第 1 章　习　题 答 案

2（4）程序的 CPI 和哪些因素有关？

参考答案：

程序 CPI 由程序中指令的组成和每条指令的 CPI 决定

程序中指令的组成由程序设计、编译器、ISA 决定

每条指令的 CPI 由计算机组织（微体系结构）决定

3．假定你的朋友不太懂计算机，请用简单通俗的语言给你的朋友介绍计算机系统是如何工作的。

参考答案：（略）

4．你对计算机系统的哪些部分最熟悉，哪些部分最不熟悉？最想进一步了解细节的是哪些部分的内容？

参考答案：（略）

6．若机器 M1 和 M2 具有相同的指令集，其时钟频率分别为 1GHz 和 1.5GHz。在指令集中有五种不同类型的指令 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。

机器	A	B	C	D	E
M1	1	2	2	3	4
M2	2	2	4	5	6

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，五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？

参考答案：

（1）M1 上可以选择一段都是 A 类指令组成的程序，其峰值 MIPS 为 1000MIPS。

M2 上可以选择一段 A 和 B 类指令组成的程序，其峰值 MIPS 为 1500/2=750MIPS。

（2）5 类指令具有完全相同的指令条数，所以各占 20%。

在 M1 和 M2 上执行程序 P 时的平均时钟周期数 CPI 分别为：

M1：20%×(1+2+2+3+4)= 0.2×12 = 2.4

M2：20%×(2+2+4+5+6)= 0.2×19 = 3.8

假设程序 P 的指令条数为 N，则在 M1 和 M2 上的执行时间分别为：

M1：2.4× N×1/1G = 2.4N (ns)

M2：3.8×N×1/1.5G = 2.53 N (ns)

M1 执行 P 的速度更快，每条指令平均快 0.13ns，也即 M1 比 M2 快 0.13/2.4×100%≈5%。

（问题：如果说程序 P 在 M1 上执行比 M2 上快 (3.8–2.4)/3.8×100% = 36.8%，那么，这个结论显然是错误的。请问错在什么地方？）

8．假设某机器 M 的时钟频率为 4GHz，用户程序 P 在 M 上的指令条数为 8×10^9，其 CPI 为 1.25，则 P 在 M 上的执行时间是多少？若在机器 M 上从程序 P 开始启动到执行结束所需的时间是 4 秒，则 P 占用的 CPU 时间的百分比是多少？

参考答案：

程序 P 在 M 上的执行时间为：$1.25\times8\times10^9\times1/4G = 2.5$ s，从启动 P 执行开始到执行结束的总时间为 4 秒，其中 2.5 秒是 P 在 CPU 上真正的执行时间，其他时间可能执行操作系统程序或其他用户程序。

程序 P 占用的 CPU 时间的百分比为：2.5/4 = 62.5%。

10．假定机器 M 的时钟频率为 1.2GHz，某程序 P 在机器 M 上的执行时间为 12 秒钟。对 P 优化时，将其所有的乘 4 指令都换成了一条左移 2 位的指令，得到优化后的程序 P'。已知在 M 上乘法指令的 CPI 为 5，左移指令的 CPI 为 2，P 的执行时间是 P'执行时间的 1.2 倍，则 P 中有多少条乘法指令被替换成了左移指令被执行？

参考答案：

显然，P'的执行时间为 10 秒，因此，P 比 P'多花了 2 秒钟，因此，执行时被换成左移指令的乘法指令的条数为 $1.2G\times2/(5-2) = 800M$。

第 二 章　习　题 答 案

2（1）为什么计算机内部采用二进制表示信息？既然计算机内部所有信息都用二进制表示，为什么还要用到十六进制和八进制数？

参考答案：（略）

2（7）为什么计算机处理汉字时会涉及到不同的编码（如，输入码、内码、字模码）？说明这些编码中哪些是用二进制编码，哪些不是用二进制编码，为什么？

参考答案：（略）

7．假定一台 32 位字长的机器中带符号整数用补码表示，浮点数用 IEEE 754 标准表示，寄存器 R1 和 R2 的内容分别为 R1：0000 017AH，R2：FFFF F895H。不同指令对寄存器进行不同的操作，因而，不同指令执行时寄存器内容对应的真值不同。假定执行下列运算指令时，操作数为寄存器 R1 和 R2 的内容，则 R1 和 R2 中操作数的真值分别为多少？

（1） 无符号数加法指令

（2） 带符号整数乘法指令

（3） 单精度浮点数减法指令

参考答案：

R1 = 0000 017AH = 0000 0000 0000 0000 0000 0001 0111 1010

R2 = FFFF F895H = 1111 1111 1111 1111 1111 1000 1001 0101

（1）对于无符号数加法指令，R1 和 R2 中是操作数的无符号数表示，因此，其真值分别为 R1：17AH，R2：FFFF F895H。（对应十进制分别为 378、$4\ 294\ 965\ 397=2^{32}-1899$）

（2）对于带符号整数乘法指令，R1 和 R2 中是操作数的带符号整数补码表示，由最高位可知， R1 为正数， R2 为负数。R1 的真值为+17AH=378, R2 的真值为–111 0110 1011= –1899。

（3）R1：符号位为 0，表示其为正数，阶码为 0000 0000，尾数部分为 000 0000 0000 0001 0111 1010，故其为非规格化浮点数，指数为–126，尾数中没有隐藏的 1，用十六进制表示尾数为 +0.0000 0000 0000 0010 1111 0100=+0.0002F4H，故 R1 表示的真值为$+0.0002F4H\times2^{-126}$。

R2: 符号位为 1，表示其为负数，阶码为 1111 1111，尾数部分为 111 1111 1111 1000 1001 0101，

故其为全 1 阶码非 0 尾数，即是一个非数 NaN。

9. 以下是一个 C 语言程序，用来计算一个数组 a 中每个元素的和。当参数 len 为 0 时，返回值应该是 0，但是在机器上执行时，却发生了存储器访问异常。请问这是什么原因造成的，并说明程序应该如何修改。

```
1    float sum_elements(float a[], unsigned len)
2    {
3        int   i;
4        float   result = 0;
5
6        for  (i = 0; i <= len–1; i++)
7            result += a[i];
8        return result;
9    }
```

参考答案：

参数 len 的类型是 unsigned，所以，当 len=0 时，执行 len-1 的结果为 11…1，是最大可表示的无符号数，因而，任何无符号数都比它小，使得循环体被不断执行，引起数组元素的访问越界，发生存储器访问异常。

只要将 len 声明为 int 型，或循环的测试条件改为 i<len。

10. 设某浮点数格式为：

数符	阶码	尾数
1 位	5 位移码	6 位补码数值

其中，移码的偏置常数为 16，补码采用一位符号位，基数为 4。

（1） 用这种格式表示下列十进制数：+1.75，+19，–1/8。

（2） 写出该格式浮点数的表示范围，并与 12 位定点补码整数表示范围比较。

参考答案：（假定采用 0 舍 1 入法进行舍入）

（1）+1.75 = +1.11B = 0.011100B× 4^1，故阶码为 1 +16 = 17 = 10001B，尾数为+0.011100 的补码，即 0.011100，所以+1.75 表示为 0 10001 011100。

+19 = +10011B = 0.010011B× 4^3，故阶码为 3 + 16 = 19 = 10011B，尾数为 0.010011，所以+19 表示为 0 10011 010011。

–1/8 = – 0.125 = – 0.001B = – 0.100000 × 4^{-1}，阶码为 –1 + 16 = 15 = 01111B，尾数为–0.100000 的补码，即 1.100000，所以–1/8 表示为 1 01111 100000。

（2）该格式浮点数表示的范围如下。

正数最大值：0.111111B × 4^{11111}，即：0.333× 4^{15} （≈2^{30}≈10^9）

正数最小值：0.000001B × 4^{00000}，即：0.001× 4^{-16} （≈2^{-34}≈10^{-10}）

负数最大值：–0.000001B × 4^{00000}，即：–0.001× 4^{-16}

负数最小值：–1.000000B × 4^{11111}，即：–1.000× 4^{15}

因此，该格式浮点数的数量级在 10^{-10}～10^9 之间。

12 位定点补码整数的表示范围为：-2^{11}～$+(2^{11}-1)$，即：–2048～2047
由此可见，定点数和浮点数的表示范围相差非常大。

13．设一个变量的值为 6144，要求分别用 32 位补码整数和 IEEE 754 单精度浮点格式表示该变量（结果用十六进制表示），并说明哪段二进制序列在两种表示中完全相同，为什么会相同？

参考答案：
6144 = +1 1000 0000 0000B = +1. 1× 2^{12}
32 位补码形式为：0000 0000 0000 0000 0001 1000 0000 0000 （00001800H）
IEEE754 单精度格式为：0 10001011 100 0000 0000 0000 0000 0000 （45C0 0000H）
蓝字部分为除隐藏位外的有效数字，因此，在两种表示中是相同的序列。因为正数的补码和原码是一致的，所以除隐藏位外的有效数字都相同。

14．设一个变量的值为–6144，要求分别用 32 位补码整数和 IEEE754 单精度浮点格式表示该变量（结果用十六进制表示），并说明哪种表示其值完全精确，哪种表示的是近似值。

参考答案：
–6144 = –1 1000 0000 0000B = –1. 1× 2^{12}
32 位补码形式为：1111 1111 1111 1111 1110 1000 0000 0000 （FFFF E800H）
IEEE 754 单精度格式为：1 10001011 100 0000 0000 0000 0000 0000 （C5C0 0000H）
32 位补码形式能表示精确的值，浮点数表示的也是精确值，因为没有有效数字被截断。

17．假定在一个程序中定义了变量 x、y 和 i，其中，x 和 y 是 float 型变量（用 IEEE754 单精度浮点数表示），i 是 16 位 short 型变量（用补码表示）。程序执行到某一时刻，x = –10.125、y=12、i=–125，它们都被写到了主存（按字节编址），其地址分别是 100，108 和 112。请分别画出在大端机器和小端机器上变量 x、y 和 i 在内存的存放位置。

参考答案：
–10.125 = –1010.001B = –1.010001 × 2^3
x 在机器内部的机器数为：1　10000010　0100010…0 (C122 0000H)
12= +1100B= +1.1 × 2^3
y 在机器内部的机器数为：0　10000010　100…0 (4140 0000H)
–125=–111 1101B
i 在机器内部的机器数为：1111 1111 1000 0011（FF83H）

	大端机	小端机
地址	内容	内容
100	C1H	00H
101	22H	00H
102	00H	22H
103	00H	C1H
108	41H	00H
109	40H	00H
110	00H	40H

111	00H	41H
112	FFH	83H
113	83H	FFH

18. 假定某计算机的总线采用偶校验，每 8 位数据有一位校验位，若在 32 位数据线上传输的信息是 8F 3C AB 96H，则对应的 4 个校验位应为什么？若接受方收到的数据信息和校验位分别为 87 3C AB 96H 和 1010B，则说明发生了什么情况，并给出验证过程。

参考答案：

传输信息 8F 3C AB 96H 展开为 1000 1111 0011 1100 1010 1011 1001 0110，每 8 位有一个偶校验位，因此，总线上发送方送出的 4 个校验位应该分别为 1、0、1、0。

接受方的数据信息为 87 3C AB 96H，展开后为 1000 0111 0011 1100 1010 1011 1001 0110，接收到的校验位分别为 1、0、1、0。

在接受方进行校验判断如下：

根据接收到的数据信息计算出 4 个偶校验位分别为 0、0、1、0，将该 4 位校验位分别和接收到的 4 位校验位进行异或，得到 1、0、0、0，说明数据信息的第一个字节发生传输错误。对照传输前、后的数据信息，第一字节 8FH 变成了 87H，说明确实发生了传输错误，验证正确。