Written Exam in
Computer Organization and Components (IS1500)
Computer Hardware Engineering (IS1200)

Example Exam
Fall 2014

Examiner: David Broman
Teacher on duty: David Broman

Instructions

• Allowed aids: One sheet of A4 paper with handwritten notes. You may write on both sides of the paper.
• Explicitly forbidden aids: Textbooks, electronic equipment, calculators, mobile phones, machine-written pages, photocopied pages, pages of different size than A4.
• Please write and draw carefully. Unreadable text may lead to zero points.
• You do not need to return these exam papers when you hand in your exam solutions.

The exam consists of two parts:
• **Part 1: Fundamentals:** The maximal number of points for part 1 is 40 points (for IS1200) and 48 points (for IS1500). There are 8 points for each of the six course modules. All questions in Part I expects only short answers. See below for details.
• **Part 2: Advanced:** The maximal number of points for part 2 is 50 points. The answers to these questions requires the student to discuss, analyze, or construct. Answers to these questions require clear motivations.

**Short Questions in Part 1**

Part 1 consists of questions with short answers. For a correct answer, at most a few sentences are needed.

**True/False/Don’t know Questions in Part 1**

These questions can give between 0 and 4 points. Each question consists of 4 statements. For each statement, you should answer either true, false, or “don’t know”. The points are calculated as follows: Each correct answer (answer true or false) gives one point. Each incorrect answer (true or false) gives minus one point. If you answer “don’t know” for a statement, it neither adds nor removes points. The rationale for introducing “don’t know” answers is to avoid that the student makes guesses.

Example: Assume that the correct answers to four statements are: true, false, true, false. Assume that the student answered: true, true, don’t know, false. The total number of points is then: 1 - 1 + 0 + 1 = 1 point. Note that even if the answers to all four statements are wrong, the points for the whole question can never be negative, that is, the final points will always be between 0 and 4.

1Updated version March 19, 2015: Fixed a type in the code example in exercise 4b and an error in the proposed solution for exercise 5a.
Grades
To get a pass grade (A, B, C, D, or E), it is required to pass Part I of the exam. For IS1500 students, it is required to get 36 points or more on Part I to pass the exam. IS1200 students should not answer question 1 in Part I. For IS1200 students, it is required to have 30 points or more in total for questions 2-6 on Part I.

Grading scale (For both IS1200 and IS1500):
• A: 41-50 points on Part II
• B: 31-40 points on Part II
• C: 21-30 points on Part II
• D: 11-20 points on Part II
• E: 0-10 points on Part II
• FX: 30-35 points (IS1500) or 25-29 (IS1200) on Part I, and 21-50 points on Part II.
• F: otherwise

Results
• The result will be announced at latest: N/A
• If a student received grade FX, it is possible to request a complementary oral examination that may result in grade E. Such complementary oral examination must be requested by the student. Please send an email to dbro@kth.se at latest: N/A
Part I: Fundamentals

1. Module 1: Logic Design

(a) Assume that the hexadecimal number C5 is a two’s complement number stored in one byte and the current word size is 16-bit. What is then the sign extended number represented as a word? Give the answer as a binary number (2 points).

(b) Consider the following circuit.

Assume that $C_{in} = 1$, $N = 4_{16}$, $A = C_{16}$, and $B = 2_{16}$. What is then the value of $S$? (2 points)

(c) For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions (4 points).

- Statement 1: If an SR Latch has inputs $S = 0$ and $R = 0$, then the output $Q$ can have either value 0 or 1 depending on the previous values of $S$ and $R$.
- Statement 2: D Flip-Flops are edge-triggered and can change their output values asynchronously at any point in time.

For statements 3 and 4, consider the following circuit and truth table.

- Statement 3: The output value marked as a question mark in the truth table is 1.
- Statement 4: The circuit is an example of a Moore machine.

2. Module 2: C and Assembly Programming

(a) Consider the following C program.
```c
#include <stdio.h>
int foo(int *x, int *k, int p){
    int i, s=0;
    for(i=0; i < p / *k; i++){
        s += x[i];
    }
    *k = s / (p / *k);
    return x[i-1] - 1;
}

int main(){
    int v = sizeof(int);
    int a[] = {3,5,12,15,4,3};
    int r = foo(a, &v, sizeof(a));
    printf("%d\n", v, v << r);
    return 0;
}
```

What has been printed to the standard output after that the program finished executing? (2 points)

(b) The MIPS R-type instruction is encoded as follows: op (bits 31 to 26), rs (bits 25 to 21), rt (bits 20 to 16), rd (bits 15 to 11), shamt (bits 10 to 6), funct (bits 5 to 0). The translation between register names and encoded numbers (in decimal form) is as follows: Names $t0$-$t7$ maps to numbers $8$ – $15$ and names $s0$-$s7$ maps to numbers $16$ – $23$. The operation of the or instruction, which is an R-type instruction, is as follows: \(R[rd] = R[rs] \| R[rt]\), where symbol \(\|\) is the or operation. The functional opcode for or is 100101 (in binary) and 25 (in hex).

Encode the 32-bit machine code for the instruction

or $t2, s4, s3$

Answer using binary representation or hexadecimal representation (2 points).

(c) For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions (4 points).

- Statement 1: A single addi instruction cannot load a constant into a register, where the constant is larger than 16 bits. To solve this problem, two consecutive addi instructions can be used to load a 32-bit constant into a register.
- Statement 2: Heap memory is in the C programming language allocated by defining local variables within a C function.
- Statement 3: After executing the following code, register $t0$ holds value 4.

```c
addi $t0, $0, 9
xori $t0, $t0, 8
xori $t0, $t0, 8
xori $t1, $t0, 8
sll $t0, $t1, 2
```
• Statement 4: The MIPS code fragment
  
  ```mips
  beq $s0, $s1, L1
  add $t0, $s0, $0
  L1:
  sub $t0, $t0, $s1
  ```

  is potentially a correct translation of the C code fragment

  ```c
  if (i == j) {
    f = i;
  }
  f = f - j;
  ```

  assuming that i is mapped to $s0, j to $s1, and f to $t0.

3. Module 3: Processor Design

   (a) Consider the following MIPS code.

   ```mips
   lw $t1, 0($s2)
   add $t0, $s0, $t0
   sub $t2, $t1, $s0
   ```

   For a five-stage pipelined MIPS processor, can there be any pipeline hazards when executing these three lines of code? If so, which kind of hazard and which instructions are involved? How can this hazards be solved? (2 points)

   (b) Why can a pipelined processor give better performance than a single-cycle processor? (2 points)

   (c) Consider the following figure that depicts a single cycle MIPS processor. The translation between register names and encoded numbers (in decimal form) is as follows: Names $t0$−$t7$ maps to numbers 8 − 15 and names $s0$−$s7$ maps to numbers 16 − 23.

   For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions. (4 points)
• Statement 1: If the current instruction that is executed is `lw $t2, 8 ($s2)`, then the control unit will set signal `ALUSrc` to 1.

• Statement 2: Assume that the current instruction that is executed is `j foo`, where `foo` is a label for an address. Then the 16 least significant bits of the instruction is sign extended to 32 bits to form the new target jump address, and the `Branch` control signal is 1.

• Statement 3: Assume that the current instruction that is executed is `add $t4, $s0, $t2`. Then the 5-bit input `A3` of the register file has value 10.

• Statement 4: If an R-type instruction that writes to the register file is executed, the control signal `MemToReg` must be 0, so that the result from the ALU bypasses the data memory.

4. Module 4: Memory Hierarchy

(a) Assume that we have an $N$-associative cache with a word size of 32-bit, a capacity of 4096 bytes, a block size of 8 words, and in total 32 sets. What is then the degree of associativity of the cache? (2 points)

(b) Assume that we have a direct mapped data cache with word size 32-bit and that the number of blocks ($B$) is 256 and the block size 4 words. Assume that the cache is initially empty. Consider the following MIPS code:

```
li   $t4, 0x02232130
li   $t5, 0x03f4a134
addi $t0, $0, 75
loop:  
    beq $t0, $0, done
    lw  $t1, 0($t4)  
    lw  $t2, 4($t5)  
    addi $t0, $t0, -1
    j   loop
done:
```
Note that the first instruction `li` (load immediate) is a pseudo instruction and is encoded using `lui` and `ori` if the immediate value is larger than 16 bits. What is the miss rate for the data cache when executing the code fragment? (2 points)

(c) For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions. (4 points)

- Statement 1: The main reason for having a multi-level cache is to have faster smaller caches closer to the processor and larger, but slower caches further down in the memory hierarchy.
- Statement 2: A write-back policy is a simple way to handle memory writes in a cache, where data is always written back to the main memory directly when data is written to the cache.
- Statement 3: Assume that we have a virtual memory, where the page size is 1024 bytes. The virtual address is 32-bit and the physical address is 28-bit. A page table is then used to translate 20-bit virtual page numbers into 16-bit physical page numbers.
- Statement 4: One major difference between DRAM and SRAM memories are that DRAM memories need to be periodically refreshed, whereas SRAM memories do not need any refresh.

5. Module 5: I/O Systems

(a) Draw a 7-segment display and give each segment clear numbers ranging from 0 to 6. Show the C code that can be used for displaying number ’1’, assuming that the display is I/O mapped to address `0xff34`. A bit value of 1 means that a segment emits light, and a bit value 0 means that the no light is emitted (2 points).

(b) Assume that a 32-bit ISA is using an exception vector with the software interrupt located at address `0x0000 000C` and an IRQ located at address `0x0000 0010`. The exception handler routines for software interrupts and IRQ are located at addresses `0x00ff 0000` and `0x00bb 0000`, respectively. Explain what data should be located in the exception vector for software interrupts and IRQ. (2 points)

(c) For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions. (4 points)

- Statement 1: Timers can be used to trigger a system to perform a task periodically.
- Statement 2: UART and RS-232 are used for fast parallel transmission of data between computers that are located far away from each other.
- Statement 3: Software interrupts (also called traps) can be used for performing systems call in an operating system.
- Statement 4: DMA can be used for fast data transfer between memory and I/O systems without the intervention by the processor.
6. Module 6: Parallel Processors and Programs

(a) Assume that we have a multiprocessor with 10 cores. The goal is to parallelize parts of a program so that we can achieve 5 times speedup. When the program is executed on just one of the cores, it takes 100 seconds to complete. In percentage, how large part of the execution on one core must be possible to parallelize to achieve this speedup, assuming that we can achieve linear speedup when parallelizing parts of the program on 10 cores? Answer as a rational number. For instance, the rational number \( \frac{100}{3} \% \) is the same as 33.333 . . . % (2 points).

(b) Assume that the following MIPS code is executed using an out-of-order processor.

\[
\begin{align*}
\text{lw} & \quad \$t1, 16($t0) \\
\text{sub} & \quad \$t0, \$t3, 10
\end{align*}
\]

What kind of dependency is this and what is the dependency? (2 points)

(c) For each of the following statements, answer if the statement is true, false, or “don’t know”. See the first page of the exam for an explanation of how the points are counted for these true/false/don’t know questions. (4 points)

- Statement 1: An out-of-order processor can rearrange the order that instructions are executed dynamically, as long the result of the program execution does not change.

- Statement 2: Data-level parallelism and MIMD have much in common, especially in that one stream of instructions is used to operate on multiple instances of data.

- Statement 3: On of the main uses of hardware multithreading is to increase throughput by switching threads when the processor needs to wait on, for instance, memory requests.

- Statement 4: If two cores in a multicore processor are reading from the same memory address in the main memory, and one of the cores always gets the data much faster than the other core, we have an example of a cache coherence problem.
Part II: Advanced

1. Construct the data path and draw a detailed diagram for a single-cycle MIPS processor that supports the following functionality:
   - Store word instruction
   - Load word instruction
   - PC increment

   For each of the instructions and the PC increment functionality, explain clearly how the instruction memory, the register file, the ALU, and the data memory interacts. Clearly motivate your answers. (15 points).

2. Explain clearly the difference between a 2-way set associative cache and a directly mapped cache. Your explanation should include at least the following:
   - The meaning of capacity, number of sets, block size, number of blocks, and degree of associativity.
   - A short example for each of the two cache types that illustrates the differences.
   - The meaning of temporal and spatial locality.

   Clearly motivate your answer (15 points).

3. Construct a C program and the corresponding MIPS or NIOS II assembly program that sorts an array of integers in increasing order. The programs should be documented and explained in detail (10 points).

4. Explain what the purpose of dynamic branch prediction, how a 2-bit prediction schema works, and why it is especially important for deep pipelines. Explain by giving a short assembly code example. Motivate your answer clearly. (10 points)