Computer Architecture and Programming Languages
Jacobs University Bremen
Jasmine Dayinta

Course: CO20-320241
October 26, 2018

Homework 6- MIPS Instruction Set Architecture

Problem 6.1

Solution:

They are 5 bits wide and not some other value because every MIPS instruction set uses 32 bits and every instruction uses 32 registers. This is then divided up into the following: op - 6 bits, rs - 5 bits, rd - 5 bits, shamt - 5 bits, and funct 6 bits. For rs, rd, and shamt, since they use 5 bits, this means that they represent values from 0 to 31, which makes sense because the maximum that it can be represented is 32 bits. If it was more bits, there wouldn't be enough space to represent that value and if the bits was less, then it would be only represent a very small value.

Problem 6.2

Solution:

a) add $t2, $t0, $t1
b) lw $s2, 4($s1)

Problem 6.3

Solution:

a)

op	rs	rt	rd	shamt	funct
000000	01000	01001	01010	00000	100010

Together: 00000001000010010101000000100010

b)

op	rs	rt	rd	shamt	const
100011	10001	10010	-	-	0000000000000100

Together: 10001110001100100000000000000100

Problem 6.4

Solution:

The value of $t2 after executing the MIPS instructions is the value 1. This is because the first line compares $t0 and $t1, which one is smaller. $t0 is smaller than $t1, which means that 1 is stored in $t2. The second line then checks if $t2 is 0. If true, ELSE would be executed, otherwise DONE is executed. In this case, $t2 is not 0, therefore we jump to DONE, which is empty. Therefore, $t2 has the value 1.

Problem 6.5

Solution:

```
lw    $t0, 24($s0)
add   $t0, $t0, $s1
sw    $t0, 24($s0)
```

Problem 6.6

Solution:

```
lui   $t0, 0000 0000 0010 0011        Loads the most significant 16 bits
ori   $t0, $t0, 0000 0000 0010 0011   Loads the least significant 16 bits
sw    $t0, $s4                        Load the 32-bit constant into $s4
```

Problem 6.7

Solution:

```
main:
loop:
        lw      $t0, 0
        bgt     $t0, 8, exit
        add     $s0, $s0, 3
        add     $t0, $t0, 1
        j loop
exit:
```