Fault-Mitigation by Adaptive Dynamic Reconfiguration 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

Abstract

We present an area-efficient dynamic fault-handling approach to achieve high survivability for DSP circuits. Fault detection, isolation, and recovery are performed using discrepancy information derived from the existing functional throughput by reconfiguring one of the N + 1 Reconfigurable Partitions (RPs) to replicate each of the N modules in succession. This differs significantly from the conventional approaches that heavily rely on static temporal/spatial redundancy and sophisticated error prediction/estimation techniques. The principal space complexity metric is the additional physical resources utilized to support the underlying fault-handling mechanism where a single RP can check the health of multiple distinct functional blocks, by leveraging the property of dynamic partial reconfiguration. We demonstrate this approach by implementing a video encoder's DCT block with a Xilinx Virtex-4 device and also numerically simulating a Canny Edge Detector.

Keywords: *Fault-resilience, dynamic partial reconfiguration, FPGAs, autonomous operation, fault-tolerance, availability, reconfigurable slack*

1. Introduction

With the advent of 20nm CMOS device technology and the emergence of nano-scale devices and vertical interconnect technology, permanent failure and aging effects can become more prominent in both logic and interconnect resources [1-3]. 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) FPGA-based system [15, 16], 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.

2. Amorphous Slack (AS) Fault-Handling Methodology

To achieve fault-handling operation, we propose an Amorphous Slack (AS) technique to time-multiplex the processing regions 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 RCEs 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) [17] 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 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 Φ_i= 1 in a vector $\mathbf{\Phi}$ of length ($N + N_s$) stands for faulty nature of the PE_i, Φ_i= 0 for healthy PE_i, and Φ_i= x for suspected PE_i. The vector $\mathbf{\Phi}$ 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.

Algorithm 1: Fault Isolation Algorithm

Require: N, N_s

Ensure: $\boldsymbol{\Phi}$

1: Initialize $\boldsymbol{\Phi}=[x\ x\ x\ \ldots\ x]^T$, $i=1$

2: **while** $(\{k \mid k \in \boldsymbol{\Phi}, k=0\} = \phi)$ **do**

3: Designate PE_s as checker(s) ; $(N+1) \le s \le (N+N_s)$

4: **while** $(i \le N)$ **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

3. Experimental Results

3.1. Case Study-1: Video Encoder

An example of the video encoder in a faulty scenario is presented in Figure 1. The faulty situation of PE_1 and PE_4 is examined here. The healthy nature of the RS makes it possible to isolate the faulty PE in the first iteration in which two reconfigurations are involved. As soon as the RS output is compared with PE_2 which is healthy, the RS is identified as healthy. 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. Figure 1 illustrates that the quality of signal in terms of Peak Signal to Noise Ratio (PSNR) is better in case of an encoder with fault handling capability than that of a baseline encoder even operating at a lower Quantization Parameter (QP) value.

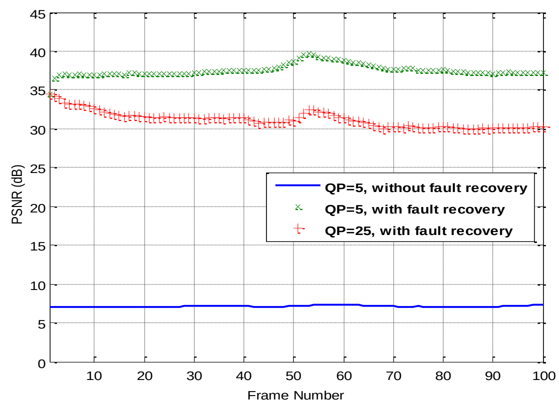


Figure 1. PSNR of recovered frames of a video sequence

3.1.1. Verilog design of the DCT core

The floating-point values of the DCT kernels matrix are represented by fixed-point values as given by hex numbers in Figure 2. Each floating point value is represented by a 12-bit fixed-point number, thus a total of 96 bits are used to specify each kernel. Then, each kernel is stored inside a PE inferred from Verilog code in Figure 3. The Multiply Accumulate (MAC) operation is synthesized by using Xilinx DSP48 elements. The sign-bit of the 21-bit dot-product result is replicated to get a 32-bit 2's complement representation. This 32-bit value from a PE's output represents a DCT coefficient. For 8x8 DCT mode, an array of 8 PEs operates in parallel on a row in 8 input pixels to produce the results of 1^{st} stage of DCT operation. The result of 1-D DCT is written into a transposition memory. The 2^{nd} stage of DCT is performed by the same array of PEs, yet with column-wise reading format of those values written by the 1st stage into the transposition memory.

```
`define DCTWIDTH 32
`define PIXWIDTH 12
`define N 8 //DCT mode
`define start contrin[0]
`define write contrin[1]
`define read contrin[2]
`define ready controut[0]
`define KERNELDC 96'h2D4_2D4_2D4_2D4_2D4_2D4_2D4_2D4 //DC
`define KERNEL0 96'h3EC_353_239_0C8_F38_DC7_CAD_C14 //AC0
```

```
`define KERNEL1 96'h3B2_188_E78_C4E_C4E_E78_188_3B2 //AC1
`define KERNEL2 96'h353_F38_C14_DC7_239_3EC_0C8_CAD //AC2
`define KERNEL3 96'h2D4_D2C_D2C_2D4_2D4_D2C_D2C_2D4 //AC3
`define KERNEL4 96'h239_C14_0C8_353_CAD_F38_3EC_DC7 //AC4
`define KERNEL5 96'h188_C4E_3B2_E78_E78_3B2_C4E_188 //AC5
`define KERNEL6 96'h0C8_DC7_353_C14_3EC_CAD_239_F38 //AC6
```

Figure 2. Parameters.v file to specify DCT core's parameters

```
`include "parameters.v"
module mac_0(
   input [`PIXWIDTH-1:0] din,
   input start,
   input clk,
   input res,
   output [`DCTWIDTH-1:0] dout,
   output ready
   );
/////////////////////////////////////
reg [95:0] c;
reg [95:0] dctM = `KERNEL0;
///////////////////////////////////////
reg [3:0] count;
always @(posedge clk or posedge res)
if (res) count<=0;
else if (start) count<=0;
else if (count==(`N+2)) count<=count;//stop
else count<=count+1;
///////////////////////////////////
always @(negedge clk or posedge res)
if (res) c <= 0;
else if (start) c <= dctM;
else if (count>0) c <= {c[83:0],12'h000};
else c<=c;
//////////////////////////////////////
reg  [`DCTWIDTH-1:0] acc;//accumulator
always @(posedge clk or posedge res)
if (res) acc <= 0; else if (start) acc <= 0;
else if (count<`N) acc <= acc + { {(`DCTWIDTH-
12){c[95]}},c[95:84]}*{{(`DCTWIDTH-`PIXWIDTH){din[`PIXWIDTH-1]}},din};
else acc <= acc;
wire [21:0] dout_rounded;
```

```
assign dout_rounded=acc[31:10]+1'b1;//Round Half up by adding 0.5
assign dout={ {11{dout_rounded[21]}},dout_rounded[21:1]};
assign ready=(count==(`N+1))?1:0;
endmodule
```

Figure 3. Verilog code to infer a MAC-based processing element

3.1.2. Hardware implementation using Xilinx ISE Design Suite: System Edition Version 14.3

While the design has been implemented using Xilinx 14.3 system edition [18-21], the implementation details can be found in [22] for Xilinx ISE 9.2 development tools. The DCT hardware core is interfaced with the on-chip processor contained in Xilinx Virtex-4 device through a Xilinx General Purpose Input/Output (GPIO) core. The GPIO core communicates to the processor via Xilinx LogiCORE Processor Local Bus (PLB). Table 1 lists the static and dynamically reconfigurable components of the hardware design implementing an H.263 video encoder. A processor-based system is instantiated in Xilinx EDK design environment. The static as well as partial reconfiguration (PR) design is synthesized in Xilinx Integrated Synthesis Environment (ISE). Then, the netlist files are imported into Xilinx PlanAhead to floorplan and implement the design. The input videos are stored on a compact flash in Xilinx ML410 development board as shown in

Figure 4. The reconfiguration status updates are communicated to a desktop computer via a serial port.

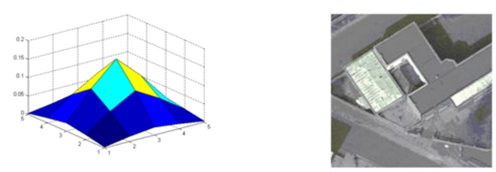
Table 1. Static and dynamic components of the design

Design	Module Name	Purpose
Static	System i	PowerPC, RS232, SystemACE, DDR2_SDRAM, DCT_GPIO,
	BUFG	Buffers to drive internal clocks signals
	DCM	Digital Clock Manager to generate clocks for DDR2, DCT core, and configuration controller
	ICAP_VIRTEX4	Internal Configuration Access Port
	config_ctrl	Configuration controller
	blockram	Block-RAM to hold configuration bitstreams
	dct_contr.	Controller to generate signals for the DCT core
	dct_p2s	Parallel to serial module
	muxdata	MUX to multiplex data to the input of PEs array from either DCT_GPIO core or transposition memory
	TransMem	A dual port transposition memory
	ICON,VIO,ILA	ChipScope debugging cores
Dynamic	PE_Array	MAC_DC, MAC_AC0, MAC_AC1, MAC_AC2, MAC_AC3, MAC_AC4, MAC_AC5 MAC_AC6

Figure 4. Xilinx ML410 development board to evaluate the proposed adaptive reconfiguration flow for video encoder application

3.2 Case Study-2: Edge Detector

The sustainability of edge detecting applications is desirable in harsh operating environments. A Canny edge detector [23-24] is popular for image-processing due to its enhanced edge detection capability. Therefore, we evaluate the behavior of faults in a Canny edge detection module. For this purpose, as shown in **Error! Reference source not found.**5(a), a 5×5 Gaussian Kernel is used for smoothing phase of the detector. We employed a distributed architecture where the convolution operation is performed by multiple PEs to accelerate the performance of the edge detection. **Error! Reference source not found.**5 illustrates the qualitative result of fault-handling for an image in the dataset available online [25].



(a) Gaussian Kernel for smoothing

(b) Original

(c) Edge detection by a faulty detector

(d) Edge detection after fault recovery

Figure 5. Gaussian kernel and qualitative results of the edge detector with fault-recovery capability

4. Conclusions

A fault handling mechanism using Amorphous Slack is introduced, which has advantages of continuous throughput with small degradation and low area overhead. Dynamic partial reconfiguration is used with hardware modularity to provide autonomous capability for survivable systems. Experiments with video coding/image processing applications indicate that fault resilience is achievable in an area efficient manner using Amorphous Slack.

Acknowledgements

This work was supported in-part by 2013 Hongik University Research Fund and the authors would like to acknowledge Xilinx and the Xilinx University Program for its generous donation of S/W design tools and H/W boards.

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 Trans. on, vol. 55, (**2008**), pp. 2259-2266.

[2] S. Mitra, K. Brelsford, Y. Kim, K. Lee and Y. Li, “Robust System Design to Overcome CMOS Reliability Challenges”, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Special Issue on the IEEE CAS Forum on Emerging and Selected Topics, (**2011**).

[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, R. Zebulum, D. Keymeulen, R. Tawel, T. Daud and A. Thakoor, "Reconfigurable VLSI architectures for evolvable hardware: from experimental field programmable transistor arrays to evolution-oriented chips", IEEE Trans. VLSI Syst., vol. 9, no. 1, (**2001**), pp. 227-232.

[5] R. Hyman Jr., K. Bhattacharya and N. Ranganathan, “Redundancy mining for soft error detection in multicore processors”, Computers, IEEE Trans. on, vol. 60, no. 8, (**2011**) August, pp. 1114 –1125.

[6] SPP1500, Dependable embedded systems, (**2012**) January 8, 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), (**2006**) June, pp. 288–291.

[8] A. Avizienis, J. C. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing", Dependable and Secure Computing, IEEE Transactions on, vol. 1, (**2004**), pp. 11-33.

[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 Computing Surveys, (**2010**) April, pp. 33.

[14] S. Mitra, N. R. Saxena and E. J. McCluskey, "Common-mode failures in redundant VLSI systems: a survey", Reliability, IEEE Transactions on, vol. 49, (**2000**), pp. 285-295.

[15] C. Carmichael, “Triple module redundancy design techniques for virtex FPGAs”, Xilinx Application Note: Virtex Series, (**2006**).

[16] N. Imran and R. F. DeMara, "A Self-Configuring TMR Scheme Utilizing Discrepancy Resolution", in proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig 2011), (**2011**) November 30 2011-December 2, pp. 398-403.

[17] N. Imran, J. Lee and R. F. DeMara, "Fault Demotion using Reconfigurable Slack (FaDReS)", IEEE Transactions on VLSI Systems, **(2012)**.

[18] Xilinx, “Planahead user guide”, uG632 (v14.3), (**2012**) October 16.

[19] Xilinx, “Partial reconfiguration user guide”, uG702 (v14.3), (**2012**) October 16.

[20] Xilinx, “Partial reconfiguration tutorial: Planahead design tool”, uG743 (v14.1), (**2012**) May 8.

[21] Xilinx, “EDK Concepts, Tools, and Techniques: A Hands-On Guide to Effective Embedded System Design”, UG683 (v14.4), (**2012**) December 18.

[22] N. Imran, J. Lee, Y. Kim, M. Lin and R. F. DeMara, "Amorphous Slack Methodology for Autonomous Fault-Handling in Reconfigurable Devices", International Journal of Multimedia and Ubiquitous Engineering (IJMUE), vol. 7, no. 4, (**2012**) October, pp. 29-44.

[23] J. Canny, “A computational approach to edge detection”, Pattern Anal. and Mach. Intell., IEEE Trans. on, vol. PAMI-8, no. 6, (**1986**) November, pp. 679 –698.

[24] T. Kim, H. Adeli, C. Ramos and B. -H. Kang, Signal Processing, Image Processing, and Pattern Recognition, ser. Springer-Verlag, Springer, (**2011**).
[25] VGG, "Oxford visual geometry group (vgg)'s images dataset: Aerial views", (**2012**) February 13, http://www.robots.ox.ac.uk/vgg/data/.
[26] N. J. Macias and P. M. Athanas, "Using Low-Level Architectural Features for Configuration InfoSec in a General-Purpose Self-Configurable System", International Journal of Advanced Science and Technology, vol. 22, (**2010**), pp. 1-12.
[27] S. Saranyadevi and M. Thangavel, "A Low Power Structure Design of 2D-LFSR and Encoding Technique for BIST", International Journal of Advanced Science and Tech., vol. 18, (**2010**), pp. 11-22.
[28] K. Shamna and S. R. Ramesh, "Design and Implementation of an Optimized Double Precision Floating Point Divider on FPGA", International Journal of Advanced Science and Tech., vol. 18, (**2010**), pp. 41-48.
[29] N. Imran, J. Lee, Y. Kim, M. Lin and R. F. DeMara, "Area-Efficient Fault-Handling for Survivable Signal-Processing Architectures", in Proceedings of First International Conference on Advanced Signal Processing, Seoul, Korea, (**2012**) March 30–31, pp. 37.

*Corresponding author: Jooheung Lee, Ph.D.
Department of Electronic and Electrical Engineering,
Hongik University, Korea. E-mail: joolee@hongik.ac.kr

** This paper is a revised and expanded version of a paper [29] entitled "Area-Efficient Fault-Handling for Survivable Signal-Processing Architectures", in Proceedings of First International Conference on Advanced Signal Processing, pp. 37, Seoul, Korea, March 30–31, 2012.

Authors

Naveed Imran received the M.S. degree in Electrical Engineering from the University 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 VLSI design for signal processing systems, FPGA-based embedded systems, power-aware and adaptive VLSI systems design, and reconfigurable hardware for image/video applications. 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. 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. 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.