Oh, P.P. Shirvani, and E.J. McCluskey, "Error Detection by Duplicated Instructions in Super-Scalar Processors," *IEEE Trans. Reliability*, Mar. 2002, pp. 63-75.
15. N. Oh, S. Mitra, and E.J. McCluskey, "ED4I: Error Detection by Diverse Data and Duplicated Instructions," *IEEE Trans. Computers*, Feb. 2002, pp. 180-199.
16. N.R. Saxena et al., "Dependable Computing and On-Line Testing in Adaptive and Reconfigurable Systems," *IEEE Design and Test of Computers*, Jan.-Mar. 2000, pp. 29-41.
17. S.S. Mukherjee, M. Kontz, and S. Reinhardt, "Detailed Design and Evaluation of Redundant Multithreading Alternatives," *Proc. Int'l Symp. Computer Architecture*, IEEE CS Press, 2002, pp. 99-110.
18. M. Nicolaidis, "Time Redundancy-Based Soft-Error Tolerance to Rescue Nanometer Technologies," *Proc. IEEE VLSI Test Symp.*, IEEE Press, 1999, pp. 86-94.
19. A. Carbine and D. Feltham, "Pentium Pro Processor Design for Test and Debug," *Proc. Int'l Test Conf.*, IEEE Press, 1997, pp. 294-303.
20. R. Kuppuswamy et al., "Full Hold-Scan Systems in Microprocessors: Cost/Benefit Analysis"; http://developer.intel.com/technology/itj/2004/volume08issue01/.

***Subhasish Mitra**, a senior staff engineer at Intel, is also a consulting assistant professor in the Electrical Engineering Department at Stanford University and the associate director of the Stanford Center for Reliable Computing. His research interests include robust system design, VLSI design and test, fault-tolerant computing, and computer architecture. Mitra received a PhD in electrical engineering from Stanford University. Contact him at subhasish.mitra@intel.com.*

***Norbert Seifert** is a design and reliability engineer at Intel. His research interests include the interdependence of design and system reliability. Seifert received a PhD in physics from the Technical University of Vienna, Austria. Contact him at Norbert.Seifert@ieee.org.*

***Ming Zhang** is an intern at Intel and a PhD candidate in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. His research interests include design and modeling of reliable circuits and systems. Zhang received an MS in electrical engineering from the University of Illinois at Urbana-Champaign. Contact him at mzhang2@uivlsi.csl.uiuc.edu.*

***Quan Shi** is a senior design engineer at Intel. His research interests include circuit-hardening techniques, circuit modeling and validation, and asynchronous circuits. Shi received a PhD in electrical engineering from the University of New Mexico. Contact him at quan.shi@intel.com.*

***Kee Sup Kim** is the director of DFX—Design for Test, Reliability, Manufacture, and Debug—for communications products at Intel. His research interests include the four DFX areas, especially structural test, speed-defect coverage, BIST, and quality risk assessment. Kim received a PhD in electrical engineering from the University of Wisconsin-Madison. Contact him at kee.sup.kim@intel.com.*