# 6

# Standby Supply Voltage Minimization for Reliable Nanoscale SRAMs

Jiajing Wang and Benton H. Calhoun
*University of Virginia*
*United States*

## 1. Introduction

Increased leakage current and device variability are posing major challenges to CMOS circuit designs in deeply scaled technologies. Static Random Accessed Memory (SRAM) has been and continues to be the largest component in embedded digital systems or Systems-on-Chip (SoCs). It is expected to occupy over 90% of the area of SoC by 2013 (Nakagome et al., 2003). As a result, SRAM is more vulnerable to those challenges. To effectively reduce SRAM leakage and/or active power, supply voltage ($V_{DD}$) is often scaled down during standby operation (e.g. (Qin et al., 2004; Flautner et al., 2002; Bhavnagarwala et al., 2004; Wang et al., 2007)) and/or active operation (e.g. (Morita et al., 2006; Joshi et al., 2007)). For ultra-low-energy applications, SRAMs operating with $V_{DD}$ near/below the threshold voltage ($V_T$) are also proposed (e.g. (Calhoun & Chandrakasan, 2007; Verma & Chandrakasan, 2008)). However, all SRAM functions, including read stability, write ability, access performance, and hold stability, are less reliable at lower voltage, which leads to the reduction of yield. The minimum supply voltage (Vmin) is limited by the lowest acceptable yield and determines the maximum achievable power reduction. Applying an underestimated Vmin will cause intolerable failures and decrease SRAM yield. On the other hand, applying an overestimated Vmin will waste power and energy. However, finding the optimum Vmin becomes difficult in the presence of global and local variations.
In this chapter, we particularly explore SRAM Vmin during standby mode, i.e. data retention voltage (DRV). We first analyze the impacts of local/random and global/systematic variations on DRV, and then present new statistical and adaptive design methods to address those impacts. The goal of this chapter is to develop effective methods for achieving the best leakage power savings while maintaining the desired yield under variations.

## 2. Variation impact on data retention voltage

### 2.1 Data Retention Voltage (DRV)

Fig. 1 shows the structure of the conventional 6T SRAM cell. The cell consists of two cross-coupled inverters ((PL,NL) & (PR,NR)) and the pass-gate transistor XL/XR on each side. Q and QB are the internal nodes storing the data. During standby mode, the WL signal remains low. BL/BLB signals are often precharged to either high or low. Although floating bitline is also proposed to further reduce BL leakage current (Wang et al., 2007), we assume that the BLs remain high in this chapter. Fig. 1 also illustrates the paths of the major leakage

Source: Solid State Circuits Technologies, Book edited by: Jacobus W. Swart,
ISBN 978-953-307-045-2, pp. 462, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

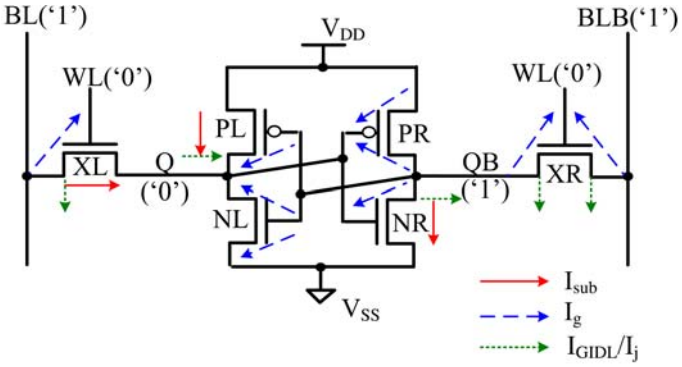


Fig. 1. 6T SRAM cell and the path of the major leakage currents.

current components during standby mode for nanometer technologies. They are sub-threshold leakage current ($I_{sub}$), gate leakage current ($I_g$), gate induced drain leakage ($I_{GIDL}$), and junction leakage current ($I_j$). $I_{sub}$ is the drain-to-source current when the transistor operates in weak inversion. It decreases exponentially with the reduction of the drain-to-source voltage ($V_{DS}$) due to the drain induced barrier lowering (DIBL) effect (Ferre & Figueras, 2005). $I_g$ is the direct tunneling current through the gate oxide to the channel as well as to the overlap region between gate and source/drain extension. Since it grows exponentially with the scaling of the gate oxide thickness, $I_g$ becomes the dominant leakage source for CMOS technologies beyond 45nm. Recent new high-k metal gate device option provides large reduction in gate leakage (Mistry et al., 2007). In addition, a lower $V_{DD}$ exponentially reduces $I_g$. $I_{GIDL}$ is caused by the high electric field under the gate-to-drain overlap region, and $I_j$ is caused by the reverse-biased pn junction (Roy et al., 2003). Both $I_{GIDL}$ and $I_j$ also decrease dramatically with $V_{DD}$. Therefore, $V_{DD}$ scaling can effectively reduce the total cell leakage current, $I_{lk,total}$. Fig. 2 shows that $I_{lk,total}$ can be reduced by more than 10× for a cell in 45nm. Due to the direct effect of $V_{DD}$, the cell leakage power, which is equal to $I_{lk,total} \cdot V_{DD}$, can be further reduced with a lower $V_{DD}$.

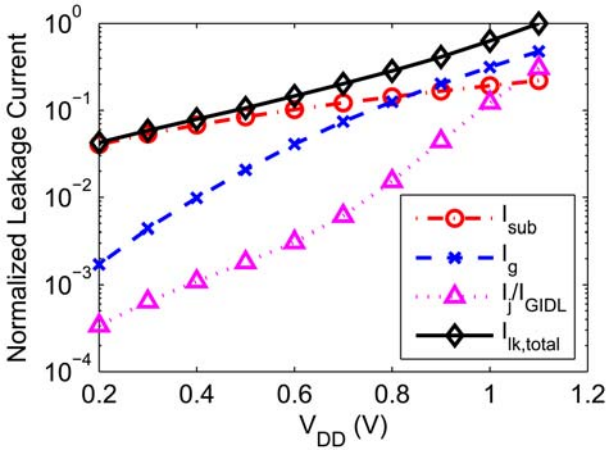


Fig. 2. The normalized cell leakage current versus $V_{DD}$.

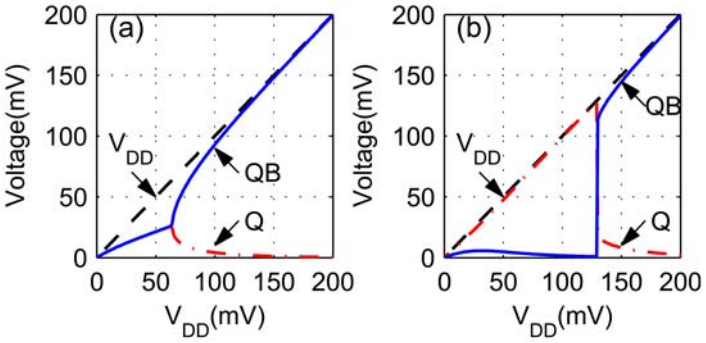


Fig. 3. The voltage of the storage nodes against $V_{DD}$ for (a) a balanced cell and (b) a imbalanced cell (© 2007 IEEE).

However, the drawback of a scaled $V_{DD}$ is the degradation of the cell stability. Fig. 3 shows that excessive $V_{DD}$ scaling results in the loss of the stored data ('0' in this example). Fig. 3(a) particularly shows the balanced case when there is no mismatch between the transistors on the left side (PL/NL/XL in Fig. 1) and those on the right side (PR/NR/XR in Fig. 1). Q and QB converge to a metastable point as a result of the degraded gain. Fig. 3(b) shows the other case when the cell is imbalanced by some mismatch in $V_T$. In this case, Q and QB flip to the more stable state ('1' here). The data retention voltage (DRV) defines the minimum $V_{DD}$ below which the SRAM cell can not preserve its data (Qin et al, 2004). So DRV is the fundamental limiter of the lower $V_{DD}$ operation and prohibits additional power savings. We define DRV0 and DRV1 as the minimum $V_{DD}$ for preserving '0' and '1' respectively. For the balanced case as in Fig. 3(a), DRV0=DRV1; for the imbalanced case, one increases while the other decreases (e.g. DRV0>>DRV1 for the example in Fig. 3(b)). To ensure the cell can safely hold both '0' and '1', the actual DRV is the maximum value of DRV0 and DRV1. Fig. 3 thereby implies that DRV increases when any mismatch occurs.

Unfortunately, device variability increases with technology scaling. In order to predict the maximum achievable power savings from lowering $V_{DD}$, we must evaluate the impact of device variability on DRV. All the variations can be categorized into two groups: *global/systematic* variation and *local/random* variation. Global variations influence all the transistors on the chip. On the other hand, local variations have a different effect on individual transistors, and thus cause mismatch between adjacent devices. Next, we will examine the impact of these variations on DRV.

### 2.2 Impact of local/random variation

Variations occur in a variety of physical parameters, mainly including the threshold voltage ($V_T$), the gate oxide thickness (Tox), the channel effect length ($L_{eff}$), and the channel effect width ($W_{eff}$). Among these parameters, DRV is most sensitive to $V_T$ (Qin et al., 2004). In addition, the variation of $L_{eff}$ can cause $V_T$ variation due to the short channel effect. Therefore, we mainly consider the impact of $V_T$ variation on DRV. Random doping fluctuation (RDF) is the dominant source of local $V_T$ variation, and it deteriorates with continuous device scaling. The RDF induced random $V_T$ variation can be modeled as a normal distribution with its standard deviation ($\sigma_{VT}$) inversely proportional to the square root of the channel area as below (Asenov et al., 2003).

$$\sigma_{V_T} \propto \frac{1}{\sqrt{W_{eff}L_{eff}}} \tag{1}$$

SRAM cells commonly use transistors with smaller geometry for higher density. Thus they are naturally more susceptible to random variations due to a larger value of $\sigma_{V_T}$.
Given the statistics of parametric variations, we can use Monte Carlo (MC) simulation to investigate the impact of variations on the figure of merit. Fig. 4 is the histogram of the cell DRV values with a 5000-point MC simulation in a commercial 90nm CMOS process. The DRV exhibits a non-Gaussian distribution with a longer tail on the right side. The tail value of the distribution is the lowest supply voltage that can be applied to the whole SRAM array without losing any data. We call it the standby Vmin for an SRAM. Vmin determines the maximum achievable power reduction for the entire SRAM array. Therefore, the estimation of the tail value becomes crucial. Modern SRAMs often contain millions of cells, thus the tail event only occurs once out of millions of cell simulations. For such a rare event, the Monte Carlo method requires at least millions of runs, thereby becoming prohibitively expensive. To speed up the estimation of these rare events, various methods arise and fall into the following two major categories.

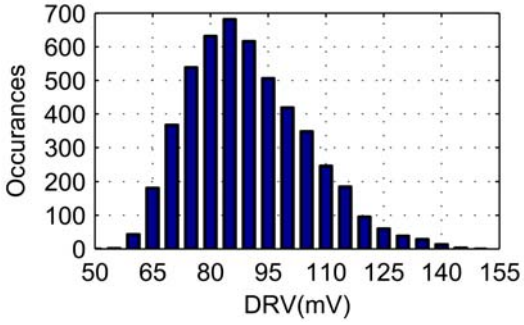


Fig. 4. The histogram of DRV from Monte Carlo simulation with 5000 samples (© 2007 IEEE).

- *Non-Monte-Carlo (non-MC) methods*
  The first non-MC method is to develop a comprehensive analytical model. Although Qin et al. (2004) proposed a theoretical model to approximate the DRV of a single cell, they did not address the statistical characteristics of DRV. The question of how variations impact the long tail of the DRV distribution is not answered. The second and more generic non-MC method is the boundary searching approach, which intends to find the boundaries in the parameter space that correspond to success/failure of the circuit without using MC sampling (Gu & Roychowdhury, 2008). The authors demonstrated its efficiency for estimating SRAM read access yield when considering only two major design parameters. However, the real access yield is also determined by other design parameters that have a minor impact on read access. When all the parameters are searched, this method becomes quite expensive.
- *Improved Monte-Carlo (MC) methods*
  The huge expense of MC for rare event estimation is mainly due to the inefficiency of the rare event sampling. Importance sampling (Kanj et al., 2006) and the Statistical Blockade (SB) tool (Singhee & Rutenbar, 2007) are two interesting techniques to hasten the generation of the rare events. However, their efficiency highly relies on the

goodness of the sampling distribution and the tail filter respectively. Extrapolation is an alternative way to avoid a full MC simulation. We can run a relatively small number of samples and fit them into a known distribution. After that, we can quickly acquire the estimates in the extreme tail region by simply calculating with the fitting distribution. Although it is much simpler, its accuracy is dependent on how good the fitting distribution is. For non-Gaussian variables like DRV, it is hard to find a proper known distribution that can well fit the skewed tail region. Fitting a normal and log-normal distribution either underestimates or overestimates the tail values, respectively. The SB tool proposes to use the generalized Pareto distribution (GPD) to particularly fit the tail samples. Its accuracy is dependent on the number of tail samples, which also requires fast Monte Carlo methods like the tail filter in the SB tool to accelerate its generation.

In this chapter, we propose a new fast method to predict the tail of the DRV distribution. We use the extrapolation method so that only a small number of Monte Carlo samples is required. High accuracy is achieved by using a dedicated statistical model for DRV (Wang, Singhee et al., 2007). We will describe the details of this method in section 3.

### 2.3 Impact of global/systematic variation

Global variations include manufacturing related process variations, voltage supply fluctuations, and temperature changes (i.e. PVT variations). We assume the temperature range is [0°C, 105°C] and the voltage fluctuation range is [-25mV, 25mV]. Fig. 5 shows the DRV histogram of a 5-Kb SRAM array at three PVT cases: typical, best-case, and worst-case. The typical case is at the TT (typical-N and typical-P) process corner, 25°C, and zero voltage fluctuation; the best case for the technology we use is at the SS (Slow-N and slow-P) process corner, 0°C, and 25mV voltage fluctuation; the worst case happens at the FS (Fast-N and slow-P) process corner, 105°C, and -25mV voltage fluctuation. Under one PVT scenario, local variations spread the DRV of the cells, and the tail of the distribution (marked with circle) determines the standby Vmin for this global condition. In contrast, global variations predominantly move the entire DRV distribution around, so the tail point, i.e. the standby Vmin, also shifts with global effects. For this 90nm process, the worst-case Vmin ($Vmin_{wc}$) is about 100mV and 140mV higher than the typical case Vmin ($Vmin_{typ}$) and the best-case Vmin ($Vmin_{bc}$) respectively. For more advanced processes, the variability of global effects might increase and result in a larger difference between $Vmin_{wc}$ and $Vmin_{typ}$/$Vmin_{bc}$. To ensure data safety under all the conditions, we must address this Vmin variability.

The most straight forward method is the worst case approach, which uses a standby $V_{DD}$ based on the worst case at design time and even adds some guard-band for more robustness. For instance, authors of the drowsy cache set the standby $V_{DD}$ 50% higher than the threshold voltage despite the fact that the actual DRV can be much smaller (Kim et al., 2004). A processor with a drowsy mode is also implemented by collapsing the supply voltage well above that required to upset the logic states during standby mode (Clark et al., 2004). Although this open-loop worst-case approach is very robust, it can potentially waste substantial power because of two reasons. First, the worst PVT scenario only occurs in extreme conditions like extremely high temperature, which is very rare for most of the applications. Second, the difference of the Vmin values between the worst case and the non-worst cases can be quite large, and it even becomes larger as CMOS technology continuously scales. We can expect that the conservative worst-case approach would sacrifice more power savings for future CMOS technologies.

In order to gain optimum power reduction for non-worst-case conditions, we propose a closed-loop standby $V_{DD}$ scaling system with online replica cells as monitors for tracking

PVT variations (Wang & Calhoun, 2008). Section 4 will present the details of this new approach.

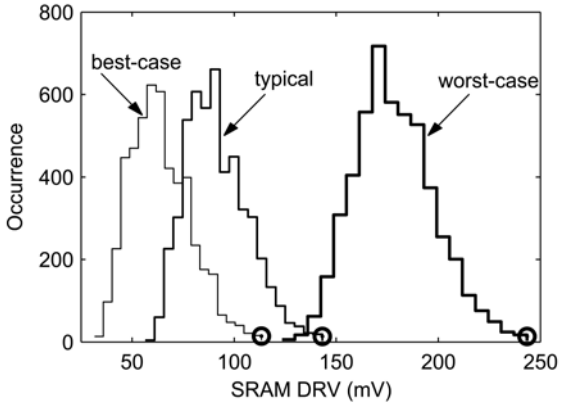


Fig. 5. DRV distribution of a 5Kb SRAM array with global PVT variations and local variations. Three PVT cases (typical, best-case, and worst-case) are shown (© 2008 IEEE)

## 3. Fast and accurate estimation of standby Vmin

In this section, we propose a fast method to predict standby Vmin, i.e. the tail of the DRV distribution in the presence of random variations. Let us define $P_{cf}(v)$ as the probability that the cell fails when $V_{DD}=v$ during standby. We can compute $P_{cf}(v)$ in two ways. First, in terms of DRV, since DRV is the minimum $V_{DD}$ below which a cell cannot preserve its data we can compute $P_{cf}(v)$ as

$$P_{cf}(v) = P(DRV > v) = 1 - F_{DRV}(v) \tag{2}$$

where $F_{DRV}$ is the cumulative density function (cdf) of DRV. We can also compute $P_{cf}(v)$ in terms of static noise margin (SNM), which is the conventional metric for cell stability. A cell fails at voltage $v$ when its SNM is less than the lowest acceptable noise margin s (e.g. s=0 in a noiseless system), so we can also compute $P_{cf}(v)$ as

$$P_{cf}(v) = P(SNM_v < s) = F_{SNM_v}(s) \tag{3}$$

where $SNM_v$ is the cell's SNM at $V_{DD}=v$ and $F_{SNM_v}$ is the cdf of $SNM_v$. As we observed in Fig. 4, DRV has a non-Gaussian distribution with a heavy tail on the right side, which makes it hard to directly fit the DRV data into a known distribution. Nevertheless, because of the equivalence of (2) and (3), we can obtain $F_{DRV}$ through the simple transformation of $F_{SNM_v}$ by

$$F_{DRV}(v) = 1 - F_{SNM_v}(s) \tag{4}$$

As we will show in the next section, it is much easier to obtain $F_{SNM_v}$. Thus we can derive the cdf of DRV from SNM and finally derive the inverse cdf or the quantile function of DRV.

### 3.1 Statistics of hold static noise margin

The most popular metric for SRAM noise margin is the butterfly curve based SNM, which is the maximum amount of dc voltage noise that a cell can tolerate (Seevinck et al., 1987) and is

equivalent to the largest square that can be embedded with the two butterfly curves as shown in Fig. 6. Particularly, the largest square inside the upper-left lobe is defined as SNMH, the SNM for holding '0'; and the largest square inside the lower-right lobe is defined as SNML, the SNM for holding '1'. The true SNM is the minimum of SNMH and SNML. Fig. 6 further shows how SNMH and SNML change with $V_{DD}$ scaling. In the case that the cell is balanced as in Fig. 6(a), both SNMH and SNML decrease to 0 when $V_{DD}$=65mV. This implies that DRV=DRV0=DRV1=65mV. On the other hand, if the cell is imbalanced by variation as the example in Fig. 6(b), SNMH first drops to 0 while SNML still maintains a positive amount of value when $V_{DD}$=130mV. Therefore, for this example, DRV=DRV0=130mV. In fact, Fig. 6 uses the same examples as Fig. 3. The same DRV results are obtained by directly simulating the collapse of the internal states as in Fig. 3 and by simulating the decrease of SNM with $V_{DD}$ scaling as in Fig. 6. This verifies that we can use SNM to explore DRV.

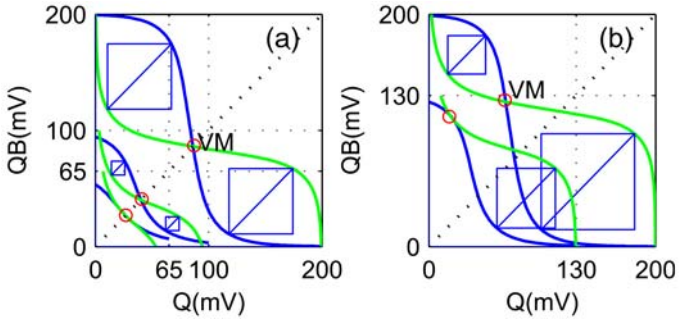


Fig. 6. Butterfly curve based SNM changes with $V_{DD}$ scaling when the cell is (a) balanced and (b) imbalanced by some mismatch (© 2007 IEEE)

The next question we should answer is how local random variations impact SNMH or SNML. Fig. 7 plots the 50,000-point MC simulation results of SNMH and SNML when $V_{DD}$=300mV. We fit a normal distribution to the data of both SNMH and SNML. The normal distribution closely matches the body of both data. The deviation in the tail points is mainly caused by the error of Monte Carlo simulation, which decreases as we use more Monte Carlo samples. Therefore, it is accurate to approximate the true SNMH and SNML with an identical normal distribution.

Since DRV is the $V_{DD}$ point when SNM is equal to the lowest noise margin (e.g. 0 here), a more important question is how those SNM distributions change with $V_{DD}$ scaling. We further examine the SNMH or SNML distribution at different $V_{DD}$ points. We find that SNMH and SNML remain normally distributed. Moreover, as shown in Fig. 8, the mean ($\mu$) is approximately linear with $V_{DD}$ while the standard deviation ($\sigma$) keeps almost constant. If we know that the estimation of the mean and the standard deviation at an initial voltage, $v_0$, are $\mu_0$ and $\sigma_0$, we can quickly obtain the new mean and standard deviation values at any arbitrary $V_{DD}$ point, $v$, with

$$\mu = \mu_0 + k(v - v_0); \quad \sigma = \sigma_0 \tag{5}$$

where $k$ is the sensitivity of $\mu$ to $V_{DD}$ and can be extracted by fitting the mean data in Fig. 8 to the linear curve.

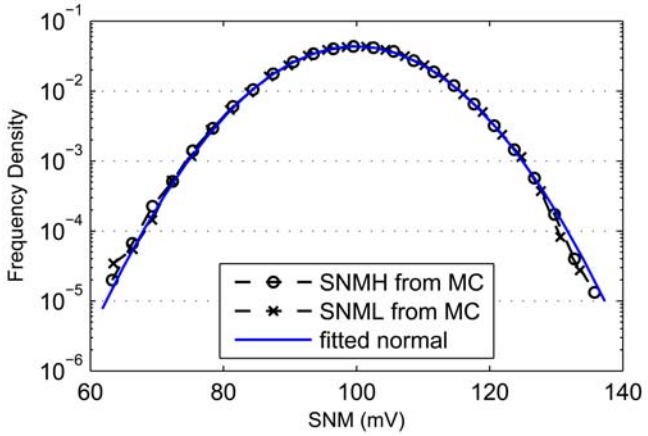


Fig. 7. 50,000-point Monte Carlo results of SNMH and SNML at $V_{DD}$=300mV and a normal distribution is fitted to both data.

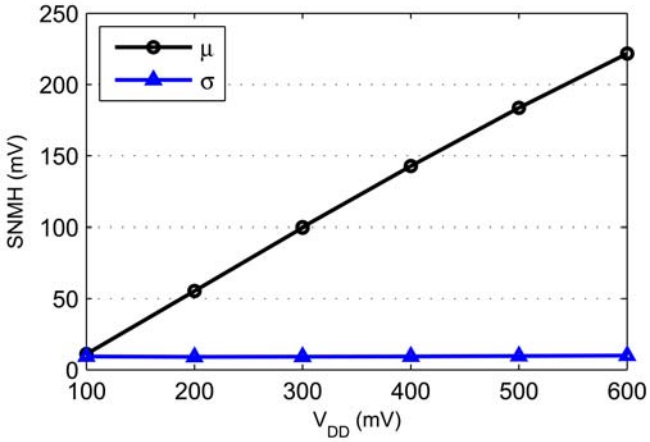


Fig. 8. Estimated mean and standard deviation of SNMH from MC simulations versus $V_{DD}$.

### 3.2 DRV and yield model

So far we are able to predict the distribution of SNMH or SNML at any $V_{DD}$ point with (5). The real SNM is the minimum of SNMH and SNML. If we assume SNMH and SNML are independent random variables, according to order statistics, the cumulative density function of the real SNM can be calculated as follows.

$$\begin{aligned} F_{\mathrm{SNM}_v}(s) &= \mathrm{P}(\mathrm{SNM}_v < s) \\ &= \mathrm{P}(\min(\mathrm{SNMH}_v, \mathrm{SNML}_v) < s) \\ &= \mathrm{P}(\mathrm{SNMH}_v < s) + \mathrm{P}(\mathrm{SNML}_v < s) - \mathrm{P}(\mathrm{SNMH}_v < s, \mathrm{SNML}_v < s) \\ &= \mathrm{erfc}(x) - \frac{1}{4}\mathrm{erfc}^2(x), \quad \text{where} \quad x = \frac{\mu_0 + k(v - v_0) - s}{\sqrt{2}\sigma_0}. \end{aligned} \tag{6}$$

Here erfc() is the complementary error function. (6) actually estimates the cell failure probability during standby as expressed in (3). Thus we can quickly estimate the yield of an SRAM array with a given capacity when the standby $V_{DD}$ is equal to $v$.
Another important estimation is the minimum standby $V_{DD}$ for a given yield or cell failure probability constraint. In other words, we want to estimate the DRV quantile. To derive DRV quantile function, we first obtain the cdf model of the DRV by substituting (6) into (4):

$$F_{\mathrm{DRV}}(v) = 1 - \mathrm{erfc}(x) + \frac{1}{4}\mathrm{erfc}^2(x), \quad \text{where} \quad x = \frac{\mu_0 + k(v - v_0) - s}{\sqrt{2}\sigma_0}. \tag{7}$$

Then we obtain the quantile function, i.e. the inverse cdf of DRV, as:

$$v = F_{\mathrm{DRV}}^{-1}(p) = \frac{1}{k}\left(\sqrt{2}\sigma_0 \cdot \mathrm{erfc}^{-1}(2 - 2\sqrt{p}) - \mu_0 + s\right) + v_0, \tag{8}$$

where $\mathrm{erfc}^{-1}()$ is the inverse function of erfc() and $p$ is the probability that DRV≤$v$.
Both (7) and (8) only require 4 parameters: $v_0$, $\mu_0$, $\sigma_0$, and $k$. First, we pick m (e.g. m≤6) typical $V_{DD}$ points, say $v_1$, … , $v_m$. Then we run $n_{MC}$ Monte Carlo samples of SNMH at $v_i$ and fit a normal distribution $N(\mu_i, \sigma_i^2)$ to the data. Since we estimate the mean and standard deviation of the distribution body instead of the distribution tail, a small scale of Monte Carlo (e.g. $n_{MC}$=1,000~5,000) is sufficient. After obtaining $\mu_i$, we extract $k$ by fitting a linear curve to the ($v_i$, $\mu_i$) data. Finally we pick one $V_{DD}$ point as the initial point $v_0$, and then $\mu_0$ and $\sigma_0$ are chosen accordingly. Therefore, the total number of Monte Carlo samples used in our method is equal to m×$n_{MC}$, which is 6×5,000 in our test case. To further reduce the run time, we can use a simpler way to approximate $k$. Instead of running MC simulations on multiple $V_{DD}$ points, we can run a nominal dc simulation of SNM with the sweep of $V_{DD}$. However, this simplification might cause a slightly larger error.

### 3.3 Experiment results

We use a 6T cell in a commercial 90nm process to test our DRV model. Without loss of generality, we choose the lowest acceptable noise margin $s$=0 in the test. Since SRAMs usually contain at least 1,000 cells, we are interested in the DRV quantiles $F_{\mathrm{DRV}}^{-1}(p)$ that have the probability $p$ ≥0.999. For the same probability $p$, the quantile of a theoretical standard normal variable $M \sim N(0,1)$ is m= $\Phi^{-1}(p)$, where $\Phi^{-1}$ is the inverse of standard normal cdf. We thereby plot the estimated DRV quantile versus the normal quantile (m) that has the equivalent probability $p$ ≥0.999. Fig. 9 plots the estimates of the DRV quantiles equivalent to m∈ [3,8] from several methods.

1. *Analytical model*: The DRV quantiles estimated from (8) with $p$ = Φ(m) are plotted with the solid curve. We select $v_0$=100mV. $\mu_0$ and $\sigma_0$ are obtained by fitting a normal distribution to the 5,000-point MC result for SNMH at $v_0$. The parameter $k$, the sensitivity of the mean of SNMH to $V_{DD}$, is obtained from linear fitting the curve in Fig. 8.
2. *Standard Monte Carlo or fast Monte Carlo with the Recursive Statistical Blockade*: The DRV quantiles estimated from a 1-million-point Monte Carlo simulation of DRV are plotted with the circles. With 1-million raw MC samples, the maximum DRV quantile we can estimate with a high confidence is equivalent to the normal quantile m≈4. For m>4, we use the fast Monte Carlo method with the recursive statistical blockade tool (Singhee et al., 2008) to reduce run time.

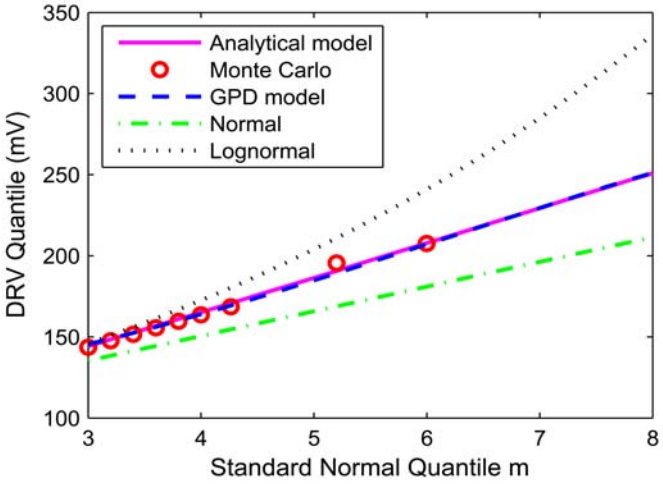


Fig. 9. The DRV quantiles estimated from different methods against the theoretical standard normal quantiles; our new model (8) and the GPD model from the Statistical Blockade tool (Singhee & Rutenbar, 2007) (lines coincident on the plot) closely track Monte Carlo simulation and match farther out in the tail (© 2007 IEEE).

3. *GPD model from the Statistical Blockade (SB)*: The 1,000 tail points from the last recursion stage of the recursive statistical blockade run are used to fit a generalized Pareto distribution (GPD) (Singhee & Rutenbar, 2007). The results estimated from the GPD model are plotted as the dashed curve.
4. *Normal model*: A normal distribution is fit to the DRV data from a 5,000-point MC simulation. The DRV quantiles estimated from the fitting normal distribution are plotted as the dash-dotted curve.
5. *Lognormal model*: A lognormal distribution is fit to the same set of the 5,000 MC points for DRV. The DRV quantiles estimated from the fitting lognormal distribution are plotted as the dotted curve.

Fig. 9 shows that both the results from our model and from the GPD model closely match the MC results up to m=6. In addition, our model matches well with the GPD model at the tail region of m>6, where the tail event has the probability smaller than 9.86e-10. Extrapolation with either normal or lognormal distribution is inaccurate, especially for the points farther out in the tail. The normal model underestimates DRV while the lognormal model overestimates it.

With the comparable accuracy, our method offers a significant speedup over the standard Monte Carlo method because it only requires a small number (e.g. 5,000) of MC simulations for SNMH at a couple of $V_{DD}$ points (totally ≤30,000) to predict any extreme DRV tail values. However, if the probability of the tail event is $p_t$, the standard MC method requires at least $1/p_t$ samples to obtain one estimate of the quantile. For example, when $p_t$=9.86e-10 (i.e. m=6), we must run at least 1-billion simulations. Thus, our method provides a speedup of at least 30,000× over standard MC. The recursive statistical blockade requires about 41,700 simulations (Singhee et al., 2008), so our method offers a slight speedup of 1.4× over it. For m>6, standard MC would need thousands of billions of simulations. In this case, the speedup over MC is extremely large.

## 4. Canary based closed-loop standby $V_{DD}$ scaling

In this section, we deal with the impact of global variations on DRV and present a closed loop $V_{DD}$ scaling system for aggressive leakage power reduction while protecting data by maintaining $V_{DD}$ above the DRV of the worst SRAM cell (Wang & Calhoun, 2007).

### 4.1 Principle

Fig. 10(a) shows the basic architecture of the system. An on-chip or off-chip voltage regulator supplies $V_{DD}$ to the SRAM cells and to the canary replicas. Multiple canary categories are designed to fail across a range of voltages above the average DRV of the SRAM cells as illustrated in Fig. 10(b). The most important feature of the canary cell is its ability to duplicate the impact of global changes on SRAM stability. With this ability, when the failure voltage of the SRAM cell increases or decreases by some amount due to certain global effect, the failure voltage of each canary category will also change by the same amount. In other words, the DRV of each canary category can maintain a predefined proximity to the DRV of the SRAM cells despite changes in global conditions. Note that, just as SRAM cells, the canary cells are also sensitive to local variations. We employ redundancy

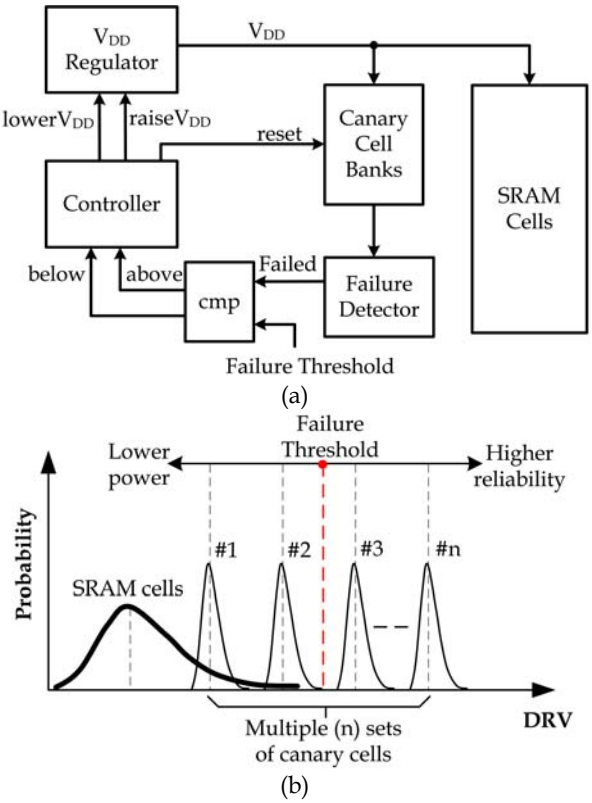


Fig. 10. (a) Architecture and (b) mechanism of the closed loop $V_{DD}$ scaling system (© 2008 IEEE).

and a voting strategy to sharpen the distribution of canary cells within the same category. The failures of the canary categories are monitored by online failure detectors. SRAM data safety is ensured by a programmable failure threshold, which defines the critical failure status of the canary categories and determines the proximity of the standby $V_{DD}$ to the tail of the SRAM DRV distribution. When entering the standby mode, the controller starts lowering $V_{DD}$ until the canary failures meet the failure threshold. Once the global stimuli occur, the canary failures will exceed or drop below the failure threshold, which triggers the controller to raise or lower $V_{DD}$ accordingly.

Besides the improvement of power reduction under variations, this system also allows a trade-off between power savings and data reliability by altering the failure threshold. When the application needs a higher data reliability, a failure threshold that allows less canary sets to fail should be chosen. On the other hand, when the data reliability constraint is lowered or some data errors can be tolerated by redundancy or error correction techniques, we can change the failure threshold to allow more canary sets to fail so that $V_{DD}$ can be reduced for more power savings.

### 4.2 Major components

#### 4.2.1 Canary cell

The canary cell is the most important component in our system. It must replicate the impact of global variations on SRAM cell stability. Moreover, it must fail before the SRAM cells to prevent the loss of data in SRAM. The canary DRV distribution is not a good indicator of the SRAM cell DRV distribution because there are too few canary cells. Therefore, we must use a design that makes it more sensitive to $V_{DD}$ than it would be simply due to the impact of local variation.

We propose the circuit in Fig. 11(a) and (b) as canary cells for holding '1' and '0', respectively. Each canary cell contains the same 6T transistors (M1~M6) as any SRAM cell, an additional pmos pass transistor (M7) for enhancing the ability of writing a '1' at lower voltage, and a pmos header transistor (M8) for tuning the virtual supply of the cell. The input signal, W, and its inversion, WB, act as the bit lines and word line. During reset mode, W rises, and the pass transistors M5~7 are turned on; '1'/'0' is written into the canary cell '1'/'0'. During standby mode, W switches to low and turns off M5~7. In addition, the bitlines are holding the opposite states with the internal nodes, which creates the worst leakage current through M5~7 and contributes to a higher DRV for the canary cell. The header M8 plays the key role for tuning canary DRV. By tuning the input signal VCTRL at its gate, the virtual supply of the canary cell, $VV_{DD}$, becomes smaller than $V_{DD}$, which results in a higher $V_{DD}$ to flip the storage nodes, i.e. a higher DRV for the canary cell. Fig. 12 shows the simulated canary DRV values against the VCTRL values. For comparison, the histogram of the SRAM cell DRV from a 5000-point Monte Carlo simulation is also plotted. Two interesting observations make this tuning knob more appealing. First, there is a nice linearity between canary DRV and VCTRL. Thereby we can create multiple canary categories by simply using regularly increased VCTRL signals, which are easy to implement (e.g. in our test chip, we use a resistor ladder to generate a series of VCTRL signals). Second, the canary DRV can be potentially moved to any point in a wide range. Thus we can always find at least one canary category with its DRV higher than the tail value of SRAM DRV distribution, which could be quite large for big SRAM arrays in scaled technologies.

Now let us further examine the canary cell's capability for tracking PVT variations, which is essential to protecting data in this approach. We use a 1-Kb SRAM and 8 canary sets (#0~#7)

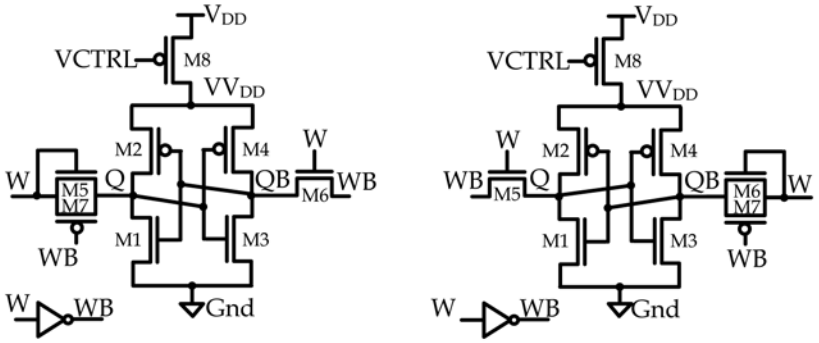


Fig. 11. Schematic of canary cell (a) for holding a '1' and (b) holding a '0' (© 2008 IEEE).

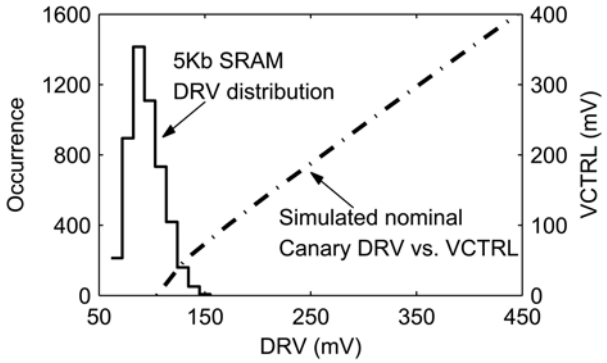


Fig. 12. Simulated nominal canary cell DRV versus VCTRL relative to a 5 Kb SRAM DRV distribution (© 2008 IEEE).

as an example. We first obtain the worst DRV value, i.e. Vmin, of the 1-Kb SRAM with Monte Carlo simulations at normal condition (i.e. at TT process corner & 25°C). Then at the same normal condition, we configure the canary cells by tuning their VCTRL values so that $DRV_{C,7} > DRV_{C,6} > \ldots > DRV_{C,1} > Vmin > DRV_{C,0}$. Here, $DRV_{C,i}$ is the DRV of the canary set #*i*. In order to protect SRAM data, the canary set #1 can be chosen as the first set that should never fail. After configuration, the canary VCTRL values are fixed. Then we change either the temperature or the process corner and rerun the simulations to obtain the new SRAM Vmin and $DRV_{C,i}$ values, which are shown in Fig. 13(a) and (b). The SRAM Vmin is plotted as the curve with circles. $DRV_{C,i}$ is plotted as the curve with triangles. For all the temperature and process changes, the DRV of each canary set moves almost by the same amount as the SRAM Vmin. This indicates that the canaries can successfully track global effects. The only exception here is the SF (Slow-N Fast-P) corner because the technology we use is a strong-N process. At the SF corner, the impact of global variation on the tail of SRAM DRV is overwhelmed by the impact of large local variations. However, the canary DRV is still affected by global variation, so $DRV_{C,1}$ becomes smaller than Vmin at SF corner. To fix this, we can either reconfigure $DRV_{C,1}$ so that $DRV_{C,1} > Vmin$ at this corner or reset the failure threshold to choose the canary set #2 as the first one that does not allow to fail.

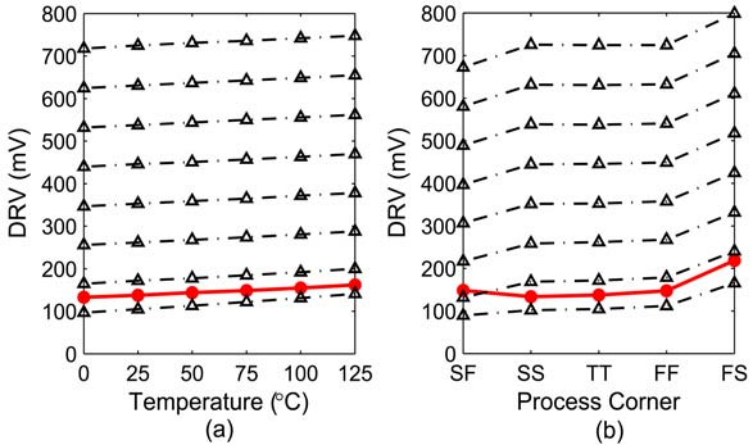


Fig. 13. Simulated DRV of the canary sets (lines with triangles and the upper ones have higher VCTRL) and the worst DRV of a 1 Kb SRAM (the line with circles) change consistently with (a) temperature and (b) process corner for the 90 nm technology (© 2008 IEEE).

#### 4.2.2 Failure detector and canary bank

In our system, the failure of the canary cell is detected online. To enable a quick sensing, the failure detector directly monitors the storage nodes Q and QB of the canary cell. As shown in Fig. 3(a), Q and QB of an SRAM cell might converge if the cell is balanced. However, we set the two bitlines of the canary cell with the complementary values of W and WB (see Fig. 11). This asymmetry makes Q and QB mainly flip when the current $V_{DD}$ is below the cell's DRV. Thus we propose to use a differential sense amplifier shown in Fig. 14 as the failure

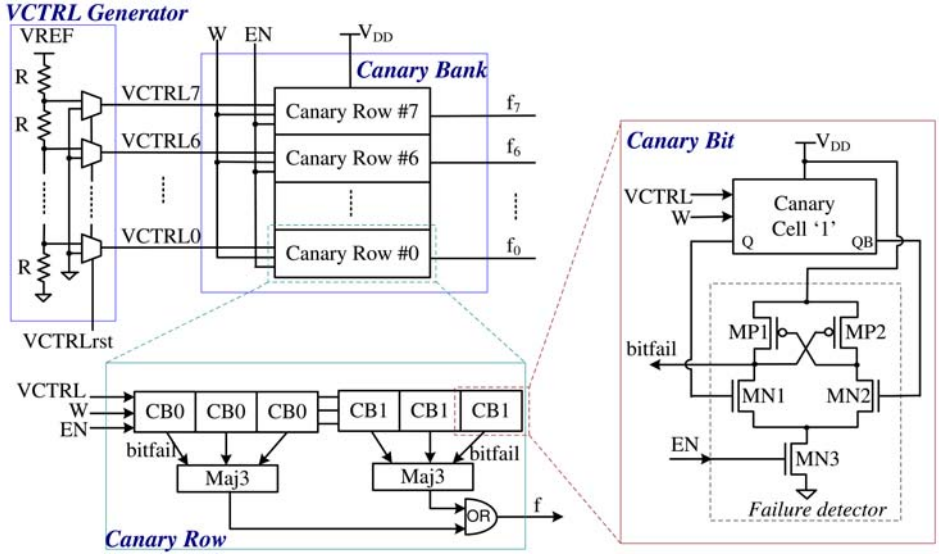


Fig. 14. Canary bank and VCTRL generator.

detector. It shares $V_{DD}$ with the canary cell, and its input differential pair MN1 and MN2 directly connect to Q and QB. One canary cell and its own failure detector compose a canary bit.

The canary sets are deployed as rows in a bank structure as illustrated in Fig. 14. Each canary set occupies one row of the bank. To reduce the variance of the canary DRV, we employ redundancy and majority voting circuits. Thus one canary set (row) consists of n copies of canary bit '1' and n copies of canary bit '0'. Although a larger n can decrease the variance, the area and complexity overhead would dramatically increase. By trading off between the efficiency of variance reduction and the overhead cost, we choose n=3. The failure signals from the three replicas of canary bit '1'/'0' go into the majority-3 gate to generate the voted failure signal. The whole canary set fails when either the majority of the canary bit '1' or the majority of the canary bit '0' fails.

Fig. 14 also shows the VCTRL generator, which is a resistor ladder with a reference voltage VREF and a series of identical resistors. Each canary set (row) is connected to one VCTRL signal from the VCTRL generator.

### 4.2.3 Feedback controller

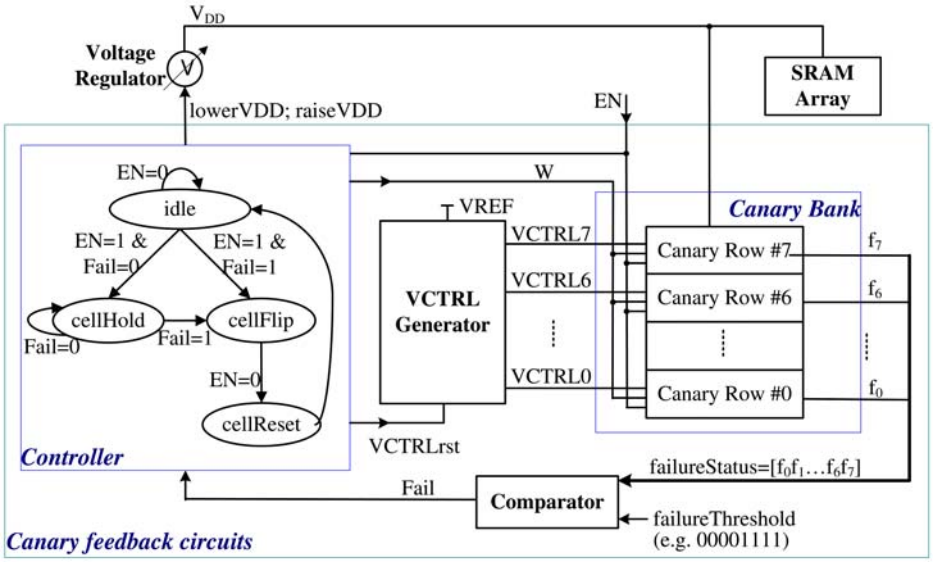


Fig. 15. The feedback controller connects other components in the feedback system.

The feedback controller plays an important role in our system. As shown in Fig. 15, it ties all the other blocks together to form a complete feedback loop. The controller receives the final failure signal 'Fail' from the comparator, which asserts 'Fail' when the failure status of the canary sets ($f_0f_1...f_6f_7$) exceeds the predefined failure threshold. The controller then sends out different control signals to different blocks. The 'lowerVDD' and 'raiseVDD' are sent to the voltage regulator to lower or raise $V_{DD}$ by one step (e.g. 10mV). The 'W' signal is sent to the canary bank for rewriting all the canary cells. The 'VCTRLrst' signal is sent to the VCTRL generator for occasionally resetting all the VCTRL signals to 0.

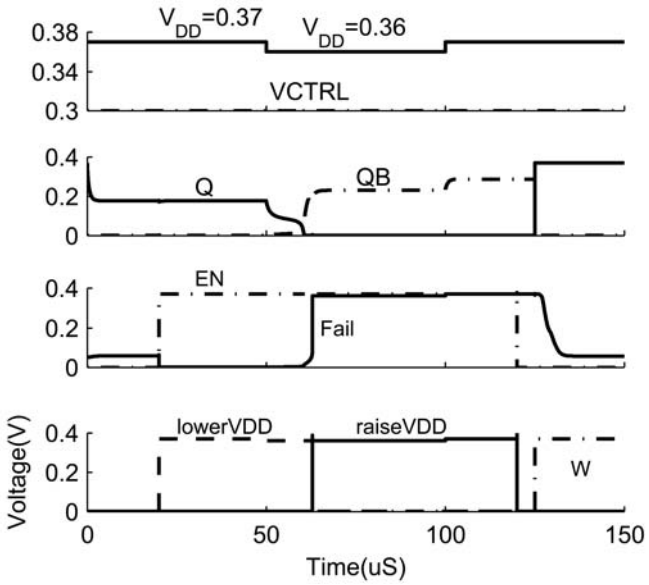


Fig. 16. The timing diagram of the controller (© 2008 IEEE).

Fig. 15 also illustrates the major state transitions in the controller. There are four states: *idle*, *cellHold*, *cellFlip* and *cellReset*. Fig. 16 gives the timing diagram that shows how the states transfer. Suppose the failure threshold is set to 00001111, which implies that the canary set #3 is the first set not allowed to fail. We configure its VCTRL=0.3V. For simplicity, we do not consider redundancy here. After we assert the enable signal 'EN', the failure detector of each canary set evaluates its own Q and QB. When $V_{DD}$=0.37V, Q and QB of the canary set #4-7 flip, but those of the canary set #0-3 maintain their original values. Thus the failureStatus is 00001111, which is no larger than the failure threshold. Therefore, 'Fail' maintains zero, which causes the controller change from the *idle* state to the *cellHold* state, and the signal 'lowerVDD' rises up to inform the voltage regulator to decrease $V_{DD}$ by 10mV. Once $V_{DD}$ is lowered to 0.36V, Q and QB of the canary set #3 flip to the opposite value, resulting in failureStatus=00011111, which is larger than the failure threshold. Thus 'Fail' rises up and the *cellFlip* state becomes valid. This state asserts 'raiseVDD'. As a result, the regulator increases $V_{DD}$ by one step and $V_{DD}$ returns to the previous value 0.37V, which is actually the DRV of the canary cell #3. After that, 'EN' goes low to disable the failure detection, and the controller enters the *cellReset* state, which asserts the 'W' signal to write the original values into Q and QB for next check.

Since SRAM Vmin can be near or even smaller than the threshold voltage $V_T$, all the circuits including the failure detector, the comparator, and the controller are designed to function in the sub-threshold region, where $V_{DD}<V_T$ (Wang et al., 2006).

### 4.3 Model for canary cell tuning

We have observed in Fig. 12 that the canary DRV changes approximately linearly with VCTRL. By analyzing the current through the pmos header (M8 in Fig. 11(a)), we can derive the theoretical model for this linear dependency. We denote $DRV_C$ as the canary DRV. It is

equal to the $V_{DD}$ value when the actual supply voltage of the canary cell, $VV_{DD}$, reaches the cell's true DRV, $DRV_t$, i.e. the cell DRV without the header. Let us denote $I_{min}$ as the leakage current for holding the cell data when $VV_{DD}$=$DRV_t$. We assume that the header M8 operates in the sub-threshold region. Since the sub-threshold leakage current is the dominant source of the leakage current, we can compute $I_{min}$ as

$$I_{min} = I_0 \cdot exp\left[\frac{DRV_C - VCTRL - V_{T,8} + \eta_8(DRV_C - DRV_t)}{n_8 V_{th}}\right] \cdot \left[1 - exp\left(\frac{-DRV_C + DRV_t}{V_{th}}\right)\right] \quad (9)$$

where $V_{T,8}$ is the threshold voltage of M8, $\eta_8$ is its DIBL coefficient, $n_8$ is its sub-threshold swing factor, $V_{th}$ is the thermal voltage, and $I_0$ is its off current. For a given canary cell, we assume that the $DRV_t$ remains the same no matter what VCTRL is. This is reasonable because M1-M7 are not changed. Therefore, $I_{min}$ also remains constant. We further ignore the rolling-off term when $DRV_C - DRV_t > 4V_{th}$ ($V_{th}$=26mV at 300K). Then we can solve $DRV_C$ as

$$DRV_C = \frac{VCTRL}{1+\eta_8} + b, \quad \text{where} \quad b = \frac{V_{T,8} + \eta_8 \cdot DRV_t}{1+\eta_8} + \frac{n_8 V_{th}}{1+\eta_8} ln\left(\frac{I_{min}}{I_0}\right). \quad (10)$$

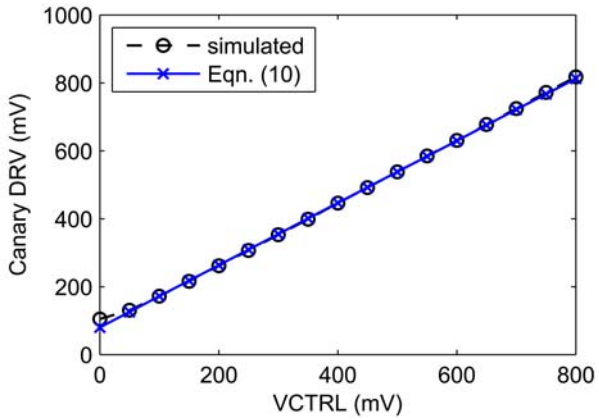


Fig. 17. Estimated canary DRV from (10) versus VCTRL is compared with the simulated result (© 2008 IEEE).

This proves the linear relationship between the canary DRV and VCTRL and implies that the slope can be approximated as $1/(1+\eta_8)$. To verify this model, we first obtain $DRV_t$ and $I_{min}$ from simulation without the header. Then we compute the canary DRV values against VCTRL with (10) and compare them with the simulated results. Fig. 17 shows that our first-order linear model provides an excellent approximation for all the VCTRL values across a wide range. A slightly larger error occurs only when VCTRL<50mV. In this region, the canary DRV ($DRV_C$) is very close to $DRV_t$, so the rolling-off term cannot be ignored, in which case numerically solving (9) can give a more accurate estimation.

In section 3.2, we proposed the model to predict SRAM DRV quantile and yield in the presence of random variations. Now by combining (8) and (10), we can estimate the VCTRL value $y$ that is needed for a canary cell in order to satisfy a given SRAM cell yield as:

$$y = \frac{1+\eta_8}{k}\left[\sqrt{2}\sigma_0 \cdot erfc^{-1}(2-2\sqrt{p}) - \mu_0 + s\right] + (v_0 - b)\cdot(1+\eta_8). \quad (11)$$

where $p$=P($DRV_S$ <$DRV_C(y)$), the probability that the SRAM DRV ($DRV_S$) is less than $DRV_C(y)$, i.e. the canary DRV when VCTRL is equal to $y$. All the other parameters are the same as in (8) and (10). Fig. 18 plots the estimated VCTRL values from (11) with the solid curve. In this figure, the point at the coordinates of ($x$,$y$) means P($DRV_S$ <$DRV_C(y)$) is equal to $\Phi(x)$, where $\Phi$ is the cdf of a theoretical standard normal variable. For instance, if one application requires 90% yield for a fault-free 100-Kb SRAM, the required failure probability is ~1e-7, which is equivalent to the probability when $x$=5.2. From Fig. 18, we quickly know that all the canary cells with VCTRL≤120mV should never fail in order to meet this yield. This gives us the guidance to choose the proper VCTRL value for each canary set. Fig. 18 gives an example of the canary configurations. We configure the canary set #2 with VCTRL=120mV. Then we assign 5 points in the region VCTRL>120mV to the canary set #3~7 and assign 2 points in the region VCTRL<120mV to the canary set #0~1. The failure threshold is set to 00011111 so that only the upper 5 canary sets are allowed to fail. This configuration ensures that SRAM can always achieve 90% yield under any PVT variations. If the application changes and needs a different reliability, we can reset the failure threshold or even reconfigure all of the canary sets (by remapping VCTRL values) for better results.

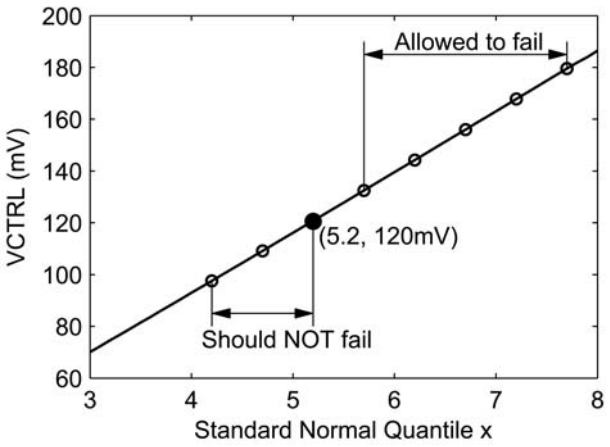


Fig. 18. Estimated VCTRL value $y$ to satisfy that P($DRV_S$ <$DRV_C(y)$) = $\Phi(x)$, where $x$ is a standard normal quantile. To achieve 90% yield for a fault-free 100Kb SRAM (i.e. $x$ = 5.2), only the canary sets with VCTRL>120mV are allowed to fail (© 2008 IEEE).

### 4.4 Test chip implementation & measurement

We implement all of the circuits in Fig. 10(a) except the $V_{DD}$ regulator in a 90nm CMOS bulk test chip. In addition to a 16×8Kb SRAM, the test chip contains the canary circuits and test circuits. The area overhead of the canary circuits is about 0.6%. Fig. 19 shows the die photo of the chip. Fig. 20(a) shows the measured average DRV of canary cells versus VCTRL at

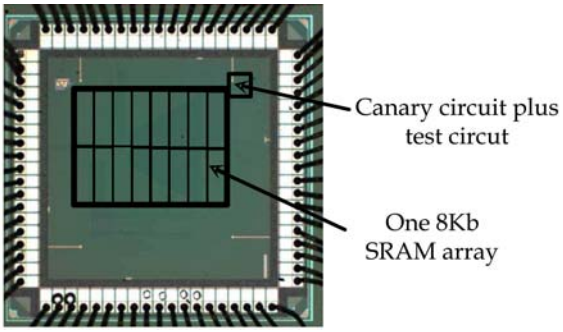


Fig. 19. The die photo of the 90nm test chip (© 2007 IEEE).

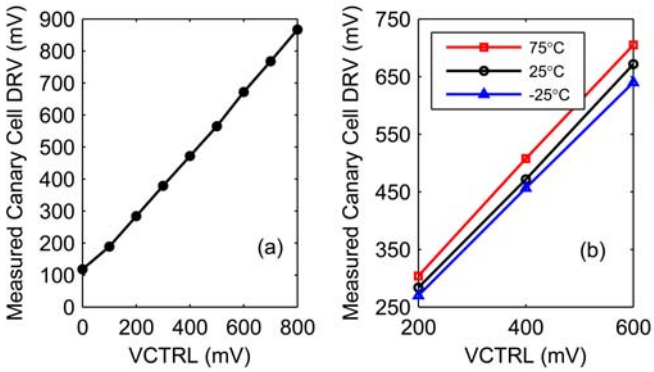


Fig. 20. The measured canary DRV against VCTRL at (a) room temperature and (b) different temperatures (© 2008 IEEE).

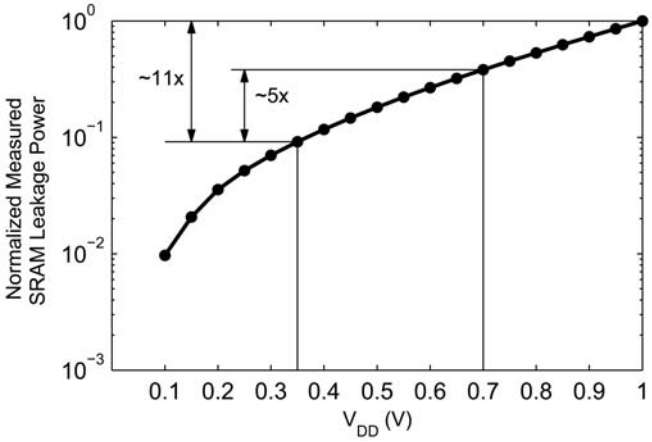


Fig. 21. The normalized measured SRAM leakage power against $V_{DD}$ (© 2008 IEEE).

room temperature. The tuning of VCTRL allows us to provide the desired continuum of failure voltages for the canary cell. It also verifies a good linear relationship between the canary DRV and VCTRL. Fig. 20(b) further shows the measured canary DRV against VCTRL at different temperatures, which verifies that our canary cell can successfully track temperature changes. Fig. 21 plots the normalized measured leakage power of the SRAM array with $V_{DD}$ scaling. Under normal environmental conditions, the measured worst DRV of one 8Kb SRAM array is 0.35V. We estimate that the worst standby Vmin for all the PVT variations plus certain guardband is equal to 0.7V. With the worst case approach, we always set the standby $V_{DD}$ to 0.7V. In contrast, by using our canary-based feedback approach, we can adjust $V_{DD}$ to near the true Vmin value (i.e. 0.35V) at the normal condition. Thereby the canary approach offers ~5× power reduction compared with the conservative worst-case approach and ~11× reduction compared with using the nominal $V_{DD}$, 1V.

## 5. Conclusion

Variation has become one of the biggest challenges for circuit design in scaled CMOS technologies. In this chapter, we first investigate the impact of both local and global variations on SRAM data retention voltage (DRV) and then present a method to deal with each type of variation. Local random variations spread the cell DRV across the same array, and the tail of the distribution is the minimum standby $V_{DD}$ (Vmin) that can be applied on the whole SRAM. We propose a fast and accurate method to predict the tail DRV. Our method offers the comparable accuracy with the standard Monte Carlo (MC) method and shows an excellent agreement with another fast method, the Statistical Blockade (SB) tool, for the tails up to $8\sigma$. It offers the speedup of > $10^4$× over MC and 1.4× over SB. Global PVT variation results in the shift of Vmin values. The worst-case design approach over-protects non-worst-case scenarios. To enable optimum power savings for any PVT scenario, we propose a closed-loop $V_{DD}$ scaling approach. It uses online canary replica cells and monitors to track global variations, and a feedback circuit to adjust $V_{DD}$ to approach the true Vmin. As device variability continues growing with CMOS technology scaling, SRAM supply voltage scaling requires efficient statistical analysis methods and smart adaptive approaches to maximize power reduction while maintaining correct functionality and acceptable noise immunity.

## 6. References

Asenov, A. et al. (2003). Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs, *IEEE Transactions on Electron Devices* 50(9): 1837–1852.

Bhavnagarwala, A. J. et al. (2004). A transregional CMOS SRAM with single, logic $V_{DD}$ and dynamic power rails, *Symposium on VLSI Circuits*, pp. 292–293.

Calhoun, B. & Chandrakasan, A. (2007). A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation, *IEEE Journal of Solid-State Circuits* 42(3): 680–688.

Clark, L., Morrow,M. & Brown,W. (2004). Reverse-body bias and supply collapse for low effective standby power, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 12(9): 947–956.

Ferre, A. & Figueras, J. (2005). *Low-Power Electronics Design, 2nd*, CRC Press, chapter Leakage in CMOS Nanometric Technologies, pp. 3_1–3_19.

Flautner, K. et al. (2002). Drowsy caches: simple techniques for reducing leakage power, *Proceeding of International Symposium on Computer Architecture*, pp. 148–157.

Gu, C. & Roychowdhury, J. (2008). An efficient, fully nonlinear, variability-aware non-montecarlo yield estimation procedure with applications to SRAM cells and ring oscillators, *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 754–761.

Joshi, R. et al. (2007). 6.6+ GHz low Vmin, read and half select disturb-free 1.2 Mb SRAM, *Symposium on VLSI Circuits*, pp. 250–251.

Kanj, R., Joshi, R. & Nassif, S. (2006). Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events, *Proceedings of Design Automation Conference (DAC)*, pp. 69–72.

Kim, N. S. et al. (2004). Circuit and microarchitectural techniques for reducing cache leakage power, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 12(2): 167–184.

Mistry, K. et al. (2007). A 45nm logic technology with high-k+metal gate transistors, strained silicon, 9 Cu interconnect layers, 193nm dry patterning, and 100% pb-free packaging, *Proceedings of IEEE International Electron Devices Meeting (IEDM)*, pp. 247–250.

Morita, Y. et al. (2006). A Vth-variation-tolerant SRAM with 0.3-V minimum operation voltage for memory-rich SoC under DVS environment, *Symposium on VLSI Circuits*, pp. 13– 14.

Nakagome, Y. et al. (2003). Review and future prospects of low-voltage RAM circuits, *IBM Journal of Reseach & Development* 47: 525–552.

Qin, H. et al. (2004). SRAM leakage suppression by minimizing standby supply voltage, *Proceedings of International Symposium on Quality Electronic Design (ISQED)*, pp. 55–60.

Roy, K., Mukhopadhyay, S. & Mahmoodi-Meimand, H. (2003). Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits, *Proceedings of the IEEE* 91(2): 305–327.

Seevinck, E., List, F. & Lohstroh, J. (1987). Static-noise margin analysis of MOS SRAM cells, *IEEE Journal of Solid-State Circuits* 22(5): 748–754.

Singhee, A. & Rutenbar, R. (2007). Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application, *Proceedings of Design, Automation & Test in Europe Conference & Exhibition DATE '07*, pp. 1–6.

Singhee, A. et al. (2008). Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design, *Proceedings of 21st International Conference on VLSI Design VLSID 2008*, pp. 131–136.

Verma, N. & Chandrakasan, A. (2008). A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy, *IEEE Journal of Solid-State Circuits* 43(1): 141–149.

Wang, A., Calhoun, B. H. & Chandrakasan, A. P. (2006). *Sub-Threshold Design for Ultra Low-Power Systems*, Springer.

Wang, J. & Calhoun, B. (2007). Canary replica feedback for near-drv standby $V_{DD}$ scaling in a 90nm SRAM, *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)*, pp. 29–32.

Wang, J. & Calhoun, B. H. (2008). Techniques to extend canary-based standby $V_{DD}$ scaling for SRAMs to 45 nm and beyond, *IEEE Journal of Solid-State Circuits* 43(11): 2514–2523.

Wang, J., Singhee, A. et al. (2007). Statistical modeling for the minimum standby supply voltage of a full SRAM array, *Proceedings of European Solid State Circuits Conference (ESSCIRC)*, pp. 400–403.

Wang, Y. et al. (2007). A 1.1GHz 12uA/Mb-leakage SRAM design in 65nm ultra-low-power CMOS with integrated leakage reduction for mobile applications, *IEEE International Solid-State Circuits Conference*, pp. 324–606.