Improving Nano-circuit Reliability Estimates by Using Neural Methods

Azam Beg

College of Information Technology, United Arab Emirates University
P.O. Box 17551, Al-Ain, UAE
abeg@uaeu.ac.ae

Abstract. The reliability of nano-sized combinational circuits can be estimated by using different techniques, such as mathematical equations, Monte Carlo simulations, algorithmic approaches, and combinations of these. Commonly used equations are functions of gate count, and of the reliability and number of devices that make up the gates. The aim of this paper is to present a(n alternative) neural-based approach which is more accurate than applying simple equations, while being faster than the time-consuming Monte Carlo technique.

Keywords: Reliability estimation, reliability model, probability of failure, nano-metric circuits, neural network model.

1 Introduction

Nano-electronics has myriads of promising applications that would improve the well-being of humans in many areas of personal and work-lives, including: computing, communications, health, entertainment, mobile commerce, etc. For example, the small sizes of nano-devices would make it possible to create 100-billion to 1-trillion transistor microprocessors or Tera-byte memory chips. Nano-based medical diagnostics and treatment have a huge potential, as well as the anti-terrorism and security applications. Intelligent, cleaner and safer transport is also expected to benefit from the nano-sized devices. Despite the great prospects of nano-electronics, among the key issues that need to be addressed are:

- The development of tools and techniques to incorporate new computing architecture and devices; and
- The production of reliably functioning systems which are made from unreliable devices [1].

Mathematical equations can be used to estimate circuits' reliability (represented by the *probability of failure* of a circuit, $PF_{circuit}$ in this paper) [2]:

$$PF_{gate} = 1 - (1 - PF_{device})^{\delta}, \text{ and} \tag{1}$$

$$PF_{circuit} = 1 - (1 - PF_{gate})^{\gamma}. \tag{2}$$

A. Schmid et al. (Eds.): Nano-Net 2009, LNICST 20, pp. 270–275, 2009.

where δ is the number of devices per gate, γ is the number of gates in the circuit, PF_{gate} and PF_{device} are the probabilities of failure of the individual gates, and of the devices they are made of, respectively. However, when we compare equation-based estimates of circuits' reliability with the ones obtained using *nano-CR-EDA* (a Bayesian Network (BN) based reliability estimation tool [3]), or Monte Carlo methods, significant discrepancies between the sets of results were found (as shown in Fig. 1 and Fig. 2). The aim of our work is to propose a neural network (NN) based method for quickly estimating the reliability of nano-metric combinational circuits.

Section 2 of this paper provides a literature review. Section 3 briefly goes over artificial NNs, and then proceeds to the data acquisition and data processing techniques used to build the NN. In the same section, we also explain the process of NN development and its use for reliability estimations. Conclusions and the future work are included in Section 4.

2 Related Research

The unreliable operation of a digital circuit may be due to manufacturing inconsistencies or external factors (such as radiation). In recent years, a variety of tools and techniques have been developed to model the reliability of shrinking-dimension devices (and circuits built using them). A review of some of these follows.

A probabilistic single-event upset (SEU) fault model presented by Rejimon and Bhanja [6] employed logic induced fault encoded directed acyclic graph structured probabilistic BNs. The authors reported 5× faster estimation of SEU faults using this approach, for ISCAS-85 benchmark circuits. Tosun *et al.* [7] approach integrated reliability along with the traditional area and timing metrics. Their hardware-software co-design methodology searches through a library of design alternatives in order to yield a design with the best reliability figures. In [8], an analytical model for estimating a circuit's sensitivity to SEUs was developed. The model was proposed as an alternative to time-consuming Spice simulations. The model, a function of gate delay, approximates the behavior of an SEU and how it propagates through the circuit. The joint effects of smaller transistor geometries and lower supply voltages causing soft errors in CMOS transistors was modeled in [9]. The authors showed that it was possible to use birth-death queue model soft error prediction over a wide time scale. In [10], Beg presented a method for calculating reliability exactly using probability transfer matrices (PTMs). That tool (*AutoPTMate*) creates Matlab m-files from Verilog dataflow-style models for (relatively small) circuits. To handle larger circuits, Ibrahim *et al.* [3] introduced a tool called *nano-CR-EDA* that uses BNs to accurately calculate circuit reliability of larger circuits and to estimate the *PF* of output signals. Beg and Ibrahim [11] investigated relationships between topologies of the combinational circuits and their $PF_{circuit}$'s. They observed high correlations between the gate count and $PF_{circuit}$, and between the circuit-level/tier count and $PF_{circuit}$. As a follow-up of [11], we gathered reliability data for several thousand combinational circuits, by using *nano-CR-EDA* [3], [12]. This exercise revealed that the equation-based technique used noticeably overestimates the $PF_{circuit}$. So, in this paper, we propose an alternative, namely a higher accuracy reliability estimation scheme (than the equation-based method).

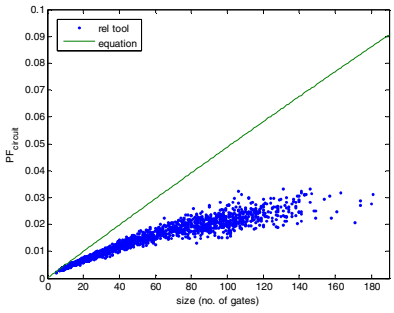

Fig. 1. Effect of *size* (gate count) on $PF_{circuit}$ when $PF_{device} = 10^{-4}$, showing that equations (1) and (2) tend to overestimate the $PF_{circuit}$

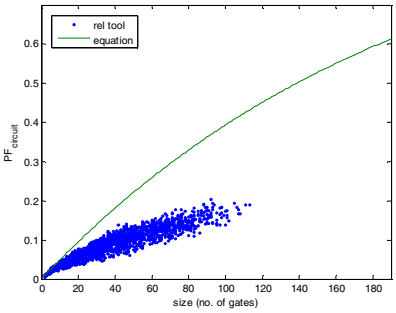

Fig. 2. Effect of *size* (gate count) on $PF_{circuit}$ when PF_{device} is 10^{-3}, showing that equations (1) and (2) tend to overestimate the $PF_{circuit}$

3 Neural Network for Reliability Estimation

3.1 What Are Neural Networks?

Artificial NNs emulate the functioning of biological brains. NNs' building blocks are called *neurons*. The neurons are interconnected to generate outputs in a parallel manner, unlike the conventional sequential computers. A *feed-forward neural network* (FFNN) generally comprises three types of neuron layers: *input*, *hidden*, and *output*. The outputs of each layer only feed the next layer and not any of the previous layers. The neurons multiply their inputs values with their respective *weights*, before summing them, and finally applying a *nonlinear activation function* (such as a sigmoid) to produce the neuron's output. The weights are determined by training the NNs on known input examples (*training sets*). The weights are iteratively adjusted in such a way that each set of inputs produces output(s) close to the example's output(s). An iteration of the weight-tuning process is known as an *epoch*. Some known input-output sets *(validation sets*) are used for validating the NN prediction accuracy. The validation sets are not "shown" to the NN during training [4].

3.2 The Training Database

First, we have randomly generated more than 8,000 Verilog files which describe combinational circuits with between 3 to 9 inputs (as random sets of sum-of-products). We have then synthesized (and optimized for *area*) the circuits using Synopsys Design Compiler's public domain library (0.35µm rev 1.3.1) [5]. The resulting (optimized) circuits have up to 194 gates in up to 14 levels. The circuits contain a mix of NOT, OR, NOR, AND, and NAND gates. For these circuits, we determined the exact $PF_{circuit}$ by using the *nano-CR-EDA* tool [3]. Several PF_{device} values (for example, 10^{-4} , 10^{-3} , etc.) were used in the many thousand runs of *nano-CR-EDA*.

The inputs to our proposed reliability-estimation NN model are: PF_{device} and γ(the number of gates in a circuit). The NN output is the estimated $PF_{circuit}$. It is well-known

that the raw form of real-world data is seldom in an NN training-friendly format. So an important (and often time-consuming step) is to *pre-process/transform* the dataset (inputs and their corresponding outputs), which ensures that the NN gives suitable importance to all variables, while training and while testing the NN's predictive performance. The pre-processing of PF_{device} data took two steps, namely, taking log_{10} and then applying Matlab's `premnmx` function (which normalizes data to [–1, 1] range. The other two variables ($PF_{circuit}$ and γ) were simply transformed with the `premnmx` function.

3.3 Training the Neural Network

We performed NN training using the Matlab's `newff` and `train` functions. We used a total of 10,376 datasets in the NN creation process. 80% of the datasets were used for training, 10% for testing, and 10% for validation. In order to tackle the issue of local minima, we trained the NNs at least 3 times in a given configuration (specifically, the number of hidden neurons). The NN models represent 2 inputs with 2 input-neurons and one output with a single output-neuron. Statistics for the best performing NNs are listed in Table 1. As we can see, just 7 hidden-neurons (see row #5) are sufficient to bring us quite close to a desired prediction error of 1%. The training progress for this network is shown in a larger NN (with 9 neurons), which provided only a marginal improvement in predictive performance. As expected, larger number of neurons helped train the NN in fewer epochs.

Table 1. Neural network training statistics. Matlab settings: net.trainParam.goal = 0.01; net.trainParam.lr = 0.01; net.trainParam.epochs = 5000; threshold functions for hidden and output layers = {'tansig','tansig'}; training function = 'traingd'.

No.	Neurons [input hidden output]	Epochs	Training error	Validation error	Testing error	"`net.trainParam.goal`" met?
1	[2 2 1]	5000	0.0944	0.0909	0.0989	No
2	[2 3 1]	5000	0.0213	0.0239	0.0210	No
3	[2 5 1]	5000	0.0128	0.0128	0.0134	No
4	[2 6 1]	5000	0.0138	0.0148	0.0134	No
5	[2 7 1]	3793	0.0100	0.0105	0.0099	Yes
6	[2 8 1]	2449	0.0100	0.0102	0.0099	Yes

3.4 Reliability Estimations

We compared the $PF_{circuit}$ predictions ($PF_{device} = 10^{-4}$) made by the NN with the ones obtained using the *nano-CR-EDA* tool. Visually, a high correlation between the two sets of data is evident (see Fig. 3). In exact numeric terms, the correlation is R = 0.961, very close to an almost perfect fit (R = 1.0).

For demonstrative purposes, we used the NN method to predict $PF_{circuit}$ for 40 different, random sets of input parameters (number of gates γ and PF_{device}). The $PF_{circuit}$ estimates generated by *nano-CR-EDA* and those generated by the *trained NN* are plotted in Fig. 4. These results provide high confidence in the use of an NN model

for the quick estimation of the reliability of combinational circuits. The NN reliability estimates are more accurate than the equation-based ones. A point to note is that the NN models are somewhat extrapolative in nature, so they can be used to estimate the reliability for input datasets having ranges beyond what was *shown* to the NN during training.

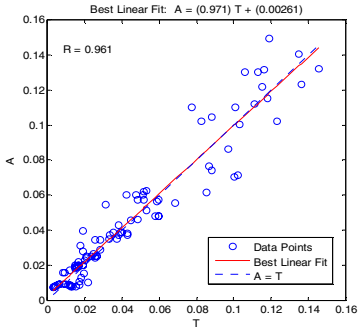

Fig. 3. A high correlation (R = 0.961) is found between the actual data points (from *nano-CR-EDA* tool) and the NN-produced estimates. (The best-fit linear equation A is compared with the *nano-CR-EDA* data points (circles). T is the ideal (45 degree) line for a perfect-fit situation).

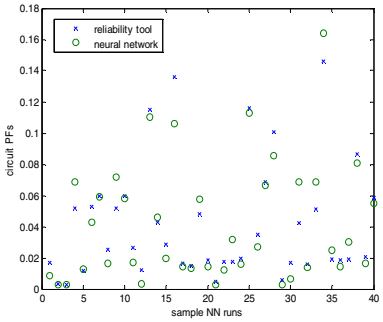

Fig. 4. Sample NN-predictions for randomly selected gate counts while PF_{device}=10^{-4}. Most NN predictions fall quite close to the exact values obtained using the *nano-CR-EDA* tool, while being obtained significantly faster.

4 Conclusions and Future Work

The NN-based nano-circuit reliability estimation method presented in this paper is an alternative to the equation-based and Monte Carlo (MC) simulation-based methods. The NN technique is quite time-efficient when compared to the MC method (but less accurate), and more accurate than the equation-based techniques (with a comparable time efficiency). As a continuation of the current work, we are planning to enhance the NN-based reliability estimations by including a larger number of circuit-configuration parameters, such as number and types of different gates, number of layers in the circuit, etc.

References

1. International Technology Roadmap for Semiconductors (ITRS). Semiconductor Industry Association, Austin, TX, USA (2007, 2008), `http://public.itrs.net/`
2. Forshaw, M.R.B., Nikolic, K., Sadek, A.: ANSWERS: Autonomous Nanoelectronic Systems With Extended Replication and Signaling. University College London, London, UK (2001); MEL-ARI #28667, 3rd Year Annual Report, 1–32 (2001), `http://ipga.phys.ucl.ac.uk/research/answers/reports/3rd_year_UCL.pdf`

3. Ibrahim, W., Beg, A., Amer, H.: A Bayesian Based EDA Tool for Accurate VLSI Reliability Evaluations. In: International Conference on Innovations in Information Technology (Innovations 2008), Al-Ain, UAE, pp. 101–105 (2008)
4. Witten, I.H., Frank., E.: Data Mining. Morgan Kaufmann, New York (2005)
5. RTL-to-Gates Synthesis Using Synopsys Design Compiler (rev. March 2, 2008), http://csg.csail.mit.edu/6.375/handouts/tutorials/tut4dc.pdf
6. Rejimon, T., Bhanja, S.: An Accurate Probabilistic Model for Error Detection. In: International Conference on VLSI Design (VLSID 2005), Piscataway, NJ, USA, pp. 717–722 (2005)
7. Tosun, S., Mansouri, N., Arvas, E., Kandemir, M., Xie, Y., Hung, W.L.: Reliability-Centric Hardware/Software Co-Design. In: International Symposium on Quality of Electronic Design (ISQED 2005), pp. 375–380. IEEE Computer Society, Los Alamitos (2005)
8. Gilson, I.W., Ivandro, R., Michele, G.V., Kastensmidt, F.G.L.: Single Event Transients in Dynamic Logic. In: Annual Symposium on Integrated Circuits and Systems Design, pp. 184–189. ACM, Ouro Preto (2006)
9. Li, H., Mundy, J., Patterson, W., Kazazis, D., Zaslavsky, A., Bahar, R.I.: A Model for Soft Errors in the Subthreshold CMOS Inverter. In: Workshop on System Effects of Logic Soft Errors (SELSE-2), Urbana-Champaign, IL, USA (2006)
10. Beg, A., Ibrahim, W.: On Teaching Circuit Reliability. In: Frontiers in Education Conference (FIE 2008), Saratoga Springs, NY, USA, pp. T3H12- T3H17 (2008)
11. Beg, A., Ibrahim, W.: Relating Reliability to Circuit Topology. In: North Eastern Workshop on Circuits and Systems (NEWCAS 2009), Toulouse, France (2009) (in press)
12. Ibrahim, W., Beiu, V.: A Bayesian–based EDA Tool for nano-Circuits Reliability Calculations. In: International ICST Conference on Nano-Networks (Nano-Net 2009), Luzern, Switzerland (2009) (in press)