

## CSER: BISER-Based Concurrent Soft-Error Resilience

<sup>1,3</sup> Laung-Terng Wang, <sup>2</sup> Nur A. Touba, <sup>1</sup> Zhigang Jiang, <sup>1</sup> Shianling Wu, <sup>3</sup> Jiun-Lang Huang, and <sup>3</sup> James Chien-Mo Li

<sup>1</sup> SynTest Technologies, Inc., 505 S. Pastoria Ave., Suite 101, Sunnyvale, CA 94086, USA

<sup>2</sup> Dept. of Electrical and Computer Engineering, University of Texas, Austin, TX 78712, USA

<sup>3</sup> Dept. of Electrical Engineering, Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan

### Abstract

*This paper presents a concurrent soft-error resilience (CSER) scheme with features that aid manufacturing test, online debug, and defect tolerance. The proposed CSER scheme is based on the built-in soft-error resilience (BISER) technique [4]. A BISER cell is redesigned into various robust CSER cells that provide slow-speed snapshot, manufacturing test, slow-speed signature analysis, and defect tolerance capabilities. The cell-level area, power, and performance overhead of the robust CSER cells were found to be generally within 1% to 22% of the BISER cell.*

### 1. Introduction

**Soft errors** are caused by transient faults induced by various types of radiation [1]. Radiation-induced transient faults can abruptly flip the stored state of a system and cause a system crash or even worse - a **silent data corruption** (SDC) - if they remain undetected [2].

Atmospheric radiation, such as cosmic rays, has long been regarded as the major source of soft errors, especially in memories. Chips used in space applications have typically employed *parity* or *error-correcting codes* (ECCs) for soft-error protection. Reduced feature sizes, higher logic densities, shrinking node capacitances, lower supply voltage, and shorter pipeline depth have significantly increased the susceptibility of *integrated circuits* (ICs) to **single event upsets** (SEUs) in memories and sequential elements (latches and flip-flops). Terrestrial radiation, such as alpha particles from the packaging materials of the chips, is also starting to cause soft errors with increasing frequency. This has also created system reliability concerns, especially for chips used for mainstream applications in automotive, mobile, banking, medical, and networking industries, to name a few.

As process geometries continue to scale down, the amount of energy required to cause an error is lowered. Logic soft errors have become the major contributors to system-level silent data corruption for designs manufactured in advanced technologies at 90nm and below [3,4]. While cost and performance have traditionally been the primary objectives of these advanced applications, the increased susceptibility to logic soft errors has resulted in new robust design approaches being proposed to mitigate

SEUs from the process to device to circuit and system levels [5].

At the circuit level, a cost-effective approach to protect sequential elements against soft errors has centered on designing robust scan cells using a basic scan flip-flop [6] or a scanout flip-flop [7]. The basic scan flip-flop consists of a system flip-flop and a scan portion for manufacturing test. The scanout flip-flop consists of a system flip-flop and a scanout (signature analysis) portion for real-time debug. Alternatively, the system flip-flop can be a pulse latch [8]. For instance, researchers at Intel® have designed a few robust scan cells using a **built-in soft error-resilience** (BISER) technique for protecting these basic scan cells against SEUs [4,5,9]. A BISER cell may consist of a basic scan flip-flop and an output joining circuit for both test and soft-error resilience. Alternatively, the BISER cell may consist of a scanout flip-flop and an output joining circuit for both debug and soft-error resilience. The output joining circuit may include a **Muller C-element**, a self-checking circuit, or an error-trapping circuit for error correction or detection.

Another cost-effective approach for designing robust scan cells to mitigate soft errors is applying the **triple modular redundancy** (TMR) concept that adds a majority voter to the basic scan flip-flop or a muxed-scan flip-flop (scan cell). This approach is different from the classical TMR technique where three copies of the system flip-flops must be used. For instance, researchers at IBM® [10] basically made use of a majority voter coupled to the outputs of the master and slave latches of a muxed-scan flip-flop as well as a duplicate of the slave latch. The reconfigured robust scan cell provides additional self-correction capability to all three latches whenever a soft error is detected. The authors in [11] added a hold-state slave latch to a muxed-scan flip-flop that acts as a shadow latch and a C-element is used for soft-error correction. The reconfigured muxed-scan flip-flop further provides an additional feature for **enhanced scan testing**. Researchers in [12], on the other hand, preserved the enhanced scan testing capability by reconfiguring three latches of the basic scan flip-flop for soft-error correction. A majority voter instead of a C-element is then used to correct a single soft error that occurs in the basic scan flip-flop.

These approaches, however, do not address the need for robust scan cells that provide the capabilities for (1) **slow-**

**speed snapshot** which allows the system to shift out the contents of the robust scan cells upon capture at a reduced shift clock frequency when the system clock is still running, (2) **manufacturing test** which allows for detecting faults within the system flip-flops when using BISER cells, (3) **slow-speed signature analysis** which allows designers to sample the contents of the robust scan cells every two or more system clock cycles, and (4) **defect tolerance** which allows the system to continue its operation even when the system flip-flop in the robust scan cell has permanent defects.

To address these limitations, a **concurrent soft-error resilience** (CSER) scheme is proposed in this paper which provides these additional capabilities. To demonstrate the effectiveness of the proposed scheme, we first develop several robust CSER cells each redesigned from a BISER cell that includes a Muller C-element for soft-error resilience. The redesigned BISER cell, called a CSER cell, can perform slow-speed snapshot during online debug by adding one additional AND gate into the BISER cell. The CSER cell can also perform manufacturing test to detect faults within the system flip-flop when an additional MUX is further inserted into the BISER cell. Online debug becomes possible when the CSER cell is redesigned for slow-speed signature analysis. Lastly, a selector (called an **S-element**), when coupled to the C-element, can provide the system with defect tolerance capability to bypass the defective system flip-flop. It has been observed through experiments that the additional area, power, and performance overhead for the CSER cells are generally within 1% to 22% of the respective BISER cell. The proposed CSER cells, however, can provide the system with additional manufacturing test, online debug, and defect tolerance capabilities.

The rest of the paper is organized as follows: Section 2 reviews the basic BISER technique. Section 3 presents the proposed robust CSER cells based on the BISER technique. Section 4 describes the proposed robust CSER cells using muxed-scan flip-flops that are widely used in the industry. Section 5 compares the overhead associated with area, power, and performance between each CSER cells and a BISER cell given in [4]. Section 6 concludes the paper.

## 2. Background

The *built-in soft-error resilience* (BISER) technique proposed in [4,5,9] has demonstrated its effectiveness in correcting logic soft errors during system operation. BISER reuses the scan portion of a basic scan flip-flop or the scanout portion of a scanout flip-flop to reduce area overhead. The scan portion includes a slow scan chain, while the scanout portion includes a debug chain.

Fig. 1 shows a BISER cell using a Muller C-element for mitigating SEUs in the latches. This BISER cell consists of a system flip-flop and a scan portion, each of which contains a two-port D latch ( $PH_1$  and  $LA$ ) and a one-port D latch ( $PH_2$  and  $LB$ ). The data signal  $D$  present in the system flip-flop which connects to the  $1D$  data port of latch  $PH_2$  further connects to the  $2D$  data port of latch  $LA$ . The functional clock  $CLK$  present in the system flip-flop which drives latches  $PH_2$  and  $PH_1$  further drives latches  $LA$  and  $LB$  in the scan portion through the additions of an OR gate and an AND gate. In so doing, the cell operates in three modes: test mode, system mode, and economy mode.

**In test mode**,  $TEST$  is set to 1, and the C-element acts as an inverter for latch  $PH_1$  output  $O_1$ . During shift operation, a test vector is shifted into latches  $LA$  and  $LB$  by alternately applying clocks  $SCA$  and  $SCB$  while keeping  $CAPTURE$  and  $CLK$  at 0. Then, the  $UPDATE$  clock is applied to move the content of  $LB$  to  $PH_1$ . As a result, a test vector is written into the system flip-flop. During capture operation,  $CAPTURE$  is first set to 1, and then the functional clock  $CLK$  is applied which captures the circuit response to the test vector into the system flip-flop and the scan portion simultaneously. The circuit response is then shifted out by alternately applying clocks  $SCA$  and  $SCB$  again.

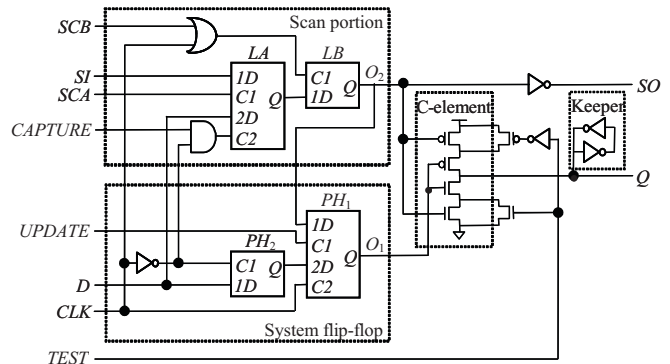


Figure 1: A BISER cell for test and error resilience [4].

**In system mode**,  $TEST$  is set to 0, and the C-element acts as a hold-state comparator. The function of the C-element is shown in Table 1.

Table 1: C-element Truth Table

$O_1$	$O_2$	$Q$
0	0	1
1	1	0
0	1	Previous value retained
1	0	Previous value retained

When inputs  $O_1$  and  $O_2$  are unequal, the output of the C-element keeps its previous value. During this mode, a 0 is

applied to the *SCA*, *SCB*, and *UPDATE* signals, and a 1 is applied to the *CAPTURE* signal. This converts the scan portion into a master-slave flip-flop that operates as a shadow of the system flip-flop. That is, whenever the functional clock *CLK* is applied, the same logic value is captured into both the system flip-flop and the scan portion. When *CLK* is 0, the outputs of latches *PH<sub>1</sub>* and *LB* hold their previous logic values. If a soft error occurs either at *PH<sub>1</sub>* or at *LB*, *O<sub>1</sub>* and *O<sub>2</sub>* will have different logic values. When *CLK* is 1, the outputs of latches *PH<sub>2</sub>* and *LA* hold their previous logic values, and the logic values drive *O<sub>1</sub>* and *O<sub>2</sub>*, respectively. If a soft error occurs either at *PH<sub>2</sub>* or at *LA*, *O<sub>1</sub>* and *O<sub>2</sub>* will have different logic values. In both cases, unless such a soft error occurs after the correct logic value passes through the C-element and reaches the keeper, the soft error will not propagate to the output *Q* and the keeper will retain the correct value at *Q*.

**In economy mode**, *CAPTURE* is set to 0 and the scan clock *SCB* is forced to 1 [5]. The power in the scan portion is completely shut off. This mode turns off soft-error protection to enable reuse of the same core for applications, such as cell phones, where error protection may not be required or power saving is more important.

The beauty of this technique is that the BISER design has self-correction capability. Each BISER cell can still function as a normal scan cell in test mode. Once the chip is in normal operation, it can be configured in soft-error resilience mode and no errors which can be detected and corrected will propagate any further than the C-element. There are no new routing and new control signals to be added other than the existing scan control signals. Another important attribute of this technique is that it is applicable for any latch-based or flip-flop-based logic design.

The only shortcoming, however, is that the scan clocks *SCA/SCB* and the system clock *CLK* cannot be activated at the same time. This prevents the system from performing a snapshot operation using the slow scan chain (through the *SI* and *SO* ports) for online debug and diagnosis. Also, this BISER cell can only perform at-speed signature analysis [9], and the faults (*e.g.*, a stuck-at fault) within latch *PH<sub>2</sub>* are not properly detected during manufacturing test because one cannot observe the *O<sub>1</sub>* output of latch *PH<sub>1</sub>* directly after each test application. These limitations are addressed in the proposed CSER cells.

### 3. Robust CSER Cells

The design of robust CSER cells is described here. They are constructed by adding additional circuitry to the BISER cell given in Fig. 1. The robust CSER cells enable the system to run in system mode for concurrent soft-error

resilience while at the same time providing manufacturing test, online debug, and defect tolerance capabilities. The use of CSER cells can be considered a *design-for-excellence* (DFX) technique because it supports test, debug, and defect tolerance to enhance product reliability.

#### 3.1 CSER Cell for Slow-Speed Snapshot

In Fig. 2, slow-speed snapshot capability is added to the BISER cell given in Fig. 1. The design and operation of the CSER cell are identical to what was previously described for Fig. 1 with the exception of an additional AND gate which is inserted before the OR gate that drives latch *LB*, to gate the functional clock *CLK* with the *CAPTURE* input.

Table 2: Snapshot Operation of the Scan Portion

Mode	Action	<i>CAPTURE</i>	<i>CLK</i>	<i>SCB/SCA</i>
Snapshot	Load <i>D</i>	1	ON	OFF
Test	Shift <i>SI</i>	0	ON/OFF	ON

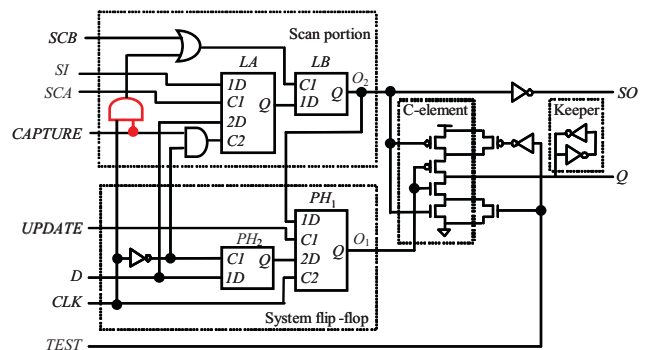


Figure 2: A CSER cell for slow-speed snapshot.

The behavior of the cell in test mode is identical to what was described earlier for Fig. 1. The behavior of the cell in system mode differs in the following way. There are now two ways that the scan portion can operate in system mode. If 0 is applied to the *SCA*, *SCB*, and *UPDATE* signals, and a 1 is applied to the *CAPTURE* signal, then the scan portion will shadow the operation of the system flip-flop, the same as described earlier for Fig. 1. However, if a 0 is applied to the *CAPTURE* signal, then the scan portion can perform slow-speed snapshot.

When *CAPTURE* is 0, the scan portion is decoupled from the system flip-flop. The functional clock *CLK* is gated by the *CAPTURE* signal so that it can no longer trigger state changes in either latch *LA* or *LB*. The circuit state can then be shifted out by alternately applying scan clocks *SCB* and *SCA* which shifts the response out through output *SO*. The frequency at which *SCA* and *SCB* are clocked need not be related in any manner to the system clock frequency, and hence a slow-speed snapshot can be performed. The snapshot operation is listed in Table 2.

### 3.2 CSER Cell for Manufacturing Test

One drawback of the BISER cell design shown in Fig. 1 is its difficulty in detecting faults within latch  $PH_2$  during manufacturing test. If the output of latch  $PH_2$  is stuck-at-1 or 0, the fault would not be detected because the circuit response is captured in latch  $LB$ , rather than latch  $PH_1$ , after loading a test pattern through the  $UPDATE$  clock. Fig. 3 shows adding manufacturing test capability in Fig. 2 to capture output  $O_1$  of latch  $PH_1$  in the CSER cell.

Table 3: Operation Modes of the Scan Portion

Mode	Action	SHIFT	CAPTURE
Snapshot	Load $D$	-	1
Test	Shift $SDI$	1	0
Test	Load $O_1$	0	0

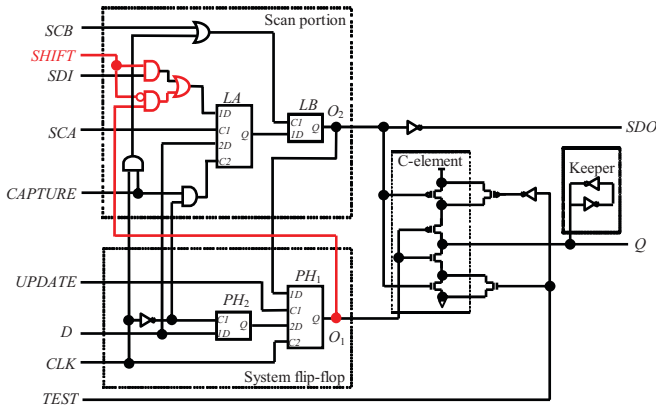


Figure 3: A CSER cell for manufacturing test.

When the  $CAPTURE$  signal is set to 1 for one system clock cycle and scan clocks  $SCB/SCA$  are set to 0, the design and operation of the CSER cell are identical to what was previously described for Fig. 2 for slow-speed snapshot. If  $SHIFT$  is set to 1 and  $CAPTURE$  is set to 0, then the scan in data  $SDI$  is scanned out through  $SDO$ . If  $SHIFT$  is set to 0 and  $CAPTURE$  is set to 0, then the output of  $PH_1$  is scanned out. This allows manufacturing test to detect faults within the system flip-flop, such as stuck-at or delay faults at the output of  $PH_2$ . The scan operations of Fig. 3 are shown in Table 3. The CSER cell that adds additional slow-speed snapshot and manufacturing test capabilities to the original BISER cell is referred to as a *modified BISER* (MBISER) cell.

### 3.3 CSER Cell for Slow-Speed Signature Analysis

Fig. 4 shows a CSER cell for online debug. A scanout flip-flop, which was used in the microprocessor described in [7], is reused to perform soft-error resilience. The design and operation of the cell are identical to what was previously described for Fig. 3 with the exception of the addition of some signature logic which consists of one XOR gate and one additional input  $LOAD$ . The  $SHIFT$ ,

$CAPTURE$ , and  $LOAD$  signals are assigned with the appropriate values as listed in Table 4 to perform the snapshot and signature analysis operations.

Table 4: Operation Modes of the Scanout Portion

Mode	Action	SHIFT	CAPTURE	LOAD
Test	Load $O_1$	0	0	1
Test	Clear data	0	0	0
Test	Shift $SDI$	1	0	0
Snapshot	Load $D$	0	1	0
Signature	Compress bit stream	1	1	0

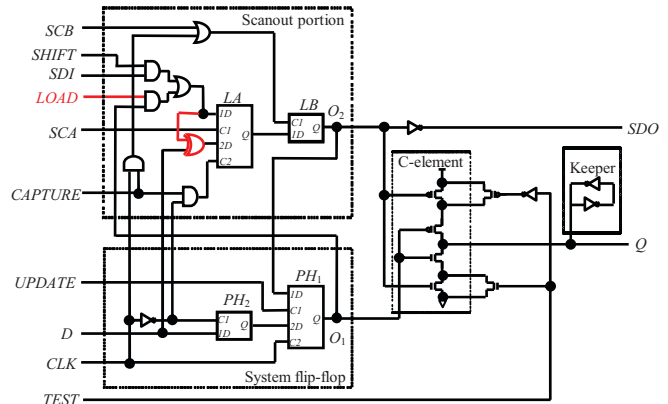


Figure 4: A CSER cell for slow-speed signature analysis.

When  $LOAD$  is set to 1 and both  $SHIFT$  and  $CAPTURE$  signals are set to 0, the design and operation of the CSER cell are identical to what was previously described for Fig. 3 for manufacturing test. When  $LOAD$  is set to 0, the CSER cell will allow for online debug. If  $SHIFT$  and  $CAPTURE$  are set to 0, then a constant value 0 is scanned into the scanout portion. If  $SHIFT$  is set to 1 and  $CAPTURE$  is set to 0, then the output of  $LB$  is scanned out. If  $SHIFT$  is set to 0 and  $CAPTURE$  is set to 1, then the output of latch  $PH_1$  is scanned out through  $SDO$ . If both  $SHIFT$  and  $CAPTURE$  are set to 1, then the XOR value (called the **signature**) of the output of latches  $PH_1$  and  $LB$  is scanned out. This allows the circuit to run in online debug modes: slow-speed snapshot and slow-speed signature analysis.

In slow-speed snapshot mode, the operation is identical to what was previously described for Fig. 2. In **slow-speed signature analysis mode**, the  $CAPTURE$  signal is set to 1 for one system clock cycle, and then set to 0 for one or more system clock cycles to match the frequency of the scan clocks  $SCA/SCB$ . For example, if the operating frequency of the functional clock  $CLK$  is 1 GHz and the shift frequency of the scan clocks  $SCA/SCB$  is 10 MHz, then the load operation may only occur every 100 or more system clock cycles to allow for enough time to shift the previous  $SDI$  value and the  $SDO$  signature value in and

out of the BISER cell. This is in sharp contrast to **at-speed signature analysis** where the *CAPTURE* signal is set to 1 all the time so that the snapshot and signature operations are performed at every system clock cycle. The *SDI/SDO* scan chain (referred to as a debug chain) must now allow shifting data in and out of the BISER cell at-speed. In the example given above, this means clocks *SCA/SCB* must operate at 1 GHz instead of 10 MHz. This may cause significant routing difficulty during layout. The CSER cell that adds additional slow-speed signature analysis capability to the *modified BISER* (MBISER) cell is referred to as a **concurrent BISER** (CBISER) cell.

### 3.4 CSER Cell for Defect Tolerance

Fig. 5 illustrates adding the ability to bypass/repair a failed flip-flop in the CSER cell. The design and operation of the CSER cell are identical to what was previously described for Fig. 2 with the exception of the addition of an S-element which has been coupled to the C-element in the output joining circuit to allow for selective bypass/repair (referred to as **defect tolerance**). The S-element truth table is shown in Table 5. When *TEST* is set to 0 and *SELECT\_O2* is set to 0, then the C-element behaves normally, the same way as previously described for Fig. 2 during system mode. When *TEST* is set to 1 and *SELECT\_O2* is set to 0, then the C-element inverts  $O_1$  and ignores  $O_2$  thereby allowing  $O_2$  to be bypassed during system operation. When *TEST* is set to 0 and *SELECT\_O2* is set to 1, then the C-element inverts  $O_2$  and ignores  $O_1$  thereby allowing  $O_1$  to be bypassed during system operation. If there is a defect in either the system flip-flop or the scan portion, then the defect can be tolerated by bypassing that particular flip-flop using the S-element.

Table 5: S-element Truth Table

Mode	<i>TEST</i>	<i>SELECT_O2</i>	$Q$
Normal	0	0	Select C-element
Select O1	1	0	$O_1$
Select O2	0	1	$O_2$

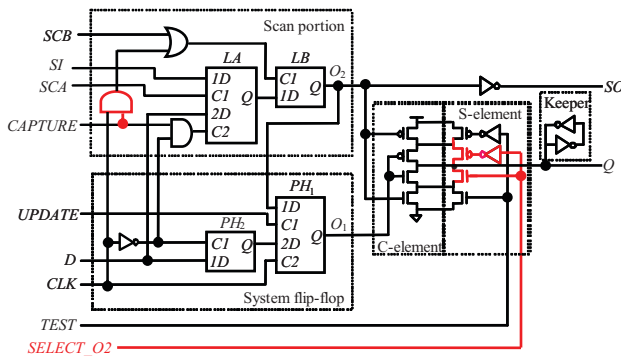


Figure 5: A CSER cell for defect tolerance.

## 4. MUX-Based CSER Cells

In Fig. 6, a MUX-based BISER cell has been redesigned as a MUX-based CSER cell with enhanced scan capability which allows the application of any pair of  $V_1$  and  $V_2$  vectors during two-pattern delay testing. The proposed CSER cell includes two muxed-scan flip-flops *SDFF1* and *SDFF2*. Each muxed-scan flip-flop consists of a *D* flip-flop and two multiplexers to select appropriate clock and data ports depending on the value of the scan enable signal *SE* or the *DEBUG* signal. The cell reuses the two multiplexers already available in modern ASIC designs, namely the clock MUX (which could be shared for use in a clock domain) and the scan enable MUX, for providing additional capability to aid in at-speed delay testing. Also, there is no need to capture the  $O_1$  value of *SDFF1* in *SDFF2* during manufacturing test because the captured response at  $O_1$  can be simply shifted out for analysis through the *SI/SO* scan chain. The cell can perform slow-speed snapshot when the *DEBUG* signal is set to 1. When signature logic and an S-element coupled to the C-element are added in accordance with what was described in Figs. 4 and 5, the cell can perform slow-speed signature analysis and defect tolerance, respectively.

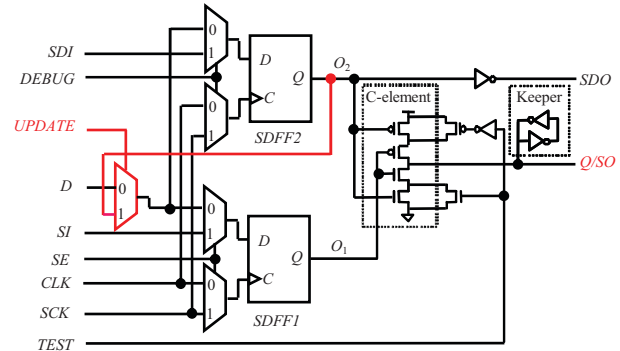


Figure 6: A MUX-based CSER cell for test and debug.

The enhanced scan capability for applying delay tests is controlled by the additional input *UPDATE*. The additional MUX allows flip-flop *SDFF1* to be loaded from either *D*, when *UPDATE* is equal to 0, or from the  $O_2$  output of flip-flop *SDFF2*, when *UPDATE* is equal to 1. The ability to load flip-flop *SDFF1* with the value stored in flip-flop *SDFF2* provides enhanced scan capability that permits the application of any two-pattern test where  $V_1$  may be scanned into flip-flop *SDFF1* through the slow *SI/SO* scan chain or the *SDI/SDO* debug chain, and  $V_2$  may be scanned into flip-flop *SDFF2* through the debug chain to launch a transition to  $V_1$ .

Fig. 7 illustrates how the MUX-based CSER cells are used in a robust scan design. A global scan enable signal, *SE*, is routed to all scan cells, and a global debug mode signal, *DEBUG*, and a global test mode signal, *TEST*, are routed to the CSER cells. Two scan paths are formed. One

is the slow scan chain which runs along the *SI/SO* path through the three cells for manufacturing test. The other is the debug chain which runs along the *SDI/SDO* path through the two CSER cells for debug.

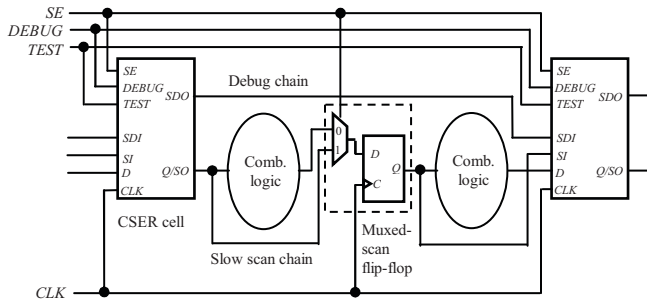


Figure 7: An example robust scan design.

## 5. Experimental Results

We designed the CSER cells using the 45-nm *predictive technology model* (PTM) released by Arizona State University and estimated their respective cell-level area, power, and timing [13]. We then compare their cell-level overhead in area, power, and performance with the BISER cell illustrated in Fig. 1. The normalized results (to BISER) are shown in Table 6. All CSER cells incur higher area, power, and delay overhead than the BISER cell. Each cell-level overhead shown in “Figs. 4/5” is the overhead of adding the defect tolerance circuit shown in Fig. 5 onto Fig. 4 for providing the CSER with test, debug, reliability, and defect tolerance capabilities. The overhead in Fig. 6 does not include the two shared clock MUXes.

Table 6: Cell-Level Area, Power, and Delay Comparison

	Area	Power	D-to-Q delay
BISER (Fig. 1)	1.00	1.00	1.00
CSER (Fig. 2)	1.05	1.02	1.00
CSER (Fig. 3)	1.14	1.02	1.00
CSER (Fig. 4)	1.18	1.02	1.00
CSER (Fig. 5)	1.08	1.01	1.00
CSER (Figs. 4/5)	1.22	1.03	1.00
CSER (Fig. 6)	1.00	0.58	0.88

SPICE simulations showed that 1) the MUX-based CSER cell illustrated in Fig. 6 yielded better results than the BISER cell given in Fig. 1, and 2) all other CSER cells incurred about 5% to 22% area overhead but up to 3% power/performance overhead over the BISER cell. This is because each CSER cell was designed and resized to match the BISER cell performance as closely as possible.

## 6. Summary and Conclusions

In this paper, we proposed a BISER-based *concurrent soft-error resilience* (CSER) scheme. The redesigned BISER cell, called a CSER cell, can perform slow-speed

snapshot and signature analysis during online debug, detect more faults during manufacturing test, and bypass a defective system flip-flop for defect tolerance. While each CSER cell incurs higher area, power, and performance overhead than a BISER cell given in [4], designers can now make use of these robust CSER cells for manufacturing test, online debug, reliability enhancement, and defect tolerance.

This paper only discussed techniques to mitigate soft errors caused by *single event upsets* (SEUs). Future work will explore design techniques to further match the BISER cell performance and investigate cost-effective robust schemes to protect combinational logic against *single event transients* (SETs) [14-16].

## 7. Acknowledgments

The authors wish to thank Mr. Shih-Lun Peng of National Taiwan Univ. for designing the BISER and CSER cells.

## 8. References

- [1] R. Baumann, “Soft Errors in Advanced Computer Systems,” *IEEE Design & Test of Computers*, vol. 22, no. 3, pp. 258–266, May-June 2005.
- [2] L.-T. Wang, C. E. Stroud, and N. A. Touba, Eds., *System-on-Chip Test Architectures: Nanometer Design for Testability*, Morgan Kaufmann, San Francisco, 2007.
- [3] S. Mitra, T. Kamik, N. Seifert, and M. Zhang, “Logic Soft Errors in Sub-65nm Technologies Design and CAD Challenges,” *Proc. ACM/IEEE Design Automation Conference*, pp. 2–4, 2005.
- [4] 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.
- [5] M. Zhang, S. Mitra, T. M. Mak, N. Seifert, N. J. Wang, Q. Shi, K. S. Kim, N. R. Shanbhag, and S. J. Patel, “Sequential Element Design With Built-In Soft-Error Resilience,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, no. 12, pp. 1368–1378, Dec. 2006.
- [6] R. Kuppaswamy, P. DesRosier, D. Feltham, R. Sheikh, and P. Thadikaran, “Full Hold-Scan Systems in Microprocessors: Cost/Benefit Analysis,” *Intel Technology Journal*, vol. 8, no. 1, Feb. 2004.
- [7] A. Carbine and D. Feltham, “Pentium Pro Processor Design for Test and Debug,” *IEEE Design & Test of Computers*, vol. 15, no. 3, pp. 77–82, July-Sept. 1998.
- [8] S. D. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. J. Sullivan, and T. Grutkowski, “The Implementation of the Itanium 2 Microprocessor,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1448–1460, Nov. 2002.
- [9] S. Mitra, M. Zhang, T. M. Mak, N. Seifert, V. Zia, and K. S. Kim, “Logic Soft Errors – A Major Barrier to Robust Platform Design,” *Proc. IEEE International Test Conference*, Paper 28.3, pp. 1-10, 2005.
- [10] A. J. Drake, A. KleinOowski, and A. K. Martin, “Self-Correcting Soft Error Tolerant Flop-Flop,” *Proc. NASA Symposium on VLSI Design*, 2005.
- [11] A. Goel, S. Bhunia, H. Mahmoodi, and K. Roy, “Low-Overhead Design of Soft-Error-Tolerant Scan Flip-Flops with Enhanced-Scan Capability,” *Proc. ACM/IEEE Asia and South Pacific Design Automation Conference*, pp. 665–670, 2006.
- [12] A. Jagirdar, R. Oliveira, and T. J. Chakraborty, “Efficient Flip-Flop Designs for SET/SEU Mitigation with Tolerance to Crosstalk Induced Signal Delays,” *Proc. IEEE Workshop on Silicon Errors in Logic – System Effects*, 2007.
- [13] “Predictive Technology Model (PTM),” Released by Arizona State University, <http://www.eas.asu.edu/~ptm>, Nov. 2008.
- [14] K. Mohanram and N. A. Touba, “Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits,” *Proc. IEEE International Test Conference*, pp. 893-901, 2003.
- [15] M. Nicolaidis, “Design for Soft Error Mitigation,” *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 405–418, Mar. 2005.
- [16] S. Mitra, M. Zhang, S. Waqas, N. Seifert, B. Gill, and K. S. Kim, “Combinational Logic Soft Error Correction,” *Proc. IEEE International Test Conference*, Paper 29.2, pp. 1-9, 2006.