CE-4
August 2011 QE

I. Processor (30 points)

1) **(20 points)** A workload has occasional bursts of memory accesses followed by a period of computation (during which only registers are accessed). The number of misses in each burst has a significant impact on overall memory stalls. If there is only one miss in a burst, the latency of that miss is fully tolerated (i.e., no stall time results) because of the adequate independent work that can be overlapped with the miss. At the other extreme, if there are 4 or more misses in a burst, the processor stalls for the full memory access time because of the lack of independent work. Assume that the stall varies linearly when the number of misses per burst varies between 1 and 4.

Assume ideal CPI to be CPI_0 and memory access latency to be M cycles. Assume the memory bursts-per-instruction is b, and that the miss-distribution is specified by $<f_0, f_1, f_2, f_3, f_{4+}>$ where the fraction of bursts that see N misses is represented as f_N. (For example, f_0 represents the fraction of memory bursts which have 0 misses, and f_{4+} represents the fraction of memory bursts with four or more misses). Note, $f_0+f_1+f_2+f_3+f_{4+} = 1$.

Provide an expression for overall CPI.

2) **(10 points)** In designs that use a unified physical register file which holds both committed and non-committed register values (similar to MIPS R10K), when can a physical register be freed?

II. Memory (20 points)

(1) **(10 points)** Explain why victim-TLBs for L1 TLBs are significantly less likely to improve performance than victim caches for L1 caches.

Hint: the reason has nothing to do with the difference between virtual memory pages and cache blocks.

2) **(10 points)** Why do memory hierarchies typically use split caches at the L1 level and unified caches at the L2 level? You answer must include justification for both the L1 and L2 claims.

III. Multicore (20 points)

1) **(10 points)** Identify the major inefficiency in the implementation of LOCK/UNLOCK shown in the table below. (The LOCK routine uses a hardware-supported atomic test-and-set (T&S) instruction.) Assume a three-state MSI (Modified/Shared/Invalid) coherence protocol and an atomic bus.

LOCK(X) {
while (T&S(X) == 1);
}
	UNLOCK(X) {
	X=0;
	}

Write in Exam Book Only

2) **(10 points)** Suggest an alternate LOCK/UNLOCK implementation that fixes the inefficiency observed in part (1). Your solution must use the same T&S hardware support.

IV. Fundamentals (30 points)

A new memory technology called SprintRAM has been developed as a replacement for DRAM. SprintRAM's memory access latency (***service_time***) is comparable to that of SRAM caches and significantly better (i.e., lower) than DRAM. However after completion of a memory access, a SprintRAM memory array is unavailable to service other requests for a period of time called ***rest_time***.

1) (6 points) What is the impact of ***rest_time*** for memory intensive workloads? Give one high level idea to minimize the impact of ***rest_time***.

2) **(12 points)** Describe a class of access patterns for which one may build a SprintRAM-based **cacheless** memory system while completely avoiding ***rest_time***-related performance degradation. Your solution must describe both the access pattern and the SprintRAM memory system. Your solution must consider memory-intensive workloads with significant data-reuse. (Designs for non memory-intensive workloads or for workloads with no data-reuse will get zero credit.)

   Hint: You should not think about the amount of reuse; that is already high as stated in the question. Rather, think about how a reuse-heavy workload could avoid ***rest_time***-related performance degradation.

3) **(12 points)** Describe the type of workloads whose performance will degrade without SRAM caches in spite of SprintRAM's superior access latency.

Write in Exam Book Only