Amorphous Slack Methodology for Autonomous Fault-Handling in Reconfigurable Devices

Naveed Imran[1], Jooheung Lee*[2], Youngju Kim[2], Mingjie Lin[1] and Ronald F. DeMara[1]

[1]*Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL, 32816 USA*

[2]*Department of Electronic and Electrical Engineering, Hongik University, Korea*

naveed@knights.ucf.edu, mingjie@eecs.ucf.edu, demara@mail.ucf.edu, joolee@hongik.ac.kr, yjkim1@hongik.ac.kr

Abstract

Amorphous Slack fault handling methodology utilizes adaptive runtime redundancy to improve survivability of FPGA based designs. Unlike conventional static redundancy based methods to achieve fault resilience, the proposed system operates in uniplex arrangement under non-contingent conditions. The proposed fault isolation algorithm is invoked upon fault detection which employs a health metric of the application operating over reconfigurable platform. This assertion applies if a signal-to-noise metric is known, as well as applications that do not possess a readily correlated metric to identify anomalous behavior. In particular, readily available processor cores allow dynamic fault identification by executing a software specification of the signal processing algorithm which is used to periodically validate critical outputs of the high-speed hardware circuit within tolerances. The results from H.263 video encoder and Canny edge detector implemented over Xilinx Virtex-4 device demonstrate autonomous recovery from permanent stuck-at faults while maintaining the throughput during fault-handling operations. The fault-detection and isolation applications are executed on on-chip PowerPC processor while the Circuit-Under-Test (CUT) is realized in hardware fabric. The proposed architecture allows on-chip processor based functional monitoring of the contained hardware resources subjected to the actual inputs of the circuit.

Keywords: *Fault-handling, FPGAs, Survivable Architectures, Dynamic Partial Reconfiguration, Reliability, Availability, Hardware-software co-design*

1. Introduction

With the advent of 20nm CMOS device technology and the emergence of nanoscale devices, permanent faults and aging-induced degradation effects can become more prominent in both logic resources and interconnects [1-3]. This threat of diminished component reliability becomes more unpredictable due to escalating thermal profiles, process-level variability, and harsh environments such as deep-space [4], high-altitude flight, or other mission critical applications. Furthermore, chip density and complexity can make the prevention of all possible design faults infeasible.

Due to these challenges, error-resiliency and self-adaptability of future electronic systems are subjects of growing interest [5-7]. In particular, a DSP device is survivable if it can continue its operation in the presence of failures, perhaps in a degraded mode with partially restored functionality [8]. For DSP devices implemented with reconfigurable digital fabric, its survivability can be achieved in various ways. Offline testing methods rely on taking the DSP device out of operation, diagnosing the faulty resources and avoiding those resources in the configured design. However, this method is less practical for real-time systems with specific timing deadlines. On the other hand, online testing methods, such as online Built-in Self-Test (BIST) techniques typically involve pseudo-exhaustive input-space testing in order to identify faults, while functional testing methods check the fitness of the datapath functions as they are utilized [9]. Because reconfigurable hardware fabric has been widely used as a platform for modern DSP applications such as image/video coding, cryptographic algorithms, and speech processing [10-12], FPGA technology offers a suitable platform for researching survivable DSP architectures. A comprehensive overview of the metrics for fault tolerance is provided in [13].

Traditionally, survivable systems employ resolution phases such as Fault Detection, Fault Isolation, and Fault Recovery. For example, the Concurrent Error Detection (CED) setup, a popular redundancy based fault-detection method, either realizes two concurrent replicas of a design [14], or two diverse duplex datapaths to avoid common mode faults. Although with costs of area and power overhead, CED achieves very low fault detection latency. A Triple Modular Redundant (TMR) system, on the other hand, utilizes three instances of a datapath module, whose outputs become the input to a majority voter. In this way, a TMR system is able to mask its faults in the output if distinguishable faults occur within one of three modules. However, such approach incurs an increased area and power requirements 3-fold that of the uniplex configuration.

In our approach, we employ dynamic redundancy to isolate and recover from faults. An on-chip processor core in FPGA fabric is used to monitor the health of contained logic resources. Unlike conventional test vectors methods, the processor performs functional testing of the resources subjected to the actual inputs of the system. In addition, we demonstrate that the real time analysis of time varying characteristics of input data is beneficial in predicting the computational complexity and hence the required hardware resources. Hardware architecture with software flexibility is desirable to provide architectural support to deal with these time-varying computing workloads. Thus, the hardware resources saved by intelligent prediction of computational resources are used to provide the capability needed for the proposed fault isolation and recovery scheme. The simulation results of the fault-isolation scheme show that fault isolation can be improved by taking into account the input signal characteristics.

The proposed fault-resilient architecture is effectively demonstrated by implementing H.263 video encoder on Xilinx Virtex-4 FPGA. The Discrete Cosine Transform (DCT) block is implemented in hardware by utilizing various Processing Elements (PEs) to accelerate the performance. Such a distributed implementation is also useful in terms of improvement in fault-resiliency. By using the proposed isolation algorithm, faulty PEs can be avoided at runtime while some identified healthy PEs recover the functionality in a fault-scenario. The Peak Signal-to-Noise Ratio (PSNR) of video frames shows that considerable throughput is available even during fault-diagnosis while complete or partial recovery from hardware failures is demonstrated by the PSNR measure after fault-handling phase.

2. Related Work

Conventional approaches to achieve fault tolerance cannot effectively ensure DSP device or system survivability, for example to multiple cumulative failures, because they rely primarily on either redundant circuit techniques or conservative design such as guard-banding, conservative voltage scaling, and even radiation hardening to ensure correct operation and to increase error resilience. However, such passive or static techniques may be inadequate in either fault coverage or recovery time, especially when unexpected operating conditions occur or input characteristics vary. In fact, most previous studies assume a priori knowledge of defects [15,16] or stable error characteristics of input data, hence requiring the use of sophisticated error prediction, fault modeling and avoidance methods [17], or test vectors which may degrade signal processing throughput, and also not always be comprehensive.

For many years, redundancy-based techniques, such as the use of redundant hardware, data integrity checking, and data redundancy across multiple devices [18, 19, 20], have been employed to provide static fault detection, masking and isolation capability. Similar principles have also been applied within software domain, such as N-version programming [21, 22], error handling in the code [23], and time-out monitoring [24]. However, most of these redundancy techniques have significant area/performance penalties and often are labor intensive, therefore having restricted benefits. Moreover, redundancy-based techniques cannot effectively handle multiple simultaneous faults in triplicated modules, such as the industry-standard and design tool supported TMR approaches including Xilinx XTMR [25].

The goal of developing autonomously achieving hardware-efficient computing survivability for DSP devices is justified by several technical trends. First, emerging integrated CPU+FPGA hybrid platforms, such as the Extensible Processing Platform architecture from Xilinx [26], offer an unprecedented opportunity to explore for intrinsic amorphous hardware redundancy. For example, there is renewed interest recently among major chip makers to add gate programmability into general-purpose CPUs partly due to the drastic increase in transistor density [27, 28, 29, 30]. Meanwhile, major FPGA vendors, such as Xilinx and Altera, have started shipping large-capacity FPGA devices with high-performance embedded processor cores [31, 32, 33]. All these, we believe, not only provide a readily available platform to prototype our proposed architecture, but also demonstrate the potential of achieving autonomous diagnosis and self-recovery at the chip level without re-architecting existing computing devices. Second, due to the significant increase in modern CMOS device’s computing performance and logic capacity, research on evolvable hardware techniques has been intensified recently in order to adapt hardware to achieve fault tolerance [13]. Commonly, these methods rely on finding a configuration which meets fitness criteria under a given fault scenario in order to avoid faulty resources [14]. Finally, as more and more applications in multimedia, cryptography and evolutionary systems can benefit from dynamic reconfiguration [34], autonomous reconfiguration of computing devices becomes an important issue. With the appearance of the partial reconfiguration technology in recent years, mainly for FPGA technology, a series of frameworks for dynamic reconfiguration have been developed [7, 35, 36]. All these technical advances significantly enhance the feasibility for implementing resilient DSP devices.

Virtually all prior studies rely on one or more critical components, sometimes referred to as golden elements [37] that are required to be operational in order for the recovery strategy to operate. Many strategies that tend to excel with respect to sustainability characteristics often do so at the expense of increased overhead characteristics. Thus, strength of the proposed approach is that it does not insert additional golden elements, such as TMR voters or redundant gates, into the signal processing data throughput path. As a result, a failure in fault-handling circuitry only loses recovery capacity, not circuit throughput functionality.

While other redundancy based methods employ either static or dynamic redundancy for fault-detection/isolation purposes as shown by CED, TMR and NMR arrangements in Figure 1 and Figure 2, we develop an adaptive redundancy scheme. The proposed fault-isolation algorithm is generalized to employ N-Modular Redundancy at runtime to achieve desired reliability levels with constraints of area and power. The mathematical Mode operation in Figure 2 denotes majority value that is passed through the voter and becomes the main output of CUT.

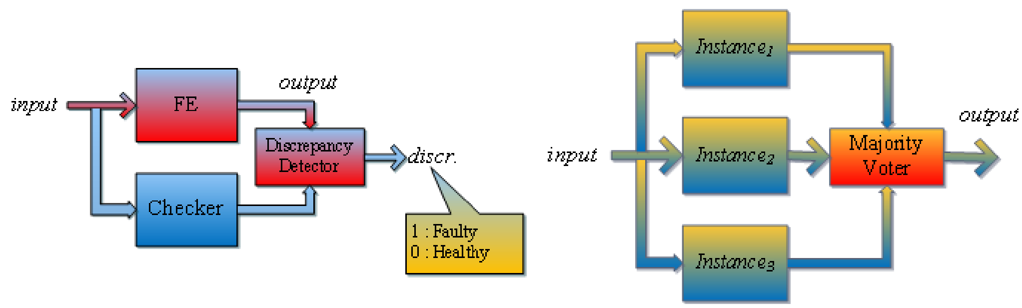


Figure 1. Conventional CED and TMR Methods of Utilizing Redundancy for Fault-tolerance

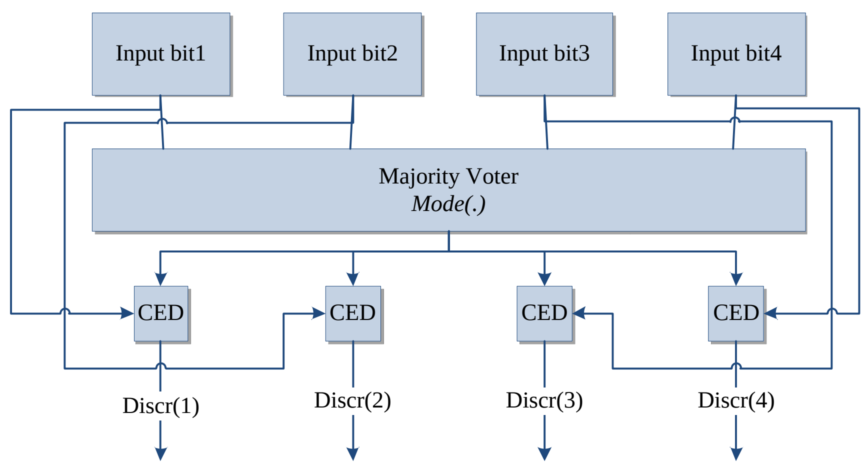


Figure 2. A Quadruple Modular Redundant (QMR) Arrangement

3. Amorphous Slack Approach

To achieve fault-handling operation, we propose an Amorphous Slack (AS) technique to time-multiplex the processing PRRs for different functions and compare their outputs with those from the active modules in the logic datapath. A discrepancy between the outputs of two modules results in them remaining in the Suspect pool, whereas the agreement marks them as Healthy after the evaluation window elapses. This diagnosis procedure runs concurrently with DSP processing, without decreasing signal processing throughput. Each processing slack can check multiple distinct functional blocks, therefore being area efficient, by leveraging the FPGA's inherent property of reconfiguration.

We consider a typical signal processing application which can be pipelined into multiple stages to accelerate the throughput. Consider a Functional Element (FE) which can be partitioned into multiple PEs. Some of the PEs operate as Reconfigurable Checker Elements (RCEs) for discrepancy checking purposes while others are kept in the throughput datapath for computation purposes. The total number of checker elements, designated as slack denoted by N_s, available for comparison purposes can be varied depending upon input signal characteristics, area margin, and power budget. These RCE can either be spares reserved at design-time, temporarily vacated PEs during runtime, or part of another FE performing some other task of lower priority. The term Reconfigurable Slack (RS) is used for the PEs corresponding to the first two cases. Algorithm 1 is used for fault isolation purpose in a core containing N PEs. Upon identifying faulty PEs, their functionality is assigned to healthy PEs which may either be slacks reserved at design time or some PEs computing lower priority-functions. In case of a DCT, the DC-coefficient computation function is more significant than AC-coefficients computing functions since the DC-coefficient contains the most content information about a natural image.

The proposed fault-handling scheme consists of the following phases: fault-detection in uniplex mode of operation by observing a health metric, identification of the faulty components by a novel fault-isolation algorithm and recovery from failures by exploiting the runtime reconfiguration capability of modern FPGAs. These phases are discussed in detail in the following subsections.

3.1. Fault Detection Mechanism

The software-based monitor running on FPGA's on-chip processor monitors the health of the CUT utilized resources by continually observing a health metric of the system. A hardware anomaly is manifested as degradation in the quality of system's output as verified by injecting Stuck-At faults in the hardware resources' models. The fault detection mechanism is illustrated by the flow chart in Figure . Initially, the Fitness State (FS) of all the PEs is healthy. However, the degradation of PSNR below a user defined threshold reveals possible faulty nature of the CUT resources. Such a detection event leaves all the PEs suspect and the hardware core needs further investigation as discussed in the fault isolation scheme further.

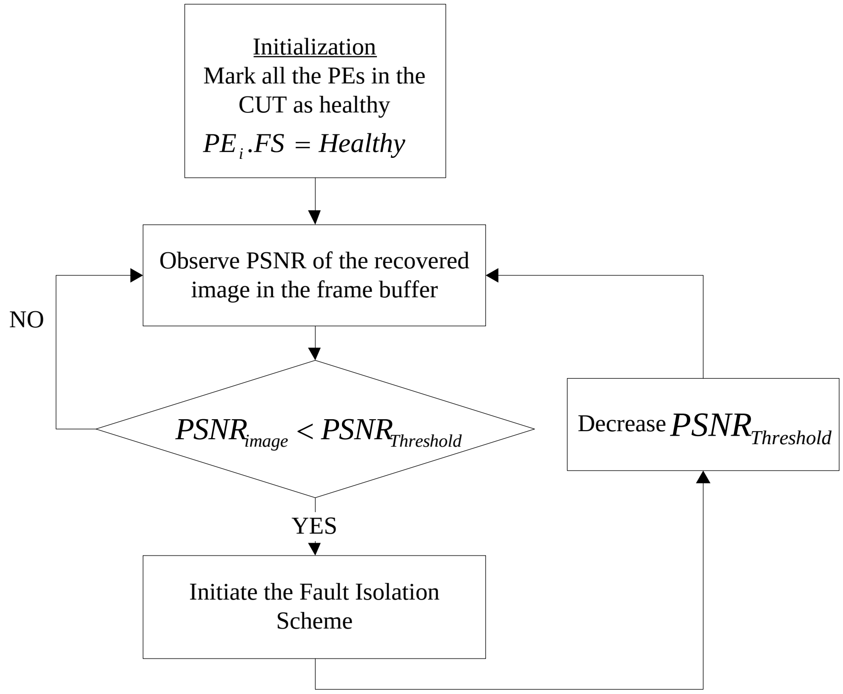


Figure 3. Fault Detection Mechanism

3.2. Fault Isolation Algorithm

For diagnosis purpose, concurrent comparison of various PEs' outputs is made which are selected from the overall pool based upon priority of the functions they implement. As shown in Figure , a discrepancy between the outputs of a CED pair reveals faulty nature of at least one of the PEs. On the other hand, a complete agreement in the output of a pair of PEs implementing a same given function over an evaluation window period is considered as presumed healthy nature of the PEs. Once a healthy PE has been found, another PE exhibiting a discrepancy in the CED setup is marked as faulty. Algorithm 1 is an identification scheme of finding healthy PEs in a pool of N PEs.

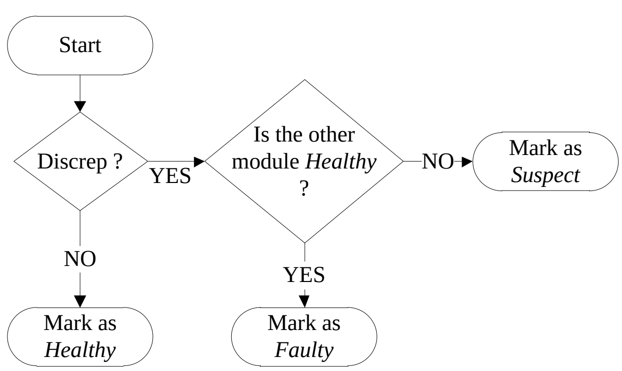


Figure 4. Fitness State Update based upon Discrepancy Information

Algorithm 1. Fault Isolation Algorithm Employing Amorphous Slacks

Input: N, N_s , Input signal characteristics, QP
Output: $\boldsymbol{\Phi}$
1: Initialize $\boldsymbol{\Phi}=[x\ x\ x\ \ldots\ x]^T$, $i=1$
2: **while** ($\{k|\ k\in\boldsymbol{\Phi},\ k=0\ \} = \phi$) **do**
3: Designate PE_s as checker(s) ; $(N+1)\leq s\leq(N+N_s)$
4: **while** ($i \leq N$) **do**
5: Reconfigure AS(s) with the same functionality as PE_i
6: Perform N-Modular Redundancy (NMR) majority voting to identify at least one healthy AS, $\Phi_i \leftarrow 0$ for PE_i which shows no discrepancy then go to step-11, $\Phi_i \leftarrow x$
otherwise
7: $i \leftarrow i+1$
8: **end while**
9: Move the AS by updating $N=N-N_s$, Re-initialize $i=1$
10: **end while**
11: Use a healthy AS to check all other PEs

The AS fault handling scheme identifies the faulty PE(s) by employing the RCE(s) as follows: Once fault is detected, the health of all the PEs in the processing datapath is suspected. Thus, step-1 of Algorithm 1 initially labels all PEs as Suspect. An entry $\Phi_i= 1$ in a vector $\boldsymbol{\Phi}$ of length ($N + N_s$) stands for faulty nature of the PE_i, $\Phi_i= 0$ for healthy PE_i, and $\Phi_i= x$ for suspected PE_i. The vector $\boldsymbol{\Phi}$ is used to maintain a record of proven healthy PEs. Initially, the set containing tested and verified fault-free healthy PEs is an empty set (ϕ) as labeled in step-2. The RCE can either be the blank PEs available in the system, some low-priority PEs, or PEs temporarily decommissioned from another FE. Initially, the RCE (or multiple RCEs) is reconfigured with the same functionality as that of the most important functional PE, for example, the module for computing DC-coefficient (step-3 and step-5). The location of a faulty PE is detected by performing the discrepancy check in an NMR arrangement (step-6). In case of a Dual Modular Redundancy (DMR) arrangement, a faulty status of one of the two modules, and a faulty status of more than N-2 modules in case of an NMR arrangement result into Suspect state of every instance. Therefore, we proceed to reconfigure the RCE with the second priority function and so on (step-3). Once an agreement between two modules over a complete evaluation window is observed, the two modules are declared as Healthy and their fitness state is updated (step-6). The identification of a healthy RCE implies that we do not need to reconfigure the PEs as checkers further. A healthy RCE can be used to check the fitness of all the modules (step-11). The discrepancy of a suspected module in pair with a healthy module reveals its Faulty nature. On the other hand, an observed discrepancy between suspected modules does not provide any information and keeps them marked Suspect. If a Healthy RCE is not identified in the first iteration even after reconfiguring with all of the functions in the datapath, it is moved to the next PE, and so on (step-9). Upon the completion of fault isolation, the priority functions are moved to the Healthy PEs, achieving recovery.

4. Experiment Setup-1: H.263 Video Encoder

To demonstrate the validity of the proposed fault handling approach, we execute H.263 video encoder's application on the Xilinx on-chip processor PowerPC while implementing the Discrete Cosine Transform (DCT) core in hardware. For this purpose, the DCT module is described in Verilog Hardware Description Language (HDL). The design is synthesized and implemented in Xilinx Integrated Software Environment (ISE) development environment with target device Virtex-4 FPGA. The detail of some of the hardware modules is given in the following:

4.1. The PowerPC 405 Processor

The PowerPC 405 processor is an optimized 32-bit implementation of the PowerPC 64-bit architecture. This on-chip processor block is optimized for embedded applications. The PowerPC 405 implements a 5-stage pipeline consisting of fetch, decode, execute, write-back, and load-write-back stages. Its memory management and cache management schemes are optimized for embedded software environments and performance in numerically intensive applications [38].

We used Xilinx Embedded Development Kit (EDK) which is a software suite of design tools to build processor-based embedded system for implementation in Xilinx FPGA. Xilinx Platform Studio (XPS) 9.2i provides the necessary interface to connect the essential peripherals to the processor block. Through this interface, the processor block and peripherals are available as embedded processing Intellectual Property (IP) cores. EDK invokes the utilities from ISE to synthesize and implement the processor-based hardware system over FPGA target [39]. Although, XPS environment can be used for the complete process starting from creating an entire design to all the way to generating and downloading the bitstream, we used ISE as the main platform for managing the overall system. The reason is that our video encoder system involves Partial Reconfiguration (PR) necessitating a PR tool to be integrated in the design flow as we will discuss further in the following.

4.2. Double Data Rate (DDR) Memory

We used Xilinx Embedded development board ML410 for evaluating the proposed FPGA design. This board has a 265 MB DDR memory interfaced through a standard 240-pin Dual Inline Memory Module (DIMM) socket [40]. The clock signal as a single differential pair is broadcast from the FPGA logic. A clock feedback signal is also used to resolve clock-skew issues by the Xilinx DCM IP. Xilinx Multi-Port Memory Controller (MPMC) core provides processor access to DDR2 via the Instruction and Data Processor Local Buses (IPLB and DPLB) as shown in Figure . The primary purpose of the DDR2RAM in this project is to store code and data as their storage size requirement is beyond the capacity of the Virtex-4 FPGA's on-chip memory [42].

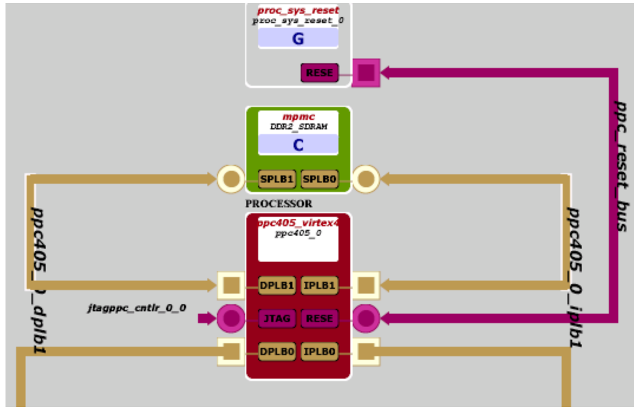


Figure 5. Processor Block Interfaced to DDR-RAM through MPMC Controller in XPS

4.3. Peripherals in the processor-based system

Other peripherals including DCT controller, DCT PEs, UART serial port, compact flash, and General Purpose Input-Output (GPIO) cores are interfaced to the processor block via PLB as shown in Figure 6. The image data is written by the processor to the frame buffer, and then upon completion of the DCT operation on a row of pixels, it is read back from the frame buffer to the PowerPC as shown in Figure 7. The data from first stage of the 2-D DCT, i.e., after 1-D DCT operation on 8 pixels, is used by the processor to diagnose the DCT core. For example, the output from an active PE and three RS's configured with the same function is used to compute the majority value. A discrepancy in output of a PE from the majority output value reveals the faulty nature of the PE. As shown in Figure 8, the output from PE_1 which is in active throughput datapath, is compared with the output from PE_2, PE_3, and PE_4. These checker PEs or RS's are configured at runtime with same functionality and provide dynamic redundancy for diagnosis purposes.

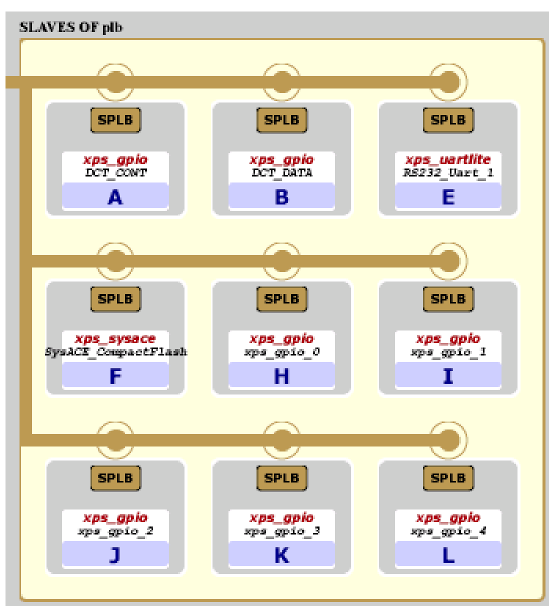


Figure 6. Peripherals Interfaced through PLB in XPS

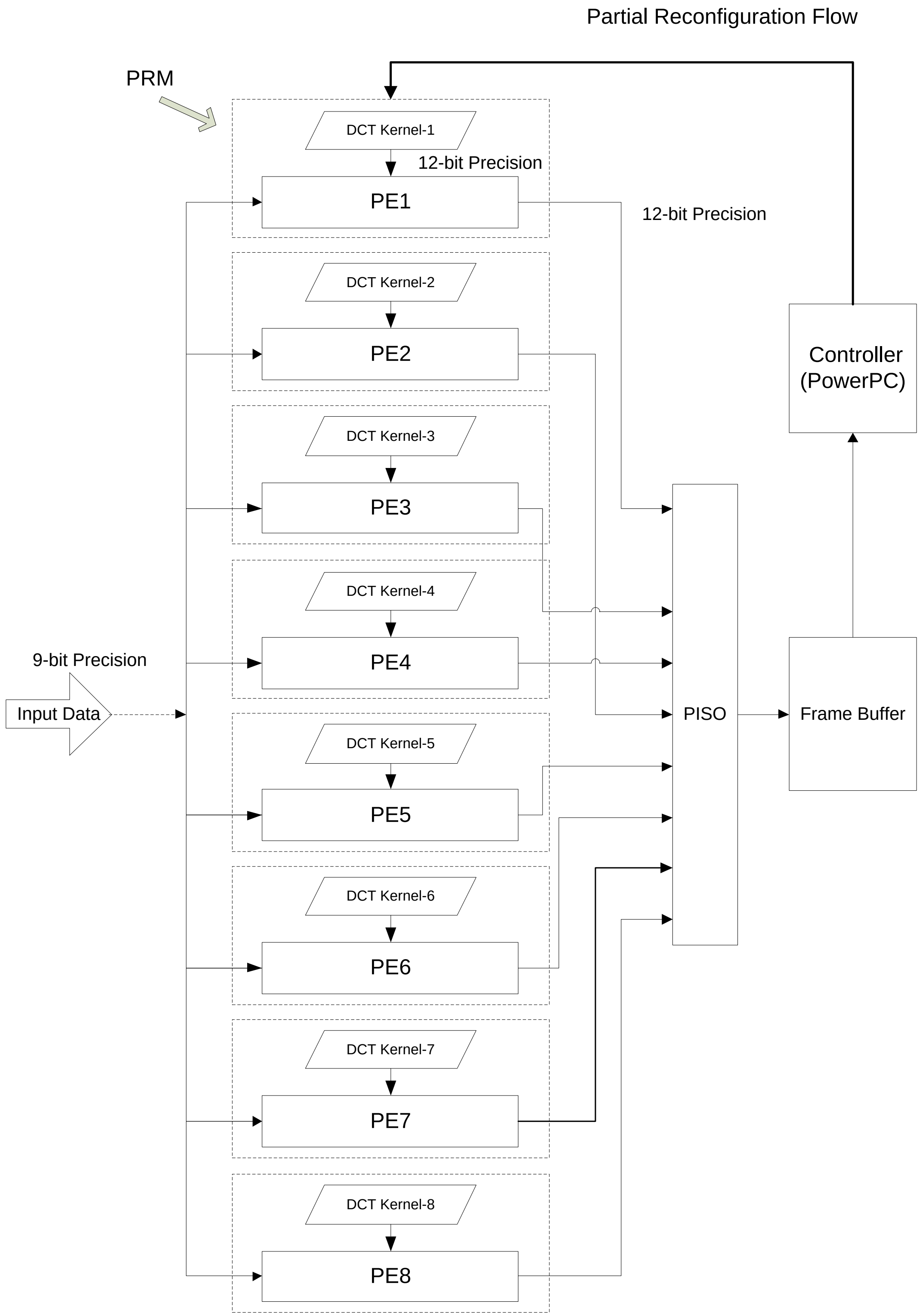


Figure 7. DCT Core Consisting of 8 PEs Interfaced with Processor Block

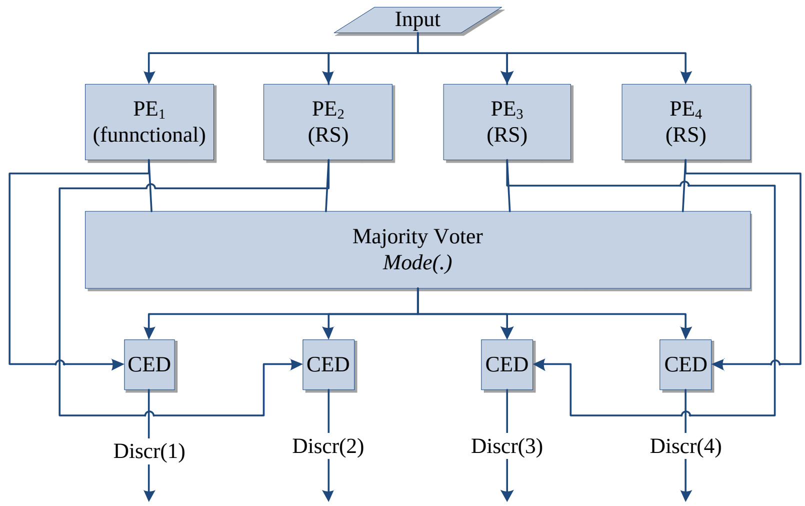


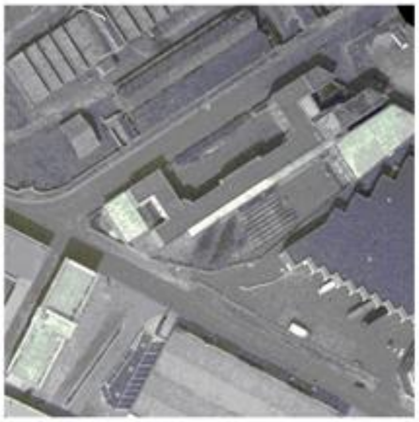
Figure 8. QMR Majority Voting Employing Three RS's

5. Experiment Setup-2: Canny Edge Detector

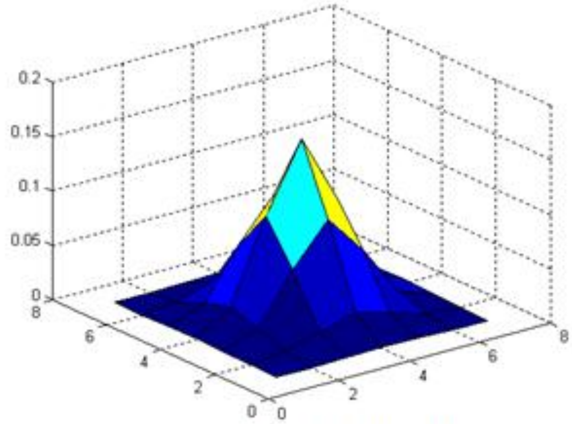
The survivability of edge detecting applications is desirable in harsh operating environments and long term missions [43]. A Canny edge detector [44-45] is characterized by its enhanced edge detection capability. Therefore, we evaluate the behavior of hardware faults in a Canny edge detection module. As shown in Figure 9(a), a 7×7 Gaussian Kernel is used in smoothing phase of the edge detector. We employed a distributed architecture where the convolution operation is performed by multiple PEs to accelerate the performance of the edge detection. Figure 9 illustrates the qualitative result of fault-handling for an image in the dataset available online [46].

6. Conclusions

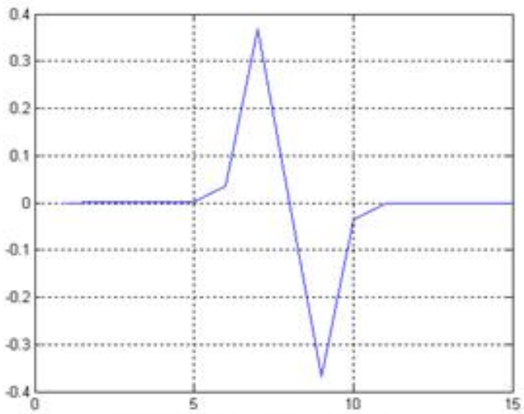
We proposed a novel dynamic fault-handling approach to autonomously achieve high survivability for the DSP circuits widely used in communications and cyber-physical systems. The benefits of Amorphous Slack methodology include 1) Multi-Resilience: signal processing throughput, as well as intrinsic graceful degradation, can be sustained even after multiple cumulative failures; 2) Model-Free Coverage: robustness is not contingent on a priori knowledge of hardware defects or input characteristics, thus avoiding error prediction challenges of emerging device technologies; 3) Autonomy: explicit fault isolation is avoided by adapting faulty modules within their own environment; 4) Compartmentalized Throughput and Recovery: the throughput datapath operates even if the fault-handling mechanism fails because golden elements such voters are not inserted into the throughput datapath; 5) Time and Area Efficiency: a single uniplex instance of the datapath achieves high throughput, while a governing health metric or flexible software detection scheme covers multiple modules using idle CPU cores. The proposed scheme could be broadened to operate in the absence of some uniplex health metric, such as PSNR, by instead using the processor cores during their idle times on the target platform.



(a). Original image

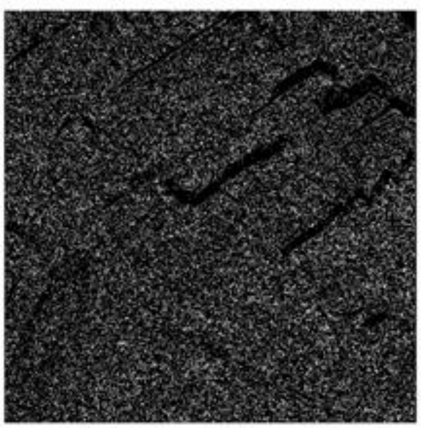


(b). Gaussian Kernel



(c). Gaussian Derivative

(d). Image from a faulty detector



(e). Image from a faulty detector

(f). Image from detector after fault-handling

Figure 9. Qualitative Results of the Survivable Edge Detector

References

[1] M. Berg, C. Poivey, D. Petrick, D. Espinosa, A. Lesea, K. A. LaBel, M. Friendlich, H. Kim and A. Phan, "Effectiveness of Internal Versus External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis", Nuclear Science, IEEE Transactions on, vol. 55, **(2008)**, pp. 2259-2266.
[2] M. Agarwal, B. Paul, M. Zhang and S. Mitra, "Circuit failure prediction and its application to transistor aging", in VLSI Test Symposium, 25th IEEE, **(2007)** May, pp. 277–286.
[3] W. Rao, C. Yang, R. Karri and A. Orailoglu, "Toward future systems with nanoscale devices: Overcoming the reliability challenge", Computer, vol. 44, no. 2, **(2011)** February, pp. 46 –53.
[4] A. Stoica, D. Keymeulen, R. Zebulum, M. Mojarradi, S. Katkoori and T. Daud, "Adaptive and evolvable analog electronics for space applications", in Proceedings of the 7th international conference on Evolvable systems: from biology to hardware, ser. ICES'07. Berlin, Heidelberg: Springer-Verlag, **(2007)**, pp. 379–390.
[5] R. Hyman Jr., K. Bhattacharya and N. Ranganathan, "Redundancy mining for soft error detection in multicore processors", Computers, IEEE Transactions on, vol. 60, no. 8, **(2011)** August, pp. 1114 –1125.
[6] SPP1500, "Dependable embedded systems", http://spp1500.itec.kit.edu/63.php, **(2012)** January 8.
[7] K. Paulsson, M. Hubner and J. Becker, "Strategies to on-line failure recovery in self-adaptive systems based on dynamic and partial reconfiguration", in Adaptive Hardware and Systems (AHS 2006), **(2006)** June, pp. 288–291.
[8] J. -C. Laprie, "Dependable computing and fault tolerance: Concepts and terminology", in Fault-Tolerant Computing, International Symposium on, **(1995)** June.
[9] M. Gericota, G. Alves, M. Silva and J. Ferreira, "Reliability and availability in reconfigurable computing: A basis for a common solution", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 16, no. 11, **(2008)** November, pp. 1545 –1558.
[10] H. Flatt, H. Blume and P. Pirsch, "Mapping of a real-time object detection application onto a configurable RISC/Coprocessor architecture at full HD resolution", in Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on, Quintana Roo, **(2010)** December, pp. 452 –457.
[11] G. Varatkar and N. Shanbhag, "Error-resilient motion estimation architecture", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 16, no. 10, **(2008)** October, pp. 1399 –1412.
[12] P. Fernando, S. Katkoori, D. Keymeulen, R. Zebulum and A. Stoica, "Customizable FPGA IP core implementation of a general-purpose genetic algorithm engine", Evolutionary Computation, IEEE Transactions on, vol. 14, no. 1, **(2010)** February, pp. 133-149.
[13] M. G. Parris, C. A. Sharma and R. F. DeMara, "Progress in autonomous fault recovery of field programmable gate arrays", ACM Comput. Surv., vol. 43, **(2011)** October, pp. 31:1–31:30.
[14] R. F. DeMara, K. Zhang and C. A. Sharma, "Autonomic fault-handling and refurbishment using throughput-driven assessment", Appl. Soft Comput., vol. 11, **(2011)** March, pp. 1588–1599.
[15] Y. Li, Y. M. Kim, E. Mintarno, D. S. Gardner and S. Mitra, "Overcoming early-life failure and aging for robust systems", IEEE Design Test of Computers, vol. 26, no. 6, **(2009)**, pp. 28-39.
[16] M. Lin, Y. Bai and J. Wawrzynek, "Discriminatively fortified computing with reconfigurable digital fabric", In Proceedings of the 13th IEEE International High Assurance Systems Engineering Symposium, **(2011)**, pp. 65–74.
[17] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri and V. Piuri, "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard", IEEE Transactions on Computers, vol. 52, no. 4, **(2003)**, pp. 492-505.
[18] P. K. Lala, "Fault tolerance and self-checking techniques in microprocessor-based system design", Software Microsystems, vol. 4, no. 3, **(1985)**, pp. 50 –52.
[19] E. Rotenberg, "AR-SMT: a microarchitectural approach to fault tolerance in microprocessors", Fault-Tolerant Computing, 1999, Digest of Papers, Twenty-Ninth Annual International Symposium on, **(1999)** June 15-18, pp. 84-91.
[20] M. A. Gomaa, C. Scarbrough, T. N. Vijaykumar and I. Pomeranz, "Transient-fault recovery for chip multiprocessors", Micro, IEEE, vol. 23, no. 6, **(2003)**, pp. 76-83.
[21] J. Gray and D. P. Siewiorek, "High-availability computer systems", Computer, vol. 24, no. 9, **(1991)** September, pp. 39-48.
[22] G. K. Saha, "Software based fault tolerance: a survey", Ubiquity, vol. 2006, **(2006)** July, pp. 1:1–1:1.

[23] J. Xie and D. Li, "Parallel error-trapping decoding cyclic burst error correcting codes", Communications, Circuits and Systems, 2009. ICCCAS 2009, International Conference on, **(2009)** July 23-25, pp.354-356.
[24] A. K. Somani and N. H. Vaidya, “Understanding fault tolerance and reliability”, Computer, vol. 30, no. 4, **(1997)** April, pp. 45 –50.
[25] C. Carmichael, “Triple module redundancy design techniques for Virtex FPGAs”, Xilinx Application Note: Virtex Series, **(2006)**.
[26] Xlinx, “Extensible Processing Platform”, http://www.xilinx.com/innovation/extensible-processing-platform.htm, **(2011)** November.
[27] D. Bouldin, “Enhancing electronic systems with reconfigurable hardware”, Circuits and Devices Magazine, IEEE, vol. 22, no. 3, **(2006)** May-June, pp. 32-36.
[28] E. Stott, J. S. J. Wong and P. Y. K. Cheung, “Degradation analysis and mitigation in FPGAs”, In Field Programmable Logic and Applications (FPL), International Conference on, **(2010)**, pp. 428–433.
[29] G. Gibeling, “GateLib: A Library for Hardware and Software Research”, UC Berkeley, **(2009)** June.
[30] B. Pratt, M. Caffrey, P. Graham, K. Morgan and M. Wirthlin, “Improving FPGA design robustness with partial TMR”, In Reliability Physics Symposium Proceedings, 44th Annual., IEEE International, **(2006)**, pp. 226 –232.
[31] Xilinx Unveils High-Performance ARM-based CPU-FPGA Hybrid Platform, http://www.bdti.com/InsideDSP/2010/05/20/Xilinx.
[32] J. R. Heath, P. J. Kuekes, G. Snider and S. R. Williams, “A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology”, Science, vol. 280, no. 5370, **(1998)** June, pp. 1716–1721.
[33] R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B. C. Lee, S. Richardson, C. Kozyrakis and M. Horowitz, “Understanding sources of inefficiency in general-purpose chips”, SIGARCH Comput. Archit. News, vol. 38, **(2010)** June, pp. 37–47.
[34] J. Kramer and J. Magee, “Dynamic configuration for distributed systems”, IEEE Trans. Softw. Eng., vol. 11, **(1985)** April, pp. 424–436.
[35] H. Tan and R. F. DeMara, “A multilayer framework supporting autonomous run-time partial reconfiguration”, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 16, no. 5, **(2008)**, pp. 504–516.
[36] J. Huang and J. Lee, "Reconfigurable Architecture for ZQDCT using Computational Complexity Prediction and Bitstream Relocation", Embedded Systems Letters, IEEE, **(2011)**.
[37] M. Garvie and A. Thompson, “Scrubbing away transients and jiggling around the permanent: long survival of FPGA systems through evolutionary self-repair”, In On-Line Testing Symposium, IOLTS 10th IEEE International, **(2004)** July, pp. 155 – 160.
[38] Xilinx, “PowerPC 405 processor block reference guide (ug018)”, Xilinx Online Documents, UG018 (v2.4), http://www.xilinx.com/support/documentation/user_guides/ug018.pdf, **(2010)** January 11.
[39] Xilinx, “Embedded system tools reference manual”, Xilinx Online Documents, EDK 10.1, Service Pack 3, http://www.xilinx.com/support/documentation/sw_manuals/edk10_est_rm.pdf, **(2008)**.
[40] Xilinx, “ML410 Embedded Development Platform,” Xilinx Online Documents, UG085 (v1.7.2), http://www.xilinx.com/products/boards/ml410/index.html, **(2008)** December 11.
[41] Xilinx, “PlanAhead 10.1 user guide”, Xilinx Online Documents, http://www.xilinx.com/support/documentation/sw_manuals/PlanAhead10-1_UserGuide.pdf**,** **(2008)** January 18.
[42] Xilinx, “Virtex-4 FPGA configuration user guide (ug071)”, Xilinx Online Documents, UG071 (v1.11), http://www.xilinx.com/support/documentation/user_guides/ug071.pdf, **(2009)** June 9.
[43] R. F. DeMara, J. Lee, R. Al-Haddad, R. S. Oreifej, R. Ashraf, B. Stensrud and M. Quist, “Dynamic partial reconfiguration approach to the design of sustainable edge detectors”, ERSA'10, **(2010)**, pp. 49–58.
[44] J. Canny, "A Computational Approach to Edge Detection", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI-8, no. 6, **(1986)** November, pp. 679-698.
[45] T. Kim, H. Adeli, C. Ramos and B.-H. Kang, “Signal Processing, Image Processing, and Pattern Recognition,” ser. Springer-Verlag, Springer, **(2011)**.
[46] VGG, “Oxford visual geometry group (vgg)’s images dataset: Aerial views”, http://www.robots.ox.ac.uk/vgg/data/, **(2012)** February 13.

Authors

Naveed Imran received the M.S. degree in Electrical Engineering from the Univeristy of Central Florida (UCF), Orlando, FL in 2010. Currently, he is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at the UCF. His research interests include FPGA-based embedded systems, design for reliability, computer vision, reconfigurable hardware for image/video applications, and signal processing. He is a student member of IEEE.

Jooheung Lee has been working on various topics in the areas of multimedia signal processing algorithms and low power VLSI systems design. His research interests include image and video coding algorithms, multimedia systems, power aware and reliable VLSI systems design, and reconfigurable computing for signal processing applications. Previously, he worked at the Wireless Multimedia Communications Laboratory at the R&D Complex of LG Electronics in 1998, where he worked on low power video codec ASIC design for mobile applications. After completing his Ph.D. at the Pennsylvania State University in 2006, he joined the Department of Electrical Engineering and Computer Science at the University of Central Florida, Orlando, Florida, USA, where he was a full-time faculty member. Currently, he is an Assistant Professor of the Department of Electronic and Electrical Engineering at Hongik University, Republic of Korea.

Youngju Kim received the B.S. and M.S. degrees in Electrical Engineering from the Seoul National University, Korea in 1980 and 1985, respectively and the Ph.D. degree in Electrical Engineering from the Polytechnic University of New York, USA, in 1995. In 1996, he joined the Hongik University, Republic of Korea, where he is now an Associate Professor. His recent research interests include the RF circuit design and LIN wireless network and plasma engineering.

Mingjie Lin joined UCF as an assistant professor of Electrical Engineering and Computer Sciences in Spring 2011. From 2008 to 2009, he worked at an FPGA startup---Tabula Inc. for one year as a senior engineer. At the beginning of 2009, he returned to academia and worked as a post-doctoral scholar at EECS of UC Berkeley for one year.

Mingjie's previous research involves VLSI reconfigurable array architecture, bio-inspired/neuromorphic arrays, and monolithically stacked 3D-IC. His current research focuses on exploring novel ways to construct scalable computing machine with high performance and low power consumption. To this end, his research activities spanned across Computer Architecture/Compiler, Reconfigurable Computing, Integrated Circuit, and System Design.

Ronald F. DeMara received the Bachelor of Science degree in Electrical Engineering with High Honors from Lehigh University in 1987, the Master of Science degree in Electrical Engineering from the University of Maryland, College Park in 1989, and 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. He is a Professor in the Department of Electrical Engineering and Computer Science and also Coordinator of Graduate Programs in Electrical and Computer Engineering. He was previously an Associate Engineer at IBM Federal and Complex Systems Division and has been a Visiting Research Scientist at NASA Ames Research Center.

Dr. DeMara's research interests are in Computer Architecture with emphasis on Evolvable Hardware and Distributed Architectures for Intelligent Systems. He has published approximately 120 articles on these topics and holds one patent. His research has been sponsored by the National Science Foundation; NASA; 7 different branches of the U.S. Army, Navy, Air Force, DARPA, and Department of Defense; National Security Agency; Harris Computer Systems; Lockheed Martin Information Systems; Theseus Logic Incorporated, and others. Research funding of $5M as PI or co-PI has led to the completion as advisor of 30 students with thesis or dissertation. At UCF, he has taught 13 different courses including 6 new courses he introduced into the curriculum. He has also served as ECE Department Associate Chair, Computer Engineering Program Coordinator, and Assessment Coordinator. He is a Senior Member of IEEE and a Member of ACM, and ASEE. He has been active in conference program committees, chaired focus sessions and technical panels, held officer positions in the Southeast Section of ASEE, and served on the UCF Faculty Senate and Graduate Council. He has been a technical referee for over 25 different journal and conference venues, and numerous proposals. He has served on the Editorial Boards of IEEE Transactions on VLSI Systems, ACM Transactions on Embedded Systems, Journal of Circuits, Systems, and Computers, and the journal Microprocessors and Microsystems. In 2008, he received the Outstanding Engineering Educator Award in the Southeastern United States from IEEE.