

**Prctice Problems for Final Exam**  
**Computer Architecture CSCI 320 – Fall 2007**  
**December 3, 2007**

**Practice Problems for Final Exam** Included are some questions from old CS320 quizzes and final exams. **I promise at least one of these will be on the exam.**

1. (10 pts) Describe the difference between reliability and availability.
2. (10 pts) How are precise interrupts defined?
3. (10 pts) A processor issues 10 disk I/Os per second. The average disk service time is 40 ms. Assume requests are exponentially distributed and FIFO.
  - a). On average, how utilized is the disk?
  - b). What is the average time spent in the queue?
  - c). What is the average response time for a disk request including the queuing time and disk service time?
  - d). What is the average length of the waiting line?
  - e). What is the average number of tasks in service?
4. (15 pts) You are given a 6-stage pipeline where each stage takes 2 nanoseconds. Assuming you have 10 items to be processed, what is the speedup of the pipeline over the serial case?
5. (10 pts) Explain the mechanism of a “delayed branch” used in some RISC architectures. Give an example to demonstrate the concept. Explain why it is used.
6. (15 pts) Assume a CPU sends 15 disk I/O requests per second (exponentially distributed). Assume the average disk service time is 20 ms and the average time spent in the queue is 5 ms.
  - a). On average, what is the utilization of the disk?
  - b). What is the length of the queue? (Hint: Use John Little’s Law.)
7. (20 pts) A memory has 2 words per block, 32-bit words and is word addressable. An address is 32 bits wide. The write-through cache is direct mapped and has 8 cache lines. The write policy is write allocate. Given the following code sequence:

Address	Instruction	Comments
0	Load R1, 7	Load the contents of address 7 into R1
1	Add R1, 8	Add the contents of address 8 to R1
2	Store R1, 9	Store the contents of R1 into address 9
3	Load R2, 18	Load the contents of address 18 into R2
4	Add R2, 19	Add the contents of address 19 to R2
5	Store R2, 20	Store the contents of R2 into address 20
6	Halt	

- a). Show a diagram of an address request from the CPU. Label block address, tag, index and block offset fields. Show the size in bits for each.
- b). Show a diagram of a cache line. Label the fields and show size in bits for each field.
- c). Draw a diagram of the layout in memory showing the placement of the 7 instructions in blocks. Label the appropriate block addresses in the memory.
- d). Determine the following from executing the above instructions. Assume the cache is empty at start and the PC starts at zero.

cache read hits —

cache read misses —

cache write hits —

cache write misses —

8. (10 pts) Suppose we could improve the speed of the CPU in our machine by a factor of five (without affecting I/O performance) for five times the cost. Also assume that the CPU is used 50% of the time, and the rest of the time the CPU is waiting for I/O. If the CPU is one-third of the total cost of the computer, is increasing the CPU speed by a factor of five a good investment from a cost/performance viewpoint?
9. (15 pts) Some people have argued that with increasing capacity of memory storage per chip, virtual memory is an idea whose time has passed, and they expect to see it dropped from future computers. Discuss reasons for and against this argument.
10. (10 pts) Find all the data hazards in the following code and label them RAW, WAR, or WAW.

```

loop: L.D    F0,0(R1)
      ADD.D  F4,F0,F2
      SUB.D  F4,F2,F4
      S.D    F4,0(R1)
      DADDUI R1,R1,#-8
      BNE   R1,R2,loop

```

11. (20pts) In class we studied hardware approaches for exploiting instruction-level parallelism (ILP). This exercise asks how well hardware can find and exploit ILP. Consider the following MIPS code fragments each containing two instructions:

i).     DADDI R1,R1,#4  
        LD     R2,7(R1)

ii).    DADD  R3,R1,R2  
        SD     R2,7(R1)

iii). SD R2,7(R1)  
S.D F2,200(R7)

iv). BEZ R1,place  
SD R1,7(R1)

a). For each code fragment (i) to (iv) identify each type of dependency that exists or that may exist (a fragment may have no dependencies) and describe what data flow, name reuse, or control structure causes or would cause the dependency. For a dependency that may exist, describe the source of the ambiguity and identify the time at which that uncertainty is resolved.

b). For each code fragment, discuss whether dynamic scheduling is, may be or is not sufficient to allow out-of-order execution of the fragment.

12. (10 pts) Assume a dynamic branch prediction scheme. Why is a 2-bit predictor scheme preferred over a 1-bit predictor scheme?

13. (10 pts) When an instruction is correctly speculated, what is the effect on the three factors comprising the CPU time formula (page 42): dynamic instruction count, average clocks per instruction, and clock cycle time? When speculation is incorrect, it is possible for CPU time to increase. Which factor(s) of the CPU time formula best model this increase and why?

14. (10 pts) Apply the renaming algorithm to the following:

```
MUL.D R8,R8,R8  
ADD.D R8,R8,R8  
SUB.D R8,R8,R8
```