Design and Realization of a Fault-Tolerant 90nm CMOS Cryptographic Engine Capable of Performing under Massive Defect Density

Milos Stanisavljevic, Frank Kagan Gürkaynak, Alexandre Schmid, Yusuf Leblebici
Microelectronic Systems Laboratory LSM, Station 11
Swiss Federal Institute of Technology EPFL
CH – 1015 Lausanne Switzerland

Maria Gabrani
IBM Zurich Research Laboratory
Säumerstrasse 4
CH-8803 Rüschlikon Switzerland

ABSTRACT
This paper presents a new approach for assessing the reliability of nanometer-scale devices prior to fabrication and a practical reliability architecture realization. A four-layer architecture exhibiting a large immunity to permanent as well as random failures is used. Characteristics of the averaging/thresholding layer are emphasized. A complete tool based on Monte Carlo simulation for a-priori functional fault tolerance analysis was used for analysis of distinctive cases and topologies. A full chip CMOS integrated design of the 128-bit AES cryptography algorithm with multiple cores that incorporate reliability architectures is shown.

Categories and Subject Descriptors
B.8.1 [**Hardware**]: Performance and Reliability – *Reliability, Testing, and Fault-Tolerance.*

General terms
Reliability.

Keywords
Fault-tolerant architecture, high defect density, reliability of submicron and nanoelectronic systems.

1. INTRODUCTION
Modern microelectronic systems are suffering from increased variability in the parameters of the fabrication technology, which may disrupt correct operation of single devices both in time and space in a random way. Still, CMOS-based systems are manufactured with a consented lowered yield. Several nanoelectronic device types have been fabricated and successfully measured. Their fabrication process however remains at a very low scale. Faulty behavior of electronic devices has been under consideration for a long time, and has been mostly handled by manufacturing test of all systems aiming at detecting and turning down all defective parts. This approach becomes intractable in presence of massive defect density, where every manufactured system is suffering from several physical defects, or unsystematic variation of vital parameters. Fault-tolerant computing has offered solutions at different abstraction levels of the integration to address this problem. For example, triple redundancy (TMR) with majority voting has been successfully applied in industrial applications, mostly considering a fairly large definition of the system to be replicated (computer, or large parts of microprocessors) [1]. However, dramatically different and new approaches may be needed to properly address the demands of safety-critical systems subject to massive defect density [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GLSVLSI'07, March 11–13, 2007, Stresa-Lago Maggiore, Italy.
Copyright 2007 ACM 978-1-59593-605-9/07/0003... $5.00.

2. PREVIOUS WORK
A four-layer fault-tolerant hardware architecture 4LRA (Figure 1) is used in order to offer a solution to the previously presented reliability issues [3]. The proposed 4LRA has been applied at the gate, or extended gate level. It can be applied hierarchically in a bottom-up way, and combined with other high-level fault absorption techniques. The details of this architecture have already been presented by the authors in earlier publications [3], [4]. Adaptable thresholding is necessary to adapt the 4LRA to the actual faulty transfer function surface as illustrated in Figures 8 and 9 in [3]. All third layer voltage levels above threshold saturate the output to V_{dd}, whereas all signal levels under threshold saturate the output to ground. Static errors can be recovered applying the proposed circuit architecture. This work has extended to the study of delay faults which can also be recovered using the proposed 4LRA.

A complete tool for a-priori functional fault tolerance analysis was developed by authors to support the development of fault-tolerant libraries of standard cells. It is a statistical Monte Carlo based tool that induces different failure models, and does subsequent evaluation of system reliability under realistic constraints. The benefit of differential circuit architectures over standard single-ended circuit architectures has been analyzed in the case of complex systems using the proposed tool and methodology.

Moreover, the analysis of reliability of different circuits has been undertaken starting from the simple NOR Boolean gate as depicted in Figure 2, where Monte Carlo (SPICE) analysis have been applied using the developed tool ([4], [5]). Two implementations of the circuits have been selected, and comparatively analyzed. Single-ended standard CMOS design is compared to differential cascode voltage switch logic (DCVS) which has been selected as a differential circuit technique. The analysis range depicted in Figure 2 is excessively large. Typically,

an maximal input range of 30% of probability of failure for each transistor is already a significant value. More complex circuits have been analyzed [5]. Finally, an analysis of the optimal granularity of the redundant blocks is presented in [5].

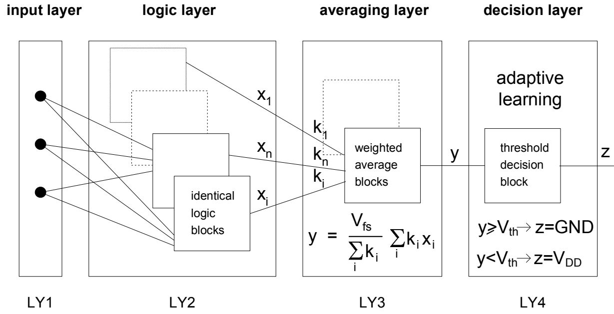


Figure 1. – Four-layer reliable architecture (4LRA)

3. DESIGN METHODOLOGY AND SAMPLE DESIGN

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 is necessary, which is supported by a command line tool that is used in two distinctive phases. The proposed flow is depicted in Figure 3. 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 declined in a number of schematic representations, each using a different architecture (redundancy factor *R*), design style, and variable parameters. In a second phase the end-user designer makes use of the pre-developed cells and combines them according to their attached performance. Hence, we extend the concept of reliability by construction to the selection of the optimal architecture.

The goal of the proposed design-flow lies its intrinsic transparency of the design complexity related to increasing reliability to the desired level. The designer perceives the development flow as similar to currently applied design-flows. The standard cell library is enhanced by the Averaging/Thresholding Cell (ATC). Several different drive strength versions of the ATC cell are designed and characterized. The design is synthesized in a standard fashion, and the resulting netlist is modified using a custom Perl script. After this, a standard back-end design flow is used. However, the optimization rules of the back-end design are modified to prevent logic optimization and critical path resynthesis. Another modification is ensuring that the analog bias voltage is routed as an analog signal without buffers.

4. RELIABLE ARCHITECTURE IMPLEMENTATION – A 128-BIT AES PROCESSOR CORE

A standard cell design of moderate complexity using a 90nm technology has been implemented in order to compare the efficiency of the reliable design methodology [6], [7]. The system consists of an implementation of the well-known Advanced Encryption Standard (AES) cryptographic algorithm using 128-bit keys [8]. In order to evaluate different fault tolerant architectures, the design has been partitioned. To minimize the I/O requirements external I/O interfaces have been limited to 8 bits. Separate input and output controllers handle the communication and store 128-bit copies of the plaintext, cipherkey, and the ciphertext. This pipelining allows, simultaneous encryption, and data I/O. A parallel on-the-fly key generator is used to derive the in every round keys used in AES from the cipherkey iteratively.

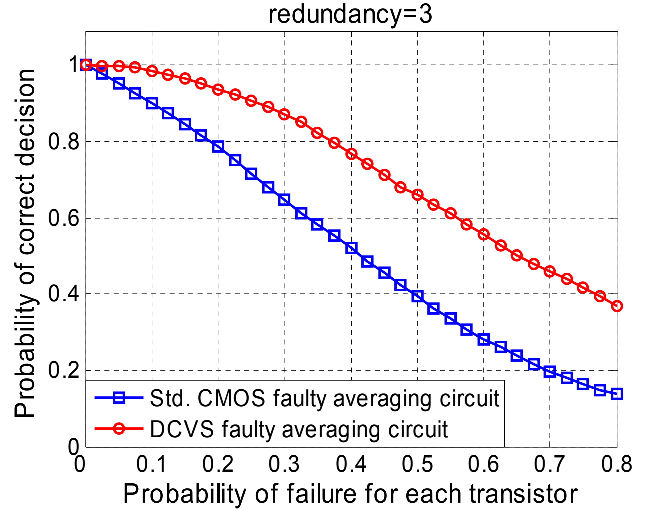


Figure 2. Comparative analysis of a 2-input NOR gate realized using DCVS and standard CMOS logic for a redundancy of three, and a faulty averaging unit.

The implementation consists of four encryption (*encrypt*) module cores, one key generation (*keygen*) module, input/output registers, and one additional module that has been integrated for the purpose of controlling induced faults.

The full development process has been inserted into an established industrial design flow, involving a number of adaptation scripts to be developed. In order to limit the number of cells to be added into the standard cell libraries, a generic averaging and thresholding circuit (ATC) with the ability to accommodate three redundant units has been designed. Consequently, the ATC can be treated by place and route tools as a regular cell.

A specific step in the development takes care of selecting circuit portions of appropriate size, and instantiating it according to the desired redundancy factor. Adding reliability feature is performed by the means of Perl scripts that take the output from the synthesizer or a preplaced netlist, and generate the final netlist for place and route tools (Figure 4(a)). Redundant units (layer two in 4LRA) are placed separately as well as the ATC (Figure 4(b)).

The ATC is realized as a standard library cell. A differential to single-ended version of the ATC has been selected as a way to keep static current below the 20μA limit that has been derived as the maximal current dissipation per ATC from the permitted overall static current dissipation, and increase linearity figures. The schematic of the implemented circuit is depicted in Figure 5(a). The cell was characterized and included in the standard design flow. The analog biasing voltage, V_{bias} is set to 300mV causing a static current dissipation of 18μA. The gate delay is equal to 280ps with a load of 6fF. Dynamic current dissipation is

equal to 8μA. Relative differential and integral linearity are both under 10% for the whole input range. The layout of the ATC is depicted in Figure 5(b). The unbalanced aspect ratio of the standard cell is due to the need to respect the regular height of standard cells, as well as the need to decrease the current density through the averager.

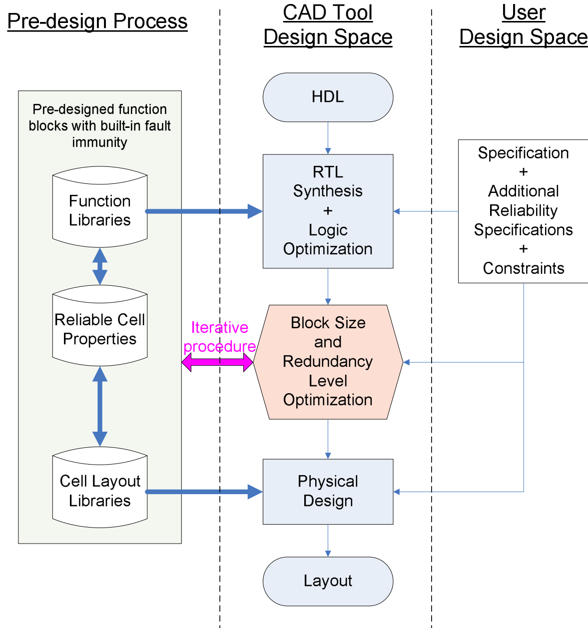


Figure 3. Schematic flow-graph of the proposed design-flow incorporating pre-designed fault-tolerant standard cells

Fault emulation has been introduced in order to provide full controllability over the faults to be injected into the AES core. The goal of the design is to demonstrate the performance of the proposed fault tolerant design methodology in a real design. Unfortunately the used 90nm design process is a commercial fabrication process and does not have the high defect density required to justify the fault tolerant design methodology. Because of this, artificial fault locations are inserted into the system at random locations. The basic circuit of the encryption core contains more than 5000 gates. To be able to simulate a wide variety of fault distributions it was decided to add 4000 fault locations.

Fault insertion is performed after all the netlists have been modified. For the redundant architectures 4000 fault locations are determined with uniform spatial distribution. Two types of faults are supported. In a hard-0 fault, the node is driven to logic-0 by the addition of a AND gate whose second input is used to control the insertion of a fault. A hard-1 fault can be inserted by adding an OR gate in a similar manner. To control all 4000 faults, a shift register with 4000 locations is used. This organization allows any combination of 4000 faults to be active at any given time. Practical experiments will be made with 20 to 100 simultaneous faults.

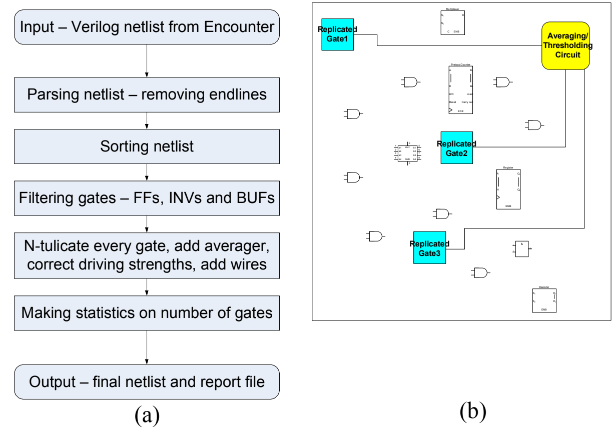
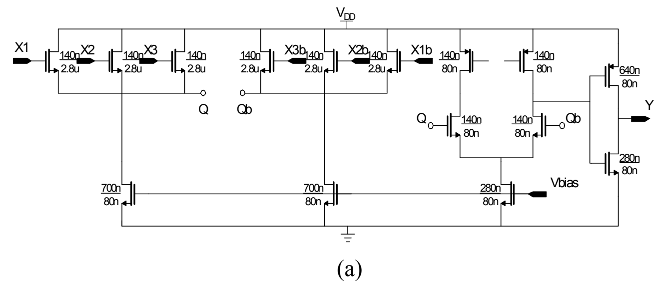
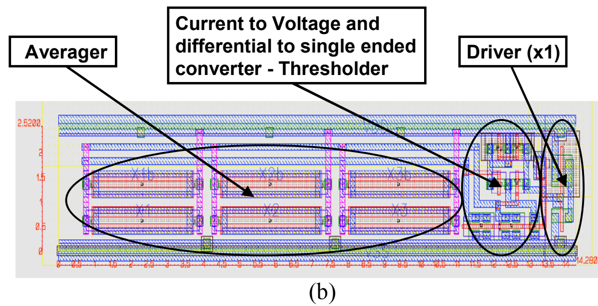


Figure 4. (a) Design-flow and (b) conceptual view of the corresponding triplication and averaging/thresholding connection schemes, where the ATC is conceived as a standalone standard cell library component.



(a)



(b)

Figure 5. (a) Schematic and (b) layout of the ATC.

The manufactured chip supports four separate incarnations of the encryption core. The first core is a straightforward implementation of the encryption (Vanilla). It has been optimized to run at 250 MHz and is able to reach a throughput of 2.9 Gb/s. However, the 8-bit I/O interface limits the throughput to 2.0 Gb/s. The second core uses the well-known majority voter approach (MAJ). Each gate in the netlist is triplicated and a simple two-out-of-three majority voter is added. This new netlist is generated automatically from the synthesized netlist. For the majority voter a standard cell that implements the functions $Y = AB + CD + EF$ is used, where A and C, B and E, as well as C and F are connected together. The third core implements the proposed multi-level fault tolerant architecture (RBS1). In this version a simplified model is used, where the logic layer consists of a single logic gate. Similar to the majority voter circuit the synthesized netlist is automatically

modified. First each gate is triplicated, then the specialized ATC cell is instantiated. The last core is once again a standard implementation (AESF). However, it also includes the artificial emulated fault locations. Note that, for each gate in the original netlist there are three gates in the redundant architecture. Therefore in some cases multiple faults in the redundant architecture map to a single fault in this architecture. In total, the redundant architectures have 4000 fault locations, while this core has only 2668.

The critical component in this design is the ATC cell. Our first version of the ATC was designed as a proof of principle and was designed as a standard cell that is compatible to the rest of the standard cell library and was not optimized for speed and area. As a result there are noticeable delay penalties for the core that uses the ATC cell. The maximum operating frequency is limited to 66 MHz. The cell also has a significant area overhead, which is an implication of the need to decrease the power dissipation of the current-based cell. A redesign of the ATC is envisioned to significantly reduce both the delay and the area.

Integrating figure results are summarized in Table 1.

Table 1. Integration result figures.

Module	N. of non-replicated gates	Num. of replicated gates	Area of replicated gates (um^2)	Area for ATC/MAJ (um^2)	Area for fault scan g. (um^2) (N. of gates)	Total gates	Total area(um^2)
Vanilla	5073	-	-	-	-	5073	**27095**
RBS1	1539	10602	67259	127173	14113 (4000)	19675	**213991**
MAJ	1539	10602	67259	32417	14113 (4000)	19675	**119235**
AESF	5073	-	-	-	9413 (2668)	9073	**36508**
Fault scan						4000	**70559**
Top+keyg.						9237	**66781**
Total						**66733**	**534160**

The analysis of the maximal fault densities for each integrated cores is given in Table 2.

Table 2. Overall fault rates

Module	Faults/μm^2	Faults/total gates
RBS1	0.019	0.2
MAJ	0.034	0.2
AESF	0.073	0.29

The test chip has 27 I/O pins and occupies 1.5x1.5mm total area out of which $0.7mm^2$ is core area. The total number of gate-equivalents is 67000 gates. The chip functional breakdown is presented in Figure 6(a), and the final chip layout is depicted in Figure 6(b). No restriction has been assigned to the exact position of the four cores, which spread over the full area.

The critical path delay for cores Vanilla and AESF is approximately equal to 4ns, is 6ns for MAJ, and for RBS1 is over 15ns due to a slow ATC cell implementation, that are common in the path. Over 90% of the ATC cells have x4 drive strength, which means that most of the ATC cells are driving at least three redundant gates. The ATC cell has also a significant impact on size which makes the RBS1 core approximately twice larger then the MAJ core. Extra hardware logic included for fault emulation (AND/OR gates) and additional wiring occupy 1.7% of the total useful area in each core.

Error!

Different sizes – no separation

Vanilla – yellow

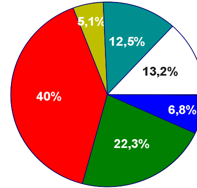
RBS1 – red

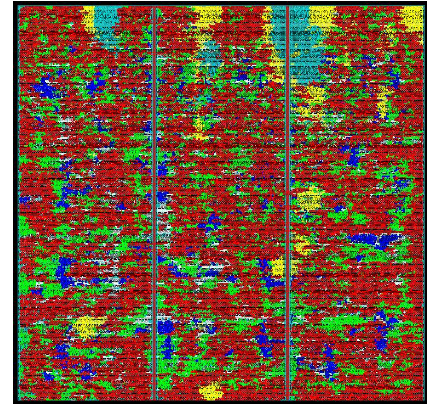
MAJ – green

AESF - blue

Fault regs. – white

Top+keygen – cyan





(a) (b)

Figure 6. (a) Final cores layout figures, and (b) layout of the integrated circuit.

5. REFERENCES

[1] J. von Neumann, Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components, *Automata Studies*, Princeton University Press, 1956.

[2] S. Roy and V. Beiu, Multiplexing Schemes for Cost-Effective Fault-Tolerance, *4th IEEE Conference on Nanotechnology (IEEE-NANO)*, pp. 589-592, Aug. 2004.

[3] Schmid and Y. Leblebici, Robust Circuit and System Design Methodologies for Nanometer-Scale Devices and Single-Electron Transistor, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 11, pp. 1156-1166, Nov. 2004.

[4] M. Stanisavljevic, A. Schmid, and Y. Leblebici, A Methodology for Reliability Enhancement of Nanometer-Scale Digital Systems Based on A-Priori Functional Fault-Tolerance Analysis, *Proc. IEEE IFIP VLSI-SoC*, October 2005.

[5] M. Stanisavljevic, A. Schmid, Y. Leblebici, Fault-Tolerance of R Publication obust Feed-Forward Architecture Using Single-Ended and Differential Deep-Submicron Circuits Under Massive Defect Density, *International Joint Conference on Neural Networks*, July 2006.

[6] Publicly available parameters for the IBM 90nm technology http://www-03.ibm.com/chips/asics/products/stdcell.html

[7] T. Schafbauer, et al., Integration of high-performance, low-leakage and mixed signal features into a 100 nm CMOS technology, 2002 Symposium on VLSI Technology, June 2002.

[8] Advanced Encryption Standard, *Federal Information Processing Standards 197* (FIPS 197), National Institute of Standards and Technology (NIST), November 2001.