Norwegian University of Science and Technology (NTNU)
DEPT. OF COMPUTER AND INFORMATION SCIENCE (IDI)

Course responsible: Professor Lasse Natvig
Quality assurance of the exam: PhD-student Magnus Jahre
Contact person during exam: Lasse Natvig, Phone: 906 44 580

Deadline for examination results: 3rd of July 2010.

EXAM IN COURSE TDT4260 COMPUTER ARCHITECTURE
Saturday 12th of June 2010
Time: 0900 – 1300

***** with preliminary solution sketches in violet *****
Last updated 2010-07-05

Supporting materials: No written and handwritten examination support materials are permitted. A specified, simple calculator is permitted.

By answering in short sentences it is easier to cover all exercises within the duration of the exam. The numbers in parenthesis indicate the maximum score for each exercise. We recommend that you start by reading through all the sub questions before answering each exercise.

The exam counts for 80% of the total evaluation in the course. Maximum score is therefore 80 points.

Exercise 1) Amdahls law and caches (Max 25 points)

a) (Max 6 points) Explain briefly Amdahl's law.
See course slides LN – lecture 1, slide 42 (and more). (See also the text book)

b) (Max 4 points) What is the main difference between a homogeneous and heterogeneous multicore chip, and explain how Amdahl's law can be used as an argument in favor of heterogeneous multicore chips.
In a homogeneous multicore chip the different cores are similar, i.e. they have the same ISA as well as same clock speed, same local cache size etc. Heterogeneous multicores have at least one core that is different from the others. It might have another ISA, and/or different cache size and architecture, and/or different clock speed. Amdahl's law explains how the serial part of a computation gives a limit of performance and speedup. The serial part cannot be parallelized, but if one of the cores is faster (more powerful) it can be used for executing the serial part.

c) (Max 5 points) Why does increasing the associativity of a cache often lower the miss rate? Give an explicit example.
Reduces conflict misses. One example would be that of a direct mapped cache, where you access A, then A', then A (both of which map to the same index). For a direct mapped cache, you miss on the second access to A because it was evicted by A'. When you have set associativity, you may cache A' without evicting A, and the third access will hit.

d) (Max 5 points) Most modern microprocessors have small level 1 data caches with a low associativity (2 for example). If increasing associativity lowers miss rate, then why does the level 1 cache often have a low associativity?

The level one access time is often on the critical path of the machine. Adding higher associativity means you have to do multiple parallel data reads (one for each way), and then mux down the data based on the tag comparison. The muxing amount increases with the number of ways, and therefore increases cycle time. Keeping it low allows you to lower the cycle time.

e) (Max 5 points) John the architect is designing a pipeline. He is having trouble meeting timing, and the critical path is shown in Figure 1. In the path, a Virtual Address (VA) is taken from a register, then used to look up a Physical Address (PA) from the TLB. The Physical Address is then used to access a level 1 data cache. The TLB page size is 4 KB and the level 1 data cache is also 4 KB. What can John the architect do to improve timing?

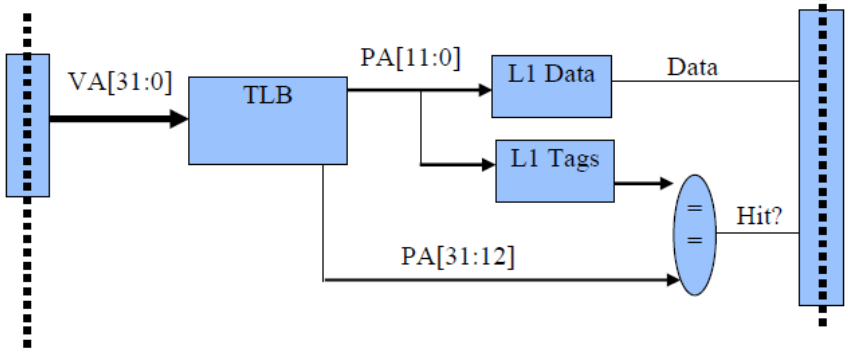

Figure 1.

Do a virtually indexed, physically tagged cache. Index the data and tags with VA bits, because those don't change from VA to PA (because the TLB page size is greater than or equal to the level 1 data cache size). See also textbook page 291 – 292.

Exercise 2) Instruction Level Parallelism (Max 10 points)

a) (Max 5 points) Instructions per cycle (IPC), the number of instructions committed per cycle for a given benchmark, is often used to compare different architectures. Why is this a poor metric for architectural comparison? Give an example of a machine that achieves an IPC of 1.2 on a given benchmark that might have lower performance than a machine which achieves an IPC of 1.0 on the same benchmark.

It does not account for ISA differences or different clock frequencies. In the above example, the machine with an IPC of 1.0 could be running at a clock frequency twice as fast as the other machine, resulting in an absolute performance delta of 1.66x (assuming the same ISA).

b) (Max 5 points) A computer architect is designing a new pipeline. She is considering a VLIW machine that can issue two instructions each cycle (ie. two-wide), and a two-wide in-order superscalar machine. Both machines have the same datapaths (ie, same number of ALUs, registers, etc). What is an advantage of choosing the VLIW machine? What is an advantage of choosing the superscalar design?

The VLIW design has simpler control, because the hardware does not have to dynamically detect parallelism. The superscalar machine will have a smaller instruction cache footprint, because VLIW often results in code bloat due to the fact that there will not always be two instructions to bundle together, thereby wasting that extra instruction encoding space.

Exercise 3) Memory Systems (Max 10 points)

a) (Max 6 points) Briefly explain three techniques that can be used to either reduce the hit time, increase bandwidth, reduce the miss penalty or reduce the miss rate in caches.

See Chapter 5.1 and especially 5.2 in the textbook

b) (Max 4 points) Your task is to evaluate the performance effects of implementing non-blocking caches. Consider a processor with a single on-chip cache and a 100 clock cycle miss penalty to off-chip memory. In program A, the misses that stall the processor occur in bursts of 4 misses every 100 clock cycles. For program B, a single miss stalls the processor every 400 clock cycles. What percentage of time will the processor be stalled waiting for memory with Program A and B for (i) a blocking cache and (ii) a cache that can service 4 misses concurrently?

To solve this assignment we rely on two observations:

- The stall time of misses that are serviced concurrently is approximately equal to the miss penalty of the first miss (i.e. the latency of the other misses are hidden).
- All misses in Program B are serialized. It does not matter if the cache is blocking or non-blocking

This gives the following stall times:

	Blocking Cache	4-way Non-Blocking
Program A	4 serialized misses → 400 clock cycles	4 parallel misses → 100 clock cycles
Program B	100 clock cycles	100 clock cycles

The stall percentage is given by the formula: stall cycles / (stall cycles + compute cycles)

This gives the following stall time percentages:

	Blocking Cache	4-way Non-Blocking
Program A	400 / (400+100) = 80%	100 / (100+100) = 50%
Program B	100 / (100+400) = 20%	100 / (100+400) = 20%

Exercise 4) Vector Processors and Interconnection Networks (Max 10 points)

a) (Max 5 points) What makes vector processors fast at executing a vector operation?
A Vector operation can be executed with a single instruction, reducing code size and improving cache utilization. Further, the single instruction has no loop overhead and no control dependencies which a scalar processor would have. Hazard checks can also be done per vector, rather than per element. A vector processor also contains a deep pipeline especially designed for vector operations.

b) (Max 5 points) Interconnection networks fall into two main routing categories (source routing and distributed routing). In a source routed scheme, a header is generated which contains what to do at each switch point along the path of the packet. In a distributed routing scheme, the header simply contains the destination address, and the router calculates what to do at each switch point. What is an advantage of the source routed scheme? What is a disadvantage?
With the source routed scheme, the switchpoints (routers) may be very simple, because they don't have to do any route calculation. This can result in lower latency. The disadvantage is that it requires a header that is larger than in a distributed routed scheme. In a source routed scheme, the destination address is decoded into a sequence of directions, in the distributed routed case, the destination address stays encoded, saving header space, but resulting in a more complex route header that must decode the destination address and calculate the proper path at each switch point.

Exercise 5) Chip Multiprocessors (Max 25 points)

a) (Max 6 points) In the paper Chip Multithreading: Opportunities and Challenges, by Spracklen & Abraham is the concept Chip Multithreaded processor (CMT) described. The authors describe three generations of CMT processors. Describe each of these briefly. Make simple drawings if you like.
Solution sketch: See lecture 6 (LN) slides 27 – 29.

b) (Max 6 points) Explain briefly the research method called design space exploration (DSE). When doing DSE, explain how a cache sensitive application can be made processor bound, and how it can be made bandwidth bound.
(Lecture 7 (LN) slide 9 and other parts of lecture) DSE is to try out different points in an n-dimensional space of possible designs, where n is the number of different main design parameters, such as #cores, core-types (IO vs. OOO etc.), cache size etc. Cache sensitive applications can become processor bound by increasing the cache size, and they can be made bandwidth bound by decreasing it.

c) (Max 4points) Discuss briefly the advantages and disadvantages of asymmetric multicore processors with homogenous instruction set architectures.
Advantages: A program can be run on any core without recompilation, power efficiency can be increased by running the program on a core where it can achieve high resource utilization, performance can be increased by utilizing a larger number of cores for scalable parallel programs, +++ Disadvantages: Increased CMP design complexity, increased complexity of system software, heterogeneity-unaware scheduling will in some cases reduce performance (i.e.

OS must be modified), +++

d) (Max 5 points) Your task is to analyze the performance and power efficiency of a parallel program on two different machines. In Machine A, the chip area is used to provide 4 high-performance processing cores where each core can complete 2 units of work each second. In Machine B, the area is used to provide 16 cores that can complete 1 work unit each second. Both machines use the same amount of power. The program is mapped to one thread per core and consists of 32 units of work. Furthermore, it scales ideally and all communication overheads are negligible. What is the runtime of the program on Machine A and B? Which machine is most power efficient for this program?

	Machine A	Machine B
Total amount of work	32	32
Work units per core	32 / 4 = 8	32 / 16 = 2
Total run time	8 /2 = 4 seconds	2 / 1 = 2 seconds

Both machines use the same amount of power. Since Machine B achieves higher performance than Machine A, it is more power efficient.

e) (Max 4 points) In Chip Multiprocessors, memory system units are often shared between processing cores. Commonly, the cores share the on-chip interconnect, the shared cache and the off-chip memory bus. Briefly describe how memory requests from different cores can interfere with each other in these units. How can this interference affect performance?

Interference can result in:

- Increased queuing delays in the on-chip interconnect
- An increased shared cache miss rate due to other processors evicting cache blocks that would otherwise be reused
- Additional queuing latencies in the memory bus
- Bonus point (not expected): latency effects due to complex interactions in the DRAM's 3D structure of banks, rows and columns

Interference increases the average memory latency, and this may increase the number of cycles a processor is stalled waiting for memory accesses. The magnitude of this impact depends on (among others) the programs ability to utilize the latency hiding mechanisms in the processor core and the parallelism available in the memory system.

…---oooOOOooo---…