CDT204 – Computer Architecture

Date: Jan. 7th 2016

Time: 8:10 – 13:30

Help: Small calculator is allowed in the exam

The exam has 33 points. The grades will be awarded as follows:

- 3 : 17 Points
- 4 : 22 Points
- 5 : 27 Points

Important Notes:

- Give as full an answer as possible to obtain full marks. All calculations, approximations, assumptions and justifications must be reported for full credit unless stated otherwise. Please use figures and examples to clarify.
- If you do not understand a question clearly, you are allowed to call the teacher and ask.
- Write the question and part number on each page clearly.
- Answer each question on a separate page.
- Bonus questions will give you extra points (they raise the total over 33) ***if*** you can solve them.

May the force be with you.

Task 1 – General (3p)

Consider the following binary representation: 1101 1001

a) What is the hexadecimal representation of that number? (1p)
b) What is its value in decimal form (base 10) if it contains an unsigned integer? (1p)
c) What is its value in decimal form (base 10) if it contains a two's compliment signed integer? (1p)

Task 2 – Performance (8p)

When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.

Assume a program requires $t = 100s$ of execution time on one processor. When run p processors, each processor requires t/p seconds, as well as an additional $4s$ of overhead, irrespective of the number of processors.

a) Compute the per-processor execution time for 2, 4, 8, 16, 32, 64, and 128 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead). (5p)
b) Draw a chart with number of processors on x-axis and the speedup on the y-axis. Plot both actual speedup and ideal speedups that you calculated for the previous part into this graph. Discuss your findings. (3p)

Task 3 – MIPS Assembly (4p)

For the following MIPS assembly instructions below:

```
sll     $t0, $s0, 2
add     $t0, $s6, $t0
sll     $t1, $s1, 2
add     $t1, $s7, $t1
lw      $s0, 0($t0)
addi    $t2, $t0, 4
lw      $t0, 0($t2)
add     $t0, $t0, $s0
sw      $t0, 0($t1)
        ...
```

a. What is the corresponding C statement? Assume that the variables *f*, *g*, *h*, *i* and *j* are assigned to registers *$s0, $s1, $s2, $s3*, and *$s4*, respectively. Assume that the base address of the arrays *A* and *B* are in the registers *$s6* and *$s7*, respectively. (3p)
b. If the variables have the following values before the code runs, what will be their values after execution the code? *f=1, g=2, h=0, i=0, j=1, A={32, 10, 7}, B={99,99,12}* (1p)

NOTE: The whole MIPS code translates to only one C statement, not a C program. So the final result for "part a" should be just a C statement, although you must show step-by-step how reached to that conclusion.

Task 4 – Pipelining (5p)

Assume the following five steps MIPS data path and their latencies

Pipeline stage	Description	Latency (ps)
IF	Instruction Fetch – Get the next instruction from memory	150
ID	Instruction Decode – Decode the op-code and read the register file	300
EX	Execute – Perform arithmetic operations / compare commands / calculate memory addresses	100
MEM	Memory – Read from or write to memory	400
WB	Write Back – Write back the result to the register file	200

Also assume that the instructions executed by the processor are broken down as follows

ALU	BEQ	LW	SW
45%	10%	25%	20%

a. What is the clock cycle time and frequency in a pipelined and non-pipelined processor? (1p)
b. What is the total latency of an LW instruction in a pipelined and non-pipelined processor? (1p)
c. If we can split one stage of the pipelined data path into two new stages, each with half the latency of the original stage, which stage would you split and what are the new clock cycle and frequency of the processor? (1p)
d. Assuming there are no stalls or hazards, what is the utilization of the data memory? (1p)
e. Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit? (1p)

Task 5 – Cache Performance (4p)

Consider the following memory hierarchy:

Level	Seek Time	Miss Rate
L1-Cache	0.5ns	20%
L2-Cache	1.8ns	5%
L3-Cache	4.2ns	1.5%
Main Memory	70ns	0%

Seek times given refer to the total time it takes to both check whether the requested data is available on the current level of hierarchy, and transmit the data to the level above (or to the CPU). Miss Rate is local, which means that it only applies to accesses that reach the current level because they have missed in all the levels above it.

(a) What is the average time for a memory access? (2p)

(b) What would the average memory access time be if we remove the L3 cache? (2p)

Task 6 – Virtual Memory (6p)

Suppose that you have a page size of 4 KB. Also suppose that you have an application in which the text segment (program code) takes up 2 KB, the data segment (global variables, etc.) takes up 5 KB, and that 1 KB allocated for the stack.

a. We do not know where in the process's private memory space these three segments are allocated, but we know that they are in 3 distinct address ranges. What is the maximum number of pages that this program might need? (1p)

Now assume that we know that the code segment is allocated in address 1000_{16} to $17FF_{16}$, data segment from 10000_{16} to $113FF_{16}$ and the stack from $FFC00_{16}$ to $FFFFF_{16}$ in a 1MB virtual memory. After the program runs for a while, some parts of the page table look like the following:

Valid	Dirty	Virtual Page No.	Physical Page No.
1	0	00010000	001100101
1	0	00000001	000110011
1	1	11111111	110110100
0	0	00000001	111001000
1	1	00010001	000110101

b. Considering that the virtual memory is 1MB in size, how many bits the virtual addresses should be? (1p)
c. Judging by the physical page numbers, how big is the physical memory? (1p)
d. What page number the code segment has got? (The whole segment is placed in one page) (2p)
e. Is all of the data segment read into the physical memory? (2p)

Task 7 – Multiprocessing (3p + 2p BONUS)

Flynn's taxonomy in its classical form (represented in the table below) defines 4 different classes in computer architecture from a multiprocessing point of view.

	Single Instruction	Multiple Instruction
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD

a. Explain each of the classes and give examples for each of them. (2p)
b. Some researchers have proposed 2 newer classes called SPMD (Single Program Multiple Data) and MPMD (Multiple Programs Multiple Data). SPMD defines a design in which multiple processors execute the same program at independent points on different data at the same time. Can you give an example of hardware which has an architecture very similar to SPMD? (1p)
c. *BONUS QUESTION (+2 Points):* NIVIDIA classifies its architecture for Tesla and Fermi line of products as SIMT (Single Instruction Multiple Threads), compare this design with classical SIMD and/or Vector architecture (for example Intel AVX).