

jne 指令(第 3 行)产生了一个控制相关(control dependency), 因为条件测试的结果会决定要执行的新指令是 irmovq 指令(第 4 行)还是 halt 指令(第 7 行)。在我们的 SEQ 设计中, 这些相关都是由反馈路径来解决的, 如图 4-22 的右边所示。这些反馈将更新了的寄存器值向下传送到寄存器文件, 将新的 PC 值向下传送到 PC 寄存器。

图 4-38 举例说明了将流水线引入含有反馈路径的系统中的危险。在原来的系统(图 4-38a)中, 每条指令的结果都反馈给下一条指令。流水线图(图 4-38b)就说明了这个情况, I1 的结果成为 I2 的输入, 依此类推。如果试图以最直接的方式将它转换成一个三阶段流水线(图 4-38c), 我们将改变系统的行为。如图 4-38c 所示, I1 的结果成为 I4 的输入。为了通过流水线技术加速系统, 我们改变了系统的行为。

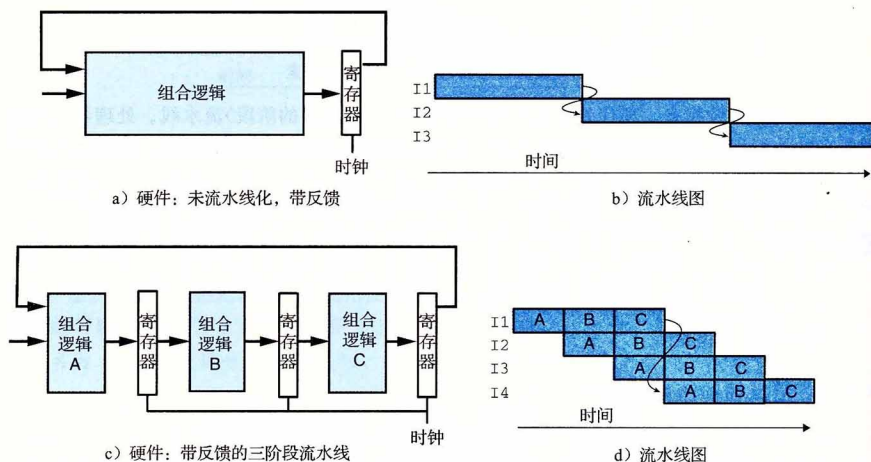


图 4-38 由逻辑相关造成的流水线技术的局限性。在从未流水线化的带反馈的系统 a 转化到流水线化的系统 c 的过程中, 我们改变了它的计算行为, 可以从两个流水线图(b 和 d)中看出来

当我们将流水线技术引入 Y86-64 处理器时, 必须正确处理反馈的影响。很明显, 像图 4-38 中的例子那样改变系统的行为是不可接受的。我们必须以某种方式来处理指令间的数据和控制相关, 以使得到的行为与 ISA 定义的模型相符。

4.5 Y86-64 的流水线实现

我们终于准备好要开始本章的主要任务——设计一个流水线化的 Y86-64 处理器。首先, 对顺序的 SEQ 处理器做一点小的改动, 将 PC 的计算挪到取指阶段。然后, 在各个阶段之间加上流水线寄存器。到这个时候, 我们的尝试还不能正确处理各种数据和控制相关。不过, 做一些修改, 就能实现我们的目标——一个高效的、流水线化的实现 Y86-64 ISA 的处理器。

4.5.1 SEQ+: 重新安排计算阶段

作为实现流水线化设计的一个过渡步骤, 我们必须稍微调整一下 SEQ 中五个阶段的顺序, 使得更新 PC 阶段在一个时钟周期开始时执行, 而不是结束时才执行。只需要对整体硬件结构做最小的改动, 对于流水线阶段中的活动的时序, 它能工作得更好。我们称这