

计算机系统结构 第一次作业

UNikeEN 2023/2/24

T1

解：优化前后时间变化： $\Delta t = t(P) - t(P') = 12 - \frac{12}{1.2} = 2s$

对应时钟周期数： $\Delta k = 2 \times 1.2G = 2.4 \times 10^9$

被替换的乘法指令数： $\Delta n = \frac{\Delta k}{\Delta(CPI)} = \frac{2.4 \times 10^9}{5-2} = 8 \times 10^8$ 条

T2

解：方案一加速比： $\frac{1}{(1-0.3)+\frac{0.3}{10}} = 1.37$

方案二加速比： $\frac{1}{(1-0.5)+\frac{0.5}{1.6}} = 1.23$

方案一加速比大于方案二，在工作量相同的情况下，设计团队选择加快平方根运算的设计方案更好。

T3

- A.错误。当 $x=\text{INT_MIN}=-2147483648$ 时，等式左侧为假，等式右侧 $x-1=\text{INT_MAX}<0$ 为假
- B.永真。若为假，需要 $((x\&7)\&7)\&\&((x<<29)>=0)$ 。当 $(x\&7)\&7$ 时， x 的形式为...1111， $x<<29$ 的形式为1111...，故 $x<<29$ 一定小于零
- C.错误。 $x*x$ 可能会溢出
- D.正确。int 范围内每个正数都有对应的相反数， x 为正数时原式一定成立，不发生溢出
- E.错误。当 $x=\text{INT_MIN}$ 时为假
- F.正确。比较时 $x+y$ 首先被转换为unsigned int，因此始终为真
- G.正确。 $(x\sim y+uy\sim ux) = x\sim(uy+\sim y) = x\sim(-1) = -x$
- H.正确。整数乘法中左移指令可以代替与2的幂次方相乘
- I.正确。先右移2位，再左移2位会让 x 的0、1位全部变成0，一定会变小或不变

T4

表达式	永真	原因
$(x < y) == (-x > -y)$	N	当 $x=\text{INT_MIN}$ 时为假，由于溢出，此时 x 与 $-x$ 相等，任取 y 等式均不成立

表达式	永真	原因
<code>((x+y)<<4) + y-x == 17*y+15*x</code>	Y	整数乘法中左移指令可以代替与2的幂次方相乘
<code>~x+~y+1 == ~(x+y)</code>	Y	可以看成先把x和y转换为无符号整数再操作
<code>ux-uy == -(y-x)</code>	Y	比较时都会转换为unsigned int，所以成立。
<code>!(x >= 0)</code>		<code>(x < ux)</code>
<code>((x >> 1) << 1) <= x</code>	Y	当x为偶时左右相等，当x为奇时x补码末位为1，右移再左移后变0，即左侧为x-1，等式成立
<code>(double)(float) x == (double) x</code>	N	左侧int转float会丢失精度，右侧直接转double不会
<code>dx + dy == (double)(y+x)</code>	N	等式右侧可能会发生溢出。如当 <code>x=INT_MAX</code> ， <code>y>0</code> 时等式不成立。
<code>dx + dy + dz == dz + dy + dx</code>	Y	此处double由int转化而来，double精度足以表示三个int之和
<code>dx * dy * dz == dz * dy * dx</code>	N	三个int相乘，可能超出尾码长度，此时进行浮点数运算会有精度缺失