Low Cost Multiple Bit Upset Correction in SRAM Based FPGA Configuration Using Erasure Code

T. Thangam#1 and P. Vaisanthi#2

1Associate Professor, ECE Department, PSNA College of Engineering & Technology, Dindigul.

2PG Scholar, ECE Department, PSNA College of Engineering & Technology, Dindigul, Anna University, India.

1Mail id: Thangam7280@gmail.com

2Mail id: januvaisanthi@gmail.com

Abstract

Radiation-induced multiple bit upsets are a major reliability concern in nanoscale technology nodes. Occurrence of such errors in the configuration frames of a field programmable gate array device permanently affects the functionality of the mapped design. Occasional configuration scrubbing joined with a low-cost error correction scheme is an efficient approach to avoid such a permanent effect. Existing procedures utilize mistake redress codes with significantly high overhead to moderate MBUs in design outlines. In this paper, we display a minimal effort mistake recognition code to identify MBUs in configuration frames and in addition a nonexclusive scrubbing scheme to reproduce the mistaken configuration frame based on the concept of erasure codes.

Index Terms—FPGA, multiple bit upsets, reliability, soft errors.
Introduction

SRAM-BASED field-programmable gate arrays (FPGAs) are widely used in a variety of application domains due to short time-to-market time, flexibility, high density, and cost-efficiency. In any case, expanding transistor check per chip combined with the lassened working voltage in the previous years brings about an exponential development in soft error rate digital circuits. Considering the expansion of FPGAs in a variety of safety- and mission critical applications, it is crucial to mitigate their susceptibility to these sorts of errors.

In order to meet the ever-increasing performance and power demands, FPGAs are, FPGAs are normally manufactured utilizing the most advanced technology nodes. As of late, FPGAs in view of a 14-nm innovation denser integration schemes, for example 3-D die stacking, have been declared. In such small device geometries, a single radiation-induced molecule strike is likely to affect several adjacent cells in a memory array, leading to a multiple bit upset. Considering the fact that the MBU rate in nanoscales is comparable with the single event upset, a proper plan is required to detect and correct multiple errors in memory arrays. More particularly, SRAM-based FPGAs are more inclined to soft errors as a particle strikes in a configuration frame permanently affects the functionality of the mapped design. Since the configuration frames constitute the majority of SRAMs in an FPGA device mitigation of MBUs in configuration frames is of conclusive significance. Several schemes have been presented to address the increasing soft error concern in the FPGA configuration frames. The main objective of these schemes is to reduce error latency, and hence, to avoid error accumulation within configuration frames. Costly modular redundancy is a traditional procedure to tolerate and soft errors in both configuration frames and functional logic. In any case, collected errors in both data and configuration bits significantly in it the mean time to failure of such schemes.

Another method is to enhance the design out configuration frame circuitry for soft errors as detailed. Be that as it may, such hardening systems are not executed in the existing FPGA devices because of their excessive area overheads. In this way, a low-cost solution is required to correct erroneous configuration frames during operation. The combination of configuration scrubbing and error correction codes is an effective solution to detect and correct radiation-induced transient errors in configuration bits.

Static random access memory

SRAM is computer memory that requires a consistent power flow to hold data. Power utilization changes widely based on how frequently the memory is accessed. Although quicker than DRAM, SRAM is more costly and holds less data per unit volume. Along these lines, it is more commonly used in cache and video card memory only.

Soft error

Soft error is a kind of error where a signal or datum isn't right. Errors might be caused by a defect, normally understood either to be a mistake in design or construction, or a broken component. A soft error is also a signal or datum isn't right, however isn't expected to imply such a mistake or breakage. After observing a soft error, there is no implication that the system is any less reliable than before. In the rocket business, this sort of blunder is called a single event upset.

Soft error mitigation

A designer can endeavor to limit the rate of delicate blunders by wise gadget configuration choosing the correct semiconductor, package and substrate materials, and the correct gadget geometry. Frequently, be that as it may, this is limited by the need to decrease gadget size and voltage, to increase operating speed and to reduce power dissipation. The susceptibility of devices to upsets is described in the industry.

One strategy that can be utilized to lessen the soft error rate in advanced circuits is called radiation hardening. This includes expanding the capacitance at selected circuit nodes in order to increase its effective Qoff value. This lessens the scope of molecule energies to which the logic value of the node can be upset. Radiation hardening is often accomplished by increasing the size of transistors
who share a deplete/source locale at the node. Apparatuses and models that can foresee which hubs are most defenseless are the subject of past and eb and flow inquire about in the territory of delicate mistakes.

Detecting soft errors

There has been work addressing soft errors in processor and memory assets utilizing both hardware and software procedures. A few research efforts addressed soft errors by proposing error detection and recovery via hardware-based redundant multi-threading. These methodologies utilized exceptional hardware to replicate an application execution to identify errors in the output, which expanded hardware design complexity and cost including high performance overhead. Software-based soft error tolerant schemes, on the other hand, are flexible and can be applied on commercial off-the-shelf microprocessors. Many works propose compiler-level direction replication and result checking for delicate mistake recognition.

Correcting soft error

Designers can acknowledge that delicate mistakes will happen and design systems with appropriate error detection and correction to recover smoothly. Ordinarily, a semiconductor memory configuration may use forward error correction, incorporating redundant data into each word to create an error correcting code. On the other hand, move-back error correction can be used, detecting the soft error with an error-detecting code such as parity, and rewriting correct data from another source. This system is frequently utilized for compose through store recollections.

Soft errors in logic circuits are some of the time detected and corrected utilizing the procedures of fault tolerant plan. These frequently incorporate the utilization of redundant circuitry or computation of data, and ordinarily come at the cost of circuit area, decreased performance, and/or higher power consumption. The idea of triple secluded repetition (TMR) can be utilized to guarantee high delicate mistake unwavering quality in rationale circuits.

The plan of mistake discovery and redress circuits is helped by the way that delicate blunders more often than not are confined to a little zone of a chip. Typically, just a single cell of a memory is influenced, although high energy events can cause a multi-cell upset. Conventional memory format usually places one bit of many different correction words adjacent on a chip. In this way, even a multi-cell upset leads to only a number of separate single-bit upsets in multiple correction words, rather than a multi-bit upset in a single correction word. In this way, a mistake remedying code needs just to adapt to a solitary piece in blunder in every adjustment word keeping in mind the end goal to adapt to all probable delicate blunders. The term ‘multi-cell’ is used for upsets affecting multiple cells of a memory, whatever correction words those cells happen to fall in. Multi-bit is used when multiple bits in a single correction word are in error.

Planners can acknowledge that delicate blunders will happen, and outline frameworks with fitting mistake location and revision to recoup effortlessly. Regularly, a semiconductor memory configuration may use forward blunder rectification, consolidating repetitive information into each word to make a mistake adjusting code. On the other hand, move back blunder amendment can be utilized, recognizing the delicate mistake with a mistake distinguishing code, for example, equality, and modifying right information from another source. This system is regularly utilized for compose through reserve recollections.

FPGA configuration frame

A field-programmable gate array is a coordinated circuit intended to be arranged by a client or a planner in the wake of assembling subsequently "field-programmable". The FPGA design is for the most part determined utilizing an equipment depiction dialect, like that utilized for an application-particular circuit.

FPGAs contain a variety of programmable rationale squares, and a progression of reconfigurable interconnects that enable the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be arranged to perform complex combinational capacities, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which might be straightforward flip-failures or more total squares of memory.
Contemporary field-programmable gate arrays have huge assets of logic gates and RAM blocks to execute complex digital computations. As FPGA outlines utilize fast I/O rates and bidirectional data buses, it becomes a challenge to verify correct timing of valid data within setup time and hold time. Floor planning enables resource allocation within FPGAs to meet these time constraints. FPGAs can be used to implement any logical function that an ASIC could perform. The capacity to refresh the functionality after shipping, partial re-configuration of a portion of the design and the low non-recurring engineering costs relative to an ASIC design, offer advantages for many applications.

**Erasure codes**

Eradication coding (EC) is a technique for information assurance in which information is broken into fragments, extended and encoded with information data pieces and stored across a set of different locations or storage media.

The objective of eradication coding is to empower data that becomes debased eventually in the disk storage process to be recreated by utilizing data about the data that's stored elsewhere in the cluster. Eradication codes are regularly utilized rather than traditional RAID because of their ability to reduce the time and overhead required to reconstruct data. The drawback of eradication coding is that it can be more CPU-intensive, and that can translate into increased latency.

Deletion coding can be valuable with huge amounts of data and any applications or systems that need to ensure failures, for example, disk array systems, data grids, distributed storage applications, object stores and archival storage. One common current use case for erasure coding is object-based distributed storage.

**Low-cost MBU detection**

The main idea of this paper is to exploit erasure codes to recover the contents of the incorrect configuration frame. By the by, since erasure codes can’t detect errors, an effective detection technique is required too. Therefore, each configuration frame has to be equipped with a low-cost error detection code. A scrubber unit periodically examines the configuration frames for possible errors. Once an error is detected, by accepting that the erroneous frame is erased, its contents are recovered utilizing a erasure code.

**Error detection and correction**

The general thought for accomplishing blunder identification and revision is to some excess to a message, which receivers can use to check consistency of the conveyed message, and to recover information that has been determined to be corrupted. Blunder location and rectification plans can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits which are gotten from the information bits by some deterministic calculation.

On the off chance that lone mistake identification is required a collector can basically apply a similar calculation to the got information bits and contrast its yield and the got check bits; if the qualities don’t coordinate, a blunder has happened sooner or later amid the transmission. In a system that uses a non-systematic code, the first message is
transformed into an encoded message that has in any event the same number of bits as the first message.

Good error control execution requires the plan to be selected based on the characteristics of the communication channel. Regular channel models include memory-less models where errors occur randomly and with a specific likelihood, and dynamic models where errors occur primarily in bursts. Therefore, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst-error-detecting/correcting. A few codes can also be suitable for a mixture of random errors and burst errors.

If the channel capacity cannot be determined, or is highly variable, an error-detection scheme may be combined with a system for retransmissions of erroneous data. This is known as automatic repeat request, and is most notably used in the Internet. An alternate approach for error control is hybrid automatic repeat request, which is a combination of ARQ and error-correction coding.
Performance Analysis

The Figure given below is shown that there is a considerable reduction in time and delay based on the implementation results which have been done by using Spartan-3 processor. The proposed algorithm significantly reduces area when compared to the existing system.

Table 7 Comparison Table

<table>
<thead>
<tr>
<th>NO</th>
<th>Parameter</th>
<th>Existing</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Size</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$4 \times 110$</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Conclusion

In this work, a cost-efficient method based on erasure codes, MBU can be used to detect and correct the error in SRAM based FPGA configuration frame. This was the method is implemented in a soft core with the user design and does not require any changes in the existing FPGA architecture. Compared with the previous method, my work will provide highest level of MBU. In my proposed work reduce area and delay by using low cost method for erasure codes.

References


