



# 第9章 并行组织与结构

山东科技大学杨晓东

[sdustyxd@163.com](mailto:sdustyxd@163.com)

QQ: 124998396

Phone:18660860091



## 如何提高计算机效率

### 1. 提高访存速度

高速芯片

Cache

多体并行

### 2. 提高 I/O 和主机之间的传送速度

中断

DMA

通道

I/O 处理机

多总线

### 3. 提高运算器速度

高速芯片

改进算法

快速进位链

### 4. 提高传输速度

提高带宽

多总线

- 提高整机处理能力

高速器件

改进系统结构，开发系统的并行性



### 1. 并行性概念

**(1) 并行性：**指计算机系统具有可以同时进行运算或操作的特性，它包括**同时性**与**并发性**两种含义

- 同时性：**两个或两个以上的事件在同一时刻发生。
- 并发性：**两个或两个以上的事件在同一时间间隔发生

**(2) 计算机系统中并行性等级：**

作业级（程序级）

任务级（进程级）

指令级（指令间）

指令内部级

**粗粒度：软件实现**

**细粒度：硬件实现**



## 1. 并行性概念

(3) 并行性等级从低到高可分为：

- ① **字串位串**：同时只对一个字的一位进行处理，这是最基本的串行处理方式，不存在并行性；
- ② **字串位并**：同时对一个字的全部位进行处理，不同字之间是串行的，已开始出现并行性；
- ③ **字并位串**：同时对多字的同一位进行处理，该方式有较高的并行性；
- ④ **全并行**：同时对多字的全部位进行处理，最高一级并行。



## 2. 提高并行性技术途径

提高并行性技术途径主要有4条：

- ◆**时间重叠**：即时间并行。多个处理过程在时间上相互错开，轮流重叠地使用**同一套硬件**设备的各个部分；
- ◆**资源重复**：即空间并行 通过**重复**设置硬件资源，大幅度提高计算机系统的性能；
- ◆**时间重叠 + 资源重复**：主流技术；
- ◆**资源共享**：用软件方法实现多个任务按一定时间顺序轮流使用同一套硬件设备。



### 3.单处理机系统中并行性

在发展高性能单处理机过程中，起着主导作用的是

**(1) 时间并行（流水线）技术。**实现时间并行的基础是“部件功能专用化”，即流水线分层。

IF	ID	EX	MEM	WB			
	IF	ID	EX	MEM	WB		
		IF	ID	EX	MEM	WB	
			IF	ID	EX	MEM	WB

**(2) 空间并行技术：**多体存储器和多操作部件

**(3) 资源共享：**即虚拟机的概念



## 4.多处理机系统中并行性

多处理机系统同样遵循时间重叠、资源重复、资源共享原理；各处理机间连接紧密程度与交叉作用能力的强弱为耦合度。

根据耦合度，多处理机系统分为：

- ◆紧耦合系统（直接耦合系统）：处理机之间物理连接具有相对较高，可以并行处理作业或任务；
- ◆松耦合系统（间接耦合系统）：通过通信线路（通道）实现处理机间互联；



## 4.多处理机系统中并行性

### (1) 异构型多处理机系统

许多主要功能如：数组运算、语言编译、数据库管理等交由**专用处理机**完成，处理机间耦合度逐渐加强。

### (2) 同构型多处理机系统

为了使并行处理的任务能在处理机之间随机地进行调度，就必须使各处理机具有**同等的功能**





## 5. 并行处理机体系结构类型

按指令流和数据流的不同组织方式，可分为4种类型：

**SISD** ( Single Instruction Single Data ) ： 单处理机；

**SIMD** ( Single Instruction Multiple Data ) ： 向量处理机；

**MISD** ( Multiple Instruction Single Data ) ： 从未实现过；

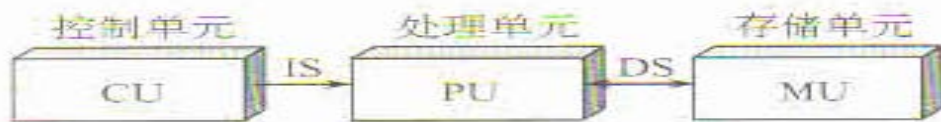
**MIMD** ( Multiple Instruction Multiple Data ) ： 代表机型是多处理机和机群系统。前者为紧耦合系统，后者为松耦合系统。





## 5. 并行处理机体系结构类型

### (1) SISD (Single Instruction Single Data)

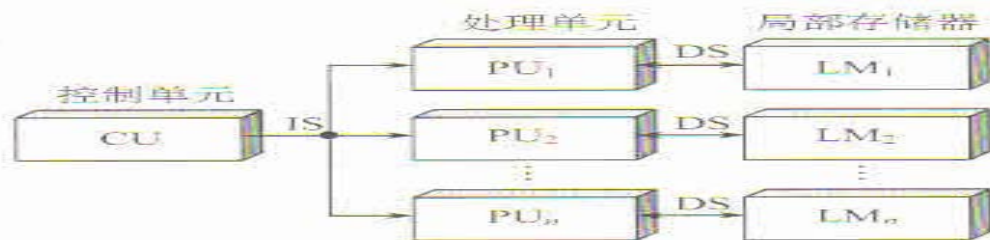


(a) SISD

**IS:** Instruction Stream

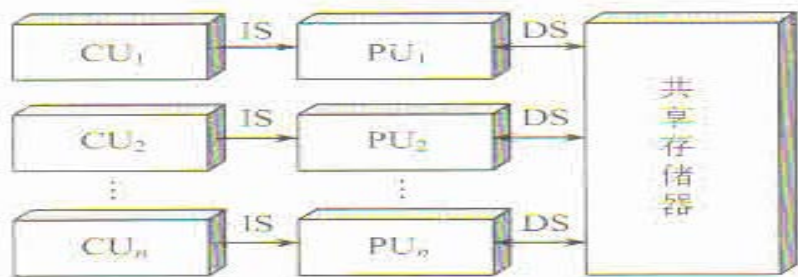
**DS:** Data Stream

### (2) SIMD (Single Instruction Multiple Data)

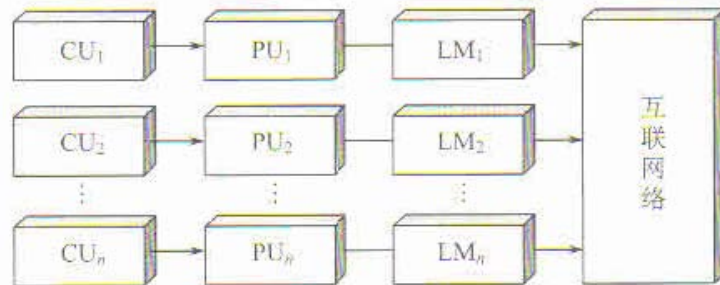


(b) SIMD(分布式存储器)

### (3) MIMD (Multiple Instruction Multiple Data)



(c) MIMD(共享存储器)



(d) MIMD(分布式存储器)



## 6. 并行处理机组织和结构

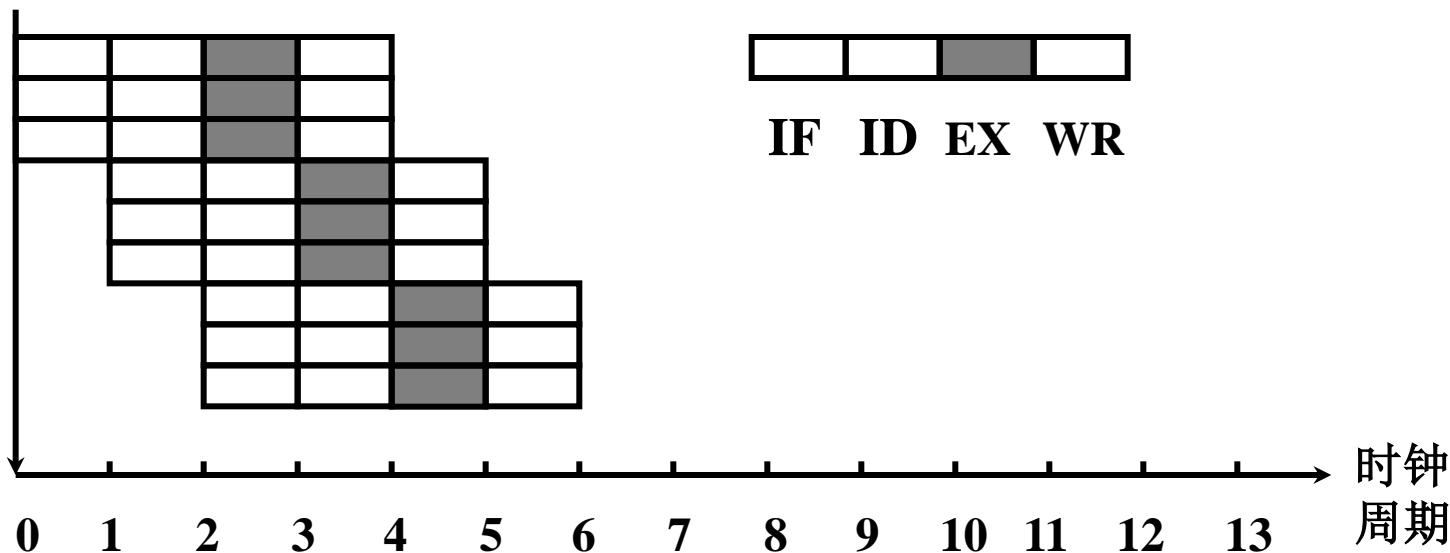
### (1) 超标量处理机

- 每个时钟周期内可 **并发多条独立指令**  
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，把可并行执行的指令搭配起来

指令序列

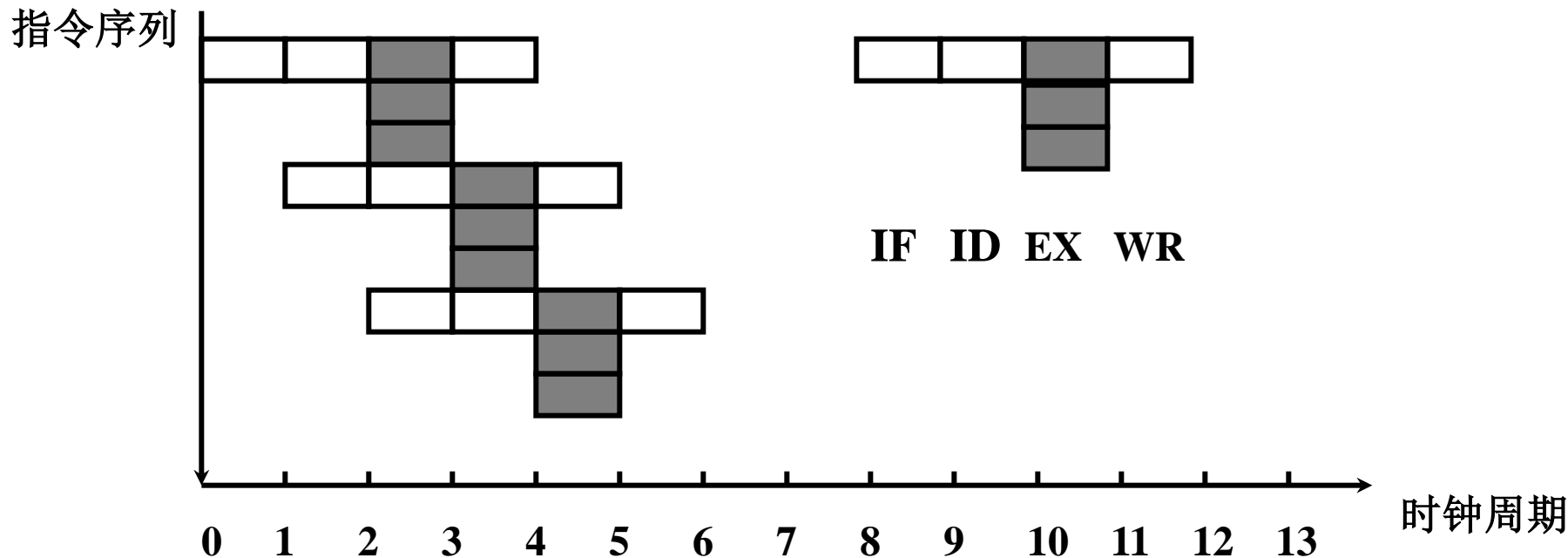




## 6. 并行处理机组织和结构

### (2) 超长指令字处理机

- 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**





## 6.并行处理机组织和结构

(3) 广义多处理机系统：多台计算机协同工作完成要求的工作

(4) 狭义多处理机系统：一台计算机内处理机之间通过共享存储器方式通信的并行计算机系统；

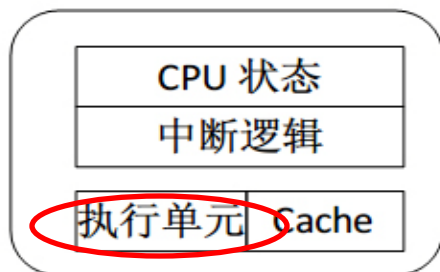
(5) 多计算机系统：不共享内存的多处理机系统构成的并行系统

(6) 多线程处理机：thread是操作系统能够进行运算调度的最小单位，是进程中的实际运作单位。一条线程指的是进程中一个单一顺序的控制流，一个进程中可以并发多个线程，每条线程并行执行不同的任务；

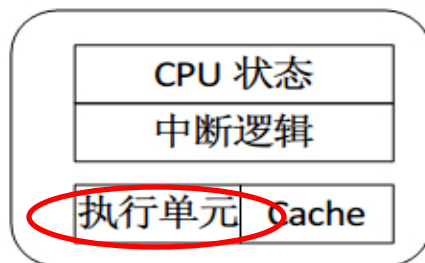
(7) 多核处理机：处理机芯片内集成两个或两个以上的完整且并行工作的计算引擎（核）



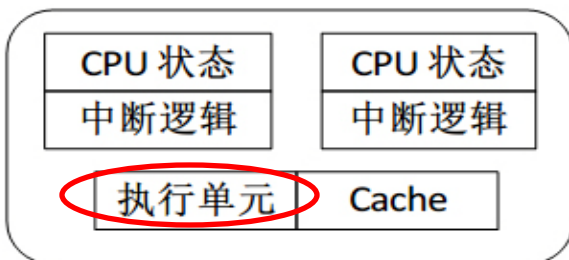
## 6. 并行处理机组织和结构



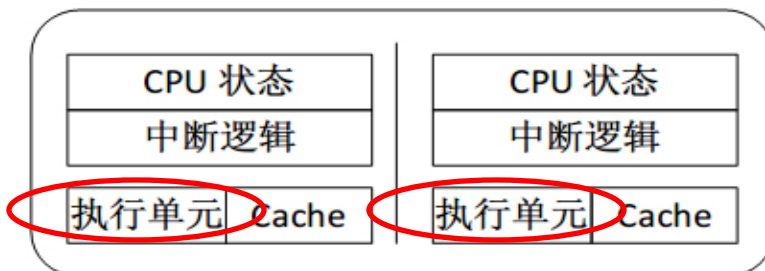
(a) 单核处理机结构



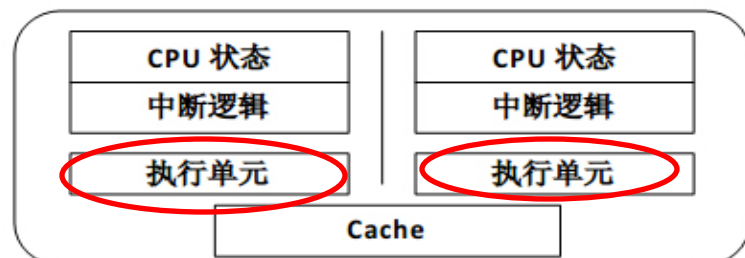
(b) 多处理机结构



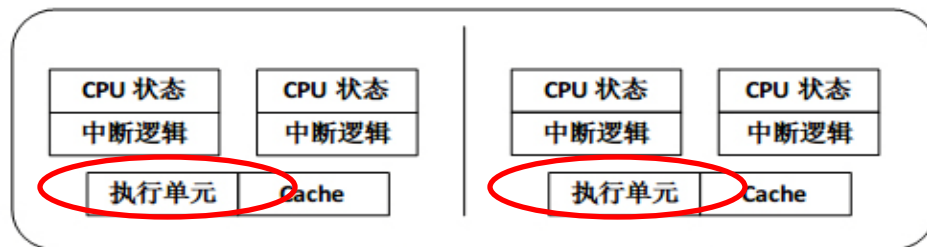
(c) 多线程处理机结构



(d) 多核处理机结构



(e) 共享 Cache 的多核处理机结构



(f) 支持多线程的多核处理机结构





## 1. 相关概念

◆ **程序 (Program)** : 静态的实体

◆ **进程 (Process)** : 执行中的程序, 动态实体; 不仅包含代码, 也包含状态和资源

◆ **传统OS**: 进程作为基本单位, **OS**将资源分配给每个进程;

◆ **现代OS**: 线程 (**Thread**) 是能独立运行的基本单位, 进程是资源分配单位, 每个进程可以拥有若干线程;

◆ **硬件线程 (Hardware Thread)** : 也称逻辑内核或逻辑处理器, **OS**每一个硬件线程识别为一个可调度的逻辑处理器, 可以运行软件线程的代码, **OS**调度器可以决定将一个软件线程赋给一个硬件线程, 通过这种方式均衡每一个硬件线程的工作负载, 以达到并行优化的作用。



## 2.超线程技术

◆超线程技术（Hyper Threading, HT）把多线程处理器内部的两个逻辑内核模拟成两个物理芯片，让单个处理器就能使用线程级的并行计算，进而兼容多线程操作系统和软件。超线程技术充分利用空闲CPU资源，在相同时间内完成更多工作。

◆超标量技术和超长指令字技术都是针对单一的指令流中的指令来提高并行处理能力的。当单一指令流出现cache缺失时，指令流就会断流；指令间的相关性也会严重影响执行单元的利用率。

标量机：是指一个数一个数地进行计算的加工处理；

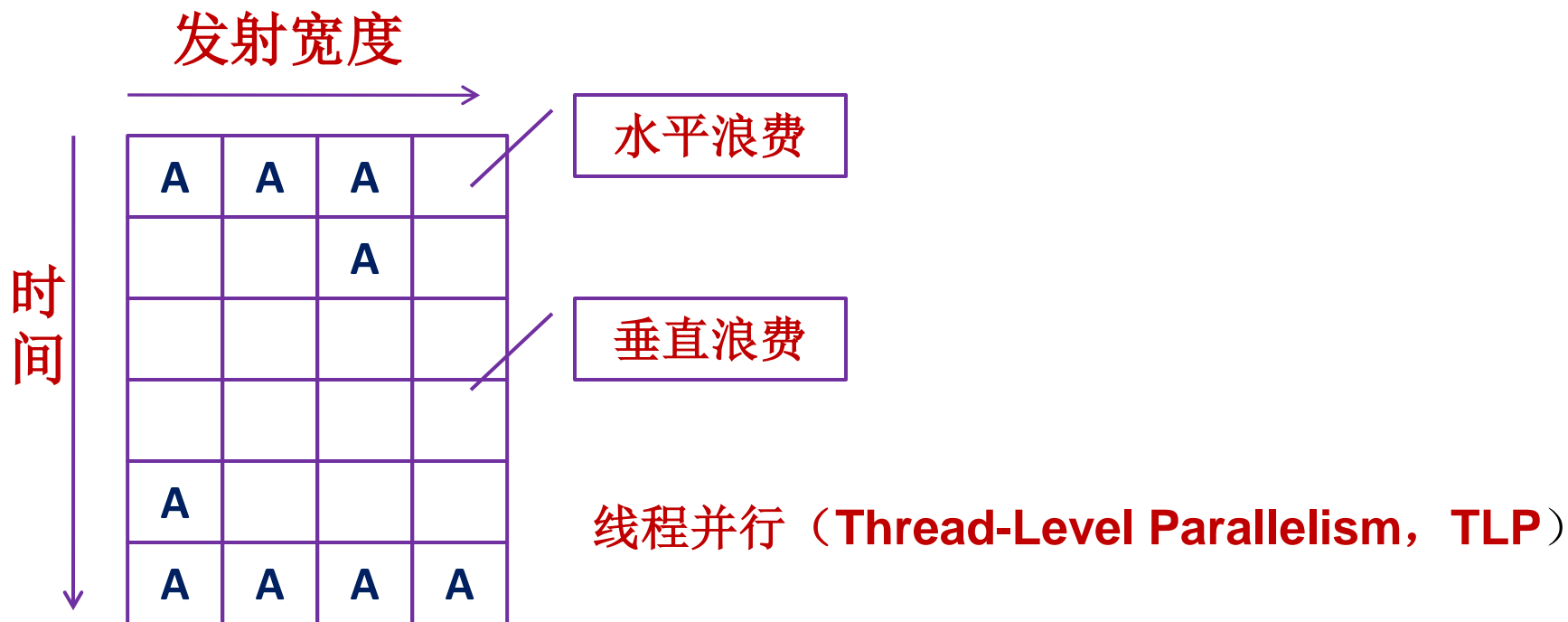
向量机：能够对一批数据同时进行加工处理。





### 3. 超标量中的相关问题

- ◆ **垂直浪费**：资源相关导致流水线不能执行新的指令
- ◆ **水平浪费**：控制相关导致流水线中部分流水线被闲置

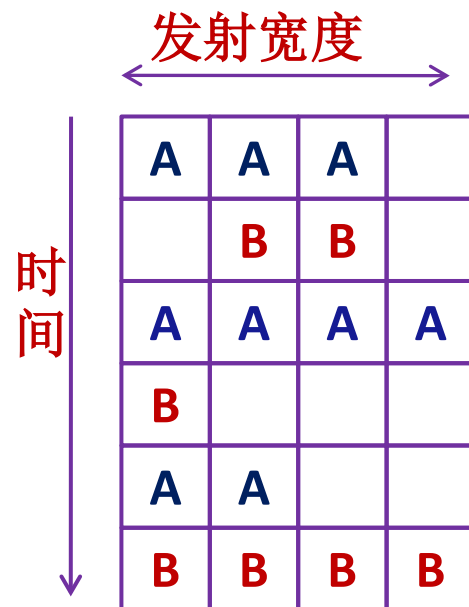




## 4. 硬件线程 (Hardware Thread)

### ◆ 细粒度多线程 (交错多线程) 处理机:

交替执行线程指令，每个T周期都进行线程切换；由于多线程交替执行，处于阻塞线程在切换时被跳过，在一定程度上降低了指令阻塞造成的处理机吞吐率损失。

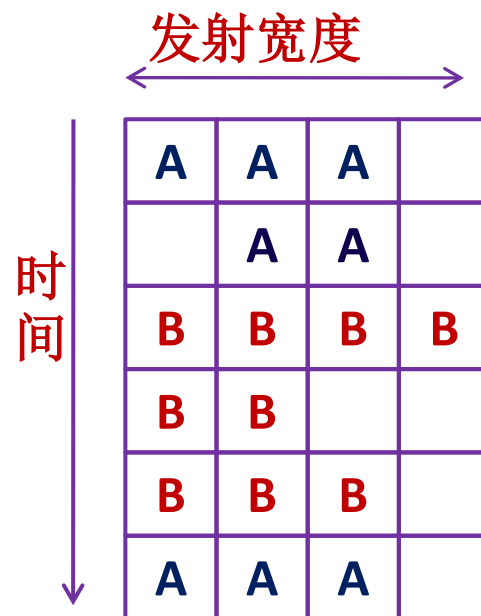




## 4. 硬件线程 (Hardware Thread)

### ◆ 粗粒度多线程 (阻塞多线程) 处理机:

只有遇到高代价的长延迟操作时才有处理机硬件进行线程切换, 否则一直执行同一线程指令; 比细粒度多线程有更低的切换开销。

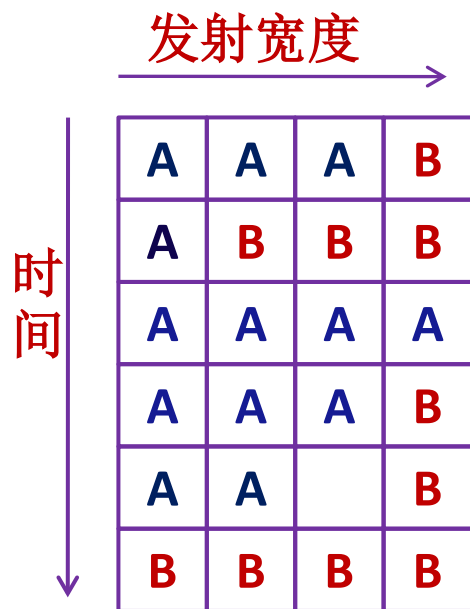




## 5.SMT: Simultaneous MultiThreading

◆同时多线程（**SMT**）结合了超标量技术和细粒度多线程技术的优点，允许在**同一**时钟周期内发射来自不同线程的多条指令。

◆**SMT**能够利用**线程级并行减少垂直浪费**，又能在一个**T**内利用**线程并行和指令并行减少水平浪费**。



**缺点：**需要设置大量**REG**保存现场信息；需要保证并发执行多线程带来的Cache冲突。

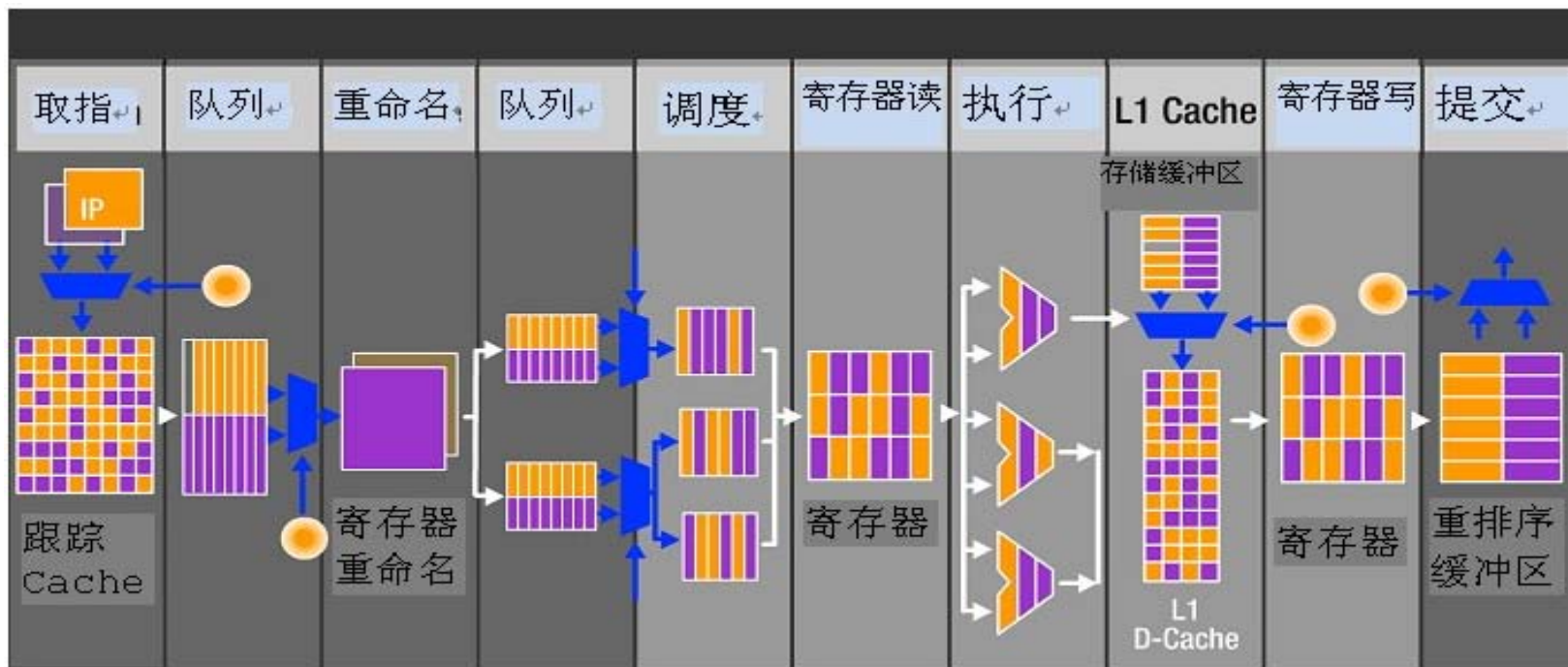


## 6.超线程机构的主要技术

NetBurst微体系结构的流水结构：

预测损失、高频率执行、缩减指令量

指令执行都要经过**10个**功能流水线。

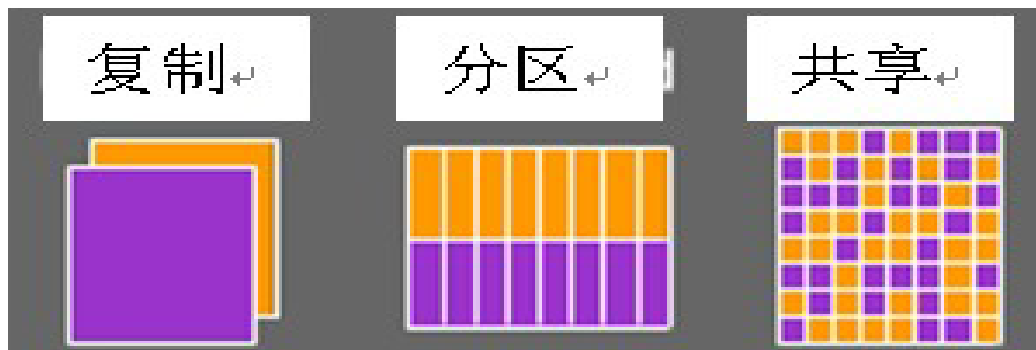




## 6.超线程机构的主要技术

为了支持两个硬件线程同时运行，让每级流水线中的资源通过三种方式复用于两个线程：**复制、分区、共享**。

- 复制**：在处理器设计时分别为两个线程**设置独立的部件**。被复制的资源包括状态、**IP**、**REG**命名部件等
- 分区**：将单线程独立资源分割成两部分。方式：**缓冲区和队列**
- 共享**：在指令执行过程中使**资源根据需要在两个线程之间共享**





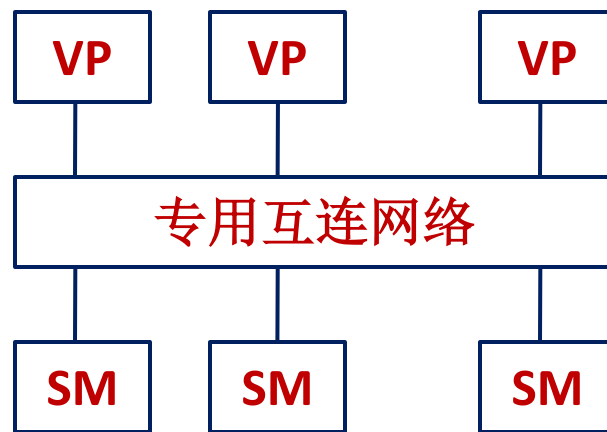
### 1. 多处理机系统的分类

多处理机系统由多个独立的处理机组成，每个处理机能够独立执行自己的程序：

- ◆ 并行向量处理机（**PVP**）：Parallel Vector Processor
- ◆ 对称多处理机（**SMP**）：Symmetrical Multi-Processing
- ◆ 大规模并行处理机（**MPP**）：Massively Parallel Processor
- ◆ 分布共享存储器多处理机（**DSM**）：Distributed Shared Memory

（1）并行向量处理机（**PVP**）：几台巨型向量处理机采用共享存储器模式互连而成，处理机数目较少。

紧耦合





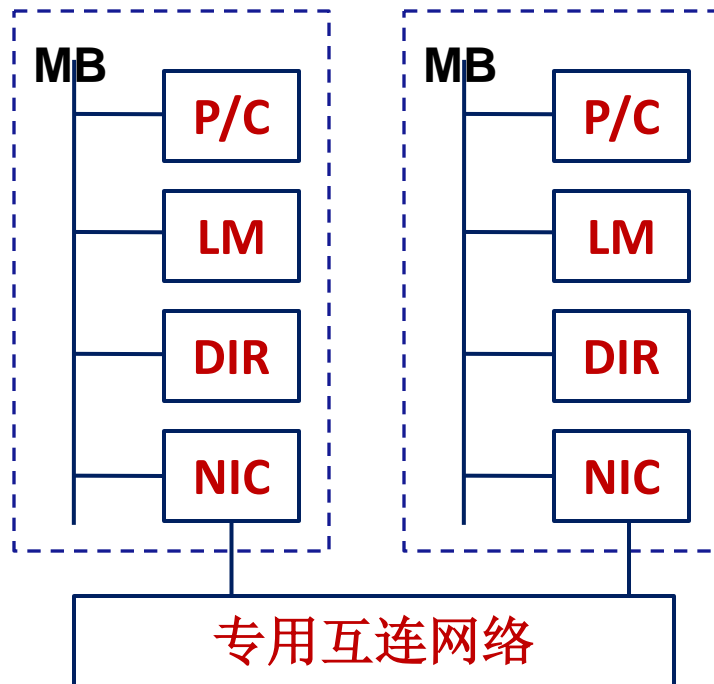
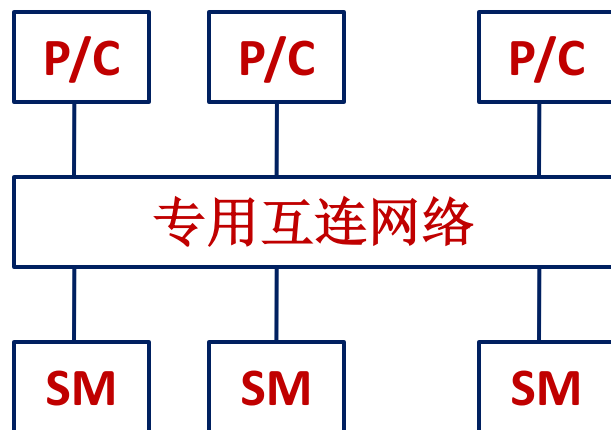
### 1. 多处理机系统的分类

(2) 对称多处理机 (SMP) :  
多个处理器和存储器互连, 处理机对称且功能相同; 存储器访问延迟相同。

紧耦合

(3) 分布共享存储器多处理机 (DSM) :  
有本地存储器但统一编址。逻辑上组成一个共享存储器。

紧耦合





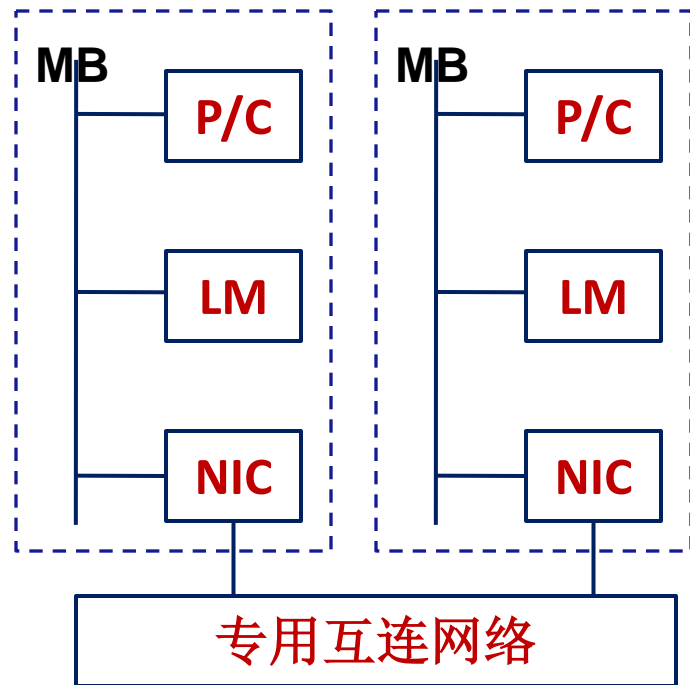


### 1. 多处理机系统的分类

#### (4) 大规模并行处理机 (MPP)

每个结点有一台处理机，有局部存储器、NIC和设备；传送数据速度低，延迟长

松耦合





## 2.SMP的基本概念

**SMP**定义为具有如下特征的独立计算机系统：

- ◆有两个以上功能相似的处理机；
- ◆这些处理机**共享同一主存和I/O设施**，以总线或其他内部连接机制互连在一起；这样，存储器存取时间对每个处理机都是大致相同的；
- ◆所有处理机共享对I/O设备的访问，或通过**同一通道**，或通过提供到同一设备**路径的不同通道**；
- ◆所有处理机能完成同样的功能；
- ◆系统被一个**集中式操作系统（OS）**控制。



## 2.SMP的基本概念

**SMP的OS能跨越所有处理机来调度进程或线程，其优点：**

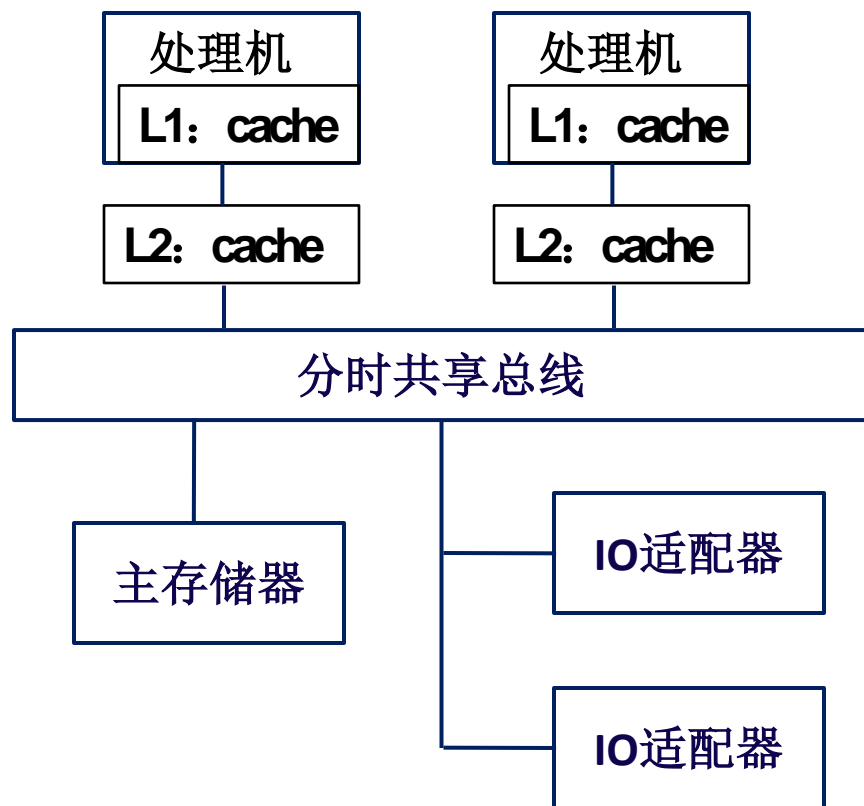
- ◆**性能：**多个处理机的系统具有同类型单处理机功能；
- ◆**可用性：**所有处理机具有相同功能，单个故障不会停机；
- ◆**增量式增长：**可以在系统中**添加**处理机来提高性能；
- ◆**可扩展性：**系统因配置不同数目的处理机有不同的性价比；
- ◆**透明性：**多个处理机对用户是透明的



### 3.SMP结构

为便利来自I/O处理器的DMA传送，应提供如下特征：

- ◆**寻址**：能区分总线上模块以确定数据的来源和目标；
  - ◆**仲裁**：能对总线控制的竞争请求进行仲裁；
  - ◆**分时共享**：一个模块使用总线，其他模块被锁住；
- 一般来说，工作站和个人机**SMP**都有两级**Cache**；现在，某些处理机还使用了**L3 Cache**。





## 1.多核处理机的优势

多核处理机的**3高3低**:

- ① **高并行性**: 可同时执行的**线程数或任务数**是单处理机的数倍;
- ② **高通信效率**: 各个处理机核只需要在核内部的相对较小的区域内交换数据;
- ③ **高资源利用率**: 多核结构可有效支持**片内**资源共享;
- ④ **低功耗**: 不再依靠提高主频改善性能;
- ⑤ **低设计复杂度**: 每个核的结构相对简单;
- ⑥ **较低的成本**: 各个核共享器件芯片封装和芯片**I/O**资源。

**多核技术**是在**超线程、超标量、多处理器**等技术基础上发展起来的



### 2. 多核处理机的组织结构

(1) 同构多核处理机与异构多核处理机:

◆ **同构多核 (homogenous multi-core)** 处理机所有计算内核结构相同，地位对等;

◆ **异构多核 (heterogeneous multi-core)** 处理机各个计算内核结构不同，地位不对等。一般多采用“**主处理核+协处理核**”的主从架构。



## 2.多核处理机的组织结构

### (2) 多核处理机的对称性

#### ◆对称(Symmetric MultiProcessing, SMP)多核:

处理机片内包含**相同结构**的核，多个核紧密耦合，并运行统一的操作系统；

#### ◆非对称(Asymmetric MulitiProcessing, AMP)多核:

若是同构多核，但每个核运行一个独立的操作系统或同一操作系统独立实例，就变成非对称多核。

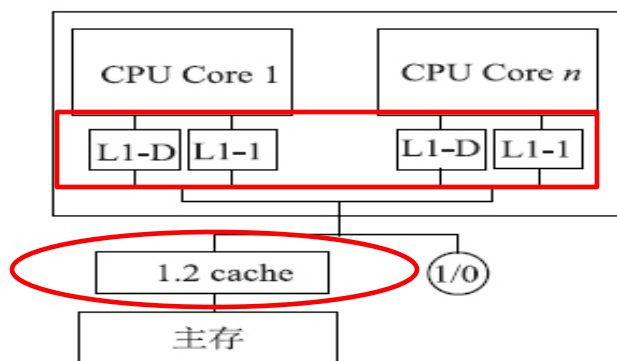
**AMP**多核系统也可采用**异构多核**和共享存储器构成。



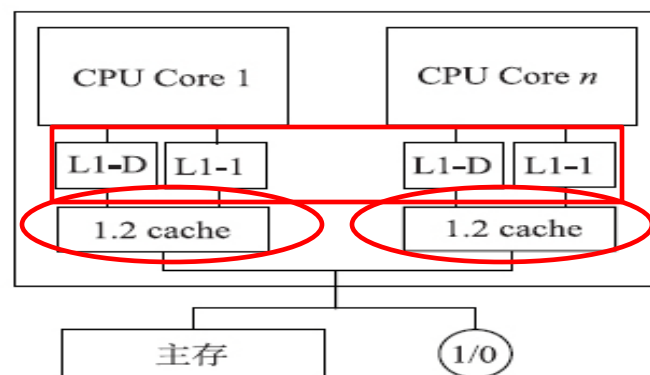
### 2. 多核处理机的组织结构

#### (3) 多核处理机的cache组织

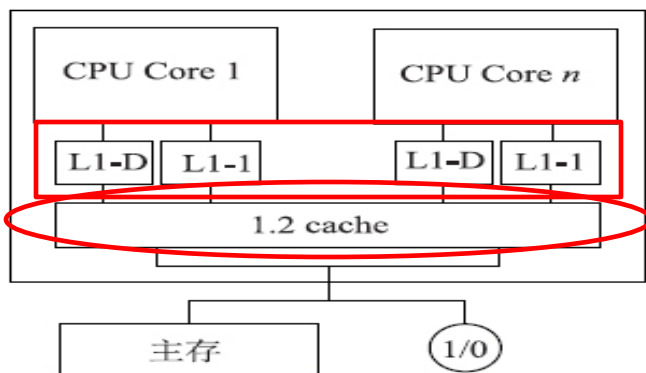
采用两级cache结构，其中L1分为指令和数据两路缓存；



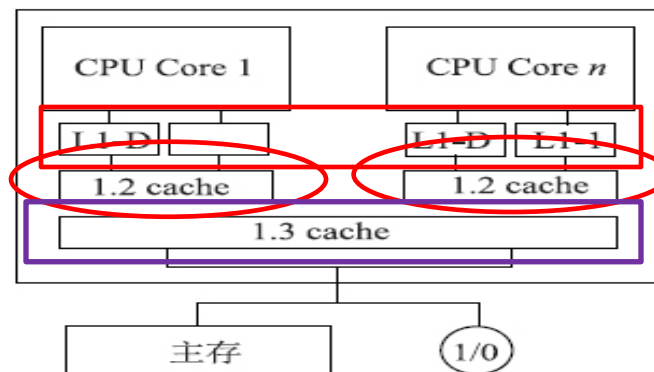
(a) 片内私有L1 cache



(b) 片内私有L2 cache



(c) 片内共享L2 cache



(d) 片内共享L3 cache





## 2.多核处理机的组织结构

### (4) 共享**cache**优缺点

- 共享**Cache**有助于提高**Cache**的命中率;
  - 共享**Cache**可以在不同核之间动态分配**Cache**空间;
  - 共享**Cache**可以作为处理机之间交互信息的通道;
- ◆多核处理机必须解决多级**Cache**的一致性问题, 共享多级**Cache**会增加解决一致性的难度。



### 3.多核处理机关键技术

#### (1) 多核处理机架构

- ◆每个核自身的结构
- ◆每个核是否应该采用超标量技术或超线程技术
- ◆多核之间的对等性，核的数目
- ◆同构多核：阿姆达尔定律（**Amdahl's law**）
- ◆异构多核：集成图形处理机（**GPU**）

#### (2) 多核系统存储结构设计

- ◆采用二级甚至三级**cache**提高等效访问速度



### 3.多核处理机关键技术

#### (3) 多核处理机的**cache**一致性

多核**cache**内容不一致的原因：

- ◆可写数据的共享：单核的全写法和回写法；
- ◆I/O活动：**IO**处理机直接接在系统总线上；
- ◆核间线程迁移：把一个没完成的线程调度到另外一个空闲核上；

**解决方法：**禁止**IO**通道与处理机共享、禁止核间线程迁移

**cache**一致性维护：

- 软件方法：**预防（分析逻辑结构和数据相关性）
- 硬件方法：**发现和解决（目录协议、监听协议）



### 3.多核处理机关键技术

#### 目录协议：

- ◆收集并维护有关数据块副本驻存在何处的信息；
- ◆任何一个能影响**Cache**全局状态的局部动作必须报告给**中央控制器**；
- ◆处理机核向局部**Cache**行副本写入信息前，必须向中央控制器请求**排他性**访问权限

#### 监听协议：

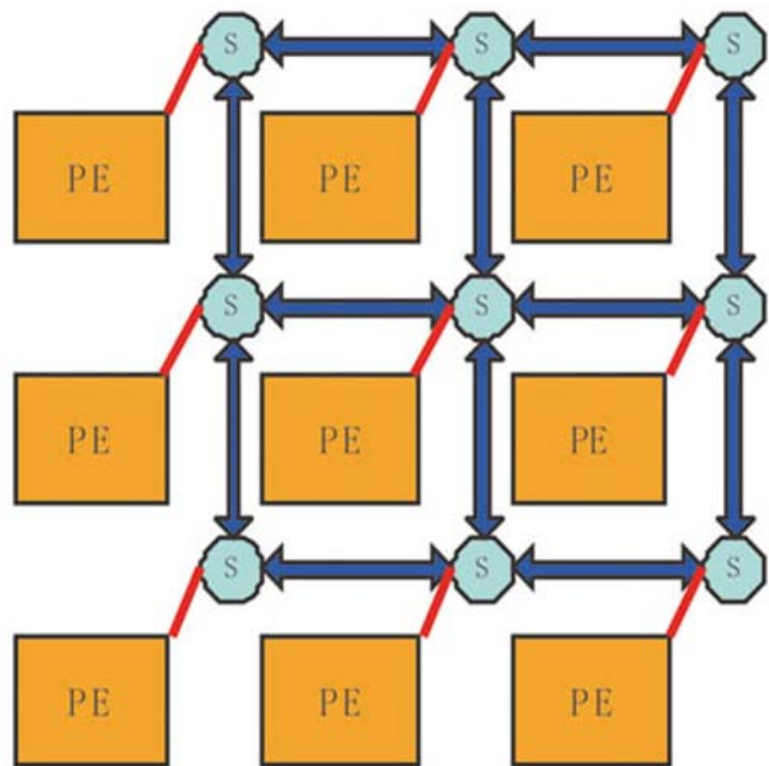
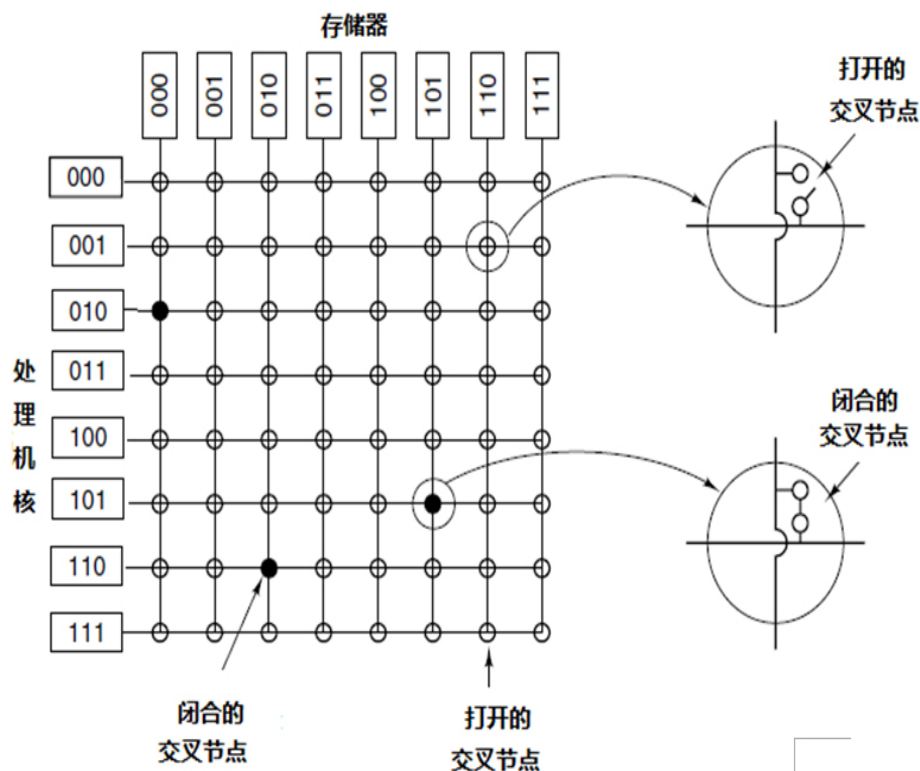
- ◆将维护**cache**一致性的责任**分布**到每个**Cache**控制器上；
- ◆对共享的**cache**进行修改时，广播机制通知所有**cache**；
- ◆两种协议：**写-作废（Write-invalidate）**和**写-更新（Write-update）**

写-作废：修改**Modified**、独占**Exclusive**、共享**Shared**、无效**Invalid**



### 4.多核处理机的核间通信与同步技术

主要通信方式有三种：**总线共享cache结构**、**交叉开关互连结构**、**片上网络结构**





## 5.低功耗设计

低功耗设计是一个多层次的问题，需要同时在**操作系统级、算法级、结构级、电路级**等多个层次上综合考虑

- ◆单处理机：降低半导体电路的动态和静态电能消耗；
- ◆多核处理机：异构结构设计、**动态线程分派与转移技术**等。

## 6.多核软件设计

多核技术与多处理机的差别：

- 多处理机间界限清晰，处理机独立运行；
- 多核系统中，资源的共享更加普遍。



## 6. 多核软件设计

支持多核的操作系统必须解决任务分配、任务调度、仲裁、负载平衡、核间动态任务迁移等问题

- 调度算法：全局队列调度、局部队列调度
- 应用软件设计：并行编程、并行设计思想
- 平衡设计原则：权衡芯片复杂度、内部结构、性能、功耗、扩展性、部件成本等