

# 《计算机组成》第五章作业(可参考附件二指令集)

(所有答案字体均已加粗)

一、

- (a) 将 10 进制数 35 转换为 8 位二进制数是 **00100011**  
(b) 将二进制数 10010111 转换为 16 进制数是 **0x97**  
(c) 将 10 进制数 -35 转换为 8 位二进制补码是 **11011101**  
(d) 将 8 进制数 156 转换为 10 进制数是 **110**  
(e) 请判断以下两个补码表示的二进制数做二进制加法后是否溢出: **溢出**

➤ 01101110  
➤ 00011010

- (f) 对 8 位 16 进制数 0x88 做符号扩展成 16 位数是: **0xFF88**  
(g) 下列代码段存储在内存中, 起始地址为 0x00012344, 分支指令执行后 PC 的两个可能的值分别是: **0x00012344** 和 **0x00012354**。同时, 请在注释位置用伪代码形式对每条指令做出描述。

➤ loop: lw \$t0, 0(\$a0) # **t0 = mem[a0]**  
➤ addi\$a0, \$a0, 4 # **a0 = a0 + 4**  
➤ andi\$t1, \$t0, 1 # **t1 = t0 & 1**  
➤ beqz \$t1, loop # **if t0 奇数 go to loop**

二、将下列汇编语言指令翻译成机器语言代码, 以 16 进制表示

loop:	addu \$a0, \$0, \$t0	# <u><b>0x00082021</b></u>
	ori \$v0, \$0, 4	# <u><b>0x34020004</b></u>
	syscall	# <u><b>0x0000000C</b></u>
	addi \$t0, \$t0, -1	# <u><b>0x2108FFFF</b></u>
	bnez \$t0, loop	# <u><b>0x1500FFFB</b></u>
	andi \$s0, \$s7, 0xffc0	# <u><b>0x32F0FFC0</b></u>
	or \$a0, \$t7, \$s0	# <u><b>0x01F02025</b></u>
	sb \$a0, 4(\$s6)	# <u><b>0xA2C40004</b></u>
	srl \$s7, \$s7, 4	# <u><b>0x0017B902</b></u>

三、写一个MIPS汇编程序, 要求对内存以“example100”为标签 (label) 的数据段中前100个字 (words) 的数据求和, 并将结果存入紧跟在这100个字之后的内存中。

```
.data
example100: .space 400
.globl main
.text
```

```
main:  
  
la $a0, example100 # Load address pointer  
  
li $t0, 0 # Clear sum  
  
li $t1, 100 # Initialize loop count  
  
loop:  
  
lw $t2, 0($a0) # $t2 = mem(a0)  
  
add $t0, $t0, $t2 # $t0 = $t0 + $t2  
  
addi $a0, $a0, 4 # Inc. address pointer  
  
addi $t1, $t1, -1 # Dec. loop count  
  
bgtz $t1, loop # if ($t1 > 0) branch  
  
sw $t0, 0($a0) # Store the result  
  
li $v0, 10 # End of program  
  
syscall
```

四、写一段MIPS汇编语言代码，将内存中“SRC”标签开始的100个字的一块数据转移到内存中另一块以“DEST”标签开始的空间中。

```
.data  
  
SRC: .space 400  
  
DEST: .space 400  
  
.globl main  
  
.text
```

```
main:  
  
la $a1, SRC #$a1 = &SRC  
  
la $a2, DEST #$a2 = &DEST  
  
li $t0, 100 #$t0 = 100  
  
loop:  
  
lw $t1, 0($a1) #$t1= mem($a1)  
  
sw $t1, 0($a2) #mem($a2) = $t1  
  
addi $a1, $a1,4 #$a1 = $a1+4  
  
addi $a2, $a2,4 #$a2 = $a2+4  
  
addi $t0, $t0,-1 #$t0 = $t0 - 1  
  
bgtz $t0, loop #Branch if $t0 > 0  
  
li $v0, 10  
  
syscall
```

五、写一个MIPS函数ABS，通过\$a0传入一个32位整数，将这个数的绝对值存回\$a0。再写一段主程序，调用两次ABS并输出结果，每次传给ABS的数不同。

```
.text  
  
ABS: bgez $a0, return # if ($a0 >= 0) done  
  
sub $a0, $0, $a0 # $a0 = 0 - $a0  
  
return: jr $ra # return  
  
.globl main  
  
.text  
  
main: li $a0, -8765
```

```
jal ABS
li $v0, 1 # Output result
syscall
li $a0, 4321
jal ABS
li $v0, 1 # Output result
syscall
li $v0, 10 # End of program
syscall
```

六、写一个函数FIB(N, &array)向内存中的一个数组（array）存入斐波那契数列的前N个元素。N和array的地址分别通过\$a0和\$a1传递进来。斐波那契数列的前几个元素是：1, 1, 2, 3, 5, 8, 13, .....

```
fib:
li $t0, 1
sw $t0, 0($a1)
sw $t0, 4($a1)
addi $a0, $a0, -2
loop:
lw $t0, 0($a1)
lw $t1, 4($a1)
add $t0, $t0, $t1
sw $t0, 8($a1)
addi $a1, $a1, 4
```

```
addi $a0, $a0, -1  
bgtz $a0, loop  
jr $ra
```

七、请分别补全以下MIPS汇编指令，使每条指令完成注释中的功能

- a) add \$s1, \$s2, \$0 # 寄存器间数据移动,  $\$s1 \leftarrow \$s2$
- b) addiu \$s1, \$s1, 1 # 自增1,  $\$s1 \leftarrow \$s1 + 1$
- c) add \$s1, \$zero, \$0 # 清零,  $\$s1 \leftarrow 0$
- d) addiu \$s1, \$s1, -1 # 自减1,  $\$s1 \leftarrow \$s1 - 1$
- e) sub \$s1, \$0, \$s1 # 求反,  $\$s1 \leftarrow -\$s1$