CompArch: exam

Renaud Pacalet

7 February 2020

You can use any document you need. Please number the different pages of your work and indicate on each page your first and last names. Write your answers in French or in English, as you wish, but avoid mixing the languages. Precede your answers with the question's number. If some extra information or hypotheses are missing to answer a question or solve a problem, decide by yourself and write down the added hypotheses or information. If you consider a question as absurd and thus decide not to answer, explain why. If you do not have time to answer a question or solve a problem but know how to, briefly explain your ideas.

Important advice #1: quickly go through the document and answer first the easy parts.

Important advice #2: copying verbatim the slides of the lectures or any other provided material is not considered a valid answer.

1 Representation of numbers (4 points)

1.1 Definitions and notations

- There are several ways to represent signed integers using bits. In computer systems, the two most frequently encountered are *sign and magnitude* and *two's complement*.
- We denote $u_{n-1}u_{n-2}\dots u_1u_0$ the n-bits representation of an integer U.
- In both representations u_0 is the **Least Significant Bit (LSB)**.
- In *sign and magnitude* representation u_{n-1} is the **sign** bit.
- In *two's complement* representation u_{n-1} is the **Most Significant Bit (MSB)**.

1.2 Problems

1.2.1 [2 points] Integers representation in *sign and magnitude*

1. What is the **minimum** number of bits m_1, including the sign bit, to represent 2047 in *sign and magnitude*?
2. What is the **minimum** number of bits m_2, including the sign bit, to represent -1023 in *sign and magnitude*?
3. What is the **minimum** number of bits m_3, including the sign bit, to represent -2024 in *sign and magnitude*?

4. Convert 2047 in m-bits *sign and magnitude* where $m = max(m_1, m_2, m_3)$.

5. Convert -1023 in m-bits *sign and magnitude* where $m = max(m_1, m_2, m_3)$.

6. Convert -2024 in m-bits *sign and magnitude* where $m = max(m_1, m_2, m_3)$.

1.2.2 [1 points] Overflow in sign and magnitude

When adding two n-bits integers represented in *sign and magnitude* representation how can we detect an overflow?

1.2.3 [1 points] Overflow in two's complement

When adding two n-bits integers represented in *two's complement* representation how can we detect an overflow?

2 Branch prediction (6 points)

2.1 Definitions and notations

- **Miss-Prediction per executed Branch Instruction (MPBI)**: the ratio between the number of times a given branch instruction has been wrongly predicted and the total number of times this same branch instruction has been executed. The lower the MPBI, the better the predictor.

- **Branch outcome**: the actual decision (**not the prediction**) for a given branch instruction; Taken or Not taken, denoted `T` and `N`, respectively.

- **Periodic infinite sequence of outcomes**: a sequence of outcomes that starts with a finite sequence, the stem, which can be empty, and continues with a finite cycle that repeats infinitely. We represent these sequences as `STEM(CYCLE)`$\star$. Example: `TNTN NNNTTTT NNNTTTT NNNTTTT`... is a periodic infinite sequence of outcomes, its stem is `TNTN`, its cycle is `NNNTTTT` and we represent it as `TNTN(NNNTTTT)`$\star$.

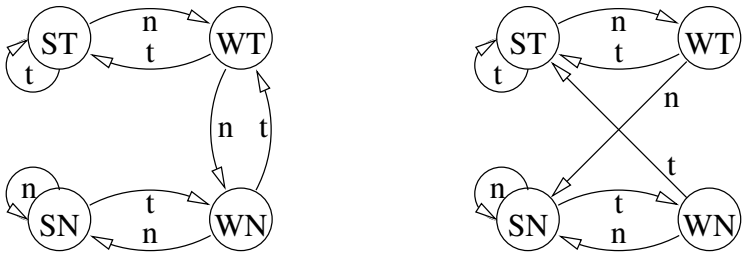
2.2 Problem

We consider the two different 4-states branch predictors studied during the lecture on pipelines: the 2-bits saturating counter and a variant.

We use these two predictors to predict a single branch instruction `B`, always the same. We measure their efficiency with their MPBI for instruction `B`. We will assume that both predictors are initialized in the Strong Taken (ST) state.

1. **[**1.5 **points]** Imagine a periodic infinite sequence of branch `B` outcomes such that the MPBI of the variant is significantly greater (that is, worse) than the MPBI of the 2-bits saturating counter. What is the MPBI of both predictors with this sequence?

1. **[**1.5 **points]** Imagine a MIPS assembly code snippet that would produce such a sequence. Label the branch `B` instruction of interest `B:`. Use the provided MIPS reference card on assembly language syntax.



(a) Saturating counter (b) Variant

Figure 1: Two 4-states branch predictors

2. **[**1.5 **points]** Imagine another periodic infinite sequence of branch `B` outcomes such that the MPBI of the 2-bits saturating counter is significantly greater than the MPBI of the variant. What is the MPBI of both predictors with this sequence?

1. **[**1.5 **points]** Imagine a MIPS assembly code snippet that would produce such a sequence.

3 Caches (6 points)

3.1 Definitions and notations

- The **breakdown** of a data structure is the partitioning of the data structure in individual fields and, for each field, its bit-width and a description of its role.
- In a set-associative cache the lines in a set are numbered starting from 0. When populating an empty set we assume that the lines are populated in increasing order of their number: line 0 first, then line 1...
- *Extra cache data*: control and management information stored in the cache (tags, flags, replacement policy information...)
- *Net cache data*: copies of memory data stored in the cache, that is, everything except *extra cache data*.

3.2 Cache architecture

We consider a computer system with 32 bits addresses and data. A cache is used to improve the performance. The total net cache data capacity of the cache is 8 k-Bytes. Its other characteristics are:

- Four 32-bits words per line.
- 2-ways set-associative.
- Write-back.
- Write-allocate.
- Least Recently Used (LRU) replacement policy.

1. **[**1/2 **points]** What is the breakdown of a 32-bits address?
2. **[**1/2 **points]** How many bits per set are needed to handle the LRU replacement policy?
3. **[**1/2 **points]** Explain how the LRU policy works and what use it makes of the bits you decided to use.
4. **[**1/2 **points]** What is the breakdown of a cache line?
5. **[**1/2 **points]** What is the breakdown of a cache set?
6. **[**1/2 **points]** What is the total size in bits, including extra cache data, of the cache?

3.3 Cache operations

Starting from an empty cache (all lines invalid), the cache receives the following sequence of consecutive 32-bits word read accesses from its processor:

```
B23C6694, 95CFF69C, 3487351C, B23C6698,
3487369C, B23C6514, 95CFF518, 3487369C,
B23C6518, 95CFF67C, 3487369C, B23C6694,
95CFF670, 95CFF690, B23C6514, 34873690
```

The byte addresses are given in hexadecimal. The sequence must be read left to right first and then top to bottom: the first access is at `B23C6694`, the second at `95CFF69C`... the last at `34873690`.

[3 **points]** Simulate the cache for this sequence of read accesses and write a 16 lines table showing how the cache behaves. The format of the table must be:

Address	Hit/Miss	Operation
AAAAAAAA	X	O
...	...	...

where:

- `AAAAAAAA` is the **hexadecimal** address that was read.
- `X` is `H` for hit or `M` for miss.
- `O` is `U` if some net cache data is updated without eviction, `E` if some net cache data is updated with eviction, `N` if no net cache data is updated.

Example: `12345678 H N` means that the read address was `12345678`, it was a hit and the net cache data has not been updated. If you wish, and if it helps, you can add other columns on the right to represent other information. If you do, add the corresponding header cells to name each extra column and add some text to explain what these extra columns represent.

4 MIPS 5-stages pipeline (4 points)

In a 5-stages MIPS pipeline we run the following assembly program:

```
    sw   $s0, 0($t3)        # Mem[0+$t3] <- $s0
    or   $s0, $s0, $s7      # $s0 <- $s0 or $s7
    lw   $t0, 0($s0)        # $t0 <- Mem[0+$s0]
    bne  $t6, $s0, label    # if($t6 != $s0) goto label
    lw   $t9, 0($t9)        # $t9 <- Mem[0+$t9]
    xori $s0, $s0, 0xff     # $s0 <- $s0 xor 0xff
label:
    ...
```

1. **[2 points]** Identify the various hazards.
2. **[2 points]** For each hazard:

a. In which class of hazards does it fall?

b. Which technique is the best to deal with it?