# Computer Architecture (263-2210-00L), Fall 2018
# HW 1: Fundamentals, Memory Hierarchy, Caches

Instructor: Prof. Onur Mutlu
TAs: Mohammed Alser, Can Firtina, Hasan Hassan, Jeremie Kim, Juan Gómez Luna, Geraldo Francisco de Oliveira, Minesh Patel, Giray Yaglikci

Assigned: Wednesday, Sep 26, 2018
Due: **Wednesday, Oct 10, 2018**

- **Handin - Critical Paper Reviews (1).** You need to submit your reviews to `https://safari.ethz.ch/review/architecture18/`. Please, check your inbox, you should have received an email with the password you should use to login. If you didn't receive any email, contact comparch@lists.ethz.ch. In the first page after login, you should click in "Architecture - Fall 2018 Home", and then go to "any submitted paper" to see the list of papers.
- **Handin - Questions (2-8).** You should upload your answers to the Moodle Platform (`https://moodle-app2.let.ethz.ch/mod/assign/view.php?id=274671`) as a single PDF file.

## 1 Critical Paper Reviews [300 points]

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 review.

- Lecture slides on guidelines for reviewing papers. Please, follow this format. `https://safari.ethz.ch/architecture/fall2018/lib/exe/fetch.php?media=onur-comparch-f18-how-to-do-the-paper-reviews.pdf`
- Some sample reviews can be found here: `https://safari.ethz.ch/architecture/fall2018/doku.php?id=readings`

(a) Write a one-page critical review for each of the following papers:

- Moscibroda and Mutlu, "Memory Performance Attacks: Denial of Memory Service in Multi-Core Systems," in Proceedings of the USENIX Security, 2007. `https://people.inf.ethz.ch/omutlu/pub/mph_usenix_security07.pdf`
- Liu et al., "RAIDR: Retention-Aware Intelligent DRAM Refresh," in Proceedings of the International Symposium on Computer Architecture, 2012. `https://people.inf.ethz.ch/omutlu/pub/raidr-dram-refresh_isca12.pdf`
- Kim et al., "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors", in Proceedings of the International Symposium on Computer Architecture, 2014. `https://people.inf.ethz.ch/omutlu/pub/dram-row-hammer_isca14.pdf`

(b) (Optional) Write a one-page critical review for the following paper:

- Patt, "Requirements, Bottlenecks, and Good Fortune: Agents for Microprocessor Evolution," in Proceedings of the IEEE, 2001. `https://safari.ethz.ch/architecture/fall2018/lib/exe/fetch.php?media=patt_ieee2001.pdf`

## 2 DRAM Refresh [150 points]

A memory system has four channels, and each channel has two ranks of DRAM chips. Each memory channel is controlled by a separate memory controller. Each rank of DRAM contains eight banks. A bank contains 32K rows. Each row in one bank is 8KB. The minimum retention time among all DRAM rows in the system is 64 ms. In order to ensure that no data is lost, every DRAM row is refreshed once per 64 ms. Every DRAM row refresh is initiated by a command from the memory controller which occupies the command bus on the associated memory channel for 5 ns and the associated bank for 40 ns. Let us consider a 1.024 second span of time.

We define *utilization* (of a resource such as a bus or a memory bank) as the fraction of total time for which a resource is occupied by a refresh command.

For each calculation in this section, you may leave your answer in *simplified* form in terms of powers of 2 and powers of 10.

(a) How many refreshes are performed by the memory controllers during the 1.024 second period in total across all four memory channels?

(b) What command bus utilization, across all memory channels, is directly caused by DRAM refreshes?

(c) What data bus utilization, across all memory channels, is directly caused by DRAM refreshes?

(d) What bank utilization (on average across all banks) is directly caused by DRAM refreshes?

(e) The system designer wishes to reduce the overhead of DRAM refreshes in order to improve system performance and reduce the energy spent in DRAM. A key observation is that not all rows in the DRAM chips need to be refreshed every 64 ms. In fact, rows need to be refreshed only at the following intervals in this particular system:

| Required Refresh Rate | Number of Rows |
|---|---|
| 64 ms | $2^5$ |
| 128 ms | $2^9$ |
| 256 ms | all other rows |

Given this distribution, if all rows are refreshed only as frequently as required to maintain their data, how many refreshes are performed by the memory controllers during the 1.024 second period in total across all four memory channels?

What command bus utilization (as a fraction of total time) is caused by DRAM refreshes in this case?

(f) What DRAM data bus utilization is caused by DRAM refreshes in this case?

(g) What bank utilization (on average across all banks) is caused by DRAM refreshes in this case?

(h) The system designer wants to achieve this reduction in refresh overhead by refreshing rows less frequently when they need less frequent refreshes. In order to implement this improvement, the system needs to track every row's required refresh rate. What is the minimum number of bits of storage required to track this information?

(i) Assume that the system designer implements an approximate mechanism to reduce refresh rate using Bloom filters, as we discussed in class. One Bloom filter is used to represent the set of all rows which require a 64 ms refresh rate, and another Bloom filter is used to track rows which require a 128 ms refresh rate. The system designer modifies the memory controller's refresh logic so that on every potential refresh of a row (every 64 ms), it probes both Bloom filters. If either of the Bloom filter probes results in a "hit" for the row address, and if the row has not been refreshed in the most recent length of time for the refresh rate associated with that Bloom filter, then the row is refreshed. (If a row address hits in both Bloom filters, the more frequent refresh rate wins.) Any row that does not hit in either Bloom filter is refreshed at the default rate of once per 256 ms.

The false-positive rates for the two Bloom filters are as follows:

| Refresh Rate Bin | False Positive Rate |
|---|---|
| 64 ms | $2^{-20}$ |
| 128 ms | $2^{-8}$ |

The distribution of required row refresh rates specified in part (e) still applies.

How many refreshes are performed by the memory controllers during the 1.024 second period in total across all four memory channels?

What command bus utilization results from this refresh scheme?

What data bus utilization results from this refresh scheme?

What bank utilization (on average across all banks) results from this refresh scheme?

## 3 Data Flow Programs [100 points]

The Fibonacci number $F_n$ is recursively defined as

$$F(n) = F(n-1) + F(n-2),$$

where $F(0) = 1$ and $F(1) = 1$. So, $F(2) = F(1) + F(0) = 1 + 1 = 2$, and so on. The Fibonacci number can be computed as $F(\texttt{n})$:

```
int fib(int n)
{
  int a = 0;
  int b = 1;
  int c = a + b;
  while (n > 0) {
    c = a + b;
    a = b;
    b = c;
    n--;
  }
  return c;
}
```

Draw the data flow graph for the `fib(n)` function. You may use the following data flow nodes in your graph:

- + (addition)
- > (left operand is greater than right operand)
- `Copy` (copy the value on the input to both outputs)
- `BR` (branch, with the semantics discussed in class, label the True and False outputs)

You can use constant inputs (e.g., 1) that feed into the nodes. Clearly label all the nodes, program inputs, and program outputs. Try to the use fewest number of data flow nodes possible.

## 4 Caches [100 points]

A byte-addressable system with 16-bit addresses ships with a two-way set associative, writeback cache with perfect LRU replacement. The tag store (including the tag and all other meta-data) requires a total of 4352 bits of storage. What is the block size of the cache? Assume that the LRU information is maintained on a per-set basis as a single bit. (Hint: $4352 = 2^{12} + 2^8$ .)

## 5 Reverse Engineering Caches [100 points]

You're trying to reverse-engineer the characteristics of a cache in a system so that you can design a more efficient, machine-specific implementation of an algorithm you're working on. To do so, you've come up with four patterns that access various *bytes* in the system in an attempt to determine the following four cache characteristics:

- Cache block size (8, 16, 32, 64, or 128 B)
- Cache associativity (1-, 2-, 4-, or 8-way)
- Cache size (4 or 8 KB)
- Cache replacement policy (LRU or FIFO)

However, the only statistic that you can collect on this system is cache hit rate after performing the access pattern. Here is what you observe:

| Access Pattern | Blocks Accessed (Oldest → Youngest) | | | | | | | | | Hit Rate |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 4096 | 8192 | 12288 | 16384 | 4096 | 0 | | | 1/7 |
| B | 0 | 1024 | 2048 | 3072 | 4096 | 5120 | 6144 | 3072 | 0 | 1/9 |
| C | 0 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 4/9 |
| D | 128 | 1152 | 2176 | 3200 | 128 | 4224 | 1152 | | | 2/7 |

Based on what you observe, what are the following characteristics of the cache? (Be sure to justify clearly your answer for full credit.)

(a) Cache block size (8, 16, 32, 64, or 128 B)?

(b) Cache associativity (1-, 2-, 4-, or 8-way)?

(c) Cache size (4 or 8 KB)?

(d) Cache replacement policy (LRU or FIFO)?

## 6 Sectored Cache vs. Smaller Blocks [100 points]

After mounds of coffee and a good number of high-level simulation cycles, you narrowed down the design choices for the L1 cache of the next-generation processor you are architecting to the following two:

**Choice 1.** A sectored 64KB cache with 64-byte blocks and 8-byte subblocks

**Choice 2.** A non-sectored 64KB cache with 8-byte blocks

Assume the associativity of the two caches are the same.

(a) What is one definitive advantage of the sectored cache over the non-sectored one?

(b) What is one definitive advantage of the non-sectored cache over the sectored one?

(c) Which of the choices has the faster cache hit latency? Circle one:
**Choice 1** **Choice 2** **Not Enough Information**
Justify your choice, describing the tradeoffs involved if necessary.

## 7 Memory Interleaving [100 points]

A machine has a main memory of 4 KB, organized as 1 channel, 1 rank and $N$ banks (where N > 1). The system does not have virtual memory.

- Data is interleaved using a cache block interleaving policy, as described in lecture, where consecutive cache blocks are placed on consecutive banks.
- The size of a cache block is 32 bytes. Size of a row is 128 bytes.
- An open row policy is used, i.e., a row is retained in the row-buffer after an access, until an access to another row is made.
- A row-buffer hit is an access to a row that is present in the row-buffer. A row-buffer miss is an access to a row that is not present in the row-buffer.

(a) For a program executing on the machine, accesses to the following bytes miss in the on-chip caches and go to memory.

0, 32, 320, 480, 4, 36, 324, 484, 8, 40, 328, 488, 12, 44, 332, 492

The row-buffer hit rate is 0%, i.e., all accesses miss in the row-buffer.

What is the minimum value of $N$ - the number of banks?

(b) If the row-buffer hit rate for the same sequence were 75%, what would be the minimum value of $N$ - the number of banks?

(c) i) Could the row-buffer hit rate for the sequence be 100%? Why or why not? Explain.

ii) If yes, what is the minimum number of banks required to achieve a row-buffer hit rate of 100%?

## 8 Virtual Memory [50 points]

An ISA supports an 8-bit, byte-addressable virtual address space. The corresponding physical memory has only 128 bytes. Each page contains 16 bytes. A simple, one-level translation scheme is used and the page table resides in physical memory. The initial contents of the frames of physical memory are shown below.

| Frame Number | Frame Contents |
|---|---|
| 0 | Empty |
| 1 | Page 13 |
| 2 | Page 5 |
| 3 | Page 2 |
| 4 | Empty |
| 5 | Page 0 |
| 6 | Empty |
| 7 | Page Table |

A three-entry translation lookaside buffer that uses Least Recently-Used (LRU) replacement is added to this system. Initially, this TLB contains the entries for pages 0, 2, and 13. For the following sequence of references, put an 'H' below those that generate a *TLB hit* and write "PF" below those that generate a *page fault*. What is the hit rate of the TLB for this sequence of references? (Note: LRU policy is used to select pages for replacement in physical memory.)

References (to pages): 0, 13, 5, 2, 14, 14, 13, 6, 6, 13, 15, 14, 15, 13, 4, 3.

(a) At the end of this sequence, what three entries are contained in the TLB?

(b) What are the contents of the 8 physical frames?