Designing Logic Circuits for Probabilistic Computation in the Presence of Noise

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

ABSTRACT

As Si CMOS devices are scaled down into the nanoscale regime, current computer architecture approaches are reaching their practical limits. Future nano-architectures will confront devices and interconnections with a large number of inherent defects, which motivates the search for new architectural paradigms. In this paper, we examine probabilistic-based design methodologies for nanoscale computer architectures based on Markov random fields (MRF). The MRF approach can express arbitrary logic circuits and the logic operation is achieved by maximizing the probability of correct state configurations in the logic network depending on the interaction of neighboring circuit nodes. The computation proceeds via probabilistic propagation of states through the circuit. Crucially, the MRF logic can be implemented in modified CMOS-based circuitry that trades off circuit area and operation speed for the crucial fault tolerance and noise immunity. This paper builds on the recent demonstration that significant immunity to faulty individual devices or dynamically occurring signal errors can be achieved by the propagation of state probabilities over an MRF network. In particular, we are interested in CMOS-based circuits that work reliably at very low supply voltages (V_{DD} = 0.1–0.2 V), where standard CMOS would fail due to thermal and crosstalk noise, and transistor threshold variation. In this paper, we present results for simulated probabilistic test circuits for elementary logic components and well as small circuits taken from the MCNC91 benchmark suite and we show greatly improved noise immunity operating at very low V_{DD}. The MRF framework extends to all levels of a design, where formally optimum probabilistic computation can be implemented as a natural element of the processing structure.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-tolerance

General Terms

Design, Reliability, Emerging technologies

Keywords

noise immunity, reliability, subthreshold operation, probabilistic computing, Markov random fields, nanodevices

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

1. INTRODUCTION

For several decades, mainstream silicon technology has relied on scaling down CMOS transistors following Moore's Law. Thus far, the semiconductor industry has successfully overcome many hurdles, including the current transition to silicon-on-insulator (SOI) technology [1]. Looking to the future, the next major challenges to Si CMOS include new materials (high-κ and low-κ dielectrics [2]), new device geometries (dual-gate or fin-FET devices [3]), and further downscaling of devices and supply voltages with attendant difficulties in manufacturing, power dissipation, and economics of commodity manufacturing [2].

Further into the future, the International Technology Roadmap for Semiconductors (ITRS) predicts that the continued shrinkage of individual transistors will stop, perhaps around 2015, due to unavoidable physical limits, with the ultimate transistor gate length near L_G $\sim$10 nm [4]. These ultimate transistors will be nanodevices in the true sense of the word. Working L_G $\sim$6 nm devices have already been reported [5], although there is some debate as to whether the performance gains of such small devices will provide adequate return to justify the enormous fabrication costs. 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 [6].

Currently, exotic materials — from carbon nanotubes to molecular transistors, to spin-based devices and superconducting junctions, to single-electron devices, all the way to DNA-based computing — are being touted as contenders for computational circuitry. According to some benchmarks, many of these devices are quite successful: they may be faster (tunneling-based devices), carry more current (carbon nanotubes), take up less area (molecular transistors), or have higher logic functionality than CMOS FETs. Yet the key issues of compatibility with the enormous installed base of VLSI fabrication tools and know-how, and with the usual operating parameters (room temperature, low-voltage, ultra-low static power consumption) remain elusive. Instead of targeting the integration of non-CMOS nanodevices into computational logic circuits, we propose to examine the performance of ultimate CMOS transistors in a new computing framework — that of probabilistic computing embedded in a Markov random network. The premises underlying this approach are as follows:

- Any computational scheme involving large numbers of nano-

devices (regardless of whether these devices are $L_G \sim$10 nm Si transistors or whether the information is stored in the charge, spin, or even quantum phase of some other material) needs a large degree of fault tolerance. It is inevitable that a significant fraction of devices and interconnections will fail, with failures occurring both during fabrication and operation.

- Since the number of devices in future computational circuits will be enormous, the constraints on power consumption will require the lowest possible supply voltages. As a result, these circuits will operate closer to the thermal limit, where the difference between the logic voltage levels will shrink well below the many tens of kT/q of today's circuits. If so, the very nature of computation will need to become probabilistic.

Probabilistic computing provides a new approach towards building more powerful fault-tolerant nanoarchitectures and systems. The future of computation beyond standard CMOS is highly uncertain, but it is a given that new fault-tolerant paradigms will be needed and CMOS will remain the dominant technology for the next decade or two. If extended to more complex circuits, our approach could lead to a paradigm shift in computing architecture that would still be compatible with real-world technology.

The main contributions of this paper are:

- We consider various forms of noise, including thermal noise and threshold variation, in CMOS devices as these devices approach their ultimate limiting size and we evaluate their impact on circuit reliability. These noise sources have not hitherto been significant but will become so as device size continues to shrink.

- We propose the first technologically realistic CMOS implementation of a new logic style based on the Markov random field (MRF) probabilistic paradigm proposed in [9]. It should be emphasized that while [9] did discuss the theoretical utility of MRF for probabilistic computation in the presence of noise and soft faults, no practical implementation was suggested. In this paper, we propose a MRF implementation using ultimate CMOS operated at ultra-low supply voltages.

- Using circuits from the MCNC benchmark suite, we show through SPICE simulation how our new family of MRF circuits can provide superior soft fault tolerance and noise immunity at very low supply voltages (e.g., 0.15V) compared to a conventional CMOS implementation.

The rest of the paper is organized as follows. The enabling theory of probabilistic computation based on the Markov random field (MRF) approach, is described in Section 2. Our noise-immune circuit designs are presented in Section 3, followed in Section 4 by a quantitative analysis of their noise immunity relative to standard CMOS designs. A discussion on higher architectural level implications is presented in Section 5, followed by conclusions and future work in Section 6.

2. MARKOV RANDOM NETWORKS

The downscaling of devices to the nanoscale, the explosion in the number of devices integrated in a digital computation circuit, and the reduction of logic levels down to the thermal limit, all conspire to produce faulty systems. We can expect the frequent occurrence of both soft faults due to noise and signal coupling, and hard faults due to process variations and defects. As a result, we propose a new approach to the design and operation of logic circuits where the logic states are considered to be random variables. 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. This joint distribution can be considered to be a distribution on random vectors, with a vector element for each logic variable in the circuit. Thus, under this new probabilistic approach, circuit design is guided by the formulation of a multivariate distribution on vectors, aiming for a distribution that attains maximum probability for the valid states of the circuit.

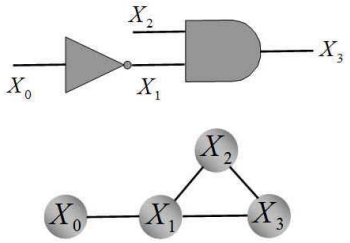

Figure 1: A logic circuit and the corresponding graph. Taken from [9].

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) [7][8]. Previous work has discussed the use of MRF for probabilistic computation in the presence of noise and faults [9]; however, no practical implementation using real devices was suggested. Now, we propose to implement MRF-based probabilistic computation in ultimate CMOS transistors.

An appropriate model for the MRF is a graph structure, where the nodes of the graph represent logic variables and the edges represent statistical dependency between the variables. An example of such a graph is shown in Figure 1 for a very small logic circuit. There are four logic variables represented by the vector $\{x_0, x_1, x_2, x_3\}$ and these variables are the nodes of the graph in the lower half of the figure. The edges of the graph indicate that the subsets of nodes $\{x_0,x_1\}$,$\{x_1,x_2,x_3\}$ directly interact. These subsets are called *cliques* from the graph theory concept of node subsets that are all mutually connected by graph edges. For any MRF graph G there exists a set of cliques C that represent the local statistical dependencies of logic states.

Let the full set of nodes (logic variables) of G be represented by the vector $\mathbf{X}$. The key result that makes the MRF an attractive formulation for probabilistic circuit design is called the Hammersley and Clifford theorem [10]. It can be shown that the joint probability distribution for $\mathbf{X}$ can be factored as,

$$p(\mathbf{X}) = \prod_{c \in C} F_c(x_c) \tag{1}$$

where x_c is the set of nodes in a clique c. The strictly positive functions $F_c(x_c)$ are known as the Gibbs energy functions and represent the joint probability of the clique variables. One form for $F_c(x_c)$ is the Gibbs distribution that represents the probability of states in a physical system,

$$F_c(x_c) = \frac{1}{Z} e^{\frac{-U(x_c)}{kT}} \tag{2}$$

where Z is the partition function and normalizes $F_c(x_c)$ to the range [0, 1]; $U(x_c)$ is the clique energy that depends only on the nodes in the clique; and kT corresponds to the thermal energy. For the purposes of probabilistic computation based on interacting nanodevices, we need to find a physical embodiment of interacting logic levels. 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, to the orientation of magnetic spins influencing each other via the exchange interaction. However, we choose to map the MRF probabilistic computation onto nanoscale CMOS, which is by far the most mature and well-controlled technology.

The key advantage of the MRF approach is that its operation does not depend on perfect devices or connections. In operation, the nodes iteratively change their logic levels and these changes propagate through the network. Successful operation only requires that the joint energy of correct states be lower than the energy of the errors, so a degree of fault tolerance is built in.

As was shown in [9], the Gibbs form of Eq. 2 makes it possible to achieve correct logic operation in the presence of structural faults. Using the *auto-model* form of the clique energy, it can be shown that by adding redundant logic elements and pathways according to the gate's clique energy function, high fault tolerance is achieved. For the inverter example given in [9], up to one third of the redundant connections or devices can be bad without destroying the correct energy relations.

3. MRF CMOS-BASED LOGIC ELEMENTS

For the MRF model to be mapped onto a CMOS circuit, we require two essential ingredients:

- Each logic state, x_i, should be represented as a bistable storage element, taking on logical values of "0" an "1". 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 functions $F_c(x_c)$ to factorize Eq. 1 and maximize the joint probability of the correct logical values.

As the simplest example, consider the inverter represented by nodes x_0 and x_1 in Figure 1. An MRF implementation of the inverter, implementing $F_{01}(x_0, x_1)$ is shown in Figure 2. The circuit consists of two "storage nodes", one for x_0 and one for x_1. The stable states of the nodes correspond to the maximum probability configurations of the variables. For example, suppose that $x_0 = 0$ and $x_1 = 1$. 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 logic "0" state. These feedback values are consistent with the input values $\{x_0, x_1\}$ and the overall circuit latches into this state. The other configuration, $x_0 = 1$ and $x_1 = 0$, corresponding to the other valid inverter logic state, is also stable.

The layout of Figure 2 suggests a programmable logic array style encoding where different functions can be achieved by varying the cross-links. Logic functions with more variables are implemented by feedback paths involving NAND gates with larger fan-in, and changing the inverters to NOR gates.

Clearly, the MRF implementation of logic functions requires many transistors. On the other hand, as our simulations will demonstrate, the MRF implementation provides significant advantages over standard, minimum-transistor CMOS designs with respect to noise immunity, fault tolerance ($e.g.$, imperfect threshold voltage control), and low-voltage operation ($e.g.$, $V_{DD} \sim V_{TH}$ of the transistors). In the simulation results to follow, we compare MRF inverters and NAND gates with their standard CMOS counterparts. The simulations were carried out in SPICE using the 70 nm CMOS Predictive Technology Model from Berkeley [11] at T = 100 °C. These transistor models have threshold voltages V_{TH} of 0.2 V for NMOS and –0.22 V for PMOS.

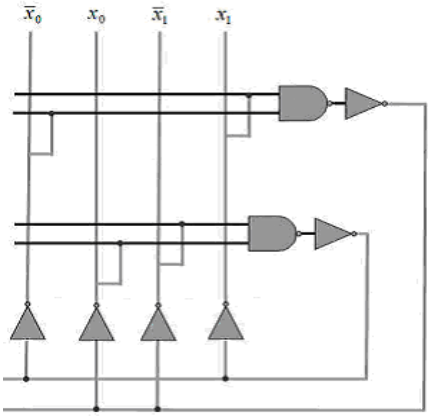

Figure 2: A circuit for encoding the clique function of two logic variables defining an inverter (total transistor count is 20, as compared to two for standard CMOS inverter)

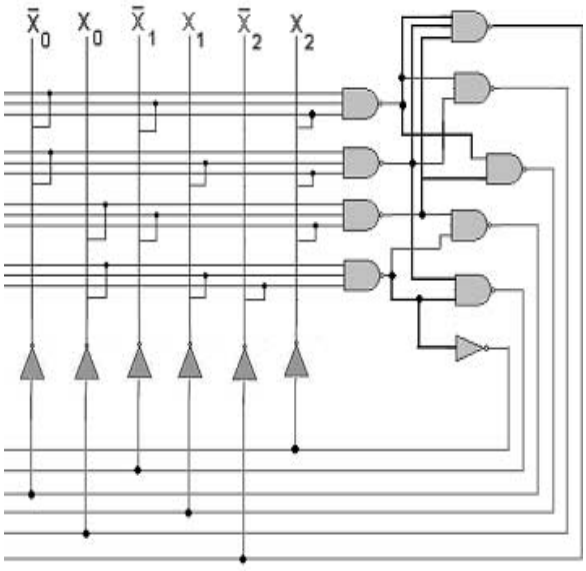

Figure 3: MRF NAND gate implementation (total transistor count is 60). The inputs are x0 and x1, the output is x2.

The MRF inverter is shown in Figure 2 and the MRF NAND gate in Figure 3. We ran two sets of simulations at V_{DD} = 0.15 V, a voltage *below* the $|V_{TH}|$ of our model transistors. First, we simulated the output of the inverter and NAND gate for a noisy input signal in comparison with the standard CMOS gates. Second, we simulated the effect of V_{TH} variation on the MRF circuits. 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 nanodevices [12]. Since we propose to run our circuits at very low V_{DD} and T = 100 °C, 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 [13] between transistors will reduce the noise margins.

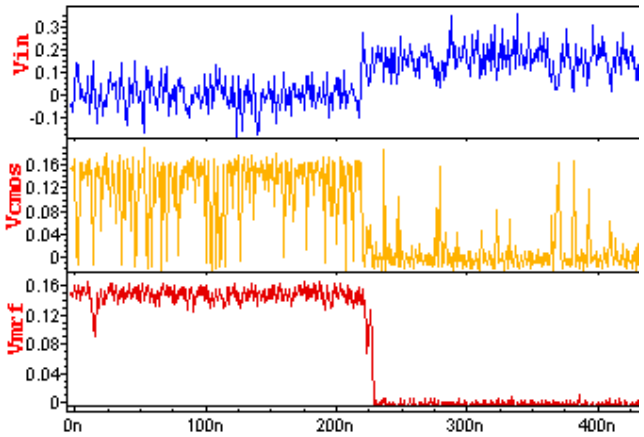

Figure 4: Standard CMOS inverter and MRF inverter operation at subthreshold supply voltage. The MRF inverter output is stable, whereas standard CMOS switches between correct and incorrect output values (no V_{TH} variation is assumed.)

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 10 inverter stages in tandem, with thresholds V_{TH} of individual transistors allowed a random variation of $\pm$ 10%. The resulting noise was roughly Gaussian with 30.2 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 capacitances. Since noise is inversely proportional to the square root of the node capacitance [14], it is expected to be higher. In addition, our thermal model leaves out crosstalk noise, which will also have a significant effect. Lacking realistic noise models for ultimate transistors, we have added Gaussian noise of zero mean and 60 mV RMS standard deviation to our 0.15 V and zero voltage levels — a value we believe to be a reasonable lower bound 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 and NAND gates, initially assuming no V_{TH} variation. The inverters are compared in Figure 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.

The same comparison for NAND gates is shown in Figure 5. Once again, noisy (uncorrelated) inputs cause the standard NAND gate operated at subthreshold V_{DD} = 0.15 V to switch between correct and incorrect output values. The MRF NAND gate provides stable and correct voltage operation, at the cost of much greater

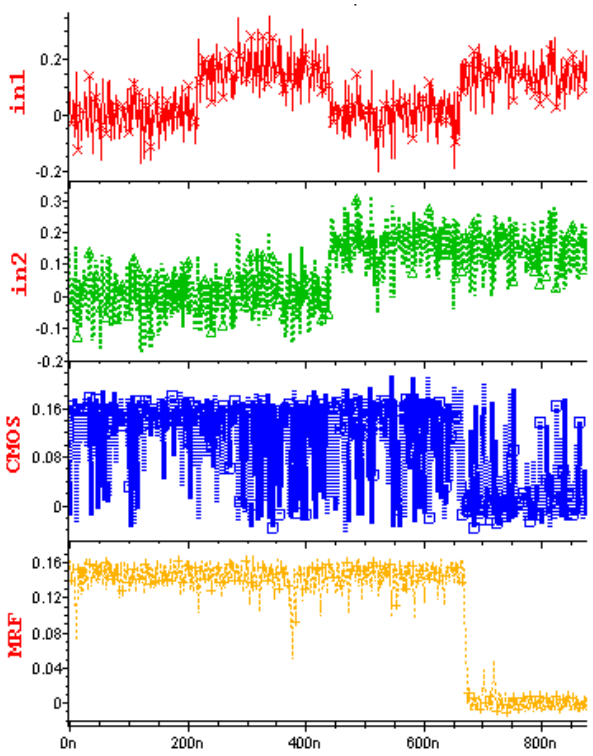

Figure 5: Same comparison for standard CMOS and MRF NAND gates with noisy inputs and no V_{TH} variation.

transistor counts (60 transistors compared to four for the standard NAND).

The MRF implementations analogous to Figures 2 and 3 provide correct probabilistic operation at low V_{DD} in the presence of noise that would defeat standard CMOS. Nevertheless, it is instructive to compare this implementation with other implementations that also have noise-immunity characteristics, as well as smaller transistor counts. 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. Figure 6 compares the DCVS inverter (see inset for layout) to our MRF inverter of Figure 2, in the presence of the same noisy input signals as in Figure 4 (i.e., Gaussian voltage noise). We find that the DCVS inverter has much better noise immunity than a standard CMOS inverter, but is 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.

We emphasize that simulations illustrated in Figures 4 and 5 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. The current expectation is that they will have fully depleted undoped SOI channels [4][1] 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. 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 [15][16]. A monolayer fluctuation in W

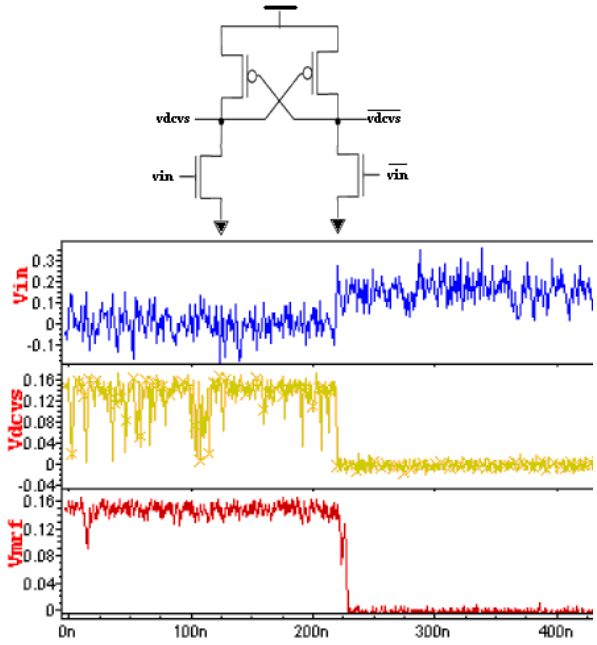

Figure 6: Comparison of DCVS (top, with inset showing the DCVS transistor layout) and MRF inverters operated at VDD = 0.15 V, given noisy voltage inputs (Gaussian noise with zero mean and 60 mV RMS amplitude). Note that DCVS provides some noise immunity over standard CMOS, but not as much as MRF.

would lead to several meV 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 their larger transistor counts, the immunity of MRF circuits in Figures 2 and 3 to V_{TH} variation is not self-evident, but our preliminary simulations, shown in Figure 7 are reassuring. Figure 7 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.

4. QUANTIFYING NOISE IMMUNITY

An appropriate measure of the discrepancy between the actual output signal probability P_{real} of a logical element or circuit and the ideal (correct) output P_{ideal} is the Kullback-Leibler distance (KLD) [17]. For a digital system with two levels ("0" and "1"),

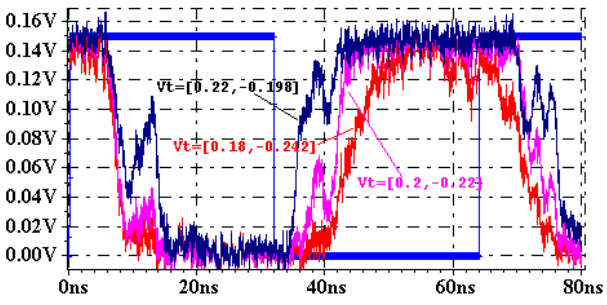

Figure 7: MRF inverter with variable transistor V_{TH}. Comparison of V_{TH} values = 0.2 and –0.22 V (standard) with worst-case $\pm$ 20 mV variation (10% of V_{TH}) for all transistors.

	INV	NAND
CMOS	3.404	3.7947
DCVS	2.1832	3.6608
MRF	0.5878	0.4126

Table 1: Comparison of Kullback-Leibler distance from correct (noise-free) output of unloaded CMOS, DCVS, and MRF logic elements fed with noisy input voltages.

circuits	# gates	inp	out	MRF	CMOS
squar5	95	5	8	0.19	0.60
rd53	67	5	3	0.12	0.25
misex1	97	8	7	0.15	0.33
con1	27	7	2	0.10	0.78
5xp1	150	7	10	0.16	0.96

Table 2: 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).

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):

$$KLD(P_{ideal}, P_{real}) = \sum_{states} P_{ideal} \log_2(\frac{P_{ideal}}{P_{real}}) \qquad (3)$$

where the smaller the KLD, the better the noise immunity of the circuit. By sampling the output voltage at discrete points (every 0.1 ns) we can quantitatively compare the noise immunity of our simple logic elements. A comparison of standard CMOS, DCVS and MRF inverters and NAND gates is shown in Table 1. Clearly, the MRF implementations have much better noise immunity as measured by the KLD (for perfectly correct operation, the KLD is zero, see Eq. 3).

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. Each implementation consisted of inverters, and two-input NAND, AND, NOR, and OR gates. These circuits were selected from the MCNC'91 combinational benchmark set and were simulated using the 70 nm Berkeley predictive technology model [11] at V_{DD} = 0.15 V and T = 100 °C using SPICE. In Table 2, we present the benchmark circuits, the number of gates, the primary inputs and outputs. The gate counts reported in column 2 were obtained after redundancy removal and mapping the gates using the SIS synthesis tool [18], with each logic cell making up the circuit driving a maximum of 4 output loads. As can be seen in Table 2, 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.

5. HIGHER LEVEL IMPLICATIONS

So far, our results have focused on combinational logic circuits; however, in order to produce an overall architecture that can carry out useful computing it will be necessary to propagate the random nature of the logic signals and their distributions to the upper levels of architectural structure, *e.g.*, register stacks, priority queues, ALU blocks, *etc.* Ultimately, the final output of computing is taken as the

highest probability outcome with confidence derived from the associated probability distribution. It will be possible to trade off time and space (redundancy) to achieve a desired level of confidence. Some less critical outputs can be less accurate than others and so achieved more quickly and with less redundant computation.

As an example, consider a finite state machine described by the following tuple, $\langle V_{IN}, S, s_0, P \rangle$, where V_{IN} is a set of input symbols, S a set of states, s_0 the initial state, and P the set of stochastic state transition rules. For a conventional state machine, the next state would depend only on the input and the current state with probability of transitioning into the correct state equal to 1. But as signals get noisier, the input symbols and the initial state s_0 have probability distributions and other state transitions can have non-zero probability leading to a stochastic finite state machine.

Conventional approaches to stochastic finite state analysis "unroll" the state transitions into a MRF based on the state-time trellis. This trellis represents all the state transition paths over some finite string of input symbols. The standard approach is to find the sequence with maximum probability using the Viterbi algorithm [19], which is equivalent to solving the trellis MRF by dynamic programming [20]. This is done by taking the current observed result and backtracking through the trellis, computing the most likely state sequences that leads to the result. From the perspective of circuit design this approach is not practical since there is no natural finite interval for the input sequence and the elements of general dynamic programming are too complex at this architectural level. However, recent work on the implementation of the Viterbi algorithm in programmable logic for speech recognition [21] illustrates that approximate solutions can be achieved at the level of complexity consistent with on-chip modules. For control logic, the implementation only needs to be approximate with a fixed, relatively small, time history (trellis) of transitions.

The probability distribution for the input symbols can be derived from the MRF logic implementation of the source logic, in a similar manner as was done for combinational logic in the previous section. Additionally, the probability of each state transition can be adaptively adjusted over time to reflect the variation in both soft and hard fault statistics. This approach provides a natural transition from the level of combinational logic and has sufficient representational power to support the direct manipulation of probabilities.

This example illustrates our overall vision for probabilistic computing where we determine the best probability model for each level of abstraction based on distributions supplied by the previous level as well as prior and learned distributions generated within the level. Each time the most probable computational decisions are reached with increasing accuracy and with a better approximation to the optimum Bayesian decision for computational results.

6. CONCLUSIONS AND FUTURE WORK

This paper is focused on a key question in semiconductor technology: How can nanotechnology extend our CMOS-based digital computing paradigm beyond the ITRS roadmap, where transistor scaling ceases? Our basic device will be the ultimate CMOS transistor (at $L_G \sim 10$ nm); however, by designing the circuits using the Markov Random Field as a framework for probabilistic computing, the design becomes noise-immune, thereby freeing the design process from the necessity for perfect operations and error-free inputs. Since we have started with CMOS, it will be possible for us to simulate the entire design spectrum based on realistic signal and hard fault distributions generated at the logic level. Ultimately, the MRF framework will allow us to extend our approach to all levels of a design.

7. REFERENCES

[1] G. K. Celler and S. Cristoloveanu, "Frontiers of silicon-on-insulator", *J. Appl. Phys.* **93**, 4955 (2003).

[2] S. Luryi, J. M. Xu, and A. Zaslavsky, eds., *Future Trends in Microelectronics: The Nano, the Giga, and the Ultra*, New York: Wiley, 2004.

[3] H. S. P. Wong, "Beyond the conventional transistor", *IBM J. Res. Dev.* **46**, 133 (2002).

[4] The latest publicly released version of the ITRS roadmap is available on the http://public.itrs.net web site.

[5] B. Doris *et al*, "Extreme scaling with ultra-thin Si channel MOSFETs", *Tech. Digest IEDM* (2002), p. 267.

[6] H. Iwai, "The future of CMOS downscaling", 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.

[7] R. Chellappa, *Markov Random Fields: Theory and Applications*, New York: Academic Press, 1993

[8] S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Berlin: Springer-Verlag, 1995.

[9] R. I. Bahar, J. Mundy, and J. Chen, "A Probabilistic-based Design Methodology for Nanoscale Computation", *International Conference on CAD*, Nov. 2003.

[10] J. Besag, "Spatial interaction and the statistical analysis of lattice systems", *J. Royal Statistical Soc. Ser. B* **36**, 192 (1994).

[11] Available at http://www-device.eecs.berkeley.edu/∼ptm/ .

[12] V. M. Polyakov and F. Schwierz, "Excessive noise in nanoscaled double-gate MOSFETs: A Monte Carlo study", *Semicond. Sci. Technol.* **19**, 145 (2004).

[13] 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* **39**, 501 (2004).

[14] R. Sarpeshkar, T. Delbrueck, C. A. Mead, "White Noise in MOS Transistors and Resistors," *IEEE Circuits and Devices Magazine*, **6** 23 (Nov. 1993).

[15] 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 Dev.* **47**, 354 (2000).

[16] 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 Dev.* **50**, 830 (2003).

[17] S. Kullback, *Information Theory and Statistics*, New York: Dover, 1969.

[18] E. M. Sentovich *et al.*, "Sequential circuit design using synthesis and optimization", *International Conference on Computer Design*, Oct. 1992.

[19] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Information Theory* **13**, 260 (1967).

[20] S. Geman and K. Kochanek, "Dynamic programming and the graphical representation of error-correcting codes", *IEEE Trans. Information Theory* **47**, 549 (2001).

[21] S. Melnikoff, S. Quigley, and M. Russell, "Implementing a hidden Markov model speech recognition system in programmable logic", *11th Intern. Workshop Field-Programmable Logic Applications*, Aug. 2001.