计算机系统结构 第一次作业

陈文迪 519021910071

问题一

描述：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?

解答：

假设共有x条乘法指令被替换成了左移指令。

由于程序P在M上的执行时间为12秒钟，而P的执行时间是P'执行时间的1.2倍，故P'在M上的执行时间为10秒钟。

根据CPU执行时间公式，我们可以列出如下方程：

$$\frac{5}{1.2\times 10^9}x-\frac{2}{1.2\times 10^9}x=12-10$$

解得$x=8\times 10^8$，故共有8×10^8条乘法指令被替换成了左移指令。

问题二

描述：图形处理器中经常需要的一种转换是求平方根。浮点（FP）平方根的实现在性能方面有很大差异，特别是在为图形设计的处理器中，尤为明显。假设FP平方根（FPSQR）占用一项关键图形基准测试中30%的执行时间。有一项提议：升级FPSQR硬件，使这一运算速度提高到原来的10倍。另一项提议是让图形处理器中所有FP指令的运行速度提高到原来的1.6倍，FP指令占用该应用程序一半的执行时间。设计团队相信，他们使所有FP指令执行速度提高到1.6倍所需要的工作量与加快平方根运算的工作量相同。试比较这两种设计方案。

解答：

我们可以根据Amdahl定律来计算改进后系统的性能加速比。

对于第一种方案，系统总加速比为：

$$\frac{1}{(1-0.3)+\frac{0.3}{10}}=1.37$$

对于第二种方案，系统总加速比为：

$$\frac{1}{(1-0.5)+\frac{0.5}{1.6}}=1.23$$

因此，第一种方案性能提升效果更好。

问题三

描述：假设我们在对有符号值使用补码运算的32位机器上运行代码。对于有符号值使用的是算术右移，对无符号值使用的是逻辑右移。变量的声明和初始化如下：

```
int x = foo(); //调用某某函数，给x赋值
int y = bar(); //调用某某函数，给y赋值
unsigned ux = x;
unsigned uy = y;
```

对于下面每个C表达式：证明对于所有的x和y 值，都为真（等于1）；或者给出使得它为假的x和y值。

```
A.  (x>0) || (x-1<0)
B.  (x&7) != 7 || (x<<29<0)
C.  (x * x) >= 0
D.  x<0 || -x<=0
E.  x>0 || -x>=0
F.  x+y == uy+ux
G.  x*~y + uy*ux == -x
H.  x*4 + y*8 == (x<<2)+(y<<3)
I.  ((x>>2)<<2)<=x
```

解答：

A：取 x = -2147483648 时，会发生溢出，此时 x < 0 且 x-1 = 2147483647 > 0 ，该表达式为假。

B：该式为真。因为若 (x&7)!= 7 为假，则x的最低三位必为111，因此 x<<29 的符号位为1，故 x<<29 < 0 为真。

C：取 x =0xF000 时， x*x 会溢出导致符号位为1，因此原表达式为假。

D：该式为真。因为32位int的范围是-2147483648~2147483647，若x>=0，则-x一定不会超出表示范围，因此

-x<=0为真。

E：取 x = -2147483648 时，-x仍为-2147483648，因此原表达式为假。

F：该式为真。有符号数在于无符号数比较时会转化为无符号数，且二者的加法运算规则相等，因此该式恒为真。

G：该式为真。根据补码的定义，-y = ~y+1，所以 ~y = -y-1。因此，左式等于x*(-y)-x+uy*ux，再依据F的结论可知左式的值正是-x。

H：该式为真。因为有符号数的乘法过程是，先将运算数当作无符号数再进行截断，其产生的结果与左移是一样的。

I：该式为真。因为左式相当于将x的最低两位抹去，相当于减去一个正数或零，因此原式恒为真。

问题四

描述：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 = –0.125、y=7.5、i=100，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。

解答：

我们先将x，y，i转化为机器数。

x = -0.125 = -0.001B = -1.0×2^{-3} ，在机器内部的机器数为：1 01111100 000...0 (BE000000H)。

y = 7.5 = 111.1B = 1.111×2^{2}，在机器内部的机器数为：0 10000001 11100...0(40F00000H)。

i = 64+32+4，在机器内部的机器数为：0000 0000 0110 0100 (0064H)。

内存位置如下图所示。

地址	大端机	小端机
100	BEH	00H
101	00H	00H
102	00H	00H
103	00H	BEH
108	40H	00H
109	F0H	00H
110	00H	F0H
111	00H	40H
112	00H	64H
113	64H	00H

问题五

描述： We are running programs on a machine with the following characteristics:

- Values of type int are 32 bits. They are represented in two's complement, and they are right shifted arithmetically. Values of type unsigned are 32 bits.
- Values of type float are represented using the 32-bit IEEE floating point format, while values of type double use the 64-bit IEEE floating point format.
- We generate arbitrary values x, y, and z, and convert them to other forms as follows:

```
/* Create some arbitrary values */
int x = random();
int y = random();
int z = random();
 /* Convert to other forms */
unsigned ux = (unsigned) x;
unsigned uy = (unsigned) y;
double dx = (double) x;
double dy = (double) y;
double dz = (double) z;
```

For each of the following C expressions, you are to indicate whether or not the expression always yields 1. If so, circle "Y". If not, circle "N" and tell why.

Expression	Always True?
`(x<y) == (-x>-y)`	Y N
`((x+y)<<4) + y-x == 17*y+15*x`	Y N
`~x+~y+1 == ~(x+y)`	Y N
`ux-uy == -(y-x)`	Y N
`(x >= 0) || (x < ux)`	Y N
`((x >> 1) << 1) <= x`	Y N
`(double)(float) x == (double) x`	Y N
`dx + dy == (double) (y+x)`	Y N
`dx + dy + dz == dz + dy + dx`	Y N
`dx * dy * dz == dz * dy * dx`	Y N

解答：

1. “N”。取x=-2147483648，y=0。此时-x仍是-2147483648，-y=0，原式为假。
2. “Y”。有符号整型的加法仍然是按位加法，因此无论有无溢出或正负，左端产生的结果在内部表示上与(x+y)*16+y-x = 17*y+15*x相同，原式恒为真。
3. “Y”。由于-x=~x+1，所以~x=-x-1。原式左端变为-x-y-1，右端也变为-x-y-1，原式恒为真。
4. “Y”。在进行比较时，无论是否是无符号数，都会转化为无符号数进行比较。左右两端虽然在意义上不同，但在数位上是相同的，原式恒为真。
5. “N”。取x=-1，由于有符号数和无符号数在比较时同一转为无符号数，x==ux，原式为假。
6. “Y”。因为左式相当于将x的最低一位抹去，相当于减去一个正数或零，因此原式恒为真。
7. “N”。取x=2111111111，由于左端先转化为float，会有精度损失，原式为假。
8. “N”。取x=y=2147483647，则y+x发生溢出得到-2，原式为假。
9. “Y”。由于double具有52个尾数位，对于int级别或比int大几个数量级的整数都可以精确表达，原式恒为真。
10. ”N“。取x=2147483647，y=21474836，z=3，由于运算顺序不同和精度损失，左端结果不等于右端，原式为假。