Microelectronics Reliability 78 (2017) 85–92

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

Estimating the SEU failure rate of designs implemented in FPGAs in presence of MCUs

Igor Villalta *, Unai Bidarte, Julen Gomez-Cornejo, Jesús Lázaro, Armando Astarloa

University of The Basque Country (UPV/EHU), Department of Electronic Technology, Spain

ARTICLE INFO

Article history:
Received 10 May 2017
Received in revised form 12 July 2017
Accepted 4 August 2017
Available online xxxx

Keywords:
SEU
FPGA
Emulation
Fault injection
SBU
MCU

ABSTRACT

Due to the continuous reduction of the transistor size in electronic devices, it is becoming more and more likely for an SEU (Single Event Upset) to provoke a flip on two or more memory cells in SRAM based FPGAs, which is called a MCU (Multiple Cell Upset). Fault injection in the configuration memory of these devices has been used for many years, in order to evaluate their reliability. Emulation of these injections using the bitstream file has always been a simple, fast and cheap solution. Most of the existent SEU emulation tools do not consider the injection of MCUs, and they do not discuss the implication MCUs have on the overall failure rate of the system.
In this work, bitstream based SEU emulators are updated to consider MCUs. It is discussed the necessity of injecting faults on physically adjacent cells, in order to emulate appropriately the effect of MCUs. Adjacent MCU injection has been compared theoretically with an approach considering MCUs as bunches of independent SBUs, as it is done in other emulation platforms. A Zynq-based fault injection platform has been used, in order to apply this way of emulating MCUs and validate the proposal.

1. Introduction

The continuous size reduction of electronic devices has brought a huge amount of advantages in the last decades in terms of performance, area and power saving. As a result, electronic devices have reached lots of applications, including safety-critical systems, where a failure can lead to high economical losses or damage to people or the environment. However, the device size reduction has made electronic devices more and more vulnerable to plenty of harmful effects related to radiation. Evaluating the radiation effects is mandatory when designing a critical system in order to avoid catastrophic failures.

One of the most relevant radiation induced effects in electronic systems is the SEE (Single Event Effect). This effect happens when a single radiation particle interacts with electronic components. Ionizing particles produced by this interaction generate a track of electron-hole (e-h) pairs while they are traveling across the semiconductor. If this e-h pair track is close to a sensitive part of the circuit, an undesired parasitic current can be generated.

The most notable SEE is the SEU (Single Event Upset). This happens when the parasitic current introduces or removes electric charge from one or more memory cells. If the charge threshold is exceeded, the logical value stored by this cell is flipped. When the SEU affects only a single bit it is called SBU (Single Bit Upset), and it is called MCU (Multiple Cell Upset) when more than one memory cell is affected. The term MBU (Multiple Bit Upset) is also used in many works. It refers to an MCU that affects two or more bits belonging to the same logical word [8].

The amount of generated electric charge in an SEE is dependent on the LET (Linear Energy Transfer) of the ionized particle that travels through the semiconductor. The LET represents the amount of energy transferred by the particle to the silicon per unit of distance. Heavy-ions are predominant in aerospace applications and have a large LET. Hence, it is very likely for them to provoke an MCU. On the other hand, neutrons and alpha particles are predominant in the terrestrial environment, since high energy ions do not cross the atmosphere. Alpha particles are produced by package impurities and they have a very low LET. Thus, are less likely to provoke an MCU. Neutrons are not ionized particles, but they can react with silicon and boron nuclei generating medium size ions such as Mg, Ne or Al, which can provoke MCUs in certain cases [19].

In FPGAs (Field Programmable Gate Arrays), an SEU can affect the sequential elements of the implemented circuit (flip-flops or Block RAMs), which can be protected by using error detection and correction codes, and the error can be recovered by rewriting the corrupted cell or applying a reset. Besides, SEUs can affect to the configuration memory.

* Corresponding author.
E-mail addresses: igor.villalta@ehu.es, http://det.bi.ehu.es/~apert/index_engl.html (I. Villalta), http://det.bi.ehu.es/~apert/index_engl.html (U. Bidarte), http://det.bi.ehu.es/~apert/index_engl.html (J. Gomez-Cornejo), http://det.bi.ehu.es/~apert/index_engl.html (J. Lázaro), http://det.bi.ehu.es/~apert/index_engl.html (A. Astarloa).

http://dx.doi.org/10.1016/j.microrel.2017.08.003
0026-2714/

This can provoke a functional interrupt that needs a device reconfiguration to be solved.

SEU is the most relevant natural threat that influences the dependability of an SRAM FPGA-based system during its operating life. Its impact has to be evaluated both qualitatively and quantitatively, in order to show compliance with dependability and safety regulations. Radiation campaigns are performed for evaluating the SEU resilience of devices, counting the amount of upsets at the configuration memory. However, these campaigns do not fully characterize the failure rate of particular designs. This happens because many configuration bits of the FPGA are associated with unused resources, and robustness can be incremented by means of redundant architectures.

SEU emulation in FPGAs is a methodology for analyzing the effects of SEEs in particular designs. It is based on programming the device with a corrupted configuration file in order to store wrong values at the configuration memory emulating the effect of SEUs. It is a cheap and simple technique, since no special laboratory instrumentation is required. This strategy can be utilized to characterize the SEU tolerance of particular designs implemented in FPGAs and obtain the failure rate.

Failure rate is a measure of the number of failures per unit of time. It indicates the mean number of failures during a known period of time. In the case of this work, SEU failure rate represents the amount of SEU-induced failures per unit of time. In reliability engineering, failure rate is usually provided in terms of FIT (Failures In Time), which represents the amount of failures in 10^9 h.

All configuration bits do not have a critical impact on the design functionality when they are flipped by an SBU, and the objective of SEU emulation is to obtain the amount of critical bits of the design. In [1] Xilinx defines the parameter DVF (Device Vulnerability Factor), which represents the probability of a configuration bit to be critical for the design. Typical values are between 2% and 10% according to Xilinx [1,10]. The DVF is a key parameter that defines the criticality of a concrete design. The failure rate of the system is calculated from DVF using expression (1).

$$Failure\ rate = Event\ rate^{*}DVF \tag{1}$$

It has been observed in the literature that MCUs have been traditionally ignored by FPGA SEU emulators [11–16]. Factors such as device size reduction and lower supply voltages decrease the critical charge of SRAM cells, increasing the probability of multiple upsets. According to Moore's law, this may continue happening in the following years. Therefore, SEU emulators should consider MCUs even when new silicon trends based on FinFET SRAM cells are presenting promising results [2,3].

The focus of this paper is to generalize SEU emulation in FPGA designs in order to consider MCUs. This work deepens on the causes of the differences between failure rate calculation approaches considering MCUs and not considering them. Mathematical expressions are proposed for estimating the failure rate taking MCUs into consideration. This theoretical model has been validated by means of a real SEU emulation platform.

2. Related work

In order to illustrate the calculation of the SEU failure rate by emulation, the equation presented in the previous section is taken as the starting point (1). This formula expresses that the failure rate of an FPGA design is dependent on two factors, the device vulnerability factor and the event rate.

The event rate is calculated by multiplying the cross section per bit (σ) by the radiation flux (φ) (2). Radiation flux represents the amount of radiation particles per area and time. The cross section is an area that represents the likelihood of interaction between an incident radiation particle and the electronic device. It is calculated experimentally in laboratory facilities by means of radiation beams. The device under study is placed under a beam of radiation particles during a period of time and the failure rate is calculated using Eq. (3). This expression gives the cross section of the device.

Cross section per bit is sometimes calculated by means of number of flipped bits, instead of using number of events, as it is done in [2] (5). This is not fully precise, since an event can flip one or more bits. The relation between these two representations of the cross section is given by the mean number of upsets per event (r) (7), which is dependent on the LET of the particle (6). In this equation P_{SBU} is the percentage of upsets that are SBUs, P_{nMCU} is the percentage of upsets that are MCUs of n bits and n is the size of the MCU.

$$Event\ rate = \int \sigma(LET)^{*}\varphi(LET) \tag{2}$$

$$\sigma = \frac{number\ of\ SEU\ events}{\varphi} \tag{3}$$

$$\sigma_{EVENT} = \frac{number\ of\ SEU\ events}{\varphi^{*}size} \tag{4}$$

$$\sigma_{ERROR} = \frac{number\ of\ flipped\ bits}{\varphi^{*}size} \tag{5}$$

$$r(LET) = P_{SBU} + \sum_{n} n^{*}P_{nMCU} \tag{6}$$

$$\sigma_{ERROR} = \sigma_{EVENT}{}^{*}r(LET) \tag{7}$$

There are multiple works measuring experimentally the cross sections of memories and the probability and size of MCUs in SRAM based FPGAs. In [4], A methodology for measuring the proton and heavy-ion cross-sections for MCUs in Xilinx FPGAs is presented, where Virtex4 family (90 nm) is analyzed. Here, the 1%–3% of the upsets produced by 63.3 MeV protons are MCUs, and for high LET heavy-ions MCUs can reach the 35%. In [5], the Virtex5 family is analyzed and the MCUs are reported between 6 and 10% for protons and about 60% for heavy ions. Similar data is calculated for a Spartan3 device in [6]. In [7], the impact of MCUs in TMR (Triple Modular Redundancy) designs is studied, and it is concluded that the probability of MCUs to corrupt a TMR circuit is 2.6 orders of magnitude more than SBUs. In [8], MCUs are analyzed for the Xilinx 7 series FPGAs (28 nm). In [9], the FIT rate of the UltraScale family is analyzed.

The DVF is the other parameter that influences the failure rate of the system according to Eq. (1). It represents the probability of functional failure when an SEU event happens.

The failure rate value depends on the ratio of unused versus used resources of the FPGA. Designs occupying very few resources may have few critical bits, being less likely for an event to provoke a failure. However, device occupation is not the only parameter that influences the DVF. Designs with similar occupation ratios can have a variation of 100% on the DVF, as it is reported in [10] (page 25). Here, it is mentioned that a typical value for DVF is 5% (1 failure in 20 upsets), but it can reach a 10% in a worst case (1 failure in 10 upsets).

This parameter is measured by means of SEU emulation, which consists of programming the configuration memory of the FPGA with intentional errors emulating the effects of SEUs. It is a cheap and simple technique, since no special laboratory instrumentation is required. Once the error has been injected the functionality of the circuit is verified. If a malfunction is observed the flipped bit is labeled as critical.

There are several works that propose a bitstream based SEU emulation approach [11–16], but none of them consider an MCU type fault injection. In these works multiple upsets are covered as bunches of independent SBUs, since utilized cross section (σ_{ERROR}) considers the amount of flipped bits instead of the amount of events. In this case the DVF is exactly the percentage of critical bits of the design (8). Other works such as [17,18], consider the injection of MCUs on circuits