

1. 请从指令编码格式的角度分析 MIPS 寄存器个数为 32 个的合理性。提示：主要分析如果寄存器个数多于 32 时，对于指令编码会带来哪些负面影响。
2. 请以指令为单位分析 jal 的被调用函数的入口地址的范围。
3. 请结合 C 语言的特点，分析 beq 指令格式中 imm 域的合理性。
4. 请把下列 C 代码翻译为汇编代码。假设 \$s0 存储着变量 i，可用的寄存器只有 \$s0~\$s3。

```
do {
    循环体;
while ( 0<i && i<100 ) ;
```

5. 请用 beq 和 bne 以及与分支无关的指令来完成下面这条语句的功能。提示：要防止溢出。
6. 程序员编写了如下汇编程序来完成 1000 个字节的复制任务。假设循环开始前，\$s0 和 \$s1 分别指向源字符串首地址和目的字符串首地址。

```
LOOP:  lb $t0, 0($s0)
      beq $t0, $0, TAIL
      sb $t0, 0($s1)
      addi $s0, $s0, 1
      addi $s1, $s1, 1
      j LOOP
TAIL:
```

- 1) 请计算上述代码执行的指令总数。
- 2) 请最大化优化上述代码。优化代码只能使用本章讲授的指令。优化后代码仍然采用循环结构，且每次循环只能有 1 次复制操作。
- 3) 请计算优化后代码执行的指令总数。
7. 请编写一个名为 fib(n) 的递归函数来计算第 n 个斐波那契数。斐波那契数列的计算公式为： $f(n+2)=f(n+1)+f(n)$ ， $f(1)=f(2)=1$ 。
- 1) 用 MARS 模拟器测试程序是否正确。
- 2) 如果栈空间容量为 4KB，请估算递归调用次数的极限。

## 答案

1. 多于 32 个，则编码位数就至少为 6 位。
  - 1) R 型移位指令：移位位数为 4 位，无法单条指令实现在 32 位内任意移位。
  - 2) I 型计算：立即数部分则会减至 14 位。扩展至 32 位时至少需要 2 条指令。
  - 3) I 型 beq：转移地址范围大幅度减少。
2. 如果将 4G 划分为 16 个 256MB，则可以在 jal 所在的那个 256MB 区段内任意跳转。
3. beq 的 imm 有 16 位，相当于 64K 条指令的范围。意味着以 beq 为基准，其上下可跳转的范围为{-32K, +32K}。一条 C 语句大于在 10 条指令左右。这意味着 if-else 语句块的大下为 6.4K 条 C 语句。从程序设计合理性角度，不应该有这么大的语句块。因此 imm 的范围足够了。
- 4.

```

Loop :
    循环体
    blez $s0, Loop_End      // i ≤ 0 时退出
    slt $s0, $s1, 100       // i ≥ 0 时, $s1 为 0
    beq $s1, $0, Loop_End   //
    j Loop

```

Loop\_End :

5.

s1 小于 s2: s0 为 1  
s1 等于 s2: s0 为 0  
s1 大于 s2: s0 为 0

```

sub $t0, $s1, $s2
beq $t0, $0, SET0      // s1 等于 s2
srl $t0, $t0, 31       // 只保留符号位
beq $t0, $0, SET0      // 符号位为 0: S1>S2
addi $s0, $0, 1        // 符号位为 1: S1<S2
j END

```

SET0:

```
addi $s0, $0, 0
```

END:

6.

- 1) 1000 字节, 还得增加 1 个字节的结束符 (即 0), 因此总计循环 1001 次。所以执行的总指令数=6\*1001=6006
- 2) 循环 250 次用于复制, 然后 S1 就指向了结束符所在的【字】。为此, 需要写入字内的最高字节 (因为最高字节是结束符)

```
addi $t1, $0, 250
```

Loop:

```

lw $t0, 0($s0)
sw $t0, 0($s1)
addi $s0, $s0, 4
addi $s1, $s1, 4
addi $t1, $t1, -1
bne $t1, 0, LOOP
sb $0, -3($s1)

```

- 3) 指令总数=6\*250+1=1501

7.

- 1) 略。
- 2) 假设函数需要入栈的寄存器总数为 n (含 PC 以及通用寄存器等), 则栈容量需求为 4n 字节。调用次数的极限为 4K/4n=1K/n。