JOURNAL OF ELECTRONIC TESTING: Theory and Applications 23, 255–266, 2007
© 2007 Springer Science + Business Media, LLC. Manufactured in The United States.
DOI: 10.1007/s10836-006-0553-9

Designing Nanoscale Logic Circuits Based on Markov Random Fields

K. NEPAL, R. I. BAHAR, J. MUNDY, W. R. PATTERSON AND A. ZASLAVSKY
Brown University, Division of Engineering, Providence, RI 02912, USA
Kundan_Nepal@brown.edu
Iris_Bahar@brown.edu
Joseph_Mundy@brown.edu
William_Patterson_III@brown.edu
Alexander_Zaslavsky@brown.edu

Received March 8, 2006; Revised July 11, 2006

Editor: M. Tehranipoor

Abstract. As devices and operating voltages are scaled down, future circuits will be plagued by higher soft error rates, reduced noise margins and defective devices. A key challenge for the future is retaining high reliability in the presence of faulty devices and noise. Probabilistic computing offers one possible approach. In this paper we describe our approach for mapping circuits onto CMOS using principles of probabilistic computation. In particular, we demonstrate how Markov random field elements may be built in CMOS and used to design combinational circuits running at ultra low supply voltages. We show that with our new design strategy, circuits can operate in highly noisy conditions and provide superior noise immunity, at reduced power dissipation. If extended to more complex circuits, our approach could lead to a paradigm shift in computing architecture without abandoning the dominant silicon CMOS technology.

Keywords: reliability, Markov random fields, probabilistic computing, noise immunity, redundancy

1. Introduction

As Si CMOS devices are scaled down into the nanoscale regime, current microarchitecture approaches are reaching their practical limits. Thus far, the semiconductor industry has successfully overcome many hurdles, including the current transition to silicon-on-insulator (SOI) technology [2]. Looking to the future, the next major challenges to Si CMOS include new materials (high-κ and low-κ dielectrics [10]), new device geometries (dual-gate or Fin-FET devices [19]), and further downscaling of devices and supply voltages with attendant difficulties in manufacturing, power dissipation, and economics of commodity manufacturing [10]. The longer-term prospects of digital computation then diverge into two interrelated areas. On the system side, there are the computer architecture issues arising from the problem of integrating billions of transistors at the lowest possible supply voltage, with tremendous constraints on total power dissipation and device reliability. On the device integration front, there is hope that hybrid systems will emerge, combining CMOS FET-based digital logic with any number of alternative devices, ranging from analog circuits, to more exotic alternatives (optical sources and detectors, quantum or molecular transistors, carbon nanotube devices, etc.) all on the same chip [5].

While there is no clear consensus on how far and how fast CMOS will downscale and which of the emerging hybrid technologies will eventually enter production, it is certain that future nanodevices will have high manufacturing defect rates. Further, it is clear that the supply voltage, V_{DD}, will be aggressively scaled down to reduce dynamic power dissipation—$V_{DD} = 0.5V$ is the current prediction for low-power CMOS in 2018 [18], although extrapolations to even lower $V_{DD} = 0.3V$ have appeared in the literature [5]. The resulting reduction in noise margins will expose computation to higher soft error rates.

Probabilistic computing provides a new approach towards building fault-tolerant nanoarchitectures and systems. We propose a new CMOS-compatible approach to the design and operation of logic circuits, where the logic states are considered to be random variables whose values can vary over the range of the logic signal level between 0

V and V_{DD}. Under this framework, one no longer expects a correct logic signal at all nodes at all times, but only that the joint probability distribution of signal values has the highest likelihood for valid logic states. The random logic variables for a circuit interact through a distribution representing their joint probability. Circuit design is guided by the formulation of a multivariate distribution on vectors of logic variables, aiming for a distribution that attains maximum probability for valid states of the circuit.

In a circuit with hundreds of logic variables it is impractical to directly consider a joint probability distribution. The number of constraints required to enforce maximum probability for the valid states grows exponentially with the dimension of the random vector space and so the computation quickly becomes intractable. Fortunately there exists a representation for high dimensional joint distributions that can be factored into low dimensional distributions known as the Markov random field (MRF) [3, 7].

In this paper, we describe how logic circuits may be designed using CMOS elements, based on principles of Markov Random Fields, such that correct logic operation may be obtained even under extremely noisy conditions. We show that with our new design strategy, the circuits provide superior noise immunity and at reduced power dissipation compared to standard CMOS counterparts.

The rest of the paper is organized as follows. A brief overview of the Markov random field theory is presented in Section 2, followed by a description of how the logic functions maybe mapped onto CMOS using ideas from MRF in Section 3. The delay, area overhead, and power consumption of the MRF circuit elements as compared to their CMOS counterparts is discussed in Section 4. A quantitative analysis of the noise immunity of the MRF elements relative to their standard CMOS counterparts is presented in Section 5, followed by conclusions and future work in Section 6.

2. Markov Random Fields: Theory

Before presenting our MRF style circuits, we first provide a brief overview of the Markov random field theory. Consider a set of random variables called *sites*, $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{k}\}$ where each variable, x_i can take on various values called *labels*. The sites in X are related to one another via a neighborhood system ($\mathcal{N}$) defined by a set of variables from $X - \{x_i\}$. This collection of random variables is called a Markov Random Field (MRF) if:

$$P(x) > 0, \quad \forall x \in \mathbf{X} \quad \text{(Positivity)} \tag{1}$$

$$P(x_i|\{\mathbf{X} - x_\mathrm{i}\}) = \mathrm{P}(x_\mathrm{i}|\mathcal{N}_\mathrm{i})\text{(Markovianity)} \tag{2}$$

In other words, a set of random variables form a MRF if all sites have a finite positive probability and the probability of a particular site in the neighborhood depends only on its immediate neighbors to which it is connected by an edge. The edges in the neighborhood represent the conditional dependence between the connected variables in the neighborhood. The conditional probability of a given site in terms of its neighborhood can be formulated in terms of the associated clique of the graph structure. Fig. 1 shows one such neighborhood with one 1^{st}order clique and one 2^{nd} order clique.

Circuit networks can be expressed in terms of such neighborhoods and the interaction of the logic states and variables can be represented as a dependence graph. Fig. 2 shows a simple multi-level circuit and its corresponding dependence graph. In this case, the graph is equivalent to a Markov random field, where the nodes are random logic variables that can hold values ranging from 0 V to V_{DD} and the edges are the conditional dependencies between the variables. Importantly, there is no notion of directed logic flow and causality, just statistical dependence. For instance, if the output of the first NAND gate is at logic 0, then both the inputs are constrained to be at logic 1—i.e., there is a (backward) statistical dependency between the output state and the input state.

All the logic variables, $\{s_0, s_1, s_2, s_3, s_4, s_5\}$, in the example, are varying in a random manner over the range of the voltage levels. The correct logic states are those that maximize their joint probability, i.e., the correct logic operation for the example corresponds to the variables that maximize, $p(s_0, s_1, s_2, s_3, s_4, s_5)$.

In the graph of Fig. 2, three distinct sets of *cliques* (i.e., the sets of fully connected subsets of the nodes in the graph) $\{s_0, s_1, s_3\}$, $\{s_2, s_3, s_4\}$, $\{s_4, s_5\}$ are observed. These cliques represent the local statistical dependencies of the logic states. The crucial factor for probabilistic circuit design is that the full set of nodes (logic variables) in the circuit can be factored into a product of joint probabilities in the set of cliques that describe the local interactions. Using the Hammersley Clifford theorem [1], the joint probability distribution can be written as,

$$p(S) = \frac{1}{Z} \prod_{c \in C} e^{\frac{-U(s_c)}{U_0}} \tag{3}$$

where S is the set of all nodes in the dependence graph, C is the set of cliques, s_c is the set of nodes in a clique c,

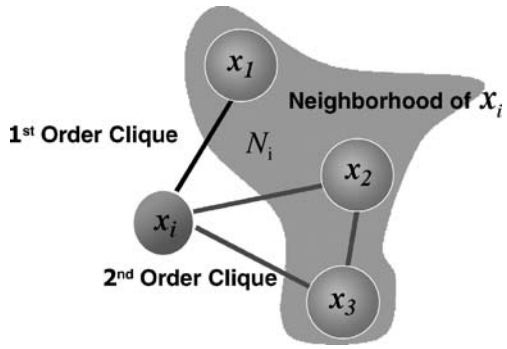

Fig. 1. The MRF neighborhood system.

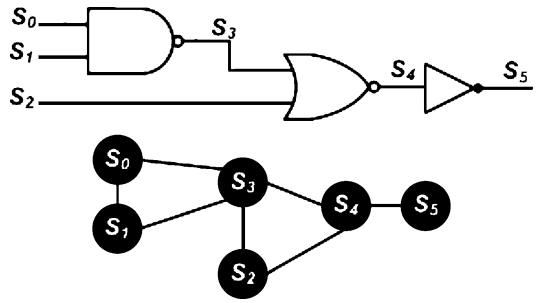

Fig. 2. A logic circuit and its dependence graph for a simple Markov random field.

$U(s_c)$ is the clique energy function also referred to as the logic compatibility function, and U_0 is an abstract term that defines the sharpness of the probability distribution. The term Z is called the *partition function* and is a constant required to normalize the probability function to [0,1].

The general algorithm for finding individual site labels that maximize the probability of the overall network is called *belief propagation* [20] and provides an efficient means of solving inference problems by propagating marginal probabilities through the network. The basic idea of belief propagation is that the probability of state labels at a given node in the network can be determined by marginalizing (summing) over the joint probabilities for the node state given just the probabilities for site labels in the Markov neighborhood, $\mathcal{N}_i$. In our example of Fig. 2 the probability $p(s_0, s_1, s_2, s_3, s_4, s_5)$ can be decomposed into:

$$p(s_0, s_1, s_2, s_3, s_4, s_5) = U(s_0, s_1, s_3)U(s_2, s_3, s_4)U(s_4, s_5) \quad (4)$$

For belief propagation, we start from the primary inputs of the circuit network. As a first step, s_0 and s_1 are eliminated by summing $U(s_0, s_1, s_3)$ over all states of s_0 and s_1 to obtain $U(s_3)$, i.e., s_0 and s_1 are marginalized out. Then s_2 and s_3 can be eliminated by summing $U(s_3)U(s_2, s_3, s_4)$ over all states of s_2 and s_3, giving $U(s_4)$. Finally, s_4 can be eliminated similarly to obtain $U(s_5)$.

This example illustrates that achieving the correct state configuration in the network corresponds to propagating state values through the network and updating each node assignment with a node state having the maximum probability. The great advantage of the Markov network model is that this probability is maximum when the total clique energy is a minimum.

Consider the inverter from Fig. 2 with input s_4 and output s_5. Successful operation of this inverter is designated by the compatibility function $f(s_4, s_5)$ as shown in Table 1. This compatibility function or the clique energy function syntactically describes the interaction of the neighboring nodes represented in the circuit dependence graph.

Here we list all possible states (input–output pairs): valid states with $f = 1$ and invalid states with $f = 0$. The valid input–output pair should have a lower energy than the invalid states. Thus, the clique energy expression is obtained by a negative sum over minterms from the valid states,

$$U(s_4, s_5) = -\sum_i f_i(s_4, s_5) = -(s_4 s_5' + s_4' s_5) \quad (5)$$

For the two valid states $\{01, 10\}$ of the inverter, the clique energy is -1 while for the two invalid states $\{00, 11\}$ the clique energy is 0. As long as the energy of the correct logic state configurations is less than that of the invalid state configurations, the logic element will operate correctly. Although the example of a two-state inverter may appear trivial, similar clique energy expressions may be written down for all elementary logic elements.

3. Building MRF Elements in CMOS

As described in the previous section, the key underpinning of the MRF circuit behavior is that the logic states of network nodes need to depend, in a probabilistic fashion, on the logic states of some finite number of neighboring nodes. For the purposes of probabilistic computation based on interacting nanodevices, we need to find a physical embodiment of interacting logic levels and the clique energy function. In principle, these could be encoded in many physical variables, from occupation of quantum dots by single electrons with occupation probability of neighboring dots mutually influenced by their Coulomb repulsion [9], to the orientation of magnetic spins influencing each other via the exchange interaction. However in this paper, we present the MRF computational paradigm in CMOS Si technology. By choosing the CMOS route, we can use proven device and circuit simulation techniques to more easily examine the higher-level architectural implications of probabilistic computing, including the power consumption, speed and fault tolerance of our circuits.

In the preceding section we discussed clique energy minimization to obtain correct circuit operation. This energy minimization can be achieved by a device or device configuration that produces a bistable energy function. A binary flip-flop circuit possesses this desired energy

Table 1. The logic compatibility function for an inverter with all possible states.

s_4	s_5	f
0	0	0
0	1	1
1	0	1
1	1	0

behavior where the required asymmetry of state energy is created by the summing mechanism just described. As such, the mapping of MRF model into CMOS circuitry requires the following two essential ingredients [13]:

- Each logic state, s_i, should be represented as a *bistable storage element*, taking on logical values of "0" an "1" with equal probability. The probability for any other signal value should be low.
- The constraints of each logic graph clique should be *enforced by feedback* to the appropriate storage elements, implementing the logic compatibility functions to maximize the joint probability of the correct logical values.

The first requirement ensures that the MRF logic states are maintained so that the conditional probabilities among the neighboring elements can propagate. The feedback paths, required by the second design principle, are based on conditional probabilities and ensure that the correct logic states are the most probable states. Each logic state, valid or invalid, has a probability distribution associated with it. In the absence of feedback the probability distribution of the circuit would be uniformly distributed between all states. Whereas the bistable element allows us to maintain a particular logic state at a given node, the feedback mechanism allows us to model the belief propagation and the dependence of a node on the state of its neighborhood, as described in Section 2.

3.1. MRF Inverter

For combinational circuits, this notion of feedback can be enforced by realizing the relationship between inputs and outputs of each gate or function. For example, consider the inverter of Fig. 2 with input variable s_4 and output variable s_5 and logic compatibility function or clique energy described by Eq. 5. Following the recipe for mapping MRF networks into CMOS structures, we can create a bistable structure with feedback reinforcement that represents the clique energy function of the inverter relation using CMOS logic gates. The MRF implementation of the inverter is shown in Fig. 3.

The circuit consists of two "storage nodes," one for s_4 and one for s_5. The stable states of the nodes correspond to the maximum probability configurations of the variables. For example, suppose that $s_4 = 0$ and $s_5 = 1$. Then the top NAND-inverter gate is active and feeds the logic state "1" back to the inputs, thereby reinforcing the expected output value. The other NAND-inverter gate feeds back the logic "0" state. These feedback values are consistent with the input values $\{s_4, s_5\}$ and the overall circuit latches into this state. The other configuration, $s_4 = 1$ and $s_5 = 0$, corresponding to the other valid inverter logic state, is also stable.

The MRF inverter and all of the more complex MRF gates and circuits described later in this paper were

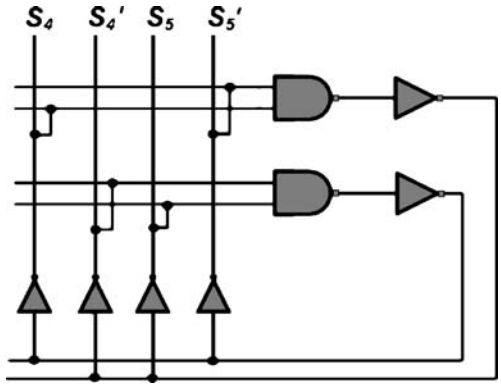

Fig. 3. A circuit for encoding the clique function of two logic variables defining an inverter.

simulated in SPICE using the 70 nm Berkeley predictive technology model (Available at http://www-device.eecs.-berkeley.edu/~ptm/) at $T = 100°C$. In order to simulate the aggressively scaled V_{DD} of future circuits, as well as noisy environment that will plague the end-of-the-road-map CMOS, we operated our MRF gates at a supply voltage of 0.15 V—a voltage level below the threshold voltages of our transistor models, which are V_{TH}=0.2 and −0.22 V for NMOS and PMOS, respectively.

We ran two sets of simulations. First, we simulated the output of the circuit for a noisy input signal in comparison with the standard CMOS gates. Second, we simulated the effect of V_{TH} variation on the MRF element. We emphasize that the sources of signal noise in ultimate transistors are a subject of current research. Some noise sources, e.g., hot-electron effects, cannot be treated analytically even for standard supply voltages but rather require Monte Carlo techniques. On the basis of such simulations, some authors have argued that current noise models will underestimate noise levels in nano-devices [15]. Since we propose to run our circuits at very low V_{DD}, both thermal noise and hot-electron effects, as well as power supply and electromagnetic coupling noise will significantly degrade the logic voltages, while substantial and unavoidable V_{TH} variation [12] between transistors will reduce the noise margins.

An estimate of the noise on a typical signal arising from thermal noise aggravated by threshold variation can be obtained in SPICE by transient simulation of a chain of standard CMOS inverters. A sample of bandwidth-limited random noise of magnitude and spectrum determined from the steady-state noise of the Berkeley transistor model was added to the output of each of ten inverter stages in tandem, with thresholds V_{TH} of individual transistors allowed a random variation of ± 10%. The resulting noise was roughly Gaussian with 30 mV RMS standard deviation. However, the Berkeley model deals with 70 nm planar bulk devices, whereas the future Si technology relies on fully depleted SOI with substantially lower node capaci-

tances. Since noise is inversely proportional to the square root of the node capacitance [16], it is expected to be higher. In addition, our thermal model leaves out crosstalk noise, which will also have a significant effect. While research is underway in trying to accurately model the noise sources in nanoscale CMOS designs [8], we have added Gaussian noise of 0 mean and 60 mV RMS standard deviation to our 0.15 V and 0 voltage levels—a value we believe to be a reasonable estimate for the true signal noise seen by ultimate transistors operated at low V_{DD}.

With this choice of noisy input signals, we have compared the noise immunity for the MRF and CMOS inverters, initially assuming no V_{TH} variation. The inverters are compared in Fig. 4, where it is evident that the noisy input causes the standard CMOS inverter to switch between correct and incorrect output values, due to the small noise margin at low V_{DD} compared to the input noise amplitude. The MRF inverter, on the other hand, provides excellent noise immunity.

We emphasize that simulation illustrated in Fig. 4 assumed noisy input signals, without any V_{TH} variation from transistor to transistor (expected to reduce the noise margins in any large-scale circuit). The expected threshold voltage variation in ultimate CMOS transistors will depend on how the threshold is controlled. Current expectation is that they will have fully depleted undoped Fin-FET channels [10, 18] and V_{TH} will be controlled by the appropriate mid-gap gate material. In order to maintain effective gate control over the potential along the channel, the channel thickness W will need to be smaller than the gate length L_G, so $W < 10$ nm for ultimate CMOS devices. At the same time, W cannot be made too small because size quantization in the channel renders V_{TH} very sensitive to any variation in W [4, 17]. A monolayer fluctuation in W would lead to several millivolts variation in V_{TH}. As a result, in the following simulations we chose a worst-case $\pm 10\%$ (that is, ± 20 mV) variation in V_{TH}.

Given the larger transistor counts, the immunity of the MRF inverter in Fig. 3 to V_{TH} variation is not self-evident, but our preliminary simulations, shown in Fig. 5

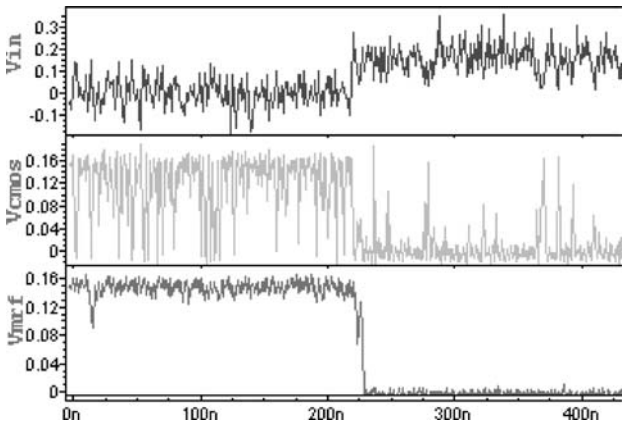

Fig. 4. Simulation of standard CMOS inverter and MRF inverter operation at subthreshold supply voltage.

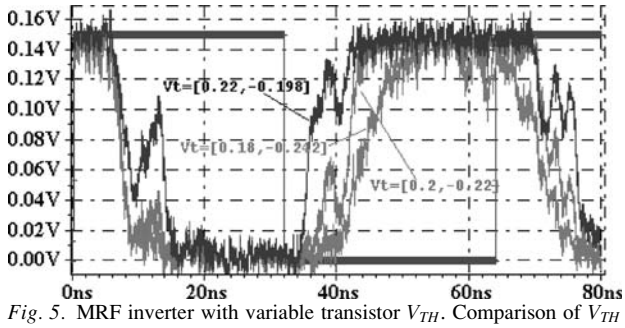

Fig. 5. MRF inverter with variable transistor V_{TH}. Comparison of V_{TH} values=0.2 and -0.22 V (standard) with worst-case $\pm$ 20 mV variation (10% of V_{TH}) for all transistors.

are reassuring. Fig. 5 compares MRF inverter operation for V_{TH}=0.2 and -0.22 V model values with the worst-case situation of ΔV_{TH}=20 mV in all transistors but with N and P devices changing in opposite senses. In all cases, the MRF inverter operates correctly.

The MRF implementations analogous to Fig. 3 provide correct probabilistic operation at low V_{DD} in the presence of noise that would ordinarily defeat standard CMOS. Nevertheless, it is instructive to compare this implementation with other implementations that also have noise-immunity characteristics. For example, consider a gate based on differential cascode voltage switch (DCVS) logic. By virtue of its differential operation and positive feedback, DCVS has some built-in noise immunity. Fig. 6 compares the DCVS inverter (see inset for layout) to our MRF inverter of Fig. 3, in the presence of the same noisy input signals as in Fig. 4 (i.e., Gaussian voltage noise). We find that the DCVS inverter has much better noise immunity than a standard CMOS inverter, but is still not as stable as our MRF inverter. At the same time, a DCVS inverter requires twice the transistor count of standard CMOS, while our MRF inverter is an order of magnitude higher.

3.2. *MRF NAND Element*

The layout of Fig. 3 suggests a programmable logic array style encoding where different functions can be achieved by varying feedback paths. Logic functions with more variables are implemented by feedback paths involving NAND/NOR gates with larger fan-in, and complex feedback elements. Here we use a 2-input NAND gate to illustrate the design of an MRF element with a three-node clique function.

Consider the truth table of a two-input NAND gate shown in Table 2. Again all valid states in the table are labeled with f=1. The clique energy function of this three-node gate can be obtained as:

$$U_c(x_0, x_1, x_2) = x_0' x_1' x_2 + x_0' x_1 x_2 + x_0 x_1' x_2 + x_0 x_1 x_2' \quad (6)$$

The clique energy function shows that there are a total of four minterms for the NAND2 element. Each minterm is a valid input–output pair whose probability must be maxi-

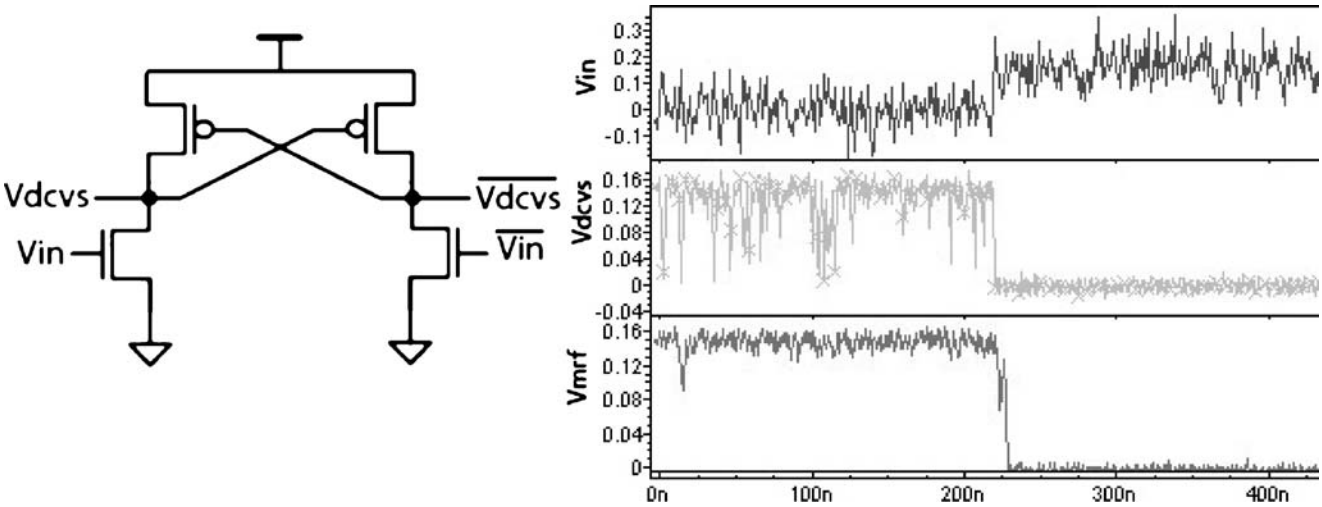

Fig. 6. Comparison of DCVS (*top*, with *inset* showing the DCVS transistor layout) and MRF inverters operated at V_{DD}=0.15 V, given noisy voltage inputs (Gaussian noise with 0 mean and 60 mV RMS amplitude). Note that DCVS provides some noise immunity over standard CMOS, but not as much as MRF.

mized using a bistable storage element. The feedback circuitry becomes slightly more complicated compared to the previous example. The feedback to x_2' comes from the first three minterms containing x_2, while the feedback to x_2 comes only from the final minterm containing x_2'. Since more than one minterm can determine the state of a logic variable, a more complex feedback network consisting of NOR logic gates are needed as shown in Fig. 7. The circuit suggests that a bistable element is required for each minterm. 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. Fortunately, an alternate mapping of the MRF elements described below provides better efficiency in terms of area and power and allows for creation of larger fan-in elements.

The clique energy function of Eq. 6 for the NAND gate can be re-expressed as:

$$U_c(x_0, x_1, x_2) = (x_0' + x_1')x_2 + x_0 x_1 x_2' \qquad (7)$$

Table 2. The logic compatibility table for a two-input NAND gate as a function of all possible input–output pairs.

x_0	x_1	x_2	**f**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The inputs are x_0 and x_1, the output is x_2.

Using this factored form of Eq. 6, a more efficient mapping of the NAND gate can be created as shown in Fig. 8.

The new mapping consists of an 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 four *(for the four minterms)* to just two. This decrease also reduced the complexity of the feedback path. In our earlier approach, 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

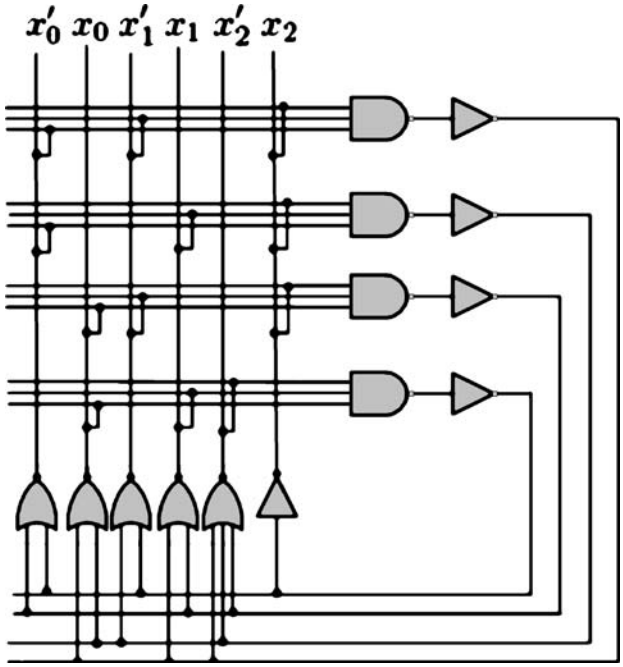

Fig. 7. Implementation of the MRF NAND2 element. The inputs are x_0 and x_1, the output is x_2.

Eq. 6). This feedback complexity 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 Fig. 7.

The simulation of the optimized MRF NAND element of Fig. 8 and its comparison to standard CMOS when subjected to uncorrelated noisy inputs is shown in Fig. 9. 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 and correct voltage operation and excellent noise immunity.

It should be noted that, for all MRF elements, the presence of a feedback loop in the circuit can result in the circuit oscillating between valid states. This oscillation behavior (although not desirable in an electrical circuit) is consistent with the theory of belief propagation in a loop, where the loop is not guaranteed to settle in a particular state but can oscillate between valid states [11]. The feedback components must be sized properly to ensure that no oscillation or metastable states are possible at the supply voltage being used. In our circuits, all feedback to the circuit output are sized slightly larger to eliminate the possibility of any metastable states that might arise due to contention between the input and feedback.

Using this factorization technique, higher fan-in circuits can be created without exponentially increasing the circuit area and complexity. Table 3 shows the transistor counts for different gates (given as a function of its inputs) mapped into MRF elements.

The fan-in of the MRF elements is limited only by the maximum number of transistors connected in series in their transistor-level implementations. For instance, in the 2-input MRF NAND element shown in Fig. 8, a 3-high stack is required to implement the OAI gate within the element. In general, an N-input MRF element would require at most $(N+1)$ transistors in series in the transistor-level implementation. While larger logic functions could be realized using higher transistor stacks, for practical purposes this is generally not preferred. When all the devices in the stack are turned on and conducting, the threshold voltage of each device effectively increases due to the stack effect and causes the drive current to decrease. This, coupled with the fact that all our circuits are operating at a subthreshold voltage regime prompted us to limit the maximum stack size of all our circuits to four transistors.

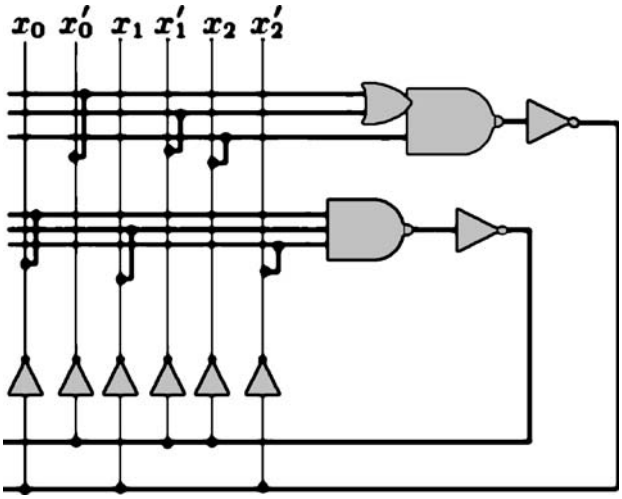

Fig. 8. Area efficient MRF NAND gate implementation (total transistor count is 28 compared to 60 of the mapping shown in Fig. 7). The inputs are x_0 and x_1, the output is x_2.

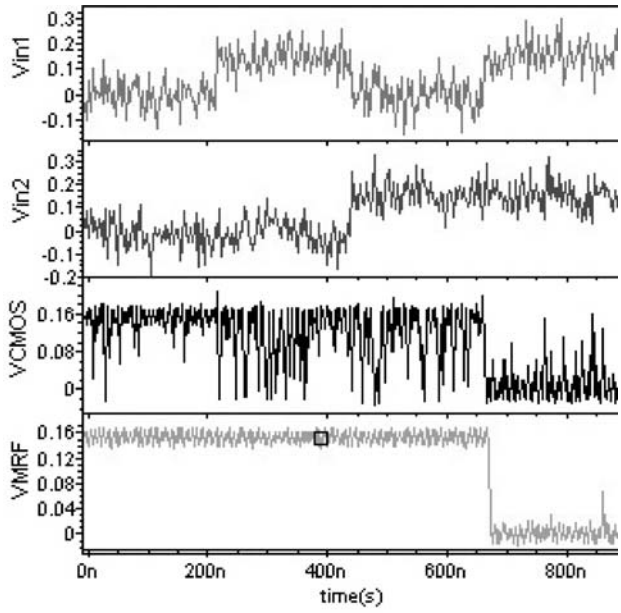

Fig. 9. Simulation of regular static CMOS NAND and optimized MRF NAND gate in presence of noise.

3.3. *Circuits with Multiple Outputs*

So far we have looked at simple logical elements. Often in real designs we encounter circuits that have multiple outputs. Usually these multiple outputs are all a function of the same primary inputs of the circuit. Consider a circuit with inputs p, q and outputs x and y. The output x is defined by the logical AND of the two inputs, i.e., $x = p.q$

Table 3. Comparison of transistor counts for multiple-input standard MRF elements.

Std. gates	MRF mapping
1-input	20
2-input	28
3-input	36
4-input	44
5-input	48

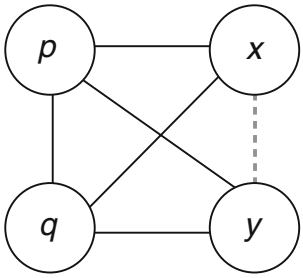

Fig. 10. A graph representing a circuit with multiple outputs. p, q are the inputs and x, y are the outputs.

and y is defined as $y = p + q'$. The clique energy function for these two relations can be written as:

$$U_c(p,q,x) = pqx + x'(q' + p') \tag{8}$$

$$U_c(p,q,y) = p'qy' + y(p + q') \tag{9}$$

The dependence graph in Fig. 10 shows the relationship between the inputs and the respective outputs. The solid lines in the graph show a dependence of the two separate outputs x, and y on the inputs. There is also a dashed line between outputs x and y. The dashed line between the two outputs represents an implied dependence between the two outputs. For example, the logical state of output x is directly dependent on inputs p and q. But the state of inputs p and q is also directly dependent on y. That means if output y was set to a 0 then inputs p and q would have to be logic 0 and 1, respectively. These two inputs being in those states means that x has to be at logic 0. This implied dependence between all the nodes of the dependence graph adds some degree of complexity but it also has an advantage. The main advantage here is that instead of treating these two outputs as two separate entities with two different clique functions, we can treat

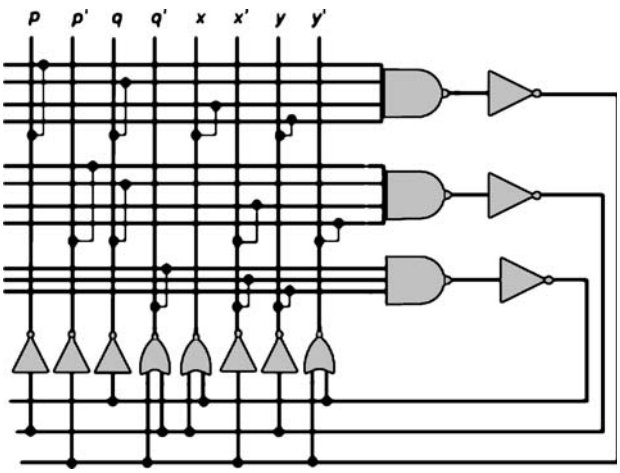

Fig. 11. MRF encoding of clique energy function shown in Eq. 10 p, q are the inputs and x, y are the outputs.

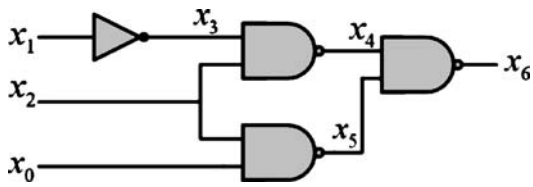

Fig. 12. A multi-level circuit.

the entire system as one large entity governed by a fourth-order clique function consisting of nodes p, q, x and y.

$$U_c(p,q,x,y) = pqxy + p'qx'y' + q'x'y \tag{10}$$

Using this combined clique energy function, we can now create an MRF mapping for the circuit the same way as we did for the MRF inverter and NAND2 elements. The circuit encoding is shown in Fig. 11. The total number of transistor required to implement this combined clique energy function is 50 compared to 84 if the individual clique energies were separately mapped.

3.4. Building Larger Circuits

The MRF approach can be generalized to larger combinations of logic gates. Consider the circuit shown in Fig. 12 which implements the function $x_6 = x_2(x_0 + x_1')$. Larger multilevel circuits such as the one shown here can always be built by cascading MRF elements like the MRF inverter and the MRF NAND gate introduced earlier. The noise tolerance of the individual MRF element cascaded to form such multi-level elements will result in a reliable signal at the output x_6. However, the total cost of this reliability in terms of transistor count is 104, which is a large area penalty to pay for what is a 14 transistor circuit in regular static CMOS.

Instead of cascading individual MRF elements naively, one can take advantage of the fact that not all internal nodes are important. If the circuit designer were interested only in the primary inputs and primary outputs of the circuit and did not care about the internal nodes, one could do away with cascading MRF gates. In the circuit shown in Fig. 12, the important nodes are just the inputs x_0, x_1, x_2 and the output x_6. As such, one can write the clique energy function directly for the circuit, ignoring all internal nodes, as: $U_c = x_6x_2(x_0 + x_1') + x_6'(x_0'x_1 + x_2')$. Implementing this clique energy function directly requires a total of only 36 transistors compared to 104.

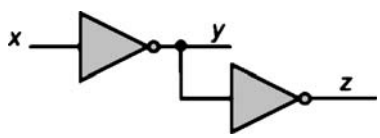

Fig. 13. An inverter cascade.

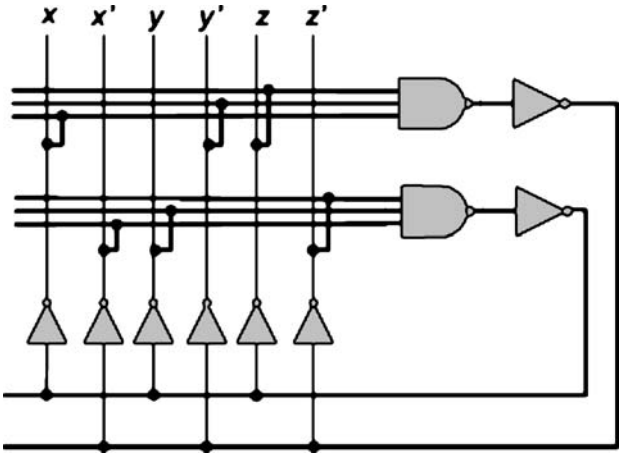

Fig. 14. Clique encoding of the inverter cascade.

While this decrease in transistor count is impressive, one cannot always ignore the internal nodes of a circuit. Consider a cascade of two inverters as shown in Fig. 13 for an application where the output of the first inverter as well as the second inverter is equally important. In such a case, one cannot ignore the middle node y. One possible solution is again to take just two MRF inverter elements and cascade them together. The total cost of this implementation would be 40 transistors. However, instead of cascading the two MRF inverter elements, we can again look at this circuit in terms of implied dependence. Signals x and y have an explicit dependence while y and z also have an explicit dependence. In such a case, the dependence of x and z is implicitly defined. Hence, an MRF encoding can be built by sharing the common label y as shown in Fig. 14. Here the total cost in terms of area is just 28 transistors.

4. Power, Area and Delay Analysis

In this section, we compare the power dissipation and area overhead in terms of transistor counts of MCNC'91 combinational benchmark circuits mapped onto our optimized MRF elements operating at 150 mV and regular static CMOS gates running at 1 V (the expected V_{DD} for 70 nm technology) [18].

Table 4 shows the comparison between the standard CMOS and MRF implementations in terms of the number of transistors and power consumption under noisy conditions. 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). For all our circuit simulations, because of the subthreshold operation of our circuits, we limit the maximum stack size to four transistors. Hence, the MRF approach used two and three input MRF elements when mapping the benchmark circuits to gates.

The table shows that the average increase in area for the MRF implementation is about 4.7 times the static CMOS implementation. At the same time, other redundancy approaches like Triple Modular Redundancy (TMR) are not that far lower (3× overhead for a fine-grained implementation). Regardless of whether a fine-grained or coarse-grained approach is used, the TMR method needs a reliable and a noise-free final majority gate. The MRF elements, however, have no such requirements. The effectiveness of the MRF approach compared to the TMR and the cascaded TMR approach was shown in [14]. In the presence of extreme noise and single-bit errors, the MRF approach produced correct output and showed superior immunity to noise compared to TMR.

The results in Table 4 also show that the MRF mapping provides a power advantage compared to static CMOS gates, particularly for circuits with larger depth and many transistors in the first stage. Specifically, the MRF implementation dissipates on average 33% less power than the standard CMOS implementation for these larger circuits (e.g., *alu*4, *cordic, ex*5, and *table*5). This is significant, since this implies that our MRF elements may be used more effectively in larger circuit designs. For circuits with

Table 4. Comparison of transistor counts and power for MCNC'91 benchmark circuits.

			CMOS V_{DD}=1 V				MRF mapping	
Circuit	In	Out	$\#tran$	$1^{st}-stage$	$depth$	$power(\mu W)$	$\#tran$	$power(\mu W)$
5xp1	7	10	568	25	10	101.4	2756	151.2
alu4	14	8	6928	153	23	875.2	33416	612.1
con1	7	2	78	6	6	16.5	356	16.9
cordic	23	2	604	32	15	89.8	2612	54.7
ex5	8	63	5448	135	13	692.5	25964	506.9
misex1	8	7	356	11	7	69.6	1700	82
o64	130	1	520	65	8	24.7	2752	44.5
rd53	5	3	232	6	9	40.7	1012	46.3
squar5	5	8	346	10	8	55.6	1532	70.1
table5	17	15	10192	237	23	1522.5	47948	936.1

Table 5. Delay of CMOS and MRF elements normalized to the delay of a CMOS inverter operated at $V_{DD} = 0.15V$.

Circuit	Delay
CMOS INV	1.0
CMOS NAND2	1.6
MRF INV	7.1
MRF NAND2	8.6

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.

For sake of completeness, a quick glance at the propagation delay of the MRF elements is also provided. As the power supply is decreased into the subthreshold operation region, the propagation delay of a circuit increases significantly. The increase in delay for the MRF elements shown in the previous section is even more obvious because of the increased level of circuitry and the existence of feedback paths. The feedback paths add capacitance at the output node and the contention between the input and the feedback values causes an increase in the latency of the circuit. Table 5 shows the delay of the MRF elements normalized to a CMOS inverter operated at the same subthreshold voltage of 150 mV.

Depending on where and how these MRF elements are used, some path in the circuit may be able to tolerate the increase in delay. However, the problem will remain one with reliability, power and delay tradeoffs.

5. Quantifying Noise Immunity

In Section 3, we showed the noise tolerance of MRF elements compared to their CMOS counterparts. Here we quantify the circuit's tolerance to noise. An appropriate measure of the discrepancy between the actual output signal probability of a logical element or circuit P_{real} and the ideal (correct) output P_{ideal} is the Kullback–Leibler distance (KLD) [6]. For a digital system with two levels ("0" and "1"), the KLD is the measure of the distance between P_{real} and P_{ideal} (where output is sampled and noise leads to some probability of finding an incorrect output value):

$$KLD(P_{ideal}, P_{real}) = \sum_{states} P_{ideal} \log_2\left(\frac{P_{ideal}}{P_{real}}\right) \qquad (11)$$

where the smaller the KLD, the better the noise immunity of the circuit. By sampling the output voltage at discrete points we can quantitatively compare the noise immunity of our simple logic elements. For the KLD calculation the voltage values are sampled at 0.1 ns, because this time is much smaller compared to the switching time of our MRF elements. A comparison of standard CMOS, DCVS and MRF inverters and NAND gates is shown in Table 6. Clearly, the MRF implementations have much better noise immunity as measured by the KLD (for perfectly correct operation, the KLD is 0; see Eq. 11).

Table 6. Comparison of Kullback–Leibler distance from correct (noise-free) output of unloaded CMOS, DCVS, and MRF logic elements fed with noisy input voltages.

	INV	NAND
CMOS	3.404	3.7947
DCVS	2.1832	3.6608
MRF	0.5878	0.4126

We have also carried out the same noise immunity simulations for several larger benchmark circuits, each with two different implementations; one based on our MRF circuits and the other based on "standard" CMOS gates. The KLD values were computed by creating a probability distribution averaged over all primary outputs. As can be seen in Table 7, the KLDs for the MRF circuits are much smaller than those of the standard CMOS circuits, indicating that the probability distributions of the MRF gates more closely mimic the ideal output probability distributions.

6. Conclusions and Future Work

As devices are sized down to the nanoscale and supply voltage is scaled down below 0.5 V, circuit designs will need to account for significant signal noise in order to guarantee reliable computation. The MRF probabilistic

Table 7. Comparison of Kullback–Leibler distance from correct (noise-free) output of MRF and standard CMOS benchmark circuits (run at V_{DD}=0.15 V, T=100 °C).

Circuit	CMOS	MRF
5xp1	1.23	0.23
alu4	0.76	0.39
con1	1.03	0.20
cordic	0.60	0.33
ex5	1.18	0.50
misex1	1.00	0.24
o64	0.85	0.37
rd53	0.98	0.11
squar5	1.13	0.28
table5	0.90	0.34

model provides a framework for designing CMOS circuits that can operate effectively under conditions of ultra-low supply voltage and extreme noise conditions. We have demonstrated that probabilistic computation based on MRF principles may be implemented in CMOS circuitry with much greater reliability in the presence of noisy inputs, but at a cost of larger transistor counts. The MRF circuits may be operated at much lower supply voltage, leading to reduced power dissipation along with improved reliability.

Our immediate goal in the future is to further reduce the area overhead to make the MRF design methodology more viable for large circuits. Our ultimate goal is to develop a noise-aware logic synthesis and technology mapping tool. Given a functional description of a circuit, the tool will produce an error-tolerant design that balances area, power, delay, and reliability constraints when generating the final mapped circuit. In addition, work is also underway in trying to accurately model the noise sources in nanoscale CMOS designs.

References

1. J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. R. Stat. Soc., Ser. B*, vol. 36, no. 3, pp. 192–236, 1994.
2. G.K. Celler and S. Cristoloveanu, "Frontiers of Silicon-on-insulator," *J. Appl. Physi.*, vol. 93, pp. 4955–4978, May 2003.
3. R. Chellappa, *Markov Random Fields: Theory and Applications*, Academic, 1993.
4. T. Ernst, S. Cristoloveanu, G. Ghibaudo, T. Ouisse, S. Horiguchi, Y. Ono, Y. Takahashi, and K. Murase, "Ultimately Thin Double-gate Soi Mosfets," *IEEE Trans. Electron Devices*, vol. 50, pp. 830–838, March 2003.
5. 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*, New York: Wiley, 2004, pp. 23–33.
6. S. Kullback, *Information Theory and Statistics*, New York: Dover, 1969.
7. S.Z. Li, *Markov Random Field Modeling in Computer Vision*, Berlin Heidelberg New York: Springer, 1995.
8. H. Li, J. Mundy, W.R. Patterson, D. Kazazis, A. Zaslavsky, and R.I. Bahar, "A Model for Soft Errors in the Subthreshold Cmos Inverter," in *Proceedings of Workshop on System Effects of Logic Soft Errors*, Nov. 2006.
9. K.K. Likharev, "Single-electron Devices and their Applications," *Proc. I.E.E.E.*, vol. 87, no. 4, pp. 606–632, April 1999.
10. S. Luryi, J.M. Xu, and A. Zaslavsky eds. *Future Trends in Microelectronics: The Nano, the Giga, and the Ultra*. New York: Wiley, 2004.
11. K. Murphy, Y. Weiss, and M. Jordan, "Loopy Belief Propagation for Approximate Inference: an Empirical Study," in *Proceedings of Uncertainty in AI*, pp. 467–475, 1999.
12. S. Narendra, V. De, S. Borkar, D.A. Antoniadis, and A.P. Chandrakasan, "Full-chip Subthreshold Leakage Power Prediction and Reduction Techniques for Sub-0.18 μm Cmos," *IEEE J. Solid-state Circuits*, vol. 39, pp. 501–510, March 2004.
13. 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 *Proceedings of Design Automation Conference*, pp. 485–490, June 2005.
14. K. Nepal, R.I. Bahar, J. Mundy, W.R. Patterson, and A. Zaslavsky, "MRF Reinforcer: A Probabilistic Element for Space Redundancy in Nanoscale Circuits," *IEEE Micro*, vol. 26, no. 5, pp. 19–27, Sept–Oct.
15. V.M. Polyakov and F. Schwierz, "Excessive Noise in Nanoscaled Double-gate Mosfets: A Monte Carlo Study," *Semicond. Sci. Technol.*, vol. 19, no. 4, pp. 145–147, 2004.
16. R. Sarpeshkar, T. Delbrueck, and C.A. Mead, "White Noise in Mos Transistors and Resistors," *IEEE Circuits Devices Mag.*, vol. 6, pp. 23–29, Nov 1993.
17. E. Suzuki, K. Ishii, S. Kanemaru, T. Maeda, T. Tsutsumi, T. Sekigawa, K. Nagai, and H. Hiroshima, "Highly Suppressed Short-channel Effects in Ultrathin Soi N-mosfets," *IEEE Trans. Electron Devices*, vol. 47, no. 2, pp. 354–359, Feb. 2000.
18. ITRS, http://www.public.itrs.net, 2004 (latest update).
19. H.S.P. Wong, "Beyond the Conventional Transistor," *IBM J. Res. Develop.*, vol. 46, no. 2–3, pp. 133–168, 2002.
20. J. Yedidia, W. Freeman, and Y. Weiss," Understanding Belief Propagation and its Generalizations," in *International Joint Conference on AI*, 2001. Distinguished Lecture.

Kundan Nepal received the BS degree in Electrical Engineering from Trinity College, Hartford in 2002 and his MSEE from the University of Southern California in 2003. He is currently at the Division of Engineering, Brown University pursuing a Ph. D. degree in Electrical and Computer Engineering. His research interests include fault tolerant computing for nano-architectures, VLSI system design and Computer-aided design of digital integrated circuits.

R. Iris Bahar received the B.S. and M.S. degrees in computer engineering from the University of Illinois, Urbana-Champaign, in 1986 and 1987, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Colorado, Boulder, in 1995. From 1987–1992 she was with Digital Equipment Corporation, responsible for the hardware implementation of the NVAX processor. Since 1996 she has been with the Division of Engineering, Brown University, where she is currently an Associate Professor. Her research interests include computer architecture; computer-aided design for synthesis, verification and low-power applications; and design, test, and reliability issues for nanoscale systems.

Joseph Mundy received the bachelor's degree in electrical engineering from Rensselaer Polytechnic Institute in 1963. He joined General Electric Research in 1963 and received the PhD degree in electrical engineering from Rensselaer Polytechnic Institute in 1969. In his early career at GE he carried out research in solid state physics and integrated circuit devices. In the early 70's he formed a research group in computer vision with emphasis on industrial inspection. His group developed a number of inspection systems for GE's manufacturing divisions including a system for the inspection of lamp filaments that exploited syntactic methods in pattern recognition. During the 1980's his group moved towards more basic research in object recognition and geometric reasoning. In 1988, he was named a Coolidge Fellow, which awarded him a sabbatical at Oxford University. At Oxford he collaborated on the development of theory and application of geometric invariants. During the 1990's his group carried out work in range-based metrology, model-supported change detection for image intelligence, target recognition and video understanding. In 2002, he retired from GE Research and joined the engineering division of Brown University. His current research interests are video-based object recognition and nano architecture.

William R. Patterson received the ScB degree in Physics and the ScM in Electrical Engineering in 1963 and 1966 from Brown University. Since 1977 he has been in the Electrical Sciences group of the Division of Engineering, Brown University, where he is currently a Senior Lecturer and Senior Research Engineer. In 1977 he received the NASA Public Service Medal for contributions to the Viking program. His current research interests include low power analog circuit design for biomedical applications, circuits and architectures for probabilistic logic, and instrumentation for geological spectroscopy.

Alexander Zaslavsky is an Associate Professor of Engineering and Physics at Brown University. His research interests include tunneling semiconductor physics and devices, alternative device materials like carbon nanotubes and germanium-on-insulator, flexible electronics, and CMOS-compatible probabilistic computing. He received an A.B. in physics from Harvard in 1986 and a Ph.D. in electrical engineering from Princeton in 1991; he is a member of the American Physical Society.