Turtle Logic: A new probabilistic design methodology of nanoscale digital circuits

Lancelot Garcia-Leyva
Universidad Autónoma de Tlaxcala
Facultad de Ciencias Básicas, Ingeniería y Tecnología.
C/Apizaquito s/n, Apizaco, 90300
Tlaxcala, México.
Email: lancelot@eel.upc.edu

Antonio Calomarde, Francesc Moll and Antonio Rubio
Universitat Politecnica de Catalunya
Department of Electronic Engineering
C/Jordi Girona, 31, 08034, Barcelona, Spain.
Email: {calomarde, moll}@eel.upc.edu,
antonio.rubio@upc.edu

***Abstract*— As devices and operating voltages are scaled down, future circuits will be plagued by higher soft error rates, reduced noise margins and defective devices. A key challenge for the future technologies is to retain circuit reliability in the presence of faults and noise. The Turtle Logic (TL) is a new probabilistic logic method based on port redundancy and complementary data, oriented to emerging and beyond CMOS technologies. The TL is a technology independent method, which aims to improve tolerance to errors due to noise in single gates, logic blocks or functional units. The TL operation is based on the consistency relation of redundant inputs. In case of discrepancy, the output of the system keeps the previous value, therefore avoiding the propagation of incorrect inputs. Simulations show an excellent performance of TL in the presence of large random noise at the inputs, as well as intrinsic noise (thermal noise and flicker noise) and shot noise in the power source.**

I. Introduction

With the progress on VLSI process technology, the design complexity and the transistor density in system increases rapidly, causing that the power consumption and power density in system rise with the same trend. The size of CMOS devices is scaled down to the nanoscale level where interferences (soft errors), becoming significant, affect the VLSI circuit performance [1]. Future electronic devices are expected to operate at lower voltage supply to save power, especially in ultimate and new technologies. The resulting reduction of logic levels approaches the thermal noise limit, and consequently signal to noise margins are reduced, exposing computations to higher soft-error rates [1].

In order to design reliable circuits with unreliable components, new design techniques must be introduced. The problem of designing reliable systems with unreliable components traces back to Von Neumann, who proposed the N-tuple Modular Redundancy (NMR) technique [2]. In the NMR each gate, logic block or functional unit is replicated N times and the final output is obtained through a majority voting circuit, where the majority voting gate need to be perfect and free of faults. Several approaches have been proposed based on the NMR such as the Triple Modular Redundancy (TMR) or the Cascaded TMR (CTMR) [3].

Additional proposals have appeared in the literature addressing the problem from the point of view of noise tolerance instead of exclusively considering hard defects. For instance, the so called Probabilistic Logic based on Markov Random Field theory (MRF) [4], identifies valid and invalid states and derives a logic implementation which tries to reinforce valid states in order to increase their probability and decrease the probability of erroneous states. Another approach is the Probabilistic CMOS (PCMOS) [5], where its principal objective is to develop and comprehensively characterize probabilistic CMOS circuits that may be used to build energy efficient computing platforms, where reliability of the circuit is not necessarily the objective.

Our proposal is based on the scenario that all components are noisy and hence they may fail. In order to increase circuit reliability, we propose to increase the number of input and output ports in each module, which is a form of data redundancy. This new proposal is different from the Von Neumann NMR approach in two aspects: that each module is not replicated n times, but instead, the module is redesigned with I/O ports replicated n times to increase the reliability, and voting circuit is not needed.

The structure of the paper is as follows: in Section II, the basic concepts and principles of error probability are introduced, as well as the error caused by noise in one digital node. In section III, Port Redundancy as a method to improve the reliability of a module under noise scenario is presented. In Section IV our proposal called Turtle Logic is introduced, explaining the implementation of a NOT, AND-NAND, OR-NOR and XOR-XNOR gates as examples. Section V presents simulation results of the different approaches using a 90nm CMOS process technology. Finally, the conclusions are presented in Section VI.

II. Probabilistic formulation

In ultra-low V_{DD} CMOS scenario and future technology, the noise margin will be too small, and therefore soft errors due to all kinds of noise sources become crucial. For simplicity,

This research work has been supported by the Spanish Ministry of Science and Innovation (MICINN) through projects TEC2008-01856 and PLE2009-0024 (ENIAC MODERN project), with the participation of FEDER funds and "Plan E" funds, and Government of Mexico under grant 164013 CONACyT and UATLX-235 PROMEP. The group of research is considered a consolidated group by the MICINN.

978-1-4244-7773-9/10/$26.00 ©2010 IEEE

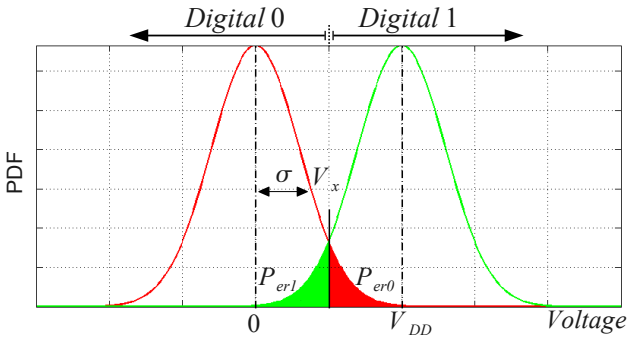

Fig. 1. The Probability Density Function for digital values with $\mu = 0V$ and $\mu = V_{DD}$ voltages in a digital node, under influence of White Gaussian Noise.

and without loss of generality, we model signals as two-logic levels, with additive white Gaussian uncorrelated noise, with mean μ, standard deviation σ and variance σ^2.

A. Error caused by noise in one digital node

Fig. 1 shows the voltage probability distribution in a noisy digital node. It can be described as two Gaussian distributions centered around low and high voltages (0 and VDD respectively). An error can be produced when a logic 1 is interpreted as a logic 0 and vice versa. The respective probabilities of such cases are denoted as P_{er1} and P_{er0}, they are obtained integrating the logic 1 and logic 0 PDF functions beyond the logic threshold (denoted as Vx in Fig. 1).

Assuming that logic 1 and logic 0 are equally probable, the probability that the data in a node is incorrect is given by:

$$P_e = \frac{P_{er0} + P_{er1}}{2} \tag{1}$$

III. Port Redundancy

In this section, we analyze the reliability improvement of input and output ports redundancy (PR) to the modules. In Fig. 2, we show how a module can have $(n-1)$ replicated ports, where the conventional differential logic is a particular case $(n = 2)$. The respective replicated ports can be either equal or complementary to the original port. In general, port redundancy has three different scenarios, which probability can be calculated as follow:

1) Correct scenario (CS): All the inputs logic values are interpreted correctly. The probability of this correct scenario when n ports are considered is given by equation (2).

$$P_{cn} = (1 - P_e)^n \tag{2}$$

2) Discrepant scenario (DS): One or several ports have voltage values which cause an incorrect interpretation (logic 0 instead of intended logic 1 or vice versa). The probability that this happens is given by equation (3).

$$P_{dn} = \sum_{k=1}^{n-1} \frac{n!}{k!(n-k)!} (1 - P_e)^{n-k} P_e^k \tag{3}$$

3) Error scenario (ES): All the ports are misinterpreted, as a logic 1 instead of an intended logic 0 or vice versa. In this case, the port interpretation is apparently coherent (they keep their internal coherence). The probability of this scenario is given by equation (4).

$$P_{en} = P_e^n \tag{4}$$

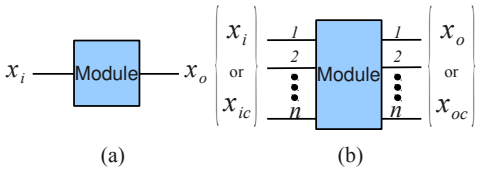

Fig. 2. Port redundancy. (a) Single gate with one input and one output port, (b) Single gate with $(n-1)$ replicated ports.

The specific case of a NOT gate with PR1 (PR with $n = 1$) will be considered in the next section.

A. Probability scenarios for a NOT gate with PR1

The probabilities for respective scenarios are given by:

1) CS: The values of the both ports are correct, Fig. 3(a).

$$P_{c2} = (1 - P_e)(1 - P_e) = (1 - P_e)^2 \tag{5}$$

2) DS: Only one value of the two ports is incorrect, Fig. 3(b).

$$P_{d2} = 2P_e(1 - P_e) \tag{6}$$

3) ES: Both ports are incorrect, Fig. 3(c).

$$P_{e2} = P_e P_e = P_e^2 \tag{7}$$

Each redundant port is generated by a different driver, so that the noise can be considered statistically independent.

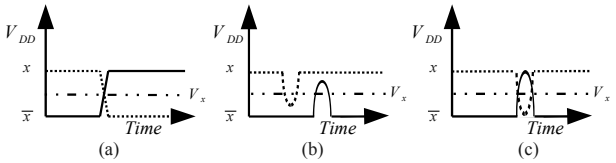

Fig. 3. Scenarios for CMOS single gate with one complementary replicated port, (a) Both signal are correct (P_{c2}), (b) One of both signals is incorrect (P_{d2}) and, (c) Both signal are incorrect (P_{e2}).

B. Analysis of error reduction with port redundancy (PR)

The error probability (P_{en}) follows an exponential relationship with the number of replicated ports as shown in equation (4). Fig. 4 shows the results of error probability for a single gate with redundancies from 0 to 9, for an SNR from -3 to 16 dB. As expected, if the number of replicated ports is increased, the error probability is reduced. For the specific case of PR1 with SNR=-3dB ($\sigma = 0.5$ Vrms and $V_{DD} = 0.5$ V), the decrease in error probability is 34.57%. When the PR is increased further to 2, 3 and 4, the corresponding decrease in error probability is smaller each time, with decrements respect

to the previous PR of 11.95%, 4.13%, 1.41%, 4.91e-3%, respectively, so that for PR beyond 4 a negligible decrement in error probability is observed. For SNR greater than -3dB, the advantage of redundancy 1 over redundancy 2, 3, etc. is lower every time. Conversely, the cost in hardware of a digital circuits with PR is highly increasing. Therefore, a redundancy of one (PR1 – increasing in one the number of ports) is considered in this work, as a good balance between reliability and overhead.

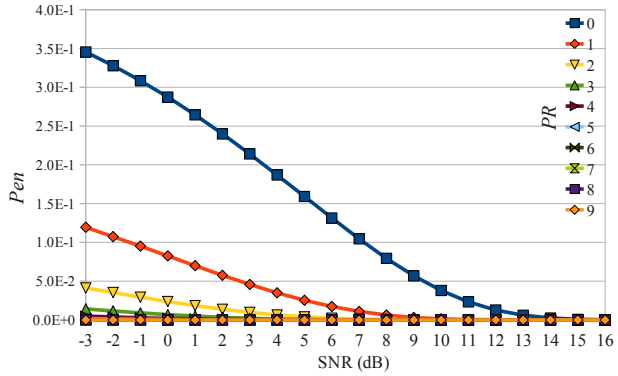

Fig. 4. Error probability for a single gate with 0-9 port redundancies.

IV. New Design for Robust Digital Circuits

The design proposal presented in this paper consists in that logic functions are implemented such that the consistency of all redundant inputs is checked. In case of discrepancy, the output of the gate holds its previous values, therefore avoiding the propagation of incorrect inputs to the outputs. We call this concept Turtle Logic (TL). To show the operating of TL design, we use a NOT gate with PR1 and complementary data. The circuit obtained is shown in Fig. 5, and this TL NOT gate has two possible cases:

- When the input ports x_i and x_{ic} have complementary values, for instance, $x_i = 0$ and $x_{ic} = 1$ or vice versa, the TL NOT identifies the inputs as correct values and forces the complementary output ports driving the appropriate values according to the inverter function ($x_o = 1$, $x_{oc} = 0$).
- When the inputs have the same value ($x_i = 0$, $x_{ic} = 0$ or $x_i = 1$, $x_{ic} = 1$), TL NOT gate detects these values as an error. Then, the output ports of the system are forced to hold their previous values.

It has to be noted that when both inputs take simultaneously an incorrect value due to noise, the TL gate admits them as correct values, thus yielding an incorrect output. However, the probability of this simultaneous event is very low when there is a statistical independence of the noise in redundant ports, as shown in equation (4) and Fig. 4.

Figs. 6 and 7 show a NAND and an XOR gate using the TL methodology described before, and the data about the overhead in gates of their respective implementations is shown in Table I.

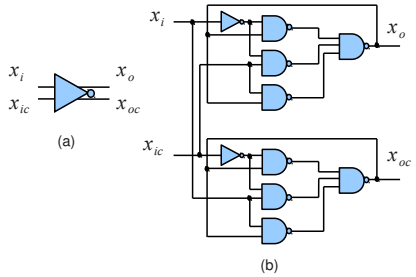

Fig. 5. NOT gate using TL with PR1. (a) Symbol, and (b) Schematic view.

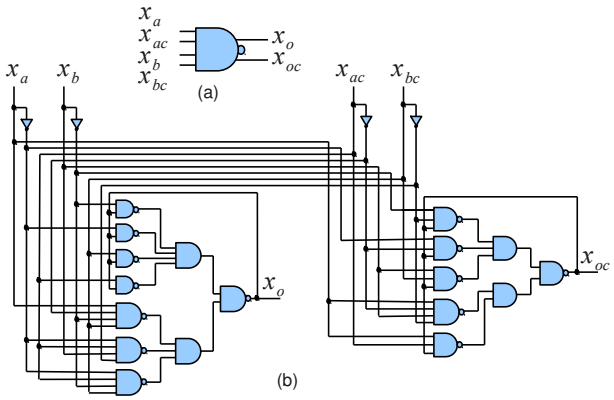

Fig. 6. NAND gate using TL with PR1. (a) Symbol, and (b) Schematic view.

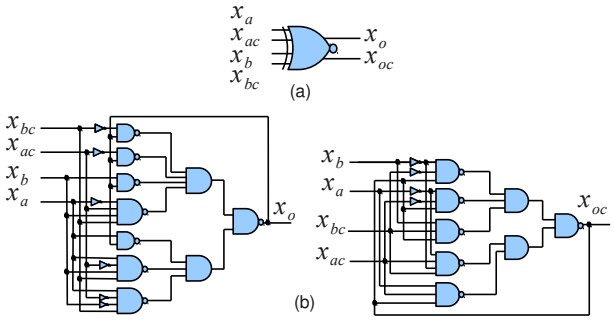

Fig. 7. XOR gate using TL with PR1. (a) Symbol, and (b) Schematic view.

V. Simulation Results

In order to validate the improved robustness of the TL proposal Fig.5(b), we implemented a NOT gate with three methods: standard CMOS, MRF [4] and TL. The results shown in Fig. 8 are the result of SPICE simulations, using 90nm technology device models with a V_{DD} of 0.5V at temperature of 100 ^{o}C. Internal noise was generated using transient-noise option of the Spectre circuit simulator, which generates thermal noise and flicker noise to the channel of each transistor, and shot noise to each power supply at each time step. The noise generated to each transistor and power source were amplified 100 times respect to intrinsic noise of the 90nm technology, in order to emulate the behaviour of future technology with very noisy components. It is worth noting that nominal V_{DD} for this technology is 1V, but in this work it was decreased to 0.5V to simulate an equivalent

TABLE I

COMPARATIVE CMOS, MRF AND TURTLE LOGIC TECHNIQUES FOR A NOT GATE, USING ALTERNATING 5,000 1'S AND 0'S, WITH PERIOD T OF 4NS.

Logic		# Inputs	# Outputs	# Gates	ER x_o	ER x_{oc}
NOT	Static	1	1	1	8,298	N. A.
	MRF	2	2	8	1,580	2,850
	TL	2	2	10	71	73
AND-NAND	Static	2	1	2/1	5,778	10,864
	TL	4	2	18	596	939
OR-NOR	Static	2	1	2/1	4,061	13,351
	TL	4	2	18	930	1,163
XOR-XNOR	Static	2	1	5/5	12,313	14,545
	TL	4	2	18	1,683	1,867

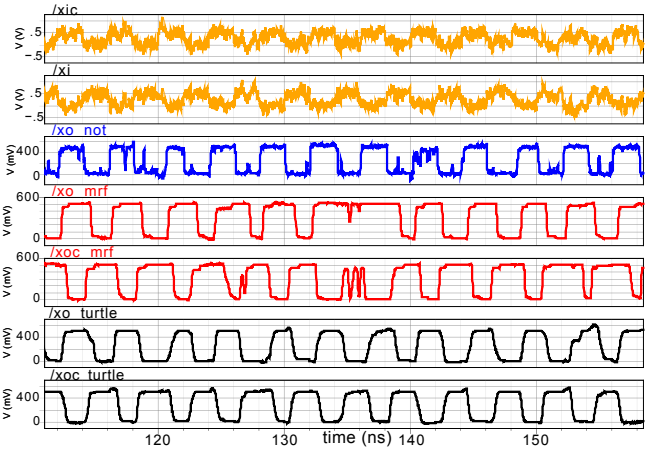

Fig. 8. Simulation results for a NOT gate using Standard CMOS, MRF and Turtle logic.

environment in future technologies, where thermal and flicker noise will have greater relevance. The input noisy signals were generated with Additive White Gaussian Noise (AWGN) modules implemented using Verilog-A, with 0V mean and standard deviation of 0.5 Vrms to both logic levels 1 and 0, using different seed for each input. Uncorrelated noise sources are applied to each of the inputs, while the outputs of the device under test have identical NOT gates as load.

The simulated transient waveforms are shown in Fig. 8, where *xi* and *xic* are common input signals to all devices. *xo_not*, *xo_mrf*, *xoc_mrf*, *xo_turtle* and *xoc_turtle* are the output signals for standard CMOS, MRF and Turtle Logic, respectively. It can be seen that Turtle Logic has a very good robustness in an environment where noise levels reach similar magnitude as V_{DD}.

Table I shows the overhead as well as the noise error robustness of standard CMOS, MRF gates and Turtle logic approaches for the NOT, AND-NAND, OR-NOR and XOR-XNOR gates. As the reliability parameter to measure and compare the noise error robustness of these approaches we use the error ratio (ER) factor, shown in the respective column of table I. The meaning of this ER factor depends on the number of spurious signals observed at the outputs of the circuits after a periodic sequence of 0's and 1's is applied at their inputs. The period of the sequence is 4 ns for the NOT gates, and for the other gates, one of the inputs has 4 ns period and the other has 8 ns period in order to have all the possible combinations. A spurious signal (SS) is any signal which has an unexpected voltage in a time interval [6]. An SS is counted in the results table as one ER when it has an amplitude greater than $V_x \pm 0.1V_{DD}$ (for a 0 logic and 1 logic, respectively) and width at least 10% of the signal period ($T = 0.4$ ns in our example).

As can be seen in Table I, in the case of a NOT gate we found 8,298 spurious for standard CMOS, 1,580 for the MRF, and 71 for TL. TL NOT gate has a much lower ER than Standard CMOS or MRF, therefore demonstrating that TL is more robust in extremely noisy environments.

VI. CONCLUSIONS

The results shown in the previous section demonstrate an excellent tolerance to high noise of the present approach compared to the previous proposals. This paper shows how increasing port redundancy decreases the error probability. This result in addition with Turtle Logic is the basis for a new way of designing logic blocks which are tolerant to a high level of noise, under the assumption of statistical independence of noise (the addition of thermal noise, flicker noise and coupled noise) in future and new technologies.

This improvement is obtained at the expense of a large overhead in area and timing that must be carefully evaluated. It can be argued that the obtained area and timing penalties would be prohibitive in today's noise environment and technologies. However, this level of overhead can be acceptable in future technologies beyond CMOS, where the scenario of low SNR (extremely high noise) considered in this paper will need reliable circuits. In this conditions, the design strategy here proposed will be an adequate option.

REFERENCES

[1] [Online]. Available: http://www.itrs.net/Links/2009ITRS/Home2009.htm

[2] J. V. Neumann, *Probabilistic Logics the Synthesis of Reliable Organisms from Unreliable Components*, C. E. S. Automata Studies and e. J. McCarthy, Eds. Automata Studies, C. E. Shannon and J. McCarthy, eds., 1956.

[3] W. H. Pierce, *Failure-Tolerant computer design*. Academic Press., 1965.

[4] K. Nepal, R. Bahar, J. Mundy, W. Patterson, and A. Zaslavsky, "Designing logic circuits for probabilistic computation in the presence of noise," in *Design Automation Conference, 2005. Proceedings. 42nd.*, June 2005, pp. 485–490.

[5] P. Korkmaz, "Probabilistic CMOS (PCMOS) in the nanoelectronics regime," Ph.D. dissertation, Georgia Institute of Technology, Dec. 2007.

[6] F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 39, no. 10, pp. 749–752, Oct 1992.