An Energy Efficient DRAM System For GPUs

Kangni Liu
Electric and Computer Engineering department
University of Pittsburgh
Pittsburgh, Uniter States
connie.liu@pitt.edu

***Abstract*—DRAM row buffer is a limited resource to concurrent threads, which lead to energy overheads when processing. An energy-efficient DRAM was proposed to reduce minimum row activation granularity. To realize it, the paper designed a structure that divides DRAM datapath into several subchannels. To lower the toggling energy caused by subchannels, a static data recording scheme was used. This newly designed reduced energy consumption by 35% and increased area of about 2.6%. Overall, the system performance improved by 13% with semi-independent subchannels.**

***Index Terms*—subchannel, row activation, DRAM**

I. Introduction

Higher compute capability and wider bandwidth are the development trend of Graphics Processing Units. It could be expected that the future system will require 1 TB/s of bandwidth. To satisfy the increasing bandwidth, the energy consumption of DRAM should also be increased greatly. However, the energy consumption of traditional bandwidth-optimized GDDR5 memories will be unbearable when bandwidth reaches 1 TB/s. To release this condition, on-package stacked DRAM was applied. It can reduce the cost of data transfer on the interface between the DRAM stack and the processor die. However, the energy used to move data from the DRAM bit cells to the I/O interface does not change significantly. Reducing the energy of DRAM itself is still a key energy of the future high bandwidth system.

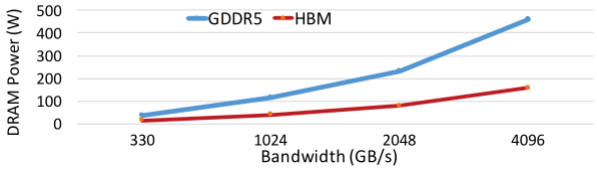

Fig. 1. GPU memory system energy vs bandwidth [1]

II. DRAM access Energy analysis

The energy of DRAM access is mainly two parts, that is, row energy and column energy. The functions of row energy including activate a DRAM row, sense the data, latch the data to row buffer and precharge the bit lines. So the row energy is determined by various factors. The function of column energy is to move the data from activated row to I/O pins. So it only depends on the distance of the data is moved, the switching rate and capacitance of wires.

It is shown in Figure 2 that in different conditions, column energy has hardly changed while row energy could vary widely. In low row locality conditions, the row energy is much higher than column energy. There are two main reasons for high row energy.

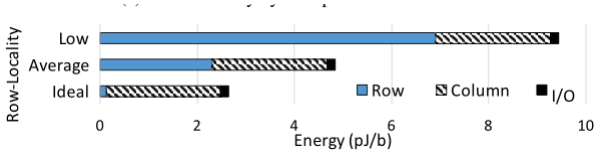

Fig. 2. memory energy in different locality [1]

First is that several GPU applications have low locality. Applications, which have dependent memory access, have low relation with successive accesses, and this is unavoidable.

Second is that the interference between the quantities of simultaneously executing threads leads to the limited row-buffer locality. Frequently bank conflict increases energy consumption.

The paper proposed a new DRAM structure to address this problem, called subchannels. Subchannels reduced the energy of row activate. This new design could reduce the energy consumption in DRAM by 35% and the total overhead is about 2.6%. On average, the system performance is improved by 13%.

III. Structure of High bandwidth DRAM

Subchannels were designed based on high bandwidth DRAM. Before we explain the working principle of subchannels, a brief introduction of high bandwidth DRAM is necessary.

A bank is a storage array and each bank has its row and column decoders. A collection of banks is called a DRAM die. Banks are connected through the bus, which sends the bank signals to I/O buffer. The I/O buffer is the same size as a DRAM atom. For example, in DDR5, it is 32B. It connected to the external I/O interface.

Back to the bank. Each bank is a 2D array of mats. The structure of the mat is shown as Fig. 4. It consists of 512×512 DRAM cells. Row-buffer, a 512-bit sense-amplifier in Fig .4, is used to amplify the signals from its bank. 32 mats form horizontally, called a subarray. In general, if a subarray was activated, all the banks in this subarray will be activated at the

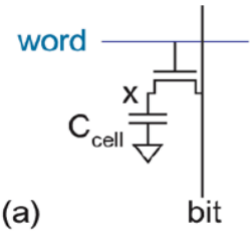

Fig. 3. Structure of DRAM cell [2]

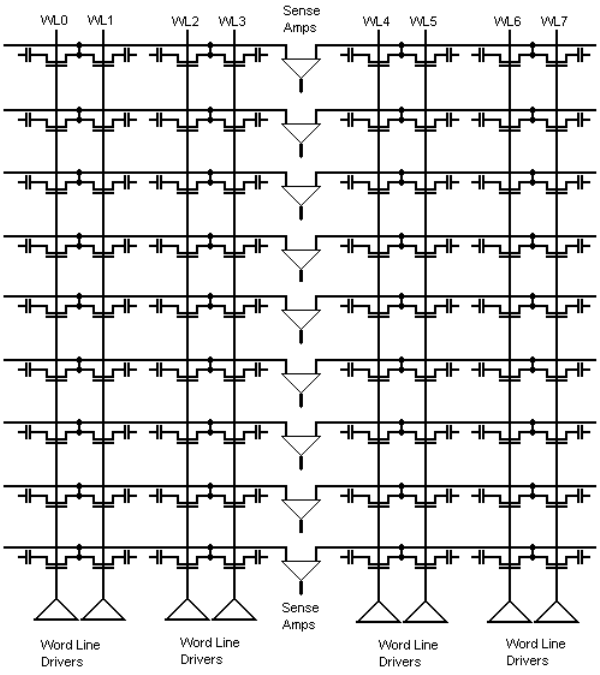

Fig. 4. An example of DRAM array [3]

same time. So a DRAM atom is spread out over the subarray, and each bank contributes to 8 bits.

Master World Line(MWL) drives the Local World Line(LWL) in each mat. Once the local world line works, it turns on the transistor of a row in the mat, which activates the row. After the row is activated, the bits are read into a row buffer. Row-buffer activates the column select line(CSL) of the mat, which sends the data to the sense amplifier. Also, the Master Data Lines(MDL) are connected to the CSL through Local Data Lines(LDL). A helper flip-flop is set between MDL and LDL to drive MDL down to the Global Sense Amplifier(GSA).

IV. Structure of Energy-Efficient DRAM and its work principle

The energy-efficient DRAM reduced the energy of row activation. It separates subarray into several groups and maps the DRAM atom. This new structure is called subchannels. The following section will introduce its detail and its working principle.

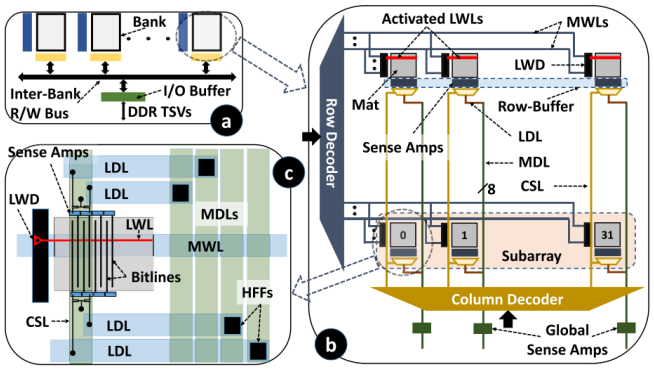

Fig. 5. High bandwidth DRAM [1]

A. Structure of Subchannels

In DRAM, MWL drives the LWL of all mats in a subarray by activating the local wordline driver(LWD). To separate the subarray, additional LWD is necessary, and segment select lines are needed to distinguish the target mat. An example structure of separated subarray is shown as Fig. 6.

As shown in Fig.6, Mat 1 and Mat 2 are separated by adding one more LWD between them. Mat 0 and Mat 1 form subchannel 1; Mat 2 and Mat 3 form subchannel 2. These two subchannels can be driven independently. If 32 mats are divided into 8 groups, it is obvious that 7 additional LWDs are needed. The extra LWD will be inserted at the boundary of each group.

Moreover, to decide which subchannel is to activate, we add Segment Select(SS) signals, which are the green lines in Fig.6. If the SS[0] in Fig.6 is asserted, Mat 0 and Mat 1 will be turned on, allowing MWL to drive their LWL. Similar to subchannels, the SS signals are also independent with each other.

The overhead area of these two additional components is estimated to be 2.6%. Compare to the enhanced performance, the negative effect of the additional area could be ignored in this case.

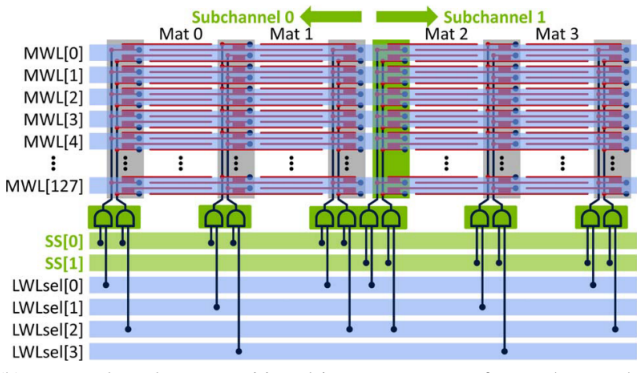

Fig. 6. separated subarray [1]

B. Pipeline Burst of Subchannels

Subchannels could be activated independently, which means a DRAM atom can be retrieved only from activated mats. Reading the DRAM atom from $1/8^{th}$ of the mats in one cycle requires increasing output width. However, this may lead to more wiring tracks of MWL, LDL, CSL and MDL. If these overhead areas count, the total mat area will be increased by 34.5% instead of 2.6%.

Instead of increasing wiring tracks, the paper keeps the mat bandwidth unchanged and transfer each DRAM atom over eight DRAM cycles. The details are shown bellowing.

In the HBM architecture, the whole DRAM atom is sent to I/O buffer in one internal cycle. With eight subchannel in each subarray, it will take 8 internal cycles to fill the I/O buffer with a DRAM atom. Instead of waiting the I/O buffer be filled up, the new DRAM pipeline the burst, both internal and external. It has to be noticed that the internal DRAM cycle needs to be short enough to allow back-to-back column accesses to each element of the burst. tCCDL, the CAS-to-CAS delay within the same bank, is the definition of this internal cycle. DDR5 already has a tCCDL of 2 ns, which is suitable for this new design.

C. Address Decoupling Register

Row-address latch and column-address latch are added to avoid address decoupling. Taking the row-address latch for example. When the row address is driven to the row-address latch, the latch could drive the MWL. After that, the row decoder is free, decoding and driving the next row address to the latch in the next cycle. The column-address latch works in the same way.

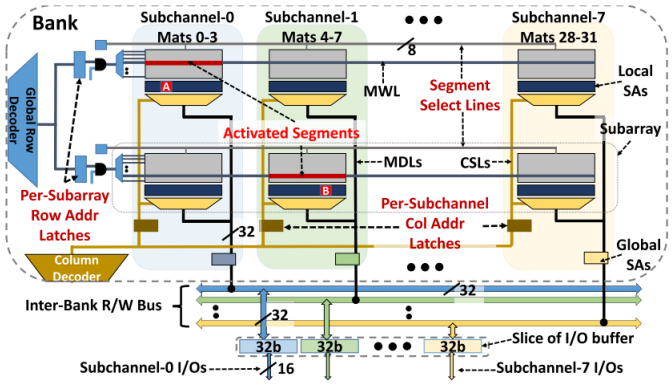

Fig. 7. subchannels with address decoupling design [1]

D. Command Coalescing

All subchannels in one subarray share the address and command bus, and the same bank in different subchannels share the row and column decoders. Focusing on this, the paper proposed an optimized control method called command coalescing.

The application may require activating the same row in different subchannels. Usually, it will require more address and command bandwidth than the HBM system. To solve this problem, the paper suggests that the memory controller could coalesce several activate operation. First, the memory controller scans the command queues and find out the activate command of the same row in different subchannels. It releases a single activate command for all these requirements. The activate command will be sent with the appropriate set. So when the locality is high, this new DRAM shows better performance than the HBM baseline.

Not only the activate command will be coalescing, but also the read command and write command. The principle is similar. If the read or write command is issued to the same bank and column in multiple subchannels, the memory controller will coalesce the read or write command. The output of the column-decoder and subchannel mask are used together, making real progress in parallel across subchannels.

E. Toggling Energy Reduction

Usually, a 32B DRAM atom consists of eight 32-bit value or four 64-bit value. Fig.8 shows how a 256-bit DRAM atom is transferred in the different datapath. In the baseline memory, the DRAM atom moves across a 256-bit internal bus. When it comes to DDR I/O, the datapath becomes 128-bit, and the order of the data is shown in Fig. 8a. In 8 subchannels, the 256-bits wide datapath is separated into 8 pieces of 32-bits, which is shown in Fig. 8b. When it comes to the I/O bus, the datapath becomes narrow, which is 16-bits. Fig. 8c shows how the data order in the I/O bus. It is obvious that the frequency should double. Because uncorrelated portions of a 64-bit value are transmitted back-to-back, the internal switching will arise. In the I/O bus, this effect will increase.

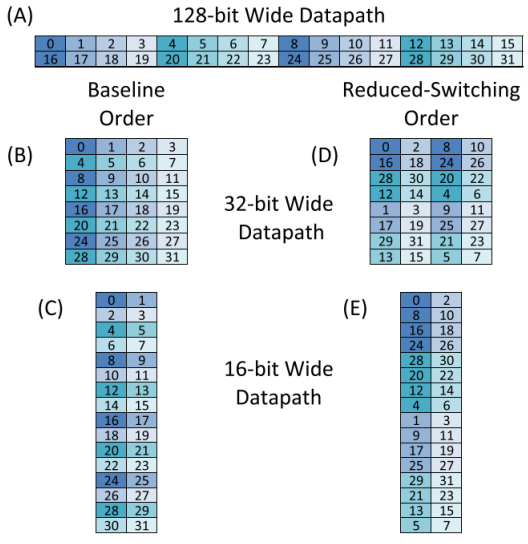

Fig. 8. Data ordering in different datapath [1]

To reduce the switching activity, the paper remaps the data within a burst, making the high correlated bits transferred on

the same wires in successive cycles. The new order is shown in Fig. 8d and Fig .8e. In successive cycles, use 8-bit offsets while the remaining bytes are transferred in 4-bits offset. When the data is stored in DRAM. it could be statistically applied to every DRAM atom without additional hardware. During the storage, the data is loaded into the buffer of the memory controller in the reorder shown in Fig .8e. When read, the data returned from the DRAM and reordered into the original one. This new simpler static scheme achieved 98% of the benefit and is much easier to be applied in GPU.

V. Results

Using command subchannels and command coalescing, the energy consumption of DRAM is decreased greatly. In this section, DRAM energy consumption under different conditions will be compared.

A. DRAM Energy Reduction

Figure 9 shows the DRAM energy consumption under three conditions. Original baseline, 8 subchannels without reordering(SC-8_No_Reordering), 8 subchannels with reordering(SC-8). The energy of the DRAM is divided into 3 parts. Row energy, column energy and I/O energy. It can be obtained that the row activation energy in SC-8 is 74% lower than Baseline on average. It is more obvious in low locality applications, such as GPUs, bh, sp, MCB.

It has to be pointed out that because of the narrow subchannel design, the energy of the column and I/O increased. Using burst reordering could low down this energy, but it still has an effect. In conclusion, the total DRAM energy with subchannels and burst reordering is 35% lower than the Baseline.

B. System Performance Enhance

Fig. 10 shows the performance under four conditions, Baseline, subchannels without parallelism(SC-8_No_Parallelism), subchannels with parallelism but no commend coalescing(SC-8_No_Coalescing) and subchannels with both(SC-8).

Overall, the performance of bandwidth-intensive benchmarks (bfs (66%), kmeans (35%), stream- cluster (9.1%), sp (39%),lulesh (10%), MiniAMR (35%), GUPS (152%)) improved a lot with parallelism and command coalescing. On average, the whole system performance improved about 13%.

References

[1] N. Chatterjee, M. O. Connor, D. Lee, D. R. Johnson, S. W. Keckler, M. Rhu, and W. J. Dally, "Architecting an Energy-Efficient DRAM System for GPUs," 2017.

[2] X. Feng, "Introduction to memory ii."

[3] T. Schwarz, "Dram overview." [Online]. Available: http://www.cse.scu.edu/ tschwarz/coen180/LN/DRAM.html

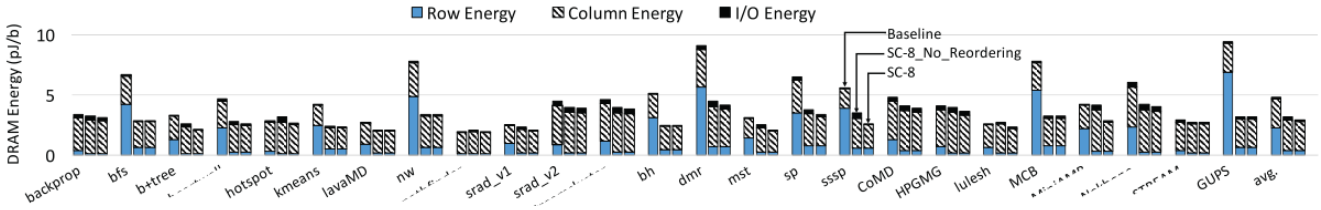

Fig. 9. Energy consumption of DRAM under different conditions [1]

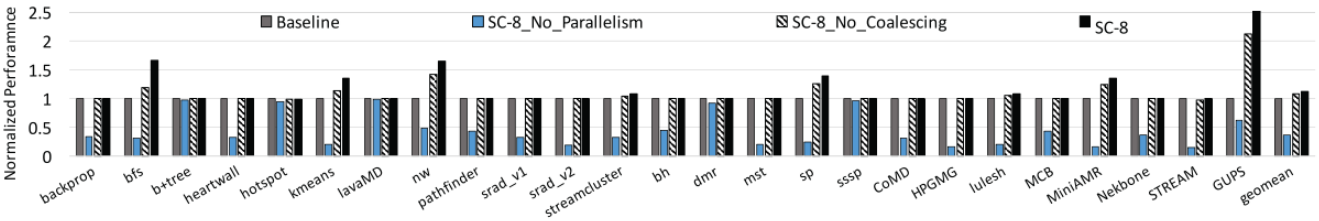

Fig. 10. Performance compare under different conditions(Higher the Better) [1]