0.2 V 8T SRAM With PVT-Aware Bitline Sensing and Column-Based Data Randomization

Do, Anh Tuan; Lee, Zhao Chuan; Wang, Bo; Chang, Ik-Joon; Liu, Xin; Kim, Tony Tae-Hyoung

2016

Do, A. T., Lee, Z. C., Wang, B., Chang, I. J., Liu, X., & Kim, T. T. H. (2016). 0.2 V 8T SRAM With PVT-Aware Bitline Sensing and Column-Based Data Randomization. *IEEE Journal of Solid-State Circuits*, 51(6), 1487-1498.

<https://hdl.handle.net/10356/84030>

<https://doi.org/10.1109/JSSC.2016.2540799>

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [Article URL/DOI].

Downloaded on 26 Apr 2023 18:15:53 SGT

0.2 V 8T SRAM with PVT-Aware Bitline Sensing and Column-based Data Randomization

Anh Tuan Do, *Member, IEEE*, Zhao Chuan Lee, Bo Wang, *Student Member, IEEE*, Ik-Joon Chang, *Member, IEEE*, Xin Liu, *Member, IEEE* and Tony T. Kim, *Senior Member, IEEE*

Abstract—In sub/near-threshold operation, SRAMs suffer from considerable bit-line swing degradation when the data pattern of a column is skewed to ‘1’ or ‘0’. The worst scenarios regarding this problem occur when the currently read SRAM cell has different data compared to the rest cells on the same column. In this work, we overcome this challenge by using a column-based randomization engine (CBRE). This CBRE circuit randomizes data stored to SRAM. This makes distribution of “1” and “0” in each column close to 50%, significantly increasing bit-line swing. To further improve the bit-line swing, we employ bit-line boost biasing and dynamic bit-line keeper schemes. Based on the mentioned techniques, we fabricated a 256 rows $\times$ 128 columns (32Kb) 8T SRAM array in 65 nm CMOS technology. In our silicon measurement, the SRAM array shows successful 200mV operation at room temperature, where energy consumption and access time are 1 pJ and 2.5 μ s, respectively.

Index Terms—8T SRAM, bitline sensing, data randomization, sub-threshold circuit, near-threshold circuit, PVT-aware design

I. INTRODUCTION

With the explosive market growth of portable electronics such as tablets and smartphones, low power consumption becomes of utmost importance to system designers [1]. Researchers have presented many low power techniques in previous literature [2, 3]. However, these techniques may not provide sufficient power saving for several applications such as wearable devices and wireless sensor nodes. These applications require ultra-low power consumption since they are operated under extremely small-size battery environment. Recently, the authors of [2] find that the energy-optimal point of digital circuits tend to appear near the threshold voltage of transistors. This motivates sub- or near-threshold operations as possible

solutions to deliver the ultra-low power demand. However, in sub-100nm technologies these ultra-low voltage operations result in many design challenges. Most of all, it is extremely difficult to obtain secure ultra-low voltage operations in SRAM. In sub- or near-threshold operations, transistor current is extremely sensitive to *process, voltage and temperature* (PVT) variations. Then, read and write stabilities of the conventional 6T SRAM become very poor at their worst PVT corners [4]. In the 6T SRAM, read and write stabilities have contradicting design requirements [5], making it difficult to improve these stabilities at the same time. We may mitigate the above problems by employing higher SRAM supply voltage (V_{DD}) compared to that of logic circuit. Then, level-shifters are necessary to interface between the logic circuit and the SRAM. Furthermore, more voltage regulator should be added in order to generate different V_{DD} 's for the logic core and the SRAM. These extra circuits incur significant power penalties, harming the energy efficiency of the whole system. Hence, researchers have made various efforts to aggressively scale down the supply voltage of SRAM. For instance, alternative SRAM cells suitable for sub- or near-threshold operations have been developed. Some representative SRAM cells are shown in Fig. 1, where read and write ports are separated by adding more transistors. Here, data nodes are fully decoupled from read access, delivering good read stability in ultra-low voltage operations. The write stabilities of these SRAM cells are enhanced by utilizing some techniques such as collapsed cell V_{DD} [6], boosted word-line voltage schemes [7], or negative bit-lines [8]. Among these decoupled SRAM cells, the 8T cell of Fig. 1(b) is the most popular owing to reasonable area overhead when compared to the 6T SRAM [9-15].

Small bit-line swing is another reason to impede robust ultra-low voltage operation of SRAM [16, 17]. As we scale down *supply voltage* (V_{DD}), on and off current ratio becomes smaller. This implies that in the ultra-low voltage operation, leakage noise due to unaccessed SRAM cells may deteriorate bit-line swing [13]. To address this challenge in the 8T SRAM, the authors of [18] employed sense amplifier redundancy. The redundancy reduces the offset of a sense amplifier, thereby improving sensing stability. However, in local bit-lines of 8T SRAM array the single-end sensing scheme based on domino-style circuit is more widely used, where the above technique may not be applicable.

In this work, we propose some techniques to enhance the sensing stability of 8T SRAM. We note that leakage noise has data dependency, which has the worst scenario when the currently read SRAM cell has different data compared to the rest cells on the same column. We randomize data stored to SRAM, namely *data randomization*. This makes that distribution of “1” and “0” in each column is close to 50% and

Anh Tuan Do is now with Institute of Microelectronics (IME), Agency for Science, Technology and Research (A*STAR), Singapore. He was with the School of EEE, Nanyang Technological University, Singapore 639798. (e-mail: doat@ime.a-star.edu.sg).

Zhao Chuan Lee and Bo Wang are with the school of EEE, Nanyang Technological University, Singapore 639798. (e-mail: e110002@e.ntu.edu.sg; bwang6@e.ntu.edu.sg).

Ik-Joon Chang is with the Department of Electrical and Electronic Manufacturing, Kyunghee University, Yongin-si, Gyeonggi-do, 446-701 Korea (e-mail: ichang@khu.ac.kr).

Xin-Liu is with Institute of Microelectronics (IME), Agency for Science, Technology and Research (A*STAR), Singapore (e-mail: liux@ime.a-star.edu.sg).

Tony T. Kim is with the School of EEE, Nanyang Technological University, Singapore. (e-mail: thkim@ntu.edu.sg).

Fig. 1 (a) Conventional 6T SRAM cell and several decoupled SRAM cell designs for ultra low-voltage applications, (b) Single-ended 9T, (c) Single-ended 8T, and (d) 10T.

Fig. 2 A typical 8T SRAM architecture with decoupled read port.

thus we are able to avoid the worst scenario with respect to leakage noise. We combine *data randomization* with the *bitline boosting* of [19], which is further enhanced by using dynamic bitline keeper. We verify our techniques through extensive simulations and actual silicon measurements. The results show that our techniques lead to robust ultra-low voltage operation of 8T SRAM.

The rest of the paper consists of five sections. In section II, we discuss issues and existing circuit techniques associated with the read port of the 8T SRAM design. Section III proposes our column-based data randomization design, followed by the dynamic bitline keeper design in section IV. Section V presents our measurement results. Section V concludes the paper.

II. BL LEAKAGE IN CONVENTIONAL 8T SRAMS AND EXISTING LEAKAGE COMPENSATION TECHNIQUES

A. Read Bit Line leakage issues in 8T SRAM

Fig. 2 shows a typical 8T SRAM organization. Each row of cells has a *Read Word Line* (RWL) attached to the read ports and a *Write Word Line* (WWL) attached to the write port of the 6T element. Each column has pair of complementary Write Bit Line (WBL and WBLB) for writing into the cell and a Read Bit Line (RBL) for reading. Write operation of the 8T SRAM is the same as that of the conventional 6T SRAM [9], whose read operation is performed through the read port. During standby,

all RBLs are pre-charged to V_{DD} by using pre-charging PMOS devices (Fig. 2). When the read cycle is activated, the pre-charging PMOSs are turned off, leaving RBLs floating. It is then discharged to ground if the accessed cell stores “0” and stays at V_{DD} otherwise.

As illustrated in Fig. 2, there are four currents associated with the read bitlines: I_{cell1} , I_{cell0} , I_{leak1} and I_{leak0} , where I_{cell1} and I_{cell0} are the read currents sunk by the accessed cell when it stores a “1” and “0”, respectively. Note that when a cell stores “1” (i.e. $Q = “1”$), its complementary bit (i.e. $QB = “0”$) is applied to the gate of the pull down NMOS in the read port. Therefore, I_{cell1} is ideally zero and much smaller than I_{cell0} . Similarly, I_{leak1} and I_{leak0} are the leakage currents flowing through the other unaccessed cells which store “1” and “0”, respectively. Ideally, both I_{leak0} and I_{leak1} are zero and hence, the sensing circuit can easily differentiate between I_{cell1} and I_{cell0} . However in reality, leakage current is not negligible, which becomes more significant under low V_{DD} , high temperature and large number of cells per RBL. Hence, the leakage currents of unselected cells may discharge RBL. For instance, if the selected cell stores a “1”, (i.e. $Q = “1”$), $QB = “0”$ is applied to the read port. Ideally, RBL is not discharged under this condition. However, in reality $I_{cell1} + I_{leak-total}$ considerably discharges RBL, where $I_{leak-total}$ is the total leakage contributed by unselected cells on the same column. To assure correct sensing despite of this problem, $I_{cell0} + I_{leak-total}$ should be significantly larger than $I_{cell1} + I_{leak-total}$ so that the sensing circuit can easily detect their difference.

However, as we scale down V_{DD} to save power, on-current (I_{on}) decreases more quickly compared to off-current (I_{off}), shown in Fig. 3(a). As a result, the I_{on}/I_{off} ratio, which is required to be very high for reliable sensing, quickly diminishes in ultra-low voltage region. For example, at 1 V supply voltage, I_{on} is 110× larger than I_{off} while they are almost comparable at 0.35V, as shown in Fig. 3(b). This makes that in such low voltage operation it is difficult to deliver the above condition.

We need to note that $I_{leak-total}$ is dependent on data stored to SRAM cells, making this situation worse. In Fig. 2, I_{leak1} flows through two stacked off-transistors while I_{leak0} has single off-transistor in the leakage path. Since in the stacked leakage path leakage current is exponentially reduced, I_{leak0} is much smaller compared to I_{leak1} . Fig. 4 shows two extreme cases considering the data-dependency of leakage current. In the column of Fig. 4 (a), QB’s of the accessed cells are “1” while QB’s of all unaccessed cells are “0”. Under this circumstance, $I_{leak-total}$ has the minimum value and hence, we express $I_{leak-total}$ of Fig. 4 (a) as $(n-1) \times I_{leak1}$ where n is the number of SRAM cells per column. On the other hand, in the column of Fig. 4 (b), QB’s of all unaccessed cells are “1” and hence, $I_{leak-total}$ has the maximum value, therefore described as $(n-1) \times I_{leak0}$. To obtain correct sensing regardless of SRAM data pattern, the following equation should be satisfied.

$$I_{cell0} + (n-1) \times I_{leak1} > I_{cell1} + (n-1) \times I_{leak0} \quad (1)$$

Under low V_{DD} , high temperature and large number of cells per RBL, it is pretty difficult to deliver equation (1). This is due to the fact that these circumstances make I_{on}/I_{off} ratio small, as mentioned above. Fig. 5 shows this well, the discharging waveforms of RBL under 0.3 V, 80 °C, and 256 SRAM cells per bitline. We perform the simulations for the best and the

Fig. 3 Transient bitline waveforms and bitline swings of the best-case and worst-case of 8T SRAM during read at 0.3 V, 80 °C, 256 cells per bitline.

Fig. 4 Worst case scenarios during read in an 8T SRAM at 0.3V supply with 256 cells per column. (a) Worst case read “0”, all unselected cells are “1” (b) Worst case read “1”, all unselected cells are “0”.

worst data patterns. In the best data pattern, it is easy to perform data sensing while for the worst one, equation (1) is not satisfied and hence, it is difficult to assure correct sensing. Therefore, compensation techniques are needed to eliminate the effect of excessive leakage noise along the RBL.

B. Existing BL leakage compensation schemes

Several works have been reported to tackle the issues associated with large RBL leakage [20-23]. Their common approach is to measure the leakage during standby and inject a similar current to the active bitlines during the read cycle. For example, in [23], a diode connected PMOSs are used to pre-charge the BL during standby (P_2 and P_4 in Fig. 6(a)). Voltages at the drains of these transistors represent the level of leakage current sunk by the entire column of cells on the BLs. At the end of the standby period, this voltage is sampled and stored on two capacitors and subsequently applied to the pull-up PMOS P_6 and P_8 to inject small currents to the BLs during the read operation. If P_6/P_8 and P_2/P_4 are matched, then the bitline leakage currents can be compensated. However, this technique cannot be used when the supply voltage is near or lower than the threshold voltage, as the voltage drop across the PMOS cannot be reliably sensed. In addition, the peripheral circuitry required for each bitline costs a significant area overhead for the SRAM.

Fig. 5 Transient bitline waveforms and bitline swings of the best-case and worst-case of 8T SRAM during read at 0.3 V, 80 °C, 256 cells per bitline.

Fig. 6 Existing BL leakage compensation schemes. (a) using diode-connected PMOS and capacitors to store the BL leakage level [20]. (b) a pair of PMOS is used for each column as static load to provide pull-up current against the bitline leakage [21]. (c) using replica BL [22].

In [22], a pair of PMOS is used for each column as static load to provide pull-up current against the bitline leakage in 6T SRAM design (Fig. 6(b)). Note that leakage on each bitline is different and finally each bitline will reach an equilibrium level. Voltage difference across each bitline pair at the end of the standby period is regarded as the offset of the sense amplifier. This voltage offset is recorded on two capacitors and will be cancelled using additional control signal Φ_1 and Φ_2 , as shown in Fig. 6(b). Due to the similar reason to [23], this design

is not suitable for ultra-low voltage operation. Furthermore, it is only applicable for differential sensing, which is not the case of 8T SRAM cell design.

Kim *et al.* proposes a marginal RBL leakage compensation technique by means of replica bitline circuit [24], shown in Fig. 6(c). Initially, all bit-line pull-up control signals (i.e. $PCHG_{[0:3]}$), are set to “1” to turn off all pull-up devices of the dummy RBL. They are sequentially switched to “0” to increase the strength of the pull-up current until it is just enough for the output of the SA to switch from “0” to “1”. Since the dummy bitline was hardwired to the worst-case scenario (i.e. all “0”), it indicates the optimum level to pre-charge RBL in the main array. As a result, each RBL is tuned to settle just above the trip point of the sense amplifier so that when logic “0” is read, only a small swing is needed to change the sense amplifier output.

However, this approach may not account for the process variations that affect the trip point of each sense amplifier. Furthermore, the worst case scenario used in the replica RBL leads to more currents to be sourced from the pull-up devices of the main array, which can potentially hurt the pull-down path if the accessed cell stores a “0” and the rest of the column store “1”.

III. COLUMN-BASED DATA RANDOMIZATION

A. Column-based data randomization concept

Our principal idea is to make the percentage of “1” and “0” in each column close to 50% so that bitline leakage is reduced and becomes independent of the input patterns. As a consequence, bitline sensing margin is improved. It also allows a given SRAM structure to widen its operation range to more extreme conditions where bitline leakage is comparable to the active current.

This concept is realized by shuffling specific data pattern such as in image processing applications so that we can eliminate the worst case scenarios of the read bitlines. In video encoders, pixels are processed in blocks (e.g. 16x16 in MPEG-2) which are stored in on-chip SRAM. Since most image frames have smooth backgrounds or large objects, nearby pixels tend to be similar (i.e. their MSBs are the same). For example in Fig. 7, the maximum intensity difference of all pixels in the green square is only 21 LSB (Fig. 7(b)). Because of that, MSB columns are most likely to have skewed data, i.e. majority of cells on the same column are ‘1’ or ‘0’. In extreme cases, an entire column may store “0” or “1”, except for only a few cells. Similar conclusion can also be applied to other applications where data in consecutive words change slowly such as image processing. If this data pattern is written into conventional 8T SRAM, reading them at low-voltage condition is challenging. On the other hand, if we can pseudo-randomly toggle consecutive data before writing, what actually stored in the memory will be approximately 50% one and 50% zero. Furthermore, if the randomization process is simple enough, output data can be reversed back to its original value without consuming too much power overhead.

There are several ways to toggle input data before writing into the main memory. Linear feedback shift register (LFSR) is one of them. LFSR is a popular hardware-based pseudorandom pattern generator [25]. It has also been used in Flash memory [26] to improve its endurance by redistributing data pattern

along the WLs. The rule to shuffle data in Flash memory is

Fig. 7 Pixels belonging to the same object or background have similar intensities. (a) Lena image (b) intensity histogram of the pixels within the green square. This lead to skewed data along the read bitline of the MSBs.

Fig. 8 Data redistribution using column-based randomization. (a) data before randomization. (b) data after randomization.

different from row to row and therefore a long random sequence is required from the LFSR. In SRAM, we need to randomize data only along the column, not the row. Furthermore, the same rule can be used in different columns. Therefore, our implementation is different from that of Flash memory. In SRAM, if skewed data flow through a randomization engine and bits are randomly flipped, the output sequence of “1” and “0” will be equally distributed. As a result, if one looks at data pattern along a bitline, the worst case scenarios are transformed to the average case where number of “0s” and “1s” are almost the same. Fig. 8 compares the percentage of “1” in a simulated RBL during 20000 successive write cycles. Fig. 8(a) shows that when 99% of input data are “1”, percentage of “1” along the BL also approaches the same number in conventional SRAM. Similar conclusion can also be made when percentage of “1” in the input data is 50% and 1%, respectively. However, if this data pass through a randomization engine, density of “1” only marginally fluctuates around 50%, as shown in Fig. 8(b), regardless of the original data distribution. Thus, its temporal and spatial BL leakage variations are significantly reduced.

Other than minimizing the bitline leakage variations, data randomization also offers a more uniform bitline delay. For 8T SRAM structure, the best delay in reading zero happens when all unselected cells store “0” while the worst delay happens when they store “1”. As illustrated in Fig. 9, the worst case delay can be up to 50% longer than that of the best case. Note that the worst case delay defines the memory’s speed performance, regardless of how fast the best case is. Bitline for reading “1” is supposed to stay at V_{DD} thus the concept of delay is not applied

to it. With randomization, bitline delay varies less than 5% around the median value, also shown in Fig. 9.

Even though the proposed technique randomizes the column effectively, there will be specific input patterns that generate skewed data after passing through the randomization engine. However it is very unlikely if the input pattern is skewed or has long sequences of continuous “0” or “1” (for example: “00001110000”). In practice, we can avoid this situation by applying the proposed data randomization only to columns of MSBs where it is more likely to have long segments of the same data (i.e. nearby pixels of the image have the same MSBs). The LSBs can be left unchanged as the correlations of LSBs between nearby pixels are statistically low. Another approach is to include column-based registers to enable/disable the flipping logic which can be programmed in real time. In short, the pseudo-randomization scheme is more effective if input data have long sequences of “0” or “1” and designers must be aware of this point to avoid extreme case scenarios after randomization.

B. Circuit realization

In this implementation, XOR-gate are used realize our column-based randomization engine (CBRE). Row address is used to control CBRE, whose output is fed to the column-based flipping logic of the writing circuits. Each input data is flipped or kept unchanged depending on the output of the CBRE. Since we only concern about the data pattern along each column, row address can be used to decide whether a bit should be flipped. This implementation is sufficient to validate the improvement of our proposed idea. Furthermore, it offers a simple way to reverse the data during read operations. Gate level schematics of the CBRE and flipping logic are shown in Fig. 10. The XNOR gates ensure that if a long chain of consecutive “1” or “0” is written to a column, it will be transformed into a sequence of alternative “1” and “0”. As *Flip* signal controls the MUX, $\sim D_i$ will be selected if *Flip* signal is high. Otherwise, D_i will be transferred to the write driver. Fig. 11 illustrates how this scheme works on a sequence of six write operations to the same column. Assuming the input bit sequence is “111100” with six different address shown on the left. The address bits will be XOR to generate *Flip*. As can be seen, only the 1st, 4th and 5th bits are flipped while the rest are kept the same. As a result, the actual sequence writtento the column is “011010”. For video encoder or similar applications, this configuration can be applied to the MSBs. In this implementation, it is used in all columns for the purpose of characterization. Note that using this CBRE is only one way to implement the data-suffling concept. This approach also makes it very easy to switch it back to original data during the read operation and thus is more preferred implementation.

Since CBRE is shared between all columns, area overhead of the design only includes a MUX and inverter per column. As a result, it is less than 3% larger compared to the original design. Its write power overhead is less than 1%. Note that if the number of rows per column is increased, area and power overhead is even smaller.

Fig. 12 verifies the effectiveness of the toggling method using CBRE in Fig. 10. We assume that the green square (the Region of Interest, i.e. RoI) in Fig. 7 moves from top left corner of the Lenna image to the bottom right corner, from left to right and top down. The RoIs are non-overlapping. Each RoI contain

Fig. 9 Bitline discharge delay of the conventional 8T and randomized 8T.

Fig. 10 Gate-level flipping control and column-based flipping logic.

Fig. 11 Operating principle of the flipping control circuit.

16x16 pixels, each has 8-bit binary value. Each RoI is then written to a SRAM bank of 256x8 cells. The first column of the bank stores the MSB (i.e. bit 7) while the last column stores bit 0. As the RoI moves, data of the first 289 squares (i.e. 17 squares from left to right and 17 squares from top to bottom) are recorded and the percentage of “1” in each column of the SRAM bank is recorded and shown in Fig. 12(a). We only show bit 7, 6, 3, 1 and 0 for the sake of clarity. It can be seen that before toggling, the original Lenna image results in extreme cases of 100% or 0% of “1” in the MSB column. Similar observation can also be seen in the column containing bit 6. Only in columns containing bit 0 and bit 1 that the percentage of “1” varies around 50%. This agrees of our hypothesis in pervious section.

When the proposed toggling method using CBRE is applied to the same image, same writing procedures, we can see in Fig.

Fig. 12 (a) Percentage of “1” in each column of a 256 row x 8 bit cells SRAM bank before and after toggling when a block of 16×16 pixels are written in the bank. (b) Maximum and minimum percentage of “1” when different images are used.

Fig. 13 (a) Schematic of the boosted bitline schem [27] (b) timing diagram during a read operation of the conventional 8T and boosted bitline 8T. (c) simplified schematic of the bias voltage generator

12(a) that the maximum percentage of “1” is only 60% while the minimum is only 40%, which is close to the ideal case of 50%. Note that the improvement in the last two columns (i.e. bits 1 and 0) is not significant as its original data is already balanced. However, we can clearly see how it transforms the skewed data to almost 50% of “1” in the MSB columns.

Similarly, the we randomly choose 10 gray scale images from Google image search, performed the same procedures and observed the maximum and minimum percentage of “1” for each image. The results is shown in Fig. 12(b). All images have the minimum percentage of “1” equal to zero. This means at least one column of one bank when stores a RoI from these images contains all zeros. Most of them have maximum percentage of one equal to 100%, i.e. the whole column is “1”. On the other hand, the proposed method offers a good balance of “1” and “0” in every column, every banks since the maximum and minimum percentage of “1” is 62% and 38.6%, respectively.

Note that the randomization scheme does not reduce the BL leakage significantly, instead makes it almost the same for both reading “1” and “0”. As the BL leakage becomes a common component, it is easier to mitigate the impact of the BL leakage on sensing by providing a common BL leakage compensating current. However, when the BL leakage (even with 50% data pattern) is much larger than the actual cell current, it is difficult to generate sensible BL swing at ultra-low voltage condition, especially with process variation. Thus a BL length equal to or less than 256-row is recommended. Furthermore, we assume that nearby input image pixels are written continuously in the same column so that we can take advantage of the similarity of the data pattern along the column. This makes the proposed SRAM particularly useful in video/image processing applications.

IV. PVT-AWARE DYNAMIC BL KEEPER

A. BL-boosting current at ultra-low voltage operation

Reliable RBL sensing requires both large RBL swing and a wide sensing window. In[19], we proposed a PVT-aware bias voltage generator that helps to boosting both the RBL swing and BL sensing windows, especially at ultra-low voltage region. The proposed BL reading scheme and timing diagram of the proposed design is shown in Fig. 13. Unlike conventional 8T, during standby the read bitline (RBL) of the proposed design is discharged to ground instead of being charged to V_{DD} . This reduces BL leakage current when it is in idle mode. When read, *Read* signal turns on P2 to charge the RBLs up. If the

Fig. 14. Impact of the proposed boosting and randomization techniques on the BL swing when compared with the conventional BL structure.

Fig. 15 Schematic of the proposed dynamic BL keeper control circuit.

Fig. 16 Transient simulation waveforms of the dynamic bitline keeper control.

accessed cell stores a “1”, QB is low and thus I_{read1} is insignificant. As a result, RBL rises to a high voltage level (V_{RBL1}). If it stores a “0”, the corresponding V_{RBL0} will be only slightly higher than ground, due to the fighting between I_{boost} and I_{read0} (Fig. 13(b)). The stacked PMOS configuration shown in Fig. 13(a) is further biased by a Bias Voltage Generator [19] which automatically tunes to bias P1 so that the voltage gap between V_{BL1} and V_{BL0} is maximized. Simplified block diagram of the Bias Voltage Generator is shown in Fig. 13 (c). It consists of two dummy columns which represent the worst-case for read “0” and “1”, respectively. Initially, the down-counter (CT0) is at the maximum value, thus V_{ref0} is at

V_{DD} and V_{RBL0} is at ground. As the CT0 counts, V_{ref0} steps down and I_{boost0} increases. This process continues until $V_{\text{BL0-buffered}}$ turns high. This defines the minimum allowable value of V_{bias0} and at the same time RBL0-buffer blocks the clock to CT0 and locks the output bits (A0-A2). Similarly, the bottom loop tracks the maximum allowable value of V_{bias1} by using the up-counter. The locked values of A0-A2 and B0-B2 are fed to a second set of reference ladders to generate the final V_{bias} whose value is theoretically equal to $(V_{\text{ref0}} + V_{\text{ref1}}) / 2$. The area overhead of this circuit is less than 5% (including the dummy column). The counters (and thus V_{bias}) can be reset and tuned again if the operating condition changes abruptly. In this example, we only use 3 bits for demonstration.

Fig. 14 shows the impact of the proposed randomization and BL boosting techniques on the BL swing when compared to the conventional design. It can be seen that conventional design has both small sensing window and sensing margin, represented by similar data “0” and data “1” BL waveforms. With the BL boosting technique alone, sensing window is visibly extended but sensing margin is still small. If randomization is applied, the sensing margin is improved, signified by a bigger gap between data “0” and data “1”, when compared to the conventional design. However, its sensing window is as small. By applying both randomization and BL boosting, one can clearly see that both sensing margin and sensing window are improved. Therefore, the SRAM read operation becomes much more robust against both process variations and timing uncertainty.

B. Dynamic BL keeper for wide operating voltage range

The above read scheme generates an optimum biasing voltage for the bitline keeper to boost the bitline swing at ultra-low voltage condition. However, it consumes a direct current from V_{DD} to ground through the read ports of the access SRAM cells during the read operation. It is slower and consumes more power than the conventional design at high voltages. Therefore, it is only beneficial to the SRAM energy efficiency when the required operating voltage is extremely low. In this section, a PVT-aware leakage sensor is proposed to dynamically switch the RBL keeper on/off so that it can achieve both high speed, low power at high supply voltage domain while reliably operate at extremely low voltage condition.

TABLE I. DESIGN METRIC COMPARISON WITH VARIOUS ULTRA-LOW VOLTAGE SRAMS

	A-SSCC 2012 [4]	JSSC 2013 [17]	JSSC 2011 [15]	A-SSCC 2012 [16]	This work
Technology	65 nm	65 nm	65nm	65 nm	65 nm
Density	128 kb	32 kb	4 kb	16 kb	32 Kb
Transistor count	8T	9T	8T	9T	8T
Cell size	N.A	1.24 x 2.31 μm^2	1.4 μm^2	2.63x0.72 μm^2	0.52x2.6 μm^2
V_{DDmin}	0.37 V	0.28 V	0.25 V	0.26 V	0.2 V
Access time	N.A	4.55 μs (0.3 V)	500 ns (0.25 V)	0.85 μs (0.26 V)	2.5 μs (0.2 V)
Leakage current	N.A	0.05 μA (0.4 V)	300 μA (0.25 V)	1.4 μA	1.1 μA (0.2 V)
Min. Energy (E_{min})	21.2 pJ	0.57 pJ	N.A	2.07 pJ	1 pJ
Normalized E_{min}	162 aJ/b	278 aJ/b	N.A	126 aJ/b	31 aJ/b

Fig. 17 Normalized read energy per bit of the proposed dynamic BL keeper scheme when compared with the conventional design, room temperature, TT corner.

Fig. 18 Simulated leakage current of the on-chip leakage sensor, room temperature, TT corner.

Fig. 19 Output voltage, normalized to different VDD, of the on-chip leakage sensor at room temperature, TT corner.

Schematic design of the proposed dynamic BL keeper control is shown in Fig. 15, including a static leakage sensor and five standard logic gates. The leakage sensor replicates a real RBL configuration where P_0 serves as a RBL keeper and

128 pairs of N_1 - N_2 which act as a replica RBL in the off state. The gate of N_1 is grounded while that of N_2 is connected to V_{DD} to model the worst-case leakage configuration. Since all N_1 are turned off, the DC current flowing through this leakage sensor is small.

At a reasonably high voltage (above 0.4 V), I_{on} supplied by P_0 is strong enough to hold node A close to V_{DD} , indicating that leakage current along the RBL is insignificant. As a result, boost-disable (BD) signal is high and PreCharge (PREC) signal is active. During standby, PREC is low to precharge RBL to V_{DD} by turning on P_1 . During Read, PREC goes high to turn off P_1 , leaving RBLs floating and then conditionally discharged by the read current of the accessed SRAM cell similar to that of the conventional design.

Once the supply voltage is reduced to a very low level (i.e. below 0.35 V) or the operating temperature is high, BL leakage current dominates I_{on} . Therefore, V_A is discharged to a low level. Once V_A crosses the threshold voltage of the buffer, it signifies that BL leakage is too high. As a result, BD goes low and boost-enable (BE) signal goes high to enable the RBL boosting scheme mentioned in section IV.A. PREC is kept at V_{DD} to disable P_1 while the pre-discharge signal (PRED) follows RD to control P_3 in the same manner shown in Fig. 9. Note that around the borderline, both schemes perform similarly and the exact criteria to switch between these two schemes is not required. Fig. 16 illustrates the operation of the sensing circuit when V_{DD} changes from 0.4 V to 0.2 V. At 0.4 V, BD is high while BE is low and thus PREC signal is activated. At the same time, PRED is always high to turn off P_3 . When V_{DD} is reduced to 0.2 V, BE goes high and BD goes low so that PREC signal is deactivated and PRED controls the read operation of the memory. Fig. 17 summarizes the normalized read energy per bit of the proposed dynamic BL keeper. It can be seen that the BL boosting design only outperforms the conventional design when V_{DD} is below 0.4 V. The proposed dynamic BL keeper scheme closely tracks both BL boosting and conventional design and seamlessly switches between the two to achieve the lowest energy consumption possible over a wide range of operating condition.

It is important to ensure that power and area overhead of the sensing circuit is low. Fig. 18 shows the simulated current of the leakage sensor at various supply voltages. Since all N_1 are hardwired to the off state, its current consumption is essentially equal to the RBL leakage of one column, which is

Fig. 20 1000 iteration Monte-Carlo simulation of the BL waveforms and the inverter-based BL buffers during a read operation at 0.2 V. (a) Temperature = 27°C; boost bias = 25 mV. (b) Temperature = 80°C; boost bias = 10 mV. Boost bias voltage is generated by the on-chip genertor circuit which tracks V_{DD} and temperature changes

Fig. 21 Structure of the proposed design

Fig. 22 1000 iteration Monte-carlo simulation of the write operation at 0.2 V, room temperature. (a) Write WL waveform. (b) Q and QB of a cell when BL= 200 mV; BLB = 0 V. (c) Q and QB of a cell when BL= 200 mV; BLB = 30 mV with no error. (d) Q and QB of a cell when BL= 200 mV; BLB = 40 mV.

less than 0.3% of the total leakage of the whole memory. For example, its current consumptions at 1 V and 0.34 V are 209 nA and 32 nA, respectively. Its area is only 0.5% when compared to the main array area.

Although output of the sensor buffer is either V_{DD} or zero, voltage at node A can have any analog value which may cause a significant DC current to be dissipated by the first inverter of the output buffer because node A is static. Fortunately, at voltage higher than 0.45 V, V_A is very close to V_{DD} . It only starts to reduce around 0.4 V and stays around $V_{DD}/2$ when the supply voltage is around 0.3 V. Since it is well within the subthreshold region, direct current through the buffer is insignificant, as indicated by the total leakage current of the sensor in Fig. 18.

Because an inverter-based buffer is used to amplify the analog voltage at node A to full CMOS level (Fig. 15), in a very rare event when V_A equals to the threshold voltage of the buffer, its output will be neither V_{DD} nor zero but an intermediate voltage value. To minimize this, the gain of the inverter must be maximize around its threshold voltage. Fig. 19 presents the static normalized output voltage of the buffer at different supply voltages at room temperate. 1000 supply voltage values from 0.2 V to 1V were used to closely examine the behaviour of the sensor. The graph in Fig. 19 has three distinctive regions: (i) At high voltages, its output equals to V_{DD} . (ii) at very low voltages, its output value is zero. (iii) there is a very sharp jump from 1 to zero, which spans 9 mV from 331 mV to 340 mV on the x-axis. We have ran more compresensive simulations at different temperatures and our results gave similar curves. To avoid this, more buffer stages can be added to make the likelihood of V_A equal to multi-stage buffer extremely rare. In our simulation, by using four inverters, the span of the sharp rising slope is reduced from 9 mV to less than 2mV.

Fig. 20 shows the 1000 Monte-Carlo simulation of the BL waveforms and the inverter-based BL buffers during a

Fig. 23 Micrograph of the test chip fabricated in 65 nm CMOS process.

Fig. 24 Measured waveforms of the test chip using consecutive read-only of "110" at 0.2 V supply voltage, room temperature. The randomized SRAM works well while the conventional SRAM fails to read "1".

readoperation at 0.2 V. It can be seen that even with process variations, there are still enough input margin for the inverter to decide the correct BL level. To minimize process variations, larger transistor sizes are used for the inverters and it is sized so that the trip point is around half- V_{DD} at the target operating voltage. Note that at higher supply voltages, much larger BL voltage margin is available, thus the drift in the inverter trip point is not as critical as that at ultra-low voltage operation.

V. TEST CHIP AND MEASUREMENT RESULTS

The proposed design has been fabricated in a 65 nm CMOS process. Its block diagram is illustrated in Fig. 21 consisting of a row decoder, an array of 256 rows $\times$ 128 columns of 8T SRAM cell. The CBREs are shared within the chip and controls the flipping logic of all columns. In this work, to test the effectiveness of the boosted bitline scheme and the randomization, all columns are read/written at the same time and all flipping logic is activated. In practice, individual column-based flipping logic can be employed to individually enable some columns while the others can be disabled. This is to support our previous hypothesis that MSBs should be randomized while the LSBs should stay untouched.

In order for the SRAM to be written successfully at ultra-low voltage condition, we avoid the half-select disturbance issue by eliminating the bit-interleaving architecture. To test our proposed read schemes, all columns are written at the same time. Hierarchical WL architecture can also be used if only a fraction of columns are to be accessed. This approach is simpler and more stable than other advanced write-assist scheme and thus is more preferable to demonstrate our improved read operation. Fig. 22 shows the Monte-Carlo simulations of the write operation at 0.2 V, room temperature. It can be seen from Fig. 22(b-c) that when $BL = 200$ mV, $BLB = 0$ V or 30 mV, no write failure was observed. However, when BLB is pulled down to 40 mV (Fig. 22(d)), write failures start to occur. This indicates that the SRAM cell has the write margin of about 30 mV without any write-assist techniques at 0.2 V. However, to avoid data destruction due to the half-select issue, bit-interleaving must be avoided. It is worth mentioning that the combination of both BL boosting and randomization makes our design much more robust in ultra-low voltage regime when compared to the conventional design[9]. Note that the proposed boosting technique will not even work if $I_{read} + I_{leak-min} < I_{leak-max}$. In other words, the proposed boosting technique enhances the correct (but small) sensing margin. Therefore, positive sensing margin should be achieved before employing the proposed boosting technique, which is achieved by the proposed randomization technique in this work.

Micrograph of the testchip is shown in Fig. 23. Various test patterns were generated using Agilent Logic Analyzer to validate the advantage of the proposed design over different data distributions. The measured waveforms in Fig. 24 confirm that the SRAM based on the proposed data randomization scheme works properly at 0.2 V while the conventional design fails to read “1”. At 0.2 V, our fabricated SRAM has power consumption and access time of 0.7 μ W and 2.5 μ s, respectively. Fig. 25 compares their minimum V_{DD} ’s (V_{DDmin} ’s) for two different temperatures (27 and 80°C). At room temperature, the V_{DDmin} of the randomized SRAM is 50 mV

Fig. 25 Minimum supply voltage of the proposed and conventional 8T SRAM at different operating temperature.

Fig. 26 Measured total and leakage current of the proposed design at room temperature.

Fig. 27 Measured read access time and minimum energy point of the proposed design at room temperature

lower than that of the conventional one. With the rising of temperature, the possible V_{DDmin} ’s of both designs tend to become higher since bit-line leakage noise increases exponentially. At 80°C, the randomized SRAM allows 90mV further scaling compared to the conventional one, which is larger than 50mV of 27°C. This shows that our proposed scheme is more effective at highly leaking PVT corners, where bit-line leakage noise is more problematic.

As supply voltage scales down, the proposed design offers much lower power consumption, as shown in Fig. 26. At extremely low supply (i.e. $V_{DD} < 0.4$ V), its leakage power becomes comparable to dynamic power. Coupled with the fact that read access time increases exponentially in this region (Fig. 27(a)), total energy per access actually increases when supply voltage reduces to below 0.4 V. As a result, the SRAM obtains its minimum energy point of 1 pJ/access at 0.4 V supply voltage, as illustrated in Fig. 27(b).

We compare our measurement results to those of recently published low-voltage SRAMs, which is summarized in table I. As mentioned above, in our test-chip we add large number of SRAM cells to single RBL, deteriorating bit-line leakage noise. Nonetheless, our design shows the lowest V_{DDmin} , showing the effectiveness of our proposed scheme. In the comparison of the minimum energy per bit, expressed by normalized E_{min} in table I, our proposed design shows considerably lower energy dissipation compared to other ones. Our design does not pre-charge RBL, significantly reducing active leakage current. Considering that in ultra-low voltage operation, active leakage energy is almost comparable to switching energy, it is obvious that such an approach considerably improves total energy dissipation.

VI. CONCLUSION

A 32Kb 8T SRAM array with column-based data randomization for ultra-low voltage applications is presented. Densities of "0" and "1" along the bit-line are well-balanced, which makes the bit-line leakage virtually independent from the input pattern. Coupled with a BL boosting and dynamic BL keeper control, the proposed SRAM provides robust ultra-low voltage operations. We fabricate a test-chip of the propose SRAM in 65nm CMOS. In our measurements, the minimum operating energy of the SRAM is 1 pJ/read, which is obtained at 0.4 V.

REFERENCES

- [1] A. Kapoor, C. Groot, G. V. Pique, H. Fatemi, J. Echeverri, L. Sevat, M. Vertregt, M. Meijer, V. Sharma, P. Yu, and J. P. d. Gyvez, "Digital Systems Power Management for High Performance Mixed Signal Platforms," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 961-975, 2014.
- [2] H. Fuketa, D. Kuroda, M. Hashimoto, and T. Onoye, "An Average-Performance-Oriented Subthreshold Processor Self-Timed by Memory Read Completion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, pp. 299-303, 2011.
- [3] K. Zhang, U. Bhattacharya, C. Zhanping, F. Hamzaoglu, D. Murray, N. Valleppalli, W. Yih, B. Zheng, and M. Bohr, "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 895-901, 2005.
- [4] Y. Sinangil and A. P. Chandrakasan, "An embedded energy monitoring circuit for a 128kbit SRAM with body-biased sense-amplifiers," in *2012 IEEE Asian Solid State Circuits Conference (A-SSCC)*, , 2012, pp. 69-72.
- [5] A. Teman, A. Mordakhay, J. Mezhibovsky, and A. Fish, "A 40-nm Sub-Threshold 5T SRAM Bit Cell With Improved Read and Write Stability," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, pp. 873-877, 2012.
- [6] M. Yamaoka, N. Maeda, Y. Shinozaki, Y. Shimazaki, K. Nii, S. Shimada, K. Yanagisawa, and T. Kawahara, "90-nm process-variation adaptive embedded SRAM modules with power-line-floating write technique," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 705-711, 2006.
- [7] B. H. Calhoun and A. Chandrakasan, P. , "A 256-kb 65-nm Sub-threshold SRAM Design for Ultra-Low-Voltage Operation," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 680-688, 2007.
- [8] C. Y. Lu, C. T. Chuang, S. J. Jou, M. H. Tu, Y. P. Wu, C. P. Huang, P. S. Kan, H. S. Huang, K. D. Lee, and Y. S. Kao, "A 0.325 V, 600-kHz, 40-nm 72-kb 9T Subthreshold SRAM with Aligned Boosted Write Wordline and Negative Write Bitline Write-Assist," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, pp. 1-1, 2014.
- [9] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T Subthreshold SRAM Employing Sense-Amplifier Redundancy," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 141-149, 2008.
- [10] A. Teman, L. Pergament, O. Cohen, and A. Fish, "A 250 mV 8 kb 40 nm Ultra-Low Power 9T Supply Feedback SRAM (SF-SRAM)," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 2713-2726, 2011.
- [11] T. Suzuki, H. Yamauchi, Y. Yamagami, K. Satomi, and H. Akamatsu, "A Stable 2-Port SRAM Cell Design Against Simultaneously Read/Write-Disturbed Accesses," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 2109-2119, 2008.
- [12] S. Ishikura, M. Kurumada, T. Terano, Y. Yamagami, N. Kotani, K. Satomi, K. Nii, M. Yabuuchi, Y. Tsukamoto, S. Ohbayashi, T. Oashi, H. Makino, H. Shinohara, and H. Akamatsu, "A 45 nm 2-port 8T-SRAM Using Hierarchical Replica Bitline Technique With Immunity From Simultaneous R/W Access Issues," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 938-945, 2008.
- [13] L. Chang, R. K. Montoye, Y. Nakamura, K. A. Batson, R. J. Eickemeyer, R. H. Dennard, W. Haensch, and D. Jamsek, "An 8T-SRAM for Variability Tolerance and Low-Voltage Operation in High-Performance Caches," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 956-963, 2008.
- [14] M. Qazi, K. Stawiasz, L. Chang, and A. P. Chandrakasan, "A 512kb 8T SRAM Macro Operating Down to 0.57V With an AC-Coupled Sense Amplifier and Embedded Data-Retention-Voltage Sensor in 45 nm SOI CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 46, pp. 85-96, 2011.
- [15] J.-J. Wu, Y.-H. Chen, M.-F. Chang, P.-W. Chou, C.-Y. Chen, H.-J. Liao, M.-B. Chen, Y.-H. Chu, W.-C. Wu, and H. Yamauchi, "A Large σ - V_{th} /VDD Tolerant Zigzag 8T SRAM With Area-Efficient Decoupled Differential Sensing and Fast Write-Back Scheme," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 815-827, 2011.
- [16] B. Wang, T. Q. Nguyen, A. T. Do, J. Zhou, M. Je, and T. T. Kim, "A 0.2V 16Kb 9T SRAM with bitline leakage equalization and CAM-assisted write performance boosting for improving energy efficiency," in *IEEE Asian Solid State Circuits Conference (A-SSCC)*, 2012, pp. 73-76.
- [17] S. Lutkemeier, T. Jungeblut, H. K. O. Berge, S. Aunet, M. Porrmann, and U. Ruckert, "A 65 nm 32 kb Subthreshold Processor With 9T Multi-Vt SRAM and Adaptive Supply Voltage Control," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 8-19, 2013.
- [18] N. Verma and A. P. Chandrakasan, "A 65nm 8T Sub-Vt SRAM Employing Sense-Amplifier Redundancy," in *IEEE International Solid-State Circuits Conference (ISSCC), Digest of Technical Papers.*, 2007, pp. 328-606.
- [19] A. T. Do, T. Q. Nguyen, K. S. Yeo, and T. T. H. Kim, "Sensing Margin Enhancement Techniques for Ultra-Low-Voltage SRAMs Utilizing a Bitline-Boosting Current and Equalized Bitline Leakage," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, pp. 868-872, 2012.
- [20] K. Agawa, H. Hara, T. Takayanagi, and T. Kuroda, "A bitline leakage compensation scheme for low-voltage SRAMs," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 726-734, 2001.
- [21] T. T.-H. Kim, J. Liu, J. Keane, and C. H. Kim, "A 0.2 V, 480 kb Subthreshold SRAM With 1 k Cells Per Bitline for Ultra-Low-Voltage Computing," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 518-529, 2008.
- [22] Y. C. Lai and S. Y. Huang, "X-Calibration: A Technique for Combating Excessive Bitline Leakage Current in Nanometer SRAM Designs," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 1964-1971, 2008.
- [23] K. Agawa, H. Hara, T. Takayanagi, and T. Kuroda, "A bit-line leakage compensation scheme for low-voltage SRAM's," in *2000 Symposium on VLSI Circuits, Digest of Technical Papers*, 2000, pp. 70-71.
- [24] T. Kim, J. Liu, and C. H. Kim, "A voltage scalable 0.26V, 64kb 8T SRAM with V_{min} lowering techniques and deep sleep mode," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2008, pp. 407-410.
- [25] K. D. Wagner, C. K. Chin, and E. J. McCluskey, "Pseudorandom Testing," *IEEE Transactions on Computers*, , vol. C-36, pp. 332-343, 1987.
- [26] J. Cha and S. Kang, "Data Randomization Scheme for Endurance Enhancement and Interference Mitigation of Multilevel Flash Memory Devices," *ETRI*, vol. 35, p. 4, 2013.