1．假定执行某种指令 I 需要 20 个时钟周期，该指令在程序中出现的频度为 10%，其他所有 指令的平均 CPI 为 5，则 CPU 执行指令 I 所用时间占整个 CPU 时间的百分比是多少？如果 通过对硬件进行改进，能使指令 I 的执行时间缩短为 10 个执行周期，但同时会使 CPU时钟周期延长10%，你认为是否应该采取这种措施？

（1）设总的指令条数为C，时钟周期为T，则执行指令I的cpu时间为T \* 20 \* C \* 10%，执行所有指令的cpu时间为T \* 5 \* C \* 90% +T \* 20 \* C \* 10%，则百分比为：

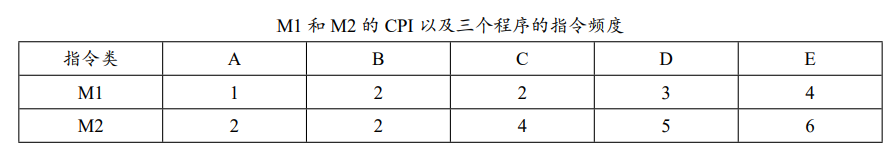
20 \* 10% /（5 \* 90% + 20 \* 10%）= 30.77%

（2）设总的指令条数为C，原cpu时间为T \* 5 \* C \* 90% +T \* 20 \* C \* 10% = 6.5CT，现将CPU时钟周期延长10%，则现在的cpu时间为T \* (1 + 10%) \* 5 \* C \* 90% + T \* (1 + 10%) \* 10 \* C \* 10% =6.05CT

6.5CT > 6.05CT

所以应该采取这种措施

2．若机器 M1 和 M2 具有相同的指令集，其时钟频率分别为 0.8GHz 和 1.6GHz。在指令集 中有 5 种不同类型的指令 A ~ E。下表给出了在 M1 和 M2 上每类指令的平均时钟周期数 CPI



请回答下列问题：

（1）M1 和 M2 的峰值 MIPS 各是多少？

（2）假定某程序 P 的指令序列中，5 类指令具有完全相同的指令条数，则程序 P 在 M1 和 M2 上运行时，哪台机器更快？快多少？在 M1 和 M2 上执行程序 P 时的平均时钟周期数 CPI 各 是多少？

（1）M1 =0.8 / (1+2+2+3+4) \* 0.2 \*106 = 0.333 \* 10-6

M2 =1.6 / (2+2+4+5+6) \* 0.2 \*106 = 0.421 \* 10-6

（2）

CPU1 = (1+2+2+3+4) \* 0.2 / 0.8 = 3 CPU2 = (2+2+4+5+6) \* 0.2 / 1.6 = 2.375

M2更快，快0.625秒

CPI1 =(1+2+2+3+4) / 5 = 2.4 CPI2 =(2+2+4+5+6) / 5= 3.8

3．某 C 语言程序，从终端输入 float 型数值：61.419997，将其打印输出，得：61.419998。 当输入分别为以下数值时：61.42，61.49，62.335，请问打印输出为多少？

输出分别为：61.419998 61.490002 62.334999

4．考虑下列 C 语言程序代码：

int i = 65535;

short si = (short) i;

int j = si;

假定上述程序段在某 32 位机器上执行，sizeof(int)=4，则变量 i、si 和 j 的值分别是多少? 为什么?

分别是：65535 -1 -1

原因：在c语言中，int类型占32位，short类型占16位，代码中均为带符号整形，65535在计算机中表示为16个1即111111111111111，当其为short类型时最高位为符号位，改二进制数表示为最大的负整数即-1

5．假设某字长为8位的计算机中，带符号整数采用补码表示，x=-68，y=-80，x和y分别存放在寄存器A和B中。请回答下列问题（要求最终用十六进制表示二进制序列)

(1）寄存器A和B中的内容分别是什么?

(2）若x和y 相加后的结果存放在寄存器C中，则寄存器C中的内容是什么?运算结果是否正确?加法器最高位的进位Cout是什么?溢出标志OF、符号标志SF和零标志ZF各是什么?

(3)若x和y相减后的结果存放在寄存器D中，则寄存器D中的内容是什么?运算结果是否正确?加法器最高位的进位Cout是什么?溢出标志OF、符号标志SF和零标志ZF各是什么?

(4）对于带符号整数的减法运算，能否直接根据CF的值对两个带符号整数的大小进行比较?

（1）寄存器A：10111100 寄存器B：10110000

（2）运算结果为：01101100 原码为：-10010100 即-148，结果不正确

最高位的进位为0

OF = 1 SF = 1 ZF = 0

（3）运算结果为：00001100 原码为：-11110010 即-242，结果不正确

最高位的进位为1

OF = 1 SF=1 ZF=0

（4）不能，还需要根据SF一起判断，若CF=SF则大于

6．已知 C 语言中的按位异或运算“XOR”用符号“^”表示。对于任意一个位序列 a 存在a^a=0。C 语言程序可以利用这个特性来实现两个数值交换的功能。以下是一个实现该功能 的 C 语言函数：

void xor\_swap(int \*x, int \*y)

{

\*y = \*x ^ \*y; /\* 第一步 \*/

\*x = \*x ^ \*y; /\* 第二步 \*/

\*y = \*x ^ \*y; /\* 第三步 \*/

}

假定执行该函数时\*x 和\*y 的初始值分别为 a 和 b，即 \*x=a 且\*y=b，请说明每一步执行结束 后 x 和 y 各自指向的内存单元中的内容分别是什么?

第一步执行结束后，x 和 y 各自指向的内存单元中的内容分别是 a和 a^b。

第二步执行结束后，x 和 y 各自指向的内存单元中的内容分别是 b 和 a^b。

第三步执行结束后，x 和 y 各自指向的内存单元中的内容分别是 b和 a。

7．假定某个实现数组元素倒置的函数 reverse\_array 调用了第 6 题中给出的 xor\_swap 函数：

void reverse\_array(int a[], int len)

{

int left, right = len - 1;

for (left=0; left <= right; left++, right--)

xor\_swap(&a[left], &a[right]);

}

当 len 为偶数时，reverse\_array 函数的执行没有问题。但是，当 len 为奇数时，函数的执行 结果不正确。请问当 len 为奇数时会出现什么问题？最后一次循环中的 left 和 right 各取什么 值？最后一次循环中调用 xor\_swap 函数后的返回值是什么？对 reverse\_array 函数做怎样的 改动就可消除该问题？

1. 当len为奇数时会出现数组中间位置的值为0
2. 最后一次循环中left和right的值相等为len/2
3. 返回值是0
4. 加入判断语句if(left == right) break;