1 Summary

Reliability issues become an important design concern with technology scaling. This paper introduces a novel methodology for balancing the desired fault masking and the consequence delay and power overheads due to redundancy. Experimental results shown that our solution outperforms similar approaches since it achieves average delay and power savings up to 25% and 35%, respectively, for comparable fault masking to existing commercially available solutions.

2 Introduction

For decades, computer architects have pursued one primary goal: performance. The even-faster transistors provided by Moore’s law were translated into remarkable gains in operation frequency and power consumption. Recently, however, the device-level size and architecture complexity impose several new challenges that have arisen, including a decrease in dependability level due to physical failures. Nowadays, fault tolerance becomes an important concern during system design not only for safety critical systems, but almost for the majority of architectures. One of the utmost important goals for system developers today is to design efficient and reliable architectures with unreliable components. Since existing approaches for reliability improvement are characterized as expensive in terms of delay, power consumption and area overhead, novel approaches should be proposed.

Even though fault tolerance is a well known technique, up to now it was mostly studied for ASIC (Application Specific Integrated Circuit) designs. However FPGAs (Field Programmable Gate Arrays) pose additional constraints (e.g., higher power density, lower operation frequencies, area overheads, etc), while the existing fault models are not necessarily applicable. Furthermore, since the majority of existing solutions are based on some form of redundancy, they are suitable only for safety/mission critical applications. To make things worse, faults at FPGAs can alter the design, not just user data. Moreover, FPGA designs utilize only a subset of the actually fabricated resources, and hence only a part of the occurred faults result to failures. Consequently, FPGA-specific mitigation techniques are required, that can provide a reasonable balance among the desired fault prevention and the consequence performance overheads due to the additional (replicated) hardware resources.

Until now, the only known commercial product that ensures fault masking in FPGAs is the Xilinx Triple Modular Redundancy (TMR) [1]. A TMR implementation includes three complete copies of the functional circuit and a voting mechanism. Even though TMR guarantees that most of the occurring faults may be prevented, it is very costly in terms of hardware resources and power consumption. Also, the performance degradation due to redundant modules usually violates the application’s timing constraints. Hence, TMR cannot be a candidate solution for general-purpose application mapping onto FPGAs.

In this paper we introduce a methodology for selectively protecting parts of applications mapped onto reconfigurable architectures. The proposed framework rather than applying redundancy to all the application’s functionalities (similar to commercially available Xilinx TMR tool [1]), it replicates only functionalities mapped onto regions with increased failure probability. Previous studies show that thermal stress is tightly firmed to reliability issues [6]. Hence, in this framework we protect resources with increased temperature values. The output of the proposed framework is a design that provides almost maximum fault masking with affordable penalties in terms of maximum operation frequency, power and area overheads.

Experimental results prove the efficiency of applying such a focused fault tolerance approach, as it achieves comparable fault masking to relevant solutions, but with significant lower mitigation cost. More specifically, our methodology leads to average gains in operation frequency and power consumption, as compared to a solution that corresponds to Xilinx TMR [1], up to 25% and 35%, respectively, for comparable fault masking. Furthermore, since our framework does not impose dedicated hardware blocks for supporting fault masking, it is applicable to commercial devices as an intermediate step between synthesis and P&R (placement and routing).

The contributions of this paper are summarized, as follows:

- Introduction of a methodology for fast yet accurate design space exploration that trades-off reliability improvement versus performance and power overheads.
- Redundancy is selectively inserted only to functionalities mapped onto resources with increased proba-
bility of failure.

- Introduction of a tool framework (available at [2]) that alleviates aging phenomena.

3 Motivation Example

This section describes the limitation of existing fault masking techniques to prevent failures occurred due to aging phenomena. For this purpose, we study a DSP application, named JPEG [3], whereas the underline platform is a Stratix-based FPGA. Note that since we target to general-purpose FPGAs, the device does not incorporate any mechanisms for fault tolerance.

The first task in order to locate regions of importance deals with the calculation of power consumption for all the FPGA’s hardware resources (both logic and interconnection). This info in conjunction to the floorplan/placement data, can be appropriately combined in order to perform a detailed thermal analysis. For this purpose, Hotspot v.5 tool [4] is employed.

Fig. 1 gives the thermal profiles regarding JPEG application under full (e.g. [1]) and proposed (Compact TMR) redundancy approaches. The granularity of this figure is defined in slice level, assuming worst-case input vectors, in order to have the maximum accuracy in the procedure of resources classification either as critical for failure, or not. Note that this map corresponds to an application mapping without fault tolerance (either in software or hardware level).

Different colors in this figure denote regions of the device that operate under different temperature values. As closer to red color a region is, the corresponding hardware resources (slices) operate under higher temperature values. Based on this thermal analysis report, we can conclude that on-chip temperature values for JPEG application range from 75.2°C up to 83.5°C, whereas for demonstration purposes, these temperatures are plotted in a normalized manner over the maximum temperature, as follows:

\[
\text{plot value}(x, y) = \frac{\text{temperature at slice}(x, y)}{\text{maximum temperature}}
\]  

A number of conclusions might be derived from these thermal maps. Among others, temperature varies a lot over the FPGA device, which in turn results to regions with different failure probabilities, and hence different requirements for fault masking. Note that different applications exhibit different regions of importance, while even the same application might result to different thermal maps (e.g. under power- and timing-aware P&R).

Another important conclusion is drawn from Fig. 1: although the majority of existing fault tolerant techniques are applied uniformly over the device since they focus solely to SEUs (Single Event Upsets), the actually critical for failure resources (due to reliability issues) provide a non-homogeneous and irregular picture. Consequently, careful analysis of the points of failure must be performed, while applications need to incorporate full-custom TMR solutions.

As an alternative to this uniform (full) insertion of TMR, Fig. 1(b) plots the thermal profile when only functionalities mapped onto slices with increased temperature values, and consequently increased failure probabilities, are triplicate. As a criterion for this triplication we assume (without affecting the generality of proposed solution) that slice has to operate under temperature value higher than half of the maximum on-chip temperature \((\text{plot}(x, y) \geq 0.5)\). Regarding the JPEG application, such a criterion results to insert TMR only to 40% of the utilized slices. Note that for Fig. 1, the maps were retrieved assuming the same P&R algorithm and target FPGA device.

Even though one would expect that power consumption with full TMR is about \(3 \times \) that of the unprotected circuit, this does not occur, since logic blocks at FPGAs dissipate about 10%-15% of the total power consumption, whereas routing infrastructure dissipate about 65%-75% [5]. This conclusion is also depicted in Fig. 1, where thermal stress does not increase linearly with the correspondence increment of application’s functionalities that are triplicate.

In other words, even if TMR is applied uniformly over the entire architecture (Fig. 1(a)), this results almost to a negligible additional thermal stress, as compared to the application mapping when only 40% of the application’s functionalities are triplicate (Fig. 1(b)).

Hence, we can summarize that full TMR does not speedup considerable aging phenomena. However, such an excessive insertion of redundancy introduces significant penalties in terms of power, delay and area overheads, which usually are not affordable for general-purpose applications.
4 Proposed Methodology

This section describes the proposed methodology (shown in Fig. 2) for inserting as much redundancy as possible, with respect to the maximum affordable performance degradation. Since we aim to improve fault masking due to aging phenomena, rather than inserting redundancy uniformly over the entire FPGA (as similar solutions do for protecting designs against to SEUs), our methodology incorporates a more aggressive approach. More specifically, only functionalities mapped onto slices with increased failure probability, are triplicate.

Initially, application’s netlist is annotated with slices with increased failure probability, resulting to the maximum possible level of fault coverage. Then, application is P&R with a timing-aware algorithm [9] in order to alleviate the delay increment due to redundant functionalities posed by redundancy. Based on P&R output, we can evaluate the sensitivity of hardware resources in errors. For this purpose, a number of upsets that follow Time-Dependent Dielectric Breakdown (TDDB) model [6] are injecting into the design. The number of injected upsets ranges from 1% up to 10% of bitstream file. By measuring the percentage of masked upsets over those reported that result to failures, it is possible to compute the efficiency in fault masking.

Having such a detailed analysis about hardware sensitivity in failures, in conjunction to thermal maps, we can claim (with good fidelity) that not all of the application’s functionalities need to be triplicate. Hence, it is possible to eliminate redundancy (through power and gate clock) from those functionalities mapped onto slices that belong to non-critical for failure regions. For this purpose, we have developed a new tool that gradually removes redundancy from netlist. The goal of this tool is to allow as maximum as possible focused redundancy at the design (applied only at application’s functionalities mapped onto slices with increased failure probability), in respect to maximum affordable power and timing overheads, as they derived from specifications.

Finally, application is re-placed and re-routed under temperature constrains. During this task, we are primarily interested to alleviate thermal stress for the already derived application mapping. For this purpose, we incorporate our previous temperature-aware P&R algorithm [8] with the already derived (with timing-aware) P&R.

In order to provide the necessary software support, we have developed a new open source tool flow, named Fault-Free [2]. Note that our proposed methodology is transparent to the employed CAD tool chain, and hence, it can be applied as an intermediate step between synthesis and P&R tools at any other existing design flow.

5 Experimental Results

This section provides a number of qualitative results that prove the efficiency of our proposed methodology. As a reference point we use two application implementations: (i) with uniform redundancy (mentioned as “Full TMR”) [1], and (ii) without TMR (marked as “No TMR”). The solutions derived from our methodology, referred as “Compact TMR”, corresponds to applications where only critical for failure functionalities (those that are mapped onto slices with temperature values \( T_{\text{slice}}(x, y) \geq T_{\text{max}} \)), are replicated.

Evaluation was performed with a number of industrial oriented kernels from [3], ranging from DSPs up to multimedia and processors. The target platform is a Stratix-based device which does not contain any fault prevention mechanism, while upsets are injected with a distribution that follows TDDB model [6].

Fig. 3 evaluates the efficiency of our compact TMR solution to mask faults, regarding JPEG application. Two different algorithms, a timing-aware [9] and a temperature-aware [8], were employed during this study. Horizontal axis at Fig. 3 corresponds to the percentage of masked over the injected faults, whereas the vertical axes give (in normalized manner) the variation in terms of maximum operation frequency and power consumption.

![Figure 2: Proposed methodology for selective TMR.](image)

![Figure 3: Fault masking under different redundancy.](image)
A number of conclusions might be derived based on Fig. 3. Among others, existing way for applying TMR [1] leads to considerable overheads in terms both of power consumption and maximum operation frequency. More specifically, the power overhead ranges up to 25%, whereas the delay degradation is up to 35%, as compared to initial (without TMR) application mapping. These penalties usually are not affordable since they violate system’s specifications. On the other hand, our proposed methodology allows to trade-off the desired reliability improvement and the consequence maximum performance degradation. For instance, assuming that a system can afford up to 5% degradation in maximum operation frequency. In case only the timing-aware P&R is employed (if we bypass the last step in our methodology), then the corresponding solution leads about to 68% fault masking (the solution is marked with a circle in Fig. 3). Note that this percentage of performance degradation corresponds to 24% fault masking regarding the temperature-aware P&R. Finally, if we try to achieve this percentage of fault masking with a temperature-aware P&R, then there is an addition penalty about 8% in maximum operation frequency.

Similar to performance exploration, our framework also enables the quantification of power consumption for different application implementations. More specifically, the fault masking that corresponds to timing-aware solution (depicted in circle) leads to 6% overhead in power consumption, as compared to the corresponding value about the temperature-aware P&R for comparable fault masking. Note that these gains can be further improved if we select a solution that corresponds to higher operation frequency (closely to left vertical axis). Another conclusion can be derived from Fig. 3. More specifically, rather than designers employ a pre-defined TMR (or partial TMR) solution, it is possible to study all the potential trade-offs between reliability improvement and performance degradation, and finally to select the solution that better meets application’s requirements.

### Table 1: Evaluation in terms of Energy×Delay Product (nsec×mW).

<table>
<thead>
<tr>
<th>Application</th>
<th>Temperature-aware</th>
<th>Timing-aware</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO TMR</td>
<td>Compact</td>
</tr>
<tr>
<td>mux64_16bit</td>
<td>3.76</td>
<td>4.46</td>
</tr>
<tr>
<td>tip_risc</td>
<td>4.25</td>
<td>6.42</td>
</tr>
<tr>
<td>sge_core</td>
<td>23.5</td>
<td>24.3</td>
</tr>
<tr>
<td>gsm_dsp</td>
<td>98.7</td>
<td>125</td>
</tr>
<tr>
<td>huffman_video</td>
<td>3.43</td>
<td>5.51</td>
</tr>
<tr>
<td>oc_mem_ctrl</td>
<td>19.9</td>
<td>25.1</td>
</tr>
<tr>
<td>oc_sbus</td>
<td>26.5</td>
<td>29.0</td>
</tr>
<tr>
<td>oc_pb dma</td>
<td>16.0</td>
<td>19.6</td>
</tr>
<tr>
<td>oc_correlator</td>
<td>3.80</td>
<td>4.82</td>
</tr>
<tr>
<td>tm_receiver</td>
<td>4.17</td>
<td>4.30</td>
</tr>
<tr>
<td>Average:</td>
<td>23.1</td>
<td>24.0</td>
</tr>
<tr>
<td>Ratio:</td>
<td>0.43</td>
<td>0.61</td>
</tr>
</tbody>
</table>

### 6 Conclusion

A new framework for supporting efficient application mapping under reliability constraint was introduced. The proposed solution provides a trade-off between the desired fault masking and the consequence delay and power overheads due to redundancy. By extending a well established technique for SEUs masking, our methodology leads to fault prevention against aging phenomena. Experimental results shown that our solution outperforms similar approaches since it achieves average delay and power savings up to 25% and 35%, respectively, for comparable fault masking to existing commercially available solutions.

### References


