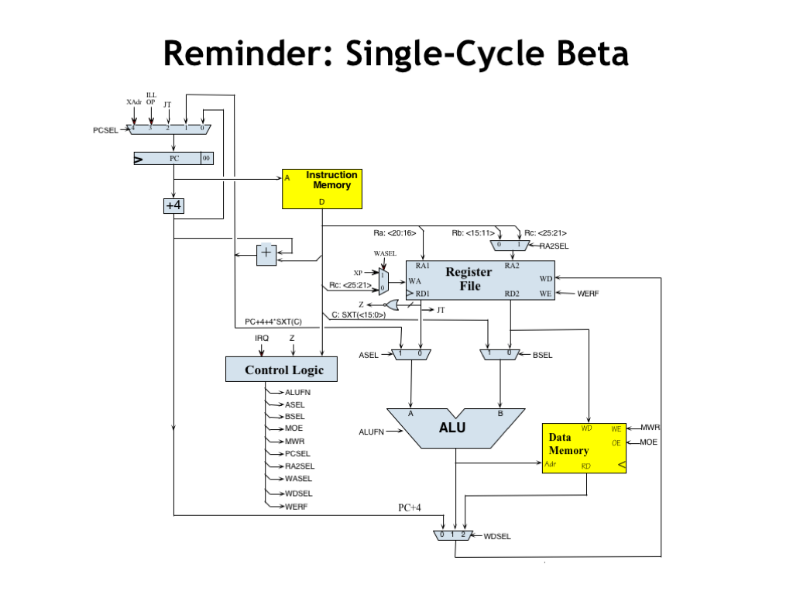
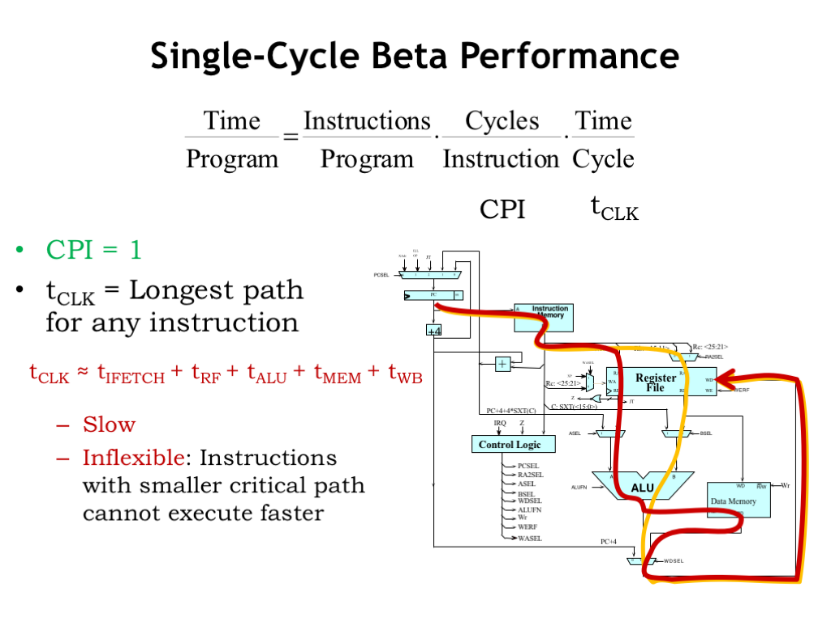
# 课件

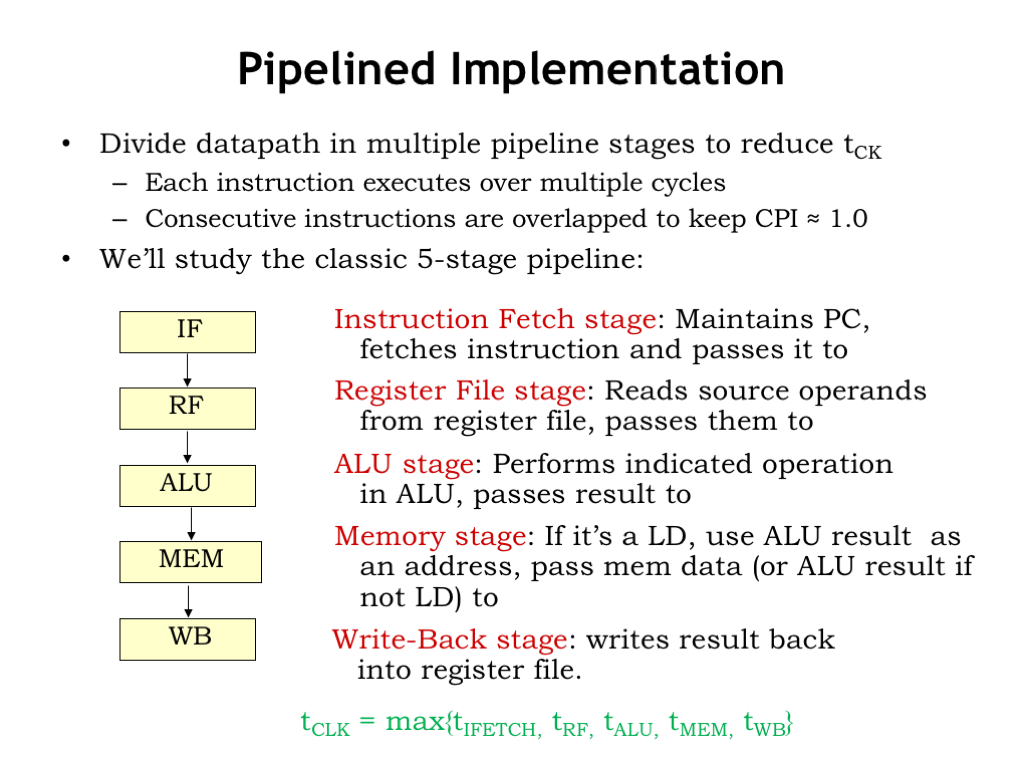
## 回顾：单周期Beta



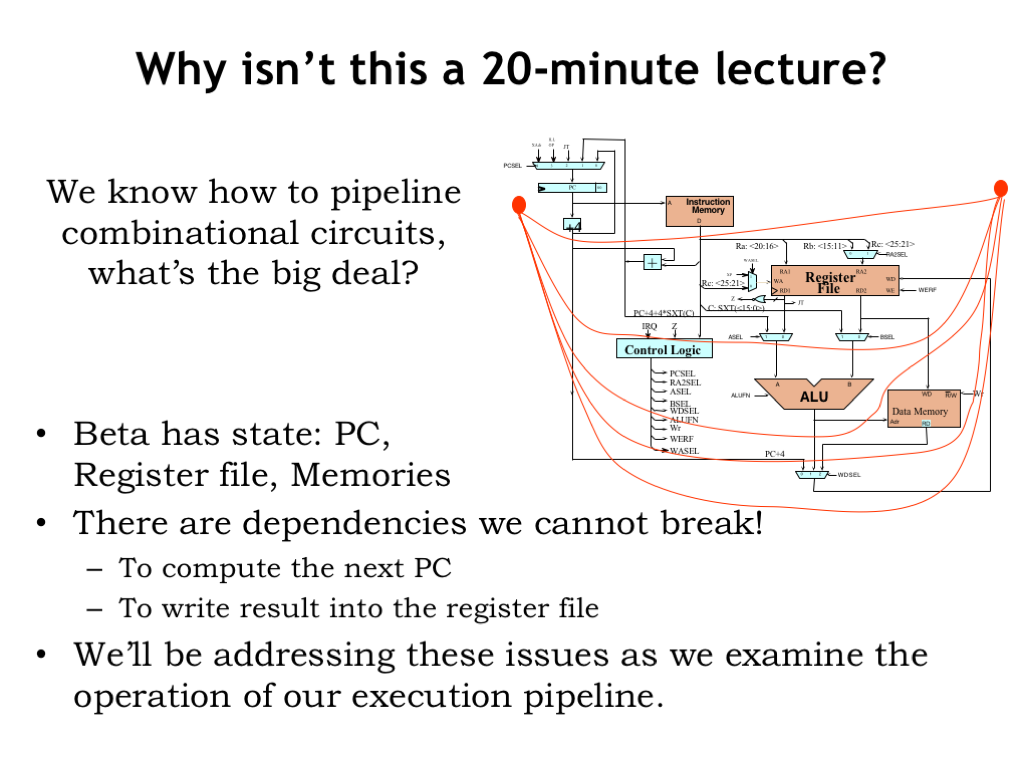
## 单周期Beta性能



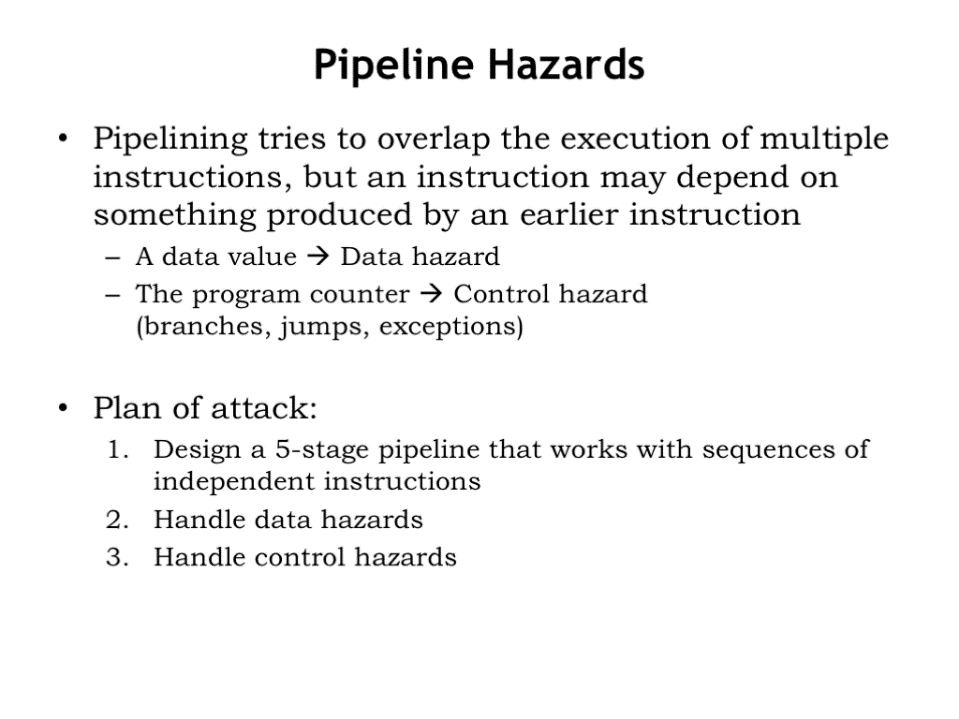
## 流水线实现



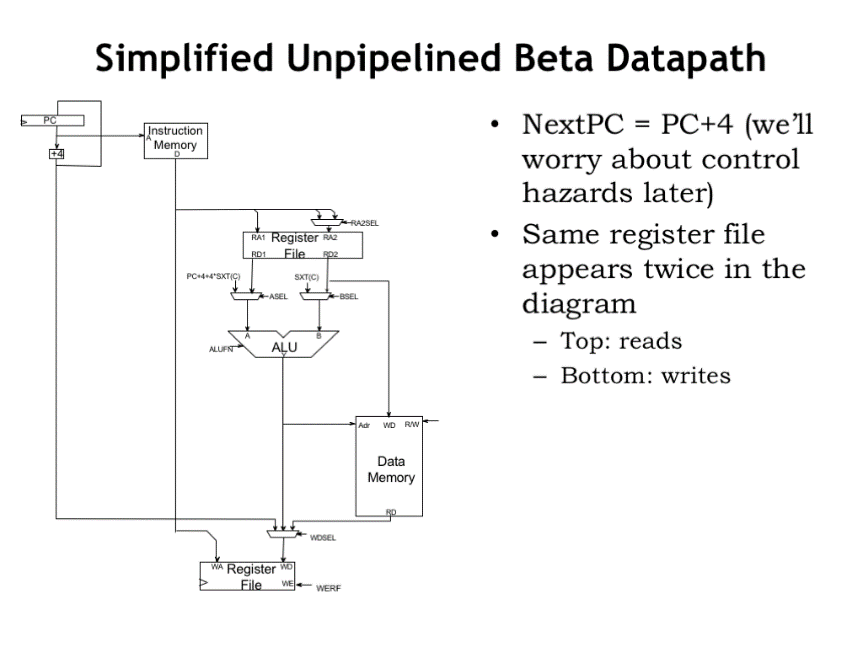
## 为什么这不是一节20分钟的课



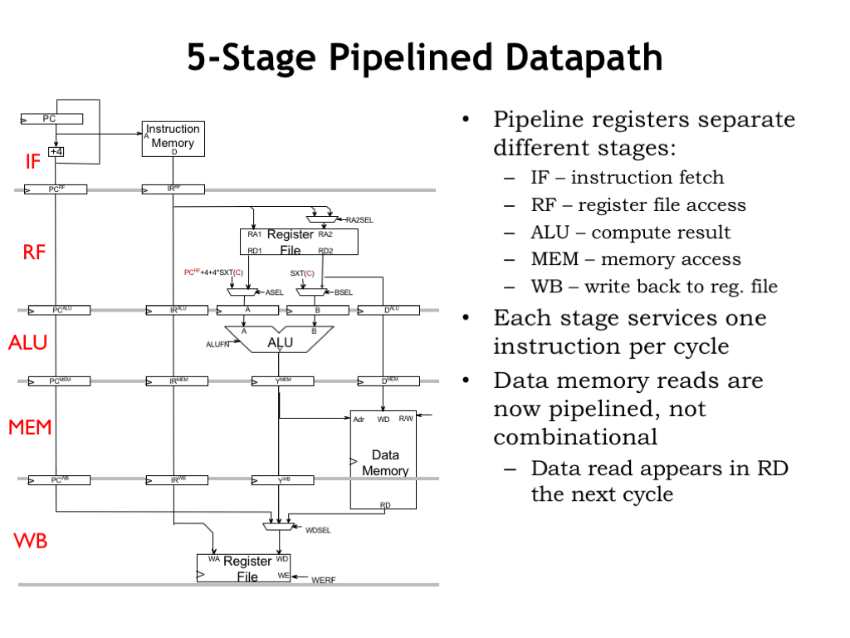
## 流水线的危害



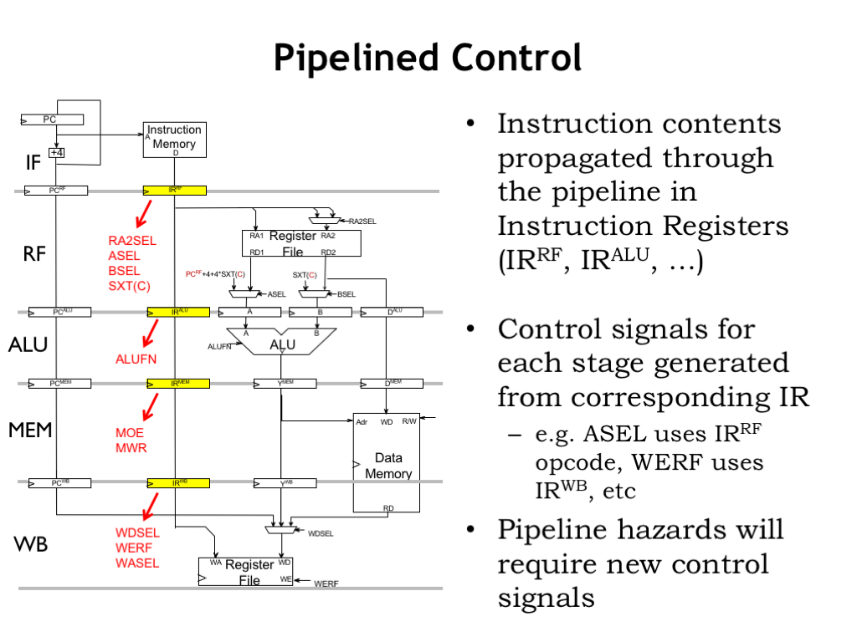
## 简化未流水线化的Beta数据通道



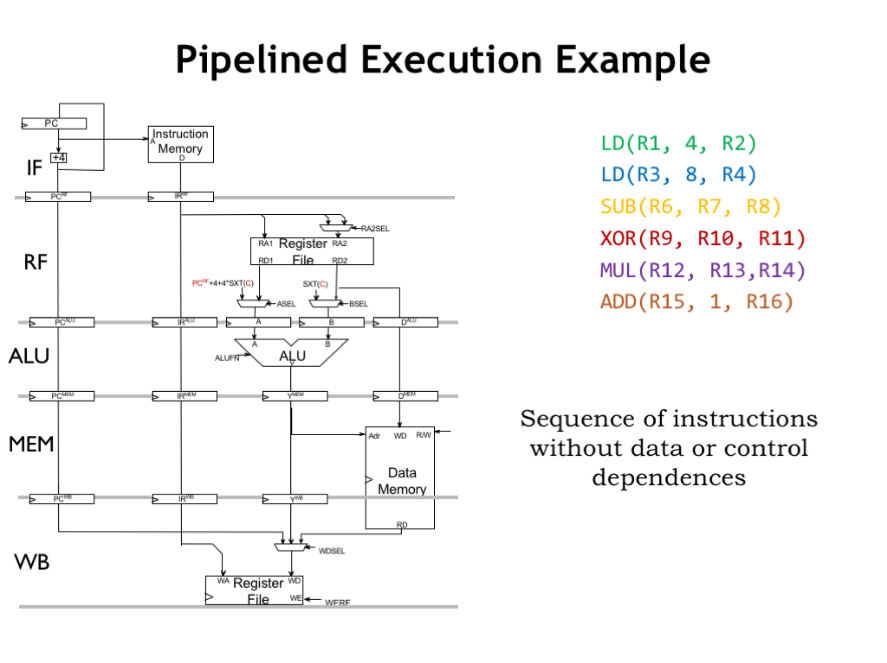
## 5阶段流水线数据通道



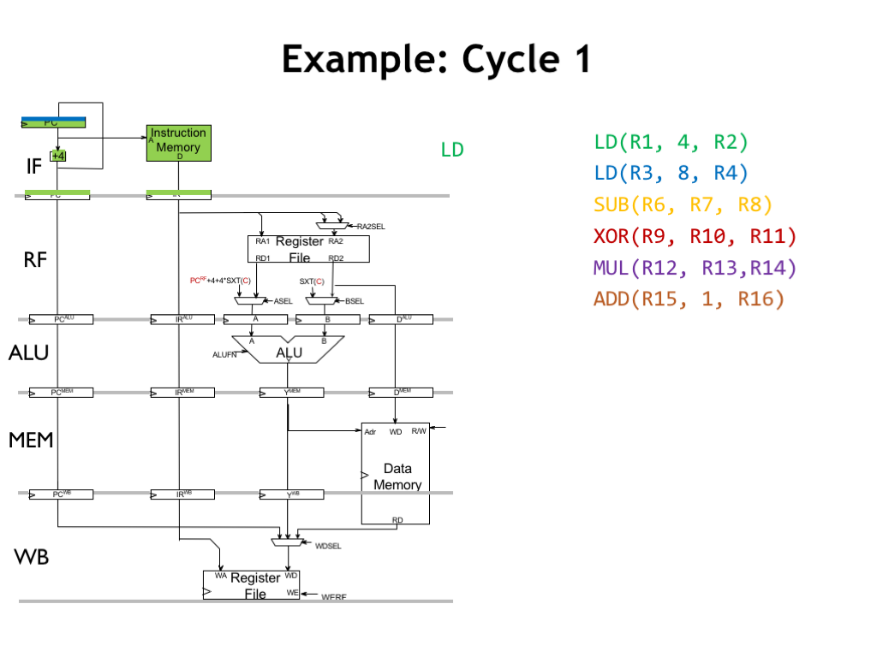
## 流水线控制



## 流水线执行例子



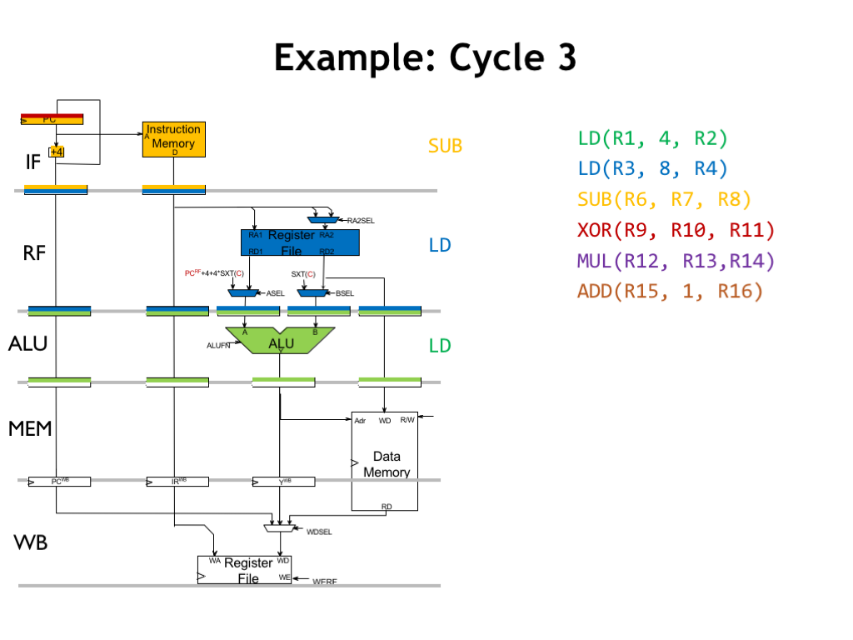
## 例子：周期1



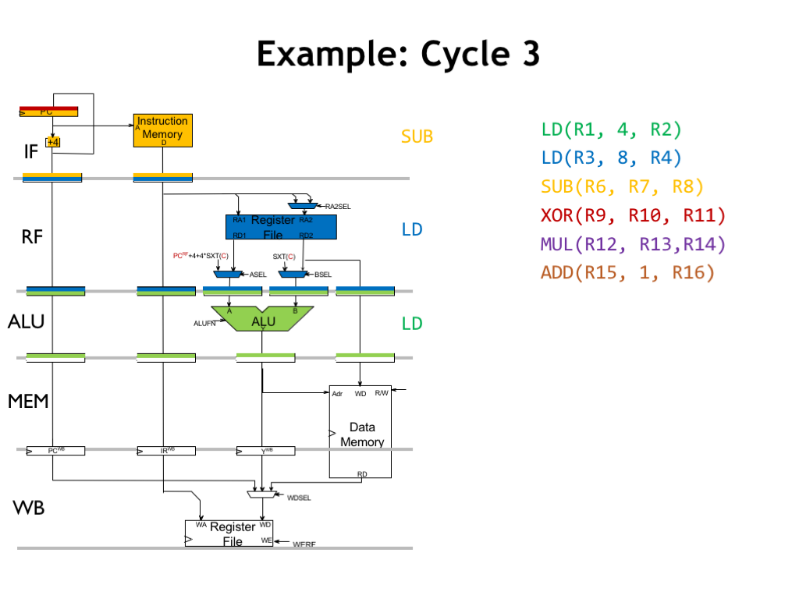
## 例子：周期2



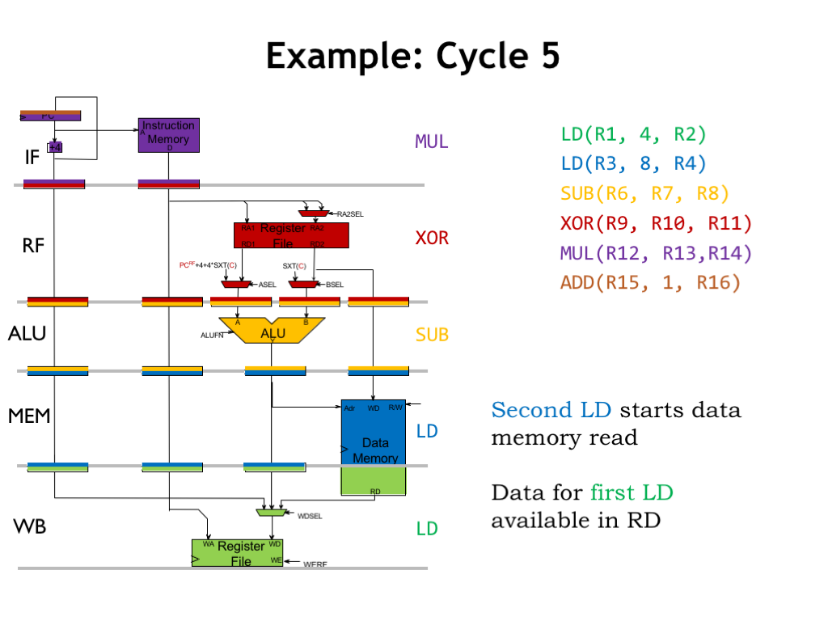
## 例子：周期3



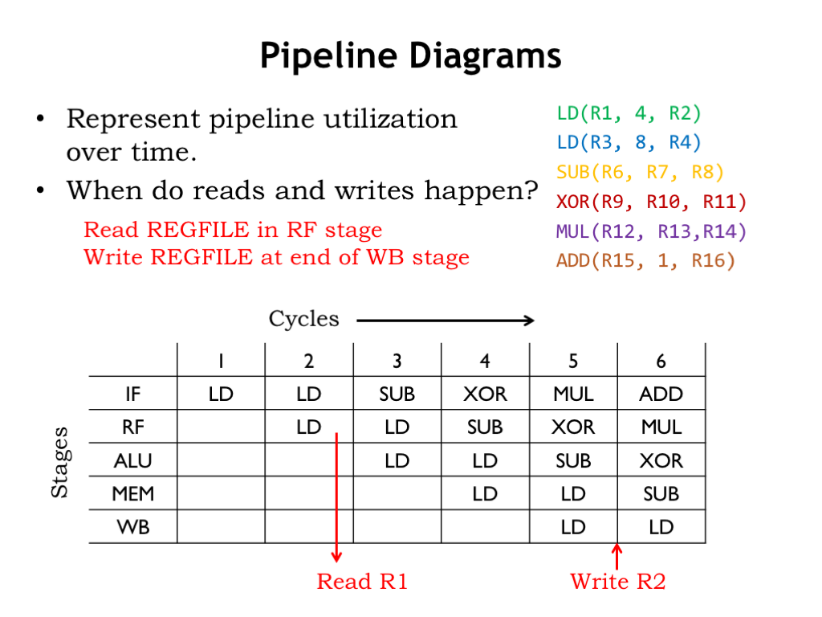
## 例子：周期4



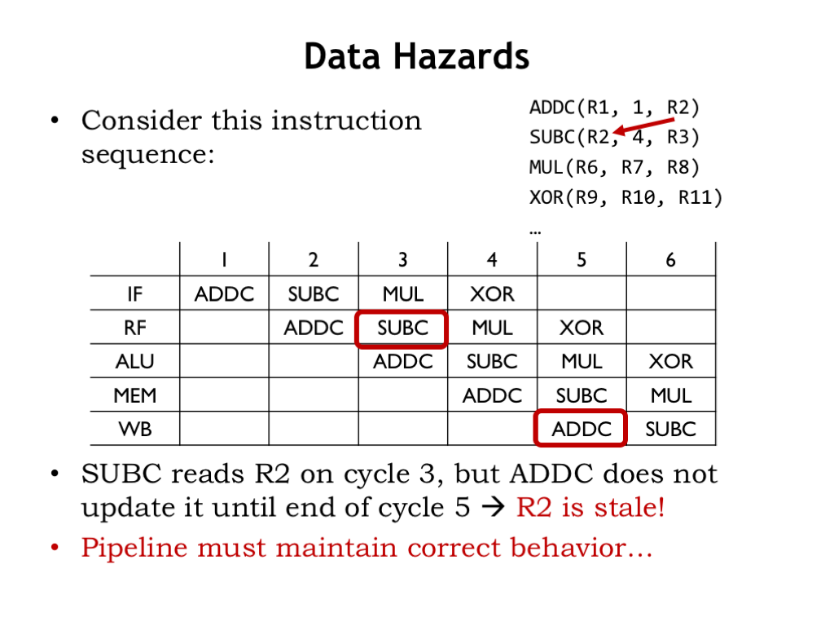
## 例子：周期5



## 流水线图表



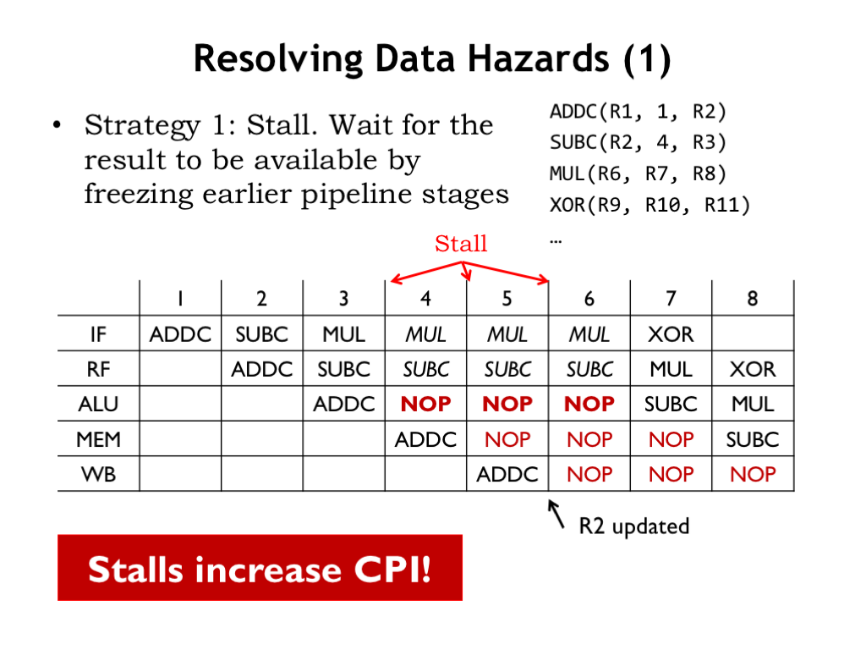
## 数据危害



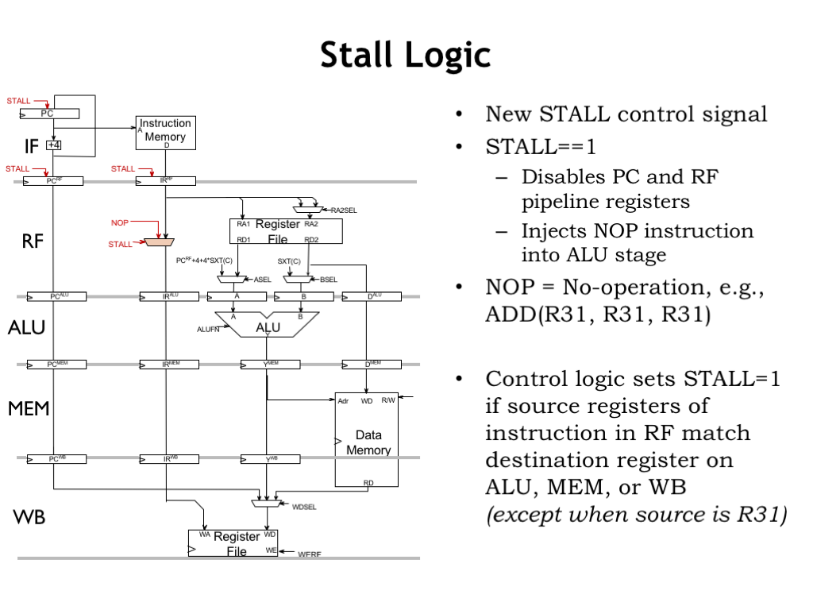
## 解决危害



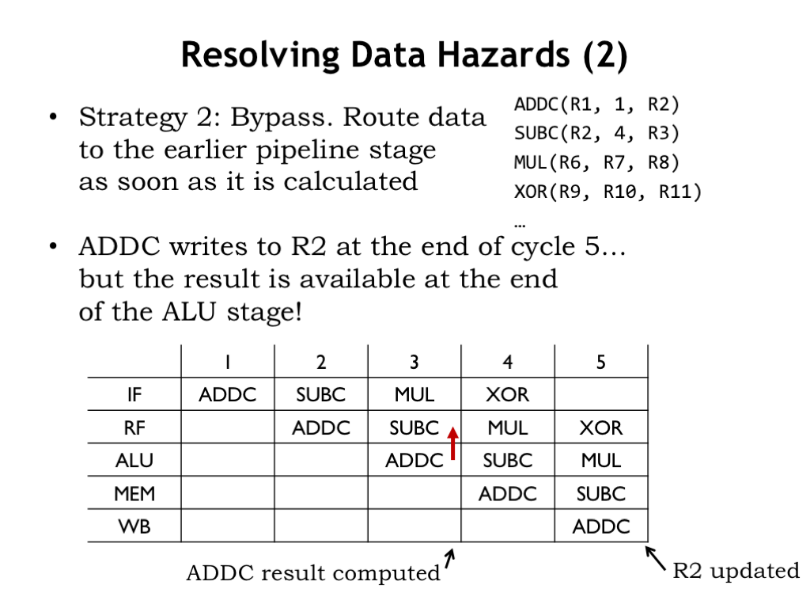
## 解决数据危害1



## stall逻辑



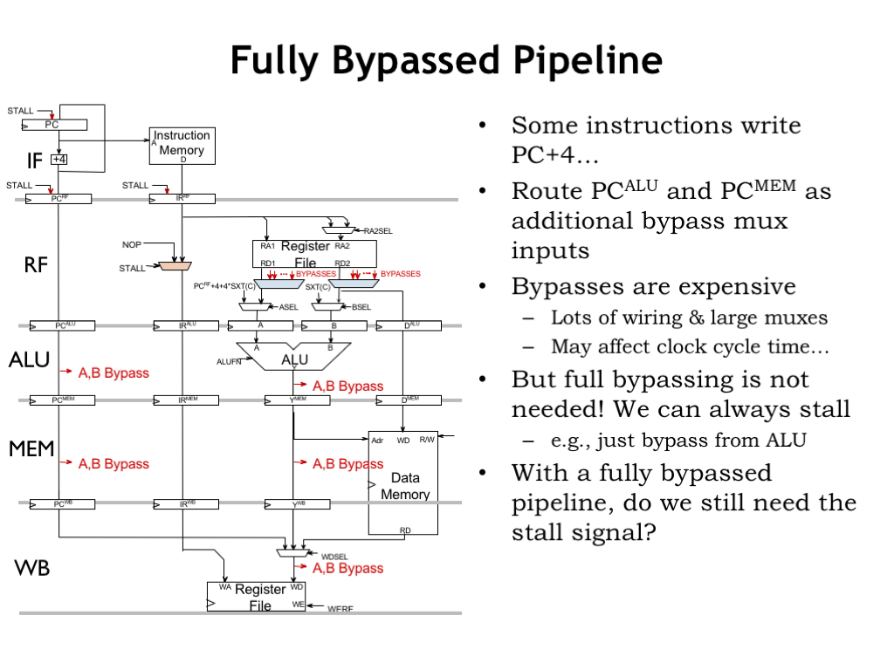
## 解决数据危害2



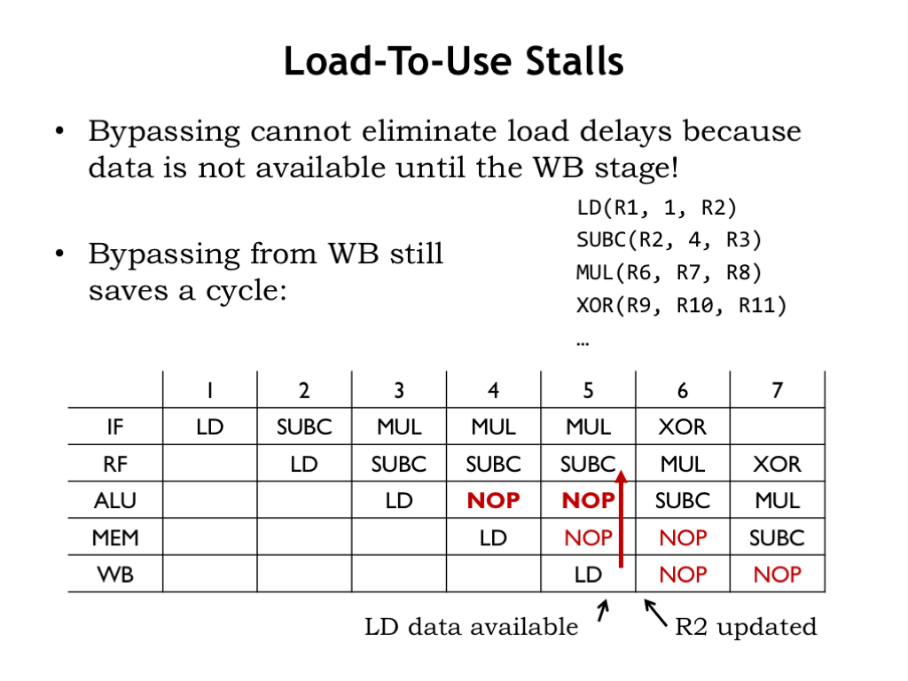
## byPass逻辑



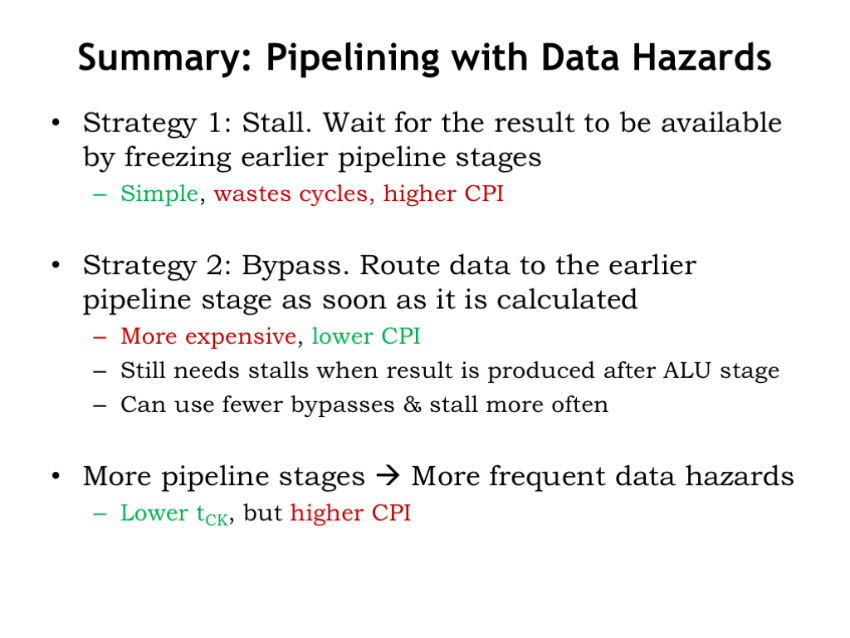
## 完全byPass流水线



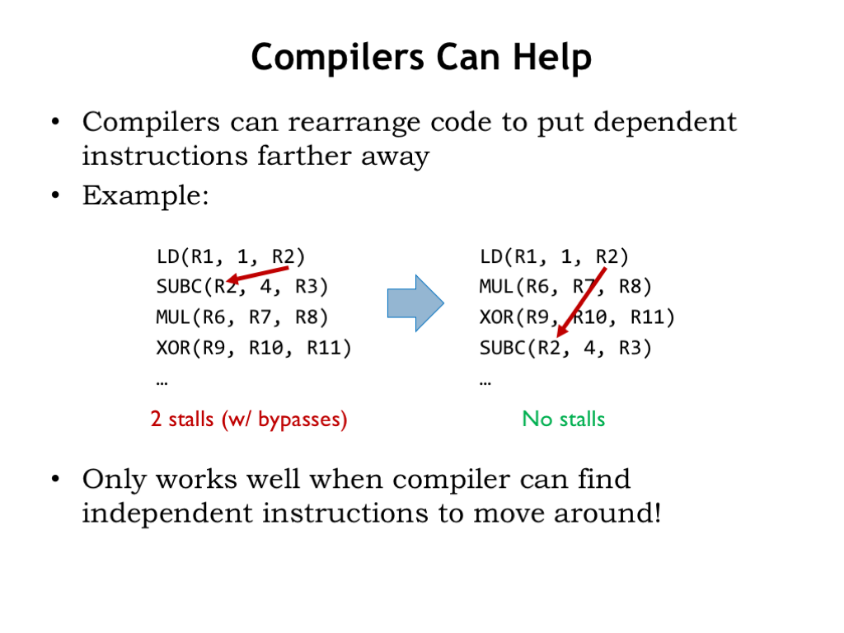
## 装载使用stall



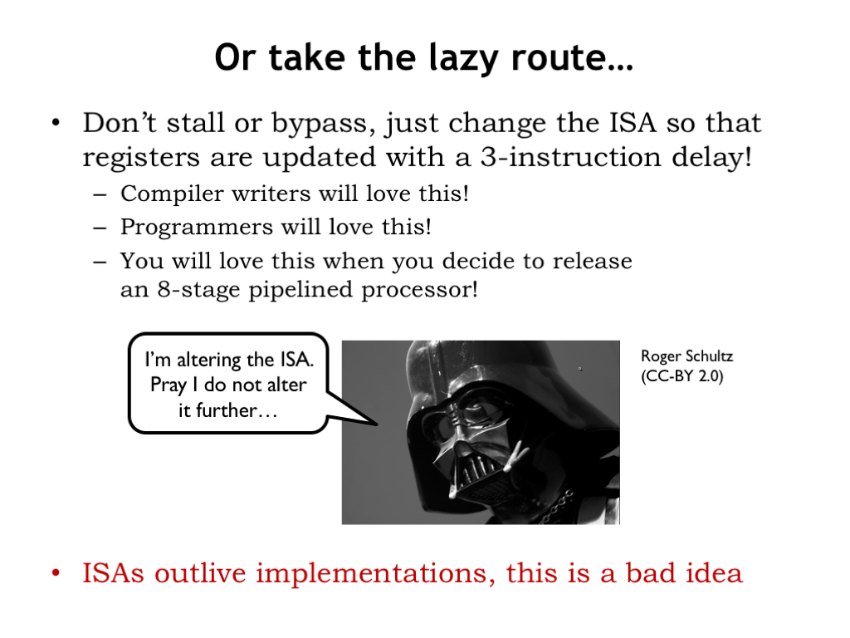
## 总结：流水线带来的数据危害



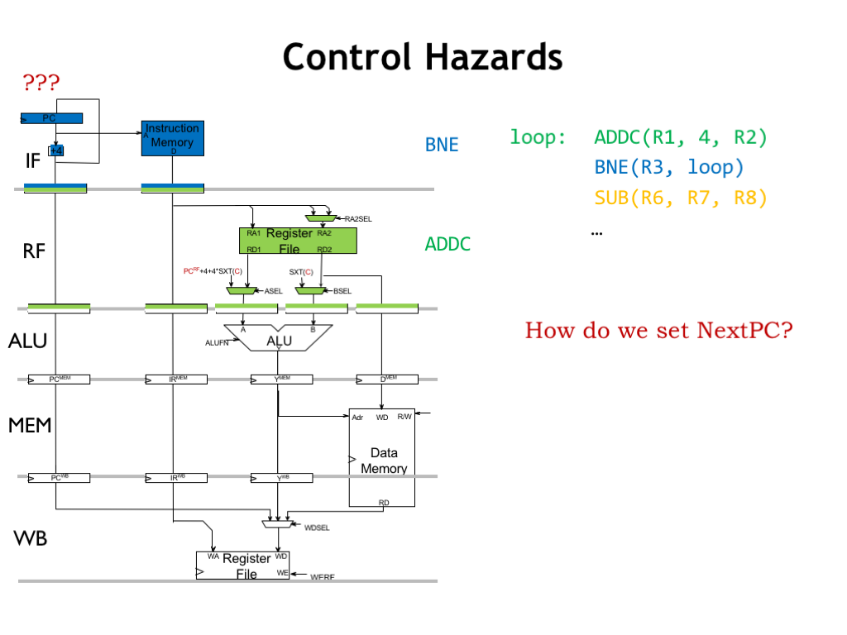
## 编译器可以帮忙



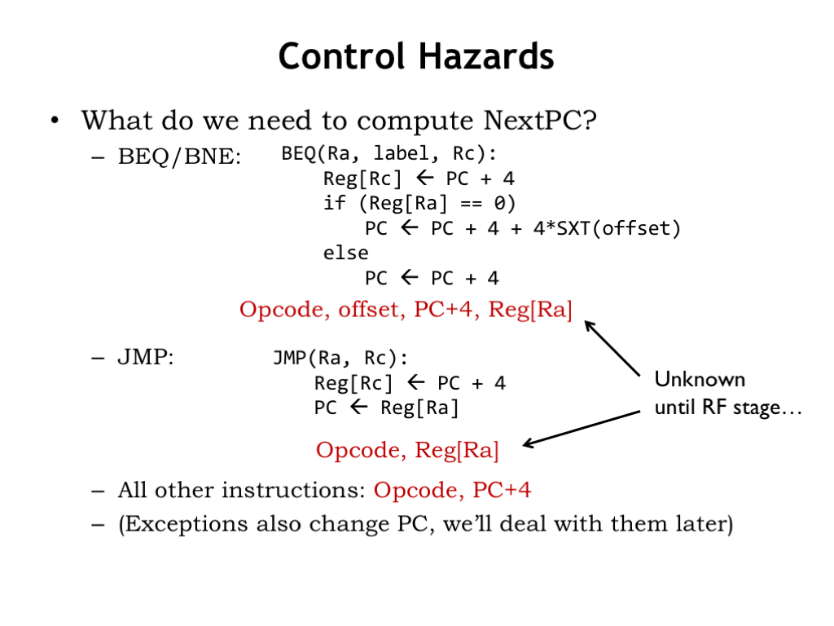
## 或者采取懒散路线



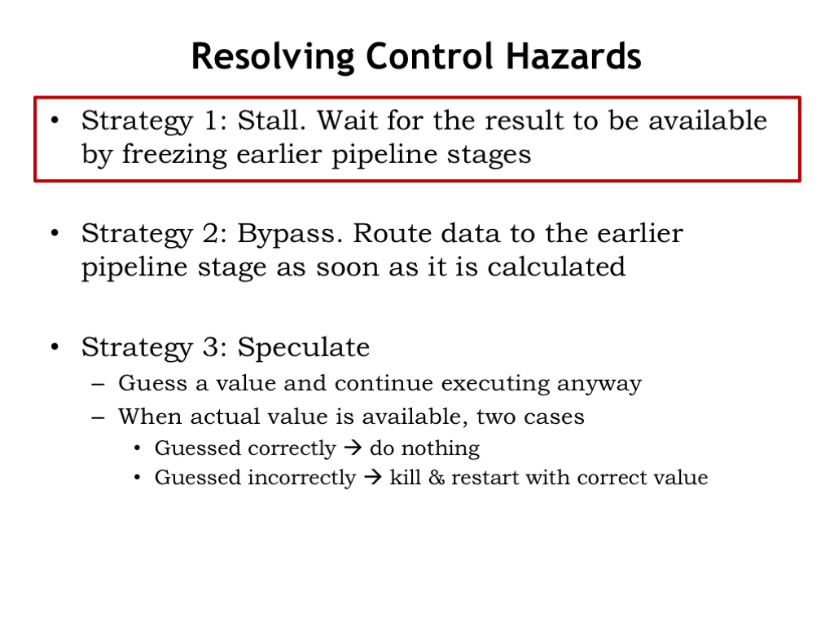
## 控制危害1



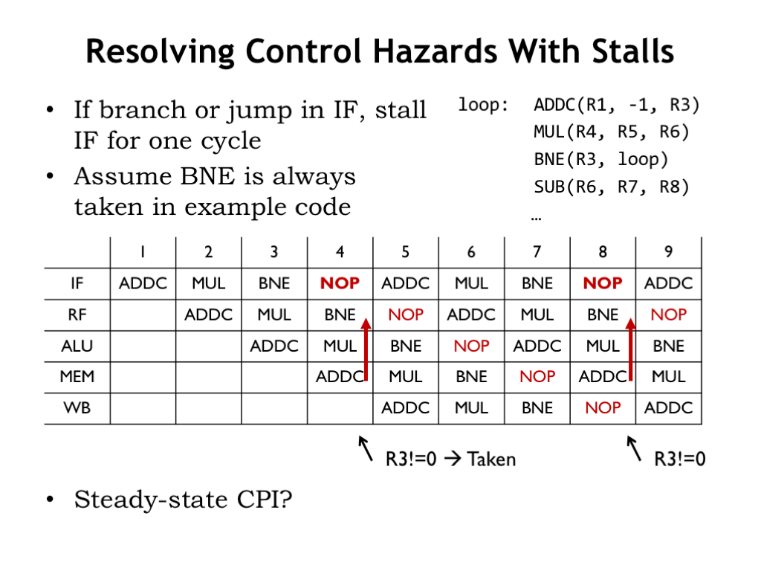
## 控制危害2



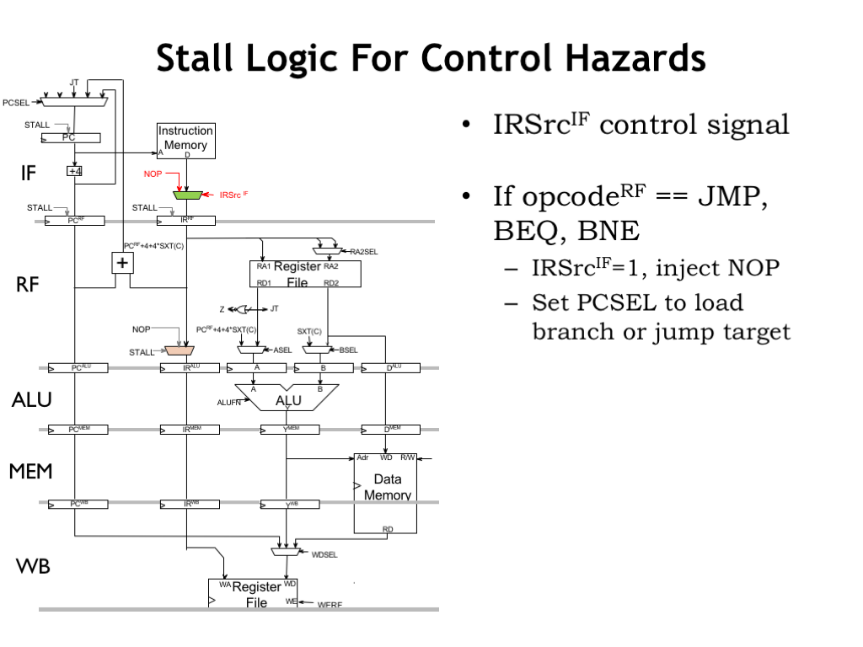
## 解决控制危害



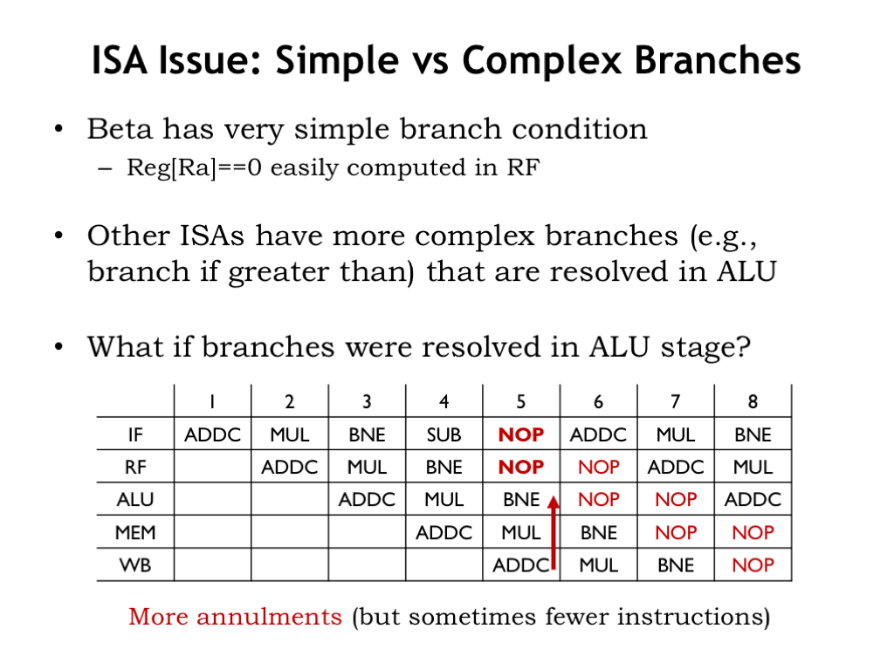
## 用stall解决控制危害



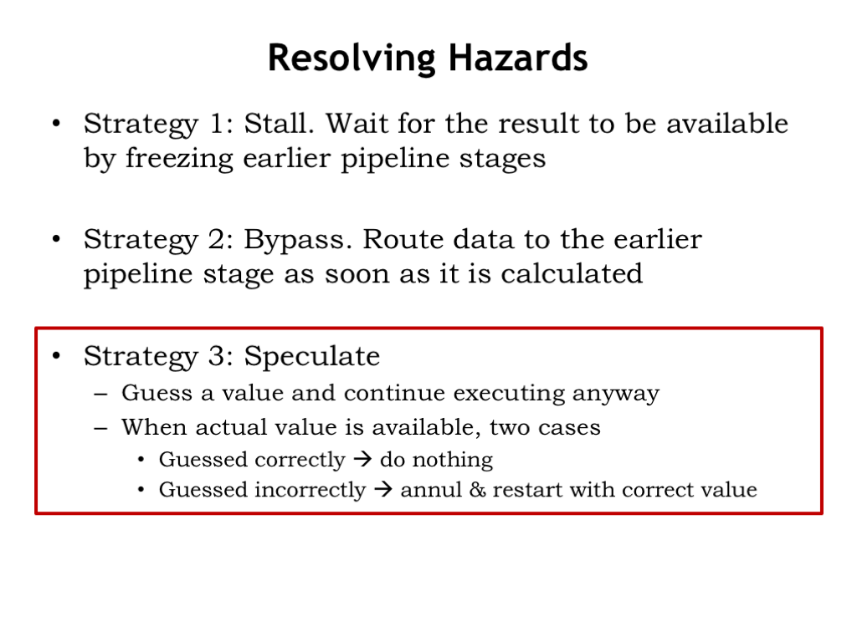
## 用于控制危害的stall逻辑



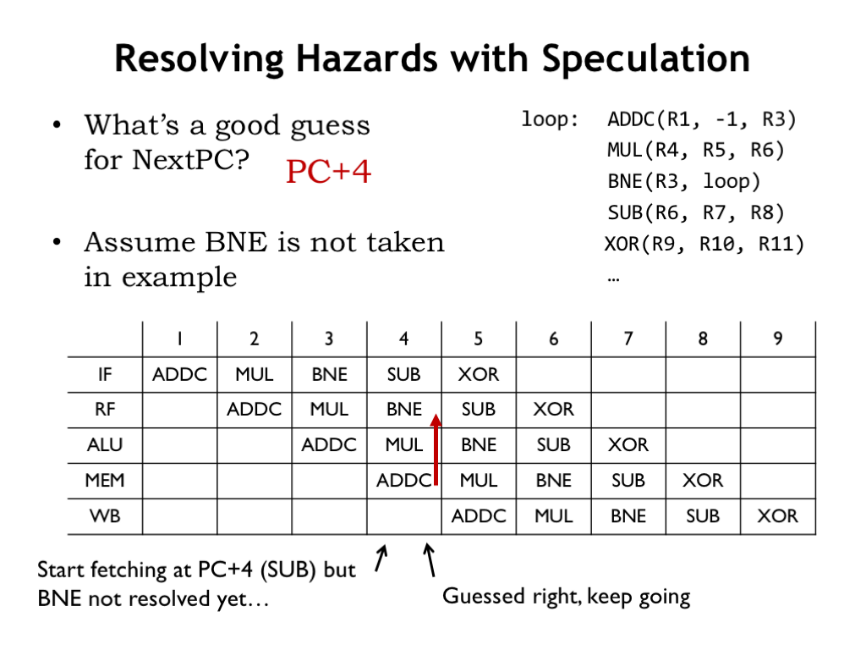
## 指令集问题：简单vs复杂分支



## 解决危害



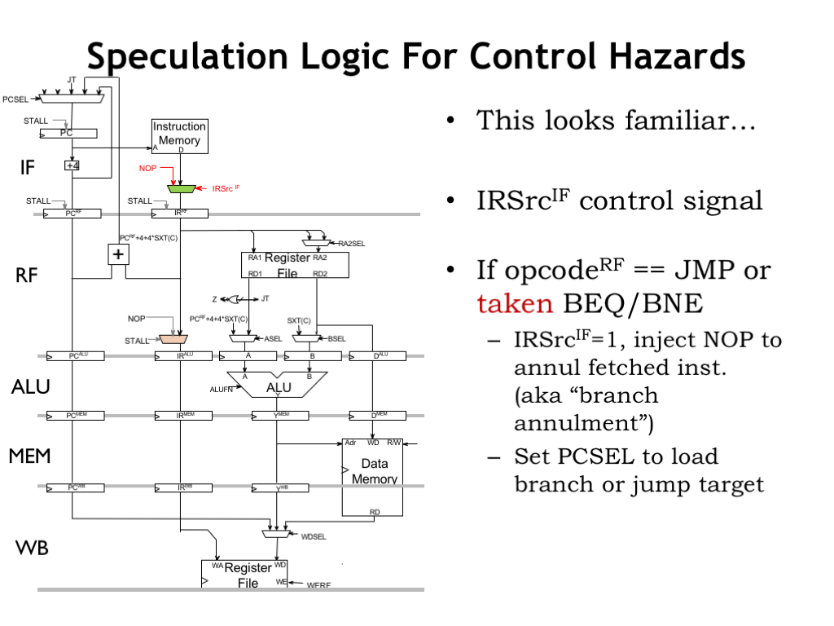
## 用推断（预测）解决危害1



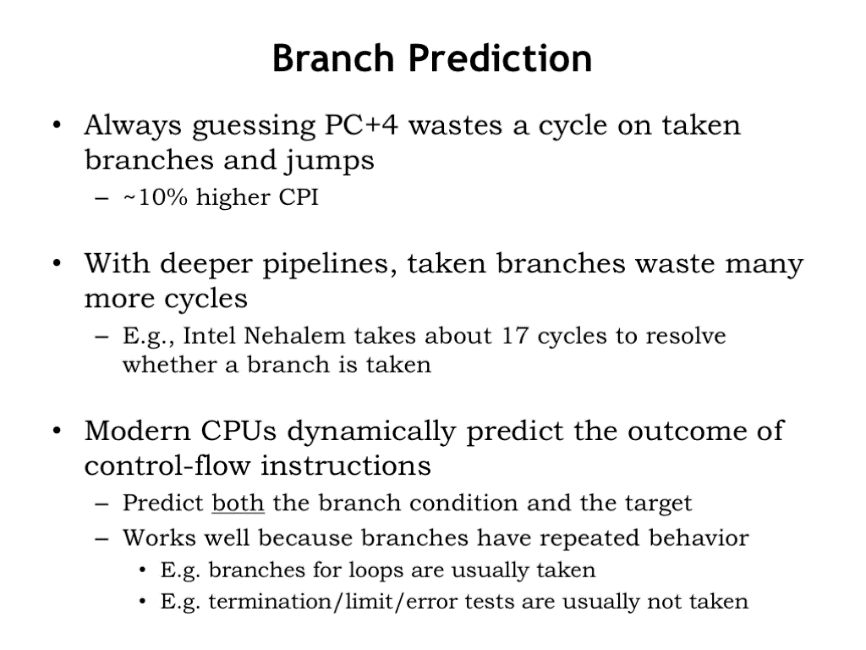
## 用推断（预测）解决危害2



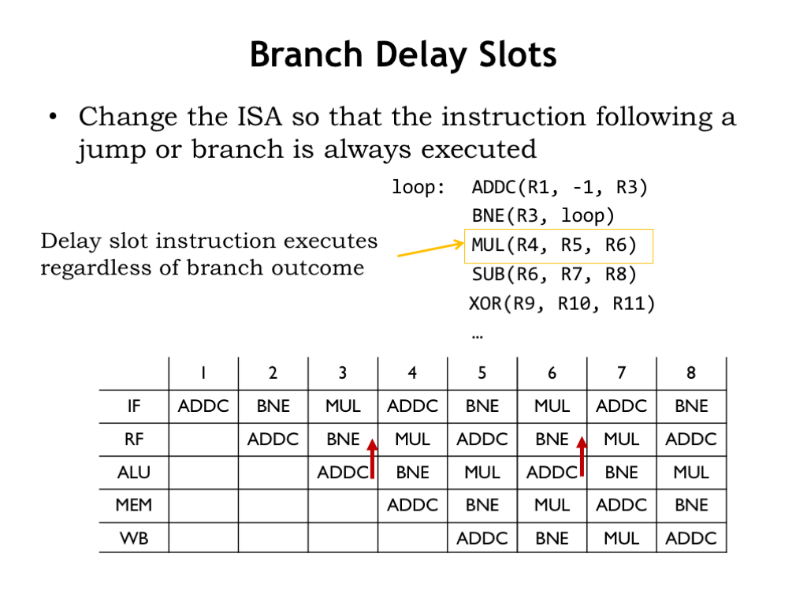
## 对于控制危害的推断逻辑



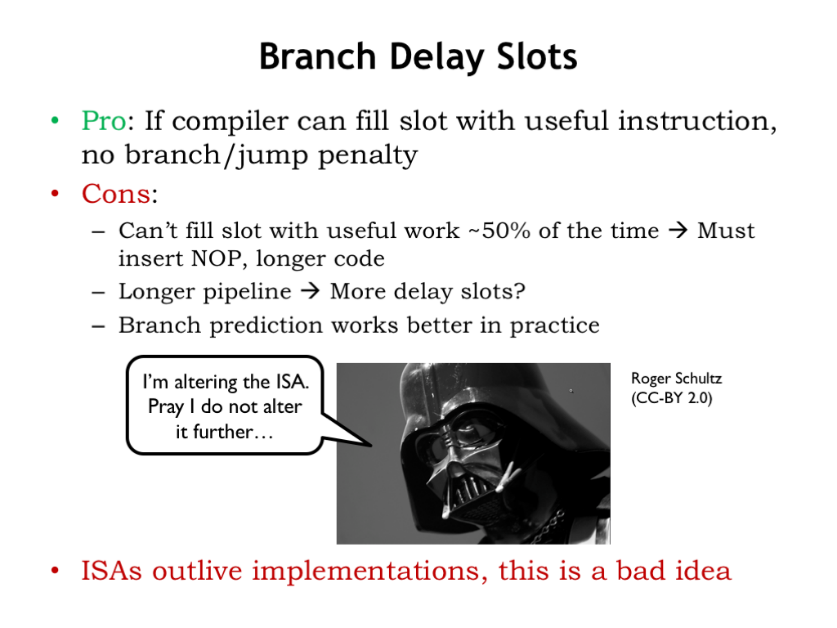
## 分支预测



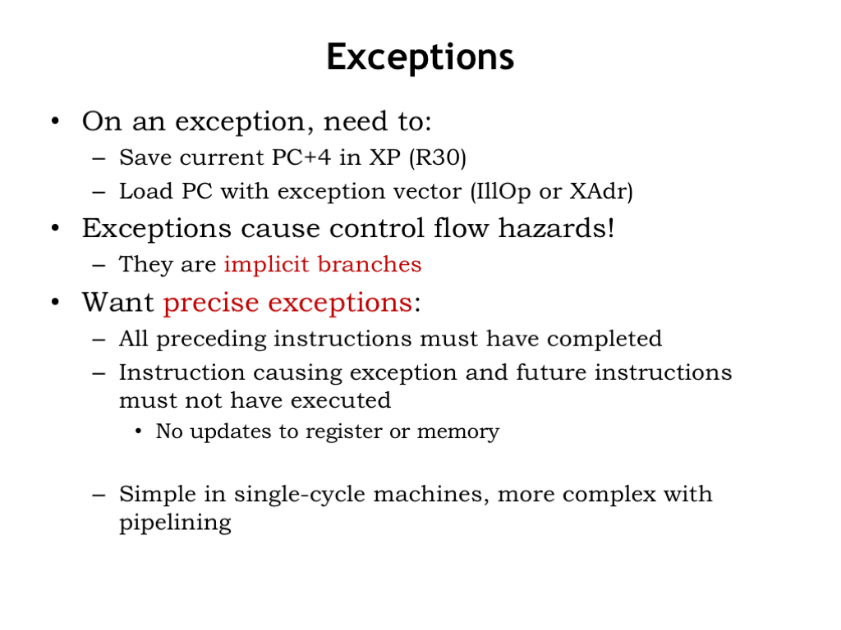
## 分支延迟单元1



## 分支延迟单元2



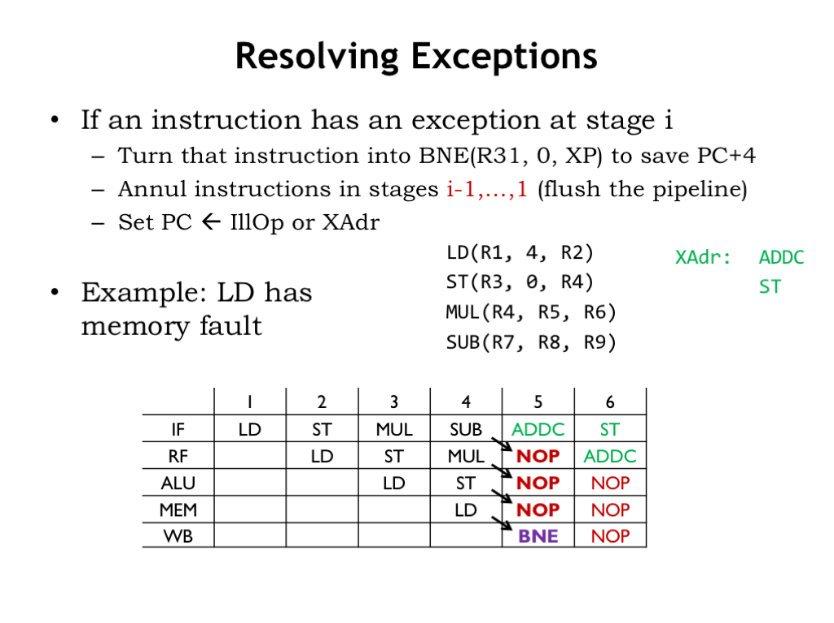
## 异常



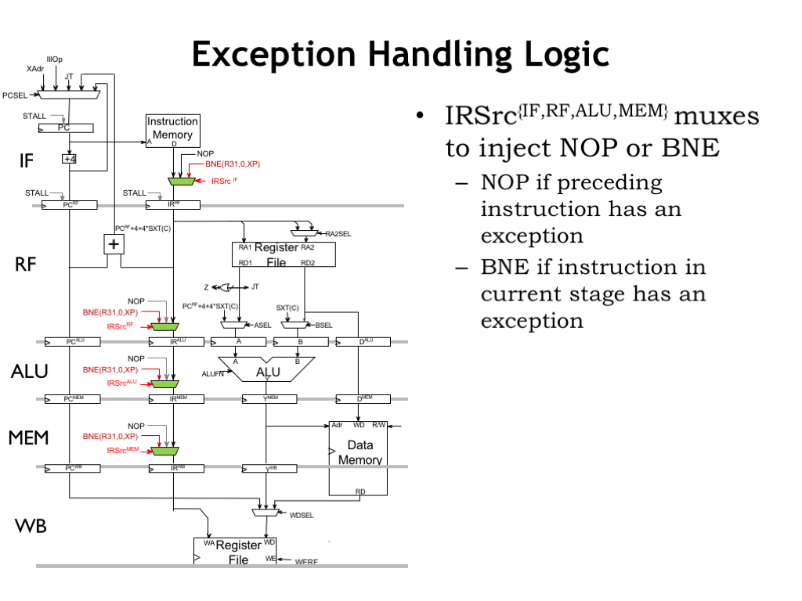
## 何时异常会发生



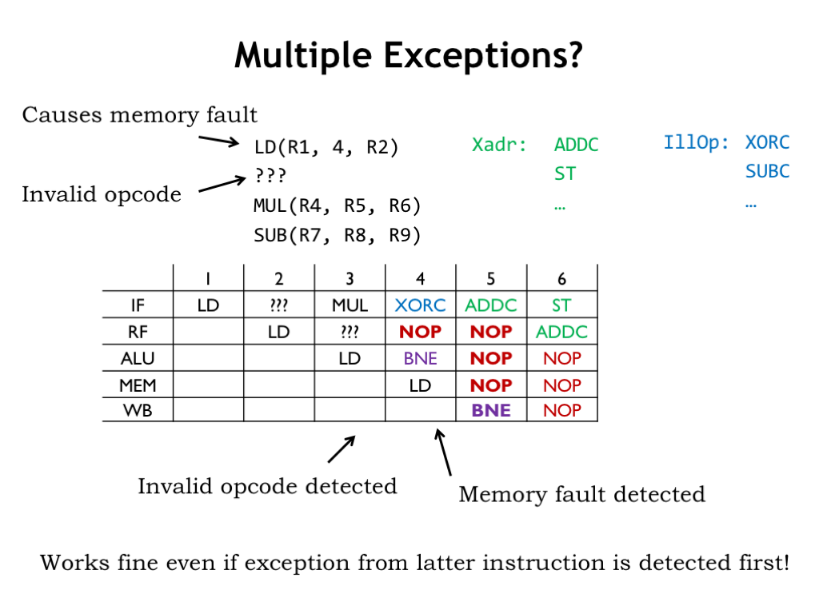
## 解决异常



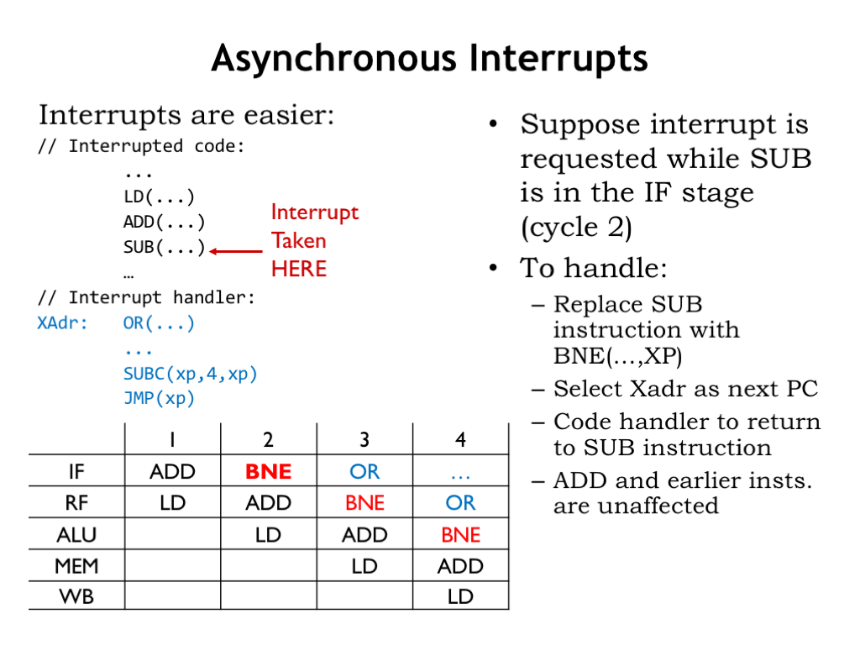
## 异常处理逻辑



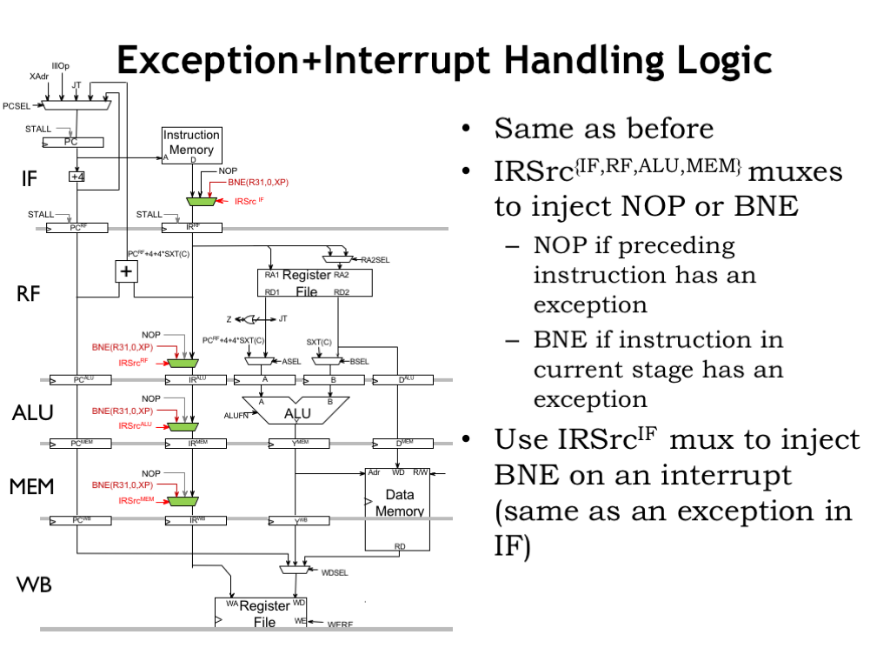
## 多个异常



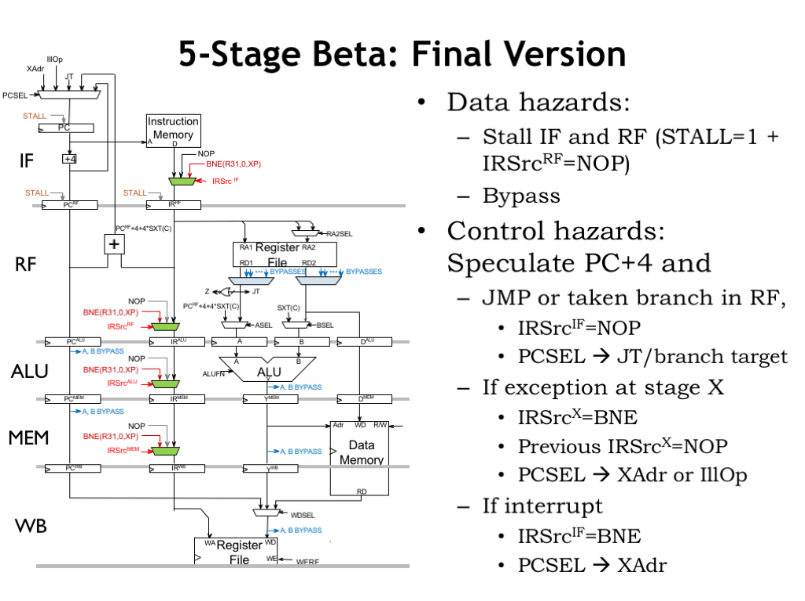
## 异步中断



## 异常、中断处理逻辑



## 5阶段 Beta：最终版



## 回顾：解决危害

