

**Umeå University**  
**Department of Computing Science**  
**5DV008 — Computer Architecture**  
**Examination: April 29, 2011**

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**  
**5DV008 — Computer Architecture**  
**Examination: April 29, 2011**

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.
4. Translations for the boxes on the answer sheets: Namn: = Name (Leave blank.) Personnr: = Swedish identity number (Put your secret exam code in this field, not your Swedish identity number.) Uppgift nr. = Problem number (Please fill this in.) Sidnr: = Page number (Please fill this in, beginning with 1.) Poäng: = Points (Leave blank.)
5. 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.
6. The examination has a total of 1000 points.
7. 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
(a)								
(b)	■						■	
(c)	■					■	■	■

(1: 100 points total) Suppose that two computers  $C_1$  and  $C_2$  implement the same instruction-set architecture. Computer  $C_1$  has a clock speed of 3 GHz, while  $C_2$  has a clock speed of only 2.5 GHz. However, for a given program  $P$ ,  $C_1$  has an effective CPI (cycles per instruction) of 2.0 while  $C_2$  has an effective CPI of 3.0. For program  $P$ , show by calculation which computer is faster and by how much (expressed as a multiplier). [A simple statement that  $C_1$  is faster or  $C_2$  is faster, without any supporting explanation, will not receive credit.]

(2: 125 points total) Consider a MIPS load instruction of the following form, where  $n$  is some number.

`sw $t1, (n)$t2`

- (a: 25 points) Identify the range of numerical values which are allowed for  $n$ .
- (b: 50 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: 50 points) Give the formula which represents the address of the memory location into which the result is stored. Indicate also any shifting or padding operations which are required to compute this address from the given fields. It is not necessary to explain the operation of the hardware which performs this computation.

(3: 125 points total) Answer the following questions about the representation of floating-point numbers using IEEE 754 format.

- (a: 50 points) Identify the three fields which comprise this format and explain briefly the purpose of each.
- (b: 25 points) Explain clearly, and illustrate by example, the notion of *overflow*.
- (c: 50 points) Explain clearly, and illustrate by example, the notion of *underflow*.

(4: 125 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)	100 ps.
Instruction decode and register file read (ID)	150 ps.
Execution or address calculation (EX)	200 ps.
Data memory access (MEM)	100 ps.
Write back (WB)	50 ps.

- (a: 50 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`, `beq`, `jr`, `lw`, and `sw`.
- (b: 50 points) Repeat part (a), this time assuming a pipelined implementation with the five stages identified above.
- (c: 25 points) Suppose that the instruction mix of a program is as follows: `add`: 10%; `addi`: 10%; `bne`: 10%; `jr`: 10%; `lw`: 30%; `sw`: 30%. For the pipelined implementation, compute the percentage of total cycles which utilize data memory. Assume that there are no stalls or hazards.

(5: 150 points total) Given is the following instruction sequence.

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

- (a: 50 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: 50 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. Also, show how these hazards may be eliminated by giving an augmented version of this instruction sequence in which explicit `nop` instructions have been inserted. 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: 50 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.

(6: 150 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	100 clock cycles
Miss rate for access to data memory	2%
Miss penalty for access to data memory	100 clock cycles
Instruction mix	Load 10%, Store 15%, Arithmetic 65%, Branch 10%

(a: 75 points) Compute the actual CPI for this processor-cache pair.

(b: 75 points) Repeat (a), this time assuming that there is also a unified L2 cache with with a miss penalty of 30 clock cycles and a miss rate of 0.20%.

(7: 100 points total) Two fundamental ways of managing cache writes are *write through* and *write back*. Describe how each of these operates, and give the advantages and disadvantages of each.

(8: 125 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	15 ms.
Controller overhead	1 ms.
Transfer rate	50M bytes/sec.

(a: 75 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 100M 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).