



图 5-11 一个乱序处理器的框图。指令控制单元负责从内存中读出指令，并产生一系列基本操作。然后执行单元完成这些操作，以及指出分支预测是否正确

ICU 从指令高速缓存(instruction cache)中读取指令，指令高速缓存是一个特殊的高速存储器，它包含最近访问的指令。通常，ICU 会在当前正在执行的指令很早之前取指，这样它才有足够的时间对指令译码，并把操作发送到 EU。不过，一个问题是当程序遇到分支^①时，程序有两个可能的前进方向。一种可能会选择分支，控制被传递到分支目标。另一种可能是，不选择分支，控制被传递到指令序列的下一条指令。现代处理器采用了一种称为分支预测(branch prediction)的技术，处理器会猜测是否会选择分支，同时还预测分支的目标地址。使用投机执行(speculative execution)的技术，处理器会开始取出位于它预测的分支会跳到的地方的指令，并对指令译码，甚至在它确定分支预测是否正确之前就开始执行这些操作。如果过后确定分支预测错误，会将状态重新设置到分支点的状态，并开始取出和执行另一个方向上的指令。标记为取指控制的块包括分支预测，以完成确定取哪些指令的任务。

指令译码逻辑接收实际的程序指令，并将它们转换成一组基本操作(有时称为微操作)。每个这样的操作都完成某个简单的计算任务，例如两个数相加，从内存中读数据，或是向内存写数据。对于具有复杂指令的机器，比如 x86 处理器，一条指令可以被译码成多个操作。关于指令如何被译码成操作序列的细节，不同的机器都会不同，这个信息可谓是高度机密。幸运的是，不需要知道某台机器实现的底层细节，我们也能优化自己的程序。

① 术语“分支”专指条件转移指令。对处理器来说，其他可能将控制传递到多个目的地址的指令，例如过程返回和间接跳转，带来的也是类似的挑战。