Enhancing DRAM Self-Refresh for Idle Power Reduction

Byoungchan Oh, Nilmini Abeyratne, Jeongseob Ahn
Ronald G. Dreslinski and Trevor Mudge
University of Michigan, Ann Arobr, MI 48109
{bcoh, sabeyrat, ahnjeong, rdreslin, tnm}@umich.edu

ABSTRACT

DRAM can enter self-refresh mode to save power during idle periods. But self-refresh mode does not modify or reduce the number of refresh operations, therefore the refresh energy stays the same. We observe that in the self-refresh mode DRAM cells are in two distinct modes, static (idle) and dynamic (refreshing), and that the switching between these modes are predictable. In this paper, we propose two new self-refresh modes to improve the power efficiency of DRAM: Enhanced Self-Refresh (ESR) and Long latency Self-Refresh (LSR). The key idea behind our observation is to optimize the leakage current of DRAM cells by selectively applying different voltage levels to the DRAM cell transistors when they are active (accessed for refreshing) and idle (pre-charged) by adjusting both the word-line and body voltages.

With our techniques, the retention time of DRAM cells is improved. In our SPICE and mathematical models, ESR and LSR modes result in a 39% and 48% DRAM self-refresh power reduction compared to the existing self-refresh mode, respectively. A workload analysis of ESR shows DRAM energy savings on average of 22%. In addition, for the long idle periods in server systems, the LSR mode can reduce DRAM idle power by nearly 50%, which results in a 6.5% total system idle power reduction.

CCS Concepts

•Hardware → Dynamic memory;

Keywords

DRAM; Self-refresh; Idle power

1. INTRODUCTION

In modern computer systems, Dynamic Random Access Memory (DRAM) is one of the components consuming significant amounts of power. To reduce DRAM idle power, the self-refresh mode, where the memory bus clock and unused circuitry are disabled, has been proposed and adopted in modern DRAM [12]. However, nontrivial power consumption still remains because of internal refresh operations.

There have been many studies to address this issue from both hardware and software perspectives. Prior work has exploited variability of retention time in DRAM cells [10, 1], dependency of leakage current on temperature [22], correctable error probabilities via strong ECC in extended refresh period [5], and bank partitioning for critical and non-critical data [15] into used and unused banks [19]. However, these prior works require large area overhead, sophisticated circuit techniques, or strong support from the OS and memory controller. In this paper, our goal is to increase the refresh period and consequently reduce refresh power without a large area overhead or modifications to the memory controller.

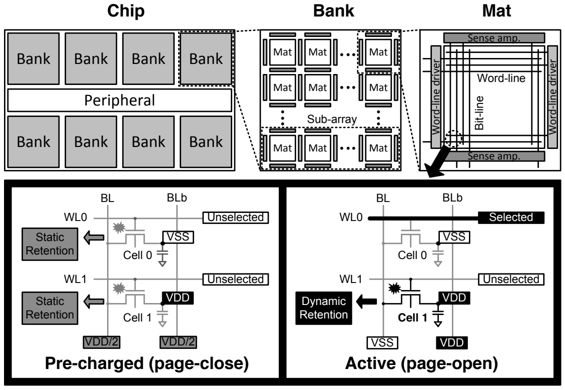


Figure 1: Configuration of a DRAM chip and static(left) and dynamic(right) retention states.

Our observation, shown in Fig. 1, is that DRAM cells have different leakage currents in the static and dynamic states. On the bottom left is a static state; the rows are pre-charged (page-close), the bit-line is at $V_{DD}/2$ and each cell is either at V_{SS} (storing bit 0) or V_{DD} (storing bit 1). The voltage differential between the bit-line and the cell is $V_{DD}/2$. On the bottom right is a dynamic state; the bit-line is driven to either V_{SS} or V_{DD} depending on the cell's value (page-open). In this situation, for all the non-active rows in the same sub-array it increases the voltage differential between the bit-line to the cell, i.e if a bit-line is at V_{SS} and a cell is at V_{DD}, the voltage differential is V_{DD}. Because of Drain-Induced-Barrier-Lowering (DIBL) [21], the large voltage differential between the bit-line and the cell generates large cell leakage current. Moreover, current systems use the same word-line and body bias for both the static and dynamic states. The current voltage levels are optimized for the dynamic state because varying traffic patterns during normal operation makes it difficult to predict the cell state. The worst case scenario where a row is always open in a refresh cycle must be considered when setting these voltages.

On the other hand, we observe that in self-refresh mode the internal refresh operations are periodic and predictable. Thus, the worst case scenario is simplified and allows for optimization for each state independently. Our key idea is to *match the word-line and body voltage levels to the state of the cell.* With our observations, we propose two new self-refresh modes. The first mode is ESR mode where Selective Word-line Bias (SWB) is applied. The ESR mode can directly replace the currently existing self-refresh mode without modification of access protocols and timing parameters. The second mode is LSR mode where Selective Body Bias (SBB) is adopted combined with SWB, bringing larger

ISLPED '16, August 08-10, 2016, San Francisco Airport, CA, USA

DOI: http://dx.doi.org/10.1145/2934583.2934632

power reduction in idle state with tolerable latency for exiting the mode.

In order to verify and confirm our idea, extensive experiments from transistor-level analysis to system-level simulation are conducted. Our SPICE and mathematical models show that retention time of DRAM cells are improved by 2.42× in ESR mode and by 3.58× in LSR mode. These improvements bring a DRAM self-refresh power reduction of 39% and 48%, respectively. With ESR mode, 22% of average energy savings in DRAM results on simulation of Spec2006 and MediabenchII workloads. Moreover, LSR mode reduces total server idle power by 6.5%.

2. MOTIVATION

2.1 DRAM Cell Leakage Current

DRAM cells have mainly five leakage components [21]. Sub-threshold leakage current (a) of the cell transistor has been increasing as technology scaling brings reduction of threshold voltage to keep operating speeds high at scaled supply voltages. Gate induced drain leakage (GIDL) current (b) is generated in the overlapped region in the gate and drain (storage node) due to the higher electric field. Thinner gate oxide and high voltage differences between the gate and storage node increases GIDL current. Junction leakage (c) at the reverse biased p-n junction is mainly affected by applied voltage, area of the junction, and doping concentration of both the p-n regions. Gate tunneling leakage (d) increases as the gate oxide thickness is decreased to maintain sufficient gate control over the channel. Lastly, there is the dielectric leakage of the storage capacitor (e), which has been reported to be not significant.

2.2 Static and Dynamic Retention

In Fig. 1, the internal structure of a DRAM chip and both states are described. When a row is closed the bit-line is accordingly pre-charged to the half V_{DD} level, this state is called the static retention state. The voltage difference of all cells remain $V_{DD}/2$ in the static states regardless of the stored data. In the dynamic retention state, a row is opened and the voltage of the bit-line becomes either V_{DD} or V_{SS} depending on the selected cell's data. If a cell (*Cell_1* in Fig. 1) in the selected sub-array has a different voltage level (V_{DD}) from the voltage level (V_{SS}) of the selected cell (*Cell_0*), the voltage difference of *Cell_1* between the bit-line and the storage node becomes V_{DD}. Previous studies have found that the differential of bit-line voltage to storage node voltage generates different amounts of leakage currents [4].

2.3 Techniques to Reduce the Leakage

Reverse Body Bias. Applying reverse body bias voltage (negative V_{BB}) can effectively suppress sub-threshold leakage current by increasing the threshold voltage of the cell transistors as proposed[11]. However, one problem is that large amounts of junction leakage can be generated by excessive negative V_{BB}. In addition, increasing threshold voltage brings substantial degradation of the cell transistor's on-current. Since the amount of the on-current directly affects operation speed, reducing sub-threshold leakage current by applying negative V_{BB} has limited impact.

Negative Word-Line Bias. Applying a negative word-line voltage (NWL) to unselected DRAM cell [23] effectively reduces sub-threshold leakage current without degrading the cell transistor's on-current, because transistors can be strongly turned off without increasing the threshold voltage. However, the drawback is that excessive NWL voltage generates large amounts of GIDL current.

Combined Technique. In most modern DRAM devices, combined negative V_{BB} and NWL techniques are applied [14]. In this paper we also exploit the combined technique to implement our idea. However, different from prior work, we dynamically adjust the voltages to achieve further gains.

2.4 Self Refresh Mode

DRAM has three power levels for the idle state which are standby, power-down, and self-refresh. Each mode has different power consumption, wake-up latency, and purposes. During standby mode, DRAM waits for the next request without any power saving techniques. Therefore, this mode has the largest standby current. The power-down mode disables the interface circuitry, but there is a short wake-up latency (t_{XP}) because enabling interface circuitry and re-synchronizing memory bus clocks takes several cycles. Thus, this mode is beneficial when the length of idle time is moderate. The most power-efficient mode is self-refresh mode. In the self-refresh mode, the interface circuitry and delay locked loop (DLL) circuit are disabled. In addition, the memory controller is disconnected and refresh operation is done autonomously by the internal counter. Although the most power efficient, this mode has several hundreds of cycles of exit latency because of the DLL's re-locking time. (or for the fast-exit, one refresh cycle should be guaranteed.) Therefore, self-refresh mode is only suitable for long idle time.

Prior work reports that timely switching to self-refresh mode significantly reduces DRAM energy in many workloads with small performance degradation by the exit latency [24]. In addition, Meisner et al. report that average system idle interval between two adjacent requests is more than 100ms in over 600 servers [16] and DRAM enters the self-refresh mode during that time to reduce the idle power.

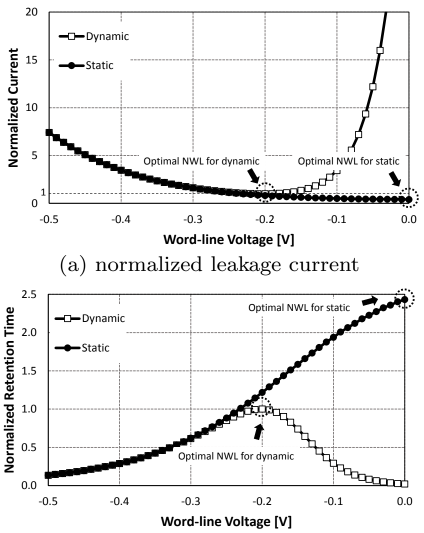
3. PROPOSED TECHNIQUE

The optimum NWL and negative V_{BB} levels for the static state are different from the optimum levels for the dynamic state. For example, a specific NWL and negative V_{BB} that minimizes the leakage current in the dynamic state cannot minimize static leakage because of the different applied voltage between the bit-line and storage nodes. Since the voltage difference between the bit-line and storage nodes is higher in the dynamic state (V_{DD}) than in the static state ($V_{DD}/2$), the dynamic leakage current is larger than the static leakage current given the same NWL and negative V_{BB} level. Conversely, DRAM cells in the static retention state can have lower leakage current than those in the dynamic retention state by applying different NWL and negative V_{BB} from the dynamic retention state.

We show that if two different optimum NWL and negative V_{BB} are selectively applied to match each retention state, the leakage current of DRAM cells can be reduced over applying fixed NWL and negative V_{BB}. Thus, this implies that DRAM cell retention time can be improved with selective biasing. However, in order to selectively apply two different voltage levels, each retention state should be clearly defined. In other words, we should know when a retention state is switched to another state and how long that state lasts. This is impossible to predict during normal operation because the retention state of DRAM cells can vary according to DRAM access patterns. Therefore, the worst case condition where DRAM cells are always in the dynamic retention is assumed when designing DRAM [4, 14]. But, we identify the opportunity to apply selective voltages in the self-refresh mode when access patterns are much more predictable.

3.1 Leakage Analysis

In order to find optimal voltage levels for each state that reduces leakage and improves retention, SPICE simulation is performed based on the parameters shown in [6].



(a) normalized leakage current

(b) normalized retention time

Figure 2: Analysis of leakage current and retention time of a DRAM cell according to word-line voltage.

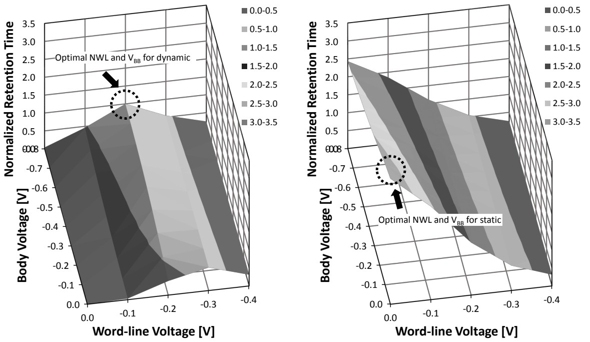
Word-Line Voltage Dependency. Figure 2a shows the dependency of the total leakage current of DRAM cells on negative word-line bias voltage in both dynamic and static retention states. In this simulation, the body is connected to V_{BB}=-0.8V, which is normally used in modern DRAM devices [14]. The optimal voltage for the dynamic state is -0.2V. A more negative voltage increases GIDL current, while a more positive voltage causes the sub-threshold leakage current to become dominant due to high voltage difference (V_{DD}) between the storage node and the bit-line. On the other hand, the sub-threshold leakage in the static state becomes dominant at a higher word-line voltage (around 0.05V, not shown in the Fig. 2a) than in the dynamic state, because of the small voltage difference ($V_{DD}/2$) between the storage node and the bit-line. Similar to the dynamic state, a more negative voltage increases GIDL current in the static state. Therefore, selectively applying -0.2V for the dynamic state and 0V for the static state can reduce the total leakage current.

Figure 2b shows the normalized retention time for the each state. If the DRAM cell is always in the static state with 0V for the word-line voltage, the retention time is improved 2.43× compared to that in the dynamic state with -0.2V of the word-line voltage.

Body Voltage Dependency. Figure 3 shows normalized retention time according to the word-line and body bias voltages. In Fig. 3a, the maximum retention time for the dynamic retention state appears around -0.2V of the word-line and -0.8V of the body bias. However, applying ground level bias to both word-line and body results in the maximum retention time for the static retention state as shown in Fig. 3b. Therefore, changing both biasing voltage levels to ground level can bring a 3.57× extension of the retention time in the static state.

3.2 Mathematical Approach

Cell Operations during Self-Refresh Mode. In the self-refresh mode, DRAM is disconnected from the memory controller and there are no random memory accesses. The only activity is a periodic refresh operation scheduled autonomously by an internal counter. As a result, the switching timing between states and the duration of each retention state is predictable.



(a) dynamic state

(b) static state

Figure 3: Normalized retention time with various word-line and body voltages.

Mathematical Model for Extended Retention Time. DRAM cells are switched from the static state to the dynamic state when other cells in the same sub-array are refreshed, and their data are different from the accessed cell on their bit-line(see Fig. 1). Since only one row can be opened simultaneously, the total time spent in the dynamic retention state in the self-refresh mode is defined as follows:

$$t_{dyn} = (n-1) \cdot t_{RC} \tag{1}$$

where t_{dyn}, n, and t_{RC} are total time of the dynamic state, the number of rows in the same sub-array, and the row cycle time, respectively. Typically $n = 512$ in modern DRAM devices [13]. In Eq. 1, it is assumed that regardless of data polarity ('0' or '1') a DRAM cell always goes to the dynamic state when other cells in the same sub-array are refreshed, which is the worst case scenario.

By selectively applying different bias voltages to both the static and dynamic modes, the reduced total leakage current can be expressed as

$$I'_{leak\_tot} = \frac{t_{dyn}}{t'_{ret}} \cdot I_{leak\_dyn} + \frac{t'_{ret} - t_{dyn}}{t'_{ret}} \cdot I_{leak\_sta} \tag{2}$$

where I'_{leak_tot} is the reduced total leakage current and t'_{ret}, I_{leak_dyn}, and I_{leak_sta} are the extended data retention time, total leakage current in the dynamic and static states, respectively. Equation 2 means that I_{leak_dyn} leaks from a DRAM cell during t_{dyn} and I_{leak_sta} leaks during the remaining time in one refresh period. Equation 2 can be converted to following equation for new retention time, t'_{ret}:

$$t'_{ret} = t_{ret\_sta} + t_{dyn} - \frac{t_{ret\_sta}}{t_{ret\_dyn}} \cdot t_{dyn} \tag{3}$$

where t_{ret_sta} and t_{ret_dyn} are data retention time in the dynamic and static states, respectively.

Mathematical Model for Reduced Power. In order to calculate current consumption in the self-refresh mode with the extended retention time, the average refresh current in the self-refresh mode ($I_{avg\_ref\_sr}$) should be defined. To obtain $I_{avg\_ref\_sr}$, we use I_{DD6} and I_{DD6ET}, average current consumption during the self-refresh mode at normal and extended temperature, respectively. Since the refresh rate is doubled at extended temperature, the difference between I_{DD6ET} and I_{DD6} represents the average current consumption for one internally issued refresh operation. With the improved retention time, the reduced current consumption during the self-refresh mode is expressed as

$$I'_{DD6} = I_{background\_sr} + \frac{t_{ret}}{t'_{ret}} \cdot I_{avg\_ref\_sr} \tag{4}$$

where $I_{avg\_ref\_sr}$, $I_{background\_sr}$, and I'_{DD6} are the average refresh current, the background leakage current, and the reduced current consumption during the self-refresh mode. Our technique is to extend the retention time, but not to reduce the amount of refresh current itself during the self-

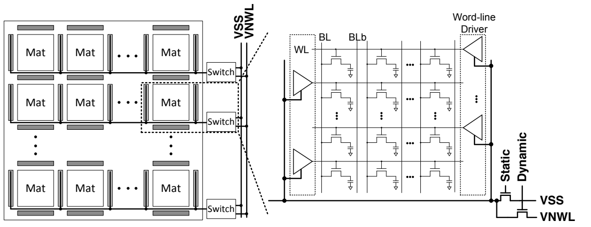


Figure 4: Circuit schematic showing the voltage selection for SWB technique.

refresh mode. However, increasing the retention time has the same effect as reducing the average refresh current during the self-refresh mode.

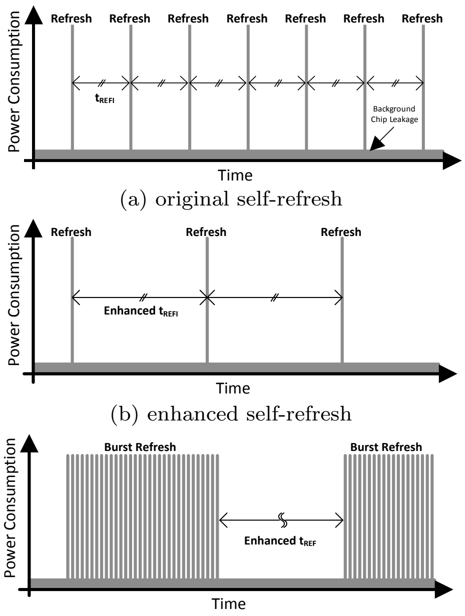
3.3 Implementation

SWB and ESR. For the selective word-line voltage, we use two voltage levels which are -0.2V and 0V. Since the negative word-line scheme is already commonly used in most modern DRAM devices, we do not have to introduce additional circuitry. Similarly, the ground level also already exists in the DRAM. In our SPICE simulation, the optimum voltage level, where minimum static leakage current appears, is not the ground level, but around 0.05 V. However, generating a new voltage level requires voltage generator circuitry and a considerable number of metal lines for the voltage routing. In order to minimize area overhead, we utilize the two existing voltage levels sacrificing a small amount of the potential energy savings for a more area efficient design.

Figure 4 shows the circuit schematic of the SWB technique. Since a refresh operation changes the retention state of cells in the same sub-array, independent voltage control is necessary for each sub-array. For example, if a row is refreshed in a sub-array, the cells connected to all other rows in this sub-array enter the dynamic state, whereas the cells in other sub-arrays remain in the static state. In order to separately control the voltage level of individual sub-arrays, at least one switch is required per sub-array. In addition, selection signals of the switch, which are static and dynamic, should be separated for each sub-array. For instance, if a row is refreshed in a sub-array, the voltage level of the rows in that sub-array should be -0.2V with *dynamic=high* and *static=low*, whereas the voltage level of the rows in other sub-arrays should be 0V with *static=high* and *dynamic=low*. These *static* and *dynamic* signals can be easily generated because each sub-array has its own row addresses.

Having only one switch in one sub-array may cause non-negligible switching latency, increasing row access time. Because many word-line drivers share one voltage line (the bold line in Fig. 4), this line has a large capacitance and leads to slow voltage transitions. The naive solution is to add more switches, because increasing the number of switches can shorten the length of the voltage line and reduce the number of word-line drivers sharing it. However, this results in a large area overhead. Instead, our solution is to predict the timing of the switch selection signals, *static* and *dynamic*. During the self-refresh mode, the timing and address for the next refresh operation can be predicted. We issue the selection signals to correspond to the switching and stabilization latency; thus eliminating switching latency.

SBB and LSR. Unlike SWB, SBB, which selectivly applies -0.8V and 0V to the body of the cell tranistors, brings long switching latencies. V_{BB} transitioning incurs long latency due to two reasons. First, shifting the output voltage level of the V_{BB} generator, which supplies V_{BB} to the cell transistors, takes hundreds of nanoseconds. Second, because the body of all cell transistors in a bank are connected to the same V_{BB} line, it has to drive a large load capacitance [18]. Then, the V_{BB} voltage level cannot be switched with the comparable speed of row accesses (nano-seconds scale).



(a) original self-refresh

(b) enhanced self-refresh

(c) long latency self-refresh

Figure 5: Power consumption and cell operations in original, enhanced, and long latency self-refresh modes.

In order to overcome the slow transitions of V_{BB} we propose a new LSR mode. In order to maximize power-savings, SWB is applied as well as SBB in the LSR mode. In LSR mode, refresh operation is changed from distributed-refresh to burst-refresh of all rows with one refresh command as shown in Fig. 5. With increasing DRAM capacity, burst-refresh has been abolished because of its long refresh cycle time. However, modern DRAM devices emply a distributed-burst-refresh where multiple rows in a bank are refreshed by one refresh command (a burst) that are then distributed across the refresh intervals. Therefore, our burst refresh can be enabled with minor modifications by changing the number of rows being refreshed to all rows.

Since the burst refresh for all rows only requires two V_{BB} transitions when the burst refresh starts and finishes, time overhead of the V_{BB} transition can be minimized. When exiting the self-refresh mode, the memory controller is reconnected to DRAM and it assumes control of the refresh operations. Since the memory controller is unaware of the last internal refresh operation, it immediately performs a refresh. The time for this refresh should be counted in the exit latency from the LSR mode, which can be expressed as

$$t_{XLSR} = m \cdot t_{RFC} + 2 \cdot t_{V_{BB\_trans}} \quad (5)$$

where t_{XLSR} is the minimum time from exit of LSR to a next valid command and $t_{V_{BB\_trans}}$ is the time to be taken for the transition of V_{BB} levels. Typical DRAM at normal temperature range issues 8192 refresh commands to refresh all rows and thus m becomes 8192. We assume $t_{V_{BB\_trans}}$ is 50μs based on previous papers [18]. In 8Gb-DDR4-2400 where t_{RFC} is 350ns, t_{XLSR} thus becomes 2.97ms. Although the exit latency of LSR mode is much longer than that of the original self-refresh mode, LSR can have substantial power savings without large performance penalty. In [16], the average idle interval between two adjacent requests in the web server systems was found to be over 100ms. Our t_{XLSR} is an acceptable latency as the transition of DRAM power mode to obtain enough power-savings.

3.4 Overhead

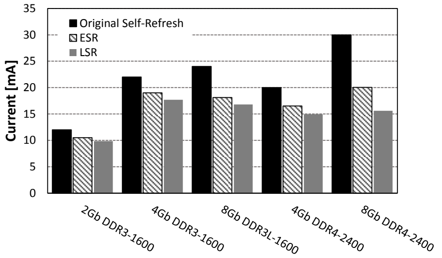


Figure 6: Comparison of current consumption between ESR, LSR, and original self-refresh modes calculated with Eq. 4.

We confirmed that ESR mode can directly replace original self-refresh mode without modification of the memory controller. In addition, SWB technique, applied in ESR mode, only requires two transistors per a sub-array. The area overhead by adding switches is negligible considering the scale of a sub-array, which consists of 1Mega cells and their wordline drivers. Moreover, because row selection signals can be reused for the switch selection signals, the area overhead to enable SWB can be ignored.

LSR mode is a new power mode having an even lower power level but with different exit latency. Therefore, modification of the memory controller is inevitable to introduce the LSR mode. However, only a slight modification is necessary because the only difference from the original self-refresh mode is the exit latency. There is negligible modification in DRAM to enable SBB technique, applied in LSR mode, because power gating technique, which drives V_{BB} level to the ground level, is applied in modern DRAM and can be reused for SBB.

4. EVALUATION METHODOLOGY

In order to evaluate our proposed techniques, we use two different approaches. First, we evaluate the impact of the ESR mode on DRAM energy by running various workloads. Second, we analyze total system idle power. Since the LSR mode is proposed for long idle times, we do not evaluate the impact of the LSR mode while applications are running.

4.1 Workload Analysis

MARSSx86 [20], a full-system x86 simulator, and DRAMPower [3], a tool for DRAM power and energy estimation, are used for the evaluation of the ESR mode. Table 1 shows the system configuration in our simulation. For speculative usage of power-down and self-refresh modes, we modified the command scheduler based on prior study [24].

Table 1: System configuration for the workload simulation.

Component	Specifications
Processor	2 GHz, single out-of-order core, 4-issue
L1 Cache	128 KB, 8-way associativity
L2 Cache	2 MB, 8-way associativity
DRAM Device	Micron- 8 Gb, DDR4-2400, x16 I/O

We employ the workloads in two benchmark suites, which are Spec2006 [8] and MediabenchII [7]. In order to confirm the merit of our technique, 18 workloads, where memory accesses are non-intensive, are selected from the two suites.

4.2 Idle Power Analysis

We use HP Power Advisor, which is a tool for estimating power requirements for HP ProLiant server systems, to analyze total system idle power [9]. Self-refresh current is used to present DRAM idle power in this tool. We substitute the self-refresh current with the current drawn in LSR mode to obtain an idle power savings. The configured system is shown in Table 2.

Table 2: Specifications of the configured system for idle power analysis

Component	Specifications
Processor	2.4 GHz, eight cores x2
Memory	32 GB, DDR4 x8
Storage	800 GB, SSD x1
Network	1 Gbps, 2 ports x1
Power Supply	550 W, 80 PLUS x1

5. RESULTS

5.1 SPICE/Mathematical Model

From Eq. 4, current consumption of ESR and LSR modes can be obtained. The improved retention time that Eq. 4 requires is obtained from SPICE simulation (see Sec. 3.1). For the remaining parameters that are shown in Eq. 4 Micron's data sheets [17] are used. Figure 6 shows the reduced current consumption of the ESR and LSR modes from original self-refresh mode for various DRAM devices. ESR and LSR mode reduces the current by an average of 20.2% and 28.2% from the original self-refresh mode, respectively.

5.2 DRAM Energy Saving in Workloads

In our system, DRAM enters the ESR mode when there is a long idle period. Our baseline it the original self-refresh mode. As discussed in Sec. 3.3, ESR mode does not bring any changes of timing parameters related to the self-refresh mode. In addition, we use the same threshold value to determine the DRAM power mode. Therefore, DRAM devices remain in ESR mode for the same number of cycles as in the original mode. The execution time of the workloads are the same. The only difference is that the ESR mode has less current consumption, which results in large DRAM energy savings during long idle periods.

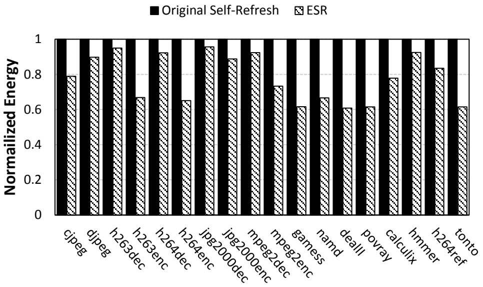


Figure 7: Effect of ESR mode on DRAM energy saving with workloads in SPEC 2006 and MediaBench II.

Figure 7 shows ESR energy normalized to original self-refresh energy for various MediaBench and Spec workloads. ESR mode reduces DRAM energy by up to 39.1% (*dealII*) and on average 22.0% without performance degradation. The maximum energy savings possible in our system with 8Gb DDR4 is 39.2% (Fig. 6). Thus, it is obvious that while running *dealII*, DRAM devices are mostly in the idle state. In the ESR mode less memory intensive workloads have more energy savings because there are more chances to enter the self-refresh mode.

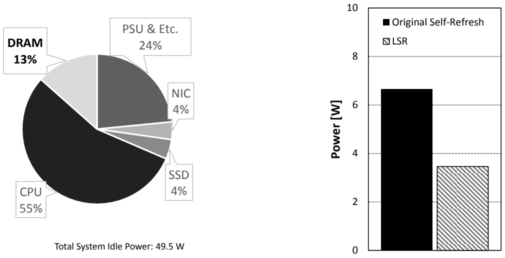
5.3 System Idle Power Reduction

The LSR mode is enabled for very long idle periods when no workloads are running. The system shown in Table 2 consumes 274.9W at the maximum load, of which 35.9% is DRAM, and 49.5W for idle state, of which 13% is DRAM (Fig. 8a). The portion of DRAM idle power in this system nearly corresponds with that shown in [2]. Thus, the configured system falls into typical server systems.

If the LSR mode is used for the system idle state instead of the original self-refresh mode, the idle power of DRAM

Table 3: Comparison between various self-refresh technique.

	ASR [1]	DPS-refresh [10]	MECC [5]	Flicker [15]	ESR	LSR
Testing required	Yes (BIST)	Yes	No	No	No	No
Area overhead in DRAM	1.5 %	1 %	0 %	0 %	0 %	0 %
Controller modification	No	No	Yes (major)	No	No	Yes (minor)
Software (or OS) support	No	No	No	Yes	No	No
Mechanism	Apply variable refresh rate through retention time profiling		Allow or correct errors with lowered refresh rate		Apply lowered refresh rate with improved leakage current	



(a) total system idle power (b) DRAM idle power reduction

Figure 8: Breakdown of the system idle power and reduction of DRAM idle power with LSR mode.

is reduced from 6.7W to 3.5W as shown in Fig. 8b. Since DRAM consumes a considerable amount of the system idle power, this reduction can bring a 6.5% reduction in total system idle power.

6. RELATED WORK

In addition to the techniques we propose, there have been several studies to reduce self-refresh. First, Idei et al. [10] proposed a power efficient self-refresh mode having dual-period refreshes. They applied different refresh rate to each row with the retention information stored in internal non-volatile storage. Second, MECC incresed refresh intervals and used ECC to correct resulting errors [5]. Last, Temperature Compensated Self-Refresh (TCSR) and Partial Array Self-Refresh (PASR) have been proposed and implemented in modern DRAM devices [19]. During TCSR mode, internal self-refresh intervals are adjusted for the ambient temperature of DRAM devices. PASR allows portions of the DRAM (bank-granularity) to be put into a no-refresh state while other portions are in normal self-refresh.

Unlike prior work, our technique does not require large area overhead in the DRAM devices, sophisticated circuit techniques, or strong support from the OS and memory controller. We summarize the difference between our proposed technique and prior work in Table 3.

7. CONCLUSION

We presented Selective Word-line and Selective Body Bias (SWB and SBB), two novel techniques to improve DRAM cell retention time in self-refresh mode. Our observations are that: 1) the retention time of DRAM cells can be improved by selectively applying different voltage levels to the word-line and body bias depending on their states: either active or pre-charged; 2) the state of the cell is periodically and predictably switched during self-refresh mode. SWB and SBB exploit the periodic and predictable cell operations of the self-refresh mode to design for a less severe worst-case scenario. With SWB and SBB, we proposed new power-efficient self-refresh modes—Enhanced and Long latency Self-Refresh (ESR and LSR). To our knowledge, the presented work is the first work to reduce DRAM idle power by exploiting variability in the leakage current of DRAM cells depending on the DRAM internal behavior.

8. REFERENCES

[1] J.-H. Ahn et al. Adaptive self refresh scheme for battery operated high-density mobile DRAM applications. In *ASSCC 2006*.

[2] L. A. Barroso et al. The datacenter as a computer: An introduction to the design of warehouse-scale machines. 2013.

[3] K. Chandrasekar et al. Improved Power Modeling of DDR SDRAMs. In *DSD*.

[4] M. Chang et al. Impact of gate-induced drain leakage on retention time distribution of 256 Mbit DRAM with negative wordline bias. *Electron Devices, IEEE Transactions on*, 2003.

[5] C. Chou et al. Reducing Refresh Power in Mobile Devices with Morphable ECC. In *DSN 2015*.

[6] DRAM Power Model. http://www.rambus.com/energy.

[7] J. E. Fritts et al. MediaBench II video: Expediting the next generation of video systems research. *Microprocessors and Microsystems*, 2009.

[8] J. L. Henning. SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 2006.

[9] HP Power Advisor. http://www.hp.com/go/hppoweradvisor.

[10] Y. Idei et al. Dual-period self-refresh scheme for low-power DRAM's with on-chip PROM mode register. *Solid-State Circuits, IEEE Journal of*, 1998.

[11] K. Itoh et al. Trends in low-power RAM circuit technologies. *Proceedings of the IEEE*, 1995.

[12] JEDEC standard for DDR/DDR2/DDR3/DDR3L/DDR4 SDRAM. http://www.jedec.org/standards-documents.

[13] Y. Kim et al. A case for exploiting subarray-level parallelism (SALP) in DRAM. In *ISCA 2012*.

[14] D.-S. Lee et al. Simultaneous Reverse Body and Negative Word-Line Biasing Control Scheme for Leakage Reduction of DRAM. *Solid-State Circuits, IEEE Journal of*, 2011.

[15] S. Liu et al. Flikker: saving DRAM refresh-power through critical data partitioning.

[16] D. Meisner et al. PowerNap: eliminating server idle power. In *ASPLOS 2009*.

[17] Micron Technology. https://www.micron.com/products/datasheets.

[18] K.-S. Min et al. A fast pump-down V BB generator for sub-1.5-V DRAMs. *Solid-State Circuits, IEEE Journal of*, 2001.

[19] Partial Array Self Refresh (PASR) TN - Micron. https://www.micron.com/support.

[20] A. Patel et al. MARSS: a full system simulator for multicore x86 CPUs. In *DAC 2011*.

[21] K. Roy et al. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proceedings of the IEEE*, 2003.

[22] D. Shim et al. A process-variation-tolerant on-chip CMOS thermometer for auto temperature compensated self-refresh of low-power mobile DRAM. *Solid-State Circuits, IEEE Journal of*, 2013.

[23] H. Tanaka et al. A precise on-chip voltage generator for a gigascale DRAM with a negative word-line scheme. *Solid-State Circuits, IEEE Journal of*, 1999.

[24] G. Thomas et al. A predictor-based power-saving policy for dram memories. In *DSD 2012*.