Fundamental Bounds on Power Reduction during Data-Retention in Standby SRAM†

A. Kumar, H. Qin, P. Ishwar*, J. Rabaey, and K. Ramchandran

EECS, University of California,
Berkeley, CA – 94720
Email:{animesh,huifangq,jan,kannanr}@eecs.berkeley.edu

*ECE, Boston University
Boston, MA – 02215
Email: pi@bu.edu

***Abstract*— We study leakage-power reduction in standby random access memories (SRAMs) during data-retention. An SRAM cell requires a minimum critical supply voltage (*DRV*) above which it preserves the stored-bit reliably. Due to process-variations, the intra-chip *DRV* exhibits variation with a distribution having a diminishing tail. In order to minimize leakage power while preserving data reliably, existing low-power design methods use a worst-case standby supply voltage. This worst-case voltage is larger than the highest *DRV* among all cells in an SRAM. In contrast, our approach uses aggressive voltage reduction and counters the ensuing unreliability by an error-control code based memory architecture. Using this approach, we explore fundamental trade-offs between power reduction and redundancy present in the SRAM. We establish fundamental bounds on the power reduction in terms of the *DRV*-distribution using techniques from information theory and algebraic coding theory. For an experimental test-chip *DRV*-distribution in the 90nm CMOS technology, we show that 49% power reduction with respect to (w.r.t.) the worst-case is a fundamental lower bound while 40% power reduction w.r.t. the worst-case is achievable by using a practical algebraic coding scheme. We also study the power reduction as a function of the block-length for low-latency codes since most applications using SRAM are latency constrained. We propose a reliable low-power memory architecture based on the Hamming code for the next test-chip implementation with a predicted power reduction of 33% while accounting for coding overheads.**

I. Introduction

As CMOS technology scales into sub-100nm domain, a number of new challenges unfold in front of the designers. Excessive leakage-power and increasing process-variations are two important design issues observed in the emerging sub-100nm CMOS technology [1]. In this paper, we will focus on these two design aspects of standby static random access memories (SRAMs).

In many chips which include an SRAM module, e.g., sensor network nodes, there are two modes of operation: (i) the *active-mode* in which the SRAM is active for reading and writing, and (ii) the *standby-mode* in which the SRAM retains the data. Since SRAM module occupies a significant portion of the total chip-area, e.g. the CHARM chip in [2], therefore, standby SRAM power is the dominant power consumption factor in applications that are primarily in the standby mode. In CMOS technology, standby power consists of leakage-power which increases with each silicon-technology generation [1]. Thus, for low-power devices, e.g. sensor nodes, standby leakage-power reduction is crucial for device-operation within the scavenging power limit [2].

An effective method to reduce leakage-power is to minimize the supply voltage while ensuring data-retention. Using this approach, it has been shown that any SRAM cell has a critical voltage (called the data retention voltage or *DRV*) at which a stored bit (0 or 1) is retained reliably [3].

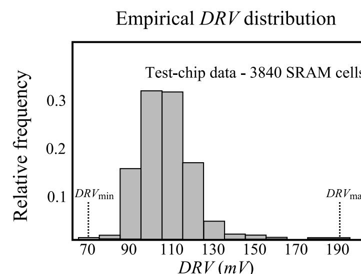

Fig. 1. **Test-chip *DRV*-distribution:** *The experimental intra-chip DRV varies from 70 to 190mV in the 90nm CMOS technology [4]. The worst-case solution for data-retention is a supply voltage of 200mV.*

The intra-chip *DRV* exhibits a distribution due to process-variations. In Fig. 1, we illustrate an experimental test-chip distribution in the 90nm CMOS technology [4]. The *DRV* varies from 70 to 190mV for 3840 SRAM cells. In order to minimize leakage power while preserving data reliably, existing low-power design method uses a worst-case approach — a standby supply voltage larger than the highest *DRV* among all cells in an SRAM is used. For the test-chip distribution in Fig. 1, the worst-case supply voltage is 200mV.

In contrast to the worst-case design, we propose aggressive reduction of the standby supply voltage with error-control coding, thereby ensuring *reliable* data-storage. Using this approach, we show the following main results in this paper:

- We establish fundamental bounds on the power reduction in terms of the *DRV*-distribution using techniques from

†This research was sponsored in part by MARCO, GSRC and supported in part by the NSF under Grant No. CCR-0330514 and Career CCF-0546598. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1-4244-0921-7/07 $25.00 © 2007 IEEE.

information and coding theory. For the test-chip *DRV* distribution in Fig. 1, we show that 49% power reduction w.r.t. the worst-case is a fundamental lower bound while 40% power reduction w.r.t. the worst-case is achievable by using a practical algebraic coding scheme.

- We study the power reduction as a function of the block-length for low-latency codes since most applications using SRAM are latency constrained. We propose a reliable memory architecture based on the Hamming code for the next test-chip implementation with a predicted power reduction of 33% while accounting for coding and latency overheads.

The first result states the fundamental bounds on the power reduction for any given *DRV*-distribution. The second result states that, *while accounting for coding overheads*, a significant portion of the optimum power reduction (33% out of 40%) can be achieved by using a low-latency Hamming code.

Related work: Reliable storage capacity and coding for storage have been studied and proposed in various memory implementations in the literature (e.g., see [5], [6]). Heegard and El Gamal have analyzed the reliable storage capacity for a memory with stuck-at faults or random errors [5]. Our work differs in two important aspects from the existing results: (i) We study power versus redundancy trade-off in standby SRAMs, unlike previous works which studied reliability versus redundancy trade-offs, and (ii) We account for coding and latency overheads in our analysis and results.

Modeling assumptions: We will model the parametric variation of the *DRV* by the observed (discrete) probability distribution $\mu(x), x \in \{70, 80, \ldots, 190\}$ (see Fig. 1). The cumulative distribution function is $F(x) = \sum_{z \leq x} \mu(z)$. Since the available *DRV* data is quantized at a resolution of $10mV$, we will sweep the supply voltage in multiples of $10mV$. A cell will retain the stored data successfully if the supply voltage is strictly greater than the cell's *DRV* voltage. We model the *DRV* as a random but fixed voltage after manufacture.

The *DRV* is obtained by solving current equations in the subthreshold regime [3]. The gate-leakage current can vary with time due to trapping and de-trapping or charges [7]. However, the gate-leakage current and its variations are much smaller at low voltages (around $200mV$) compared to the subthreshold currents. Therefore, *DRV* in the 90nm CMOS process does not depend significantly on gate-leakage, and is approximately constant with time. [1]

Notation: In the rest of the paper, the *standby power* will be called as *power* for brevity. The distribution in Fig. 1 will be referred as $F(x)$. The supply voltage will be represented by v_S. The symbol $\mathbb{P}$ will be used for the probability of a set with respect to the distribution $F(x)$. Vectors like $(x_1, x_2, \ldots, x_n)$ will be represented as x_1^n. Finally, $h(t) = -t \log_2 t - (1-t) \log_2 (1-t)$ stands for the binary entropy function [9].

Organization: We present the proposed standby SRAM architecture in Sec. II. We discuss fundamental bounds on power reduction in Sec. III. We also discuss the power reduction for a few known family of codes in the same section. Finally, we conclude in Sec. IV.

[1]Gate-leakage decreases exponentially with the supply voltage [8]. However, the subthreshold leakage decreases linearly with the supply voltage (see Fig. 4).

II. Standby SRAM: Low-power architecture

We will present the SRAM cell retention model followed by our proposed standby SRAM architecture. The description of the retention model is important for understanding the architecture.

***DRV* based retention model:** For each SRAM cell, there is a data-retention-voltage (*DRV*), above which the stored data bit (0 or 1) is stored reliably [3]. However, if the supply voltage is lowered below the *DRV*, then the stored bit degenerates to a preferred digital state $S \in \{0, 1\}$ [3].

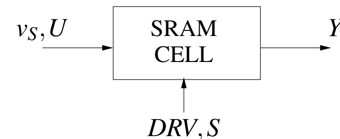

Fig. 2. *DRV* **based retention model:** *The SRAM cell has two statistically independent parameters: (i) a time-invariant positive continuous-valued threshold-voltage called DRV, and (ii) a binary bias-state $S \in \{0, 1\}$. The inputs are the supply voltage v_S and a bit $U \in \{0, 1\}$ to be stored. The output is $Y = U$ if $v_S > DRV$ and S otherwise.*

We capture these features of an SRAM cell in the following mathematical model (see Fig. 2). The cell has two statistically independent parameters: (i) a time-invariant, positive and continuous-valued threshold-voltage *DRV*, and (ii) an equally likely binary bias-state $S \in \{0, 1\}$. The inputs to the cell are the supply voltage v_S and a bit $U \in \{0, 1\}$ to be stored. The retention model for the SRAM cell is as follows:

$$\begin{aligned} Y &= U \quad \text{if} \quad DRV < v_S, \\ &= S \quad \text{if} \quad DRV \geq v_S, \end{aligned} \tag{1}$$

where $Y \in \{0, 1\}$ is the output bit. If $v_S \leq DRV$, then there is a *DRV failure*. This digital abstraction is sufficient for this paper. We discuss the proposed standby SRAM architecture next.

Standby SRAM low-power architecture: Let the standby supply voltage be $v_S \in \{0, 10, \ldots, 200\}$ in mV at $10mV$ resolution. The worst-case solution is to use $v_S = 200mV$ in which every cell retains the data reliably (see Fig. 1).

In contrast, we propose an error-protected SRAM as follows. Let $B_1^k = (B_1, B_2, \ldots, B_k)$ be the data vector to be stored. Using a suitable error-control code, B_1^k is encoded into U_1^n and stored in n memory cells ($n \geq k$). Cells have i.i.d. pairs of independent DRV, S realization. [2] The jth stored bit is stuck-at S_j if $DRV_j \geq v_S$, otherwise U_j is successfully retained. At the end of standby, Y_1^n is decoded to $\widehat{B}_1^k$. Let $0 \leq i \leq 2^k - 1$ be the integer representation of B_1^k. For any error-control code, the voltage v_S is chosen such that the outage probability,

$$\mathbb{P}(\text{outage}) = \mathbb{P}\left(\exists i, \text{ such that } \widehat{B}_1^k \neq i | B_1^k = i\right), \tag{2}$$

[2]The assumption that *DRV* across cells are independent is a worst-case assumption as discussed at the end of Sec. III.

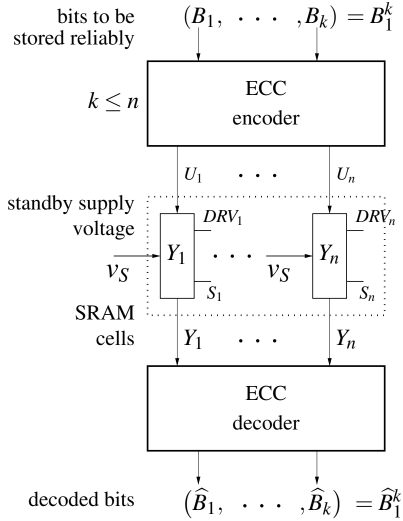

Fig. 3. **Standby-SRAM architecture:** *Let B_1^k be the data vector to be stored. Then B_1^k is encoded into U_1^n and stored in n memory cells. The jth stored bit is stuck-at S_j if $DRV_j \geq v_S$, otherwise U_j is read-out. The decoder reads Y_1^n and outputs $\widehat{B}_1^k$. The voltage v_S is selected such that $\mathbb{P}(outage)$ is negligible (see (2)).*

is negligible. This condition ensures that an n-bit row of memory stores all input words B_1^k with high reliability. The outage failures will be corrected by testing and row-redundancy [10].

Since v_S is a free variable, power per useful-bit (or other performance metrics) can be *optimized* over its range. For an outage of ε, we define the power per bit as,

$$\mathcal{P}_\varepsilon(v_S) := \frac{1}{k} \cdot \big(\text{Total standby power}\big). \tag{3}$$

If ε can be made arbitrarily small by choosing $n \longrightarrow \infty$, then the power per bit function will be called as $\mathcal{P}(v_S)$. The dependence of power per bit on v_S will be established next.

III. Bounds on power reduction

In this section, we will derive DRV-distribution dependent fundamental bounds on the power per bit $\mathcal{P}(v_S)$. We first discuss the standby power dependence on the supply voltage.

A. Power dependence on the supply voltage

Let T_s be the standby duration. Let E_C be the average encoder-decoder computational energy (over codewords B_1^k) of the error-control code C. The total standby power is,

$$P_T = P_L + \frac{E_C}{T_s}, \tag{4}$$

where P_L is the total leakage-power.

The leakage-current in the $100-200mV$ range is approximately linear in the supply voltage, i.e., $I_L = Gv_S$, where G is a constant. This is confirmed by our test-chip leakage-current measurements (see Fig. 4). Thus, the power per bit of the SRAM cell is,

$$\mathcal{P}_\varepsilon(v_S) = \frac{n}{k} \cdot Gv_S^2 + \frac{E_C}{kT_s}, \tag{5}$$

where the code C has an outage ε.

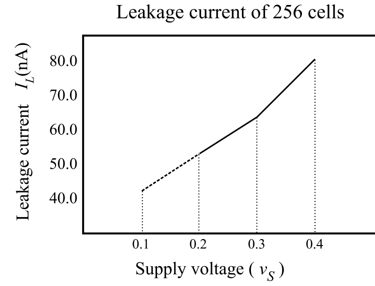

Fig. 4. **Average leakage-current:** *The measured leakage-current for 256 SRAM cells is shown as a function of the supply voltage. In the range $100-200mV$, the leakage-current is approximately linear.*

B. Fundamental bounds on the power reduction

For deriving bounds, we note the following important points: (i) For $T_s \longrightarrow \infty$, i.e., when the standby time is much larger than the encoding-decoding time, the coding energy overhead becomes negligible. Under this condition, the standby power is minimum and will be considered first, (ii) We will account for the coding and latency overheads (see Sec. III-C and Sec. III-D) after establishing fundamental benchmark asymptotic bounds, and (iii) The outage $\varepsilon > 0$ can be made arbitrarily small in an asymptotic setting, i.e., when $n \longrightarrow \infty$. The DRV-failure probability for an SRAM cell is given by,

$$p(v_S) = \sum_{z \geq v_S} \mu(z). \tag{6}$$

Then, we have the following theorem:

Theorem 3.1: Let v_S be the standby supply voltage and $p(v_S)$ be as in (6). For each voltage $v_S : p(v_S) < 0.25$, the minimum power per bit satisfies,

$$\frac{Gv_S^2}{1-h\big(p(v_S)/2\big)} \leq \mathcal{P}(v_S) \leq \frac{Gv_S^2}{1-h\big(2p(v_S)\big)}, \tag{7}$$

where G is a constant. The reduction in $\min_{v_S} \mathcal{P}(v_S)$ w.r.t. the worst-case is between 40% and 49%. ■

The bounds on $\mathcal{P}(v_S)$ are derived using ideas from Information theory [9, Ch. 8] and error-control code theory [11], respectively. We omit the details for brevity. Observe that the denominator $1-h\big(p(v_S)/2\big)$ and the numerator v_S^2 increase as v_S increases. When v_S is small (around $70mV$), the increase in denominator term is rapid compared to the numerator. The trend reverses for large v_S (around $200mV$). Thus, we see that the optimum power per bit is achieved for an intermediate values of v_S. Similar argument holds for the upper bound.

Fig. 5 illustrates the power per bit bounds as a function of $p(v_S)$. The minimum value of the upper bound and the lower bound are 40% and 49% less than the worst-case, respectively.

C. Power reduction with low-latency codes

After the standby mode, practical SRAM design requires data-output within a latency of a few clock cycles. We explore power per bit reduction as a function of the block length

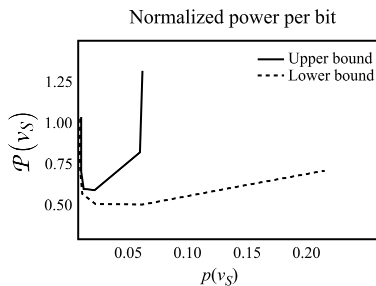

Fig. 5. **Bounds on $\mathcal{P}(v_S)$:** *Power per bit bounds are plotted as functions of the DRV-failure rate $p(v_S)$. The minima of upper and lower bounds are 40% and 49% lower than the worst-case.*

n for Hamming and Reed Muller codes. We will study the power reduction at an outage of $\varepsilon = 0.01$. As noted, rows in outage will be corrected by row-redundancy [10]. The outage condition simplifies to

$$\varepsilon = \mathbb{P}\left[DRV_{(n-u)} \geq v_S\right], \tag{8}$$

where the code can correct up to u errors and $DRV_{(t)}$ is the tth largest random DRV. The power per bit function is

$$\mathcal{P}_{0.01}(v_S) = G \cdot \frac{n}{k} \cdot (v_S)^2. \tag{9}$$

The trade-off curves are shown in Fig. 6. For Hamming codes, the minimum $\mathcal{P}_{0.01}(v_S)$ is 33% less than the worst-case for $n = 31$. The corresponding numbers for Reed Muller code are 33% and 256, respectively. A significant fraction, 33% out

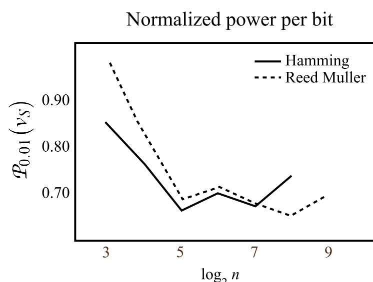

Fig. 6. $\mathcal{P}(v_S)$ **for finite n:** *For an outage $\varepsilon = 0.01$, the optimum power per bit for Hamming and Reed Muller codes are plotted. Maximum power reduction is achieved at $n = 31$ for Hamming codes and at $n = 256$ for Reed Muller codes.*

of the optimum 40% (see Thm. 3.1), power per bit reduction is achieved with a single clock-cycle latency Hamming code. The gap can be reduced with higher-complexity coding. The returns are marginal, e.g., 2% extra power per bit can be saved by a Reed Muller code with an 8-times larger block length.

D. Coding and latency overheads

We selected the Hamming code with a block length $n = 31$ for implementation. We synthesized the encoder-decoder using CAD tools (90nm CMOS technology). The estimated average encoding and decoding energy for 26-bit word were $0.93pJ$ and $2.32pJ$, respectively. The measured leakage-current at $200mV$ for 256 cells was $55.76nA$. Based on this data, *we estimated that $T_s \geq 100ms$ is sufficient to achieve power per bit reduction of 33%.* The latencies of the Hamming encoder and decoder are 1-clock cycle each (2ns).

Independence of *DRV*: Correlations in the *DRV* can be exploited with better coding strategies. However, from the test-chip measurements, we observed a small spatial correlation factor (< 0.1) in the *DRV* data. Since the measured correlation is small, the resulting gains will not be significant. Therefore, we work with the pessimistic i.i.d. assumption.

IV. Conclusions

We studied reduction of leakage-power during data-retention in standby SRAMS. For reliable retention, the supply voltage of an SRAM cell should be greater than a threshold voltage *DRV*. In the presence of process-variations, existing low-power design method uses a supply voltage larger than the largest *DRV* in an SRAM. Instead, we have advocated aggressive voltage reduction with an error-control code based SRAM architecture to reduce standby power. We established fundamental bounds on standby power reduction. We also studied the dependence of power-reduction on block-length for low-latency codes. We showed that a significant portion of the optimum power-reduction can be achieved by a Hamming code with a block-length 31. We proposed a practical reliable memory architecture based on the Hamming code for the next test-chip implementation with a predicted power reduction of 33% while accounting for the coding and latency overheads.

References

[1] System Drivers, "International Technology Roadmap for Semiconductors," *http://www.itrs.net*, pp. 1–25, 2005.

[2] M. Sheets, B. Otis, F. Burghardt, J. Ammer, T. Karalar, P. Monat, and J. Rabaey, "A (6x3)cm^2 self-contained energy-scavenging wireless sensor network node," in *Wireless Personal Multimedia Communications, WPMC, Abano Terme, Italy*, 2004.

[3] H. Qin, Y. Cao, D. Markovic, A. Vladimirescue, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," in *ISQED'04: Proc. of Fifth Intl. Symposium on Quality Electronic Design*, 2004, pp. 55–60.

[4] H. Qin, R. Vattikonda, T. Trinh, Y. Cao, and J. Rabaey, "SRAM cell optimization for ultra-low power standby operation," *Journal of Low Power Electronics*, vol. 2, no. 3, pp. 401–411, Dec 2006.

[5] C. Heegard and A. E. Gamal, "On the capacity of computer memory with defects," *IEEE Trans. on Information Theory*, vol. 29, no. 5, pp. 731–739, Sept 1983.

[6] C. W. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations," *IEEE Trans. on Reliability*, vol. 5, no. 3, pp. 397–404, Sept 2005.

[7] M. Agostinelli et al., "Erratic fluctuations of SRAM cache vmin at the 90nm process technology node," in *IEEE International Electron Devices Meeting, 2005.* IEDM Technical Digest, Dec 2005, pp. 655–658.

[8] K. M. Cao et al., "BSIM4 gate leakage model including source-drain partition," in *IEEE International Electron Devices Meeting, 2000.* IEDM Technical Digest, Dec 2000, pp. 815–818.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley, 1991.

[10] W. K. Huang, Y. Shen, and F. Lombardi, "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp. 323–328, Mar. 1990.

[11] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, 1st ed. NJ, USA: Prentice Hall, 1995.