

2023 春季学期-计算机组成原理-作业四

(要求：独立完成，5 月 26 日上课时上交)

1. (8 points) (课本 P203 页习题 3) 假定在一个 5 级流水线处理器中，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，能加快多少？如果不能，为什么？

(2) 若 ALU 操作时间增加 20%，则对流水线的性能有何影响？

(3) 若 ALU 操作时间增加 40%，则对流水线的性能有何影响？

(1) 流水线的执行速度取决于最慢的功能部件所用时间，因此，将 ALU 操作时间缩短不能加快流水线执行速度。

(2) ALU 执行时间延长 20% 以后，执行时间变为 180ps，仍然比存储器时间短，因此对流水线仍然没有影响。

(3) ALU 执行时间延长 40% 以后，执行时间变为 210ps，此时，流水线的时钟周期变为 210ps，比原来速度降低了 5%。

2. (8 points) (课本 P203 页习题 5) 假定最复杂的一条指令所用的组合逻辑分成 6 部分，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以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段寄存器？

(1) 插入 1 个流水段寄存器，得到一个两级流水线。

(2) 插入 2 个流水段寄存器，得到一个三级流水线。

(3) 插入 3 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。

(1) 流水线的划分原则：各段延迟尽量均衡。

插入一个流水段寄存器时，最优的划分方式为 80ps 30ps 60ps | 50ps 70ps 10ps，此时，两段组合电路的延迟分别为 170ps 和 130ps。加上流水段寄存器的 20ps 延时，时钟周期应为 190ps，吞吐率为每秒执行 $1/190ps = 5.26 \times 10^9$ 条指令，每条指令执行时间为 $190ps \times 2 = 380ps$ 。

(2) 插入两个流水段寄存器时，最优的划分方式为 80ps 30ps | 60ps 50ps | 70ps 10ps，此时，三段组合电路的延迟分别为 110ps、110ps 和 80ps。加上流水段寄存器的 20ps 延时，时钟周期应为 130ps，吞吐率为每秒执行 $1/130ps = 7.69 \times 10^9$ 条指令，每条指令执行时间为 $130ps \times 3 = 390ps$ 。

(3) 插入三个流水段寄存器时，最优的划分方式为 80ps | 30ps 60ps | 50ps | 70ps 10ps，此时，四段组合电路的延迟分别为 80ps、90ps、50ps 和 80ps。加上流水段寄存器的 20ps 延时，时钟周期应为 110ps，吞吐率为每秒执行 $1/110ps = 9.09 \times 10^9$ 条指令，每条指令执行时间为 $110ps \times 4 = 440ps$ 。

(4) 要设计吞吐量最大的流水线，需要使每个流水段的延迟最小。本题中，组合电路部分最大延迟为 80ps，因此划分方式为 80ps | 30ps | 60ps | 50ps | 70ps 10ps，是一个 5 段流水线设计。组合电路

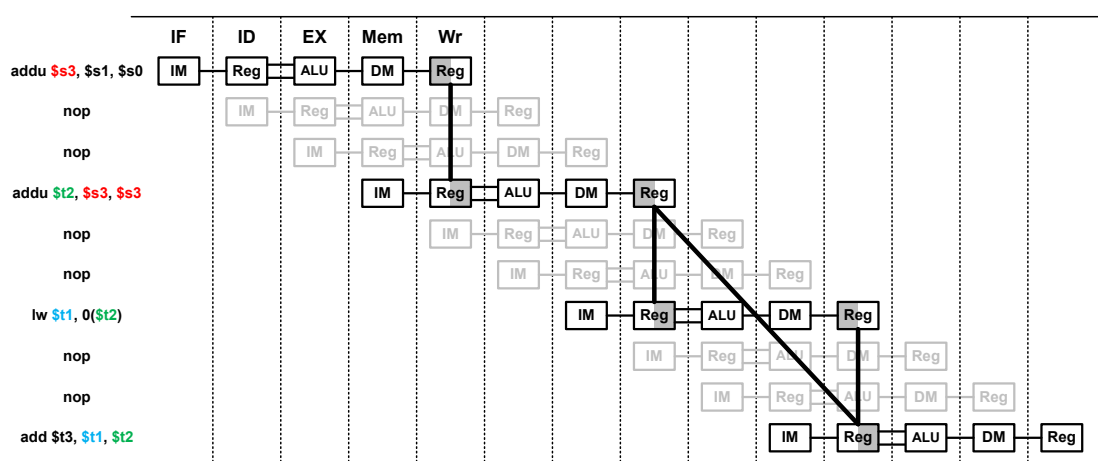
延迟加上流水段寄存器的 $20ps$ 延时, 时钟周期应为 $100ps$, 吞吐率为每秒执行 $1/100ps = 10 \times 10^9$ 条指令, 每条指令执行时间为 $100ps \times 5 = 500ps$ 。

流水线划分的流水段越多, 时钟周期越短, 吞吐量越大。但是, 流水段寄存器引入的额外开销变大, 使得一条指令的执行时间变长。

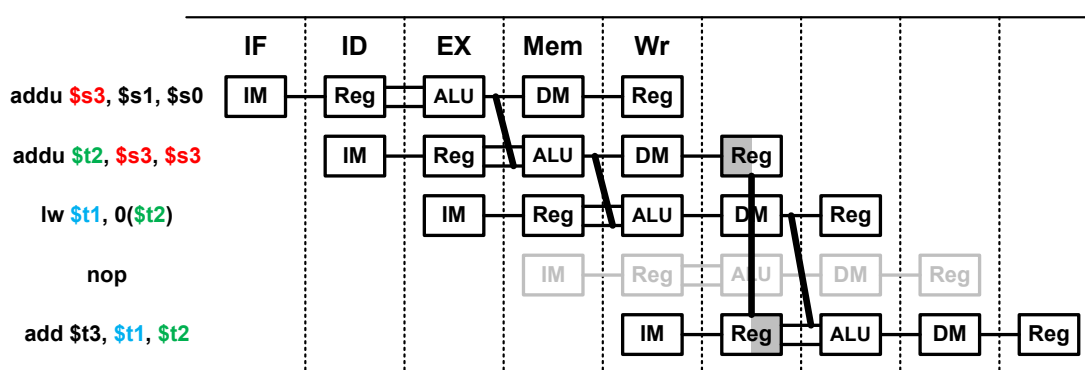
3. (8 points) (课本 P204 页习题 6) 以下指令序列中, 哪些指令对之间发生数据相关? 假定采用“取指、译码/取数、执行、访存、写回”5 段流水线方式, 如果不用转发技术, 需要在发生数据相关的指令前加入几条 `nop` 指令才能使这段程序避免数据冒险? 如果采用转发技术是否可以完全解决数据冒险? 不行的话, 需要在发生数据相关的指令前加入几条 `nop` 指令才能使这段 MIPS 程序不发生数据冒险?

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

不用转发技术, 假设寄存器的写入和读取分别安排在时钟周期的前、后半周期。需要两条 `nop`。



使用转发技术, 可以消除部分 `nop`, 但是 `load-use` 冒险仍然需要一个 `nop`。



4. (8 points) (课本 P204 页习题 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 5 段流水线中执行该程序，其 CPI 是多少？

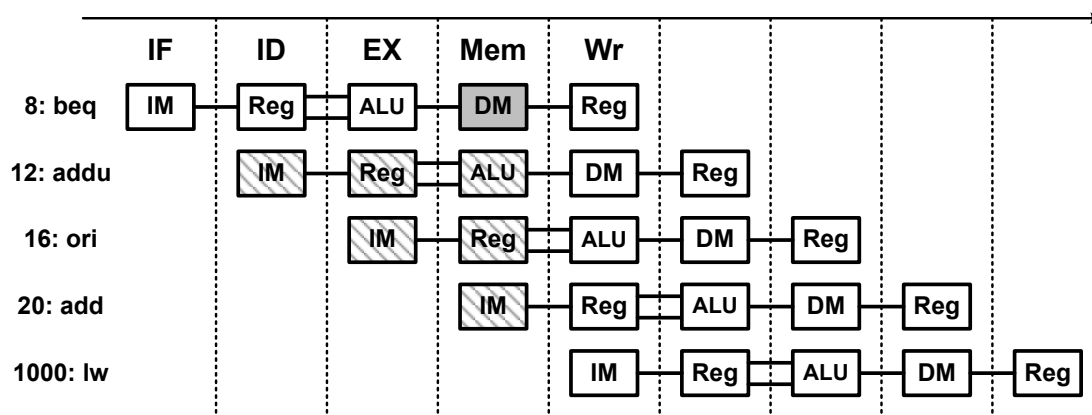
(2) 在不带转发的 5 段流水线中执行该程序，其 CPI 是多少？

(1) 若流水线包含转发电路，且寄存器每个周期内可以进行读写两个操作，则只存在 lw 和 add 之间的一个 load-use 冒险，即每个 lw 和 add 之间有一次流水线阻塞。因此，每对指令需要三个时钟周期，即 $CPI = 1.5$ 。

(2) 若没有转发电路，由于寄存器每个周期内可以进行读写两个操作，则每两条指令之间会有两个周期的阻塞，相当于每条指令需要三个时钟周期才能完成，即 $CPI = 3$ 。

5. (8 points) (课本 P204 页习题 10) 在一个采用“取指、译码/取数、执行、访存、写回”的 5 段流水线中，若检测结果是否为 0 和将转移目标地址 (Btarg 和 Jtarg) 送 PC 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？在带转发的 5 段流水线中，对于以下 MIPS 指令序列，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:
```



若判断结果是否为 0 的操作在执行阶段进行，则执行阶段完成后就可以判定是否需要跳转，则分支延迟损失时间片 $C = 2$ 。

由于存在数据转发电路，则除了 load-use 之外的数据冒险都可以通过转发来解决，因此，会发生流水线阻塞的情况包括 load-use 冒险以及 bne 和 j 引起的控制冒险：

- 题目中, lw 和 bne 存在关于 \$t0 的 load-use 冒险, 需要阻塞流水线一个时钟周期。
- 对于 bne 引起的控制冒险, 若检测结果是否为零并更新 PC 的操作发生在访存阶段, 则分支延迟槽 $C = 3$, 即需要阻塞流水线三个时钟周期。若检测发生在执行阶段, 则只需要阻塞两个时钟周期。
- 对于 j 引起的控制冒险, 若更新 PC 的操作发生在执行阶段, 则流水线需要被阻塞两个时钟周期。若更新 PC 的操作发生在译码阶段, 则流水线只需要被阻塞一个时钟周期。

6. (8 points) (课本 P205 页习题 11) 假设数据通路中各主要功能部件的操作时间是: 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 为 52%、分支 11%、跳转 2%。假设控制单元和传输线路等延迟都忽略不计, 则以下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。时钟周期取存储单元操作时间的一半, 每类指令时钟数是: 取数为 7、存数为 6、ALU 为 5、分支为 4、跳转为 4。

(3) 流水线方式。时钟周期取存储单元操作时间的一半, 采用“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为 $1/2, 1/4, 1/8, \dots$; 分支延迟损失时间片为 2, 预测准确率为 75%; 不考虑异常、中断和访问缺失引起的流水线冒险。

(1) 单周期方式中, 执行时间最长的指令为 lw, 其需要的时钟周期为 $200ps + 50ps + 100ps + 200ps + 50ps = 600ps$, 因此指令的执行时间为 600ps。

(2) 多周期方式中, $CPI = 0.25 \times 7 + 0.1 \times 6 + 0.52 \times 5 + 0.11 \times 4 + 0.02 \times 4 = 5.47$ 。由于时钟周期为 100ps, 执行一条指令的时间为 $100ps \times 5.47 = 547ps$ 。

(3) 流水线方式中, 对于涉及的指令类型分别做分析:

- 对于取数指令, 虽然存在转发电路, 发生 load-use 数据冒险时仍然需要阻塞流水线。由于访存操作占用两个时钟周期, 因此, load 指令后第一条指令需要阻塞两个时钟周期, 其后第二条指令需要阻塞一个时钟周期。因此, 取数指令的 $CPI = 1/2 \times 3 + 1/4 \times 2 + (1 - 1/2 - 1/4) \times 1 = 2.25$ 。
- 对于存数指令, 不存在数据冒险, 其 $CPI = 1$ 。
- 对于 ALU 指令, 其数据冒险可通过数据转发解决, 因此其 $CPI = 1$ 。
- 对于分支指令, 由于分支延迟损失时间片为 2, 因此预测失败时, 需要 3 个时钟周期。分支指令的 $CPI = 0.75 \times 1 + 0.25 \times 3 = 1.5$ 。
- 对于跳转指令, 最快需要在译码阶段确定转移地址, 因此需要阻塞两个时钟周期, 其 $CPI = 3$ 。

综合以上情况, 流水线方式的 $CPI = 0.25 \times 2.25 + 0.1 \times 1 + 0.52 \times 1 + 0.11 \times 1.5 + 0.02 \times 3 = 1.4075$ 。平均一条指令的执行时间为 $100ps \times 1.4075 = 140.75ps$ 。

综合三种方式的性能, 流水线方式执行速度更快, 其速度为单周期方式的 4.26 倍, 多周期方式的 3.89 倍。