

第6章 布置作业

- 课后3、4、5、6、7、11

P232 第3题 参考答案

(1) ALU操作时间缩短20%，不能加快流水线执行速度，因为指令流水线的执行速度取决于最慢的功能部件所用时间，最慢的是存储器，只有缩短了存储器的操作时间才可能加快流水线速度。

(2) ALU操作时间延长20%时，变为180ps，比存储器所用时间200ps还小，因此，对流水线性能没有影响。

(3) ALU操作时间延长40%时，变为210ps，比存储器所用时间200ps大，因此，在不考虑流水段寄存器延时的情况下，流水线的时钟周期从200ps变为210ps，流水线执行速度降低了 $(210-200)/200=5\%$ 。

P232 第4题 参考答案

ms(毫秒), μs (微秒), ns(纳秒), ps(皮秒) 10^{-12}

(1) 非流水线处理器上执行该程序的时间大约为： $100\text{ps} \times 10^6 = 100\mu\text{s}$ 。

(2) 若在一个20级流水线的处理器上执行，忽略流水段之间的寄存器延时，理想情况下，每个时钟周期为 $100/20 = 5\text{ps}$ ，所以程序执行时间大约为 $5 \times 10^6 = 5\mu\text{s}$ 。因此，大约快 $100/5 = 20$ 倍。

(3) 流水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，即影响了指令执行时间；同时也延长了每个流水段的时间，即影响了指令吞吐率。

P233 第5题 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 $50+70+10=130\text{ps}$ 。最长功能段延时为 170ps ，加上流水段寄存器延时 20ps ，因而时钟周期为 190ps ，理想情况下，指令吞吐率为每秒钟执行 $1/190\text{ps}=5.26\text{G}$ 条指令。每条指令在流水线中的执行时间为 $2*190=380\text{ps}$ 。

(2) 两级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 $80+30=110\text{ps}$ ，中间第二段的延时为 $60+50=110\text{ps}$ ，最后一段延时为 $70+10=80\text{ps}$ 。这样每段流水段所用时间都按最长延时调整为 $110+20=130\text{ps}$ ，因而时钟周期为 130ps ，指令吞吐率为每秒钟执行 $1/130\text{ps}=7.69\text{G}$ 条指令。每条指令在流水线中的执行时间为 $3*130=390\text{ps}$ 。

(3) 三级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 ，中间第二段的延时为 $30+60=90\text{ps}$ ，第三段延时为 50ps ，最后一段延时为 $70+10=80\text{ps}$ 。这样每段流水段所用时间都按最长延时调整为 $90+20=110\text{ps}$ ，因而时钟周期为 110ps ，指令吞吐率为每秒钟执行 $1/110\text{ps}=9.09\text{G}$ 条指令。每条指令在流水线中的执行时间为 $4*110=440\text{ps}$ 。

(4) 因为各功能部件对应的组合逻辑中最长延时为 80ps ，最合理的划分是每个时钟周期 $80+20=100\text{ps}$ 来进行，流水段至少按5段来划分，分别把流水段寄存器插入在A和B、B和C、C和D、D和E之间，这样各段的组合逻辑延时为 80ps 、 30ps 、 60ps 、 50ps 、 80ps 。其中，最后一个延时 80ps 是E和F两个阶段的时间相加而得到的。这样时钟周期为 100ps ，指令吞吐率为每秒钟执行 $1/100\text{ps}=10\text{G}$ 条指令，每个指令的执行时间为 $5*100=500\text{ps}$ 。

结论：划分的流水段多，时钟周期就变短，指令执行吞吐率就变高，而相应的额外开销（即插入的流水段寄存器的延时）也变大，使得一条指令的执行时间变长。

P233 第6题 参考答案

addu \$s3, \$s1, \$s0

addu \$t2, \$s3, \$s3

lw \$t1, 0(\$t2)

add \$t3, \$t1, \$t2

(1) 发生数据相关的是：第1和第2条指令之间关于\$s3，第2和第3条之间关于\$t2，第2和第4条指令之间关于\$t2，第3和第4条指令之间关于\$t1。

(2) 不采用“转发”处理的话，需要分别在第2、3、4条指令前加2条nop指令才能避免数据冒险，共加了6条nop指令。

(3) 通过“转发”可以避免在第1和第2、第2和第3、第2和第4条指令之间的数据相关；但是第3和第4条指令之间是load-use数据相关，因此无法用“转发”消除冒险，而需要在第4条指令前加入1条nop指令。

P233 第7题 参考答案

(1)发生数据相关的是：第1和第2条指令之间关于\$s3，第2和第3条指令之间关于\$t2，第2和第4条指令之间关于\$t2，第3和第4条指令之间关于\$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和第4条指令之间是load-use数据冒险，所以无法用“转发”消除冒险。

(3)第5周期结束时，各条指令执行状态如下：

指令1在“WB”阶段，控制信息等MEM/WB.Reg中，\$s3正在被写，结束时写完。

指令2在“MEM”阶段，subu指令在该阶段进行的是空操作；在转发检测单元中，因为流水段寄存器Ex/Mem中的目的寄存器RegRd为\$t2，流水段寄存器ID/Ex中的源寄存器也为\$t2，同时流水段寄存器Ex/Mem中的RegWr信号为1，所以转发条件满足，因而此时subu指令在上一个时钟周期中的执行结果正被回送到ALU的输入端；结束时转发完成。

指令3在“EXE”阶段，ALU正在执行“add”操作，进行地址运算，ALU输出结果将被写入流水段寄存器Ex/Mem中；结束时运算完成，控制信息等ID/EX.Reg中，正在检测是否load-use冒险。

指令4在“ID/REG”阶段，指令在IF/ID.Reg中，\$t1和\$t2正在被读出。在load-use冒险检测单元中，因为流水段寄存器IF/ID中源操作数寄存器Rs为\$t1，流水段寄存器ID/Ex中的目的操作数寄存器Rt也为\$t1，同时，因为上条指令是lw，故流水段寄存器ID/Ex中的MemRead控制信号为1，所以在该阶段检测到load-use冒险条件满足，此时，需要进行load-use冒险处理，在流水线中插入一个“气泡”，将指令的执行阻塞一个时钟周期，包括以下三个步骤：一是将流水段寄存器ID/Ex中的控制信号全部清0，以保证第4条指令被堵塞一个时钟周期执行；二是将流水段寄存器IF/ID中的指令维持不变，以保证第4条指令重新译码后执行；三是将PC的值维持不变，以保证根据PC的值重新取出第5条指令。结束时完成上述工作。

指令5在“IF”阶段，指令正被读出。结束时已送到流水段寄存器IF/ID的输入端。因为之前发生了load-use数据冒险，所以该指令将在随后的第6个时钟周期内重新被读出。

补充题

下面是一段MIPS指令序列：

- 1 add \$t1, \$s2,\$s0
- 2 sub \$t2, \$s0,\$s3
- 3 add \$t1,\$t1,\$t2

假定在一个采用“取指、译码/取数、执行、访存、写回”的5段流水线处理器中执行上述指令序列，请回答下列问题：

- (1)以上指令队列中，哪些指令之间发生数据相关？
- (2)不采用“转发”技术的话，需要在何处、加入几条nop指令才能使这段指令序列的执行避免数据冒险？
- (3)如果采用“转发”技术，是否可以完全解决数据冒险？不行的话，需要在何处、加入几条nop指令才能使这段指令序列的执行避免数据冒险？

补充题 参考答案

下面是一段MIPS指令序列：

```
1 add $t1, $s2,$s0
2 sub $t2, $s0,$s3
3 add $t1,$t1,$t2
```

参考答案：

(1)因为第1条和第2条指令都会更新第3条指令用到的寄存器的值，有可能导致第3条指令取操作数时得到的是更新前的数据，这样第3条指令就不能正确执行。因此第1和第3条指令、第2和第3条指令之间发生数据相关。

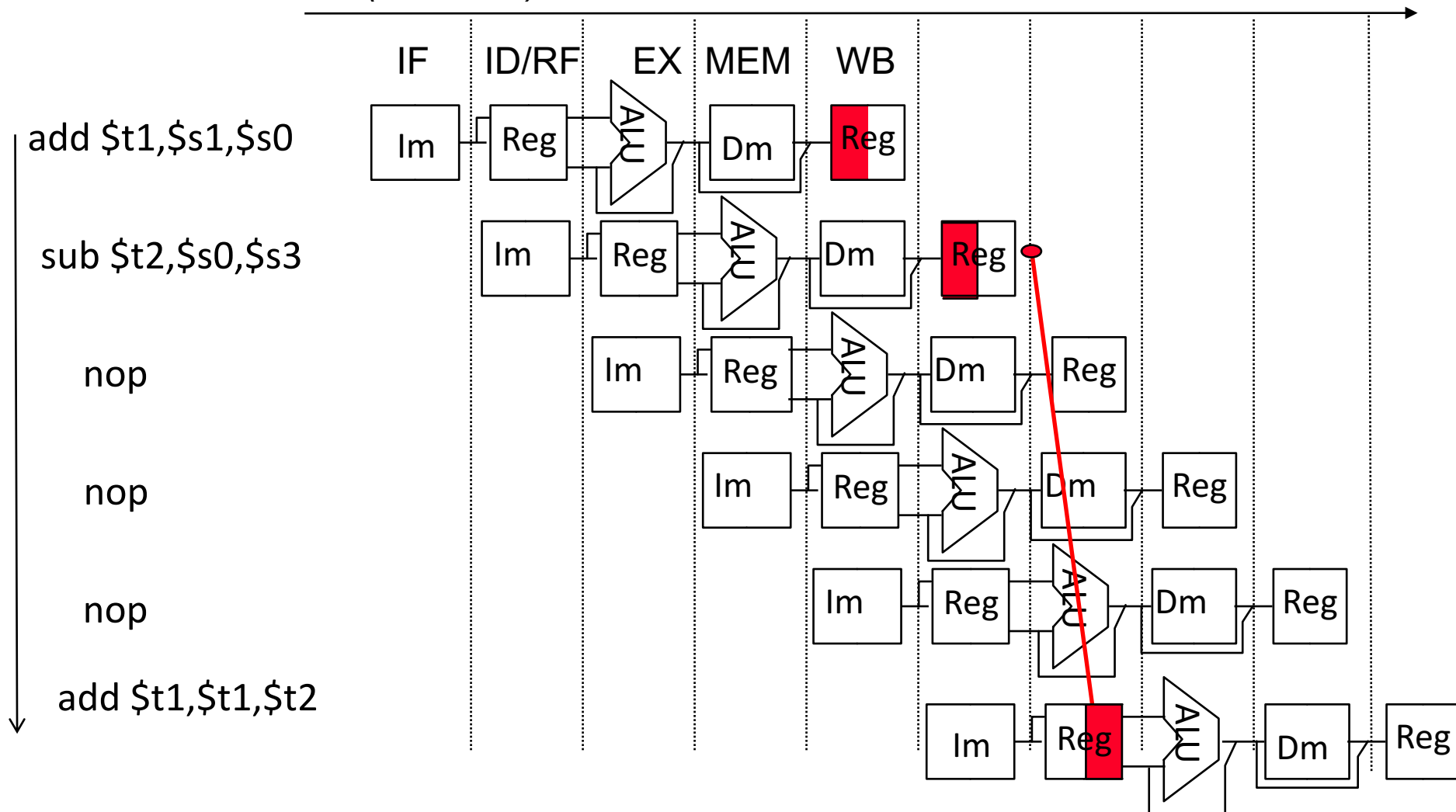
(2)不进行“转发”的话，就只能通过在第3条指令前加nop指令来延迟第3条指令的执行。因为只有第2条指令把数据写回到\$t2，第3条指令才能从\$t2取到正确的值，所以，第2条指令的写回(Wr)流水段后面才应该是第3条指令的译码/取数(ID)流水段，为此，在第2和第3条指令之间必须插入3条nop指令，如图1所示。

若将寄存器写口和寄存器读口分别安排在一个时钟周期的前、后半周期内独立工作，使得前半周期写入寄存器的内容在后半周期能够正确读出，那么，只要加入两条nop即可，如图2所示。

(3)采用“转发”技术，上述程序可以完全避免数据冒险。只要把第1条指令访存(Mem)段结束时在流水段寄存器中的\$t1的值和第2条指令执行(Ex)段结束时在流水段寄存器中的\$t2的值同时“转发”到第3条指令的执行(Ex)段中ALU的两个输入端，这样，在ALU中运算的两个操作数都是正确的值，不会发生数据冒险，无须再插入nop指令，如图3所示。

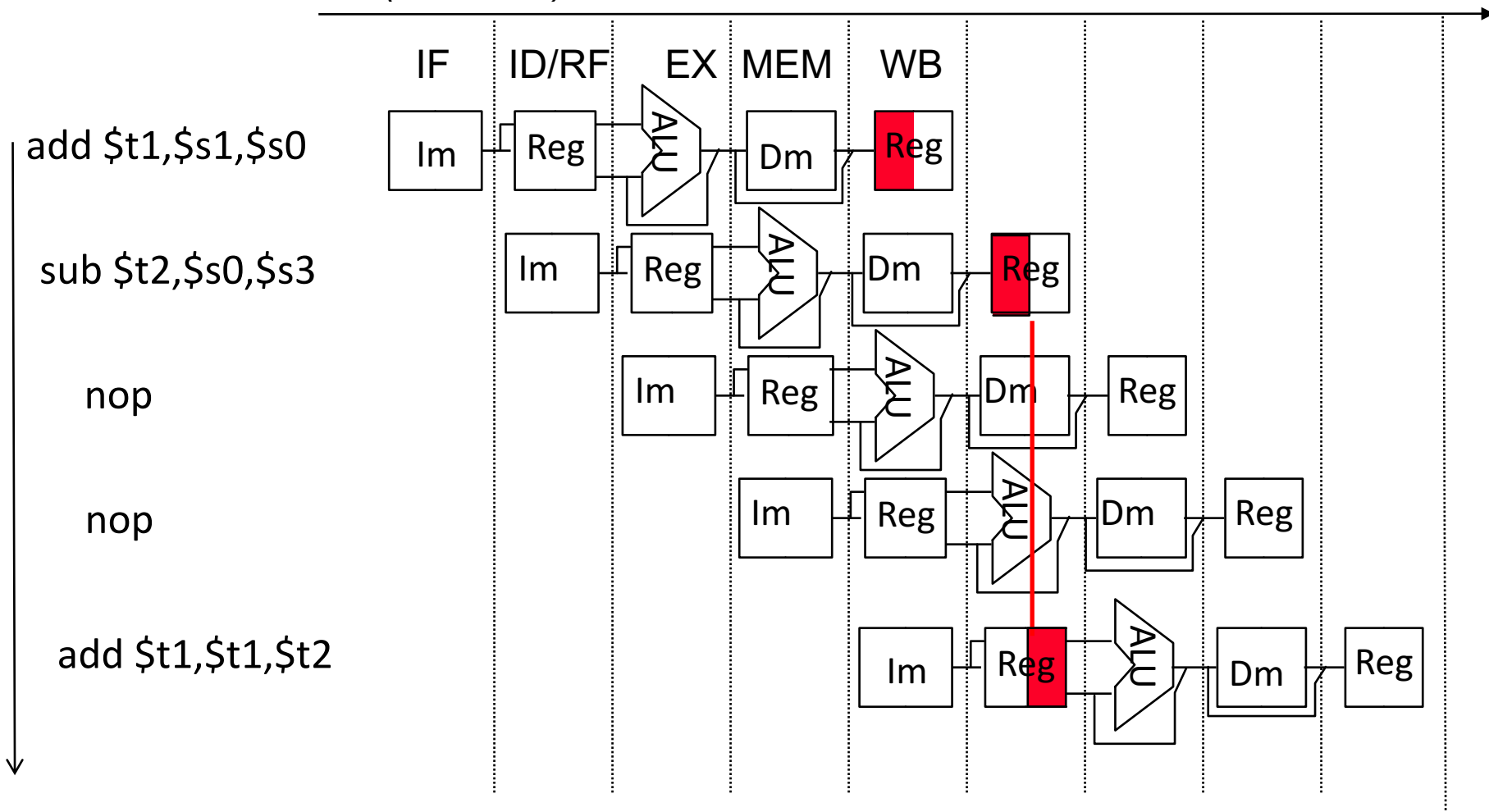
□ 补充题参考答案 图1

时间 (时钟周期)



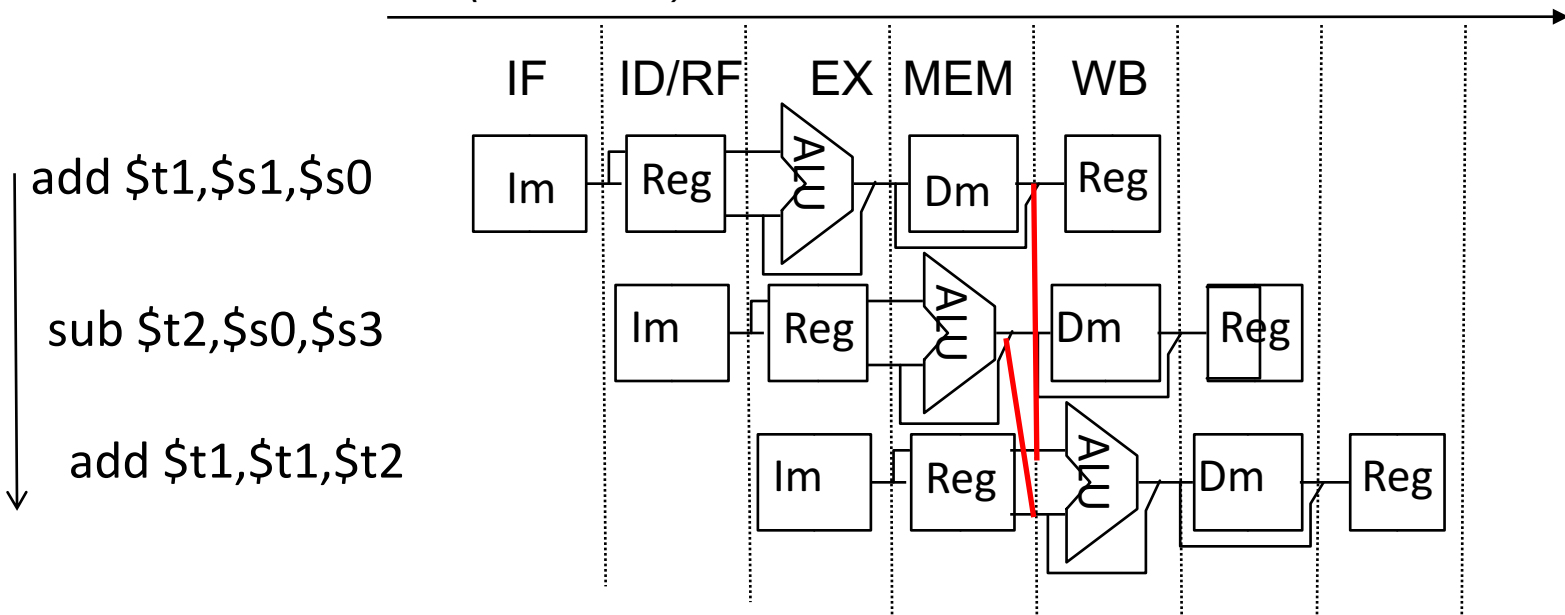
□补充题参考答案 图2

时间 (时钟周期)



□ 补充题参考答案 图3

时间 (时钟周期)



P234 第11题 参考答案

(1)单周期方式下，时钟周期为 $200+50+100+200+50=600\text{ps}$ ，故一条指令的执行时间为 600ps 。

(2)多周期方式下， $\text{CPI}=0.25*7+0.10*6+0.52*5+0.11*4+0.02*4=5.47$ ，存储器操作变为在两个时钟周期内完成后，多周期数据通路的时钟周期为 100ps ，故平均一条指令的执行时间为 $100*5.47=547\text{ps}$ 。

(3)流水线方式下，存储器操作变为在两个时钟周期内完成后，其流水线包含了7个阶段。

对于分支指令，若预测正确，则不需要额外时钟周期，故只需1个时钟周期；若预测错误，则因为分支延迟损失时间片为2，所以应该将错误预取的2条指令冲刷掉，额外多用了2个时钟周期，因此预测错误时共需3个时钟周期，故分支指令的 $\text{CPI}=0.25*3+0.75*1=1.5$ 。

对于load指令，因为一个存储操作占用2个时钟周期，所以随后第1条指令则需3个（其中阻塞2个）时钟周期；随后第二条指令需2个（其中阻塞1个）时钟周期，以后的指令都不需要阻塞，故 $\text{CPI}=1/2*3+1/4*2+2/8*1=2.25$ 。

对于ALU指令，随后的数据相关指令都可以通过转发解决，故 $\text{CPI}=1$ 。

对于store指令，不会发生数据冒险，故 $\text{CPI}=1$ 。

对于jump指令，最快也要在译码阶段才能确定转移地址，因此需阻塞2个时钟周期，加上本身一个时钟周期，共3个时钟周期，故 $\text{CPI}=3$ 。

因此综合 $\text{CPI}=0.25*2.25+0.10*1+0.52*1+0.11*1.5+0.02*3=1.41$ ，所以，平均一条指令的执行时间为 $1.41*100=141\text{ps}$ 。

由上述分析可知，流水线处理器的指令执行速度最快，是单周期的 $600/141 \approx 4.26$ 倍，是多周期的 $547/141 \approx 3.844$ 倍。