



HAL
open science

Techniques and Prospects for Fault-tolerance in Post-CMOS ULSI

Yangyang Tang, Sundararajan Gopalakrishnan, Chris Winstead, Emmanuel
Boutillon, Christophe Jego, Michel Jezequel

► **To cite this version:**

Yangyang Tang, Sundararajan Gopalakrishnan, Chris Winstead, Emmanuel Boutillon, Christophe Jego, et al.. Techniques and Prospects for Fault-tolerance in Post-CMOS ULSI. ULSIWS 2012: 21st International Workshop on Post-Binary ULSI Systems, May 2012, France. pp.1-7. hal-00955755

HAL Id: hal-00955755

<https://hal.science/hal-00955755>

Submitted on 5 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Techniques and prospects for fault-tolerance in post-CMOS ULSI

Yangyang Tang^{*†}, Gopalakrishnan Sundararajan[†], Chris Winstead[†], Emmanuel Boutillon^{*},
Christophe Jégo[‡] and Michel Jézéquel[§]

^{*}Université de Bretagne Sud, UMR CNRS 6258 Lab-STICC, Lorient, France

[†]Dept. of Electrical and Computer Engineering, Utah State University, Logan, Utah 84322

[‡]Institut Polytechnique Bordeaux, UMR CNRS 5218 Lab-IMS, Bordeaux, France

[§]Institut TELECOM/TELECOM Bretagne, UMR CNRS 6258 Lab-STICC, Brest, France

Abstract—This paper presents a survey of fault-masking techniques suitable for tolerating short-duration transient upsets in minimum-scale switching devices. Two types of fault masking are considered. The first type, coded dual-modular redundancy (cDMR), represents a family of parity-checking methods suitable for correcting a low rate of transient upsets. The second type, Restorative Feedback (RFB), is a triple-modular solution suitable for compensating a higher rate of transient upsets. We show that cDMR can be used efficiently for crossbar-style logic, but is not efficient in general for all logic styles. By contrast, RFB offers a fixed redundancy, and can be applied in general to any logic circuit. Finally, we propose novel circuits for ternary Muller C implementation based on carbon nanotube FET devices.

I. INTRODUCTION

When devices are scaled to minimum physical dimensions, very small energy is available for storing and communicating logic levels between gates. Additionally, when devices are integrated at maximum density, the resulting high thermal density may seriously increase transient noise. Logic circuits must therefore operate with reduced signal-to-noise ratios that result in a high rate of momentary logic faults and memory state upsets. Transient signal upsets can be caused by thermal noise, electromagnetic interference (e.g. from switching events in neighbouring devices on the same chip), timing failures in synchronous pipelines, power supply noise, high-energy particle collisions and other sources [1]–[4]. Various techniques are available to mask the occurrence of momentary faults for binary and multiple-valued logic system [5]–[11].

In this work we present a survey of methods for tolerating transient upsets in nano-CMOS and post-CMOS ULSI systems. We first present a coded dual-modular redundancy (cDMR) technique, which has low redundancy in some cases. The cDMR method is permissible if certain constraints are imposed on the logic synthesis. The required constraints sometimes result in more complex logic synthesis, which may limit the generality of cDMR solutions.

Triple-Modular Redundancy (TMR) [12] provides a more general solution for protecting black-box logic modules without requiring any constraints on the logic synthesis method. TMR-style solutions have a fixed redundancy of two, whereas the cost of cDMR varies for each synthesized circuit. Hence it may be preferable to use TMR because it provides more certain characteristics during architecture planning.

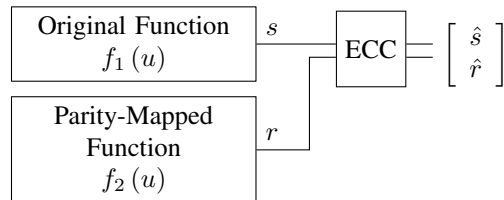


Figure 1: Architecture of the LFCT method. The original logic function f_1 is composed with a block ECC code to create the parity-mapped function f_2 . Outputs are then corrected by the ECC circuit.

The Restorative Feedback method (RFB) [13] is a TMR-style solution which is specialized for correcting transient faults in digital logic. The RFB method was previously shown to be applicable to multiple-valued logic circuits. In this paper, we present novel ternary implementations of the Restorative Feedback (RFB) method, which is comprised of Muller C-elements. We present two ternary C-element implementations using carbon nanotube FET (CNTFET) devices, which may be used to realize a multiple-valued RFB solution. A review of alternative implementations and tradeoffs is also presented.

II. CODED DUAL-MODULAR REDUNDANCY TECHNIQUES

One of the authors (Winstead) has proposed an LDPC-coded Fault Compensation Technique (LFCT) [14] which achieves reliable operation in the presence of transient and permanent defects. The LFCT method uses Gaudet and Rapley’s theory [15] of stochastic decoding to correct errors that appear at the output of some logic computation.

The LFCT architecture is depicted in Fig. 1. In the proposed LFCT system, a logic function f_1 is subject to transient internal upsets, causing one or more errors to appear on the function’s outputs, s . To correct these error, a second function f_2 is introduced which maps the input data u to a vector of parity-check bits r , so that the output $[s \ r]$ forms a codeword according to a traditional block error correction design. In this system, we assume that the input vector u is error-free. If the number of errors in $[s \ r]$ is sufficiently small, then they are corrected by the ECC module, resulting in a correct output vector $[\hat{s} \ \hat{r}]$. Finally the corrected data \hat{s} can be propagated to other logic modules for additional computations.

If the logic function is constrained to use a crossbar synthesis architecture, as commonly proposed for post-CMOS electronics [16], [17], then the fault statistics are suitable for using the LFCT technique. To explain this constraint, we may contrast the two circuits shown in Fig. 2. Both circuits implement a two-bit adder function. Fig. 2(a) shows a traditional ripple-carry implementation, and Fig. 2(b) shows a crossbar implementation. In the ripple-carry implementation, a single gate error may propagate to the several of the output signals. An example of error propagation is indicated by the \star symbol in Fig. 2(a). In the LFCT system, error-propagation may induce many simultaneous faults in the $[s r]$ codeword, which are not likely to be correctable.

In the crossbar implementation, as shown in Fig. 2(b), logic is implemented by fabrics of AND-logic and OR-logic. The “dots” in Fig. 2(b) indicate the placement of junctions which physically implement the logic operations. In this style of implementation, every operation is associated with only a single output. If a momentary fault occurs at some junction, it will propagate only to a single output. This style of logic guarantees that single-error events are correctable. The major disadvantage of crossbar logic is that the operation counts are not optimal. The crossbar adder in Fig. 2(b), for instance, has 57 separate operations. Crossbar logic does not generally obtain minimized gate complexity, but it offers improved reliability by eliminating error propagations. The added gate complexity could be considered as a form of redundancy. Furthermore, if crossbar-style logic is used with the LFCT method, then the logic function can be duplicated only once to provide error correction.

The authors (Tang, Winstead, Boutillon, Jégo and Jézéquel) have developed a specialized LFCT method, called the LDPC Stochastic Decoding (LSD) architecture [18], which can be considered as a circuit-level implementation of the LFCT concept. To evaluate the LSD solution, an example system was simulated using a variety of error-correcting codes. All codes are rate 1/2, meaning that there is one redundant parity bit for each systematic bit. The Bit Error Rate (BER) results for the LDPC codes applying LSD algorithm are shown in Fig. 3. In our simulations, the output bits s and r from $F(x)$ and $E \cdot F(x)$, respectively, are assumed to have a uniform independent error probability of α . Internal errors are also modeled in the correction circuit, with a uniform error probability equal to ϵ . Stuck-at faults were also inserted in uniformly random positions in $[s r]$ with a fault rate of β , here, $\beta = 0.001$.

As long as $\alpha < 0.05$, the output s from $F(x)$ can be recovered with a significantly reduced error probability. As α is reduced below β , the performance becomes dominated by the gate-level fault probability ϵ . The results show that, in the case of $\epsilon = 10^{-5}$, the LSD architecture introduces gains about two orders of magnitude by comparison with uncoded data (output s from $F(x)$). Consequently, the proposed decoding method is able to suppress the resulting error probability to a level equal to the decoder’s internal fault rate.

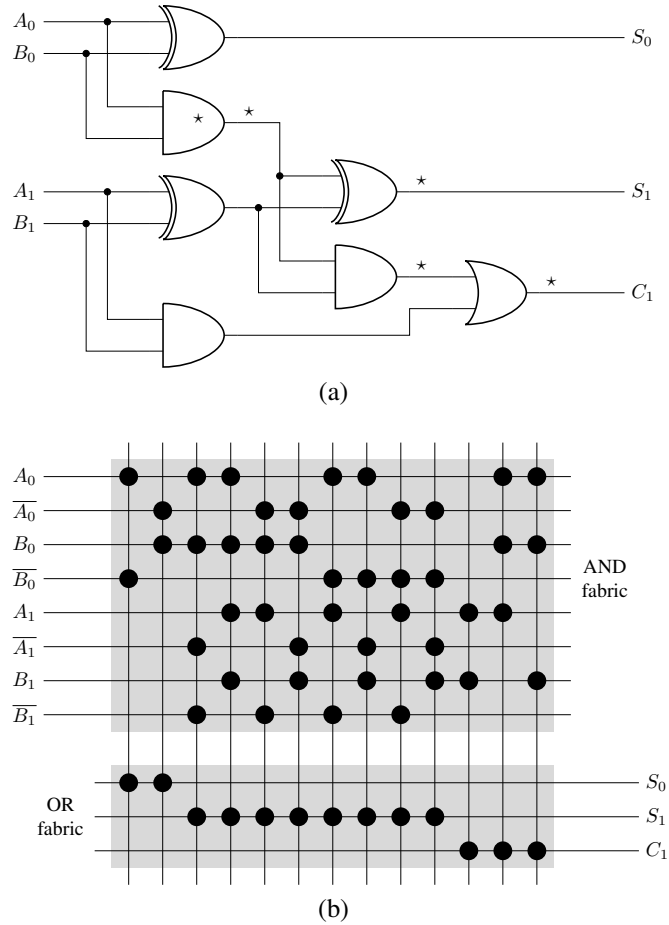


Figure 2: Implementations of a two-bit binary adder function, representing a traditional ripple-carry design (a), and a crossbar design suitable for some nanoelectronic device families (b). The ‘ \star ’ symbol indicates the occurrence of an error which propagates to multiple signals.

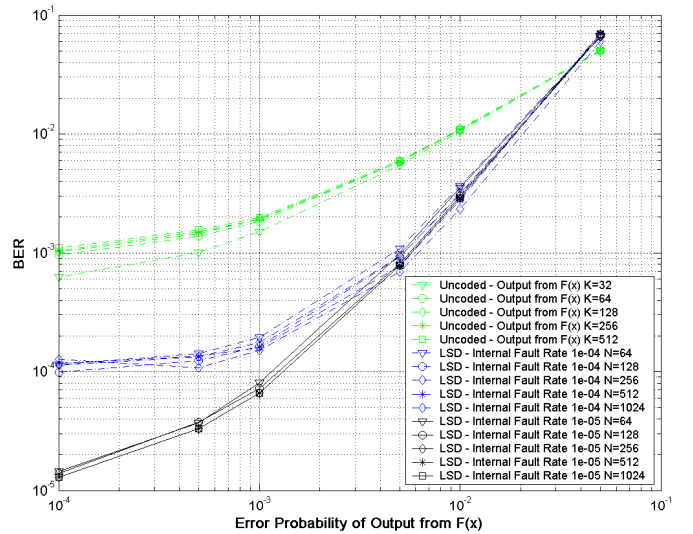


Figure 3: Simulation results for rate-1/2 LDPC codes based on LSD architecture with five iterations. The hard-fault rate β is 0.001.

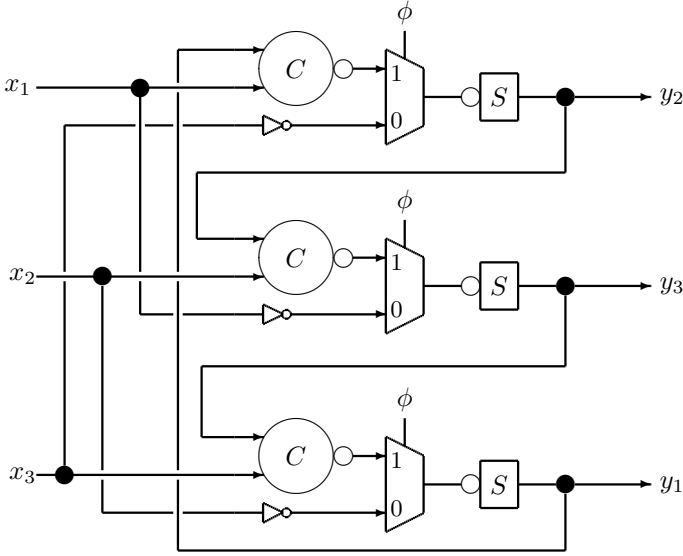


Figure 4: The RFB circuit based on Muller C-element gates. The S gate is a storage element.

III. TMR METHOD: RESTORATIVE FEEDBACK

TMR methods require duplicating a logic function twice, and require no constraints on how the logic functions are synthesized (i.e. TMR methods can correct cases of error-propagation). Although cDMR requires only a single duplication, it requires suboptimal synthesis that may significantly increase the operation count. Therefore TMR methods are more attractive to be used with traditional non-crossbar logic styles. A variant of TMR, referred to as Restorative FeedBack (RFB), was previously described by one of the authors (Winstead) [13]. The RFB method is based on Muller C-element gates, as shown in Fig. 4. The RFB circuit is derived from the theory of stochastic iterative decoding, and achieves better performance than traditional majority-based TMR for correcting momentary upsets. The RFB method is also applicable to M -ary logic systems as well as conventional binary logic circuits. A ternary RFB circuit was previously described using CMOS semi-floating gate circuits.

In this section, we consider a post-CMOS implementation based on carbon nanotube FET devices (CNTFETs). We first review the basics of ternary logic, and then investigate the CNTFET transistor, its operation and multi V_{th} design in the context of CNTFET devices. Muller C Element is presented afterwards. At last, two proposed circuits for implementing ternary C-element are illustrated, as well the simulation results.

A. CNTFET Transistor

Fig. 5 shows the cross-section of a CNTFET transistor. CNTFET transistors are constructed by replacing the silicon channel of traditional MOSFETs with semiconducting carbon nanotubes (CNT). The CNTs deployed in the channel region can be either metallic or semiconducting depending on the angle of atom arrangement along the tube. Similar to traditional MOSFETs, CNTFETs have four terminals: drain, gate, source and back-gate [19].

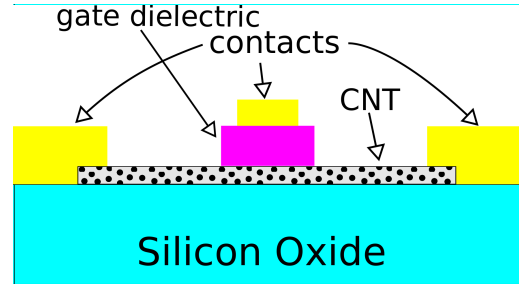


Figure 5: Side view of a CNTFET transistor

B. Overview of Ternary Logic

In this subsection, we review the basics of ternary logic. Ternary logic functions are defined as those having importance if a third value is introduced into the binary logic value. Ternary Logic values consists of three states (1, 2, 3) representing false, undefined and true respectively [20]. Table I shows the logic values and the corresponding voltage levels used in this paper to represent ternary logic values.

The inverter is a basic logic gate used in digital design. In ternary logic, there are three types of inverters: positive ternary inverter (PTI), standard ternary inverter (STI) and negative ternary inverter (NTI). The truth table for all the inverters mentioned above is shown in table II.

The current through the CNTFET transistor is controlled by adjusting the device parameters such as gate length, number of nanotubes, chirality vector and the pitch distance [21]. The operation of CNTFETs is very similar to traditional MOSFETs except for differences in the device orientation. In contrast to MOSFETs, the source and drain terminals of CNTFETs transistors are not interchangeable. This is because doping variations are introduced in the source and drain regions during lithography rendering it impossible to interchange them [22].

For CNTFET circuits, multiple V_{th} transistors can be achieved by tuning the chirality of each transistor. Even though CNTFET are relatively new, plenty of existing literature has documented the feasibility of adjusting the V_{th} of transistors by growing different nanotube diameters [23].

The V_{th} of the intrinsic CNT channel is given by :

$$V_{th} \approx \frac{E_g}{2e} = \frac{\sqrt{3}}{3} \frac{aV_\pi}{eD_{CNT}}$$

where $a = 2.49 \times 10^{-10} m$, $V_\pi = 3.033 ev$, and

$$D_{CNT} = \frac{\sqrt{3}a_0}{\pi} \sqrt{n^2 + m^2 + nm}$$

where $a_0 = 0.142 nm$ is the inter-atomic distance between each carbon atom and its neighbor and (m, n) is called the chirality vector that describes the structure of a carbon nanotube [24].

C. Muller C Element

In this subsection, we describe the binary and ternary Muller C elements and its operation. The C element has been employed in asynchronous circuits design [25]. Fig. 6 shows the circuit and the schematic of the Muller C element. The

voltage-Level	Logic Value
0	0
$0.5V_{DD}$	1
V_{DD}	2

Table I: Ternary Logic Symbols

Input X	STI	PTI	NTI
0	2	2	2
1	1	2	0
2	0	0	0

Table II: Ternary Inverter Truth Table

Muller C-element is a fundamental circuit element widely used for control synchronization in asynchronous designs. In general, a C-element is a state holding circuit which is transparent when all its inputs are equal, and holds the previous output otherwise. The C element consists of two inputs a, b and one output c. Table III shows the truth table of the binary Muller C element. The logic equation describing the behaviour of the Muller C element is described as follows:

$$c = \bar{c} (a + b) + a.b$$

The ternary Muller C element is similar to its binary counterpart except for the fact that the inputs and the output could take three logic values. Hence there will be an increase in the number input combinations. Table IV shows the truth table of ternary Muller C element. From the table, we find that a new value is latched at output c when the two inputs a and b are equal such that $c = a = b$. Otherwise, the ternary Muller C retains its old value.

D. Static Complementary CNTFET Logic Implementation

Static complementary logic design is a widely used design style in MOS technology because of its robustness and low

Input a	Input b	Output c
0	0	0
0	1	<i>S</i>
1	0	<i>S</i>
1	1	1

Table III: Muller C-element truth table. The *S* value indicates that the C-element holds its state (i.e. does not change value) for the corresponding pattern of inputs.

Input a	Input b	Output c
0	0	0
0	1	<i>S</i>
0	2	<i>S</i>
1	0	<i>S</i>
1	1	1
1	2	<i>S</i>
2	0	<i>S</i>
2	1	<i>S</i>
2	2	2

Table IV: Ternary Muller C-element truth table.

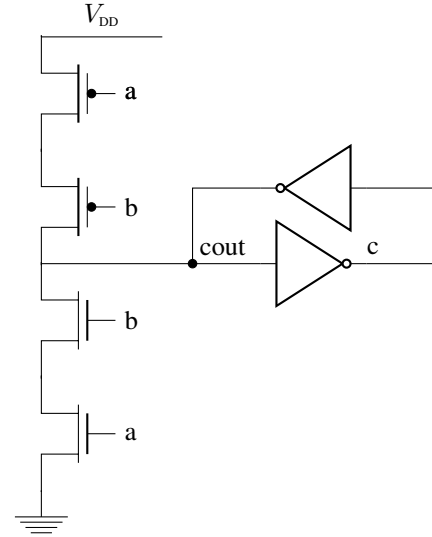


Figure 6: Binary CMOS Muller C element

power consumption. In CNTFETs also static complementary design consisting of N-CNTFETs and P-CNTFETs can be employed resulting in significant advantages. In this design, we use a combination P-CNTFETs and diode connected CNTFETs. Fig. 7 shows the circuit diagram of the Static Complementary CNTFET Muller C element. The initial portion of the Muller C consists of five P-CNTFETs and five N-CNTFETs. The inverter latch consisting of cross coupled inverters are also implemented using static complementary CNTFET design from literature [26].

The operation of the circuit is as follows. When the input voltages are below 300 mV P-CNTFET transistors T1, T2, T4 and T5 turn on thereby pulling the node cout to a logic high value. This logic value is passed on to cross coupled inverters thereby pulling the output c to a low value. When the input voltages are raised above 300 mV transistors T4, T5, T9 and T10 are off and the combination of transistors T1, T2, T7 and T8 along with the diode connected transistors T3 and T6 produce a voltage drop of 0.45 V or logic 1 at node cout. This intermediate logic value is fed to the inverter latch thereby producing a voltage drop of 0.45 V or logic 1 at the output c. As the input voltages are raised above 600 mV, transistors T4 and T5 are off and transistors T9 and T10 are on which pull the node cout to a logic low value. This low logic value is fed to the cross coupled inverters thereby driving the node output to a high value. The Transistors T1, T2, T7 and T8 have diameter of 1.487 nm. Transistors T4, T5, T9 and T10 have diameter of 0.783 nm. Diode connected transistors T3 and T6 have diameter of 1.018 nm.

E. CNTFET Muller C Element with different back gate biased voltages

Fig. 8 shows the circuit diagram of CNTFET based Muller C element based on different back gate biasing voltages. In this implementation, two transistors are reduced compared to

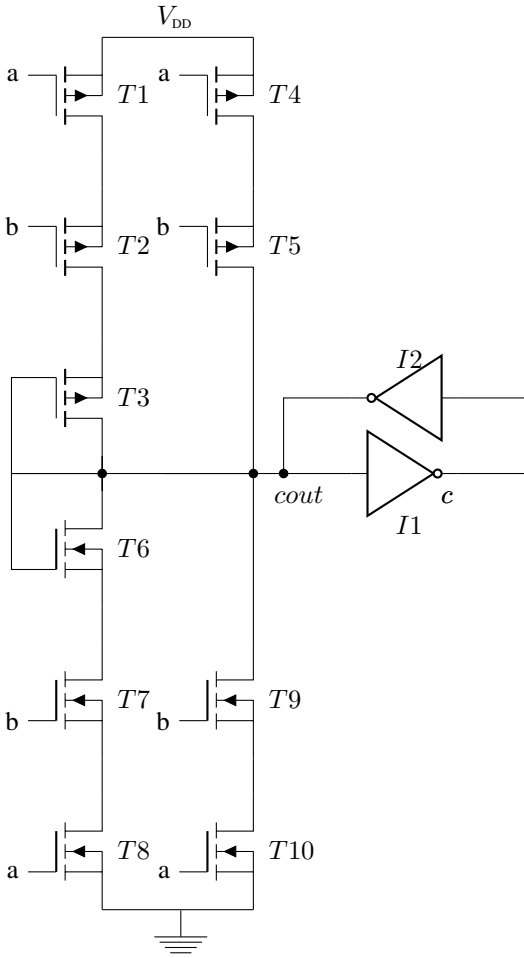


Figure 7: Muller C Circuit in Ternary Logic using Complementary CNTFET Logic

the previous static complementary CNTFET implementation. Two supply voltages are used in this implementation. The first supply voltage is 0.45V and the second supply voltage is 0.9V. When the input voltage is below 0.3V, T1, T2, T3 and T4 are on and the remaining transistors are off. A high output voltage resulting from first part is then fed to the inverter latch thereby resulting in low output voltage at the output c. The output voltage is 0V. When the input voltage is between 0.3V and 0.6V T5 and T6 are on and T1, T2 are operating in sub-threshold region. As discussed in [27] lower threshold CNTFET devices operating in sub-threshold region with forward biasing offers high drive current which is equal in measure compared to the drive current provided by higher threshold CNTFET device operating in super-threshold region. Therefore, the output is the first supply voltage VDD1 which is 0.45V. When the input voltages are above 0.6V then transistors T1, T2, T3 and T4 are off and low voltage value appears at the cout which is fed to the inverter latch resulting in high output at c.

The designs were built and evaluated using HSPICE circuit simulator. The technology node used in this paper to evaluate

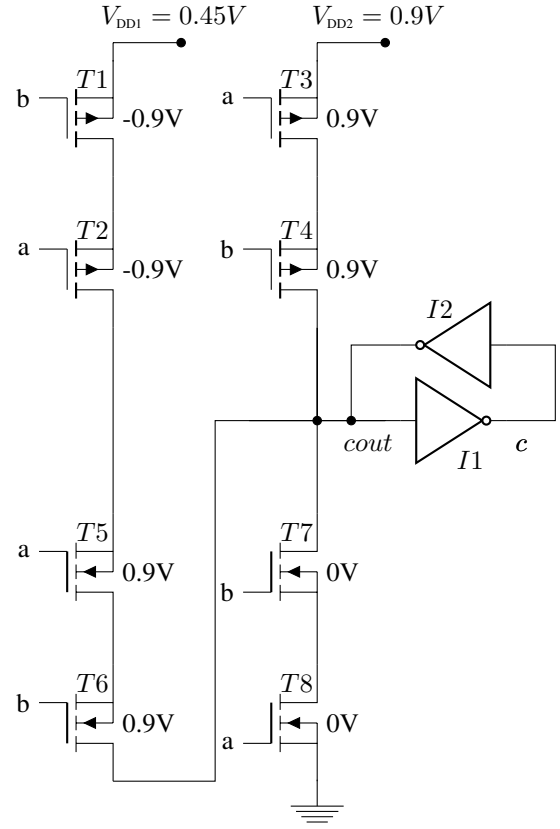


Figure 8: CNTFET Muller C Element with different back gate biased voltages

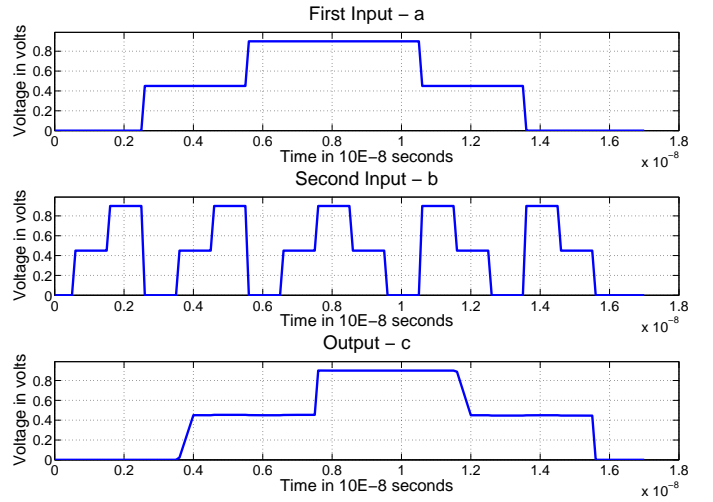


Figure 9: Transient Results Of Static Complementary Ternary CNTFET Muller C Element

the two designs was 32 nm. Stanford CNTFET device models were used to design we perform simulations in HSPICE using CNTFET parameter models by Stanford Nanoelectronics Group in the 32nm technology node [28]. All the designs were simulated at supply voltage of 0.9 V and at room temperature of $T = 27$ C.

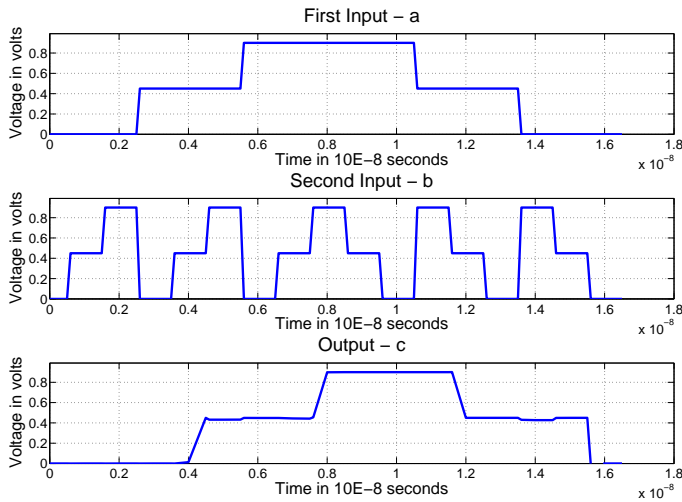


Figure 10: Transient Results Of Back-Gate biased Ternary CNTFET Muller C Element

Design	Delay(ps)	Power(uW)	PDP(E-16J)
Complementary CNTFET	120.704	3.417	4.12
Back gate biased CNTFET	359.925	1.548	5.57

Table V: Performance Comparison Of Two Proposed Designs

F. Simulation Results

In this subsection we present our simulation results. The static and the resistive load CNTFET designs were built and simulated using HSPICE. Transient simulations were run on the two designs to verify the functionality of the designs. Figure 9 and Figure 10 show the transient simulation results for Complementary CNTFET design. From the transient simulations we observe that when the first input a is held at a steady state and input b is swept from logic low to a high value the output changes state when the two inputs are at the same logic level. When the inputs a and b are at a different logic level the output holds the previous latched logic value. The inputs are swept from logic low to logic high and then changed to logic low in constant steps and the output is observed for a full low to high and a high to low transition. From the analysis, we can conclude that the ternary C element works as intended.

Table V shows the average delay, average power and the Power Delay Product (PDP) for the two proposed CNTFET ternary Muller C designs. From the table, we could see that though back gate design dissipates less power compared to the complementary CNTFET design it has a larger propagation delay which results in a higher power delay product.

IV. CONCLUSION

While electronic circuit is downsized into the nanometer scale, efforts are necessary to jointly optimize robustness and efficiency, while accounting for the probabilistic nature of nanometer devices. Among all available fault-tolerant methods to deal with transient fault occurrence, two latest works, the DMR-based LDPC-coded Fault Compensation Technique

(LFCT) and TMR-based Restorative FeedBack (RFB) technique, are first reviewed. Finally, two novel implementations of ternary Muller C-element using carbon nanotube FETS (CNTFETs) are presented for the purpose of error-resilience in multi-valued logic circuit.

ACKNOWLEDGEMENT

This work was supported by the US National Science Foundation under awards ECCS-0954747 and CCF-0916105.

REFERENCES

- [1] S. Kamal and C. Page, "Intermittent faults: A model and a detection procedure," *IEEE Transactions on Computers*, vol. 23, pp. 713–719, 1974.
- [2] F. K. E.H. Neto and G. Wirth, "Tbulk-bics: A built-in current sensor robust to process and temperature variations for soft error detection," *IEEE Transactions on Nuclear Science*, vol. 55, pp. 2281–2288, 2008.
- [3] J. Sequera and C. F. Hawkins, *MOS Electronics: How it works, how it fails*. Wiley-IEEE Press, 2004.
- [4] R. C. Baumann, "Soft errors in advanced semiconductor devices-part i: the three radiation sources," *IEEE Transactions on Device and Materials Reliability*, vol. 1, no. 1, pp. 17–22, 2001.
- [5] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Trans. VLSI Syst.*, vol. 12, no. 5, pp. 497–510, 2004.
- [6] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. VLSI Syst.*, vol. 14, no. 4, pp. 336–348, 2006.
- [7] C. Winstead and M. El Hamoui, "Reducing clock jitter by using muller-c elements," *Electronics Letters*, vol. 45, no. 3, pp. 150–151, 29 2009.
- [8] C. Winstead, "C-element multiplexing for fault-tolerant logic circuits," *Electronics Letters*, vol. 45, no. 19, pp. 969–970, 10 2009.
- [9] A. Courtay, E. Boutillon, and J. Laurent, "A convolutional code for on-chip interconnect crosstalk reduction," in *ISCAS'09*, 2009, pp. 145–148.
- [10] L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra, "Ersa: error resilient system architecture for probabilistic applications," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10, 2010, pp. 1560–1565.
- [11] Y. Tang, E. Boutillon, C. Jégo, and M. Jézéquel, "A new single-error correction scheme based on self-diagnosis residue number arithmetic," in *DASIP*, 2010, pp. 27–33.
- [12] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Dev.*, vol. 6, pp. 200–209, 1962.
- [13] C. Winstead, Y. Luo, E. Monzon, and A. Tejada, "An error correction method for binary and multiple-valued logic," *Multiple-Valued Logic, IEEE International Symposium on*, vol. 0, pp. 105–110, 2011.
- [14] C. Winstead and S. Howard, "Probabilistic LDPC-coded fault compensation technique for reliable nanoscale computing," *IEEE Transactions on Circuits and Systems II – Express Briefs*, vol. 56, no. 6, pp. 484–488, June 2009.
- [15] V. Gaudet and A. Rapley, "Iterative decoding using stochastic computation," *Electronics Letters*, vol. 39, no. 3, pp. 299–301, 2003.
- [16] A. DeHon and K. Likharev, "Hybrid cmos/nanoelectronic digital circuits: devices, architectures, and design automation," in *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, nov. 2005, pp. 375–382.
- [17] W. Rao, A. Orailoglu, and R. Karri, "Logic mapping in crossbar-based nanoarchitectures," *Design Test of Computers, IEEE*, vol. 26, no. 1, pp. 68–77, jan.-feb. 2009.
- [18] Y. Tang, C. Winstead, E. Boutillon, C. Jégo, and M. Jézéquel, "An ldpc decoding method for fault-tolerant digital logic," in *ISCAS*, 2012, p. To be published.
- [19] J. Appenzeller, "Carbon nanotubes for high-performance electronics x2014;progress and prospect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 201–211, feb. 2008.
- [20] Y. Yamamoto and M. Mukaidono, "P-functions-ternary logic functions capable of correcting input failures and suitable for treating ambiguities," *Computers, IEEE Transactions on*, vol. 41, no. 1, pp. 28–35, jan 1992.

- [21] J. Deng and H.-S. Wong, "Modeling and analysis of planar-gate electrostatic capacitance of 1-d fet with multiple cylindrical conducting channels," *Electron Devices, IEEE Transactions on*, vol. 54, no. 9, pp. 2377–2385, sept. 2007.
- [22] P. Avouris, J. Appenzeller, V. Derycke, R. Martel, and S. Wind, "Carbon nanotube electronics," in *Electron Devices Meeting, 2002. IEDM '02. International*, dec. 2002, pp. 281–284.
- [23] Y. Li, W. Kim, Y. Zhang, M. Rolandi, D. Wang, and H. Dai, "Growth of single-walled carbon nanotubes from discrete catalytic nanoparticles of various sizes," *The Journal of Physical Chemistry B*, vol. 105, no. 46, pp. 11 424–11 431, 2001. [Online]. Available: <http://pubs.acs.org/doi/abs/10.1021/jp012085b>
- [24] S. Lin, Y.-B. Kim, F. Lombardi, and Y. J. Lee, "A new sram cell design using cntfets," in *SoC Design Conference, 2008. ISOCC '08. International*, vol. 01, nov. 2008, pp. I–168 –I–171.
- [25] M. Shams, J. Ebergen, and M. Elmasry, "A comparison of cmos implementations of an asynchronous circuits primitive: the c-element," in *Low Power Electronics and Design, 1996., International Symposium on*, aug 1996, pp. 93–96.
- [26] S. Lin, Y.-B. Kim, and F. Lombardi, "A novel cntfet-based ternary logic gate design," in *Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on*, aug. 2009, pp. 435–438.
- [27] H. Nan and K. Choi, "Novel ternary logic design based on cntfet," in *SoC Design Conference (ISOCC), 2010 International*, nov. 2010, pp. 115–118.
- [28] Stanford, "Stanford university cntfet model website [online 2008]," 2008. [Online]. Available: <http://nano.stanford.edu/model.php?id=23>.