A Case Study in Reliability-Aware Design: A Resilient LDPC Code Decoder

Matthias May, Matthias Alles, Norbert Wehn

Microelectronic Systems Design Research Group
University of Kaiserslautern, 67663 Kaiserslautern, Germany
{may, alles, wehn}@eit.uni-kl.de

Abstract

Chip reliability becomes a great threat to the design of future microelectronic systems with the continuation of the progressive downscaling of CMOS technologies. Hence increasing the robustness of chip implementations in terms of error tolerance becomes an important issue. In this paper we present a case study in reliability-aware design tolerating transient errors. A state-of-the-art WiMAX channel decoder for LDPC codes is investigated on all design levels to increase its reliability for a given system performance with minimum hardware overhead. We show that an efficient exploitation of the algorithmic fault-tolerance yields a fairly small area overhead with nearly no degradation in communications performance even under high error injection rates.

1. Introduction

Error sources such as radiation induced soft errors, erratic intermittent errors due to, e.g., crosstalk, power supply and temperature variations, make microelectronic circuits and systems less and less reliable in future scaled CMOS technologies. The traditional worst case design methodology becomes infeasible due to the large area and energy overhead and the required a priori knowledge of all error sources at design time. As a result, future microelectronic systems must correct or mitigate the occurrence of errors. Thus it is mandatory to consider reliability as a cross-cutting problem concerning not only technology and test engineers but also system designers [1, 10]. Classical techniques like triple modular redundancy (TMR) and radiation hardening are very expensive in terms of area, power, and performance and would diminish the realized performance and area benefit of further scaling. Thus, more sophisticated approaches are necessary.

In this paper we present a case-study in reliability-aware design. A Low-Density Parity-Check (LDPC) Code decoder is selected as case study. LDPC codes belong to the most powerful forward error correction codes known today and are part of many communication standards like DVB-S2, WiMAX, WiFi and space application standards [8, 13, 14, 18]. Especially the last one is very susceptible to soft errors. The decoding of LDPC codes belongs to the important class of probabilistic belief propagation algorithms which have in addition to communication a broad applicability such as data mining, image recognition and case synthesis. The latter three applications are often referred to as Recognition, Mining, and Synthesis (RMS) which are to be considered to be the next killer applications [4].

The LDPC decoding algorithm has some inherent fault tolerance, i.e., not all errors in the computation and transmission of data have an impact on the algorithmic performance. This is a typical property of many multimedia applications, too. Thus, the approach presented in this paper is not restricted to LDPC code decoders only.

Here we consider erratic intermittent transient errors, which come about on the physical and circuit level due to, e.g., soft errors and timing errors. From a system perspective a large design space exists to mitigate or correct these errors. Many techniques were developed to avoid soft errors or reduce interconnect noise on the circuit level [6, 15]. On higher levels spatial- and time redundancy, error detection and correction codes can be applied [5, 17, 19, 20, 21]. However the general application of these techniques imply a large overhead in area, energy and timing.

In this paper we focus on a system approach. In other words, the application, in our case the LDPC decoding, is taken into account in the design space exploration and its inherent fault tolerance is exploited. Hence, we advocate a joint algorithm, architecture, and error handling co-design considering the following strongly interrelated levels:

- On the application level different decoding algorithms (see Section 2) are investigated with respect to error resilience and error propagation.
- On implementation level, different architectures, various data representations and the criticality of the individual signals are examined.
- Finally, the ”error sensitivity” of each subblock of the architecture is checked in the system context and an appropriate error protection technique is selected such that the *overall system reliability* is increased.

Only few publications exist which investigate the reliability issue for this application domain. In [16] a timing error tolerant design of a Turbo-Code decoder was presented to reduce the energy by voltage overscaling. A worst case design technique is applied, but the area overhead is reduced by focusing only on the important signals in the decoder. An important-aware clock skew scheduling technique is presented that assigns at design time circuit paths associated with important bits a longer timing slack. However this approach requires an a priori knowledge of all transient errors at design time and the decrease in the communications performance is quite large. In [3] an extension of the LDPC decoding algorithm was considered to make it more robust against timing errors. However, in this work, only timing errors in the connectivity network of the decoder are considered from the algorithmic side. Memories and functional units are not taken into account. Moreover the impact on the hardware is not investigated.

The paper is structured as follows. In Section 2 LDPC codes and various decoding algorithms are discussed. Section 3 introduces the selected decoder architecture. In Section 4 the new reliability-aware decoder is presented. Results are given in Section 5 and Section 6 concludes the paper.

2. LDPC Codes

LDPC codes are linear block codes defined by a sparse binary matrix H, called the parity check matrix. The set of valid codewords C satisfies

$$Hx^T = 0, \qquad \forall x \in C. \tag{1}$$

A column in H is associated to a codeword bit, and each row corresponds to a parity check. A nonzero element in a row means that the corresponding bit contributes to this parity check. The complete code can best be described by a Tanner graph, a graphical representation of the associations between code bits and parity checks. Code bits are shown as so called variable nodes (VN) drawn as circles, parity checks as check nodes (CN) represented by squares, with the edges connecting them according to the parity check matrix. Figure 1 shows a Tanner graph for a generic irregular LDPC code with N variable and M check nodes with a resulting code rate of $R = (N - M)/N$.

The number of edges connected to a node is called the node degree. If the node degree is constant for all CNs and VNs, the corresponding LDPC code is called regular, otherwise it is called irregular. Note that the communications performance of irregular LDPC codes is generally superior to that of regular LDPC codes.

Decoding Algorithm

LDPC codes are decoded using a probabilistic message passing algorithm [9]. Messages between both kind of

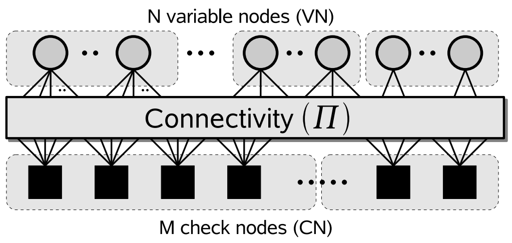

Figure 1. Tanner graph of an LDPC code

nodes in the Tanner graph are iteratively exchanged. Soft decision algorithms use several bits to represent the received symbols and the messages. This yields a much better decoding performance than hard decision decoder. The sign of the exchanged soft values represents the decision made by the node, i.e., a one or a zero respectively, and the additional bits correspond to the probability whether the made decision is correct or not. This message passing algorithm has some inherent fault tolerance as will be shown later.

Updating the nodes can be done with a canonical, two-phased scheduling: In the first phase all variable nodes are updated, in the second phase all check nodes respectively. The processing of individual nodes within one phase is independent and can thus be parallelized. Other updating schemes are for example layered decoding algorithms [12]. However our investigations have shown that the layered algorithm shows a significant error propagation due to the accumulation of the a priori values. If an a priori value is corrupted, the values in the subsequent iterations are all corrupted. Hence we decided to use the more robust two-phase scheduling algorithm in our reliability-aware decoder.

For a soft decision decoder the exchanged messages are assumed to be log-likelihood ratios (LLRs). The variable node j of degree d_v calculates an a posteriori information according to:

$$\lambda_j = \lambda_j^{ch} + \sum_{i=1}^{d_v} \lambda_{ij}, \tag{2}$$

with λ_j^{ch} the channel LLR associated with VN j and λ_{ij} the LLR of the connected check node i. The sign of λ_j can be used as a hard decision for the decoded bit. For the outgoing message λ_{ji} to check node i the incoming message λ_{ij} from check node i has to be subtracted from the a posteriori information:

$$\lambda_{ji} = \lambda_j - \lambda_{ij}. \tag{3}$$

Check node i then computes new messages that are sent back to the variable nodes again. When the parity check in a check node succeeds, all edges to the connected variable nodes will preserve their signs. On the other hand, when the parity check failed, the signs of all edges are changed. The magnitude of the outgoing messages corresponds to the belief whether the made decision is correct.

Various algorithms for the message passing calculation exist: Sum-Product, Min-Sum and λ-Min [9, 11]. They differ in their implementation complexity, communications performance and error sensitivity. The Sum-Product has a high implementation complexity and is very sensitive to errors. We selected the 3-Min algorithm which yields a near optimum communications performance [11], low implementation complexity and good robustness with respect to errors. This algorithm uses the 3 smallest absolute input values and applies a correction term to counter the introduced approximation.

3. Decoder Architecture

High throughput LDPC code decoders require parallel or partly parallel architectures. Fully parallel decoders are infeasible for large block sizes or when flexibility in block sizes is requested. Thus, we focus on partly parallel architectures in which only a subset of nodes is instantiated and the variable and check nodes are time multiplexed processed on these instantiated nodes.

As mentioned in the previous section, we apply the two-phase scheduling scheme with the 3-Min algorithm which yields the architecture shown in Figure 2. A subset P of variable nodes and check nodes are instantiated as variable node functional units (VFU) and check node functional units (CFU) respectively. Hence each VFU has to process N/P variable nodes and each CFU has to process M/P check nodes in a time multiplexed way. The functional units sequentially process the incoming messages which provides flexibility when different node degrees have to be supported as requested by many standards.

The VFU contains four different RAMs. All N/P channel values λ_j^{ch} which are processed in the corresponding VFU are stored in the channel RAMs. The sum RAMs are used to treat the N/P sums according to the sum term in Equation 2. While one sum RAM is used to accumulate the incoming messages the second RAM contains the sums that were build in the previous iteration. In the subsequent iteration, both sum RAMs are swapped. The message RAM stores the messages λ_{ij} that are received from the corresponding CFUs. The values in this RAM are used to calculate the outgoing messages according to Equation 3 for the subsequent iteration.

The connectivity between the nodes is realized by a flexible permutation network. Since all standardized LDPC codes are designed using only permuted identity matrices of size P in the parity check matrix H, a logarithmic barrel shifter can be used for implementation of this network.

4. Reliability-Aware Decoder Design

This section describes the modifications of an existing LDPC code decoder to improve its reliability. We use a

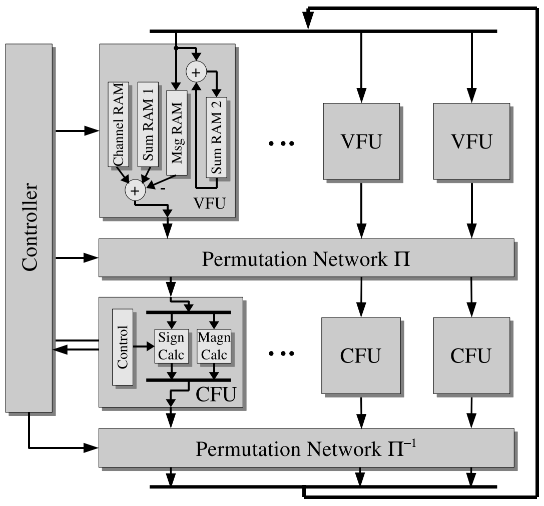

Figure 2. Two-phase decoder architecture

WiMAX decoder as reference [2]. This decoder has a parallelism degree of 96, achieving a throughput of up to 333 Mbit/s at 400 MHz on a 65nm technology. The total area is 1.3 mm^2. As reference for the communications performance we selected the block length N=2304, the code rate $R = 1/2$ and 30 iterations.

In the following we investigate for each building block of the architecture (memories, functional units, controller and permutation network) its ”error sensitivity” and derive appropriate error detection and correction techniques for each block to maximize the system reliability. Two important characteristics of the message passing algorithm are exploited in all building blocks:

- Not all data bits of a message or channel value have the same importance. Corruption in higher significant bits has a larger impact on the overall communications performance than corruption in lower bits.
- If an LLR value which is calculated by a functional node is corrupted, no error correction is mandatory. Instead we can ”puncture” the value, i.e., reset it to 0. This means that the belief of this node in its decision is a 50:50 probability for a zero and a one. In other words, the corresponding node/edge is temporarily removed for the current iteration in the tanner graph. Thus, it has a diminishing impact on the communications performance.

4.1. Memories

Memories are very susceptible for soft errors. Hence we assume that each bit in a RAM flips with a specific probability within a specific period of time. All RAMs are supposed to have the same mean time between failure (MTBF). Figure 3 shows the frame error rates (FER) of the decoder under different scenarios. The decoder works without loss

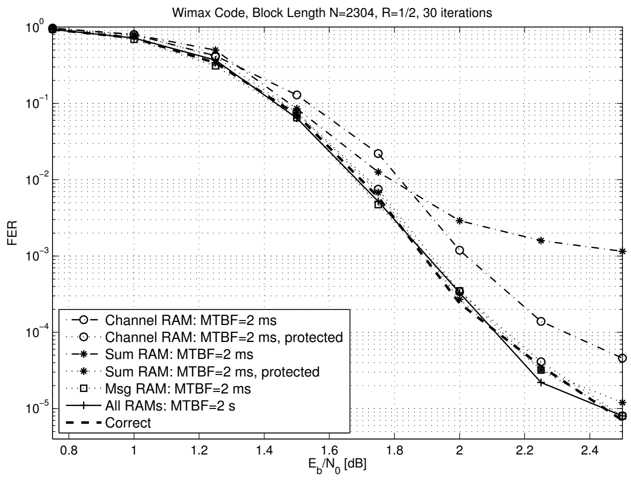

Figure 3. Performance: RAM reliability

in communications performance up to an MTBF of 2 seconds. The error protection presented in the following allows for nearly the same decoding performance up to an MTBF of $2ms$.

Channel RAM: As already mentioned not all bits in the RAM have the same importance. Our investigations have shown that it is sufficient to only protect the MSB. Thus we double the MSB when writing the values into the RAM. During reading we compare the two bits. If there is a mismatch we simply force the value to 0. It is important to mention that the values in the channel RAM are updated only once before decoding a block. Therefore, in the last iterations the probability of error occurrences increases. However due to the strong extrinsic values during the last iterations, the puncturing approach is sufficient as error handling.

Sum RAM: Similar to the channel RAM only the MSB has to be protected. However, a puncturing of an erroneous value would mean that all informations of the previous iterations are lost. Hence, we have to correct the error. Thus we triplicate the MSB during writing and vote during reading.

Message RAM: Errors in the message RAM have no impact on the overall performance, even with an MTBF of $2ms$, since they correspond to an error on a single edge for the current iteration, whereas a decision of a VN depends on multiple edges.

4.2. Controller

It is obvious that errors in the controller are very critical for the overall system. Therefore, the controller has to be strongly protected. This is provided by a triplication of the controller and 2-out-of-3 voting at its outputs. When an output signal of a decoder deviates from the two others, it has to be assumed that the controller can run into a false state. In this case, the controller is reset when the other controllers get into the initial state after the decoding of the current block.

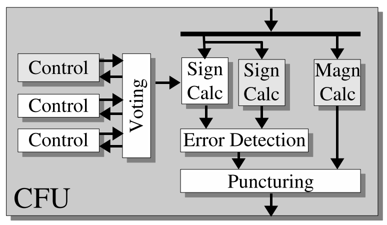

Figure 4. Error protection in CFU

4.3. Functional Units

Error detection and correction of the CFUs is shown in Figure 4. Like the main controller, the control unit in the CFU is protected by triplication with an 2-out-of-3 voting mechanism.

An analysis of the dynamic range has proven that a sign-magnitude representation of the exchanged messages is preferable to a two's complement representation. It saves power and simplifies error protection. Investigations have shown that the sign bits are by far the most important bits since they represent the hard decision of the decoded bits and exploit the inherent fault-tolerance of the message passing and 3-Min algorithms with respect to magnitudes. Therefore, only the parity check unit which calculates the signs of the messages has to be protected. This is implemented by duplicating the unit and comparison of its outputs. If an error occurs we again apply puncturing, i.e., the corresponding value is reset to 0.

4.4. Permutation Networks

Interconnection is very prone to errors. The use of ARQ is not possible due to strong latency constraints. Thus we implemented a combined soft and timing error detection mechanism for both networks. Each message sent from a VFU to a CFU is protected by a duplication of the sign bit. To detect not only bit flip but also timing errors the duplicated sign signal is flipped every second clock cycle, see Figure 5. We don't use a Razor Flip-Flop [7] for detection of timing errors, since the dimensioning of such a flip-flop is very critical and assumes a special cell which is normally not available in a standard cell library. On the receiver side we perform the disjunction of all error signals of the sign bits of all messages (note that we process the received messages in a time-multiplexed way) of a single check node. The result of this error check is fed to the CFU. If an error happened in the transmission of the sign bits all output messages of the currently processed check node by the CFU are set to 0. Here we can reuse the puncturing unit of the CFU error handling.

Error protection for the second permutation network is done in the same way. Simulations have shown that this error protection and correction mechanism has no negative impact on the system performance assuming a soft error rate (SER) and timing error rate (TER) of 10^{-4} respectively,

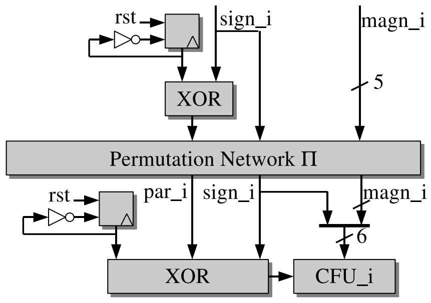

Figure 5. Error handling in network

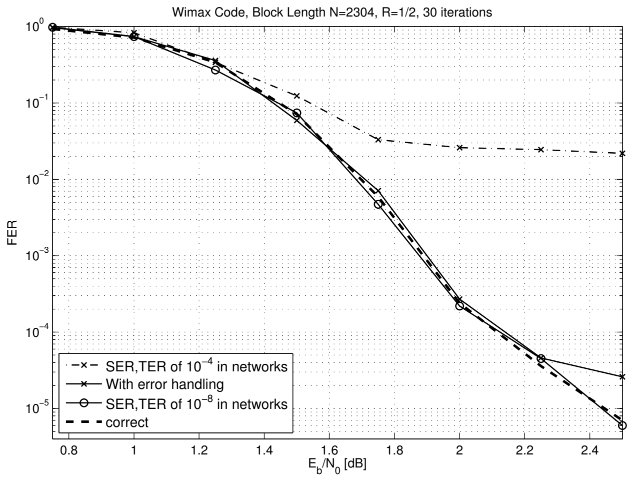

Figure 6. Performance: network reliability

apart from a slightly higher error floor, see Figure 6. The SER corresponds to 9 flipped bits, the TER to 9 delayed bits or 1.5 delayed words, on average per iteration respectively. Without error protection, the decoder works adequately up to SER=TER=10^{-8}.

5. Results

Table 1 summarizes the various error detection and correction techniques we applied in the individual subblocks of the decoder.

Figure 7 shows the FER for the selected WiMAX code with rate R=$1/2$ for the unprotected and error resilient LDPC decoder respectively with error injection in all units. Soft and timing errors were induced with very high error rates. Soft errors in the RAMs are induced with an MTBF of $2ms$. SER and TER in the permutation networks are 10^{-4}, which corresponds to an MTBF of $25\mu s$. Errors in the sign and magnitude calculation of the CFUs are induced with a bit-flip error rate (iBER) of 10^{-4}. Channel values and messages are quantized with 6 bits respectively, the sum RAMs have a width of 9 bits. Simulation is based on an AWGN channel with BPSK modulation. The dashed line is the performance graph of the reference WiMAX decoder. If we induce the aforementioned errors without any detection and

Unit	Technique	Error Correction
Controller	triplication	voting
CFU	controller triplication sign duplication	voting alg. puncturing (= 0)
Permutation Networks	sign duplication	alg. puncturing (= 0)
Channel RAM	MSB duplication	alg. puncturing (= 0)
Sum RAMs	MSB triplication	voting
Message RAM	none	none

Table 1. Techniques for error handling

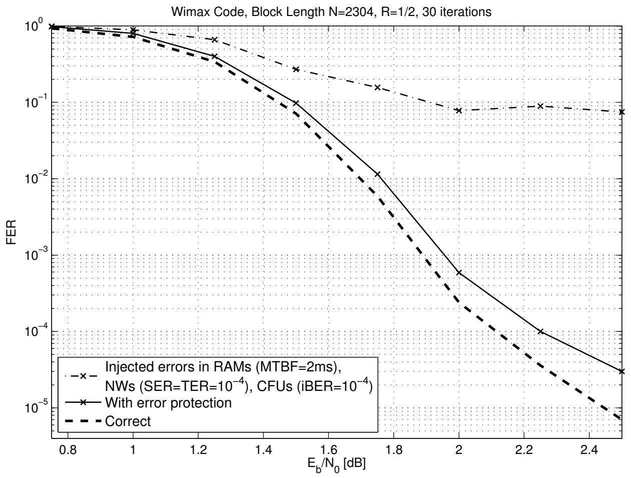

Figure 7. Overall performance

correction mechanisms, we can see an enormous impact on the decoding performance, so that the decoder practically refuses to work. With our error handling, the performance of the decoder is near that of the error free decoder. Our investigations of WiMAX codes with other code rates show similar improvements of the communications performance under injected errors compared to the unprotected decoder.

Table 2 shows the area overhead of the error resilient WiMAX decoders in comparison with the area of the state-of-the-art decoder without any error detection and protection mechanism [2]. The decoders were synthesized with the Synopsys Design Compiler on a 65 nm low power standard cell library at 400 MHz under worst case conditions.

Only the controller as highly critical unit is triplicated. Due to the exploration of the inherent fault tolerance of the selected decoding algorithm and the consideration of the "error sensitivity" in the system context, no protection for the message RAMs is needed and the area of the other units is increased very moderate. The total area increase accounts for 21%. Up to an MTBF of 2 seconds for the RAMs and an MTBF of 0.25 seconds for the rest, only the controllers need to be protected. This yields an area increase of 6% only. The additional error correction and detection mechanism had no impact on the circuit performance. Thus all decoders could be synthesized with the same clock frequency providing the same throughput and latency. It is important to mention that our presented techniques for increasing the

Unit	Un-protected	Resilient decoder	Only controllers protected
Controller	0.03	0.09 (+200%)	0.09 (+200%)
VFUs (w/o RAMs)	0.11	0.11	0.11
CFUs	0.43	0.55 (+ 26%)	0.46 (+ 5%)
Perm. Networks	0.21	0.25 (+ 23%)	0.21
Sum RAMs	0.21	0.25 (+ 22%)	0.21
Channel RAM	0.07	0.09 (+ 23%)	0.07
Message RAMs	0.25	0.25	0.25
Overall area	**1.31**	**1.59 (+ 21%)**	**1.39 (+ 6%)**

Table 2. Area[mm^2], 65 nm @ 400 MHz

reliability had absolute no impact on the overall control flow of the decoder and, thus, can be applied to other decoders, too.

6. Conclusion

In this paper we have presented a case study in reliability-aware design. We investigated an LDPC code decoder architecture and developed extensions to increase the robustness of the decoder against sporadic errors. Thereby, we regarded all levels of design in combination. The decoder provides even under much higher error rates than current SER nearly the same decoding performance as the state-of-the-art decoder without transient errors. Synthesis results show a fairly small additional area overhead. LDPC decoding is an important representative of emerging probabilistic applications such as data recognition, mining, and synthesis (RMS). Hence the methodology and results presented in this paper are not limited to LDPC decoders only.

Acknowledgement

The work presented in this paper was partly done within the scope of the AIS project (funding label: 01 M 3083), which is partly supported in accordance with the BMBF "Nanoelectronics" sponsorship scheme within the scope of the Ekompass-sponsorship complex.

References

[1] Reliability-Aware Microarchitecture. *IEEE micro*, 25(6), Nov./Dec. 2005.

[2] M. Alles, T. Brack, T. Lehnigk-Emden, F. Kienle, N. Wehn, N. Insalata, F. Rossi, M. Rovini, and L. Fanucci. On the Implementation of Low Complexity LDPC Code Decoders for Next Generation Standards. In *Proc. 2007 Design, Automation and Test in Europe (DATE '07)*, Nice, France, Apr. 2007.

[3] M. Alles, T. Brack, and N. Wehn. A Reliability-Aware LDPC Code Decoding Algorithm. In *Proc. 56th Vehicular Technology Conference (VTC Spring '07)*, pages 1544–1548, Dublin, Ireland, Apr. 2007.

[4] P. Dubey. Recognition, Mining and Synthesis Moves Computers to the Era of Tera. *Technology@Intel Magazine*, Feb. 2005.

[5] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S. G. Miremadi. Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks. In *Proc. 2007 Design, Automation and Test in Europe (DATE '07)*, Apr. 2007.

[6] M. A. Elgamel and M. A. Bayoumi. Interconnect Noise Analysis and Optimization in Deep Submicron Technology. *IEEE CIRCUITS AND SYSTEMS MAGAZINE*, 3(4):6–17, 2003.

[7] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation. In *Proc. 36th International Symposium on Microarchitecture*, pages 7–18, Dec. 2003.

[8] European Telecommunications Standards Institude (ETSI). Digital Video Broadcasting (DVB) Second generation framing structure for broadband satellite applications; EN 302 307 V1.1.1. www.dvb.org.

[9] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, Massachusetts, 1963.

[10] Giovanni De Micheli. Designing Robust Systems with Uncertain Information. ASP-DAC 2003 Keynote Speech.

[11] F. Guilloud, E. Boutillon, and J. Danger. λ-Min Decoding Algorithm of Regular and Irregular LDPC Codes. In *Proc. 3nd International Symposium on Turbo Codes & Related Topics*, pages 451–454, Brest, France, Sept. 2003.

[12] D. Hocevar. A reduced complexity decoder architecture via layered decoding of LDPC codes. In *Proc. IEEE Workshop on Signal Processing Systems (SiPS '04)*, pages 107–112, Austin,USA, Oct. 2004.

[13] IEEE 802.11n. Wireless LAN Medium Access Control and Physical Layer specifications: Enhancements for Higher Throughput. IEEE P802.16n/D1.0, Mar 2006.

[14] IEEE 802.16e. Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE P802.16e/D12 Draft, oct 2005.

[15] S. Krishnamohan and N. Mahapatra. A highly-efficient technique for reducing soft errors in static cmos circuits. In *Proc. IEEE International Conference on Computer Design (ICCD 2004)*, pages 126–131, 11-13 Oct. 2004.

[16] Y. Liu, T. Zhang, and J. Hu. Low Power Trellis Decoder with Overscaled Supply Voltage. In *Proc. 2006 Workshop on Signal Processing Systems (SiPS '06)*, pages 205–208, Banff, Canada, Oct. 2006.

[17] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. D. Micheli. Analysis of Error Recovery Schemes for Networks on Chips. *IEEE Design and Test of Computers*, 22(5):434–442, 2005.

[18] NASA. Low Density Parity Check Code for Rate 7/8. GSFC - STD - 9100, May 2006.

[19] M. Nicolaidis. Time redundancy based soft-error tolerance to rescue nanometer technologies. In *VLSI Test Symposium, 1999. Proceedings. 17th IEEE*, pages 86–94, 25-29 April 1999.

[20] N. R. Shanbhag. Reliable and Efficient System-on-Chip Design. *Computer*, 37(3):42–50, 2004.

[21] F. Worm, P. Ienne, P. Thiran, and G. De Micheli. On-Chip Self-Calibrating Communication Techniques Robust to Electrical Parameter Variations. *IEEE Design & Test of Computers*, 21(6):524–535, Nov. 2004.