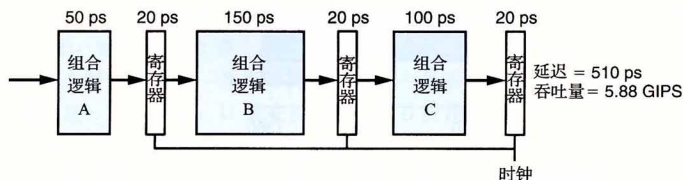
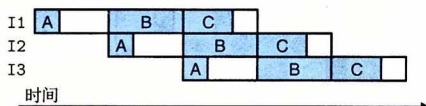


了, 延迟也增加到了 510ps。



a) 硬件: 三阶段流水线, 不一致的阶段延迟

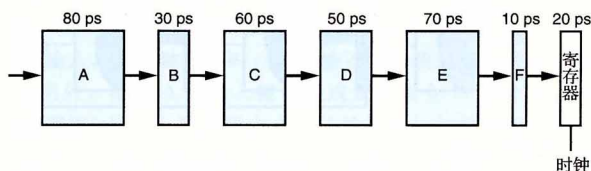


b) 流水线图

图 4-36 由不一致的阶段延迟造成的流水线技术的局限性。系统的吞吐量受最慢阶段的速度所限制

对硬件设计者来说, 将系统计算设计计划分成一组具有相同延迟的阶段是一个严峻的挑战。通常, 处理器中的某些硬件单元, 如 ALU 和内存, 是不能被划分成多个延迟较小的单元的。这就使得创建一组平衡的阶段非常困难。在设计流水线化的 Y86-64 处理器中, 我们不会过于关注这一层次的细节, 但是理解时序优化在实际系统设计中的重要性还是非常重要的。

练习题 4.28 假设我们分析图 4-32 中的组合逻辑, 认为它可以分成 6 个块, 依次命名为 A~F, 延迟分别为 80、30、60、50、70 和 10ps, 如下图所示:



在这些块之间插入流水线寄存器, 就得到这一设计的流水线化的版本。根据在哪里插入流水线寄存器, 会出现不同的流水线深度(有多少个阶段)和最大吞吐量的组合。假设每个流水线寄存器的延迟为 20ps。

- 只插入一个寄存器, 得到一个两阶段的流水线。要使吞吐量最大化, 该在哪里插入寄存器呢? 吞吐量和延迟是多少?
- 要使一个三阶段的流水线的吞吐量最大化, 该将两个寄存器插在哪里呢? 吞吐量和延迟是多少?
- 要使一个四阶段的流水线的吞吐量最大化, 该将三个寄存器插在哪里呢? 吞吐量和延迟是多少?
- 要得到一个吞吐量最大的设计, 至少要有几个阶段? 描述这个设计及其吞吐量和延迟。

2. 流水线过深, 收益反而下降

图 4-37 说明了流水线技术的另一个局限性。在这个例子中, 我们把计算分成了 6 个阶段, 每个阶段需要 50ps。在每对阶段之间插入流水线寄存器就得到了一个六阶段流水线。这个系统的最小时钟周期为 $50 + 20 = 70\text{ps}$, 吞吐量为 14.29 GIPS。因此, 通过将流