International Conference on ReConFigurable Computing and FPGAs (ReConFig 2011)
30th Nov- 2nd Dec 2011, Cancun, Mexico

Heterogeneous Concurrent Error Detection (*hCED*) Based on Output Anticipation

Naveed Imran and Ronald F. DeMara
Department of Electrical Engineering and Computer Science,
University of Central Florida
E-mail: naveed@knights.ucf.edu, demara@mail.ucf.edu

Agenda

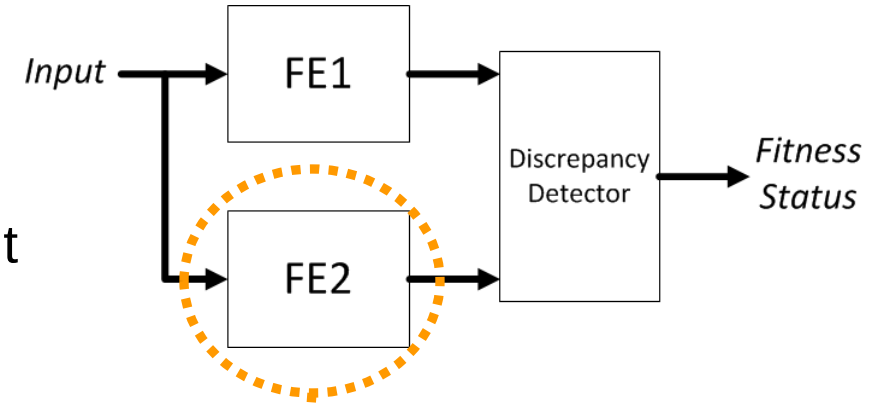
- Introduction
- Related Work
- Case-study: DCT Hardware Core
- The Baseline Arrangement
- Spatial Heterogeneous CED
- Temporal Heterogeneous CED
- Conclusions

Introduction

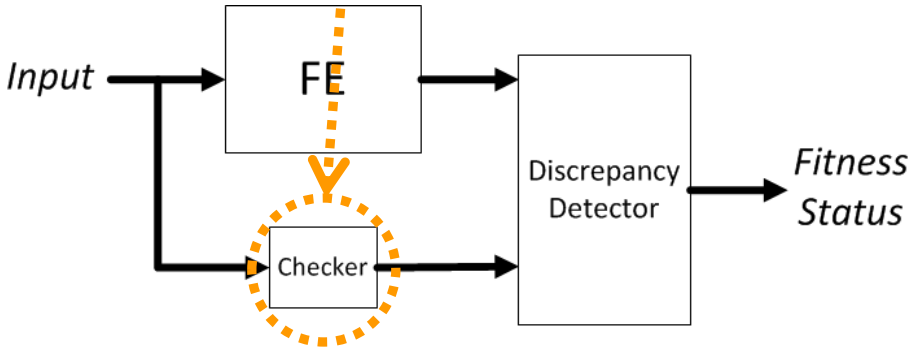
- Redundancy-based **Fault Detection** → mission critical applications
- A conventional ***Concurrent Error Detection (CED)*** technique → Two **identical** replicas of a given module are required to provide fault detection capability [1],[2]
- A ***Triple Modular Redundancy (TMR)*** *based design* → faults can be handled faults via majority **Voting** [3]
- The redundancy based schemes have **overhead** of resource and power [4]

Various Forms of CED for FPGAs

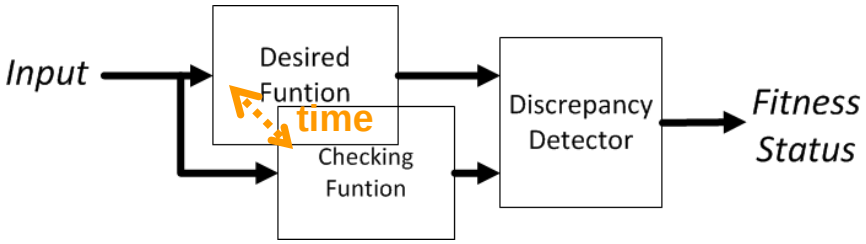
Conventional Concurrent Error Detection arrangement



***Heterogenous* Spatial** Concurrent Error Detection arrangement



***Heterogenous* Temporal** Concurrent Error Detection arrangement



Related Work

- A research of different ***forms*** of CED arrangement was presented by Mitra *et. al* [1] → e.g. **Parity-Based** systems
- Built-in Self Test (BIST) → **Detection Latency**
- *Temporal* redundancy techniques [11] → A typical error detection scheme involves running a duplicate thread for the comparison purpose on a ***chip multiprocessor (CMP)***

  → Effective for transient errors, but **permanents faults** are not detectable
- A ***resource testing*** scheme by Gao *et. al* [15] using time multiplexing of different components through the ***reconfiguration capability*** of FPGA → drawbacks of BIST along with the **reconfiguration times**

Salient Features of the Proposed Work

- As the **reconfiguration time** is a considerable entity in current FPGA technology, we propose to **multiplex** the inputs to a fixed hardware fabric instead of reconfiguring the resources with alternating functions.

Spatial *h*CED:

- Fault detection in Spatial *h*CED mode with **resource saving**
  → Area of Checker << Area of Functional Module

Temporal *h*CED:

- Fault detection in Temporal *h*CED mode with **uniplex chip area** requirement at the cost of reduced throughput.
  → Same fabric resources are time-multiplexed across different inputs
- The coverage of transients as well as **permanent faults** in the fault detection using Temporal hCED.
  → Since Checker design differs from Functional Module design, based on the inputs

Case Study - DCT hardware Core

- Common in **image/video compression applications**, and hardware implementation is highly desired in many applications due to parallel nature of the image processing related tasks and their **throughput requirements**
- **Video encoder application** → image frame is first divided into Macroblocks and the ***transform*** operation is performed on these Macroblocks
- **DCT Transformation** uses following kernels [16],[17]:

$$r(x,y,u,v) = \alpha(u)\alpha(v)cos\frac{(2x+1)u\pi}{2n}cos\frac{(2y+1)v\pi}{2n} \quad (2)$$

where

$$\alpha(u) = \begin{pmatrix} \sqrt{\frac{1}{n}} & for u = 0 \\ \sqrt{\frac{2}{n}} & for u = 1, 2, ..., n-1 \end{pmatrix}$$

A Necessary Condition for Fault-free DCT

The **1D DCT operation** on a row of pixels in a Macroblock is defined by:

$$Y = \Phi.X$$

Where

X = A row of input pixels macroblock.i.e., $\{x_1, x_2, ..., x_8\}$

Φ = Matrix of DCT kernels

Y = A row of output DCT coefficients.i.e., $\{y_1, y_2, ..., y_8\}$

As the DCT operation is a **linear transformation** from input pixels space to output coefficients space, we can make the following derivations:

$$(Y - \grave{Y}) = \Phi.(X - \grave{X})$$

$$\Delta X = X - \grave{X}\ ,\ \Delta Y = Y - \grave{Y}$$

$$\Delta Y = \Phi.\Delta X$$

The current DCT coefficients can be **estimated** from the previous coefficients using the difference values. $\hat{Y} = \grave{Y} + \Delta Y$

A **necessary condition** for a fault free DCT block can be written as:

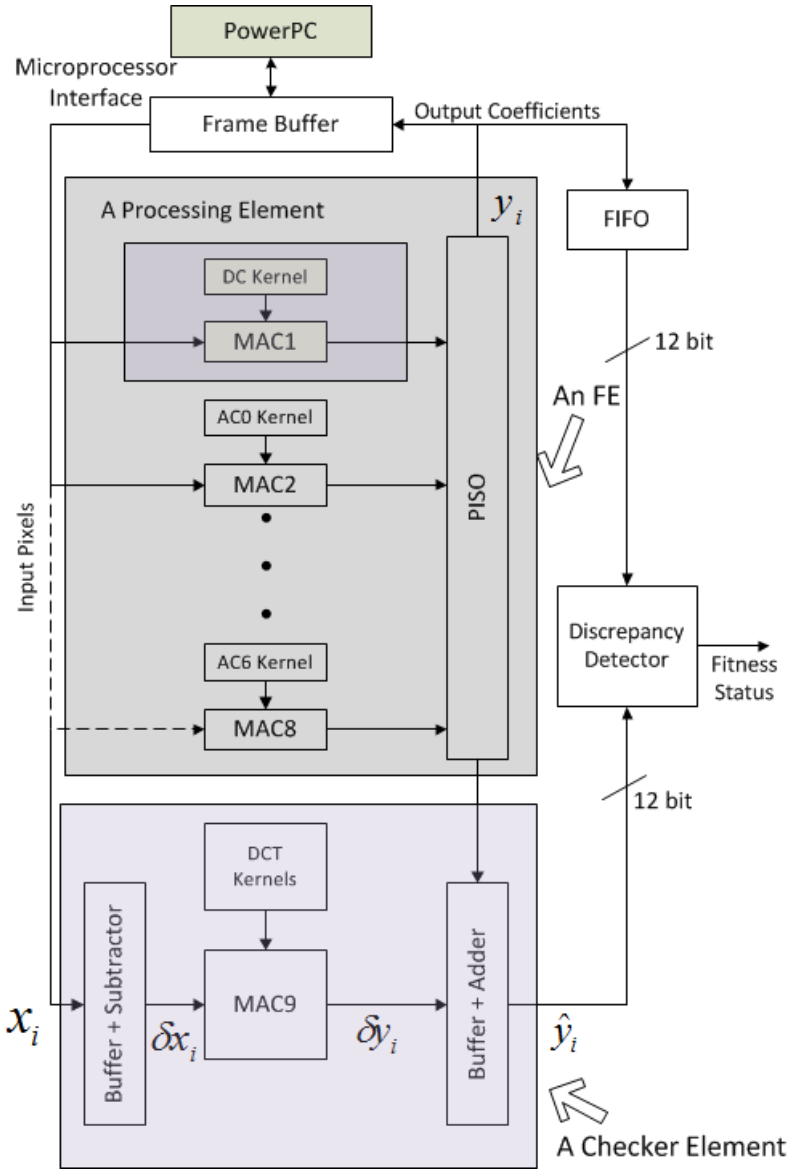
$$Y = \hat{Y}$$

Estimated = Actual

Baseline Arrangement

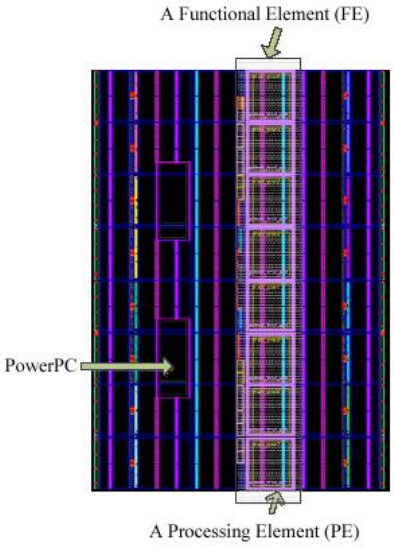
- An **8-point 1-D DCT** implemented in Verilog HDL using Xilinx ISE 9.2i development environment.
- Xilinx PlanAhead for the partial reconfiguration design
- Xilinx development board ML410 ,Virtex-4 FPGA.
- Multiple ***Processing Elements (PEs)*** used to compute DCT coefficients, **one coefficient per PE**
- A ***Functional Element (FE)*** consists of 8 *Processing Elements* (PEs), where a FE will compute the 8-point 1D DCT.
- For example, PE1 contains the **DC-kernel**, PE2 has the **AC0**-kernel and so on.
- The DCT core is interfaced with on-chip PowerPC microprocessor through GPIO core.

Spatial Heterogeneous CED

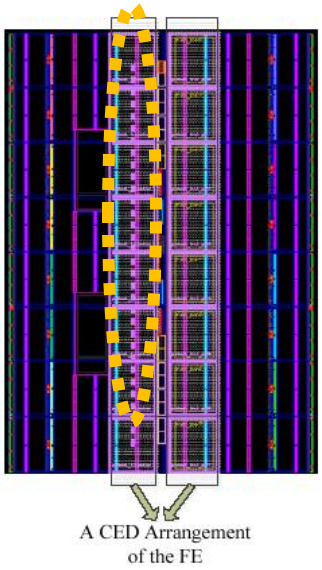


Compared to conventional CED Arrangement, the Checking Element contains a **single PE** *instead of 8 PEs.*

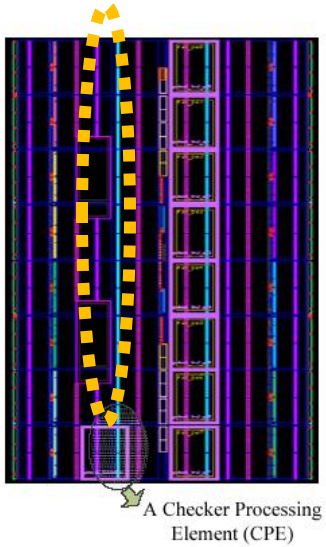
Spatial *h*CED



(a) An FE realizing the DCT function



(b) CED arrangement



(c) Spatial *h*CED arrangement

Spatial Heterogeneous CED

- PE1 through PE8 form an FE computing the DCT of input data.
- The PE containing a MAC9 serves as a ***Checker Processing Element (CPE)***. The CPE performs MAC operation on the difference input, instead of the input pixel values.
- The CPE utilizes the previous output from FE to predict the current output → discrepancy reveals error

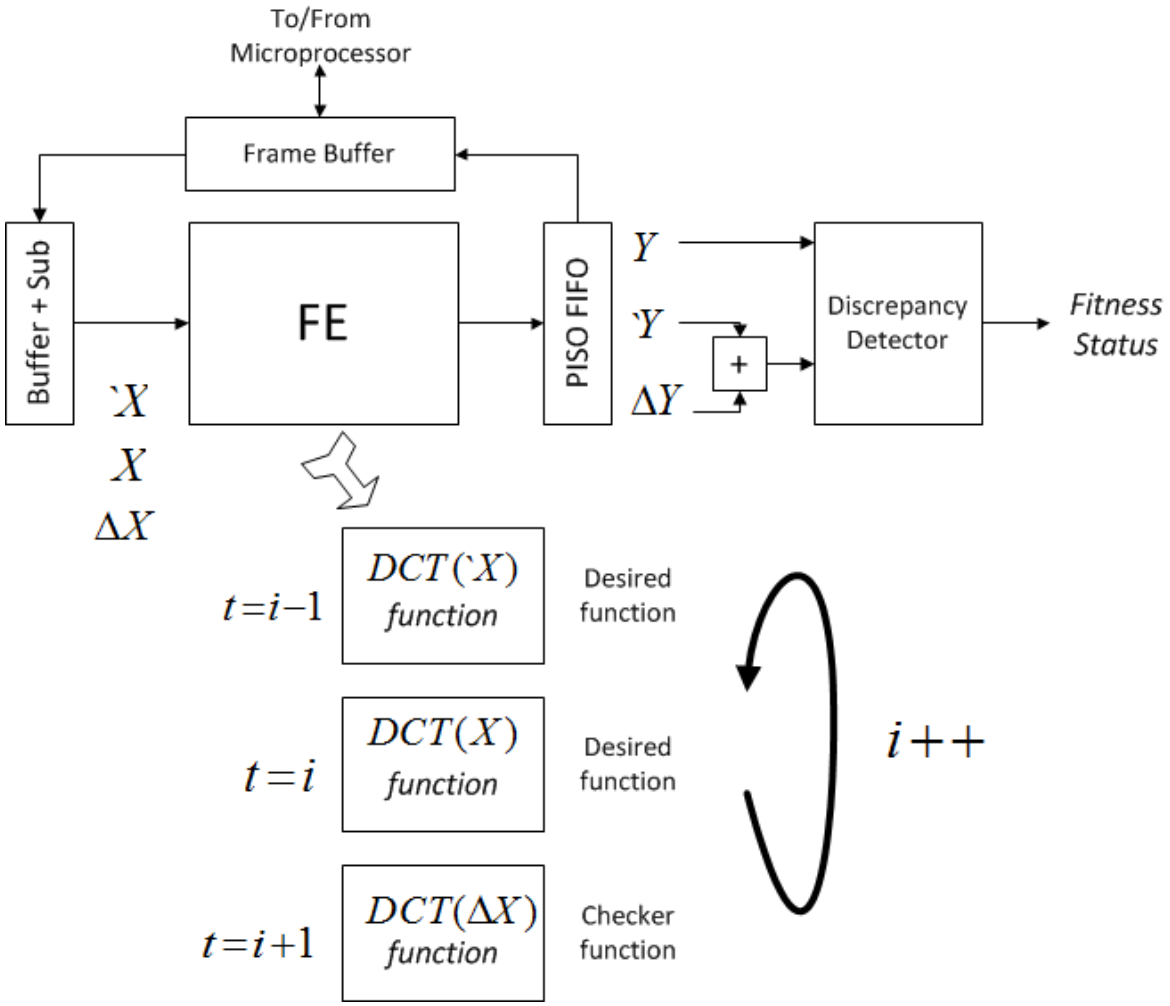
RESOURCE UTILIZATION FOR SPATIAL HCED ARRANGEMENT

Component	PE	FE	CPE
Number of slices	91	728	167
Number of slice FFs	46	368	75
Number of 4 input LUTs	161	1288	308

Spatial Heterogeneous CED

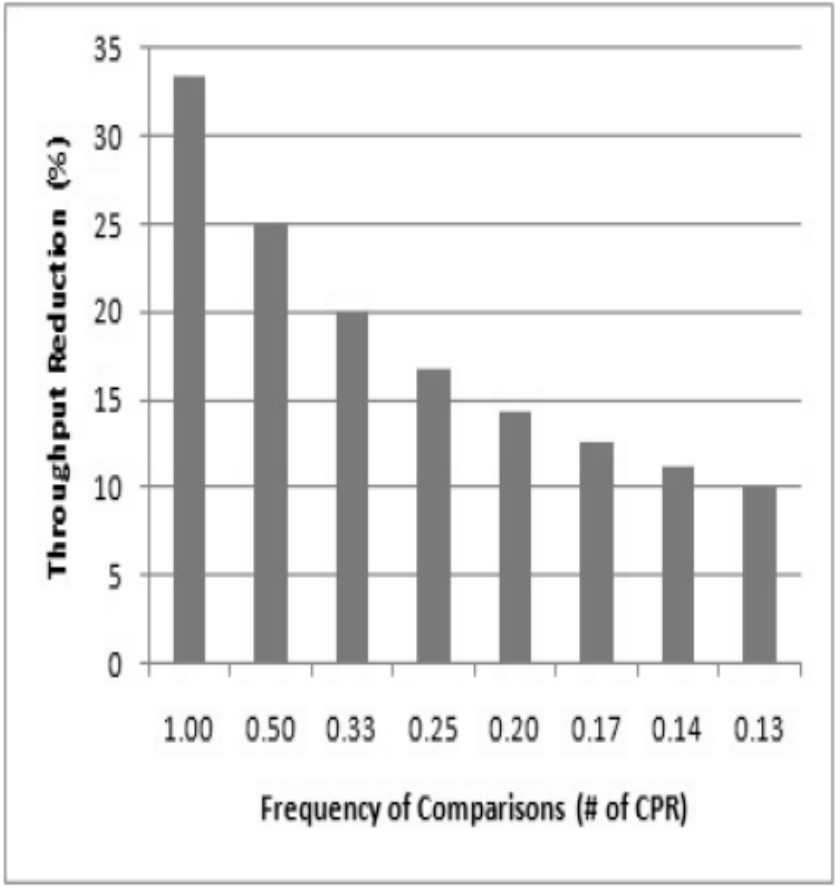
- The 8 PEs utilize 728 Configuration Logic Block (CLB) slices of the chip, while one CPE implementation requires only 167 slices, saving **overall 77%** of the FPGA resource.
- A ***conventional CED*** would require approximately **2X resource count**
- One PE requires an estimated power of 12.39 mW. On the other hand, a CPE's estimated dynamic power consumption is 13.33 mW using a clock frequency of 100 MHz → **Power saving 1/8th**
- Inject **Stuck-At faults** at LUT inputs and analyze their behavior in post Place-and-Route simulation model.
- Saving in resource and power at the expense of fault detection latency compared to conventional CED

Temporal Heterogeneous CED



Temporal Heterogeneous CED

- **Throughput reduction** of *Temporal hCED* arrangement
- For example, if one time instant out of 3 instants is dedicated for the prediction computation, the *effective throughput* would **reduce to 66.7%**
- By performing the redundant computation of difference input data **less frequently**, the useful throughput can be improved from the worst case.



Conclusions

- *h*CED provides a **flexible** fault detection scheme.
- The ***spatial hCED*** *form exhibits reduced resource* requirements than the conventional CED technique. Thus, **area and power** can be saved using the proposed approach at the cost of a negligible fault detection latency overhead.
- The ***temporal hCED*** *form* has error detection capability of fault coverage that includes **permanent faults** in logic resources, in addition to transient faults.
- The ***temporal*** error detection form has **uniplex area** requirement avoiding redundancy in the resources at the cost of throughput reduction.
- The *h*CED scheme appears to be readily applicable to any **linear transformation** that is pipelineable.

References

[1] S. Mitra and E. McCluskey, “Which concurrent error detection scheme to choose ?” in Test Conference, 2000. Proceedings. International, 2000, pp. 985 –994.

[2] S. Mitra, W.-J. Huang, N. Saxena, S.-Y. Yu, and E. Mc-Cluskey, “Reconfigurable architecture for autonomous selfrepair,” Design Test of Computers, IEEE, vol. 21, no. 3, pp. 228 – 240, May-June 2004.

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

[4] N. Imran and R. F. DeMara, “Cyclic NMR-based fault tolerance with bitstream scrubbing via Reed-Solomon codes,” in Presentations at the ReSpace/MAPLD Conference, Albuquerque, New Mexico, 2011.

[5] E. Stott, J. S. J. Wong, and P. Y. K. Cheung, “Degradation analysis and mitigation in FPGAs,” in Field Programmable Logic and Applications (FPL), 2010 International Conference on, pp. 428–433.

[6] P. Ostler, M. Caffrey, D. Gibelyou, P. Graham, K. Morgan, B. Pratt, H. Quinn, and M. Wirthlin, “SRAM FPGA reliability analysis for harsh radiation environments,” Nuclear Science, IEEE Transactions on, vol. 56, no. 6, pp. 3519 –3526, Dec. 2009.

[7] C. Bolchini and C. Sandionigi, “Fault classification for SRAM-Based FPGAs in the space environment for fault mitigation,” Embedded Systems Letters, IEEE, vol. 2, no. 4, pp. 107 –110, Dec. 2010.

[8] R. F. DeMara, K. Zhang, and C. A. Sharma, “Autonomic fault-handling and refurbishment using throughput-driven assessment,” Appl. Soft Comput., vol. 11, pp. 1588–1599, March 2011.

[9] R. F. DeMara and C. A. Sharma, “Self-checking fault detection using discrepancy mirrors,” in International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA05), Las Vegas, Nevada, 2005.

[10] 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, no. 1, pp. 11 – 33, Jan.-March 2004.

References (contd.)

[11] 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.

[12] 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, Dec. 2010, pp. 452 –457.

[13] D. Bouldin, "Enhancing electronic systems with reconfigurable hardware," Circuits and Devices Magazine, IEEE, vol. 22, no. 3, pp. 32 –36, May-June 2006.

[14] S. Dutt, V. Verma, and V. Suthar, "Built-in-self-test of FPGAs with provable diagnosabilities and high diagnostic coverage with application to online testing," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 27, no. 2, pp. 309 –326, Feb. 2008.

[15] M. Gao, H. Chang, P. Lisherness, and K. Cheng, "Time multiplexed online checking," Computers, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2011.

[16] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," Computers, IEEE Transactions on, vol. C-23, no. 1, pp. 90 –93, Jan. 1974.

[17] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed. Prentice Hall, Upper Saddle River, NJ., 2008.

[18] J. Huang, M. Parris, J. Lee, and R. F. Demara, "Scalable FPGA-based architecture for DCT computation using dynamic partial reconfiguration," ACM Trans. Embed. Comput. Syst., vol. 9, no. 1, pp. 1–18, 2009, 1596541.