Adaptive Resilience Approaches for FPGA Fabrics

Ronald F. DeMara
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816-2362
E-mail: demara@mail.ucf.edu

November 20, 2017

Abstract — This Work-In-Progress Article overviews the broad categories of autonomous self-repair approaches to Evolvable Hardware (EHW) through the years of myself, valued collaborators, and others. The paper concentrates on a creating a taxonomy of existing techniques. Techniques addressed concentrate on EHW using SRAM-based FPGAs. These range from Competitive Runtime Reconfiguration (CRR) relying on Genetic Algorithm based approaches to design-time only methods such as design disjunction which rely on design-time diversity for resilience at run-time. A common attribute is the creation of an initial population of functionally identical (same input-output behavior), yet physically distinct (alternative design or place-and-route realization) FPGA configurations is produced at design time. At run-time, these individuals compete for selection based on a fitness function favoring fault-free behavior. Hence, any physical resource exhibiting an operationally-significant fault decreases the fitness of those configurations which use it. Through runtime competition, the presence of the fault becomes occluded from the visibility of subsequent FPGA operations. The use of adaptive hardware approaches across multiple phases of the fault handling process including Detection, Isolation, Diagnosis, and Recovery are discussed in this survey.

Keywords — Field Programmable Gate Arrays (FPGAs), Fault-handling, Survivable Architectures, Dynamic Partial Reconfiguration, Reliability, Availability.

1.0 Introduction

Fault tolerance, high reliability, and availability are major desired characteristics of a mission critical system. Harsh operating environments, manufacturing defects, and component aging are contributing causes of hardware faults that make realizing these characteristics difficult. Many hardware reliability approaches have been proposed in the literature such as fault avoidance, design margin, modular redundancy, and fault refurbishment. Fault avoidance based design approaches aim to avoid possible faults that could occur at run time. Such approaches impose minimal size, weight, and power overheads. Meanwhile, design margin approaches rely on an increased number of redundant system components and capabilities to enhance reliability by designing with a margin for fault tolerance. A vast category of alternatives to these methods, based on active approaches to compensate for fault impacts, is overviewed herein.

Evolutionary mechanisms can actively restore mission-critical functionality in SRAM-based reprogrammable devices [1-8]. They provide an alternative to device redundancy for dealing with permanent degradation due to radiation-induced stuck-at-faults, thermal fatigue, oxide breakdown,
electromigration, and other local permanent damage [9-10]. Potential benefits include recovery without the increased weight and size normally associated with spares. Also, failures need not be precisely diagnosed due to intrinsic evaluation of the FPGA’s residual functionality through assessment of the Genetic Algorithm (GA) fitness function.

Autonomous repair of FPGAs is of particular interest in aerospace applications for both in-flight and Ground Support Equipment devices. Some deep space satellites utilize over 100 FPGA devices [11], and more recently, space-qualified versions of high-density SRAM-based FPGAs have become commercially-available [12]. Meanwhile NASA terrestrial applications routinely employ FPGAs for tasks ranging from launch control to signal processing. SRAM-based FPGAs are of significant importance due to their high density and increasing use in mission-critical/safety-impacting applications. Meanwhile, they offer unlimited reprogrammability that can enable autonomous repair.

For in-flight applications, FPGA devices encounter harsh environments of mechanical/acoustical stress during launch, high doses of ionizing radiation, and thermal stress. Simultaneously, they are required to operate reliably for long mission durations with limited or absent capabilities for diagnosis/replacement and little onboard capacity for spares. Hence, research has focused on employing the reconfigurability inherent in field programmable devices to increase reliability and autonomy [12]-[15].

### 2.0 Taxonomy of Autonomous Fault-Handling Approaches

Over the last few years, *adaptive regeneration* has been investigated as an alternative to pre-determined spares. As listed in Table I, Vigander’s [13] regeneration approach extends Triple Modular Redundancy (TMR) to utilize faulty FPGAs that have been partially regenerated using evolutionary algorithms. He demonstrates that FPGA-based implementations of 4-bit x 4-bit multipliers can be automatically reconfigured to realize partial refurbishment. Since each partially refurbished multiplier is deficient with respect to only selected input pairs, a voting arrangement of partially refurbished devices exhibits complete functionality. TMR, Vigander’s, and other q-plex spatial voting approaches deliver real-time fault resolution, but increase power consumption q-fold during fault-free operation.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Fault Handling Method</th>
<th>Fault Detection</th>
<th>Resource Coverage</th>
<th>Fault Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR</td>
<td>Spatial voting</td>
<td>Negligible</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Vigander [13]</td>
<td>Spatial voting &amp; offline evolutionary regeneration</td>
<td>Negligible</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lohn, Larchev,</td>
<td>Offline evolutionary regeneration</td>
<td>Negligible</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DeMara</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Lach98</td>
<td>Static-capability tile reconfiguration</td>
<td>Relies on independent fault detection mechanism</td>
<td></td>
<td>Voting element</td>
</tr>
<tr>
<td>STARS [Abramovic01]</td>
<td>Online BIST</td>
<td>Up to 8.5M outputs on ORCA FPGA</td>
<td>Test pattern transients</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LUT function</td>
</tr>
<tr>
<td>[Keymeulen,</td>
<td>Population-based fault insensitive design</td>
<td>Design-time prevention emphasis</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Stoica,</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Zebulum00]</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>CRR</td>
<td>Competing configurations with temporal voting and</td>
<td>Negligible</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>online regeneration</td>
<td>Transients are attenuated</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>automatically</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Table I: Characteristics of Related FPGA Fault-Handling Schemes.
Lohn, Larchev, and DeMara [16] develop a FPGA bit-string representation along with mutation and two-point crossover operators for actively refurbishing interconnect as well as logic resources. Their approach regenerated a Quadrature Decoder circuit on a Xilinx SRAM-based Virtex XCV1000 FPGA. It shows that a stuck-at-fault on the input to a Configurable Logic Block (CLB) can be resolved through GA-based repair. The GA synthesizes a new alternative configuration within a population of 40 competing configurations after a few hundred generations. Their GA was shown to be capable of recycling faulty components as partially-damaged CLBs were sometimes reassigned to alternative functions based on their residual functionality. While Lohn, Larchev, and DeMara’s approach demonstrated complete regeneration for a modestly-sized sequential circuit, the refurbishment was performed offline and required exhaustive fitness test vectors.

Lach’s deterministic approach [17] segments the FPGA into static tiles at design time with a known functionality, some redundant resources, and a pre-designed alternate configuration. Spare tiles can be selected when needed, but their functionality is predetermined and thus limited. On the other hand, STARS [14] is a resource-oriented diagnostic test approach that performs Built-in Self-Tests (BISTs) on roving sub-sections of the FPGA. Portions are continually taken offline in succession and tested while the functionality is moved to a new location within the reprogrammable fabric. Detection latency can be large since tests must sweep through all intervening resources before a fault is detected. Potential throughput unavailability due to diagnostic reconfigurations when no faults have yet occurred is also consideration. Nonetheless, it provides a useful approach for deterministic exhaustive online constructive repair.

Figure 1 introduces a new taxonomy to categorize approaches to Design Approaches for Evolvable Hardware for Fault-Handling. This taxonomy is based on a few orthogonal dimensions. The first level of the time of synthesis of the configuration used to facilitate fault-handling. The categories at this level are configuration synthesis at 1) design-time, 2) runtime, and 3) over the operational lifetime. The taxonomy also identifies multiple subcases within each approach.

In each case, the fabric of the underlying hardware resources are organized to advance reliability objectives. In the first approach, the emphasis is on forming one or more resilient designs, each of which is represented as an FPGA bitstream. In the latter two approaches, the post-design-time reconfiguration ability allows dynamic response to faults that occur with the fielded system such that new bit-streams becomes evolved to the conditions at-hand using intrinsic evolution and/or evaluation of fabric resources.
3.0 Design-Time Configuration Synthesis

3.1 Diversity Formation at Design-Time

In the case of design-time configuration synthesis, diversity formation approaches target a-priori evolvable hardware methods to impart fault-resilience into a baseline design that lacks sufficient resilience. For instance, a foundational approach was taken by Keymeulen, Stoica, and Zebulem [12] using a design-time emphasis. They develop evolutionary techniques so that a circuit is initially designed to remain functional even in presence of various faults. Their population-based fault tolerant design method evolves diverse circuits and then selects the most fault-insensitive individual. This method enables passive runtime operation. It is also applicable for constructing a diverse initial population for which design diversity can be measured and quantified [16]. Once quantified, diversity can be expressed and optimized by grouping FPGA configurations into alternate sets which are maximally diverse [17]. Due to the lack of ability to adapt after design-time, this survey concentrates on techniques which extend these approaches with more dynamic methods.

3.2 Non-Adaptive Group Test

Non-adaptive group testing is another design-time bit-stream synthesis technique to realize a novel low overhead fault localization/fault isolation capability along with rapid fault recovery from temporary and permanent faults in reconfigurable fabrics while incurring minimal area, power, and perturbation to normal system throughput [18]. This approach focuses on providing fault resilience against extensive fault scenarios by reusing a subset of the configurations to ensure continual execution with minimal recovery time [19][20]. Design disjunction can offer a broad coverage, high resolution, and low overhead approach to online diagnosis and recovery of reconfigurable fabrics [21]. Design disjunction leverages the condensed diagnosability of $T$ logic resources to achieve self-recovery using partial reconfiguration in $O(\log T)$ steps of comparing outputs of alternate configurations produced through design-time synthesis [22].

3.3 Functional Assembly

Genetic Algorithm (GA) based techniques are known to optimize multiple objective and automate the process of digital circuit design. They can be used achieve the synthesis of digital circuits for LUT-based FPGA architectures. Since this is performed at design-time for the entire circuit, various optimizations have been considered [23]-[25]. For example, parallel modes of the GA such as Master-Slave and the Island model are compared to see which scheme results in better speedup and quicker convergence for effectively utilization of current multicore hardware. Speedup of about 5-fold over the sequential single-threaded implementation are achieved with both the schemes on six-core machine. Convergence is also found in fewer number of generations. The methods described here-in can be employed in evolvable hardware systems as well as FPGA CAD tools.

A time and area sparing approach called ECHELON or Embedded-Cascaded Hierarchically-Evolved Logic Output Network computes special mathematical functions by cascading incremental functionality of reconfigurable analog and digital blocks [26]. ECHELON relies on evolutionary algorithms to search the design space based on the intrinsic switching behavior of the target circuit as depicted in Figure 2. This method extends the work done in [27] to exploit the interaction between general purpose analog and digital reconfigurable fabrics under the control of an embedded ARM core. The analog computation engine evolves analog reconfigurable fabric composed of switched capacitor op-amp blocks which may be configured to realize various amplifier, mixer and modulator topologies. This circuit produces analog signals roughly approximating the mathematical function desired and performs intelligent operations with self-scaling genetic algorithm (SSGA) to compress outputs beyond the device ADC range. It realizes a more sophisticated elaboration of hybrid analog and digital computation whereby an unrefined analog GA
evolves a coarse solution in a narrow voltage range. Next, the output is scaled to allow a more computationally-tractable range. The entire set of operations performed up-to this stage is referred to here as analog pre-processing. This output is then converted to digital signals which are then adaptively refined by PLD-based digital fabric and combined in cascaded stages with digitally-evolved weights, using techniques for spatial self-adaptation of digital fabrics.

4.0 Runtime Configuration Synthesis

4.1 Dynamic Modular Redundancy Methods

Starting with conventional TMR, various dynamic extensions have been developed [28]-[30]. Each actively manages some aspect of modular redundancy, ranging from varying from Dual Modular Redundancy (DMR) up through TMR, as well as maintaining or even evolving spare pools for failed voting elements. For instance, a sustainable modular adaptive redundancy technique (SMART) composed of a two-layered organic system. The hardware layer is implemented on a Xilinx Virtex-4 Field Programmable Gate Array (FPGA) to provide self-repair using a novel approach called reconfigurable adaptive redundancy system (RARS). The software layer supervises the throughput computations on the FPGA and extends the self-healing capabilities through application-independent, intrinsic, and evolutionary repair techniques that leverage the benefits of dynamic partial reconfiguration (PR). SMART was evaluated using a Sobel edge-detection application and was shown to tolerate stressful sequences of injected transient and permanent faults while reducing dynamic power consumption by 30% compared to conventional triple modular redundancy (TMR) techniques, with nominal impact on the fault-tolerance capabilities. Moreover, PR is employed to keep the system on line while under repair and also to reduce repair time. Experiments have shown a 27.48% decrease in repair time when PR is employed compared to the full bitstream configuration case [28].

4.2 Genetic Algorithm Driven Adaptation Approaches

Despite the advantages of above-mentioned approaches, anticipating all the possible faults before the system is operational is difficult. Specifically, modular redundancy approaches utilize multiple identical modules each of which is capable of delivering the desired functionality. These increase size, weight, and power consumption. Additionally, the recovery capacity of these approaches is limited to the number and granularity of the available redundant modules. Fault refurbishment approaches, such as the proposed approach offer a very competitive option because of the high recovery capacity and adaptability to unforeseen faults. However, fault refurbishment is challenging due to the complexity involved in generating configurations for implementing fault-free digital circuits on reconfigurable devices. Genetic Algorithms (GAs) [2] are guided trial-and-error search techniques that use the principles of Darwinian evolution which target the survival of the fittest by casting a net over the entire solution space to find high fitness regions. The reprogrammability of Field Programmable Gate Arrays (FPGAs) provides an efficient platform highly suitable for evolutionary fault refurbishment experiments [3]. In the event of faults in FPGAs, a GA can be used to search and implement alternate configurations that circumvent the faulty resource, thus providing device refurbishment. This paper introduces the concept of improving the performance of GAs by generating and utilizing information regarding the location of faulty resources on FPGAs.

Genetic Algorithm based FPGA fault-handling approaches aim at actively recouping the residual functionality that remains when a fault-induced error occurs [1-8]. These approaches typically focus first on hard-fault coverage, yet due to bitstream refresh may cover a number of soft errors also with reconfigurable logic LUT and interconnect storage. As mentioned, these provide an alternative to device redundancy for dealing with permanent degradation due to radiation-induced stuck-at-faults, thermal fatigue, oxide breakdown, electromigration, and other local permanent damage [9-10]. Potential benefits include recovery without the increased weight and size normally associated with spares. Also, failures need not be precisely diagnosed due to intrinsic evaluation of the FPGA’s residual functionality through
assessment of the GA fitness function. Going further, additional optimizations of the GA processing itself are developed in [31]-[36].

For instance, autonomous repair and refurbishment of reprogrammable logic devices using Genetic Algorithms can improve the fault tolerance of remote mission-critical systems. The goal of increasing availability by minimizing the repair time is addressed in this paper using a CGT-pruned Genetic Algorithm [33]. This method utilizes resource performance information obtained using Combinatorial Group Testing (CGT) techniques to evolve refurbished configurations in fewer generations than conventional genetic algorithms. A 3-bit x 2-bit Multiplier circuit was evolved using both conventional and CGT-pruned genetic algorithms. Results show that the new approach yields completely refurbished configurations 37.6% faster than conventional genetic algorithms. In addition it is demonstrated that for the same circuit, refurbishment of partially-functional configurations is a more tractable problem than designing the configurations when using genetic algorithms as results show the former to take 80% fewer generations.

4.3 Oblivious Reconfiguration Strategies

While the aforementioned approaches utilize a pre-conceived externally-defined construct for fitness of alternative bitstreams to deliver resiliency, alternatives have been pioneered whereby such determinations arise as emergent behavior among a population of alternatives [16][37]-[47]. This has numerous advantages, including automatically ranking fitness on a smooth continuum of functionality, as opposed to a binary feedback of fully-operational or entire-faulty condition. An archetype of such a novel autonomous fault handling approach to these problems is Competitive Runtime Reconfiguration (CRR) [16][47]. Some distinguishing features of CRR are:

1. integration of an adaptive computing approach throughout the device lifecycle,
2. device refurbishment without additional function or resource test vectors, and
3. a novel fitness assessment approach via pairwise discrepancy detection without a pre-conceived oracle for correctness.

Fault handling lifecycle support is realized using a pairwise comparison to detect faults against a diverse population of competing configuration alternatives. The genetic operator of crossover isolates the failed physical resource, and alone or with mutation, realizes a failure-specific repair during normal operations to make detailed physical failure mode diagnosis unnecessary. CRR integrates competition and evolution wholly within the FPGA’s normal data throughput processing flow to eliminate the need for additional test vectors. Because a fitness function is used that favors fault-free behavior, the FPGA’s normal input data stream can be used to evaluate fitness. This also ranks competing alternatives with regards to their relative performance to provide graceful degradation even in the presence of multiple faults.

CRR utilizes a temporal voting approach whereby the outputs of just two competing instances are compared at any instant and alternative pairings are considered over time. The presence or absence of a discrepancy is used to adjust the fitness of both individuals without rendering any judgment at that instant on which individual is actually faulty. The faulty, or later exonerated, configuration is determined over time through other pairings of competing configurations. The competitive process is applied repeatedly to form a strong consensus across a diverse pool of alternatives. Under CRR, the FPGA’s outputs are compared before they leave the reconfigurable device so fault detection occurs on the first erroneous output and detection latency is negligible. Fault isolation in the TMR, Vigander, and Lach approaches is restricted to coarse predefined functional granularities. Meanwhile, STARS attempts to isolate faults at only the very finest resource granularity. Alternatively, CRR does not impose a predetermined fault isolation granularity in order to achieve refurbishment.

5.0 Lifetime Configuration Synthesis
Lifetime-oriented strategies go a step further than recovering from a specific fault at runtime. Throughout the system mission lifetime, they strive to optimize an overall system performance health metric, such as viable throughput, commonly referred to as “goodput.” Thus, the focus in these techniques is beyond reliability and into the realm of sustainability, even at degraded performance, by evolving new configurations of fabric resources which can best sustain the mission. The scope is broad support to support mission objective through active reconfiguration, using the subcategories of methods identified below.

5.1 Adaptive Group Test

In FPGA devices, the available number of unit cells such as Look Up Tables (LUTs) can comprise many thousands of physical resources. In this paper, the difficult problem of rapidly identifying a failure among these resources is addressed by extending methods from the algorithmic work on Combinatorial Group Testing (CGT) [48]. These concepts are used to develop new adaptive methods that utilize only the FPGA’s usual runtime inputs to identify and isolate faults. This maximizes the device’s online availability even while fault isolation is in progress. The emphasis of Adaptive Group Test techniques is on fast and reliable fast isolation online by creating new bitstreams on-demand that successively narrow down the suspect pool of N potentially fault resources in \(O(\log N)\) testing steps [49]-[52].

For instance, the Competitive Dueling approach is introduced in [50]. Therein, a runtime fault detection without using special test vectors is achieved in a Concurrent Error Detection (CED) strategy by comparing the outputs of two identical functional circuits for discrepancies. The discrepancy detector provides information regarding whether or not the outputs of two competing configurations resident on the FPGA produce outputs which are in bitwise agreement with each other. Using only this information, an adaptive method for reconfiguring the FPGA's functional logic can enable fault isolation because alternative CED configurations use varying subsets of resources. When these instances of the functional elements are paired over time, the accumulated correctness behavior along with their resource utilization characteristics are used to isolate the physical resource fault.

5.2 Observer-Actuator Strata

An Organic Embedded System (OES) architecture is developed for sustainable performance using SRAM-based Field Programmable Gate Arrays (FPGAs), an Organic Computing (OC) observer/controller organization, and regeneration with Genetic Operators [8]. Innovations include availability during online regeneration, aging and outlier driven repair assessment, and a uniform design for Autonomic Elements (AEs) despite the fact that they monitor different types of Functional Elements (FEs). Using logic circuits from the MCNC-91 benchmark set, availability during repair phases averaged 75.05%, 82.21%, and 65.21% for the z4ml, cm85a, and cm138a circuits respectively under stated conditions. In addition to simulation, the proposed OES architecture synthesized from HDL was prototyped on Xilinx Virtex II Pro FPGA device supporting partial reconfiguration to demonstrate the feasibility of OC approaches for intrinsic regeneration of the selected circuit.

5.3 Encoding-Driven Adaptation Strategies

Encoding-driven Adaptation Strategies explicitly leverage properties of the data representation to further guide and optimize resilience handling [53]-[55]. This can guide the reconfiguration of FPGA fabric elements, far beyond an error encoding strategy alone. For instance, in [53] a Heterogeneous Concurrent Error Detection (hCED) scheme is introduced which is based on output anticipation to guide the efficient runtime reconfiguration of the fault-handling process. It presents an alternative to conventional Concurrent Error Detection (CED) techniques which rely on two exact replicas of a given module to provide redundancy in fault-tolerant systems. A discrepancy in one of the two instances flags at least one of them as faulty. It developed a heterogeneous redundant FPGA-based system by exploiting the application properties. Consequently, the replicated module is not necessarily an exact copy of the original module but is much less resource and power hungry. In the paper, we discuss two forms of the heterogeneous structure which are spatial and temporal redundancy based. These forms are evaluated using FPGA based hardware
implementation of the Discrete Cosine Transform (DCT) block. A necessary condition is derived to declare the DCT block as fault-free. The results show that the heterogeneous spatial redundancy can realize a resource efficient CED pair at the cost of a small latency in error detection. On the other hand, the heterogeneous temporal redundancy can provide permanent faults resource coverage at the cost of reduced throughput with negligible resource overhead.

6.0 Energy versus Resilience Tradeoffs

A capstone topic in autonomous reconfiguration strategies for reconfigurable fabrics is are active management of resilience versus energy consumption [56]-[61]. For instance, the Quality and Resilience of an Embedded Video Encoder is evaluated against a Continuum of Energy Consumption in [59]. This approach utilized an adaptive redundancy-based fault-handling approach exploiting the partial dynamic reconfiguration capability of SRAM-based FPGAs. Fault detection in Signal Processing Systems is accomplished using a simplex hardware arrangement while a deterministic fault isolation scheme is employed, which neither requires test vectors nor suspends the computational throughput. The approach is validated by implementation of Discrete Cosine Transform (DCT) and Motion Estimation (ME) blocks for a H.263 video encoder benchmark in Xilinx Virtex-4 FPGA. Such techniques can be viewed as multi-objective optimization search problems by the underlying evolvable hardware algorithm.

7.0 Device Infrastructures

Finally, all of the aforementioned techniques rely on hardware and middleware layers for feasible operation. A number of advances in such infrastructures for various FPGA devices continue to be developed and refined [62]-[71]. For instance, in [68] an integrated platform for fast genetic operators is presented to support intrinsic evolution on Xilinx Virtex II Pro Field Programmable Gate Arrays (FPGAs). Dynamic bitstream compilation is achieved by directly manipulating the bitstream using a layered design. Experimental results on a case study have shown that a full design as well as a full repair is achievable using this platform with an average time of 0.4 microseconds to perform the genetic mutation, 0.7 microseconds to perform the genetic crossover, and 5.6 milliseconds for one input pattern intrinsic evaluation. This represents a performance advantage of three orders of magnitude over JBITS software-based FPGA infrastructure and more than seven orders of magnitude over the Xilinx design tool driven flow for realizing intrinsic genetic operators on a Virtex II Pro device. These developments typically rely on a runtime resource management system for fabric resources [62] and may extend to FPGA-specific features such as defragmentation [70][71] beneficial for practical application of the resilience techniques discussed herein.

8.0 Conclusion

Faults in FPGA fabrics, configuration-storing memory, and error detection circuitry can be addressed through autonomous reconfiguration methods. These can be detected and ultimately handled from simple TMR-style voting up through emergent behavior processes whereby faulty competing functional outputs will indicate discrepancy. These can also cover the configurations loaded with incorrect bit streams from configuration storing memory, and whenever faulty XNOR comparators disagree. If any of these faults happens to be a transient, the bitstream reconfiguration process will scrub the SEU(s) automatically. In the most robust of these strategies, the fitness will be decreased initially due to transient faults, but later rise so that transients are resolved instantly and attenuated automatically over time.

Nonetheless, some failure exposures inevitably remain. If the reconfigurability of the FPGA is impaired then any approach based on reconfiguration would by definition become infeasible. Likewise a catastrophic failure impacting a vast majority of the half-configurations could make correct consensus impossible. Vulnerabilities in the CRR control process, such as the fitness of individuals can be mitigated by maintaining a local fitness table within each half-configuration; other control functions in the GA can
be made to be competitive using alternate physical resources as well. We are currently studying these extensions, and also broadening the CRR paradigm and its application to other self-healing systems.

Using adaptive competition, the vast majority of resources in the FPGA can be covered. Faults are detected, isolated, and resolved without additional test vectors while allowing the device to remain partially online. Furthermore, this approach has the potential to provide these benefits with no increase in chip count in the fielded system if the approach is embedded within the FPGA fabric.

References


