Modeling and Designing of a PVT Auto-tracking Timing-speculative SRAM

Shan Shen, Tianxiang Shao, Ming Ling, Jun Yang, Longxing Shi
Nation ASIC System Engineering Technology Research Center
Southeast University
Nanjing, China
{shanshen, txshao, trio, dragon, lxshi}@seu.edu.cn

***Abstract*—In the low supply voltage region, the performance of 6T cell SRAM degrades seriously, which takes more time to achieve the sufficient voltage difference on bitlines. Timing-speculative techniques are proposed to boost the SRAM frequency and the throughput with speculatively reading data in an aggressive timing and correcting timing failures in one or more extended cycles. However, the throughput gains of timing-speculative SRAM are affected by the process, voltage and temperature (PVT) variations, which causes the timing design of speculative SRAM to be either too aggressive or too conservative.**

This paper first proposes a statistical model to abstract the characteristics of speculative SRAM and shows the presence of an optimal sensing time that maximizes the overall throughput. Then, with the guidance of the performance model, a PVT auto-tracking speculative SRAM is designed and fabricated, which can dynamically self-tune the bitline sensing to the optimal time as the working condition changes. According to the measurement results, the maximum throughput gain of the proposed 28nm SRAM is 1.62X compared to the baseline at 0.6V VDD.

Keywords—Timing-speculation, SRAM, low-power design, performance model, PVT

I. INTRODUCTION

In recent years, energy efficiency has become more important for the System on a Chip (SoC) as the demand of Internet of Things (IoT) and other mobile devices increases in the market. Scaling down the supply voltage is one of the most commonly used methods in the low-power (LP) design, which brings the energy efficiency close to the optimal point [1]. Operating at low supply voltages, whereas, static random access memory (SRAM) is more prone to faults under the process variations due to its minimum-sized transistors. There are two major types of failures in memory cells: (1) timing failures that increase the cell access time and (2) unstable read/write operations [2]. The later problems can be solved by the dedicated read port in a memory cell, such as 8T [3][4] and 10T [5], or using the assist peripheral [6-8], to strengthen the read/write ability of 6T SRAM. To solve the former challenge, timing-speculative approaches [10-16] are proposed. The basic idea of timing speculation is that the SRAM employs two-phased voltage sensings (called the risk-sensing and the confirm-sensing, respectively) during an access, where the first sensing speculatively reads the small voltage difference between bitline (BL) pair while the second is used to confirm the read results. By comparing these two sensing outcomes, a possible timing error can be identified and then corrected after an extended reading or even more readings in the following cycles. Since the weak cells that cause timing failures are minor, most of the sensing results from strong cells can be directly used after the first sensing [12]. Theoretically, these timing-speculative SRAM (or called speculative SRAM for short) can increase the operating frequency to 2X or more compared to that of the conservative SRAM design.

However, the risk-sensing time in the speculative SRAM, which is equivalent to the wordline (WL) enable time, is a critical point that directly determines the SRAM frequency and throughput in the LP design. This is because the performance penalty of timing speculation highly correlated with the risk-sensing time. It affects the read failure rate, which decides the probability of doing error corrections in speculative SRAM, and the latency of a single error correction, where the risk-sensing time is the major component of reading delay. Moreover, due to the variations of error rates in different PVT, a pre-configured risk-sensing time makes the throughput gains vary significantly when the working condition has changed. In some cases, the error correction penalty even nullifies the performance improvement brought by timing speculations (Fig. 9).

On the other hand, directly using Monte Carlo (MC) analysis in SPICE to cover all working conditions is painfully inefficient when the capacity of SRAM becomes large. To address the above challenge, this work first proposes a statistical model to estimate the performance of timing-speculative SRAM under different PVT. Meanwhile, this model also predicts the best risk-sensing time to maximum the throughput gains of the speculative SRAM under a specified PVT condition. With the fast and precise performance evaluation, a self-tuned PVT auto-tracking SRAM is designed and fabricated, which can automatically probe the optimal frequency when it works in the transient voltages/temperatures (V/T).

II. RELATED WORK & MOTIVATION

A. Related Work

Regarding the timing speculation technique, Karl et al. [11] first applied the in-situ timing error detection to SRAM, which contains shadow SAs in addition to the main SA. The main SA is triggered speculatively at the clock's negative edge. After a while, the shadow SA re-samples the bitlines to confirm the result. The system detects the number of errors where the two samples are different during voltage scaling. Khayatzadeh et al. [12] proposed the Razor SRAM that reads memory twice with

This work was supported in part by the National Natural Science Foundation of China under Grant 61974024 and Grant 61874152, and in part by the Provincial Natural Science Foundation of Jiangsu Province under Grant No. BK20181141.

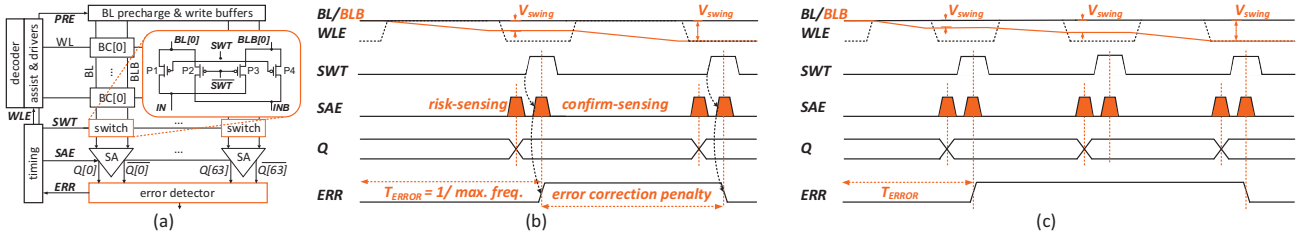

Fig. 1. (a) The structure of CS-SRAM and (b) the corresponding timing diagram with 1 extended cycle, and (c) with 2 extended cycles to correct errors.

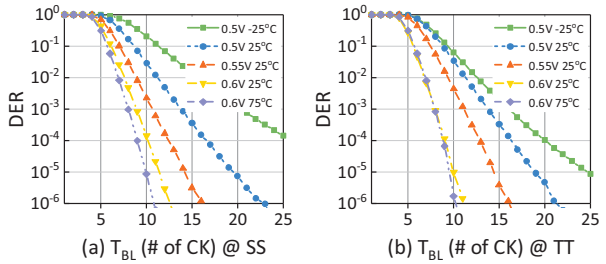

Fig. 2. Double-word error rate vs. risk-sensing time (T_{WL}) at different (a) supply voltages and (b) temperatures.

dual ports in a pipelined manner. In most cases, the read output is available after the first cycle and then confirmed by comparing with the second sample in the next cycle. For weak bits, the error flag will be triggered due to the two unequal samples. Kumar et al. [10] proposed speculative SRAM with two sense amplifiers (SA) that have opposite offset voltages by using compensatory capacitors. Also by comparing the two sensing outcome, the weak bits can be identified. However, these solutions only provide a limited latency reduction due to the too-late error detections, and they introduce large area overhead for the error detection and correction logic. To achieve a higher throughput, Yang et al. [13] proposed a double sensing scheme with selective bitline voltage regulation (DS-SBVR), where the V_{BL} is dynamically regulated by charge sharing between two sensing steps (comparisons of different timing speculations are shown in [13][14]). Shang and Ling et al. [15][16] proposed RRS caches to hide the error correction penalty of speculative SRAM in an architectural solution.

In the aspect of PVT tracking SRAM, Banerjee et al. [9] proposed a wide-voltage-range SRAM using three combined read/write assists and canary-based V_{MIN} tracking. By detecting the errors in a canary row, the SRAM self-tunes to the V_{MIN} across process and temperature for a target frequency, which achieves a great power reduction.

B. Motivation

In the work of [14], Shen et al. proposed a Timing-Speculation (TS) cache, with very limited area overhead, to boost the cache frequency and improve energy efficiency under low supply voltages, in which multiple speculative SRAM arrays with cross-sensing technique (CS-SRAM) are used for the implementation of the data arrays. During a cross-sensing reading, the voltage differences on BL are continuously evaluated twice by a sense amplifier (SA). Fig. 1 shows the (a) basic structure and (b) the timing diagram of CS-SRAM. A switch in each data column is disabled by a low *SWT* signal at the first *SAE* to connect the BL and BLB to IN and INB of a SA, while is activated by an *SWT* assertion to exchange the connections (BL to INB and BLB to IN) at the second *SAE*. The two outcomes from cross-sensing (CS) are compared by the error detector, and the *ERR* keeps high if the two outcomes are equal. The timing errors can be corrected by extending the BL discharging time in the next cycle, which is controlled by another *WLE* pulse. The error correction is also a CS operation but without BL precharging. All timing pulses are configured as multiple cycles of clock (CK) period generated by the replica bitlines [13][14].

Assuming the risk sensing follows the negative edge of *WLE* closely, the risk-sensing time equals the width of *WLE* pulse (T_{WL}), which consists of the BL discharging time (T_{BL}) and the wordline driving time. In low supply voltages, T_{BL} is the largest part of T_{WL}, which affects the performance of speculative SRAM from two aspects, the probability of timing error occurrences (identified by the error detection logic) and the latency of a single error correction. Fig. 2 shows the relationship between the 64-bit doubleword error rate (DER) and T_{BL} under different PVT. The results are collected from 2 million fast Monte Carlo simulations of a 256-row * 64-column SRAM array with the 28nm process (the fast MC simulation method is same as the work in [13] and [14]). Obviously, the DER exhibits a monotonically decreasing function of T_{BL}. Moreover, for a given T_{BL}, it decreases as the V/T scales up, or the global process corner becomes faster. A pre-set and fixed risk-sensing time would make the probability of timing errors of speculative SRAM significantly various in the transient PVT conditions and may degrade the speculative SRAM performance unexpectedly in some cases. Meanwhile, a short T_{BL} worsens the latency of a single error correction since the voltage differential on BL and BLB may not sufficient to be recognized by SAs in only one extended cycle. Consequently, a multi-cycle error correction will occur in the speculative SRAM. Fig. 1 (c) shows an example where the latency of an error correction increases from 1 to 2 cycles for a shorter risk-sensing time.

Combining these two aspects, though an aggressive timing design of risk-sensing largely boosts the SRAM frequency, it brings a higher DER and a longer latency of error correction. Oppositely, a conservative risk-sensing time reduces the DER and the error correction delay, however, it limits the overall frequency and throughput due to a relatively conservative timing margin. Therefore, to further harvest the throughput gains in speculative SRAM, the optimal risk-sensing time needs to be predicted. Unfortunately, the traditional SRAM design is based on the large-scale MC simulations to cover the worst-case situation with the presence of PVT variations. Such a time-consuming method fails to evaluate the speculative SRAM when

the capacity of SRAM grows and the peripheral circuits become complicated.

III. Performance Model

A. Modeling the Read Delay of Speculative SRAM

When a timing error is detected, previous work uses two different error correction strategies, conservative sensing [11][12][13] or continuously cross-sensing [14] in the following cycles. This work bases on the CS-SRAM, where the CS operation can be executed several times until the timing errors are completely corrected. Although the destructive read could happen if the V_{BL} is discharged too lower, such a situation does not happen in our simulations using the standard 6T cells provided by TSMC foundry under low supply voltages. Therefore, our model does not consider the destructive read for weak cells. The average read delay D under different times of CS can be modeled as

$$D = P(C_1)\cdot(t+r) + P(E_1)\cdot \begin{pmatrix} P(C_2|E_1)\cdot 2(t+r) + P(E_2|E_1)\cdot \\ (P(C_3|E_2)\cdot 3(t+r) + P(E_3|E_2)\cdot(\cdots)) \end{pmatrix} \quad (1)$$

where $P(C_N)$ and $P(E_N)$ are the probabilities that a data word is correctly or wrongly read at the Nth operation, respectively. $P(C_N \mid E_{N-1})$ denotes the probability that an error word detected by the $(N-1)$th CS is correctly read in the next operation:

$$P(C_N|E_{N-1}) = \frac{P(C_N \cap E_{N-1})}{P(E_{N-1})} = \frac{P(E_{N-1}) - P(E_N)}{P(E_{N-1})} \quad (2)$$

For example, in a given working condition and T_{WL}, the first CS operation reports an error word with 10% probability while the next CS detects the same error with 1% probability, $P(C_2|E_1)$ is 90% ($=\frac{10\%-1\%}{10\%}$). Similarly, $P(E_N|E_{N-1})$ is the probability where the result of a data word read by the Nth CS is still a timing error:

$$P(E_N|E_{N-1}) = \frac{P(E_N \cap E_{N-1})}{P(E_{N-1})} = \frac{P(E_N)}{P(E_{N-1})} \quad (3)$$

Let $T_{BL} = t$ and r be the remaining delay in an SRAM reading (including the WL driving time, SA enable time, etc.). By replacing the conditional probabilities with the doubleword error rates, $P_E(t)$, which is a function of t, (1) can be re-written as

$$D = (1-P_E(t))\cdot(t+r) + P_E(t)\cdot \begin{pmatrix} \frac{P_E(t)-P_E(2t)}{P_E(t)}\cdot 2(t+r) + \frac{P_E(2t)}{P_E(t)}\cdot \\ \left(\frac{P_E(2t)-P_E(3t)}{P_E(2t)}\cdot 3(t+r) + \frac{P_E(3t)}{P_E(2t)}\cdot(\cdots)\right) \end{pmatrix} \quad (4)$$

and after simplification, the average read delay of CS-SRAM is

$$D = (t+r)\cdot(1 + P_E(t) + P_E(2t) + P_E(3t) + \cdots) \quad (5)$$

To get a precise average delay estimation from (5), it seems that we need DER results with a wide range of t. However, since the maximum t can be 25 CKs to achieve the required 5σ access-time yield, calculating $P_E(Nt)$ will take an unacceptable time and storage overhead even for the fast MC simulation method. Fortunately, a large t leads to an infinitesimal DER, (5) can be further simplified by ignoring the small terms of $P_E(2t)$, $P_E(3t)$, ⋯ , the average read delay can be expressed as:

$$d \approx (t+r)\cdot(1 + DER(t)) \quad (6)$$

Then, by enumerating the possible values of t and calculating the corresponding hit latencies from (6), the best T_{WL} can be found. Note that the best sensing time is the local minimal extremum of (6) rather than the global minimum value because the simplified model introduces a large discrepancy in the range of small t. In Section V-A, the complete and simplified models will be compared in detail. The model of the speculative SRAM using another error correction strategy (conservative sensing) has an analogical form, and the properties of (5) and (6) will be analyzed in our future work. Besides, we emphasize that the DER under various PVT can be obtained through either a fast spice-like simulation or a yield model [17]. This work uses a fast MC simulation based on the work of [13].

IV. PVT Auto-tracking SRAM

Based on the insights provided by our model, this work proposes a self-tuned PVT auto-tracking SRAM, where we employ the model in the design flow to replace the large-scale SPICE simulations and evaluate the overall performance of the proposed design.

A. Trial-and-error Strategy

To simplify the implementation and the hardware overhead, a simple yet efficient trial-and-error strategy is adopted. It dynamically adjusts the *WLE* pulse width (T_{WL}) step by step in a working phase to configure the different risk-sensing time and obtain different throughputs. The whole SRAM working duration is partitioned into several execution phases with respect to the number of SRAM readings, in which we assume the PVT condition doesn't change acutely. During each phase, regardless of the working mode, the PVT auto-tacking SRAM monitors the read number, the error-word counts, and the total read latencies from the last and the current execution phases. At each end of the phase, as Fig. 3 shows, the PVT auto-tracking SRAM checks its working mode.

1) In the normal mode: the SRAM calculates the error word difference between the current and the last phases and compares the difference to the pre-set threshold. If the DER reduction exceeds the threshold, which means the margin of T_{WL} is larger for the current operating condition, the SRAM enters the calibration mode, where the T_{WL} will be decreased by 1 CK step. Conversely, the T_{WL} will be increased by 1 CK when the DER augmentation is larger than the threshold and results in frequent error-corrections. Such a strategy also matches the property of the performance model (5) (more analyses in Section V-A).

2) In the calibration mode: If the PVT auto-tracking SRAM already runs in calibration mode, the total read latencies (rather

the DER) of the current and the last phases are compared. If the read latency keeps reducing, which means it is approaching the optimal performance point, the T_{WL} of the speculative SRAM will be continuously increased or decreased in the following phase. Otherwise, the overall throughput starts deteriorating, and the T_{WL} is restored to its last value that provides the lowest read latency. At the same time, the SRAM exits the calibration mode.

Before a new execution phase starts, the counters that record the information of the last phase copy the current error-word count and the total read latency, while other counters are reset for the next new phase.

Note that the DER/latency comparisons are only activated when the last SRAM reading in the execution phase is completed, and T_{WL} calibrating finishes before the next SRAM response starting. Hence, the PVT auto-tracking SRAM can be accessed and interfaces with other control logic normally regardless of its operating mode.

B. Hardware Design

Fig. 4 illustrates the hardware architecture of the proposed SRAM. The basic SRAM module including the SRAM array, read/write assist circuits, a row decoder, an error detector, and a timing module. The replica bitline generates the basic CK for SRAM, which is also designed to auto-track the worst-case memory cell under PVT variations. The T_{WL} calibrator collaborates with the read/write controller to dynamic configure the SRAM timing. In an execution phase, the calibrator records the number of error flags generated from the error detector, the number of read operations, and the total read latency. The error flags are also received by the R/W controller to do the error corrections. When the read number reaches the phase length, the T_{WL} calibrator changes the working mode according to the DER/total latency changing and sends T_{WL} configuration to the timing module that can also be manually configured through the interface.

C. Overhead

After synthesis, the T_{WL} calibrator only takes 598um^2 chip area, which is only 2.1% area overhead of a 28nm 16KB SRAM. The energy consumed by the calibrator can be ignored compared with that of the SRAM arrays. When the PVT auto-tracking SRAM boots, it takes time to complete the first calibration if the initial T_{WL} is set to the largest, or the most conservative value (32 CKs in our design). Fortunately, since the mechanism of cross-sensing ensures the final data is always correct [14], the SRAM can still be accessed even though it works with an extreme T_{WL} that provides poor throughput. Commonly, the calibration mode takes no more than 4 execution phases after the first adjustment (shown in the next section). Furthermore, recording and comparing the DER/latency are parallel with the SRAM reading without introducing any delay.

V. Evaluation

In this section, a 256-row * 64-column CS-SRAM is used as the basic sub-arrays in the evaluation. The results are collected from 512 fast MC simulations for each PVT condition with a 32KB SRAM (i.e., sixteen basic sub-arrays), r = 4 CKs.

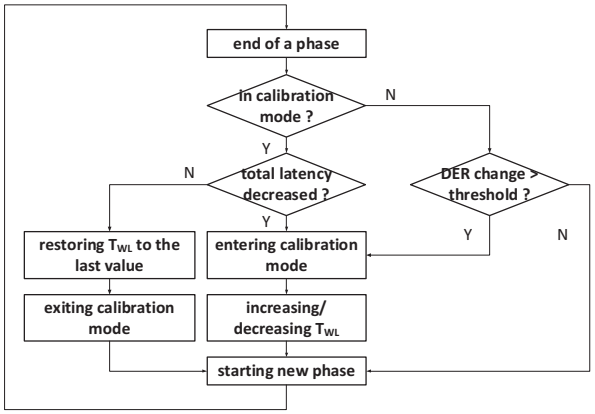

Fig. 3. The flowchart of PVT auto-tracking SRAM at the end of an execution phase.

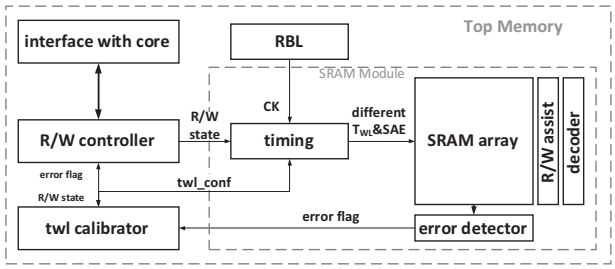

Fig. 4. The hardware architecture of PVT auto-tracking SRAM.

A. Accuracy of Performance Model

Fig. 5 (a) - (d) compares the complete model (5), the simplified model (6), as well as the simulation results. Since the pulse width of the bitline discharging time is a discrete value, t varies from 1 to 25 CKs. The average delay is also shown in the form of CK numbers. The complete performance model shows a relative high accuracy to depict the behavior of the speculative SRAM. The 'W' shaped curves of the simulation results and the complete model demonstrate that the best performance points (i.e., the minima on the complete model curves) exist across different PVT. Compared to the optimum, a more conservative T_{WL}, which introduces large timing margine, or an more aggressive T_{WL}, which results in frequent error correcting, has a worse read delay. Although the simplified model has significant errors at the section of lower t, it perfectly converges to the complete model when t grows larger than the first local maximal extremum. Note that the precise part of the simplified model has the same minimal extremum value as (5), which can be used to estimate the optimal risk-sensing time without any error. Therefore, the inconsistency at small T_{BL} does not impact the optimal risk-sensing time prediction, and such a short timing strobe would not be used in the real LP design either. Moreover, the optimal T_{WL} reduces as the V/T rises. Such property of the performance model is leveraged by the PVT auto-tracking SRAM. In addition, the optimal T_{WL} is insensitive to the global variations since it stays the same value at SS, TT, FF global corners (the graphs are not shown due to the space limitation).

B. Calibration Accuracy of the PVT Auto-tracking SRAM

To mimic the environment changing during SRAM working, as Table I shows, we set up to 10 sets of experiments that contain

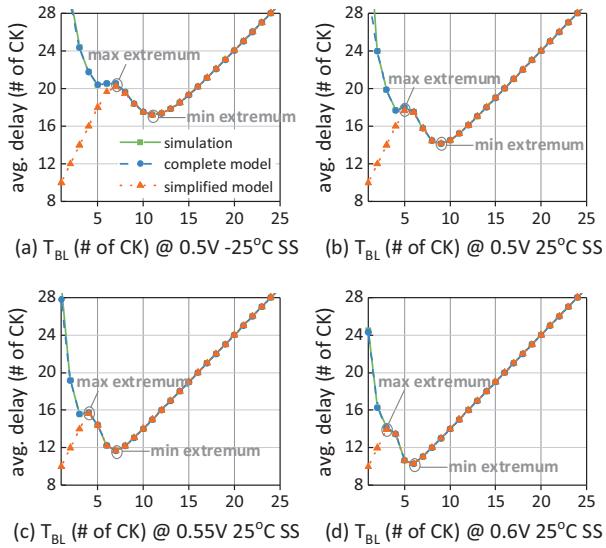

Fig. 5. (a)-(d) Average delay estimated from the complete/simplified models and the simulation result at different V/T conditions.

different PVT variations. The steps of voltage and temperature variations are 50mV and 25℃, respectively. The phase length is 4K SRAM readings and the threshold is set to 8 error-words . Fig. 6 (a) shows the average differences between the self-tuned T_{WL} using the trial-and-error strategy and the true optimal T_{WL} from simulations. The average difference doesn't exceed 0.02 CK. Comparing the results from Set1~3 and Set4~6, the variation of temperature affects the precise of T_{WL} adjustment more seriously than that of voltages.

Assuming a 10ns latency per SRAM reading at low voltages, an execution phase takes tens of milliseconds (4K readings per phase in this paper). Fig. 6 (b) shows the average number of phases consumed by the calibration mode to probe the best T_{WL}. On average, it takes 3 working phases to converge at the best T_{WL}, which is a relatively short period compared to the whole SRAM working duration.

TABLE I. COMBINATIONS OF PVT CHANGING IN EXPERIMENTS.

index	Process Corner	Voltage (V)	Temperature (°C)
Set1	SS	0.5	-25 ~ 75 ~ -25
Set2	TT	0.5	-25 ~ 75 ~ -25
Set3	FF	0.5	-25 ~ 75 ~ -25
Set4	SS	0.5 ~ 0.65 ~ 0.5	-25
Set5	SS	0.5 ~ 0.65 ~ 0.5	25
Set6	SS	0.5 ~ 0.65 ~ 0.5	75
Set7	SS	0.5 ~ 0.65	-25 ~ 75
Set8	SS	0.65 ~ 0.5	-25 ~ 75
Set9	SS	0.5 ~ 0.65	75 ~ -25
Set10	SS	0.65 ~ 0.5	75 ~ -25

C. Performance Improvement

Three strategies to configure the risk-sensing time are compared. The first one is the fixed risk-sensing time regardless of environment changes. In our experiments, the risk-sensing time is set to 9 CKs for this strategy that achieves a relatively high throughput under most circumstances. Another scheme is the half-of-worse-case (half-of-WC) configuration, which is

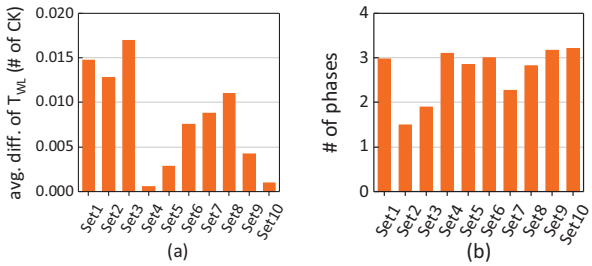

Fig. 6. (a) The average differences between the self-tuned T_{WL} using trial-and-error strategy and the the true optimal T_{WL} from simulations in the PVT transitions. (b) The average number of phases consumed in the calibration mode after PVT changing.

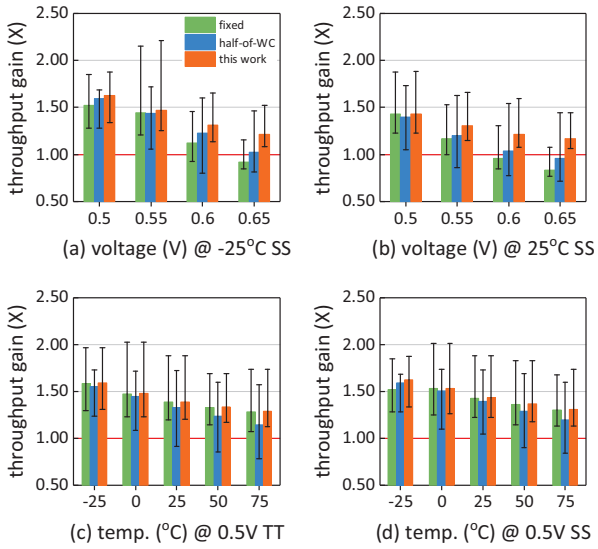

Fig. 7. (a)-(d) The average (bar), max (upper lines), and min (lower lines) throughput gains for CS-SRAM using three T_{WL} configuring strategies at different PVT.

used in the work [11][12][14][16]. The final strategy uses the trial-and-error method proposed by this work.

Fig. 7 compares the average (colored bars), maximal (the upper vertical lines), and minimal (the lower vertical lines) throughput gains of the three strategies at different working conditions. The throughput gain is defined as the ratio of the maximal throughput to that of the baseline SRAM without using any timing-speculation (same as [14]). The throughput gains are reduced when V/T rises at SS corner. For the prior fixed and half-of-WC strategies, the average throughput gains are lower than 1X (i.e., degrade the SRAM performance) at 0.65V, while the minimal throughput gains are further lowered when the supply voltage exceeds 0.55V. The reason for this phenomenon is the slopes of error rates become steeper, as the V/T increases. It leads to a dramatic increase of the error word count even reducing T_{WL} by one CK. Moreover, the reduced T_{BL} at a higher V/T improves the throughput of the baseline SRAM, which, in turn, limits the performance benefit provided by timing speculation. Oppositely, the proposed solution provides a 2.21X maximum throughput gain at 0.55V -25℃, and the lowest minimum gain is 1.07X at 0.65V 25℃, which never hurts the SRAM performance at all PVT combinations in our simulations.

VI. Measurements

The simulation results often provide optimistic evaluations on SRAM performance due to the underestimated backend parasitic parameters. To further show the functionality of PVT auto-tracking SRAM, twenty 28nm 16KB SRAM chips are fabricated using TSMC technology, each consisting of eight 256-row * 64-column sized CS arrays. The chip micrograph is shown in Fig. 8 (a). The self-tuned T_{WL} variation from 8 representative SRAM chips in PVT transitions is plotted in Fig. 8 (b). The direction of T_{WL} calibration matches the simulation and performance model results, where the self-tuned risk-sensing time increases as V/T drops. Fig. 9 (a) - (d) further depicts the throughput gain distribution at different V/T. The max throughput reaches 1.63X at 0.6V 25℃ compared to the baseline SRAM. At 0.7V VDD 25℃ and 0℃, most PVT auto-tracking SRAM chips provide the throughput gains by 1.2X-1.4X.

VII. Conclusion

This paper first proposes an analytical model to abstract the characteristics of timing-speculative SRAM. It shows that an optimal timing exists and changes with PVT variations. To leverage the property of speculative SRAM, we also design and fabricate a timing-speculative SRAM that can self-tuned its sensing time to the optimal value that maximizes the overall throughput gains in terms of PVT transitions.

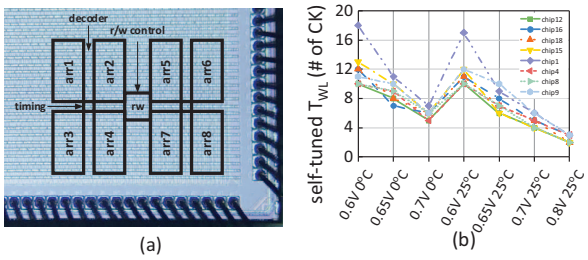

Fig. 8. (a) Chip micrograph of PVT auto-tracking SRAM. (b) The variation of the self-tuned T_{WL} in the proposed SRAM under different PVT.

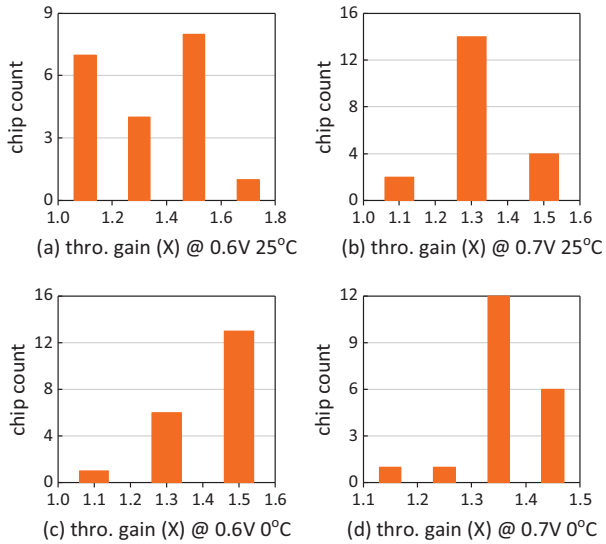

Fig. 9. Chip distributions of overall throughput gain at different working conditions. The total number of test chips is 20.

References

[1] Alioto, Massimo. "Ultra-low power VLSI circuit design demystified and explained: A tutorial." IEEE Transactions on Circuits and Systems I: Regular Papers 59.1 (2012): 3-29.

[2] Mukhopadhyay, Saibal, et al. "Modeling and estimation of failure probability due to parameter variations in nano-scale SRAMs for yield enhancement." 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525). IEEE, 2004.

[3] Chen, Chien-Fu, et al. "A 210mV 7.3 MHz 8T SRAM with dual data-aware write-assists and negative read wordline for high cell-stability, speed and area-efficiency." 2013 Symposium on VLSI Circuits. IEEE, 2013.

[4] Wu, Jui-Jen, et al. "A Large σ VTH/VDD Tolerant Zigzag 8T SRAM With Area-Efficient Decoupled Differential Sensing and Fast Write-Back Scheme." IEEE Journal of Solid-State Circuits 46.4 (2011): 815-827.

[5] Calhoun, Benton Highsmith, et al. "A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation." IEEE journal of solid-state circuits 42.3 (2007): 680-688.

[6] Raychowdhury, Arijit, et al. "PVT-and-aging adaptive wordline boosting for 8T SRAM power reduction." 2010 IEEE International Solid-State Circuits Conference-(ISSCC). IEEE, 2010.

[7] Sinangil, Mahmut E., et al. "A 28nm high-density 6T SRAM with optimized peripheral-assist circuits for operation down to 0.6 V." 2011 IEEE International Solid-State Circuits Conference. IEEE, 2011.

[8] Fujimura, Yuki, et al. "A configurable SRAM with constant-negative-level write buffer for low-voltage operation with 0.149 μm 2 cell in 32nm high-k metal-gate CMOS." 2010 IEEE International Solid-State Circuits Conference-(ISSCC). IEEE, 2010.

[9] Banerjee, Arijit, et al. "A 256kb 6T self-tuning SRAM with extended 0.38 V–1.2 V operating range using multiple read/write assists and V MIN tracking canary sensors." 2017 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2017.

[10] Kumar, Ashish, G. S. Visweswaran, and Kaushik Saha. "Low voltage error resilient SRAM using run-time error detection and correction." ESSCIRC Conference 2015-41st European Solid-State Circuits Conference (ESSCIRC). IEEE, 2015.

[11] Karl, Eric, Dennis Sylvester, and David Blaauw. "Timing error correction techniques for voltage-scalable on-chip memories." 2005 IEEE International Symposium on Circuits and Systems. IEEE, 2005.

[12] Khayatzadeh, Mahmood, et al. "17.3 a reconfigurable dual-port memory with error detection and correction in 28nm fdsoi." 2016 IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2016.

[13] Yang, Jun, et al. "A Double Sensing Scheme With Selective Bitline Voltage Regulation for Ultralow-Voltage Timing Speculative SRAM." IEEE Journal of Solid-State Circuits 53.8 (2018): 2415-2426.

[14] S. Shen et al., "TS Cache: A Fast Cache With Timing-Speculation Mechanism Under Low Supply Voltages," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2019). doi: 10.1109/TVLSI.2019.2935227

[15] Shang, Xiaojing, et al. "RRS Cache: A Low Voltage Cache based on Timing Speculation SRAM with a Reuse-aware Cacheline Remapping Mechanism" 2019 IEEE International Symposium on Memory Systems (MEMSYS 2019), in press.

[16] Ling, Ming, et al. "Lowering the Hit Latencies of Low Voltage Caches Based on the Cross-Sensing Timing Speculation SRAM." IEEE Access 7 (2019): 111649-111661.

[17] Sun, Zhongmao, et al. "Approximate Yield Estimation of Correlated Failure Events for Near-threshold SRAM." IOP Conference Series: Materials Science and Engineering. Vol. 565. No. 1. IOP Publishing, 2019.