**The 1st International Conference on Advanced Signal Processing 2012**

# Area-Efficient Fault-Handling for Survivable Signal-Processing Architectures

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*
*naveed@knights.ucf.edu, mingjie@eecs.ucf.edu, demara@mail.ucf.edu*
*[2]Department of Electronic and Electrical Engineering, Hongik University, Korea*
*joolee@hongik.ac.kr, yjkim1@hongik.ac.kr*

*PRESENTER:*
Jooheung Lee

# CONTENTS

- Introduction
- Related Work
- The Proposed Approach
- Fault Isolation Algorithm
- Case Study-1: Video Encoder
- Case Study-2: Edge Detection
- Conclusions

# INTRODUCTION

- Permanent failure and aging effects in both logic and interconnect resources with the advent of 20nm CMOS device technology and the emergence of nano-scale devices [1][2][3]

- Unpredictable component-reliability due to escalating thermal profiles, process-level variability, and harsh DSP environments such as deep-space [4] and high-altitude flights

- Due to these challenges, error-resiliency and self-adaptability of future electronic systems are subjects of growing interest [5-7]

- Survivable of operation in the presence of failures, perhaps in a degraded mode with partially restored functionality [8]

# RELATED WORK

Offline testing method:

- By taking the circuit out of operation, diagnosing the faulty resources and avoiding those resources in the configured design
- Less practical for real-time systems with specific timing deadlines

Online testing methods

- Online Built-in Self-Test (BIST) techniques typically involve pseudo-exhaustive input-space testing in order to identify faults
- Functional testing methods check the fitness of the datapath functions as they are utilized [9]

Redundancy-based techniques

- Concurrent Error Detection (CED) arrangement
- Triple Modular Redundant(TMR) arrangement -> Fault masking in the system output via a majority voter

# FPGA AS A PLATFORM

- 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]
- FPGAs in mission-critical space applications
- FPGA technology offers a suitable platform for researching fault-tolerant architectures [13]
- A design can be recovered from faulty logic-resources by reconfiguring it to a fault-free area at runtime
- Dynamic Partial Reconfiguration capability of FPGAs

# OUR APPROACH

- Employ dynamic redundancy to isolate and recover from faults
- 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
- 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

# AMORPHOUS SLACK (AS) FAULT-HANDLING METHODOLOGY

- Time-multiplex the processing PRRs for different functions and compare their outputs with those from the active modules in the logic datapath
- Each processing slack can check multiple distinct functional blocks, therefore being area efficient, by leveraging the FPGA's inherent property of reconfiguration
- 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 slacks for comparison purposes can be varied depending upon input signal characteristics, area margin, and power budget

# FAULT-ISOLATION ALGORITHM

**Require:** N, $N_s$
**Ensure:** $\Phi$
1: Initialize $\Phi=[x\ x\ x\ \ldots\ x]^T$, $i=1$
2: **while** $(\{k|\ k \in \Phi,\ k=0\ \} = \phi)$ **do**
3: Designate $PE_s$ as checker(s) ; $(N+1) \leq s \leq (N+N_s)$
4: **while** $(i \leq N)$ **do**
5: Reconfigure RCE(s) with the same functionality as $PE_i$
6: Perform N-Modular Redundancy (NMR) majority voting to identify at least one healthy RCE, $\Phi_i \leftarrow 0$ 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 RCE by updating $N=N-N_s$, Re-initialize $i=1$
10: **end while**
11: Use a healthy RCE to check all other PEs

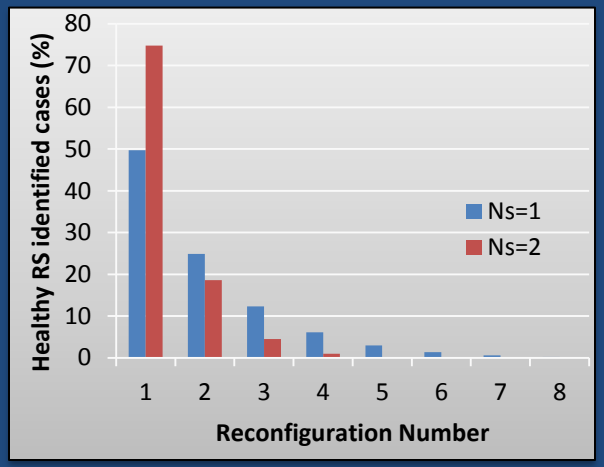
- Upon identifying faulty PEs, their functionality is assigned to healthy PEs which may either be slacks reserved at design time or some PEs computing lower priority-functions
- In case of a DCT, the DC-coefficient computation function is more significant than AC-coefficients computing functions

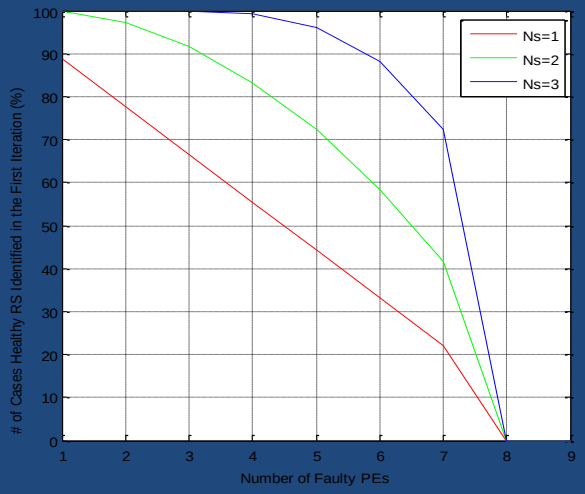
# FAULT-ISOLATION RESULTS

- DCT core. The core consists of 8 PEs, each PE computing one coefficient of the 8×8 DCT

- Figure 1 --> if the current DCT mode is 8×8 and one RS is available, approximately 50% of the fault scenarios are successfully isolated in the first iteration

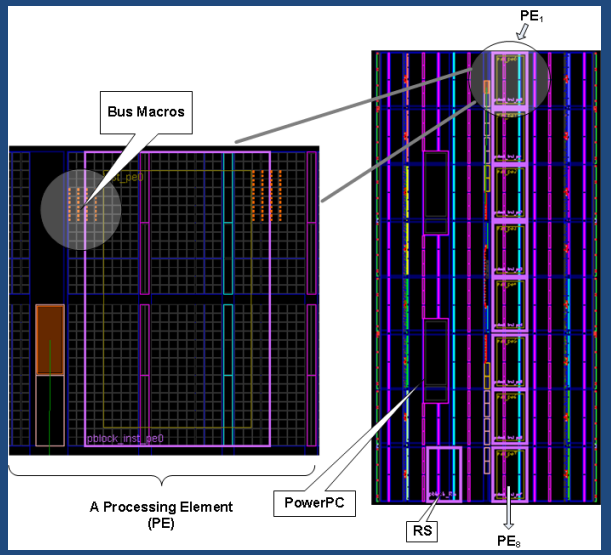
- Figure 2 shows the probability of isolating the faulty modules in the first iteration for different number of RS. Compared to a single slack, more failure scenarios are resolved with two slacks

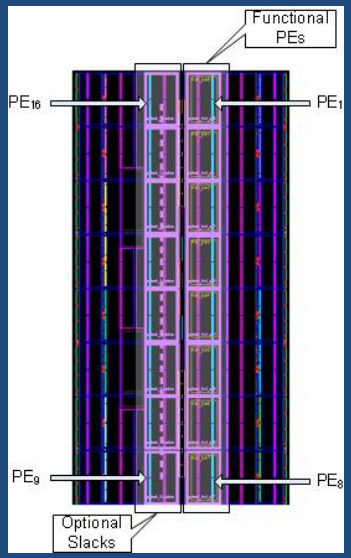


# CASE STUDY-1: VIDEO ENCODER

- H.263 video encoder application on-chip PowerPC processor of a Virtex-4 FPGA provided on Xilinx ML-410 development board
- DCT module in hardware specified by Verilog HDL
- Xilinx PlanAhead for Partial Reconfiguration (PR) flow while the software and hardware system is built using Xilinx Platform Studio (XPS)
- Xilinx Internal Configuration Access Port (ICAP) for downloading the partial bitstreams from external compact flash

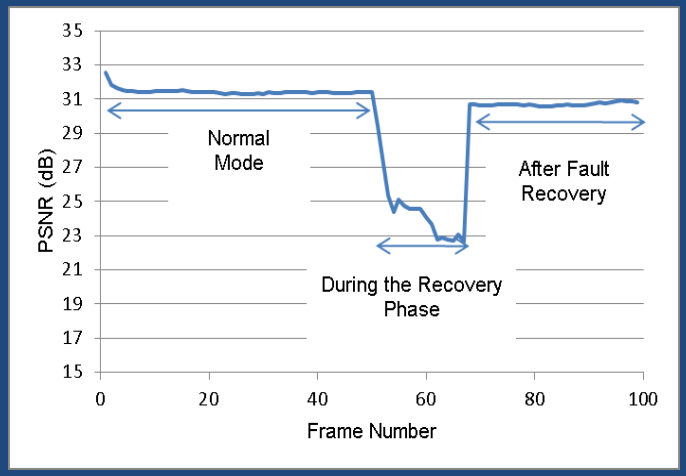


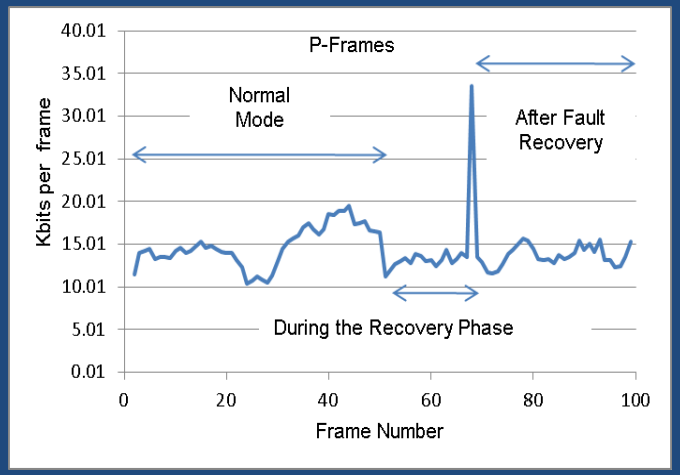




The floorplan of the hardware

# FAULT-RESILIENT DCT MODULE

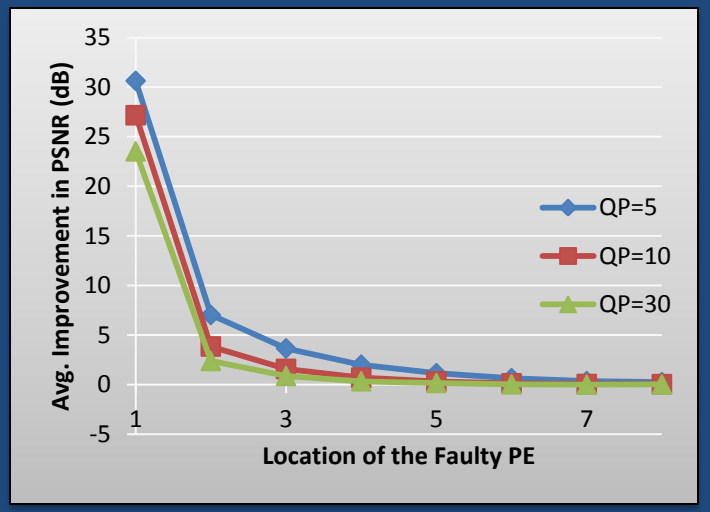
- Simulation of fault injections in the post-place and route simulation model of the circuit generated by Xilinx software flow
- An example of the video encoder in a faulty scenario in which $PE_1$ and $PE_4$ are faulty
- As the faulty $PE_1$ was performing an important function, that is, the computation of the DC coefficient, therefore a healthy PE is assigned to this functionality
- Quality degradation is spanned to a few frames during which partial throughput is available



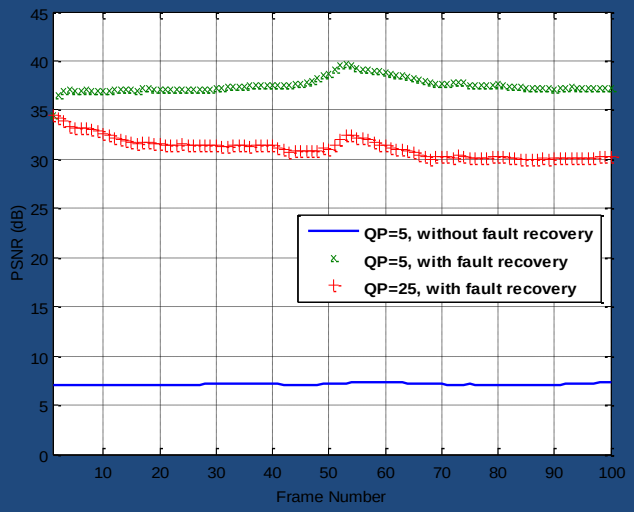



# EFFECT OF QUANTIZATION PARAMETER

Improvement in PSNR in case of different faulty PEs location, for three QP values

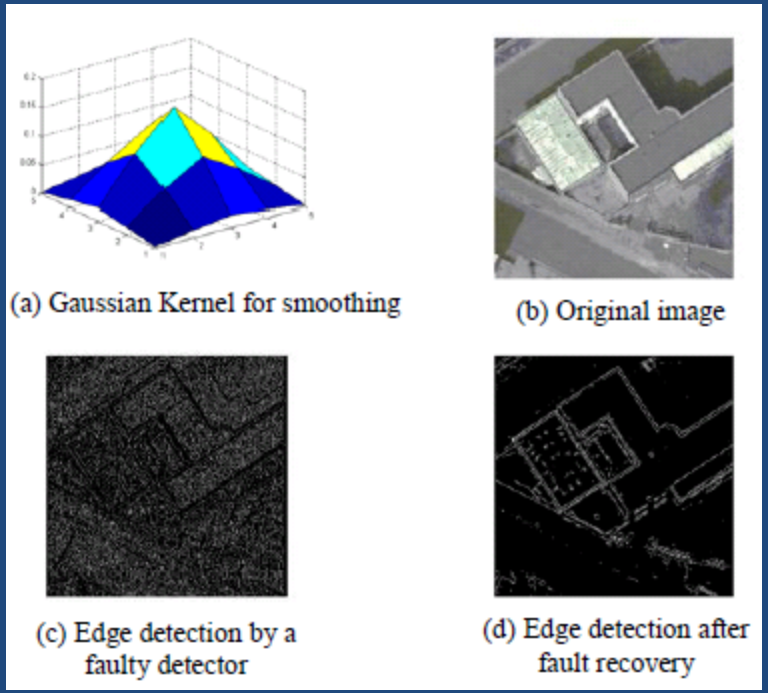



- The faulty behavior of a PE computing the DC coefficient impacts the video quality more than that of a PE computing the higher frequency AC coefficients.
- In general, a high value of QP results into a higher compression ratio at the cost of low visual quality
- A fault handling encoder even for a high QP value outperforms that with a low QP value without fault handling.



# CASE STUDY-2: EDGE DETECTOR

- The sustainability of edge detecting applications is desirable in harsh operating environments [16]
- A Canny edge detector [17-18] for image-processing due to its enhanced edge detection capability → Evaluate the behavior of faults in a Canny edge detection module
- A 5×5 Gaussian Kernel for smoothing phase of the detector
- The convolution operation is performed by multiple PEs to accelerate the performance of the edge detection



# CONCLUSIONS

- As opposed to a test vectors strategy, the proposed scheme evaluates the modules 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 PRR consists of 1152 LUTs in addition to resources like FF, BRAM and DSP48 blocks. In a BIST-based resource testing scheme, these resources need to be tested exhaustively, at all times even before a fault occurrence.
- The fault coverage includes logic resources as well as routing resources. The manifested malfunctioning of any of them will result in the utilizing PE to be flagged as faulty, and then the assigned function is moved to another area in the chip.
- The dynamic reconfiguration capability of the devices is essential to implement the proposed fault handling flow.
- Some limitations of the proposed scheme include the lack of fault masking provided by a TMR.

# REFERENCES


[1]. M. Berg, C. Poivey, D. Petrick, D. Espinosa, A. Lesea, K. LaBel, M. Friendlich, H. Kim, and A. Phan, IEEE Transactions on Nuclear Science 55, 2259 (**2008**)

[2]. M. Agarwal, B. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," in VLSI Test Symposium, 2007. 25th IEEE, May 2007, 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, pp. 46 –53, Feb. 2011.

[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, pp. 1114 –1125, Aug. 2011.

[6]. SPP1500, "Dependable embedded systems," retrieved on January 8, 2012 [Online] http://spp1500.itec.kit.edu/63.php.

[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), June 2006, pp. 288–291.

[8]. J.-C. Laprie, "Dependable computing and fault tolerance : Concepts and terminology," in Fault-Tolerant Computing, 1995, ' Highlights from Twenty-Five Years'., Twenty-Fifth International Symposium on, Jun 1995, p. 2.

[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, pp. 1545 –1558, Nov. 2008.

[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, Dec. 2010, 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, pp. 1399 –1412, Oct. 2008.

# REFERENCES (Contd…)


[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, pp. 133-149, Feb. 2010.

[13]. S. C. Parris, M.G. and R. DeMara, Progress in Autonomous Fault Recovery of Field Programmable Gate Arrays. ACM Computing Surveys, p. 33 pages, April 2010.

[14]. R. DeMara and K. Zhang, "Autonomous FPGA fault handling through competitive runtime reconfiguration," in Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on, July 2005, pp. 109 – 116.

[15]. Xilinx, "Embedded system tools reference manual," 2008, retrieved on January 8, 2012

[Online] http://www.xilinx.com/support/documentation/sw manuals/edk10estrm.pdf.

[16]. 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." in ERSA'10, 2010, pp. 49–58.

[17]. J. Canny, "A computational approach to edge detection," Pattern Anal. and Mach. Intell., IEEE Trans. on, vol. PAMI-8, no. 6, pp. 679 –698, Nov. 1986.

[18]. T. Kim, H. Adeli, C. Ramos, and B.-H. Kang, Signal Processing, Image Processing, and Pattern Recognition, ser. Springer-Verlag. Springer, 2011.

[19]. VGG, "Oxford visual geometry group (vgg)'s images dataset : Aerial views," 2012, retrieved on Feb. 13, 2012 [Online] http://www.robots.ox.ac.uk/vgg/data/.

[20]. Nicholas J. Macias and Peter M. Athanas, Using Low-Level Architectural Features for Configuration InfoSec in a General-Purpose Self-Configurable System, International Journal of Adv. Science and Tech. vol. 22, pp. 1-12 (**2010**)

[21]. Saranyadevi.S, Thangavel.M, A Low Power Structure Design of 2D-LFSR and Encoding Technique for BIST, International Journal of Advanced Science and Technology. vol. 18, pp. 11-22 (**2010**)

[22]. Shamna.K, S.R Ramesh, Design and Implementation of an Optimized Double Precision Floating Point Divider on FPGA, International Journal of Advanced Science and Technology. vol. 18, pp. 41-48 (**2010**)