Designing MRF based Error Correcting Circuits for Memory Elements

K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky
Brown University, Division of Engineering, Providence, RI 02912

Abstract

As devices are scaled to the nanoscale regime, it is clear that future nanodevices will be plagued by higher soft error rates and reduced noise margins. Traditional implementations of error correcting codes (ECC) can add to the reliability of systems but can be ineffective in highly noisy operating conditions. This paper proposes an implementation of ECC based on the theory of Markov random fields (MRF). The MRF probabilistic model is mapped onto CMOS circuitry, using feedback between transistors to reinforce the correct joint probability of valid logical states. We show that our MRF approach provides superior noise immunity for memory systems that operate under highly noisy conditions.

1. Introduction

Numerous methods have been proposed to improve reliability of circuits in the presence of increased soft error rates and single event upsets (SEU). Traditional approaches include triple-modular redundancy (TMR), N-modular redundancy, cascaded TMR [1] and error correcting codes (ECC) [2]. Multi-bit data in memory elements is protected using error correcting codes (ECC), where a number of redundant bits are encoded into the data bits forming a *codeword*. When the codeword is accessed from memory, a decoder checks the parity bits to detect and correct any errors caused by corrupted data. Hamming code is one very popular code used in error-detection and correction in memory structures. A detailed description of conventional Hamming ECC implementation is given in [3].

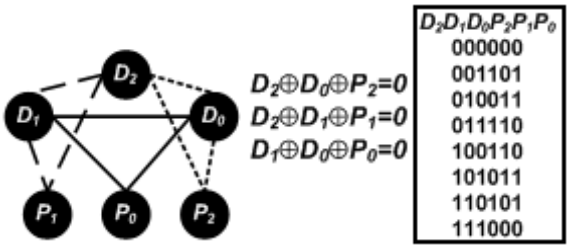


Figure 1: Dependence graph, constraint equations and corresponding codewords.

While ECC has proved to be effective (and even essential) in achieving high reliability in memory systems, traditional means of implementing this logic are not in-and-of themselves fault tolerant. That is, under highly noisy operating conditions, the ECC logic may fail to properly correct corrupted data. In this paper, we propose an implementation of the ECC based on probabilistic Markov random fields (MRF). The use of Markov Random Fields as a foundation for a probability-based design methodology for nanoscale computation in the presence of noise and circuit errors was first proposed in [4] and mapping of this probabilistic framework to modified CMOS-based circuitry was shown in [5]. The mapping enabled circuits to operate reliably at very low subthreshold voltages (e.g. 150mV). The application of MRF theory to error correcting codes using dynamic programming was explored in [6]. In this paper, our goal is to map the MRF-based error correcting code onto a physical implementation using ultimate CMOS technology.

Consider a 3-bit data protection scheme using Hamming encoding. Given the $\lceil \log_2 N \rceil + 1$ requirement for protecting N bits, our 3-bit data needs 3 additional redundant bits, resulting in a 6-bit codeword. The interaction between the data and redundant bits can be represented with a dependence graph and constraint equations shown in Figure 1. The dependence graph shows that all data and parity bits are explicitly or implicitly dependent on each other. The PLA type MRF implementation, based on the recipe for mapping MRF networks into CMOS structures [5], is shown in Figure 2. The mapping was created by directly taking the codewords from the constraint table and creating eight different bistable elements (one for each row) and a feedback path from the output of each of the bistable elements to the storage nodes.

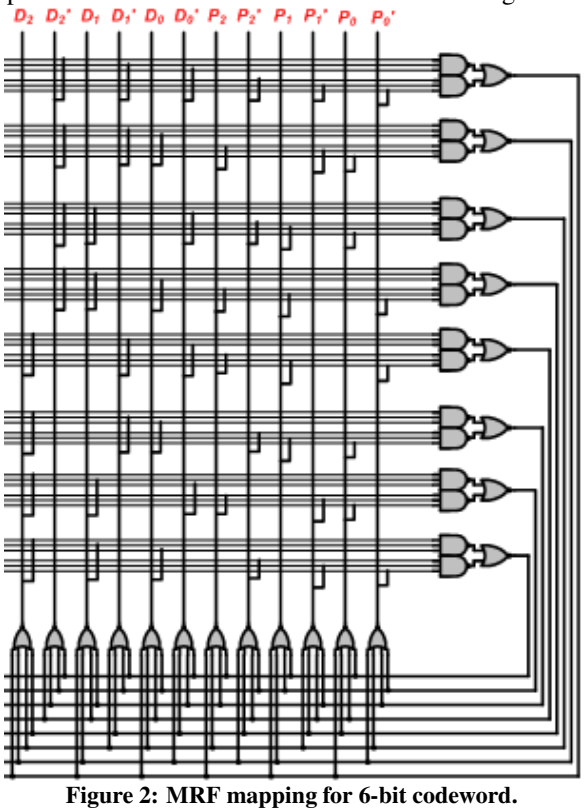


Figure 2: MRF mapping for 6-bit codeword.

The circuit consists of twelve "storage nodes", one for each data and parity bit and its complement. The stable states of the nodes correspond to the maximum probability configurations of the variables. These occur when all nodes are storing values that correspond to a correct codeword. The values are reinforced using the feedback connection to the complemented inputs. If one of the nodes flips to an incorrect value, the feedback will restore the correct value at that

3-9810801-0-6/DATE06 © 2006 EDAA

node. The CMOS representation of the MRF ECC guarantees that that probability distribution of the *valid* codewords is maximized. Note that the codewords were generated with a minimum Hamming distance of 3, so in the event of a single bit error, the distance of the invalid code will be closer to one codeword compared to the other and the circuit will settle at a *valid and correct* codeword.

We emphasize that our MRF ECC differs from traditional ECC in two fundamental ways: (i) both data and parity bits are treated equally in the MRF graph node, and (ii) error detection and correction is done naturally in the system without explicit decoding.

2. Simulation setup and results

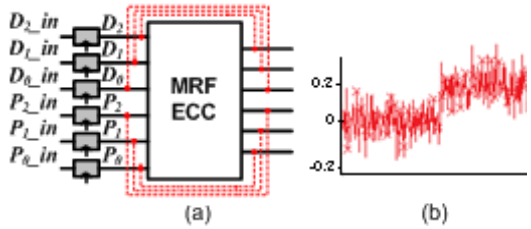
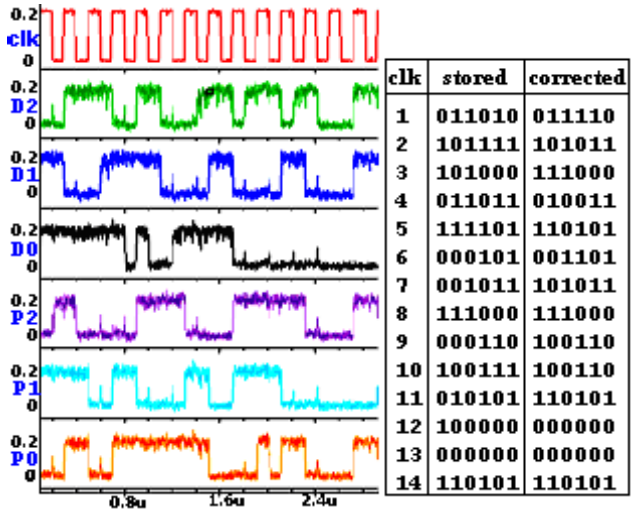


Figure 3: (a) Simulation setup. (b) Sample input signal with added noise.

For our simulations, the data and parity bits are first stored in a memory element (i.e., flip-flop) and subjected to noisy conditions. The output of the memory element with possible one-bit error is then sent to the MRF ECC circuit. The feedback from the MRF circuit acts on the stored data and parity bits when the clock signal is low *(i.e. when new values are not allowed to propagate from the input to the output of the flip-flop)* and reinforces the correct codeword. All devices including the storage elements and the components of the MRF circuits are subject to the same noisy signal conditions. The simulation setup is shown in Figure 3. The circuits were simulated in SPICE using the 70 nm BPTM [7] at V_{DD} = 0.2 V *(i.e. subthreshold operation)*, T = 100 °C and the Gaussian noise model from [5].



clk	stored	corrected
1	011010	011110
2	101111	101011
3	101000	111000
4	011011	010011
5	111101	110101
6	000101	001101
7	001011	101011
8	111000	111000
9	000110	100110
10	100111	100110
11	010101	110101
12	100000	000000
13	000000	000000
14	110101	110101

Figure 4: Simulation of a set of random codewords.

Figure 4 shows the first 14 clock cycles of the simulation of a random sequence of codewords with possible one-bit errors. Again the feedback from the MRF circuit corrects any one-bit deviation from the correct codeword and restores the state of the memory element to the correct codeword. The adjoining table in the figure summarizes the clock cycle, the stored codeword and the corrected codeword. By comparing the corrected codewords to the set of valid codewords, it is clear that the MRF ECC circuit works for any one-bit error correction.

Note, that the CMOS mapping of the ECC circuit shown in Figure 2 is a PLA type mapping and not necessarily the most optimal implementation in terms of area. While an optimal mapping is being investigated, the benefits of MRF circuits in terms of superior noise immunity can be exploited by using a modular approach when protection of a large data set is needed. Using a modular approach to protect data using three or four of the data bits at a time, more than single-bit error protection can be achieved at a reduced transistor count. Figure 5 compares the number of transistors required in a Hamming decoder built using traditional techniques and one built using modular MRF ECC. We see that while the modular MRF implementation is less efficient compared to regular ECC in terms of transistor count, the difference is not prohibitive, especially considering the superior noise immunity provided by the MRF approach.

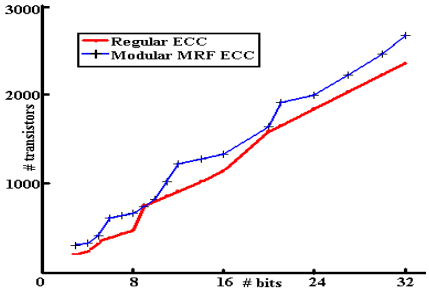


Figure 5: Transistor count comparison between regular hamming decoder and modular MRF ECC.

3. Conclusions

The MRF probabilistic model provides a framework for designing CMOS circuits that can operate effectively under ultra-low supply voltage and extreme noise conditions. A new approach to implementing Hamming ECC codes based on Markov random fields has been demonstrated to provide significant reliability against single event upsets and noisy transients. A more area optimal mapping is under investigation.

4. References

[1] S. Spagocci and T. Fountain. Fault rates in nanochip devices. In *Electrochem. Soc.*, pages 354–368, 1999.

[2] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Netherlands: North-Holland, 1977.

[3] F. J. Aichelmann Jr. Fault-tolerant design techniques for semiconductor memory applications. *IBM Journal of Research and Development*, 28(2):177–183, March 1984.

[4] R. I. Bahar, J. Mundy, and J. Chen. A probabilistic-based design methodology for nanoscale computation. In *ICCAD*, Nov. 2003.

[5] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky. Designing logic circuits for probabilistic computation in the presence of noise. In *DAC*, June 2005.

[6] S. Geman and K. Kochanek. Dynamic programming and the graphical representation of error-correcting codes. *IEEE Tran. on Information Theory*, 47:549–568, April 2001.

[7] http://www-device.eecs.berkeley.edu/∼ptm/.