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。

答案：

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。

1. 请以指令为单位分析jal的被调用函数的入口地址的范围。

答案：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。

1. 请结合C语言的特点，分析beq指令格式中imm域的合理性。

答案：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。

1. 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 ) ;

答案：

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 :

1. 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。

slt $s0, $s1, $s2

答案：

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:

1. 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。

答案：

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

1. 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。

答案：

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 $s0, $s1, 4

addi $t1, $t1, -1

bne $t1, 0, LOOP

sb $0, -3(s1)

1. 请计算优化后代码执行的指令总数。

答案：

指令总数=6\*250+1=1501

1. 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。
2. 用MARS模拟器测试程序是否正确。

答案：

略。

1. 如果栈空间容量为4KB，请估算递归调用次数的极限。

答案：

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。