Postsilicon Adaptation for Low-Power SRAM under Process Variation

Minki Cho
Georgia Institute of Technology

Hamid Mahmoodi
San Francisco State University

Jason Schlessman
Princeton University

Marilyn Wolf and Saibal Mukhopadhyay
Georgia Institute of Technology

Editor's note:
Due to the high density requirement for embedded memories, such memories are highly vulnerable to process variation–induced failures. A conservative design approach can largely affect memory density and access performance. This article analyzes variation effects in SRAM and presents low-cost, adaptive postsilicon repair mechanisms.

—Swarup Bhunia, Case Western Reserve University

Die-to-die and within-die statistical variations in process parameters result in parametric failures (read, write, access, and hold failures) in SRAM cells, thereby degrading design yield. The principal reason for parametric failures is intradie variation in the cell transistors' threshold voltage due to random dopant fluctuations (RDFs).[1] On the other hand, die-to-die variation in process parameters (such as threshold voltage V_{TH}) can significantly increase the impact of within-die random variations.[2] From a yield perspective, this increasing variation manifests itself as yield degradation. In addition, variation is a critical bottleneck for reduction of array operating voltage, and hence power. This is because a lower supply voltage makes the SRAM more sensitive to manufacturing variations, and the probability of parametric failures increases with voltage scaling.[2] Therefore, from a system perspective, a lower supply voltage leads to a higher bit-error rate (BER) and limits the opportunity for power savings.

Design-level approaches such as transistor sizing and optimization of different array voltages can help increase tolerance to process variations,[1,2] but the design considerations for different failure modes have conflicting requirements. For example, write failures decrease with stronger access transistors, but this degrades read-disturb failures. To cope with these limitations, researchers have proposed various circuit techniques to dynamically improve read and write stabilities, depending on the type of operations.[3,4] Presilicon design and optimization approaches help improve tolerance to random process variations, but they often fall short if die-to-die variations are nonnegligible. Employing postsilicon adaptive repair, along with presilicon design techniques, provides a new opportunity to improve SRAM yield, considering both local and global variations.[2]

In this article, we describe postsilicon adaptation methods that improve SRAM yield and reduce power dissipation under process variation. We discuss these methods in the context of on-chip SRAM for multimedia systems. The yield of on-chip SRAM can significantly impact the cost of multimedia systems. Furthermore, because SRAM power is a significant component of system power consumption in multimedia systems, power reduction in SRAM is important.

In multimedia systems, the SRAM array is shared between data- and image-storage applications. Typically, data-processing applications cannot tolerate any memory failures. Thus, we consider postsilicon repair of SRAM arrays to ensure correct functionality at nominal supply voltages. The different types of

0740-7475/10/$26.00 © 2010 IEEE Copublished by the IEEE CS and the IEEE CASS

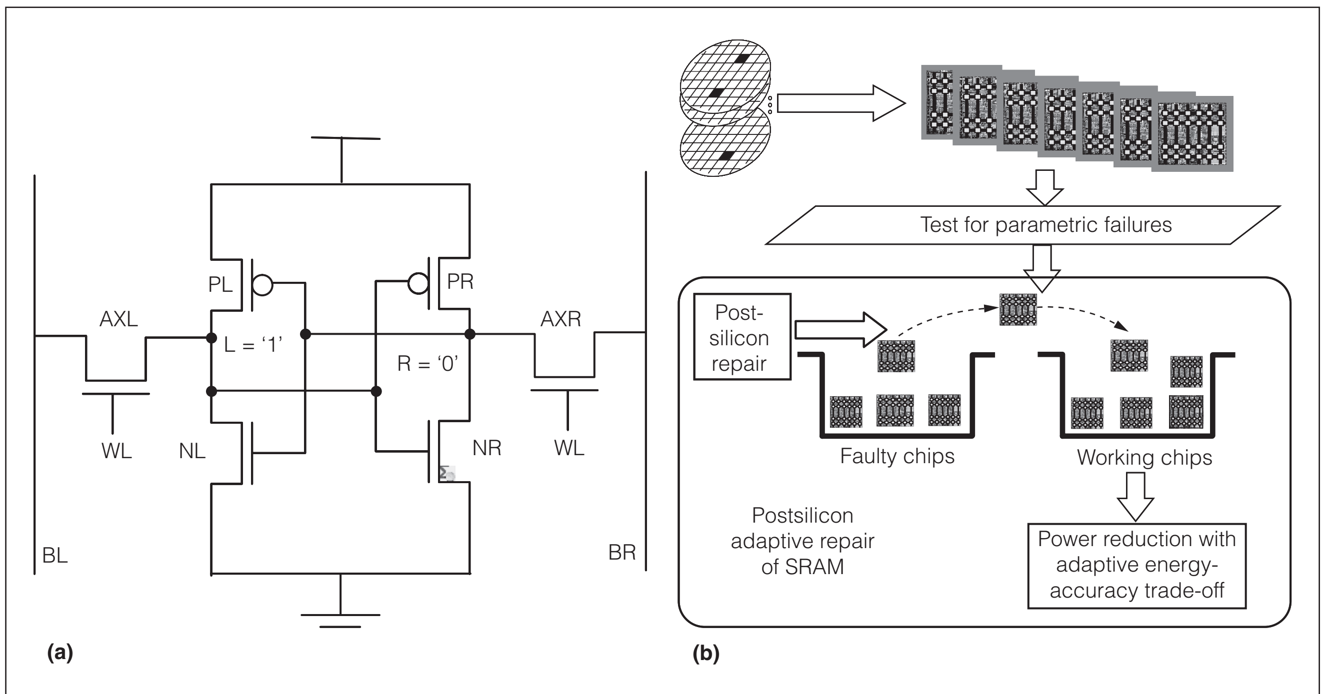

Figure 1. Postsilicon adaptive repair for yield enhancement and power reduction in SRAM: circuit schematic of an SRAM cell (a), and overall methodology for postsilicon adaptation (b). (AXL and AXR: access transistors; BL: left bitline; BR: right bitline; L: left node; NR and NL: pull-down NMOS transistors; PL and PR: pull-up PMOS transistors of an SRAM cell; R: right node; WL: wordline.)

parametric failures can increase at different interdie corners. Accordingly, postsilicon repair in SRAM arrays aims to reduce the dominant types of failures at different interdie corners using adaptive biasing, thereby correcting faulty arrays and improving design yield.[3] With such repairs, the SRAM arrays operate in an "always correct" mode at nominal supply voltages. However, an increase in parametric failures at lower supply voltage limits the voltage, and hence power reduction, in SRAM. Although this causes a serious bottleneck when SRAM arrays store data, image-processing and multimedia applications can provide acceptable quality of service even with nonnegligible BERs.[5-7] We discuss a reconfigurable SRAM architecture to exploit this property and to provide the necessary accuracy during data storage, while performing an accuracy-energy trade-off when storing images.[7]

Parametric failures in SRAM cell and array

A failure in an SRAM cell (see Figure 1) is mainly due to a mismatch in the device parameters (channel length L, channel width W, and/or V_{TH}) of different transistors (intradie) in the cell. The principal source of the device mismatch is the intrinsic fluctuation of different transistors' V_{TH} due to RDFs.[1] The parametric failures in an SRAM cell result from the following:[1]

- *Read failures*. These parametric failures occur when the cell content flips during reading. Read failures increase when the difference is reduced between the voltage rise at the node storing 0 while reading (read-disturb voltage) and the trip point of the inverter associated with the node storing 1.
- *Write failures*. These failures occur when the node storing 1 cannot be discharged (through the access transistors) below the trip point of the inverter associated with the node storing 0 within the time interval for which the wordline remains high.
- *Access failures*. These failures occur when the voltage difference between the two bitlines (bit differential) during the sense amplifier's firing drops below the sense amplifier's offset voltage.
- *Hold failures*. At a lower supply voltage V_{DD}, NMOS leakage causes the node storing 1 to decrease from V_{DD} (which is enhanced by a weak PMOS). If this voltage becomes lower than the trip point of the inverter storing 0, the cell flips when it is in standby mode.

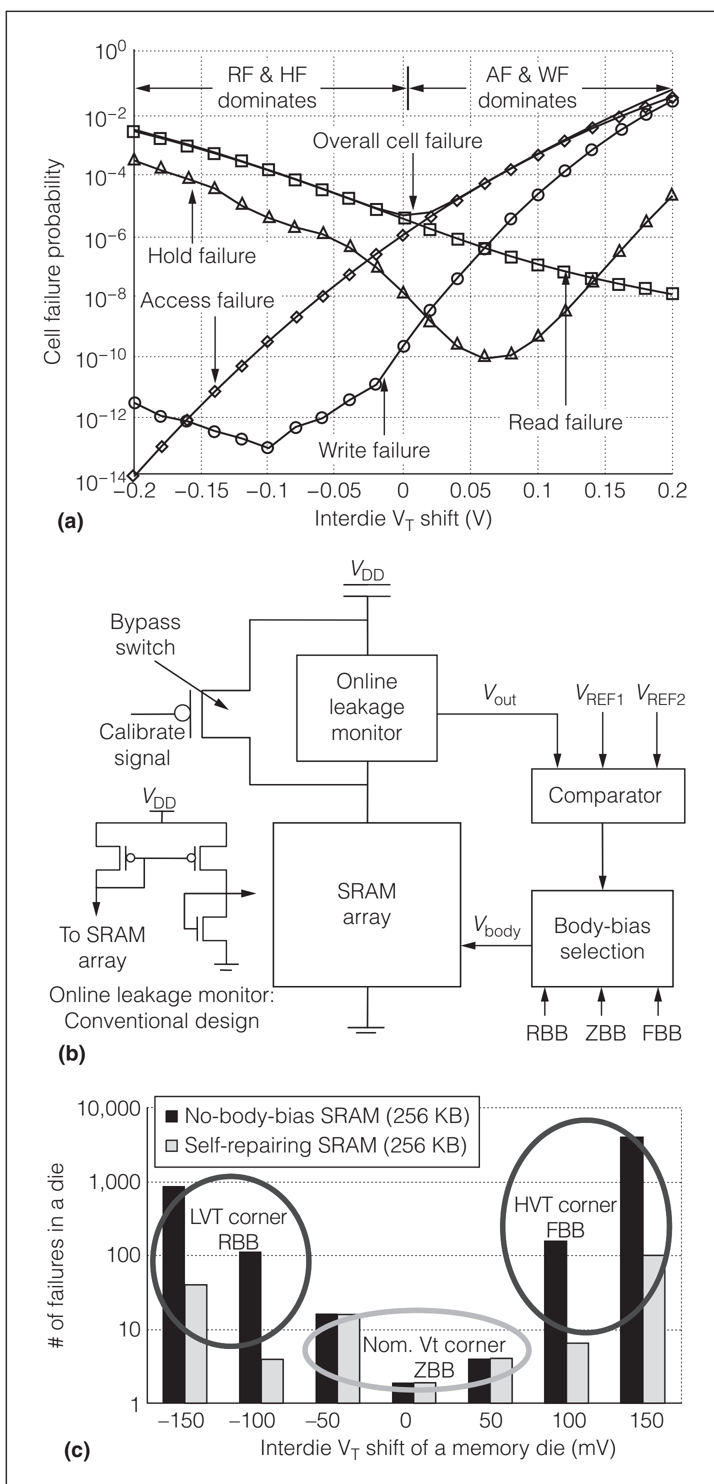

Figure 2. Adaptive repair of SRAM: cell failure probability with interdie threshold voltage V_{TH} shift (a), self-repairing array (b), and reduction in number of failures in a 256-Kbyte self-repairing array (c). (AF: access failure; FBB: forward body bias; HF: hold failure; HVT: high-V_{TH} process corner; LVT: Low-V_{TH} process corner; RBB: reverse body bias; RF: read failure; WF: write failure; ZBB: zero body bias.)

If any cell in a column fails because of any of these failure events, the column is replaced by an available redundant column. If the number of faulty columns is larger than the number of redundant columns, the SRAM array fails (*memory failure*).[1]

Postsilicon adaptive repair of SRAM

The primary cause of parametric failures in SRAM arrays is local device variations. The local nature of these faults makes it impossible to correct every faulty cell in a faulty array. The available redundancy can help mask the effects of a certain number of faults. The aim of postsilicon adaptive repair of a faulty SRAM array is to reduce the total number of faults to a level such that they can be masked by available redundancy (see Figure 1).[2]

Die-to-die variation and adaptive repair

We first analyze the effect of global die-to-die variations on the different types of failures.[2] The lower device V_{TH} reduces the difference between the read-disturb and trip voltages. It also increases the leakage of the NMOS devices. Consequently, the read and hold failures increase for the SRAM arrays shifted to the low-V_{TH} process corners (see Figure 2a). Reducing the access transistors' current drive increases the access and write failures for the SRAM arrays shifted to the high-V_{TH} process corners. Hence, the overall cell failure increases at both low- and high-V_{TH} corners and is minimal for arrays in the nominal corner.

Adaptive repair of SRAM arrays aims to identify the global process corners of an SRAM array and apply proper correction mechanisms to reduce the dominant types of failures in that corner. One method to accomplish this goal is to apply an adaptive body bias (ABB). A global positive bias can be applied to all the cells in SRAM arrays shifted to high-V_{TH} corners. This can partially compensate for high global V_{TH}, and hence reduce access and write failures. Similarly, a global negative bias can be applied to SRAM arrays shifted to low-V_{TH} corners. This can help increase the global V_{TH}, and hence reduce read and hold failures. Because access and write failures dominate in high-V_{TH} process corners, whereas read and hold failures dominate in low-V_{TH} process corners, adaptive body bias can reduce the total number of failures in all process corners.

Self-repair in SRAM arrays

We've demonstrated the concept of adaptive repair in a 256-Kbyte SRAM array designed in predictive

70-nm technology. We designed a self-repairing SRAM array that monitors the leakage of the total SRAM array to determine the interdie corner (see Figure 2b).[2] On the basis of the sensed leakage, we can apply a forward, reverse, or zero body bias to reduce the total number of failures. We first optimize the size of the transistors in the SRAM cell to minimize the cell failure probability of considering no (0 mV) interdie shift in the threshold voltage. This optimization ensures that the number of faulty cells are very low at the nominal interdie corner (i.e., when the interdie V_{TH} shift is negligible). Without adaptive repair, the designed SRAM array has a large number of failures at low- and high-V_{TH} interdie corners. Applying reverse body bias (RBB) at low-V_{TH} corners and forward body bias (FBB) at high-V_{TH} corners significantly reduces the number of failures at both low and high interdie V_{TH} corners (see Figure 2c, in which the self-repairing SRAM has an RBB of −0.3 V and an FBB of +0.3 V). The measurement results from a 128-Kbyte SRAM test chip in CMOS at 130 nm verify the effectiveness of adaptive repair in SRAM. (Details regarding the test chip design and measurement results are available elsewhere.[2])

An important factor in the context of adaptive repair using ABB is that the body bias modifies the standard deviation of V_{TH} variation. This is demonstrated in Figure 3, which shows the measurement results from a test chip designed specifically to extract the local mismatch in the devices' threshold voltage at the 130-nm technology node.[8] The test chip measurement results clearly show that RBB increases local device variability, whereas FBB reduces it. Designers must consider this effect while designing body-bias-based self-repair schemes for SRAM.

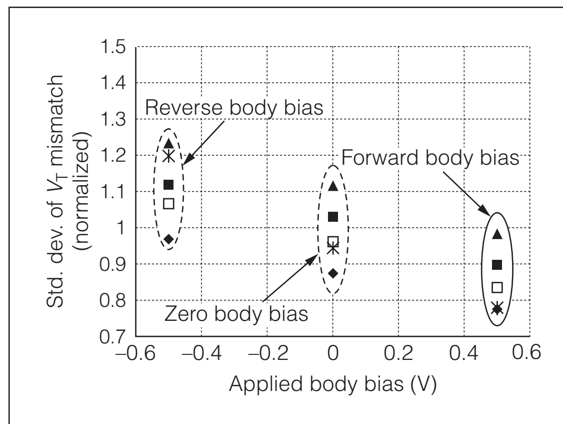

Figure 3. Test-chip measurement result for 130-nm CMOS, showing the effect of body bias on local device variability.

Related work on postsilicon adaptive repair of SRAMs

The uncorrelated global variations in PMOS and NMOS impact the dependence of parametric failures on global corners. For example, write failures are maximal at low-V_{TH} PMOS and high-V_{TH} NMOS corners, whereas read failures are minimal at those corners. Similarly, at high-V_{TH} PMOS and low-V_{TH} NMOS corners, read failures are maximal, whereas write failures are minimal. Hence, more-efficient repair methods are necessary to adapt to global corners of both NMOS and PMOS. In one approach, Yamaoka et al. have proposed to separately measure PMOS and NMOS leakage in SRAM arrays to determine their individual global corners, and accordingly apply individual PMOS and NMOS body biases.[9] Alternatively, Mojumder et al. have proposed a different approach for implementation of adaptive repair.[10] They use an SRAM-cell-based test circuit to directly measure the global read and write margin corners for every die. For dies shifted to worse global read margin corners, they use a higher cell supply and lower wordline voltage (a difference of approximately 100 to 200 mV is sufficient). For dies shifted to worse global write margin corners, they use lower cell supply and higher wordline voltages.

Adaptive accuracy-energy trade-off in SRAM

The adaptive repair technique we've presented can help provide high-yield functional SRAM arrays for multimedia systems during data storage at nominal voltages. Here, we discuss a methodology for run-time adaptation of an accuracy-energy trade-off in SRAM. The array remains at a high accuracy and nominal power mode during data storage. During image storage, it can be adapted to an acceptable accuracy and low-power mode.

Effect of voltage scaling on failures and image quality

Figure 4 shows the effects of operating-voltage scaling (cell supply, bitline, and wordline) on parametric failures. We designed the SRAM cell in a 70-nm predictive model to achieve an access speed of approximately 1 GHz. We assume the frequency of operation for multimedia applications is about 250 MHz.

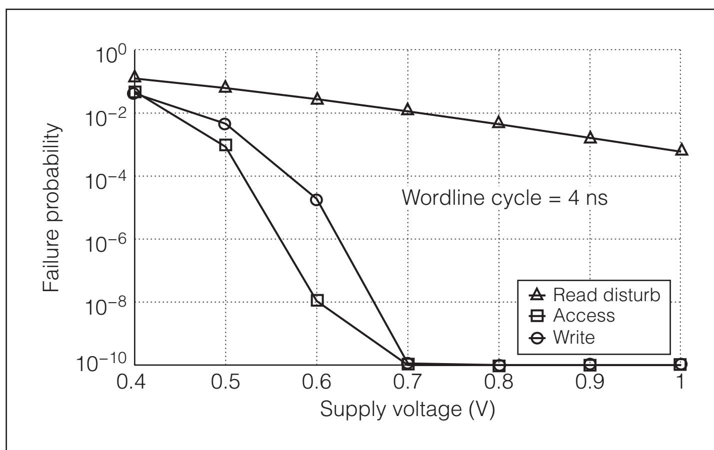

Figure 4. Effect of supply voltage on SRAM cell failures.

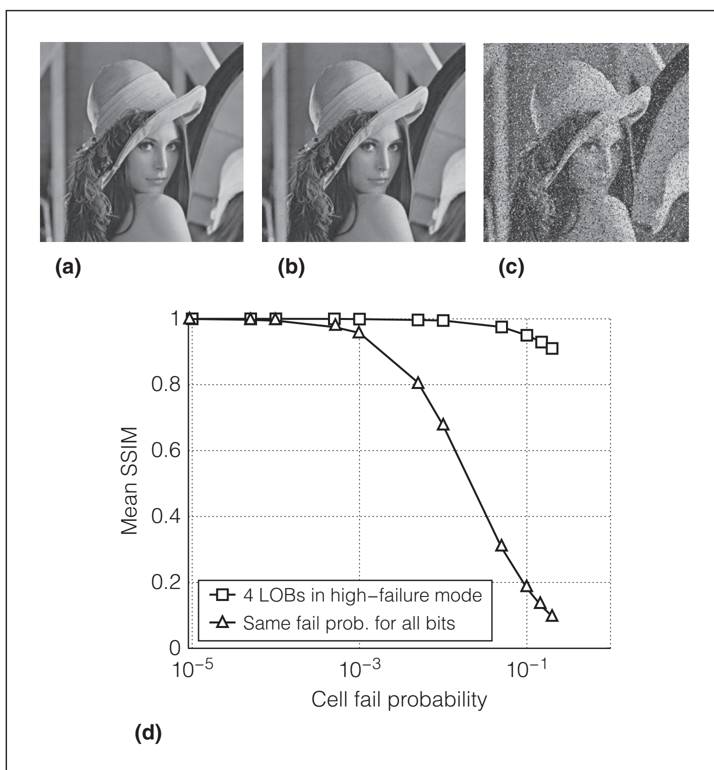

Figure 5. Effect of different bit-error rates (BERs) of lower-order bits (LOBs) and higher-order bits (HOBs): original photo (a); same photo with 4 LOBs operating at $P_f = 10^{-2}$, and 4 HOBs operating at $P_f = 10^{-5}$ (b); and same photo with all bits operating at $P_f = 10^{-2}$ (i.e., both 4 LOBs and 4 HOBs) (c). (P_f is the total cell failure probability.)

The operation at below-maximum frequency results in very low access-failure and write-failure probabilities at nominal conditions. However, at a lower operating voltage, even for a 250-MHz operation, access and write failures become appreciable. The increased failures manifest themselves as higher BERs and increased image quality degradation. We estimate the image quality degradation by comparing the mean structural similarity (MSSIM) index of the original and modified image. If two images are visually identical, the MSSIM is 1.[11] A degradation in the image quality results in a lower MSSIM. It's well-known that, in typical multimedia applications, an image pixel's lower-order bits (LOBs) are more tolerant to noise than its higher-order bits (HOBs). If we scale only the voltages of the LOBs (i.e., we allow only the BERs of LOBs to increase), the net effect on the image quality degradation is smaller (see Figure 5). Hence, we can achieve appreciable power savings with minimal image quality degradation by operating LOBs at a low voltage and HOBs at a nominal voltage.

Runtime adaptation of the accuracy-energy trade-off

On the basis of these observations, we present a runtime adaptation of the accuracy-energy trade-off. The main idea is to consider only two voltage levels: nominal (about 1 V) and low (about 0.4 V).[7] The operating voltages for different bits are adapted during runtime to either of these levels. The number of bits in low-voltage mode can be adapted using a hardware-based reconfiguration framework. For the highest accuracy, all bits remain at a nominal voltage while storing data. During image processing, depending on accuracy requirements, the number of low-voltage bits can be increased. We refer to this approach as *spatial voltage scaling*.

Architecture of the adaptive SRAM

Figure 6 shows the overall SRAM architecture to implement the runtime adaptive accuracy-energy trade-off.[7] We consider a conventional SRAM array with column multiplexing, where particular bits of different words are grouped together, which we refer to as a multiplexer (mux) group. A single read/write circuit is used for each mux group. All the cells in a mux group, along with the corresponding precharge device and write drivers, share the same power supply. A voltage-switching network (the top PMOSs in Figure 6) can change the cell supply, bitline, and write voltages for the array. A reconfigurable wordline structure ensures that the wordline voltage for a mux group is the same as the cell supply voltage. In this structure, the local wordlines (LWLs) of a row of

cells in a mux group are connected together and driven by reconfiguring inverters. The LWLs of different mux groups are disconnected. The inputs of all the reconfiguring inverters are connected to the global wordline (GWL). The power supplies of these inverters are connected to the supply voltage of that mux group. For a selected row, GWL is 0, which makes the LWL voltage (which is 1) the same as the cell supply and bitline voltages. Each reconfiguring inverter is sized to minimize its impact on the wordline delay at nominal operation, while keeping area overhead under control (approximately 6%).

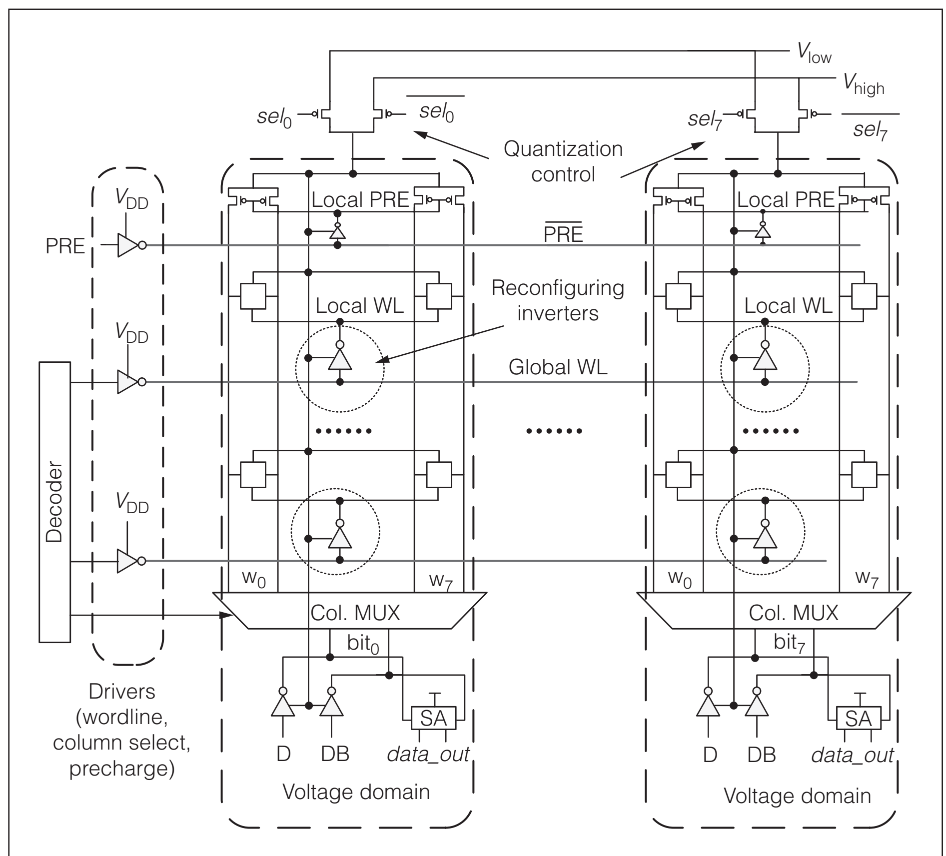

Figure 6. Adaptive accuracy-energy trade-off in SRAM. (PRE: precharge signal.)

The voltage reconfiguration network consists of two PMOSs: one connected to the low voltage, and the other connected to the high voltage. The gate voltage of the PMOSs for the *j*th bit is controlled by the *j*th *select* signal (sel_j). To reconfigure a bit to the low voltage, sel_j is set to 0. The switching PMOSs in the voltage reconfiguration network for the array and precharge devices are sized to ensure a low voltage drop and short reconfiguration time (the time required to change a bit's voltage level). Our analysis shows that if the PMOSs in the reconfiguration network are sized to 5× larger than the cell PMOSs, the area overhead for 256 cells per column is limited to about 0.4%, and the reconfiguration time to about 4 ns (1 cycle for a 250-MHz operation). If we consider read and write operations of a 256 × 256 pixel image, even if reconfiguration is performed once for every image, only one additional clock cycle per image is required for reconfiguration.

Simulation results for image storage

We evaluated the effect of the adaptive accuracy-energy trade-off on a standard 8-bit gray-scale test image suite.[7] The read access, write access, and active leakage power considering the bitline and wordline (including the reconfiguring inverters) switching energy was considered in the power computation. We computed the power savings with reference to a regular array (all bits at a nominal voltage of 1 V). To simulate the array's error characteristics, we performed system-level fault simulation. First, we performed circuit simulations to estimate the distributions of the read margin, write margin, and access time of cells at different voltages. Next, we created a random array instance, which generates the locations of faulty cells by randomly associating cell margin values (generated from the estimated distributions) with each cell and comparing them against a target. During image mapping, a bit is flipped if it maps to a faulty cell location.

As Figure 7 shows, significant power savings can be achieved via graceful degradation of image quality for different images.[7] Using four low-voltage bits ($L_{bit} = 4$), a power savings of about 45% can be achieved compared to a regular array with a 10% reduction in quality. Power savings is 20% higher than the savings achievable by reducing the voltage of all bits at the same degradation level. Increasing the number of low-voltage bits degrades quality with an increase in power savings over the regular array (see Figure 7c). Simultaneously reconfiguring V_{low}

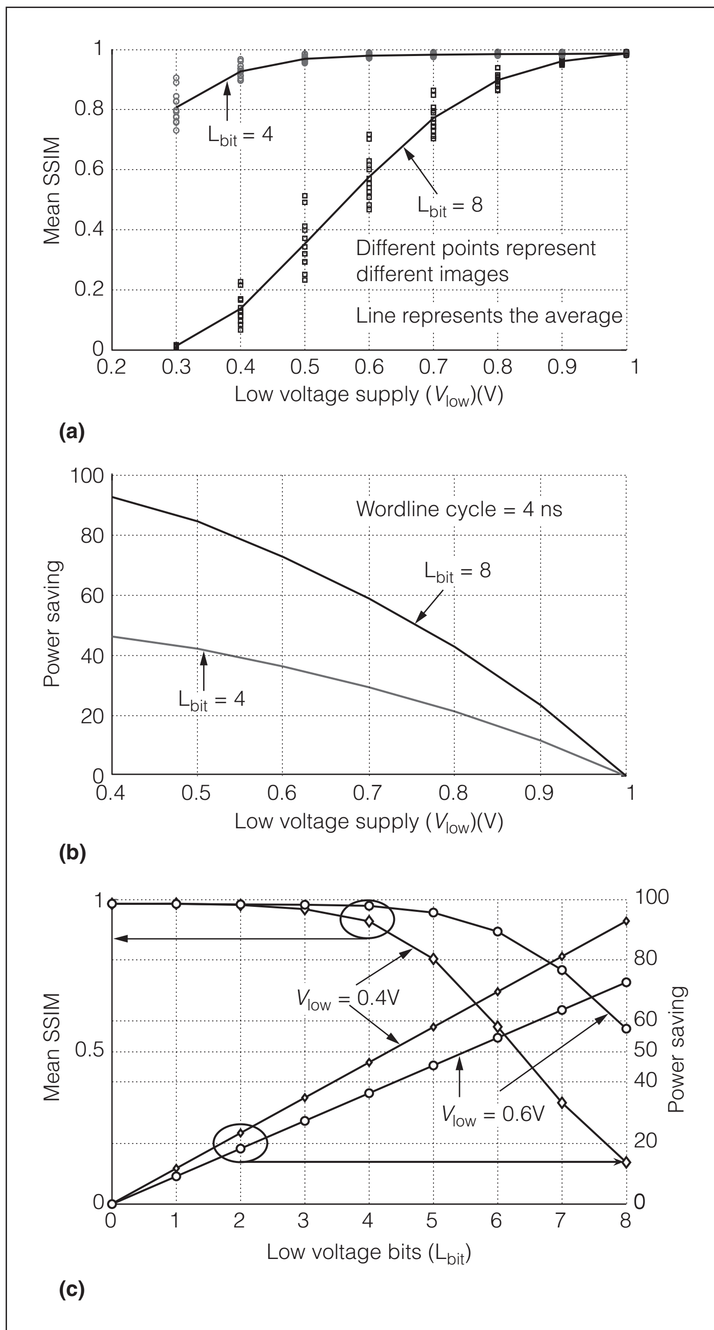

Figure 7. Effect of supply voltage scaling on image quality (a) and power (b), and effect of the number of LOBs on image quality and power (c).

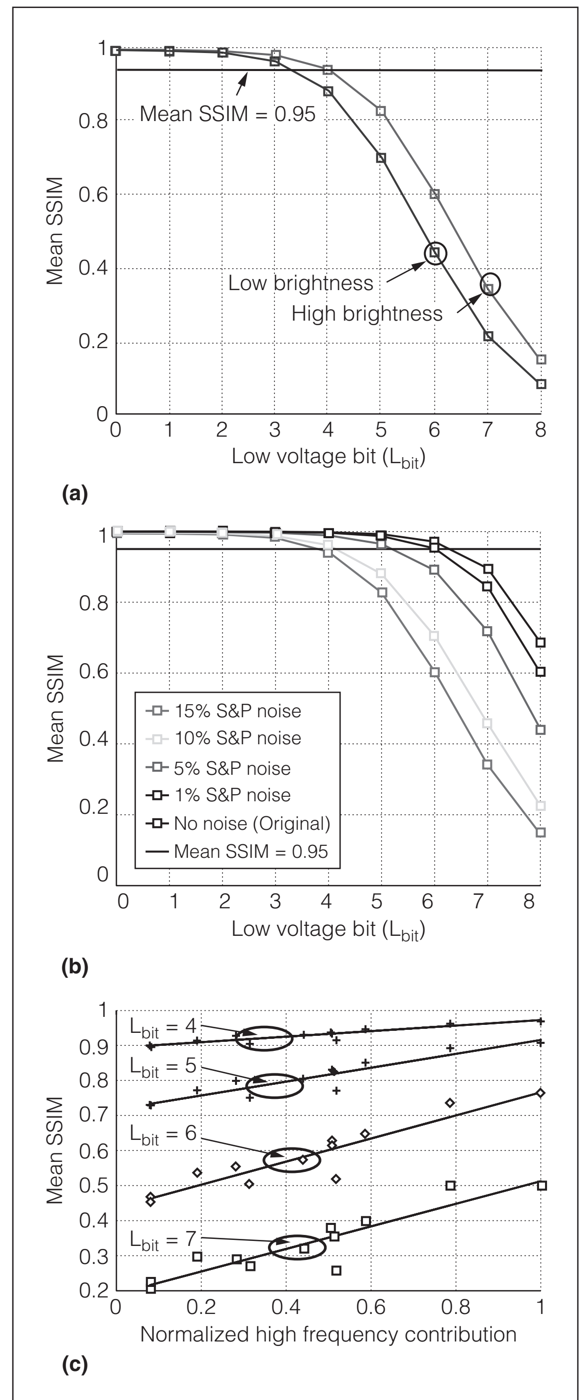

Figure 8. Effect of image characteristics on brightness (a), noise (b), and frequency characteristics (c). (MSSIM: mean structural similarity; S&P: salt and pepper.)

and L_{bit} can achieve an even better energy-accuracy trade-off and higher power savings.

Effect of image properties on the adaptation

The effectiveness and permissible extent of the adaptive energy-accuracy trade-off depends on the image's properties. For example, images with less brightness suffer more errors under reconfiguration (see Figure 8a). Similarly, images with low noise levels (salt-and-pepper noise) maintain a target quality with a higher L_{bit} (see Figure 8b). Finally, we evaluated the

correlation of frequency characteristics with quality degradation under reconfiguration. We estimated the high-frequency components in different images by employing a 2D wavelet decomposition of each image.

Figure 8c shows that images with a higher contribution of high-frequency components can maintain a desired MSSIM for a higher L_{bit}. This analysis also shows the power advantages of the proposed reconfigurable architecture over a fixed L_{bit} designed on the basis of a worst-case corner. For example, consider the effect of noise. To account for worst-case (assume 15%) noise, the design time choice of L_{bit} could be 4. Using the proposed architecture, L_{bit} could be reconfigured to 6 under better-than-worst-case conditions, leading to an additional 15% power savings.

Application to optical flow

Optical flow is a pattern of apparent visual motion created between an observer and the scene.[12] Estimation of optical flow has been used in many image-processing applications, such as pattern recognition and computer vision. The typical implementation of optical flow involves calculating the deformations of two adjacent image frames using motion-flow vectors for each pixel.

Considering the optical-flow problem, we can evaluate the effectiveness of the reconfigurable memory architecture in performing an adaptive accuracy-energy trade-off. To perform this evaluation, we coupled an image-processing system in Matlab to the system-level fault simulator. We chose a frame sequence containing 30 seconds of 8-bit gray-scale video for experimentation. This video involved feed from a crowded public area, along with motion occurring at different velocities and in various directions. We modified the individual video frames according to the fault model, considering different L_{bit} values, and for $V_{low} = 0.4$ and 0.8 V. We processed each variation using the optical-flow system. After processing all the frames, we compared the resulting flow vectors, $\mathbf{V}_x$ and $\mathbf{V}_y$, with those of the original video.

As Figure 9a shows, the comparable average MSSIM over all frames can be achieved with uniform voltage scaling to 0.8 V ($L_{bit} = 8$), or using spatial scaling with $L_{bit} = 4$ and $V_{low} = 0.4$ V. Reconfiguring L_{bit} provides further energy-accuracy trade-offs. However, image storage is only one component of the embedded optical-flow system. Image quality declines over time because of pixel faults propagating both in a temporal fashion between frames and in an intraframe fashion between flow operations. We determined the mean percent error between optical-flow vectors for the original video and those acquired using varying numbers of low-voltage bits and voltage values (see Figure 9b). In terms of overall flow quality, up to four low-voltage bits was acceptable when processed through later flow-vector grouping stages. Another interesting finding is that the error increased more uniformly with scaling of L_{bit}. Although there is a significant difference between the average MSSIM for $V_{low} = 0.4$ V and $V_{low} = 0.8$ V for $L_{bit} = 6$, the mean percent error in flow error is far smaller. Figure 9b also helps explain the effect of *uniform voltage scaling* (when voltage levels of all bits are reduced simultaneously) and *spatial voltage scaling* (when the voltage levels of only LOBs are reduced using the

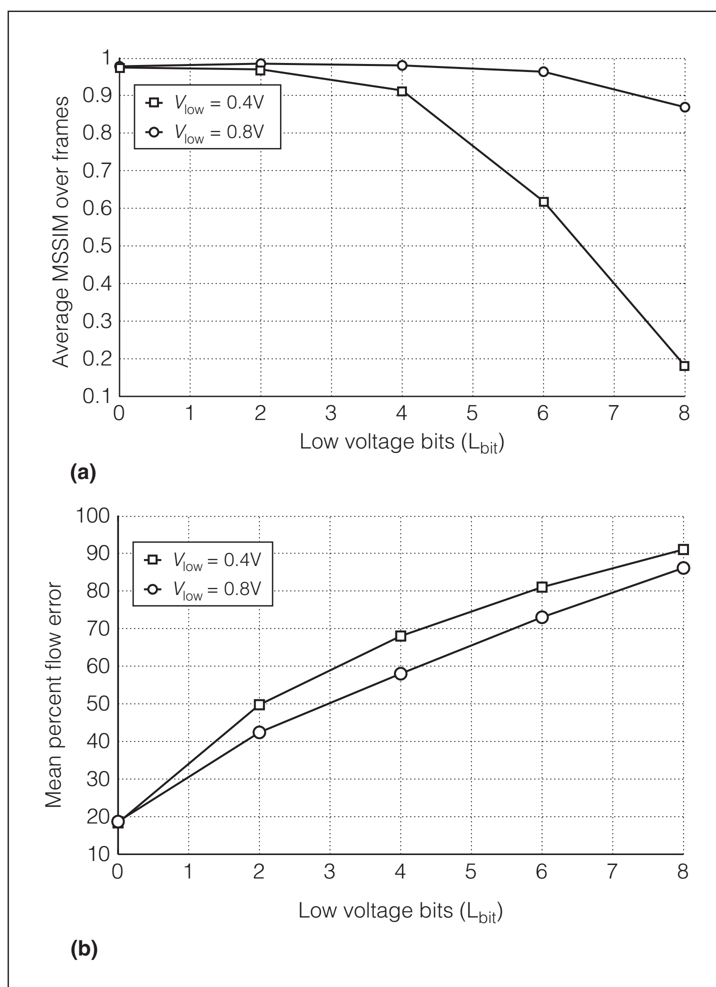

Figure 9. Impact of proposed array on optical flow: average mean structural similarity (MSSIM) over frames (a) and average error (b).

reconfiguration functionality of the proposed SRAM array) on error propagation. Although uniform scaling to 0.8 V and spatial scaling with $L_{\text{bit}} = 4$ and $V_{\text{low}} = 0.4$ V provide a comparable MSSIM, the mean error is significantly large with uniform voltage scaling.

Related work on the energy-accuracy trade-off in SRAMs for multimedia applications

The reduction of memory power in multimedia systems is a well-known problem. Djahromi et al.[5] have performed aggressive voltage scaling in SRAM arrays for multimedia and communication applications, exploiting these applications' inherent error tolerance. However, aggressive voltage scaling of all bits in a pixel (uniform voltage scaling) increases the error rates in all bits and leads to faster degradation of image quality with voltage scaling. To counter this effect, researchers have recently explored spatial voltage scaling. In a recent study, Yamaoka et al. showed the advantage of using different voltages across a word in improving the effective yield of video memories.[9]

Our present work has shown that simultaneous considerations of technology, circuit, system, and application properties can lead to new opportunities for designing high-yield, robust, and low-power SRAM. Although this article illustrates the promise of circuit-system-application co-analysis and co-design, future research needs to consider how to effectively incorporate such methodologies in the SRAM design flow. The challenges that remain are in modeling the correlation between error characteristics and energy in SRAM and developing effective statistical analysis and design techniques to modulate this correlation. The effective modulation of this correlation through circuit, architecture, and system-level postsilicon adaptation is the key to designing high-yield, robust, and low-power SRAM in future technologies. ■

Acknowledgments

This work was supported in part by the National Science Foundation under grant CCF # 0916083.

References

1. S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems,* vol. 24, no. 12, 2005, pp. 1859-1880.
2. S. Mukhopadhyay et al., "Design of a Process Variation Tolerant Self-Repairing SRAM for Yield Enhancement in Nanoscaled CMOS," *IEEE J. Solid State Circuits,* vol. 42, no. 6, 2007, pp. 1370-1382.
3. K. Zhang et al., "A 3-GHz 70MB SRAM in 65nm CMOS Technology with Integrated Column-Based Dynamic Power Supply," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), vol. 1, IEEE Press, 2005, pp. 474-475, 611.
4. S. Ohbayashi et al., "A 65 nm SoC Embedded 6T-SRAM Design for Manufacturing with Read and Write Cell Stabilizing Circuits," *Proc. Symp. VLSI Circuits,* IEEE Press, 2006, pp. 17-18.
5. A.K. Djahromi et al., "Cross Layer Error Exploitation for Aggressive Voltage Scaling," *Proc. 8th IEEE Int'l Symp. Quality Electronic Design* (ISQED 07), IEEE CS Press, 2007, pp. 192-197.
6. K. Yi et al., "A Partial Memory Protection Scheme for Higher Effective Yield of Embedded Memory for Video Data," *Proc. 13th Asia-Pacific Computer Systems Architecture Conf.* (ACSAC 08), IEEE CS Press, 2008, pp. 273-278.
7. M. Cho et al., "Reconfigurable SRAM Architecture with Spatial Voltage Scaling for Low Power Mobile Multimedia Applications," to be published in *IEEE Trans. Very Large Scale Integration (VLSI) Systems.*
8. S. Mukhopadhyay et al., "Statistical Characterization and On-Chip Measurement Methods for Local Random Variability of a Process Using Sense-Amplifier-Based Test Structure," *Proc. IEEE Int'l Solid State Circuits Conf.* (ISSCC 07), IEEE Press, 2007, pp. 400-401, 611.
9. M. Yamaoka et al., "65 nm Low-Power High-Density SRAM Operable at 1.0 V under 3σ Systematic Variation Using Separate Vth Monitoring and Body Bias for NMOS and PMOS," *IEEE Int'l Solid State Circuits Conf.* (ISSCC 08), IEEE Press, 2008, pp. 383-385, 622.
10. N.N. Mojumder et al., "Self-Repairing SRAM Using On-Chip Detection and Compensation," *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 1, 2010, pp. 75-84.
11. Z. Wang et al., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing,* vol. 13, no. 4, 2004, pp. 600-612.
12. J. Schlessman et al., "Hardware/Software Co-design of an FPGA-Based Embedded Tracking System," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPRW 06), IEEE Press, 2006, pp. 123-133.

Minki Cho is pursuing a PhD in electrical and computer engineering from the Georgia Institute of Technology, Atlanta. His research interests include design

of accuracy-aware low-power systems, self-adaptive system design for multi- and many-core processors, and design for 3D integration. He has an MS in electrical and computer engineering from the Georgia Institute of Technology, Atlanta. He is a student member of IEEE.

Jason Schlessman is pursuing a PhD in electrical engineering at Princeton University. His research interests include embedded computer vision, parallel computing, and distributed smart cameras. He has a master's degree in electrical engineering from Princeton University.

Hamid Mahmoodi is an assistant professor in the School of Engineering at San Francisco State University. His research interests include low-power, reliable, and high-performance circuit design for nanoscale technologies. He has a PhD in electrical and computer engineering from Purdue University. He is a student member of IEEE.

Marilyn Wolf is the Rhesa "Ray" S. Farmer Jr. Distinguished Chair of Embedded Computing Systems and Georgia Research Alliance Eminent Scholar in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta. Her research interests include embedded computing, computer vision, and VLSI systems. She has a PhD in electrical engineering from Stanford University. She is a Fellow of IEEE and the ACM, and a Golden Core member of the IEEE Computer Society.

Saibal Mukhopadhyay is an assistant professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta. His research interests include low-power variation-tolerant VLSI circuits and systems, embedded systems, and 3D integration. He has a PhD in electrical and computer engineering from Purdue University. He is a member of IEEE.

■ Direct questions and comments about this article to Saibal Mukhopadhyay, Klaus Advanced Computing Bldg., #2366, 266 Ferst Dr., Atlanta, GA 30332; saibal@ece.gatech.edu.

cn **Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.**

Call for Articles

IEEE Software seeks practical, readable articles that will appeal to experts and nonexperts alike. The magazine aims to deliver reliable information to software developers and managers to help them stay on top of rapid technology change. Submissions must be original and no more than 5,400 words, including 200 words for each table and figure.

IEEE Software

Author guidelines: www.computer.org/software/author.htm
Further details: software@computer.org
www.computer.org/software