Computer Architecture and Programming Languages
Jacobs University Bremen
Jasmine Dayinta

Course: CO20-320241
November 6, 2018

Homework 7 - MIPS Instruction Set Architecture

Problem 7.1

Solution:

```
my_function:
   addi $sp, $sp, -8          #create space for variable
   sw $s0, 4($sp)             #save register $s0
   move $fp, $sp
   sw $4, 8($fp)
   sw $5, 12($fp)
   lw $2, 8($fp)
   slti $2, $2, 11            #if x < 11
   beq $0, $t0, L1            #jump to L1 if x >= 11
   add $0, $a0, $a1           #x+y

$L1:
   sub $s0, $a0, $a1          #x-y

$L2:
   add $v0, $s0, $0           #return value stored into register
   lw $fp, 4($sp)             #restore for caller
   addi $sp, $sp, 8           #delete 1 item
   jr $ra                     #jump to caller
```

Problem 7.2

Solution:

```
is_more_than_fifty:
   addiu $sp, $sp,-32          #stack to add a variable
   sw $31, 28($sp)            #save register for variables
   jal prod
   move $fp, $sp
   sw $4, 32($fp)            #saving more register for more variables
   sw $5, 36($fp)            #saving more register for more variables
   lw $5, 36($fp)
   lw $4, 32($fp)
   slti $t0, $v0, 51
   beq $0, $t0,0x1
   addi $0, $0,0x00ff
   beq $2, $0, $L1
   b $L2
$L1:
   move $2, $0            #putting 0 to $0
$L2:
   move $sp, $fp
   lw $s0, 24($sp)            #restore $0
   addi $sp, $sp, 32
   jr $31
prod:
   addi $sp, $sp,-8
   sw $fp, 4($sp)
   move $fp, $sp
   mul $3, $2            #multiply a and b
   addiu $sp, $sp,8
   jr $ra                #jump back to caller
```

Problem 7.3

Solution:

```
int i = 0;
while(A[i] != -1){
   i++;
}
```

Problem 7.4

Solution:

PC	Machine Code	Binary representation
60000	0 0 19 9 2 0	000000 00000 10011 01001 00010 000000
60004	0 9 22 9 0 32	000000 01001 10110 01001 00000 100000
60008	35 9 8 0	100011 01001 01000 0000000000000000
60012	4 8 21 2	000100 01000 10101 0000000000000010
60016	8 19 19 1	001000 10011 10011 0000000000000001
60020	2 15000	000010 00000000000011101010011000
60024		

Problem 7.5

Solution:

a)

i) $0x0C000000 = 0C000000_{16} = 0000\ 1100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2$
$= 2^{26} + 2^{27}$
$= 201326592_2$

ii) $0xC4630000 = C4630000_{16} = 1100\ 0100\ 0110\ 0011\ 0000\ 0000\ 0000\ 0000_2$
$= 2^{17} + 2^{19} + 2^{20} + 2^{21} + 2^{24} + 2^{25} + 2^{26} + 2^{28} + 2^{29} + 2^{30}$
$= -1000144896_{10}$

b)

i) $0C000000_{16} = 0000\ 1100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2$
ii) $0xC4630000 = 1100\ 0100\ 0110\ 0011\ 0000\ 0000\ 0000\ 0000_2$

c)

i) It would need to be shifted to the right (sra).
ii) OP code doesn't exist in MIPS assembly language.