INFORMER: An Integrated Framework for Early-Stage Memory Robustness Analysis

Shrikanth Ganapathy* Ramon Canal* Dan Alexandrescu† Enrico Costenaro† Antonio González‡* Antonio Rubio§
*Department d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, Spain
†Engineering Division, iRoC Technologies, Grenoble, France
‡Intel Barcelona Research Center, Intel Labs-UPC, Barcelona, Spain
§Department d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Barcelona, Spain
Email: *{sg, rcanal}@ac.upc.edu †{dan.alexandrescu, enrico.costenaro}@iroctech.com ‡antonio.gonzalez@intel.com §antonio.rubio@upc.edu

***Abstract*—With the growing importance of parametric (process and environmental) variations in advanced technologies, it has become a serious challenge to design reliable, fast and low-power embedded memories. Adopting a variation-aware design paradigm requires a holistic perspective of memory-wide metrics such as yield, power and performance. However, accurate estimation of such metrics is largely dependent on circuit implementation styles, technology parameters and architecture-level specifics.**

In this paper, we propose a fully automated tool - INFORMER - that helps high-level designers estimate memory reliability metrics rapidly and accurately. The tool relies on accurate circuit-level simulations of failure mechanisms such as soft-errors and parametric failures. The statistics obtained can then help couple low-level metrics with higher-level design choices. A new technique for rapid estimation of low-probability failure events is also proposed. We present three use-cases of our prototype tool to demonstrate its diverse capabilities in autonomously guiding large SRAM based robust memory designs.

I. Introduction

The design of embedded memories has garnered much significant importance over the last decade as it expected that they will occupy more than 70% silicon real-estate by 2017 [1]. In deep sub-micron technologies, the task of design is compounded with new challenges introduced by parametric variations of process and environmental parameters. Such variations introduce non-uniform characteristics in memory components leading to indeterministic behaviour post-manufacturing worsening the reliability metric. In this paper, we focus on the design of fault tolerant embedded memories subject to limits set on system level metrics such as energy, performance and area.

Limited by factors such as memory density and low-power consumption, embedded memories using minimum geometry six-transistor (6T) SRAM cells under the effects of parametric variability become unstable and start failing under different scenarios. Variations in the threshold voltage can reduce the stability of the cell (lower noise margins) increasing the susceptibility to a bit flip leading to a failure. Then, there are soft-errors that are caused due to energization of bitcells due to the collision of impurity induced alpha particles or neutrons generated from cosmic rays. The random nature of such failures as a rare phenomena can become quite significant in the context of memory yield. Techniques like error-correcting codes (ECC) and redundancy help lower the failure rate and improve memory yield albeit incurring area, performance and power overheads. Therefore, for any given design, while improving robustness is critical, it is also important that system-level metrics like performance is enhanced whilst lowering power consumption. In that regard, recent research has highlighted the need to analyse the global figures of merit defined by energy, yield and performance when evaluating new proposals targeting robust memory design [2]. Moreover, trends in the EDA industry have shown that reliable embedded memory design will be the biggest challenge in the next decade; and developing design flows and tools centered around it will represent the fastest growing product segment [3].

978-3-9815370-2-4/DATE14/©2014 EDAA

In this paper, we introduce a novel tool *(INFORMER: An Integrated Framework for Early-Stage Memory Robustness Analysis)*, that allows users to estimate memory reliability under the impact of parametric (soft) failures for a given SRAM cell design, technology, working environment and memory architecture. The primary contributions of the paper can be summarized as:

- A novel technique for estimating the failure probability of SRAM cells by leveraging transistor dimensions.
- INFORMER tool that provides a simple and transparent interface to estimate memory robustness. It aggregates memory specification from different levels of design-abstraction → builds a prototype memory at the circuit-level → estimates memory reliability → applies it on higher-level specifications → optimizes simultaneously multiple-objectives if needed.

INFORMER is then interfaced with another highly specialized and accurate tool - TFIT for soft-error rate analysis [4].

The main objective of the tool is to provide a stable platform for higher-level designers to evaluate the impact of soft failures and errors in embedded memories for a wide range of design specifications within a reasonable amount of time.

The remainder of the paper is organized as follows. Section II discusses failure probability estimation methodology. Section III discusses the design flow of INFORMER and its internal components. In section IV, we discuss three case studies on the diverse uses of INFORMER. Finally, in section V, the concluding remarks are presented.

II. Rapid failure estimation methodology

In order to estimate the SRAM-cell failure probability, we consider four scenarios where the cell can potentially fail :

Write Failure : This type of failure occurs when a '0' cannot be written into the node storing a 1 when the wordline is high.

Read Failure : Disturbance on the bitlines causes the cell to flip during a read operation. In other words, an accidental write operation happens during a read access. For the sake of brevity, we don't consider failures from *column half-select* disturbances prevalent in 8T-SRAM(s) like cells employing single-ended sensing scheme.

Access Failure : The cell is not able to create a bit-differential equal to sense amplifier offset in the time it is enabled. As a result, the output of the sense-amplifier is faulty.

Hold Failure : During the standby mode, when reducing the standby supply voltage beyond a certain level, the contents of the cell are destroyed.

The total failure probability $P_{failure}$ is then given by P[WF $\cup$ RF $\cup$ AF $\cup$ HF]. At nominal voltages, the failure probability is

extremely low owing to sufficient noise margins requiring exhaustive number of simulations for estimating failure probability. As a result, estimating the failure probability in minimum time requires special approximation techniques. It is possible to minimize the quantum of simulations by reducing the search space by determining *apriori* the failure region of maximum likelihood. The accuracy of measurements in such cases is largely dependent on the technique used. *Most probable failure point* analysis is one such technique that can be leveraged to improve the efficiency and accuracy of traditional monte-carlo based methods [5]. The task of failure probability estimation can be simplified into the problem of finding in the variation space the most probable point of occurrence for which the SRAM cell fails. In a regular six-transistor (6T) cell, random variations in the threshold voltage are primarily responsible for parametric failures. The number of *directions* in which the threshold variations of all six transistors can vary within the variation space is $2^6 = 64$. Previous published work assumes that for each of the four failure mechanisms, there is only one direction where failures are dominant. However, our simulations have demonstrated that the *length* and *width* of the six constituent transistors can greatly influence the failure probability as they modify read/write parameters, cell leakage and propagation delays making it harder to determine accurately the failure probability accurately when limiting the search space to just one direction. We introduce a

Algorithm 1: *Leveraged* MPFP analysis

Input: Cell Dimensions (Symmetric), σV_{th}, $P_{fail} = 0$
Output: failure probability
for *Each Failure Mechanism* **do**
  $Dim = \{L_{PU}, L_{PD}, L_{NA}, W_{PU}, W_{PD}, W_{NA}\}_{normalised}$
  $Failure_{Directions} = \bigcup_{i=1}^{|Dim|} Direction^{i}_{lookup}$, i$\in$*Dim*
end
foreach $Failure_{Direction}$ **do**
  determine *failure* and *no-failure* boundary ;
  while *Simulation number* $<$ *limit* **do**
    generate ΔV_{th_1} ... ΔV_{th_6} (within boundaries);
    run *Simulation* ;
    if *Failure criteria* == *True* **then**
      $P_{failure} = \prod_{i=1}^{6} P(\Delta V_{th_i})$
    end
  end
  $P_{fail} = P_{fail} + max(P_{failure})$
end

new technique called *leveraged* MPFP that calculates the MPFP for more than one direction based on cell dimensions. The concept of using cell dimensions to trade-off estimation accuracy for simulation speed is based on the notion that with increase in cell dimensions, improved noise margins lower failure probability. This will enable us to to tune the estimation technique so as to reduce/increase the number of required simulations dependent on cell dimensions. The basic methodology of *l* MPFP is shown in algorithm 1. For each direction (out of 64) where a failure event is likely to occur, the technique searches for the MPFP based on threshold variability. We improve the computational complexity of the search algorithm by first using a 1-D approximation by generating the ΔV_{th} values in equal normalised variations and determining the boundary between a failure event and a no-failure event. Within the region between the two boundaries, the traditional MPFP technique is executed to determine the combination of ΔV_{th} values of the six transistors that maximizes the failure probability. The accuracy of our proposed technique is largely dependent on the increments of ΔV_{th} values used to determine the two boundaries. Assuming a 3σ variation, we were able to determine the boundaries with acceptable levels of accuracy within 100 increments. As for determining the failure directions with maximum likelihood, we use data obtained *apriori* by running exhaustive simulations for a wide range of cell sizes and using a look-up table to reference based on normalised λ values of transistor dimensions.

The simulations were performed for a range of threshold deviation values for a cell designed in 22nm HP-PTM operating at 0.7V with dimensions obtained from [6], [7]. For the reminder of the paper, we will present results for the same configuration unless it is explicitly mentioned. The results are presented in figure 1. We used the HSPICE simulator for traditional monte-carlo based simulations. At low variation corners where failure probability is extremely low, both MPFP and lMPFP estimate failure probabilities that is an order of magnitude different from SPICE estimates. This is primarily because

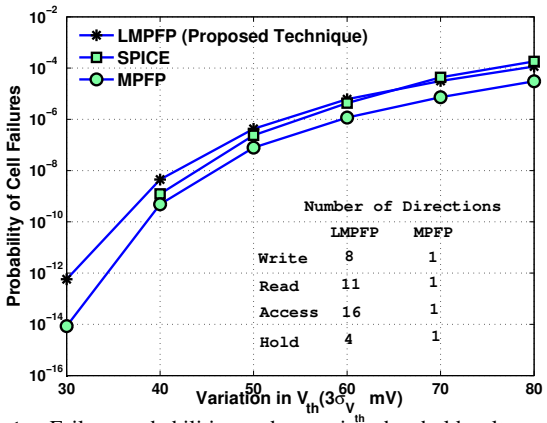

Fig. 1. Failure probabilities under varying threshold voltage deviation

both techniques derive the probability density function (PDF) of only one point in the variation space with maximum probability. However, the presence of more than one failing point with near-identical probability can reduce the effectiveness of our technique. The number of directions where failures are more probable to occur as obtained by our technique is shown within the figure. As the threshold variation increases, lMPFP is able to perform on par with traditional monte-carlo simulations while MPFP still suffers from such primitive issues. For the case of failure probability $\approx 10^{-5}$, the simulations were run for more than 3 hours in HSPICE compared to the 31 seconds it took for our technique to complete achieving 350X speed-up. It should also be noted that there are several other techniques such as importance sampling and probability collectives that build on top of MPFP to improve the accuracy of the measurement tremendously. As mentioned earlier, our technique in conjunction with the INFORMER framework is designed to help derive a trend in memory robustness when considering multiple design choices (across different levels of abstraction) simultaneously and is not designed to be a memory robustness sign-off tool.

III. Overview of INFORMER

INFORMER has been designed as a generic tool capable of guiding designs across different process nodes. It has been written in *Python* on top of SPICE. The tool supports a wide variety of input parameters describing the different memory components with varying levels of detail. The primary inputs of the configuration file include :

- Technology specific transistor models - *SPICE* libraries

- Architecture-level memory specifics - number of rows and columns/array, number of redundancy columns, total size of memory, read-out width and ECC strength
- Bitcell specifications - maximum wordline pulse-width, SA enable time, SA offset, Hold voltage
- Parametric variability information - threshold, channel-length & width variability, Temperature range
- extra parameters needed for soft-error simulations like particle flux and pulse-width of strike

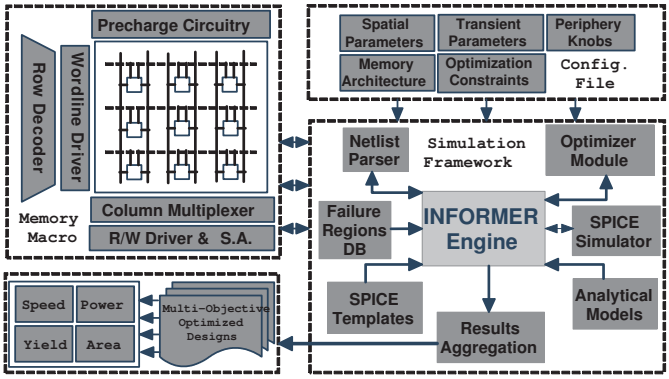

Fig. 2. INFORMER design flow.

As shown in figure 2, at the heart of the system lies the engine that is completely responsible for all tasks ranging from characterizing SRAM cells to designing whole memory macros. The simulation and statistics generation is done in two phases. In the first phase, based on user inputs, a representative memory critical path is generated in a *spice* format. Either pre-written templates or user provided netlists can be used for this purpose. The netlists generally contain all components of the memory including write buffers, column and row decoders, multiplexers, precharging circuitry and sense amplifiers. The advantage of generating an approximate critical path from individual components is the ability to port designs from one technology to another and rapidly estimate the memory-wide statistics to better understand the changes in the design. While the design of each peripheral circuitry can change, their input/output ports are fixed to ensure coherency in simulation results. Therefore, their interfaces and certain timing related specifications are static. Once the memory macro is generated, all the different failure criteria are evaluated based on input specifications. The influence of the peripheral circuitry on the failure probability is captured through the use of one or more *tuning-knobs* that control the timing and bias sources of the memory array. The *bias* knobs can typically control all voltage sources connected to the write, read and precharge lines while the *timing* knobs can limit read/write and sense amplifier enable times. We estimate the failure probabilities while tuning the design knobs like wordline pulse width, sense-amplifier offset, standby voltage and temporally varying parameter like temperature. The obtained results are presented in figure 3. The values are normalised to the minimum failure probability observed. Note that in some results the increase in failure probability is not necessarily an increase in the overall failure probability. Each failure mode exhibits different trends for different tuning parameters. We have highlighted only those failure modes that are the most affected by such design parameters.

In the second phase of simulation, the obtained estimates from the simulation are fed into black-box analytical models that provide accurate memory-wide statistics of power, yield and area overhead. We have also included an optimization layer that can take control of the simulations independently and guide design choices autonomously. For a minimum number of input constraints, it can optimize the

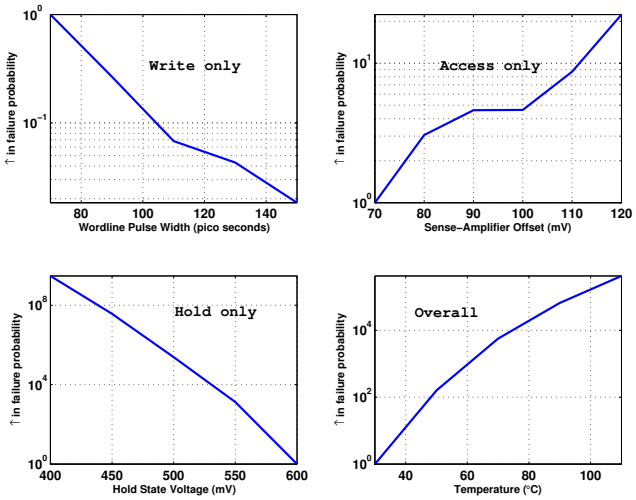

Fig. 3. Estimated increase in failure probabilities as a function of varying control knobs. The values are normalised to the lowest failure probability.

cell specifications subject to limits set on the input parameters. Another unique feature of our tool is the seamless integration with a state of the art soft-error rate estimation tool, TFIT [4]. We evaluate the tolerance of the cell to soft-errors through a technological process characterization database allowing the generation and evaluation of any transient currents that may be induced by single events. This database built from prior silicon test data is primarily a response model that tracks the effect of transient current pulses on the individual transistors of the cell. The cell response is then evaluated w.r.t. these events in a given working environment. In its current iteration, TFIT does not support *variability* specific control statements that many simulators accept. A custom wrapper-like *TFIT interface* interacts with TFIT and provides it with the necessary configuration files and a modified cell netlist with the necessary variation information injected.

All simulations have been performed on a 6-core machine running at 3.0 GHz and using 32GB of main memory. INFORMER supports multi-core processing and for each run of the input configuration, six operations (4 failure modes, soft-error, power/latency simulation) are performed in parallel to further enable rapid design space exploration.

IV. Use Cases of Informer

A. Constraint based optimization

Figure 4 shows the percentage yield as a function of the spread in leakage power and cell area under the impact of spatial variations. The analysis was performed for a lot of 729 cells with different transistor dimensions. It was observed that the 1-σ deviation of leakage and cell area is 51.2% and 5.4% respectively. From the figure, it should be clear that the yield by itself does not display any particular trend with respect to leakage and cell area. Thus the problem of cell sizing can be viewed as a non-linear optimization with a set of input constraints. We defined the constraint problem as, *minimize cell area* subject to *yield* $>$ *90%* and *std. dev. of leakage* $<$ *-1.2*σ. For this case, the optimized cell area was 3.2% lower than that of the mean close to the nominal design. When the constraint on *std. dev. of leakage* was ≤ 0, the cell was further optimized and the obtained area estimates were 8.25% lower than the mean.

B. Column redundancy limits on yield

Redundancy has been proposed as a low-cost technique to improve yield and reduce test cost by replacing defective (hard faults)

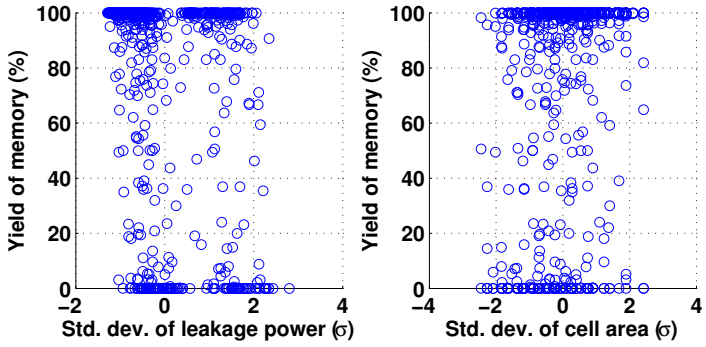

Fig. 4. Impact of cell area and power on yield

rows/columns with redundant ones. More recently, redundancy has been used to improve the functional yield at low supply voltages where the functional margin is very poor [8]. In order to achieve maximum yield through the use redundancy, it is important to know exact fault locations *apriori*. That way, redundant columns (rows) can be allocated to the column (rows) with most number of faults with significant increase in yield. Figure 5 shows the estimated yield as a

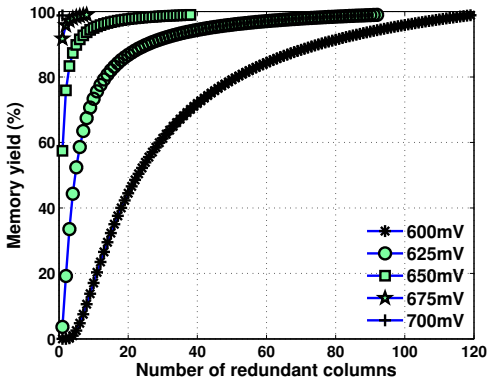

Fig. 5. Yield w.r.t. the number of redundant columns in a 256x128 Array

function of the number of redundant columns for five different cases of supply voltage. It is well known that supply voltage scaling is the most effective technique for power savings. However, beyond a point it becomes a critical reliability constraint and no longer is energy efficient due to area and power-overheads of the recovery mechanisms (redundant columns). There is approximately a 33% reduction in power when lowering supply voltage from 700mV to 600mV and the area overhead needed to ensure 90% yield is nearly 47% which makes it a non-viable option for dense embedded memories. Instead, a small overhead in the cell area can be incurred to achieve lower failure probability and still be able to operate at lower voltages.

C. Variability aware soft-error rate estimation

As the soft-error rate is exponentially dependent on the critical charge, the impact of process variations on soft-error rate is not trivial. Previous studies have highlighted that gate-length variations has the most impact on the critical-charge and can be as high as 80% [9]. We use a 45nm cell designed in conjunction with a 45nm CMOS database. The choice of technology node for this study was prompted by the lack of accurate design files for advanced nodes. Figure 6 shows the distribution of the SER (FIT/1Mb) of memory designed using 3 different cells. The cell dimensions were obtained from [8] and scaled accordingly. Normalising the area estimates of the

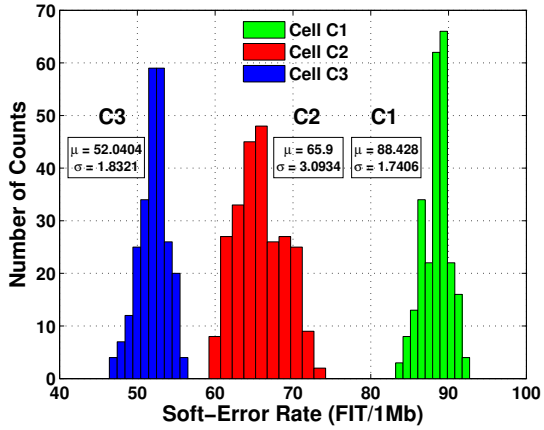

Fig. 6. Soft-error rate under process variability

different cells w.r.t. to the smallest cell, the areas of the intermediate and largest are 23% and 46% higher (reported estimates were for 32nm technology). We notice here that the spread is very small unlike leakage power or access latency. This is mainly because the effects of independent variation of channel length and width on the critical charge are quite different and they eventually cancel out each other. Further, as the soft-error phenomena is observed only in the hold states, the access transistors can be completely ignored in the analysis. It was shown in [8] that the probability of joint occurrence of a soft-error and parametric failure in the same memory word is extremely low. Therefore, it is sufficient if both types of errors are treated in an orthogonal manner.

V. Conclusion

In this paper, we have proposed a new tool, INFORMER, that helps to estimate memory-wide critical statistics like power, performance and yield early in the design stage. The tool captures reliability failure mechanisms using circuit-level simulations and applies it on higher-level specifications for rapid design space exploration. Finally, we have demonstrated three different applications of the tool that can be leveraged to make intuitive design choices that ultimately lead to designing more robust memories.

VI. Acknowledgments

This work has been partially supported by the Spanish Ministry of Education and Science under grant TIN2010-18368 and TEC2008-01856, the Generalitat of Catalunya under grant 2009SGR1250 and Intel Corporation.

References

[1] S. Kaushik *et al.*, "Embedded memory test and repair optimizes soc yields." [Online]. Available: http://www.edn.com/design/manufacturing/4390489/Embedded-memory-test-and-repair-optimizes-SoC-yields-

[2] Nalam *et al.*, "Virtual prototyper (vipro): An early design space exploration and optimization tool for sram designers," in *DAC'10*.

[3] "Spice memory battle," 2012. [Online]. Available: http://www.deepchip.com/items/0502-08.html

[4] H.Belhaddad *et al.*, "Circuit simulations of seu and set disruptions by means of an empirical model built thanks to a set of 3d mixed-mode device simulation responses," in *RADECS'06*.

[5] D. Khalil *et al.*, "Sram dynamic stability estimation using mpfp and its applications."

[6] J. Kim *et al.*, "Modeling SRAM failure rates to enable fast, dense, low-power caches," in *SELSE'09*.

[7] "Predictive technology models," *http://ptm.asu.edu/*.

[8] Z. Shi-Ting *et al.*, "Minimizing total area of low-voltage SRAM arrays through joint optimization of cell size, redundancy, and ecc," in *ICCD'10*.

[9] Qian *et al.*, "Impact of process variation on soft error vulnerability for nanometer vlsi circuits," in *ASICON'05*.