



第七讲

计算机组成的新发展



第八讲

计算机结构的新发展

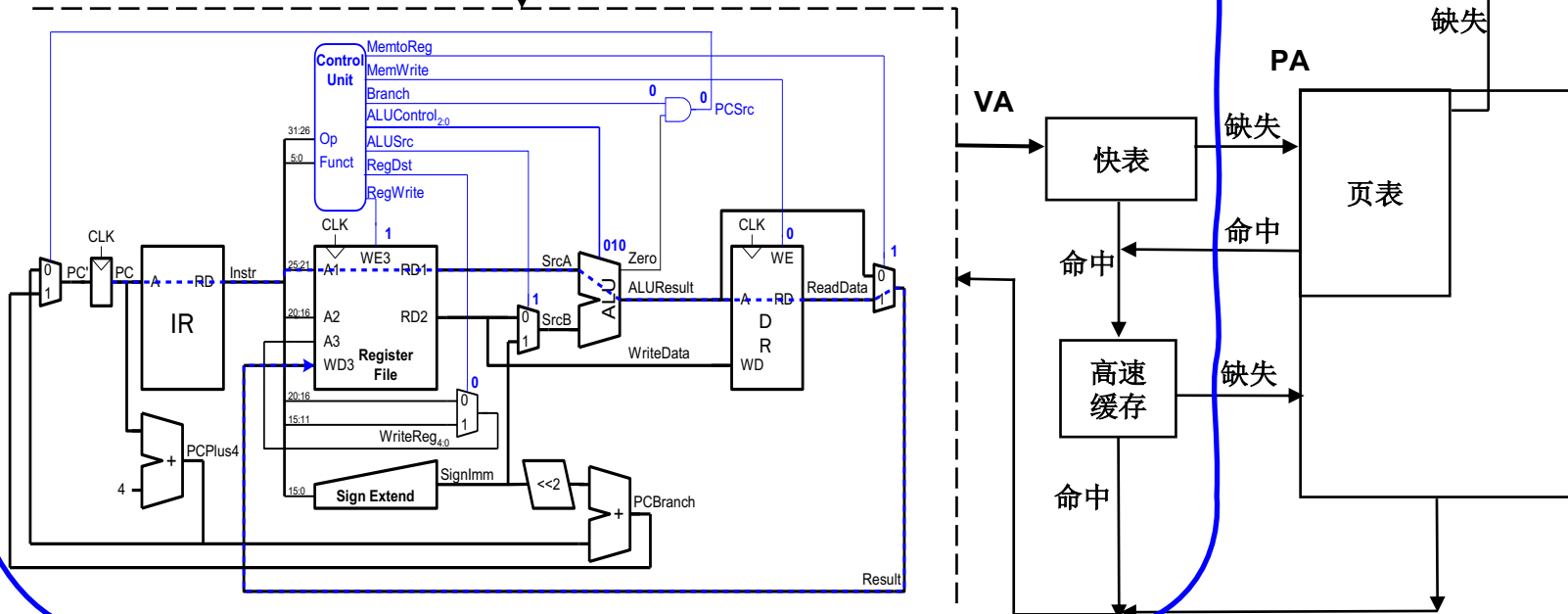


教学型计算机

I/O子系统

主机系统

中断判别与处理机制





为什么要学习教学型的计算机

- 硬件无所不在，软、硬件紧密相关
- 新技术总是在基础技术上一步步发展而来
- 需求时时变化，但基础技术不变
- 基于技术的能力是未来发展的基础
- 对于计算机领域不同层次和方向的研究生，“基础”技术的定义不同



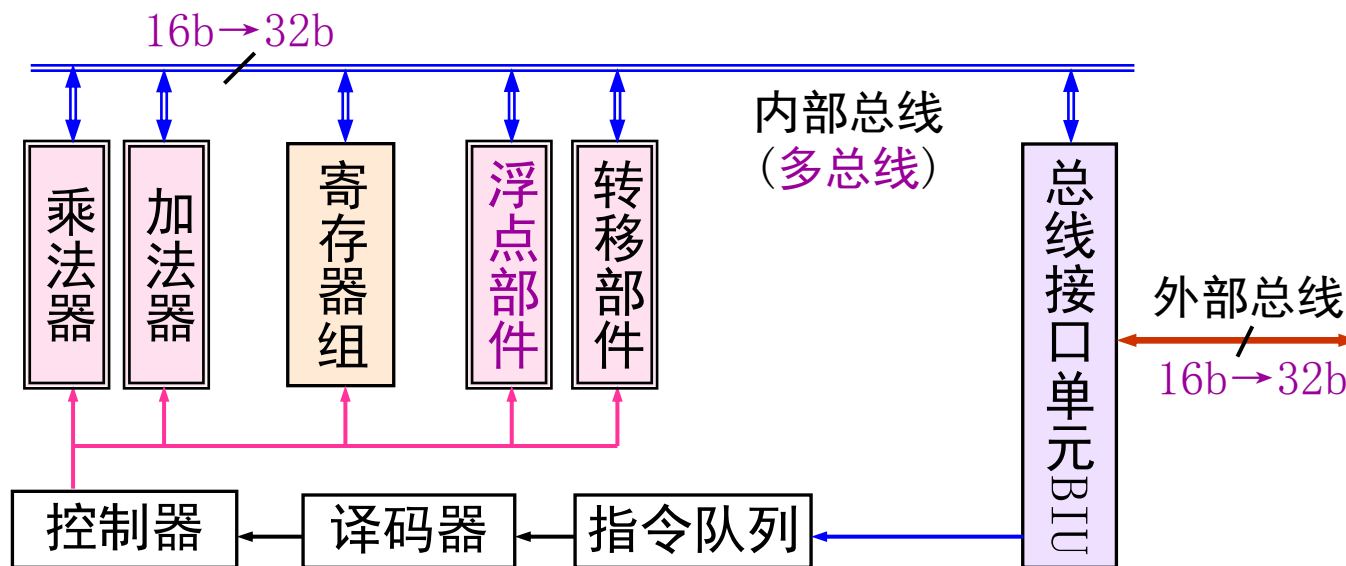
如何提升CPU的性能

- 冯诺依曼模型的性能瓶颈
 - CPU-MEM、指令串行执行
- 提高性能的方法
 - 更快的速度、并行工作方式
- 硬件方面的优化思路
 - 提高指令系统性能—增加新功能指令等
 - 提高主频—改进器件、电路等技术
 - 改进结构
 - 提高CPI (Cycles per Instruction)
 - 提高ILP (Instruction Level Parallel)
 - 提高OS效率—增加便于OS工作的相应硬件
 - 提高访存速度—选择快速器件、改进MEM结构
 - 提高I/O速度—采用I/O接口、改进I/O方式等
- 体系结构是提升硬件性能的基础



操作级结构：强化CPU组件

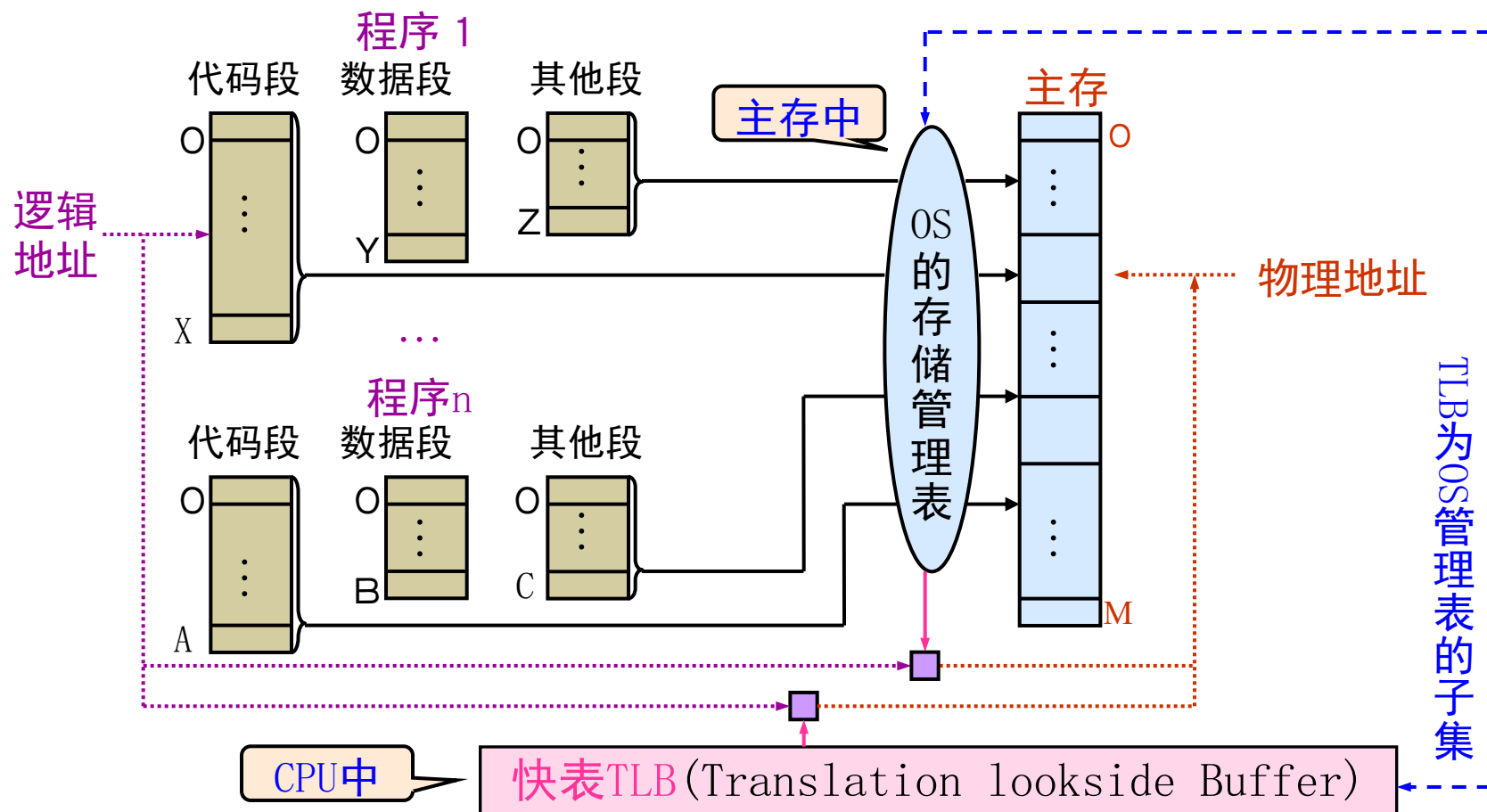
- 强化CPU组件
 - 优化组件性能：增加CPU字长、提高主频、多总线通路
 - 增加指令功能：扩展指令系统(需相应增加硬件)





操作级结构：改进存储系统（1）

- 改进存储系统
 - 采用虚拟存储器
 - 有效支持多任务OS(减轻程序员负担)
 - 增设快表提高地址变换速度。





- 增设Cache→多级Cache

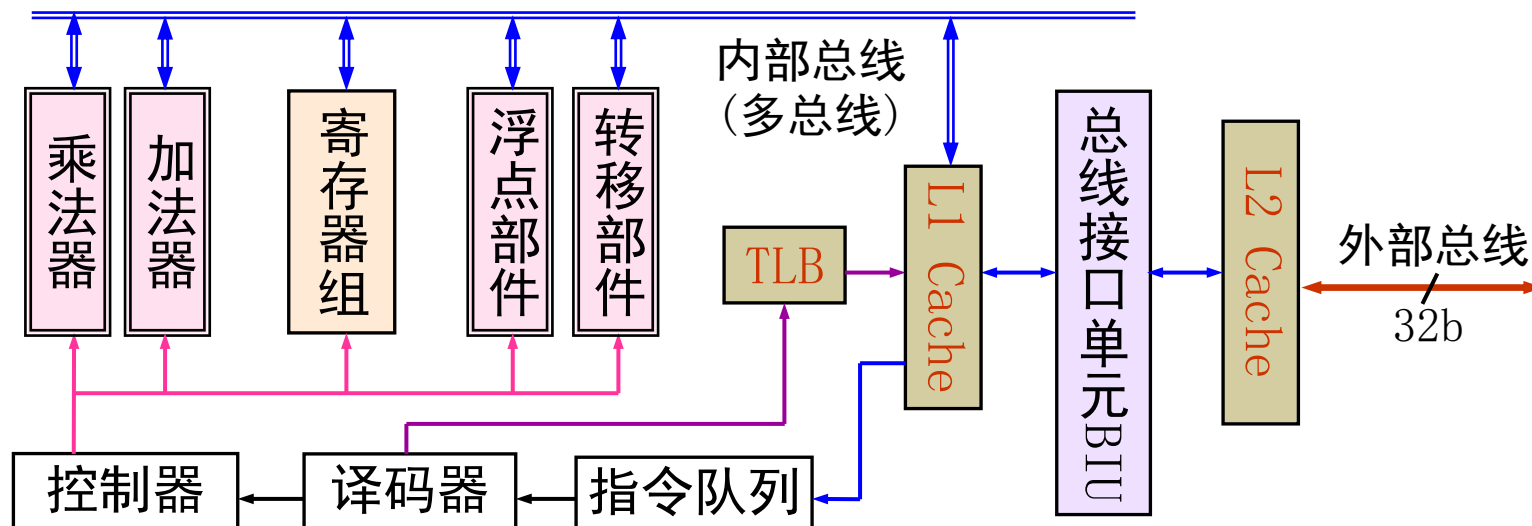
- 利用程序访存局部性规律，提高访存速度(弱化CPU-MEM瓶颈)及性能-价格比(容量 $\text{Cache} \ll \text{容量}_{\text{MEM}}$)。





操作级结构：改进存储系统（3）

- 改进存储系统
 - 改善系统总线性能：
 - 提高总线时钟频率、增加A/D宽度、多级总线等

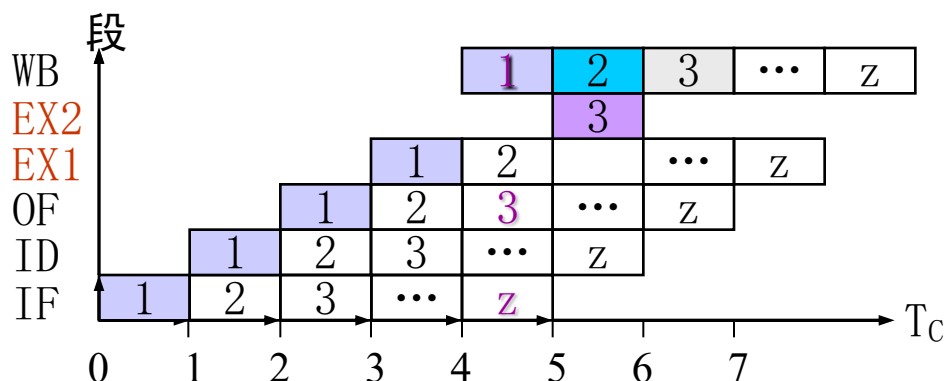




操作级结构：改进CPU结构

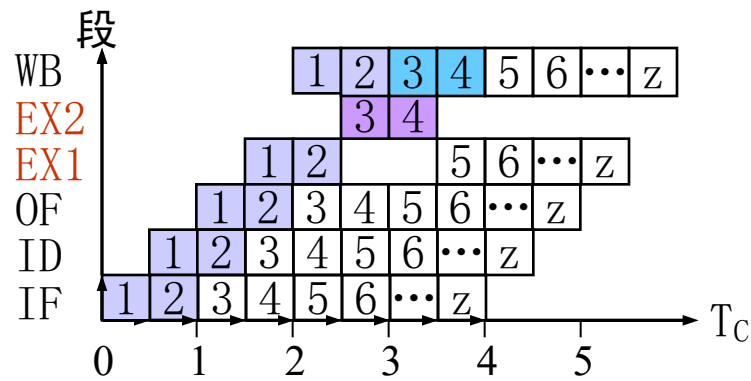
改进CPU结构

- 开发操作级并行性：流水线及超级流水线技术
- 流水线技术：指令执行各步骤重叠，减少程序执行时间



普通流水线：ILP(1, 1)

$$T_{CPU} = [5 + (I_N - 1)] * T_c$$



超级流水线：ILP(1, 2)

$$T_{CPU} = [5 + (I_N - 1)] * (T_c / 2)$$

说明：ILP(m, n) —— m为同时启动的指令或操作，n为每个 T_c 启动次数。

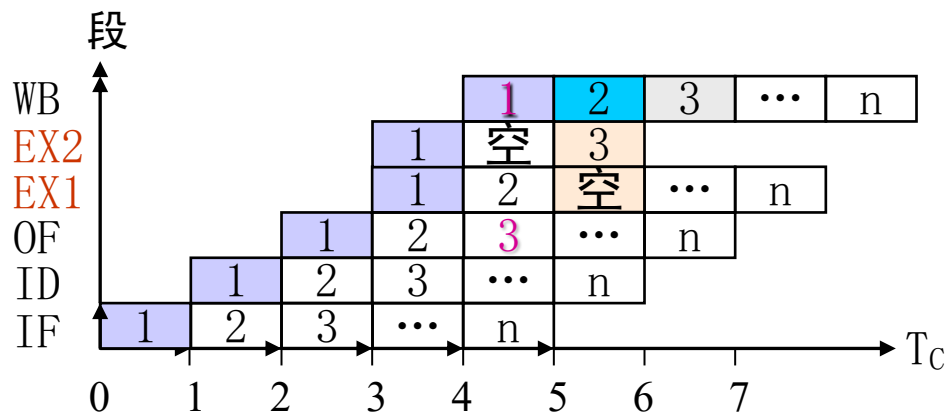
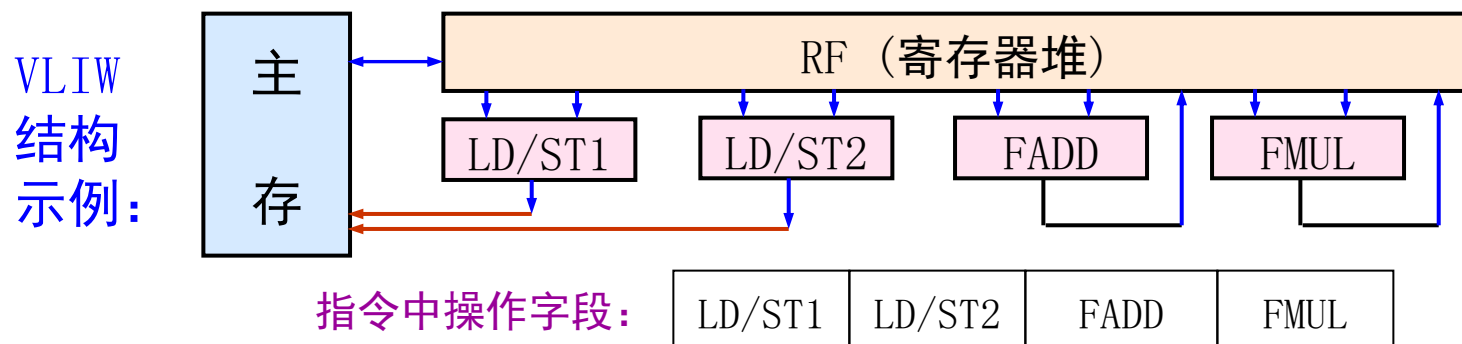
需解决问题：

- (1) 结构相关 (资源使用冲突)；
- (2) 指令间的数据相关及控制相关；
- (3) 提高执行部件EX利用率。



指令级结构：指令并行（1）

- 提升指令级并行性
 - VLIW
 - 流水指令包含多种操作，提高EX的利用率；

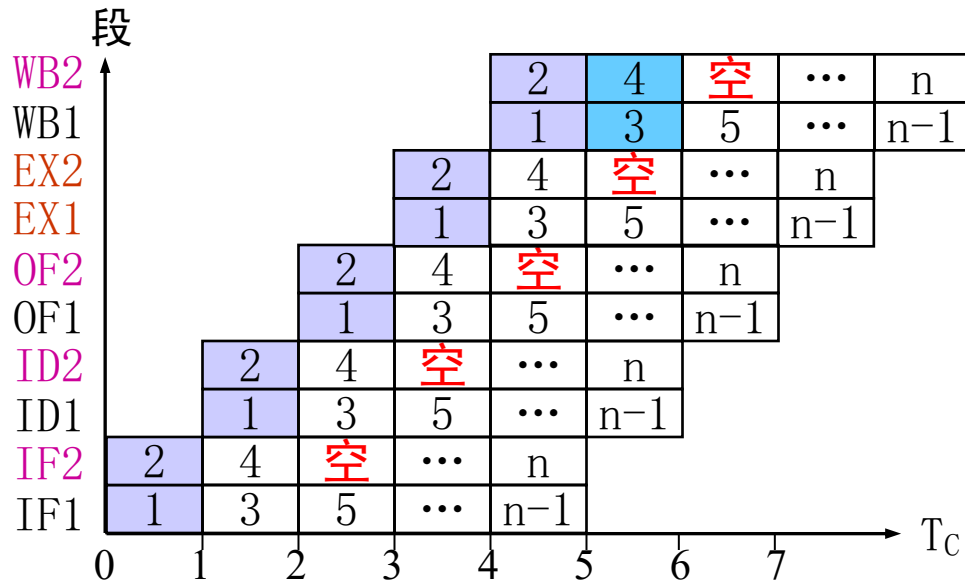


需解决问题：
目标代码效率（即编译程序效率）。



指令级结构：指令并行（2）

- 提升指令级并行性
 - 超标量流水技术：
 - 多条指令并行流水



超标量流水线：ILP(2, 1)

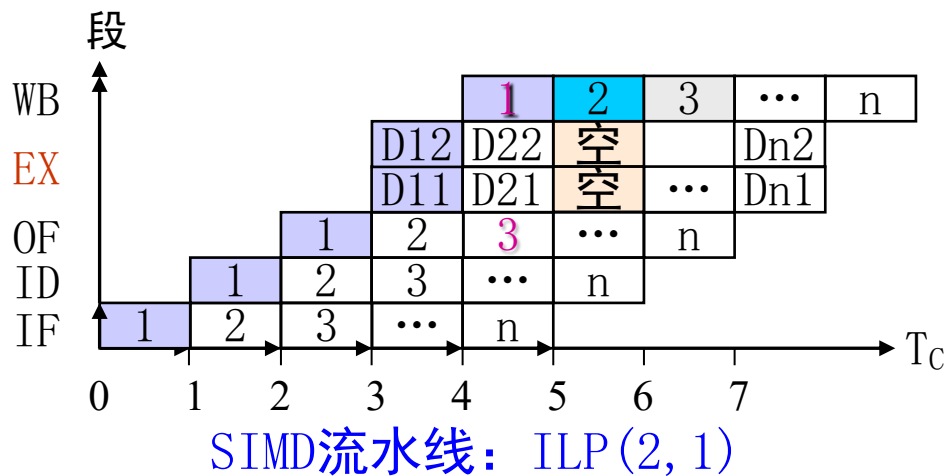
需解决问题：

- (1)多I、D并行存取；
- (2)更严重的数据相关。



指令级结构：指令并行（3）

- 提升指令级并行性
 - SIMD流水技术：
 - 流水指令可处理多个数据



◇需解决问题：

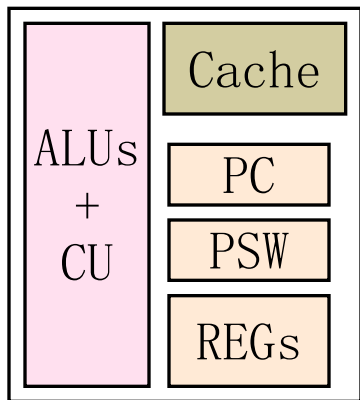
(1)数据存取宽度；

(2)支持新数据类型。

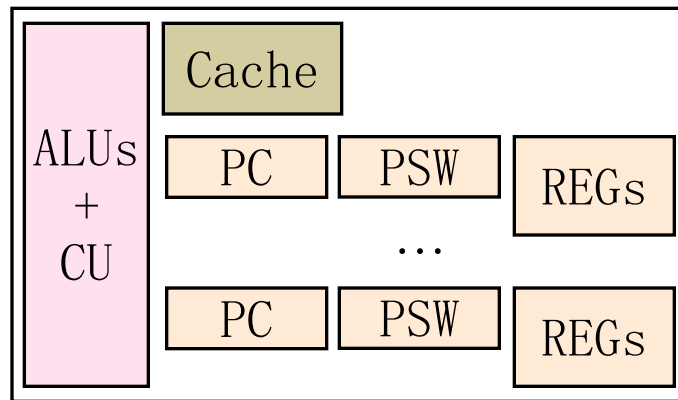


程序级结构：程序并行

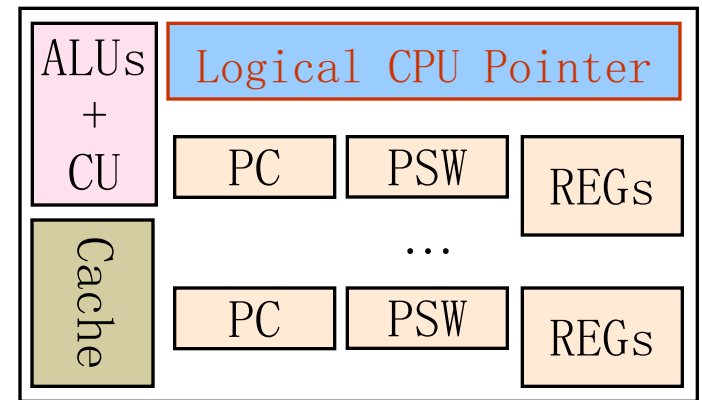
- 提升程序级并行性
 - 多线程(MT)
 - 超线程(HT) (即同时多线程(SMT))



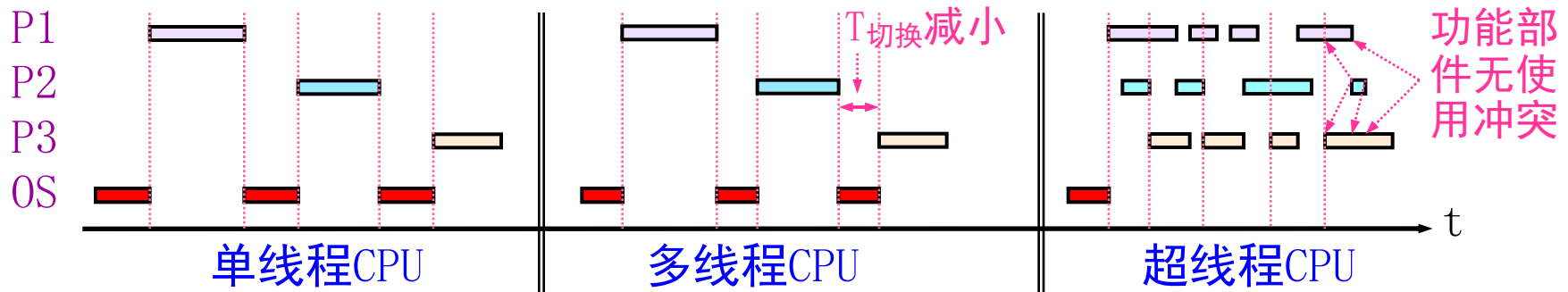
单线程 (ST) CPU



多线程 (MT) CPU



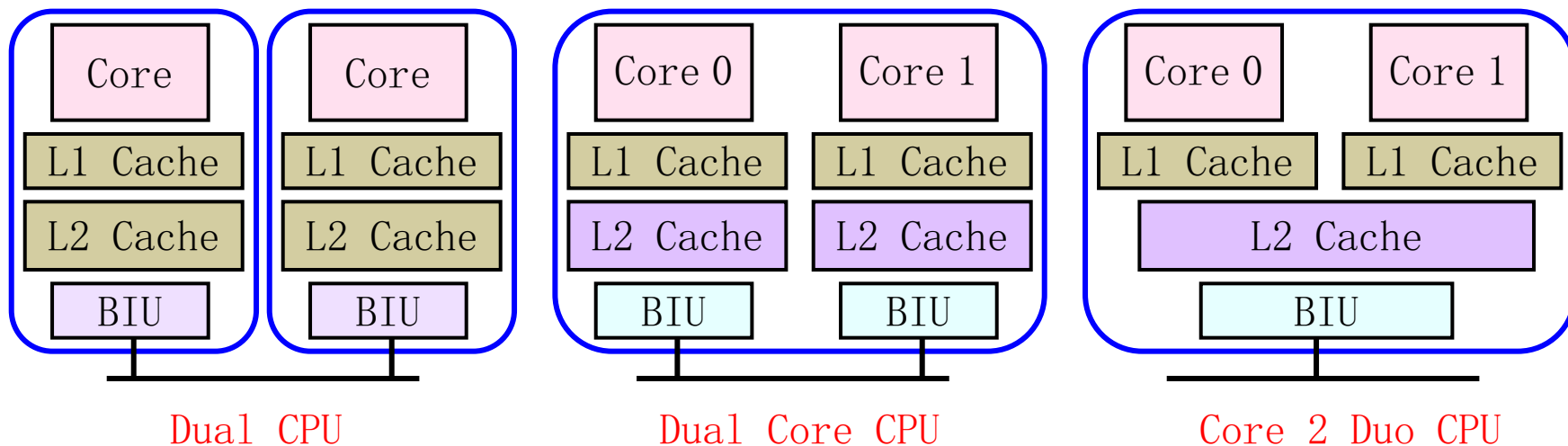
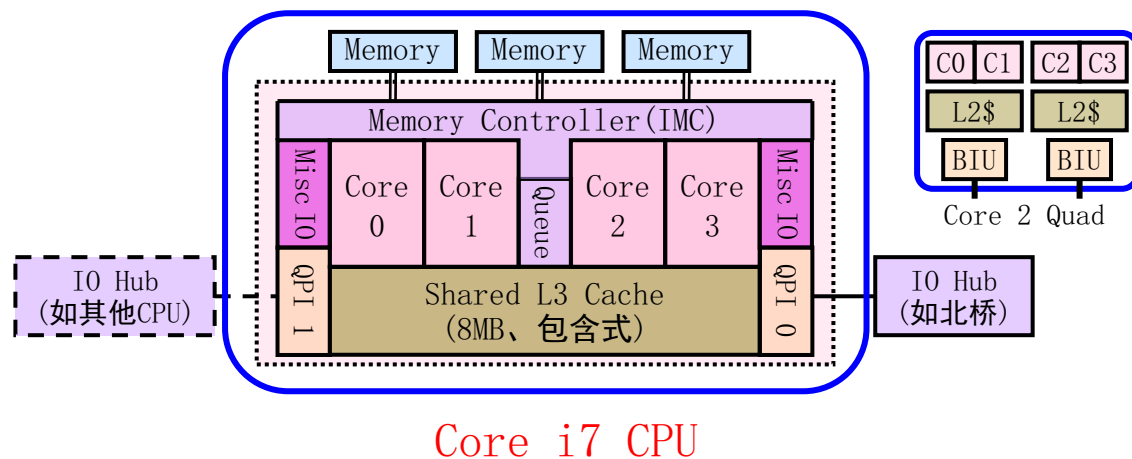
超线程 (HT) CPU





CPU级结构：核并行

- 提升CPU级并行性
 - 多CPU
 - 多核CPU



多核CPU与超线程CPU:

- *超线程CPU—多个逻辑CPU(要求资源使用不冲突);
- *多核CPU—多个物理CPU(资源使用不会冲突)。



问题和讨论