Computer Architecture and Programming Languages
Jacobs University Bremen
Fjolla Dedaj

Course: CO20-320241
21st of October, 2019

Homework 6

Problem 6.1

In the register instruction format (used e.g., for add), the fields rs, rt, and rd occupy 5 bits each. Why are they 5 bits wide, and not 4 or 6 or some other value?

Solution:
Because in total, we have 32 registers (0 - 31) and 5 bits are enough to represent them in binary ($31_{10} = 11111_2$) using the GNU MIPS register allocation.

Problem 6.2

Solution:

a) op (*operation*) = 0, rs (*source*1)= 8, rt (*source*2) = 9, rd (*destination*) = 10, shamt (*shift*) = 0, funct (*operation*) = 34

Since funct = 34 and op = 0, we have a subtraction instruction. Also, because 8 = $t0, 9 = $t1 and 10 = $t2, the instruction is as follows:

sub $t2, $t0, $t1

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

$23_{26} = 3 * 16^0 + 2 * 16^1 = 3 + 32 = 35_{10}$
rs (source 1) = 17 = $s1
rt (source 2) = 18 = $s2
op = $0x23 = 35_{10}$ = lw

Thus, the instruction is:

lw $s1, 4($s2)

Problem 6.3

Solution:

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

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

rs $= 8_{10} = 01000_2$
rt $= 9_{10} = 01001_2$
rd $= 10_{10} = 01010_2$
funct $= 34_{10} = 100010_2$

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

op	rs	rt	address
100011	10001	10010	0000 0000 0000 0100

op = 0x23 $= 35_{10} = 100011_2$
rs $= 17_{10} = 10001_2$
rt $= 18_{10} = 10010_2$
const = 0x4 $= 4_{10} = 0000\ 0000\ 0000\ 0100_2$

Problem 6.4

. slt $t2, $t0, $t1
. beq $t2, $0, ELSE
. j DONE
ELSE: addi $t2, $0, 2
DONE:

$t0 $= 0010\ 0100\ 1001\ 0010\ 0100\ 1001\ 0010\ 0100_2 = 613566756_{10}$
$t1 $= 0011\ 1111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000_2 = 1073217536_{10}$

Solution:

slt $t2, $t0, $t1 if(**$t0** $<$ **$t1**) **$t2** = 1 else **$t2** = 0
. Since **$t0** $<$ **$t1**, **$t2** = 1

beq $t2, $0, ELSE if(**$t2** $==$ **$0**) goto ELSE
. Since **$t2** != 0 we don't go to **ELSE** but we jump to **DONE**.
. Thus, the value of **$t2** remains 1.

Problem 6.5

Solution:
Offset: A[i] = 4 * i, thus: A[6] = 4 * 6 = 24.

lw $t0, 24($s0)
add $t0, $t0, $s0
lw $t1, 0($t0)
add $t1, $t1, $s1
sw, $t1, 0($t0)

Problem 6.6

Solution:

lui $s4 35 (load upper immediate)
ori $s4, $s4, 35 (bitwise logical OR)

Problem 6.7

Solution:

. li $t0, 0 $\Longrightarrow$ load immediate for i
. li $t1, 8 $\Longrightarrow$ load immediate for the limit of i
LOOP: beq $t0, $t1, END $\Longrightarrow$ if ($t0 == $t1) go to **END**, else $\rightarrow$ execute the instructions below
. addi $s0, $s0, 4 $\Longrightarrow$ add 4 to a
. addi $t0, $t0, 1 $\Longrightarrow$ add 1 to i
. j LOOP $\Longrightarrow$ jump to the start of the loop again
END: ...