计算机组成原理第三次理论作业

1. 将下列 MIPS 指令翻译为机器代码

1. add $s1$,s2, $s3 机器代码：0x02538820
2. sub $s4$,t5, $t9 机器代码：0x01b9a022
3. and $t1$,t2, $s3 机器代码：0x014b4824
4. or $t0$,t1, $t2 机器代码：0x012a4025
5. slt $t0$,s3, $s4 机器代码：0x0274402a
6. lw $t0, 32$(s3) 机器代码：0x8e680020
7. sw $t0, 48$(s3) 机器代码：0xae680030
8.

20000H BEQ $s3$,s4, EXIT ;BEQ 指令所在地址为 20000H

...

40000H ;EXIT 指令所在地址为 40000H

机器代码: 0x12740c00

2. Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields: op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

op=0, funct=34 -> sub

rs=3 -> $v1rt = 2- >$v0

rd=3 -> $v1

shamt=0 -> 0

type -> R-type

assembly language instruction -> sub $v1$,v1, $v0

binary representation -> 000000 00011 00010 00011 00000 100010

hexadecimal representation -> 0x00621822

3. Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields: op=0x23, rs=1, rt=2, const=0x4

op=0x23 -> lw
rs=1 -> $atrt = 2 - >$v0
const=0x4 -> 4

type -> I-type
assembly language instruction -> lw $v0, 4$(at)
binary representation -> 100011 00001 00010 0000000000000100
hexadecimal representation -> 0x8c420004

4. 假设下面的程序放在内存 60000开始的单元，请将下列程序翻译为机器代码

```
Loop:
    srl $t6, $s4, 2         60000  000000 00000 10100 01110 00010 000010  0x00147082
    and $t1, $t3, $s4       60004  000000 01011 10100 01001 00000 100100  0x01744824
    sw $t2, 32($t1)         60008  101011 01001 01010 0000000000100000  0xad2a0020
    beq $t5, $s3, Exit      60012  000100 01101 10011 0000000000000010  0x11b30002
    addi $s4, $t5, -1       60016  001000 01101 10100 1111111111111111  0x21b4ffff
    j Loop                  60020  000010 00000000001110101001100000  0x0800ea60
Exit: ...                   60024
```

5. 编写实现下列C语言描述的功能的MIPS程序和80x86程序

(1)

```
f = (g-h) + (i-j);
```

MIPS程序: (假设 g, h, i, j 分别存放在 $s0$,s1, $s2$,s3 中，f 最终存放在 $s0 中)

```
sub $t0, $s1, $s2
sub $t1, $s3, $s4
add $s0, $t0, $t1
```

80x86程序: (假设 g, h, i, j 分别存放在 eax, ebx, ecx, edx 中，f 最终存放在 eax 中)

```
sub eax, ebx
sub ecx, edx
add eax, ecx
```

(2)

```
g = h + A[2];
```

MIPS程序: (假设 h 存放在 $s0$中，A的首地址存放在s1 中，g 最终存放在 $s0 中)

```
lw $t0, 8($s1)
add $s0, $s0, $t0
```

80x86程序: (假设 h 存放在 eax 中，A的首地址存放在 ebx 中，g 最终存放在 eax 中)

```
mov eax, [ebx+8]
add eax, [ebx]
```

(3)

```
A[8] = h + A[4];
```

MIPS程序: (假设 h 存放在 $s0$中，A的首地址存放在s1 中)

```
lw $t0, 16($s1)
add $t0, $s0, $t0
sw $t0, 32($s1)
```

80x86程序: (假设 h 存放在 eax 中，A的首地址存放在 ebx 中)

```
add eax, [ebx+16]
mov [ebx+32], eax
```

(4)

```
if (i == j) f = g - h;
else f = g + h;
```

MIPS程序: (假设 i, j, g, h 分别存放在 $s0$,s1, $s2$,s3 中，f 最终存放在 $s4 中)

```
bne $s0, $s1, Else
sub $s4, $s2, $s3
j Exit
Else:
add $s4, $s2, $s3
Exit:
```

80x86程序: (假设 i, j, g, h 分别存放在 eax, ebx, ecx, edx 中，f 最终存放在 eax 中)

```
cmp eax, ebx
jne Else
sub eax, edx
jmp Exit
Else:
add eax, edx
Exit:
```

(5)

```
while (save[i] < k) i += 1;
```

MIPS程序: (假设 save 的首地址存放在 $s0$中，k存放在s1 中，i 存放在 $s2 中)

```
Loop:
sll $t0, $s2, 2
add $t0, $s0, $t0
lw $t1, 0($t0)
bge $t1, $s1, Exit
addi $s2, $s2, 1
j Loop
Exit:
```

80x86程序: (假设 save 的首地址存放在 ebx 中，k 存放在 ecx 中，i 存放在 edx 中，其中i为int型占4个字节)

```
Loop:
shl edx, 2
cmp [ebx+edx], ecx
jge Exit
inc edx
jmp Loop
Exit:
```