



All Faculty Publications

2003-09-11

Evaluating TMR Techniques in the Presence of Single Event Upsets

Paul S. Graham

Nathaniel Rollins

See next page for additional authors

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the [Electrical and Computer Engineering Commons](#)

Original Publication Citation

13

BYU ScholarsArchive Citation

Graham, Paul S.; Rollins, Nathaniel; Wirthlin, Michael J.; and Caffrey, Michael P., "Evaluating TMR Techniques in the Presence of Single Event Upsets" (2003). *All Faculty Publications*. 1048.

<https://scholarsarchive.byu.edu/facpub/1048>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Authors

Paul S. Graham, Nathaniel Rollins, Michael J. Wirthlin, and Michael P. Caffrey

LA-UR-03-7525

Approved for public release;
distribution is unlimited.

Title:

Paper:

Evaluating TMR Techniques in the Presence
of Single Event Upsets

Author(s):

Nathan Rollins, Michael Wirthlin,
Brigham Young University, Provo UT
Paul Graham, Michael Caffrey
Los Alamos National Laboratory, Los Alamos NM

Submitted to:

Military and Aerospace Programmable Logic Devices
International Conference, Washington DC
9/9-9/11/2003



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (8/00)

Evaluating TMR Techniques in the Presence of Single Event Upsets

Nathan Rollins¹, Michael J. Wirthlin¹,
Michael Caffrey², and Paul Graham²

nhr2@ee.byu.edu, wirthlin@ee.byu.edu, mpc@lanl.gov, and graham@lanl.gov

¹Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT. 84602

²Los Alamos National Laboratory, Los Alamos, NM

Abstract

Field programmable gate arrays (FPGAs) are sensitive to radiation-induced single event upsets (SEUs) within the configuration memory. Triple modular redundancy (TMR) is a technique commonly used to mitigate against design failures caused by SEUs. This paper evaluates the effectiveness and cost of TMR on two different counter designs in the presence of SEUs. The evaluation measures the reliability, area cost, and speed of different TMR styles. The tests show that when feedback TMR is used with triplicated clocks, it is possible to have a counter design which is insensitive to any single configuration upset.

I Introduction

Field programmable gate arrays (FPGAs) are an attractive hardware design option for many space-based computing applications. They can be reprogrammed while in orbit to adapt to changing mission needs and allow for design modifications. While SRAM-based FPGAs offer several advantages for space-based operations, they are sensitive to single event upsets (SEUs)[1]. SRAM-based FPGAs are especially sensitive to SEUs within the configuration memory of the device. The configuration memory defines the operation of the configurable logic blocks (CLBs), routing resources, input/output blocks (IOBs), and other FPGA resources and upsets in the configuration memory can change the operation of the circuit. To operate properly in space, SRAM-based FPGA circuit designs must anticipate and mitigate against any configuration SEU which could alter the design.

Several techniques have been proposed to make designs reliable in the presence of event upsets. Triple modular redundancy (TMR) is a technique commonly used to provide design hardening[2]. The purpose of this paper is to investigate the effectiveness and cost of different TMR styles in order to im-

prove the reliability of SRAM-based FPGA designs in the face of single event configuration upsets.

II SEU Simulator

In order to evaluate the effectiveness of TMR, it is important to identify the number of sensitive configuration bits in a design hardened with each TMR style. A sensitive configuration bit is a configuration bit that directly affects the behaviour of the design. Any single configuration bit that is upset due to radiation will change the behaviour of the design. If a sensitive configuration bit is upset and the behaviour of the design changes, we indicate that a failure has occurred.

A simulator developed at BYU[3] is used to exhaustively test the sensitivity of every configuration bit of a Virtex V1000 FPGA. This simulator was developed in order to evaluate how sensitive a given design is to configuration SEUs. Based on the SLAAC-1V FPGA computing board[4], the simulator places the design to be tested in two separate Virtex V1000 FPGAs. One of the two FPGAs is targeted for testing, and the second one is used as a 'golden' design. During the SEU simulation, every configuration bit is systematically upset. The outputs of both FPGAs are fed to a third FPGA. This third FPGA compares the output from the golden design to the design under test. If there are differences in the outputs, we know that an error has occurred in the design under test. This third FPGA communicates this information to a host via a PCI bus.

TMR is used to decrease the sensitivity of a design to single event upsets. The goal of this work is to evaluate the effectiveness and cost of each TMR style. Increased reliability from the use of TMR comes at the cost of greater overhead (in the form of LUTs, IOBs and routing) and reduced design speed. Each TMR approach will be evaluated in terms of the number of sensitive configuration bits, the number

of sensitive configuration bits relative to the amount of logic used, the overhead required, and the speed at which the design can run. Also, the use of TBUF voters in place of LUT voters will be evaluated.

III Baseline Designs

We apply each TMR style on two different counter designs. The first design is a simple 8-bit incrementer (Figure 1(a)). The second design is a more complex 8-bit incrementer/decrementer, loadable counter (Figure 1(b)). The performance and area of the unmitigated designs can be used as a baseline benchmark to measure the cost and effectiveness of each redundancy method. The first reported baselines are area and speed. Without TMR the simple incrementer requires only 8 LUTs and can run at 220 MHz. The complex counter consumes 10 LUTs and operates at 220 MHz.

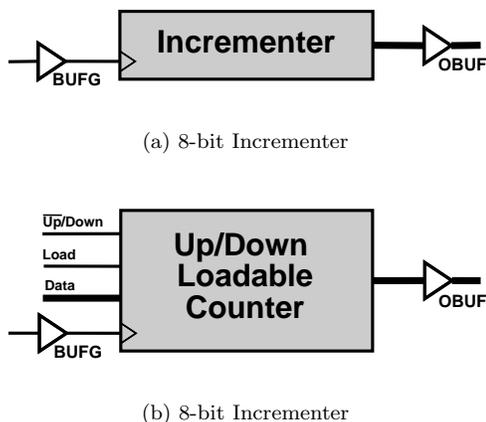


Figure 1: Baseline 8-bit Counter Designs

Next, we report the baseline benchmark for reliability. With no redundancy, the simple counter has 446 sensitive configuration bits. In other words, there are 446 specific bits in the configuration bit-stream that will cause this circuit to fail (or 58 sensitive configuration bits per LUT). The extra logic required by the second counter design makes it more sensitive to configuration upsets. The SEU simulator identified 463 sensitive configuration bits in this more complex design (or 46 sensitive configuration bits per LUT).

IV TMR Techniques

Triple Modular Redundancy (TMR) is a common technique used to harden circuits to prevent design failures due to single event upsets (SEUs)[2]. TMR is able to mask single circuit failures by triplicating the circuit of interest and voting on the circuit outputs. Several TMR styles are applied to both the simple and complex counters. The first TMR style triplicates the counters and uses a single voter (Figure 2). The next style triplicates the voters as well as the counters (Figure 3). The third method implements TMR on the feedback path of the counters (Figure 4). We also look at how triplicating the clock affects all of the previous TMR styles. The results of the simulations for each TMR design are summarized in Table 1. It is important to note that in each of the TMR design, half-latches have been completely removed[5].

A. TMR with 1 Voter

The first redundancy design triplicates the counters and then votes on the three designs with a single voter (Figure 2). Table 1 shows that the number of LUTs used with this style increases dramatically by a factor of 4. Further, the extra logic provides a limited increase in reliability. The problem with this method of hardening is that the voter is a single-point of failure. Any sensitive configuration bits associated with the voter will cause the design to fail. Since the voter uses the same number of LUTs as the original non-redundant counter, we see no increase in reliability. However, if we apply single voter TMR to a larger circuit (relative to the voter), we would see an increase in reliability.

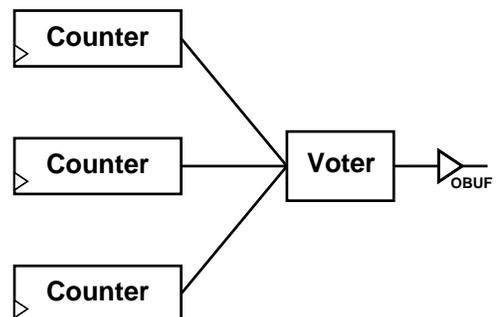


Figure 2: 8-bit Counter with TMR and One Voter

B. TMR with 3 Voters

We can improve the reliability of this first TMR approach by triplicating the voters (Figure 3). Using three voters removes the single-point of failure found in the previous approach. Table 1 shows that

this approach significantly reduces the configuration sensitivity of the design. However, using TMR with three voters consumes significantly more logic resources and is slightly slower than the non-redundant circuit. More specifically, when hardened this way, the design requires about six times more resources than the non-redundant counter. Improved reliability comes at the cost of additional hardware resources and slower operating speed.

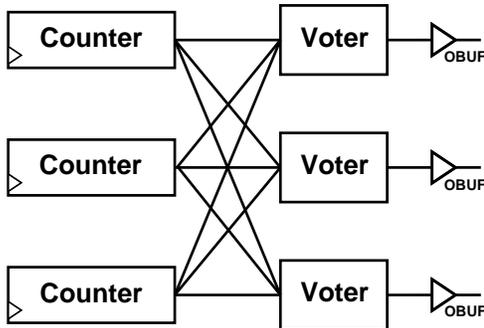


Figure 3: 8-bit Counter with TMR and 3 Voters

C. Feedback TMR

Up to this point, the hardening techniques discussed all suffer from resynchronization problems. If a faulty counter is repaired through bitstream scrubbing, the repaired counter will not be synchronized with the other two counters. This problem can be prevented by placing the voting circuitry within the feedback path of the circuit[6]. Placing voters within the feedback path of the counters can be seen in Figure 4. When the voters are put within the feedback path of the circuitry, synchronization errors are prevented, regardless of where the configuration upset occurs.

Table 1 shows that feedback TMR significantly increases the reliability of the counters while consuming the same number of resources as conventional 3-Voter TMR. Increased reliability does not come at

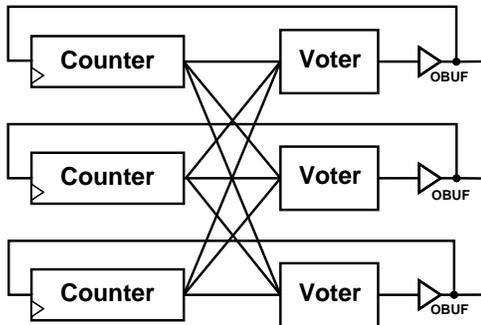


Figure 4: 8-bit Counter with Feedback TMR

the cost of area, but it comes at the cost of speed. Due to the additional logic within the feedback path, the feedback TMR method runs 30% slower than the non-redundant counter.

Strategic mapping techniques can be used to significantly reduce the logic resources of the feedback TMR incrementer circuit. Specifically, a single-bit voter circuit can be merged with the increment logic in a single LUT (see Figure 5). This technique produces the voting logic at no additional cost. Further, the feedback delay is reduced relative to the non-mapped counter circuit. In this special case, redundancy comes at a cost of only three times the original circuit size.

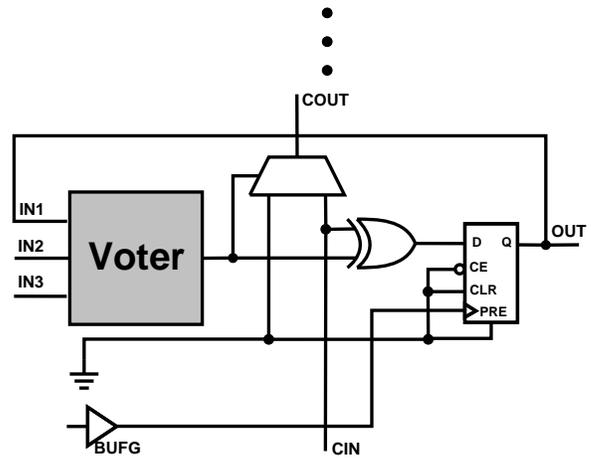


Figure 5: Incrementer with Mapped Feedback TMR

Unfortunately, this mapping technique is not always possible. Circuits which include more logic than a simple incrementer cannot take advantage of this mapping technique. In general, if the circuit requires more than two LUT inputs, the voter cannot be merged with the design into a single LUT. The complex counter for example, requires four LUT inputs, thus the merged-voter mapping technique cannot be applied. The six times cost associated with a 3-voter feedback TMR applies to these more complex circuits.

V Architectural TMR Techniques

The reliability of TMR methods discussed can be improved with additional architectural techniques. At this point, none of the TMR styles has been able to provide an absolutely reliable design (i.e. completely eliminating all sensitive configuration bits). By triplicating the global clock this goal can be reached. In addition to clock replication, TBUF voters can be used rather than LUT voters to reduce the logic resource requirements of the circuit.

Table 1: Evaluation of TMR on 8-bit Counters

Design (single clock)	Simple Incrementer			Up/Down Loadable Counter		
	LUTs	Failures	Speed (MHz)	LUTs	Failures	Speed (MHz)
No Redundancy	8	446	220	10	463	220
1 Voter	35 (~4x)	410	217 (99%)	41 (~4x)	484	217 (99%)
3 Voters	51 (~6x)	14	199 (91%)	57 (~6x)	14	213 (97%)
Feedback	51 (~6x)	14	160 (73%)	57 (~6x)	15	157 (72%)
Map Feedback	27 (~3x)	15	194 (88%)	N/A		

A. Triplicated Clocks

In order to improve the reliability of the different TMR styles, we triplicate the clock in each of the TMR styles (Figure 6). Using three clocks eliminates the single point of failure in the single clock domain. The results of triplicating the clock in each design are reported in Table 2. The most important result from this table is that *all* sensitive configuration bits have been removed from the design. In other words, *no* single-bit configuration bit upset will cause the design to fail. This result demonstrates that it is possible to make a counter design 'immune' to single event upsets.

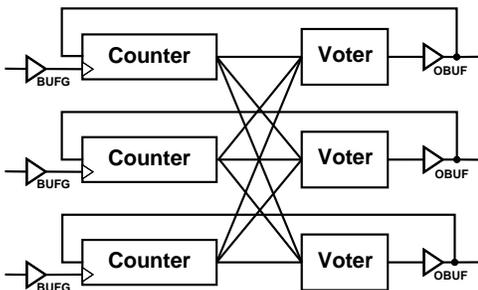


Figure 6: Resilient Counter Design

Although triplicating the clock can bring the number of sensitive configuration bits to zero, it comes at a cost. Triplicating the clock requires the use of three of the four Virtex global clock buffers and may increase the power consumption.

B. LUT Voters vs. TBUF Voters

The TMR voting can be implemented with Virtex TBUFs instead of LUT logic[6]. The TBUF voter shown in Figure 7(b) drives an active low when all three of its inputs or two out of three of its inputs are low. When all three of the inputs are high, the tristate buffers are disabled and the pull-up resistor pulls the output signal high. When two of the inputs are high, the pull-up resistor is also pulls a high signal to the output. Thus the TBUF voter produces the same results as the LUT voter.

Each of the TMR styles were implemented with both LUT and TBUF voters. As expected, the designs which used TBUF voters consumed fewer LUTs. However, a TBUF voter requires three TBUFs per bit while one a single LUT voter is required for each voter bit. A major difference between LUT and TBUF voters is design speed. Feedback TMR designs which use TBUF voters run at half the speed of TMR designs using LUT voters (see Table 3).

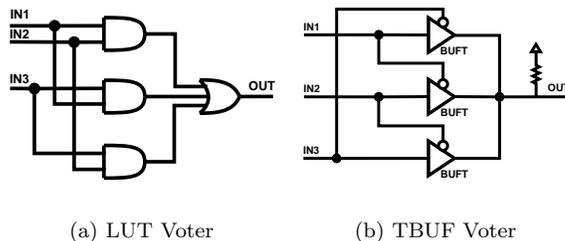


Figure 7: Voters Used with TMR

VI Conclusion

This paper evaluates the performance of several TMR design hardening techniques. The results reported in Tables 1-3 indicate that significant improvements in reliability can be made using appropriate redundancy techniques. Both the three voter TMR and feedback TMR styles can be used to eliminate all sensitive configuration bits. The mapped feedback TMR style proves to be the most effective TMR method. Using TMR in the feedback path of the 8-bit counters and triplicating the global clock reduces the number of configuration bits sensitive to single event upsets to zero and as well as eliminating resynchronization problems. In some cases, the mapped feedback TMR style allows us to pack the voter into the same LUTs as the counter. In most cases however, a completely resilient design comes at a much greater cost.

Table 2: Evaluation of Triplicated clocks and TMR on 8-bit Counters

Design	Simple Incrementer		Up/Down Loadable Counter	
	Failures	Speed (MHz)	Failures	Speed (MHz)
3 Voters	0	201 (91%)	0	218 (99%)
Feedback	0	167 (76%)	0	158 (72%)
Map Feedback	0	204 (93%)	N/A	

Table 3: Evaluation of TBUF Voters with TMR on 8-bit Counters

Design	Simple Incrementer			Up/Down Loadable Counter		
	LUTs	Failures	Speed (MHz)	LUTs	Failures	Speed (MHz)
1 Voter	27 ($\sim 3x$)	293	219 (100%)	33 ($\sim 3x$)	425	212 (96%)
3 Voters	27 ($\sim 3x$)	14	219 (100%)	33 ($\sim 3x$)	14	213 (97%)
3 Voters, 3 Clk	27 ($\sim 3x$)	0	219 (100%)	33 ($\sim 3x$)	0	215 (98%)
Feedback	27 ($\sim 3x$)	19	106 (48%)	33 ($\sim 3x$)	14	102 (46%)
Feedback, 3 Clk	27 ($\sim 3x$)	0	123 (56%)	33 ($\sim 3x$)	0	117 (53%)
Map Feedback	27 ($\sim 3x$)	19	105 (48%)	N/A		
Map Feedback, 3 Clk	27 ($\sim 3x$)	0	123 (56%)			

In the future we will evaluate the power consumption costs of these TMR styles and investigate the impact of TMR on other architectural features.

References

- [1] J. Wang, R. Katz, J. Sun, B. Cronquist, J. McCollum, T. Speers, and W. Plants. SRAM based re-programmable FPGA for space applications. *IEEE Transactions on Nuclear Science*, 46(6):1728–1735, December 1999.
- [2] J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, (Annals of Math Studies No. 34), 1956. Princeton University Press.
- [3] Eric Johnson, Michael J. Wirthlin, and Michael Caffrey. Single-event upset simulation on an FPGA. In Toomas P. Plaks and Peter M. Athanas, editors, *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, pages 68–73. CSREA Press, June 2002.
- [4] USC-ISI East. *SLAAC-1V User VHDL Guide*, October 1, 2000. Release 0.3.1.
- [5] Paul Graham, Michael Caffrey, Michael Wirthlin, Eric Johnson, and Nathan Rollins. Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip. In *24th Annual IEEE Aerospace Conference*, 2003. To be published.
- [6] Carl Carmichael. Triple module redundancy design techniques for Virtex FPGAs. Technical report, Xilinx Corporation, November 1, 2001. XAPP197 (v1.0).