

## 第六章 指令流水线答案

3. 假定在一个五级流水线（如图 7.5 所示）处理器中，各主要功能单元的操作时间为：存储单元：200ps；ALU 和加法器：150ps；寄存器堆读口或写口：50ps。请问：

（1）若执行阶段 EX 所用的 ALU 操作时间缩短 20%，则能否加快流水线执行速度？如果能的话，能加快多少？如果不能的话，为什么？

（2）若 ALU 操作时间增加 20%，对流水线的性能有何影响？

（3）若 ALU 操作时间增加 40%，对流水线的性能有何影响？

参考答案：

（1）ALU 操作时间缩短 20% 不能加快流水线指令速度。因为存储单元的时间为 200ps，所以流水线的时钟周期不会因为 ALU 操作时间的缩短而变短。

（2）ALU 操作时间延长 20% 时，变为 180ps，比 200ps 小，对流水线性能没有影响；

（3）ALU 操作时间延长 40% 时，变为 210ps，比 200ps 大，所以，流水线的时钟周期将变为 210，其效率降低了  $(210-200)/200=5\%$ 。

6. 以下指令序列中，哪些指令对发生数据相关？假定采用“取指、译码/取数、执行、访存、写回”五段流水线方式，那么不用“转发”技术的话，需要在发生数据相关的指令前加入几条 nop 指令才能使这段程序避免数据冒险？如果采用“转发”是否可以完全解决数据冒险？不行的话，需要在发生数据相关的指令前加入几条 nop 指令才能使这段程序不发生数据冒险？

```
add $s3, $s1, $s0
sub $t2, $s0, $s3
lw  $t1, 0($t2)
add  $t1, $t1, $t2
```

参考答案：

发生数据相关的有：第 1 和 2 间关于 \$s3、第 2 和 3 间关于 \$t2、第 2 和 4 间关于 \$t2、第 3 和 4 间关于 \$t1。

不进行“转发”处理的话，需要分别在第 2、3、4 条指令前加三条 nop 指令才能避免数据冒险。而通过“转发”可以避免 1 和 2、2 和 3、2 和 4 间的数据相关；但第 3 和 4 间是 load-use 数据相关，所以无法用“转发”消除冒险，因此，需在第 4 条指令前加入一条 nop 指令。

7. 假定以下 MIPS 指令序列在图 7.18 所示的流水线数据通路中执行：

```
addu  $s3, $s1, $s0
subu  $t2, $s3, $s3
lw    $t1, 0($t2)
add   $t3, $t1, $t2
add   $t1, $s4, $s5
```

请问：

（1）上述指令序列中，哪些指令的哪个寄存器需要转发，转发到何处？

（2）上述指令序列中，是否存在 load-use 数据冒险？

（3）第 5 周期结束时，各指令执行状态是什么？哪些寄存器的数据正被读出？哪些寄存器将被写入？

参考答案：

（1）发生数据相关的有：第 1 和 2 间关于 \$s3、第 2 和 3 间关于 \$t2、第 2 和 4 间关于 \$t2、第 3 和 4 间关于 \$t1。通过“转发”可以避免 1 和 2、2 和 3、2 和 4 间的数据相关；

（2）第 3 和 4 间是 load-use 数据相关，所以无法用“转发”消除冒险。

（3）第五个时钟各条指令的执行情况如下：

指令 1 在“WB”阶段，控制信息等在 MEM/WB.Reg 中，\$s3 正在被写

指令 2 在“MEM”阶段，控制信息等在 EX/MEM.Reg 中

指令 3 在“EXE”阶段，控制信息等在 ID/EX.Reg 中

指令 4 在“ID/REG”阶段，指令在 IF/ID.Reg 中，\$t1 和 \$t2 正在被读出

指令 5 在“IF”阶段，指令正被读出

10. 在一个采用“取指、译码/取数、执行、访存、写回”的五段流水线中，若检测结果是否为“零”的操作在执行阶段进行，则分支延迟损失时间片（即分支延迟槽）为多少？以下一段 MIPS 指令序列中，在考虑数据转发的情况下，哪些指令执行时会发生流水线阻塞？各需要阻塞几个时钟周期？

```
loop: add $t1, $s3, $s3
      add $t1, $t1, $t1
      add $t1, $t1, $s6
```

```
lw    $t0, 0($t1)
bne   $t0, $s5, Exit
add   $s3, $s3, $s4
j      Loop
```

Exit:

参考答案:

若检测操作在执行阶段进行, 则分支延迟损失时间片 (即分支延迟槽) 为 2。

**bne** 指令执行时会发生流水线阻塞, 阻塞 1 个时钟周期。**j** 指令如果在译码阶段就根据译码结果计算跳转目标地址, 那么 **j** 指令后面指令会被阻塞 1 个时钟周期, 若在执行阶段计算, 则要阻塞 2 个时钟周期。

//\*\*\*\*\* (以下为没有布置作业的其它题目的参考答案)

4. 假定某计算机工程师想设计一个新 CPU, 一个典型程序的核心模块有一百万条指令, 每条指令执行时间为 100ps。请问:

- (1) 在非流水线处理器上执行该程序需要花多长时间?
- (2) 若新 CPU 是一个 20 级流水线处理器, 执行上述同样的程序, 理想情况下, 它比非流水线处理器快多少?
- (3) 实际流水线并不是理想的, 流水段间数据传送会有额外开销。这些开销是否会影响指令执行时间 (Instruction latency) 和指令吞吐率 (Instruction throughput)?

参考答案:

- (1) 非流水线处理器上执行该程序的时间为:  $100\text{ps} \times 10^6 = 100\mu\text{s}$ 。
- (2) 若在一个 20 级流水线的处理器上执行, 理想情况下, 每个时钟周期为:  $100/20 = 5\text{ps}$ , 所以, 程序执行时间约为  $5 \times 10^6 = 5\mu\text{s}$ 。快  $100/5 = 20$  倍。
- (3) 流水线段之间数据的传递产生的额外开销, 使得一条指令的执行时间被延长, 即影响 Instruction latency; 同时也拉长了每个流水段的执行时间, 即影响 Instruction throughput。

5. 假定最复杂的一条指令所用的组合逻辑分成 6 块, 依次为 A~F, 其延迟分别为 80ps、30ps、60ps、50ps、70ps、10ps。在这些组合逻辑块之间插入必要的流水段寄存器就可实现相应的指令流水线, 寄存器延迟为 20ps。理想情况下, 以下各种方式所得到的时钟周期、指令吞吐率和指令执行时间各是多少? 应该在哪里插入流水线寄存器?

- (1) 插入一个流水段寄存器, 得到一个两级流水线
- (2) 插入两个流水段寄存器, 得到一个三级流水线
- (3) 插入三个流水段寄存器, 得到一个四级流水线
- (4) 吞吐量最大的流水线

参考答案:

(1) 两级流水线的平衡点在 C 和 D 之间, 其前面一个流水段的组合逻辑延时为  $80+30+60=170\text{ps}$ , 后面一个流水段的组合逻辑延时为  $50+70+10=130\text{ps}$ 。这样每个流水段都以最长延时调整为  $170+20=190\text{ps}$ , 故时钟周期为 190ps, 指令吞吐率为  $1/190\text{ps}=5.26\text{GOPS}$ , 每条指令的执行时间为  $2 \times 190 = 380\text{ps}$ 。

(2) 两个流水段寄存器分别插在 B 和 C、D 和 E 之间, 这样第一个流水段的组合逻辑延时为  $80+30=110\text{ps}$ , 中间第二段的时延为  $60+50=110\text{ps}$ , 最后一段延时为  $70+10=80\text{ps}$ 。这样每个流水段都以最长延时调整为  $110+20=130\text{ps}$ , 故时钟周期为 130ps, 指令吞吐率为  $1/130\text{ps}=7.69\text{GOPS}$ , 每条指令的执行时间为  $3 \times 130 = 390\text{ps}$ 。

(3) 三个流水段寄存器分别插在 A 和 B、C 和 D、D 和 E 之间, 这样第一个流水段的组合逻辑延时为 80ps, 第二段时延为  $30+60=90\text{ps}$ , 第三段时延为 50ps, 最后一段延时为  $70+10=80\text{ps}$ 。这样每个流水段都以最长延时调整为  $90+20=110\text{ps}$ , 故时钟周期为 110ps, 指令吞吐率为  $1/110\text{ps}=9.09\text{GOPS}$ , 每条指令的执行时间为  $4 \times 110 = 440\text{ps}$ 。

(4) 因为所有组合逻辑块中最长延时为 80ps, 所以, 达到最大可能吞吐率的划分应该是以一个流水段延时为  $80\text{ps}+20\text{ps}$  来进行, 因此, 至少按五段来划分, 分别把流水段寄存器插在 A 和 B、B 和 C、C 和 D、D 和 E 之间, 这样第一段的组合逻辑延时为 80ps, 第二段为 30ps, 第三段为 60ps, 第四段为 50ps, 最后一段为  $70+10=80\text{ps}$ 。这样每个流水段都以最长延时调整为  $80+20=100\text{ps}$ , 故时钟周期为 100ps, 指令吞吐率为  $1/100\text{ps}=10\text{GOPS}$ , 每条指令的执行时间为  $5 \times 100 = 500\text{ps}$ 。

8. 假定有一个程序的指令序列为“lw, add, lw, add, ...”。add 指令仅依赖它前面的 lw 指令，而 lw 指令也仅依赖它前面的 add 指令，寄存器写口和寄存器读口分别在一个时钟周期的前、后半周期内独立工作。请问：

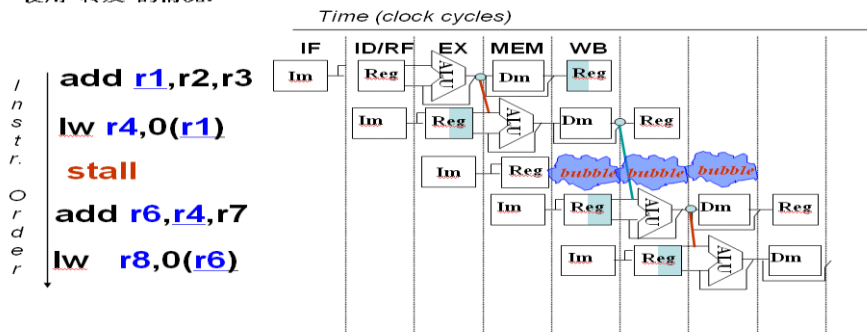
(1) 在带转发的五段流水线中执行该程序，其 CPI 为多少？

(2) 在不带转发的五段流水线中执行该程序，其 CPI 为多少？

参考答案：

(1) 因为 lw 指令和 add 指令之间存在一个 load-use 数据冒险，所以每个 lw 指令和 add 指令之间要有一次流水线阻塞。而 add 指令和 lw 指令之间的数据冒险可通过数据转发解决。即：CPI 为 1.5。

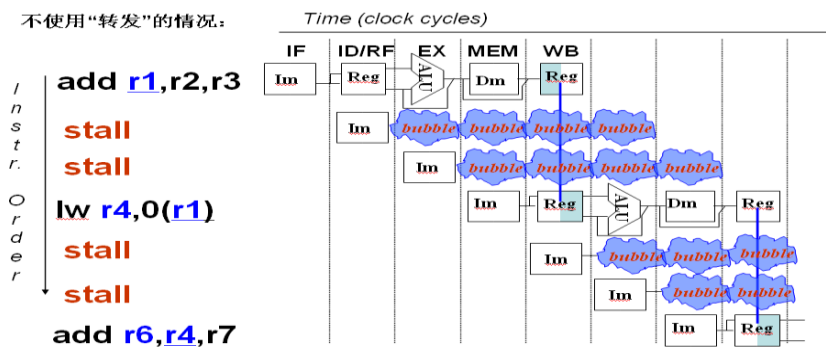
使用“转发”的情况：



使用“转发”时，只有lw指令后需要一次阻塞！

(2) 如果没有转发，而寄存器写口和寄存器读口分别在一个时钟周期的前、后半周期内工作，则在每条 lw 指令和 add 指令之间将会有两个阻塞，这样每条指令相当于都要有三个时钟才能完成。即：CPI 为 3

不使用“转发”的情况：



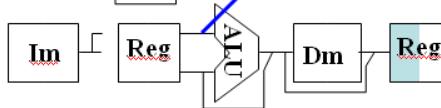
通过寄存器写口/读口分别安排在前半/后半周期，在不使用“转发”时使得每条指令之间只要阻塞两次就可解决！

9. 假定在一个带转发功能的五段流水线中执行以下程序段，则可以怎样调整以下指令序列使其性能达到最好？

```
lw    $2, 100($6)
add   $2, $2, $3
lw    $3, 200($7)
add   $6, $4, $7
sub   $3, $4, $6
lw    $2, 300($8)
beq   $2, $8, Loop
```

参考答案：

```
lw    $2, 100($6)
add   $6, $4, $7
add   $2, $2, $3
lw    $3, 200($7)
lw    $2, 300($8)
sub   $3, $4, $6
beq   $2, $8, Loop
```



11. 假设数据通路中各主要功能单元的操作时间为: 存储单元: 200ps; ALU 和加法器: 100ps; 寄存器堆读口或写口: 50ps。程序中指令的组成比例为: 取数 25%、存数 10%、ALU 52%、分支 11%、跳转 2%。假设时钟周期取存储器存取时间的一半, MUX、控制单元、PC、扩展器和传输线路等的延迟都忽略不计, 则下面的实现方式中, 哪个更快? 快多少?
- (1) 单周期方式: 每条指令在一个固定长度的时钟周期内完成;
  - (2) 多周期方式: 每类指令时钟数: 取数-7, 存数-6, ALU-5, 分支-4, 跳转-4;
  - (3) 流水线方式: 取指 1、取指 2、取数/译码、执行、存取 1、存取 2、写回 7 段流水线; 没有结构冒险; 数据冒险采用“转发”技术处理; load 指令与后续各指令之间存在依赖关系的概率分别 1/2、1/4、1/8、...; 分支延迟损失时间片为 2, 预测准确率为 75%; 不考虑异常、中断和访问失效引起的流水线冒险。

参考答案:

单周期: 存储器操作变为两个时钟周期后, 其数据通路的时钟周期不变, 为 600ps

多周期:  $CPI = 0.25 \times 7 + 0.10 \times 6 + 0.52 \times 5 + 0.11 \times 4 + 0.02 \times 4 = 5.47$

存储器操作变为两个时钟周期后, 多周期数据通路的时钟周期为 100ps,

故一条指令的执行时间为  $100 \times 5.47 = 547ps$

流水线: 存储器操作变为两个时钟周期后, 其流水线包含了 7 个阶段。

对于 beq, 若预测正确, 则为 1 个周期, 若预测错误, 则为 3 个周期 (与原五段流水线相比, 多一个取指周期, 多阻塞了 1 个周期), 故  $CPI = 1/4 \times 3 + 3/4 \times 1 = 1.5$

对于 load, 随后第一条则为 3 个 (阻塞 2 个) 周期; 随后第二条则为 2 个 (阻塞 1 个) 周期, 以后的指令都不需要阻塞, 故  $CPI = 1/2 \times 3 + 1/4 \times 2 + 3/8 \times 1 = 2.25$

对于 ALU 指令, 随后的数据相关指令都可通过转发解决, 故  $CPI = 1$

对于 Store 指令, 不会发生数据冒险, 故  $CPI = 1$

对于 Jump 指令, 总要等到译码结束才能确定转移地址, 故  $CPI = 2$

平均 CPI 为:  $2.25 \times 25\% + 1 \times 10\% + 1 \times 52\% + 1.5 \times 11\% + 2 \times 2\% = 1.39$

所以, N 条指令的执行时间为  $1.39 \times 100 \times N = 139N(ps)$

流水线速度是单周期的  $600/139 = 4.32$  倍, 是多周期的  $547/139 = 3.94$  倍。

1. 假设有一段程序的核心模块中有五条分支指令, 该模块将会被执行成千上万次, 在其中一次执行过程中, 五条分支指令的实际执行情况如下 (T: Taken; N: not Taken)。

分支指令 1 (B1): T - T - T。

分支指令 2 (B2): N - N - N - N。

分支指令 3 (B3): T - N - T - N - T - N。

分支指令 4 (B4): T - T - T - N - T。

分支指令 5 (B5): T - T - N - T - T - N - T。

假定各个分支指令在每次模块执行过程中实际执行情况都一样, 并且动态预测时, 每个分支指令都有各自的预测表项, 每次执行时的初始预测位都相同。请给出以下几种预测方案的预测准确率。

(1) 静态预测, 总是预测转移 (Taken)。

(2) 静态预测, 总是预测不转移 (not Taken)。

(3) 一位动态预测, 初始预测转移 (Taken)。

(4) 二位动态预测, 初始预测弱转移 (Taken)。

【分析解答】

预测准确率 = 预测正确次数 / 总预测次数  $\times 100\%$ 。以下 R 表示正确预测次数, W 表示错误预测次数。

- (1) B1: R-3, W-0; B2: R-0, W-4; B3: R-3, W-3; B4: R-4, W-1; B5: R-5, W-2; 60%
- (2) B1: R-0, W-3; B2: R-4, W-0; B3: R-3, W-3; B4: R-1, W-4; B5: R-2, W-5; 40%
- (3) B1: R-3, W-0; B2: R-3, W-1; B3: R-1, W-5; B4: R-3, W-2; B5: R-3, W-4; 52%
- (4) B1: R-3, W-0; B2: R-3, W-1; B3: R-3, W-3; B4: R-4, W-1; B5: R-5, W-2; 72%