

程序员可见的状态都没有影响。

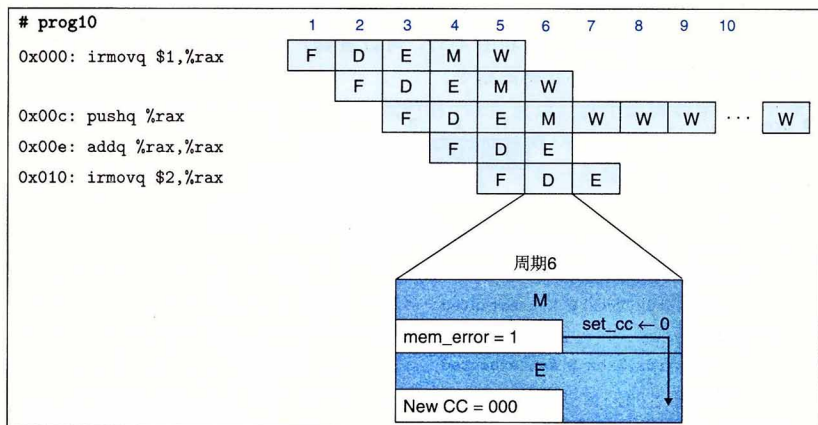


图 4-63 处理非法内存引用异常。在周期 6，pushq 指令的非法内存引用导致禁止更新条件码。流水线开始往访存阶段插入气泡，并在写回阶段暂停异常指令

## 2. 发现特殊控制条件

图 4-64 总结了需要特殊流水线控制的条件。它给出的表达式描述了在哪些条件下会出现这三种特殊情况。一些简单的组合逻辑块实现了这些表达式，为了在时钟上升开始下一个周期时控制流水线寄存器的活动，这些块必须在时钟周期结束之前产生出结果。在一个时钟周期内，流水线寄存器 D、E 和 M 分别保持着处于译码、执行和访存阶段中的指令的状态。在到达时钟周期末尾时，信号  $d\_srcA$  和  $d\_srcB$  会被设置为译码阶段中指令的源操作数的寄存器 ID。当 ret 指令通过流水线时，要想发现它，只要检查译码、执行和访存阶段中指令的指令码。发现加载/使用冒险要检查执行阶段中的指令类型 (mrmovq 或 popq)，并把它的目的寄存器与译码阶段中指令的源寄存器相比较。当跳转指令在执行阶段时，流水线控制逻辑应该能发现预测错误的分支，这样当指令进入访存阶段时，它就能设置从错误预测中恢复所需要的条件。当跳转指令处于执行阶段时，信号  $e\_Cnd$  指明是否要选择分支。通过检查访存和写回阶段中的指令状态值，就能发现异常指令。对于访存阶段，我们使用在这个阶段中计算出来的信号  $m\_stat$ ，而不是使用流水线寄存器的  $M\_stat$ 。这个内部信号包含着可能的数据内存地址错误。

条件	触发条件
处理 ret	$IRET \in \{D\_icode, E\_icode, M\_icode\}$
加载/使用冒险	$E\_icode \in \{IMRMOVL, IPOPL\} \ \& \ E\_dstM \in \{d\_srcA, d\_srcB\}$
预测错误的分支	$E\_icode = IJXX \ \& \ ! e\_Cnd$
异常	$m\_stat \in \{SADR, SINS, SHLT\} \    \ W\_stat \in \{SADR, SINS, SHLT\}$

图 4-64 流水线控制逻辑的检查条件。四种不同的条件要求改变流水线，暂停流水线或者取消已经部分执行的指令

## 3. 流水线控制机制

图 4-65 是一些低级机制，它们使得流水线控制逻辑能将指令阻塞在流水线寄存器中，