Written test on 13^{th} of December 2017

Course: Computer Architecture and Engineering Course no. 02155

Exam time: 2 hours

No aids – Allowed: pocket calculator

Weighting: Each of the 5 problems counts 20%. All questions in a problem are weighted equally.

Answers are evaluated based on correctness, completeness and conciseness (i.e., answers must be correct, show how the result is obtained, and contain only the necessary text).
If some specification is missing, just state your own specification and continue the problem according to your stated specification.

Problem 1

Answer the following questions.

Question 1.1
How long are RISC-V instructions?

Question 1.2
What is forwarding good for?

Question 1.3
Which of the following is true?

1. Register fields are always in the same position in the instruction to simplify pipelining.
2. RISC-V can add two numbers that are in memory in a single instruction.
3. A single-cycle datapath is the best performing implementation of RISC-V.

Question 1.4
Why do caches work?

Question 1.5
Which of the following is true?

1. Fully associative caches are the caches with the simplest implementation.
2. An instruction set simulator of RISC-V can boot Linux.
3. Pipelining improves latency of instructions compared to non-pipelining implementations.

Problem 2

Consider the following RISC-V processors:

P1: single cycle processor (each instruction is executed in one clock cycle) with $T_{clock} = 10\ ns$.

P2: 5-stage pipelined processor (stages: F, D, E, M, W) without data-forwarding, with $T_{clock} = 2\ ns$. Branches are assumed not-taken and the decision on taking the branch is made at the end of stage D.

P3: as **P2**, but with data-forwarding implemented.

The processors are used to execute the following fragment of RISC-V code.

```
main:
        addi a2, x0, 5
        lw a1, 0(a0)
        addi a1, a1, 5
        sw a1, 0(a0)
        add a1, x0, a2
        beq a1, x0, branch1
        beq a1, a2, branch2
branch1:
        add a1, a1, a1
        add a2, a2, a2
branch2:
        addi a1, a1, 6
        addi a2, a2, 7
```

Question 2.1

For each processor, show the timing diagram[1] (instructions executed in each clock cycle) of the execution of the above fragment of RISC-V code.

Question 2.2

For each processor, compute the execution time of the fragment of RISC-V code.

Question 2.3

What is the speed-up of **P3** over **P1** and **P2**?

[1] For the timing diagram you should use the following compact representation, for example:

```
                1 2 3 4 5 6 7 ...
addi a2, x0, 5  F D E M W
lw a1, 0(a0)      F D E M W
...                 ...
```

 ... continued on the next page

Problem 3

A single core processor can operate at the following Voltage Scaling (VS) and Frequency Scaling (FS) modes:

		High-perf.	Medium	Low-power
frequency	[GHz]	4	3	2
supply V_{DD}	[V]	1.0	0.9	0.8

Its Floating-Point (FP) unit can issue per clock cycle:

- 1 double-precision (binary64) FP operation;
- 2 single-precision (binary32) FP operations in parallel;
- 4 half-prexision (binary16) FP operations in parallel.

For all three types of operations, the FP-unit latency is the same.

We run a C program on the processor and we want to determine its performance and power efficiency under the three types of FP operations (and operands).
Assume that

- the execution of the program requires 10^9 cycles when using binary64 FP operations;
- 60% of the cycles are spent FP operations;
- All FP operations are independent.

Question 3.1

In the High-performance ($f = 4$ GHz, $V_{DD} = 1$ V) mode, determine the execution time for the three types of FP operations and the speed-ups with respect to the double-precision case.

Question 3.2

The processor's power dissipation is 10 W when operating in High-performance ($f = 4$ GHz, $V_{DD} = 1$ V) mode. We assume that the power dissipation in the FP-unit stay constant when executing 1 binary64 operation, or 2 binary32 operations or 4 binary16 operations.

In the half-precision (binary16) execution, we apply frequency and voltage scaling to reduce the power dissipation. Determine:

a) the minimum power dissipation that can be achieved by maintaining a speed-up over the double-precision (binary64) execution in High-performance mode;

b) the improvement in energy efficiency over the double-precision (binary64) execution in High-performance mode.

... continued on the next page

Problem 4

Consider the following caches. For both, assume 4KB size, 32-bit addresses, and byte addressing:

C1: direct mapped with 8 word blocks and write-through policy.

C2: 4-way set associative with 4 word blocks, LRU replacement, and write-back policy.

For each cache:

Question 4.1
Show how binary addresses are divided into tag, index, block offset, and byte offset.

Question 4.2
Calculate the total number of bits required for the caches.

Question 4.3
For each memory access below, state whether the access is a hit or miss:

	C1				C2			
Address	Tag	Index	Offset	Hit/Miss	Tag	Index	Offset	Hit/Miss
0x0022								
0x0023								
0x0034								
0x0420								
0x1420								
0x0422								
0x1430								

Question 4.4
Calculate the hit ratio.

... continued on the next page

Problem 5

Consider three systems with three different cache configurations:

- Cache 1: Direct-mapped with one-word blocks
- Cache 2: Direct-mapped with four-word blocks
- Cache 3: Two-way set associative with four-word blocks

For the three configurations, the following miss rate measurements have been made:

- Cache 1: Instruction miss rate is 4%; data miss rate is 9%.
- Cache 2: Instruction miss rate is 2%; data miss rate is 5%.
- Cache 3: Instruction miss rate is 2%; data miss rate is 3%.

For these machines, 40% of the instructions contain a data reference. Assume that the cache miss penalty is 6 + Block size in words. The CPI for this workload was measured on a machine with Cache 1 and was found to be 2.0.

Question 5.1

Determine which machine spends the most cycles on cache misses.

End of the Exam