A 40-nm Ultra-Low Leakage Voltage-Stacked SRAM for Intelligent IoT Sensors

Jingcheng Wang, *Member, IEEE*, Hyochan An, *Student Member, IEEE*, Qirui Zhang, *Graduate Student Member, IEEE*, Hun Seok Kim, *Member, IEEE*, David Blaauw, *Fellow, IEEE*, and Dennis Sylvester, *Fellow, IEEE*

***Abstract*—A stacked voltage domain SRAM is proposed as an effective leakage reduction technique where bit-cell arrays are split into two voltage domains (top and bottom) connecting in series between VDD and GND to generate a subthreshold retention voltage directly from a nominal supply with no area penalty or efficiency loss compared to the conventional voltage regulator approach. The Zigzag 8T bit-cell structure is chosen with an optimal transistor sizing to balance among hold stability, leakage, and area density. SRAM peripherals remains at full supply domain resulting in super-cutoff read for improved sensing margin and word-line overdrive for better write margin. A novel array swapping mechanism with a comprehensive timing control ensure stable access to arbitrary arrays within one system clock cycle. The proposed SRAM achieves 1.03-pW/b leakage at 0.58 V in 40 nm.**

***Index Terms*—Charge recycling, low leakage, SRAM, sub-/near-threshold, voltage stacking.**

I. Introduction

There is substantial recent interest in implementing deep learning techniques within IoT devices to enable intelligence in edge devices and avoid the need for expensive wireless communication to the cloud. In addition, off-chip DRAM accesses are costly for highly miniaturized and power-constrained devices. As a result, it is beneficial to fit entire neural network into on-chip memories, most commonly SRAM. Given their relatively low density, these memories can easily consume >80% of total chip area [1]. As a result, standby power of these battery-powered devices becomes dominated by SRAM leakage. For example, in the low-power, motion-triggered smart image sensor considered in this letter, the firmware, reference frame, and neural network weights require a total of 8.9-Mb SRAM that consumes up to 90% of the chip's standby power, dictating battery life.

Prior work has shown many leakage reduction techniques like the use of large long-channel thick-oxide transistors [2], [3], SOI devices with strong reverse body bias [4], [5], floating bit-line [6], raising VSS [7], [8], and lowering VDD [9], [10]. Beside using high VT transistor, which enables an order of magnitude leakage reduction and is readily deployed, lowering supply voltage is one of the most effective approaches to reduce leakage due to the DIBL effect. Scaling the bit-cell VDD to 0.3 V can further reduce array leakage by 11×. However, it raises two issues: 1) commercial bit-cells provided by foundries are not sized for holding data at very low voltages (e.g., subthreshold regime) and, therefore, require a redesign with careful hold margin/leakage/density tradeoff and 2) voltage regulation is required to generate a separate voltage level for SRAM arrays. LDOs are conventionally used, incurring area and power overheads due to efficiency loss. Voltage stacking provides an alternative way to generate an intermediate voltage level by placing voltage domains in series and has been previously used in microprocessors [11] and high bandwidth data buses [12].

Manuscript received August 16, 2020; revised November 9, 2020; accepted November 28, 2020. Date of publication December 9, 2020; date of current version January 20, 2021. This work was supported by Sony Semiconductor Solutions Corporation/Sony Electronics Inc. This article was approved by Associate Editor Vinayak Honkote. *(Corresponding author: Jingcheng Wang.)*

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 USA (e-mail: jiwang@umich.edu).

Digital Object Identifier 10.1109/LSSC.2020.3043461

In this letter, we show the first design and implementation of voltage stacking technique applied to SRAM arrays without additional voltage regulator to lower the bit-cell leakage. Section II describes the basic concept of voltage stacking and novel array swapping mechanism. Bit-cell design is covered in Section III, followed by the bank architecture and memory control in Section IV. Section V presents the measurement results.

II. Voltage Stacking and Array Swapping

Voltage stacking is a high-efficiency power delivery method by stacking two sets of circuits with similar current load in series. Voltage is divided between the two and charges flow through the top set is recycled by the bottom sets, enabling a high ratio voltage down conversion with only nominal supply, without need for a low efficiency on-chip regulator. The main challenge of voltage stacking is to balance the active current between top and bottom domain. To maintain a stable mid-rail voltage, it usually requires an additional small voltage regulator for mid-rail, reducing some power benefits gaining from stacking. SRAM arrays, however, are dominated by near-constant leakage current (writing a bit only draws 10 s of pA average active current, negligible compared to μA-level background leakage), making them ideal for voltage stacking.

Since the SRAM, we developed in this letter, targets to low-activity IoT sensors which would stay in sleep most of the time, the active power reduction is not the main concern. Besides, to prevent load imbalance caused by instantaneous large current during active operation, SRAM peripheral and other logic circuits on-chip are not stacked as shown in Fig. 1 (peripheral can be frequently power gated to reduce leakage) and, therefore, word-line and bit-line voltages remain at VDDcore (∼2VDDmid) resulting in an inherent write/read noise margin enhancement. This also removes the need for level converters between SRAM interface and the rest of the chip, but, as a result, only bottom arrays can be accessed directly. To allow arbitrary address access while maintaining a stable mid-rail voltage, we propose a new array swapping mechanism. When a top array is accessed, it will first be swapped with a bottom array in the same quad-array SRAM bank. This swap mechanism ensures the leakage current from top and bottom sets remains balanced at all times. Thanks to the low clock frequency of IoT processers (100 kHz), the swap operation can be completed in one system clock cycle and allows all bit-cell arrays to remain at their minimum retention voltage for maximum leakage reduction. Beside the leakage reduction from voltage scaling, stacking offers an additional 2× leakage reduction due to the body effect and reduced bit-line leakage in top arrays.

III. Bit-Cell and Super-Cutoff Read

To get a robust bit-cell operating in subthreshold regime, we choose the Zigzag 8T structure [13]. It decouples read/write operation like a traditional 8T bit-cell, while its differential sensing provides faster read speed and larger sensing margin at low voltage. Fig. 2 shows the bit-cell schematic and layout whose area is almost the same as traditional 8T. The cross-coupled 4T and 2 write port transistors uses HVT to minimize leakage while LVT devices in the read port provide faster sensing speed. During the read operation, the read word-line (RWL) of the selected row will be pulled to ground by the WL driver, while

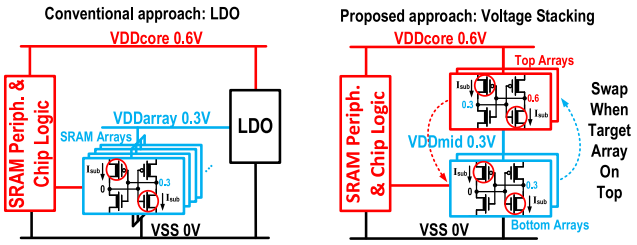

Fig. 1. Conventional approach (left), proposed voltage stacking, and array swapping technique (right).

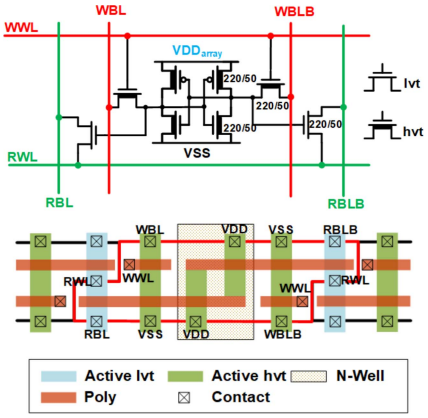

Fig. 2. Bit-cell schematic (top) and layout (bottom).

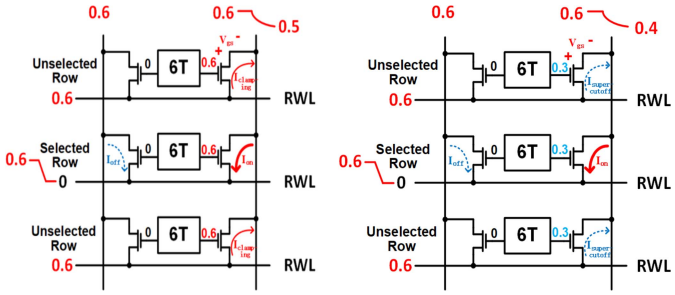

Fig. 3. Clamping current issue in original Zigzag 8T read operation (left), super-cutoff read limits clamping current on the bit-line (right).

RWLs of all the other unselected rows remains at full VDD (i.e., 0.6 V). Then the read current, I_{on}, of the selected cell begins to pull the read bit-line (RBL) low. However, at the same time, the read transistors of unselected rows may contribute undesired current flowing back onto the RBL as their positive V_{GS} increases, shown in Fig. 3. This current will clamp the bit-line to a certain voltage level which may result in a read failure if the sense amp offset is larger than RBL voltage drops. The worst-case clamping current occurs when all the cells in the column store the same value. However, in our stacked SRAM, since the array voltage (VDDmid) is $\sim 1/2$ the bit-line voltage (VDDcore), it gives the RBL at least half VDD swing before the clamping current problem can happen, as all the unselected cells are super-cutoff with negative V_{GS} when bit-line voltage is still higher than VDDmid. As a result, RBL in our design can drop to a much lower level and offers better sensing margin. Besides, during the write operation, since write word-line (WWL) is over-driven to a peripheral voltage twice as large as array voltage, it also largely improves write margin and write speed.

As mentioned before, bit-cell sizing requires a careful hold noise margin, leakage power, and area density tradeoff analysis. First, bit-cell needs to be upsized for improved hold noise margin (HNM) which can yield a lower leakage due to lower retention voltage.

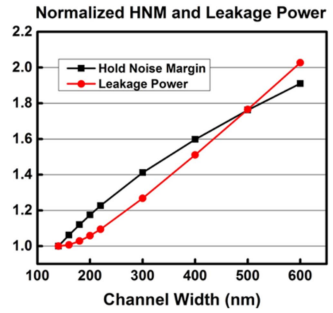

Fig. 4. Hold noise margin and leakage power versus bit-cell sizing.

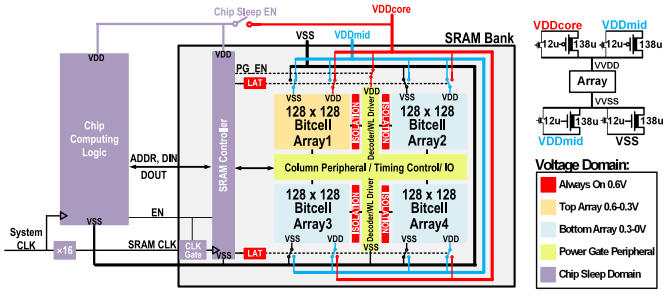

Fig. 5. SRAM bank architecture (left), and array header and footer (top-right).

However, the larger bit-cell incurs an area penalty and may lead to larger leakage compared to the denser and less robust bit-cell. In our design, channel length is increased to 50 nm where leakage is minimum (18% less), also improving HNM by 10% while incurring only 8% area penalty. In Fig. 4, we observe that as channel width is increased, HNM improves faster than leakage power, providing a favorable tradeoff for channel width between 200 nm and 400 nm. The final bit-cell width is chosen to be 220 nm to balance among HNM, leakage, and area density. As shown in Fig. 2, the size ratio among pull-up (PU), pull-down (PD), and pass-gate (PG) transistor is 1:1:1. PU and PD are sized the same for improved HNM. Since WWL voltage over-drive already helps boost write margin exponentially, we can size PG the same as PD for a higher area density.

IV. Bank Architecture and Swapping Control

Fig. 5 shows the architecture of one SRAM bank and voltage domain of each part. Each SRAM bank contains power-gated row/column peripheral, an SRAM controller, always-on configuration latches, and 4 bit-cell arrays with headers and footers that connect them either between VDDcore and VDDmid or VDDmid and VSS. SRAM peripheral operates under the full VDD, and they are power-gated to each system CLK cycle ($\sim$ 2 us) to save leakage. The controller operates under the same voltage domain as the rest of chip and goes into sleep when chip is idle. It runs under a gated SRAM CLK domain that is 16 times faster than the system CLK, because the function of the controller is to wake up the peripheral (release the power gates, isolation, and reset signal) before accessing the array and then put the peripheral back to gated and isolated state immediately afterward within a single system CLK period.

We do not allow direct access to a top array for two reasons.

1) Since the virtual ground of the top array bit-cell is VDDmid, in a write operation where we drive WBL/WBLB to either VDDcore or VSS, VSS can be short with bit-cell virtual ground, namely, VDDmid, through pass-gate and pull-down nMOS transistors.
2) Since top array bit-cells hold either VDDcore and VDDmid in its internal storage nodes and RBL/RBLB are precharged to VDDcore, we will not be able to have the benefit of super-cutoff read as mentioned before and read noise margin will be compromised.

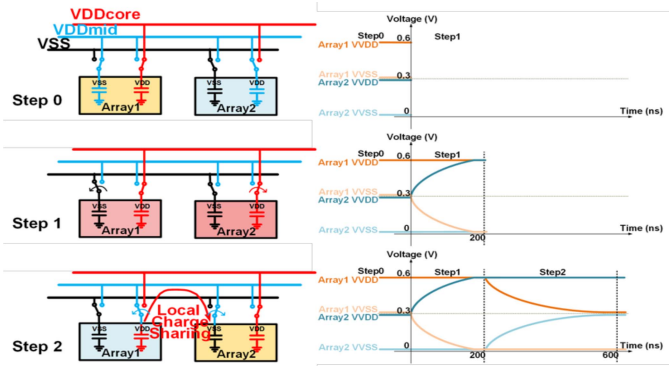

Fig. 6. Two-step array swapping mechanism.

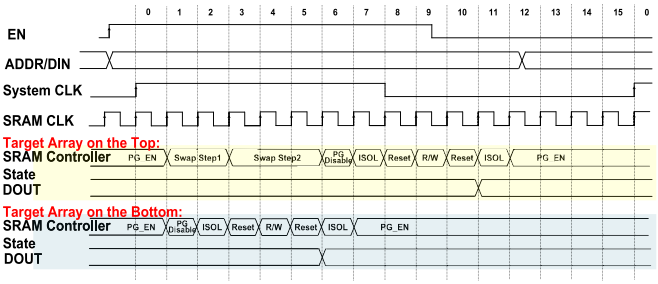

Fig. 7. Timing diagram of SRAM controller.

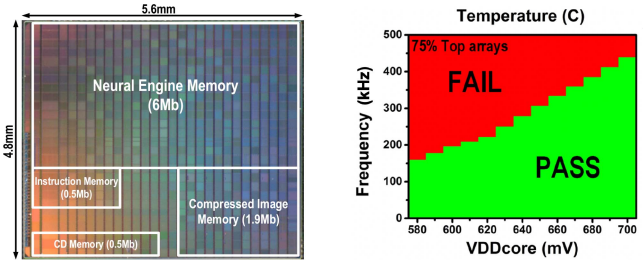

Fig. 8. Die photograph and shmoo plot.

Therefore, if memory address targets to a top array, the SRAM controller will automatically find a bottom array in the same bank and swap it with the top array before the access.

As illustrated in Fig. 6, the swap process takes two steps. Initially, target array1 is at top and swapping array2 is at bottom domain. In step1, the two arrays are expanded to full voltage range by switch array1's footer to VSS and array2's header to VDDcore. In step2, they collapse to the appropriate half range by turning both array1's header and array2's footer to VDDmid. Since the two arrays are in the same bank and physically close to each other, local charge sharing minimizes the disturbance to the mid-rail. To smooth transition and reduce inrush current and coupling noise, each power switch consists of a pair of small (12 um) and large (138 um) headers/footers that are turned on in sequence.

Fig. 7 shows the SRAM controller timing diagram. The chip operates in the low frequency system CLK while SRAM controller operates under a 16× faster SRAM CLK synchronous to the system CLK. When accessing a top array, the controller decodes the address and picks a bottom array to swap in SRAM cycle 0. Swap step1 starts in cycle 1. After waiting one extra cycle for the array supply/ground rail to settle down, swap step2 starts in cycle 3. It takes more time for mid-rail voltage of the two arrays to stabilize. Thus, array access will not happen until five cycles later. Three cycles before the access, controller starts to power up the peripheral, release isolation, and reset signals. Then, in cycle 9, read/write operation is performed, after

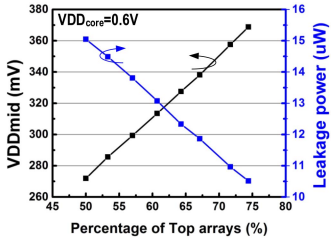

Fig. 9. Mid-rail voltage and leakage versus percentage of top arrays.

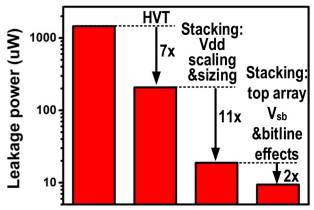

Fig. 10. Leakage reduction contribution.

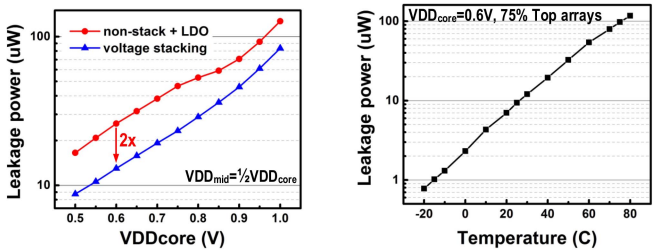

Fig. 11. Memory retention leakage across voltage and temperature.

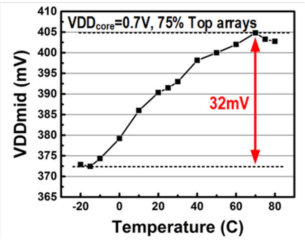

Fig. 12. Mid-rail voltage across 100 degree.

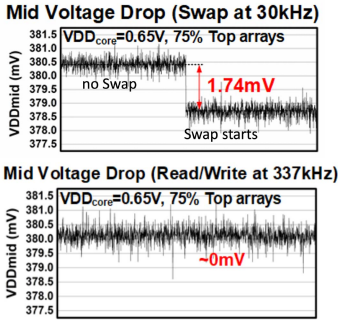

Fig. 13. Mid-rail voltage droop in different memory activities.

which arrays are isolated and peripheral is shut down in three cycles. If it is a read operation, the data is captured and outputted by the controller in the 11th cycle of SRAM CLK. When accessing a bottom array, swap steps are skipped, saving 5 cycles. The output data shows up in the 6th cycle of SRAM CLK.

Programmers can choose the number of arrays allocated to the top and bottom domains by writing to the configuration latches in each bank. Latches are under always-on domain to retain the configuration

	This work	ISSCC10 [2]	ISSCC14 [3]	VLSI13 [4]	VLSI17 [5]	JSSC09 [8]	JSSC08 [10]
Process	40nm	180nm	65nm Thick Oxide	28nm FDSOI	65nm SOTB	130nm	65nm
Cell type	8T	10T	6T	6T	6T	8T	8T
Cell Area (um^2)	0.82*	17.48	2.159	0.12	0.5408	0.61	-
On-chip SRAM Capacity	8.91Mb	24kb	128kb	1Mb	8Mb	64kb	256kb
Leakage	1.03pW/bit (0.58V)	3.3fW/bit (0.4V)	32.4fW/bit (1.2V)	300fW/bit (0.6V)	13.7fW/bit (0.5V)	70pW/bit (0.23V)	6.45pW/bit (0.3V)
Extra Supply Level Required	No	Yes	No	Yes	No	Yes	No
Body Bias Voltage Required	No	No	No	-1.5V	-2V	No	No
Access time	143ns (0.7V)	13700ns (0.4V)	7ns (1.2V)	-	31.4ns (0.75V)	66.7ns (0.6V)	1250ns (0.6V)
Access Energy	67fJ/bit (0.6V)	-	195fJ/bit (1.2V)	-	224fJ/bit (0.75V)	25fJ/bit (0.4V)	-
*Logic design rule							

	JSSCC11 [13]	VLSI17 [14]	JSSC11 [15]	JSSC13 [16]	ISSCC13 [17]	ISSCC14 [18]	VLSI13 [19]
Process	90nm	55nm	40nm	28nm	28nm	28nm	20nm
Cell type	8T	6T	9T	6T	6T	6T	6T
Cell Area (um^2)	-	0.803	1.058	-	0.12	-	-
On-chip SRAM Capacity	64kb	16kb	8kb	512kb	2Mb	32kb	128kb
Leakage	305pW/bit (0.23V)	97.6pW/bit (0.2V)	6.25pW/bit (0.4V)	10.6pW/bit (0.8V)	3.5pW/bit (0.7V)	109pW/bit (0.33V)	33.6pW/bit (0.6V)
Extra Supply Level Required	No	Yes	No	No	Yes	No	No
Body Bias Voltage Required	No	No	No	No	No	No	No
Access time	111ns (0.3V)	250ns (0.25V)	1000ns (0.4V)	0.42ns (1V)	-	-	1.16ns (0.9V)
Access Energy	1.15pJ/bit (0.5V)	5fJ/bit (0.25V)	11.3fJ/bit (0.4V)	-	-	93fJ/bit (0.6V)	69fJ/bit (0.6V)

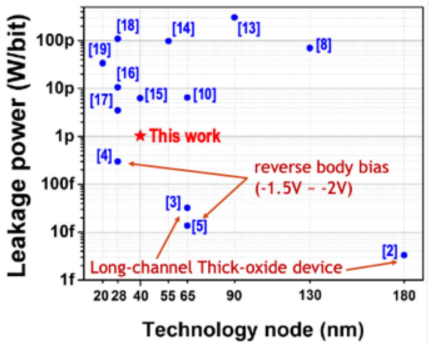

Fig. 14. Comparison table and design space landscape.

during sleep. Each bank can have up to 3 top arrays and at least one bottom array in case of an array swapping. To minimize the frequency of swaps, programmers can do some clever domain partition of the memory arrays. For example, instruction memories, exhibiting mostly random accesses, are placed all in the bottom, whereas neural engine memories with mostly sequential access patterns can be primarily placed in the top for balance.

V. Measurement Results

Over 200 voltage-stacked SRAM banks, a total of 8.9 Mb, were implemented in a 40-nm CMOS image processing IoT chip which operates at 438 kHz at 0.7 V. Fig. 8 shows the die photograph and shmoo plot measured with 75% SRAM arrays in top domain, which leads to the highest mid-rail voltage, the fastest access speed and lowest leakage (Fig. 9). Fig. 11 shows the measured leakage across voltage and temperature, with 1.03 pW/b reported at 0.58 V. Compared to conventional SRAM in the same technology, it achieves over 100× leakage reduction from mainly three sources: 1) the use of high Vt transistor reduces leakage by 7×; 2) supply voltage scaling and bit-cell sizing provide another 11× reduction; and 3) due to voltage stacking, the body effect (negative Vsb) and bit-line leakage reduction in top arrays result in an additional 2× reduction (Fig. 10). Power-rail voltage stability is crucial to this SRAM in that large voltage disturbance in mid-rail may cause data retention failure. Fig. 12 shows that VDDmid varies by ±16 mV across 100 °C. Fig. 13 shows that VDDmid drops only 1.74 mV at 0.65 V when one bank is swapped every 11 cycles at 330-kHz clock. And there is no need to worry that array swapping would stress power distribution network with large instantaneous voltage drop (IVD) caused by huge current spike. First, mid-rail power grid of all 200 SRAM banks are connected together to create a huge amount of decoupling cap. Second, each swap consumes only 8-pJ active energy, similar to a 128-bit read operation. Fig. 14 compares this letter to other state-of-the-art low leakage SRAMs whose leakage current ranges from few fW/b to hundreds of pW/b. Many have leakage power above 1pW/b and require extra supply levels for data retention. And those with leakage lower than pW/b, either used large long-channel thick-oxide device or SOI devices with strong reverse body bias voltage applied.

References

[1] S. Han *et al.*, "EIE: Efficient inference engine on compressed deep neural network," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2016, pp. 243–254.

[2] G. Chen *et al.*, "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2010, pp. 288–289.

[3] T. Fukuda *et al.*, "13.4 A 7ns-access-time 25μW/MHz 128kb SRAM for low-power fast wake-up MCU in 65nm CMOS with 27fA/b retention current," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2014, pp. 236–237.

[4] R. Ranica *et al.*, "FDSOI process/design full solutions for ultra low leakage, high speed and low voltage SRAMs," in *IEEE Symp. VLSI Circuits Dig.*, Jun. 2013, pp. 210–211.

[5] M. Yabuuchi *et al.*, "A 65 nm 1.0 V 1.84 ns Silicon-on-Thin-Box (SOTB) embedded SRAM with 13.72 nW/Mbit standby power for smart IoT," in *IEEE Symp. VLSI Circuits Dig.*, Jun. 2017, pp. 220–221.

[6] Y. Wang *et al.*, "A 4.0 GHz 291Mb voltage-scalable SRAM design in 32nm high-κ metal-gate CMOS with integrated power management," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2009, pp. 456–457.

[7] J. Chang *et al.*, "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel Xeon processor 7100 series," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 846–852, Apr. 2007.

[8] T.-H. Kim, J. Liu, and C. H. Kim, "A voltage scalable 0.26 V, 64 kb 8T SRAM with V_{min} lowering techniques and deep sleep mode," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, pp. 1785–1795, Jun. 2009.

[9] F. Hamzaoglu *et al.*, "A 153Mb-SRAM design with dynamic stability enhancement and leakage reduction in 45nm high-κ metal-gate CMOS technology," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2008, pp. 376–377.

[10] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 141–149, Jan. 2008.

[11] K. Blutman *et al.*, "A low-power microcontroller in a 40-nm CMOS using charge recycling," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 950–960, Apr. 2017.

[12] J. M. Wilson *et al.*, "8.6 A 6.5-to-23.3fJ/b/mm balanced charge-recycling bus in 16nm FinFET CMOS at 1.7-to-2.6Gb/s/wire with clock forwarding and low-crosstalk contraflow wiring," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2016, pp. 156–157.

[13] J.-J. Wu *et al.*, "A large σV_{TH}/VDD tolerant zigzag 8T SRAM with area-efficient decoupled differential sensing and fast write-back scheme," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 815–827, Apr. 2011.

[14] Q. Dong *et al.*, "A 0.3V VDDmin 4+2T SRAM for searching and in-memory computing using 55nm DDC technology," in *IEEE Symp. VLSI Circuits Dig.*, Jun. 2017, pp. 160–161.

[15] 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 J. Solid-State Circuits*, vol. 46, no. 11, pp. 2713–2726, Nov. 2011.

[16] N. Maeda *et al.*, "A 0.41 μA standby leakage 32 kb embedded SRAM with low-voltage resume-standby utilizing all digital current comparator in 28 nm HKMG CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 917–923, Apr. 2013.

[17] F. Tachibana *et al.*, "A 27% active and 85% standby power reduction in dual-power-supply SRAM using BL power calculator and digitally controllable retention circuit," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Feb. 2013, pp. 320–321.

[18] F. Frustaci, M. Khayatzadeh, D. T. Blaauw, D. Sylvester, and M. Alioto, "13.8 A 32kb SRAM for error-free and error-tolerant applications with dynamic energy-quality management in 28nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2017, pp. 244–245.

[19] H. Fujiwara *et al.*, "A 20nm 0.6V 2.1μW/MHz 128kb SRAM with no half select issue by interleave wordline and hierarchical bitline scheme," in *IEEE Symp. VLSI Circuits Dig.*, Jun. 2013, pp. 118–119.