Fault Tolerance Implementation within SRAM Based FPGA Designs based upon Single Event Upset Occurrence Rates

Melanie Berg:
Chief Staff Electrical Engineer and Principle Investigator
NASA Goddard Space Flight Center Radiation Effects and Analysis Group/ MEI Technologies

1 ABSTRACT

Emerging technology is enabling the design community to consistently expand the amount of functionality that can be implemented within Integrated Circuits (ICs). As the number of gates placed within an FPGA increases, the complexity of the design can grow exponentially. Consequently, the ability to create reliable circuits has become an incredibly difficult task. In order to ease the complexity of design completion, the commercial design community has developed a very rigid (but effective) design methodology based on synchronous circuit techniques.

In order to create faster, smaller and lower power circuits, transistor geometries and core voltages have decreased. In environments that contain ionizing energy, such a combination will increase the probability of Single Event Upsets (SEUs) and will consequently affect the state space of a circuit. In order to combat the effects of radiation, the aerospace community has developed several "Hardened by Design" (fault tolerant) design schemes.

This paper will address design mitigation schemes targeted for SRAM Based FPGA CMOS devices. Because some mitigation schemes may be over zealous (too much power, area, complexity, etc....), the designer should be conscious that system requirements can ease the amount of mitigation necessary for acceptable operation. Therefore, various degrees of Fault Tolerance will be demonstrated along with an analysis of its effectiveness.

2 SYNCHRONOUS DESIGN AND SOFT ERRORS WITHIN CMOS TECHNOLOGY

The foundation of Synchronous Methodology is to assure deterministic behavior. One of the major key components of synchronous design techniques is employing clocks as a source of control. Designs are created such that portions of functionality can be completed within one clock cycle. This rule of thumb establishes a discrete means of verification, determinism, and traceability.

Due to the reduction in core voltage, decrease in transistor geometry, and increase in switching speeds, CMOS transistors have become more susceptible to incurring faults. Upsets can be in the form of a flipped bit (DFF or memory) or a transient (glitch) within a combinatorial logic path (i.e. functional logic, clock, reset, receive buffers, transmit buffers, etc....). Soft Flipped bit faults are labeled as Single Event Upsets (SEUs). Glitches within combinatorial circuitry are identified as Single Event Transients (SETs). Terrestrial devices have fault vulnerability mostly due to: alpha particles (from packaging and doping) and Neutrons (caused by Cosmic Ray Interactions that enter into the earth's atmosphere). Designs targeted for higher altitude operations (Aerospace and Military) are more prone to upsets caused by heavy ions and protons.

What is of major concern to a designer is that these upsets are unpredictable in occurrence and therefore asynchronous in nature. A great deal of research has been done concerning memory and DFF bit upsets and their associated fault tolerant solutions (generally assumes synchronous fault generation and synchronous correction). However, as system clock speeds increase, the probability of capturing combinatorial transients also increases. This error cross section can be greater than the probability of SEU error rates at high speed operation. The major caveat is that due to the asynchronous characteristics of the transients, metastability can occur and cause major system malfunction. Unfortunately, finding a 100% effective solution is impossible when protecting circuitry from asynchronous events. However, the designer can reduce the probability of error by creating "clean" synchronous designs while strategically choosing fault tolerant mitigation techniques.

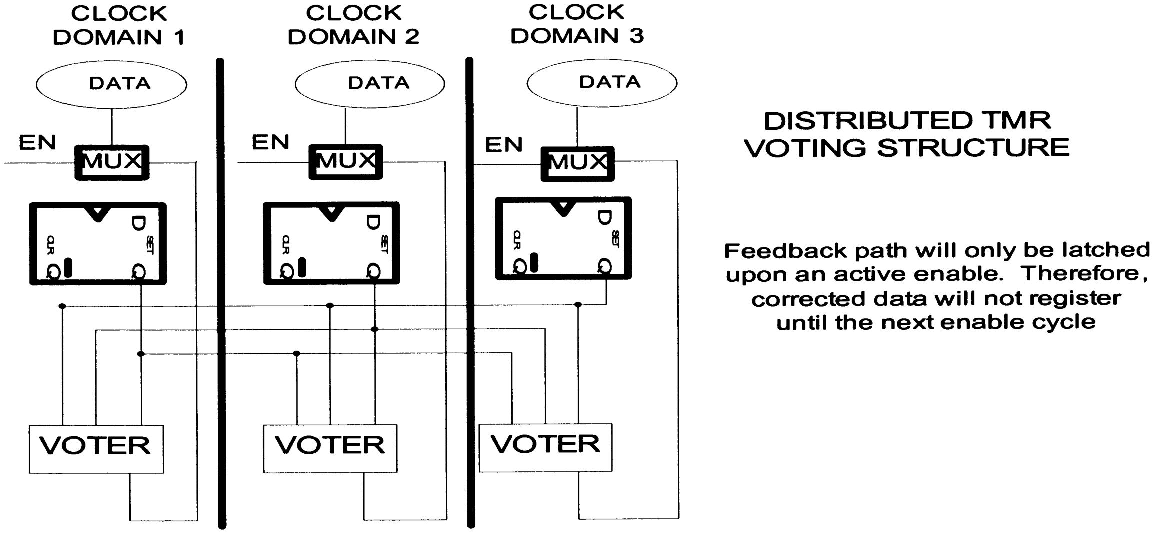
Careful analysis of the points of failure (within the combinatorial paths and sequential components) must be performed in order to determine if the chosen error correction (or detection) scheme is beneficial or has created a more susceptible design. The analysis includes: definition of SEU and SET error cross sections, knowledge of maximum transient widths (based on process, implementation and operating environment), point of failure identification, probability analysis of multiple failures upon SEU/SET generation (due to fan – out and/or metastability), and probability of SET propagation (can be depleted on high capacitance nets or gates).

3 FAULT TOLERANCE AND SRAM BASED FPGAS

The definition of Fault Tolerance is the ability to mask or recover from erroneous conditions in a system once an error has been detected. The degree of fault tolerance implementation is defined by your system level requirements... i.e. specifications that clearly state acceptable behavior upon error.

3.1 SRAM Based FPGAs and Aerospace Radiation Environments

Due to the higher SEU and SET error rates within Aerospace and Military environments, the level of necessary fault tolerance can be very high. Proposed schemes for SRAM based FPGA designs can be very complex and almost impossible to implement. A maximally fault tolerant system (for SRAM based technology) would require: A scrubbing circuit for the SRAM configuration space (usually implemented within an anti-fuse FPGA technology), redundant memory (containing the configuration bit-stream), and a distributed logic voting scheme (tripling the design including I/O, clocks, resets, and all functional logic paths). Such an implementation can lead to extremely complex board designs with signal integrity issues and to unverifiable designs (very dangerous). Due to the lack of verification integrity and the extreme increase in complexity and cost, designers are investigating partial mitigation techniques with SRAM Based FPGAs (when acceptable to project specifications).



Feedback path will only be latched upon an active enable. Therefore, corrected data will not register until the next enable cycle

Figure 1: Internal Implementation of Distributed Triple Mode Redundancy (TMR)

3.2 SRAM Based FPGAs and Terrestrial Environments

For terrestrial environments, requirements are more relaxed. The system may only necessitate scrubbing. However, partial mitigation strategies (for the functional logic) may also be necessary. A very popular example of partial mitigation is Error Detection and Correction circuitry. The point that is incredibly overlooked is that due to the ever increasing SET rates (due to the faster logic and faster clock frequencies), current EDAC implementations may increase the fault cross section rather than decrease it. As demonstrated in Figure 2, the EDAC circuitry will more likely have a higher probability of SET generation due to its transistor level complexity. At high operational frequencies, a SET within the feedback path can get caught by Several DFFs and cause uncorrectable results or metastability. An alternate approach can be to triple the EDAC logic, vote, and then feedback. The error cross section is now decreased from a higher probabilistic SET cloud of logic to a very small cross section consisting of the transistors that comprise the voter logic.

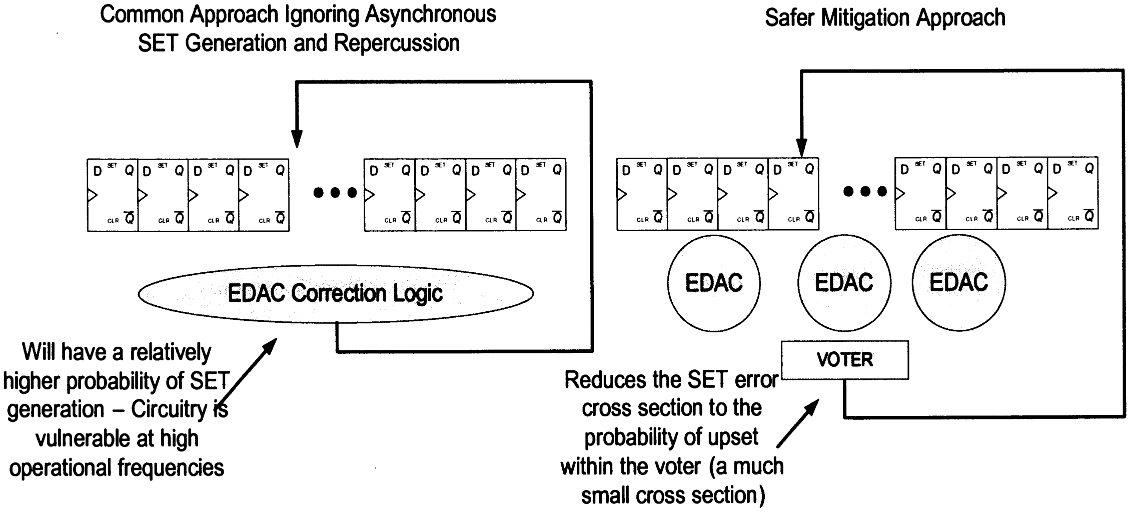

Common Approach Ignoring Asynchronous SET Generation and Repercussion
Safer Mitigation Approach
D SET Q
CLR Q̄
EDAC Correction Logic
EDAC
EDAC
EDAC
VOTER
Will have a relatively higher probability of SET generation – Circuitry is vulnerable at high operational frequencies
Reduces the SET error cross section to the probability of upset within the voter (a much small cross section)

Figure 2: Simplified Depiction of EDAC Circuitry and Proper Mitigation Techniques Concerning Asynchronous Fault Generation.

4 SUMMARY

CMOS technological improvements have lead to transistor level vulnerability to ionized particles. Based on the system level requirements and the operational environment, careful analysis of the proposed fault tolerant implementation must be performed. There exist mitigation strategies that can be overly complex and very costly. It is the designers' responsibility to perform a detailed trade space of the necessary level of redundancy vs. its expense of implementation. This paper investigates types of fault tolerant schemes, and their effectiveness towards reaching given specification objectives.