CPU

指令流水线

一条指令的执行过程可以分为多个阶段或过程。

1. 顺序执行方式
    - 传统冯诺依曼机采用顺序执行（也称串行执行方式）。总耗时$T = n\times3t = 3nt$
    - 控制简单，硬件代价小；
    - 但执行指令的速度慢，任何时刻只要有一条指令在执行，各功能部件的利用率低。
2. 一次重叠执行方式
    - 总耗时$T = 3t+(n-1)\times2t = (1+2n)t$
    - 程序执行时间缩短了$1/3$，各功能部件利用率明显提高；
    - 但硬件开销较大，控制过程比顺序执行复杂。
3. 二次重叠方式
    - 总耗时$T = 3t+(n-1)\times t = (2+n)t$
    - 指令执行时间缩短近$2/3$。一种理想的指令执行方式，正常情况下，处理机同时又2条指令在执行。
    - （还有三次、四次或更多折叠方式）

流水线的性能指标

1. **吞吐率**：单位时间内流水线所完成的任务数量，或是输出结果的数量。若任务数为n，处理完成n个任务所用的时间Tk，则计算流水线吞吐率TP的基本公式为：$TP = n/Tk$。
    - 当连续输入任务$n->\infty$时，最大吞吐量$TPmax = 1/\triangle t$。
    - 一条指令的执行分为k个阶段，每个阶段耗时$\triangle t$，一般取$\triangle t$=一个始终周
    - 流水线实际吞吐率为$TP = n/((k+n-1)\triangle t)$
    - 流水线时空图在头尾有装入时间和排空时间。
2. **加速率**：完成同样任务，不使用流水线与否的时间之比。
    - 不使用流水线的执行时间T0（顺序执行所用时间），使用流水线的执行时间Tk，流水线加速比$S = T0/Tk$
    - 当连续输入任务$n->\infty$时，最大加速比$Smax = k$
    - 顺序完成n个任务耗时$T0 = nk\triangle t$，流水线完成n个任务耗时$Tk = (k+n-1)\triangle t$，实际加速比$S = kn\triangle t/(k+n-1)\triangle t = kn/(k+n-1)$
3. **效率**：流水线的设备利用率。
    - 流水线效率$E = 完成n个任务占用的时空区有效面积/n个任务所用时间与k个流水段所谓成的时空区总面积 = T0/(kTk)$

指令流水线的影响因素

MIPS架构的五段流水线：取指令IF，指令译码ID，指令执行EX，访存M，写回通用寄存器WB

为了方便流水线设计，会将每阶段耗时取一样（以最长的为准），则流水线每个部件后面都要有一个缓冲寄存器（锁存器）来保存本流水段的执行结果，供下一段流水使用。

1. **结构相关**（资源冲突）
    - 多条指令在同一时刻争用同一资源而形成的冲突
    - 解决方法：
        1. 可以让后一相关指令暂停一周期
        2. 或资源重复配置：数据存储器+指令存储器
2. ***数据相关***（数据冲突）
    - 一个程序中，存在必须等前一条指令执行完后才能执行后一条指令的情况
    - 解决方法：
        1. 相关指令以及后续指令都暂停1~数个时钟周期，直到数据相关问题消失后继续。可用*硬件阻塞stall（纯硬件阻塞）或软件插入NOP（空指令阻塞）*
        2. 数据旁路技术：处理后的数据除了写回原定寄存器外，还额外送入下一步需要进行操作的部件
        3. 编译优化：编译器调整指令顺序来解决
3. **控制相关**（控制冲突）
    - 放流水线遇到转移指令和其他改变PC的指令而造成断流
    - 解决方法：
        1. 转移指令分支预测。有*简单预测（永远猜TrueOrFalse）*、*动态预测（根据历史情况动态调整）*
        2. 预取转移成功和不成功的两个控制流方向上的目标命令
        3. 加快和提前形成的条件码
        4. 提高转移方向的猜准率

流水线的分类

1. **部件功能级、处理机级、处理机间级流水线**：根据流水线的级别不同而分类
    1. 部件功能级流水：将复杂的算术逻辑运算组成流水线。例如将浮点加减操作分为求接差、对阶、尾数相加、结果规格化等过程
    2. 处理机级流水：一条指令解释过程分为多个子过程。例如取址、译码、执行、访存、写回
    3. 处理机间流水：宏流水，每个处理机完成某一专门任务，各处理机所得到的结果许存放在与下一个处理机所共享的存储器中
2. **单功能流水线、多功能流水线**：根据流水线可完成的功能分类
    1. 单功能流水线：只能实现一种固定的专门功能的流水线
    2. 多功能流水线：通过各段间的不同连接方式可以实现多种功能的流水线
3. **动态流水线、静态流水线**：根据同一时间内各段之间的连接方式分类
    1. 静态流水线：同一时间内，流水线各段只能按某一功能的连接方式工作

2. 动态流水线：同一时间内，当某些段正在实现某种运算，另一些段正在进行另一种运算。高效，但流水线控制复杂

4. **线性流水线、非线性流水线**：根据各个功能段之间是否有反馈信号分类

    1. 线性流水线：从输入到输出，各功能段只允许经过一次，不存在反馈回路
    2. 非线性流水线：存在反馈回路。适合进行线性递归运算

流水线的多发技术

1. **超标量技术**：

    - 每个时钟周期内可*并发多条独立指令*
    - 要配置多个功能部件
    - *不能调整*指令的*执行顺序*
    - 通过编译优化技术，把可并行的执行指令搭配

2. **超流水技术**：

    - *在一个时钟周期*内*再分段*（3段）
    - 在一个时钟周期内，*一个功能部件使用多次*（3次）
    - *不能调整*指令的*执行顺序*
    - 靠*编译程序*解决*优化问题*

3. **超长指令字**：

    - 由*编译程序挖掘*出指令间*潜在*的*并行性*
    - 将*多条*能*并行操作*的指令组合成*一条*
    - 具有*多个操作码字段*的*超长指令字*（可达几百位）
    - 采用*多个处理部件*

五段式指令流水线

MIPS架构的五段流水线：取指令IF，指令译码ID，指令执行EX，访存M，写回通用寄存器WB

常见五类指令：运算类、LOAD、STORE、条件转移、无条件转移

运算类：
IF：根据PC从指令Cache取指令到IF锁存器;
ID：去除操作数到ID段锁存器；
EX：运算，将结果存入EX段锁存器；
M：空段 WB：将运算结果写回指定寄存器

LOAD：
IF：根据PC从指令Cache取指令到IF锁存器；
ID：将基址寄存器的值放到锁存器A，将偏移量的值放到Imm；
EX：运算，得到有效地址；
M：从数据Cache中取数并放入锁存器；
WB：将取出的数写回寄存器

STORE:
IF：根据PC从指令Cache取指令到IF段锁存器；
ID：将基址寄存器的值放到锁存器A，将偏移量的值放到Imm。将要存的数放到B；
EX：运算得到有效地址。并将锁存器B的内容放到锁存器Store；
M：写入数据Cache；
WB：空段

条件转移:
IF：根据PC从指令Cache取指令到IF段的锁存器；
ID：进行比较的两个数放入锁存器A、B，偏移量放入Imm；
EX：运算，比较两数；
M：将目标PC值写回PC WB：空段

无条件转移:
IF：根据PC从指令Cache取指令到IF段锁存器；
ID：偏移量放入Imm；
EX：将目标PC值写回PC；
M：空段；
WB：空段

多处理器

- 多处理器
    - 多处理器multiprocessor：两个或以上处理器的计算机系统
    - 任务级并行task-level parallelism或进程级并行process-level parallelism：同时运行独立程序来利用多处理器
    - 并行处理程序：同时运行在多个处理器上的单一程序
    - 集群cluster：局域网连接的一组计算机，等同于一个大型多处理器
    - 多核处理器multicore microprocessor：单一集成电路上包含多个处理器的微处理器
    - 共享内存处理器shared memory processor(SMP)：共享一个物理地址空间的并行处理器（等同于多核处理器，只是命名角度不同）

- 处理器和向量处理器
    - SISD：单指令流单数据流的单处理器
        - 一个处理器+一个主存储器
        - 各指令序列只能并发、不能并行、每条指令处理一两个数据
        - 不是数据级并行技术
        - 若采用指令流水线，需要设置多个功能部件，采用多模块交叉存储器
    - SIMD：单指令流多数据流（同样的指令在多个数据流上操作，和向量处理器中一样）。
        - 一个CU+多个处理单元或执行单元+多个局部存储器+一个主存储器
        - 各指令序列只能并发、不能并行，但每条指令可以同时处理多个具有相同特征的数据
        - 数据级并行技术
        - 有三种变体：向量体系结构、多媒体SMD指令集扩展、图形处理单元
    - MIMD：多指令流多数据流的多处理器（共享存储多处理器系统）
        - 个指令并行执行，分别处理多个不同数据
        - 一种线程级并行、甚至是线程级以上的并行技术
        - 细分：

        - 多处理器系统：多个处理器共享单一的物理地址空间；
          各处理器间通过LOAD/STORE访问同一个主存储器，通过主存相互传送数据
        - 多计算机系统：多台计算机之间只能通过"消息传递"相互传送数据；
          每台计算机拥有各自的私有存储器，物理地址空间相互独立
    - SPMD：单程序多数据流（一种传统MIMD编程模型，其中一个程序运行在所有处理器之上）
    - 数据级并行：不同数据执行相同操作所获得的并行
    - 陈列处理器：SIMD的一种，并行处理机
    - 向量机（向量处理器）：SIMD的一种
        - 多个处理单元，多组向量寄存器
        - 一条指令的处理对象是"向量"，擅长对向量型数据并行计算、浮点数运算，常用于超算
        - 主存采用"多个端口同时读取"的交叉多模块处理器
    - 标量体系结构：常规指令集体系结构

- 硬件多线程

    - 硬件多线程：线程阻塞时处理器可切换到另一线程（MIMD相关）。有三种种实现方法：
        - 细粒度多线程
        - 粗粒度多线程
        - 同时多线程SMT
    - 进程process：一个进程包含≥1个线程、地址空间、操作系统。进程切换通常需要操作系统介入
    - 线程thread：一个线程包含计数器、寄存器状态、内存栈，一个轻量级进程，多个线程共享一个地址空间（进程不是）

-	**细粒度多线程**	**粗粒度多线程**	**同时多线程SMT**
指令发射	各个时钟周期，轮流发射多个线程指令	连续几个时钟周期都发射同一线程的指令序列； 流水线阻塞时切换到另一个线程	一个时钟周期内，同时发射多个线程指令
线程切换频率	每个时钟周期	当流水线阻塞时	NULL
线程切换代价	低	高，需要重载流水线	NULL
并行性	指令级并行，线程间不并行	指令级并行，线程间不并行	指令级并行，线程级并行

- 多核处理器

    - CPU架构由 控制单元、运算单元、存储单元 这三个模块，由CPU内部总线连接。
      多核CPU则是多个核组织（多个控制单元和多个运算单元）共用存储单元。
      多核结构则在CPU内布置多个执行核（执行单元），如ALU、FPU、L2缓存，其它部分由多个核共享。
      由于布置了多个核，其指令级并行是真正并行，而非超线程结构半并行。

- 共享内存多处理器SMP

    - 共享内存多处理器SMP（共享地址多处理器）：
        - 硬件为所有处理器提供单一物理地址空间

        - 用锁来同步共享变量
            - 同步：对可能运行不同处理器上的两个或更多进程的行为进行协调
            - 锁：一个时刻进允许一个处理器访问数据的同步装置
    - SMP体系结构
        - 采用相关技术以确定和同步对资源的相关占用声明，确保进程不会从队列中丢失。但由于必须确保两个处理器不会选择同一个进程，增加了操作系统的复杂性。
    - SMP组织结构
        - 有多个处理器，每个都有自己的控制单元、算术逻辑单元、存储器
        - 每个处理器可以通过某种形式的互联机制访问一个共享内存和IO设备
        - 共享总线是一个通用方法

- 多处理器操作系统的关键涉及问题：

    - 同时对并发进程或线程
    - 调度
    - 同步
    - 存储管理
    - 可靠性和容错