

Half-DRAM: a High-bandwidth and Low-power DRAM Architecture from the Rethinking of Fine-grained Activation

Tao Zhang^{†‡} Ke Chen[§] Cong Xu[†] Guangyu Sun[§] Tao Wang[§] Yuan Xie[†]

[†]Pennsylvania State University [‡]NVIDIA Corporation*

[§]Oracle Corporation [§]Peking University

{zhangtao, czx102, yuanxie}@cse.psu.edu

ke.c.chen@oracle.com

{gsun, wangtao}@pku.edu.cn

Abstract

*DRAM memory is a major contributor for the total power consumption in modern computing systems. Consequently, power reduction for DRAM memory is critical to improve system-level power efficiency. Fine-grained DRAM architecture [1, 2] has been proposed to reduce the activation/precharge power. However, those prior work either incurs significant performance degradation or introduces large area overhead. In this paper, we propose a novel memory architecture **Half-DRAM**, in which the DRAM array is reorganized to enable only half of a row being activated. The half-row activation can effectively reduce activation power and meanwhile sustain the full bandwidth one bank can provide. In addition, the half-row activation in Half-DRAM relaxes the power constraint in DRAM, and opens up opportunities for further performance gain. Furthermore, two half-row accesses can be issued in parallel by integrating the sub-array level parallelism to improve the memory level parallelism. The experimental results show that Half-DRAM can achieve both significant performance improvement and power reduction, with negligible design overhead.*

1. Introduction

The power consumption for DRAM memory can compose a significant percentage of the total power consumption in modern computing systems, especially for servers and data center systems. For example, prior work has demonstrated that DRAM can consume significant amount of power (sometimes more than 25% of the total power in a datacenter) [3, 4, 5, 6, 7]. As a result, how to improve the power efficiency of DRAM is one of the major challenges in the memory architecture design. Furthermore, the performance improvement of DRAM for both traditional commodity DRAM [8] and emerging 3D-stacked DRAM [9] is limited due to the power constraints. For example, the well-known timing constraints t_{RRD} and t_{FAW} (t_{TAW} in 3D wide-IO DRAM standard [9]) must be obeyed, which limits the activation frequency and thus diminishes the opportunity of potential performance gain from better memory parallelism.

The *activation and precharge* power is the major contributor to the DRAM power consumption. The activation and precharge power can be around 25% of the total DRAM power (as illustrated in Figure 2. See more details in Section 2.1). Consequently, fine-grained activation techniques [1, 2] have been proposed to reduce activation power¹. These work, however, suffers from either large area overhead or significant performance degradation due to the severely reduced bank bandwidth. As a result, they are impractical to be implemented in modern memory systems unless the fine-grained activation is re-designed to mitigate the overhead on either performance or area. In this work, we propose a novel DRAM architecture called *Half-DRAM* to address the design challenge of fine-grained activation. In Half-DRAM, only half of a row is activated such that the activation power is reduced substantially. Furthermore, as the fine-grained activation relaxes the power constraints, Half-DRAM can further improve the memory performance with sub-array level parallelism [10]. In general, our contributions can be summarized as follows.

- We demonstrate that the prior work on fine-grained activation techniques can result in significant memory performance degradation without careful design. We show in-depth details of the bandwidth degradation of previous work, which directly motivates our work.
- We propose *Half-DRAM* as an effective solution, targeting at low activation power and high memory bandwidth. In Half-DRAM, the *1RD-2HFF* layout is designed to reorganize the cell arrays in order to enable half-bank level activation and thus reduce activation power without sacrificing memory bandwidth. To the best of our knowledge, this is also the first work to enable the fine-grain activation without performance overhead by reorganizing DRAM.
- We further propose a *Half-DRAM-2Row* technique to enhance the performance by relaxing the power constraint in Half-DRAM. This technique integrates sub-array level parallelism (SALP [10]) to exploit the half-bank level parallelism. The experimental results verify that Half-DRAM can achieve better performance with lower power consumption, which makes it promising for future memory systems.

*The work was done when Tao Zhang was in Pennsylvania State University. He is now with NVIDIA as a senior architect.

¹Without specific comment, the term “activation power” represents both activation and precharge power consumption in the rest of paper.

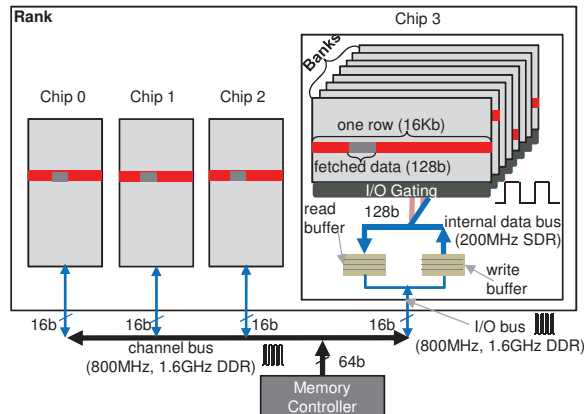


Figure 1: DRAM hierarchy – a 2Gb-8bank \times 16 example

2. Background and Motivation

In this section, we first review the fundamental of conventional DRAM memory architecture to help understand the innovative techniques in Half-DRAM. Then, we show the power inefficiency in the commodity DRAM and the impracticability of prior fine-grained activation techniques.

2.1. DRAM Preliminary

The DRAM Structure. Without loss of generality, commodity DRAM has a top-down hierarchical structure, including channel, rank, chip, and bank as shown in Figure 1. One channel is composed of a memory controller and a few ranks that share the same command/address and data bus. Since the I/O data width of a single memory chip is limited, multiple chips are populated in a rank and operate in lockstep to feed a wider data bus. In this paper, we use Micron DDR3_1600_8bank_x16 [11] DRAM chip as an example and the baseline design, as illustrated in Figure 1. Since each chip can only provide 16-bit (16b) data (so-called $\times 16$ device), four chips are organized in a single rank to satisfy the total bus width of 64-bit (64b). Inside one chip, eight banks are deployed as DRAM cell arrays. A bank can be accessed independently so that bank level parallelism is extensively exploited to assure memory bandwidth. In the figure, each bank has 16,384 rows and 16,384 columns. Even though we focus on the commodity DDR3[8] in this work, the idea of Half-DRAM can be easily extended to other DRAM technologies that have similar hierarchy, such as low-power DDR (LPDDR[12]) or 3D-stacked DRAM (e.g., Wide I/O[9], Hybrid Memory Cube[13]).

The DRAM Power Breakdown. To evaluate the DRAM power consumption, we leverage Micron’s DDR3 power calculator [14] and run STREAM benchmark to collect the related statistics. Figure 2a shows the power breakdown with the simulation results². In the simulation, aggressive fast-exit powerdown mode is adopted so that DRAM is powered down

²Due to space limitation, we do not explain how DRAM power is calculated. Readers can refer to Micron’s technical note [15] for the detail.

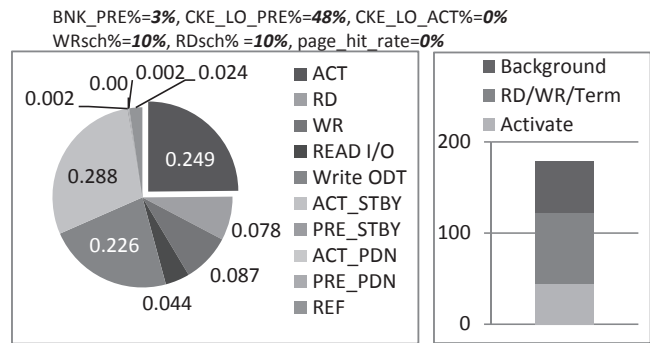


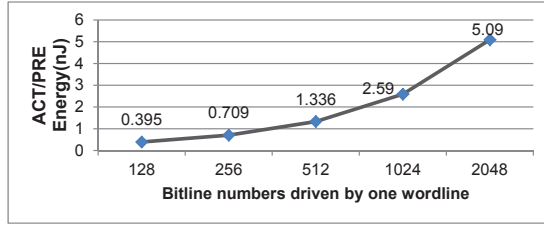
Figure 2: Power breakdown of STREAM

whenever the request queue is empty and the system is idle (The experiment setup is described in Section 5).

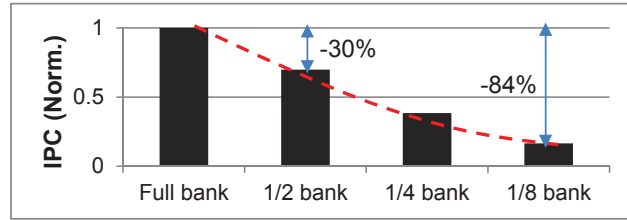
The power breakdown in Figure 2a can be classified into 3 categories in Figure 2b: (1) *Background Power*. The background power takes up 31.6% of total power that accounts for the static power and refresh power, which consists of the powers from ‘ACT_STBY’ (active standby), ‘PRE_STBY’ (precharged standby), ‘ACT_PDN’ (active powerdown), ‘PRE_PDN’ (precharged powerdown) and ‘REF’ (refresh). (2) *RD/WR/Termination Power*.³ About 43.5% of the power comes from RD/WR/Termination which represents the power for the data movement, including the powers from ‘RD’ (read burst), ‘WR’ (write burst), ‘READ I/O’ (read bus driver), and ‘Write ODT’ (write signal on-die termination). (3) *Activation Power*. This consists of 24.9% of the DRAM power. Many prior work has been proposed to reduce the *Background Power* [16, 17, 18, 19, 20] or *RD/WR/Termination Power* [21, 22]. In this paper, our focus is on the minimization of the *Activation Power*.

Row Overfetching and n-bit Prefetching. To reduce the activation power, we first need to understand how the data are fetched in a DRAM memory. In JEDEC-DDR, there are two terms describing the internal data fetching: *Row Overfetching* and *n-bit Prefetching*. As shown in Figure 1, *Row Overfetching* mandates the entire row to be activated (red block) even though only a small portion of data are fetched at a time (gray block). In our baseline model, a 16Kb row is activated while only 128b data are fetched per request. Ideally, row overfetching is supposed to be helpful for performance since it prevents repeated activations if multiple requests access the same row (so-called row buffer hit). It is supposed to also improve power efficiency since the activation power can be amortized over these accesses. Unfortunately, in modern CMP architectures the application interference from other cores randomizes the requests and thus lowers the row buffer hit rate [2, 24]. Moreover, a memory controller should take into account the starvation and fairness issue among requests. Therefore, only a few accesses (<5) can be served for one activation [25]. The

³RD/WR power accounts for the power dissipation on the internal bus during the data transfer, which includes both read/write FIFO access and internal bus traverse. Alternatively, termination power is generated due to the on-die termination and delay-locked loop (DLL) at chip interface.



(a) Energy Proportionality to Bitline Number



(b) The Impact of Reduced Data Bandwidth

Figure 3: Motivation study. (a) The activation/precharge energy proportionality to the number of bitlines based on CACTI-3DD [23]; (b) The Impact of reduced data bandwidth in prior work [1, 2]. These are simulation results of the STREAM benchmark, and are normalized to the speedup of the baseline full bank activation.

Table 1: DRAM Area and Power Breakdown by CACTI-3DD [23]

Area per Chip (mm ²)			
DRAM cell	14.677	sense amplifier	3.189
row predecoder	0.051	column decoder	0.002
local wordline driver			6.789
total area			37.129
Energy per MAT (nJ)			
local bitline	1.336	local wordline	0.005
local SA	0.117	column select line	0.08
row decoder	0.004	column decoder	0.001

low row utilization incurs the power inefficiency because the row overfetching consumes lots of power for activation and precharge.

On the other hand, *n-bit Prefetching* is widely used in the DDRx DRAM family [8, 26], where *n* stands for the multiplier of the width of internal data bus over I/O bus. 2-/4-bit prefetching is implemented in DDR and DDR2 while 8-bit prefetching is in DDR3 and DDR4⁴. The *n*-bit prefetching is used to address the asymmetric bus frequencies between the I/O bus and the internal bus. As highlighted in Figure 1, the data frequency on I/O bus is 1.6GHz (double data rate with 800MHz clock) while the internal bus only runs at 200MHz (single data rate)⁵. Since data bandwidth is calculated as the product of data width and data frequency (as shown by Equation 1), 128b data is fetched at each time to provide the same bandwidth as the I/O bus requires. Accordingly, the data burst length is eight and only 16b data are sampled per clock edge within the bank.

$$B_{DataBandwidth} = W_{DataWidth} \times F_{DataFrequency} \quad (1)$$

2.2. The Opportunity for Activation Power Reduction

The Power Inefficiency in Overfetching. The first issue we seek to address is the power inefficiency caused by the row overfetching. To understand why and how the row overfetching introduces power inefficiency, CACTI-3DD [23] is

⁴Please do not confuse the term ‘8-bit’ as the 8b data width used in many other literatures. In DRAM, 8-bit prefetching in fact determines the internal data fetching primitives and the corresponding burst length.

⁵The internal bus can run faster with narrower bus width. For simplicity, we assume all data is given in one transfer as the data bandwidth is constant.

Table 2: Power Parameters Based on Micron Datasheet [11]

Row Size	IDD0	IDD3N	IDD2N	tRAS	tRC
16,384	49mA	37mA	23mA	35ns	48.75ns
8,192	42mA	35mA	23mA	35ns	48.75ns

deployed to further break down the power inside a DRAM chip. The result is shown in Table 1. It is obvious that the power is mainly consumed on the bitline (local bitline) and the sense amplifier (local SA), where activation and precharge operations manifest. Moreover, **the activation power is proportional to the number of bitlines being activated during a memory access**. Figure 3a shows the change of activation energy as the number of bitlines driven by a single wordline increases, where the energy proportionality is clearly captured. For future memory chips with larger capacity and more bitlines, this power efficiency problem of row activation will obviously become even worse.

We also use data from industrial reports to further prove the necessity of fine-grained activation. We compare the **pure** current dissipated by activation in two DRAM chips that have 16Kb and 8Kb row buffer sizes, respectively. According to the power calculator [15], the pure activation current is calculated by Equation 2, where *IDD0* is the raw activation current during a row cycle *tRC*, and *IDD2N* and *IDD3N* are the static currents for the situation that all banks are idle or at least one bank is active, respectively. Table 2 shows the power parameters we collect from Micron DDR3 DRAM power datasheet [11]. With these parameters, the activation current consumed for opening a 16Kb row is 16mA and becomes 10mA for 8Kb row. As a result, 37.5% activation current is reduced by halving the row size, which demonstrates the effectiveness of fine-grained activation. Note that the power saving is less than the expected 50%, which stems from the larger static current *IDD3N* in a 16Kb row. If we hypothetically assume that 8Kb row has the same *IDD3N* as 16Kb has (37mA), then the current saving can be as much as 43.7% and is close to 50%⁶. Based on these results, we are confident that halving the row size (or bitline

⁶The assumption here is realistic since both chips have the same capacity. To our knowledge, the more complicated row decoder in 8Kb DRAM consumes more power during activation. Therefore the power saving can never reach the ideal 50%.

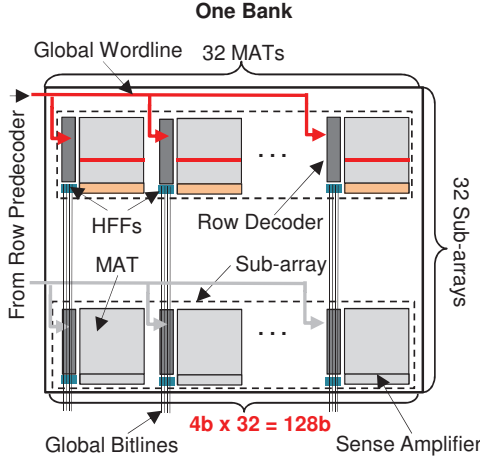


Figure 4: Zoom-in view of the fine-grained structure inside one DRAM bank. The bank has 32 sub-arrays and each sub-array contains 32 MATs. Global bitlines are shared by all sub-arrays. Every MAT can only provides 4b data to form 128b/bank data width.

number) can significantly reduce the activation power.

$$I_{ACT} = IDD0 - \frac{IDD3N \times tRAS + IDD2N \times (tRC - tRAS)}{tRC} \quad (2)$$

The Dilemma of Fine-Grained Activation. According to the analysis above, it is straightforward that activation power can be reduced by limiting the number of involved bitlines during a row activation. Previously, *fine-grained activation*[1] and *selective bitline activation*[2] have been proposed to take advantage of the intrinsic fine-granularity structure of DRAM so as to reduce the number of active bitlines. We generally denote these techniques as fine-grained activation.

Figure 4 illustrates the fine-grained memory architecture inside a DRAM bank. In fact, a bank can be further vertically divided into many sub-arrays as the figure shows. Each sub-array horizontally consists of multiple cell matrices (MATs), which are the atomic access units (square blocks in the figure) for a single memory operation. Each MAT has a local row decoder and a local sense amplifier array (a.k.a row buffer). Typically, a MAT has 512×512 storage cells in row (wordline) and column (bitline) dimensions. Given the 512×512 MAT size, there are totally 32 sub-arrays in one bank and 32 MATs per sub-array. Note that the prefetched data chunk comes from **all** 32 MATs within a sub-array and each MAT only contributes 4b data. There are dedicated helper flip-flops (HFFs) in each MAT to further latch the selected data and then relay them on the internal data bus [27].

Figure 5 further illustrates how the intrinsic fine-grained structure inside a DRAM bank is leveraged by fine-grained activation in the prior work [1, 2]. As shown in the figure, an activation decoder is deployed to determine which MATs should be activated within a sub-array. The posted-CAS (column address strobe) command is used so that the activation

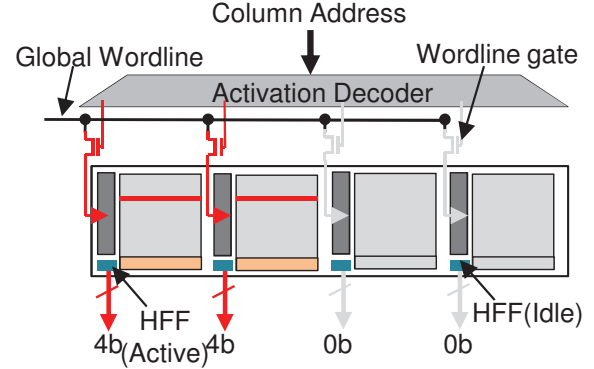


Figure 5: Fine-grained activation in prior work [1, 2]. An activation decoder is introduced to control the number of active MATs. However, halving the active MATs also halves the data width and thus reduces bandwidth.

decoder can know the MAT ID in advance from the column address.

Even though the idea of fine-grained activation is attractive, naively implementing it without careful re-design inevitably incurs significant performance degradation as they fail to comply with the n-bit prefetching [1, 2]. As a simple example illustrated in Figure 5, the left two MATs of a sub-array are normally activated while the right two are still inactive. Therefore, the output data width is reduced by half correspondingly. Deducing the example to the commodity DRAM, the fine-grained activation can severely destroy the n-bit prefetching and thus reduce the data bandwidth of one bank. To mitigate the reduction on bandwidth, it is straightforward to increase the narrow data width of a MAT for compensation, similar to what the selective bitline activation [2] does. However, this approach is not practical because the data width is constrained by the limited routing resources as well as the available number of HFFs and secondary latches used as drivers for a long intra-chip data path. In other words, it is difficult to increase the data width without incurring large area overhead, which has also been pointed out by T. Vogelsang [28]. Since the data frequency does not change, the only way to deliver an atomic 64Byte data chunk⁷ is to increase the burst length accordingly. However, this will increase the data transfer time on a DRAM bus because the bursts are serialized, and therefore will degrade the memory performance.

To evaluate the impact of data bandwidth loss on the system performance, we run the STREAM benchmark [29] as a representative of memory-intensive applications. Figure 3b shows the respective results for the cases when only 1/2, 1/4, or 1/8 of a bank can be activated at a time. Compared to the full bank activation, the original method suffers a lot from the narrow data width a MAT can provide. In particular, the 1/2 bank activation without careful design can experience a 30% performance loss. Even worse, an unacceptably high 84%

⁷Given a 64b I/O width and a burst length of eight, one memory request can deliver $64 \times 8 = 512b$ data, which is equal to 64Byte.

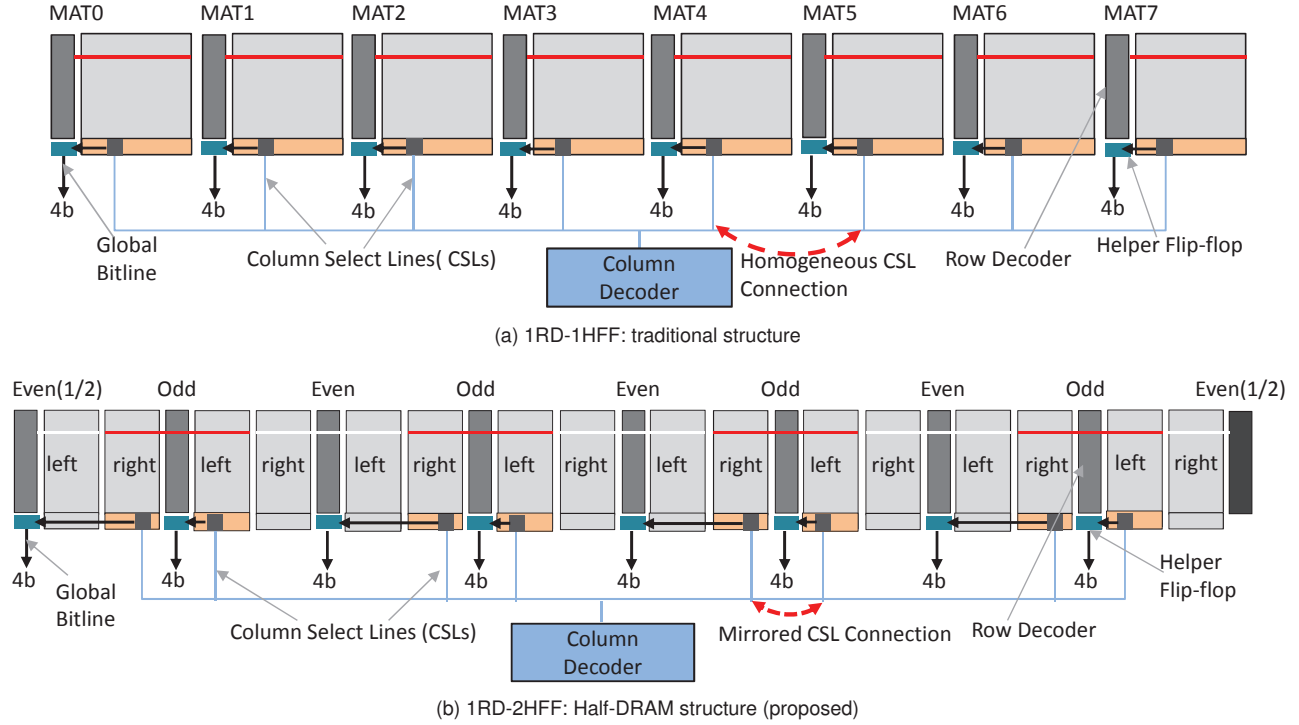


Figure 6: The reorganization of sub-array in Half-DRAM. (a) Traditional sub-array schematic. One-to-one relationship exists between row decoder and HFF since the row decoder drives the entire wordline of one MAT; (b) A MAT is split into “left” and “right” block and they are driven by different row address decoders. The sub-array is further divided into Odd and Even groups.

performance loss is observed when 1/8 bank is activated (four MATs involved as the example in Figure 4). Therefore, we conclude that **naively activating fewer MATs to reduce activation power can destroy the n-bit prefetching and thus cause significant memory bandwidth drop**. In the extreme scenario of 1/8 bank activation, 8-bit prefetching is completely destroyed and 7/8 of the bandwidth is wasted. The wasted bandwidth explains the significant performance degradation as it now needs 64 bursts to deliver the amount that could be transferred within 8 bursts originally (i.e., a 64Byte last-level cache line).

Due to the aforementioned issues, fine-grained activation must be carefully rethought to make sure it does not degrade memory bandwidth nor incur large area overhead. In this work, Half-DRAM is proposed to make better trade-offs among performance, power, and area. A novel fine-grained memory architecture is designed so as to activate half the MATs in one sub-array while still providing full 128b data with negligible area overhead.

3. Design of Half-DRAM

From our observation, the major problem in prior fine-grained activation techniques is the incapability of reusing the row buffer and the associated functional components on the data path (e.g., HFFs and global bitlines). The incapability stems from the one-to-one relationship between the row decoder/row buffer and the HFF.

Figure 6 shows the reorganization of sub-array in Half-DRAM with comparison to the baseline DRAM design. As shown in Figure 6a, the local row decoder drives the associated wordline, which further activates the entire row data into the row buffer. All bits in the row buffers compete for a common HFFs shown on the left. We name such one-to-one relationship as “1RD-1HFF”, which means one HFF group is dedicated to one row decoder and vice versa. This 1RD-1HFF hardware organization in traditional DRAM is kept the same in the prior work of fine-grained activation [1, 2], which is the substantial reason why the bandwidth cannot be further improved.

3.1. 1RD-2HFF v.s. 1RD-1HFF

We propose a novel structure to enable the fine-grained activation without bandwidth loss. Figure 6b shows the basic idea of Half-DRAM. First of all, Half-DRAM breaks the original 1RD-1HFF restraint by splitting one MAT into two identical slabs: **left** and **right**. Distinct from 1RD-1HFF, a row decoder is now driving the left and right slabs that originally belong to two neighbor MATs. Consequently, once the row decoder selects a wordline, both MATs are activated but each with half a row. In other words, *even if every other row decoder is disabled* (white wordlines in 6b), *all HFFs can still be active whilst only half of a row has valid data* (red wordlines). Therefore, the sub-array is **logically** divided into two groups, which are respectively labeled as *Odd* and *Even* as Figure 6b shows. Similar to previous work, a transistor is deployed to

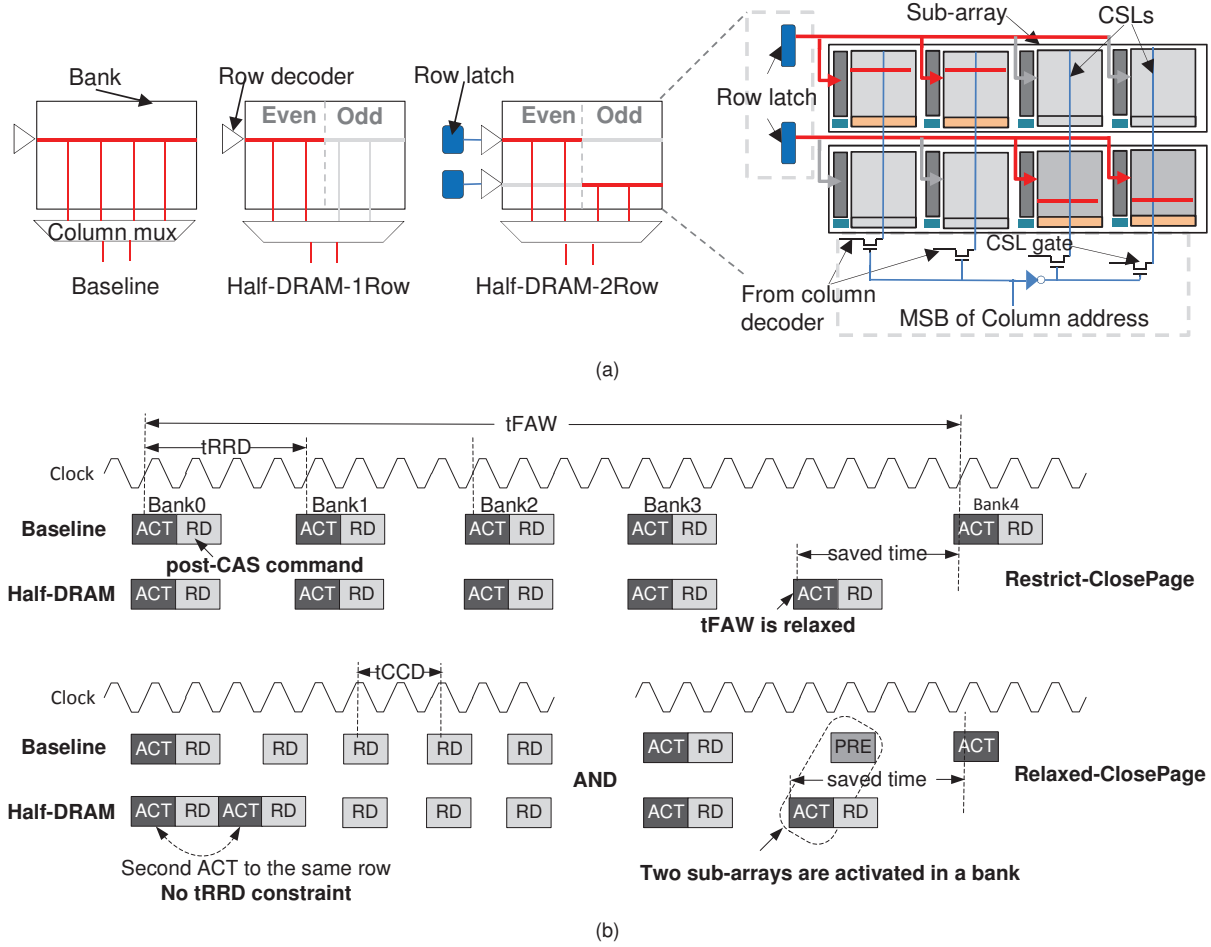


Figure 7: The proposed Half-DRAM models with the timing illustration. (a) The logic view of Half-DRAM-1Row and Half-DRAM-2Row (left) and the row/column control design for Half-DRAM-2Row; (b) Timing diagram to illustrate the relaxation of tFAW as well as the integration of sub-array level parallelism [10] for performance improvement.

control the gating of the undesired MATs, which is shown in Figure 5. Note that there are two slabs standing alone at the head and tail of a sub-array, which actually belong to the Even group. An additional row decoder is introduced to drive the tail (left) slab for simplicity. The extra row decoder can even be removed to eliminate area overhead as we will elaborate in Section 4. In this way, Half-DRAM establishes a new “1RD-2HFF” relationship between a row decoder and a HFF.

3.2. Half-active DRAM

Figure 7 shows the logic views of various Half-DRAM designs. By leveraging the column select line (CSL) gate shown in Figure 7a, fine-grained activation can be easily enabled in 1RD-2HFF as the Even group is active while the Odd group is idle, or vice versa. The MSB of column address is used to determine which half of a bank should be activated. The posted-CAS technique allows ACT and CAS commands to be issued back-to-back, which can be leveraged in this work to know the column address in advance. Once such fine-grained activation is applied to commodity DRAM, only half of a DRAM chip

is active for each memory request (so-called “Half-DRAM”). Distinct from prior work [1, 2], all components on the global data paths, including the HFFs and the global bitlines, can be fully utilized to sustain the 8-bit prefetching and provide the required 128b data of each DRAM chip (8 bursts of 16 bits).

In addition, the column select logic should be redefined to make sure only the valid data can be selected out of the row buffer. Figure 6 also illustrates the different column select logic for this goal. For the traditional 1RD-1HFF, every MAT has the same semantic to interpret the inputs of CSLs. This is achieved by the homogeneous CSL connection to the output of column decoder. Alternatively, in Half-DRAM the semantic is completely mirrored in any two neighboring MATs. If we assume one MAT still complies with the original column select semantic, then the neighboring MAT needs to have the reversed semantic for the correct selection. Fortunately, this can be simply achieved by mirroring the CSL connection without any overhead (see Section 4).

Even though Half-DRAM intelligently mitigates the bandwidth reduction problem, it only has half of the row buffer size,

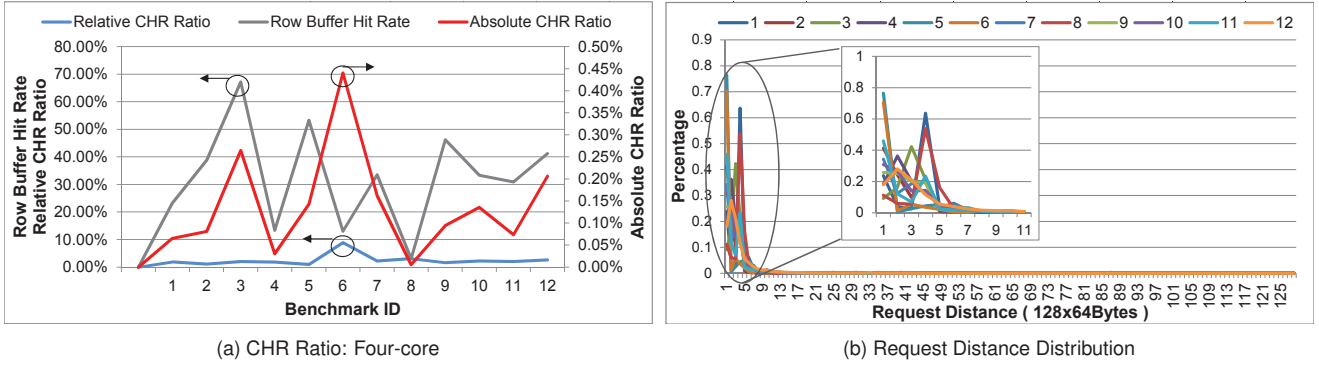


Figure 8: The impact of half-size row buffer. All benchmarks have good data locality so that only <0.5% accesses cross half row during a burst of row buffer hits.

which may induce performance overhead due to degraded row buffer hit rate. Fortunately Half-DRAM can be easily extended to the traditional full bank activation without any extra design effort to provide full row buffer size. The second activation does not have to bear the tRRD because the row address has been pre-decoded by the previous activation. Consequently, it neither incurs timing penalty as shown in Figure 7b.

Depending on the row buffer management policy, Half-DRAM can render different behaviors. In this work, two close-page policies are evaluated. The first close-page policy, *Restrict-ClosePage*, closes the row immediately after the access is completed. No row buffer hit can be exploited in this policy. The other close-page policy, *Relaxed-ClosePage*, closes the row only when there is no more requests in the queue that can benefit from row buffer hit. Obviously, for the *Restrict-ClosePage*, Half-DRAM can gain maximum power saving without performance overhead. For *Relaxed-ClosePage*, two half-row activations are issued if and only if two requests in the queue reference to both Odd and Even group. In this way, Half-DRAM enables a fine-grained, on-demand-activation memory system.

3.3. Half-size Row Buffer: Challenge or Opportunity

Even though Half-DRAM can effectively reduce the activation power, it may induce more activations because of the smaller row buffer size. To justify the feasibility of Half-DRAM, we evaluate the performance impact of half-size row buffer. We first define the *ratio of crossing half row* (CHR ratio) as the percentage of two requests referencing to the same row but different half rows when *Relaxed-ClosePage* policy is applied. The lower the ratio is, the more requests are served within a half row. Specifically, zero CHR ratio means all requests go to the same half row at each memory access and there is no performance penalty at all. Note that we only count the requests that have row buffer hits, other requests that close the row immediately (due to row conflicts) have been excluded.

Two CHR ratios are defined in Equation 3 and 4. The *absolute CHR ratio* represents the percentage of half row crossing over all requests, while *relative CHR ratio* shows the actual

number of requests that have data accesses across half rows upon a row buffer hit. Therefore, relative CHR ratio filters out the noise from requests with low row buffer hit rates. Figure 8a shows the results of the 4-core simulation. Obviously, all benchmarks have quite low absolute CHR ratios (<0.5%, red curve in Figure 8a). Even the relative CHR ratio (blue curve) is very low, which is no more than 8%. The low absolute and relative CHR ratio indicates a great opportunity for Half-DRAM to save power without performance loss. Also, note that the CHR ratio and row buffer hit rate are not necessarily correlated. For example, *test3* has high row buffer hit rate (67.2% in gray curve) but it only has 0.26% row crossing rate. On the other hand, *test6* has 13% row buffer hit rate while it shows the highest CHR ratio as 0.44%.

$$CHR_Ratio_{absolute} = \frac{NUM_{CHR}}{NUM_{total_req}} \quad (3)$$

$$CHR_Ratio_{relative} = \frac{NUM_{CHR}}{NUM_{row_buffer_hit}} \quad (4)$$

To understand the cause of low CHR ratio, we also quantify the request distance, which is defined as the distance between two requests that access the minimum and maximum column address during a series of row buffer hits. The distance distribution is shown in Figure 8b with 64Byte sampling unit. Given our baseline has 64Kb row in a rank (4chip×16Kb), there are totally 128 sampling units on the X-axis. As shown, the requests have pretty high data locality as >99% requests occur within 8-unit distance. As a result, the CHR ratio is minimized by the high data locality in a row and thus reveals a promising opportunity for Half-DRAM to reduce activation power.

3.4. Chance of Improving Memory Parallelism

To prevent too frequent activations from generating a high current that exceeds the predefined current threshold, two timing parameters are defined in commodity DRAM. While the row-to-row activation delay (tRRD) specifies the minimum interval of two successive activations, the four-activation-window constraint requires that there must be no more than four ac-

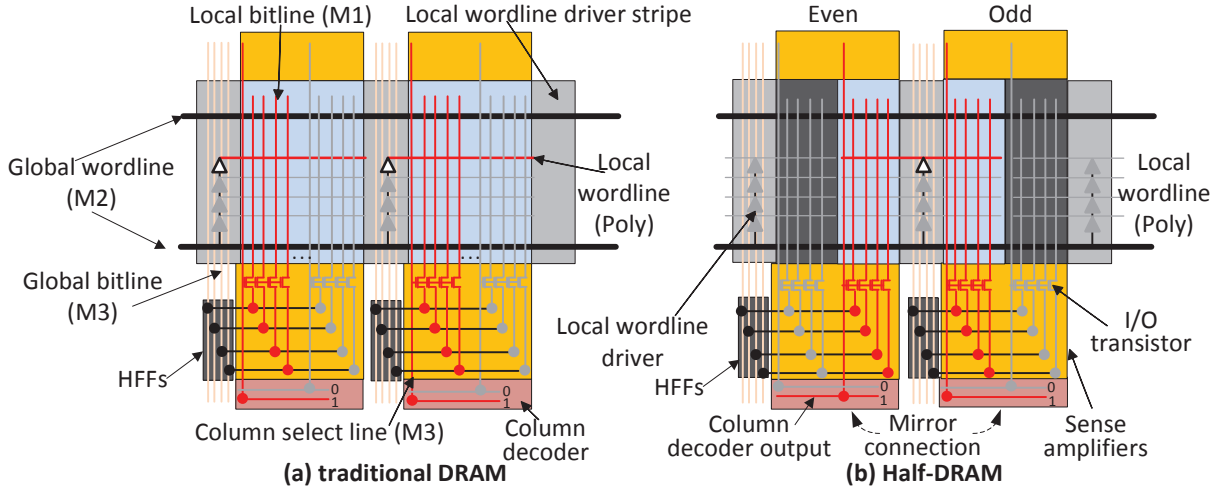


Figure 9: The circuit design of Half-DRAM-1Row. All metals layers are given for the illustration. Note that how data is selected out and relayed on HFFs [27]. Obviously, Half-DRAM-1Row can completely reuse all components and wires without any overhead. The connection of column decoder output in an Even MAT is mirrored in an Odd MAT.

tivations in the rolling window tFAW as shown in Figure 7b. Half-DRAM is promising in that it can relax these power constraints (manifested as timing constraints), resulting from its design to activate fewer bitlines each time and thus consume less activation power. Specifically, the four-activation window constraint can be relaxed since now eight half-row activations are permitted in any tFAW window. Such improvement can help the memory with Restrict-ClosePage applied since the performance of such memory is mainly limited by tFAW [24].

In addition, *Half-Bank level parallelism* is introduced in Half-DRAM to further improve the memory parallelism. Different from the basic design, two half bank latches are utilized to decouple Odd and Even groups from each other. As a result, sub-array level parallelism [10], which uses sub-arrays as independent memory operation corresponder, can be integrated into Half-DRAM seamlessly. As shown in Figure 7b, two half-rows in different sub-arrays can be activated without data path contentions as long as they belong to different half-rows, which effectively doubles the memory parallelism.

In this work, two Half-DRAM schemes are proposed as shown in Figure 7a with the baseline illustrated at the leftmost. The intermediate Half-DRAM that only permits one half-row activation is presented in the middle (*Half-DRAM-1Row*), meaning that no further activation can be issued even if the next activation goes to the inactive half-row. Alternatively, the Half-DRAM that allows any two half-rows to be active is given at the rightmost side of the figure (*Half-DRAM-2Row*). In fact, Half-DRAM-1Row can be easily extended to *Half-DRAM-1Row-Demand*, in which the other half of the same row can be activated immediately as long as no precharge is issued. Note that we retain the same assumptions that no more than eight half-row activations can be issued in the tFAW rolling window and two continuous half-row activations must comply with the tRRD constraint whenever they go to different rows.

4. Design Overhead Analysis

As DRAM is competing at a thin profit margin, and the cost (known as \$/bit) is very sensitive to area increase, any change in DRAM structure should be assessed with area overhead analysis. In this section, we will conduct a detailed analysis on DRAM area overhead to justify the practicality of our Half-DRAM design. Typically, the commodity DRAM is implemented with three metal layers [23, 28]. This design can be well leveraged by our Half-DRAM design. Figure 9 presents the circuit design of Half-DRAM. Instead of driving a local wordline that traverses all 512 bitlines in a single MAT (Figure 9a), the wordline can be horizontally shifted and cover a half row from both neighboring MATs, respectively. Note that the two half-row wordlines share a common row logic stripe (Figure 9b). In other words, the local wordline driver becomes bi-directional, and drives the wordline (or row) at both directions. Obviously, this design does not incur extra area or routing overhead, given that it only involves a horizontal shift of the metal wordline and only requires a few additional metal vias to connect to the driver at circuit level.

Figure 9 also shows the proposed design change of column select lines (CSLs) routing. In the original DRAM layout as Figure 9a shows, the CSLs are organized in an order that is identical between MATs. One CSL connects to four I/O transistors to select data out. Therefore, each MAT can have only one $\times 4$ column selected and buffered in the HFFs assigned to the MAT. In Half-DRAM, in order to select the desired data within one even or odd DRAM row, the connection of CSLs to the output of column decoder is mirrored between even and odd MATs. For example, two CSLs are shown in the figure with each from the left and right slab, respectively. Originally, if the output of column decoder is ‘10’, then the columns on the left half in both MATs are selected due to the homogeneous connection (in red color in Figure 9a). Instead, as the CSL connection is mirrored, the same decoding output

‘10’ selects the right slab of the first MAT (Figure 9b). In this way, two columns (or $2 \times 4 = 8$ bits) in each activated even (or odd) row are selected and transferred to their corresponding HFFs. Note that these “mirrored” CSL design will apply to each pair of MAT across the entire sub-array. Because the mirrored wire connection can be done outside the cell array, there is no circuit or routing overhead incurred.

The only overhead in Half-DRAM-1Row is the extra row decoder at the tail of each sub-array. According to Table 1, the total footprint of the row decoder is 6.84 mm^2 (local wordline driver + row predecoder), which is 18.4% of the DRAM die area (37.129 mm^2). Since the ratio of the extra row decoder is 1/32, the area overhead is roughly 0.58% ($18.4\% \times 1/32$). In addition, as the original row decoders already consume much lower power than the bitline and SA, the power overhead caused by the additional row decoders is negligible. We denote this design option as *Power-OPT* (power optimization) since it maximizes the power saving with modest area overhead. Furthermore, even this marginal area overhead can be saved by removing the extra tail row decoders. Instead, the half-row at the tail can be driven by the last Odd row decoder and this decoder is always activated once the row is selected, similar to what the baseline does. We denote this design option as *Area-OPT* (area optimization). Of course, Area-OPT is at the cost of lowering the energy savings from roughly 1/2 to 29/64 as the last three half MATs are always activated. The elimination of area overhead, however, encourages us to apply Area-OPT in our work. In summary, Half-DRAM-1Row does not incur any area overhead.

On the other hand, Half-DRAM-2Row requires extra row and column logic shown in Figure 7a. Similar to sub-array level parallelism [10], two row address latches are deployed and each is sized around 40b. Note that **Half-DRAM-2Row naturally avoids the global bitline contention due to the exclusive column selection**. Therefore it removes the designated-bit latches used in multiple activated sub-arrays [10]. This is a unique advantage of Half-DRAM-2Row. In the column decoder, the most-significant bit of column address is used to gate half of CSLs and its reverse signal gates the other half. As one bank has 1,024 CSLs, 8,192 gates are needed for a whole chip. Similarly, 8,192 gates are also needed for wordline gating. Since the gate is implemented by a single transistor, its area overhead is completely negligible (note that one chip has $>1\text{G}$ bits and each bit has a transistor). In addition, we use Design Compiler [30] to synthesize the row latches with TSMC 45nm-1.05V process. The results show that these latches only occupy $786 \mu\text{m}^2$ and consume $427 \mu\text{W}$ in a chip. Compared to the activation power that can be as much as 24 mW ($=16 \text{ mA} \times 1.5 \text{ V}$), the power overhead of the latches is trivial. The wire routing overhead is also negligible since the area of the second 40b address bus is only about $0.081 \mu\text{m}^2$ at 45nm node. In summary, the total area and power overhead caused by Half-DRAM is conservatively estimated as less than 0.003%, which is negligible.

Table 3: Simulation Platform Configuration

Cores	4, ALPHA, out-of-order
CPU Clock Freq.	3 GHz
LDQ/STQ/ROB Size	32 / 32 / 128 entries
Issue/Commit Width	8 / 8
L1-D/L1-I Cache	32kB / 32kB 4-way 2-cycle latency
D-TLB/I-TLB Size	64 / 48 entries
L2 Cache	Shared, Snooping, 4MB, LRU 8-way, 10-cycle latency
Memory	
General	DDR3-1600, 4GB, 64bit I/O, 4 ranks, 2Gb chip, 8 banks($\times 16$), FR-FCFS, RD/WR queues, 64/32 entries,
Timing (in cycle)	tRCD-tCAS-tRP-tWR: 11-11-11-12, tRAS-tCCD-tRRD-tFAW: 28-4-6-32
Power (fast-exit PD)	49/42mA(IDD0), 15mA(IDD2P), 23mA(IDD2N), 37mA(IDD3N), 135mA(IDD4R), 146mA(IDD4W), 182mA(IDD5A), 1.5V(V_{DD})
test# Benchmarks (SPEC2006+STREAM)	
¹ STREAM $\times 4$, ² bwaves $\times 4$, ³ gobmk $\times 4$, ⁴ leslie3d $\times 4$, ⁵ libquantum $\times 4$, ⁶ lbm $\times 4$, ⁷ mcf $\times 4$, ⁸ milc $\times 4$, ⁹ STREAM-gobmk-lbm-libquantum, ¹⁰ bwaves-leslie3d-mcf-milc, ¹¹ lbm-libquantum-bwaves-leslie3d, ¹² STREAM-gobmk-mcf-milc	

5. Evaluation Results

In this work, we used gem5 [31] as our simulation platform. We integrated the NVMain [32] into gem5 as the DRAM model. Table 3 shows the simulation setup. The DRAM timing and power parameters are excerpted from Micron’s data sheet [11]. Based on the power analysis in Section 2.2, the IDD0 of an 8Kb row (42mA) is used as the half-row activation current. FR-FCFS memory scheduling policy [33] is deployed in the memory controller with separate read/write queue. The selected SPEC2006 CPU benchmarks with reference input size [34] and STREAM with all functions [29] are evaluated as multi-programmed tests. Eight benchmarks that have high MPKIs (miss per kilo instructions) are selected and each benchmark is either duplicated or mixed for the four-core simulation. The four-core benchmarks are listed at the bottom of Table 3, where each of them is given a test number. We run all benchmarks for 500 million instructions for cache warmup and then the following 100 million instructions for statistics. The weighted IPC (instructions per cycle) defined in Equation 5 is used as the performance criteria for the four-core simulation. The aforementioned two Half-DRAM models: Half-DRAM-1Row and Half-DRAM-2Row, are evaluated and compared to the baseline. Obviously, they represent the lower and upper bound of performance improvements, respectively.

$$WeightedSpeedup = \sum_{i=1}^n \frac{IPC_{multi-core}^i}{IPC_{standalone}^i} \quad (5)$$

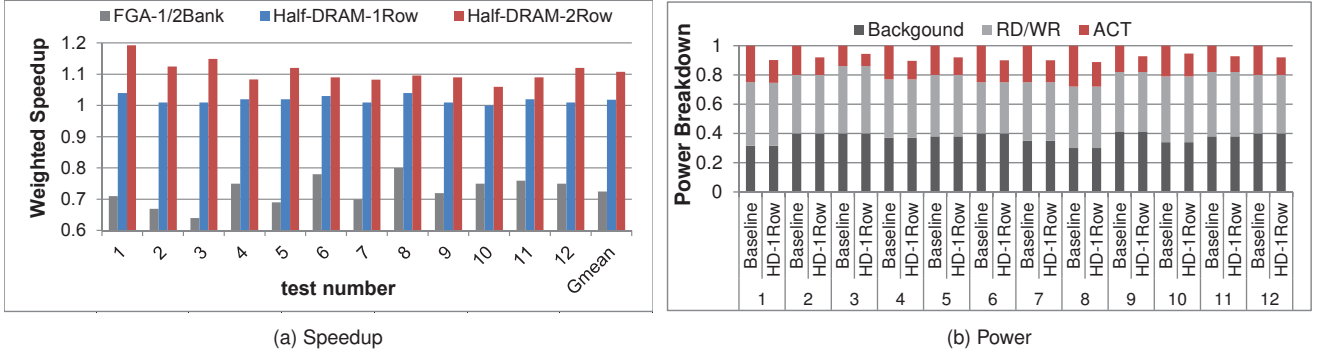


Figure 10: Four core simulation results. All tests are run with the Relax-ClosePage scheme and all results are normalized to baseline. The FGA-1/2Bank corresponds to 1/2 bank activation in prior work [1, 2]. The burst length of FGA-1/2Bank is 16.

5.1. Performance Analysis

The performance results of the four-core simulation are shown in Figure 10a. To show the advantage of Half-DRAM, the result of the prior work with 1/2 bank activation is also evaluated and denoted as *FGA-1/2Bank*. The burst length in FGA-1/2Bank is set to 16 (8×2) to compensate the bandwidth loss discussed in Section 2.2. All tests have been normalized to the baseline where full bank activation and the Relaxed-ClosePage policy are applied. First, all tests suffer severe performance degradation, from 20% (test8) to 36% (test3). Since test3 has high row buffer hit rate and intensive memory accesses, the reduced bandwidth leads to much more contention on the data bus, which can explain the performance drop.

In contrast, thanks to the extremely low CHR ratio, no tests suffer from performance drop in Half-DRAM-1Row. Moreover, test6 that has the highest CHR ratio even shows 3% performance improvement. The reason of the improvement is that it has a low row buffer hit rate so that Half-DRAM-1Row can take advantage of the relaxation of Four-activation-window constraint to overwhelm the slight increase of activation number. On average, Half-DRAM-1Row can improve the performance by 1.3%. Even though the performance improvement is trivial, Half-DRAM-1Row does not induce performance degradation. In addition, Half-DRAM-2Row shows a promising performance improvement over baseline by leveraging the sub-array level parallelism. In particular, test1 can achieve as much as 19% performance improvement. The performance gain comes from the relatively low data locality, which can utilize the half-bank parallelism well. Again, the relaxation of tFAW also boosts the performance gain. The average performance improvement in Half-DRAM-2Row is 10.7%.

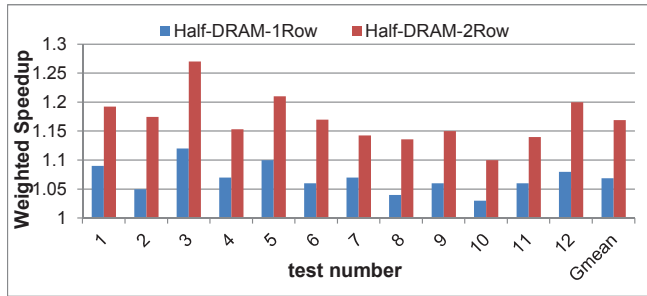
5.2. Power Analysis

To verify the power reduction of Half-DRAM, we use NVMain to collect the active, standby, and powerdown cycle numbers as well as the read and write request numbers that are required by Micron’s power calculator [15]. Note that the IDD0 of half-bank activation is set as 42mA so that every activation can save 43.7% power as discussed in Section 2.2. The gen-

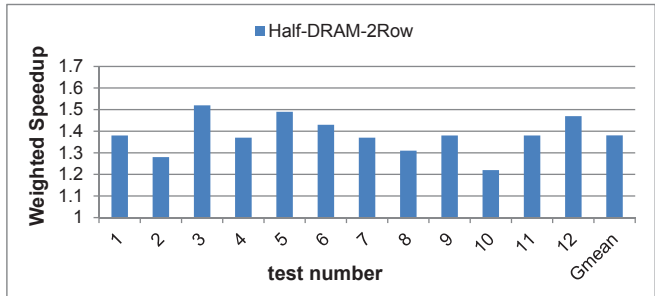
erated four-core power results are shown in Figure 10b with the power breakdown in detail. Since Half-DRAM-1Row has almost the same runtime as the baseline, its power is given so that we can concentrate on the power reduction from activation. As shown in the figure, the power efficiency varies among the benchmarks. For instance, test1 and test6 have significant power reduction, which is up to 9.1% and 10%, respectively. In contrast, the power reduction from test3 and test9 is relatively small (5.8% and 7%). The effect of fine-grain activation in Half-DRAM is mainly determined by the intensity of activations at runtime. From Figure 10b, it is obvious that the activation power in test1 is big enough to reflect the power efficiency of Half-DRAM. On the other hand, the power gain is limited because the high buffer hit rate can effectively amortize the activation power, like what happens in test3. In general, Half-DRAM-1Row can achieve 8.4% improvement on power efficiency over the baseline.

5.3. The Effect of the Relaxation of Power Constraint

Impact of Four-activation-window Constraint Compared to Relaxed-ClosePage policy, the performance of Restrict-ClosePage policy is limited by Four-activation-window constraint because any memory request requires an activation/precharge while the constraint limits the activation frequency [24]. According to the value of tRRD and tFAW shown in Table 3, there can be $5.3 - (\frac{tFAW}{tRRD})$ activations in a tFAW if no Four-activation window constraint is applied. In other words, DRAM loses 25% ($= \frac{tFAW - 4 \times tRRD}{tFAW}$) activation bandwidth due to the tFAW constraint. As Half-DRAM alleviates the power constraints, the activation rate can be improved accordingly. To verify the advantage of Half-DRAM, we re-run the simulation and the results are shown in Figure 11a. The average speedup in Half-DRAM-1Row is 6.9%. Compared to the results of Relaxed-Close Policy is applied (Figure 10a), Half-DRAM-1Row has even higher performance improvement, which unsurprisingly comes from the relaxation of four-activation-window constraint. In particular, test3 has 11.8% improvement as the requests going to the same rows that originally have to wait for the row re-activation can now



(a) Restrict-ClosePage + Half-DRAM



(b) Wide I/O + Half-DRAM-2Row

Figure 11: Simulation results of power constraint study. Restrict-ClosePage management policy is applied. (a) Weighted speedup by relaxing tFAW constraint; (b) Better performance improvement due to higher tFAW penalty in Wide I/O.

be served earlier. Even better, Half-DRAM-2Row has further performance improvement over all tests. The additional improvement is from enhanced memory parallelism as expected. At this time, two requests that go to the Odd and Even group separately can be served in parallel. In general, 16.9% performance improvement is observed in Half-DRAM-2Row.

A Case Study of Half-DRAM So far, it is clear that the performance advantage of Half-DRAM is promising in a memory that suffers from the restricted power constraint. Wide I/O [9] is such a memory in which the Four-activation-window is further restricted to Two-activation-window (tTAW) due to the challenging power delivery issue in 3D-stacked memory [35, 36]. In other words, in any tTAW there can be only two activations allowed. Considering the larger window in Wide I/O memory (tTAW=50ns and tRRD=10ns), the power constraint more severely suppresses the performance⁸. We apply Half-DRAM-2Row to Wide I/O memory to assess the possible performance gain. Note that all requests still go to one channel because the Two-activation-window is applied within each channel. The results are given in Figure 11b. Not surprisingly, the average performance improvement is up to 38.1% while test3 and test5 gain as much as 52.4% and 49.2% speedup, respectively. Therefore, Half-DRAM is very effective in Wide I/O memory to exploit the potential performance benefit.

6. Related Work

Several fine-grained DRAM structures have been proposed in conventional 2D DRAM. For example, Fujitsu implements the Fast-cycle RAM (FCRAM) that has a sub-bank structure and achieves faster access speed and lower power consumption than the baseline 2D DRAM [37]. Unfortunately, FCRAM has low cell density that leads to limited memory capacity. Similarly, Reduced-latency DRAM (RLDRAM) [38] was introduced with a smaller bank size for low access latency. RLDRAM, however, also induces large area overhead so that

⁸Wide I/O augments the channel-level parallelism to alleviate the constraint as it has four independent channels. Our concern, however, is that the power constraint has effect within a channel so that it eventually suppresses the performance when DRAM utilization is high.

the capacity was only about 40% of commodity DRAM. As mentioned earlier in the paper, Cooper-Balis *et al.* [1] and Udipi *et al.* [2] proposed to leverage fine-grained access in the commodity DRAM to reduce power. However, both work fail to comply with the n-bit prefetching, and the implementation overhead to sustain full data bandwidth is significant [28].

Sub-rank level parallelism has been studied as another level of fine-grained structure. Zheng *et al.* introduced a bridge chip MRB to split the original rank into mini-ranks [39]. The mini-rank design not only significantly reduces power consumption, but also improves the memory parallelism so as to compensate the potential performance degradation from narrowing down the data bus. In spite of the power optimization, the extra MRB increases the DRAM cost and it still suffers from the bandwidth loss. Leveraging the mini-rank structure, D. Yoon *et al.* implemented a memory system that has adaptive access granularity at rank level [40]. The adaptive granularity memory system, however, requires the co-design of a corresponding fine-grained cache architecture, which significantly affects its flexibility. To summarize, without reasonable optimizations in DRAM core, it is hard to achieve a good trade-off between performance (bandwidth) and power. All above approaches always improve one aspect by sacrificing the other. Distinguished from the previous work, Half-DRAM takes a holistic consideration of both performance and power and thus can achieve better compromise in between.

7. Conclusion

The power consumption for DRAM memory can be a significant portion of the total power consumption in modern computing systems, especially for server and data center systems. Consequently, reducing DRAM power is critical for power-efficient computing. In this work, we proposed *Half-DRAM* as a novel memory architecture that can practically activate only half a DRAM bank to reduce activation/precharge power. Compared to previous work [1, 2] that may induce severe performance or area overhead due to the reduced data bandwidth, Half-DRAM can enable the fine-grained activation with full data bandwidth by leveraging the “1RD-2HFF” structure. In addition, by exploiting the sub-array level parallelism [10],

half-bank level parallelism is developed to take advantage of the relaxed power constraint to further improve the memory performance. Depending on the row buffer management policy, the experimental results show that Half-DRAM can achieve 10.7% and 16.9% performance improvement with 8.4% power saving, with negligible hardware overhead. With the integration of Half-DRAM, Wide I/O can obtain 38.1% performance gain that is promising for its improvement. As a result, we believe that Half-DRAM can be a promising enhancement to the conventional DRAM for future power-efficient computing.

Acknowledgement

We would like to thank reviewers' valuable comments on the paper. This work is supported in part by National Science Foundation (1213052, 1218867), Department of Energy under Award (DE-SC0005026), National Natural Science Foundation of China (61103028, 61202072), National High-tech R&D Program of China (2013AA013201), and AMD Grant.

References

- [1] E. Cooper-Balis and B. Jacob, "Fine-Grained Activation for Power Reduction in DRAM," *MICRO*, vol. 30, no. 3, pp. 34–47, 2010.
- [2] A. N. Udipi, N. Muralimanohar, N. Chatterjee, R. Balasubramanian, A. Davis, and N. P. Jouppi, "Rethinking DRAM Design and Organization for Energy-constrained Multi-cores," in *ISCA'10*, Jun. 2010, pp. 175–186.
- [3] U. Hoelzle and L. Barroso, *The Datacenter as a Computer*. Morgan & Claypool Publishers, 2009.
- [4] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, "Disaggregated Memory for Expansion and Sharing in Blade Servers," in *ISCA'09*, Jun. 2009, pp. 267–278.
- [5] K. Malladi, F. Nothaft, K. Periyathambi, B. Lee, C. Kozyrakis, and M. Horowitz, "Towards Energy-Proportional Datacenter Memory with Mobile DRAM," in *ISCA'12*, Jun. 2012, pp. 37–48.
- [6] D. H. Yoon, J. Chang, N. Muralimanohar, and P. Ranganathan, "BOOM: Enabling Mobile Memory based Low-Power Server DIMMs," in *ISCA'12*, Jun. 2012, pp. 25–36.
- [7] K. Sudan, K. Rajamani, W. Huang, and J. B. Carter, "Tiered Memory: An Iso-Power Memory Architecture to Address the Memory Wall," *IEEE Transactions on Computers*, vol. 61, no. 12, pp. 1682–1696, 2012.
- [8] JEDEC Solid State Technology Association, "JEDEC Standard: DDR3 SDRAM Specification," <http://www.jedec.org/standards-documents/docs/jesd-79-3d>, Sep. 2009.
- [9] JEDEC Solid State Technology Association, "JEDEC Standard: Wide I/O Single Data Rate Specification," <http://www.jedec.org/standards-documents/results/jesd229>, Dec. 2011.
- [10] Y. Kim, V. Seshadri, D. Lee, J. Liu, and O. Mutlu, "A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM," in *ISCA'12*, Jun. 2012, pp. 368–379.
- [11] Micron, "MT41J128M16HA-125 Data Sheet," <http://www.micron.com/products/dram/>.
- [12] JEDEC Solid State Technology Association, "JEDEC Standard: LPDDR3," <http://www.jedec.org/standards-documents/results/jesd209-3>, May 2012.
- [13] Hybrid Memory Cube Consortium, "Hybrid Memory Cube Specification 1.0," <http://hybridmemorycube.org>, Apr. 2013.
- [14] Micron, "DDR3 Power Calculator," <http://www.micron.com/products/dram>.
- [15] Micron, "TN-41-01: Calculating Memory System Power for DDR3," <http://www.micron.com/products/dram>.
- [16] M. Ghosh and H.-H. S. Lee, "Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Conventional and 3D Die-Stacked DRAMs," in *MICRO'07*, Dec. 2007, pp. 134–145.
- [17] I. Hur and C. Lin, "A Comprehensive Approach to DRAM Power Management," in *HPCA'08*, Feb. 2008, pp. 305–316.
- [18] H. David, C. Fallin, E. Gorbato, U. R. Hanebutte, and O. Mutlu, "Memory Power Management via Dynamic Voltage/Frequency Scaling," in *ICAC'11*, Jun. 2011, pp. 31–40.
- [19] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn, "Flikker: Saving DRAM Refresh-power through Critical Data Partitioning," in *ASPLOS'11*, Mar. 2011, pp. 213–224.
- [20] K. Malladi, I. Shaeffer, L. Gopalakrishnan, D. Lo, B. Lee, and M. Horowitz, "Rethinking DRAM Power Modes for Energy Proportionality," in *MICRO'12*, Dec. 2012, pp. 131–142.
- [21] Y. Lee, S. Kim, S. Hong, and J. Lee, "Skinflint DRAM System: Minimizing DRAM Chip Writes for Low Power," in *HPCA'13*, Feb. 2013, pp. 25–34.
- [22] JEDEC Solid State Technology Association, "JEDEC Standard: LPDDR2," <http://www.jedec.org/standards-documents/docs/jesd209-2e>, Dec. 2010.
- [23] B. Jacob, S. W. NG, and D. T. Wang, *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann, 2007.
- [24] D. Kaseridis, J. Stuecheli, and L. K. John, "Minimalist Open-page: A DRAM Page-mode Scheduling Policy for the Many-core Era," in *MICRO'11*, Dec. 2011, pp. 24–35.
- [25] K. Chen, S. Li, N. Muralimanohar, J. H. Ahn, J. Brockman, and N. Jouppi, "CACTI-3DD: Architecture-level Modeling for 3D Die-stacked DRAM Main Memory," in *DATE'12*, Mar. 2012, pp. 33–38.
- [26] JEDEC Solid State Technology Association, "JEDEC Standard: DDR4 SDRAM," <http://www.jedec.org/sites/default/files/docs/JESD79-4.pdf>, Sep. 2012.
- [27] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM Circuit Design: Fundamental and High-Speed Topics*. Wiley-IEEE Press, 2007.
- [28] T. Vogelsang, "Understanding the Energy Consumption of Dynamic Random Access Memories," in *MICRO'10*, Dec. 2010, pp. 363–374.
- [29] J. D. McCalpin, "STREAM Benchmark," <http://www.cs.virginia.edu/stream>.
- [30] Synopsys, "Design Compiler," <http://www.synopsys.com>.
- [31] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi et al., "The gem5 Simulator," *Computer Architecture News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.
- [32] M. Poremba and Y. Xie, "NVMain: An Architectural-Level Main Memory Simulator for Emerging Non-volatile Memories," in *ISVLSI'12*, Aug. 2012, pp. 392–397. [Online]. Available: <http://www.nvmain.org>
- [33] S. Rixner, W. J. Dally, U. J. Kapasi, P. Mattson, and J. D. Owens, "Memory Access Scheduling," in *ISCA'00*, Jun. 2000, pp. 128–138.
- [34] Standard Performance Evaluation Corporation, "SPEC2006 CPU," <http://www.spec.org/cpu2006>.
- [35] M. B. Healy and S. K. Lim, "Power-Supply-Network Design in 3D Integrated Systems," in *ISQED'11*, Mar. 2011, pp. 223–228.
- [36] P. Jain, D. Jiao, X. Wang, and C. H. Kim, "Measurement, Analysis and Improvement of Supply Noise in 3D ICs," in *VLSI Circuits Digest*, Jun. 2011, pp. 46–47.
- [37] Fujitsu, "Memory Consumer FCRAM 512M Bit MB81EDS516445," 2010.
- [38] Micron, "MT44K32M18RB-093 rldram3," <http://www.micron.com/products/dram>.
- [39] H. Zheng, J. Lin, Z. Zhang, E. Gorbato, H. David, and Z. Zhu, "Minirank: Adaptive DRAM Architecture for Improving Memory Power Efficiency," in *MICRO'08*, Nov. 2008, pp. 210–221.
- [40] D. H. Yoon, M. K. Jeong, and M. Erez, "Adaptive Granularity Memory Systems: A Tradeoff between Storage Efficiency and Throughput," in *ISCA'11*, Jun. 2011, pp. 295–306.
- [41] B. Jacob, *The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It*. Morgan & Claypool Publishers, 2009.
- [42] D. H. Woo, N. H. Seong, D. L. Lewis, and H.-H. S. Lee, "An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth," in *HPCA'10*, Jan. 2010.
- [43] D. H. Woo, N. H. Seong, and H.-H. S. Lee, "Pragmatic Integration of an SRAM Row Cache in Heterogeneous 3-D DRAM Architecture Using TSV," *TVLSI*, pp. 1–13, Dec. 2011.
- [44] Q. Zhu, X. Li, and Y. Wu, "Thermal Management of High Power Memory Module for Server Platforms," in *ITHERM'08*, May 2008, pp. 572–576.