Abstract — This paper proposes novel system-level protection techniques for feed forward equalizers, which exploit application and system knowledge, resulting in a more intelligent protection with a 71% saving of circuit complexity in comparison to XTMR.

Index Terms — Field Programmable Gate Array (FPGA), Single Event Effects, Fault Injection, Design Cross-section, Adaptive Filters.

I. INTRODUCTION

A dvances in the semiconductor industry leading to high density chips by technology scaling, reduced voltage operation and higher frequencies make integrated circuits more susceptible to soft errors caused by energized particles, reducing their reliability [1]. Although soft errors do not cause physical damage on the chip, memory cell content may be altered, leading to incorrect processing of the stored value [2].

SRAM-based Field Programmable Gate Arrays (FPGAs) have the capability to implement any kind of digital circuit by loading a bitstream onto its configuration memory. The configuration memory, which totally controls the implemented circuit, is composed of static RAM cells. Therefore, SRAM-based FPGAs have high sensitivity to soft errors, especially single event upsets (SEUs), which may change the behavior of the originally implemented circuit [3]. This high sensitivity to SEUs limits the usage of SRAM-based FPGAs in safety or mission-critical applications, unless suitable hardening techniques are adopted.

Several studies by radiation [4] or by emulation of the effects of SEUs in the FPGA’s configuration memory as bit-flips in the memory content [5]-[8] have been carried out to investigate the sensitivity of SRAM-based FPGAs to ionizing radiation. Of all above mentioned platforms emulating the effects of SEUs, the ones proposed in [5], [6] and [7] are consecutive enhancements of one of the others doing mostly static analysis of the loaded design. In [8], the authors presented a platform called FLIPPER, which performs fault injection tests by applying workload to the loaded Design Under Test (DUT). Furthermore, FLIPPER has been validated by radiation tests showing that the cross section of the design implemented in the FPGA is quite well emulated by its fault-injection tests [9].

In the recent years, the number of FPGAs used in space applications is increasing, starting with the Mars Lander [10], Mars Rover [10] and Venus Express [11]. This trend will continue growing in a near future due to their re-programmability [12]. As communications are fundamental to space borne applications, such as satellites, spacecrafts, unmanned missions, digital filters play an important role in space systems [13]. Commonly used for equalization and noise or interference cancellation are adaptive filters as part of an adaptive Feed Forward Equalizer (FFE), whose effects of SEUs have been studied in [14] showing that their adaptive nature is well suited for circuit specific protection techniques [15][16]. The main goal of an equalizer is to compensate the effects introduced by the channel in the transmitted signal. One of those effects is the Inter Symbol Interference (ISI) which poses a major problem to high speed communication as the transmitted symbols are spread by the channel and interfere with nearby symbols [13], causing bit errors.

To guarantee fault tolerance of applications implemented in FPGAs, there is the need of system-level protection techniques. One option for system-level protection is Duplicating with Comparison (DwC), which is capable of detecting the fault event but cannot identify which of the duplicated components is the faulty one [17]. Another one is called Recomputing with Shifted Operands (RESO) that calculates the result of the functional unit twice, posing an intrinsic delay of more than double of the execution time [18]. The preferred choice to improve the reliability is Triple Modular Redundancy (TMR) or Xilinx’s TMR (XTMR) for Virtex FPGAs, since it does not require any architectural innovation and it is function-independent. The drawback is that the area and the power consumption are triplicated too [19][20].

This paper presents novel system-level protection techniques based on DwC for an adaptive FFE. The FLIPPER platform was used to evaluate the design cross section showing a cost reduction of up to 71% in comparison to XTMR. The paper is organized as follows. Section II gives an overview of the adaptive FFE, while Section III presents the XTMR and the proposed protection techniques. The test setup...
and procedure are illustrated in Section IV, while Section V presents the result analysis. Finally, conclusions are drawn in Section VI.

II. PRELIMINARIES

This section describes the adaptive Feed Forward Equalizer focusing on the slicer error used for adaption. This error will play an important role in identifying fault occurrences.

The Feed Forward Equalizer (FFE), illustrated in Fig. 1, is composed of three main blocks, a traditional Finite Impulse Response (FIR) filter with coefficients \( h[i] \), input signal \( x[n] \) and output signal \( d[n] \), an adaptation logic that updates the values of the coefficients \( h[i] \) periodically according to the error \( e[n] \) and a slicer unit estimating the received symbol \( y[n] \) from the filter output \( d[n] \). The error \( e[n] \) is calculated as a difference of \( t[n] \) and \( y[n] \) during training and the difference between \( d[n] \) and \( y[n] \) in normal operation.

![Block diagram of an adaptive FFE.](image)

Fig. 1: Block diagram of an adaptive FFE.

In this section, the slicer error and its function in the FFE have been explained, highlighting that the slicer error is a good indicator for anomalies like fault occurrences. This fact is used for the proposed techniques in the next section.

III. SYSTEM LEVEL PROTECTION TECHNIQUES

This section presents system-level protection techniques based on redundancy. The simplest and the most effective form of fault tolerance by redundancy is TMR. As it is very costly there is the need for other techniques. Therefore, this section also presents novel techniques which are based on the properties of the FFE for fault detection, localization and mitigation.

A. Xilinx Triple Modular Redundancy (XTMR)

The simplest technique is Triple Modular Redundancy (TMR). The FPGA vendor Xilinx provides a tool called Xilinx’s TMR (XTMR) which is applying TMR on a design for a specific Xilinx FPGA. This XTMR tool has become an industrial standard in the past few years for designs targeting Xilinx FPGAs and it is commonly used as a reference [19][20].

Concerning the XTMR implementation of the FFE, the following two options have been applied [21]. For “Controlling Component Triplication”, the option of “XTMR Types Standard” has been selected as it triplicates for a hierarchical design, all the inputs and all underlying primitives. All the inputs are assigned to unique input pins. The option “Triple-Voted” for “Controlling Output Triplication” has been applied to the dataout signal, as shown in Fig. 3.

![Schematic of the XTMR of the FFE.](image)

Fig. 3: Schematic of the XTMR of the FFE.

In this scheme, the minority voter inhibits, by acting on a tristate buffer, the propagation to the output of the signal that differs from the other two replicas.

As this fault tolerance technique is based on TMR, the resource consumption and power consumption is about three times of the unprotected version.

B. Proposed Techniques

The proposed Dual Modular Redundancy (DMR) methods are based on DwC but instead of the comparator they use enhanced voting logic that can detect failures, distinguish the faulty module and avoid failure propagation, by exploiting the observations on the slicer error discussed in the previous section.

In the normal steady state, the slicer error rarely surpasses the
threshold, which is chosen to be twice the error magnitude in the steady state, as shown in Fig. 4.a. Occasionally, the threshold is exceeded for a few cycles due to random bit errors caused by noise or other imperfections in the received signal. When a SEU hits the configuration memory of the FPGA, it might change the original design architecture or functionality, affecting the slicer error or even altering the dataout:

1. A minor change in the FFE circuitry can cause an increase in the slicer error, as shown in Fig. 4.b, but might not be sufficient to alter the dataout. For avoiding a bigger impact by accumulation of SEUs, the circuitry should be re-configured, but only in the cases when the slicer error exceeds the threshold for a certain number of cycles. In this way, those can be distinguished from a common bit error.

2. A major change in the FFE circuitry is causing an error that alters the output signal from its occurrence. Hence, the slicer error increases without any possibility of recovering the correct output values, as illustrated in Fig. 4.c.

![Fig. 4: Slicer error when a) slicer error in normal state, b) slicer error after a minor change, c) slicer error after a major change.](image)

These two cases have different symptoms and therefore, also need different approaches of detection and mitigation. With this understanding of the circuitry and with the help of additional logic, the faulty module can be localized and the propagation of the fault can be mitigated.

1) **Dual Modular Redundancy (DMR)**

As mentioned before, the proposed architecture consists of two FFE modules placed in parallel with a voting logic, resembling DMR. All input signals are duplicated outside of the FPGA and fed inside on different pins in order to avoid a single point of failure in case a SEU changes a configuration bit.

By using a reduced replica the overall area consumption is reduced and therefore also the probability of SEUs hitting the configuration bits used by the circuitry.

![Fig. 5: Design with two parallel sub-FFEs and a voter.](image)

Failures are detected by observing the slicer error as mentioned previously and can have the following effects:

- Mismatch of dataouts.
- One of the slicer errors exceeding the threshold.

At all times, the smart voters constantly observe the error signals and whenever one exceeds the threshold for a given number of cycles, a fail signal called scrub is activated, which is stored in the tri_reg register. The value of the tri_reg is fed back to both smart voters so that the faulty module is not considered anymore and the smart voters forward the correct dataout directly to the main output of the device. When the output of the two replicas differs, each smart voter compares the error values coming from each replica. The module with the greater error value is identified as the corrupted replica and the corresponding scrub signal is activated.

These scrub signals could be used to trigger partial re-configuration by an external scrubbing logic to recover the faulty module and resetting the tri_reg, while the system continues to operate with the other FFE module. The topic of scrubbing is out of scope of this paper and is mentioned for completeness.

2) **DMR with Reduced Replica (DMRwRR)**

This version is based on the same ideas as the proposed DMR version. However, instead of using the same replica for the duplication, it uses a reduced replica which works with less precision by using fewer bits in the coefficients and delay-line. This replica provides worse equalization performance but a significant area reduction.

The error_1 signal is truncated from 22 bits to 17 bits of the error_2 signal to enable the comparison for the smart voters. The rest of the structure is completely identical as the DMR version as illustrated in Fig. 6.

![Fig. 6: Schematic of the DMR with reduced replica.](image)

Fig. 6: Design with two parallel sub-FFEs and a voter.

By using a reduced replica the overall area consumption is reduced and therefore also the probability of SEUs hitting the configuration bits used by the circuitry.

## IV. TEST SETUP AND FAULT MODEL

This section presents the fault injection setup and the fault model used for retrieving the results discussed in the next section. The proposed techniques have been evaluated by a fault injection experiment. This has been accomplished by comparing SEU sensitivity of the following design versions: plain, XTMR, DMR and DMRwRR.

The fault injection experiment has been performed by using the FLIPPER platform [8]. The fault model adopted in
FLIPPER is the bit-flip of configuration memory cells [9]. For each of the four design variants, an injection campaign has been accomplished aimed at analyzing the design sensitivity to a single bit-flip occurring in random locations of the device configuration memory based on the assumption that between the occurrences of the SEUs, there is sufficient time for re-configuration of the faulty module. These campaigns consists of 300,000 fault injections runs each.

V. EXPERIMENTAL RESULTS AND ANALYSIS

All mentioned designs have been synthesized for the Xilinx XQR2V6000 FPGA [22] and their resource usage has been documented in Table I.

Table I

<table>
<thead>
<tr>
<th>Multipliers</th>
<th>Designs</th>
<th>LUTs</th>
<th>Slices</th>
<th>IOBs</th>
<th>FFs</th>
<th>MULTs18x18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DMR</td>
<td>1704</td>
<td>2250</td>
<td>1316</td>
<td>880</td>
<td>8200</td>
</tr>
<tr>
<td></td>
<td>DMRwRR</td>
<td>1704</td>
<td>2250</td>
<td>1316</td>
<td>880</td>
<td>8200</td>
</tr>
</tbody>
</table>

The table also includes the relative values comparing the system-level protection techniques to the plain version next to the absolute values of the resource usage with the percentage of overall chip usage.

As expected, Table I shows that the DMR version is a little bit more than two times bigger than the plain version. The DMRwRR is achieving a reduction and situates itself around 1.6 times of the plain version. The expected overhead for XTMR of three times is only kept for the IOBs, FFs and MULTs18x18 but when looking to the LUTs, the overhead increases to 4.20 and 5.27 for slices. When taking the slice usage as only measurement and the XTMR as the reference of comparison, the proposed techniques achieve for the DMR and the DMRwRR a reduction of 59% and 71%, respectively.

The fault injection results of 300,000 configuration memory bit-flips for each of the four designs are summarized in Table II, where the incidents are the cases when there is a difference between one of the slicer errors and the golden error vector and a failure occurs when the dataout differs from the golden.

Table II

<table>
<thead>
<tr>
<th>Design</th>
<th>Total Incidents</th>
<th>Total Failure Rate</th>
<th>Rel. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMR</td>
<td>282</td>
<td>0.006%</td>
<td>0.06%</td>
</tr>
<tr>
<td>XTMR</td>
<td>1669</td>
<td>0.000%</td>
<td>0.00%</td>
</tr>
<tr>
<td>DMRwRR</td>
<td>1291</td>
<td>0.033%</td>
<td>3.19%</td>
</tr>
</tbody>
</table>

The plain version has a failure rate of 0.094% where 282 out of 890 incidents caused a failure on the dataout. For the DMR and the DMRwRR, the total number of incidents increases with the factor discussed for the resources. For the XTMR version, the total number of incidents did not increase by the factor of the resource usage. This is due to the insertion of voters in each feedback path. The failure rate for the XTMR is 0.000% which is the expected value based on the test setup. For the DMR and DMRwRR versions, bit-flips affecting the smart voter logic are thought to be causing failures on the main dataout, resulting in failure rates of 0.004% and 0.003%, respectively. The DMRwRR achieves a higher failure rate reduction due to the fact that the reduced replica used for the DMR uses less complexity. This results into a smaller total area and also a smaller number of total incidents in comparison to the DMR version as Table II show. The last column shows the remaining failure rate when applying one of these system-level protection techniques to the plain version resulting 0.000% for XTMR and 4.26% and 3.19% for DMR and DMRwRR, respectively.

VI. CONCLUSIONS

This paper proposes novel system-level protection techniques for feed forward equalizers, which exploit application and system knowledge, resulting in a more intelligent protection with a 71% saving of circuit complexity in comparison to XTMR, as it has been shown in the paper. As future work, the fault model will be changed and all four designs tested for fault accumulation to estimate the mean time to failure for each design.

REFERENCES