Name _____

EE 7700-2

Take-Home Final Examination

Tuesday, 10 May 2011 to Friday, 15 May 2011

Work on this exam alone. Regular class resources, such as notes, papers, documentation, and code, can be used to find solutions. Do not discuss this exam with classmates or anyone else. Any questions or concerns about problems should be directed to Dr. Koppelman.

Some questions in this exam are based on the paper "Debunking the 100X GPU vs. CPU Myth" by Lee *et al*, which will be referred to as Lee 2010 in the exam. The paper itself and a full reference are linked to the course references page, as are other papers cited in the exam.

Problem 1	_____	(10 pts)
Problem 2	_____	(14 pts)
Problem 3	_____	(14 pts)
Problem 4	_____	(14 pts)
Problem 5	_____	(14 pts)
Problem 6	_____	(18 pts)
Problem 7	_____	(16 pts)
Exam Total	_____	(100 pts)

Alias _____

Good Luck!

Problem 1: [10 pts]Table 2 in the paper Lee 2010 describes several performance measures for the Core i7-960 and GTX 280. Show what a row in this table would look like for GTX 480. Use data in the CUDA C Programming Guide 3.1 (or later) to answer your question. Show data for the following columns: Num PE, SP SIMD width, DP SIMD width, Peak SP SIMD FLOPS, and Peak DP SIMD FLOPS. Base the FLOPS calculation on fused multiply/add instructions. Use a clock frequency of 1.4 GHz.

Problem 2: [14 pts]In Lee 2010 Section 3.2, under the heading "Cache size/Multi-threading" the claim is made that "Since all the threads within a warp execute the same instruction, the warps are switched out upon issuing memory requests." (Please see the paper for the full context.) That does not sound exactly like the way memory access timing was described in class.

(*a*) Based on the description in class, how is the scheduling of warps that issue memory requests different than the statement above.

(*b*) Explain the implications for hiding memory latency.

Problem 3: [14 pts]In Lee 2010 Section 2 item 10 (the description of the sort algorithm) the number of bits per digit (called h in class and referred to as "bits considered per pass" in Lee 2010) is said to be limited by the amount of local storage. According to the radix sort paper, Satish 2009 (linked to the course references page), and similar to the algorithm presented in class, the size of storage of the histogram itself, 2^h integers, was not a major factor.

(*a*) What was the factor limiting the number of bits per digit?

(*b*) Though there was enough storage for a larger histogram, there was a disadvantage. What was the disadvantage?

Problem 4: [14 pts]One might argue that the warp size in NVIDIA CC 1.X devices should really be 8 and in CC 2.0 devices it should really be 16.

(*a*) Why might one think that those are the more natural warp sizes?

(*b*) If those really were the warp sizes, how might that affect the performance of CUDA code that was written for those new WARP sizes? (Assume that the clock frequencies in these smaller warp systems was the same as the original systems.)

(*c*) Describe the impact on the cost of the smaller warp systems. Be specific as possible.

Problem 5: [14 pts]The size of an application's working set relative to on chip (die) storage can determine whether it is bandwidth-bound or compute-bound. Lee 2010 Section 4.3.3 describes how several applications' working set sizes determine their bandwidth- or compute-bound status.

It should be clear that the Fermi L1 cache, shared memory, and registers count towards this on-die storage. However one can argue either that the texture cache does not count as on-die storage or that it does count as on-die storage. For this problem assume that the texture cache is used for ordinary indexed accesses and ignore texture cache features such as texture filtering. Also assume that there is no need to cache writable data. That is, an answer can't be that the texture cache isn't writable in CC 1.X. *Hint: For the solution to both parts below think about one of the surprising things about the texture cache. For the first part it shouldn't matter, for the second part it should matter.*

(*a*) Explain why the texture cache should count as on-die storage for purposes of reducing the bandwidth consumption of a CUDA program.

(*b*) Explain why the texture cache should not count as local storage. That is, having a real L1 cache would allow code to do something that the texture cache would not allow.

Problem 6: [18 pts]Section 5.2 of Lee 2010 offers some hardware recommendations. Describe what you believe is the most viable hardware recommendation for next-generation GPUs, something that might become available with two or three years. Be sure to justify why you believe that recommendation to be most viable. Full credit will be given to any of the recommendations so long as the justification makes good points. The justification should contrast your chosen recommendation with at least one of the others.

Problem 7: [16 pts]Answer each question below:

(*a*) What was the working set size of the matrix multiply algorithm on the NVIDIA GPUs based upon? Would having more local storage for a larger working set have helped performance very much?

(*b*) Lee 2010 explains that the NVIDIA GPU is more efficient than the Intel i7 at gather/scatter operations. One example of a gather operation is a global memory load of non-contiguous addresses. We know that such a load will result in multiple requests, normally a bad thing. Why then is that better than a gather on the i7?