Fault-Tolerance of Robust Feed-Forward Architecture Using Single-Ended and Differential Deep-Submicron Circuits Under Massive Defect Density

Milos Stanisavljevic, Alexandre Schmid, *Member, IEEE*, and Yusuf Leblebici, *Senior Member, IEEE*

***Abstract*— An assessment of the fault-tolerance properties of single-ended and differential signaling is shown in the context of a high defect density environment, using a robust error-absorbing circuit architecture. A software tool based on Monte-Carlo simulations is used for the reliability analysis of the examined logic families. A benefit of the differential circuit over standard single-ended is shown in case of complex systems. Moreover, analysis of reliability of different circuits and discussion on the optimal granularity of redundant blocks was made.**

I. Introduction

The increased prominence of embedded systems, which are applied in safety-critical systems such as in-situ medical prosthetic microelectronic applications as well as space applications, where component maintenance is virtually out of question, emphasizes the need for increased system reliability. At the same time, modern microelectronic systems make use of advanced deep-submicron, and nanoelectronic fabrication technologies, exhibiting increased rates of defect density. The granularity of fault-tolerant "islands" must be adapted to new rates of failure densities that occur in nanometer-scale technologies. In this paper an architectural-level methodology allowing significant improvement in system reliability, which is applicable to deep-submicron as well as nanoelectronic systems, is presented. Single-ended and differential signaling circuits, built into the proposed feed-forward fault-tolerant architecture, are assessed from the perspective of their robustness to failure. A software tool based on SPICE Monte-Carlo (MC) simulations, which allows a-priori data analysis of defect-sensitive deep-submicron digital microelectronic circuits has been specifically developed.

Differential Cascode Voltage Switch (DCVS) logic [1] is a circuit technique which has potential advantages over conventional static CMOS NAND/NOR logic in terms of circuit delay, layout density, power dissipation, and logic flexibility. In this paper we demonstrate key advantages in terms of reliability of DCVS logic in comparison with standard CMOS logic, both used in a four-layer robust architecture developed for error absorption in high-density of failure environment. Moreover some insight is given into optimal redundant block sizing and design methodology.

In Section II, the applied defect modeling is described. The reliability architecture as well as averaging circuits are presented in Section III. The tool for analysis is described in Section IV. The fault tolerant properties of differential signaling, and a comparative analysis of obtained results are shown in Sections V and VI respectively. Finally, a discussion of optimal granularity and design methodology is presented in Sections VII and VIII.

The authors are with the Microelectronic Systems Laboratory (LSM), Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland (email: {milos.stanisavljevic, alexandre.schmid, yusuf.leblebici}@epfl.ch).

II. Defect Modeling and Analysis Tool

A major step in any design automation process consists of simulation. In order to perform a simulation for reliability, an accurate fault model of physical defects and fault modes with a netlist fault description is necessary. The Stuck-At approach which is traditionally used in fault coverage is not sufficient to handle the analysis of various faults in nanometer-scale devices. There are two basic approaches in device fault modeling: Inductive Fault Analysis (IFA) [2] and transistor level fault modeling [3], both of which are proven to be complex problems.

The transistor-level fault modeling is applied at an abstraction level above physical layout. It usually incorporates only stuck-on, stuck-off models of transistors for representing faults. These models represent a very reduced set of possible physical defects and therefore they are not sufficient. On the other hand, the IFA approach has some drawbacks, mainly high computational complexity of used tools, complete dependency on geometrical characteristics and difficulty of handling properly analog layouts.

A three-level hierarchical fault modeling is proposed in this paper in order to overcome shortfalls of transistor level fault modeling using some results of IFA approach and also to cover as wide as possible range of impacts that device faults have on the circuit behavior (Figure 1). The first level consists of transistor model parameters (e.g. threshold voltage V_{th}, oxide thickness t_{ox}, geometric parameters L, W) whose process-dependent variation have a significant influence on the dynamic behavior and can lead to "dynamic" faults, or violation of design time constraints. Here, each parameter can be represented by its distribution function $f_i(\ldots)$ and a nominal mean value.

In the second level, models for various physical defects such as missing spot, unwanted spot, Gate Oxide Short (GOS) with channel, floating gate coupled to a conductor, and bridging faults are adopted [4], [5]. These models have been developed from structural and lithography defects, and each defect model is described in terms of electrical parameters of its components. Thus, for simulation purposes,

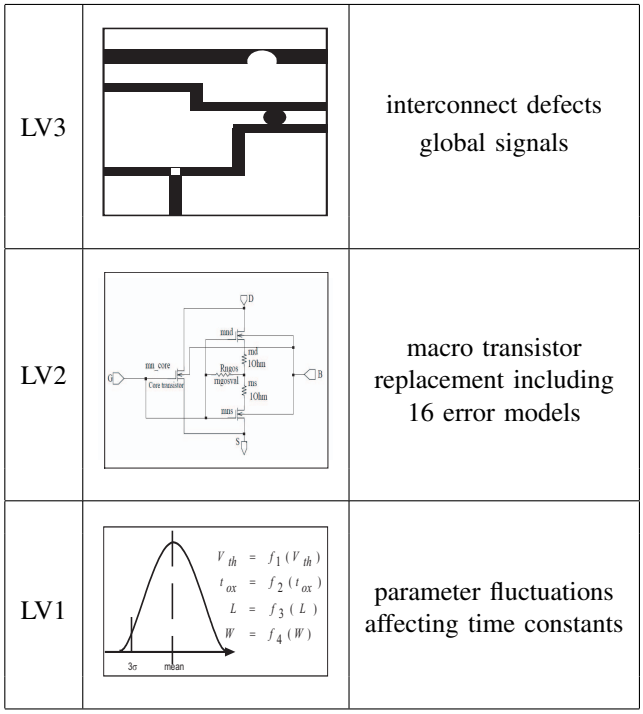


Fig. 1. Three layer fault modeling.

TABLE I

LIST OF LOW-LEVEL FAILURES MODELED IN LEVEL LV2.

Acronym	Failure Type
DHO	**D**rain **H**ard **O**pen, resulting in stuck-off fault
DSO	**D**rain **S**oft **O**pen, resulting in partial stuck-off fault
SHO	**S**ource **H**ard **O**pen, resulting in stuck-off fault
SSO	**S**ource **S**oft **O**pen, resulting in partial stuck-off fault
FLG	**FL**oating **G**ate resulting in disconnected input
DSHS	**D**rain **S**ource **H**ard **S**hort, resulting in stuck-on fault
DSSS	**D**rain **S**ource **S**oft **S**hort, resulting in partial stuck-on fault
DGHS	**D**rain **G**ate **H**ard **S**hort, resulting in input-output bridging fault
DGSS	**D**rain **G**ate **S**oft **S**hort, resulting in partial input-output bridging fault
GSHS	**G**ate **S**ource **H**ard **S**hort, resulting in input stuck-at fault
GSSS	**G**ate **S**ource **S**oft **S**hort, resulting in partial input stuck-at fault
DBHS	**D**rain **B**ulk **H**ard **S**hort, resulting in excessive current flowing through the substrate
DBSS	**D**rain **B**ulk **S**oft **S**hort, resulting in partial excessive current flowing through the substrate
GOS	**G**ate **O**xyde **S**hort, resulting in an excessive current flowing through the gate oxyde insulator

physical defects are translated into equivalent electrical linear devices such as resistors, capacitors, and nonlinear devices such as diodes and scaled transistors. A total of sixteen possible defects are considered for each transistor, which are listed in Table I.

The third layer of the fault model represents the mapping of interconnection defects into their electrical models (open spots and bridging faults) [6]. The actual model is highly dependent on geometrical characteristics of layout, where maintaining correspondence between physical and electrical parameters remains as a problem that needs to be solved. In the transistor level simulations this layer can be excluded, considering that more than 80% [7] of signal errors in modern circuits are due to global signals stuck-at supply or ground.

III. RELIABILITY ASSESSMENT APPROACH AND AVERAGING CIRCUITS

The fault-tolerance property of the proposed redundant four-layer feed-forward architecture has been applied previously to the case of single Boolean gates [8]. An array arrangement has been proposed to fabricate a multiple-input NOR slice, with fault-absorption capability [9]. Single-ended and differential circuits realizing the critical third layer are proposed in this Section.

The fault-tolerant architecture depicted in Figure 2 consists of four layers in which the data is strictly processed in a feed-forward manner. The first layer is denoted as the input layer, accepting conventional Boolean (binary) signal levels. The core operation is performed in the second layer, which consists of a number of identical, redundant units implementing the desired logic function. Fault immunity increases with the number of redundant units, yet the operation is quite different from the classical majority-based redundancy. The third layer receives the outputs of the redundant logic units in the second layer, creating a weighted average with re-scaling. Note that the output of the third layer becomes a multiple-valued logic level. Finally, the fourth layer is the decision layer where a binary output value is extracted using a variable threshold.

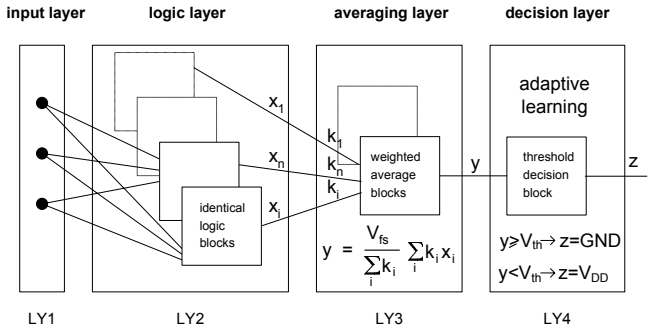


Fig. 2. Four-layer, feed-forward fault-tolerant architecture, with adaptive final decision stage.

If we measure the output of the third layer for all input values we can the construct circuit's transfer function surface. The acceptance condition for a transfer function surface to be considered as operating correctly, despite of any errors in the circuit, can be limited to critical intervals dictated by the input noise margin of the next stage, as later depicted in Figure 3. The condition for accepting or rejecting the transfer function surface is dictated by the possibility to place a threshold value V_{th} and its tolerance interval in a way that permits a correct separation of Logic *1* and Logic *0* outputs.

The variable threshold is clearly necessary in order to select the appropriate threshold level. Also, a method allowing the auto-adjustment of the threshold voltage is highly

desirable. Incorporating adjustment mechanisms into every fault-tolerant Boolean gate would require a large amount of extra hardware. Possible ways to explore include local malfunction detection, and report to a central control unit, which selectively applies learning algorithms inspired from artificial neural network theory to adapt the threshold and restore correct operation.

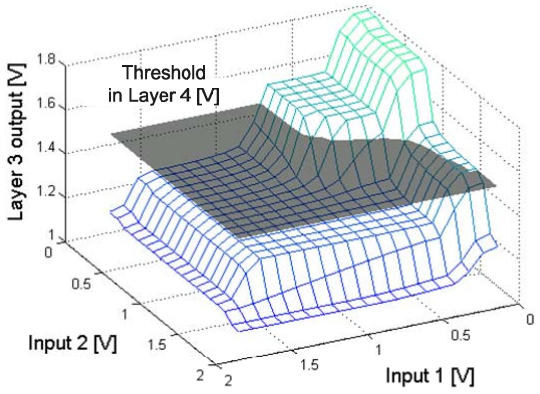


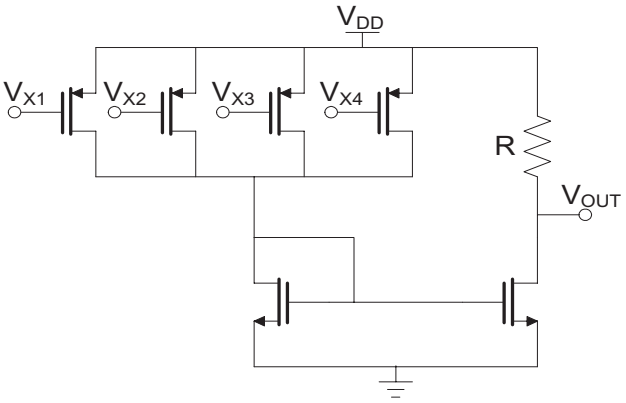
Fig. 3. Faulty transfer function surface showing significant distortion at the output of LY3 and correctly set threshold.

Fixed-weight, single-ended and differential realization of the third-layer which execute weighted average and rescaling of their inputs are shown in Figure 4. The circuits have been designed to limit the circuit area. Generally speaking, the single-ended realization should be selected to conform to area constraints, whereas the differential realization should be selected where higher linearity of the transfer function is demanded. This will be discussed in detail in Section VI.

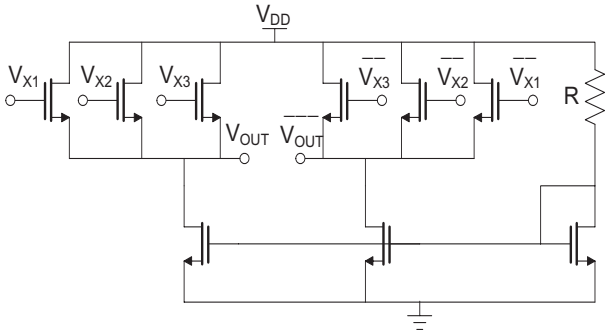
IV. Statistical Analysis Tool

The proposed redundancy scheme (explained in Section II) does not allow to extract a simple reliability rule, such as a majority rule applied in Triple Modular Redundancy (TMR) systems. In our case, every system state corresponds to an individual combination of transistor states that manifest themselves as degenerated DC transfer function surfaces, some of which still operate correctly. In order to perform analytical assessment of reliability of four-layer architecture it is necessary to extract a rule-set which describes the combination of transistor states that allow correct circuit operation. For each state, knowing probabilities of the transistors to be in that state allows to calculate the probability of correct system operation as a sum of products of probabilities. The rule-set should be disjunctive and provide full coverage for all cases of correct circuit operation.

The described method is very complex. Every constituting element of the block, such as a MOS transistor, can be in a number of states dictated by the fault occurring. Calling ε the number of faults, and n the number of transistors under consideration, the total number of system states is given as $(\varepsilon + 1)^n$. For a full statistical coverage it is possible to consider a limited number of cases, given that the redundancy in the logic layer does cause a number of cases to appear as



(a) CMOS realization



(b) DCVS realization

Fig. 4. Averaging layer LY3.

identical in their DC transfer function, and also taking into account that faults are not totally statistically independent. Nevertheless, the actual number of states is exponentially dependent on the number of transistors. Moreover, the extraction of a rule-set that describes correct functioning turns out to be intricate considering that the rule-set should cover the whole state space and that the number of rules is also exponentially dependent on the number of transistors. Above all, the task of mapping rules into actual probabilities is trivial.

In this case, the use of Monte Carlo based approach offers a relevant solution. The Monte Carlo approach implies state sampling. In this technique, a subset of states (sample) is randomly chosen from the set of all possible states. The states in a sample are simulated and the ratio of states where correct operation can be extracted (working states) over all states in the sample is used as an estimate of reliability in the complete state set. The accuracy (or error bound) of the estimated coverage depends on the absolute number of states in the sample. This number is known as *sample size* (in our case the necessary number of Monte Carlo iterations). The error bound of the estimate can be reduced by increasing the sample size. The total number of states, N_p, is called the *population size*. We want to determine the population fraction R that represents working states and randomly collect a sample of N_s states (sample size $= N_s$). If r is a random variable representing probability of correct operation and x is an estimated value of R determined by Monte Carlo

simulation, than the number of ways to obtain sample states N_w is given in Equation 1.

$$N_w = \binom{RN_p}{xN_s}\binom{(1-R)N_p}{(1-x)N_s} \tag{1}$$

The probability of a state sample giving a value x for the random variable r is given in Equation 2.

$$p(x) = \text{Prob}(r = x) = \frac{\binom{RN_p}{xN_s}\binom{(1-R)N_p}{(1-x)N_s}}{\binom{N_p}{N_s}} \tag{2}$$

This represents the *hypergeometric probability density function* of a discrete-valued random variable r. When N_s is large, r can be treated as a continuous variable and Equation 2 is conventionally approximated by a *Gaussian probability density function* with mean, $\varepsilon(r) = R$, and variance σ^2 as expressed in Equation 3.

$$p(x) = \text{Prob}(x \leq r \leq x + dx) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-R)^2}{2\sigma^2}} \tag{3}$$

Here R represents the true probability of correct operation, as the mean (or an unbiased estimate) of r. The variance of r is given according to [10] in Equation 4

$$\sigma^2 = \frac{R(1-R)}{N_s}(1 - \frac{N_s}{N_p}) \approx \frac{R(1-R)}{N_s} \tag{4}$$

The actual probability of x being within the 3-sigma range is given in Equation 5.

$$|x - R| = 3\sqrt{\frac{R(1-R)}{N_s}} \tag{5}$$

For example for $R = 0.5$ we need only 1000 Monte Carlo iterations ($N_s = 1000$) to guarantee an error smaller than 1.5%. This makes the Monte Carlo based approach very suitable.

A software tool using MC, SPICE and MATLAB has been developed in order to automate the reliability analysis process. First, a netlist is acquired from the appropriate schematic acquisition tool. Then, in each of the applied MC iterations, a faulty pattern is generated. Standard BSIM3 models of the transistors that are affected by faults are replaced with the appropriate fault models according to the error model described in Section II. A multi-variable DC sweep analysis for the acquired circuit netlist is then executed, thus forming transfer function surfaces for the considered block under failure. Subsequent Monte Carlo iterations are run applying different failure patterns, and performing sweep analysis in the transistor fault probability space. In a next step, simulation results are processed and discrimination of the transfer function surfaces where proper operation can possibly be recovered is performed. The related probability of correct operation with respect to the probability of fault of a single transistor can be calculated in the next stage. In the final step, this probability for a unit block is compared to additional trials in order to select an appropriate redundancy factor, a circuit topology, and a transistor size.

The acceptance condition for a transfer function surface to be considered as correctly operating, despite of any errors in the circuit, can be limited to critical intervals dictated by the input noise margin of the next stage.

The described steps are shown in Figure 5.

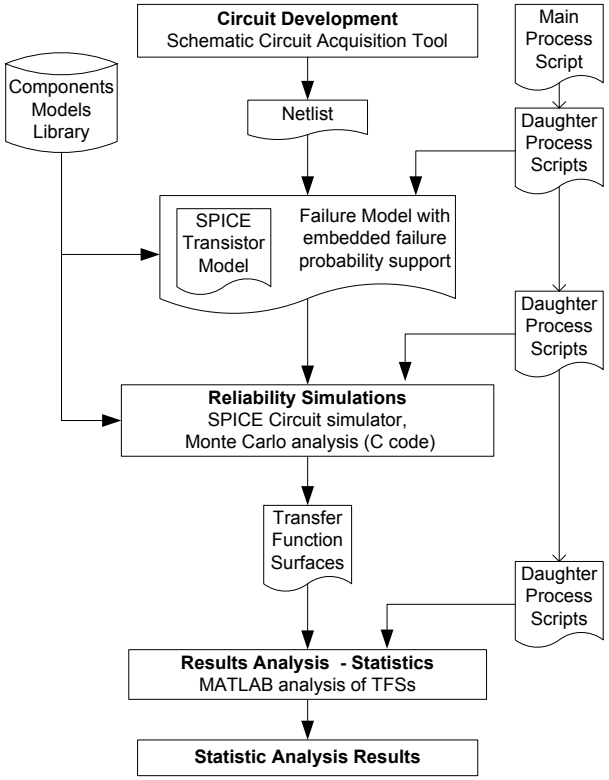


Fig. 5. Synthetic flow-graph of the tool for the a-priori analysis of reliability.

Fault distribution models adapted for nanometric technologies require monitoring the failure behavior of actual devices in mass production. Also, the computational load shows an exponential dependency with the number of input variables as well as faulty states, using conventional models.

However, in the case of the Monte Carlo based approach computational load is exponentially dependent only on the number of input variables, and not on the number of faulty states and fault modeling parameters. Moreover, faulty states and fault modeling parameters have a limited impact on single iteration time in order of logarithmic proportion.

This is an additional advantage of the Monte Carlo based approach over any analytical approach. The total time for simulations to be run is expressed in Equation 6.

$$T_{sim} = N_{sp}^{(N_{var}-1)} N_{it} N_{prob} T_{it} \quad \text{with} \quad T_{it} \approx N_{var}\log(\varepsilon) \tag{6}$$

Here, N_{sp} is the number of sweep points for each variable, N_{var} the number of input variables, N_{it} the number of MC iterations, N_{prob} the number of probability iterations, T_{it} is the time of one iteration and ε the number of different simulated fault states. Results of this reliability analysis approach will be shown in Section VI.

V. Fault-Tolerant Properties of Differential Signaling Circuits

In differential circuits, information is processed and transmitted in a redundant and complementary way, intrinsically offering an increased resistance to failures. In case of failure, the correct output signal can still be recovered, if i) the complement signal is available, and ii) the circuitry which can decode this state is available.

The logic decision is based on the possibility to define a decision interval centered around a threshold value $[V_{th} - \Delta V_{th},\ V_{th} + \Delta V_{th}]$ separating the complementary output line voltage values representing the Logic High and Logic Low. The exact values of V_{th} and ΔV_{th} are dictated by the circuit construction of the next layer input stage. Nevertheless, some conclusions with significant practical impact can be drawn out of the theoretical rationale.

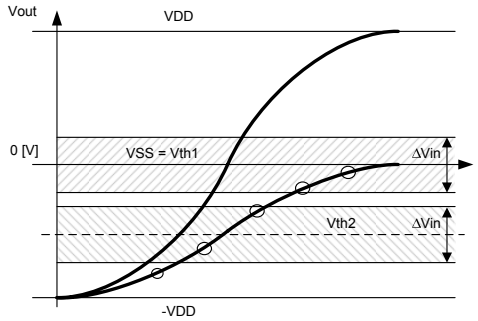


Fig. 6. Effect of stuck-at errors on the transfer function, and corresponding adaptive value of V_{th}: 1) failure free (plain line, V_{th1}), 2) stuck-at-zero (circled line, V_{th2}).

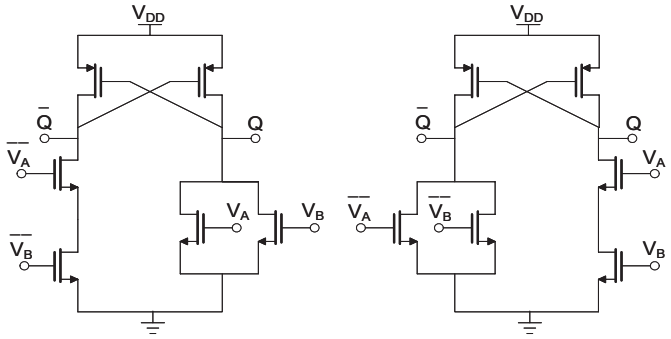
The impact of a stuck-at fault is perceived on the transfer function surface as a compression of the analog output range, as depicted in Figure 6. Consequently, a variable decision threshold V_{th} is mandatory in order to handle all possible combinations of failure distribution. Let the output of layer three (LY3) be complementary lines a and b. Calling the values on the lines a and b corresponding to Logic *1* and Logic *0*, a_1, b_1 and a_0, b_0 respectively, the value of V_{th} which is appropriate to handle errors is the arithmetical average of differential signals, taking actual values into account, i.e. any stuck signal is assigned its actual stuck voltage value, and is expressed as:

$$V_{th} = \frac{(a_1 - b_1) + (a_0 - b_0)}{2}$$

A thresholding circuit complying with this property is applied in the four-layer architecture and used in the analysis in the following Section.

VI. Comparative Analysis of Obtained Results

Different types of circuits (basic gates, Boolean, as well as more complex ones as full adders) and different circuit topologies (standard CMOS Logic, static DCVS-Differential Cascode Voltage Switch Logic) with various redundancy factors have been extensively analyzed, using the presented software tool. The reliability figure has been defined as the probability of correct operation with respect to the probability of failure of each transistor. Some basic DCVS gates are depicted in Figure 7.



(a) NOR2/OR2 (b) NAND2/AND2

Fig. 7. DCVS realization of Boolean gates.

The analysis of CMOS and DCVS logic realizations of a 2-input NOR gate is depicted in Figure 8. In both topologies, blocks for averaging and thresholding (decision) were not affected by induced faults (this condition is referred in the following as *fault-free averaging circuit*). No significant difference in circuit reliability can be observed under the aforementioned conditions. The reliability of the NOR2 function remains very close to 1 even for device failure rates exceeding 10%.

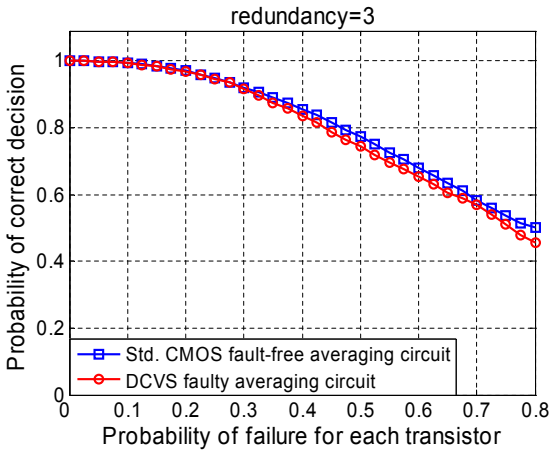
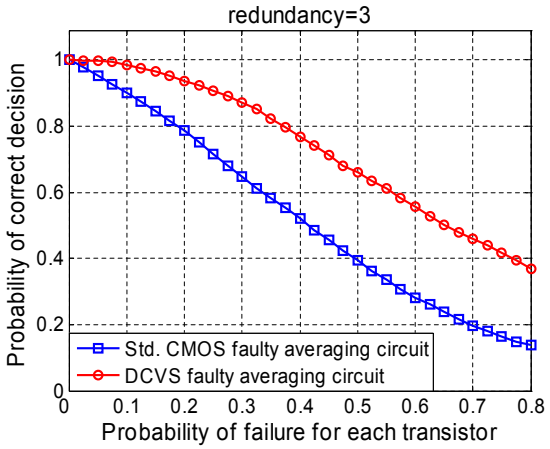
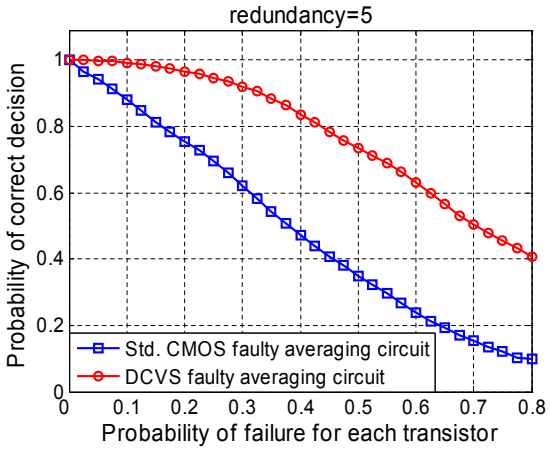


Fig. 8. Comparative analysis of the 2-input NOR gate in DCVS and standard CMOS logic with fault-free averaging circuit.

If faults are induced in the averaging circuit to represent more realistic conditions, however, the overall system reliability drops more rapidly, and the curve is not showing saturation for low fault density. Instead, it becomes quasi-linear in the critical working range, i.e. for transistor failure probability lower than 0.3, as seen in Figure 9(a) and (b) for different redundancy factors. In this case, the use of a differential averaging circuit shows improved reliability in comparison to standard CMOS, with significant difference in case of a larger redundancy factor. Moreover, there is no improvement in reliability with respect to redundancy for standard CMOS, which is due to the averaging circuit.

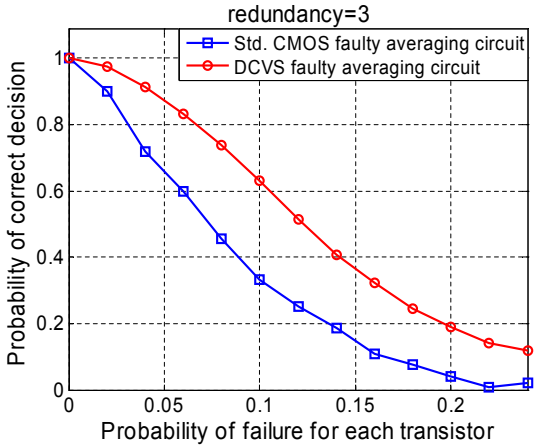


(a) Redundancy factor R=3.

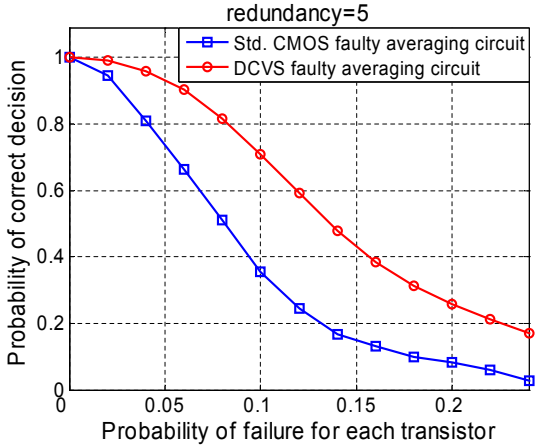


(b) Redundancy factor R=5.

Fig. 9. Comparative analysis of the 2-input NOR gate in DCVS and standard CMOS logic with faulty averaging circuit for a redundancy of R=3 and R=5.



(a) Redundancy factor R=3.



(b) Redundancy factor R=5.

Fig. 10. Comparative analysis of the 4-input gate realizing a function $f(x1, x2, x3, x4) = x1x4' + (x2x3)' + x1(x2x3)' + x1'x2x3x4$ in DCVS and standard CMOS logic with faulty averaging circuit for a redundancy of R=3 and R=5.

It can be seen that an overall reliability of >99% can be maintained with the DCVS solution under device failure rates of up to 10%, while the reliability of the CMOS solution drops rapidly below 90% for the same device failure rate.

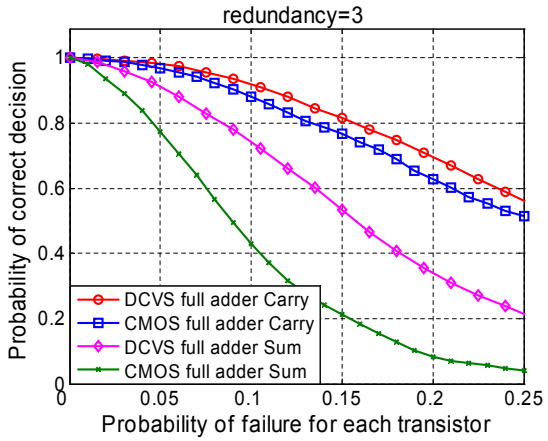
When larger blocks are used as redundant units, the probability of correct operation is reduced with respect to the size of a block. This is depicted in Figures 10 and 11 where a gate realizing the complex 4-input function, described in figure caption, and a full adder cell are used respectively as redundant blocks. Nevertheless, a clear advantage of DCVS logic realization in comparison with standard CMOS is observed in both cases. In the 4-input function case there is an improvement in reliability with respect to increase of redundancy for both configurations (Figures 10(a) and (b)). In the full adder case advantage of differential configuration is emphasized in the *Sum* signal output, whose path has an increased logic depth and complexity in case of standard CMOS when compared to DCVS. The benefits of the differential architecture are even more obvious in case where faults are induced into the averaging circuit (Figure 11(b)), when compared to fault-free averaging circuit (Figure 11(a)).

The simulation conditions and results, reported in Table II, demonstrate the extensive use of computational resources. Simulation time grows exponentially with the number of input variables, which is however dedicated to library development, and as such does not affect the end user simulations in phase two described in Section VIII. Multiprocessor systems, which optimally support parallel operation of the Monte-Carlo algorithm can be used to limit simulation time.

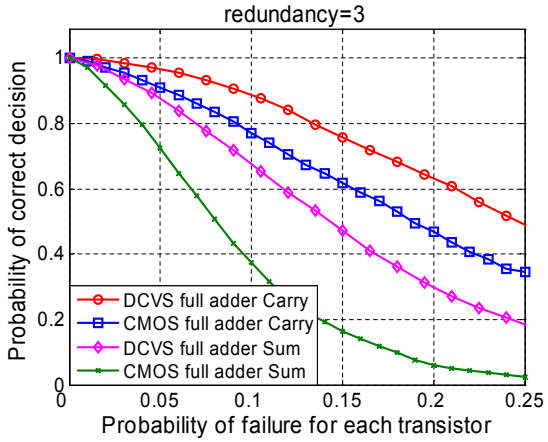
VII. Discussion on Optimal Granularity of Redundant Blocks

The granularity at which the cell size is to be considered must be adapted to new rates of failure densities that occur in nanometer-scale technologies. Besides acquiring information related to the probability of correct operation of a block, the optimal size of a block is also an important factor in design methodology for unreliable architectures. Insight in optimal sizing of redundant blocks is provided in following.

Due to the fact that all analysis are performed at transistor level, optimization is only possible according to the transistor count and not block area. However, this is not a drawback taking into account that this information should be used for



(a) Fault-free model of averaging circuit.



(b) Faulty model of averaging circuit.

Fig. 11. Comparative analysis of the full adder block in DCVS and standard CMOS logic for a redundancy of R=3 in case of fault-free and faulty averaging circuit models.

choosing an optimal subset of library of reliable components for synthesis in a given technology as explained in the Section VIII.

The cost function C_F for the optimal size analysis in case of a given defect density is chosen as:

$$C_F = P_{cor}/F_{OH}$$

where P_{cor} is a probability of correct operation for a given defect density acquired using the tool and F_{OH} is the normalized overhead function given by:

$$F_{OH} = NT_{tot}/NT_{block}$$

with NT_{tot} representing the total number of transistors in a cell realized using four-layer architecture and NT_{block} representing the number of transistors in a single redundant unit. The analysis is performed for various circuits (NOR2, NAND2, NOR3, NAND3, 1b FA, 2b FA, 3b FA, 4b FA, XOR2, MUX42, ISCAS-C17, AOI21), in case of redundancy factor R=3, in order to get broad coverage. Results are depicted in Figure 12.

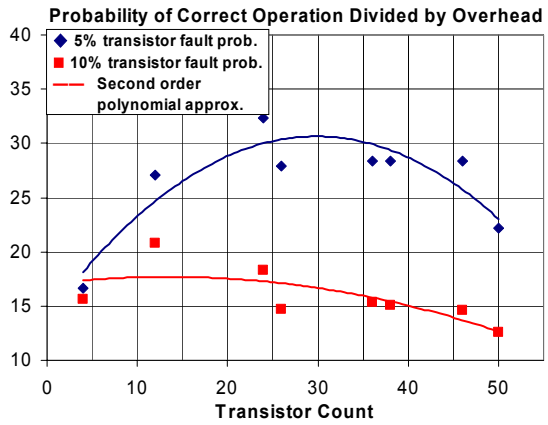
The optimal number of transistors in a block NT_{opt} is found as the block size for which the cost function reaches its maximum. Despite the fact that various circuits have been used, an optimal block size can be observed as a local maximum, and the second-order polynomial approximation of the curve provides satisfying correlation between the points. For a defect density of 5%, the optimal block size for standard CMOS cells is equal to 30 transistors, whereas it is limited to only 15 in the case of 10% defect density. The reliability figure is improving using DCVS circuits, as it can be observed on Figure 12(b).

TABLE II
SIMULATION CONDITIONS

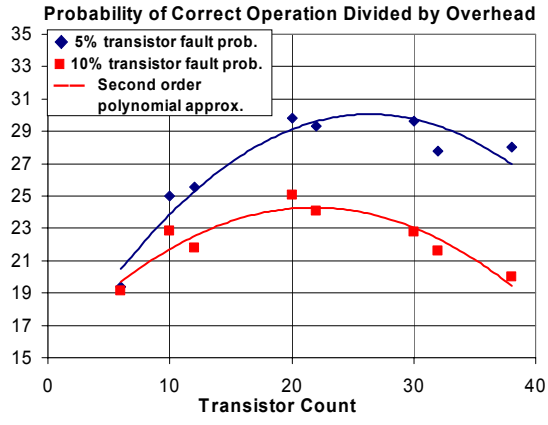
system	ULTRASPARC III+ 900MHz 16GB RAM
software	Solaris 9, Cadence 5.0.32 SPECTRE
technology	UMC 0.18 μm logic, 1.8V, 1P6M

Boolean NOR: CMOS vs. DCVS for R=3,5						
R	16 sweep pts		49 sweep pts		transistors count	
	CMOS	DCVS	CMOS	DCVS	CMOS	DCVS
3	72h	85h	85h	100h	17	27
5	80h	95h	100h	118h	22	43

4-bit ripple-carry full-adder: CMOS vs. DCVS for R=3				
R = 3	1024 sweep pts		transistors count	
input variables per cell (4) = 5	CMOS	DCVS	CMOS	DCVS
	180h	165h	392	336



(a) CMOS realization



(b) DCVS realization

Fig. 12. Optimal block size for a given technology

An increase in the defect density apparently demands a smaller optimal block size. An excessively small optimal size

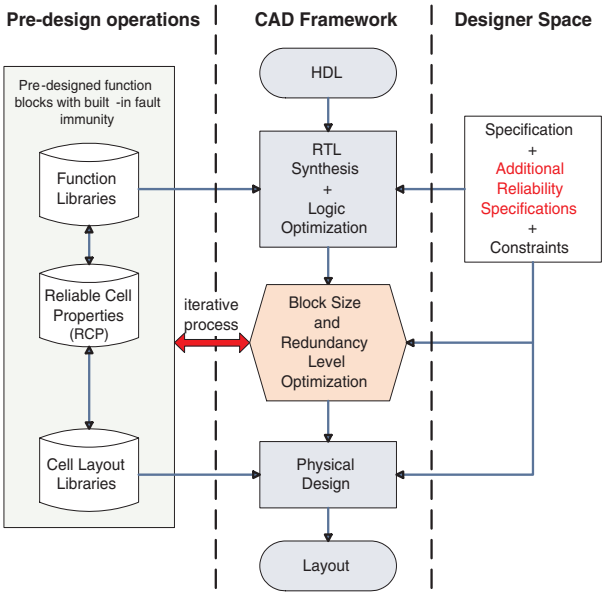


Fig. 13. Schematic flow-graph of the proposed design-flow

(less then 10 transistors) is the indication that the applied reliability architecture is not appropriate to support the given defect density with the required fault tolerance. Moreover, the small size of the optimum block that is observed verifies the correctness of the software tool addressing the analysis of circuits of a size up to approximately 100 transistors.

VIII. Design Reliability Exploration Framework

In order to provide the end-user digital IC designer with a tool allowing the exploration of the reliability design space, an adaptation of the standard design-flow applied nowadays is proposed according to Figure 13, which is supported by a command line tool that is used in two distinctive phases.

The software reliability assessment tool is used in a first phase to develop fault-tolerant standard cells forming libraries of blocks with various levels of immunity to failure. Every standard cell is described in a number of schematic representations, each using a different architecture (redundancy factor R), design style, and variable parameters (transistor sizes). After the analysis, results are attached to every developed cell as an extended cell model parameter (RCP in Figure 13). In a second phase currently under development, the end-user designer makes use of the pre-developed cells and combines them according to their attached performance. Intensive reliability simulations are no longer required at this stage, allowing the designer to experiment various scenarios. Hence, we extend the concept of *reliability by construction* to the selection of the optimal architecture.

IX. Conclusions

A circuit architecture inspired by the feed-forward artificial neural network model is presented in order to improve reliability of very-deep submicron and future nanoelectronic circuits which are expected to be fabricated in technologies prone to exhibit high density of failures. The impact of using such failure prone fabrication technologies on the reliability of different circuits and circuit architectures has been explored using an a-priori assessment software tool, with the goal of constructing and delivering libraries of fault-tolerant cells. The proposed error-absorbing, four-layer feed-forward architecture has been demonstrated to perform better using a differential averaging circuit. Moreover, the DCVS logic shows benefits in comparison with standard CMOS logic in all cases where complex gates or cells (such as a full adders) are used, demonstrating the benefits of extending differential signaling to the development of digital cells. Clearly, bio-inspired systems and methodologies offer original solutions to acute reliability issues related to emerging and future fabrication technologies.

Acknowledgment

The authors acknowledge the support of the Swiss National Science Foundation under grant 200021-112291, and the EPFL Centre SI project NANO-PLA.

References

[1] L. G. Heller, W. R. Griffin, J. W. Davis, and N. G. Thoma, "Cascode voltage switch logic: A differential CMOS logic family," *Proc. ISSCC Dig. Tech. Papers*, pp. 16–17, 1984

[2] J. P. Shen, W. Maly, F. G. Ferguson, "Inductive fault analysis of MOS Integrated Circuits," *Proc. IEEE Design and Test of Computers*, 2(6), pp. 13–26, 1985

[3] T. Olbrich, J. Perez, I. Grout, A. Richardson, C. Ferrer, "Defect-Oriented Vs Schematic-Level Based Fault Simulation for Mixed-Signal ICs," *Proc. International Test Conference*, pp. 511, 1996

[4] D. Al-Khalili, S. Adham, C. Rozon, M. Hossain, D. Racz, "Comprehensive defect analysis and defect coverage of CMOS circuits," *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 84–92, 1998

[5] R. Rodriguez-Montanes et al., "Current vs. logic testing of gate oxide short, floating gate short and bridging failures in CMOS," *Proc. International Test Conference*, pp. 510–519, 1991

[6] T. M. Storey and W. Maly, "CMOS bridging faults detection," *Proc. International Test Conference*, pp. 1123–1131, 1991

[7] V. Krishnaswamy, A. B. Ma, P. Vishakantaiah, "A study of bridging defect probabilities on a Pentium (TM) 4 CPU," *Proc. International Test Conference*, pp. 688–695, 2001

[8] A. Schmid and Y. Leblebici, "Robust circuit and system design methodologies for nanometer-scale devices and single-electron transistors," *Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 11, pp. 1156–1166, 2004

[9] A. Schmid and Y. Leblebici, "Regular array of nanometer-scale devices performing logic operations with fault-tolerant capability," *Proc. IEEE Conference on Nanotechnology (IEEE-NANO)*, pp. 399–401, 2004

[10] V. D. Agrawal, "Sampling techniques for Determining Fault Coverage in LSI Circuits," *Journal of Digital Systems*, vol. V, no. 3, pp. 189–202, 1981