

CMU 18-447 INTRODUCTION TO COMPUTER ARCHITECTURE, SPRING 2015
HW 5: VIRTUAL MEMORY AND MAIN MEMORY

Instructor: Prof. Onur Mutlu

TAs: Kevin Chang, Rachata Ausavarungrun, Albert Cho, Jeremie Kim, Clement Loh

Assigned: Mon., 3/23, 2015

Due: **Sun., 03/29, 2015 (Midnight)**

Handin: **Autolab**

1 Magic-RAM

Assume you developed the next greatest memory technology, MagicRAM. A MagicRAM cell is non-volatile. The access latency of a MagicRAM cell is 2 times that of an SRAM cell but the same as that of a DRAM cell. The read/write energy of MagicRAM is similar to the read/write energy of DRAM. The cost of MagicRAM is similar to that of DRAM. MagicRAM has higher density than DRAM. MagicRAM has one shortcoming, however: a MagicRAM cell stops functioning after 2000 writes are performed to the cell.

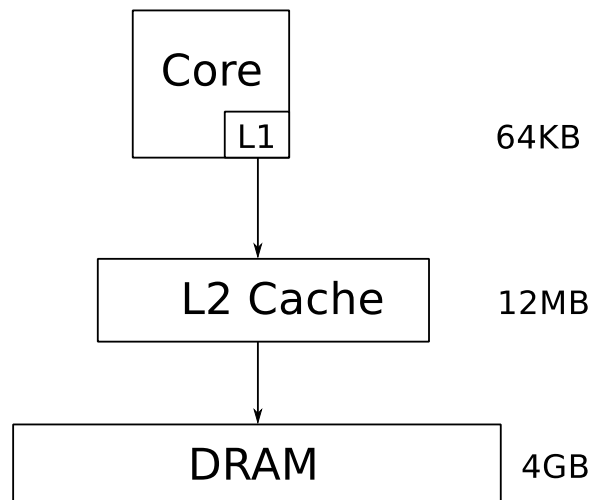
- (a) Is there an advantage of MagicRAM over DRAM other than its density? (Please do not repeat what is stated in the above paragraph.) Circle one: **YES** **NO**

Explain.

- (b) Is there an advantage of MagicRAM over SRAM? Circle one: **YES** **NO**

Explain.

- (c) Assume you have a system that has a 64KB L1 cache made of SRAM, a 12MB L2 cache made of SRAM, and 4GB main memory made of DRAM.



Assume you have complete design freedom and add structures to overcome the shortcoming of MagicRAM. You will be able to propose a way to reduce/overcome the shortcoming of MagicRAM (note that you can design the hierarchy in any way you like, but cannot change MagicRAM itself).

Does it makes sense to add MagicRAM anywhere in this memory hierarchy given that you can potentially reduce its shortcoming? Circle one: **YES** **NO**

If so, where would you place MagicRAM? **Depict in the figure above clearly** and describe why you made this choice.

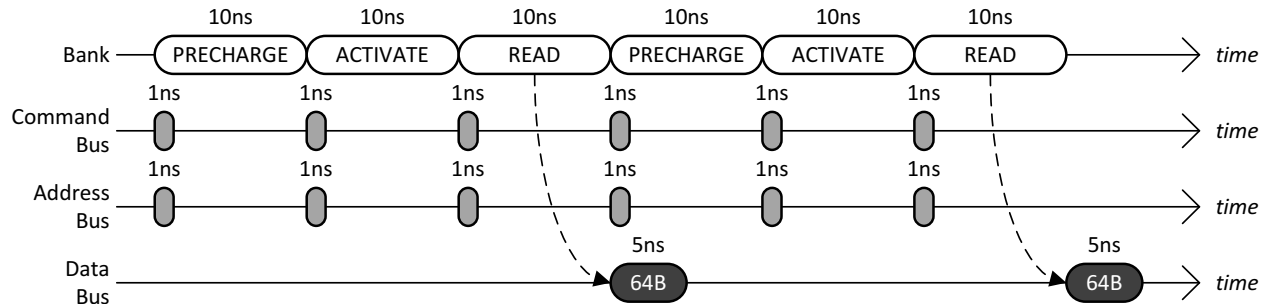
If not, why not? Explain below clearly and methodically.

- (d) Propose a way to reduce/overcome the shortcoming of MagicRAM by modifying the given memory hierarchy. Be clear in your explanations and illustrate with drawings to aid understanding.

Explanation:

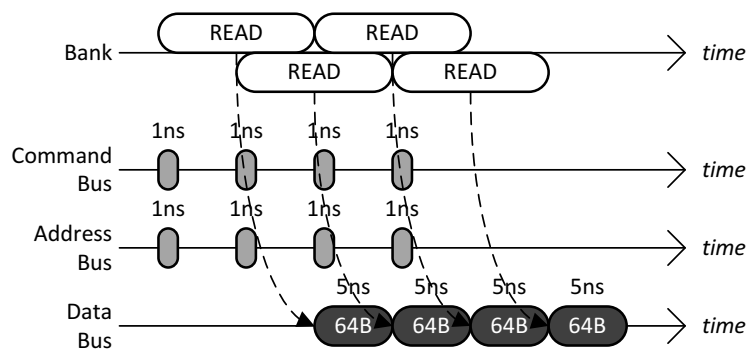
2 Memory Scheduling

Row-Buffer Conflicts. The following timing diagram shows the operation of a single DRAM channel and a single DRAM bank for two back-to-back reads that conflict in the row-buffer. Immediately after the bank has been busy for 10ns with a READ, data starts to be transferred over the data bus for 5ns.



- Given a long sequence of back-to-back reads that always conflict in the row-buffer, what is the data throughput of the main memory system? Please state your answer in **gigabytes/second**.
- To increase the data throughput, the main memory designer is considering adding more DRAM banks to the single DRAM channel. Given a long sequence of back-to-back reads to all banks that always conflict in the row-buffers, what is the minimum number of banks that is required to achieve the maximum data throughput of the main memory system?

Row-Buffer Hits. The following timing diagram shows the operation of the single DRAM channel and the single DRAM bank for four back-to-back reads that hit in the row-buffer. It is important to note that row-buffer hits to the same DRAM bank are pipelined: while each READ keeps the DRAM bank busy for 10ns, up to at most **half** of this latency (5ns) can be overlapped with another read that hits in the row-buffer. (Note that this is different from Lab 6 where we unrealistically assumed that row-buffer hits are non-pipelined.)

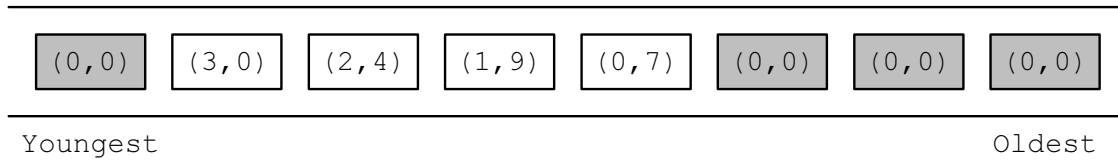


- Given a long sequence of back-to-back reads that always hits in the row-buffer, what is the data throughput of the main memory system? Please state your answer in **gigabytes/second**.
- When the maximum data throughput is achieved for a main memory system that has a single DRAM channel and a single DRAM bank, what is the bottleneck that prevents the data throughput from becoming even larger? **Circle** all that apply.

BANK COMMAND BUS ADDRESS BUS DATA BUS

Memory Scheduling Policies. The diagram below shows the memory controller's *request queue* at time 0. The shaded rectangles are read requests generated by thread T_0 , whereas the unshaded rectangles are read requests generated by thread T_1 . Within each rectangle, there is a pair of numbers that denotes the request's (*BankAddress, RowAddress*). Assume that the memory system has a **single** DRAM channel and **four** DRAM banks. Further assume the following.

- All the row-buffers are **closed** at time 0.
- Both threads start to stall at time 0 because of memory.
- A thread continues to stall until it receives the data for all of its requests.
- Neither thread generates more requests.



For extra credits (15 points), please make sure that you model contention in the banks as well as in all of the buses (address/command/data).

- For the FCFS scheduling policy, calculate the memory stall time of T_0 and T_1 .
- For the FR-FCFS scheduling policy, calculate the memory stall time of T_0 and T_1 .
- For the PAR-BS scheduling policy, calculate the memory stall time of T_0 and T_1 . Assume that all eight requests are included in the same *batch*.

3 Cache and Virtual Memory

A four-way set-associative writeback cache has a 2^{11} · 89-bit tag store. The cache uses a custom replacement policy that requires 9 bits per set. The cache block size is 64 bytes. The cache is virtually-indexed and physically-tagged. Data from a given physical address can be present in up to eight different sets in the cache. The system uses hierarchical page tables with two levels. Each level of the page table contains 1024 entries. A page table may be larger or smaller than one page. The TLB contains 64 entries.

- (a) How many bits of the virtual address are used to choose a set in the cache?
- (b) What is the size of the cache data store?
- (c) How many bits in the Physical Frame Number must overlap with the set index bits in the virtual address?
- (d) On the following blank figure representing a virtual address, draw in bitfields and label bit positions for “cache block offset” and “set number.” Be complete, showing the beginning and ending bits of each field.

Virtual Address:

- (e) On the following blank figure representing a physical address, draw in bitfields and label bit positions for “physical frame number” and “page offset.” Be complete, showing the beginning and ending bits of each field.

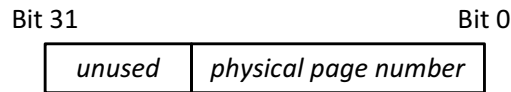
Physical Address:

- (f) What is the page size?
- (g) What is the size of the virtual address space?
- (h) What is the size of the physical address space?

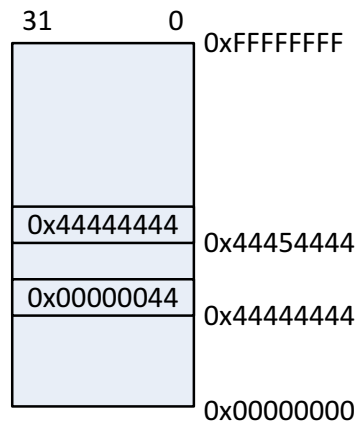
4 0x44444444

A 32-bit processor implements paging-based virtual memory using a single-level page table. The following are the assumptions about the processor's virtual memory.

- The number of bytes in a page is greater than four and is also a power of two.
- The base address of the page table is page-aligned.
- A page table entry (PTE) stores **only** the physical page number and has the following format. All of the unused bits in the PTE are set to 0.



The following figure shows the physical memory of the processor at a particular point in time.

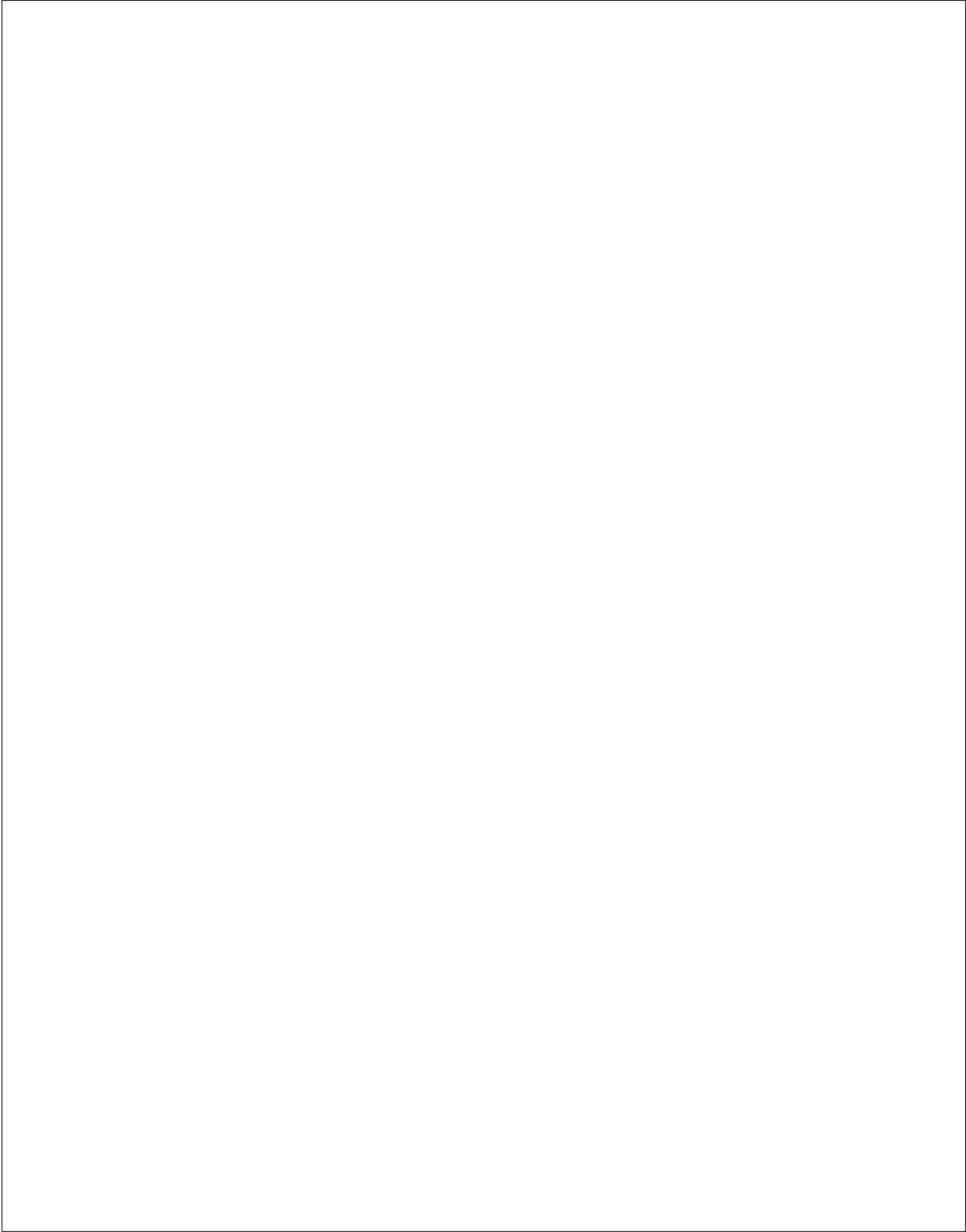


4GB Physical Memory

At this point, when the processor executes the following piece of code, it turns out that the processor accesses the page table entry residing at the physical address of 0x44444444.

```
char *ptr = 0x44444444;  
char val = *ptr; // val == 0x44
```

What is the page size of the processor? Please show work for partial credit. (The answer box is continued on the next page.)



5 Research Paper Summaries

Please read the following handout on how to write critical reviews. We will give out extra credit that is worth 0.5% of your total grade for each good summary/review.

- Lecture slides on guidelines for reviewing papers.
<http://www.ece.cmu.edu/~ece447/s15/lib/exe/fetch.php?media=onur-447-s15-how-to-do-the-paper-reviews.pdf>
- (a) Write a half-page summary for the following paper:
Onur Mutlu, Justin Meza, and Lavanya Subramanian, "The Main Memory System: Challenges and Opportunities" Invited Article in Communications of the Korean Institute of Information Scientists and Engineers (KIISE), 2015.
http://users.ece.cmu.edu/~omutlu/pub/main-memory-system_kiise15.pdf
- (b) **Extra credit (0.5% of your grade)**. Write a half-page summary for the following paper:
Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry, "Row-Clone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization" Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013.
http://users.ece.cmu.edu/~omutlu/pub/rowclone_micro13.pdf