

作业

1953729

吴浩泽

4.1 ① 指令级并行：让所有的处理器都利用流水线来使指令重叠并行执行，以达到提高性能的目的。这种指令之间存在的潜在并行性称为指令级并行ILP
 ② 保留站设置在运算部件的入口。每个保留站都有一个标识字段，唯一地标识3级保留站。每个保留站中保存一条已经流出并等待到本功能部件执行的指令。
 ③ CDB：是该结构中的一条重要的数据通路。所有功能部件的计算结果都是送到CDB上，由它把这些结果直接播送到各个需要该结果的地方。
 ④ 分支历史表：分支预测缓冲器。最简单的动态分支预测方法。用BHT来记录分支指令最近一次或几次的执行情况（成功或不成功），并据此进行预测。
 ⑤ ROB是为实现前门缓冲而设置在指令操作完成后到指令被确认这段时间，为指令保存数据。

4.4 (1) 程序执行的CPI = 没有分支的基本CPI + 分支带来的额外开销
 分支带来的额外开销是指在分支指令中，缓冲命中但预测错误带来的开销与缓冲没有命中带来的开销之和。
 分支带来的额外开销 = $15\% \times (90\% \text{命中} \times 10\% \text{预测错误} \times 4 + 10\% \text{没命中} \times 3) = 0.099$

所以，程序执行的CPI = $1 + 0.099 = 1.099$
 (2) 采用固定的2个时钟周期延迟的分支处理CPI = $1 + 15\% \times 2 = 1.3$
 由(1)(2)可知，分支目标缓冲方法执行速度快。

4.6 (1) 共需14个时钟周期，其中有5个空转周期。
 (2) 循环顺序展开4次，不进行任何调度，则指令1~5及其间的Stall都是必要的，只是指令6~9只需执行一次，因此，共有 $10 \times 4 + 4 = 44$ 个时钟周期。计算出4个Y值，所以计算一个Y值需要11个时钟周期。加速比为 $14/11 \approx 1.27$ 。加速主要是来自减少控制开销，即减少对R1、R2的整型操作以及比较、分支指令带来的。

(3)

L.D	F2, 0(R1)
L.D	F8, 8(R1)
L.D	F14, 16(R1)
L.D	F20, 24(R1)
MUT.D	F4, F2, F0
MUT.D	F10, F8, F0
MUT.D	F16, F14, F0
MUT.D	F22, F20, F0
L.D	F6, 0(R2)
L.D	F12, 8(R2)
L.D	F18, 16(R2)
L.D	F24, 24(R2)
ADD.D	F6, F4, F6
ADD.D	F12, F10, F12
ADD.D	F18, F16, F18

ADD.D	F24, F22, F24
S.D	F6, 0(R2)
S.D	F12, 8(R2)
S.D	F18, 16(R2)
S.D	F24, 24(R2)
DADDIU	R1, R1, #32
DADDIU	R2, R2, #32
DSUBIU	R3, R1, #DONE
BNEZ	R3, F00

共用324个时钟周期，计算一个Y值平均需要 $24/4 = 6$ 个时钟周期，加速比为 $14/6 \approx 2.33$

作业

1953729

吴浩泽

5.1 ①命中时间: 访问 Cache 命中时所用的时间. ②不命中率: CPU 访存时, 在一级存储器中找不到所需信息的概率. ③不命中开销: CPU 向二级存储器发出访问请求到把这个数据调入一级存储器所需时间. ④直接映像: 主存中的每一块只能被放置到 Cache 中唯一的一个地方. ⑤组相联映像: 将 Cache 分成若干组, 每组由若干块组成. 主存中的每块可以放置到 Cache 中唯一的一组中任何一个地方.

5.2 存储层次

比较项目

Cache-主存层次

主存-辅存层次

目的

弥补主存速度的不足

弥补主存容量的不足

存储管理的实现

全部由专用硬件实现

主要由软件实现

访问速度的比值
(一级比二级)

几比一

几万比一

典型的块(页)大小

几~几十字节

几千字节或更大

CPU对第二级的访问方式

可直接访问

均通过第一级

不命中时CPU是否切换

不切换

切换到其他进程

5.10. 平均访问时间 = 命中时间 + 失效率 \times 失效开销

$$\text{平均访问时间}_{1\text{路}} = 2.0 + 1.4\% \times 80 = 3.12\text{ns}$$

$$\text{平均访问时间}_{2\text{路}} = 2.0 \times (1 + 10\%) + 1.0\% \times 80 = 3.0\text{ns}$$

$$\text{访问速度比} = 3.12 / 3 = 1.04$$

2路组相联的平均访问时间比较1.

$$\text{CPU时间} = (\text{CPU执行} + \text{存储器等待周期}) \times \text{时钟周期}$$

$$\text{CPU时间} = \text{IC} \times (\text{CPI执行} + \text{总失效率} / \text{指令量数} \times \text{失效开销})$$

$$\times \text{时钟周期} = \text{IC} [(\text{CPI执行} \times \text{时钟周期}) + (\text{每条指令的访存次数} \times \text{失效率} \times \text{失效开销} \times \text{时钟周期})]$$

$$\text{CPU时间-1路} = \text{IC} (2.0 \times 2 + 1.2 \times 0.014 \times 80) = 5.344\text{C}$$

$$\text{CPU时间-2路} = \text{IC} (2.2 \times 2 + 1.2 \times 0.01 \times 80) = 5.36\text{C}$$

$$\text{相对性能比} = \frac{\text{CPU时间-2路}}{\text{CPU时间-1路}} = 5.36 / 5.344 \approx 1.003$$

直接映像Cache的访问速度比2路组相联映像Cache快1.04倍, 而2路平均性能高1.003倍, 因此选2路组相联.