Computer Architecture and Programming Languages
Jacobs University Bremen
Arsenij Percov

Course: CO20-320241
October 21th, 2019

Homework 6

Problem 6.1

Solution:

Biggest reason for that is the fact that we have 32 registers on most architectures. In order to represent all 32 possible options we need exactly 5 bits, since 11111 represents 31, and 00000 represents 0.

Problem 6.2

Solution:

a) op = 0, rs = 8, rt = 9, rd = 10, shamt = 0, funct = 34
sub $t2 $t0 $t1

b) op = 0x23, rs = 17, rt = 18, const = 0x4
op = 35_10
lw $s2 4($s1)

Problem 6.3

Solution:

a) op = 0, rs = 8, rt = 9, rd = 10, shamt = 0, funct = 34

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

b) op = 0x23, rs = 17, rt = 18, const = 0x4

op	rs	rt	const
100011	10001	10010	0000000000000100

Problem 6.4

Solution:

$t0 = 0010 0100 1001 0010 0100 1001 0010 0100
$t1 = 0011 1111 1111 1000 0000 0000 0000 0000
After comparing bits from most significant bit to least one, we can notice that 4 most significant bit is bigger at $t1, so $t1 is bigger.
Therefore, slt will set value of $t2 as 1.
Next instruction is beq, and since 1 and 0 are not equal, we go to next instruction, which is to jump to done, ans our final value is 1.

Problem 6.5

Solution:

addi $t0, $0, 6
sll $t0, $t0, 2
add $t0, $t0, $s0 - get adress of A[6]
lw $t1, 0($t0) - load value of A[6]
add $s1, $s1, $t1
sw, $s1, 0($t0)

Problem 6.6

Solution:

Add 16 most significant bits:
0000 0000 0010 0011 = 35_10
lui $s4, 35
Load 16 least significant bits
ori $s4, $s4, 35

Problem 6.7

Solution:

```
        addi $t0, $0, 0
        addi $t1, $0, 8
LOOP:beq $t0, $t1, EXIT
        addi $s0, $s0, 4
        addi $t0, $t0, 1
        j LOOP
 EXIT:
```