Power and Quality-Aware Image Processing Soft-Resilience using Online Multi-Objective GAs

Naveed Imran

Department of Electrical Engineering and Computer Science,
University of Central Florida,
Orlando, FL 32816-2362, United States
E-mail: naveed@knights.ucf.edu

Rizwan A. Ashraf

Department of Electrical Engineering and Computer Science,
University of Central Florida,
Orlando, FL 32816-2362, United States
E-mail: rizwan.ashraf@knights.ucf.edu

Ronald F. DeMara

Department of Electrical Engineering and Computer Science,
University of Central Florida,
Orlando, FL 32816-2362, United States
E-mail: demara@mail.ucf.edu

Abstract: A self-aware signal processing architecture is proposed based on adaptive resource escalation which is guided by a multi-objective Genetic Algorithm (GA). The GA prioritizes tasks within a reconfigurable hardware fabric to maintain the quality-of-service and power consumption objectives. Attainment of these objectives is subject to the intrinsic reliability and performance of the computational elements in the resource pool. A health metric at the application layer, such as Peak-Signal-to-Noise Ratio (PSNR) measurement in a Discrete Cosine Transform (DCT) or Measure of Confidence in a Support Vector Machine (SVM) classifier, is used to assess throughput performance. When performance decreases beyond acceptable tolerances, the primary objective is to maximally recover output quality. The secondary objective is to minimize power consumption which also depends upon the input signal characteristics, in addition to the utilized computational resources. An adaptive guidance function for GA-driven recovery is proposed and validated for these objectives. It retains healthy processing elements in the throughput datapath to gracefully-degrade throughput by optimizing resource selection.

Keywords: Soft-resilient Signal Processing, Reconfiguration for Autonomous Operation, Runtime Multi-objective Optimization, Evolvable Hardware, Field Programmable Gate Array devices

Reference to this paper should be made as follows: Naveed Imran, Rizwan A. Ashraf, and Ronald F. DeMara (2014) 'Power and Quality-Aware

Image Processing Soft-Resilience using Online Multi-Objective GAs', *Int. J. of Computational Vision and Robotics (IJCVR)*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Naveed Imran received the Ph.D. degree and M.S. degree in Electrical Engineering from the University of Central Florida (UCF), Orlando, Florida, USA in 2010 and 2013, respectively. His research interests include hardware design of DSP systems, computer architecture, FPGAs, reconfigurable hardware for image/video applications, and reliable VLSI architectures.

Rizwan A. Ashraf received the Bachelor's Degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan in 2007. He is currently a Ph.D. Candidate in Computer Engineering at the University of Central Florida, where he has obtained the Master's Degree in Computer Engineering. His research interests are in reliability-aware and low-power autonomous techniques for reconfigurable logic and custom ASIC devices.

Ronald F. DeMara received the Ph.D. degree in Computer Engineering from the University of Southern California in 1992. Since 1993, he has been a full-time faculty member at the University of Central Florida. His research interests are in Computer Architecture with emphasis on Evolvable Hardware and Distributed Architectures for Intelligent Systems. He has published approximately 140 articles on these topics and holds one patent. He is a Senior Member of IEEE and a Member of ACM, and ASEE, and serves as an Associate Editor of IEEE Transactions on Computers. In 2008, he received the Outstanding Engineering Educator Award in the Southeastern United States from IEEE.

NOMENCLATURE

N	Total Number of Processing Elements (PEs)
N_a	Number of Active PEs
$\mathbf{T} = \{T_1 T_2 \dots T_N\}$	Set of Throughput Tasks
T_0	Zero-task, i.e., Empty Task
$\mathbf{P} = \{p_1 p_2 \dots p_N\}$	Set of Task Priorities
$\mathbf{H} = \{h_1 h_2 \dots h_N\}$	Set of PE Health Status
$\mathbf{V_a} = \{PE_i\}; \forall_i T_i \neq T_0$	Set of Active PEs
f	Multi-Objective Function
f_c	Crossover Fraction for GA
Γ	Current Value of Health Metric
$\ddot{\Gamma}$	Target Value of Health Metric
π_k	Power Consumption of k^{th} Task
d_i	Defectiveness Estimate of i^{th} PE
p	Population Size for Genetic Algorithm (GA)

1 Introduction

Autonomicity is a desirable property for Digital Signal Processing (DSP) architectures in dynamic real-time environments. Ideally, image processing systems should maintain

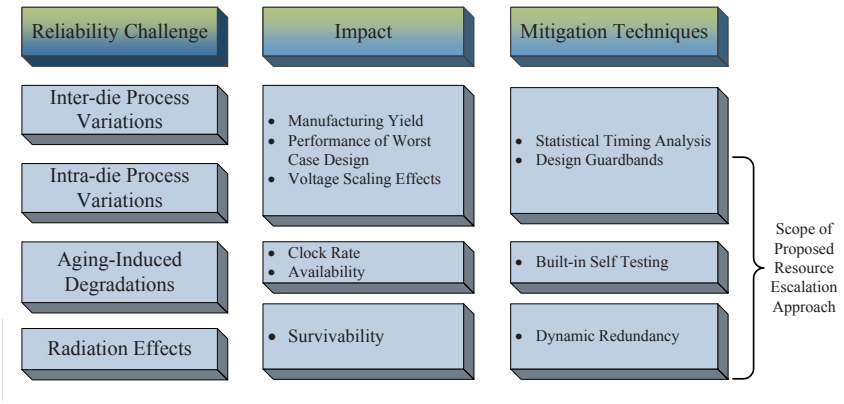


Figure 1: Reliability challenges facing DSP systems due to increased technology scaling

the desired levels of accuracy with rapid convergence and optimized power consumption throughout a range of operating and reliability conditions. Survivability under these constraints can be enhanced by the provision of self-awareness properties at the system-level [1]. These attributes are most critical in real-time environments where the reliability of CMOS devices in nanoscale regimes is becoming increasingly sensitive to variations in temperature, process manufacturing tolerances, aging effects, and supply voltage stability. For example, to achieve energy-efficiency in nanoscale CMOS circuits, voltage scaling [2] continues to be realized as one of the most effective methods. However, near threshold voltage operation of these circuits can manifest process defects and variations as run-time computational errors which appear in the context of signal processing applications as accuracy degradation [3]. This paper develops a cross-layer signal processing architecture which uses self-adaptation to address these concerns.

Fig. 1 provides an overview of stability and reliability issues in sub-45 nm CMOS systems and some popular corresponding mitigation techniques. This layered model can be adapted and leveraged in DSP applications due to their inherent *soft-resilience* to errors. The soft-resilience property arises from redundancies in input data and at the application-level from inexact perception of output quality by the user [4]. For instance, an example of soft-resiliency at the algorithm-level is Kalman filtering in which errors in prediction at a given instance are corrected in subsequent iterations. Soft-resiliency is compatible with a recent trend of attempting to sustain Moore's law by designing computing systems using various error-permissible computing models [5][6]. The inherent resiliency of signal processing algorithms allows some relaxation of exact computation to embrace this type of soft-computing paradigm. In particular, the provision of error de-sensitizing mechanisms and graceful degradation is desirable to maintain output quality objectives.

In this paper, a cross-layer soft-computing approach leveraging the different priorities of DSP tasks on the overall output accuracy is presented. These are evaluated at runtime by monitoring a specific health metric which is a dynamic operating condition observed at the cognitive layer to trigger adaptation. This avoids the complexity of rigid exhaustive fault coverage by handling only those subset of errors which affect the output quality beyond acceptable tolerances. Thus, the system adapts concisely to manifested errors while nullifying false-positive demands. In addition, the need to synthesize test vectors with high resource coverage becomes unnecessary.

The proposed system is demonstrated using Field Programmable Gate Array (FPGA) devices which are widely chosen to accelerate DSP applications in hardware. As a computational platform, an additional major advantage of FPGAs is their runtime reconfigurability. Reconfigurable regions can be defined at design-time for a circuit and later at runtime these regions can be re-assigned to alternative tasks dynamically. To reconfigure a Processing Element (PE) with an alternative task, the other regions of the device which are not being reassigned do not need to be removed from service. This online partial reconfiguration ability provides great flexibility for novel soft-computing approaches at the architectural level.

Herein, the approach of reconfiguration to maintain accuracy and power efficiency is formulated as a multi-objective optimization problem. The term *device configuration* will be used to denote a distinct mapping of tasks assigned to PEs. The event of *reconfiguration* will be used to denote the task re-allocation process within a reprogrammable hardware fabric. As tasks have different priority levels, the tasks are best mapped for execution when their priorities are positively correlated to the healthiness of underlying resources in the computational fabric. Thus, those configurations are preferred in which prioritized tasks are mapped to healthier elements in the resource pool of reconfigurable regions. As the number of potential mappings can be quite large, the optimization capability of multi-objective Genetic Algorithms (GAs) to search the mapping space for throughput quality and power consumption alternatives is employed. The proposed approach is evaluated for a Support Vector Machine (SVM) and a Discrete Cosine Transform (DCT) implemented in FPGA hardware. Performance metrics such as power consumption, measure of confidence, and PSNR demonstrate that a *health metric based multi-objective online evolution approach* achieves those objectives while incurring acceptable runtime overhead costs.

The following are the main contributions of this work:

1. The tradeoffs of reliability and power savings are formalized as a generalized runtime mapping problem based on the underlying resource performance and operating workload.

2. A multi-objective GA approach is demonstrated for this mapping optimization problem in which a population of solutions is guided by a novel adaptive guidance function.

3. Instead of requiring redundant units for fault-detection, a throughput health metric is identified. Thus, fault-detection is feasible using a uniplex instance of the datapath without requiring redundancy for error checking. This also allows a consolidation phase to distinguish transient conditions in the detection method.

4. Soft resilience is introduced as an iterative task remapping process to maintain the output quality metric within acceptable limits. Namely, an integrated diagnosis and recovery scheme is presented which neither requires a voting mechanism nor bringing the system entirely offline as recovery progresses.

2 Related Work

Biological systems have inherent self-repairing capabilities which have inspired signal processing research to mimic these natural adaptive processes in reconfigurable digital fabrics. Thus, research interest has been increasing toward electronic systems which can sustain adverse events, yet remain operational or at least partially operational. Consequently,

self-repair and self-healing mechanisms have been proposed for hardware by various researchers [7],[8]. These mechanisms rely on identifying or employing some form of redundancy, reconfiguration, or both. To realize these properties in a DSP system, it is useful to identify how a layered model emphasizing the impact of signal processing tasks on output correctness and the runtime reconfiguration of FPGA resources based on Evolvable Hardware can be leveraged.

Evolvable Hardware has been proposed as a reconfiguration-based approach to achieve fault tolerance in electronic designs. These methods extend static fault tolerance techniques at design-time which attempt to make designs more robust to faults [8]. In particular, runtime techniques reconfigure hardware resources at runtime to refurbish the circuit [9]. Previous works establish the successful use of GAs for adaptive self-recovery of hardware systems based on reconfigurable logic platforms, especially in FPGA-based systems [9]. A survey of fault-handling techniques ranging from passive to dynamic in classification are presented in [10] to tackle hard faults in SRAM-based FPGAs for relatively small-sized circuits.

Researchers have devised runtime evolutionary techniques to realize fault-resilient electronics through iterative selection [11]. Conventionally, fault-tolerance at the system level is attained by either employing passive redundancy to mask these output errors immediately or by executing a phased fault-handling flow consisting of fault-detection, diagnosis, and recovery stages. Although, previous attempts have been made to combine architecture and algorithm level knowledge [12], there remains a need to develop frameworks utilizing cross-layer information in a way that leverages the soft-resilience present in DSP applications.

It is worth-mentioning that a greedy algorithm like [13] is successful at small-scale optimization with single objectives (i.e., throughput), large-scale multi-objective problems necessitate meta-heuristic algorithms to explore the associated large search space. The proposed scheme is based on the technique of performing iterative reconfigurations until the system's output meets quality objectives. To avoid the requirement of redundancy which can incur significant area overhead in the case of cold-spares and power consumption in the case of replicated paths for comparison-based detection, the proposed approach leverages a health metric and inherent computational priority in its system design. Such an approach is especially promising for DSP applications which can accommodate a graceful degradation of functionality.

As illustrated in Fig. 2, there is a spectrum of techniques dealing with error-tolerance of DSP systems ranging from the device-level up to the system-level. Fault-handling at the *architectural-level* is often oblivious to the error-mechanisms in the underlying hardware. For example, a Triple Modular Redundancy (TMR) arrangement is a technique in which a datapath is replicated to create three identical instances and then each output is passed into a majority voter for selection. Although, a TMR scheme maintains all three instances in the datapath thereby achieving fault-masking capability, the resource overhead is considerable in both area and power consumption, even for vast majority of system lifetime which may be fault-free. Namely, a TMR arrangement incurs a power consumption overhead that is approximately three-folds as compared to a simplex arrangement even if the overhead of voter overhead is considered negligible. On the other hand, a Concurrent Error Detection (CED) arrangement detects faults by comparing the output of two replicas subjected to the same inputs [14]. A discrepancy reveals faulty nature of either instance without pinpointing which of the modules is faulty. Again, the area and power overheads are significant concerns in CED. As an alternative, Built In Self Test (BIST) mechanisms diagnose faulty components by evaluating them with some test inputs generated by an Automated Test Pattern Generator

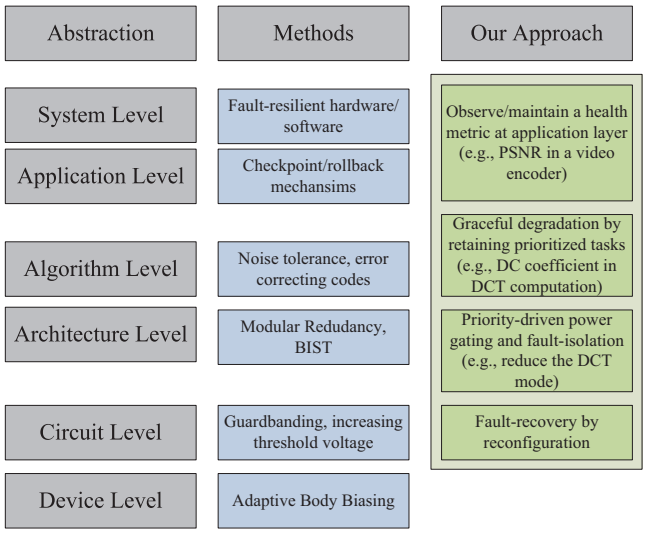


Figure 2: Hierarchy of fault-mitigation techniques at various abstraction levels

(ATPG) to provide one-time or periodic fault-assessment. In practice, a BIST scheme rarely achieves 100% coverage, yet may generate false alarms [15]. Moreover, an evaluation of some test vectors may not necessarily correspond to the actual runtime scenario of a module under test. Thus, the proposed technique of *health metric based multi-objective online evolution* relies on the actual behavior of the signal processing module under runtime conditions. It is shown that an evolutionary-inspired scheme of reconfiguration which correlates the output healthiness information with the task mapping history can meet these goals.

Algorithmic-level fault-handling approaches exploit signal processing algorithm properties to make the system robust and error-resilient. Hegde and Shanbhag *et al.* [2] proposed an Algorithmic Noise-Tolerance (ANT) technique to compensate the errors introduced into DSP architectures due to voltage scaling beyond the nominal operating point. Such voltage scaling is an effective method of reducing power consumption, yet the correctness of throughput becomes an issue when the supply voltage is scaled beyond a critical voltage. To mitigate these concerns, the authors developed a prediction-based error control scheme which requires knowledge of the system transfer function which was a digital filter in their case study. Applying algorithmic-level fault-handling to video processing, Varatkar *et al.* [16] proposed a sub-replica of the motion estimation block to concurrently check the error-prone main block. Meanwhile for image processing, Kim *et al.* [17] proposed a soft voter employing a Bayesian detection technique. The soft voter is demonstrated to provide correct output in a Discrete Cosine Transform (DCT) based image coder. Lisboa *et al.* [18] proposed a fault-tolerance technique to mitigate faults in matrix multiplication algorithms, which comprise the heart of many signal/image processing applications.

Finally, power consumption remains one of the key issues even in deep-scaled CMOS technology. This is especially true in both high-density deep submicron ASIC/FPGA designs due to cooling considerations, as well as portable electronic systems where battery life, size, and weight are concerns. Although voltage scaling has been used to drastically reduce power

consumption, this increases the circuits' susceptibility to faults, and hence the desirability for soft-resilient operation under these conditions. While there is a body of research work dealing with power versus fault-tolerance tradeoff at design time [19], there remains a need to develop runtime techniques to autonomously manage these tradeoffs. Runtime techniques are also promising to handle faults in unforeseen mission-critical scenarios as well as commonly-encountered manufacturing-induced process variations that impact yield, and transistor aging characteristics. The presented *health metric based multi-objective online evolution* scheme addresses the issue of power consumption and quality tradeoffs through a novel runtime architectural adaptation technique formalized in the next section.

3 Problem Formulation and Methodology

Consider a computing Array-Under-Test (AUT) realized by a set of N-Processing Elements (PEs), namely PE_1, PE_2, ..., PE_N each executing a task T_1, T_2, ..., T_N, respectively. The priority of the tasks assigned to the PEs is given by a vector $\mathbf{P} = \{p_1 p_2 \dots p_N\}$ having its i^{th} component denote the priority of the i^{th} task. As a fault-recovery provision, a PE can be reconfigured to an alternative function or equivalently, a task can be re-assigned to an alternate PE at runtime. Given a homogeneous computing array of PEs, a reconfiguration controller can re-assign an alternative task to any PE in the array. Let the healthiness of resources which comprise the PEs be denoted by a vector $\mathbf{H} = \{h_1 h_2 \dots h_N\}$ as illustrated below. In the proposed fault-handling scheme, the defectiveness degree of PEs is assumed to be unknown. The formulation here allows utilization of a-priori information about the priority of tasks mapped at runtime as discussed in Sections 4.2 and 4.3.

Power consumption in such an AUT can be reduced by power gating of some of the PEs [20],[21] which acts to exclude them from operation. The choice of PEs selected for use depends upon input signal characteristics, assigned tasks priorities, and desired quality levels. To maintain the generality of the notation without becoming restricted to a specific signal processing algorithm, consider a *zero-task* denoted by T_0 which corresponds to the power-gating OFF condition of the underlying computational resources for which the task has been mapped. In practice, a T_0 task can be realized by a power-gating technique in an ASIC implementation or by configuring a blank bitstream into the reconfigurable region in an FPGA. The latter approach is selected for the case studies as a FPGA device is utilized for experiments in this work.

A set of active PEs is defined as $\mathbf{V_a} = \{PE_i\}; \forall_i\ T_i \neq T_0$. Thus, $\mathbf{V_a}$ contains those PEs which are assigned a non-zero task and thus these PEs realize the computational throughput of the system. There is a one-to-one mapping of tasks to active PEs such that the cardinality of the set of active PEs is given by $|\mathbf{V_a}| = N_a$ where $N_a \leq N$. In the following discussion, the terms processor *node* and *PE* are used interchangeably. It is worth highlighting the assumptions of the above formulation:

1. A PE can be configured with any task, namely, a homogeneous array of PE resources is considered here.

2. Input data can be multiplexed to any or all of the PEs.

Each PE can be configured with any task, as the size of PE is determined by the resource intensive task in $\mathbf{T}$. Consequently, the largest configuration used determines the quantity of resources available to the set of PE nodes. This along with the reconfiguration capability of

FPGAs allows to map any task to any physical resource on the FPGA fabric. Furthermore, the input data can be applied to any node by using bus-macros in the target FPGA platform, e.g., as per the Xilinx-specified partial reconfiguration based design flow.

Fig. 3 shows an architectural view of the proposed fault-handling approach and the corresponding mapping algorithm is given in Algorithm 1. The computationally-demanding portion of a DSP application has been mapped to an array of PEs to accelerate throughput. The reconfigurable array of PEs is managed by the reconfiguration controller which maps tasks into the computational regions. A health metric is communicated from the software application to the reconfiguration controller. The top-level software application can be executed on a PowerPC processor as such on-chip processors are provided in most commercially-available FPGA chips. The value of the health metric will vary due to either input signal characteristics or hardware defects. To identify the latter case, a health metric outside nominal operating range triggers the fault-identification process. To keep the area overhead minimal, fault-identification is performed by a comparison-based discrepancy detector on a PE-scale resolution rather than at the AUT level. In particular, a Reconfigurable Slack (RS) [13] region is utilized to consolidate a non-transient fault-detection of decreased health metric value. In particular, a RS is a single task-grained tile reserved as a cold-spare for the entire design; only one RS is needed regardless of N. Operationally, an RS is loaded to test suspect PEs successively, and in order of their priority of impact on the output quality. Namely, a discrepancy between the output of an active PE and RS indicates a transient or permanent hardware fault. Thus, fault-identification is asserted without rendering any decision about exact location of fault being either in the active PE or the RS. Afterwards, the diagnosis and recovery process is carried out by the GA engine embedded in reconfiguration controller to locate which of these two is actually faulty. However, if no discrepancy is observed between the active PE and RS, then the health metric is assumed to have exceeded tolerance simply due to the input characteristics or due to a transient fault in the computational resources which can subsequently be resolved. Thus, it is necessary to focus on the case where the active PE and the RS outputs are discrepant as discussed below.

Algorithm 1 Guidance function driven multi-objective GA

Require: $\mathbf{T}, \mathbf{P}, N, w_1, w_2$
Ensure: $\ddot{\Gamma}$
  Initialize $\mathbf{V_a} = PE_i$, $i = 1$ to N_a, $N_a = N$
  while $f_1 < \ddot{\Gamma}$ // System is unhealthy **do**
    Evaluate $f = w_1 * f_1 + w_2 * f_2$
    while $(\{k | k \in \mathbf{V}, k = 0\} = \phi)$ // Identify at least one healthy node **do**
      Designate v_s as checker(s) $(N_a + 1) \leq s \leq (N_a + N_s)$ thus $V_s = \{v_s\}$
      while $i \leq N_a$ **do**
        Reconfigure RS(s) with the same functionality as v_i
        Perform CED among CUTs when $N_s = 1$, $d_i \leftarrow 0$ for v_i which shows no discrepancy
        Evaluate h_i based on Eq. 4 for each node
        $i \leftarrow i + 1$
      end while
      Relocate the RS by updating $N_a = N_a - N_s$, Re-initialize $i = 1$
    end while
    Evaluate the guidance function in Eq. 5 to determine task mapping using updated N_a
  end while

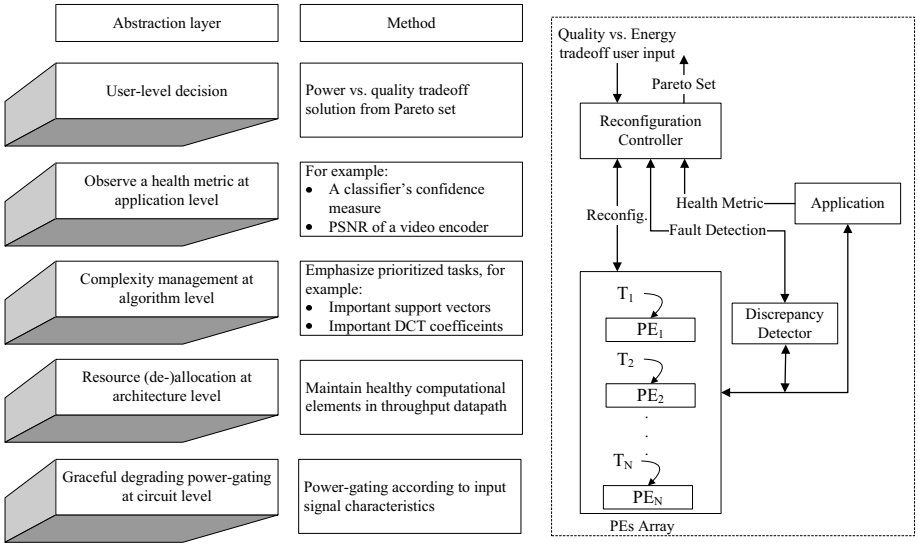


Figure 3: Cross-layer fault-handling architecture with hierarchical support: pareto set solutions, health metrics, task priorities, computational resources, and on-demand power-gating

The fault-handling processes employs a data structure representing the task mapping to PE resources. The PE array and corresponding task mappings use a fixed-length chromosome in this formulation which is suitable for GA processing. The genetic representation is illustrated in Fig. 4 which shows array of 7 PEs concurrently executing a set of tasks. A PE can be configured with any task T_i where $0 \leq i \leq N$. An example of the task-mapping chromosome is shown in Fig. 4. The number of fields in a chromosome is equal to the number of PEs in the processing array. The value of a particular field identifies the task number allocated to the corresponding PE. For example, the third field in the chromosome contains the value 4 implying that PE_3 is assigned to execute task T_4. It is worth noting the exemplified task mapping of PE_4 being allocated T_0 which corresponds to configuring a blank bitstream on this particular PE. Here, zero or more PEs may be configured with T_0 based on the instantaneous or near-term throughput quality requirements. Such a dynamic assignment of blank tasks acts to reduce and dynamically optimize power consumption at the expense of some quality degradation whereby a functional task, for instance the corresponding low priority DCT coefficient, is decommissioned from the datapath. The formulation of the tradeoff of these objectives is described below.

3.1 Multi-Objective function

The power versus quality tradeoff in DSP systems is formalized as a optimization problem using the composite function f to be minimized given by:

$$f = w_1 f_1 + w_2 f_2 \tag{1}$$

where w_1 and w_2 are corresponding weights of the opposing functions f_1 and f_2. The functions f_1 and f_2 represent the throughput degradation and power consumption,

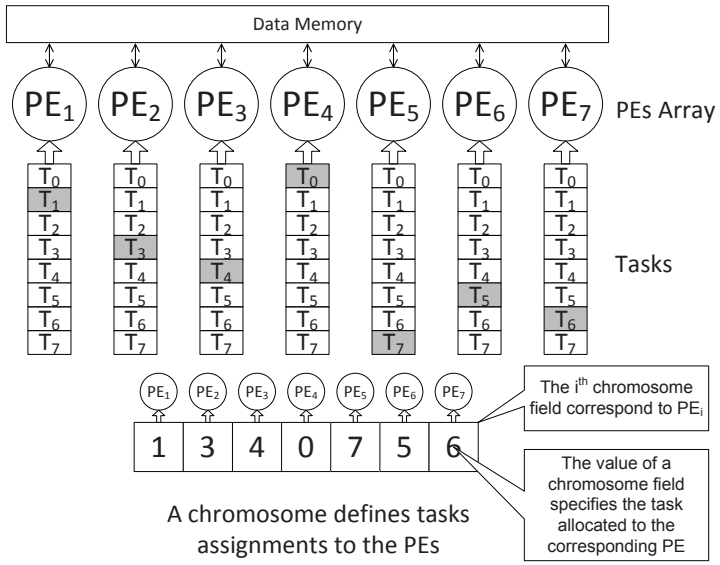


Figure 4: An array of 7 configurable PEs and its genetic representation

respectively, of the current task mapping using the selected computational resources. An effort to improve f_1, i.e. minimize quality loss, results in degradation in f_2, i.e. consumption of more power, and vice-versa. The pareto solution set to this problem corresponds to a set of configurations aimed at exploring the design space of the quality versus power efficiency tradeoff as follows: 1) The objective of soft-resilience is achieved by mapping prioritized tasks to the healthy resources, 2) The objective of power efficiency is achieved by loading blank bitstreams into both failed and healthy PEs. Of course, disabling healthy PEs while saving power, degrades throughput as discussed below.

3.1.1 Throughput Degradation

The evaluation interval size τ is defined as the period of calculations over which the fitness assessment of an AUT is performed, expressed in units of the number of inputs as specified by the user. For higher throughput quality and accuracy over an evaluation interval, the following metric which is essentially a measure of Mean Squared Error (MSE), should be minimal:

$$f_1 = \frac{1}{\tau}\sum_{i=1}^{\tau} ||\Gamma_i - \ddot{\Gamma}||^2 \tag{2}$$

where Γ_i health metric value at time i and $\ddot{\Gamma}$ is the target value of health metric as set by the user. The health metric selected can include the PSNR, bitrate, measure of confidence, or other application-level throughput quality indicator.

3.1.2 Power Consumption

Power consumption of an array of PEs is directly proportional to its size N. Therefore, a normalized power consumption measure is defined in terms of N and is given by:

$$f_2 = \frac{\sum_{\forall k \in V_a} \pi_k}{\sum_{i=1}^{N} \pi_i} \tag{3}$$

where V_a is the set of PEs assigned with non-zero tasks and π_k is the power consumption of the k^{th} task. Thus, the AUT's power consumption is maximized when all N PEs are assigned to have active tasks resulting in $N_0 = 0$. On the other hand, power consumption is minimized when all the PEs are assigned with zero-task assignment yielding $N_0 = N$, yet throughput quality in that case is also non-existent and hence not an option selected in practice.

The objective functions given in Eq. 2 and Eq. 3 are oppositional. A higher number of active PEs results in increased throughput quality at the expense of increased power consumption. On the other hand, *blanking* [22] of the PEs results in reduced power consumption while incurring output quality degradation. A runtime multi-objective GA approach is used in finding a pareto optimal set as described below, thus spanning throughput versus power optimization, and also soft-resilience against faults using an autonomous strategy based on a feedback arrangement.

3.2 Guidance Function

Although a solution to the minimization of Eq. 1 is the objective that realizes the desired soft-resilience operating point, the search space of the mapping problem is considerably large. For example, an exhaustive search will require $(N+1)!$ reconfigurations in a cluster of size N to explore the search space. Thus, exhaustive or randomized approaches can be intractable for absolute minimization of large-size problems which render the practicality of non-guided search. To this end, it is proposed to incorporate evaluation history information of the influence of mapping on throughput quality which further guides the population towards the pareto front. The history information of the individuals maintains a health estimate of the computational resources which in turn prunes the search space of the problem. Thus, adaptive guidance of the population using a runtime healthiness estimate will be shown to benefit the convergence of the online multi-objective GA.

An a-priori knowledge of tasks' default priorities is generally useful in terms of carrying out a graceful degradation strategy, and is available in many cases such as computing the coefficients in a DCT core. Here the DC coefficient should be computed on the healthiest resource. However, such a knowledge of healthiness of computational resources is often dynamic and may be subject to soft-errors due to aggressive voltage scaling, aging, and supply variations, or even permanent faults. Thus, it is beneficial to estimate the healthiness of computing resources at runtime to evaluate Eq. 5. This uses the nodes's output discrepancy history to develop its healthiness estimate h_i. The overall error observed in the output is also weighted by the priority level of its assigned task. Thus, defectiveness estimate at evaluation instance t, $d_i(t)$ is estimated as follows:

$$d_i(t) = d_i(t-1) + p_j * |\Gamma - \ddot{\Gamma}| \tag{4}$$

where $d_i(t) = 1/h_i(t)$ and p_j = priority value of task j assigned to PE_i.
Thus, to calculate the defectiveness estimate d_i of a node i, the throughput degradation

is weighted by the task-priority value and accumulated into the previous estimate of d_i. By employing the fault articulation history as well as the task priorities, the defectiveness estimate becomes an effective measure to lead the adaptation towards a preferred mapping.

The guidance function g can thus be realized as:

$$g = \frac{|\sum_{i=1}^{N} p_i h_i - \sum_{\forall_k k \in V_a} p_k h_k|}{\sum_{i=1}^{N} p_i h_i} \tag{5}$$

Its minimization guides the GA to find the pareto front while maintaining partial throughput during fault-resolution phase. Here, the tasks' priorities are weighted by the healthiness of the underlying resources on which the tasks are mapped to. As Eq 5 reveals, a minimum value of g corresponds to the mapping when vectors **H** and **P** are highly correlated. That is, high priority tasks are mapped to healthier resources. Such a guidance function assists in guiding the GA according to the fitness function when system is faulty. Otherwise, the fitness function continues to use f_1 and f_2 functions for throughput assessment and power optimization, respectively.

The proposed fault-handling methodology is summarized below:

1. isolate faulty resources when the application-level health metric exceeds tolerance,
2. consolidate non-transient fault conditions via a CED-based discrepancy-based detection using RS,
3. invoke the GA during the resource escalation phase of task remapping, and
4. select an individual from the obtained pareto set to finally map tasks on to the fabric based upon their quality and power consumption tradeoff.

4 Execution Results

4.1 Synthetic Nodes Simulation

To illustrate the process and the impact of the function given in Eq. 1, the approach is evaluated using an array of simulated nodes. For this purpose, a PE-array of size $N = 7$ is chosen and the fault scenarios are simulated by assigning healthiness values to the PEs as listed in Table 1. The priority values are assigned to various tasks such that T_1 receives the highest priority (i.e., the maximal value of 7 for an arrangement comprising 7 possible tasks, while T_7 receives the least priority, i.e.,the value minimal non-zero task value of 1. Thus, **P** vector's component values reflects the reverse ordering of Task Numbers. For example, in a DCT, Task T_1 would correspond to the computation of the DC coefficient. However in this illustrative example, for generality assume the effect of priority values on the overall output is unknown at this point. Therefore, it is not feasible to initially evaluate the first term of the objective function given in Eq. 1. Instead, the healthiness values are assumed to be already available in this scenario in the form of a monotonically decreasing linear function while the guidance function of Eq. 5 is considered to be the first term of the objective function of Eq. 1. The duration of the evaluation interval is considered as one sample such that $\tau = 1$, i.e., the objective function is evaluated for every input in this synthetic nodes case study. It is worth mentioning here that although the healthiness and priority values are generated by a linear function in the synthetic nodes simulation case, these values can be substituted

Table 1 Example of priority values, **P** , and healthiness of resources, **H**, in a synthetic array of $N = 7$ nodes

PE Number i	1	2	3	4	5	6	7
H_i	0.25	0.2143	0.1786	0.1429	0.1071	0.0714	0.0357
Task Number j	1	2	3	4	5	6	7
P_j	7	6	5	4	3	2	1

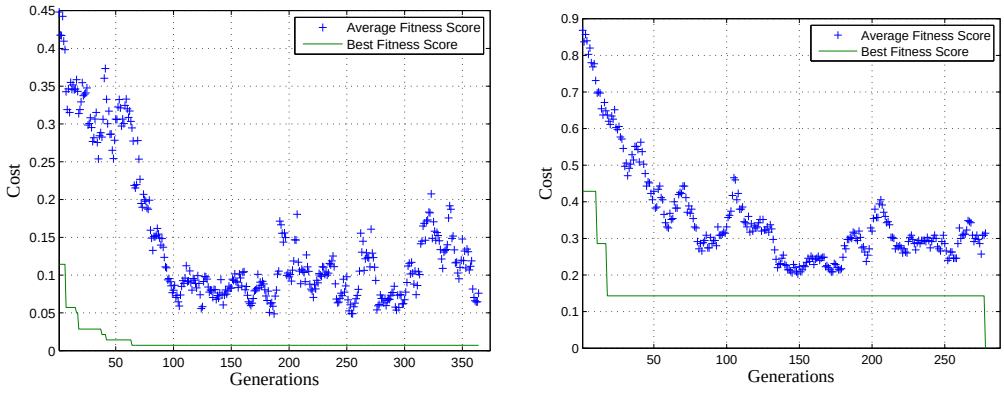
Table 2 GA parameters

Parameter	Value
Population Size	50
Migration Direction	forward
Migration Interval	20
Migration Fraction	0.200
Population Creation Function	Uniform
Fitness Scaling Function	Rank
Selection Function	Uniform
Crossover Function	Two-point
Elite Count	2
Crossover Fraction	0.7
Mutation Rate	0.01

with results from any fault model and the impact on output quality in the practical case studies as discussed further below. For example, the fault impact is simulated by a stuck-at fault model and the corresponding PEs are evaluated for functional output in practical case studies with favorable results.

The GA parameters used are given in Table 2. The Population Size corresponds to various tasks configurations of the AUT. Migration parameters specify the individuals of population's movement among multiple sub-populations. The individuals are created by a uniform function while being selected using rank criteria [23]. The standard two-point crossover operator is used with the mutation option compatible with a generational GA. Elitism ensures that some of the best individuals are guaranteed to be propagated to the next generation.

Figure 5 shows the throughput degradation and power consumption objective costs on the vertical axis for various iterations as two curves over time in units of generation number on the horizontal axis. The two curves depict the average behavior of the population as the upper scatter plot and the best-performing individual's behavior as the lower curve on each plot. The throughput degradation is described in terms of the guidance function. The optimal solution reached by the GA was $\{1,2,4,3,5,6,7\}$ after 500 generations. As Figure 5 shows, the average behavior significantly improves within 100 generations, and then fluctuates due to the mutation operation. A sufficient population size together with a mutation function is necessary in order to diversify the population to reach a good solution in terms of meeting multiple criteria. The cost scores are defined in terms of the number of active PEs as well as the synthetic priority and health values. Then, after normalization, the unit-less ratios are the cost scores to be minimized. A converging trend of the cost plots after 100 generations implies that the proposed evolutionary methodology can achieve power and quality goals by employing the runtime-behavior information of the processing array. The global optimum



(a) Throughput degradation (b) Power consumption

Figure 5: Cost functions

solution for this problem is $\{1,2,3,4,5,6,7\}$ as it corresponds the resource escalation of the weighted prioritized tasks over the reconfigurable fabric. Thus, the GA is successful in improving the configuration, within a reasonable number of generations suitable for runtime operation.

Figure 6 shows the pareto set of solutions of the Multi-Objective Online Evolution (MOOE) problem. Here, both costs, namely throughput degradation and power consumption are employed to engage quality and energy efficiency tradeoffs, respectively. For example, a 40% tolerable behavior in terms of throughput degradation allows power consumption reduction to 30% of the maximum budget. A further reduction in power consumption is feasible as low as only 10% if approximately 80% throughput degradation can be tolerated. As the result shows, the proposed evolvable hardware MOOE recovery formulation allows finding a set of optimal solutions which facilitates design space exploration in terms of quality and energy efficiency tradeoffs as a continuum.

4.2 A Computer Vision Case-Study: Support Vector Machine (SVM)

A second case study is undertaken using a SVM to evaluate the *health metric based multi-objective online evolution* scheme. A hardware core of a SVM is monitored for its health status by observing the Measure of Confidence. The control feedback mechanism is that an unusually low confidence measure from a SVM can indicate hardware failures. Thus, the proposed online evolution mechanism architecturally adapts the SVM core to recover from failures by utilizing a health metric based feedback in the recovery loop.

SVMs are popular as supervised machine learning methods in classification problems. While the learning phase can either be carried out offline or online, the testing phase is usually desired online due to real-time requirements of many applications. Thus, hardware implementation is favorable, to accelerate intensive computations involved. For training purpose, LIBSVM [24] is employed, and thereafter the learned kernels are implemented in hardware by Multiply Accumulate-based PEs. Because SVMs are favorable in image detection tasks in space missions [25], they are considered as a case study herein to evaluate

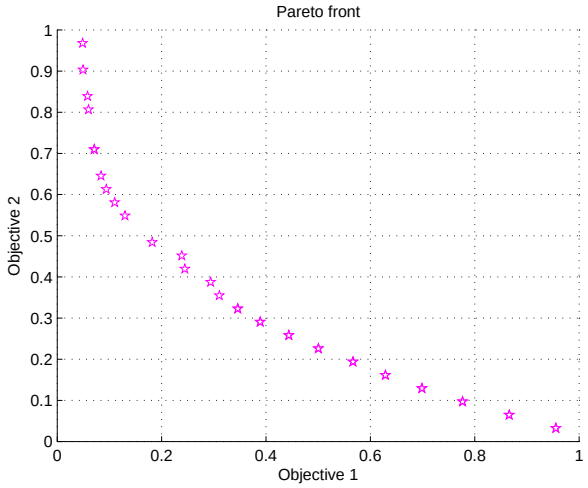


Figure 6: Pareto set of solutions for the synthetic graph MOOE problem

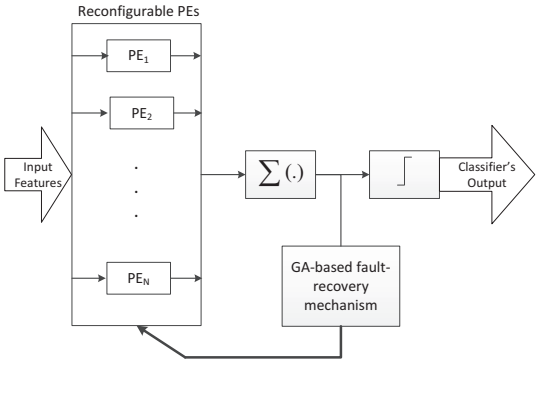


Figure 7: Functional block arrangement for proposed approach in a Self-Healing SVM case study

the proposed self-healing mechanism. Furthermore, the approximate classification process from multi-dimensional dataset allows to study interesting trade-offs of power and quality.

An architectural view of the proposed self-healing SVM is provided in Fig. 7. In this pattern recognition task, the SVM's measure of confidence is employed as a health metric to guide the architectural adaptation through fault scenarios and power efficiency tradeoffs. As the objectives such as power consumption are secondary to minimally-acceptable throughput quality, first the proposed approach is evaluated in terms of correctness under fault-handling conditions. Fault injection and fault recovery results are listed in Table 3 and Table 4, respectively. Here negative values indicate that this sample belongs to "False" class and positive or zero values indicate "True" class. The impact of fault for samples shown in Table 3 does not impact the classification. Results shown in Table 4 for CoverType [26]

Table 3 Fault impact on the classifier output

	Fault-free Classifier		Faulty Classifier		
Sample Number	Estimation Function	Detector Output	Estimation Function	Detector Output	Actual Class
1	-1.0675	False	-0.8485	False	False
2	-1.0645	False	-0.8472	False	False
3	-1.0019	False	-0.7211	False	False
4	-0.8932	False	-0.6180	False	True
5	-1.0126	False	-0.7939	False	False

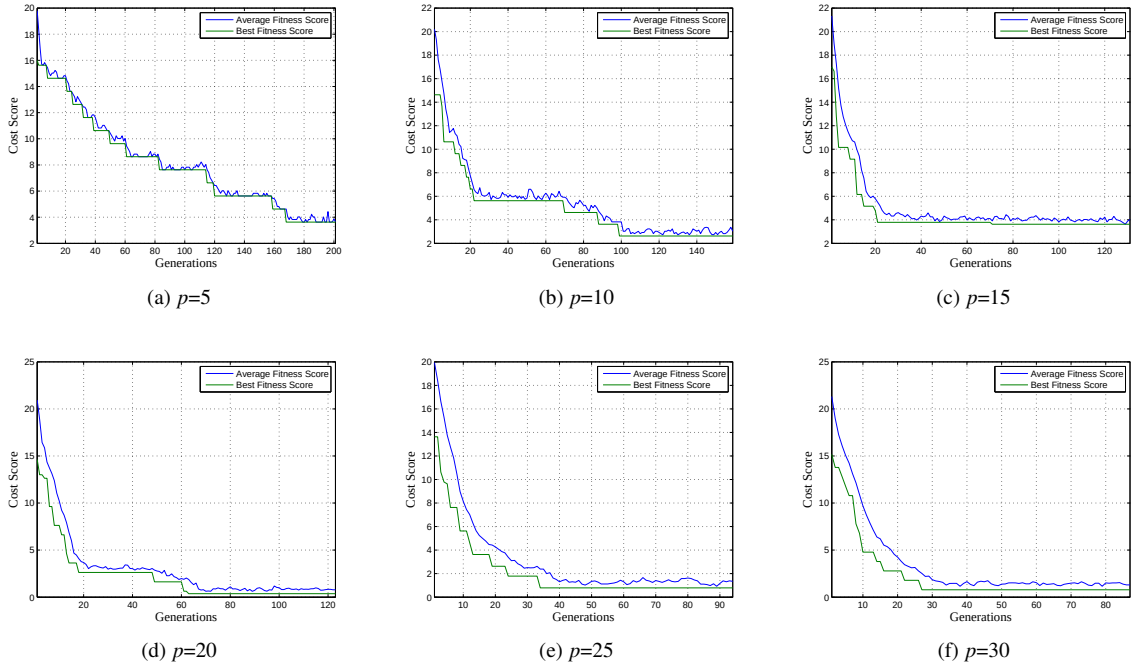
Table 4 Fault recovery for *Covertype*[26] data set compared to 75.68% Fault-free Classifier's Accuracy

Number of Faulty PEs	*Faulty Classifier's Accuracy*	*Recovered Classifier's Accuracy*
1	69.12%	75.19%
2	59.09%	73.24%
3	58.09%	73.02%
4	52.26%	72.83%
5	50.24%	69.12%

dataset demonstrate that the proposed method is able to recover a faulty SVM classifier with only 50.24% classification accuracy to 69.12% accuracy whereas the original fault-free classifier had a 75.68% accuracy. Such graceful degradation can be acceptable, or even desirable in many image pattern recognition tasks, especially when low-power and survivability objectives are to be sustained simultaneously.

Fig. 8 illustrates the effect of population sizes on convergence of a single objective GA. A large population size is advantageous in terms of exploring the problem's search space as it is evident for population size of 30 as compared to a population size of 5 which needs far more number of generations of the GA to converge. Convergence required approximately 160 generations for population size of 5, approximately 100 generations for population size of 10, and about 30 generations for larger population sizes. However, it's worth mentioning that a large population size requires a longer duration to evaluate the individuals for the purpose of estimating their fitness behavior. Thus, a large population size may not necessarily correspond to faster convergence. Regardless of population size selected, it is important to note that only single instance of hardware resources is used; the population size represents only the number of entries in the data structure used to represent the dynamic set of mapping permutations being explored by the GA.

A critical operation in GAs is crossover which combines attributes of two existing individuals in the population to create a novel individual. Fig. 9 illustrates the impact of the cross-over operation. In this experiment, a fixed population size of 25 is selected based on the sufficiency of that population size indicated by the previous experiment. In this experiment where 20% of the population undergoes crossover operation (i.e., $f_c = 0.2$), the cost score improves after 100 generations with elitism retaining the two best-performing individuals. Thus, the guidance function is effective at escalating the computational resources as per application needs. On the contrary, the cost score levels out only after 25 generations for an excessive crossover fraction parameter (i.e., $f_c = 0.9$). However, it is to be noted in the latter case that the algorithm cannot further improve the best fitness value after generation 14, because all the individuals in the population essentially become identical. Such an overly-early convergence does not help to find the best individual in a fewer number of generations. Thus, this case study illustrates the benefit of latency to converge the reconfiguration solution

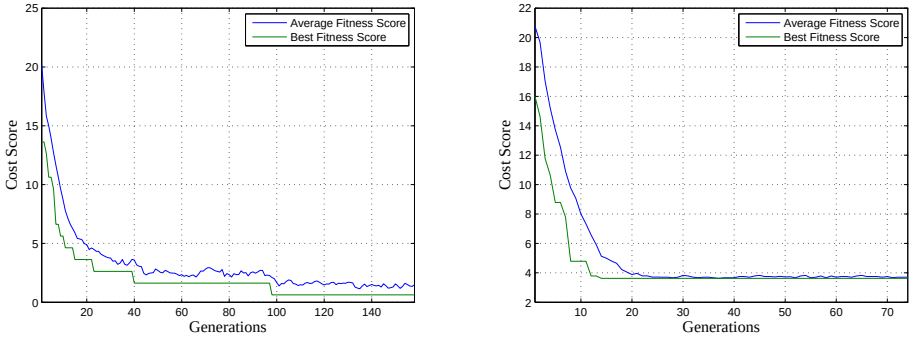


(a) p=5 (b) p=10 (c) p=15

(d) p=20 (e) p=25 (f) p=30

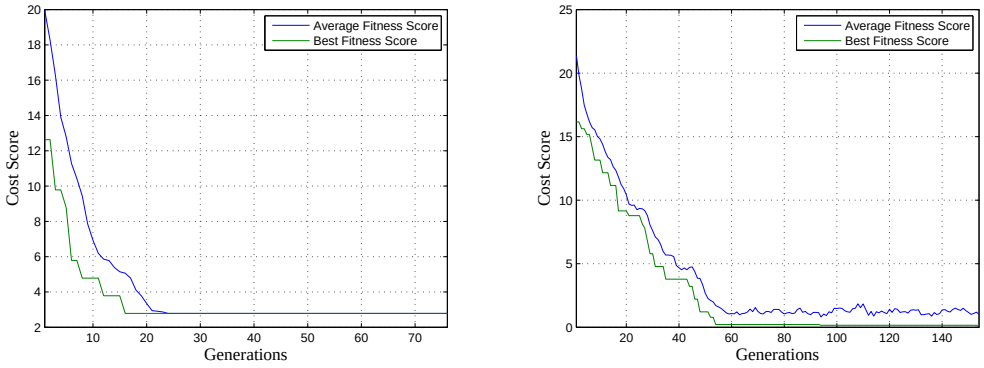
Figure 8: Effect of Population Size on Recovery Results. Curves indicate $p \geq 15$ results in rapid convergence with further increases in p resulting in reduction only of the cost function.

and the quality of the desired solution should be taken into account to determine the crossover fraction parameter in practice.



(a) f_c=0.2 (b) f_c=0.9

Figure 9: Effect of Crossover Fraction on Convergence Property of the GA, p=25. Low crossover rate is seen to improve the exploration of the solution space.



(a) With No Mutation

(b) With Only Mutation

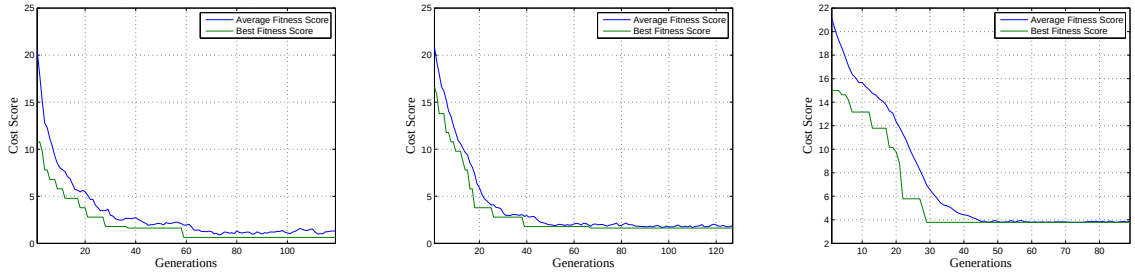
Figure 10: Effect of Mutation on Convergence Property of the GA. Use of mutation alone is seen to provide basis of comparison corresponding to random search.

Fig. 10 depicts the effect of the choice of the mutation operation on the soft-resilience search progression. As Fig. 10a reveals, a mere use of crossover without any mutation improves the fitness behavior of the population to some initial level. However, a local minimum solution is reached and no further improvement is observed beyond 20 generations. Further GA operations without mutation are seen to not improve the average nor best-performing objective score. On the other hand, using a mutation operation only as in Fig. 10b, the random changes applied by the algorithm exploit the diversity in solutions and hence a better solution is eventually realized, although after a larger number of generations than use of crossover and mutation together with suitable occurrence probabilities.

To analyze the effect of elitism, a fixed population size of 25 is used with a crossover fraction of 0.5 in Fig. 11. A lower number of elite count, such as 2, maintains the opportunity of realizing diverse individuals through the rest of the population. On the other hand, a very high number of elite count can result in slower progression towards convergence when poor average behavior occurs as those elite members become the dominating individuals and prevent more diverse exploration of the search space.

Fig. 12 shows the pareto set of solutions for the multi-objective evolution problem. The health metric degradation is specified in terms of degradation in measure of confidence on a normalized-to-maximum value scale. Similarly, the other objective cost to minimize, i.e., power consumption, is described on a normalized scale. For an example, if a throughput degradation of 40% is acceptable, it reduces power consumption to 30%. A further throughput degradation to an extent of 60% allows degraded operation at only 15% power consumption of the maximum power budget.

Thus, the measure of confidence results obtained with the SVM core demonstrate the applicability of *health metric based multi-objective online evolution* approach to realize self-recovery. The effect of GA parameters on the convergence properties of evolving hardware at runtime is also investigated. By carefully choosing a set of parameters, the designer can tradeoff various objective metrics such as power consumption, quality in terms of measure of confidence, throughput degradation of the SVM core during the recovery phase, latency



(a) n_E=2 (b) n_E=10 (c) n_E=20

Figure 11: Effect of Elite Count on Convergence Property of the GA, p=25, f_c=0.5. Elite count above two is not beneficial for convergence.

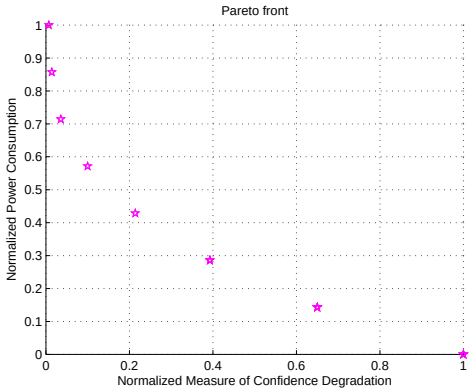


Figure 12: Pareto set of solutions for the SVM MOOE problem. Maintaining narrow degradation in classification confidence, e.g. 0.1 degradation, is seen to require roughly half the power budget.

of fault-recovery, and the reconfiguration controller overhead. Furthermore, these diverse objectives are achieved using a single cohesive strategy.

4.3 An Image/Video Processing Case-Study: Discrete Cosine Transform

Another case-study, DCT, is used to evaluate the *health metric based multi-objective online evolution* scheme to recover from hard-faults within the DCT core. In the hardware arrangement, PSNR is employed as a health metric to guide the architectural adaptations needed for fault-mitigation. It is demonstrated that PSNR based fault-detection and fault-recovery together with the proposed online multi-objective hardware evolution framework is a low-overhead technique to realize a fault-tolerant, self-healing, and low-power version of the DCT core.

To analyze the quality degradation of a faulty DCT core during the fault-handling process, the H.263 video encoder application is executed on the on-chip PowerPC processor of a Virtex-4 FPGA provided on a Xilinx ML-410 development board. The DCT module is

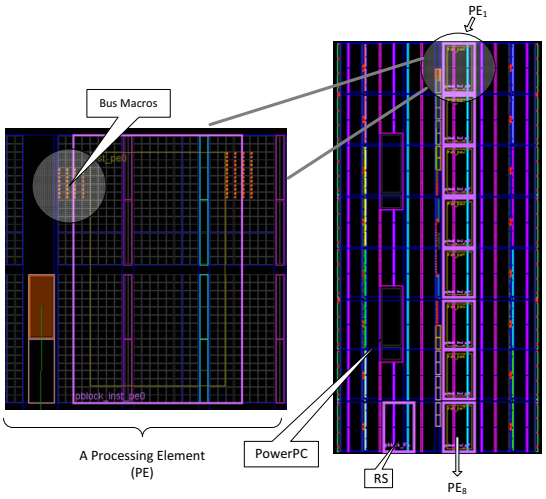
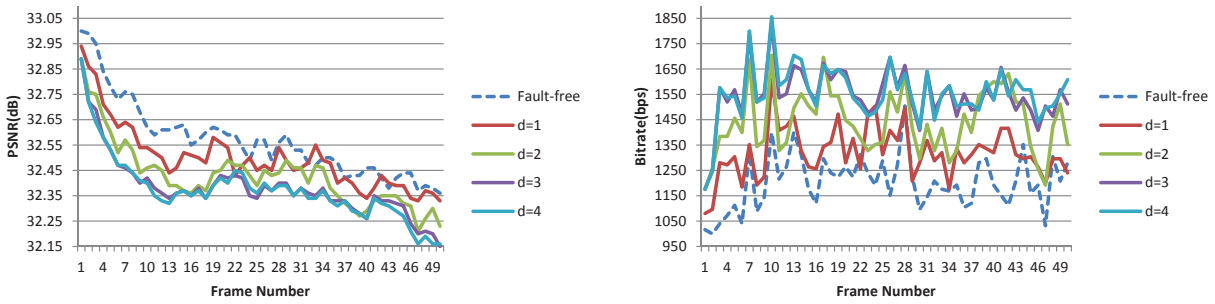


Figure 13: Floorplan of DCT module for Virtex-4 device

implemented in hardware. A 256MB memory module is used to hold the executable code (`.elf` file) of the video encoder as well as providing the data memory required to hold the images. Namely, the data from the first stage of the DCT is not overwritten, rather it is kept in its own span of the frame buffer. Xilinx PlanAhead is used for Partial Reconfiguration (PR) flow while the software and hardware system is built using Xilinx Platform Studio. Various Partial Reconfiguration Regions (PRRs) are defined where each PRR corresponds to a PE of the DCT core. The Xilinx Internal Configuration Access Port (ICAP) is used for downloading the partial bitstreams from external compact flash. The Xilinx System ACE is a controller to manage configuration data. It provides an interface between CompactFlash and the FPGA. This controller is connected in slave mode over the PLB bus and the embedded processor can read the bitstreams stored on the Compact Flash. The combined ACE file consisting of full system reconfiguration file (`.bit`) and the executable file (`.elf`) can be stored on Compact Flash. The FPGA chip is configured with the stored ACE file upon a power-ON event.

The floorplan of the DCT hardware is shown in Fig. 13. There are 9 reconfigurable PEs shown, each PE communicates to the static logic through the Bus Macros. The static modules of the design include PowerPC, DCT controller, Frame Buffer, Digital Clock Manager, DDR SDRAM controller, CompactFlash controller, and GPIO cores. The RS is reserved at design time to provide redundancy needed for fault-handling. Initially, the RS is configured with a blank bitstream. After fault-detection, iterative reconfiguration of the slack is performed to identify faulty PEs in the throughput datapath. If a faulty PE is identified in the datapath, the RS is configured with its functionality and introduced into the datapath thereby completing the recovery process.

In H.263 video encoders, PSNR is possibly maintained by fixing the QP while allowing the bitrate of the encoded bitstream output to vary. In an experiment to evaluate the quality-oriented health metrics, variable bitrate mode is selected for video encoder. In addition to scene's high activity, a failure in Motion Estimation (ME) processing due to



(a) PSNR for various number of defective PEs

(b) Bitrate for various number of defective PEs

Figure 14: Fault injection results for container.qcif video sequence

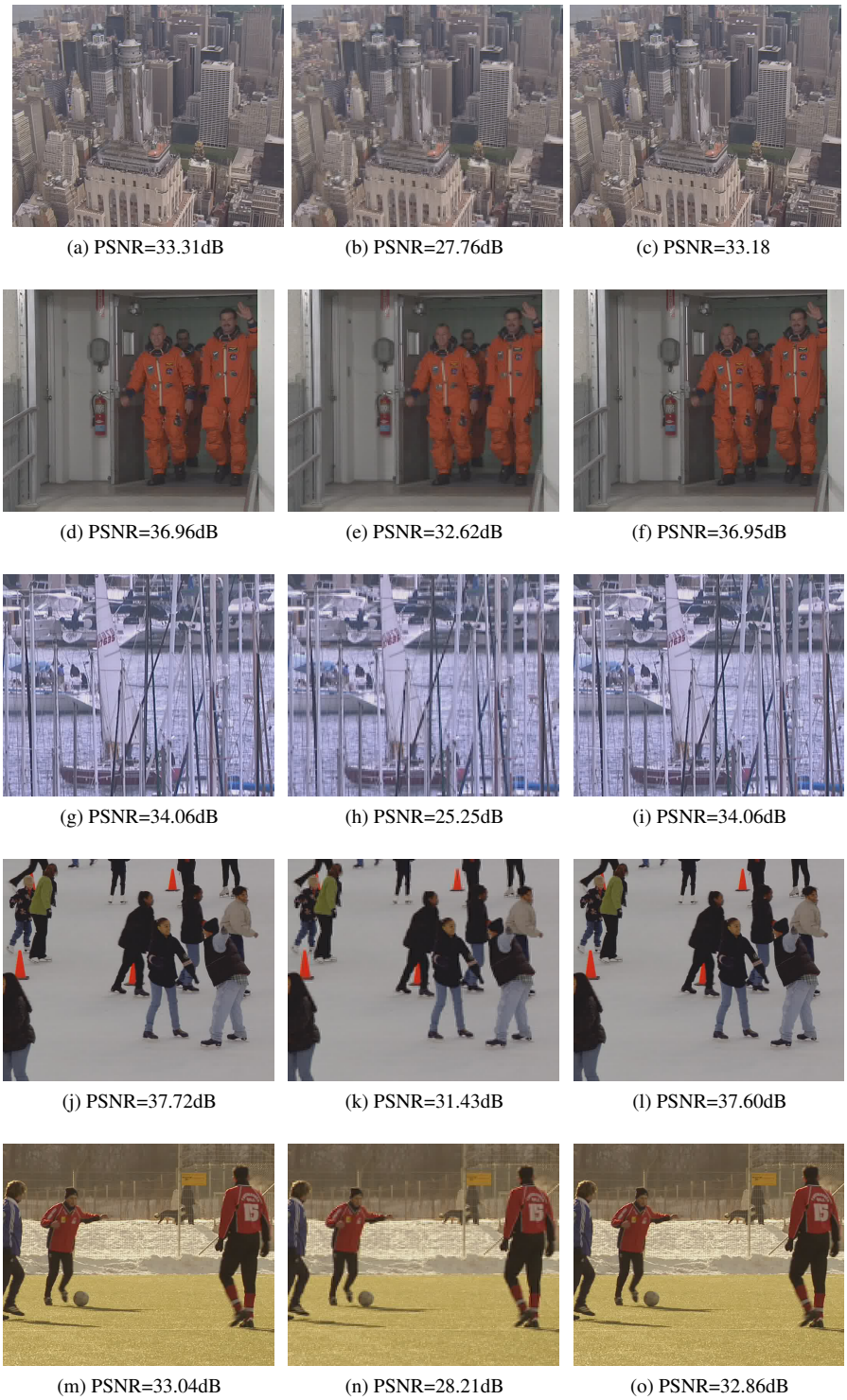
hardware faults can also be causal in increasing the bitrate of encoded video stream, thereby degrading overall compression efficiency. The PSNR and bitrate of encoded bitstream for the `container.qcif` input video sequence for various fault-scenarios are shown in Figure 14a and Figure 14b, respectively, in which d corresponds to number of faulty PEs. As it is evident, an increase in the number of defective PEs results in quality degradation of the compressed video stream in terms of both PSNR and Bitrate.

In order to demonstrate fault recovery capability of the proposed MOOE resource escalating approach, throughput degradation is described in terms of PSNR-degradation. To demonstrate the energy-saving capability of the proposed adaptive methodology, power consumption is reported as an evaluation metric. Fig. 15 shows the qualitative and quantitative results of fault-tolerant DCT module. In this evaluation scenario, no availability of a slack PE is considered, i.e, $N_s = 0$. Thus, fault recovery is realized by the successive re-mappings of DCT functions on the reconfigurable fabric. As the PSNR results show, the proposed soft-computing framework can realize near-healthy quality objective by architectural adaptations. For example, a PSNR of 32.86dB is achieved with the power consumption of 119mW after fault-recovery when the faulty-DCT provided a PSNR of 28.21dB at 142mW power consumption. This quality recovery is reasonably comparable to the fault-free DCT's output which was 33.04dB. The reduction in power consumption becomes feasible due to the feasibility of blanking least priority PE whose output was not a significant contributing factor in terms of the PSNR. In this way, the PSNR-based multi-objective online evolution explores the search space by the architectural re-mappings and their corresponding effect on output quality.

5 Comparison of Proposed Approach with Conventional Fault-Handling Techniques

5.1 Modular Redundancy

Comparing the proposed technique to the conventional approaches used in the fault-tolerance domain, there are several criteria of improvement. For example, TMR will require



(a) PSNR=33.31dB (b) PSNR=27.76dB (c) PSNR=33.18

(d) PSNR=36.96dB (e) PSNR=32.62dB (f) PSNR=36.95dB

(g) PSNR=34.06dB (h) PSNR=25.25dB (i) PSNR=34.06dB

(j) PSNR=37.72dB (k) PSNR=31.43dB (l) PSNR=37.60dB

(m) PSNR=33.04dB (n) PSNR=28.21dB (o) PSNR=32.86dB

Figure 15: Fault recovery results for various 4cif test video sequences [27] - left: images in frame buffer of 142mW healthy DCT, center: 142mW faulty DCT, right: post fault-recovery 119mW DCT

24 modules for 8x8 DCT computations and the fault capacity would be limited to errors in only one voting path. However, the proposed approach allows additional modules during normal operations, and can handle even the case when 6 out of 8 modules are faulty. Thus, compared to the TMR scheme, the area and power requirements are about one third, yet fault tolerance is improved. Moreover, fault-handling can be adjusted by the DSP circuit designer based upon the tradeoff desired between detection latency and the area overhead incurred. In addition to fault-capacity, TMR power consumption is significantly higher. On the other hand, the proposed health metric based multi-objective online evolution strategy achieves power and quality objective at uniplex area cost and significantly reduced power consumption especially for the majority portion of the mission lifetime which is fault-free.

5.2 BIST-based Evaluation

An exhaustive test vector strategy would require 2^{96} vectors (8 values of 12 bit precision) to exercise all the logic inside a module computing a DCT function, which is computationally intractable. However, the proposed scheme evaluates the modules when subjected to their actual inputs. Given the contained faulty resources do not interfere with the desired functionality, a PE can be continued to be deployed in the circuit. In the DCT core, each PE spans one Partial Reconfiguration Region (PRR) and each PRR consists of 1152 LUTs. In addition, there are other resources like FF, BRAM, and DSP48 blocks. In a BIST-based resource testing scheme [28], these resources need to be tested exhaustively, at all times even without occurrence of a failure. This affects throughput as well as power consumption. However, in the proposed approach, the fault isolation phase is initiated only after a fault is detected as significant. Here, the PRR is treated as a black box in terms of the contained resources to check its health. Thus, a health metric based multi-objective online evolution offers a promising soft-resilience technique which tackles *operationally significant* faults rather than *innocuous faults*. Meanwhile, it covers both quality and power optimization using the same cohesive strategy.

6 Conclusions and Future Directions

A throughput-driven runtime resource configuring scheme to realize soft-resiliency in self-repairing computational platforms for signal processing is presented. A health metric-based feedback method is used by the multi-objective online evolution to dynamically adapt the processing blocks to achieve the desired levels of power and quality. The scheme is validated by implementation on a commercial off-the-shelf Xilinx Virtex FPGA to validate the feasibility of a fault-tolerant and energy-efficient design. Moreover, the scheme is not dependent upon the technology model of a specific device. Nonetheless, a dynamic reconfiguration capability of the devices is essential to implement the proposed fault handling flow.

The fault coverage provided includes logic resources as well as routing resources as their performance is intrinsic to the observed quality metric. The malfunctioning of any of them will result in the utilizing PE to be flagged as faulty, and then its assigned function is moved to another area in the chip only if it is found to exhibit a sufficient operational priority on the output quality. This self-organizing hardware architecture maintains energy efficiency and quality under various operating conditions by sacrificing non-critical computations based on input signal characteristics and escalating critical tasks to healthy computational resources.

Overall, an autonomous soft-resilience approach can be advantageous to the tradeoffs of accuracy and energy efficiency. A multi-objective GA approach is promising in solving such large search space problems using the proposed guidance function along the pareto front. The proposed scheme performs well for a synthetic node case study as well as SVM and DCT computations. The recovery results demonstrate self-healing capability, as well as power efficient circuits with provision of the adaptive resource escalation approach. For example, the PSNR of a faulty DCT module is successfully recovered from 25.2dB to 34.1dB along with a power saving of 16.2%. An interesting future direction would be to develop a scheme for priority estimation at runtime for other applications where task priority information is not known a-priori.

References

[1] Hartmut Schmeck, Christian Muller-Schloer, Emre Cakar, Moez Mnif, and Urban Richter. Adaptivity and self-organisation in organic computing systems. In Christian Muller-Schloer, Hartmut Schmeck, and Theo Ungerer, editors, *Organic Computing: A Paradigm Shift for Complex Systems*, volume 1 of *Autonomic Systems*, pages 5–37. Springer Basel, 2011.

[2] R.A. Abdallah and N.R. Shanbhag. Minimum-energy operation via error resiliency. *Embedded Systems Letters, IEEE*, 2(4):115–118, Dec. 2010.

[3] R.G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge. Near-threshold computing: Reclaiming Moore's Law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, Feb. 2010.

[4] Debabrata Mohapatra, Georgios Karakonstantis, and Kaushik Roy. Significance driven computation: a voltage-scalable, variation-aware, quality-tuning motion estimator. In *14th ACM/IEEE international symposium on Low power electronics and design (ISLPED)*, pages 195–200, New York, NY, USA, 2009. ACM.

[5] Krishna V. Palem, Lakshmi N.B. Chakrapani, Zvi M. Kedem, Avinash Lingamneni, and Kirthi Krishna Muntimadugu. Sustaining moore's law in embedded computing through probabilistic and approximate design: retrospects and prospects. In *International conference on Compilers, architecture, and synthesis for embedded systems*, CASES, pages 1–10, New York, NY, USA, 2009. ACM.

[6] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, and C. Lucas. Bio-inspired imprecise computational blocks for efficient vlsi implementation of soft-computing applications. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(4):850–862, April 2010.

[7] W. Barker, D.M. Halliday, Y. Thoma, E. Sanchez, G. Tempesti, and A.M. Tyrrell. Fault tolerance using dynamic reconfiguration on the POEtic tissue. *Evolutionary Computation, IEEE Transactions on*, 11(5):666–684, Oct. 2007.

[8] D. Keymeulen, R. S. Zebulum, Y. Jin, and A. Stoica. Fault-tolerant evolvable hardware using field-programmable transistor arrays. *Reliability, IEEE Transactions on*, 49(3):305–316, 2000.

[9] Ronald F. DeMara, Kening Zhang, and Carthik A. Sharma. Autonomic fault-handling and refurbishment using throughput-driven assessment. *Applied Soft Computing*, 11:1588–1599, March 2011.

[10] Matthew G. Parris, Carthik A. Sharma, and Ronald F. DeMara. Progress in autonomous fault recovery of field programmable gate arrays. *ACM Comput. Surv.*, 43:31:1–31:30, October 2011.

[11] Rizwan A. Ashraf and Ronald F. DeMara. Scalable FPGA refurbishment using netlist-driven evolutionary algorithms. *IEEE Transactions on Computers*, 62(8):1526–1541, 2013.

[12] M.A. Makhzan, A. Eltawil, and F.J. Kurdahi. Architectural and algorithm level fault tolerant techniques for low power high yield multimedia devices. In *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, SAMOS*, pages 124–131, July 2008.

[13] N. Imran, J. Lee, and R. F. DeMara. Fault demotion using reconfigurable slack (FaDReS). *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(7):1364–1368, 2013.

[14] Subhasish Mitra, W.-J. Huang, N.R. Saxena, S.-Y. Yu, and E.J. McCluskey. Reconfigurable architecture for autonomous self-repair. *Design Test of Computers, IEEE*, 21(3):228–240, May-June 2004.

[15] Ricky W. Butler. A primer on architectural level fault tolerance. Tech. Report NASA/TM-2008-215108, The NASA STI Program Office, Feb. 2008.

[16] G.V. Varatkar and N.R. Shanbhag. Energy-efficient motion estimation using error-tolerance. In *International Symposium on Low Power Electronics and Design, ISLPED*, pages 113–118, Oct. 2006.

[17] E.P. Kim and N.R. Shanbhag. Soft N-Modular redundancy. *Computers, IEEE Transactions on*, 61(3):323–336, March 2012.

[18] C.A. Lisboa, L. Carro, C. Argyrides, and D.K. Pradhan. Algorithm level fault tolerance: A technique to cope with long duration transient faults in matrix multiplication algorithms. In *26th IEEE VLSI Test Symposium, VTS*, pages 363–370, May 2008.

[19] A. Maheshwari, W. Burleson, and R. Tessier. Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(3):299–311, March 2004.

[20] H. Singh, K. Agarwal, D Sylvester, and K.J. Nowka. Enhanced leakage reduction techniques using intermediate strength power gating. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(11):1215–1224, 2007.

[21] Bruno Zatt, Muhammad Shafique, Sergio Bampi, and Jörg Henkel. A low-power memory architecture with application-aware power management for motion & disparity estimation in multiview video coding. In *Proceedings of the International Conference on Computer-Aided Design*, ICCAD '11, pages 40–47, Piscataway, NJ, USA, 2011. IEEE Press.

[22] Shaoshan Liu, R.N. Pittman, A. Forin, and J.-L. Gaudiot. On energy efficiency of reconfigurable systems with run-time partial reconfiguration. In *Application-specific Systems Architectures and Processors (ASAP), 2010 21st IEEE International Conference on*, pages 265–272, July 2010.

[23] MATLAB. Genetic algorithm options. *MathWorks Documentation Center*, 2013.

[24] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[25] Dennis Decoste and Bernhard Scholkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3):161–190, 2002.

[26] Jock A. Blackard and Denis J. Dean. The forest covertype dataset. http://archive.ics.uci.edu/ml/machine-learning-databases/covtype/covtype.info.

[27] TNT. Video test sequences, institute for information processing, leibniz university of hannover. Retrieved on May 26, 2013 [Online] `ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/`.

[28] M. Abramovici, C.E. Stroud, and J.M. Emmert. Online BIST and BIST-based diagnosis of FPGA logic blocks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(12):1284–1294, 2004.