MS108: Computer System 1

Spring 2013

Homework #4

Due: Two Weeks from Assignment

TA: Haopeng Liu

Course Email: ms108_2013@163.com

Collaboration Policy

These homework sets will be extremely valuable as tools for learning the material and for doing well on the midterm and final. You are required to obey the following rules:
(a) Each student should write out their solution independently and in their own words.
(b) Same applies to programming assignments – you should do your own coding.
Above all, make sure that you understand the solution to these homework problems. They really are assigned to help you understand the material and be prepared for the types of problems on the midterm and final!

Q1 Simultaneous multithreading (SMT)

Consider a Simultaneous Multithreading (SMT) machine with limited hardware resources. **Circle** the following hardware constraints that will limit the total number of threads that the machine can support. For the item(s) that you circle, **briefly** describe the minimum requirement to support **N** threads.

(A) Number of execution functional unit

(B) Number of physical registers

(C) Data cache size

(D) Data cache Associativity

(E) Number of instruction caches (suppose we have one instruction cache in a non-SMT processor)

Q2 Amdahl’s Law and Gustafson’s Law

Suppose program P takes 200 seconds to execute on a uniprocessor system. Of this time, 30% execution time is spent on the sequential part of the program, and 70% time is spent on the parallel part. The execution of the parallel part can be accelerated by using more processors, and the speedup for the parallel part is the same as the number of processors. For example, if it takes T seconds to execute the parallel part with one processor, using k processors would reduce the amount of time to T/k seconds for that part.

i. [4 marks] On a multiprocessor system with 5 processors, how much time would it take to execute program P?

ii. [6 marks] Using the Amdahl’s Law, find limit of speedup we can get by using more processors to execute program P without scaling up the problem size.

iii. [8 marks] Suppose we can scale up the problem size (e.g., giving the program a larger input data) when adding more processors into the system. Specifically, we scale up the problem so that the parallel part of P takes 8 times as much time as before to execute on a uniprocessor system. The scaled-up problem does not change the execution time of the sequential part. Compute the speedup of program P when we run it to solve the scaled-up problem on an 8-processor system.

Q3 Coherence Protocol Design

Add a clean exclusive state (a state where the block is being read *only* by the local processor) to the basic snooping cache coherence protocol we introduced in the class. Assume that the cache can distinguish a read miss that will retrieve a block destined to have a private state (non-shared) from a read miss delivering a shared block. In other words, instead of just having Read Hit/Read Miss you will now have Read Hit/Read Miss Exclusive, and Read Miss Shared transition events.

Show the protocol in the format of state machine we showed in class and describe all of the necessary changes made to the basic protocol in words.

Q4 GPU

H&P Problem 4.13 in Chapter 4, on page 339