An Efficient DRAM with Reduced Energy Consumption in Video Driver

Kasif Khan
Department Of EC
TIT Collage
Bhopal

Ankur Beohar
Department Of EC
TIT collage
Bhopal

Manish Kumar Gurjar
Department Of EC
TIT Collage
Bhopal

ABSTRACT

Power reduction in electronic and computing system is one of the basic requirements and is increasingly demanded for the battery operated mobile systems. Although the power is required for every part of the system but the devices accessed most frequently (processor, DRAM) are takes special attention, because the improvement in power dissipation in these devices can dramatically reduce the overall power requirement. Since the many approaches have been already proposed for the power reduction in processor this paper focuses on the power reduction in DRAM. The DRAM may be considered as most power consuming device after processor, even when it is idle. Although the DRAMs inherently supports different power saving modes, like self-refresh and power-down, but these techniques are not as efficient and also causes the unwanted delay which in non-comprisable for the many multimedia applications. Hence in this paper, we propose and evaluate an efficient DRAM rank grouping and power gating technique for power-saving that optimizes the power saving with marginal performance degradation. The proposed approach is developed and tested on several multimedia operations and the experimental results show that it reduces the total DRAM energy consumption between 56% and 183%(approx)at a negligible performance penalty between 3% and 5%(approx).

Keywords:

Group-based Power Saving, Power Gating, and DRAM-Memory

1. INTRODUCTION

Energy has become a non-avoidable design requirement in almost every computer systems, specifically in battery powered mobile devices. In any battery power device, energy saving can increase battery life and makes it more applicable during long travelling.

Recent growth in semiconductor technology made possible to design compact high power processors which turns the mobile devices like phones into general purpose computing platforms, with their own operating system, often called smartphones. But unlike the traditional computers they typically used in short-bursts over extended periods of time, alternatively can be said that they remains idle at most of the time, because of requirement of such devices to wake up and Restore their last accessed state immediately; they are required to remain in responsive state even when it is not being used for a long time. This requires power because Dynamic Random Access Memories (DRAMs) gradually leaks the stored charge and required to be refreshed periodically prevent loss of data. Since the DRAM consumes a large percentage of the total power required by any computing device, the designers are developing the new ways to manage the power characteristics of DRAM. The power efficiency has been achieved by many ways from low level semiconductor physics, power-gating to software level resource control. The semiconductor level development is very costly and already reaching to their limitation which makes the improvements in this level much difficult. Hence the other techniques are utilized to improve the efficiency of DRAM. The commercially available DRAM, have integrated mechanism for power reducing which can be used to put idle memory into a low power mode. Even with the availability of these facilities it is difficult to decide the applicability of it in the dynamic and multitasking environment which leads increased latency. This paper presents an efficient group (rank) based controlling technique which reduces the power at acceptable latency. The rest of this paper is organized as follows. Section II provides the related work to the study. In Section III, DRAM is discussed followed by the some controlling algorithm in section IV. The next section presents the proposed method Section VI shows the simulation results in various different scenarios. Finally, Section VII presents our conclusions and further research directions.

2. LITERATURE REVIEW

The literature on the DRAM discharging characteristics for different refreshing rate and corresponding wakeup latency with experimental results is presented by Krishna T. Malladi et al [9] they also presented an approach to turn on the DRAM for less than recommended while increasing the sampling duration. MingliXie et al [10] proposed optimal page policy selection techniques based on the application characteristics, which optimize DRAM performance or minimize power consumption. As an improvement, the literature also proposed a power aware bank partitioning to balance DRAM performance and power consumption. Jishen Zhao et al [5] propose a hybrid graphics memory architecture which utilizes different memory technologies (DRAM, STT-RAM, and RRAM), to improve the memory bandwidth and reduce the power consumption and for further memory powerreduction an adaptive data migration mechanism that exploits various memory access patterns of GPGPU applications is also addressed. An application-level technique to reduce refresh power in DRAMmemories namedFlikker is proposed by Song Liu et al [4]. The technique facilitates the developers to specify critical and non-critical data in programs which is used by the runtime system to allocate these data in separate parts of memory. The refresh rate of each part of memory is reduced on the basis of critical requirements of the data. This partitioning and refreshing saves energy at the cost of a slightly increased data error in the non-critical data.Flikker thus finds an interesting trade-off between energy consumption and hardware correctness. Token-Based Adaptive Power Gating (TAP) [13], a technique to power gating the cores during memory accesses, TAP works by observing and predicting the system memory request power gate the core without performance or energy loss. A mixed approach for the same goal is presented in [2] the literature describe a simple power down policy for exploiting low power modes of DRAMs, with adaptive history-based memory schedulers and throttling approach that arbitrarily reduces DRAM activity by delaying the issuance of memory commands.

3. DRAM EXECUTION MODEL

A DRAM accessing requires sequential execution of three operations, called pre-charge (charge a DRAM bank before a row access), activation (activate a row (page) of a DRAM bank), and data read/write (column access, select and return a block of data in an activated row). In the close page mode, the common execution sequence of all three operations is activation, data read/write, and then pre-charge for the next access.

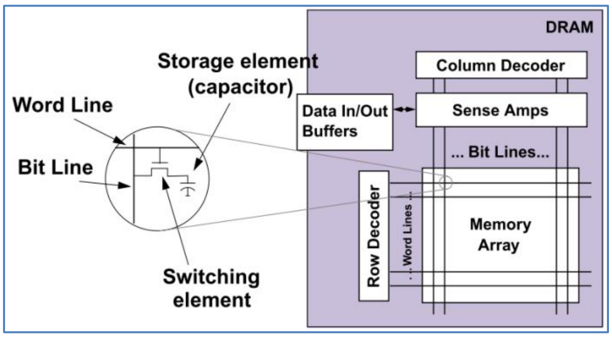

Figure 1: The basic architecture of DRAM.

In the open page mode,the common sequence for a row buffer miss (if the data in not in same page) is pre-charge,activation, and data read/write; and only data read/write isneeded for a row buffer hit.

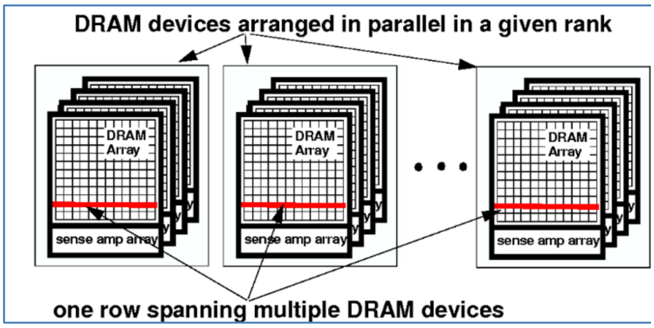
DRAM devices arranged in parallel in a given rank
DRAM
Array
sense amp array
DRAM
Array
sense amp array
DRAM
Array
sense amp array
one row spanning multiple DRAM devices

Figure 2: Representing the relation of BANK (DRAM Array), RANK (Group of Array) and PAGE (the Complete Figure)

The power consumption for the DRAM can be divided into four categories:

1. Background: the power that a DRAM chip consumes all the time with or without operations.
2. Operation: to perform activation and pre-charge.
3. Read/write: data read and write.
4. I/O: driving the data bus and terminating data.

There is only the background power which consumes all the time irrespective of operations, hence to reduce it DRAMs support multiple low power modes.

4. BASIC POWER SAVING MODES OF DRAM

The details of the power modes are discussed below:

Active: In this mode, the DRAM module is ready and can transition immediately to read or write mode. As the memory unit is ready to service any read or write request, the resynchronization time for this mode is the least (zero units), and the energy consumption is the highest.

Standby: In this mode, the column multiplexers are disabled resulting in significant reduction in energy consumption compared to the active mode. The resynchronization time for this mode is typically one or two memory cycles. Some state-of-the-art RDRAM memories already exploit this mode by automatically transitioning into the standby mode at the end of memory transaction.

Napping: The ROW de-mux circuitry is turned off in this mode, leading to further energy savings over the standby mode. When napping, the DRAM module energy consumption is mainly due to the refresh circuitry and clock synchronization that is initiated periodically to synchronize the internal clock signals with the system clock. This mode can typically consume two orders of magnitude less energy than the active mode, with the resynchronization time being higher by an order of magnitude than the standby mode.

Power-Down: This mode shuts off the periodic clock synchronization circuitry resulting in another order of magnitude saving in energy. The resynchronization time is also significantly higher (typically thousands of cycles).

Disabled: If the content of a module is no longer needed, it is possible to completely disable it (saving even refresh energy). There is no energy consumption in this mode, but the data is lost.

5. PROPOSED ALGORITHM

The proposed model can be as a hybrid cluster approach which controls the DRAM operations. The memory controller selects a BANK within the RANK in such a way that it will save thepower to a greater extent while maintain the latency and also maintains the efficient utilization of the DRAM BANKs.

5.1. The proposed algorithm can be explained in the following steps:

1. The memory controller initially forms the Indexes of RANKs, Indexes of each BANK within each RANK, and Indexes of each BANK available within each RANK.

2. Next the memory controller collects the Indexes of current BANKs in use and indexes of RANKs in which currently active BANK is present.

3. Now it generates a reference table to store the status of different BANKs and RANKs for example, let the number of RANK = 4, number of BANKs within selected RANK = 3 and all others remains idle then it will be represented as:

Table 1: RANK Activity Status Monitoring Table

	Bank 1	Bank 2	Bank 3	Bank 4	Busy
Rank 1	0	0	0	0	0
Rank 2	0	0	0	0	0
Rank 3	0	0	0	0	0
Rank 4	0	0	1	0	1

4. One extra column is added in reference table which shows the status of that RANK whether it is gated or not in case of RANKs.

'0' represents RANK is off.

'1' represents RANK is gated.

In case of BANKs,

'0' represents that BANK is free.

'1' represents that BANK already has data.

Whenever any RANK is selected for the 1st time its entry is made in the reference table along with its RANKs and all the BANKs within it, and the table is immediately updated after any action is performed on BANK or RANK level either assigning workload (wakeup) or turning off (power down) it.

Initially all the RANKs are in off state and all the BANKs within them remain gated.

This is done in order to assure that whenever a new cluster is selected that was in off state, only one wakeup signal will be sufficient to wakeup whole RANK for reducing further latency to wake up each BANK.

5.1.1. Procedures for selecting memory

1. Memory controller will search for the free BANKs in the currently active BANKs within the currently pointed active RANK. If free BANK is Available then it will select that BANK.

Else,

2. Memory controller will search the table for the already active RANKs with free BANKs from the scratch. If found then, it will select that BANK.

Else,

3. Scheduler will select next lowest indexed RANK to gate it and the lowest indexed BANK will be selected finally its entry will be made in the table immediately.

Suppose a RANKS entry is present in the table showing that it was previously used but presently it is in power off state ,then that RANK will not be selected (unless and until that is the last free available RANK) , for efficient utilization of all memory blocks.

5.1.2.Gating procedure for RANKs and BANKs

Suppose any BANK is free for time equal to greater than the threshold value, then the memory controller will turned off, that BANK, but controller will look if all the BANKsat sameRANK are also in turned off state then, it will turned off the whole RANK and turn on all the BANKs just after it, otherwise simply turned off that BANK.

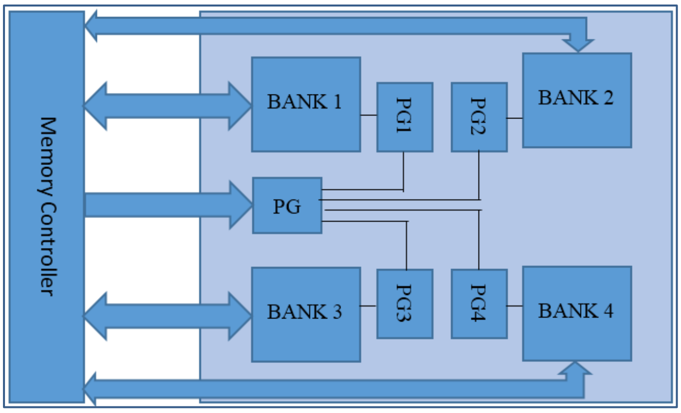

Figure 3: Block Diagram of Memory RANK for the Proposed System.

6. SIMULATION RESULTS

The implementation and simulation of the proposed algorithm is performed using MATLAB. The simulation is performed for random load and multimedia requests. The DRAM is formed by using sub-Array size (16x4) 16 BANKS and 4 RANKS. Finally the simulation results are presented in the form of tables and graphs.

Table 2: Comparison of Wakeup latency for Different Video Files

	Akiyo	Foreman	Hall	Mobile	Random
SBS	0.41	0.41	0.36	0.29	0.38
SBSTO	0.41	0.41	0.36	0.29	0.38
RBS	0.84	0.91	0.88	0.80	0.91
RBSTO	0.90	0.93	0.87	0.86	0.94
Cluster	0.07	0.02	0.10	0.15	0.08
Cluster-Mod.	0.47	0.31	0.41	0.33	0.38

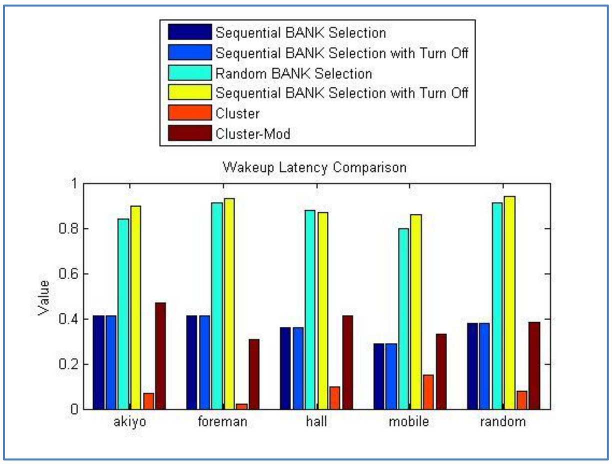

Figure 4: Comparison of Wakeup latency for Different Video Files.

Table 3: Comparison of Leakage Loss for Different Video Files

	Akiyo	Foreman	Hall	Mobile	Random
SBS	10.78	10.52	9.48	7.53	11.78
SBSTO	6.06	5.26	5.33	4.23	4.42
RBS	10.78	10.52	9.48	7.53	11.78
RBSTO	6.06	5.26	5.33	4.24	4.42
Cluster	3.17	3.02	2.90	2.57	3.19
Cluster-Mod.	2.83	2.86	2.62	2.38	2.99

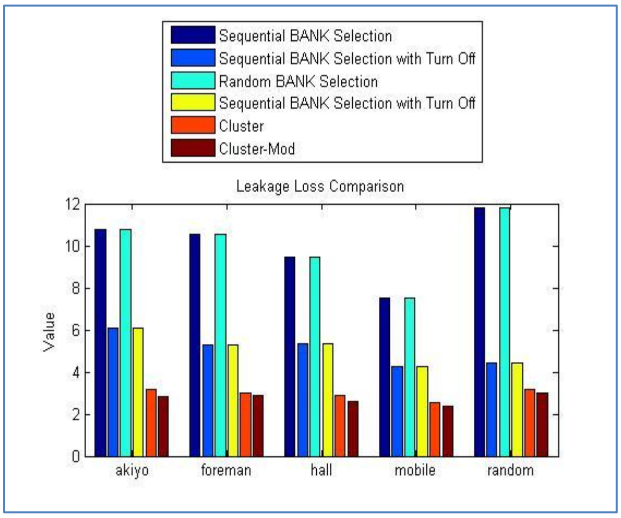

Figure 5: Comparison of Leakage Loss for Different Video Files

Table 4: Comparison of BANKs Utilization for Different Video Files

	Akiyo	Foreman	Hall	Mobile	Random
SBS	5.68	3.63	6.25	7.10	2.60
SBSTO	5.68	3.63	6.25	7.10	2.61
RBS	19.33	18.35	16.68	19.56	17.14
RBSTO	19.71	20.06	17.79	19.51	20.47
Cluster	17.27	15.14	17.61	20.16	14.46
Cluster-Mod.	18.95	16.40	17.10	19.07	16.46

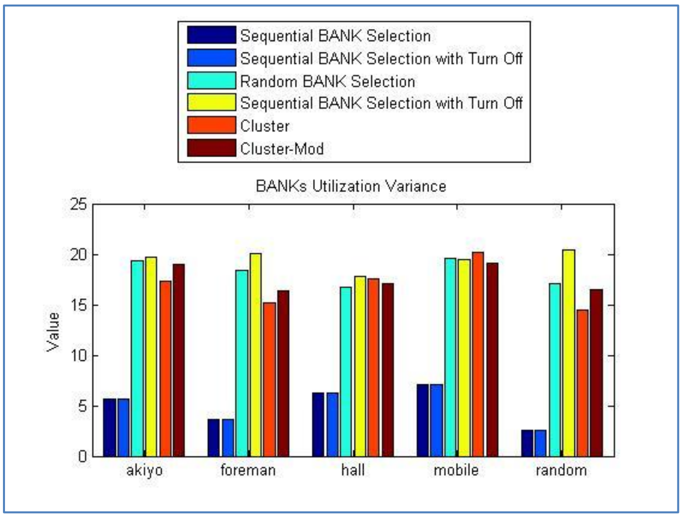

Figure 6: Comparison of BANKs Utilization for Different Video Files.

7. CONCLUSION

This paper presents a new clustered gating approach for the power saving and efficient and uniform utilization of DRAM while minimizing power requirements and without compromising latency. The simulation results show that the proposed technique uniformly utilizes the DRAM blocks from all RANKs (as shown in table 4). It also reduces the wakeup latency and leakage power (as shown in table 2 and 3 respectively) frequency of individual BANKS which reduces the accessing delay. These results validates that the proposed algorithm can provide a better solution for power and latency minimization of DRAM. For the further work we can improve and also investigate these methods by increasing the scalability of DRAM technique.

8. REFERENCES

[1] Yiran Li and Tong Zhang, “Reducing DRAM Image Data Access Energy Consumption in Video Processing”IEEE Transactions on Multimedia, Vol. 14, No. 2, April 2012

[2] Ibrahim Hur and Calvin Lin “A Comprehensive Approach to DRAM Power Management”, High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on 16-20 Feb. 2008.

[3] HongzhongZheng, Jiang Lin, Zhao Zhang, Eugene Gorbatov, Howard David and Zhichun Zhu “Mini-Rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency”, Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on 8-12 Nov. 2008.

[4] Song Liu, KarthikPattabiraman, Thomas Moscibroda and Benjamin G. Zorn “Flikker: Saving DRAM Refresh-power through Critical Data Partitioning”, ASPLOS XVI Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems, 2011-03-05.

[5] Jishen Zhao and Yuan Xie “Optimizing Bandwidth and Power of Graphics Memory with Hybrid Memory Technologies and Adaptive Data Migration”, Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on 5-8 Nov. 2012.

[6] Sparsh Mittal "A Cache Reconfiguration Approach for Saving Leakage and Refresh Energy in Embedded DRAM Caches", arXiv:1309.7082v1 [cs.AR] 26 Sep 2013.

[7] Gervin Thomas, KarthikChandrasekar, Benny Akesson, Ben Juurlink and KeesGoossens "A Predictor-based Power-Saving Policy for DRAM Memories", Digital System Design (DSD), 2012 15th Euromicro Conference on 5-8 Sept. 2012.

[8] Howard David, Chris Fallin, Eugene Gorbatov, Ulf R. Hanebutte, OnurMutlu "Memory Power Management via Dynamic Voltage/Frequency Scaling", ICAC '11 Proceedings of the 8th ACM international conference on Autonomic computing Pages 31-40, 2011-06-14.

[9] Krishna T. Malladi, Ian Shaeffer, LijiGopalakrishnan "Rethinking DRAM Power Modes for Energy Proportionality", http://doi.ieeecomputersociety.org/10.1109/MICRO.2012.21.

[10] MingliXie, Dong Tong, Yi Feng, Kan Huang, Xu Cheng"Page Policy Control with Memory Partitioning for DRAM Performance and Power Efficiency", Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on 4-6 Sept. 2013.

[11] Qingyuan Deng, David Meisner, Luiz Ramos, Thomas F. Wenisch, Ricardo Bianchini "MemScale: Active Low-Power Modes for Main Memory", ASPLOS XVI Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems, 2011-03-05.

[12] Jung Ho Ahn, Norman P. Jouppi, Christos Kozyrakis , Jacob Leverich, Robert S. Schreiber "Improving System Energy Efficiency with Memory Rank Subsetting", ACM Transactions on Architecture and Code Optimization (TACO) TACO Homepage archive Volume 9 Issue 1, March 2012.

[13] Andrew B.Kahng, Seokheong Kang, TajanaRosing and Richard Strong "TAP Token-Based Adaptive Power Gating" ISLPED'12 Proceeding of the 2012 ACM/IEEE International Symposium of Low Power Electronics and Design

[14] Andrew B.Kahng, Seokheong Kang, TajanaRosing and Richard Strong "Many-Core Token-Based Adaptive Power Gating"IEEE Transactions On Computer-Aided Design Of Integrated Circuits and Systems, Vol.32,Issue: 8 , month Aug, 2013.