CS61C Spring 2025 — Instruction Translation, AMAT — Discussion 5

1 RISC-V Instruction Translation

1.1 In this question, translate the following RISC-V instructions into their binary and hexadecimal values.

a) `addi s1 x0 -24` = 0b_____ (1.1)

= 0x_____ (1.1)

b) `sh s1 4(t1)` = 0b_____ (1.1)

= 0x_____ (1.1)

1.2 In this question, translate the following hexadecimal values into RISC-V instructions.

a) `0xFE05 0CE3` = _____ 1.2

b) `0x2345 54B7` = _____ 1.2

1.3 Given the following RISC-V code and instruction addresses, translate the **`jal`** and **`bne`** instructions (you'll need your RISC-V reference sheet!) and determine the value of **`R[ra]`** during the execution of **`loop`**.

```
      loop:
0x002CFF00:     add t1, t2, t0        0x00538333

0x002CFF04:     jal ra, foo           _____ 1.3

0x002CFF08:     bne t1, zero, loop    _____ 1.3
                ...
      foo:

0x002CFF2C:     jr ra         R[ra] = _____ 1.3
```

2 RISC-V Addressing

We have several *addressing modes* to access memory (immediate not listed):

a) Base displacement addressing adds an immediate to a register value to create a data memory address (used for `lw, lb, sw, sb`).

b) PC-relative addressing uses the PC and adds the immediate value of the instruction to create an instruction address (used by branch and jump instructions).

c) Register Addressing uses the value in a register as an instruction address. For instance, `jalr, jr,` and `ret`, where `jr` and `ret` are just pseudoinstructions that get converted to `jalr`.

2.1 What is the range of 32-bit instructions that can be reached from the current PC using a single branch instruction? Note that RISC-V branch instructions must support branching to 16-bit "compressed" instructions (enabled via an optional RISC-V extension).

2.2 What is the maximum range of 32-bit instructions that can be reached from the current PC using a jump instruction?

3 Caches Intro: AMAT

Recall that AMAT stands for Average Memory Access Time. The main formula for it is:

$$\text{AMAT} = \text{Hit Time} + \text{Miss Rate} * \text{Miss Penalty}$$

3.1 Suppose your system takes 100ns to access main memory. We decide to add a cache with a measured hit time of 25ns and miss rate of 25%. What is the average memory access time of the system?

3.2 In a new 2-level cache system, after 100 total accesses to the cache system, we find that the L2$ (L2 Cache) ended up missing 20 times. What is the global miss rate of L2$?

3.3 Given the system from the previous subpart (100 total accesses, 20 L2$ misses), if L1$ had a local miss rate of 50%, what is the local miss rate of L2$?

For the following subparts, suppose we have a new system that consists of:

1. An L1$ that has a hit time of 2 cycles and a local miss rate of 20%
2. An L2$ that has a hit time of 15 cycles and has a global miss rate of 5%
3. Main memory where accesses take 100 cycles

EDIT 2/26: the original worksheet listed L2$ hit time as 16 cycles. The correct L2$ hit time is 15 cycles.

3.4 What is the local miss rate of L2$?

3.5 What is the AMAT of the system?

3.6 Suppose we want to reduce the AMAT of the system to 8 cycles or lower by adding in a L3$. If the L3$ has a local miss rate of 30%, what is the largest hit time that L3$ can have?