21st IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'06), pp. 274-279, October 2006.

Low-Cost Hardening of Image Processing Applications Against Soft Errors

Ilia Polian[1,2] Bernd Becker[1] Masato Nakasato[2] Satoshi Ohtake[2] Hideo Fujiwara[2]

[1]Albert-Ludwigs-University
Georges-Köhler-Allee 51
79110 Freiburg i. Br., Germany
{polian|becker}@informatik.uni-freiburg.de

[2]Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma City, Nara, Japan
{masato-n|ohtake|fujiwara}@is.naist.jp

Abstract

Image processing systems are increasingly used in safety-critical applications, and their hardening against soft errors becomes an issue. We propose a methodology to identify soft errors as uncritical based on their impact on the system's functionality. We call a soft error uncritical if its impact is provably limited to image perturbations during a very short period of time (number of cycles) and the system is guaranteed to recover thereafter. Uncritical errors do not require hardening as their effects are imperceivable for the human user of the system. We focus on soft errors in the motion estimation subsystem of MPEG-2 and introduce different definitions of uncritical soft errors in that subsystem. We propose a method to automatically determine uncritical errors and provide experimental results for various parameters. The concept can be adapted to further systems and enhance existing methods.

Keywords: Soft errors, Image processing systems, Late-age silicon, MPEG, Threshold testing

1 Introduction

Soft errors caused by ionizing radiation are a prime concern in late-age silicon [1, 2]. Traditionally, image processing applications have not been hardened against soft errors. One reason was the low likelihood of their occurrence (according to the roadmaps, the estimated frequency of soft errors, called soft error rate (SER), in low-radiation environments is one in several months or even years). Furthermore, image processing applications are normally associated with consumer electronics for which an occasional failure is not critical: most soft errors would happen unnoticed by the end customer, and should an error lead to deterioration of the system's functionality, the customer would switch the equipment off and on again, and no further damage would be caused. The handling of such rare events did not justify adding significant costs for handling soft errors by using radiation-hardened (radhard) manufacturing technology or redundancy in case of consumer electronic which is generally characterized by low profit margin per IC.

On the other hand, image processing applications are increasingly used in safety-critical systems in fields including automotive, avionics, medical and military. The decisions made based on the output of these systems may lead to a damage or even loss of life. In many instances, the tasks solved by the systems have real-time constraints, and switching off and on or rebooting is not an option. At the same time, the SER is expected to increase over time [3, 4, 5]. It appears that soft errors in image processing applications should be handled, but the cost pressure remains. As a consequence, it is still unrealistic to employ radhard technology or redundancy techniques based on duplicating or triplicating the whole chip in image processing systems.

One recently proposed approach suggests to harden only a subset of possible soft error sites [6]. The *soft error susceptibility* of the candidate nodes (i.e., the probability that radiation induces a bit flip on a node) is calculated based on the technology parameters and the circuit's layout. The nodes with a high susceptibility are hardened locally while the remaining nodes are left unprotected. By doing so, the SER of the chip is reduced with minimal costs. The local hardening can be done by adding shielding layers, creating dummy junctions below the active area, and altering the package design [7, 8]. Alternatively, the data in [9] suggests that by simply duplicating a gate without need for any voting circuitry the soft error susceptibility of its output node is decreased by an order of magnitude. If an area or power consumption constraint is given, a number of nodes can be selected for hardening such that the constraint is not violated. Hence, selective hardening may be a practical approach to decrease the SER of image processing systems at acceptable costs.

It has been noticed that not every soft error will result in erroneous behavior of the system [10, 11, 12]. Some authors even argued that detection of such errors should be prevented in order not to reduce the system's performance by unnecessary counter-measures [13, 14]. In this paper, we argue that *the soft errors which are guaranteed not to result in an unacceptable behavior of the system do not require hard-*

Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)
0-7695-2706-X/06 $20.00

ening. We identify *uncritical* soft errors which do not produce unacceptable system behavior under any possible input sequence irrespective of the system's state. Uncritical soft errors should not be considered when calculating the SER. Under a selective hardening strategy, hardening against uncritical soft errors can be avoided even if they have high soft error susceptibility, such that other nodes can be hardened and SER is further reduced.

In case of general computing systems, any deviation of an output value from the reference value given by the specification is unacceptable. Under this definition, any uncritical soft error would simply correspond to a redundant node which has no influence on the system behavior, as we require that no unacceptable system behavior is produced under any input sequence and state. Such a node could be removed without any consequence for the system's functionality. However, an image processing application can produce output which does not match exactly the reference output but is still acceptable as the human viewer would not notice the difference. This is particularly true in a video application if subsequent video pictures are fault-free. In an image processing application, a soft error could be uncritical in sense that if it occurs very infrequently its impact on the system functionality would be minimal, and not redundant at the same time. One instance of such error would change a value of a pixel which is going to be overwritten by a correct value in the next picture, such that the wrong pixel would be visible only for a fraction of a second.

In this paper, we focus on identifying uncritical soft errors in the motion estimation (ME) logic of MPEG-2. We allow that the ME circuit calculates a wrong reference sample for a given number of clock cycles, but we require that the system must return to the error-free behavior thereafter, i.e., it must assume the same state as it would have assumed if the soft error had not happened. Since a soft error does not create a lasting damage in the chip, the system will continue to operate error-free for a long period of time (until the next soft error). The wrong ME results will create transient deterioration of the system's performance which will impact a very low number of pictures and most probably won't be noticed by the user. Note that MPEG-2 is designed to deal with computation errors due to wrong input data arriving through a potentially noisy communication data. Although in our scenario it is not the input data which is corrupted but rather the chip's logic itself, the effect (transient deterioration of the chip's performance) is similar.

The remainder of the paper is organized as follows. The next section provides details on motion estimation in MPEG-2 and formulates the criterion for a soft error being uncritical. Section 3 describes the methodology to identify uncritical soft errors based on an automatic construction of a mathematical proof. Relation to existing work is discussed in Section 4. Section 5 concludes the paper and gives some directions for future research.

2 Uncritical Soft Errors in Motion Estimation Circuits

Before defining the uncritical soft errors, the functionality of MPEG-2 is reviewed in order to facilitate the understanding by an unfamiliar reader. The discussion is on an intuitive level and some details such as handling of interlaced pictures are not covered. Please refer to text books such as [15] for detailed treatment.

In MPEG-2, the video stream consists of *pictures*. Every picture corresponds to a full screen and is broken into 16×16 pixel *macroblocks*. There are I pictures, P pictures and B pictures. The macroblocks in I pictures are encoded using a lossy compression technique similar to JPEG. For the macroblocks in P and B pictures, a "similar" macroblock, called *reference sample*, in a neighboring (previous or subsequent) picture is determined and its coordinates are encoded by a *motion vector*. Only the difference between the macroblock and the reference sample, called *residual*, is encoded using the same compression method. If a macroblock with low difference from the macroblock has been found, i.e., the residual consists mostly of zeros, then the compression ratio is improved significantly. Hence, macroblocks in the P and B pictures are represented by the motion vector and the compressed residual whereas in case of an I picture the complete macroblock needs to be compressed.

Motion estimation (ME) is the process of identifying the best reference sample for a given macroblock. It is possible to perform ME by calculating the difference from every possible reference sample and choosing the one with the lowest difference, but it is expensive. There are heuristic ME methods which solve this task using less comparisons (they are not guaranteed to come up with the best solution). ME circuits consist of a number of *processing elements* (PE), which calculate the difference between two pixels, and logic used to determine the motion vector, i.e., select the reference sample with the minimal difference calculated by the PEs. Special hardware architectures exist for increasing the performance of ME.

If the ME circuit is affected by a soft error, it may fail to determine the best reference sample resulting in a larger residual and a larger amount of the compressed data. As a consequence, either a higher butrate must be provided to transmit more data, or, if the bitrate is fixed, lower image quality must be enforced by manipulating the parameters of the compression method. If the effect of the soft error persists only for a few clock cycles, its impact will disappear completely when the next I picture is transmitted at the latest. Given that soft errors happen once in a few months or years, a deterioration which affects only a few pictures or an increased bitrate for a few clock cycles both appear to be acceptable.

The situation is different if a soft error changes the state of the system permanently, i.e., the system does not return into an error-free state and keeps calculating wrong reference samples for a long time. In this case, the bitrate increase or the quality deterioration could be permanent and the system would stop meeting its specification. Note that in applications such as satellite communication an increased bitrate automatically means increased energy consumption during communication and is a severe restriction of functionality. Hence, such a soft error is critical. However, even critical soft errors in the ME circuits do not lead to computation of wrong images, only to quality or bitrate deterioration which could persist for a long time.

In the next section, we introduce the methodology to automatically distinguish between critical and uncritical soft errors.

3 Proposed Method

We start by describing a ME system on which the experiments were performed. Then, we introduce the actual method and prove its correctness before reporting the results. Finally, an extension to find uncritical soft errors which result in less deterioration is proposed.

3.1 Benchmark circuit

We used a ME circuit of an MPEG-2 system implementing heuristic 2-stage motion estimation with half-pixel accuracy. The circuit consists of a first-stage processing element (PE) array, which calculates the differences between the current macroblock and a number of candidate reference samples (first search); the logic which determines the minimum difference and calculates the motion vector for the first stage; the PE array for the second search; the motion vector calculation for the second search; and a controller. We removed the motion vector calculation logic for the second search, such that the intermediate results of the second search are directly visible on the chip's output and the impact of soft errors is easily monitored. The circuit we use in the experiments has 16 8-bit data buses, which are the residuals of candidate macroblocks during the second search, as outputs, and no control output.

A soft error anywhere within the circuit may lead to a deviation of a value on the chip's output. The subsequent motion vector calculation logic could select the wrong matching vector and a residual which is difficult to compress.

3.2 Uncritical soft error identification

We have seen that a soft error is uncritical if its effects disappear after a given number of clock cycles under any state and input sequence. In our methodology, this number k is specified by the user. For a larger number k, more errors will be classified as uncritical but for a smaller k the maximal duration of the period in which the system produces incorrect results is reduced.

For simplicity, we assume that a soft error may occur on any output of a flip-flop or combinational logic and persists for exactly one clock cycle. We distinguish between the flip-to-1 and flip-to-0 errors. There are more accurate soft error models (see [2] for an overview and [1, 3] for factors involved including different masking concepts). The methodology can be easily extended to deal with a different soft error model if one is available.

Problem formulation: Given a circuit C with primary inputs PI, secondary inputs (flip-flop outputs) SI, primary outputs PO, secondary outputs (flip-flop inputs) SO, the constant k, the node n affected by the soft error and the flip direction g ($g = 1$ for a flip-to-1 error and 0 for a flip-to-0 error), determine whether there exists a state s and a sequence of input vectors $i_1, i_2, \ldots, i_k$ such that the circuit state after k clock cycles is different in presence and in absence of the error. □

In order to solve the formulated problem, we construct an *auxiliary circuit* C_{aux} such that the soft error in the original sequential circuit C is identical to a permanent stuck-at fault in the combinational auxiliary circuit. We will then prove that the soft error in C is uncritical if and only if the stuck-at fault in C_{aux} is redundant. By running a redundancy check on C_{aux}, which could be done by an ATPG or a (bounded) model checker, it is determined whether the soft error in C is critical or not.

The auxiliary circuit C_{aux} is illustrated in Figure 1: k-frame expansion of the circuit is generated. The PIs of the circuit are (SI, PI_1, ..., PI_k). The PIs of all k time frames and the SIs of the first time frame are controllable (this can be restricted if not all states are reachable). The SOs of frame i are connected to the SIs of frame $i + 1$ and are not observable. The POs of the first k frames are made unobservable by adding masking logic (AND gates with an input set to zero). The SOs of frame k are observable. The stuck-at-g fault on node n is injected into the first time frame but not into the subsequent frames.

Theorem: Soft error flip-to-g on node n in C is uncritical if and only if the stuck-at-g fault on node n in the first time frame of C_{aux} is redundant. □

Proof: Suppose that the stuck-at-g fault on node n in the first time frame of C_{aux} is not redundant and the test vector that detects it is (SI, PI_1, ..., PI_k). This means that under this vector C_{aux} produces different outputs at the SOs of frame $k + 1$. If soft error flip-to-g on node n happens in circuit C when it is in state SI and the input sequence (PI_1, ..., PI_k) is applied in the subsequent cycles, the state of the circuit after k cycles will deviate from the error-free state, so the soft error is critical.

Suppose that the soft error in C is critical. Then, there must be a state s and a sequence of input vectors $i_1, i_2, \ldots, i_k$ such that the circuit state after k clock cycles is different in presence and in absence of the error. Applying the input vector $(s, i_1, i_2, \ldots, i_k)$ to circuit C_{aux} will lead to

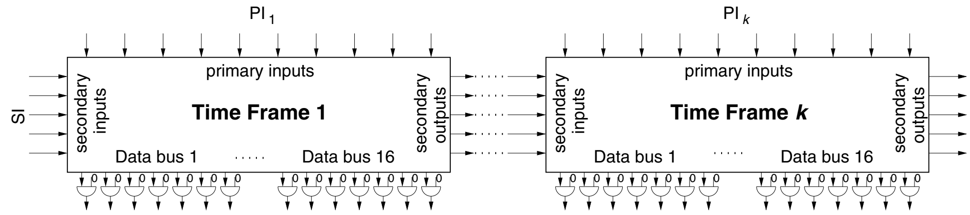

Figure 1: Circuit C_{aux} used for uncritical soft error identification

different outputs when the error is present or absent. Consequently, this vector detects the fault and it cannot be redundant. q.e.d.

Note that the stuck-at fault is not injected in the frames 2 through k because the soft error is assumed to persists for only one clock cycle. It is possible to inject different faults according to a different soft error model. If the soft error model describes effects which last more than one cycle, the injection must take place in more than one time frame.

3.3 Results

We constructed the circuit from Figure 1, injected a stuck-at fault into the first time frame, and searched for an input assignment using a commercial ATPG tool. All errors identified as redundant are uncritical. There was a total of 82,664 stuck-at faults in the first time frame of MPEG-2 ME circuit's time frame expansion. The run times were low as the circuit was relatively small even when expanded several times. Note that combinational ATPG is routinely employed in the industry for multi-million gates designs. Table 1 summarizes the influence of k, the number of cycles which the circuit is allowed to deviate from its reference behavior, on the number of uncritical soft errors. The percentage of the identified uncritical soft errors among all 82,684 soft candidate soft errors is quoted in the last column of the table. It can be seen that a significant fraction of soft errors is uncritical and can be excluded from selective hardening.

k	Number of uncritical faults	Percentage
1	645	0.78
2	1645	1.99
3	2649	3.21
4	3655	4.42
5	4687	5.67
6	5685	6.88
7	6715	8.13
8	7713	9.33
16	14763	17.86
24	22093	26.73
32	29173	35.30
48	43871	53.06
64	58413	70.65
96	58801	71.12

Table 1: Number of uncritical soft errors as function of k

3.4 Deviation limit

In this section, we introduce additional requirements for a soft error to be classified as uncritical. We require that the *extent of deviation* caused by the error in the worst case is limited. In particular, we require that the values calculated on the circuit's outputs in presence and in absence of the error do not differ too much. This is related to the concept of threshold testing [16]. The underlying assumption is that if a soft error leads to calculation residuals which are not the best but are *close* to the best ones, the resulting increase of the bitrate or deterioration of the quality will be of limited extent.

We define that the system behavior after a soft error has happened is *acceptable* if any deviation will be observed only on τ least significant bits (LSB) of any of the 16 output buses. If $\tau = 1$, one LSB is allowed to deviate, if $\tau = 4$, 4 LSBs are allowed to deviate, and so forth. The requirement that the error effect must be completely gone after k clock cycles remains in place, so this new definition is stricter than the previous one. $\tau = 8$ corresponds to the previous definition as all the output bits are allowed to deviate.

The circuit used for identification of uncritical soft errors leading to acceptable behavior for $\tau = 2$ is shown in Figure 2. Note that it has $k + 1$ time frames and any deviation is forbidden on frame $k + 1$.

Results for $k = 8$ and different values of τ are given in Table 2. It can be seen that the new definition is indeed much stricter and not many soft errors qualify as uncritical with respect to this definition. It may be necessary to apply transformations to the circuit in order to increase the number of soft errors which are uncritical considering a deviation limit.

4 Related Work

There is a significant body of research on fault tolerance, including classical textbooks [17]. Of particular importance with respect to this work are the architectural redundancy techniques as they can include the ability of a system to return from an erroneous state to an error-free state. Never-

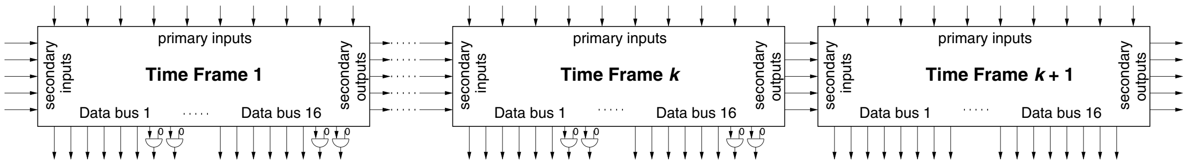

Figure 2: Circuit used for uncritical soft error identification with a deviation limit

theless, the classical fault tolerance for digital systems does not allow any deviations from the reference behavior and its techniques are designed to guarantee output values which could not be distinguished from ones produced by a system not affected by errors. This is different for analog circuits for which the reference behavior is given by a range rather than by a value.

Soft errors with no effect on the system behavior have been described in [10, 11, 12] but this was with respect to a given input sequence. In our work, the errors identified as uncritical are mathematically proven to be uncritical for any possible input sequence and state. The soft errors that do change the system output but in a way that is irrelevant for the application have been mentioned in [3] (soft errors in parts of 64-bit registers used to process 32-bit data) and [13] (instructions that will never commit and data which will never be read).

There is some recent interest in a methodology referred to as *error tolerance* [18]. An error tolerant system is allowed to deviate with respect to a metric which is part of the system's specification. The same IC can be sold for its regular price if it is manufactured fault-free and for a lower price if it contains a defect which leads to acceptable yet not reference behavior. Threshold testing mentioned above is an instance for a metric of error tolerance. Up to now, we are not aware of any work on error tolerance which considered soft errors.

Some works specifically addressed error tolerance aspects of image processing [19, 20, 21]. In particular, [20] addressed error-tolerant motion estimation, considering permanent defects which were restricted to the motion vector calculation (PEs were assumed to be hardened and not susceptible to defects). In contrast, we assume soft errors and consider all elements of the circuit.

There is significant research on *error concealment* in video coding [22, 23]. The assumed errors are due to video data being transmitted through an unreliable communication channel. Packets containing video data are lost or arrive too late to be processed. Typical error concealment techniques try to reconstruct missing data from available information such as neighboring pixels. In contrast, our approach considers errors in the video processing hardware itself rather than erroneous or missing input data.

τ	# uncritical faults	τ	# uncritical faults
1	13	5	71
2	30	6	82
3	45	7	97
4	56		

Table 2: Number of uncritical soft errors with acceptable behavior for $k = 8$ as function of deviation limit τ

Performing computation with potentially imprecise result has been investigated in the field of real-time computing [24]. The underlying assumption of *imprecise computing* is that there are two versions of a task: one accurate and slow and one imprecise and fast. Various kinds of scheduling problems with different optimization goals and deadlines are solved. Propagation and amplification of errors through a sequence of imprecise tasks can be considered.

5 Conclusions and Future Work

We proposed a methodology for identifying uncritical soft errors in image processing based on their impact on the system functionality. A soft error is considered uncritical if its consequence is a deterioration in image quality which is so short that the end user would not notice it and the system is guaranteed to recover thereafter. Up to around 70% of soft errors are found to be uncritical. The methodology complements existing low-cost selective hardening strategies which are based on the soft error susceptibility of a node [6]. By combining both strategies, a better protection against critical soft errors is possible.

While the experiments have been run on a motion estimation circuit, it is possible to extend the methodology to generic image processing applications. The uncritical faults can be re-defined as the ones which do not create deviations which are visible for the human user. Psychovisual metrics such as one proposed in [21] can be employed. It is likely that the approach can also be generalized to audio and other signal processing applications.

A useful extension would be to define *severity* of an error rather than to declare it critical or not. [2]. This would yield a list of soft errors sorted by severity. Then, the hardening approach similar to [6] could be used. Severity is also useful for design of optimal online BIST logic [2]. Methods to calculate severity may include simulation, emulation or hardware experiments (similar to fault injection campaigns run for airspace applications [25, 11]) or formal methods including PTMs [26] and probabilistic model checking [27].

Finally, *synthesis* of circuits in which a large portion of soft errors are uncritical is of interest. This requirement is obviously less strict than the problem solved by the classical fault tolerance, namely to produce a circuit which never deviates from its reference behavior. Consequently, there is hope that low-cost solutions could be obtained.

Acknowledgments

This work was supported in part by 21st Century COE (Center of Excellence) Program "Ubiquitous Networked Media Computing", in part by JSPS (Japan Society for the Promotion of Science) under Grants-in-Aid for Scientific Research B(2)(No. 15300018) and in part by the DFG project RealTest under grant BE 1176/15-1. We are grateful to John P. Hayes, Melvin Breuer and Rob Aitken for discussions on the subject and encouragement. We thank the Semiconductor Technology Academic Research Center (STARC) for providing the MPEG circuit.

6 References

[1] P.E. Dodd and L.W. Massengill. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Trans. on Nuclear Science*, 50(3):583–602, 2003.

[2] I. Polian, J.P. Hayes, S. Kundu, and B. Becker. Transient fault characterization in dynamic noisy environments. In *Int'l Test Conf.*, 2005.

[3] H.T. Nguyen and Y. Yagil. A systematic approach to SER estimation and solutions. In *Int'l Reiliability Physics Symp.*, pages 60–70, 2003.

[4] P. Shivakumar, M. Kistler, W. Keckler, D. Burger, and L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Int'l Conf. on Dependable Systems and Networks*, pages 389–398, 2002.

[5] N. Seifert, X. Zhu, and L.W. Massengill. Impact of scaling on soft-error rates in commercial microprocessors. *IEEE Trans. on Nuclear Science*, 49(6):3100–3106, 2002.

[6] K. Mohanram and N.A. Touba. Cost-effective approach for reducing soft error failure rate in logic circuits. In *Int'l Test Conf.*, pages 893–901, 2003.

[7] R. Baumann. Tutorial: Ghosts in the machine: A tutorial on single-event upsets in advanced commercial silicon technology. In *Int'l Test Conf.*, 2004.

[8] M.G. Pecht, R. Radojic, and G. Rao. *Managing Silicon Chip Reliability*. CRC Press, 1998.

[9] T. Heijmen and A. Nieuwland. Soft-error rate testing of deep-submicron integrated circuits. In *European Test Symp.*, pages 247–252, 2006.

[10] N.J. Wang, J. Quek, T.M. Rafacz, and S.J. Patel. Characterizing the effects of transient faults on a high-performance processor pipeline. In *Int'l Conf. on Dependable Systems and Networks*, 2004.

[11] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Sonza Reorda, and M. Violante. An FPGA-based approach for speeding-up fault injection campaigns on safety-critical circuits. *Jour. of Electronic Testing: Theory and Applications*, 18(3):261–271, 2002.

[12] S. Kundu, M.D.T. Lewis, I. Polian, and B. Becker. A soft error emulation system for logic circuits. In *Conf. on Design of Circuits and Integrated Systems*, page 137, 2005.

[13] C. Weaver, J. Emer, S.S Mukherjee, and S.K. Reinhardt. Int'l symp. on comp. architecture. pages 264–275, 2004.

[14] S. Kundu and I. Polian. An improved technique for reducing false alarms due to soft errors. In *Int'l On-Line Test Symposium*, pages 105–110, 2006.

[15] V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers, kluwer international series in engineering and computer science edition, 1995.

[16] Z. Jiang and S.K. Gupta. An ATPG for threshold testing: Obtaining acceptable yield in future processes. In *Int'l Test Conf.*, pages 824–833, 2002.

[17] D.P. Siewiorek and R.S. Swarz. *Reliable Computer Systems – Design and Evaluation*. Digital Press, 1992.

[18] M. Breuer. Error-tolerance and related test issues. In *Asian Test Symp.*, page xxi, 2004.

[19] I. Chong and A. Ortega. Hardware testing for error tolerant multimedia compression based on linear transforms. In *Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 2005.

[20] H. Chung and A. Ortega. Analysis and testing for error tolerant motion estimation. In *Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 2005.

[21] I. Chong, H. Cheong, and A. Ortega. New quality metric for multimedia compression using faulty hardware. In *Int'l Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.

[22] Y. Wang, S. Wenger, J. Wen, and A.K. Katsaggelos. Error resilient video coding techniques. *IEEE Signal Process. Mag.*, 17(4):61–82, 2000.

[23] W.-Y. Kung, C.-S. Kim, and C.-C.J. Kuo. Spatial and temporal error concealment techniques for video transmission over noisy channels. *Trans. Circuits and Systems for Video Tech.*, 16(7):789–802, 2006.

[24] J.W.S. Liu, W.-K. Shin, K.-J. Lin, R. Bettati, and J.-Y. Chung. Imprecise computations. *Proc. of the IEEE*, 82(1):83–94, 1994.

[25] F. Lima, S. Rezgui, L. Carro, R. Velazco, and R. Reis. On the use of VHDL simulation and emulation to derive error rate. In *RADECS*, pages 253–260, 2001.

[26] S. Krishnaswamy, G.F. Viamontes, I.L. Markov, and J.P. Hayes. Accurate reliability evaluation and enhancement via probabilistic transfer matrices. In *Design Automation and Test in Europe*, pages 282–287, 2005.

[27] M. Kwiatkowska. Model Checking for Probability and Time: From Theory to Practice. In *Proc. 18th IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 351–360. IEEE Computer Society Press, June 2003. Invited Paper.