Radiation effects and mitigation strategies for modern FPGAs

M. Stettler, M. Caffrey, P.Graham, J. Krone

Los Alamos National Laboratory, Los Alamos, NM, USA
Stettler@lanl.gov

Abstract

Field Programmable Gate Array devices have become the technology of choice in small volume modern instrumentation and control systems. These devices have always offered significant advantages in flexibility, and recent advances in fabrication have greatly increased logic capacity, substantially increasing the number of applications for this technology. Unfortunately, the increased density (and corresponding shrinkage of process geometry), has made these devices more susceptible to failure due to external radiation. This has been an issue for space based systems for some time, but is now becoming an issue for terrestrial systems in elevated radiation environments and commercial avionics as well. Characterizing the failure modes of Xilinx FPGAs, and developing mitigation strategies is the subject of ongoing research by a consortium of academic, industrial, and governmental laboratories. This paper presents background information of radiation effects and failure modes, as well as current and future mitigation techniques. In particular, the availability of very large FPGA devices, complete with generous amounts of RAM and embedded processor(s), has led to the implementation of complete digital systems on a single device, bringing issues of system reliability and redundancy management to the chip level. Radiation effects on a single FPGA are increasingly likely to have system level consequences, and will need to be addressed in current and future designs.

I. INTRODUCTION

For some time, field programmable gate arrays (FPGAs) have been an attractive choice in small volume instrumentation and control system electronics. Recent advances in process technology have greatly increased the logic capacity of these devices, allowing their use for sophisticated processing applications, and at the same time reducing the cost of entry level devices to the point where they can compare favourably with custom devices for larger volume applications. In addition, FPGAs based on SRAM technology can be reconfigured at will, allowing unmatched flexibility in the face of changing requirements. The latest offerings from the FPGA vendors also include embedded processors, allowing an entire system on a chip to be fielded in a single programmable device.

However, the advanced process technology which makes these devices possible comes with a price, far greater susceptibility to upset by radiation. Radiation (energetic charged particles) affects semiconductor devices by leaving a wake of electron/hole pairs along the path of the particle through the silicon. If this charge is deposited in a control structure, such as a transistor gate, it can momentarily change the output state. If the affected transistor is part of a persistent circuit, such as a flip flop or RAM cell, the change becomes permanent. In addition, as part of the modern CMOS fabrication process, parasitic bipolar transistors are formed in the substrate of the device. These parasitic transistors can also be activated by radiation induced charge, creating a virtual short from power to ground and damaging the device. Fortunately, the lower internal voltage supplies of modern devices (1.5V in current designs) make this second type of failure far less likely, since the parasitic transistors are barely forward biased even if they are activated.

Mitigation techniques for FPGA logic depend somewhat on the underlying technology of the device. Antifuse, or one time programmable, FPGA's logic and routing are insensitive to upset, and one only needs to deal with persistent logic structures. In SRAM FPGAs however, configuration memory cells hold the definition of the user logic and routing information as well, exposing the definition of the logic as well as any persistent logic to the possibility of upset. In fact, over 90% of SRAM cells in a typical SRAM FPGA control logic configuration and routing, making this by far the dominant failure mode.

Scrubbing, or verifying configuration memory content, is commonly used to detect and repair configuration upsets. Many FPGAs allow partial reconfiguration, allowing the part to be "repaired" without resetting the entire device. Using the fastest readback modes, a typical 1 million gate part (Xilinx XQVR1000) can be verified at approximately30Hz.

In addition to scrubbing, triple module redundancy (TMR) design techniques are typically used to provide immunity from a single upset. The assumption behind this approach is that upsets will be relatively rare, and that no more than one SRAM bit will be upset by a single particle's interaction with the device. It turns out that this assumption is not a particularly good one, but it doesn't negate the value of TMR for vastly improving the reliability of user logic.

II. UPSETS

The susceptibility and functional effects of upsets vary depending on the family of FPGA devices used. At this point the focus will be on the Xilinx vertex and vertexII SRAM based families, due to the significant amount of characterization and experience fielding these devices in elevated radiation environments. In addition, redundancy techniques will focus on recovery from single event upsets (SEUs), where only one SRAM bit is altered by a single particle. Although it is known that multiple event upsets (MBUs) can occur in significant numbers, the single event model is still a very useful assumption.

Another type of upset, single event latchup (SEL), occurs when one of the parasitic bipolar transistors created as a by product of the CMOS fabrication process is activated by a charged particle. This type of upset is very serious, and results in a short being created from power to ground on the chip. Special fabrication processes using epitaxial substrate eliminate the parasitic bipolar transistors, and the susceptibility to SEL. In addition, increased density of newer device families and the corresponding lower core voltage is making SEL less likely. Due to the lower core voltage, it is significantly more difficult to forward bias the parasitic bipolar transistors. Virtex II devices, which have a 1.5V core voltage, are latchup immune to 160 MeV (protons)[1]. As core voltages drop towards 1V, devices will become virtually immune to this type of failure.

A. FPGA Architecture

The Xilinx vertex series FPGA family provides a variety of logical resources to implement user designs[2]. The core of the device consists of an array of configurable logic blocks (CLBs), each of which consists of two slices. Each slice contains two 4 input look up tables for logic generation, two flip flops, and arithmetic carry and clocking functions. Flanking the CLB matrix are two columns of dual port RAM, divided into 4Kbit blocks. The edges of the device are populated by input/output blocks, which support several I/O standards.

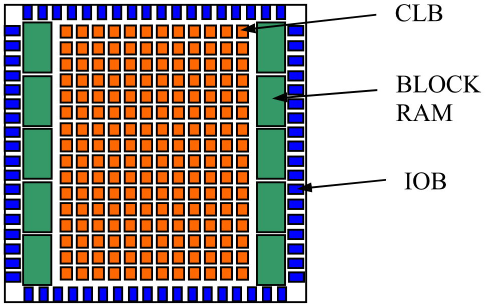

Figure 1. Simplified View of Xilinx Vertex FPGA[2]

The Xilinx vertex II series FPGA family has a similar architecture to the virtex with the addition of hardware multipliers to the block RAM. In addition, the virtex II is fabricated using a smaller process geometry, yielding larger gate counts and higher speed operation.

In addition the resources for implementing user logic, a large amount of programmable routing is available for connecting the CLBs, block RAM, IOBs, and other functional elements.

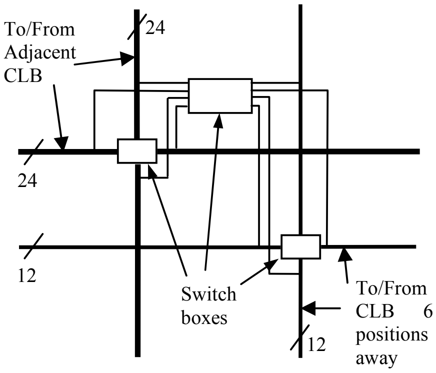

Figure 2. Simplified Virtex CLB routing[2]

The routing for the CLB array consists mainly of wires that connect to the adjacent CLBs, and to CLBs 6 rows or columns away (known as hex wires). Switch boxes connect the wires via a matrix of pass transistors, known as programmable interconnect points (or PIPs) and buffers.

It is apparent even from the rudimentary discussion of the virtex architecture that the amount of configuration information is substantial, and will easily dominate the user design when measured in RAM bits utilized. In addition, the routing resources utilize approximately 70% of the available silicon. Table 1 defines typical memory utilization for a virtex device.

Table 1: Virtex XCV1000 memory Utilization[2][3]

Memory Type	**# of bits**	**%**
Configuration	5,810,048	97.4
Block RAM	131,072	2.2
CLB flip-flops	26,112	0.4

As can be seen in table 1, the configuration information dominates the content of the RAM on these devices, and thus also dominates the cross section for radiation upsets.

B. SEUs

Single event upsets, or SEUs, occur when a RAM cell's state is changed due to exposure to energetic particle(s). The function of the particular RAM cell will determine the effect. The effects can be altered user logic state, or content, altered logic configuration, where the function of the logic is changed, or altered routing, where the connection between logic elements is changed.

Altered logic content is perhaps the most straightforward effect, and results in a flip-flop transitioning to the incorrect state. If the user logic is not part of a feedback element, the result will be a "glitch", or momentary bad data. In almost all cases, this momentary failure will go unnoticed. However, if the user logic is part of a feedback element (a counter bit, for example), the error will be persistent, and very likely to cause undesirable operation. In this case, a device reset may need to be performed to restore proper operation.

Another manifestation of altered logic content is when a global device function becomes activated due to an SEU. FPGAs support global functions for programming, initialization, and debug. Activating these functions improperly can cause the device to reset or enter configuration mode, immediately interrupting all user functionality. These events are known as single event functional interrupts (SEFIs), and always require a complete reconfiguration for recovery. In many cases, the only indication a SEFI has occurred (other than complete loss of functionality) is a configuration readback that indicates a huge number of errors, usually indicative of configuration memory erasure.

Altered logic configuration bits change the function of the user logic, and are always persistent. These errors are detectable via configuration memory readback, and easily repairable via partial reconfiguration. However, the user logic will likely malfunction randomly during the time the logic is altered, designs with a high degree of state interdependency may need a device reset to restore proper operation.

Altered routing is statistically the most likely effect of an SEU, but also the least likely to cause a logic failure. Since most of the routing is unused, even in designs which fully utilize the logic resources of a device, there is a high probability that the upset will connect unused wires, and be a "don't care" as far as user logic is concerned. In many cases, the only observable effect of routing faults is a gradual rise in device power consumption as parasitic segments are added to the design. These parasitic loads have the effect of degrading the timing margin of the design, and eventually will cause logic failures if not repaired by partial configuration. Of course, shorts or opens in wires utilized for the design have an immediate persistent effect.

In Xilinx's vertex architecture, another failure mode is possible due to the implementation of many of the logical constants in user logic[4]. By using weak keeper circuits, or half latches, to produce constant logic values, more expensive logic resources such as look up tables (LUTs) can be conserved, allowing greater logic density. At the chip level, half latches are present on many of the inputs to I/O, RAM, clocking, and logic resources. Easily overcome by the drive of an active circuit, they come into play only when the input is left unconnected. From a chip design standpoint, they are an efficient and ubiquitous source of constant "0" and "1" throught the device.

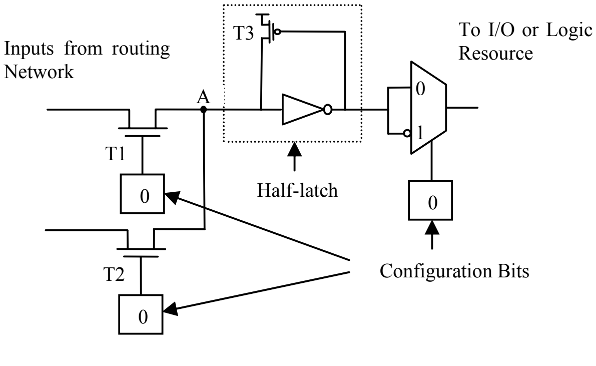

Figure 3. Simplified half-latch circuit in the virtex Architecture[4]

Figure 3 is a simplified schematic of the half latch at the circuit level. The half latch, or weak keeper, structure consists of a weak PMOS transistor (T3) and inverting buffer between the input multiplexer to the I/O or logic, and the two NMOS transistors (T1 and T2) to the FPGA routing network. The circuit is designed to hold a logic '1' at node A when both T1 and T2 are off. When either of the input transistors is on, they easily overwhelm T3, allowing node A to follow the state of the routing network signal. The mux which follows the half latch allows the circuit to supply either a logic '1' or '0' to the following logic as required by the user design. To insure proper initialization, all of the half-latches in the device are driven to logic '1' (at node A) as part of the device start up sequence.

Unfortunately, these half-latch structures are susceptible to radiation upset. When upset, the output of the half-latch inverts and the circuit remains in this state for a considerable length of time. Although it is possible for the half-latch to recover due to leakage through T3, this behaviour has not been studied in detail. Since the state of the half-latch cannot be discerned through reading the configuration bits, this type of upset cannot be detected through readback, or repaired by any means except a complete device reprogramming cycle, which will re-initialize all half-latches.

The solution to the problem created by half-latches is to alter the FPGA design to remove them, or modify the software synthesis tools with a switch to force other resources to be used for constant generation. Neither of these has been implemented to date. A tool has been written at Los Alamos to replace half-latch structures with observable and repairable constant sources[4], but as with any add-on tool, there are limitations and caveats to its use. The half-latch remains a trouble spot in the Xilinx vertex architecture that awaits a proper fix by Xilinx.

III. Mitigation

Mitigation involves both repairing altered configuration and logic design that is resistant to failure. Repairing altered configuration involves reading back the configuration from the FPGA, and comparing it to a known good copy. Xilinx vertex FPGAs allow partial readback and configuration, facilitating efficient repair of configuration memory. Failure resistant logic design involves redundancy in user logic. The most widely used technique involves triple module redundancy (TMR), which provides immunity from a single configuration or state upset[5].

A. Scrubbing

Scrubbing refers to the periodic readback of the FPGA's configuration memory, comparing it to a known good copy, and writing back any corrections required. By periodically scrubbing a device, maximum limits may be placed on the period of time that a configuration error can be present in a device. In some applications, using this technique alone is enough to satisfy operational requirements.

Xilinx vertex devices support readback and configuration modes that operate on only a portion of the device[2]. This is known as partial readback and configuration, and allows a more efficient means of repairing configuration upsets. Unlike complete configuration, partial configuration does not reset the device, which allows the uninterrupted operation of user logic.

B. TMR

Triple module redundancy, or TMR, is an effective technique creating fault tolerant logic[5][6]. In TMR, the logic of the design can simply be triplicated, with redundant voters on the output, but this is seldom the best implementation. In order to recover smoothly from logic upsets, the internal state of the design must be restored to the repaired logic. This is best illustrated by a simple example.

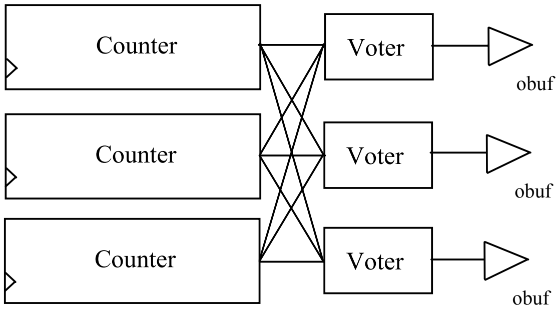

Figure 4. TMR counter

In the TMR counter design in figure 4, any single upset failure will be successfully tolerated (note that voter failures will be caught by a final off-chip voter assumed to be rad-hard), but there is still a potential problem. If the failure upsets the internal state of the counter, repairing the upset will not be enough – the state of the repaired counter must be resynchronized to match the other two. Of course, this can be accomplished by a global reset, but this is not desirable in many applications.

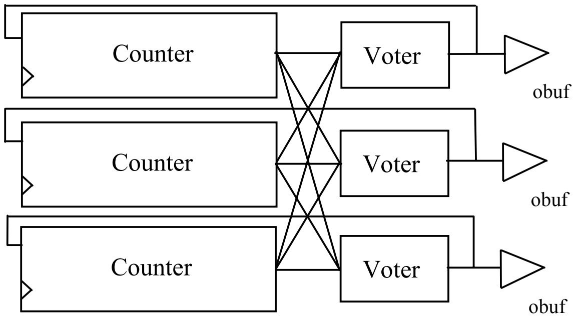

Figure 5. Feedback counter with TMR in the feedback path

In the feedback counter in figure 5, the state of the counters is obtained from the output of the voters. This feature has the effect of always presenting the correct state to the counter logic, resulting in the logic being self restoring in the event of an upset and subsequent repair. This form of TMR is desired in most applications with internal state dependencies[6]. At this time, Xilinx has a TMR tool in beta test that automatically applies this form of TMR to most user designs.

TMR does not come without a price. Obviously, designs are at least 3 times as large as a non TMR design, and suffer from speed degradation as well (25% in the counter example)[7]. In particular, feedback TMR degrades the speed of operation by introducing a longer feedback path including the voter. Power consumption is also tripled along with the logic.

The underlying assumption of TMR is that only one upset will occur within a given logic block. This is not always a good assumption to make. In virtex II devices, recent testing resulted in approximately .3-.5% of upsets causing multiple bit upsets within the device[8]. Also, the scrubbing frequency defines the rate at which upsets can be detected – this combined with the rate of upsets provides the actual tolerance of the design. This being said, a proper TMR implementation combined with fast scrubbing can provide better than an order of magnitude increase in the radiation tolerance of a given design.

IV. ALTERNATIVES

SRAM based FPGAs are widely used due their density, cost, and in system programmability. However, another option exists in antifuse technology. In addition, antifuse vendors also offer rad-tolerant versions of some product lines which are intrinsically resistant to SEUs to a degree not available in SRAM devices.

A. Antifuse

Antifuse has several advantages to SRAM. These one time programmable devices use physical shorts between metal routing layers to configure their logic. Aside from being faster and more power efficient than comparable SRAM based switches, they are immune to radiation effects. As can be seen from table 1, this eliminates 97% of sensitive bits (in a device of similar density). Application of TMR in an antifuse part is usually less costly in resources, since in general only the state dependent logic needs to be triplicated. The more efficient logic switching results in lower power consumption and quieter operation, important considerations in mixed mode designs.

The main drawback of antifuse is its one time programmability; it is best suited for applications where the initial requirements are stable and not expected to evolve over time. In addition, antifuse parts are not available in as high logic densities as SRAM devices.

B. *Rad-hard*

Some antifuse vendors (notably Actel[9], although there are others), provide rad-hard versions of some of their product lines. These devices are even more radiation tolerant than standard antifuse, with internal flip flops TMRed in silicon (a device by Quicklogic/Aeroflex even has hardware TMRed RAM arrays)[10]. These devices completely remove the need to TMR user designs, and are suitable for the highest reliability requirements. However, the selection of devices is constrained, and is not available in the highest densities supported by antifuse.

V. Conclusion

Modern FPGAs are already at the heart of most low to mid volume electronic systems, and their capabilities will continue to improve in the future. However, with the continuous shrinking of device geometry, the susceptibility to radiation upset will continue to grow. Upset tolerant design techniques, both from a system and device level, are already becoming a requirement for many systems.

SRAM FPGAs, such as the Xilinx vertex series, have long been favoured due to their unmatched performance, density, and in system programmability, yielding a powerful and flexible solution chosen by many designers. However, their relatively high susceptibility to radiation upset is a factor to be considered in a growing number of environments. The added complexity of scrubbing and TMR needs to be weighed against the advantage of unlimited in system programmability when designing a system that needs to function in an elevated radiation environment. In addition, system on a chip designs including soft processors and sophisticated peripherals, possibly including purchased IP cores (commercially available logic functions included in a user design), can be very difficult to TMR. Verifying the performance and reliability of a TMRed design can be non trivial, and negate some of the advantages of off the shelf IP.

Antifuse FPGAs are a natural choice for elevated radiation environments, but are simply unsuitable for applications that require in system reprogrammability. However, even in SRAM FPGA systems, there is often an anitfuse part tasked with scrubbing the SRAM devices. As stated previously, in systems where requirements are stable, and density requirements are not beyond their capacity, antifuse remains an attractive solution for radiation tolerance and power consumption reasons.

VI. References

[1] Private communication, Xilinx XRTC (Xilinx radiation test consortium) 9/04 unpublished

[2] “Virtex architecture guide”, Xilinx, San Jose, CA, 9/00

[3] “Single-Event Upsets in SRAM FPGAs”, M.Caffrey et al, Military and Aerospace Applications of Programmable Logic Devices (MAPLD), 9/02

[4] “SEU Mitigation for Half-Latches in Xilinx Virtex FPGAs”, P. Graham et al, IEEE Transactions on Nuclear Science (journal) in relation to IEEE Nuclear and Space Radiation Effects Conference, 7/03

[5] “Triple module redundancy design techniques for Virtex FPGAs” (xAPP197, v1.0), C, Carmichael, Xilinx, 11/01

[6] “Hardness By Design Techniques for Field Programmable Gate Arrays”, M. Wirthlin et al, 11th Annual NASA Symposium on VLSI Design, 5/03

[7] “Evaluating TMR Techniques in the Presence of Single Event Upsets”, N. Rollins et al, Military and Aerospace Programmable Logic Devices International Conference”, 9/03

[8] Private Communication, Test data from LANL Virtex II proton test at UC Davis 6/04, unpublished, J Krone, 9/04

[9] “Military and Aerospace product line”, Actel, www.actel.com

[10] “UT6250 RadHard Eclipse”, Aeroflex, www.aeroflex.com