Uppsala Universitet
Instutitionen för informationsteknologi
Avdelningen för datorteknik

Kursens Namn: Datorarkitektur 2		Datum 2001-12-19
Namn (efternamn först, v.g. texta)		Utbildningsprogram (eller liknande)
Namn (namnteckning)		Personnummer (10 siffror)
Termin och år då du först registrerade kursen	Inlämningstid	Bordsnummer

X= Solution provided B=I have bonus point ☺		Your Score	Max score
1			8
2			8
3			8
4			8
5			8
6			8
7			8
8			10
9			8
10			8
	Summa poäng:		82
	Betyg:		-

Tentamen i Datorarkitektur 2

(1DT636, Datorarkitektur MN2 och 1IT190, Datorarkitektur IT)

Onsdagen 2001-12-19

Place:	Postscriptum 2
Time:	14:00 – 19:00
Allowed help:	Calculator, pencil, and eraser.
Language:	Answer in Swedish or English

General information

- I will turn up at 16:00 and 17:30 to answer questions.
- Solve at most one “problem” on each sheet(there are 10 problems in total)
- Write your name on all pages
- Do not cross-reference between the solutions of different problems
- If anything seems unclear, state your assumptions clearly.
- Not readable and/or not understandable answers result in zero points.
- If you did not bring a calculator you still need to estimate the answer value (and state that you did not bring a calculator).
- Mark each question for which you are providing a suggested solution with an "**X**" in the first column on the cover page.
- Check the bonus pages to see if you have earned any “bonus questions”. You will automatically be given maximum points for the corresponding bonus questions. If so, put a "**B**" in the first column for those questions on the cover page.

May the force be with you,

// Erik Hagersten

I can be reached at 070-425 0502 during the exam.

1. (Bonus 1) System performance (8p)

A system has the following properties

- 10% of the instructions are stores
- 20% of the instructions are loads

- 90% of the loads hit in the 1st level data cache
- 90% of the load accesses to the 2nd level cache hit there

- 90% of the stores hit in the 1st level data cache
- 70% of the store accesses to the 2nd level cache hit there
- All the instruction fetches hit in the separate L1 instruction cache
- The CPU is a "perfect" 1.0 CPI pipeline, if all the loads and stores hit in the 1st level cache.

- The CPU clock frequency is 100MHz, in-order, and has an infinitely large write buffer.
- There is a 10 cycle stall if you miss in L1 data cache and hit in the L2 cache.
- There is a 100 cycle stall if you miss in both L1 and L2 caches
- There is unlimited bandwidth at all levels of the hierarchy
- The cache line size is 64 bytes

a) What is the CPI for the system if the overhead in the memory system is also taken into account? (3p)
b) What is the bandwidth of all the memory traffic assuming a copy-back cache strategy and that 20% of the L2 misses causes a write-back (and that there is inclusion between the L1 and L2)? (3p)
c) What is the speedup if the second-level cache was made larger and thus its miss rate was cut in half?(1p)
d) What would be the speedup if instead the CPU was clocked twice as fast, but the memory and cache latency (measured in time) stayed the same? (1p)

2. (Bonus 2) Loop scheduling (8p)

Consider the loop and the corresponding compiler-generated code

```
for (i=1; i<=1000; i = i+1)
   x[i] = x[i] + x[i+1];

loop:       LDD F0, 0(R1)          ;line 1
            LDD F2, -8(R1)         ;line 2
            ADDD F4, F0, F2        ;line 3
            SDD 0(R1), F2          ;line 4
            SUBI R1, R1, #8        ;line 5
            BNEZ R1, loop          ;line 6
```

*) LDD = Load double, SDD=store double

Assume that the array is stored in "backward order" in the array, i.e., that the address on x[i+1] is 8 bytes smaller than the address of x[i].

The pipeline has the following characteristics	Delay
INT ALU OP OUTPUT to INT ALU OP INPUT:	0 cycles
FP ALU OP OUTPUT to FP ALU INPUT:	2 cycles
LD DATA to INT/FP INPUT (load delay)	2 cycles
FP ALU OUTPUT to ST INPUT:	2 cycles
Branch delay slots:	2 cycles

a) Write one-sentence comments for each line (2p)

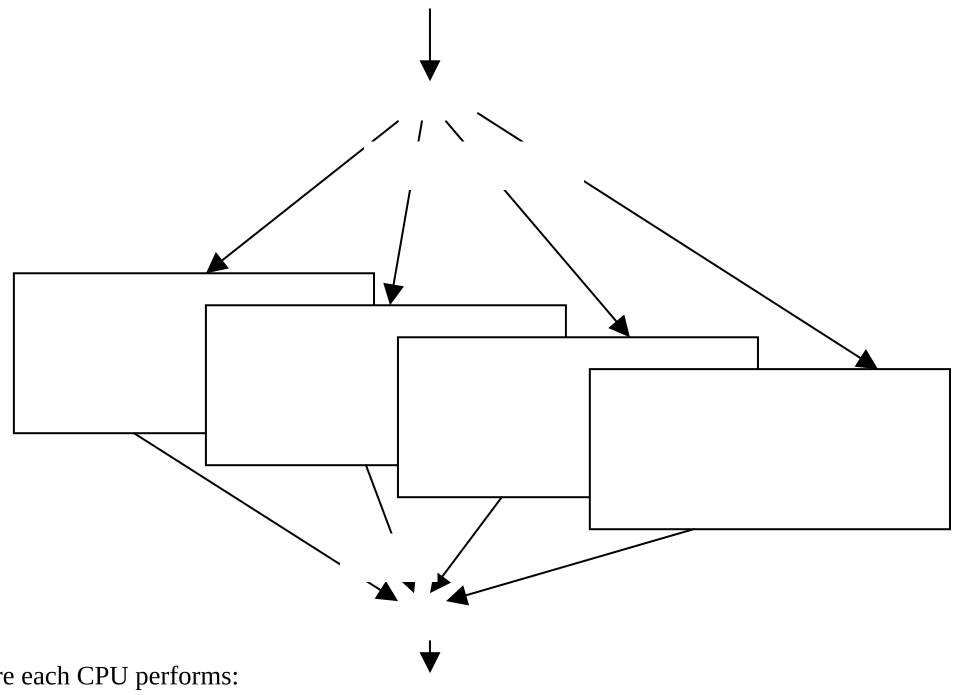
b) Show where the bubbles are in each and calculate the number of cycles in each iteration? (2p)

c) Show how the loop can be statically scheduled to improve the performance and calculate the number of cycles required for each iteration. (2p)

d) Show how loop unrolling with maximum optimizations could help avoiding all the stalls in this loop using the smallest amount of unrolling. You can not introduce any new instruction types. You can only re-schedule, remove or re-offset existing ones. You do not have to show the set-up/clean-up code before and after the loop (2p)

3. (Bonus 3) Synchronization and memory ordering (2+2+4=8p)

Consider the following parallel example:



Where each CPU performs:

```
while (sum < threshold)
      sum := sum + 1;
```

in the parallel section.

a) i) What is the highest and lowest number the printf(sum) may print?
ii) What is the highest and lowest number of additions that may be performed?
(2p)

b) Replace each parallel section with the following code:

```
lock(L)
while (sum < threshold)
      sum := sum + 1;
unlock(L)
```

i) What is the highest and lowest number the printf(sum) may print?
ii) What is the highest and lowest number of additions that may be performed?
(2p)

c) Write pseudo code for an implementation of lock(L) and unlock(L) using the function test-and-set(L) that return the old value stored in the memory location L and atomically writes the value “1” to that memory location. The implementation should produce a reasonable amount of bus coherence traffic (i.e., no banging!) (4p)

4. Random Questions (8p)

(At most two sentences per subquestion)

a) For embedded CPUs, a low power consumption is often essential. Write an expression for power consumption using the terms: voltage, frequency and area (i.e., capacitance)

b) Suggest two techniques for lowering power consumption

c) Which (one or many…) of the programming paradigms OpenMP, MPI or Posix threads (ptherads) supports a coherent shared memory view?

d) IA64 (as well as some other microprocessors) uses a technique to avoid branches for small if/then/else statements. Describe briefly how it works and what it is called.

e) What is a queue-based lock and what is its advantage over spin locks?

f) What do the acronyms UMA, NUMA and COMA stand for. What special application optimizations are needed for each one of the in order to achieve a good performance (two sentences each).

g) Software-based distributed shared memory, as suggested by Kai Li, is an alternative way of cheaply achieving a shared-memory image between more loosely couple nodes. It relies on a commonly existing hardware/software support to do its coherence check (i.e., am I allowed to perform this operation?) with no extra access time penalty. How is this achieved?

h) How are coherence check done in DSZOOM and what is the potential pros/cons of that technique compared with Li's?

5. Virtual Memory (2+2+2+2=8p)

(At most two sentences per subquestion)

The virtual memory system can be viewed as a "cache system":

a) what part of it corresponds to a cache line

b) what is its associativity

c) what is its write strategy

d) what piece of hardware is used to determine if a requested item is in the "cache" or not

6. Optimize for Caches (2+2+4=8p)

Make <u>trivial</u> changes to the following C programs such that the cache will be better utilized.

a) (2p)

```
for (j = 0; j < N; j = j + 1)
            for (i = 0; i < N; i = i + 1)
                        x[i][j] = 2 * x[i][j];
```

b) (2p)

```
for (i = 0; i < N; i = i + 1)
            for (j = 0; j < N; j = j + 1)
                        a[i][j] = 2 * b[i][j];

for (i = 0; i < N; i = i + 1)
            for (j = 0; j < N; j = j + 1)
                        c[i][j] = K * b[i][j] + d[i][j]/2
```

c) What kind of locality have you improved under a) and b) respectively? (4p)

7. Random questions (8p)

a) Describe the different types of pipeline hazards: (2p)

i) structural hazards,

ii) data hazards

b) Describe the two architectures: super-scalar and VLIW, and clearly state the difference between the two approaches (2p).

c) What is a victim cache and what kind of cache misses (Mark Hill's three Cs) does it remove (2p)?

d) IA64 calls it *register stack*, SPARC and the original RISC architecture call it *register window*. Describe briefly how one of these techniques work and what advantage it has over "normal" registers. (2p)

8. Cache coherence (10p)

All the three CPUs in a MOSI shared-memory multiprocessor executes the following code almost at the same time:

```
lock(L);        /* an optimistic spin lock with local spinning in the cache*/
A = A + 1;      /*is compiled to 3 instr: LD R1,#A; ADD R1,R1,#1; ST R1,#A*/
unlock(L);
<after a long time …>
<some other execution replaces A and L from the cache, if still present>
```

CPU1 is running slightly ahead of CPU2 and CPU3 and will get first to the critical section, but all of them will still have made their first attempt to acquire the lock L before CPU1 gets into the critical section. CPU2 is the second one to enter the critical section, and CPU3 is last.
The following four bus transaction types can be seen on the snooping bus connecting the CPUs:

- RTS: ReadtoShare (reading the data with the intention to read it)
- RTW, ReadToWrite (reading the data with the intention to modify it)
- WB: Writing data back to memory
- INV: Invalidating other caches copies

Initially, both A and L only reside in memory, both with a data value of zero.

Rip out and fill in the solution sheet at the end of the exam. Show every state <u>change</u> and/or value <u>change</u> of A and L in each CPU's cache according to a possible interleaving of the parallel memory instruction (including all memory instructions inside the lock and unlock routine). After the parallel execution is done for all of the CPUs, the cache lines still in the caches are being replaced. These actions should also be shown. For each line, also state what bus transaction occurs on the bus (if any) as well as which device is providing the corresponding data (if any).

This small example shows a parallel program where CPU1 executes a load(A) instruction after which CPU2 executes a load(L) instruction, followed by the replacement actions to empty the caches.

CPU action	**Bus transaction**	**State/value after the bus transaction**						**Data is provided by**
		CPU1		CPU2		CPU3		[CPU 1, 2, 3 or Mem]
		A	L	A	L	A	L	
Initially		I	I	I	I	I	I	
CPU1: LD A	RTS(A)	S/"0"						Mem
CPU2: LD L	RTS(L)				S/0			
… some time elapses ..								
CPU1: replace A	-	I						-
CPU2: replace L	-				I			-

9. Caches (6+2 = 8p)

a) Make a drawing of a physical cache with the following data

- Cache size: 4Mbyte
- Cache line size: 64byte
- Organization: 2-way
- CPU word size: 8 byte (this is the size of the data unit delivered from the cache to the CPU)
- Physical address size: 42 bits (i.e., 4Tbyte of address space can be addressed)

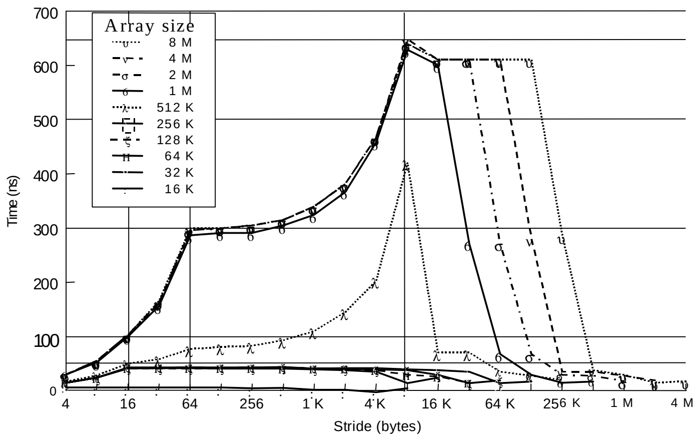
Clearly show the address bits ranges used for the indexing, the comparisons, and multiplex (mux) selects, etc. Describe, either in words or in logic, the functionality of all logic needed to resolve a read access to the cache.(6p)

b) Which bit ranges would change if the cache size was increased to 8Mbyte while all the other parameters stayed the same? State the new bit ranges (you do not need to make completely new drawing) (2p)

10. Microbenchmarks (2+2+2+2=8p)

A simple microbenchmark code below is used to draw curves that characterizes the properties of the underlying memory system

```
for (times = 0; times < Max; time++) /* many times*/
  for (i=0; i < ArraySize; i = i + Stride)
    dummy = A[i]; /* touch an item in the array */
```



Next page contains four such diagrams (A, B, C and D).

a) One generation computer is used to produce graph A, the next generation product is used to produce graph B. What improvement does the new product contain?

b) One computer system is characterized by the curve C. After updating its operating system its curve looks like B. What feature does the new operatingsystem implement?

c) One generation computer is used to produce graph C, the next generation product is used to produce graph B. Both are running identical operating systems What improvement does the new cache hierarchy contain? (Clarification: B and C both have a two-level cache hierarchy)

d) One generation computer is used to produce graph D, the next generation product is used to produce graph B. What improvement does the new product contain?

Solution sheet Question 9 Your Name ____

CPU action	**Bus transaction**	**State/value after the bus transaction**						**Data is provided by** [CPU 1, 2, 3 or Mem]
		CPU1		CPU2		CPU3		
		A	L	A	L	A	L	
Initially		I	I	I	I	I	I	