高等计算机体系结构，2020年春季

作业3： 流水线1

主讲教师：栾钟治

助讲教师：杨海龙；助教：孙庆骁

作业下发时间：2020年4月10日

作业回收时间：2020年4月24日

# 流水线 50分

观察以下程序：

MUL R3 R1, R2

ADD R5 R4, R3

ADD R6 R4, R1

MUL R7 R8, R9

ADD R4 R3, R7

MUL R10 R5, R6

分别计算该程序在以下的机器上执行时花费的时钟周期数:

(a) 非流水线机器

(b) 采用计分板（scoreboarding）的流水线机器，有5个加法器、5个乘法器，没有数据转发逻辑

(c) 采用计分板（scoreboarding）的流水线机器，有5个加法器、5个乘法器，带数据转发逻辑

(d) 采用计分板（scoreboarding）的流水线机器，有1个加法器、1个乘法器，没有数据转发逻辑

(e) 采用计分板（scoreboarding）的流水线机器，有1个加法器、1个乘法器，带数据转发逻辑

对于所有上述的机器模型，采用以下4阶段的基本指令周期:

1）取指 (1 个时钟周期)

2）译码 (1 个时钟周期)

3）执行

MUL(5个时钟周期)

ADD(2 个时钟周期)

乘法器和加法器内部不是流水线的

4） 写回 (2 个时钟周期)

请列出你为了计算对流水线结构的所有假设（例如如何在流水段之间做数据转发）

**答：首先做如下的设计假设：**

1. **非流水线机器采用多周期CPU结构，多周期CPU的设计是将整个CPU的执行过程分成几个阶段，每个阶段用一个时钟去完成。不仅能提高CPU的工作频率，还为组成指令流水线提供了基础。**
2. **记分牌算法的实现结构：**
3. **记分牌基本思想：记分牌硬件实现了对指令的动态调度。支持乱序执行，在没有结构冲突时尽早地执行没有数据冲突的指令，使多条指令同时处于执行阶段。**
4. **动态调度技术需要将ID译码段分成两个阶段：1是发射，2是读取操作数。**

**发射阶段对指令进行译码，检查结构冒险（例如有四个运算器：整数运算、加法器、乘法器、除法器，检查该指令需要使用的运算器是否正在被占用）。**

**读取操作数阶段检查数据冒险（读之前检查寄存器的值是否已经写回，或者是否会覆盖了之前的值）。数据冒险的解决方法（读写冒险(RAW)：将指令和操作数保存起来，然后只能在读操作数阶段进行读取；写写冒险(WAW)：检测是否有其它指令会写回到相同的寄存器（检测到冒险），有则等待，直到其它的完成）**

1. **发射阶段：假如检测到没有结构冒险和数据冒险，那么记分板将会将指令发射到相关的运算器，假如结构冒险或者写写冒险发生了，那么该指令将会等待，直到冒险消失为止。**
2. **读取操作数：没有数据相关以后（之前的指令不会写回源寄存器或者正在写寄存器的情况时，读写冒险），读取操作数。读取操作数后将交给运算器，之后开始运算。发送到运算器的顺序可能是乱序的。**
3. **记分板由三个部分组成，维护三张表项：**

**part1 指令状态表：记录指令究竟处于4个阶段中的哪一个阶段**

**part2 功能部件状态表：有9个域，分别是：**

**a) Busy 标识该器件是否正被使用**

**b) OP 该器件正在执行或将要执行的运算 例如 + - \* / 等等**

**c) Fi 目标寄存器**

**d) Fj, Fk 源操作寄存器**

**e) Qj,Qk指出源操作寄存器Fj, Fk正在被什么单元所处理**

**f) Rj, Rk 表示Fj Fk 是否准备好的标志位,yes表示Fj, Fk中的操作数就绪且还未被取走，否则就被置为No**

**part3 寄存器结果状态表：指出哪个功能部件将把结果写入该寄存器。**

1. **假设记分牌算法按照有（2）中介绍的完整设计思路，并且设计如下的电路：**
2. **指令缓存队列，用于储存每一条译码后的指令，以及相应指令的状态。（包括发射状态Issue、读操作数状态Read Operands、执行完毕状态Execution Complete、回写状态Write Result），对应指令状态表。**
3. **功能部件状态电路，为每个功能部件设计相应的功能部件状态电路，对应功能部件状态表。**
4. **寄存器状态电路。为每个寄存器设计相应的寄存器状态电路，用于指示哪一个功能部件将要把结果写回哪一寄存器，对应寄存器结果状态表。**
5. **假设记分牌算法三个状态表的更新通过设计电路实现，不占用任何的时间周期。**
6. **假设寄存器的写操作发生在时钟周期的前半段，读寄存器发生在后半段，因此读操作可以读取到最新的写入的数值。即在WB的两个时钟周期中前一个时钟周期写入数据，后一个时钟周期ID就可以读取数据。（\*\*重点设计思路\*\*：详见(b)机器WB和ID阶段，WB两个阶段相当于MEM+WB两个，书上设计是在WB前半时钟沿写入寄存器，在后边时钟沿ID读取寄存器，参考如上设计，在WB第二个时钟周期就可以执行ID读取数据）**
7. **假设需要做数据转发的时候，数据在ALU单元计算结束就能够得到结果，即在EX的末尾就能够得到数据并进行转发。即数据转发会发生在乘法和加法的执行周期（EX）后，写回（WB）周期前。**
8. **非流水线机器**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| **MUL R3,R1,R2** | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  | **IF** | **ID** | **EX** | | **WB** | |  |  |
| **ADD R6,R4,R1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **IF** | **ID** |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **MUL R10,R5,R6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** |
| **MUL R3,R1,R2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** | **EX** | | **WB** | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  |  |  |  |  |  | **IF** | **ID** | **EX** | |
| **MUL R10,R5,R6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **35** | **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** | **46** | **47** | **48** | **49** | **50** | **51** |
| **MUL R3,R1,R2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R4,R3,R7** | **WB** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **MUL R10,R5,R6** |  |  | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |  |  |

**总共需要45个时钟周期。**

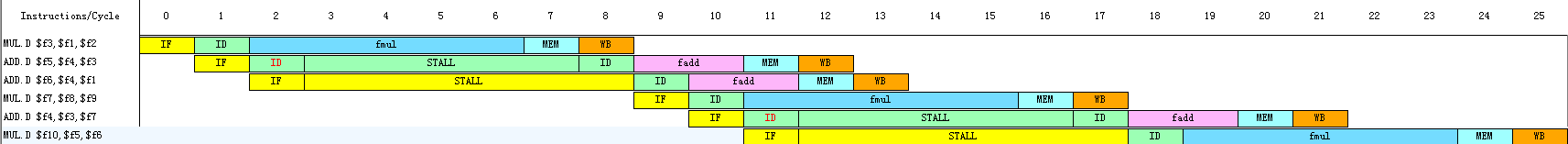
1. **采用计分板（scoreboarding）的流水线机器，有5个加法器、5个乘法器，没有数据转发逻辑**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **MUL R3,R1,R2** | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  | |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  | **IF** | **ID** | **STALL** | | | | | **ID** | **EX** | | **WB** | | |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  | **IF** | **STALL** | | | | | | **ID** | **EX** | | | **WB** | |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  | **IF** | **ID** | | **EX** | | | | | **WB** | |  |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  |  |  | **IF** | | **ID** | **STALL** | | | | | **ID** | **EX** | |
| **MUL R10,R5,R6** |  |  |  |  |  |  |  |  |  |  |  | | **IF** | **STALL** | | | | | | **ID** | **EX** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | |
| **MUL R3,R1,R2** |  |  |  |  |  |  |  |  |  |  | |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  |  | |
| **ADD R6,R4,R1** |  |  |  |  |  |  |  |  |  |  | |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  |  | |
| **ADD R4,R3,R7** | **WB** | |  |  |  |  |  |  |  | |  |
| **MUL R10,R5,R6** | **EX** | | | | **WB** | |  |  |  | |  |

**总共需要26个时钟周期。**

**用MIPSsim模拟器模拟：总共需要26个时钟周期。**

****

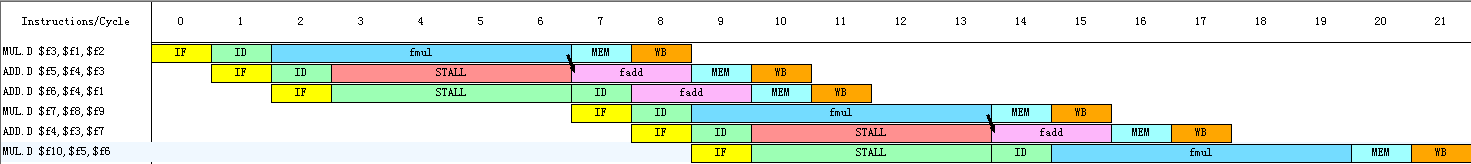
1. **采用计分板（scoreboarding）的流水线机器，有5个加法器、5个乘法器，带数据转发逻辑**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **MUL R3,R1,R2** | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  | **IF** | **ID** | **STALL** | | | | **EX** | | **WB** | |  | |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  | **IF** | **STALL** | | | | **ID** | **EX** | | **WB** | |  |  |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  | **IF** | **ID** | **STALL** | | | | **EX** | | **WB** | |  |  |
| **MUL R10,R5,R6** |  |  |  |  |  |  |  |  |  | **IF** | **STALL** | | | | **ID** | **EX** | | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **MUL R3,R1,R2** |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  |  |  |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  |  |  |
| **MUL R10,R5,R6** | **WB** | |  |  |  |  |  |  |  |  |

**总共需要22个时钟周期。**

**用MIPSsim模拟器模拟：总共需要22个时钟周期。**

****

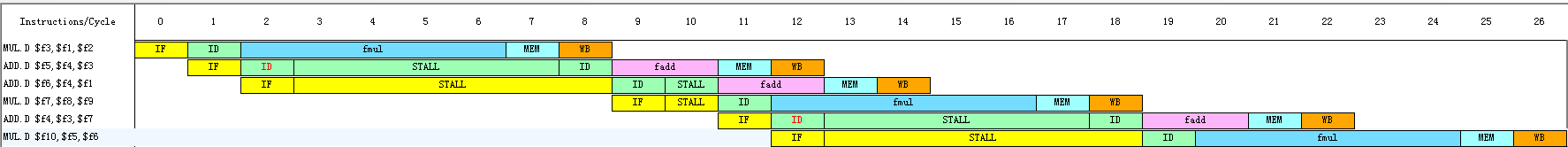
1. **采用计分板（scoreboarding）的流水线机器，有1个加法器、1个乘法器，没有数据转发逻辑**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **MUL R3,R1,R2** | **IF** | **ID** | **EX** | | | | | **WB** | |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  | **IF** | **ID** |  |  |  |  |  | **ID** | **EX** | | **WB** | |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  | **IF** |  |  |  |  |  |  | **ID** | **STALL** | **EX** | | **WB** | |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  | **IF** | **STALL** | **ID** | **EX** | | | | | **WB** | |  |
| **ADD R4,R3,R7** |  |  |  |  |  |  |  |  |  |  |  | **IF** | **ID** | **STALL** | | | | | **ID** | **EX** |
| **MULR10,R5,R6** |  |  |  |  |  |  |  |  |  |  |  |  | **IF** | **STALL** | | | | | | **ID** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **MUL R3,R1,R2** |  |  |  |  |  |  |  |  |  |  |
| **ADD R5,R4,R3** |  |  |  |  |  |  |  |  |  |  |
| **ADD R6,R4,R1** |  |  |  |  |  |  |  |  |  |  |
| **MUL R7,R8,R9** |  |  |  |  |  |  |  |  |  |  |
| **ADD R4,R3,R7** | **EX** | **WB** | |  |  |  |  |  |  |  |
| **MUL R10,R5,R6** | **EX** | | | | | **WB** | |  |  |  |

**总共需要27时钟周期。**

**用MIPSsim模拟器模拟：总共需要22个时钟周期。**

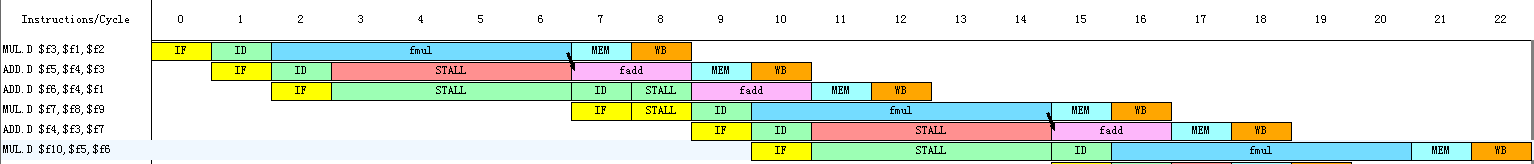
****

**(e) 采用计分板（scoreboarding）的流水线机器，有1个加法器、1个乘法器，带数据转发逻辑**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| MUL R3,R1,R2 | IF | ID | EX | | | | | WB | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ADD R5,R4,R3 |  | IF | ID | STALL | | | | EX | | WB | |  |  |  |  |  |  |  |  |  |  |  |  |
| ADD R6,R4,R1 |  |  | IF | STALL | | | | ID | STALL | EX | | WB | |  |  |  |  |  |  |  |  |  |  |
| MUL R7,R8,R9 |  |  |  |  |  |  |  | IF | STALL | ID | EX | | | | | WB | |  |  |  |  |  |  |
| ADD R4,R3,R7 |  |  |  |  |  |  |  |  |  | IF | ID | STALL | | | | EX | | WB | |  |  |  |  |
| MULR10,R5,R6 |  |  |  |  |  |  |  |  |  |  | IF | STALL | | | | ID | EX | | | | | WB | |

**总共需要23钟周期。**

**用MIPSsim模拟器模拟：总共需要23时钟周期。**

****

# 延迟槽 50分

一台五阶段流水线的机器，五个流水段分别是：取指、译码、执行、访存和写回。该机器采用延迟槽技术处理控制相关。无条件分支和有条件分支都在执行阶段获得分支的结果。

(a) 需要多少个延迟槽才能够确保正确的操作?

**答：因为题目说明了无条件分支和有条件分支都在执行阶段获得分支的结果，所以BEQ 在执行阶段后才能知道准确的跳转地址，因此需要添加2个Nop才能使程序得到正确的跳转地址。无条件分支J系指令在执行阶段才能得到准确的跳转地址，因此需要添加2个Nop。**

**TIPS：引入分支延迟槽的目的主要是为了提高流水线的效率。**

**流水线中，分支指令执行时因为确定下一条指令的目标地址（紧随其后 or 跳转目标处？）一般要到第 2 级以后，在目标确定前流水线的取指级是不能工作的，即整个流水线就“浪费”（阻塞）了一个时间片，为了利用这个时间片，在体系结构的层面上规定跳转指令后面的一个时间片为分支延迟槽（branch delay slot）。位于分支延迟槽中的指令总是被执行，与分支发生与否没有关系。这样就有效利用了一个时间片，消除了流水线的一个“气泡”。**

(b) 按照你在(a)中设计的延迟槽数量，下列汇编指令序列中，哪（些）条指令可以放入延迟槽？请使用合适的延迟槽填充方案重写下边的汇编指令代码。

(i) ADD R5 R4, R3

OR R3 R1, R2

SUB R7 R5, R6

J X

延迟槽

LW R10 (R7)

ADD R6 R1, R2

X:

(ii) ADD R5 R4, R3

OR R3 R1, R2

SUB R7 R5, R6

BEQ R5 R7, X

延迟槽

LW R10 (R7)

ADD R6 R1, R2

X:

(iii) ADD R2 R4, R3

OR R5 R1, R2

SUB R7 R5, R6

BEQ R5 R7, X

延迟槽

LW R10 (R7)

ADD R6 R1, R2

X:

**答：(i)假如没有采用数据转发的时候，只有第二条和第三条能放进去，第一条和第三条有数据冲突,在没有采用数据转发逻辑时候，会产生停顿，影响流水线性能。假如有采用数据转发逻辑，那么延迟槽还可以放第一和第三条。**

**(i) ADD R5 R4, R3**

**OR R3 R1, R2**

**SUB R7 R5, R6**

**J X**

**LW R10 (R7)**

**ADD R6 R1, R2**

**X:**

**(ii)同(i)解释，假如没有采用数据转发的时候，第二条和第三条放进去才不会产生停顿，有数据转发则还可以放第一和第三条。**

**(ii) ADD R5 R4, R3**

**BEQ R5 R7, X**

**OR R3 R1, R2**

**SUB R7 R5, R6**

**LW R10 (R7)**

**ADD R6 R1, R2**

**X:**

**(iii)这里BEQ指令判断R5和R7是否相等，R5和R7的结果取决于OR R5 R1, R2和SUB R7 R5, R6的结果，因此无法放进延迟槽中，而第一条ADD结果写入R2需要在OR中用到R2的结果所以也不能放入分支延迟槽。**

**(iii) ADD R2 R4, R3**

**OR R5 R1, R2**

**SUB R7 R5, R6**

**BEQ R5 R7, X**

**Nop**

**Nop**

**LW R10 (R7)**

**ADD R6 R1, R2**

**X:**

(c) 你能修改流水线减少延迟槽的数量吗(不使用分支预测的技术)? 请清楚地说明你的方法并解释为什么这样能减少延迟槽。

**答：使用延迟时隙法，将分支前一些无关指令填充至延迟槽处执行即可。这样执行的都是有用指令，延迟槽被有用指令填充，即减少了延迟槽。**