1. **系统架构重大转折** 指令级并行（流水线）转向线程级并行（多核）和数据级并行（并行计算）
2. **Flynn 分类法** 指令流+数据流（单指令流单数据流 SISD SIMD MISD MIMD）
3. **冯氏分类法** 用最大并行度分类
4. **提高并行性的技术途径** 时间重叠 资源重复 资源共享
5. **多机系统 3 种多处理机** 异构型多处理机 同构型多处理机 分布式系统

1. **指令集结构的三种类型** 堆栈结构 累加器结构 通用寄存器结构

1. **流水线的每个子过程及其功能部件称为** 流水线的级/段
2. **流水线的段数称为** 流水线的深度
3. **部件级流水线/运算操作流水线** 把处理机的算术逻辑运算分段，流水进行
4. **按照同一时间内各段之间的连接方式** 静态流水线 动态流水线
5. **按照流水线所完成的功能来分类** 单功能流水线 多功能流水线
6. **按照流水的级别来进行分类** 部件级流水线 处理机级流水线（指令） 处理机间流水线
7. **按照流水线中是否有反馈回路** 线性流水线与非线性流水线
8. **根据任务流入和流出的顺序是否相同** 顺序流水线与乱序流水线
9. **解决流水线瓶颈的方法** 细分瓶颈段 重复设置瓶颈段
10. **三种相关** 数据相关（写后读） 名相关 控制相关（分支）
11. **名相关的两个种类** 反相关（读后写） 输出相关（写后写）
12. **三种冲突** 结构冲突（硬件资源）数据冲突 控制冲突
13. **三种通过软件来减少分支延迟** 预测分支成功 预测分支失败 延迟分支（延迟槽）
14. **MIPS 指令的五个时钟周期**

    取指令周期（IF）

    指令译码/读寄存器周期（ID）

    执行/有效地址计算周期（EX）

    存储器访问/分支完成周期（MEM）

    写回周期（WB）

1. **两种解决相关问题的方法**

   保持相关，避免冲突

   代码变换（换名），消除冲突

2. **正确执行程序必须保持的两个属性**

3. **译码阶段细分为：**流出（IS） 读操作数（RO）

4. **两个典型的动态调度算法：**记分牌算法 Tomasulo 算法

5. **浮点加法器有 3 个保留站，浮点乘法器有 2 个保留站**

1. **三级存储系统** Cache 主存储器 磁盘存储器（辅存）

2. **Cache-主存层次主要由硬件实现，主存-辅存系统主要由软件实现**

3. **存储层次的四个问题：**

   映像规则 查找算法 替换算法 写策略（过一遍 cache 工作流程）

4. **三个映像规则：**直接映像 全相联映像 组相联映像

5. **查找方式：**通过查找目录表实现，分为并行查找和顺序查找

6. **主要的三种替换算法：**随机法 FIFO 先进先出法 LRU 最近最少使用法

7. **两种写策略** 写回法 写直达法

8. **写不命中时候的两种策略：**按写分配（写时取） 不按写分配法（绕写法）

9. **CPU 时间 =** IC*(CPI+每条指令平均访存次数*不命中率*不命中开销)*时钟周期

10. **三个方面改进 cache 性能：**

    降低不命中率 减少不命中开销 减少 Cache 命中时间

11. **三种类型的不命中：**强制性不命中 容量不命中 冲突不命中

12. **降低不命中率 （首先考虑三种类型不命中）**

    增加块大小（减少强制不命中）

    增加 Cache 容量（减少容量不命中）

    提高相联度（减少冲突不命中）

    伪相联 Cache、硬件预取、编译器控制预取、编译器优化、牺牲 Cache

在逻辑上把Cache的空间上下平分为两个区。对于任何一次访问，伪相联Cache先按直接映像Cache的方式去处理。若命中，则其访问过程与直接映像Cache的情况一样。（快）

若不命中，则再到另一区相应的位置去查找。若找到，则发生了伪命中，否则就只好访问下一级存储器。（慢） 至 CPU

13. 减少不命中开销

两级 Cache、让读不命中优先于写、写缓冲合并、请求字处理、非阻塞 Cache

14. **减少命中时间**

小容量简单 Cache、虚拟 Cache（直接用虚拟地址访问）、流水化 Cache 访问、踪迹 Cache

1. **小容量简单Cache**：通过减少容量和简化映射策略（如直接映射）降低访问延迟，但可能增加冲突不命中。
2. **虚拟Cache**：直接使用虚拟地址访问Cache，省去TLB转换时间，但需处理进程切换和别名问题。
3. **流水化Cache访问**：将Cache访问拆分为流水段以提高主频，适合高频CPU，但增加分支预测失败惩罚。
4. **踪迹Cache**：缓存动态指令序列（如已解码指令），减少解码时间，尤其适合含分支的代码段。

15. 并行主存系统主要技术（2 种） 单体多字存储器 多体交叉存储器

1. **反映外设可靠性能的参数：**可靠性 可用性 可信性
2. **评价 IO 系统性能的参数：**连接特性 容量 响应时间、吞吐率
3. **三种通道：**

字节多路通道 以字节为单位轮流为多个低速设备服务，适合连接键盘、鼠标等低带宽设备。

数组多路通道 以数据块（数组）为单位分时服务多个中速设备，平衡效率与并发性。

选择通道 独占式服务单个高速设备，传输完成后才切换，适合高带宽需求。

试比较三种通道的优缺点及适用场合。

答：(1) 字节多路通道。一种简单的共享通道，主要为多台低速或中速的外围设备服务。(2) 数组多路通道。适于为高速设备服务。(3) 选择通道。为多台高速外围设备（如磁盘存储器等）服务的。

1. **MIMD 机器的两类存储器组织结构：**

集中式共享存储器结构（SMP/UMA）

分布式存储器多处理机

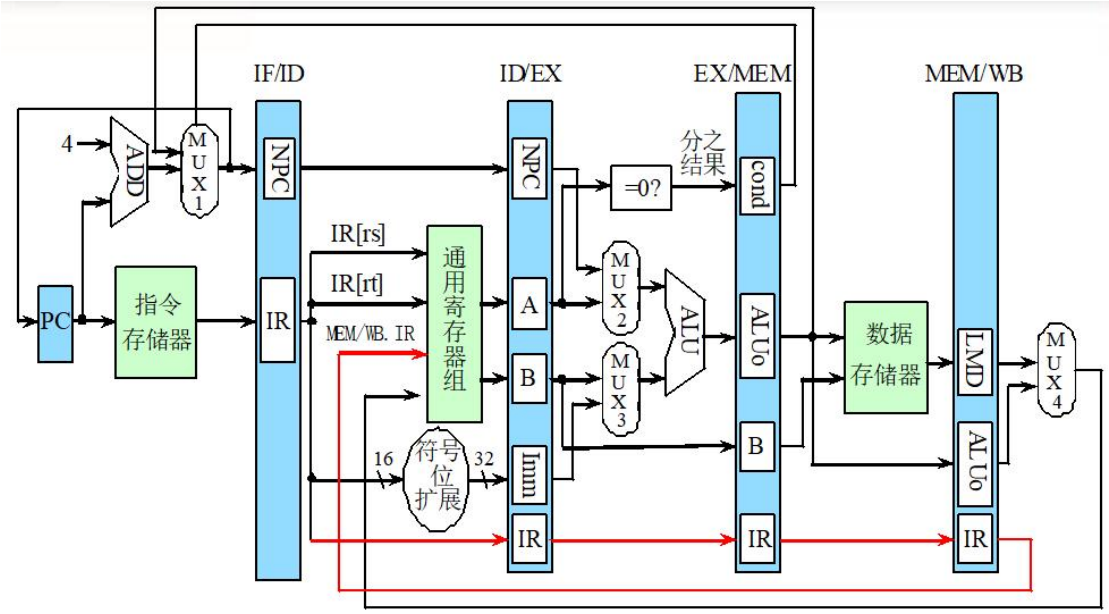
1. **单一系统映像四重含义（SSI）：**单一系统、单一控制、对称性、位置透明
2. **机群的特点：**开发周期短、可靠性高、可扩缩性强、性能价格比高、用户编程方便
3. **机群的分类（按使用目的分）：**高可用性机群、负载均衡机群、高性能机群

1. 指令系统编码格式
   a) 定长编码
   b) 变长编码
   c) 混合编码。
2. **延迟分支方法的调度策略**（无关指令+分支两种情况）
   a) **从前调度**（将无关指令提前到分支指令前）
   b) **从目标处调度**（填充分支目标指令）
   c) **从失败处调度**（填充分支失败路径指令）。

计算机系统设计中常用的 4 个定量原理及含义：

1. **大概率事件优先原理**：将资源优先分配给发生概率高的事件，如 CPU 缓存优先存储高频访问数据，提升整体效率。
2. **阿姆达尔定律（Amdahl's Law）**：计算系统中对某部件改进后，整体性能提升程度，受该部件执行时间占比和改进加速比影响，公式$S = \frac{1}{(1-f)+\frac{f}{k}}$，$S$为加速比，$f$为改进部件时间占比，$k$为改进加速比。
3. **程序的局部性原理**：程序执行时，访问内存具有时间局部性（近期访问过的内容易再访问）和空间局部性（访问地址相邻内容易被访问），用于优化存储系统，如 Cache 设计。
4. **CPU 性能公式**：CPU时间 = 指令数 × CPI × 时钟周期时间，明确指令数、每条指令周期数（CPI）、时钟周期对性能的影响，指导性能优化。

第一个是以**经常性事件为重点**



1. 一条指令的执行过程分为以下5个周期：

- 取指令周期（IF）
  - IR ← Mem[PC] 。
  - PC值加4。（假设每条指令占4个字节）
- 指令译码/读寄存器周期（ID）
  - 译码。
  - 用IR中的寄存器编号去访问通用寄存器组，读出所需的操作数。
- 执行/有效地址计算周期（EX）

  不同指令所进行的操作不同：
  - 存储器访问指令：ALU把所指定的寄存器的内容与偏移量相加，形成用于访存的有效地址。
  - 寄存器—寄存器ALU指令：ALU按照操作码指定的操作对从通用寄存器组中读取的数据进行运算。

- 寄存器—立即数ALU指令：ALU按照操作码指定的操作对从通用寄存器组中读取的第一操作数和立即数进行运算。
- 分支指令：ALU把偏移量与PC值相加，形成转移目标的地址。同时，对在前一个周期读出的操作数进行判断，确定分支是否成功。

➢ 存储器访问／分支完成周期（MEM）

该周期处理的指令只有load、store和分支指令。其他类型的指令在此周期不做任何操作。

- load和store指令

load指令：用上一个周期计算出的有效地址从存储器中读出相应的数据。

store指令：把指定的数据写入这个有效地址所指出的存储器单元。

- 分支指令

分支“成功”，就把转移目标地址送入PC。

分支指令执行完成。

➢ 写回周期（WB）

ALU运算指令和load指令在这个周期把结果数据写入通用寄存器组。

ALU运算指令：结果数据来自ALU。

load指令：结果数据来自存储器系统。

- 磁盘阵列：使用多个磁盘的组合来代替一个大容量磁盘。
- 廉价磁盘冗余阵列：存放冗余信息以便磁盘信息丢失时进行恢复的磁盘阵列。
- RAID分级

RAID级别	名称	说明
RAID0	非冗余磁盘阵列	条带存放，没有纠错能力

RAID级别	名称	说明
RAID1	镜像磁盘	复制一份，成本高
RAID2	存储器式磁盘阵列	利用Hamming纠错码，代价为log级
RAID3	位交叉奇偶校验磁盘阵列	空间开销为1，大规模读写带宽高
RAID4	块交叉奇偶校验磁盘阵列	空间开销为1，小规模读写带宽高
RAID5	块交叉分布奇偶校验磁盘阵列	空间开销为1，小规模读写带宽高
RAID6	P+Q双校验磁盘阵列	能容忍2个故障，但空间开销为2

- RAID1+0和RAID0+1：RAID0和RAID1结合的结果

  RAID10：先进行镜像，然后再进行条带存放

  RAID01：先进行条带存放，然后再进行镜像

条带 镜像