Empirical Performance Models for 3T1D Memories

Kristen Lovin, Benjamin Lee, Xiaoyao Liang, David Brooks, Gu-Yeon Wei

TR-03-08

Computer Science Group
Harvard University
Cambridge, Massachusetts

Empirical Performance Models for 3T1D Memories

Kristen Lovin[1], Benjamin Lee[2], Xiaoyao Liang[1], David Brooks[1], Gu-Yeon Wei[1]
[1] School of Engineering and Applied Sciences, Harvard University
[2] Computer Architecture Group, Microsoft Research

Harvard University Technical Report TR-03-08
3 October 2008

***Abstract*—Process variation poses a significant threat to the performance and reliability of the 6T SRAM cell. In response, research has turned to new memory cell models, such as the 3T1D DRAM cell, as potential replacement designs. If designers are to seriously consider this new design, performance models are needed to better understand the behavior of this cell. We propose a decoupled approach for collecting Monte Carlo HSPICE data, reducing simulation times by simulating memory array components separately based on their contribution to the worst-case read critical path. We use this Monte Carlo data to train regression models, which accurately predict retention and access times of a 3T1D memory array with a median error of 7.39%.**

I. Introduction

For decades, technology scaling has boosted performance and increased density in integrated circuits. However, shrinking device feature sizes mean that process variation has become a significant hindrance, reducing reliability and limiting performance gains from technology scaling. With process variation, traditional memory circuit designs need revisiting.

In particular, process variation directly attacks the weakenesses of 6T SRAM producing transistors that deviate from their specified sizes, thereby causing device mismatches. This reduces reliability and adversely affects performance. In addition to device mismatch, process variation limits 6T performance scalability by causing variation in the operating speed of individual cells and memory lines. Thus, the 6T cell is not sufficiently robust to withstand the challenges that come with future technology scaling.

Recent research has turned to alternative designs that can replace the 6T cell. One such design is that of the 3T1D DRAM cell, which promises operating speed comparable to that of SRAM without the destructive reads of the standard 1T DRAM cell. Furthermore, 3T1D does not rely on matched transistor strengths, so its reliability is not affected by process technology in the same way that the 6T cell's is. Recent research indicates 3T1D can be used directly in the place of 6T SRAM in structures, such as L1 caches, with negligible performance loss [6].

If chip designers are to consider the 3T1D cell as a practical design option, they need high-level models to quickly estimate 3T1D memory performance and its implications for the overall system. Prior work in memory models consider only 6T SRAM for on-chip memory or implement less analytical DRAM models (e.g., CACTI [2]). However, on-chip DRAM will figure prominently in future variation-tolerant designs. Prior work also overly emphasizes detailed circuit simulation, which makes performance estimates prohibitively expensive for early stage design space exploration by architects. To be widely adopted by architects and to be integrated into chip-level simulators, memory models must be computationally efficient.

To address these fundamental challenges, we propose empirical performance models that combine new circuit simulation methodology with best-known practices in regression modeling. After surveying the background and motivation for 3T1D memories in Section II, this paper presents:

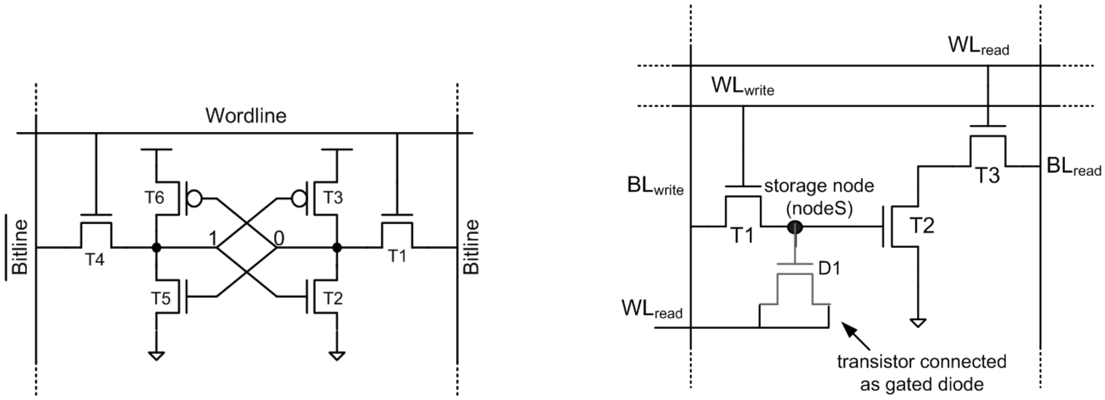
- Circuit simulation methodology that decouples memory array components along the critical path, reducing the size of simulated circuits and capturing performance characteristics more efficiently. (Section III).
- Application of spline-based regression models, which are empirically derived from circuit simulations to accurately estimate performance with the speed of solving analytical regression equations (Section IV).

We apply these new methodologies to construct models to estimate retention and access times for a given set of 3T1D array design parameters, operating conditions, and expected device size variation.

II. Background and Motivation

A. Process Variation and 6T Limitations

Process variation can affect the speed of a 6T SRAM cell, and consequently jeopardize the operating frequency of an entire array. Figure 1a shows the schematic of a standard 6T cell. Reads are performed by precharging both bitlines (the bitline and the inverted bitline) to high, strobing the wordline, and seeing which bitline discharges. If the inverted bitline discharges, a 1 is read from the cell. If the regular bitline discharges, a 0 is read from the cell. For example, this discharge path moves through transistors T1 and T2 to read a 0. Any variation in the gate length or threshold voltage of these transistors changes the current driving capability of the read path, and thus affects the access times of these cells. Within-die variation further complicates this scenario: because these transistor sizes can vary from cell to cell, each memory cell and memory line may operate at different speeds. Circuit designers must clock the circuit at the worst-case operating



(a) Schematic of 6T Cell

(b) Schematic of 3T1D Cell

Fig. 1. Comparison of 6T and 3T1D memory cells.

frequency, leading to significant performance penalties for the entire memory array.

Process variation also attacks the stability of a 6T SRAM cell. For example, transistor T2 is designed to be very strong, transistor T1, moderately strong, and transistor T3, weak. In reads, this allows T2 to quickly discharge the necessary bitline while ensuring the intermediate node between T2 and T3 does not rise enough to store a 1 when it is supposed to store a 0. Any variation within the cell changes the strength of each transistor, and may lead to a weaker T2 that does not discharge the bitline quickly enough. Such variation allows the value at the intermediate node to rise completely and flip the bit stored in the circuit, causing a pseudo-destructive read. The same analysis holds for transistors T4, T5, and T6.

Variation also causes instability in writes. Normally, a value is written to the cell by forcing the normal bitline to the value we want to store, the inverted bitline to the opposite value, and strobing the wordline. The access transistor T1/T4 and the write transistor must be strong enough to overcome the pull-up strength of T3/T6 to flip the bit. By changing the relative strength of each transistor, variation may prevent writes from occurring. Thus, in the case of both reads and writes, process variation makes it hard to ensure that a 6T cell can function reliably. Studies show that even small error rates in an SRAM array can lead to a huge performance loss [1].

B. The 3T1D Cell

In light of such problems with the standard 6T SRAM design, researchers are investigating new cell designs that can better withstand process variation. One such design is that of the 3T1D cell, first proposed by Luk, et. al [7]. 3T1D design is a DRAM memory cell that, unlike a typical 1T or 1T1C design, provides non-destructive reads and high-speed operation that is comparable to (and in some cases better than) the standard 6T SRAM cell. 3T1D is also more compact and produces less leakage power than the 6T cell [6]. Moreover, it does not suffer from the stability issues that are present in the 6T design: no transistor needs to overpower another for this cell to function properly, nor are any two transistors required to be of relatively equal strength. Variation only affects the operating frequency of the cell, making it much more robust to process variation than the 6T design.

Figure 1b presents a schematic of a 3T1D cell. To write to the cell, the write bitline is charged to the value we wish to store in the cell, and the write wordline is strobed. Because of the threshold voltage of transistor T1, a degraded value is stored at storage node S. To read from the cell, the read bitline is precharged high and the read wordline is strobed. If a 1 is stored in the cell, transistor T2 turns on and the bitline discharges. The key to fast access times is the gated diode, which is tied to the read wordline. When a 1 is stored in the cell, the diode provides a "boosting" effect to the value at the storage node and temporarily gives it a value close to (and sometimes greater than) Vdd, which allows T2 to turn on quickly and discharge the bitline.

When a 0 is stored in the cell, the capacitance of D1 is smaller and little to no voltage boosting occurs, keeping T2 turned off. Because the 3T1D is a dynamic memory cell, the value at the storage node leaks away as time passes. As this happens, accesses to the cell become slower and slower. Eventually, this access time becomes so slow that it is no longer comparable to that of the 6T cell. Eventually, the stored value degrades completely. While the fast access times and non-destructive reads of the 3T1D cell position it as a good replacement candidate for the 6T cell, its dynamic nature introduces a new issue that SRAM designers need not consider.

Although there are many instances where static data storage is desirable, it is also important to remember that most data used by modern processors is transient, and need not be stored for large periods of time. Data stored in structures like L1 caches, register files, and TLBs change quickly, and these structures do not necessarily need (or desire) static storage. Following this line of reasoning, recent work investigates the viability of building cache structures out of 3T1D arrays [6]. According to preliminary estimates, 3T1D performs within 2% of 6T memories under typical variation, outperforms 6T by 36% under severe variation, and offers lower leakage power in both cases [6]. 3T1D thus seems to solve a lot of the problems

Parameter	**Retention Time**	**Access Time**	**Access Time**	**Access Time**
		Wordline	Local Bitline	Global Bitline
Vdd	0.8V - 1.4V	0.8V - 1.4V	0.8V - 1..4V	0.8V - 1.4V
Temperature	$0\,^\circ$C-$126\,^\circ$C	$0\,^\circ$C-$126\,^\circ$C	$0\,^\circ$C-$126\,^\circ$C	$0\,^\circ$C-$126\,^\circ$C
Technology	45nm	45nm	45nm	45nm
No. Wordlines		8, 16, 32, 64, 128		
No. Local Bitlines			8, 16, 32, 64, 128	8, 16, 32, 64, 128
No. Global Bitlines				8, 16, 32, 64, 128
T1 length	$4\lambda\pm15\text{nm}^\star$		$4\lambda\pm15\text{nm}^\star$	
T1 width	$3\lambda\pm15\text{nm}^\star$		$3\lambda\pm15\text{nm}^\star$	
T2 length	$2\lambda\pm15\text{nm}^\star$		$2\lambda\pm15\text{nm}^\star$	
T2 width	$16\lambda\pm15\text{nm}^\star$		$16\lambda\pm15\text{nm}^\star$	
T3 length	$2\lambda\pm15\text{nm}^\star$		$2\lambda\pm15\text{nm}^\star$	
T3 width	$4\lambda\pm15\text{nm}^\star$		$4\lambda\pm15\text{nm}^\star$	
D1 length	$8\lambda\pm15\text{nm}^\star$		$8\lambda\pm15\text{nm}^\star$	
D1 width	$20\lambda\pm15\text{nm}^\star$		$20\lambda\pm15\text{nm}^\star$	
Mstart length				$45\text{nm}\pm15\text{nm}^\star$
Mstart width				$2\text{um}\pm15\text{nm}^\star$
Mend length	$45\text{nm}\pm15\text{nm}^\star$		$45\text{nm}\pm15\text{nm}^\star$	
Mend width	$90\text{nm}\pm15\text{nm}^\star$		$90\text{nm}\pm15\text{nm}^\star$	
Time elapsed between write and read			0ns - 500ns	

TABLE I

Performance model parameters and Monte Carlo simulation range. * The range of 15nm is based on the equation: 0.5x(previous technology node - next technology node), measuring approximately 30% variation.

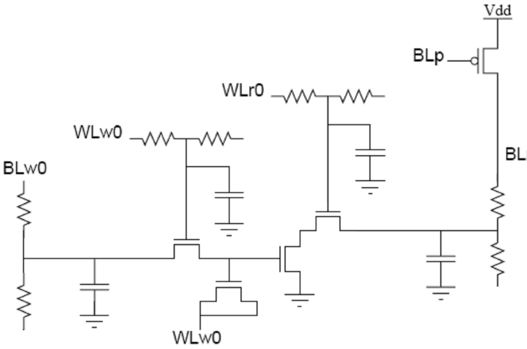


Fig. 2. Circuit schematic for retention time simulation.

encountered by the 6T cell, and stands as a viable replacement option for transient, on-chip memory structures.

III. Circuit Models

Architects rely on memory models to determine what kind of memory structures they need to get desired performance. For example, an L1 data cache requires fast access time but can tolerate a low retention time, but an L2 or L3 cache needs a higher retention time and can tolerate slower access times. Each of these performance targets calls for different transistor sizing and array structuring, and models are essential for figuring out what exactly these targets are. There already exist many different models to help designers understand 6T SRAM cell arrays (e.g. CACTI [2]). Thus, if architects are to really consider 3T1D as a potential replacement for 6T structures, performance models are need to estimate 3T1D performance within architectures.

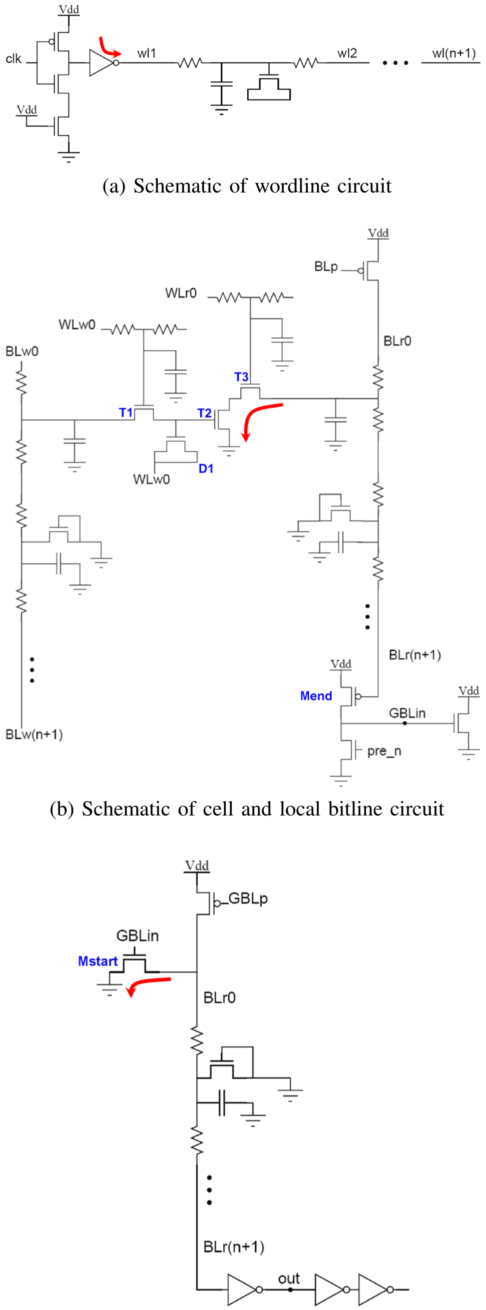
Our performance model consists of two components: a retention time model, which gives the time it takes for the storage node in an individual 3T1D cell to decay to Vdd/4, and an access time model, which gives the time it takes to perform a single read in the array. We do not model writes, as we consider reads to be the worst-case delay.

We perform Monte Carlo simulation using HSPICE to get a large sample of timing data across the different input parameters of Table I. The 3T1D model is intended for chip-level memory hierarchy design and, therefore, estimate performance from high-level design parameters such as the number of wordlines and bitlines. However, given the challenges of technology scaling, architects must also incorporate low-level parameters for environmental conditions (temperature, voltage), device parameters (transistor sizes), and variations in all of the above. We take transistor sizes as proxies for device variations and do not model threshold voltages, which could also be included.

A. Retention Time Model

Our retention time model calculates the retention time for an individual 3T1D cell, given the following input parameters: supply voltage, temperature, and expected variation expressed in terms of transistor sizes. We define retention time as the time required for the storage node in the cell to decay to Vdd/4 after a 1 has been written to the cell. Such understanding allows architects to better identify refresh policies or invalidation schemes necessary for their array.

To generate the retention model, we conduct Monte Carlo simulations of a 3T1D cell using HSPICE, randomly varying input parameters and measuring the cell's retention time. Figure 2 shows the schematic for the simulated circuit. It consists of a single 3T1D cell, connected to simple precharge and driver circuitry as appropriate. Wire parasitics are determined using standard recommendations from the Predictive Technology Models (PTM) [9]. Driver circuitry was sized to produce a fanout-of-four delay, and all input signals were shaped appropriately. To simulate this circuit, we write a 1 to our cell, wait for the value of the storage node to decay past Vdd/4, and report this value.



(a) Schematic of wordline circuit

(b) Schematic of cell and local bitline circuit

(c) Schematic of global bitline circuit

Fig. 3. Circuit schematic for access time simulation.

B. *Access Time Model*

The model calculates access time for an individual 3T1D cell in an array, given the following input parameters: Vdd, temperature, wordline length, local bitline length, global bitline length, time elapsed between write and read, and expected variation expressed in terms of transistor sizes. We define access time as the time required to perform a single read when a 1 has been written to the cell. We consider this read to be the worst case operation of the cell; modeling this read is sufficient to describe the expected performance from a cell.

We adopt an array structure similar to that of the IBM Power6 SRAM arrays, with hierarchical bitlines and no sense amplifiers (often referred to as a domino sense scheme) [11]. This structure is better suited for high frequency operation, typical for 3T1D cells. We also chose a compositional approach to this model, breaking the array into three separate pieces: (1) wordline, (2) cell and local bitline, and (3) global bitline. We model the delay of each piece separately. This compositional approach is more practical for empirical modeling, as users will want varying combinations of wordline and bitline lengths and modeling every combination of these lengths would be intractable. To get the delay for the entire array, we only need to compute the delay through each of these paths and add them together.

To generate the access time model, we conduct Monte Carlo simulations using HSPICE for the three separate pieces. The schematics for each of these components are shown in Figure 3. The wordline circuit (Figure 3a) consists of a simple driver, which is composed of a NAND and NOR gate, and a chain of “dummy” wordline cells, which consist of a single access transistor and the appropriate wire parasitics. The NAND and NOR gates are sized to have a fanout-of-four delay. The local bitline circuit (Figure 3b) consists of a single 3T1D cell, where the read and write bitlines are connected to a chain of dummy bitline cells, which consist of a single access transistor and wire parasitics. The end of the read bitline is connected to circuitry that activates the global bitline. The global bitline circuit (Figure 3c) consists of a precharge transistor, a select transistor that turns the bitline on, a chain of dummy bitline cells, and an output driver.

To simulate these circuits, we consider a typical read operation proceeding through the array and emulate this path in each of the three circuits. In the wordline circuit, we measure the propagation of a clock signal through a driver and chain of dummy (i.e. access transistor only) cells. In the local bitline circuit, we write a 0, write a 1, wait a given time interval, and measure the delay between a read signal on the wordline and the discharging of the last part of the local bitline. In the global bitline circuit, we measure the propagation of a signal through a chain of dummy bitline cells and an output driver.

C. *HSPICE Simulation Results*

Figure 4 presents representative scatter plots of Monte Carlo HSPICE simulations used to construct regression models. We observe a wide range of retention times and delays from the space of parameters in Table I, which highlight the challenges for empirical modeling. Retention times span a range between zero and three microseconds. Figure 4a indicates three discrete segments in wordline delays for our 1000 Monte Carlo trials. These segments indicate three different delay regions, which we find correspond to array configurations with 128 wordlines, 64 wordlines, and fewer than 32 wordlines; the number of wordlines is the most significant determinant of wordline delay. We use this Monte Carlo data to construct empirical regression models, which predict retention times and delays

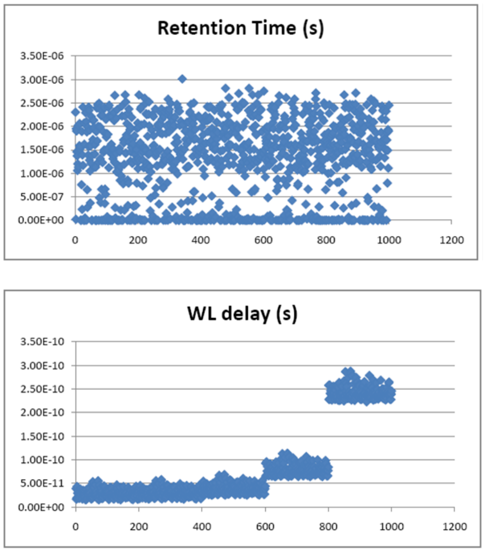


Fig. 4. HSPICE Monte Carlo simulation results.

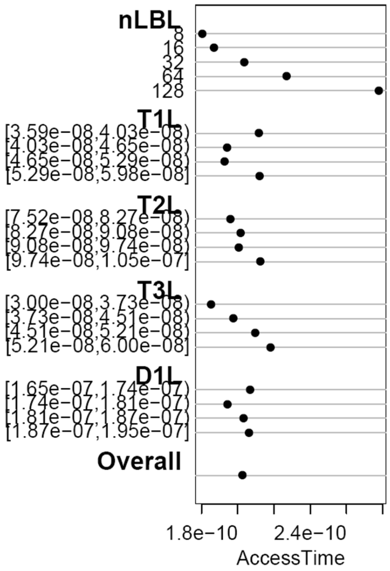


Fig. 5. Association between local bitline access time (sec) and device parameters (number of local bitlines and device lengths (m)) as reported by Monte Carlo simulations. Device lengths are grouped into four intervals and average access time across Monte Carlo simulations within each interval are plotted.

as a function of memory parameters and serve as surrogates for detailed HSPICE simulations.

IV. Regression Models

Regression models are empirically derived equations that express a response as a linear combination of predictors. In computer engineering, such models are often used as computationally efficient surrogates for detailed microarchitectural or circuit simulation [3], [5]. In this paper, we simulate 3T1D circuits to train regression models that estimate performance as a function of input parameters. Thus, we combine the detail of HSPICE data with empirically derived regression equations, simultaneously achieving accuracy close to HSPICE simulation and speed similar to that of purely analytical approaches. Without regression models, traditional Monte Carlo circuit simulations are far too slow for microarchitectural design space exploration.

A. Model Formulation

Notation. Suppose we have a set of n training observations for which values of a response $y = y_1, \ldots, y_n$ and predictors $x_i = x_{i,1}, \ldots, x_{i,p}$, $i \in [1, n]$, of that response are known. Let $\beta = \beta_0, \ldots, \beta_p$ denote regression coefficients used in describing the response as a linear function of predictors plus a random error ϵ_i as shown in Equation (1). F and G are non-linear transformations to capture non-linearity and improve model fit. The errors ϵ_i are independent random variables with zero mean and constant variance. Least squares is commonly used to identify the best-fitting model for the training observations.

$$F(y_i) = G(X_i)\beta + \epsilon_i \tag{1}$$

In this work, 3T1D retention and access times are the responses. We construct four separate regression models: (1) retention time, (2) wordline delay, (3) local bitline delay, (4) global bitline delay. These models are trained with HSPICE simulations of circuits in Figures 2–3. Predictors of these responses are the parameters of Table I.

Predictor Interactions. In some cases, the effect of two predictors x_1 and x_2 on the response cannot be separated; the effect of x_1 on y depends on the value of x_2 and vice versa. This interaction may be modeled by constructing a third predictor $x_3 = x_1x_2$ to obtain $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_1x_2 + \epsilon_i$. We specify these interactions using domain-specific knowledge. For example, devices D1 and T2 of the 3T1D cell in Figure 1b likely interact to affect the local bitline delay. T2's ability to discharge the bitline is determined by the boosting effect, which depends on the size of D1.

Non-Linearity. As illustrated by the non-linearity of Figure 4 in Section III-C, our regression models must capture discrete segments or non-linear trends. We use cubic splines to model non-linearity. Splines are piecewise polynomials, dividing the fitted function into intervals and fitting different polynomials to each interval. Splines of higher order polynomials may offer better fits and cubic splines have been found particularly effective [4], [5].

We determine the number of intervals based on a predictor's significance. If a predictor is highly correlated with the response, we use a greater number of intervals because modeling the predictor's non-linearity is likely more important to overall model accuracy. Less significant predictors will use fewer intervals. This link between significance and spline intervals requires exploratory data analysis to identify strong associations and correlations during model derivation.

Figure 5 illustrates an association analysis for local bitline access times. The scatter plots reveal strong monotonic trends

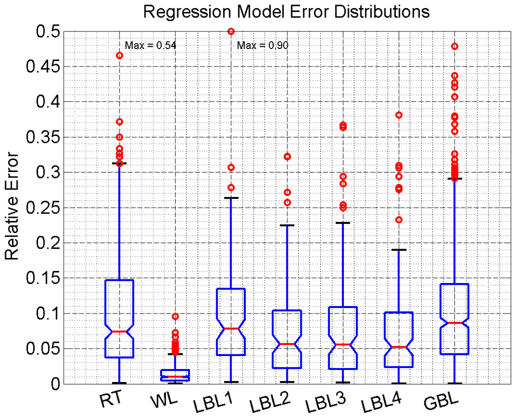


Fig. 6. Box plot of error rates for regression model.

for the number of local bitlines (nLBL) and T3 length (T3L) and a weaker trend for T2 length (T2L). There is no obvious relationship between access times and devices T1 or D1. We reconcile these trends with domain-specific knowledge. Access times depend on the speed at which transistors T2 and T3 discharge the bitline, which depends on transistor sizes. Access times also depend on bitline parasitics, which are affected by the number of connected access transistors. We use a larger number of spline intervals for nLBL, T3L, and T2L given their significance, providing model flexibility to capture trends likely important to predictive accuracy.

B. Fit and Accuracy

We collect 1000 Monte Carlo HSPICE simulations, using 800 for training and reserving 200 for validation. The R^2 statistic quantifies fit to training data with $R^2 \rightarrow 1$ indicating a better fit. The retention time model achieves $R^2 = 0.95$. For components of the access time model, we observe $0.93 < R^2 < 0.98$. Thus, we observe good fit to training data.

For non-training data, Figure 6 illustrates error distributions for model predictions on the 200 independent validation points. These plots illustrate quartiles (horizontal lines), dispersion (vertical lines), and outliers (circles). The performance model is comprised of the retention time model (RT) and the access time model, which is further comprised of wordline (WL), local bitline (LBL), and global bitline delays (GBL).

Retention time is predicted with a median error of 7.39%. Wordline delays are predicted with a median error of 1.01%, the lowest of models presented. These low errors are likely due to the smaller number of predictors used to estimate the response (Table I). Global bitline delay is estimated with a median error of 8.65%.

We construct multiple local bitline delay models with an overall median of 6.65%. Each model predicts delay for a particular time elapsed between write and read. In Figure 6, LBL1 estimates delay for reads that occur immediately after a write. LBL2, LBL3, and LBL4 estimate delays for reads that occur 20ns, 40ns, and 60ns after the write, respectively. Models for other access times were constructed with similar accuracy and we present these four representative models for illustrative purposes.. To combine these separate LBL models into an integrated model and to predict delays for elapsed times not explicitly modeled, we interpolate between predictions from models capturing adjacent points in time.

Across all models, we observe a median error rate of 7.39%. Such error rates are likely sufficient for early stage design optimization. Should designs need greater accuracy later in the design process, additional HSPICE simulations and model refinements might be applied.

V. Conclusions and Future Directions

This paper presents a performance modeling methodology for 3T1D memories. Models are constructed empirically using detailed HSPICE simulations made tractable by breaking the simulated circuit into smaller parts of the read critical path. Spline-based regression on HSPICE training data provides equations to quickly estimate performance metrics. Such models are a necessary tool if computer architects are to successfully implement 3T1D memories and effectively combat the effects of process variation on memory performance.

In the future, we might apply spline-based regression to a number of different memory cells, including 6T, 2T, and 1T1C. We might also use these models to build up models for higher-level structures, like L1 caches or register files. These models might be integrated into simulators or other pieces of software that need information about memory performance. This project provides a powerful proof-of-concept for a promising methodology, opening the doors to many new avenues of research.

References

[1] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, and K. Roy. A process-tolerant cache architecture for improved yield in nanoscale technologies. *IEEE Transactions on Very Large Scale Integration Systems*, 13(1), January 2005.

[2] N. Muralimanohar, R. Balasubramonian, N. Jouppi. Optimizing NUCA Organizations and Wiring Alternatives for Large Caches With CACTI 6.0. *International Symposium on Microarchitecture (MICRO-40)*, Chicago, IL, December 2007.

[3] S. Duvall. Statistical circuit modeling and optimization. *5th International Workshop on Statistical Metrology*, June 2000.

[4] F. Harrell. *Regression modeling strategies.* Spring, New York, NY, 2001.

[5] B. Lee, D. Brooks. Accurate and Efficient Regression Modeling for Microarchitectural Performance and Power Prediction. *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-XII)*, San Jose, CA, October 2006.

[6] X. Liang, R. Canal, G-Y. Wei, D. Brooks. Process Variation Tolerant 3T1D-Based Cache Architectures. *International Symposium on Microarchitecture (MICRO-40)*, Chicago, IL, December 2007.

[7] W. K. Luk, J. Cai, R. H. Dennard, M. J. Immediato, and S. V.Kosonocky. A 3-transistor DRAM cell with gated diode for enhanced speed and retention time. *Symposium on VLSI Technology and Circuits*, June 2006.

[8] K. Meng and R. Joseph. Process variation aware cache leakage management. *International Symposium on Low Power Electronics and Design (ISLPED)*, October 2006.

[9] Predictive Technology Model. http://www.eas.asu.edu/p̃tm/.

[10] D. W. Plass and Y. H. Chan. IBM Power6 SRAM Arrays. *IBM Journal of Research and Development*, 51(6), November 2007.

[11] R. Sam. ZOOM: *A Performance-Energy Cache Simulator*. Thesis, MIT 2002. http://www.cag.csail.mit.edu/scale/papers/regisam-meng.pdf