Error tolerance: Why and how to use slightly defective digital systems*

Melvin Breuer, Keith Chugg, Sandeep Gupta, and Antonio Ortega

Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089
mb@poisson.usc.edu; chugg@usc.edu; sandeep@poisson.usc.edu; ortega@sipi.usc.edu

ABSTRACT

We provide an overview of the notion of error tolerance and describe the context that motivated its development. We then present a summary of some of our case studies, which demonstrate the significant potential benefits of error tolerance. We present a summary of testing and design techniques that we have developed for error tolerant systems. Finally, we conclude by identifying shifts in paradigm required for wide exploitation of error tolerance.

I. BACKGROUND AND MOTIVATION

The notion of error tolerance is motivated by three important trends in information processing, namely changes in fabrication technology, changes in the mix of applications, and emergence of new paradigms of computation.

Fabrication technology: As we get closer to what some call the "end of CMOS", we see the emergence of highly unreliable and defect-prone technologies. This is accompanied by rapid development of new computing technologies such as bio, molecular, and quantum devices. Most of these new technologies are also extremely unreliable and defect-prone (e.g., see [12]). However, these new technologies also provide the ability to carry out massive numbers of computations in parallel and at speeds that far exceed those currently achieved by CMOS devices.

Applications: Increasingly larger fractions of the total number of chips fabricated in any given year implement multi-media applications and process signals representing audio, speech, images, video and graphics. The outputs of such systems eventually become input signals to human users. There are several interesting aspects to the computational requirements for such systems.

1) The result of computation, i.e., the output data, is not measured in terms of being right or wrong, but rather on perceptual quality to its human users. For example, in the case of an image the perceptual quality may be defined in terms of absence of visible artifacts, clarity, color and intensity. In other words, the criterion is not correctness but whether the end product is acceptable to the human user.
2) Most such systems are by design *lossy*, in the sense that the outputs deviate from perfection due to sampling of input signals, conversion to digital, quantization, lossy encoding, decoding and conversion to analog signals.
3) Many such applications require parallel architectures as they are *computationally intensive* and have *real-time performance constraints*.

Emerging paradigms of computation: Several new paradigms are emerging on how functions are computed and what requirements are placed on the "correctness" and "accuracy" of the results. With tongue in cheek, in our school systems 5+7=13 is not considered to be "wrong," but rather "that is close Jimmy." Increasingly this is also the case for many emerging computation paradigms, which carry out computations somewhat differently than classical computations carried out for applications like bookkeeping and flight control systems. Consider the following paradigms.

Evolutionary computation is a simplified attempt to solve a problem based on several analogies made with evolution as it occurs in biological systems. One important aspect of such heuristic computations, and of many other heuristics, can be summarized as in [14]: "Evolutionary computing deals with the process where 'a computer can learn on its own and become an expert in any chosen area.' Such systems often rely on neural nets for their implementation. The process can adapt over time, e.g. one can modify the score function."

Neural nets also define acceptability of the results of computation in a similarly less stringent manner. For example [25] states: "Neural nets typically provide a greater degree of fault tolerance than von Neumann sequential computers because there are many more processing elements, each with primarily local connections. Damage to a few elements or links thus need not impair the overall performance significantly."

Approximate computations: In [24], Partridge states that a challenge is "to develop a science of approximate computation and derive from it a well-founded discipline for engineering approximate software. In order to meet this challenge, a radical departure from discrete correct/incorrect computation is

*This research was supported by National Science Foundation (0428940)

required, a shift away from logics towards statistical foundations, such that meaningful estimates of 'confidence' emerge with each approximate result. This implies that probabilities play an integral part in computation throughout the process. The component probabilities and the eventual confidence estimates, if secured by large numbers (e.g. repeated sampling from a proposed distribution), imply a computational effort that is becoming increasingly feasible as a result of hardware advances as well as innovative developments in statistical modeling theory (e.g. reversible-jump Markov Chain Monte Carlo methods)."

Probabilistic computations: In [3] and [4] as well as [32], researchers have also focused in part on a new probabilistic computing paradigm. Here the motivation is to reduce energy, and is based on the thesis that the energy consumed by a computation is proportional to the associated accuracy. In [3] and [4], accuracy is measured as the probability of a result being correct. This work shows that by building, in hardware, probabilistic switches that compute with a fixed probability of error, one can create complex architectures that implement probabilistic algorithms that operate at a fraction of the energy needed by conventional hardware**.**

As computing becomes more pervasive, we will soon be engulfed with digital devices in our homes, cars, workplace and even our clothing. In [31] it is stated "E-textiles inherently have high defect rates, as well as high fault rates and, thus, must by necessity provide mechanisms for extracting useful work out of the unreliable substrate."

This paper is organized as follows. In Section II we provide an overview of the notion of error tolerance. Section III presents a summary of some of our case studies, which demonstrate the potential benefits of error tolerance. Sections IV and V discuss testing and design techniques for error tolerant systems, respectively. Section VI presents our conclusions.

II. The notion of error tolerance

Due to the abovementioned emerging trends, it is critical to develop new paradigms for digital systems design and test that deal efficiently with high levels of ***imperfections***, i.e., process variations, defect densities, and noise sensitivity, inherent in modern CMOS and new technologies. All existing digital system design approaches, including defect and fault tolerance (DT and FT, respectively), strive to provide error-free system outputs. For large classes of systems, including those described above, which deem certain errors at outputs as acceptable, the application of DT or FT approaches will lead to unnecessarily inefficient designs.

With the goal of mitigating such inefficiencies, we define an ***error tolerant*** (ET) system as a system where certain types of errors at system outputs can be tolerated, i.e., deemed acceptable, provided the severities of different types of errors are below certain levels.

At the system level, we have used ***error metrics*** such as PSNR, distortion, and bit error rate to characterize the severities of errors. At the level of logic blocks, so far we have concentrated on two types of error metrics, namely error significance and error rate. An ***error significance*** metric quantifies the amount by which the response at a circuit's outputs, interpreted as a numeric or coded symbol, deviates from the corresponding error-free response. (The outputs can be divided into busses and error significance computed separately for each bus.) A significance metric can use one of many measures of deviation, such as the signed numerical error, the absolute value of a numerical error, the percentage numerical error or the Hamming distance. ***Error rate*** is defined as the asymptotic value of the fraction of a large number of functional responses from a circuit that are erroneous. (This metric can also be computed separately for each output bus.)

The characteristics of the application determine what types and specific combinations of metrics are appropriate. Analysis of the entire application also determines the limits for acceptability, or ***thresholds***, on each error severity metric by itself or for each appropriate combination.

The first task we undertook after we conceptualized the notion of error tolerance was to analyze a wide range of applications to determine their inherent abilities to tolerate errors. In particular, we were interested in estimating the potential benefits – higher yields, lower costs, lower power, and/or higher performance – that may be enabled by the exploitation of the notion of error tolerance. Once our initial case studies demonstrated potential for significant benefits for a wide range of high-volume applications, we started developing design and test approaches that exploit error tolerance to provide these benefits in the era of fabrication processes with high variations and high defect rates.

III. Case studies to demonstrate potential benefits of error tolerance

We studied several systems, including MPEG encoders (its motion estimation (ME) and DCT sub-systems), JPEG encoders, digital answering machines and hardware decoders for modern (turbo-like) error control codes. We primarily focused on hard faults, including single and multiple stuck-at faults in logic gates, interconnects and memory cells. For MPEG encoder, we also studies soft errors, in particular those induced by reduction of power supply voltage to levels where the circuit delay exceeds the clock period. Finally, for some systems, we studied different designs to identify the impact of the choices made during algorithm-, architecture- and circuit-level designs. These case studies are detailed in [5]-[11], [13], [22], and [33]-[37].

A. Approach used for our case studies

We start by outlining and illustrating a framework for determining the applicability of error-tolerance to a system. We will use a digital telephone/answering device (DTAD) as the vehicle for this study. We assume that there are defects in the memory of the device and we will investigate the relationship between defect density and acceptable performance.

A DTAD is usually programmed to answer and record a call. At the front-end, an ADC samples and quantizes the caller's speech, a codec encodes the speech, and the output bit-stream of the encoder is stored in a flash memory. When a user wants to listen to a recorded message, a microcontroller extracts the encoded speech stored in the flash memory, a codec decodes the data and finally the output speech is produced. Due to sampling, quantization and lossy compression, the quality of the output speech is less than that of the original speech. In terms of error-tolerance, one key question of interest is: what, if any, defects in the flash memory of a DTAD result in acceptable performance?

A key element in our methodology for the synthesis and analysis of error-tolerant systems deals with determining acceptable performance. For our DTAD example we are lucky that there exists a database of original speech patterns and automated measures for determining the quality of output speech, which is often measured in terms of a mean opinion score (MOS). Luckily, a perceptual evaluation of speech quality (PESQ) algorithm exists that can process speech and accurately estimate a MOS.

In our experiments we considered several codec algorithms, but here we report only on the G.723.1 dual-rate codec that is part of the H.324 multimedia compression and transmission standard. One frame contains 158 bits and represents approximately a 12:1 compression.

We modeled defects using a multiple stuck-at fault model. The chance that a bit is stuck-at-1 or stuck-at-0 is 50/50. Faults are randomly allocated through the memory based upon a uniform distribution. The probability that a fault results in an error is assumed to be 0.5. The fault density is defined as the ratio of the number of faults and the size of the flash memory. Twenty different fault densities between 0% and 1% were simulated. For each fault density, 50 different random distributions of faults were considered.

The results for G.723.1 produced a MOS score of 3.6 when there were no defects in the memory, and a monotonically decreasing score as the defect density increases. We considered any MOS score above 3.0 to produce acceptable results. This 3.0 score occurs for a defect density of about 0.2%.

Thus, during the test process for the flash, if we can determine the defect density we can bin these memories as to their predicted quality of service in a DTAD. However, there are other ways of sorting these devices, such as the one discussed next that is based on the concept of sensitivity.

The encoder converts frames of sampled speech to an encoded bit-stream that represents speech parameters. Faults in the flash memory corrupt these parameters, thus degrading the quality of the decoded speech. Each bit's contribution to the decoded speech quality can be different. Thus, the number of faults and the actual distribution of these faults in a flash memory are two factors that affect the degree of degradation of the output speech quality. Usually more faults cause more degradation. Flash memories with the same number of faults but different fault distributions can have different output speech attributes. Analyzing bit sensitivity gives a better understanding of how a fault's location affects speech quality.

For simplicity, we first determined the impact of a single error on the MOS as a function of the location of the error in a frame. We then derived an expression that assigns a score to a flash based upon the positions in the memory that are defective. Based on this score, we were able to accurately sort about 57% of our test cases into memories that result in either acceptable or unacceptable performance. For the remaining 43% of the chips, functional tests (MOS) are used for binning.

In summary, the important components of our proposed methodology for the analysis of the applicability of error-tolerance for a system are delineated below, along with the corresponding specific component illustrated by our DTAD example, shown in parenthesis.

1) Identify a **domain** where error-free computation is not essential. (Audio reproduction.)
2) Identify a **target** subsystem. (DTAD.)
3) Identify a **circuit** in the target subsystem that is a good candidate to be defective. (The flash memory.)
4) Identify a class of **defects** that are likely to occur. (Multiple stuck-at faults.)
5) Identify qualitative attributes and corresponding quantitative values to **specify levels of acceptability for the system**. (MOS; others attributes for different domains include temperature, clock-rate, latency, throughput, dB loss, distortion, skew, overshoot and undershoot, error-rate and error-significance.)
6) Identify a method to **determine the acceptability level** of the output of the system. (PESQ MOS algorithm; methods exists to quantify the quality of other attributes such as blur, color, jitter, intensity, resolution, smell and clarity.)
7) (Optional) Identify a method to **determine levels of acceptability** at the subsystem or circuit level. (Sensitivity values; for other situations, sensitivity analysis is commonly used to determine the effect of noise in one part of a system on another part.)
8) Determine a **test methodology** to partition circuits into levels of acceptability. (Standard speech patterns for PESQ MOS; classical memory BIST/ATE for sensitivity score; classical ATPG, DFT and BIST; other test methods include threshold testing and error-rate testing.)
9) Determine a method to **enhance** the effective yield. (Not addressed in this study; error correction codes.)
10) Determine a way to predict the **yield enhancement** due to error-tolerance. (Not addressed in this study.)
11) Analyze the **financial and marketing aspects** of this application of error-tolerance. (Not addressed in this study.)

B. Summary of results from our case studies

Our case studies on MPEG and JPEG encoders, answering machine, and decoders for turbo codes clearly support three main conclusions.

1) The notion of error tolerance can indeed provide significant benefits for a wide range of applications.

For example, in the video coding case, our experimental results, which are consistent with our analytical conclusions, show that more than 99.2% of single stuck-at faults within some ME implementations result in less than 0.01dB degradation in output video quality [5], [6], [13]. Similar results can be seen for the widely used discrete cosine transform (DCT), e.g., 50% of single stuck-at faults in a given architecture lead to acceptable degradation (less than 0.01% of image blocks show any perceptually observable degradation) [8], [9].

2) The notion of error tolerance can be applied in different ways to provide benefits with respect to a wide range of criteria, including yield, power and delay.

For example, we designed an MPEG encoder that combines two different ME strategies, each having different characteristics with respect to computation errors. For this encoder design, aggressive power supply voltage reductions, i.e., reductions which cause erroneous outputs, provide 10% power savings with essentially no rate penalty and up to 37% power savings with a small, 3%, rate penalty [10], [11].

3) The characteristics of algorithm-, architecture-, and circuit-level designs have significant impact on the benefits that error tolerance can provide. In particular, as a rule-of-thumb, parallel architectures provide significantly higher benefits than lower-cost serial ones.

For example, while the enhanced predictive zonal search (EPZS) and full search (FS) algorithms for ME for MPEG encoding are comparable under fault-free conditions (e.g., a 0.01dB difference), we showed that EPZS performs significantly better in the presence of faults (e.g., up to 2.5dB gain in some cases) [5], [6]. EPZS provides added error tolerance because algorithmically the search is constrained to a subset of motion candidates, so that even the worst case error leads to an acceptable motion vector choice. Also different hardware architectures performing the same metric computation result in significant variations in error tolerance. In particular, the optimal ME hardware architecture in terms of error tolerance is a balanced binary tree structure, which also enhances parallelism. This architecture can reduce the expected error due to a fault by up to 95% compared to other architectures [6].

IV. Development of Test Techniques for Error Tolerant Systems

One key way in which the notion of error tolerance can be exploited is by developing new test techniques that will increase yield. While classical test techniques strive to identify and reject every chip with an imperfection, i.e., a defect or variations that induces erroneous operation, our new test techniques focus on identifying and rejecting every chip with an imperfection that causes unacceptable errors. In addition to imperfection-free chips, such a new test approach exploits error tolerance to improve yield by enabling us to sell chips that have an imperfection that only causes acceptable errors.

We have developed an extensive suite of such test approaches. In particular, we have developed new approaches to generate and apply test vectors that enable testing for logic blocks where error tolerance is defined in terms of error significance and error rate. Some of our techniques assume the existence of a fault model, and others deal only with the existence of errors. Finally, we have developed built-in self-test (BIST) approaches as well as approaches that use automatic test equipment (ATE). These test approaches are described in [16]-[21], [23], and [26]-[29].

Our effort in this direction establishes the following key results.

1) It is indeed possible to develop testing approaches that can exploit the notion of error tolerance to significantly improve yield.

In particular, we have shown that it is possible to identify and reject every chip with an imperfection that causes any unacceptable error – defined in terms of error significance or error rate – while selling every chip with an imperfection that only causes acceptable errors.

2) It is possible to achieve this type of testing at test application costs that are close to the costs for classical testing. This shows that we can accrue the yield benefits of ET without significant increase in costs.

For example, for error significance testing, for a wide range of circuits, the cost of proposed testing approaches is within 20-50% of that for classical testing approaches. Similarly, the BIST approaches can make use of the same hardware used for classical BIST, and can execute in a comparable amount of time.

V. Development of Design Techniques for Error Tolerant Systems

We developed new approaches that exploit error tolerance during design, instead of during test, to obtain circuit/sub-system designs which improve cost, yield, power, and/or performance.

In particular, we developed design techniques at two different levels of abstraction, namely logic-block and sub-system. At the combinational logic block level, we developed two different approaches. First, we developed a logic synthesis approach that synthesizes approximate combinational circuits, starting with a given logic function and a given error rate threshold. Second, for a given error significance threshold, we developed an approach to simplify designs of commonly used data-path components, such as adders and multipliers, which are typically implemented using custom-designed architectures. In both these cases, our approaches aim to maximally simplify circuit designs to reduce circuit area, reduce circuit delay, and increase functional as well as parametric yields. At the sub-system level we concentrated on design approaches that enable maximum reductions in power. These approaches are detailed in [7] , [10], [11], and [30].

Our research has demonstrated the following.

1) It is possible to significantly improve yield, reduce cost, and/or improve performance by exploiting the notion of error tolerance during circuit-level design.

   For example, we show that it is possible to redesign a wide range of well known adder architectures – from the inexpensive ripple-carry to the fast Kogge-Stone – to significantly increase yield, even for low thresholds for error rate and significance.

2) It is possible to obtain significant power reductions by exploiting the notion of error tolerance during sub-system design.

   For example, we have used our error tolerance findings to identify novel tools for architectural simplification. As an example, in ME we have shown that standard 8 bit absolute difference (AD) operations can be replaced with quantized ADs. The performance loss can be as low as 0.05dB even when using a very aggressive simplification (quantizing ADs to 1 bit), which leads to 50 to 70% complexity reduction as compared to using full adders [7].

VI. Conclusion

We have demonstrated that by employing the concept of error tolerance, system designers can achieve benefits in one or more of the following areas: yield, power, costs and performance. But to achieve such benefits, many classical design or test approaches need to be discarded and new ones developed. Some of these new methodologies include:

- Specifications: Designers need to determine how to translate inexactness in functional requirements of the end user into attributes that can be measured by test equipment, as well as assign numeric values to determine ranges of acceptability.
- Design: Designers and design tools usually focus on attributes such as performance, power and logic minimization, and not issues such as defects and inexact computation. This narrow focus must end, possibly starting at such a basic level of allowing for some flexibility in what are the minterms of a Boolean switching function. This is also the case for design at higher levels, including architecture and algorithm.
- Test technologies: The classical test methodologies focus on discarding every chip with a manufacturing-induced imperfection. These approaches must be replaced by new ones that exploit the flexibility provided by the inexactness of functional requirements.
- Marketing: New strategies for branding, pricing, and labeling fabricated chips are also required.

Acknowledgement

We would like to acknowledge the students who have contributed to this project, including H.-Y. Cheong, I. S. Chong, H. Chung, N. K. Gokli, T.-Y. Hsieh, Z. Jiang, J. Melzer, Z. Pan, S. Shahidi, D. Shin, O. Yeung, and H. Zhu.

References

[1] M. A. Breuer, "Let's Think Analog", *Annual Symp. on VLSI*, pp. 2-5, March 2005.

[2] M. A. Breuer, "Multi-media Applications and Imprecise Computation", *8th EUROMICRO Conf. on Digital System Design*, September 2005.

[3] L. N. Chakrapani et al., "Ultra-efficient (embedded) SOC architectures based on probabilistic CMOS (PCMOS) technology," *Proc. Conf. on Design Automation and Test in Europe*, pp. 1110-1115, 2006.

[4] S. Cheemalavagu, P. Korkmaz, and K.V. Palem, "Ultra Low-energy Computing via Probabilistic Algorithms and Devices: CMOS device primitives and the energy-probability relationship," *Proc. Int'l. Conf. on Solid State*, 2004.

[5] H.-Y. Cheong, I. Chong and A. Ortega, "Computation Error Tolerance in Motion Estimation Algorithms", *IEEE Int'l. Conf. on Image Processing, (ICIP)*, Oct. 2006.

[6] H.-Y. Cheong and A. Ortega, "Motion Estimation Performance Models with Application to Hardware Error Tolerance ", *Visual Communications and Image Processing (VCIP),* 2007.

[7] H.-Y. Cheong and A. Ortega, ``Distance Quantization Method for Fast Nearest Neighbor Search Computations with applications to Motion Estimation", *41st Asilomar Conference on Signals, Systems and Computers*, Nov. 2007.

[8] I. S. Chong and A. Ortega, "Hardware Testing for Error Tolerant Multimedia Compression Based on Linear Transforms," *Defect and Fault Tolerance in VLSI Systems Symp.*, 2005.

[9] I. Chong, H.-Y. Cheong and A. Ortega, "New Quality Metrics for Multimedia Compression using Faulty Hardware", *Second Int'l. Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)* January 2006.

[10] I. S. Chong and A. Ortega, "Dynamic Voltage Scaling Algorithms for Power Constrained Motion Estimation" *Int'l. Conf. on Acoustics, Speech, and Signal Processing,* April 2007.

[11] I. S. Chong and A. Ortega, "Power Efficient Motion Estimation Using Multiple Imprecise Metric Computations" *Int'l Conf. on Multimedia & Expo,* July 2007.

[12] I. L. Chuang and M. A. Nielsen. Quantum Computation and Quantum Information. Cambridge Series on Information, 2000.

[13] H. Chung and A. Ortega, "Analysis and Testing for Error Tolerant Motion Estimation," *Defect and Fault Tolerance in VLSI Systems Symposium*, 2005.

[14] D. B. Fogel, "What is Evolutionary Computation?" *IEEE Spectrum*, pp. 26-32, Feb. 2000,.

[15] N. K. Gokli, Software Interface, Hardware Architecture, and Test Methodology for the Nvidia NV34 Graphics Processor, MS Thesis completed under the guidance of M. A. Breuer, 2006.

[16] T-Y Hsieh, K-J Lee and M.A. Breuer, "An Error-oriented Test Methodology to Improve Yield with Error-tolerance," *VLSI Test Symp. (VTS),* 2006.

[17] T.Y. Hsieh, K.J. Lee and M.A. Breuer. "Reduction of Detected Acceptable Faults for Yield Improvement via Error-tolerance," *Proc. of Design Automation and Test in Europe (DATE),* April 2007.

[18] T.Y. Hsieh, K.J. Lee and M.A. Breuer, "Preventing Over-detection of Acceptable Faults for Yield Enhancement, accepted for publication in *Int'l J. of Electrical Engineering*.

[19] Z. Jiang and S. Gupta, "An ATPG for threshold testing: Obtaining acceptable yield in future processes," International Test Conference, 2002.

[20] Z. Jiang and S. Gupta, "Threshold Testing: Covering Bridging and Other Realistic Faults," *Asian Test Symposium*, 2005.

[21] K. J. Lee, T. Y. Hsieh and M. A. Breuer, "A Novel Testing Methodology Based on Error-rate to Support Error-tolerance", *Int. Test Conf. (ITC)*, 2005.

[22] J. Melzer and K. M. Chugg, "Irregular Designs for Two-State Systematic with Serially Concatenated Parity Codes," submitted to *MILCOM,* 2006.

[23] Z. Pan and M.A. Breuer, "Estimating Error-rate in Defective Logic Using Signature Analysis," *IEEE Trans. on Computers*, vol. 56, no. 5, May 2007.

[24] D. Patridge, "A science of approximate computation," http://www.nesc.ac.uk/esi/events/Grand_Challenges/paneld/d17.pdf.

[25] M. Sadiku and M. Mazzara, "Computing with neural networks," *IEEE Potentials*, October 1993, pp. 14-16.

[26] S. Shahidi and S. Gupta, “Estimating Error Rate During Self-Test via One's Counting,” *In’l. Test Conf.*, 2006.

[27] S. Shahidi and S. Gupta, “A Theory of Error-Rate Testing,” *Int’l Conf. on Computer Design*, 2006.

[28] S. Shahidi and S. Gupta, “ERTG: A Test Generator for Error-Rate Testing,” *Int’l Test Conf.*, 2007.

[29] S. Shahidi and S. Gupta, “Multi-Vector Tests: A Path to Perfect Error-Rate Testing,” to be submitted to *Design and Test in Europe*, 2008.

[30] D. Shin and S. Gupta, “A Technique for Re-designing Datapath Units for Improving Manufacturing Yield by Exploiting Given Error Tolerance,” *Workshop on Design for Manufacturability and Yield*, 2007.

[31] P. Stanley-Marbell, D. Marculescu, R. Marculescu and P. Khosla, “Modeling, Analysis, and Self-management of Electronic Textiles,” *IEEE Trans. on Computers*, Vol. 52, No. 8, pp. 996-1010, August 2003.

[32] G. V. Varatkar and N. R. Shanbhag. “Energy-efficient motion estimation using error-tolerance.” *Int’l. Symp. on Low Power Electronics and Design (ISLPED)*, 2006.

[33] O. Yeung and K. M. Chugg, "An Iterative Algorithm and Low Complexity Hardware Architecture for Fast Acquisition of PN Codes in UWB Systems," *J. VLSI Signal Processing (special issue on Ultrawideband Systems),* vol. 43, April 2006.

[34] O. Yeung and K. M. Chugg, "An Iterative Algorithm and Low Complexity Hardware Architecture for Fast Acquisition of Long PN Codes in UWB Systems," *Proc. Allerton Conf. Commun., Control, Comp.,* October 2005.

[35] H. Zhu and M.A. Breuer, “Error-tolerance and Multi-media,” *IEEE Int’l. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Dec. 2006.

[36] H. Zhu and M.A. Breuer, “A Framework for the Analysis of Error-tolerance,” accepted for publication *IEEE Design and Test Magazine*, March/April 2007.

[37] H. Zhu, Error-tolerance in Digital Speech Recording Systems, MS Thesis completed under the guidance of M. A. Breuer, 2006.