



Optimizing Noise-Immune Nanoscale Circuits using Principles of Markov Random Fields

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

ABSTRACT

As CMOS devices and operating voltages are scaled down, noise and defective devices will impact the reliability of digital circuits. Probabilistic computing compatible with CMOS offers a possible solution. In this work, we present a new area and power efficient design methodology for the implementation of a probabilistic framework into CMOS technology based on Markov Random Fields (MRF). Using SPICE, we simulate elementary logic components and sample circuits from the MCNC'91 benchmark set and show the area and power benefits compared to older MRF mapping strategies. We also extend our area and power efficient approach to improving the design of a Hamming decoder based on MRF principles.

Categories and Subject Descriptors: B.8.1 [Performance and Reliability]: Reliability, Testing, and Fault-tolerance

General Terms: Design, Reliability, Emerging technologies.

Keywords: Markov Random Fields, Noise Immunity, Circuit optimization, Subthreshold Operation, Error correcting codes.

1. INTRODUCTION

As CMOS technology downscales, circuit designers have to contend with defective devices operated in a noisy signal environment at low V_{DD} [1]. The resulting reduction in noise margins will expose computation to higher soft error rates, impacting the viable microarchitecture approaches of the future. Thus, it is unlikely that the circuit designers of the future will be able to assume error-free operation at the device level.

Numerous probabilistic methods have been proposed as models for computing in the presence of noise and device defects [2, 3, 4]. The use of Markov Random Fields (MRF) as a foundation for probabilistic computation in the presence of noise and circuit errors was proposed in [4]. The mapping of the MRF-based probabilistic framework to ultimate CMOS circuitry was shown in [5] and the approach was extended to the design of a Hamming decoder for protection of memory against single event upsets in [6]. In particular, the approach of [5] showed that subthreshold operation was viable for reliable computation at reduced dynamic power levels and with high level of noise immunity; however the improvements came with an order of magnitude increase in area.

In this paper, we expand on the previous works of [5] and [6], and in particular focus on providing an improved mapping of MRF circuits onto CMOS, in terms of both area and power dissipation. We identify two different design strategies for the MRF design. The first strategy involves the creation of simple MRF elements with a single output using the concept of factorization and the second strategy involves the use of multiple constraint equations for a circuit that has multiple outputs. We show that our new mapping can operate at highly noisy conditions and provide superior noise immunity, as was shown also in [5, 6]. However, our new mapping achieves this noise immunity with significantly reduced device overhead. We also show the power advantage of our mapping compared to that of [5, 6].

2. OPTIMIZATION OF MRF ELEMENTS

The basic requirements for mapping MRF clique energy functions into CMOS structures was presented in [5]. The clique functions were taken from the logic compatibility table and all valid states (i.e. input and output pairs) were enumerated into separate bistable storage elements. The results from these storage elements were taken back to the individual inputs and outputs through appropriate feedback elements. If explicit enumeration of all valid input-output pairs were necessary, creating a MRF element with a larger fan-in would cause an explosion in the transistor count, severely limiting the applicability of this approach. As such, in this paper we show an alternate mapping of the MRF elements which provides better efficiency in terms of area and power and allows for creation of larger fan-in elements.

Consider a NAND gate with input variables x_0, x_1 , and output variable x_2 . There are a total of four valid states for the NAND2 element — three states where the output variable is a logic 1 when either or both of the input variables is at logic 0 and the fourth state where the output variable is at 0 if both input variables are at logic 1. By summing over all valid states, the logic compatibility function or the clique energy function of the NAND2 gate can be obtained:

$$U_c(x_0, x_1, x_2) = x'_0 x'_1 x_2 + x'_0 x_1 x_2 + x_0 x'_1 x_2 + x_0 x_1 x'_2 \quad (1)$$

This equation for the NAND gate can be re-expressed as:

$$U_c(x_0, x_1, x_2) = (x'_0 + x'_1)x_2 + x_0 x_1 x'_2 \quad (2)$$

Note that, combining the first three terms of Equation 1 into a single term in Equation 2 does not result in the loss of any valid input-output pair. Using this factored form of Equation 1, an area efficient mapping of the NAND gate can be created as shown in Figure 1.

The mapping consists of a OAI (OR-AND-INV) gate implementing the first term $(x'_0 + x'_1)x_2$ and a 3-input static CMOS NAND gate implementing the second term $x_0 x_1 x'_2$. The number of bistable elements required decreased from 4 (*for the four minterms*) to just 2. This decrease also reduced the complexity of the feedback path.

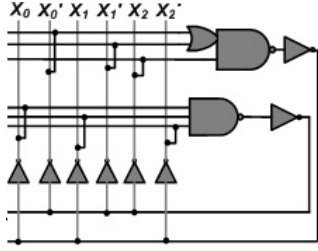


Figure 1: Area efficient MRF NAND gate implementation. The inputs are x_0 and x_1 , the output is x_2 .

In the approach of [5], the feedback to x_2' came from the output of a NOR gate whose inputs were three elements representing the minterms containing the term x_2 (see Equation 1). This now reduced from a three-input NOR (or its DeMorgan's equivalent) to a simple inverter that takes the output of the topmost complex gate and feeds back to x_2' . Similarly, the feedback to other nodes are also reduced. Mapping the simplified equation now produces a circuit that uses only 28 transistors compared to the 60 transistors shown in [5]. Using this factorization technique, higher fan-in circuits can be created without exponentially increasing the circuit area and complexity. Table 1 shows the reduction in transistor counts for multiple-input standard MRF elements mapped using our new area-efficient mapping. For all MRF circuits, the feedback components must be sized slightly larger to eliminate the possibility of any metastable states that might arise due to contention between the input and feedback.

std. gates	mapping from [5]	new mapping
2-input	60	28
3-input	144	36
4-input	352	44
5-input	832	48

Table 1: Comparison of transistor counts for multiple-input standard MRF elements.

The new modified MRF NAND gate was simulated in SPICE using the 70 nm Berkeley predictive technology model [7] at $V_{DD} = 0.15V$ and $T = 100^\circ C$. The use of subthreshold V_{DD} allows us to show the advantage of probabilistic computation in ultimate CMOS devices and also to capture the noise margin reduction due to thermal noise effects, electromagnetic coupling, hot-electron effects as well as threshold variations. For our simulations, noise is modeled using a 60mV RMS Gaussian model, which is of the order expected for ultimate CMOS [5]. The simulation of the optimized NAND gate subjected to uncorrelated noisy inputs is shown in Figure 2. As can be seen from the figure, the output of a regular static CMOS NAND gate is very noisy, rendering the gate unusable. However, the MRF NAND gate provides stable voltage operation and excellent noise immunity, similar to the mapping presented in [5], but at a reduced transistor count

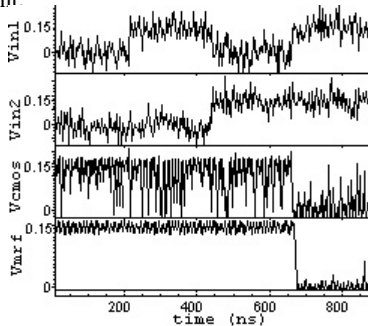


Figure 2: Simulation of regular static CMOS NAND and optimized MRF NAND gate in presence of noise.

Table 2 shows the comparison between the method of [5] and the

new optimized implementation in terms of the number of transistors and power consumption under noisy conditions for circuits selected from the MCNC'91 benchmark set. Also shown in the table are the number of *first-stage* transistors (i.e., the number of transistors gated by primary inputs) and the maximum number of gates along any path from primary input to output (i.e., the *depth* of the circuit). The table shows that the new optimized method results in an average of 63% reduction in area in terms of transistor counts and an average of 65% reduction in power dissipation compared to the method from [5]. The table also compares the power dissipation of the benchmark circuits synthesized using regular static CMOS gates running at 1V (*the expected V_{DD} for 70nm technology*). The results show that the new MRF method provides a power advantage, particularly for circuits with larger depth and many transistors in the first stage. Specifically, the new MRF implementation consumes on average 33% less power than the standard CMOS implementation for these larger circuits (e.g., *alu4*, *cordic*, *ex5*, and *table5*). This is significant, since this implies that our MRF elements may be used more effectively in larger circuit designs. For circuits with shallower depth, there is not as much flexibility available in the MRF mapping so a power advantage may not always exist. In these cases, as a power/reliability tradeoff, it might be advantageous to evaluate the circuit areas most vulnerable to defects and noise, and selectively introduce MRF elements as needed to achieve desired reliability.

3. OPTIMIZATION OF A HAMMING DECODER

In this section, we show area and power optimization of MRF circuitry for systems with multiple outputs and multiple constraint equations using a probabilistic Hamming decoder as an example. A circuit implementation of a (6,3) Hamming decoder using the MRF framework was proposed in [6]. In the following, we show an alternate mapping strategy for the MRF-based error correcting code for reliable error correction in a noisy environment.

A (6,3) code has a minimum Hamming distance of 3 for the detection and correction of all one-bit errors. Given a set of data bits d_2, d_1 and d_0 , the three parity bits p_2, p_1 and p_0 can be computed from the following set of constraint equations,

$$\{d_2 \oplus d_0 = p_2\}; \{d_2 \oplus d_1 = p_1\}; \{d_1 \oplus d_0 = p_0\} \quad (3)$$

These constraint equations can be used to generate all possible codewords for the (6,3) Hamming decoder. The generated codewords are shown in Figure 3(a).

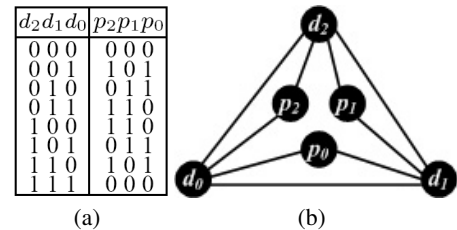


Figure 3: (a) Codewords (b) MRF dependence graph for parity equations of a (6,3) Hamming decoder.

A MRF dependence graph can be drawn to show the interaction between the data and the parity bits represented by Equation 3. The dependence graph in Figure 3(b) shows that there is an implied dependence between data bits and parity bits that do not appear in the same equation of the constraint equations. For example, the logical state of p_2 is directly dependent on the states of d_2 and d_0 , the state of d_2 is in turn dependent on d_1 and p_1 and d_1 is dependent on d_0 and p_0 . This means that p_2 is directly or indirectly dependent on the states of all the data and the other parity bits. This

Circuit	in	out	CMOS $V_{DD}=1V$				method from [5]		New MRF	
			# tran	1 st -stage	depth	power(μW)	# tran	power(μW)	# tran	power(μW)
5xp1	7	10	568	25	10	101.4	7188	380.5	2756	151.2
alu4	14	8	6928	153	23	875.2	94164	1617.4	33416	612.1
con1	7	2	78	6	6	16.5	952	50.2	356	16.9
cordic	23	2	604	32	15	89.8	7924	129.45	2612	54.7
ex5	8	63	5448	135	13	692.5	75648	1312.6	25964	506.9
misex1	8	7	356	11	7	69.6	4536	237.6	1700	82.0
o64	130	1	520	65	8	24.7	7224	173.6	2752	44.5
rd53	5	3	232	6	9	40.7	2576	156.3	1012	46.3
squar5	5	8	346	10	8	55.6	3920	233.2	1532	70.1
table5	17	15	10192	237	23	1522.5	141220	2242.7	47948	936.1

Table 2: Comparison of transistor counts and power for MCNC'91 benchmark circuits.

implied dependence between all the nodes of the dependence graph adds some degree of complexity to the overall circuit design but it also has an advantage. Traditional methods of Hamming decoding proceed by first computing a syndrome, locating the error position based on the syndrome generated and then explicitly correcting the erroneous bit. However, due to all nodes interacting with each other and being interdependent, explicit location and correction of single-bit error is not required for the MRF approach. If an error occurs on any of the data or the parity bits, feedback from all the other error-free nodes allows the circuit to correct the incorrect node back to the correct state without explicit identification of the erroneous node.

The codeword shown in Figure 3(a) can be considered as the compatibility function of the MRF Hamming decoder. The authors of [6] used this codeword table and generated a unified constraint equation from the table. The unified constrain equation was then mapped into a PLA style network. The biggest drawback of their approach is the exponential growth of the circuit height with the increase in the number of data bits being protected. For instance, in [6], for 3-bit data protection, eight different states are enumerated using 6-input NAND gates as bistable elements. Similarly, protection of four data bits using the same Hamming technique would require implicit enumeration of sixteen codewords in the circuit. This quickly begins to add up causing an explosion in the number of transistors required to implement the MRF technique.

In this paper, we propose a CMOS mapping of the MRF framework using the individual constraint equations rather than the unified equation. Consider the first of the three equations from Equation 3 where d_2 and d_0 are XORed to get p_2 . This relation can be expanded and written as the compatibility function similar to the one shown in Equation 1.

$$U_c(d_2, d_0, p_2) = d_2 d_0' p_2' + d_2' d_0 p_2 + d_2 d_0' p_2 + d_2 d_0 p_2' \quad (4)$$

The constraint equation for the remaining two parity equations can be written similarly. We now take these three separate constraint equations and create the circuit shown in Figure 4. The circuit consists of *storage nodes*, one each for d_2 , d_1 , d_0 , p_2 , p_1 , p_0 and their complements. The stable states of these nodes correspond to the maximum probability configurations of the variables. The top four NAND gates shown in the circuit are for the minterms of the first parity equation and the remaining for the second and third equations. The feedback circuitry becomes slightly more complicated. The feedback to d_2 not only comes from the minterms containing its complement in the first equation, but also from the terms of the second one. Similarly for d_1 and d_0 .

The CMOS representation shown in Figure 4 guarantees that the probability distribution of the valid codewords is maximized. On a single-bit error, the distribution of the incorrect codeword is closer to one of the valid codewords because of the Hamming distance constraint. When an error occurs in the storage nodes, interaction between the different nodes causes the system to be unstable and

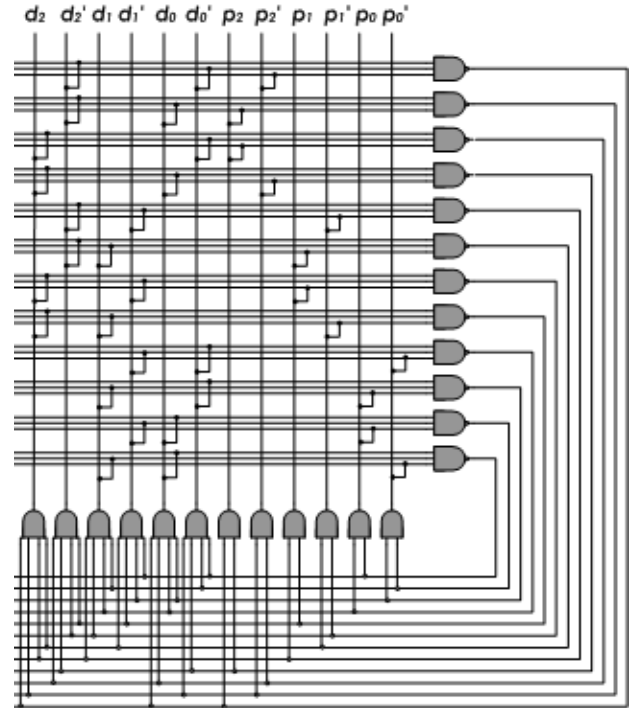


Figure 4: Circuit schematic for a new mapping of MRF (6,3) Hamming decoder.

eventually gain stability by forcing the incorrect node to its correct value using the feedback path. If no error is present, the feedback path just reinforces the correct values back to the nodes. Note that the feedback is in contention with the input values to the nodes. In our design, the feedback gates are sized slightly larger to prevent any metastable states that might arise due to this contention.

To measure the area and power effectiveness of our mapping we use the same simulation setup and subject our circuits to the same noise conditions described in [6]. The data and parity bits are stored in a memory element on the high phase of the clock and the error correction is done on the low phase of the clock. All devices are subject to noisy transients and the circuit operation is again in the subthreshold regime. The simulations of the circuit for one codeword (111000) is presented in Figure 5. Only the output data and the parity bits are shown for the sake of space. All possible one-bit error scenarios for the codeword 111000 are simulated. In the first clock cycle, the flip-flops store the correct code 111000 when the clock signal is high thus requiring no change to the codeword. In this case, the circuit causes no change to the value stored in the memory element. In the next cycle, the value of d_2 stored in the memory element is flipped due to some unknown transient or single event upset leading to the storage of 011000 — an invalid codeword. On the same clock cycle when the clock is low, the circuit

reacts to the one-bit error and via the feedback path reinforces the correct value on d_2 forcing the stored codeword to 111000. In the subsequent cycles, one-bit errors are introduced to the remaining five bits of the codeword. As is clear from the simulation in Figure 5, all one-bit deviations from the correct codeword are fixed by the reinforcing principle of the MRF circuit.

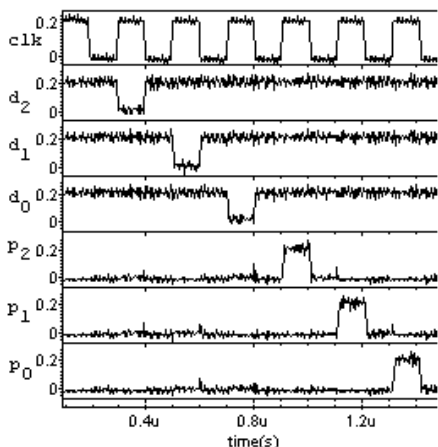


Figure 5: Result of the (6,3) decoder for 111000 codeword.

	# transistors	Power (μW)	
		no noise	noise
Traditional	136	0.42	1.71
MRF from [6]	224	1.14	3.75
New MRF	168	0.53	1.55

Table 3: Comparison of area in terms of transistor counts and power for (6,3) Hamming decoding.

Table 3 shows the improvement over the approach of [6] in terms of transistor count and power dissipation. Note that the new mapping uses only two-thirds the transistors of the mapping shown in [6] and is comparable to the area of the traditional syndrome decoder. The power measurements are also included for the circuits with and without any noisy inputs. Noise on input signals and gates causes an increase in the short-circuit power thereby increasing the overall power of the circuit. The measurement of power under noisy conditions shows that there is a 60% reduction in power when the new approach is used compared to [6]. The power for noisy conditions shows that the regular syndrome decoder uses slightly more power compared to our MRF mapping. This is attributed to the fact that the noise tolerance and filtering property of the MRF reduces the short-circuit power. We emphasize that the traditional scheme would not be able to operate correctly at such low-voltage and high noise levels. Hence we also simulated the traditional Hamming decoder at the recommended power supply of 1V where the signal to noise ratio was much higher, leading to reliable computation. The added reliability with increase in power supply voltage came with a power overhead of $19.2\mu W$ — a 12 fold increase in power consumption compared to our MRF technique.

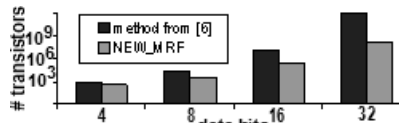


Figure 6: Comparison of transistor count of the ECC schemes.

So far we have shown the implementation of the (6,3) MRF Hamming decoder. While the MRF circuit provides excellent tolerance to noise and is comparable in power under noisy conditions to a regular syndrome decoder, protection of large data field causes a quick explosion in the area required. The added area requirement for a nibble (4-bits) memory protection scheme is very small (268

transistors compared to 232 for a traditional approach) but a byte-protection scheme using the new approach requires seven times as many transistors compared to the traditional approach. This is a considerable improvement compared to the approach of [6] which has a 40-fold increase in the number of transistors. Figure 6 shows the area savings of our new approach compared to that of the MRF approach shown in [6].

Data bits	Traditional	nibble-protection		
		Traditional	MRF from [6]	NEW
4	232	232	704	268
8	466	464	1408	536
16	1136	928	2816	1072
32	2346	1856	5632	2144

Table 4: Comparison of transistor counts for nibble-protection scheme.

Regardless of the increase in reliability attributed to the MRF scheme, straightforward implementation of the MRF-scheme for large data sets becomes impractical as illustrated in Figure 6. One option to protect large data sets while maintaining the reliability provided by MRF mapping scheme is to use a nibble-protection scheme. In this scheme, instead of employing single error correction on the entire data set, the data is segmented into multiple 4-bit nibbles. Each nibble of data is protected at a time allowing for multi-bit protection using the MRF approach and maintaining the tolerance to noise. Table 4 shows that the nibble protection using the new scheme is comparable to that of a traditional protection scheme in terms of transistor count. However, the traditional scheme would not be able to provide data protection due to excess noise. For the same cost in hardware, by switching to a MRF nibble-protection approach, a noise-immune Hamming decoder system is possible.

4. CONCLUSIONS

As devices are sized down to the nanoscale and supply voltage scale down below 0.5V, circuit designs will need to account for significant signal noise in order to guarantee reliable computation. Previous works have proposed probabilistic computation as a means of dealing with signal noise; however, at a high area overhead. We have demonstrated that probabilistic computation based on MRF principles may be implemented efficiently in CMOS circuitry. Our new MRF mapping provides over a 60% reduction in area and power dissipation compared to MRF-based implementations presented in previous work. For the specific example of a (6,3) Hamming decoder, the new MRF technique provides enhanced reliability at lower power consumption with practically no area overhead.

5. REFERENCES

- [1] H. Iwai. *The future of CMOS downscaling*, chapter in: S. Luryi, J. M. Xu, and A. Zaslavsky, eds., *Future Trends in Microelectronics: The Nano, the Giga, and the Ultra*, pages 23–33. New York: Wiley, 2004.
- [2] D. Bhaduri and S. Shukla. Nanoprism: A tool for evaluating granularity vs. reliability trade-offs in nano architectures. In *Proc. GLSVLSI*, April 2004.
- [3] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes. Accurate reliability evaluation and enhancement via probabilistic transfer matrices. In *Proc. DATE*, March 2005.
- [4] R. I. Bahar, J. Mundy, and J. Chen. A probabilistic-based design methodology for nanoscale computation. In *Proc. 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 *Proc. DAC*, June 2005.
- [6] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky. Designing MRF based error correcting circuits for memory elements. In *Proc. DATE*, March 2006.
- [7] Available at <http://www-device.eecs.berkeley.edu/~ptm/>.