# Heterogeneous Concurrent Error Detection (*h*CED) Based on Output Anticipation

Naveed Imran and Ronald F. DeMara
*Department of Electrical Engineering and Computer Science*
*University of Central Florida*
*Orlando, FL 32816-2362*
*Email: naveed@knights.ucf.edu, demara@mail.ucf.edu*

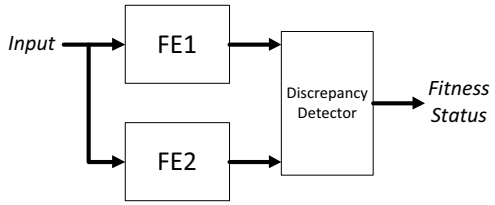
***Abstract*—A conventional *Concurrent Error Detection* (CED) technique usually relies on two exact replicas of a given module to provide redundancy in fault-tolerant systems. A discrepancy in one of the two instances flags at least one of them as *faulty*. We propose a heterogenous redundant FPGA-based system by exploiting the application properties. Consequently, the replicated module is not necessarily an exact copy of the original module but is much less resource and power hungry. In the paper, we discuss two forms of the heterogeneous structure which are spatial and temporal redundancy based. These forms are evaluated using FPGA based hardware implementation of the Discrete Cosine Transform (DCT) block. A necessary condition is derived to declare the DCT block as fault-free. The results show that the heterogeneous spatial redundancy can realize a resource efficient CED pair at the cost of a small latency in error detection. On the other hand, the heterogeneous temporal redundancy can provide permanent faults resource coverage at the cost of reduced throughput with negligible resource overhead.**

***Keywords*-fault-detection; FPGAs; partial reconfiguration;**

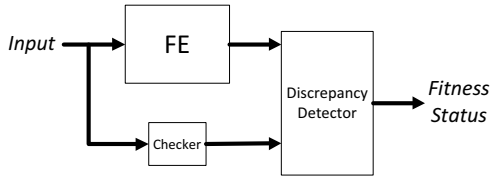
## I. Introduction

Redundancy based fault tolerant systems are common in mission critical applications. A CED scheme utilizes two replicas of a module to compute for the system output [1], [2]. A discrepancy in the output of two modules flags the system as faulty. A better approach in terms of fault capacity is *Triple Modular Redundancy* (TMR) based design in which three instances of a module concurrently operate on the same input. A majority voter is used to provide the system's main output by majority voting of the output values from the three instances. In this way, a faulty module can be easily identified as far as faults are manifested in only one of three instances. Faults in one of three modules do not cause system failure as faults can be masked via majority voting [3].

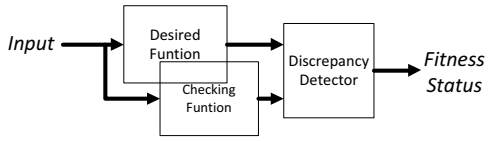
The redundancy based schemes have overhead of resource and power. A TMR and higher-MR systems [4] have better fault coverage and fault masking capabilities than a CED based system, at the expense of increased power consumption and the number of components requirement. In this work, our focus is CED setup and thus, fault detection in a fault tolerant system. A conventional CED setup is shown in Fig. 1(a), in which two exact replicas of a given module concurrently operate to compute for the same input.




(a) Conventional Concurrent Error Detection arrangement




(b) Heterogenous Spatial Concurrent Error Detection arrangement




(c) Heterogenous Temporal Concurrent Error Detection arrangement

Figure 1. Various CED configurations

In the figure, a *Functional Element* (FE) is the main required component of the system which provides the system's throughput. FE1 and FE2 are two replicas and their output is monitored by a discrepancy detector. In this way, one of the FEs can provide the system's output, whereas the other FE can be thought of as a *Checker* for the main FE.

Because the sole purpose of a CED setup is the error detection, we can reduce the checker size compromising some of its capabilities. For example, by marginalizing the checker's throughput, a considerable number of resources can be saved. Thus, the checker does not need to be an replica of the functional module. We present two forms of heterogeneous CED exploiting some properties of the functional module. In the spatial CED type (Fig. 1(b)), the

978-0-7695-4551-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ReConFig.2011.48

checker size is reduced by marginalizing its throughput at the cost of some increased fault detection latency. In the temporal CED form (Fig. 1(c)), the same hardware fabric is alternately switched between the actual function and the checker function. Thus, error detection is possible with uniplex resource requirement at the cost of some reduction in throughput.

Field Programmable Gate Arrays (FPGAs) are a suitable candidate as a platform for exploring fault handling techniques due to their flexible nature [5]. Various configurations of a design can be studied for throughput, power, and reliability analysis. Another reason for research interest in fault-tolerance of FPGA based design is due to their popularity in mission critical systems [6]. FPGAs are widely used in space applications, however they are susceptible to transient as well as permanent faults, for example *Single Event Upsets* (SEUs) in the configuration memory, and *Stuck-At* (SA) faults in the logic resource [7]. For fault detection capability, a duplex of the design can be instantiated on the chip and a discrepancy in the output can be monitored via a discrepancy detector [8], [9].

## II. Related Work

Fault tolerant systems are characterized by the reliability and dependability they provide in mission critical systems. The *detectability* is an important attribute of the fault tolerant systems by which faults or a system failure can be detected [10]. An indication is required in situation of faults when the output of the system deviates from its desired operation. While the detectability can be implemented by observing the behavior of a system through certain variables, another way is to replicate the system to realize a duplex. A disagreement in output of the two instances indicates faulty nature or an error in at least one of the two instances.

A research of different forms of the CED setup was presented by Mitra et. al [1]. The CED schemes rely on some form of redundancy for fault detection purpose. Sometimes, the functional modules are implemented in the form of a diverse duplex system which involves two alternative implementations of the same design. This helps error detection in case of common mode failures. Besides duplex systems, CED schemes also realize parity based systems [1]. For example, an even/odd parity can be used to ensure the correctness of an output sequence of a digital system.

Temporal redundancy techniques have been explored in fault-tolerant microprocessor systems. A typical error detection scheme involves running a duplicate thread for the comparison purpose on a chip multiprocessor (CMP). Hyman et. al [11] proposed an extension to the scheme by exploiting various redundancies in instructions in multi-core processors framework. Thus, if an instruction is affected by transient errors in the execution path, the duplicate execution would provide a fault detection capability. However, if the faults are of permanent nature, the re-execution of an instruction would have the same result and the error detection becomes impossible. This is because a given input data will exercise the logic resource in the same way no matter how many times an instruction is executed. In our approach, we apply the desired function to the input data at one instant, and the redundant computation is performed on the difference data in the second instant. Thus, the logic resource is exercised with different input data in each case. In this way, the approach is able to detect errors in case of permanent faults also, in addition to transient errors.

FPGAs are widely used in signal processing, image processing and video applications [12] due to their parallel nature. In addition, the reconfiguration capability [13] provides flexibility in exploring different hardware architectures. The dynamic reconfiguration of FPGA resources can be performed in a fault handling scheme to avoid the faulty resources.

Built-in Self Test (BIST) is a generally used technique to test the logic resources to identify possible faults. Dutt et. al [14] proposed an online BIST technique for FPGAs to detect and isolate faults of permanent nature. BIST techniques are characterized by the fact that fault detection latency may be long depending upon the chip area. Moreover, transient errors are not detectable in these schemes. In our approach, by introducing some redundancy for error checking purpose, the transient errors are also detectable with a negligible fault detection latency. Gao et. al [15] proposed a resource testing scheme using time multiplexing of different components through the reconfiguration capability of FPGA. As the reconfiguration time is a considerable entity in current FPGA technology, we multiplex the inputs to a fixed hardware fabric instead of reconfiguring the resources with alternating functions.

In the context of previous work, the followings are the key-points of the presented work.

- Fault detection in spatial *h*CED mode with resource saving.
- Fault detection in temporal *h*CED mode with uniplex chip area requirement at the cost of reduced throughput.
- The coverage of transients as well as permanent faults in the fault detection using temporal *h*CED.

## III. Alternate CED Arrangements

As a case study, we take a DCT hardware core to evaluate two forms of the heterogeneous CED. DCT function is widely used 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. One example is video encoder application in which an image frame is first divided into macroblocks and the transform operation is performed on these macroblocks.

$$\phi = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

Figure 2. The DCT matrix

The DCT was introduced by Ahmed et. al [16] in 1974, when an efficient computation of Fourier Transform (FT) was desired. They defined the DCT of a data sequence as:

$$G_x(0) = \frac{\sqrt{2}}{M} \sum_{m=0}^{M-1} X(m)$$

$$G_x(k) = \frac{2}{M} \sum_{m=0}^{M-1} X(m) cos \frac{(2m+1)k\pi}{2M} \tag{1}$$

where $G_x(k)$ is the $k$-th DCT coefficient [16]. As a digital image is represented in computers via 2-D matrix notation, it is often desired to define 2-D transforms for image processing tasks. The DCT is used to represent an image by sum of varying sinusoidal amplitudes and frequencies. After the transform operation, the information content of a natural image is usually concentrated in only first few coefficients of the DCT. Due to this property, the DCT can be used 1) for image compression because the information can be contained in fewer coefficients. 2) for pattern recognition as the feature size to be used in classification can be reduced considerably. Moreover, scale and rotation variance are easier to handle in frequency domain than that in pixel domain. Gonzalez and Woods [17] represent the 2-D DCT operation as:

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

where

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

The details are given in their book [17].

The DCT matrix is commonly calculated for 2-D input data samples using the equation, and is given in Fig. 2.

Our current focus is fault detection of the DCT block. In the following, we derive a sufficient condition to mark a DCT hardware block as *faulty*.

The 1-dimensional DCT operation on a row of pixels in a macroblock is defined by:

$$Y = \Phi.X \tag{3}$$

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

$$(Y - \grave{Y}) = \Phi.(X - \grave{X}) \tag{4}$$

where $\grave{X}$ and $\grave{Y}$ are the vectors for the previous time instant. Let's define:

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

then

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

The current DCT coefficients can be estimated from the previous coefficients using the difference values. i.e.,

$$\hat{Y} = \grave{Y} + \Delta Y \tag{5}$$

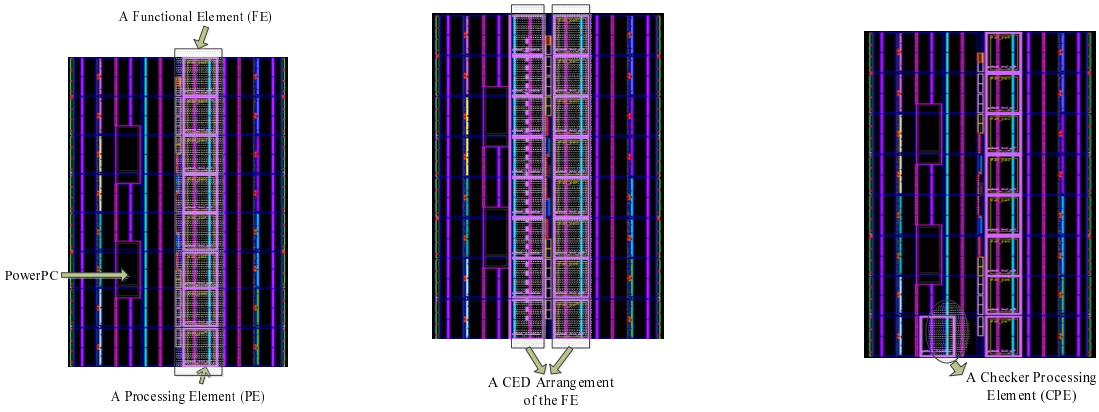
Thus, a necessary condition for a fault free DCT block can be written as:

$$Y = \hat{Y} \tag{6}$$

If the equality is not met, the DCT block may be flagged as faulty. It may be noted that this is a necessary condition, not sufficient. Thus, even if the equality in the above expression is met, it does not imply fault-free nature of the module. Manifested fault in the logic or routing resource of the FE renders it faulty. Manifested faults are those faults which affect the output of any FE. We compute the output $Y$ of DCT module through the FE, and the predicted output $\hat{Y}$ through the *Checker*. Thus, the checker serves as a predictor of the output that is desired at the module output for its correct operation.

## IV. The Baseline Setup

An 8-point 1-D DCT is implemented in Verilog HDL using Xilinx ISE 9.2i development environment. The place-and-route report shows that the design can run up to a clock frequency of 108 MHz. An FE consists of 8 *Processing Elements* (PEs), where each PE computes one DCT coefficient of a row of input pixels. Through the pipelining scheme, we are able to output one DCT coefficient every clock cycle. The DCT kernels are stored inside these PEs using the Look-up Tables (LUTs) of the FPGA chip. For example, $PE_1$ contains the DC-kernel, $PE_2$ has the $AC_0$-kernel and so on. Internally, the PEs use Multiply-Accumulate (MAC) units to perform the dot product of input data with the kernels. To provide scalability in the implementation [18], different Partial Reconfiguration Regions (PRRs) are defined to accommodate the PEs. Xilinx PlanAhead is used for the partial reconfiguration design, and ExploreAhead is used to




(a) An FE realizing the DCT function (b) CED arrangement (c) Spatial *h*CED arrangement

Figure 3. Floorplan of various FE configurations realizing a DCT module

generate partial bitstreams. The design is tested on a Xilinx development board ML410 which has a Virtex-4 FPGA. We report the utilized resources in the next section when the resource requirement in two forms of the CED is compared.

The DCT core is interfaced with on-chip PowerPC microprocessor through Xilinx provided GPIO core. The PowerPC writes the fixed-point format pixels data to the transposition memory. Also, it generates the control signals for the hardware DCT controller to sequence reads and writes operations from the frame buffer (transposition memory). The DCT controller manages to perform the 1-D operation on the pixels data row-wise in the first stage of the DCT, whereas the 2-D operation is accomplished by repeating the DCT operation on the input data column-wise. The completion of DCT operation on a macro-block is acknowledged to the PowerPC through GPIO core after which the processor can read the DCT coefficients from the transposition memory. A dual port RAM is instantiated to serve as the frame buffer. The physical layout of the design is shown in Fig 3(a). An FE realizing the DCT function and containing 8 PEs, is highlighted. A conventional arrangement of the CED is realized by the placement illustrated in Fig 3(b).

## V. Spatial Heterogeneous CED

The architecture of the spatial *h*CED is given in Fig. 4. In the context of Fig. 1(b), PE1 through PE8 form an FE computing the DCT of input data. On the other hand, the PE containing 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 of the FE. If a discrepancy is observed between the current output from FE and the predicted output from CPE, it gives an indication of fault(s) in one of the two elements.

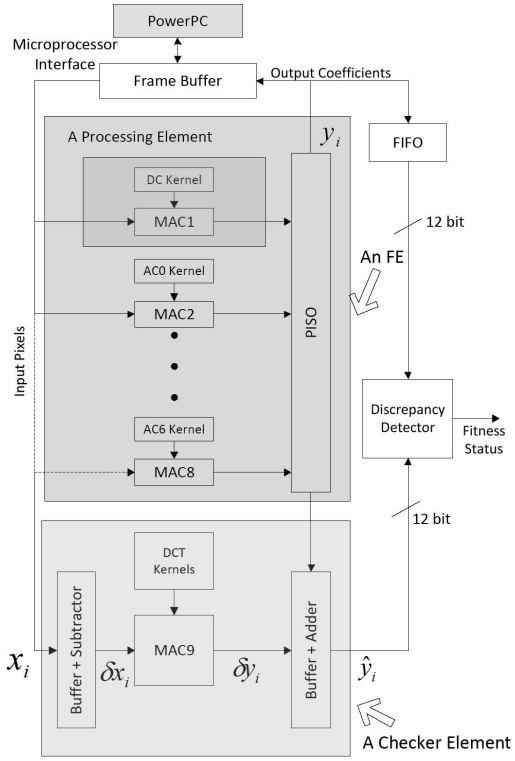



Figure 4. Spatial Heterogeneous CED arrangement realizing a DCT module

All the DCT kernels for the checker unit are stored inside the CPE. The placement layout of the processing units is given in Fig. 3(c). The CPE computes the DCT coefficients sequentially unlike the FE which computes all the 8 coefficients in parallel owing to the contained multiple PEs. Thus, one DCT coefficient is available at the checker's output

Table I
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 |

after every 8 clock cycles which can be compared with that residing in the output buffer of the FE. A discrepancy between the predicted coefficient and the actual coefficient indicates an error.

Table I lists the resource requirement; the chip resources utilized by a PE are given in the first column, the second column provides a resource count of the 8 PEs in the DCT module. A conventional CED would require two times of this resource count. By using a CPE instead of a replica, a considerable amount of chip resources is saved as evident in the third column, which in turn saves power. 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. The resource requirement of a CPE is not $\frac{1}{8}$th of that of an FE as all the DCT kernels are stored in the CPE. On the other hand, each PE of the FE contains one kernel only.

It may be noted here that this is not very area-efficient implementation of the DCT core. Our objective is to evaluate the fault detection methodology. We avoid the Xilinx built-in hard multipliers known as DSP48 blocks, and employed LUTs while synthesizing the design as a generic implementation is desired to be analyzed for error detection purpose. Moreover, we can inject SA faults at LUT inputs and analyze their behavior in post place-and-route simulation model.

An analysis of the design by using Xilinx XPower tool reveals that 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 the clock frequency of 100 MHz. As a CED arrangement will require a total of 8 PEs to serve as a checker, therefore the power requirement for the checker is approximately $\frac{1}{8}$th which uses a CPE.

This saving in resource and power, however, comes at a cost. Because of the sequential nature of the CPE, number of comparisons for the discrepancy check that can be made in a given time-period, are reduced by a factor of 8. The frequency of comparisons defines the latency of detection in case of fault occurrence. By increasing the processing units for the checker module, the fault latency can be improved. It may be a worth to be noted here that even a detection latency of 64 clock cycles may be negligible as the design is running at 108 MHz, especially when the resource saving is considerably large. The latency of fault detection also depends upon the location of faults and input data. Although, *single-event upsets* could be missed when using reduced sampling rate for error detection, they can incur only transient noise in the output image.

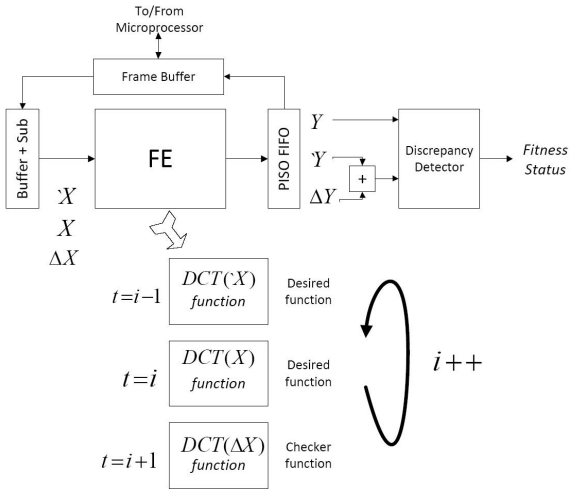



Figure 5. Temporal Heterogeneous CED arrangement of the DCT module

## VI. TEMPORAL HETEROGENEOUS CED

In this form of the *h*CED, temporal redundancy is used for fault detection in DCT module. Instead of applying a repeated function on the same input, we alternately apply the same function on two different types of inputs. In the first instant, the DCT computation is performed on the input pixels set. In the next time slot, the same computation is performed on the difference values. In this way, the hardware fabric is time multiplexed between two types of inputs. In this two shot operation mode, even permanent faults may be manifested as the diverse inputs exercise the underlying logic resources in a different way.

Fig. 5 shows the temporal *h*CED setup. At the first time instant (t=1), FE computes the DCT coefficients ($\dot{Y}$) for the given input row of pixel values ($\dot{X}$). At the second instant (t=2), the DCT coefficients ($Y$) are computed for the next row of pixels. At the instance (t=3), same FE is used for the DCT computation of the $\Delta X$, which is a difference between two consecutive rows of input pixels. The output $\Delta Y$ is used as an alternative computation (i.e., prediction $\hat{Y}$), which can be used for the discrepancy check.

If we dedicate one time instant out of 3 instants for the prediction computation, the *effective* throughput would reduce to 66.7%. It means that 2 of the three computations are providing the actual desired output of the DCT module. By performing the redundant computation of difference input data less frequently, the useful throughput can be improved from the worst case. Fig. 6 shows the overhead in terms of throughput reduction. The plot provides the frequency of comparisons in terms of *Comparisons per Row* (CPR) vs. throughput reduction. As it is evident, reducing the frequency of redundant computations for the comparison

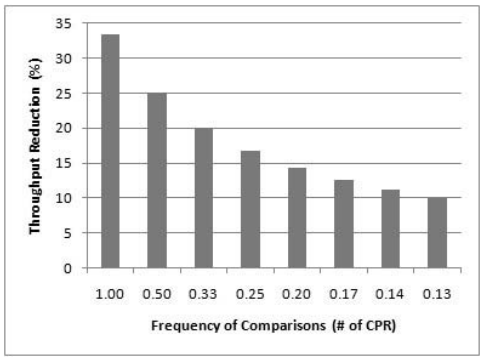



Figure 6. Throughput reduction of Temporal Heterogeneous CED arrangement

purpose, the throughput reduction can be improved.

## VII. Conclusions

We proposed two forms of the *Concurrent Error Detection* method of fault detection in FPGA-based designs. The *spatial heterogeneous CED* 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 heterogeneous CED* form's error detection capability of fault coverage includes permanent faults in logic resources, in addition to transient faults. Moreover, the temporal error detection form has uniplex area requirement avoiding redundancy in the resources. It has the capability to manifest permanent faults due to diverse inputs. These results are promising in the sense that fault coverage is enhanced with negligible resource overhead at the cost of reduced throughput. While DCT is used as a case study, this work readily appears to be applicable to any linear transformation that is pipelineable. As a future extension, the latency of fault detection in spatial mode may be improved by randomly checking some of the coefficients. An important area of future work is the derivation of necessary fault-free conditions for a generic FE design.

## 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. McCluskey, "Reconfigurable architecture for autonomous self-repair," *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.

[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.