

Umeå University
Department of Computing Science
5DV118 — Computer Organization and Architecture
Examination: January 19, 2012

Name (printed): _____

Swedish ID number: _____

Computer User-ID: _____

Signature: _____

Secret code number: _____

Instructions: / Instruktioner:

This examination will be graded anonymously. This page will be removed before the instructor receives the examination for grading. The secret code number given above must therefore be written on every answer page which you turn in to the examination proctor.

Denna skrivning rättas kodad. Detta blad kommer att avskiljas innan läraren får skrivningen för rättning. Ovanstående kod måste därför finnas på samtliga svarsblad när du lämnar skrivningen till skrivvakten.

To the proctor of the examination: / Till skrivningsbevakaren:

Detach this cover sheet from the examination and put it in the envelope which is addressed to Yvonne Löwstedt, Department of Computing Science.

Avskilj detta försättsblad och stoppa i kuvert som skickas till Yvonne Löwstedt, Datavetenskap.

Umeå University
Department of Computing Science
5DV118 — Computer Organization and Architecture
Examination: January 19, 2012

Secret code number: _____

1. Answers may be written in English or Swedish. However, all technical terms which do not have an absolutely standard representation in Swedish must be given in English.
2. A pocket engineering calculator (miniräknare) and English/X - X/English dictionary may be used. No other help materials are allowed.
3. Answers must be written on the official university answer sheets which are provided the sheets in numerical order of the problems, and write on only one side of the paper. Write only the question number and your secret code number on these pages; do not write your name or ID. Fields on the answer sheets: Kod: your secret code; Uppgift nr: problem number; Sidnr: page number; poäng: points (leave blank).
4. Show your work. For questions which require that an answer be computed, numerical answers without derivations will not receive credit. It is furthermore a good idea to show the symbolic formula used to obtain the answer, since that will result in more credit in the case that an error is made.
5. The examination has a total of 1000 points.
6. For problems with multiple parts, you have the choice, for each part, to do the problem, or to skip it for partial credit. In the table below, place an X in the position for any problem for which you have attempted a solution, and which you wish to have graded. It is extremely important that you fill in this table properly, because of the following option. For any box which is left blank, the associated question will not be graded, and you will instead be awarded 15% of the points for that question. Your decision to leave a box blank is definitive, so be very careful. For example, If you leave box 8(b) blank, your answer to that question will not be graded, even if it is completely correct. On the other hand, if you place an X in box 8(b), but provide no answer whatsoever to that question, you will not receive 15% of the points for that question. It is strongly recommended that you use a pencil, in case you change your mind!

Prob	1	2	3	4	5	6	7	8	9	10	11
(a)											
(b)					■						
(c)	■		■		■				■		■

(1: 100 points total) The average CPI (cycles per instruction) for different kinds of instructions for a computer is given in the table below, together with the frequency of each type of instruction for a given program.

Operation	Frequency	CPI
Fixed-point ALU	40%	1
Floating point	10%	10
Load	20%	3
Store	20%	3
Branch	10%	2

- (a: 50 points) Compute how much faster (as a percentage) the program would run if the average CPI for floating-point operations were reduced to 6.
- (b: 50 points) Compute how much faster (as a percentage) the program would run if the average CPI for both load and store operations were reduced to 2. (The CPI for floating-point operations remains at 10 for this part.)

(2: 100 points total) Consider a MIPS load instruction of the following form, where n is some number.

$$\text{lw } \$t1, (n)\$t2$$

- (a: 20 points) Identify the range of numerical values which are allowed for n .
- (b: 40 points) Illustrate how this instruction is represented, field by field, as a 32-bit word in MIPS. It is not necessary to give the opcode or numerical values for the register names, but all other information should be provided.
- (c: 40 points) Explain precisely how the memory address whose contents is to be loaded is computed. (Do not give details about how it is implemented in hardware; just give the formula for the final address in terms of register values, the program counter, etc.)

(3: 100 points total) Answer the following questions about the MIPS `jal` (jump and link) instruction.

- (a: 50 points) Explain how `jal` is used for calls to and returns from a leaf procedure in a high-level language. (A leaf procedure is one which does not call other procedures.) Illustrate with a simple example.
- (b: 50 points) Repeat (a) for a non-leaf procedure, indicating in particular any special measures which must be taken. Illustrate with a simple example.

(4: 60 points total) Consider a floating-point number represented using IEEE 754 format:

$$(-1)^{\text{Sign}} \times \text{Significand} \times 2^{\text{Exponent}}$$

- (a: 20 points) In *normalized* format, the value of the significand satisfies $n_1 \leq \text{Significand} < n_2$. Give the values for n_1 and n_2 .
- (b: 20 points) The Significand is usually represented in a compact form as a *fraction*. Explain what this means and illustrate with a simple example. Identify in particular how much storage is saved via this representation.
- (c: 20 points) The exponent is usually represented in *bias* format. Explain what this means and why this representation is preferable to two's complement.

(5: 40 points total) Explain why almost all digital computers use a complement representation (such as two's complement) for negative fixed-point numbers, rather than a sign-magnitude representation.

(6: 100 points total) An implementation of the MIPS architecture following the model developed in the textbook has the following delays for each of the five main stages:

Component	Delay
Instruction fetch (IF)	160 ps.
Instruction decode and register file read (ID)	100 ps.
Execution or address calculation (EX)	220 ps.
Data memory access (MEM)	180 ps.
Write back (WB)	90 ps.

- (a: 40 points) Assuming a non-pipelined implementation, and assuming all delays not explicitly identified above to be negligible, compute the latency for each of the five instructions `addi`, `bne`, `jr`, `lw`, and `sw`.
- (b: 40 points) Repeat part (a), this time assuming a pipelined implementation with the five stages identified above.
- (c: 20 points) Suppose that the instruction mix of a program is as follows: `add`: 15%; `mul`: 15%; `beq`: 25%; `jr`: 10%; `lw`: 20%; `sw`: 15%. For the pipelined implementation, but with no forwarding or other special hazard-resolution units, compute the percentage of total cycles which utilize the ALU (arithmetic-logic unit). Assume that there are no stalls or instruction flushes.

(7: 100 points total) Given is the following instruction sequence.

```
lw   $t1, 0($t2)
sub  $t3, $t2, $t1
sw   $t4, 0($t3)
add  $t3, $t1, $t1
sub  $t1, $t1, $t2
sra  $t3, $t2, 12
xor  $t5, $t1, $t2
```

- (a: 30 points) Identify all *read after write* (RAW) data dependencies, regardless of the distance. (Note: These dependencies are called *read before write* in the slides.)
- (b: 35 points) For the five-stage pipeline model of the textbook, indicate which of these dependencies results in a hazard in the case that there is no forwarding, and show how `nop` operations may be inserted to avoid these hazards. Find one solution for the instruction sequence which resolves all hazards, not one solution for each hazard. The solution must not contain any unnecessary `nop`'s.
- (c: 35 points) Repeat part (b) in the case that forwarding is employed. Insert `nops` only in the case that the hazard cannot be avoided by forwarding alone.

Note: If you make an error in your solution to part (a), you may lose substantial credit for parts (b) and (c), even if those answers are correct relative to your answer for (a). This deduction will occur particularly if your answer to (a) makes the solutions for (b) and/or (c) much simpler.

(8: 100 points total) Answer the following questions concerning address spaces and their support within computer hardware.

- (a: 40 points) Explain the difference between a *physical address* and a *virtual address*.
- (b: 30 points) Explain what a *translation lookaside buffer (TLB)* is, and clarify its rôle in the support of efficient cache processing.
- (c: 30 points) The address associated with a cache entry is typically a physical address and not a virtual address. Explain the reason for this design decision.

(9: 100 points total) Consider a processor with an L1 cache for the MIPS instruction set and the following parameters.

Ideal CPI	5
Miss rate for access to instruction memory	2%
Miss penalty for access to instruction memory	150 clock cycles
Miss rate for access to data memory	3%
Miss penalty for access to data memory	125 clock cycles
Instruction mix	Load 10%, Store 15%, Arithmetic 65%, Branch 10%

(a: 50 points) Compute the actual CPI for this processor-cache pair.

(b: 50 points) Repeat (a), this time assuming that there is also a unified L2 cache with with a miss penalty of 20 clock cycles for access to this L2 cache from the L1 cache. Assume further that the miss rate for access to the L2 cache is 0.10%.

(10: 100 points) A memory system operates on a bus which is one-word (32-bits) wide. It is governed by the following parameters:

Time to send the address to memory	1 clock cycle
Row cycle time	10 clock cycles
Column access time	4 clock cycles
Time to return one word from memory	1 clock cycle

(a: 40 points) Compute the number of clock cycles necessary to fetch one word from memory.

(b: 30 points) Compute the number of clock cycles necessary to fetch eight words from memory, assuming that the memory is not interleaved, and that there are two blocks of four words, with the words of each block in the same row, but the two blocks in different rows.

(c: 30 points) Compute the number of clock cycles necessary to fetch eight words from memory, assuming that the memory is interleaved with four banks. Assume further that the eight words are arranged into two blocks of four words each, with each word of a block in a different memory bank, but with the two blocks in different rows.

(11: 100 points total) A hard drive for a laptop computer has the parameters given in the table below.

Parameter	Value
Rotational speed	4200 rpm
Average seek time	16 ms.
Controller overhead	1 ms.
Transfer rate	25M bytes/sec.

- (a: 50 points) Compute the average amount of time required to read or write 100K Bytes. Assume that the entire transfer requires only one seek and latency penalty.
- (b: 50 points) To increase performance, two alternatives are possible. The first alternative has a rotational speed of 5400 rpm, with all other parameters the same as in the table above. The second alternative has a transfer rate of 50M bytes/sec., with all other parameters the same as in the table above. Determine which one of these alternatives will give the greater improvement for the average amount of time required to read or write 100K Bytes over that which was computed in part (a).