EFFICIENT APPROACH FOR SRAM TIMING FAILURE PREDICTION

1Sindhumathi.K, 2Srilatha.K
1PG Scholar, Department of VLSI, Sathyabama University, India
2Assistant Professor, Department of VLSI, Sathyabama University, India

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. In previous works all the results are discussed in a theoretical manner. That not to be practically implemented. This paper also investigates comparison of small signal read path and large signal read path. The array structure was designed by inserting small signal read path and also large signal read path. Finally in this paper we calculate the delay by using T-spice 7.1.

Keywords: TSPICE, SRAM, memory


1 Introduction

The performance evaluation of circuit designs, SRAM in particular, is becoming an increasingly difficult task due to the process variation in deep-submicron technologies. The complex structure of SRAM—which is an assembly of multiple components (memory cells, sense amplifiers, delay chains) at different rates of repetition—makes it very challenging to estimate the effect of process variation on critical circuit properties, such as the timing delay.

In such scenarios, the analytical distribution of the performance metrics is not known. As a consequence, any statistical simulation method unavoidably resorts to numerical solvers such as SPICE. Classical approaches like the Monte Carlo methods 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: (a) dealing with the structural complexity of the timing delay evaluation problem, and (b) estimating timing delay statistics to a very high accuracy. In this paper, we shall overcome these two challenges for the timing delay analysis of SRAM by means of two proposed methods of Loop flattening and Spherical Importance Sampling, respectively.

II. Small Signal Sensing

As the main result of this paper, we put forth a novel statistical methodology to quickly and accurately evaluate timing delay in an SRAM. The two key novel aspects of our methodology are: (i) the Loop flattening method and (ii) the Spherical Importance Sampling. The loop flattening approach addresses the challenge of dealing with the added difficulty of the timing delay estimation caused by the complex structure of SRAM. We show that, somewhat surprisingly, the naive adaptation of the classical critical path methodology—in which a Monte Carlo simulation of a chain of component circuits (such as row driver, memory cell, and sense amplifier) disregards the relative rate of replication—produces an estimate that is always conservative and highly accurate. Namely, if the loop flattening based approach indicates that the delay exceeds 130 ps with probability 10−5, then the actual delay will exceed 130 ps with probability less than or equal to 10−5. More importantly, unlike the worstcase estimation, this conservative approach is increasingly accurate at lower failure levels.

Fig. 1. The conventional Monte Carlo (nested sampling) estimation and the proposed loop flattening technique are compared in a simplified read path example in MATLAB. The delays associated with the two curves for a common level of failure tightly converge with the decreasing failure rate. High accuracy is achieved at relevant (low) levels of failure.
The reduction obtained by loop flattening still leaves us with the problem of evaluating the probability density function(pdf) of a single chain of SRAM components (a single memory cell and a single sense amplifier), to a very high accuracy. This problem requires sampling a 12-parameter distribution, wherein a standard Monte Carlo is clearly not useful. The importance sampling -based approach in the recent work utilizes ‘uniform sampling,’ and as such does not scale to higher dimensionality (12 dimensions in the case of interest). To cope with the dimensionality, we use a spherical sampling based approach that samples space in a cleverly designed, adaptive manner. The combination of these two methods, loop flattening and Spherical Importance Sampling, yields an efficient and an accurate approach for estimating the timing delay in SRAM.

As a representative application of our method, consider results plotted in Fig. 1. Given a specific setup (1mv/ps bitline discharge with standard deviation of 10%, 20mV of sense amplifier offset, and 256 cells per bitline), the curves of sense amplifier timing (X-axis) versus the probability (Y-axis) of incorrect sense amplifier output are shown for: 1) exhaustive Monte Carlo (solid red), 2) the worst-case estimation (dashed gray) and 3) our approach (dashed blue).

We make the following observations based on this figure. First, our estimation is always conservative, as expected. Second, it becomes highly accurate for low probability of error, which is the regime of interest. For example, for the probability of error of 10−5 of a single memory column, the timing delay prediction of our method is off by 1.9% compared to the Monte Carlo approach.

This corresponds to the 99% yield of a modest 512kb memory. Third, the worst-case estimation—in which the weakest memory cell must overcome the largest sense amplifier offset—is far too conservative for this probability of error. Even in MATLAB, the exhaustive Monte Carlo simulation took over six hours; whereas, the loop flattened curve was obtained instantaneously as it arises from the sum of two Gaussians. In the experimental results that follow, our approach exhibits 650X or greater speedup over a Monte Carlo approach, while still using the more generally applicable SPICE solver.

### III. Loop Flattening

Let Sk be the worst-case signal in the kth memory column. Let S = min1≤k≤M Sk. It is reasonable to model each Sk as independent and identically distributed. Therefore, to evaluate the quantity of interest, we have

\[ F(T) = Pr(S \leq 0) = 1 - Pr(S > 0) = 1 - Pr(S > 0)M \]

We may focus on evaluating Pr (S1 ≤ 0) for a single memory column. As shown in Fig. 2, let Xi be the rate of signal development by the ith memory cell of the column, and let Y be the sense amplifier offset. Then,

\[ S_1 = \min_{1 \leq i \leq R} Z_i, \]

Where \( Z_i = T \cdot X_i - Y. \) Therefore, using the standard union bound

\[ F_1(T) = Pr(S1 \leq 0) = Pr(\cup_{i=1}^{R} \{Z_i \leq 0\}) \leq R \cdot Pr(Z_1 \leq 0) = R \cdot f_1(T), \]

Where \( f_1(T) = Pr(Z_1 \leq 0). \) Putting the above into the context of M independent memory columns,

\[ F(T) \leq 1 - (1 - Rf_1(T))^M. \tag{1} \]

Further, for T large enough, i.e., for \( f_1(T) \) small enough (which is indeed the case of interest), \( Rf_1(T) \) is small, and we obtain an approximation (using \( 1 - x \approx e^{-x} \) for small \( x)\)

\[ F(T) \leq 1 - e^{-MRf_1(T)} \approx MRf_1(T). \tag{2} \]

Equation (2) suggests that \( F(T) \approx MRf_1(T). \) We call this the loop flattening approximation—it is conservative as the above derivation shows. It can be shown to be asymptotically correct, i.e.,

\[ \lim_{T \to \infty} \frac{1}{T} \log \frac{F(T)}{MRf_1(T)} = 0, \tag{3} \]
When \( X_i, Y \) have reasonable distributions such as Gaussian. Intuitively, the approximation (2) says that a large delay primarily happens due to only one of the cells, that is, multiple cells are unlikely to simultaneously induce a large delay. In Fig. 1 \( R = 256 \), \( X_i \) is a Gaussian with the mean of \( 1 \text{mV/ps} \) and relative standard deviation of \( 10\% \), and \( Y_i \) is a zero-mean Gaussian with \( 20 \text{mV} \) of standard deviation. The evaluation of these parameters show that the convergence of (3) is very tight at practical strobe timings.

**IV. Spherical Importance Sampling**

Equation (2) suggests that we need to evaluate \( f_1(T) = \Pr(Z_1 \leq 0) \) for \( T \geq 0 \). To do so, we propose an importance sampling based approach with spherical sampling. We quickly recall the basics of the importance sampling in the context of our setup. Variable \( Z_1 = T \). \( X_1 - Y \) is determined by the 12 parameters, say \( A_l, 1 \leq l \leq 12 \), that correspond to the random threshold voltage variation of transistors in the critical path (drawn in dark lines) of Fig. 3.

![Fig. 3. Transistor level schematic of a representative memory column with 128 cells per bitline and additional multiplexing of 4 columns per sense amplifier. Precharge devices are not shown for clarity.](image)

**Step 1. Perform Spherical Search**

Given the initial tolerable failure floor \( p_{\text{floor}} \) from a user (e.g., \( 10^{-12} \)) initialize the algorithm parameters: \( R_{\text{high}} = 2\varphi^{-1}(1-p_{\text{floor}}) \), \( R_{\text{low}} = 0 \), and \( N_{\text{iter}} = 0 \). \( \varphi(\cdot) \) is the normal CDF. For the allocated complexity parameter \( N_1 \), set \( L \) and \( U \) as the lower and upper bounds on the target number of fails. A good design choice is \( U = 0.5N_1 \) and \( L = 1 \). (a) While \( M_f \in [L, U] \):

- Set \( R_1 := 1/2(R_{\text{low}} + R_{\text{high}}) \)
- Sample the radius-\( R_1 \) spherical shell \( N_1 \) times, and record the total number of failures, \( M_f \).
- If \( M_f > U \), set \( R_{\text{high}} := R_1 \)
- If \( M_f < L \), set \( R_{\text{low}} := R_1 \)
- \( N_{\text{iter}} := N_{\text{iter}} + 1 \).

(b) Record the final number of iterations as \( B_1 = N_{\text{iter}} \). Average over the direction vectors associated with all the failing vectors in the last iteration in step (a). Initialize \( s_1 \) with this average as the starting mean-shift for Step 2. Record the quadratic norm of this shift as the current minimum norm, \( \min_{\text{Norm}} = \text{norm}(s_1) \).

**Step 2. Perform Local Exploration.**

Let \( N_2 \) be the allocated complexity of this step. Set \( \text{runCount} = 0 \), initialize \( R_2 = R_1/2 \), and \( a = (0.05/R_2) 1/N_2 \). The latter parameter governs how the resolution of local exploration will gradually try to zoom in. While \( \text{runCount} \leq N_2 \):

- Sample uniformly from a spherical distribution of radius \( R_2 \), around the point \( s_1 \). Call this point \( s_x \).
  - If \( \text{norm}(s_x) < \min_{\text{Norm}} \),
    - Set \( \text{runCount} := \text{runCount} + 1 \).
    - Run a SPICE simulation with \( s_x \) as its input. If the simulation results in a failure, record the displacement \( d = \text{norm}(s_1 - s_x) \) and then update the mean shift vector : \( s_1 := s_x \). Otherwise \( d = 0 \).
    - Geometrically shrink \( R_2 \), while factoring in the displacement:
      \( R_2 := aR_2 + (1 - a)d \)
Fig. 4 Illustration of Spherical Importance Sampling Algorithm. In Step 1, samples on spherical shells quickly find a good starting direction enabling the local exploration in Step 2 to gravitate to the mean shift of minimum quadratic norm.

**Step 3. Run The Off-The-Shelf Importance Sampler.**

This step is done as per (4) with \( s = s_1 \). Record \( N_3 \) as the number of steps it takes the estimator to reach the FOM value of 0.1, corresponding to the accuracy and confidence of 90% each.

**V. Simulation Result**

The schematic in Fig. 5 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 5 schematic tree of large signal read path**

The following Fig 6 explains the simulated waveform of large signal read path as the existing system.

**Fig 6 Simulation results of existing system**

**Fig 7 schematic tree of large signal read path using master slave**

The following Fig 7 explains the proposed system simulated waveform.
VI. Comparison Table

Table 1: comparison of proposed and existing system delays

<table>
<thead>
<tr>
<th></th>
<th>SSRP</th>
<th>LSRP (LATCH)</th>
<th>LSRP (MS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1BIT SRAM</td>
<td>5.08</td>
<td>4.33</td>
<td>2.67</td>
</tr>
<tr>
<td>2BIT SRAM</td>
<td>8.20</td>
<td>7.12</td>
<td>6.71</td>
</tr>
<tr>
<td>4BIT SRAM</td>
<td>10.76</td>
<td>9.88</td>
<td>8.33</td>
</tr>
</tbody>
</table>

VII. Conclusion

As a contrast to prior SRAM yield analysis, the consideration of intermediate metrics (bitline signal, sense amplifier offset) has been replaced by a full-scale SPICE simulation requiring only the indication of pass or fail. As future work, this method can be extended to the complete, global row and column path of large embedded SRAM, in addition to other highly structured circuits such as adders, FIR filters, and FFT accelerators.

References