Homework Assignment 4

CS/ECE 6810: Computer Architecture
Nov 28, 2018
Jake Pitkin, u0891770

Memory Systems

Due Date: December 11, 2018.
(120 points)

1. **Virtually Indexed Cache.** Referring to the lecture slide on virtual address translation and TLB, explain the challenges if the number of bits in the page offset does not equal the number of bits in the sum of "Index" and "Byte" (Please see `http://www.cs.utah.edu/~bojnordi/classes/6810/f18/slides/18-tlb.pdf`). You are asked to identify the issues and their corresponding solutions from the literature. **(20 points)**

   Special Instructions :

   i Strictly NO collaboration or brainstorming on this question is accepted.

   ii Mention ALL references used to answer this question. If no references are mentioned, 0 points will be awarded.

2. **Virtual Memory and TLB.** Consider an operating system (OS) using 1KB pages for mapping virtual to physical addresses. Initially, the TLB is empty and all of the required pages for the user application as well as the page table are stored in the main memory. (There is no need to transfer data between main memory and the storage unit.) Every access to the TLB and main memory takes respectively 1 and 200 processor cycles. The TLB can store up 16 entries.

   i Assuming that the main memory size is 1MB and the page table has 2048 entries, show the number of required bits for the physical and virtual address fields. **(5 points)**

   Given 1KB pages, both the physical and virtual address fields will require 10 offset bits. The page table has 2048 entries which will require 11 bits to address. Main memory is 1MB with 1KB pages, so it contains 1024 pages which will require 10 bits to address.

   ii Assume that the user application only generates five memory requests to the virtual addresses `0000`, `0004`, `0008`, `0800`, and `0804` (all in hexadecimal); the first request is ready at time 0; and the processor generates every next request immediately after serving the current one. Please find the execution time with and without using TLB in the proposed system and compute the attainable speedup due to using TLB. **(15 points)**

3. **DRAM Address Mapping.** Consider a simple in-order DRAM command scheduler. Initially, all DRAM banks are precharged and the scheduling queue contains seven read requests to the following physical addresses: `00040108`, `01040101`, `FF042864`, `A5181234`, `A5184321`, `00161804`, and `01040104` (all in hexadecimal). Using the following address mapping scheme, show all the required commands issued by the controller to serve the memory requests. **(20 points)**

row (10)	bank (4)	rank (2)	channel (0)	column (16)

4. **DRAM Row (Page) Management.** A computer system includes DRAM and CPU. The DRAM subsystem comprises a single channel/rank/bank. CPU generates a sequence of memory requests to rows `X` and `Y`. Table below shows the accessed rows by the memory requests and their arrival times at the DRAM controller. Assume that the DRAM bank is precharged initially; the DRAM scheduling queue has infinite size; and the memory interface must enforce the following timing constraints (all in ns): tCAS = 3, tRAS = 20, tRP = 20, tRCD = 10, and tBURST = 4.

Accessed Row	**Arrival time** (ns)
X	10
X	70
Y	80
X	85
Y	110
X	210

The goal is to evaluate two row management policies, namely *open-page* and *closed-page.* Show all the necessary transactions on the command, address, and data buses for each policies. Discuss which policy performs better for the given example. (Note: you are allowed to reorder requests already waiting in the memory controller.) **(20 points)**

Cmd --

Addr --

Data --

5. **Data TLB.** Consider the following pseudo code, in which **X** and **Y** are allocated as contiguous integer arrays in memory and are aligned to the 4KB boundaries. Assume that **k** and **l** are allocated in the register file with no need for memory accesses. We execute the code on a machine with virtual memory where the integer type (**int**) is 4 bytes wide. Assume that the system include a direct-mapped D-TLB with 1024 entries for serving all data accesses only. Initially, the D-TLB is empty. Find the hit rate of the D-TLB when running the code. **(20 points)**