Hierarchical Probabilistic Macromodeling for QCA Circuits

Saket Srivastava and Sanjukta Bhanja
Department of Electrical Engineering
University of South Florida, Tampa, USA
(ssrivast, bhanja)@eng.usf.edu

Abstract

With the goal of building an hierarchical design methodology for quantum-dot cellular automata (QCA) circuits, we put forward a novel, theoretically sound, method for abstracting the behavior of circuit components in QCA circuit, such as majority logic, lines, wire-taps, cross-overs, inverters, and corners, using macromodels. Recognizing that the basic operation of QCA is probabilistic in nature, we propose probabilistic macromodels for standard QCA circuit elements based on conditional probability characterization, defined over the output states given the input states. Any circuit model is constructed by chaining together the individual logic element macromodels, forming a Bayesian network, defining a joint probability distribution over the whole circuit. We demonstrate three uses for these macromodel based circuits. First, the probabilistic macromodels allow us to model the logical function of QCA circuits at an abstract level – the "circuit" level – above the current practice of layout level in a time and space efficient manner. We show that the circuit level model is orders of magnitude faster and requires less space than layout level models, making the design and testing of large QCA circuits efficient and relegating the costly full quantum-mechanical simulation of the temporal dynamics to a later stage in the design process. Second, the probabilistic macromodels abstract crucial device level characteristics such as polarization and low-energy error state configurations at the circuit level. We demonstrate how this macromodel based circuit level representation can be used to infer the ground state probabilities, i.e. cell polarizations, a crucial QCA parameter. This allows us to study the thermal behavior of QCA circuits at a higher level of abstraction. Third, we demonstrate the use of these macromodels for error analysis. We show that that low-energy state configurations of the macromodel circuit matches those of the layout level, thus allowing us to isolate weak points in circuits design at the circuit level itself.

Index Terms

quantum-dot cellular automata, bayesian networks, probabilistic computing, qca computing, qca macromodel

I. Introduction

Quantum-dot Cellular Automata (QCA) is an emerging technology that offers a revolutionary approach to computing at nano-level [1]. What sets it apart is that it exploits, rather than treat as nuisance properties, the inevitable nano-level issue of device to device interaction at nano-scales to perform computing. Each cell consists of two electrons that can occupy four dots, resulting in two ground states configurations, which can be taken to represent the logic states of zero or one. Two or more cells interact by Coulombic interaction, with an arrangement of cells settling to the lowest energy state. Since there is no flow of electrons involved, there is no need for traditional interconnects, and it has potential for extremely low-power computing, even below the traditional kT [2]. Both individual QCA cell (semi-conductor and metallic) and multiple QCA arrangement have been fabricated and tested [3], [4]. Significant progress is also being made in using molecules to implement QCAs [5], [6], which will make it possible to operate in room temperature, possibly alleviating the initial criticisms of this technology. It will also connect the areas of molecular computing and QCAs.

Time is ripe to look beyond just device level research in emerging devices and explore circuit level issues so as to scope out the types of circuits that can be built [7], [8], [9], [10], [11], [12]. However, QCA modeling tools available for such designs have been at the layout level. There are several approximate simulators available at the layout level, such as the bistable simulation engine and the nonlinear approximation methods [13], [14], [15]. These methods are iterative and do not produce steady state polarization estimates. In other words, they estimate just state assignments and not the probabilities of being in these states. The coherence vector based method [16], [15] does explicitly estimate the polarizations, but it is appropriate when one needs full temporal dy-

namics simulation (Bloch equation), and hence is extremely slow; for a full adder design with about 150 cells it takes about 500 seconds for 8 input vectors. Perhaps, the only approach that can estimate polarization for QCA cells, without full quantum-mechanical simulation is the thermodynamic model proposed in [17], but it is based on semi-classical Ising approximation. In [18], [19], it was shown that layout-level QCA cell probabilities can be modeled using Bayesian probabilistic networks.

To advance design with QCA, it is necessary to look beyond the layout level. Hierarchical design at multiple levels of abstraction, such as architectural, circuit, layout, and device levels, has been a successful paradigm for the design of complex CMOS circuits. It is only natural to seek to build a similar design structure for emerging technology. Henderson *et al.* [20] proposed an hierarchical CMOS-like top-down approach for QCA blocks that are analyzed with respect to the output logic states; this is somewhat similar to functional logic verification performed in CMOS. We also advocate building an hierarchical design methodology for QCA circuits. However, such an hierarchy should be built based on not just the functionality of the circuit, but it should also allow the abstraction of important nano-device parameters. It is not sufficient just to abstract a QCA circuit in terms of 0-1 boolean logic based majority gates and other logic components, we have to also represent the probabilistic nature of the operations. Thus, for each logic variable X, we have to assign the probabilities associated with the logic values, i.e. $P(X = 1)$ or $P(X = 0)$. In the parlance of QCA, the specific design variable is the "polarization" of cell, which is $P(X = 1) - P(X = 0)$. These probabilities (or polarizations), which are governed by quantum mechanics, are dependent on temperature, which is an important design variable for QCAs that needs to be represented at upper design levels. Another need for probabilistic representations arise due to the nature of the QCA operations. QCA circuits are designed so that the intended logic is mapped to the lowest-energy (ground state) of the cell arrangement. So, it is important that the circuit be kept near

ground state during operations, using mechanisms such as four-phased adiabatic clocking. Logical errors in QCA circuits can arise due to the failure to the settle to the ground state. It is important to compute the difference between the probability of lowest-energy state configuration that results in *correct* output and the lowest-energy state configuration that results in *erroneous* output. It would indeed be useful to be able to compute these erroneous configurations at higher levels of design. Building a device-level characterization sensitive macromodel will facilitate answering the following kinds of questions at higher design levels of abstraction itself. What is expected polarization of the outputs? How does it change with temperature? How sensitive is the design with respect to operational errors?

In this work, we formulate a probabilistic framework for higher level of abstraction of QCA circuits that would enable one to characterize designs with respect to thermal profiles and errors, the two most important design issues in nano-circuit design. Standard QCA circuit elements such as majority logic, lines, wire-taps, cross-overs, inverters, and corners are represented using conditional probability distributions defined over the output states *given* the input states. The probabilistic macromodels allow us to model QCA circuits at an abstract level above the current practice of layout level; we term this higher level as the "circuit" level. The full circuit level model is constructed by chaining together the individual logic element macromodels. This circuit represented using the graphical probabilistic models known as Bayesian networks, where the nodes of the graphs are the individual macromodels and the links represent the connection between them. The nodes are quantified by the macromodel conditional probabilities. The complete network represents a joint probability distribution over the whole circuit. Since conditional distribution over the inputs and outputs are obtained based on quantum mechanical probabilistic characterization, the circuit level model is also faithful to the underlying quantum-mechanical phenomena.

Computations using the macromodel translates to different kinds of probabilistic inference

problems. For instance, computation of ground state polarization is done using the *average* likelihood propagation on the built Bayesian network macromodel. Similarly, the most-likely configuration of the internal nodes corresponding to first-excited, also called near-ground state or the most likely error state at the outputs, can be isolated at the macromodel circuit level itself using *maximum* likelihood propagation on the same Bayesian network macromodel. We demonstrate and validate our model using commonly studied QCA circuits and elements, whose behaviors are pretty well understood by others. First, we show that the ground state polarization probabilities of the output nodes as well as the intermediate nodes in the macromodel of the QCA logic circuit closely match with those obtained from a full layout level implementation [18] at different temperatures. We show examples of characterization of thermal behavior of a QCA logic circuit that can be carried out. Second, we demonstrate that both the ground and the next excited (error) state configuration of the macromodel exactly match the corresponding configurations of the detailed layout cells. The mismatch between the ground and the next excited error state configuration can be used to identify weak spots in circuit design. Using the macromodel, this can now be done at an higher level of abstraction. Isolation of error-prone components would be useful in applying redundancy selectively to the necessary components rather than to the whole circuit. Third, we use the circuit level implementation to vet between alternate design choices. We show examples of this design space exploration process with the example of two adders. We find that one adder design, Adder-1, in spite of its larger area, is better in terms of polarization which is an extremely important measure for the QCA circuits. Also, we see that for Adder-1, number of error-prone components is less than a second adder design, Adder-2, and hence the needed redundancy measures would be less for Adder-1.

The organization of this paper is as follows. In Section II, we begin by explaining the hierarchical modeling scheme used in this work. Then we proceed in Subsection A to summarize the

quantum-mechanical nature of the probabilities associated with the QCA cells. In Section II(B), we show how an arrangement of QCA cells can be modeled by a joint probability function, represented as a Bayesian network. Further down in Section II(C) we present the theory behind the macromodels. We demonstrate how using these macromodels we can (i) model full circuits Section II(D), (ii) explore design space exploration in QCA circuit layouts (Section IV(C)), and (iii) conduct error studies (Section III). We comment on the computational advantage of the circuit level representation over the layout level one in Section IV and we conclude with Section V.

II. Modeling Theory

In this section, we explain the hierarchical modeling scheme. We focus on two levels: the layout level and the circuit level, where groups of QCA cells, corresponding to a basic logic element, are represented as one macroblock. For both these levels, we will use the graphical probabilistic model called Bayesian Networks to represent the underlying joint probability of the entire set of nodes. Note that probabilistic representation is essential to capture the inherently uncertain nature of the computing with QCAs.

Bayesian Networks[21] are efficient representations of the joint probability distribution over a set of random variables using a Directed Acyclic Graph (DAG). Each random variable of interest is represented as a node and links between the nodes denote direct dependencies (cause-effect interactions) between the random variables. For our problem, the random variables are the states of the QCA cells at the layout level or the I/O states of the macromodels. The links are guided by the interaction neighborhood of the cells and the logical flow of information from inputs to the outputs. For QCA circuits these cause-effect directions would be determined by direction of propagation of quantum-mechanical information propagation with change in input. Clocks determine the causal order between cells. Within each clock zone, ordering is determined by the direction of propagation of the wave function [22]. Since the Coulombic interaction between cells

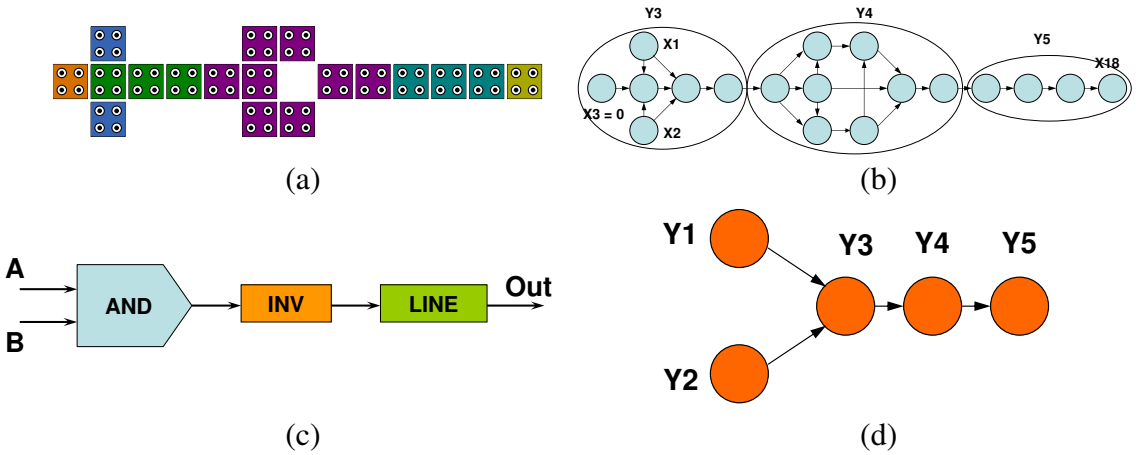


Fig. 1. A NAND logic gate (a) QCA layout (b) Bayesian model of QCA layout (c) Macromodel block diagram (d) Bayesian network of macromodel block diagram.

fall off faster than the fifth power of the distance between them, we need to consider links between cells that are within a small neighborhood of each other, typically 2 cell distance.

In Fig. 1(a), we show the QCA layout of a NAND gate. Fig 1(b) shows the layout level Bayesian representation. Note that we have 18 random variables representing the state of 18 QCA cells. Fig. 1(c) shows the circuit level abstraction of a NAND gate. The Bayesian representation of circuit level abstraction as shown in Fig. 1(d) has fewer cells. Note that each node at the circuit level is the collection of cells from the layout level.

In this work, we will use X to represent the random variable denoting the states of a QCA cell at the layout level (Fig. 1(b)). The input cell states will be denoted by $X_1, \cdots, X_r$, the non-input QCA cells will be $X_{r+1}, \cdots X_N$ and X_s will denote one of the output cell where $r+1 \geq s \geq N$. Similarly for the circuit level, we will use Y to represent the random variable denoting the line states. The $Y_1, \cdots, Y_r$ are set of input cells, $Y_{r+1}, \cdots Y_M$ are the non-input QCA cells and Y_s denotes one of the output cell where $r+1 \geq s \geq M$.

The nodes of the Bayesian network are quantified by the conditional probabilities. At the layout level, we need to specify the conditional probability of the state of a cell given the states of

parent neighbors, i.e. $P(x|pa(X))$ where $Pa(X)$ are the direct causes of the random variable X or the parents of the node X in the directed graph representation[1]. We estimate this using the quantum mechanical modeling of QCA cells. At the circuit level, we need to specify the conditional probability of the output states of a macromodel given the states of the inputs, $P(y|Pa(Y))$. These conditional probabilities are estimated from the conditional probabilities for in the layout level model of the QCA cells comprising the macromodel, at different temperatures.

In general, a Bayesian network encodes the joint probability function as a set of factored conditional probabilities, of minimal representational complexity. Proof of minimality can be found in standard Bayesian network texts such as [21].

$$P(x_1,\cdots,x_n) = \prod_{k=1}^{m} P(x_k|pa(x_k)) \tag{1}$$

In the conditional probability term $P(x|pa(X))$, $pa(X)$ represents the values taken on by the parent set, $Pa(X)$.

Inference or computation with Bayesian networks exploits the sparsely connected graph structure. The most common schemes involve passing messages among the nodes. As we shall see, for we will need to conduct both average case and maximum likelihood inferences. For both the *average* and *maximum* likelihood propagation, we adopt the cluster based exact inference scheme. We refer the reader to [21], [23], [19] for details on the inference scheme. However, it suffices to note that the propagation schemes are based on message passing and are similar, differing only in the kinds of messages that are passed. The original Bayesian network, which is a DAG structure, is first transformed into a junction tree of cliques and then marginal probabilities are computed by local message passing between the neighboring cliques. These methods result in exact inference of probabilities.

In the rest of this section, we provide details of the process. We start with discussion of the

[1] We use lowercase to indicate value of a random variable. i.e. $P(x)$ denotes the probability of the event $X = x$ or $P(X = x)$

macromodel construction process by the Bayesian network model at the layout level, which was proposed in [19]. Then, we present the construction of the macromodels and circuit level Bayesian representation.

A. Quantum Mechanical Probabilities

We sketch how the state probabilities of a QCA cell are dependent on the state probabilities of its layout neighbors, distance to the neighbors, and temperature. Each cell has 2 electrons that can occupy 4 possible dots. Among all the possible occupancy configurations, there are two lowest energy configurations corresponding to the diagonal occupancy of the cells. These represent the two logical states, 0 or 1. So, following Tougaw and Lent [22] and other subsequent works on QCA, we use the two-state approximate model of a single QCA cell. We denote the two possible, orthogonal, eigenstates of a cell by $|1\rangle$ and $|0\rangle$. The state at time t, which is referred to as the wave-function and denoted by $|\Psi(t)\rangle$, is a linear combination of these two states, i.e. $|\Psi(t)\rangle = c_1(t)|1\rangle + c_2(t)|0\rangle$. Note that the coefficients are function of time. The expected value of any observable, $\langle\hat{A}(t)\rangle$, can be expressed in terms of the wave function as $\langle\hat{A}\rangle = \langle\Psi(t)|\hat{A}(t)|\Psi(t)\rangle$ or equivalently as $\mathrm{Tr}[\hat{A}(t)|\Psi\rangle(t)\langle\Psi(t)|]$, where Tr denotes the trace operation, $\mathrm{Tr}[\cdots] = \langle 1|\cdots|1\rangle + \langle 0|\cdots|0\rangle$. The term $|\Psi(t)\rangle\langle\Psi(t)|$ is known as the density operator, $\hat{\rho}(t)$. Expected value of any observable of a quantum system can be computed if $\hat{\rho}(t)$ is known.

A 2 by 2 matrix representation of the density operator, in which entries denoted by $\rho_{ij}(t)$ can be arrived at by considering the projections on the two eigenstates of the cell, i.e. $\rho_{ij}(t) = \langle i|\hat{\rho}(t)|j\rangle$. This can be simplified further.

$$\begin{aligned} \rho_{ij}(t) &= \langle i|\hat{\rho}(t)|j\rangle \\ &= \langle i|\Psi(t)\rangle\langle\Psi(t)|j\rangle = (\langle i|\Psi(t)\rangle)(\langle j|\Psi(t)\rangle)^* \\ &= c_i(t)c_j^*(t) \end{aligned} \tag{2}$$

The density operator is a function of time and using Loiuville equations we can capture the tem-

poral evaluation of $\rho(t)$ in Eq. 3.

$$\hbar\frac{\partial}{\partial t}\rho(\mathbf{t}) = \mathbf{H}\rho(\mathbf{t}) - \rho(\mathbf{t})\mathbf{H} \tag{3}$$

where H is a 2 by 2 matrix representing the Hamiltonian of the cell and using Hartree approximation. Expression of Hamiltonian is shown in Eq. 4 [22].

$$\mathbf{H} = \begin{bmatrix} -\frac{1}{2}\sum_i E_k P_i f_i & -\gamma \\ -\gamma & \frac{1}{2}\sum_i E_k P_i f_i \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}E_k\bar{P} & -\gamma \\ -\gamma & \frac{1}{2}E_k\bar{P} \end{bmatrix} \tag{4}$$

where the sums are over the cells in the local neighborhood. E_k is the "kink energy" or the energy cost of two neighboring cells having opposite polarizations. f_i is the geometric factor capturing electrostatic fall off with distance between cells. P_i is the polarization of the i-th cell. And, γ is the tunneling energy between two cell states, which is controlled by the clocking mechanism. The notation can be further simplified by using $\bar{P}$ to denote the weighted sum of the neighborhood polarizations $\sum_i P_i f_i$. Using this Hamiltonian the steady state polarization is given by

$$P^{ss} = -\lambda_{\mathbf{3}}^{\mathbf{ss}} = \rho_{11}^{ss} - \rho_{00}^{ss} = \frac{E_k\bar{P}}{\sqrt{E_k^2\bar{P}^2 + 4\gamma^2}}\tanh\left(\frac{\sqrt{E_k^2\bar{P}^2/4 + \gamma^2}}{kT}\right) \tag{5}$$

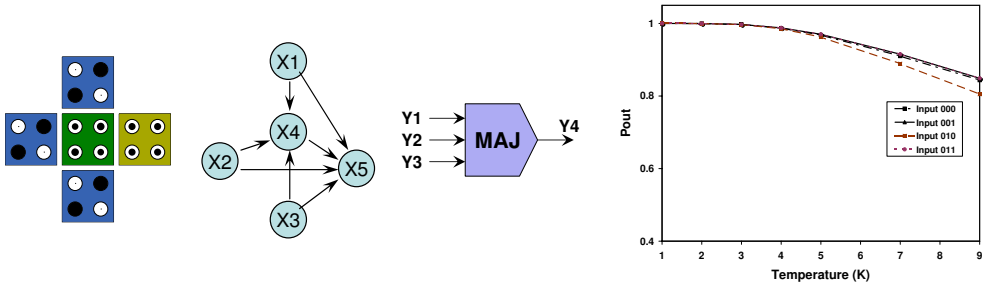
Eq. 5 can be written as

$$P^{ss} = \frac{E}{\Omega}\tanh(\Delta) \tag{6}$$

where $E = 0.5\sum_i E_k P_i f_i$, the total kink energy, $\Omega = \sqrt{E_k^2\bar{P}^2/4 + \gamma^2}$, the Rabi frequency, and $\Delta = \frac{\Omega}{kT}$ is the thermal ratio. We use the above equation to arrive at the probabilities of observing (upon making a measurement) the system in each of the two states. Specifically, $P(X=1) = \rho_{11}^{ss} = 0.5(1+P^{ss})$ and $P(X=0) = \rho_{00}^{ss} = 0.5(1-P^{ss})$, where we made use of the fact that $\rho_{00}^{ss} + \rho_{11}^{ss} = 1$.

B. Layout Level Model of Cell Arrangements

To enable us to form macromodels of various cell arrangements, we need to represent the joint state probabilities of a collection of cells at the layout level. In this section, we summarize how this



(a) (b) (c) (d)

Fig. 2. Majority logic (a) QCA cell layout (b) Bayesian network model (c) Macromodel (d) Probability of the *correct* output value for a 5 cell majority gate at different temperatures and for different inputs.

joint probability can be efficiently represented using Bayesian networks, as shown in [19], [18]. We will use the majority logic arrangement of QCA cells in Fig. 2(a) to illustrate the process.

Each cell is represented by a random variable, taking on two possible values, shown in the Bayesian network in Fig. 2(b). Each node in the network has a conditional probability table (CPT), capturing the probabilities of that node, given the states of the parent (cause) nodes. For example, the center node $X4$, will be associated with the conditional probability $P(x4|x1,x2,x3)$. The product of these CPTs determine the joint probability distribution over all the variables in the network. Thus, the joint probability $P(x1,x2,x3,x4,x5) = P(x4|x1,x2,x3)P(x5|x4,x3,x2,x3)$. The polarization of the output cell $X5$ is a function of the remaining four cells in the layout. The center node $X4$ is actually the one which gets polarized based on the majority of inputs. The output cell depicted here receives the polarization of the central cell $X4$ and also the three inputs, $X1$, $X2$, and $X3$. The interaction between the output cell and the central cell will be much more than the inputs. This is because the kink energy (which determines the amount of interaction between two neighboring cells), decays as the fifth power of distance.

For a given set of possible parent node assignments, the conditional probability values are computed using the Hartree-Fock approximation, applied locally. The parent states are constrained to be as specified in the required conditional probability. We fix the children states (or polarization)

so as to maximize $\Omega = \sqrt{E_k^2 \bar{P}^2/4 + \gamma^2}$, which would minimize the ground state energy over all possible ground states of the cell. Thus, the chosen children states are

$$ch^*(X) = \arg\max_{ch(X)} \Omega = \arg\max_{ch(X)} \sum_{i \in (Pa(X) \cup Ch(X))} E_k \bar{P} \tag{7}$$

The steady state density matrix diagonal entries (Eq. 6 with these children state assignments are used to decide upon the conditional probabilities in the Bayesian network (BN).

$$\begin{aligned} P(X=0|pa(X)) &= \rho_{00}^{ss}(pa(X), ch^*(X)) \\ P(X=1|pa(X)) &= \rho_{11}^{ss}(pa(X), ch^*(X)) \end{aligned} \tag{8}$$

Note that once the conditional probabilities between the nodes and its parents are obtained the Bayesian Network is quantified completely. Some of the important parameters used in this model that effect the polarization of a cell apart from temperature are: *relative permitivity = 12.9, radius of effect = 4, cell dimension = 20nm, cell to cell pitch = 10nm,* $CLOCK\_HIGH = 6.1 * 10^{-2} eV$ *and* $CLOCK\_LOW = 1.9 * 10^{-15} eV$.

C. Macromodel

The basic circuit elements of a QCA circuit consists of typical logic elements, such as Majority, NAND, AND, OR, and NOT, and QCA specific elements such as wires and crossbars. The macromodels of different circuit elements are the conditional probability of output cells given the values of the input cells. We compute this by marginalizing over the internal cells. The underlying premise of the macromodeling is that if the joint probability distribution function $P(x_1, \cdots, x_n)$ over all the n cells in the layout is available, using the process outlined in the previous subsection B, then we can always obtain the exact distribution over subset of cells by marginalizing the probabilities over rest of the variables. For instance, the joint probability over just three cells, x_i, x_j, and x_k, can be obtained by

$$P(x_i, x_j, x_k) = \sum_{\forall x_m, m \neq i,j,k} P(x_1, \cdots, x_n) \tag{9}$$

TABLE I

MACROMODEL DESIGN BLOCKS

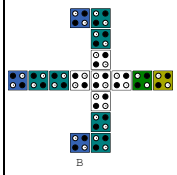
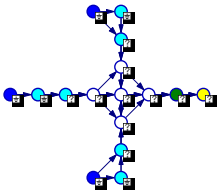
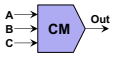
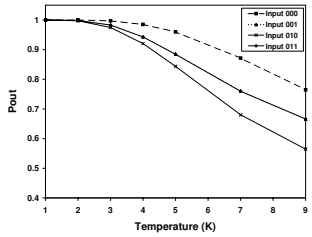
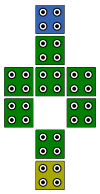
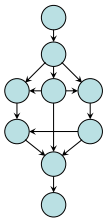
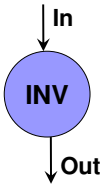
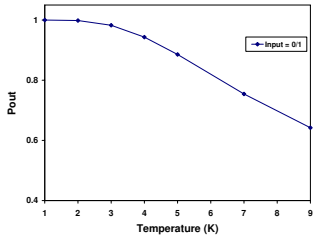
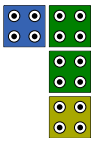
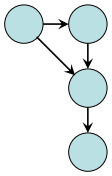
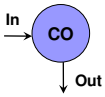
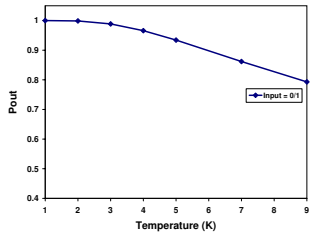
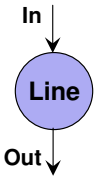
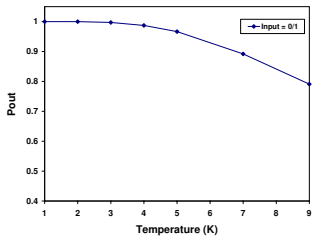
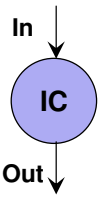
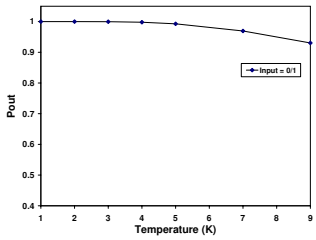
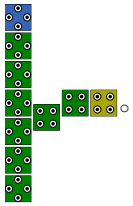
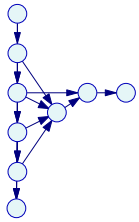
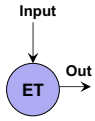
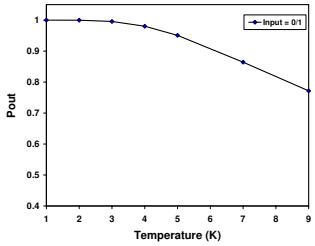
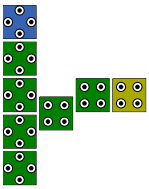
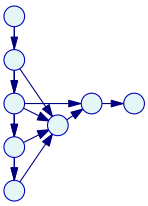
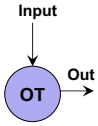
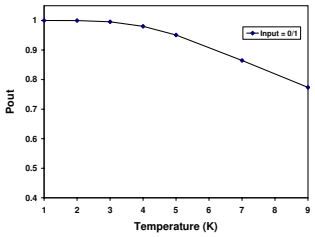
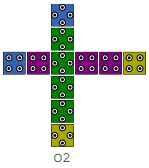
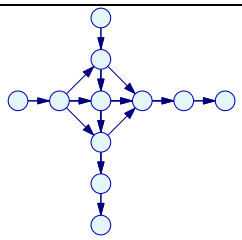
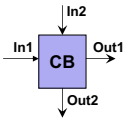
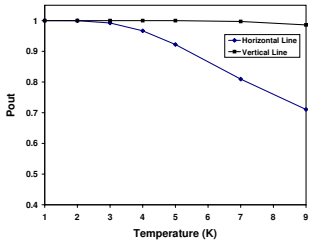
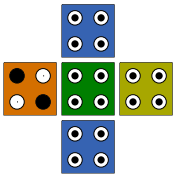
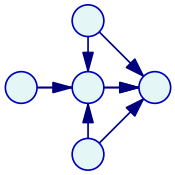
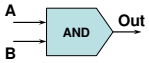
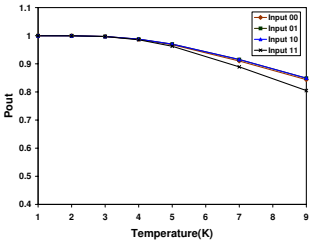
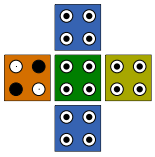
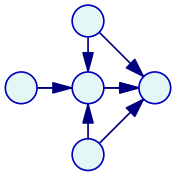
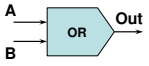
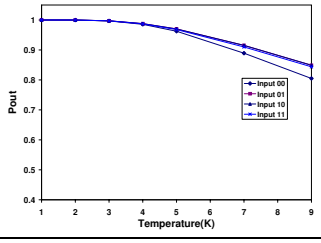
Macromodel	QCA Layout	Bayesian Model	Block Diagram	Thermal Properties
(a) Clocked Majority				
(b) Inverter				
(c) Corner				
(d) Line				
(e) Inverter Chain				

TABLE II

MACROMODEL DESIGN BLOCKS

Macromodel	QCA Layout	Bayesian Model	Block Diagram	Thermal Properties
(a) Even Tap				
(b) Odd Tap				
(c) Cross-bar				
(c) And Gate				
(c) OR Gate				

Hence, at the circuit level, we do not represent all the m internal cells. Note that at circuit level, we only represent $P(x_i, x_j, x_k)$ and represent them with different variable Y, which essentially captures the input-output dependence but is faithful to the layout level quantum interaction since the macromodel is built by marginalizing the layout level cells. This marginalizing is achieved by conducting *average* likelihood inference [21], [23] on the Bayesian network representation over all the cells in the macromodel unit. Note that Eq. 9 will yield different results at different temperatures and we store the conditional probabilities at various temperature points.

Fig. 2(d) shows the thermal models for the majority gate in Fig. 2(a). The macromodel probability distribution is defined over the output and the 3 input nodes. At a temperature of 1K, if inputs are 0, 0 and 0 then the probability of output node is at state 0 is "0.999963". As the temperature is increased, this probability decreases. We also notice that the thermal behavior is dependent on the input values. Note that, for correct operation, the probability of *correct* output should be greater than 0.5.

In the rest of this section, we present results for other basic building blocks: clocked majority gate (Table. I(a)),inverter (Table. I(b)), line (Table. I(c)), corner (Table. I(d)), inverter chain (Table. I(e)), even tap (Table. II(a)), odd tap (Table. II(b)), crossbar (Table. II(c)), AND gate (Table. II(d)) and OR gate (Table. II(e)). For each macro-cell, we show the QCA layout, layout level Bayesian model, circuit level input-output relation and magnitude of polarization drop with temperature. All the conditional probabilities are stored at various point of temperatures.

We make three important observations. First, a clocked majority gate, which is necessary to synchronize all the input signals reaching the majority gate, has weaker polarization at higher temperature compared to the simple majority shown in Fig. 2(d) as number of cells are higher in the clocked majority gate. Hence if inputs to a majority gate are arrive at the same time, then simple majority yields better polarizations at higher temperatures. Second, inverters have larger drop of

TABLE III
ABBREVIATIONS USED FOR MACROMODEL BLOCKS FOR DESIGNING QCA ARCHITECTURES OF FULL ADDERS AND MULTIPLIER

Symbol	Macromodel
Maj	Simple Majority Gate
CM	Clocked Majority Gate
Inv	Inverter
Line	Line Segment
CO	Corner
IC	Inverter Chain
OT	Odd Tap
ET	Even Tap
CB	Crossover
AND	And Gate
OR	Or Gate
ZL	z-line

polarization over the odd-tap structure at higher temperatures. Third, the crossbar structure, which allows two signal to cross each other in a coplanar way, has a different drop for the two signals.

D. Circuit Level Modeling

Table III lists all the symbols used for macromodel design blocks that we have used in our designs. A macromodel library stores the input-output characteristics (output node probabilities for each input vector set) of each macromodel block based on temperature. That means for each temperature, we have a library of macromodel blocks listed in the Table III. Once we know the logic components required to build a circuit, we simply extract the macromodel logic blocks and the required connectivity blocks (e.g. Line, Corner, Inverter Chain, etc.) from the library at a given temperature and use them to build the logic circuit. We form a Bayesian macromodel using the input-output probabilities of each block. The output from one macromodel block is fed to the input(s) of next macromodel block.

We illustrate the process using the full adder circuit, Adder-1, shown in Fig. 3(a). It consists of five majority gates with no inverters. Fig. 3(b) shows the corresponding layout level Bayesian network. We model the circuit level QCA macromodel shown in Fig. 3(c) which is the circuit level abstraction of Fig. 3(a). The Bayesian macromodel is shown in Fig. 3(d). Each signal (node)

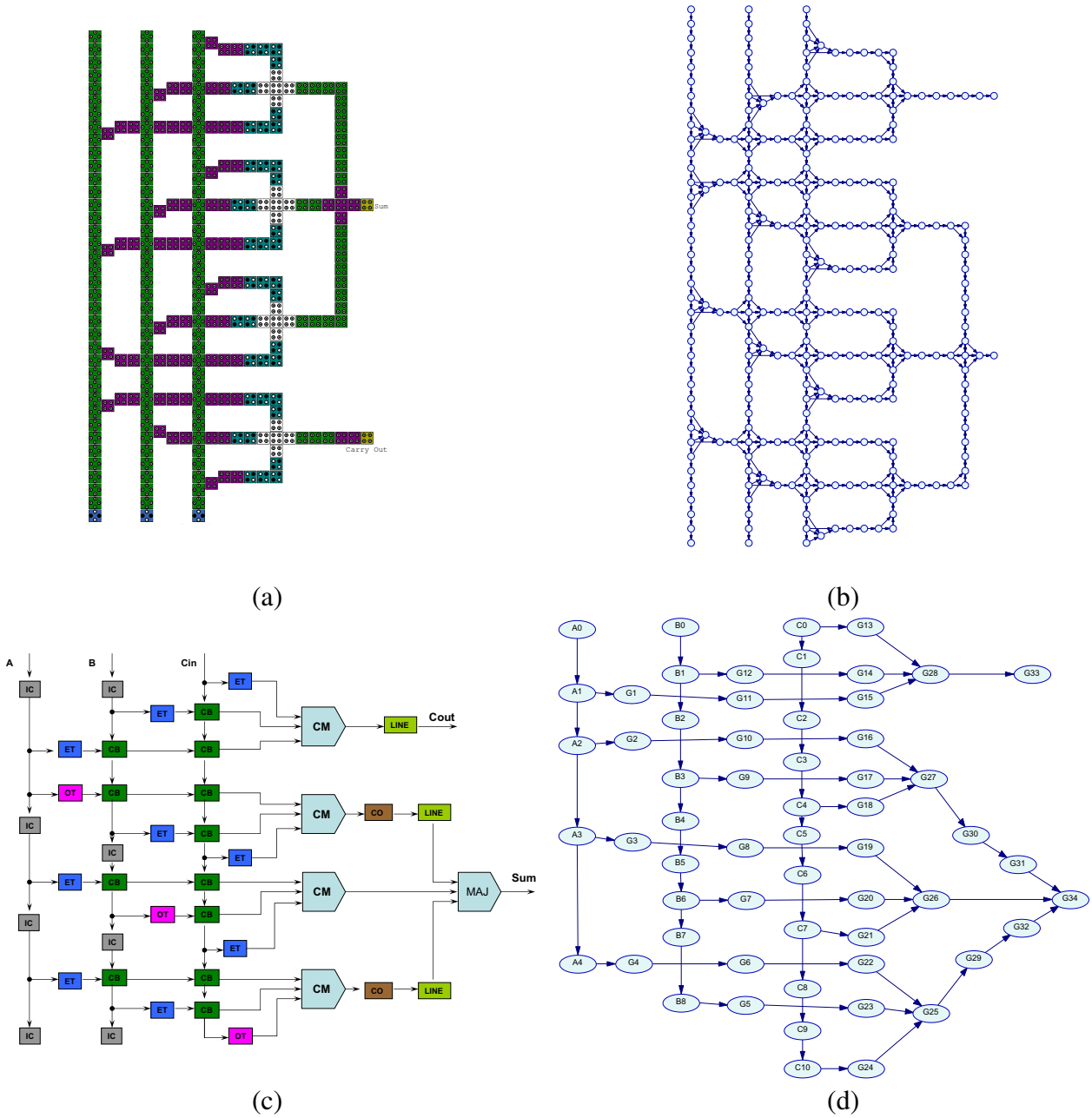


Fig. 3. A full adder circuit (Adder-1) (a) QCA cell layout (b) Layout level Bayesian network representation. (c) Circuit level representation. (d) Circuit level Bayesian network macromodel.

can either be a primary input, or an output cell of a macroblock like line, inverter etc. The links are directed from the input to the output of each macroblock and are quantified by the device macromodels. Thus, we arrive at directed acyclic graph easily from the circuit model in Fig. 3(c).

III. Error Computation

Apart of the computation of the polarization of each QCA cell or macromodel line, which we can arrive at by using *average* case propagation, another analysis of interest when comparing designs is the comparison of the least energy state configuration that results in correct output versus those that result in erroneous outputs. What is the probability of the minimum energy configuration that results in *error* at the output, x_s, for a given input assignment, $x_1, \cdots, x_r$? This can be arrived at by conditional maximum likelihood propagation. In essence, we compute $\arg\max_{x_1,x_2,\cdots,x_r} P(x_{r+1}, \cdots, x_N | x_1, \cdots, x_r, x_s)$ and the minimum energy configuration of all the cells that generates the erroneous output x_s is $\{x_1^e, x_2^e, \cdots x_{r+1}^e, \cdots x_N^e\}$. This configuration corresponds to the most likely error state at the output x_s. Whenever we have $x_i^g \neq x_i^e$, the i^{th} cell is considered sensitive to error at output x_s (also termed as weak spots).

The above computational problem of maximization of a product of probability functions can be factored as product of the maximization over each probability functions, these maximizations can also be computed by local message passing [21]. The exact maximum likelihood inference scheme is based on local message passing on a tree structure, whose nodes are subsets (cliques) of random variables in the original DAG [23]. This tree of cliques is obtained from the initial DAG structure via a series of transformations that preserve the represented dependencies. The details of the inference scheme can be found in [19]. At this transformed point, we have a tree of cliques where each clique is a sub-set of random variables. Two adjacent cliques that share a few common variable play a key role in inference. The joint probability of all the variables can be proven to be the product of individual clique probabilities. Since the problem of maximization

of a product of probability functions can be factored as product of the maximization over each probability functions, this maximization can also be computed by local message passing [23]. The overall message passing scheme involves the neighboring cliques using the maximum operator where the clique probabilities are updated till the marginal probability of the shared variables are the same.

This kind of maximum likelihood analysis can be conducted both at the layout and the circuit levels. Let us say that the circuit level macroblocks have $Y_1, \cdots, Y_r$ as inputs and $Y_{r+1}, \cdots, Y_M$ as internal circuit level lines (nodes). Let us say that the ground state macroblock cell polarizations are denoted by $\{y_1^g, y_2^g, \cdots y_{r+1}^g, \cdots y_M^g\}$. With respect to the the erroneous output y_s, let the minimum energy configuration is $\{y_1^e, y_2^e, \cdots y_{r+1}^e, \cdots y_M^e\}$. As in the case of layout, whenever we have $y_j^g \neq y_j^e$, the j-th cell is considered sensitive to error at output y_s.

In the next section, we will presents results that show that the error modes of the circuit and layout levels match. That is, whenever Y_j is sensitive to the first-excited error state for output Y_s, the corresponding layout level model, shows the set of $\{X_i\}$ that constituted the macroblock Y_j is also sensitive . This is an extremely important finding that indicates that weak spot in the design can be identified at the circuit level itself without obtaining the cell layout. Also this is an important design metrics and can be used to vet one design over and above the thermal profile of the output polarization.

IV. Results

We present results using the full adder design, which has been widely studied by others. We also use a multiplier design, which is a somewhat larger design. First, we will show that the ground state polarization probabilities of the output nodes as well as the intermediate nodes in the macromodel of the QCA logic circuit closely match with those obtained from a full layout level implementation [18] at various temperatures. Second, we demonstrate that both the ground and

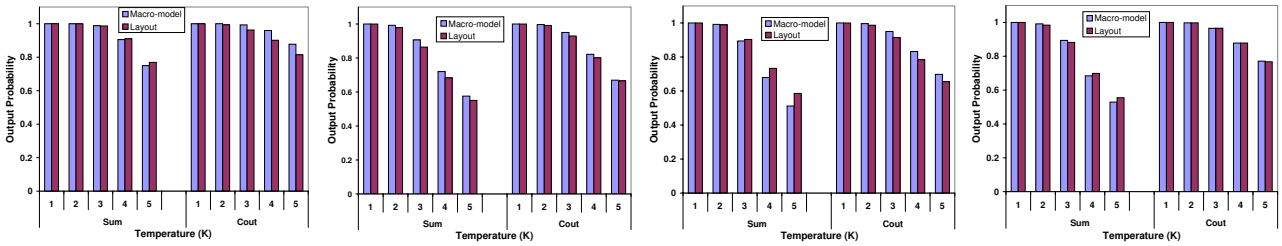


Fig. 4. Probability of correct output for sum and carry of Adder-1 based on the layout-level Bayesian net model and the circuit level macromodel, at different temperatures, for different inputs (a) (0,0,0) (b) (0,0,1) (c) (0,1,0) (d) (0,1,1).

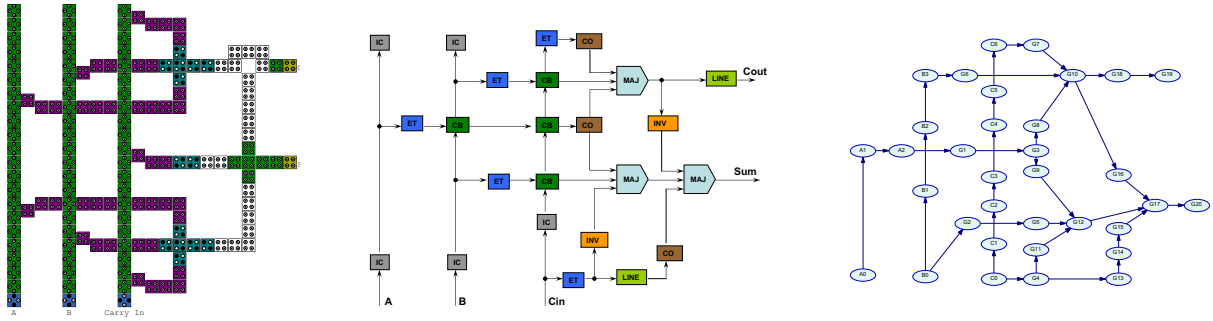


Fig. 5. A QCA Full Adder circuit (Adder-2) (a) QCA Fulladder cell layout (b) Macromodel representation (c) Macromodel Bayesian network

the next excited (error) state configuration of the macromodel exactly match the corresponding configurations of the detailed layout cells for two full adders designs. Third, we use the circuit level implementation to vet between alternate design choices. We show examples of this design space exploration process with the example of two adders.

A. *Polarization*

Fig. 4 plots the polarization estimates at the layout and the circuit levels for various temperature, and for different inputs for Adder-1 architecture shown inFig. 3a (layout level) and Fig. 3c (circuit level). Fig. 5(a) shows second adder architecture (Adder-2), consisting of three majority gates and two inverters [24]. Fig. 6 plots the polarization estimates at the layout and the circuit levels for various temperature, and for different inputs. We see that the difference in probability of

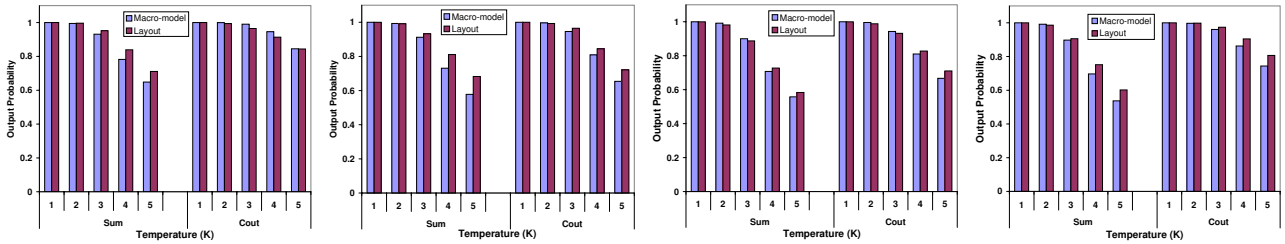


Fig. 6. Probability of correct output for sum and carry of Adder-2 based on the layout-level Bayesian net model and the circuit level macromodel, at different temperatures, for different inputs (a) (0,0,0) (b) (0,0,1) (c) (0,1,0) (d) (0,1,1).

correct output node between circuit and layout level model design is low for both the adders. We also see that in both layout and circuit level designs, the probability of the output node is dependent on the input vector set.

Similar trends is also seen for the 2x2 multiplier circuit shown in Fig. 7(a). The multiplier circuit is somewhat larger than the full adder circuit and consists of two AND gates and two half adders. We made use of a half adder similar to Adder-2 full adder design, for the simple reason that it occupies less area. The polarization of the output nodes in the multiplier layout is almost similar to that obtained at the outputs of multiplier circuit designed using the macromodel blocks. In Fig. 9 and 10, we show the variation of output nodes C0,C1,C2 and C3 of the multiplier with respect to temperature for both layout and macromodel design.

B. Error Modes

We compute the near-ground state configurations that results in error in the output carry bit C_{out} of the QCA full adders (Adder-1 and Adder-2) using both the layout and circuit level models. These are shown in Fig. 11 and 12 and Fig. 13 and 14. We show four cases, for input vectors (0,0,0), (1,0,0), (0,1,0) and (1,1,1). The other four input vector sets will have similar results due to symmetry in design. We use red marker to point to the components that are weak (high error probabilities) in both the layout and circuit level. We can easily see that the nodes with high

error probabilities in QCA layout are the ones that are clustered to form an erroneous node in the macromodel circuit design. In other words, if a node (a macromodel block) in macromodel circuit layout is highly error prone for a given input set, then some or all the QCA cells forming that macromodel block are highly prone to error. This indicates that weak spot in the design can be identified early in the design process, at the circuit level itself.

C. Design Space Exploration

We show that even at the macromodel circuit level, we have the ability to explore the design space with respect to different criteria. In addition, to obvious criteria such as gate count, we can use polarization as a design metric. The probabilistic macromodel allows us very fast estimates of polarization that correlate very well with layout level estimates. As an example we use the two adders in Fig. 3(a) and Fig. 5(a). The two adders shown here have been designed using different macromodel blocks, occupying different design areas.

The outputs of Adder-1 circuit is given by

$$\begin{aligned} Sum &= A \cdot B \cdot C_{in} + \bar{A} \cdot \bar{B} \cdot C_{in} + \bar{A} \cdot B \cdot \bar{C_{in}} + A \cdot \bar{B} \cdot \bar{C_{in}} \\ &= m(m(\bar{A}, B, C_{in}), m(A, B, \bar{C_{in}}), m(A, \bar{B}, C_{in})) \\ C_{out} &= m(A, B, C_{in}) \end{aligned} \tag{10}$$

where $m(A,B,C_{in})$ is the majority gate containing A,B and C_{in} as inputs. Similarly, for Adder-2 circuit the outputs are given by [24]

$$\begin{aligned} Sum &= m(\bar{C_{out}}, C_{in}, m(A, B, \bar{C_{in}})) \\ C_{out} &= m(A, B, C_{in}) \end{aligned} \tag{11}$$

We see that Adder-1 circuit uses five majority gates and three inverters for implementation while Adder-2 circuit uses three majority gates and two inverters. Hence the design circuit design of Adder-2 is certainly superior to Adder-1 in terms of area. However, as it can be seen from the thermal study, inverter has one of the worst polarization drop with respect to temperature and

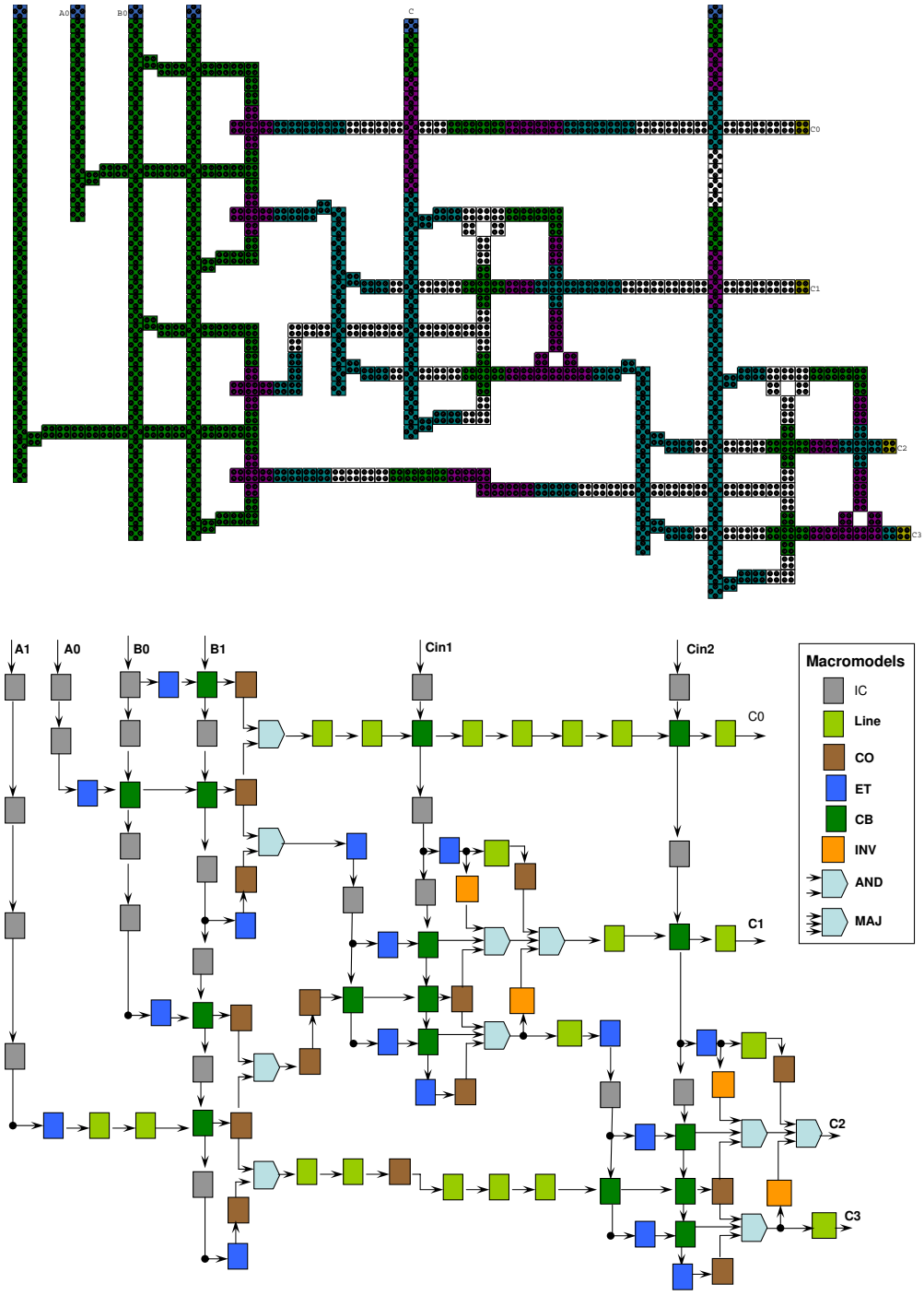


Fig. 7. A QCA 2x2 Multiplier circuit(a) QCA multiplier cell layout (b) Macromodel representation

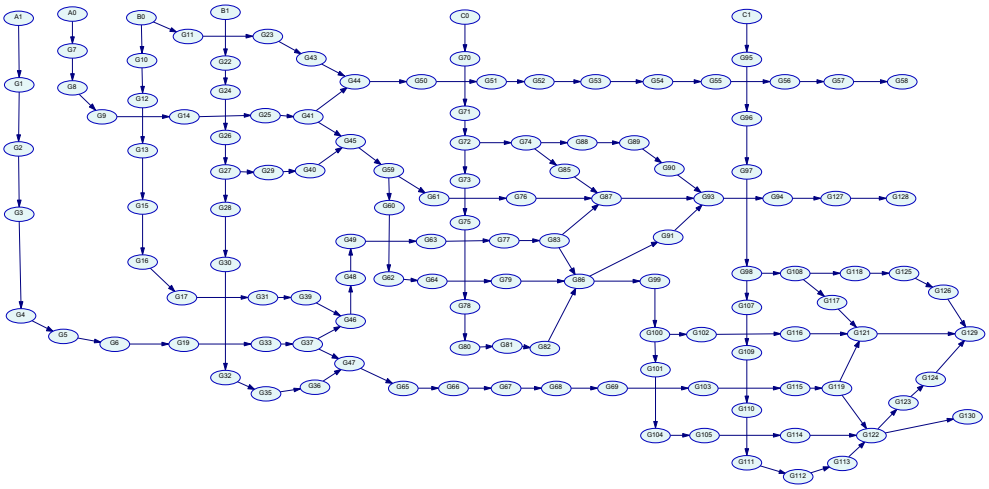


Fig. 8. Macromodel Bayesian network of a QCA 2x2 Multiplier circuit

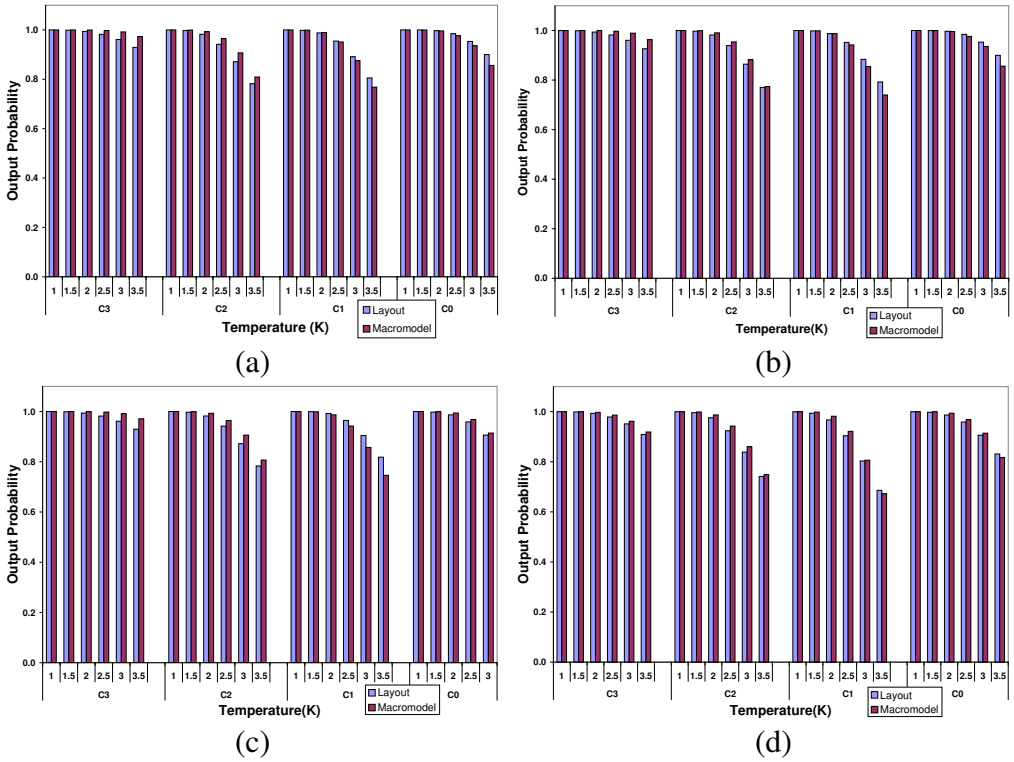


Fig. 9. Probability of correct output at the four output nodes of 2x2 Multiplier circuit based on the layout-level Bayesian net model and the circuit level macromodel, at different temperatures, for different inputs (a) (0,0),(0,1) (b) (0,0),(1,1) (c) (0,1),(0,1) (d) (0,1),(1,1)

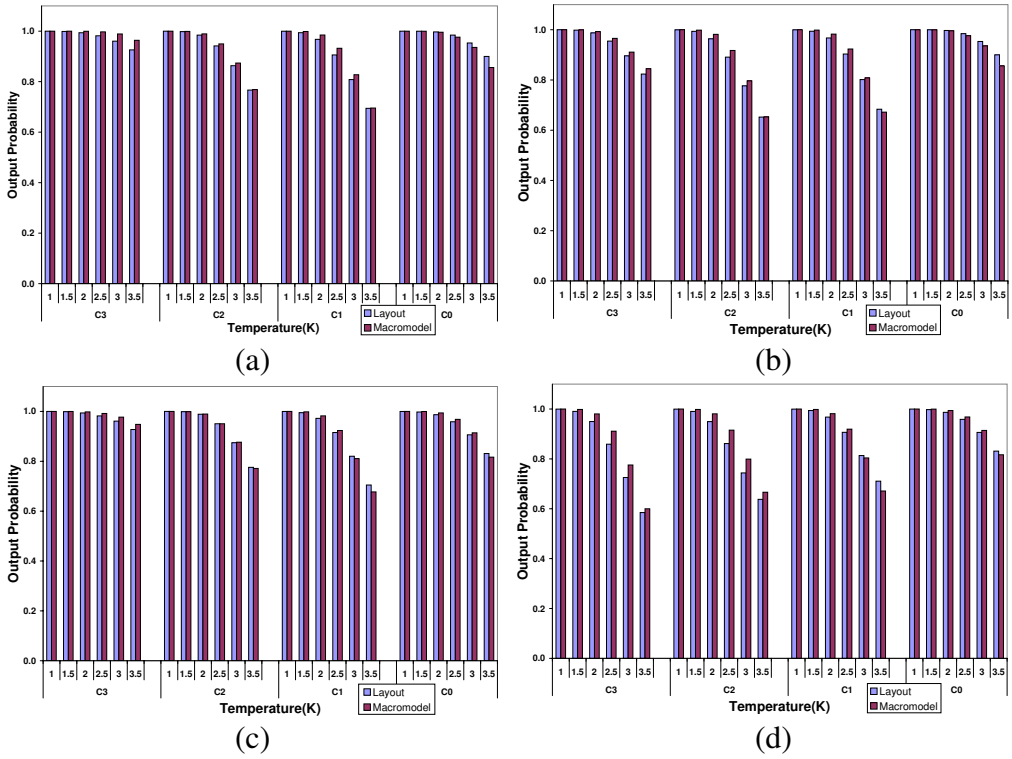


Fig. 10. Probability of correct output at the four output nodes of 2x2 Multiplier circuit based on the layout-level Bayesian net model and the circuit level macromodel, at different temperatures, for different inputs (a)(1,0),(0,1) (b) (1,0),(1,1) (c) (1,1),(0,1) (d) (1,1),(1,1).

inverters in series path will reduce the overall polarization by a great extent. Hence for larger circuits, a design criteria might look at Adder-1 in a different light.

Note that in the context of error modes, presented earlier, we saw that Adder-1 again shows *less* number of error-prone nodes than Adder-2 (Fig. 11 shows error-prone nodes for first-excited state at carry output) for most likely errors in the outputs. Note that, ideally this conclusion requires the detailed layout, however, maximum-likelihood propagation of the circuit level Bayesian Network yields the same error modes as the detailed layout. This measure indicates that cost of addition error correction required for Adder-2 would be more than that of Adder-1.

Last but not the least, we observe that an odd tap shown in Section II is a good target for one inverter as the polarization loss is less than an inverter and an even tap works better than an

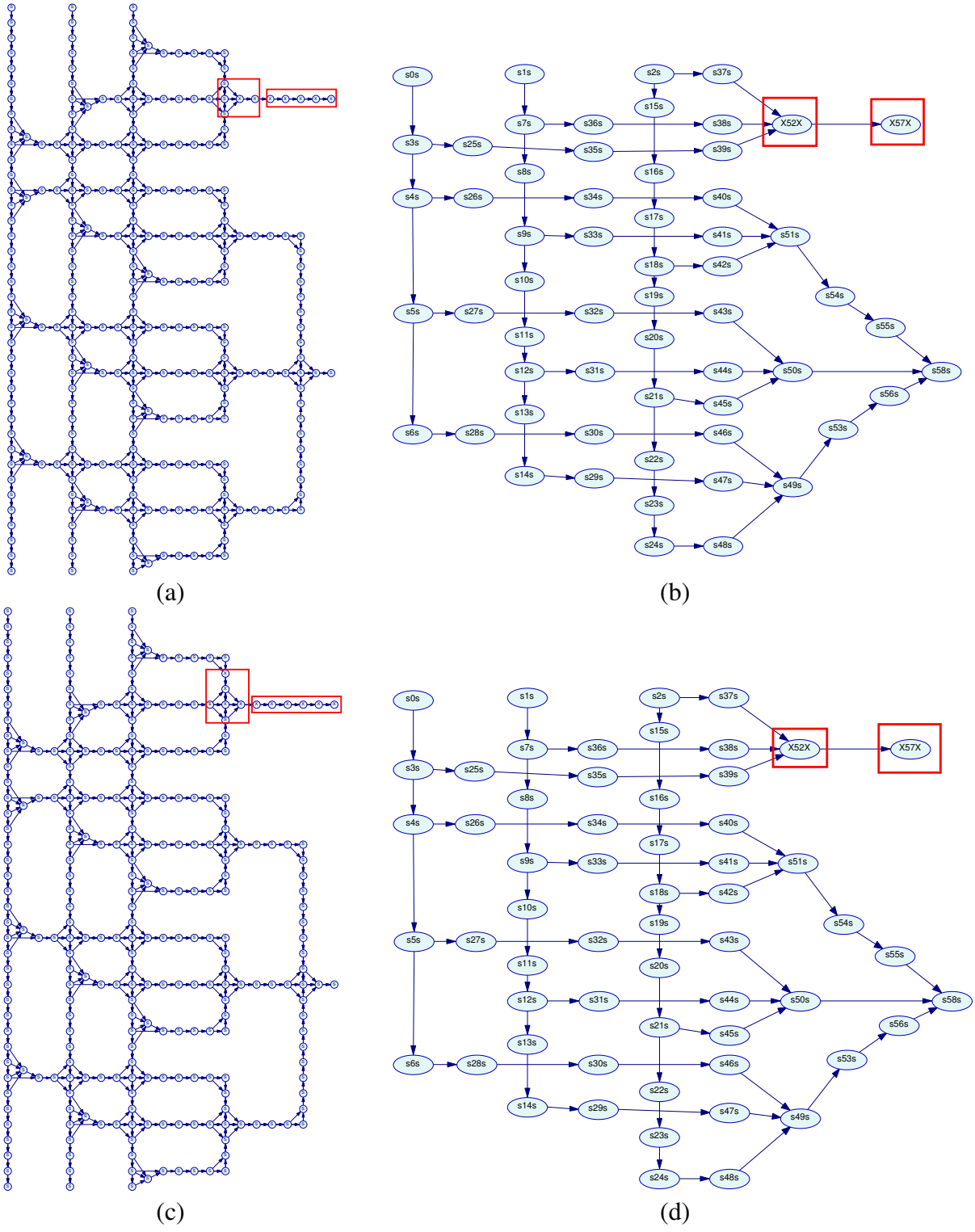


Fig. 11. Error-prone nodes for first-excited state at carry output QCA Adder-1 Circuit and its Macromodel design. It can be seen that the erroneous nodes in the layout are effectively mapped in the macromodel design. Input vector set for (a) and (b) is (0,0,0) and that for (c) and (d) is (1,0,0).

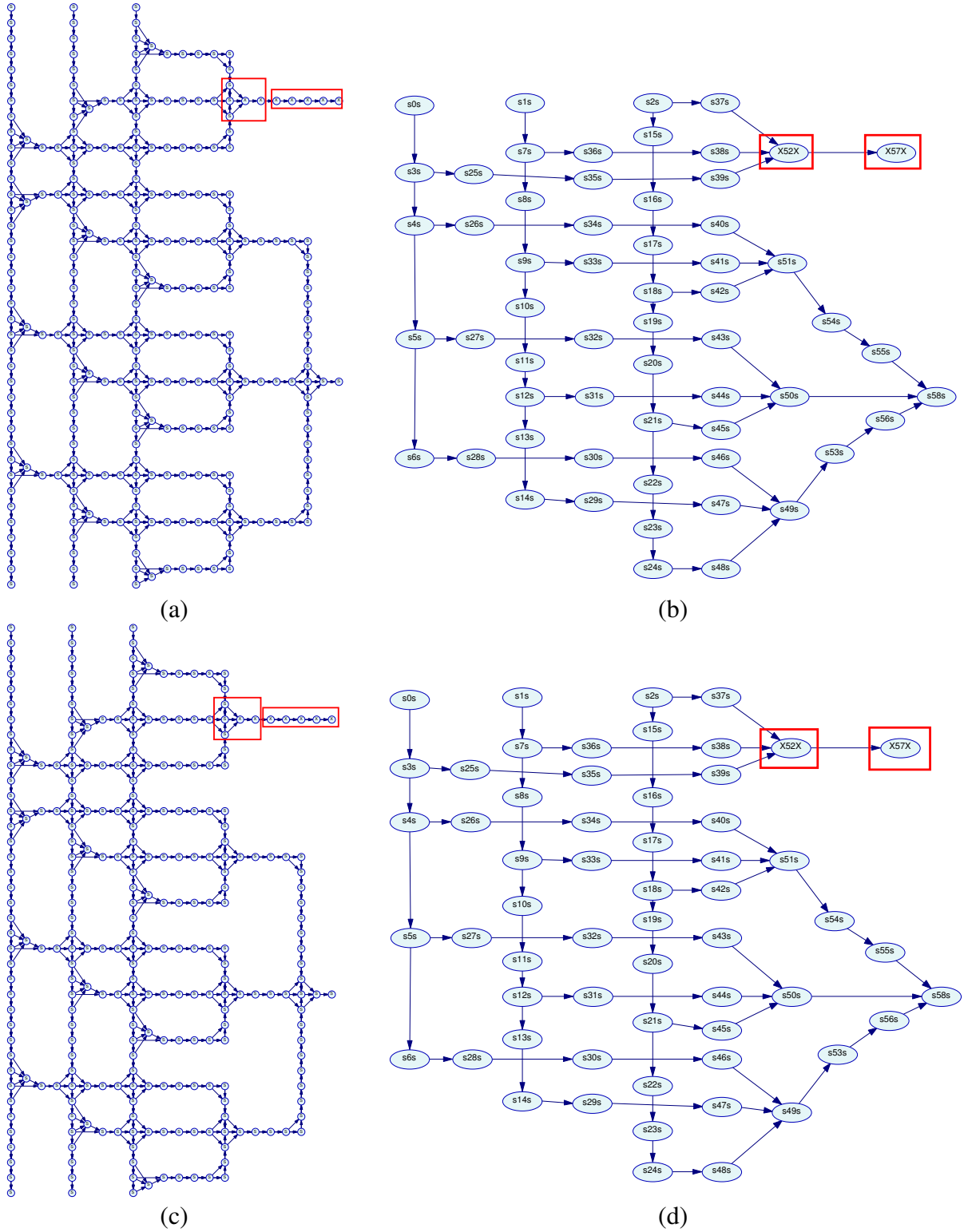


Fig. 12. Error-prone nodes for first-excited state at carry output QCA Adder-1 Circuit and its Macromodel design. It can be seen that the erroneous nodes in the layout are effectively mapped in the macromodel design. Input vector set for (a) and (b) is (0,1,0) and that for (c) and (d) is (1,1,0).

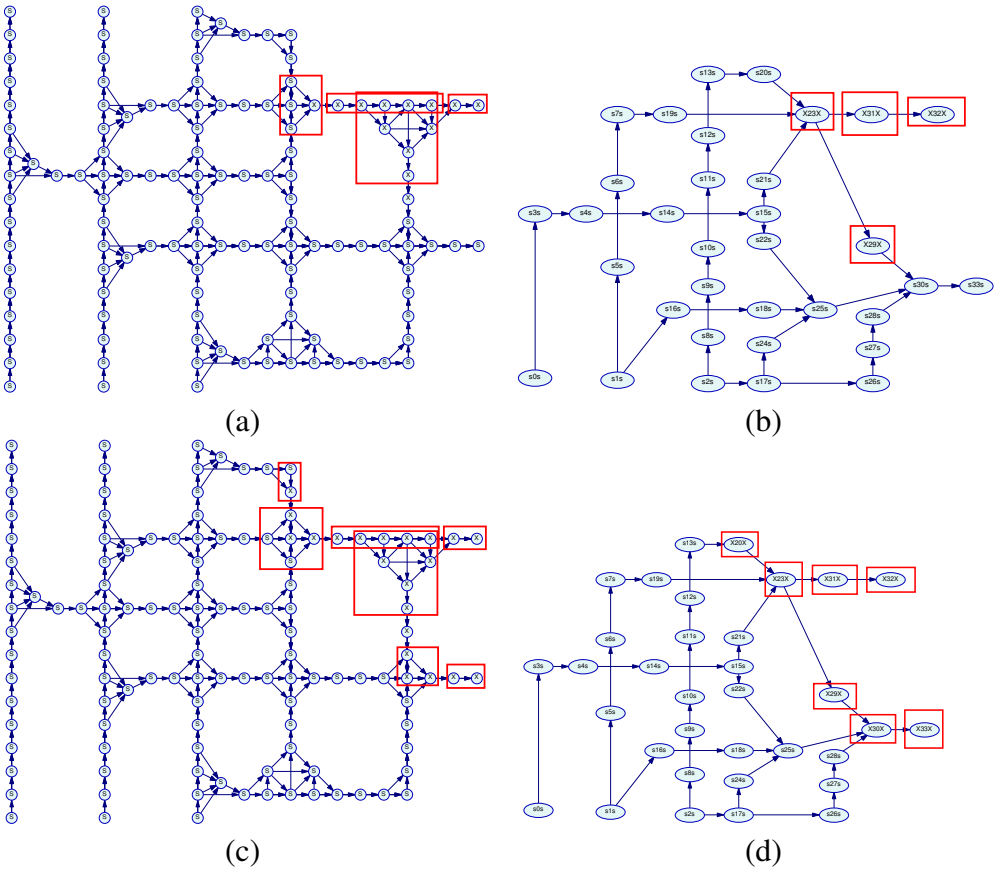


Fig. 13. Error-prone nodes for first-excited state at carry output QCA Adder-2 Circuit and its Macromodel design. It can be seen that the erroneous nodes in the layout are effectively mapped in the macromodel design. Input vector set for (a) and (b) is (0,0,0) and that for (c) and (d) is (1,0,0).

even number of inverter chains. The multiplier design that we show, utilizes these facts to arrive at better design with respect to output polarization and this, in turn, improves the multiplier's thermal characteristics.

D. Computational Advantage

To quantify the computational advantage of a circuit level macromodel with a layout level model, we consider the complexity of the inference based on the Bayesian net models for each of them. As we mentioned earlier, in the cluster-based inference scheme, the Bayesian Network is converted into a junction tree of cliques and the probabilistic inference is performed on the junction

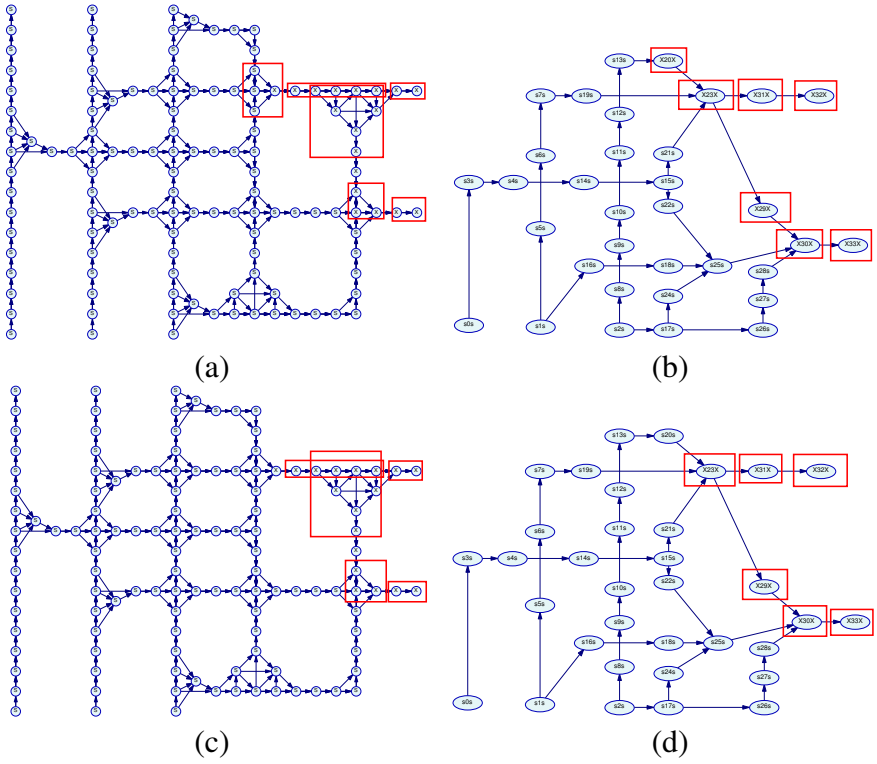


Fig. 14. Error-prone nodes for first-excited state at carry output QCA Adder-2 Circuit and its Macromodel design. It can be seen that the erroneous nodes in the layout are effectively mapped in the macromodel design. Input vector set for (a) and (b) is (0,1,0) and that for (c) and (d) is (1,1,0)

tree by local computation between the neighboring cliques of the junction tree by local message passing [21], [25]. Space complexity of Bayesian inference is $O(n.2^{|C_{max}|})$ where n is the number of variables, $|C_{max}|$ is the number of variables in the largest clique. Time complexity is $O(p.2^{|C_{max}|})$, where p is the number of cliques in the junction tree. We tabulate the complexity terms for the two adder designs in Table IV, along with the corresponding values for n, p and $|C_{max}|$. We can see that macromodel is order of magnitude faster especially due to the reduction in $|C_{max}|$ which would be important in synthesizing larger networks of QCA cells. Another observation is that Adder 2 is less expensive in terms of computation even though polarization drops are more due to the presence of inverters.

As we can see from the Table V, the simulation time required to evaluate a circuit is orders

TABLE IV

LAYOUT AND MACROMODEL TIME (T_c) AND SPACE (T_s)COMPLEXITIES. PLEASE SEE TEXT FOR AN EXPLANATION $C_{max}|$, n, AND p.

	Adder 1		Adder 2		Multiplier	
Parameters	Layout model	Macromodel	Layout model	Macromodel	Layout model	Macromodel
Cmax	15	8	10	5	15	5
p	215	57	96	30	436	119
n	278	64	125	34	539	130
$T_c = p.2^{\lvert C_{max}\rvert}$	7045120	14592	98304	960	14286848	3808
$T_s = n.2^{\lvert C_{max}\rvert}$	9109504	16384	128000	1088	17661952	4160

TABLE V

COMPARISON BETWEEN SIMULATION TIMING (IN SECONDS) OF A FULL ADDER AND MULTIPLIER CIRCUITS IN QCADESIGNER(QD) AND GENIE BAYESIAN NETWORK(BN) TOOL FOR FULL LAYOUT AND MACROMODEL LAYOUT

Simulation Time	Adder-1	Adder-2	2x2 Multiplier
	278 cells	125 cells	539 cells
QD Coherence Vector	566	253	966
QD Bistable Approx.	5	3	15
QD Nonlinear Approx.	3.5	2	8
BN Full Layout model	0.240	0.030	0.801
BN Macromodel Layout	0.010	0.000	0.08

of magnitude lower than that in QCADesigner tool. Moreover, we see that the simulation timing for bayesian macromodels of the adder circuit are much lower than bayesian full layout model. The graphs depicted in Fig. 4, Fig. 6, Fig. 9 and Fig. 10 present the crux of this work. The drooping characteristic of output node polarization with rise in temperature is a universally known fact. What we have shown in this work (as depicted in these graphs) is that the polarization of the output node in our macromodel design is showing the same drooping characteristics and is almost the same as that of the full layout. We can see that macromodel is order of magnitude faster specially due to the reduction in $|C_{max}|$ which would be important in synthesizing larger networks of QCA cells. Another observation is that Adder 2 is less expensive in terms of computation even though polarization drops are more due to the presence of inverters.

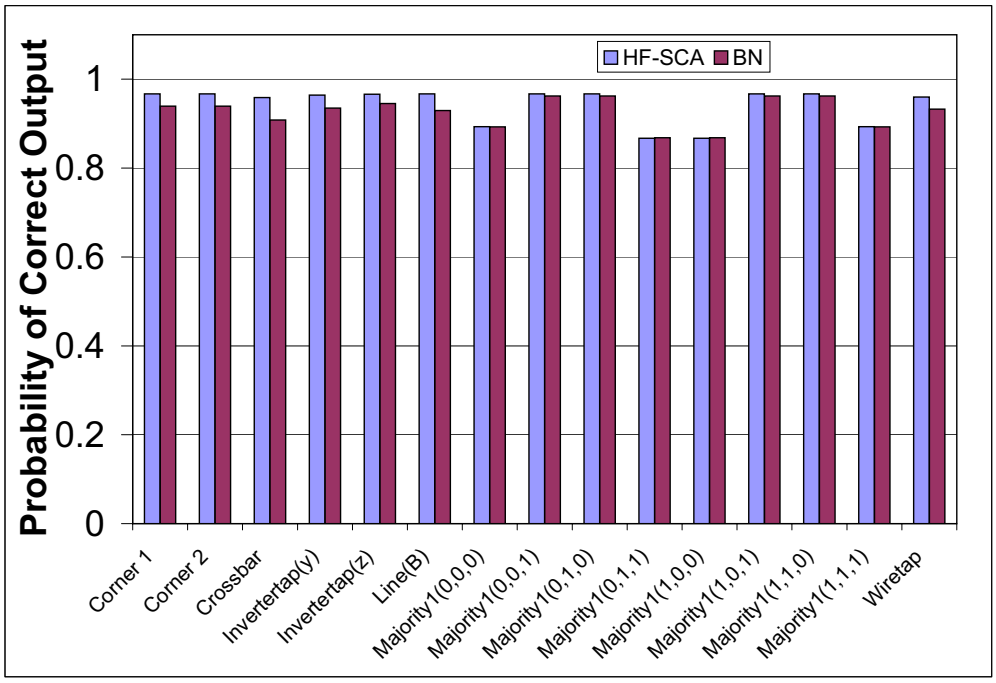


Fig. 15. Validation of the Bayesian network modeling of QCA circuits with Hartree-Fock approximation based coherence vector-based quantum mechanical simulation of same circuit. Probabilities of correct output are compared for basic circuit elements.

V. Conclusion

We proposed an efficient Bayesian Network based probabilistic macromodeling strategy for QCA circuit that can estimate cell polarizations, ground state probability, and lowest-energy error state probability, without the need for computationally expensive quantum-mechanical computations. We showed that the polarization estimates at layout and circuit levels are in good agreement. In our previous work [19], we had validated the layout level Bayesian network model with quantum-mechanical simulation Hartree-Fock, Self Consistent Analysis (HF-SCA) based estimates (see Fig. 15). In this work, we illustrate our macromodeling idea using two full adder macro model design implementations and a somewhat larger QCA design of a 2x2 Multiplier implementation. We found that both the polarization and the error mode estimates at the circuit level match those at the layout level.

The Bayesian macromodel should be useful for vetting QCA circuit designs at higher levels of abstraction in terms of not only the ground state, but also polarization, thermal dependence, and error modes. The contributions of this work can be broadly classified to be in the area of "Design for low Error" that considers error-tolerant circuit synthesis, taking into account circuit overhead considerations. The developed models in this work can be used to selectively identify weak components in a design early in the design process. It would then be possible to reinforce those weak spots in the design using reliability enhancing strategies. The error modes can also be used to compare multiple designs early on in the process.

One possible future direction of this work involves the extension of the BN model to handle sequential logic. This is possible using an extension called the dynamic Bayesian networks, which have been used to model switching in CMOS sequential logic [26].

REFERENCES

[1] C. Lent and P. Tougaw, "A device architecture for computing with quantum dots," in *Proceeding of the IEEE*, vol. 85-4, pp. 541–557, April 1997.

[2] J. Timler and C. Lent, "Maxwell's demon and quantum-dot cellular automata," *Journal of Applied Physics*, vol. 94, pp. 1050–1060, July 2003.

[3] A. Orlov, R. Kummamuru, R. Ramasubramaniam, C. Lent, G. Bernstein, and G. Snider, "Clocked quantum-dot cellular automata shift register," *Surface Science*, vol. 532, pp. 1193–1198, 2003.

[4] R. Kummamuru, A. Orlov, R. Ramasubramaniam, C. Lent, G. Bernstein, and G. Snider, "Operation of a quantum-dot cellular automata (qca) shift register and analysis of errors," *IEEE Transactions on Apllied Physics*, vol. 50, pp. 1906–1913, September 2003.

[5] C. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," *Journal of American Chemical Society*, vol. 125, pp. 1056–1063, 2003.

[6] W. Hu, K. Sarveswaran, M. Lieberman, and G. H. Bernstein, "High-resolution electron beam lithography and dna nano-patterning for molecular qca," *IEEE Transactions on Nanotechnology*, vol. 4, pp. 312–316, May 2005.

[7] M. Niemier, R. Ravichandran, and P. Kogge, "Using circuits and systems-level research to drive nanotechnology," in *IEEE International Conference on Computer Design*, pp. 302–309, 2004.

[8] R. Zhang, P. Gupta, L. Zhong, and N. Jha, "Synthesis and optimization of threshold logic networks with application to nanotechnologies," in *Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, pp. 904–909, 2004.

[9] K. Walus, G. Schhof, Z. R., G. Jullien, and W. Wang, "Circuit design based on majority gates for applications with quantum-dot cellular automata," *Copyright IEEE Asimolar Conference on Signals, Systems, and Computers*, 2004.

[10] R. Zhang, P. Gupta, and N. Jha, "Synthesis of majority networks for qca-based logical devices," *International Conference on VLSI Design*, pp. 229–234, 2005.

[11] M. Ottavi, S. Pontarelli, V. Vankamamidi, and F. Lombardi, "Novel approaches to qca memory design," in *IEEE Conference on Nanotechnology*, pp. 699–702, 2005.

[12] M. Ottavi, S. Pontarelli, V. Vankamamidi, A. Salsano, and F. Lombardi, "Design of a qca memory with parallel read/serial write," in *IEEE Computer Society Annual Symposium on VLSI*, vol. 1, pp. 292–294, 2005.

[13] G. Toth and C. Lent, "Role of correlation in the operation of quantum-dot cellular automata," *Journal of Applied Physics*, vol. 89, pp. 7943–7953, June 2001.

[14] I. Amlani, A. Orlov, G. Snider, C. Lent, W. Porod, and G. Bernstein, "Experimental demonstration of electron switching in a quantum-dot cellular automata (qca) cell," *Superlattices and Microstructures*, vol. 25, no. 1/2, pp. 273–278, 1999.

[15] K. Walus, T. Dysart, G. Jullien, and R. Budiman, "QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata," *IEEE Trans. on Nanotechnology*, vol. 3, pp. 26–29, March 2004.

[16] J. Timler and C. Lent, "Power gain and dissipation in quantum-dot cellular automata," *Journal of Applied Physics*, vol. 91, pp. 823–831, January 2002.

[17] Y. Wang and M. Lieberman, "Thermodynamic behavior of molecular-scale quantum-dot cellular automata (QCA) wires and logic devices," *IEEE Transactions on Nanotechnology*, vol. 3, pp. 368–376, Sept. 2004.

[18] S. Bhanja and S. Srivastava, "A bayesian computing model for qcas," *NSTI Nanotechnology Conference*, 2005.

[19] S. Bhanja and S. Sarkar, "Graphical probabilistic inference for ground state and near-ground state computing in qca circuits," *IEEE Nanotechnology Conference*, pp. 444–447, 2005.

[20] S. Henderson, E. Johnson, J. Janulis, and P. Tougaw, "Incorporating standard cmos design process methodologies into the QCA logic design process," *IEEE Transactions on Nanotechnology*, vol. 3, pp. 2–9, March 2004.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. Morgan Kaufmann, 1988.

[22] P. Douglas Tougaw and C. S. Lent, "Dynamic behavior of quantum cellular automata," *Journal of Applied Physics*, vol. 80, pp. 4722–4736, Oct 1996.

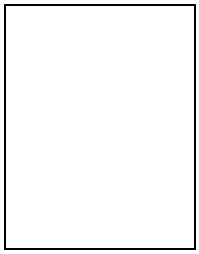
[23] R. G. Cowell, A. P. David, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag, 1999.

[24] W. Wang, K. Walus, and G. Jullien, "Quantum-dot cellular automata adders," in *IEEE Conference on Nanotechnology*, vol. 1, pp. 461–464, 2003.

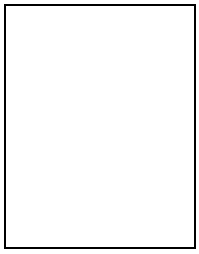
[25] S. Srivastava and S. Bhanja, "Bayesian modeling of quantum-dot-cellular-automata circuits,"

NSTI Nanotech Conference, May 2005.

[26] S. Bhanja, K. Lingasubramanian, and N. Ranganathan, "Estimation of switching activity in sequential circuits using dynamic bayesian networks," *Accepted for publication in International Conference in VLSI Design*, pp. 586–591, January 2005.



Saket Srivastava is currently a Doctoral candidate at the University of South Florida, Tampa. He received his BE in Electrical and Electronics from Regional Engineering College, Trichy(India) in 2003. His research interests include Nano-CAD tools, reliability centric Nano-CMOS and VLSI design automation.



Sanjukta Bhanja received her BE, MSc(engg.) and PhD from Jadavpur University, Indian Institute of Science, Bangalore and University of South Florida respectively. She is working as an assistant Professor in the Department of Electrical Engineering at the University of South Florida, since 2002. Her research interests include design automation, low power VLSI design, error and reliability-centric Statistical modeling and CAD, graphical probabilistic belief models, nano-CAD tools modeling quantum-mechanical behaviors for emerging technologies and nano-computing.