Umeå University
Department of Computing Science
5DV118 — Computer Organization and Architecture
Examination: April 03, 2013

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: April 03, 2013

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 cycles per instruction (CPI) for various classes of instructions of an implementation I of a given architecture are given below. Also given is the frequencies Frequency_1 and Frequency_2 of each class of instruction for each of two programs P_1 and P_2, respectively.

Operation	CPI	Frequency_1	Frequency_2
Fixed-point ALU	6	40%	20%
Floating point	12	10%	20%
Load	3	20%	25%
Store	3	20%	25%
Branch	2	10%	10%

(a: 50 points) Assume that program P_1 consists of 1×10^7 instruction executions while P_2 consists of 9×10^6 instruction executions. Compute how much faster or slower P_2 runs relative to P_1 on I.

(b: 50 points) Compute how much larger or smaller the value of CPI for floating point operations must be in order that the two programs will run in exactly the same amount of time. If no such value is possible, indicate clearly (and quantitatively) why this is the case. The computed value need not be an integer.

(2: 100 points total)

(a: 50 points) Consider the 32-bit word $b = b_{31}b_{30}\ldots b_1b_0$, with each b_i one bit in the binary representation. Suppose further that $b_{31}b_{30}b_{29}b_{28}b_{27}b_{26} = 000010$, which is the opcode for the MIPS jump instruction `j`. When b is interpreted as a MIPS instruction and executed, explain how the target address for the jump is computed from b, given that the value of the program counter, in binary, at the conclusion of the fetch of the instruction is $p = p_{31}p_{30}\ldots p_1p_0$. Express the target address as a sequence of 32 bits, just as b and p are expressed above.

(b: 50 points) Suppose that it is necessary to jump to an arbitrary address `z` in the 32-bit address space. Assume further that the value of `z` is stored in the memory location whose address is contained in register `$t0`. Give a sequence of no more than three MIPS instructions, expressed in assembly language, which will accomplish that task.

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

(a: 50 points) Explain how `jal` is used in calls to a leaf procedure in a high-level language, and in particular how returns are made from such calls. (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) Answer the following questions about the representation of floating-point numbers using IEEE 754 format.

(a: 20 points) Identify the three fields which comprise this format and explain briefly the purpose of each.

(b: 20 points) Explain clearly, and illustrate by example, the notion of *overflow*.

(c: 20 points) Explain clearly, and illustrate by example, the notion of *underflow*.

(5: 40 points total)

(a: 20 points) 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.

(b: 20 points) Explain why two's complement is preferred to one's complement for the representation of fixed-point numbers in modern computers.

(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:

Stage	Delay
Instruction fetch (IF)	200 ps.
Instruction decode and register file read (ID)	100 ps.
Execution or address calculation (EX)	300 ps.
Data memory access (MEM)	400 ps.
Write back (WB)	150 ps.

(a: 40 points) For each of the instructions `bne`, `jr`, `lw`, and `sw`, compute the *critical-path time* for that instruction; that is, the amount of time required to execute that instruction individually, with unused stages consuming zero time. Assume that there are no special units to accelerate performance, such as dedicated adders to compute jump addresses.

(b: 40 points) Assume a pipelined implementation, with the five stages identified by the components above, compute the amount of time required to complete an entire instruction.

(c: 20 points) Suppose that the instruction mix of a program is as follows: `add`: 15%; `mul`: 15%; `beq`: 10%; `bne`: 10%; `jal`: 10%; `jr`: 5%; `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 pipeline flushes, and ignore the time required to fill and to empty the pipeline at the beginning and end of the instruction sequence.

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

```
add  $t5, $t1, $t5
lw   $t1, 8($t5)
sub  $t2, $t5, $t1
addi $t2, $t1, 3
sw   $t4, 0($t2)
add  $t2, $t1, $t4
sub  $t1, $t1, $t5
sra  $t3, $tx, 12
xor  $t5, $t1, $t5
```

(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.) For each read, consider only the most recent preceding write of the same data item to cause a RAW dependency.

(b: 35 points) For the five-stage pipeline model of the textbook, indicate which of these dependencies results in a data 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 `nop`s 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) A two-way set-associative cache is to hold 256K bytes of data. Assuming that a one-bit valid field is required (for each way), answer the following questions for two-word data blocks (per way) and words of 32 bits.

(a: 35 points) Compute the size of the index needed for the cache.

(b: 35 points) Determine the size of the tag field (for each way).

(c: 30 points) Compute the total number of bits required for the cache.

Hint: Draw a picture of a typical row in this cache, to make sure that you have accounted for all things. (A row consists of all of the elements for a given index — n elements for an n-way set-associative cache.) The total size is then the size of one row times the number of rows. If you draw such a row correctly, it will help the grader understand what you did, and may result in greater partial credit in case there are errors in your solution.

(9: 100 points total) Consider a processor with separate L1 caches for instructions and for data, for the MIPS instruction set and with the following parameters.

Ideal CPI	5
Miss rate for access to instruction memory	2%
Miss penalty for access to instruction memory	130 clock cycles
Miss rate for access to data memory	3%
Miss penalty for access to data memory	110 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 a miss penalty of 25 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 0.15%, and that the miss penalty for access to both instruction and data memory from the L2 cache is 150 clock cycles.

(10: 100 points total) 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	9 clock cycles
Column access time	4 clock cycles
Time to return one word from memory	1 clock cycle

In answering each of the following questions, in addition to showing how the numerical answer is obtained, draw a diagram which illustrates the actions which occur over time and how they overlap.

(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 twelve (12) 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 twelve (12) 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 desktop computer has the parameters given in the table below.

Parameter	Value
Rotational speed	5400 rpm
Average seek time	11 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 10000 rpm, with all other parameters the same as in the table above. The second alternative has a seek time of 9 ms., 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).