Homework 3

Md Ali
CS 402: Advanced Studies II

October 16th, 2022

Exercise 2.12.1.

Proof. Taking the following assembly code:

add \$t0, \$s0, \$s1

$\boxed{\text{We get the theoretical value of \$t0 to be 0x150000000 but we don't have that much so we get \$t0 = 0x50000000}}$

□

Exercise 2.12.2.

Proof. $\boxed{\text{No, an overflow has occurred due to \$t0 had to be 0x150000000 but this is above the allowed value hence we get an overflow and left with 0x50000000.}}$

□

Exercise 2.12.3.

Proof. Taking the following assembly code:

sub \$t0, \$s0, \$s1

$\boxed{\text{Hence, we get that \$t0 = 0xB0000000}}$

□

Exercise 2.12.4.

Proof. $\boxed{\text{We don't get an overflow in this case, hence we do get the desired results.}}$

□

Exercise 2.12.5.

Proof. Taking the following assembly code:

```
add $t0, $s0, $s1
add $t0, $t0, $s0
```

We get the theoretical value of \$t0 to be 0x1D0000000 but we don't have that much so we get \$t0 = 0xD0000000

□

Exercise 2.12.6.

Proof. No, an overflow has occurred due to \$t0 had to be 0x1D0000000 but this is above the allowed value hence we get an overflow and left with 0xD0000000.

□

Exercise 2.19.1.

Proof. Taking the following assembly code:

```
sll $t2, $t0, 44
or $t2, $t2, $t1
```

We get the value of \$t2 to be 0xBABEFEF8

□

Exercise 2.19.2.

Proof. Taking the following assembly code:

```
sll $t2, $t0, 4
or $t2, $t2, -1
```

We get the value of \$t2 to be 0xAAAAAAA0

□

Exercise 2.19.3.

Proof. Taking the following assembly code:

```
srl $t2, $t0, 3
or $t2, $t2, 0xFFEF
```

We get the value of \$t2 to be 0x00005545

□

Exercise 2.20.

Proof. Below is the sequence:

```
srl $t0, $t0, 11
sll $t0, $t0, 26
ori $t2, $0, 0x03FF
sll $t2, $t2, 16
ori $t2, $t2, 0xFFFF
and $t1, $t1, $t2
or $t1, $t1, $t0
```

□

Exercise 2.22.

Proof. Below is the sequence:

```
lw $t3, 0($s1)
sll $t1, $t3, 4
```

□

Exercise 2.23.

Proof. $\boxed{\text{The value of \$t2} = 3 \text{ after the execution of the instructions}}$

□

Exercise 2.26.1.

Proof. $\boxed{\text{The value of \$s2 will be 20 after the execution of the instructions}}$

□

Exercise 2.26.2.

Proof. The C code equivalent is below:

```
i = 10;
do {
B + = 2;
i − = 1;
} while(i > 0)
```

□

Exercise 2.26.3.

Proof. $\boxed{5 \cdot N \text{ instructions will be executed}}$

□

Exercise 2.27.

Proof. The MIPS code is below:

```
addi $t0, $0, 0
beq $0, $0, TEST1
LOOP1: addi $t1, $0, 0
beq $0, $0, TEST2
LOOP2: add $t3, $t0, $t1
sll $t2, $t1, 4
add $t2, $t2, $s2
sw $t3, ($t2)
addi $t1, $t1, 1
TEST2: slt $t2, $t1, $s1
bne $t2, $0, LOOP2
addi $t0, $t0, $s0
TEST1: slt $t2, $t0, $s0
bne $t2, $0, LOOP1
```

□

Exercise 2.28.

Proof. $\boxed{\text{It takes at least 14 instructions and a total of 158 MIPS instructions to execute}}$ □

Exercise 2.38.

Proof. After excuting the first instruction, the contents of \$t0 will be 0x00000011.

$\boxed{\text{Hence, we get that the value that is stored in \$t2 is 0x00000011}}$

□

Exercise 2.39.

Proof. Below is the code:

```
lui $t1, FRONT-16
ori $t1, END-16
```

□

Exercise 2.46.1.

Proof. We are given a scenario where we have an original CPI and then seeing if with some customization if the new "improvements" indeed increase performance.

First, let's calculate the original CPI. We get the following:

$$\text{CPI}_{Original} = 1 \cdot \frac{5 \cdot 10^8}{9 \cdot 10^8} + 10 \cdot \frac{3 \cdot 10^8}{9 \cdot 10^8} + 3 \cdot \frac{1 \cdot 10^8}{9 \cdot 10^8}$$

$$\text{CPI}_{Original} = \frac{38}{9} \approx 4.22$$

So we end up with the initial execution to be:

$$\text{Initial Execution} = \frac{38}{9} \cdot 9 \cdot 10^8 \cdot f = 3.8 \cdot 10^9 \cdot f$$

Now, let's look at the "improvements" that were made. First we have to calculate the new number of arithmetic instructions involved.

$$\text{New Arithmetic Instructions} = (5 \cdot 10^8) \cdot (0.75) = 3.75 \cdot 10^8$$

This gives our new total to be:

$$\text{New Total Instructions} = 3.75 \cdot 10^8 + 3 \cdot 10^8 + 1 \cdot 10^8 = 7.75 \cdot 10^8$$

With the following values we get our new CPI to be:

$$\text{CPI}_{New} = 1 \cdot \frac{3.75 \cdot 10^8}{7.75 \cdot 10^8} + 10 \cdot \frac{3 \cdot 10^8}{7.75 \cdot 10^8} + 3 \cdot \frac{1 \cdot 10^8}{7.75 \cdot 10^8}$$

$$\text{CPI}_{New} = \frac{36.75}{7.75} \approx 4.74$$

So we end up the new execution to be:

$$\text{New Execution} = \frac{36.75}{7.75} \cdot 1.1 \cdot 7.75 \cdot 10^8 \cdot f = 4.0425 \cdot 10^9 \cdot f$$

Hence we can see that the older execution time is faster, so the new "improvement" are actually slower in comparison. The new design is a poor design choice for this computing scenario.

□

Exercise 2.46.2.

Proof. Now we are asked if we doubled the performance of the arithmetic instructions what would the speed up be, as well as what the speed up would be if we 10 times the performance of the arithmetic instructions. So now we have that the CPI below for doubling the performance of the arithmetic instructions:

$$\text{CPI} = 0.5 \cdot \frac{5 \cdot 10^8}{9 \cdot 10^8} + 10 \cdot \frac{3 \cdot 10^8}{9 \cdot 10^8} + 3 \cdot \frac{1 \cdot 10^8}{9 \cdot 10^8} = \frac{35.5}{9} \approx 3.94$$

The speed up for doubling the performance of the arithmetic instructions are as follows:

$$\frac{38}{9} \cdot \frac{9}{35.5} = 1.07$$

Hence our speed up is approximately 7% when we double the performance of the arithmetic instructions. Now let's take a look at if we 10x the performance of the arithmetic instructions.

$$\text{CPI} = 0.1 \cdot \frac{5 \cdot 10^8}{9 \cdot 10^8} + 10 \cdot \frac{3 \cdot 10^8}{9 \cdot 10^8} + 3 \cdot \frac{1 \cdot 10^8}{9 \cdot 10^8} = \frac{33.5}{9} \approx 3.72$$

The speed up for 10 times the performance of the arithmetic instructions are as follows:

$$\frac{38}{9} \cdot \frac{9}{33.5} = 1.13$$

Hence our speed up is approximately 13% when we 10 times the performance of the arithmetic instructions.

$$\boxed{\text{Hence with doubling the arithmetic instructions we get a speed up of approximately 7\% and when we 10 times the arithmetic instructions we get a speed up of approximately 13\%}}$$

□

Exercise 2.47.1.

Proof. Taking each component we can get the average CPI to be:

$$\text{CPI}_{average} = 0.7 \cdot 2 + 0.1 \cdot 6 + 0.2 \cdot 3 = 2.6$$

$$\boxed{\text{Hence the average CPI value is 2.6}}$$

□

Exercise 2.47.2.

Proof. If we have a 25% improvement, we must reduce the CPU to:

$$2.6 \cdot 0.75 = 1.95$$

Hence we get a formula to solve for x:

$$1.95 \geq 0.7 \cdot x + 0.1 \cdot 0.6 + 0.2 \cdot 3$$

Solving for x we get that:

$$x = 1.07$$

$$\boxed{\text{Hence we get that the the arithmetic instruction must have a CPI of 1.07 at most for a 25\% improvement}}$$

□

Exercise 2.47.3.

Proof. If we have a 50% improvement, we must reduce the CPU to:

$$2.6 \cdot 0.5 = 1.3$$

Hence we get a formula to solve for x:

$$1.3 \geq 0.7 \cdot x + 0.1 \cdot 0.6 + 0.2 \cdot 3$$

Solving for x we get that:

$$x = 0.14$$

$$\boxed{\text{Hence we get that the the arithmetic instruction must have a CPI of 0.14 at most for a 50\% improvement}}$$

□