Simultaneous Multithreading

Multithreading

对于一个单线程来说，已经很难在 ILP 和 DLP 上更进一步了，但是很多现实中的工作很适合 TLP (Thread-Level Parallelism)

多线程技术就是在单处理器上提升处理器利用率

在流水线中出现依赖时，一般是采用 interlocking（慢）或是数据的 bypassing（需要硬件支持，不一定适用于所有情况）

还有一种解决方法是在同一流水线交叉地运行不同线程的指令，如一个五段流水线可以运行4个线程的指令，保证每个线程中前一条指令的写回一定早于下一条指令读寄存器

每个线程需要其

- 独特的上下文：PC，寄存器
- 独特的系统状态：页表，异常处理寄存器
- 更多的 cache/TLB 冲突

SMT for OoO

之前讨论的都是纵向的多线程，每个流水线 stage 同时只处理一个线程的任务

乱序执行的超标量处理器中，细粒度的 SMT 可以让不同线程的指令在同一周期内一起运行，进一步提高利用率

理想的情况是不同线程的指令完全占满流水线，但现实中的超标量处理器往往有各种程度的浪费

- vertical waste：整个周期内流水线都是 idle
- horizontal waste：一个周期内流水线未占满

vertical multithreading 即当前周期流水线完全空闲时，插入别的线程的指令，能消除 vertical waste 但是还会有 horizontal waste

chip multithreading 即将流水线分成多组，每组执行一个线程，可以一定程度地减少 horizontal waste，而且限制了单线程能达到的最大吞吐率

在乱序执行中，可以添加更大的上下文和 fetch engine 来同时 issue 不同线程的指令，如果是一个线程则可以占用整个机器，这样在并行度高的情况下实现 TLP，没有 TLP 也可以有超标量的 ILP