Technique for Efficient Evaluation of SRAM Timing Failure

Masood Qazi, Mehul Tikekar, Lara Dolecek, Devavrat Shah, and Anantha P. Chandrakasan

***Abstract*— This brief presents a technique to evaluate the timing variation of static random access memory (SRAM). Specifically, a method called loop flattening, which reduces the evaluation of the timing statistics in the complex highly structured circuit to that of a single chain of component circuits, is justified. Then, to very quickly evaluate the timing delay of a single chain, a statistical method based on importance sampling augmented with targeted high-dimensional spherical sampling can be employed. The overall methodology has shown 650× or greater speedup over the nominal Monte Carlo approach with 10.5% accuracy in probability. Examples based on both the large-signal and small-signal SRAM read path are discussed, and a detailed comparison with state-of-the-art accelerated statistical simulation techniques is given.**

***Index Terms*— Cache memories, CMOS memory, process variation, random access memory, sense amplifier, static random access memory (SRAM).**

I. Introduction

Embedded static random access memory (SRAM) is a vital component of digital integrated circuits and often constitutes a dominant portion of the chip area [1]. Therefore, the specifications of embedded SRAM have significant implications on the overall chip cost, power, performance, and yield. Shown in Fig. 1(a) is a plot of reported cell areas in fully functional SRAM macros versus the technology node for the past few years. The cell area has scaled with the scaling of the critical feature size. Fig. 1(b) plots an unconventional metric—the number of SRAM bits per square millimeter of silicon in high-performance microprocessor chips—which reveals that reduced SRAM cell area does not readily translate into increased SRAM utilization.

This discrepancy in trends is due to a number of limitations of SRAM, all related to local variation: SRAM often needs a separate elevated power supply; excessive SRAM timing variation degrades performance; and uncertain aging effects show up first in the highly integrated and highly sensitive SRAM cells. As the overarching goal of this brief, we seek to increase the SRAM utilization by propagating the physical trend of shrinking cell area into the overall system-on-chip improvement. This goal can be achieved if designers have a way to quickly assess the impact of circuit solutions on the operating constraints (e.g., minimum VDD, frequency) to ultimately preserve the overall chip yield.

This brief focuses on read access yield because it has been observed in measurements that ac failures, manifested as too slow an access time from one or more addresses, are encountered before dc failures, manifested as the corruption of data at one or more addresses [2]. Therefore, dc stability (write and read margin) is necessary but not sufficient for yielding a memory chip. A significant degree of additional margin must be inserted to meet the performance requirements.

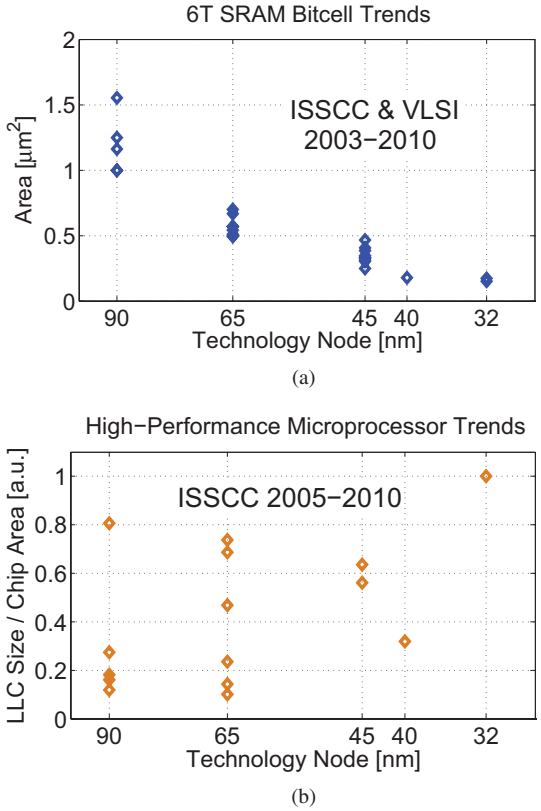

Fig. 1. (a) SRAM cell area scaling. (b) Utilization of SRAM in recent high-performance microprocessors.

In general, the exact distributions of the relevant SRAM performance metrics are not known. As a consequence, any statistical simulation method unavoidably resorts to numerical solvers such as SPICE. Classical approaches such as the Monte Carlo method require too many iterations of such SPICE evaluations because of the circuit complexity and extremely low tolerable failure probabilities of individual components (10^{-8} and below). Thus, the primary challenges to any statistical simulation method are: 1) dealing with the structural complexity of the timing delay evaluation problem and 2) estimating timing delay statistics to a very high accuracy.

A. Prior Work

A lot of exciting recent work has made important progress toward the eventual goal of designing generically applicable efficient simulation methodologies for circuit performance evaluation. To begin with, in [3]–[7], the authors developed efficient sampling-based approaches that provide significant speedup over the Monte Carlo method. However, these works do not deal with the interconnection complexity, i.e., do not address challenge 1) stated in the previous section.

Other authors have addressed the issue of structural complexity. In [8], by modeling the bitline signal and the sense amplifier offset (and the timer circuit) with Gaussian distributions, the authors proposed a linearized model for the read path. As this model can be simulated in MATLAB, the SRAM structure can be emulated and the evaluation time can be significantly improved. Additional approaches such as [9] and [10] apply more sophisticated techniques involving Gumbel

Manuscript received November 20, 2011; revised April 12, 2012; accepted June 3, 2012. Date of publication September 10, 2012; date of current version July 22, 2013. This work was supported by the C2S2 Focus Center, one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation Entity.

M. Qazi, M. Tikekar, D. Shah, and A. P. Chandrakasan are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: mqazi@mit.edu; mtikekar@mit.edu; devavrat@mit.edu; anantha@mtl.mit.edu).

L. Dolecek is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1592 USA (e-mail: dolecek@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2012.2212254

distributions and sensitivity analysis but still do not incorporate a full-scale SPICE functionality check to directly evaluate extreme delay statistics, which is generally necessary to handle all possible operating scenarios (e.g., low-voltage operation).

B. Contributions

In this brief, we show how to overcome the two challenges for the timing delay analysis of SRAM by means of two proposed methods of loop flattening and spherical importance sampling (IS), respectively. These techniques were introduced in [11], and in this brief, we add: 1) a theoretical justification of loop flattening; 2) new evidence of the loop-flattening accuracy in the large signal SRAM read path under general conditions of non-Gaussian delays, multiple levels of nested sampling, and correlated fluctuations; 3) a detailed breakdown of the simulation cost of spherical IS; and 4) a quantitative comparison with other works regarding simulation cost versus failure probability level and dimensionality.

II. Loop Flattening for Timing Variation

In this section, we describe the problem of statistically analyzing the SRAM read path which contains circuit blocks repeated at different rates. Then we introduce and justify the loop-flattening approximation to enable the application of accelerated statistical simulation techniques. In the representative block diagram of an SRAM array of Fig. 2(a), there are highly repeated structures: memory cells, sense amplifiers, row decoders and drivers, and timing circuits. There are several distinct cascaded circuit stages, some of which may be correlated. The circuit is also big. A straightforward way to simulate this circuit is to take a complete schematic and address each location in simulation while noting the behavior for each address location. This method would cost too much computational resources, so a circuit designer turns to a simulation of a critical path by taking a representative member of each group of circuits and adding appropriate parasitic loading in parallel.

A statistical analysis of a memory-critical path requires additional insight into the architecture. For now, consider a single column of 256 memory cells as in Fig. 2(b) with $R = 256$. When the wordline goes high at time $t = 0$, the memory cell develops a differential signal (voltage difference between BLT and BLC), and when the enable signal goes high at time $t = T$, the sense amplifier resolves that differential signal to a logic-level output (voltage difference between SAT and SAC). One can represent the bitcell signal of cell i as $TX_i = T(\sigma_X \tilde{X}_i + \mu_X)$ and the sense amplifier offset as $Y = \sigma_Y \tilde{Y}$ [$\tilde{Y}$ and $\tilde{X}_i$ are $\mathcal{N}(0, 1)$]. The failure of this read operation is determined by the interaction of two random variables sampled at different rates. The probability P_f that this single-column memory fails for a given strobe timing T is the probability that the sense amplifier offset overpowers the bitcell signal for one or more paths in the column

$$P_f := \Pr\left(\bigcup_{i=1}^{R} \{Y - TX_i > 0,\}\right) \tag{1}$$

$$\leq \; R \cdot \Pr(Y - TX_1 > 0) =: P_u \tag{2}$$

where P_u is the conservative union-bound estimate of P_f.

Because of the different rates of repetition, a proper Monte Carlo simulation on a critical path with one cell and one sense amplifier must sample variables in a nested for loop: for each sense amplifier, simulate over 256 cells and check for one or more failing paths, then sample a new sense amplifier and repeat over 256 new cell realizations and so on, as suggested in [8]. If one wishes to apply an accelerated statistical simulation to evaluate the failure of this circuit, the "for loop" sets an unfavorable lower bound on the number of simulations needed just to emulate the architecture.

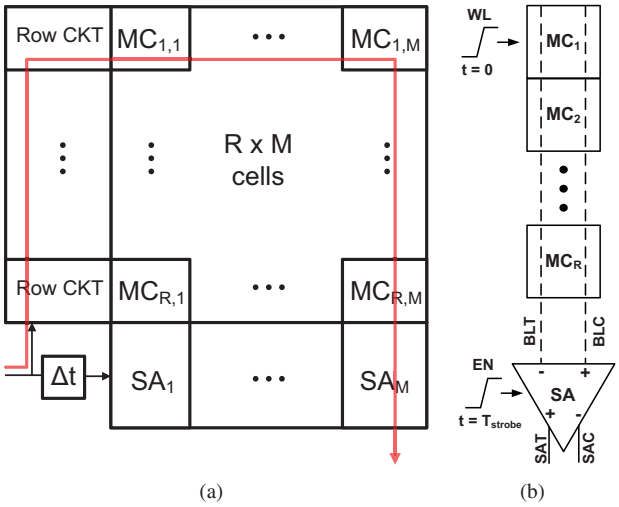

Fig. 2. (a) Representative SRAM array. (b) Simplified schematic of the small-signal read path.

We observed that the simple union-bound estimate P_u provides an accurate way to bypass this requirement. Just the path failure probability is simulated and the result is multiplied by R. The estimate is guaranteed to be conservative and in practice agrees very well at low levels of failure probability. In [11], this loop-flattening estimate was shown to be accurate for the small-signal SRAM read path. As in [8], the bitline signal development and sense amplifier offset were parametrized by Gaussian random variables—$\{\mu_X = 1$ mV/ps, $\sigma_X = 0.10 \times \mu_X$, $R = 256$, $\sigma_Y = 25$ mV$\}$ for (1). Specifically, the loop-flattening estimate was only 1.9% pessimistic in strobe timing for a modestly sized 512-KB memory (2048 memory columns) at a yield of 99%, and increased in accuracy at even lower levels of failure.

The schematic in Fig. 3(a) is the schematic tree of the large signal read path. For the case of cascaded random delays, we can also see the applicability of the loop-flattening estimate. This circuit is simulated in a production-quality 45-nm CMOS technology, where each shaded transistor (or gate input) exhibits local mismatch modeled as a fluctuation of its threshold voltage. Fig. 4 shows the Monte Carlo SPICE simulation result of this circuit for 8 cells per local bitline ($N_{\mathrm{LBL}} = 8$) and 16 local evaluation networks ($N_{\mathrm{SEG}} = 16$). In this picture, there is a delay Z_i ($1 \leq i \leq 256$) associated with each of the 256 cells on the column, and the probability of failure associated with a target delay t is

$$P_f := \Pr\left(\bigcup_{i=1}^{R} \{Z_i \geq t\}\right) \tag{3}$$

with $R = 256$. The solid curve gives the conventional nested Monte Carlo simulation result by sampling random variables in proportion to the rate of repetition of their associated transistors. The dashed curve gives the loop-flattening estimate in which a simple chain of representative transistors is simulated with all random variables sampled at the same rate. Even for this example, in which the delays are not perfectly normal and delay stages are correlated, the loop-flattening technique produces a tight estimate. The single solid black dot gives a specific example for how an IS simulation with an appropriately chosen mean shift can evaluate the loop flattening (dashed curve) with significant speedup, consuming approximately 1100 SPICE simulations in this example. The loop-flattening approximation suggests that this IS estimate in turn will match the result produced by a proper Monte Carlo simulation with nested sampling,