Integrating scan design and soft error correction in low-power applications

Michael E. Imhof, Hans-Joachim Wunderlich, Christian G. Zoellin

Institut fuer Technische Informatik, Universitaet Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
email: {imhof, wu, zoellin}@iti.uni-stuttgart.de

***Abstract*— In many modern circuits, the number of memory elements in the random logic is in the order of the number of SRAM cells on chips only a few years ago. In arrays, error correcting coding is the dominant technique to achieve acceptable soft-error rates. For low power applications, often latches are clock gated and have to retain their states during longer periods while miniaturization has led to elevated susceptibility and further increases the need for protection.**

This paper presents a fault-tolerant register latch organization that is able to detect single-bit errors while it is clock gated. With small addition, single and multiple errors are detected in the clocked mode, too. The registers can be efficiently integrated similar to the scan design flow, and error detecting or locating information can be collected at module level. The resulting structure can be efficiently reused for offline and general online testing.

I. Introduction

Today's technology scaling comes with effects concerning both power consumption and reliability. The paper presented here deals with the combination of both aspects. Single event effects (SEEs) are of concern for static memories [1], latches and flip-flops [2] and even for random combinational logic [3, 4]. As the soft error rate (SER) of memory elements in random logic is continuously increasing [5] and as the amount of flip-flops and latches is rapidly growing, the protection of these storage elements is of major concern.

Besides the high vulnerability to SEEs, scaling leads to an increased power density on chip which prohibits a frequency increase as seen in the past. The classic low power design techniques have still increasing relevance, but must be complemented by massive parallelism [6] and power management in networks and systems on a chip. While power gating is employed, if modules will be unused for a rather long period of time, clock gating is favored for shorter breaks where the computation will be resumed and the state must be retained.

There is a variety of protection schemes against single-event upsets (SEU) for flip-flops available [7, 8, 9, 2, 10, 11, 12, 13, 14, 15], all of them introduce redundancy, additional activity and additional power consumption. Significant progress has also been made to limit this power increase by special designs like BISER [2], RAZOR [16], DF-DICE [17] and others. However, these schemes do not target the clock gated phase, which is actually in most cases the larger time period a state must be retained than the clocked phase.

In very-low power designs, a latch-based design style provides benefits with respect to power, area and robustness. To avoid the overhead for LSSD, usually only a partial scan path is implemented. Because of this, the common approach to duplicate or triplicate the latches for error protection incurs immense overhead.

II. Low power SEU detection and identification

We present a technique to protect especially the clock gated phase, but it can also be used for fault detection in the clocked phase. The latch-based cells presented here have very low overhead and can be used in the same way as scan elements for an automated integration. A lightweight observation tree can be synthesized for fault location, indicating which register has been corrupted and if a restart after clock gating is required. Small extensions allow to reuse the scheme for clocked online fault detection and offline test.

A. SEU detection at gate level

Rather low impact on power, delay and area can be achieved if fault correction and fault tolerance at register level are not required and only fault detection is needed. Here, just a simple parity signal may be computed for a 8 bit register. The parity tree is gated by the inverted clock gating signal to reduce switching activity during operation. The parity logic itself is integrated into the latch design, and allows synthesis just by abutment. The final routing of the parity lines does not differ from any scan chain routing. Figure 1 shows the schematic of the parity computation of 2 latches.

With these parity pair latches (PPL) a register is formed like in figure 2. Since the parity signals have to be amplified anyway the additional impact of gating the tree is just

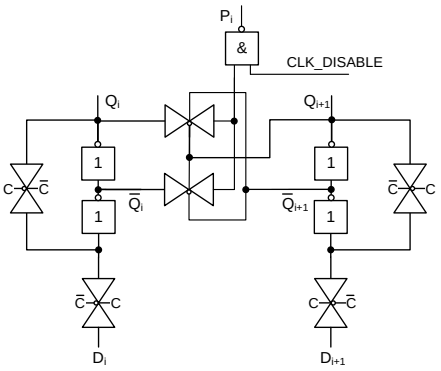

Fig. 1. Parity computation between 2 latches

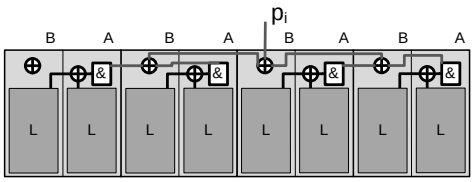

Fig. 2. Parity tree, consisting of two cell types

the difference between an inverter and a NAND gate. For the complete tree only the PPL and an XOR cell are required as seen in figure 2.

B. SEU identification at module level

The error information of the basic registers have to be passed to module's top level and will be used for register identification. For this, the transparent test technique for memory arrays from [18] is adapted to random logic. The modulo-2 address characteristic of a bit-oriented memory is computed by a bit-wise XOR of the addresses of those memory cells which contain a 1 (Figure 3).

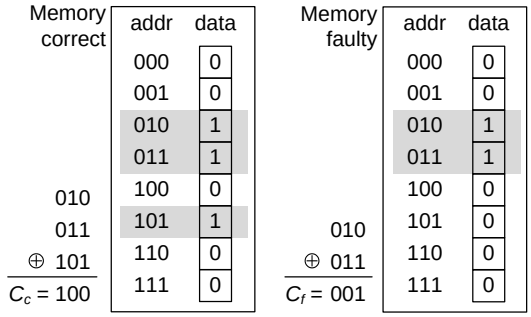

Fig. 3. Modulo-2 address characteristic

Compared to signature analysis or error correcting codes, the location of single errors is very easy. Here, the bit-wise XOR of the characteristic of the correct memory content c_c and of the incorrect one c_f provides the address of the erroneous bits. The R registers are numbered starting from 1 to R and p_1 to p_R are the parity bits. Address 0 does not contribute to error detection.

In a straight-forward way the characteristic is computed by an XOR tree whose leaves are the N bit binary words $p_i \wedge i$. A substantial amount of hardware can be saved if only significant bits are passed between the levels (Fig. 4).

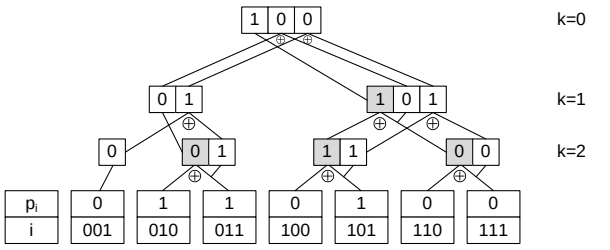

Fig. 4. Optimal tree organization

For a node $v_{k,l}$ on level k ($k \in \{0, ..., N-1\}$), the k most significant address bits $(a_{N-1}, \ldots, a_{N-k+1})$ of the preceeding vertices of $v_{k,l}$ are identical. Hence, the result bits $(c_{k,i,N-1}, \ldots, c_{k,i,N-k+1})$ only depend on the parity of the 2^{N-k} leaf vertex register-parities. We do not need to compute this vector of k bits. Instead, the generation of the information can be deferred to the successor in level $k-1$.

Therefore, each vertex on level k has just 2 connections to read each parity of the 2 predecessors and $2 \cdot (N-k)$ connections to read the characteristic of the predecessors.

C. Online detection and offline test

For LSSD-based approaches the GRAAL technique protects each single latch [19], the basic idea for online error detection here is not to compare each register latch but the parities (Figure 5). This extension is optional and may be avoided in ultra-low power circuits. The error signal allocated to each register may be evaluated and used for restart in the same way as for the RAZOR approach.

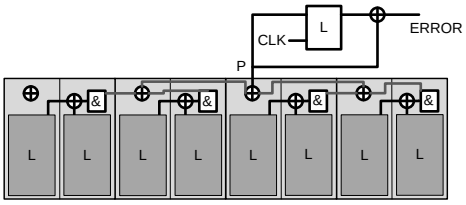

Fig. 5. Local error detection with low delay overhead

Furthermore, the error detecting design scheme presented so far is especially appropriate for partial scan design, and can be reused to support offline testing. The characteristic at module level increases the observability of all of the latches and compresses the test response. In [20] it has been shown, that this extreme response compaction does not affect fault coverage.

III. Experimental Results

In this section, we compare the technique presented here with the error masking BISER flip-flop presented in [2] and the error detecting RAZOR shadow latch presented in [16]. For each technique, the number of transistors to implement all the required logic for a LEON3 core is estimated and compared (Table I).

The configuration of LEON3 evaluated here has about 16k memory elements in the random logic. The memory elements are clustered into 2047 registers of up to 8 bits.

Column *PPL* gives the number of transistors for the different building blocks of the Parity Pair Latch scheme. For the presented scheme, *Offline detection* includes the characteristic computation as well as a simple comparator for the characteristic check. The *Online detection* row contains the additional latches, XORs as well as the OR tree to combine the $ERROR$ lines into one global signal.

The columns *RAZOR* and *BISER* give the same information for RAZOR and BISER. For RAZOR, online detection also requires a simple OR-tree to provide a core-level signal but over all latches instead of just the registers.

From the analysis it is obvious that the overhead of the presented technique is lower by an order of magnitude compared

	PPL	RAZOR	BISER
Per register	114	368	640
All registers	233358	753296	1310080
Offline detection	24626	n.a.	n.a.
Online detection	36842	65504	n.a.
Overall	294826	818800	1310080

TABLE I
TRANSISTOR COUNT

to the mentioned methods, which are based on duplication and triplication.

IV. CONCLUSION

The presented method is able to easily locate the register affected by a soft error in a low-power design. Compared to other soft error resilient memory elements, the presented structure requires up to 77% less overhead. This has direct impact on both the silicon area and most important on the associated leakage current. Furthermore, there is only insignificant impact on the switching activity since the NAND-gate built into the PPL avoids any superfluous switching activity in the characteristic computation tree.

V. ACKNOWLEDGMENT

This work has been supported by the DFG Project Realtest under grant Wu245/5-1.

REFERENCES

[1] B. Gill, M. Nicolaidis, *et al.*, "Radiation Induced Single-Word Multiple-Bit Upsets Correction in SRAM," in *11th IEEE International On-Line Test Symposium (IOLTS)*, 2005, pp. 266–271.

[2] S. Mitra, N. Seifert, *et al.*, "Robust system design with built-in soft-error resilience," *IEEE Computer*, vol. 38, no. 2, pp. 43–52, 2005.

[3] A. Nieuwland, S. Jasarevic, *et al.*, "Combinational Logic Soft Error Analysis and Protection," *12th IEEE International On-Line Test Symposium (IOLTS)*, pp. 99–104, 2006.

[4] M. Nicolaidis, "Design for soft error mitigation," *Device and Materials Reliability, IEEE Transactions on*, vol. 5, no. 3, pp. 405–418, 2005.

[5] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design & Test of Computers*, vol. 22, no. 3, pp. 258–266, 2005.

[6] S. Borkar, "Thousand core chipsa technology perspective," in *Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007*, 2007, pp. 746–749.

[7] T. Calin, M. Nicolaidis, *et al.*, "Upset hardened memory design for submicron CMOS technology," *Nuclear Science, IEEE Transactions on*, vol. 43, no. 6, pp. 2874–2878, 1996.

[8] T. Monnier, F. Roche, *et al.*, "Flipflop hardening for space applications," *Intl. Workshop on Memory Technology*, 1998.

[9] S. Krishnamohan and N. Mahapatra, "A highly-efficient technique for reducing soft errors in static CMOS circuits," *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, pp. 126–131, 2004.

[10] M. Nicolaidis, "Design for mitigation of single event effects," in *IOLTS*. IEEE Computer Society, 2005, pp. 95–96.

[11] A. Drake, A. KleinOsowski, *et al.*, "A Self-Correcting Soft Error Tolerant Flop-Flop," *12th NASA Symposium on VLSI Design, Coeur dAlene, Idaho, USA, Oct*, pp. 4–5, 2005.

[12] A. Goel, S. Bhunia, *et al.*, "Low-overhead design of soft-error-tolerant scan flip-flops with enhanced-scan capability," *Proceedings of the 2006 conference on Asia South Pacific design automation*, pp. 665–670, 2006.

[13] M. Omaña, D. Rossi, *et al.*, "Latch susceptibility to transient faults and new hardening approach," *IEEE Trans. Computers*, vol. 56, no. 9, pp. 1255–1268, 2007.

[14] M. Fazeli, A. Patooghy, *et al.*, "Feedback Redundancy: A Power Efficient SEU-Tolerant Latch Design for Deep Sub-Micron Technologies," *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 276–285, 2007.

[15] M. Zhang, T. Mak, *et al.*, "Design for Resilience to Soft Errors and Variations," *Proceedings of the 13th IEEE International On-Line Testing Symposium*, pp. 23–28, 2007.

[16] D. Ernst, N. Kim, *et al.*, "Razor: a low-power pipeline based on circuit-level timing speculation," *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pp. 7–18, 2003.

[17] R. Naseer and J. Draper, "The DF-dice storage element for immunity to soft errors," *Proceedings of the 48th IEEE International Midwest Symposium on Circuits and Systems, August*, 2005.

[18] S. Hellebrand, H.-J. Wunderlich, *et al.*, "Efficient online and offline testing of embedded drams," *IEEE Trans. Computers*, vol. 51, no. 7, pp. 801–809, 2002.

[19] M. Nicolaidis, "Graal: a new fault tolerant design paradigm for mitigating the flaws of deep nanometric technologies," *IEEE International Test Conference, 2007. ITC 2007*, pp. 1–10, 21-26 Oct. 2007.

[20] M. Elm, S. Holst, *et al.*, "Scan chain organization for extreme response compaction," *Submitted for publication*, 2008.