

第一章作业

20195633 李燕琴

1.2

采用流水线提高性能: a,

通过冗余提高性能: b

采用预测提高性能: c

加速大概率事件: d

存储器层次: e

并行提高性能: f

面向摩尔定律的设计: g

使用抽象简化设计: h

1.3

C语言程序 经过编译器, 编译为汇编语言程序(助记符形式的机器指令), 再汇编为二进制机器用语言程序, 得到可执行文件(.exe), 能够直接在CPU上执行。

1.4

(1) $Cache = 1280 * 1024 * 24bit = 30Mbit = 30/8MByte = 3.75MByte$, 即需要3.75MB缓存

(2) $t = 30Mbit / (100Mbit/s) = 0.3s$

1.5

(1) 由 $IC = \frac{time * CR}{CPI}$, $time = 1s$ 得

$IC_1 = 1s * 3 * 10^9 Hz / 1.5 = 2 * 10^9$ 条,

$IC_2 = 1s * 2.5 * 10^9 Hz / 1.0 = 2.5 * 10^9$ 条,

$IC_3 = 1s * 4 * 10^9 Hz / 2.2 = 1.82 * 10^9$ 条,

(2) 由 $CC = time / T = time * CR$ 得时钟周期数

$CC_1 = 10s * 3 * 10^9 Hz = 3 * 10^{10}$

$CC_2 = 10s * 2.5 * 10^9 Hz = 2.5 * 10^{10}$

$CC_3 = 10s * 4 * 10^9 Hz = 4 * 10^{10}$

由 $IC = CC / CPI$ 得指令数

$IC_1 = 3 * 10^{10} / 1.5 = 2 * 10^{10}$,

$IC_2 = 2.5 * 10^{10} / 1.0 = 2.5 * 10^{10}$,

$IC_3 = 4 * 10^{10} / 2.2 = 1.82 * 10^{10}$,

(3) $time = IC * CPI / CR$,

$0.7 * time = IC * 1.2CPI / (k * CR)$.

得 $k = 12/7$, 即时钟频率应该是原来的 $12/7$ 倍。

1.6

$$(1) CPI_A = 1 * 10\% + 2 * 20\% + 3 * 50\% + 3 * 20\% = 2.6$$

$$CPI_B = 2 * 100\% = 2$$

(2) 时钟周期 $T = 1/CR$, CR 为时钟频率。则

$$T_A = 1/(2.5 * 10^9 Hz) = 0.4ns, \quad T_B = 1/(3 * 10^9 Hz) = 0.33ns.$$

1.7

(1) 由 $\overline{CPI} = \frac{time}{T * IC}$ 得,

$$\overline{CPI}_A = \frac{1.1s}{1 * 10^{-9}s * 10^9} = 1.1$$

$$\overline{CPI}_B = \frac{1.5s}{1 * 10^{-9}s * 1.2 * 10^9} = 1.25$$

(2) $\overline{CPI}_A * 1.0 * 10^9 * T_A = \overline{CPI}_B * 1.2 * 10^9 * T_B$, 得 $T_B/T_A = 0.73$, 即B的时钟是A的0.73倍

(3) $T_A/T_{new} = \frac{1.0 * 10^9 * 1.1}{6.0 * 10^8 * 1.1} = 1.67$, 即该处理器相对于A的加速比为1.67倍;

$$T_B/T_{new} = \frac{1.2 * 10^9 * 1.25}{6.0 * 10^8 * 1.1} = 2.27, \text{ 即该处理器相对于B的加速比为2.27倍;}$$

1.8

(1) 假设 $P_{动} = \frac{1}{2} * C * V^2 * f$, 则 $C = \frac{2P_{动}}{V^2 f}$

$$C_A = 32 * 10^{-9} F; \quad C_B = 29.04 * 10^{-9} F.$$

(2)

	静/总	静/动
Pentium 4 Prescott	$\frac{1}{10}$	$\frac{1}{9}$
Core i5 Ivy Bridge	$\frac{3}{7}$	$\frac{3}{4}$

$$(3) P_{静2} + P_{动2} = 0.9 * (P_{静1} + P_{动1}),$$

$$P_{静2} = V_2 * I, P_{动2} = \frac{1}{2} * C * V_2^2 * f;$$

$$P_{静1} = V_1 * I, P_{动1} = \frac{1}{2} * C * V_1^2 * f.$$

对于Pentium 4 Prescott,

$$P_{静1} = 10W, P_{动1} = 90W, f = 3.6 * 10^9 Hz, V_1 = 1.25V, C = 32 * 10^{-9} F$$

$$\text{得 } V_2 = 1.1825V, \text{ 电压降低 } (1.25 - 1.1825)/1.25 = 5.4\%$$

对于Pentium 4 Prescott,

$$P_{静1} = 30W, P_{动1} = 40W, f = 3.4 * 10^9 Hz, V_1 = 0.9V, C = 29.04 * 10^{-9} F$$

$$\text{得 } V_2 = 0.8414V, \text{ 电压降低 } (0.9 - 0.8414)/0.9 = 6.51\%$$

1.13

(1) 总时间减少量: $t_{\text{浮点操作}} * 20\% = 70 * 0.2 = 14s$,

减少百分比: $14/250 = 5.6\%$

(2) 假设除去浮点指令、L/S指令, 分支指令, 就只有整数指令, 则

$$t_{\text{整数操作old}} = 250 - 70 - 85 - 40 = 55s$$

如果总执行时间减少20%, 且只有整数操作时间减少了, 则

$$\Delta t_{\text{整数操作}} = t_{\text{总}} * 20\% = 250 * 0.2 = 50s$$

减少百分比为: $\Delta t_{\text{整数操作}} / t_{\text{整数操作old}} = 50/55 * 100\% = 90.91\%$

(3) 由(2)知, 总时间减少20%即节省50s时间, 而分支指令总40s, 故而不能通过只减少分支指令来减少总时间20%。根据Amdahl定律, 计算证明如下:

$$t_{\text{总}} * 0.8 = (250 - 40) + t_{\text{分支指令old}}/k, \text{ 即 } 250 * 0.8 = (250 - 40) + 40/k \text{ 得 } k < 0,$$