UCF

COMPUTER ARCHITECTURE LABORATORY

Electrical and Computer Engineering

cal.ucf.edu

Dynamic Partial Reconfiguration Approach to the Design of Sustainable Edge Detectors

Ronald F. DeMara, Jooheung Lee

Rawad Al-Haddad, Rashad Oreifej, Rizwan Ashraf

University of Central Florida

Orlando, FL

Brian Stensrud and Michael Quist

Soar Technology, Inc.

Orlando, FL

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract #W31P4Q-08-C-0168

Introduction

- **Overall Design Objectives:**
  - Mission-critical applications in harsh operating environments
  - Sustainable design to increase reliability and application life-span
  - Autonomous operation and adapted to situation at hand
  - Self-recovery using flexible and dynamic fault tolerance strategy
  - Software control and monitoring of the hardware system
- **Use Case:** Sobel-Edge Detection (Image-Processing Application)
- **Reconfigurable Adaptive Redundancy System (RARS):**
  - Dynamically adapts to operational environment by changing topology
  - Maximizes system performance based on run-time criteria
  - Self-aware of component status to reconfigure fabric when needed
  - Interfaced with PC-based monitoring system using JTAG to perform higher-level control and management
  - implemented on Xilinx Virtex-4 (XCV4SX35) device

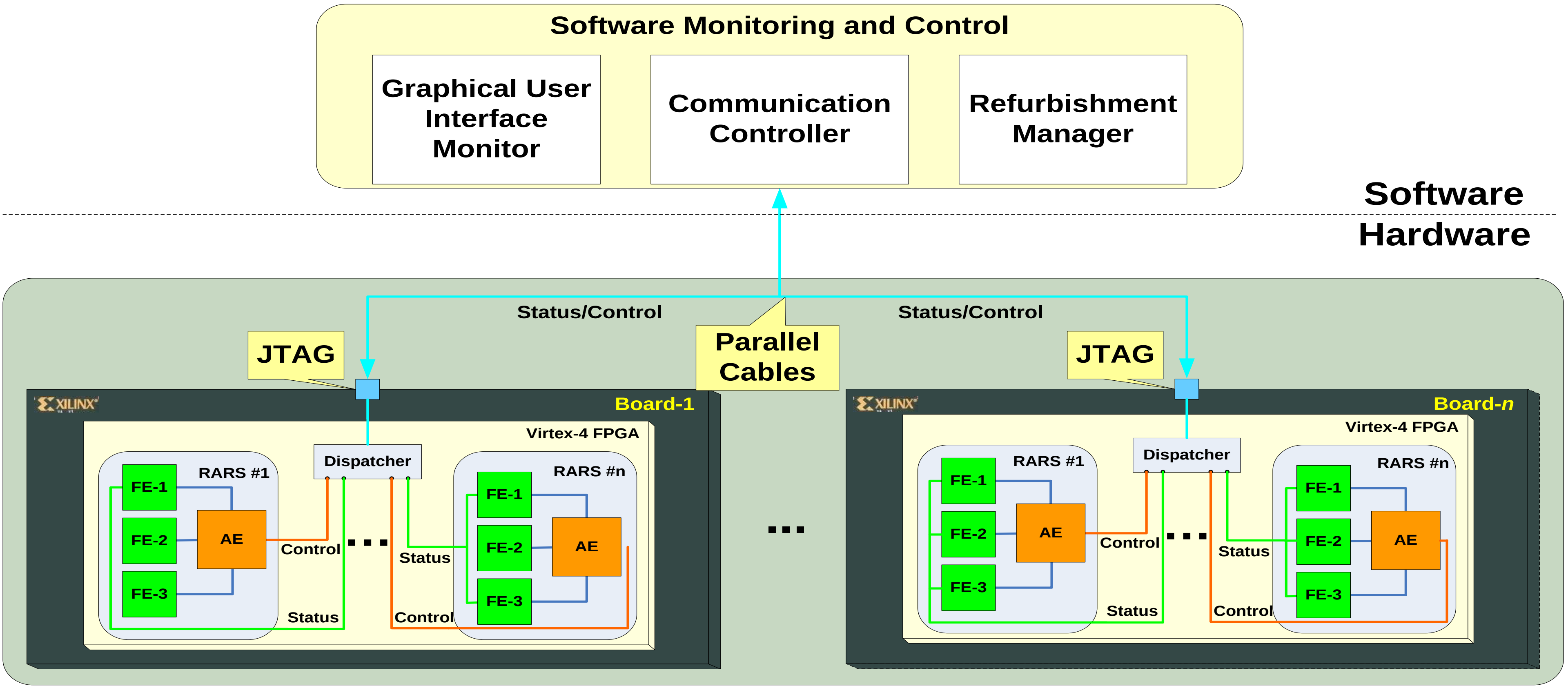
Related Work

- **Evolutionary Algorithms and Evolable Hardware:** Evolution of *fault-tolerant electronic circuits* on devices such as FPGAs [12], [16] extensive, also FPTAs.
- **Image Processing and Evolution:** Tyrrell [15] and Ross [20] used genetic programming with *software model-based* fitness evaluation to evolve edge detectors.
- **Multilayer Runtime Reconfiguration Architecture (MRRA)** A software framework [16] capable of communicating with the FPGA through high level API calls to perform *direct bitstream manipulation*.

- **Fault Detection:**
  - ✓ Modular redundancy
  - − CBS readback & compare
  - − Concurrent Error Detection
  - − BIST

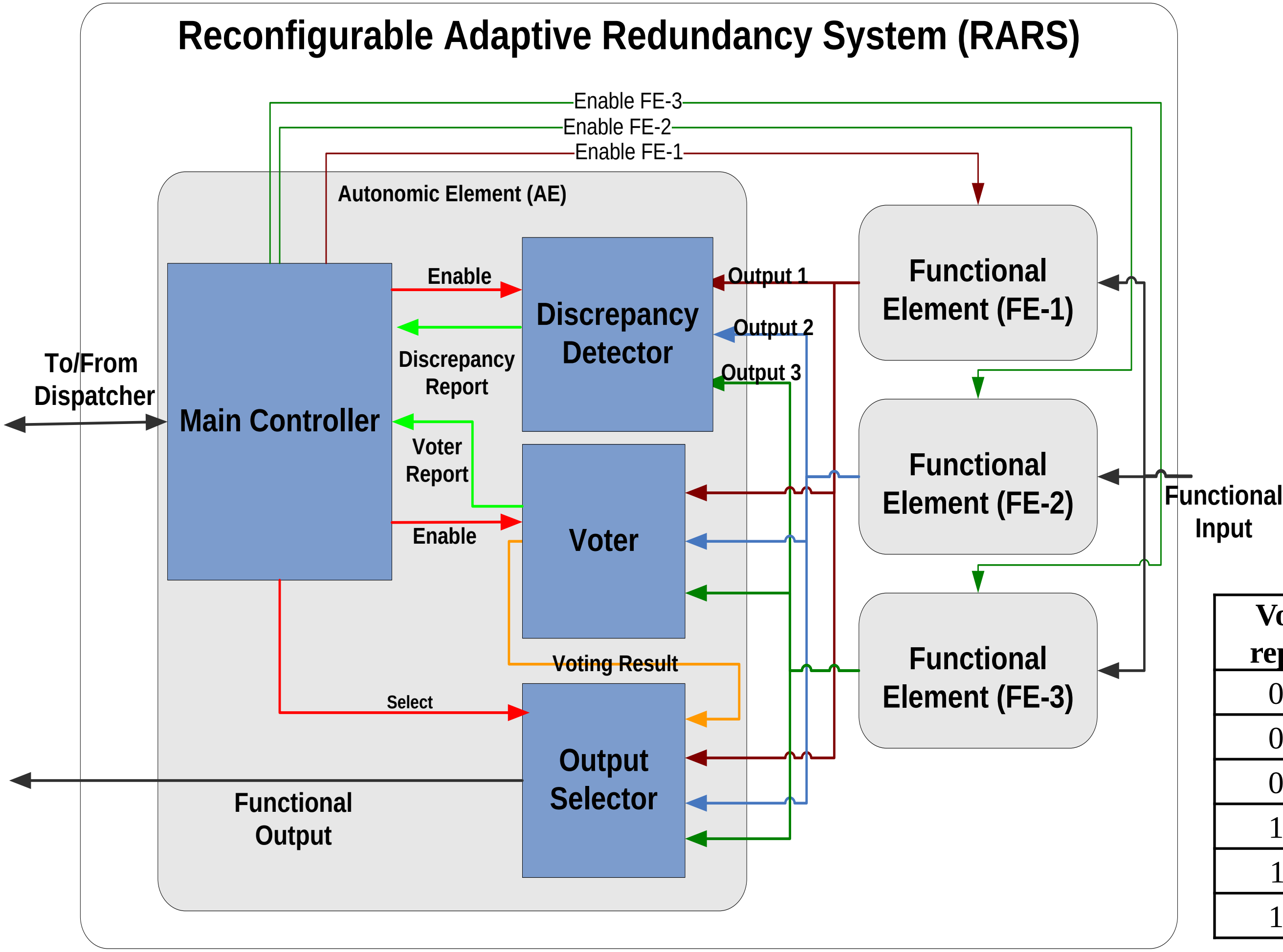
- **Fault Repair:**
  - − Blind Scrubbing
  - − TMR with recovery
  - − Reconfiguration (A-priori Synthesized Allocations)
  - − Roving STARs (Online BIST)
  - ✓ Evolutionary Techniques

System Design



System Level Schematic

RARS Architecture

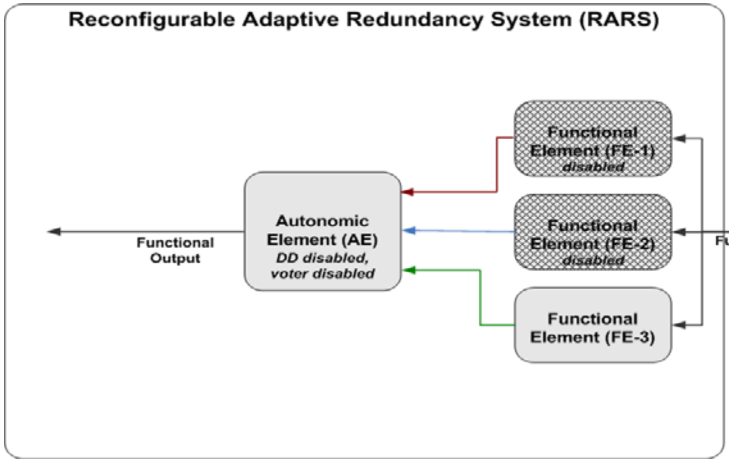


Voter report	Description
000	No discrepancy among the FEs
001	FE1 is discrepant
010	FE2 is discrepant
100	FE3 is discrepant
111	The three FEs are discrepant
101	Voter is disabled

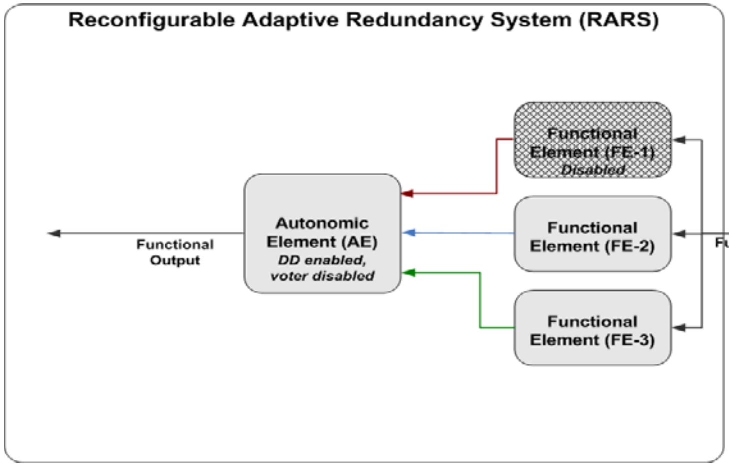
Autonomic Element (AE): Autonomous and application-independent

Functional Element (FE): Any application can be implemented that provides specified status signals for monitoring and control for reconfiguration

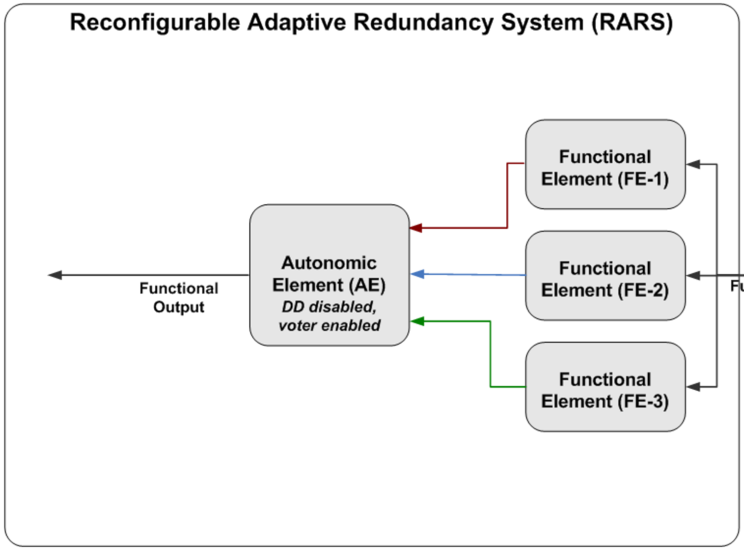
RARS Supported Modes



Simplex : One FE is functional



Duplex: : Two FEs are functional with discrepancy detection



TMR: All three FEs are functional with voter enabled

Hybrid mode: ***Temporal combination of the other modes***

- **Functional overview:**
  - *Performance Manager (PM)* compares the high-level throughput against system requirement
  - PM monitors status of each FE, configuration of AE and the overall performance level of system
  - Implements special-purpose communication protocol
  - Carries messages through the JTAG interface to the hardware via the General-purpose Native JTAG Tester (GNAT)
  - Serves two purposes:

1. Provide the monitoring module with graphical interface (Java applet)

2. Enable higher-level fault-detection and recovery techniques (GA)

Message Name	Description
FE_STATUS_REQUEST	Request status of a certain FE
AE_STATUS_REQUEST	Request status a certain of AE
PERFORMANCE_REQUEST	Request performance value

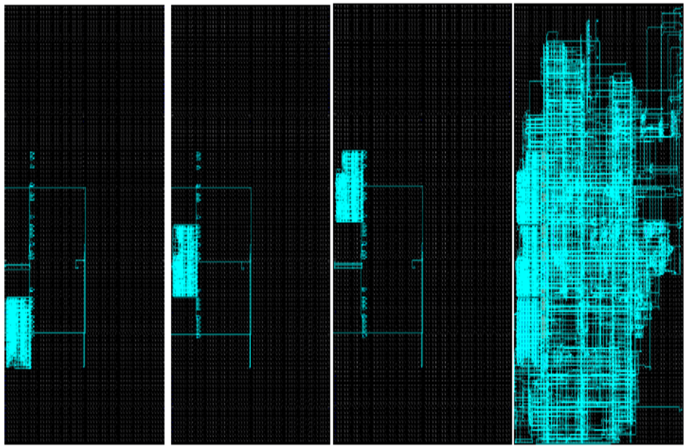
Table: Software to hardware messages

Message Name	Description
FE_STATUS_REPORT	Report certain FE status (online healthy, online faulty, offline, etc…)
AE_STATUS_REPORT	Report certain AE status (Duplex, Voter)
PERFORMANCE_REPORT	Report performance value

Table: Hardware to software messages

Dynamic Partial Reconfiguration

- *Early Access Partial Reconfiguration (EAPR)* design flow was used to achieve dynamic partial reconfiguration
- Reduce repair time due to small bitstream size (PR configures the FE in in 1.8% of the time required to reconfigure whole system)
- Facilitates repair while system is kept online



PRR's FE1, FE2 and FE3, and full design respectively

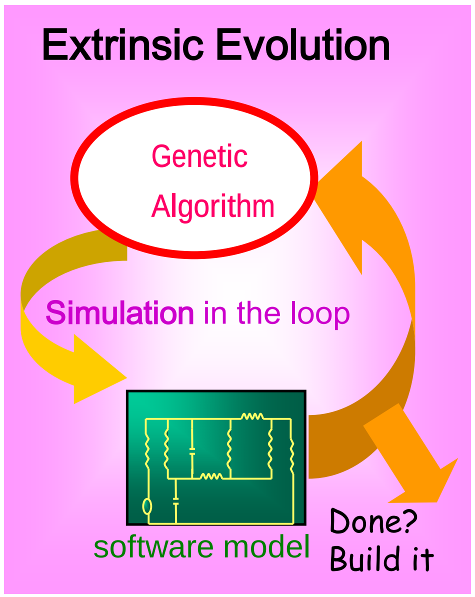
Approach	Virtex-4 Full	Virtex-4 Partial
Device	Virtex-4	Virtex-4
Bitstream Size	1.633 MB	30.61 KB
JTAG Cable	parallel cable IV 5Mbps	parallel cable IV 5Mbps
Config time (msec)	2613	48

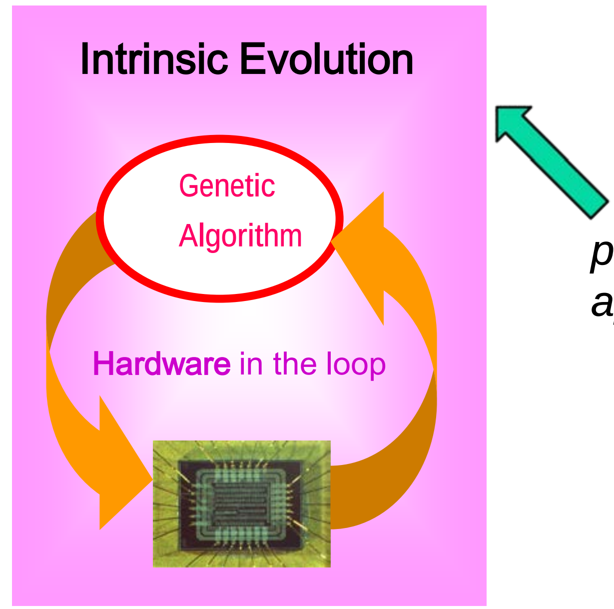
Approach	Virtex-4 Full	Virtex-4 Partial
LUTs	2785 (9%)	601 (67%)
Slices	1734 (11%)	368(82%)
IOB	70 (15%)	0 (0%)
BUFGs	11 (34%)	0 (0%)
RAMB16s	98 (51%)	0 (0%)
DCM	1 (12%)	0 (0%)
BSCAN	1 (25%)	0 (0%)

Partial Vs Full bitstream Comparison

Evolutionary Repair Strategies

- Genetic Algorithms: based on principles of natural selection.
- Genetic operators such as *mutation* and *crossover* used to search within a large irregular complex solution space, especially multi-objective optimization





preferred approach

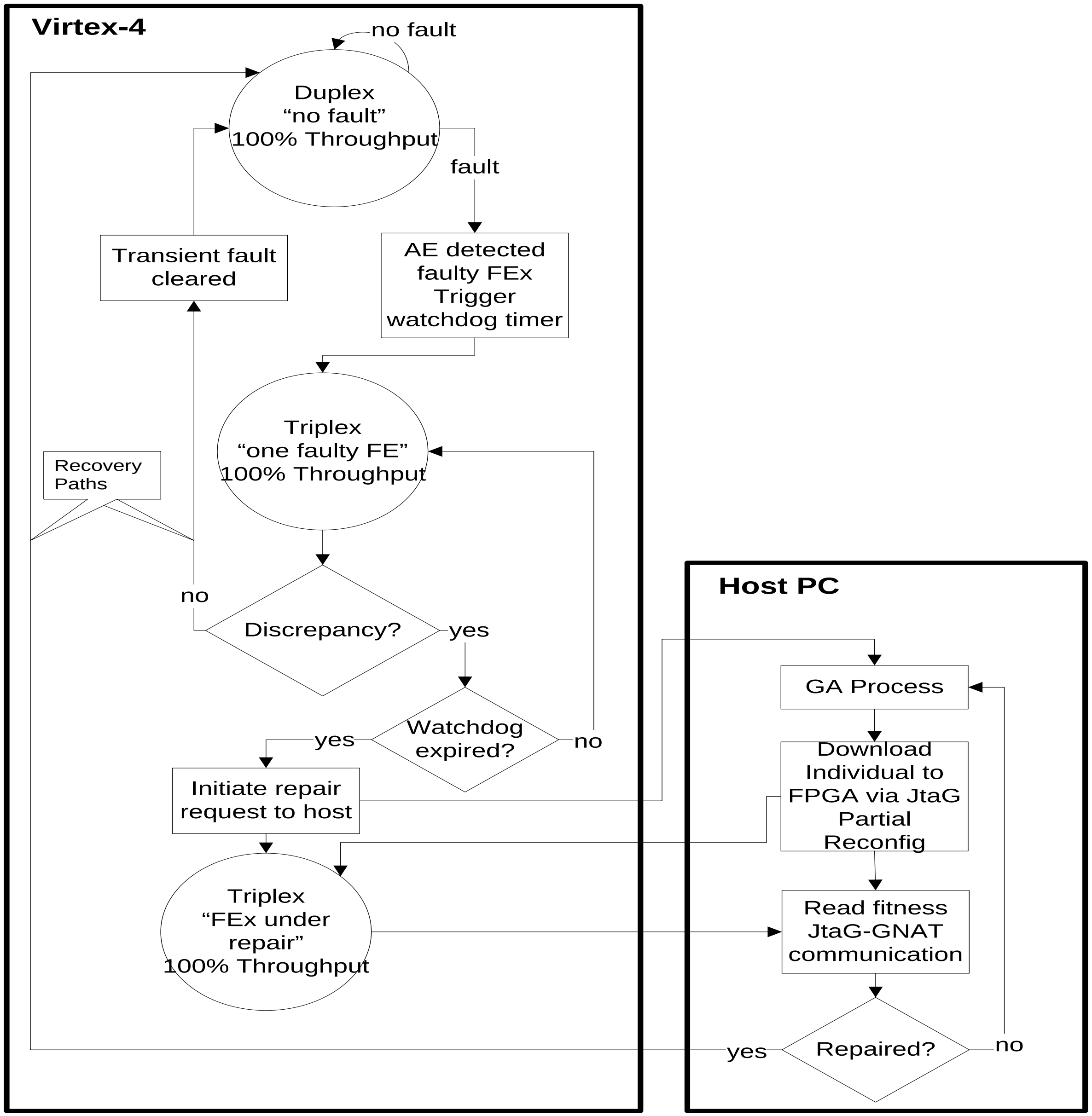
- Fitness function is measured out of the physical device output
- Constraints imposed by the device's internal structure
- Demonstrates that the resultant design will actually fit on to the implementation platform

•Functional models that abstract the physical aspects of the real device
•Representation has to undergo placement and routing before implementation.

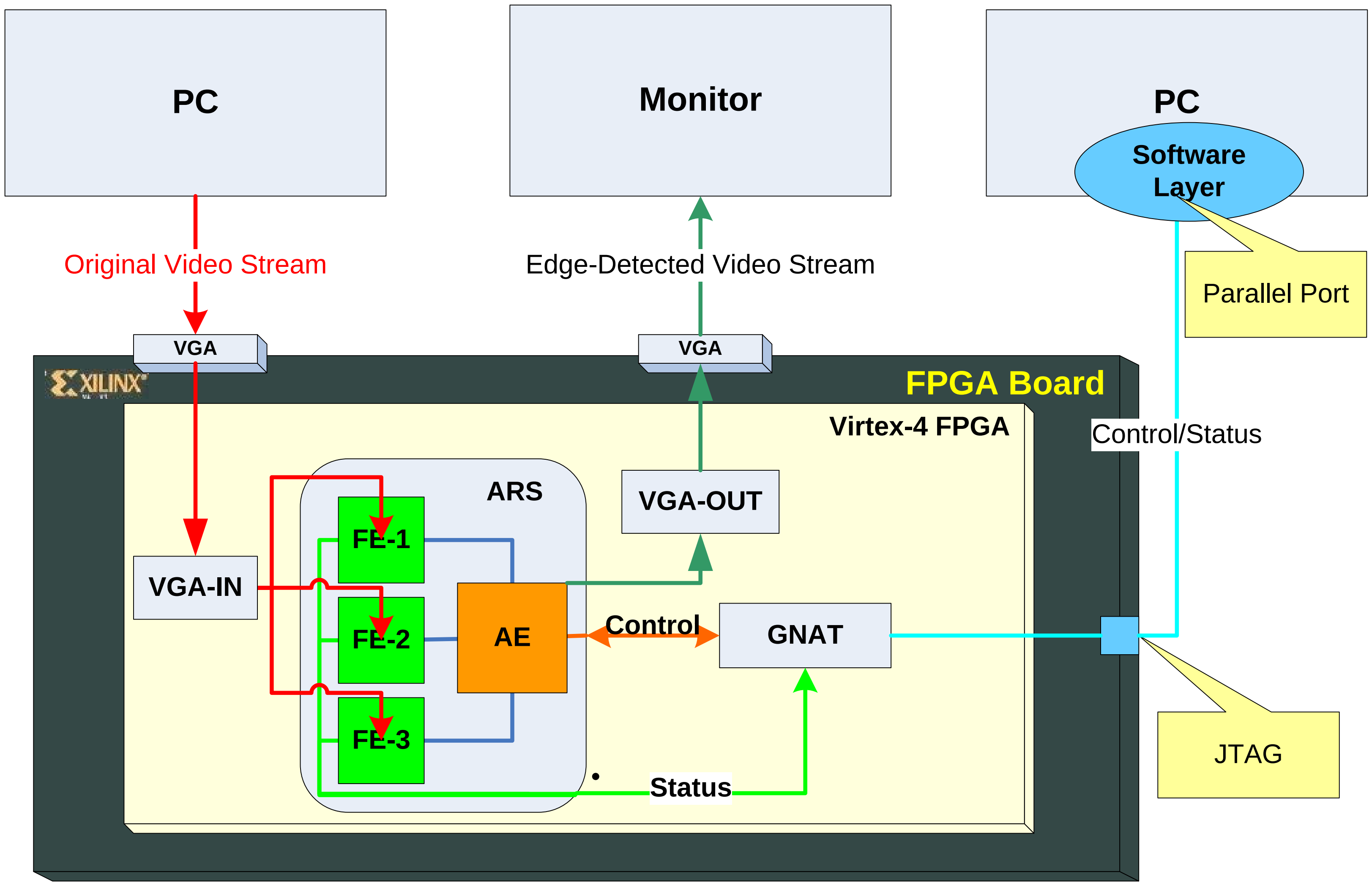
Monitoring & Control Components

- **GA Engine:** C++ application implements a **customizable standard GA** and is also able to directly read the fitness value from hardware
- **Chromosome Manipulator:** Hardware **abstraction** from the GA engine perspective
- **Multi Runtime Reconfiguration Architecture (MRRA):** partial bitstream **manipulation and decoding**, modular architecture provides the logic, translation, and reconfiguration layers which facilitates **communication with the FPGA**
- **JTAG (IEEE 1149.1):** serial port on hardware side, also used for dynamic partial reconfiguration purposes
- **General-purpose Native JTAG tester (GNAT):** configured on the device to support Input/Output operations with the **user implemented** circuit

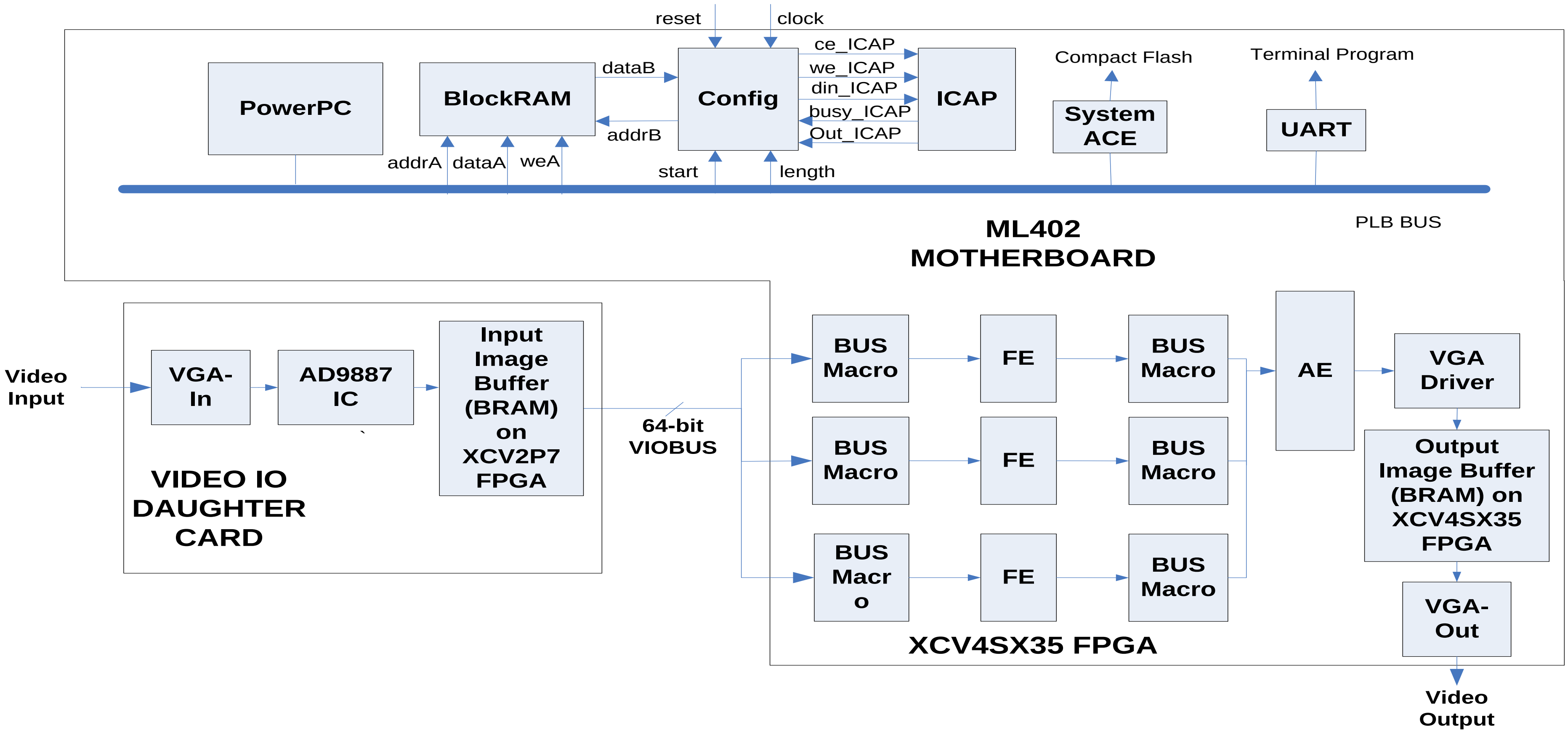
System Repair Cycle



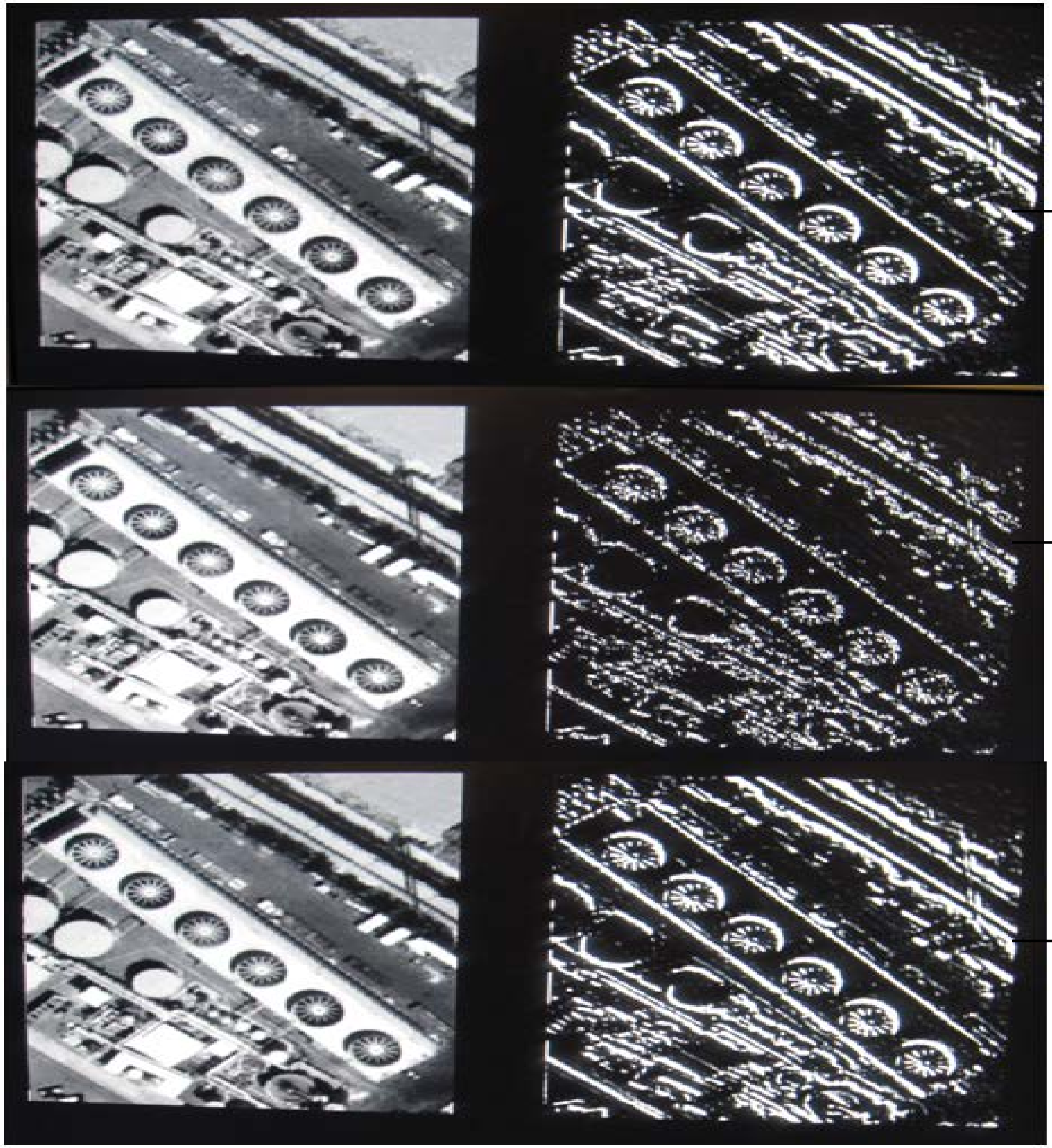
Edge Detection Use Case



Physical Design of the Use-Case on the Xilinx VSK platform



Results



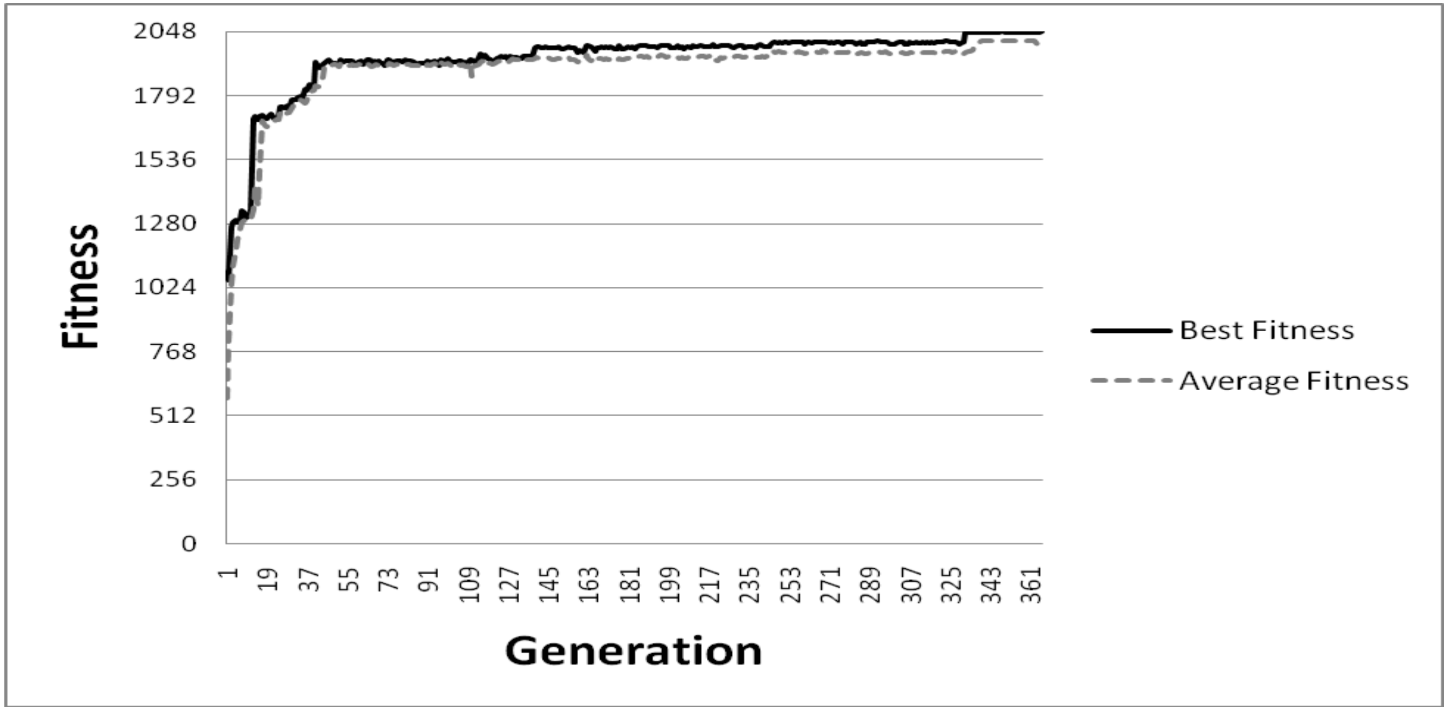
Normal fault-free output

Degraded output

Refurbished 100% performance output

GA-Based Repair Results

- GA operated on 8 LUTs: Critical LUT's that are highly influential on the performance of the Edge Detector
- 5 runs. The maximum fitness is 2048, which means that out of 2048 discrepancy reading samples, the correct solution does not show any discrepancy when its output is compared to a completely healthy Sobel edge detector.



Comparison

	Hollingworth [15]	Gudmundsson [13]	Ross [20]	RARS
Application	Generic images (fairly simple)	Unfragmented localized thin edges in medical images.	Microscopic images from mineral samples	*Generic (satellite images, uniform patterns, etc…)*
Methodology	Exploiting inherent parallelism in images	Split image into linked sub-images. Maintain links between adjacent pixels	Training stage (requires sampling 23.6% of image), followed by Genetic programming.	*Evolving a subset of the Edge Detector (critical LUTs) in order to recover from faults.*
Fitness Evaluation	Software model	Software model	Software model	***Intrinsic Evolution (HW in the loop)***
Evolutionary Algorithm	Genetic Programming	2D Genetic Algorithm problem-specific operators.	Genetic Programming Training stage (~25%) Evolution (~75%)	*Genetic Algorithm*
Genetic String Coding	Four node functions (and, or, not, xor), 8 terminal values for pixels around the evolved one	Edge Map: image pixels are masked with corresponding values in pixel map (0: not edge, 1: edge)	High-level functions (avg, min, max, stdev) Terminal Pixels and high-level ephemeral (gradient, intensity)	***Direct Bitstream Evolution. The solution coding is the actual bitfile.***
Fitness Function	Pratt Figure of Merit (PFM) relative to Sobel edge-detector F = 1/(1+Pef + Pnf)	Highly complex cost function based on 5 cost factors	Biased random sampling fitness evaluation for training. Program fitness is similar to PFM.	***Model-Free triplex discrepancy based. No application specific a-priori knowledge needed.***
Evolution Speed	Partial solution in 2333 generations (24 hours of evolution time)	2300 generations for rings image. 300 generations for thin, well-localized edges	75 generations. 25% of the images for training, Very large population size of 2000	*361 generations, low population size of 10 on Lena benchmark. 8 critical LUTs evolved.*
Best Fitness	Not reported	0.85 PFM with scaling factor of 0.01.	0.590 for Image 1 0.633 for Image 2	*100% compared to a pristine Sobel edge detector output.*

References

1. M. Parris, C. Sharma, and R. F. DeMara, “Progress in Autonomous Fault Recovery of Field Programmable Gate Arrays,” accepted to *ACM Computing Surveys, in-press*.
2. R. E. Lyons, and W. Vanderkulk, “The use of triple-modular redundancy to improve computer reliability,” *IBM Journal of Research and Development*April 1962, pp. 200-209.
3. P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, “An overview of reconfigurable hardware in embedded systems,” *EURASIP Journal on Embedded Systems,* Volume 2006, Article ID 56320, pp. 1-19.
4. M. Hubner, and J. Becker, “Exploiting dynamic and partial reconfiguration for FPGAs: toolflow, architecture and system integration,” *Proc. 19th SBCCI Symp. on Integrated Circuits and Systems Design*, Ouro Preto, Minas Gerais, Brazil, August 28- September 1, 2006, pp. 1-4.
5. C. Kao, “Benefits of partial reconfiguration,” *Xilinx Xcell Journal,* Dec. 2005, pp. 65–67.
6. J. Huang, and J. Lee, “A self-reconfigurable platform for scalable DCT computation using compressed partial bitstreams and BlockRAM prefetching,” *Special Issue on Algorithm/Architecture Co-Exploration of Visual Computing, IEEE Transactions on Circuits and Systems for Video Technology (TCSVT),* Nov. 2009, pp. 1623-1632.
7. C. Carmichael, M. Caffrey, and A. Salazar, “Correcting single-event upsets through Virtex partial configuration,” *Xilinx Application Note: XAPP216,* June, 2000.
8. J. Heiner, B. Sellers, M. Wirthlin, and J. Kalb, “FPGA partial reconfiguration via configuration scrubbing,” *International Conference on Field Programmable Logic and Applications, 2009*, Aug. 2009, pp. 99-104.
9. K. Zhang, G. Bedette, and R. DeMara, “Triple modular redundancy with Standby (TMRSB) supporting dynamic resource reconfiguration,” *IEEE Autotestcon,* Los Angeles, CA, Sept. 2006, pp. 690-696.
10. J. M. Emmert, C. Stroud, and M. Abramovici, “Online fault tolerance for FPGA logic blocks,” *IEEE Transactions on VLSI Systems,* Vol. 15, No. 2, Feb. 2007, pp. 216-226.
11. D. Keymeulen, R.S. Zebulum, Y. Jin, and A. Stoica, “Fault-tolerant evolvable hardware using field-programmable transistor arrays,” *IEEE Transactions on Reliability,* Vol 49, No. 3, Sept. 2000, pp. 305-316.
12. S. Vigander, “Evolutionary fault repair of electronics in space applications,” *Dissertation, University of Sussex, Brighton, UK,* February, 28, 2001.

References (Cont…)

13. M. Gudmundsson, E.A. El-Kwae, and M.R. Kabuka, “Edge detection in medical images using a genetic algorithm,” *IEEE Transactions on Medical Imaging,* Vol. 17, No. 3, June 1998, pp. 469-474.
14. J.F. Cayula, and P. Cornillon, “Edge detection algorithm for SST images,” *Journal of Atmospheric and Oceanic Technology,* Vol. 9, 1992, pp. 67–80.
15. G.S. Hollingworth, S.L. Smith, and A.M. Tyrrell, “Design of highly parallel edge detection nodes using evolutionary techniques,” *Proc. 7th Euromicro Workshop on Parallel and Distributed Processing*, Funchal, Portugal, February 3-5, 1999.
16. H. Tan, and R. DeMara, “A multilayer framework supporting autonomous run-time partial reconfiguration,” *IEEE IEEE Transactions on VLSI Systems,* Vol. 16, No. 5, May, 2008, pp. 504 – 516.
17. R. Oreifej, R. Al-Haddad, H. Tan, and R. F. DeMara, “Layered approach to intrinsic evolvable hardware using direct bitstream manipulation of Virtex II Pro devices,” *Proc. IProceedings of the 17th International Conference On Field Programmable Logic And Applications (FPL'07),* Amsterdam, Netherlands, August 27 – 29, 2007. pp. 299-304.
18. S. Mitra, N.R. Saxena, and E.J. McCluskey, “A design diversity metric and reliability analysis for redundant systems,” Proc. *1999 IEEE International Test Conference*, September 28-30, 1999, pp. 662-671.
19. Brian Pratt, Michael Caffrey, James F. Carroll, Paul Graham, Keith Morgan, and Michael Wirthlin, “Fine-Grain SEU Mitigation for FPGAs Using Partial TMR”, *IEEE Transactions on Nuclear Science*, Vol. 55, No. 4, pp. 2274-2280.
20. B.J. Ross, F. Fueten, and Y.Y. Dmytro, “Edge detection of petrographic images using genetic programming,” *Proc. Genetic and Evolutionary Computation Conference*, July 8 - 12, 2000, Las Vegas, NV, USA, pp. 658–665.