

# gxp班作业答案勘误

在作业答案中有许多错误/不严谨/不完整之处，在答疑课询问完后进行了整理勘误。

## HW4

T1.解答中：“I型计算……扩展至32位时至少需要2条指令”应修改为**处理32位数据时至少需要3条指令**

T5.解答中的MIPS代码仍不能实现避免溢出。应采用如下方式：

```
srl $t1, $s1, 31
srl $t2, $s2, 31
bne $t1, $t2, diff_sign
same_sign:
    sub $t0, $s1, $s2
    srl $s0, $t0, 31
    beq $zero, $zero, end
diff_sign:
    add $s0, $t1, $zero
end:
```

T6.(1) 6002次

(2)如下

```
addi $t2, $s0, 1000    # 1. 预先计算源字符串的结束地址 (1000字节后的地址)
LOOP: lw    $t0, 0($s0)      # 2. 加载一个字 (4字节)
      sw    $t0, 0($s1)      # 3. 存储一个字 (4字节)
      addi $s0, $s0, 4        # 4. 源地址 + 4
      addi $s1, $s1, 4        # 5. 目的地址 + 4
      bne  $s0, $t2, LOOP    # 6. 检查源地址是否到达结束地址
TAIL:
```

(3) $1 + 250 * 5 = 1251$ 次

T7.原版无答案，这里给出一份仅供参考：

```
fib:
    addi $sp, $sp, -12
    sw  $ra, 8($sp)
    sw  $s0, 4($sp)
    sw  $s1, 0($sp)
    move $s0, $a0
    li  $t0, 2
    bgt $s0, $t0, L_recurse
    li  $v0, 1
    j   L_return
L_recurse:
    addi $a0, $s0, -1
    jal fib
    move $s1, $v0
    addi $a0, $s0, -2
    jal fib
    add $v0, $s1, $v0
```

```

_L_return:
    lw $s1, 0($sp)
    lw $s0, 4($sp)
    lw $ra, 8($sp)
    addi $sp, $sp, 12
    jr $ra

```

## HW5

---

题目表述不清/答案错误的重灾区。

T2.

(4) 答案,  $N = 11$ 时性能最好

(5) 题目应改为: .....“每增加一级”性能提升最显著

T5.

(1) `lw` 与 `sw` 间数据冲突仅需暂停一个周期。因为 W 级到 E 级存在转发通路, 也对  $GPR[rt]$  的值进行了修正。共暂停两周期。

T6. 哪怕答案进行修正, 题目仍有不严谨或歧义之处。但至少这种做法是可以理喻的。

$$CPI_{load} = 1 + 30\% \times 1 + 5\% \times 1 = 1.35$$

$$CPI_{beq} = 1 + 25\% \times 1 = 1.25$$

$$CPI_{else} = 1$$

$$CPI = 15\% \times 1.35 + 10\% \times 1.25 + 75\% \times 1 = 1.0775$$

T7.

(2) 最后两条 `I2`, `I4`, 可调换顺序。

(3) 优化前, 由于寄存器无内部转发, 所以 `lw` 和 `sw` 间需要3条 `nop` 指令。由于支持 M 级向 D 级转发, `sub` 和 `add` 间仅需一条 `nop`。歪打正着总数仍为13个时钟周期。

优化后, 因为 `lw` 取出的\$1数据要隔三个 `nop` 才能使用, 故第一次使用\$1必须在第五条指令。故无论 `I2` `I4` 谁在前, 都得在之前暂停一拍。10个时钟周期。

## HW6

---

T4.“代价100周期”, 所以是在原有1周期基础上的。故每个块是  $100 + 8 = 108$  个时钟周期。总时间  $5056K$  个时钟周期

## 补充注意

---

HW1中化简结果可能不唯一。