

尚德机构

计算机系统结构

讲师：孙小涵

学习是一种信仰！ IN LEARNING WE TRUST

SUNLAND



讲师介绍

- 主讲老师：孙小涵（尚德机构-小涵老师）
- 主讲课程：计算机类、数学类
- 邮箱：sunxiaohan@sunlands.com



课程章节

计算机系统结构

第1章 计算机系统结构概论

第2章 数据表示、寻址方式与指令系统

第3章 存储、中断、总线与I/O系统

第4章 存储体系

第5章 标量处理机

第6章 向量处理机

第7章 多处理机

第8章 数据流计算机和归约机

第5章 标量处理机

第5章 标量处理机



5.2.3标量流水机的相关处理和控制机构

1.局部性相关的处理(填空)

重叠机器处理这些局部性相关的方法有两种。一种是**推后后续指令对相关单元的读**，直至在先的指令写入完成；另一种是设置**相关直接通路**，将运算结果经相关直接通路直接送入所需部件。

5.2.3标量流水机的相关处理和控制机构

1.局部性相关的处理（单选、填空）

任务在流水线中流动顺序的安排和控制可以有两种方式。一种是让任务（指令）流出流水线的顺序保持与流入流水线的顺序一致，称为**顺序流动方式**或**同步流动方式**；另一种是让流出流水线的任务（指令）顺序可以和流入流水线的顺序不同，称为**异步流动方式**。

5.2.3标量流水机的相关处理和控制机构

2.全局性相关的处理（简单了解单选）

全局性相关指的是已进入流水线的**转移指令**（尤其是条件转移指令）和其后续指令之间相关。

- (1) 使用猜测法
- (2) 加快和提前形成条件码
- (3) 采取延迟转移
- (4) 加快短循环程序的处理

5.2.3标量流水机的相关处理和控制机构

3.流水机器的中断处理（简单了解填空）

中断会引起流水线断流，但出现概率比条件转移的概率要低得多，且又是随机发生的。所以，流水机器中断主要是如何处理好断点现场的保存和恢复，而不是如何缩短流水线的断流时间。

5.2.3标量流水机的相关处理和控制机构

4.非线性流水线的调度

【1904真题】现设流水线由5段组成,段号 k 分别为1~5,任务经过流水线总共需要9拍,其预约表如表所示。

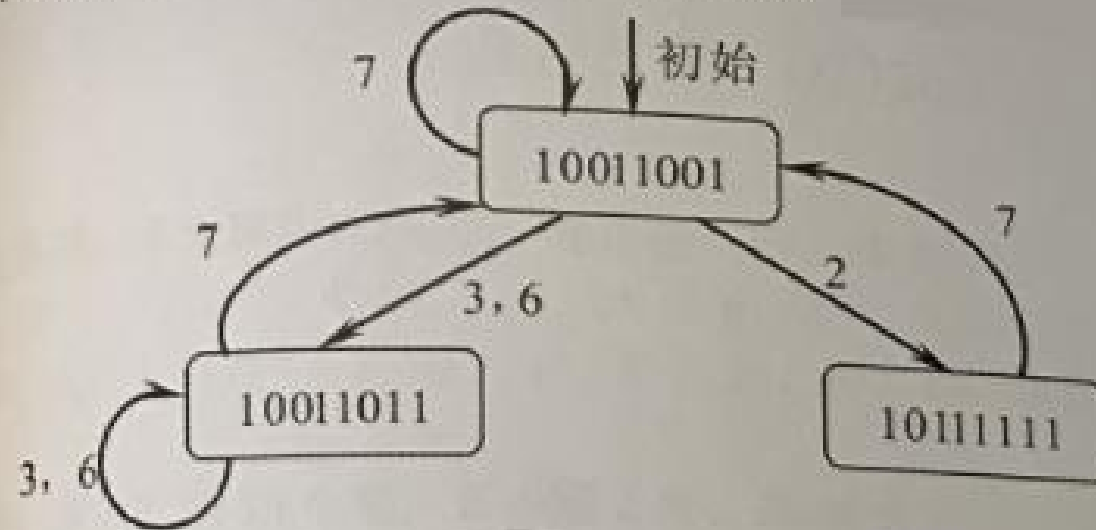
- (1)写出延迟禁止表 F 、冲突向量 C ;
- (2)画出流水线状态转移图;
- (3)求出最佳调度方案、最小平均延迟及流水线的最大吞吐率。

拍号 n 段号 k	1	2	3	4	5	6	7	8	9
1	✓								✓
2		✓	✓				✓		
3				✓					
4					✓	✓			
5							✓	✓	

5.2.3 标量流水机的相关处理和控制机构

4. 非线性流水线的调度

- (1) 延迟禁止表 $F = \{1, 4, 5, 8\}$
冲突向量 $C = (10011001)$
(2) 流水线状态转移图如答题图所示。



答图

- (3) 从状态转移图可看出, 最佳调度方案为 $(2, 7)$
则, 最小平均延迟为 4.5 拍
最大吞吐率为: $T_{p_{max}} = 1/4.5$ (任务/拍)

5.3指令级高度并行的超级处理机

本节主要内容：

超标量 (Superscalar)处理机、超长指令字 (VLIW)处理机、超流水线 (Superpipelining)处理机和超标量超流水线处理机。

5.3.1 超标量处理机（一定要会画）

假设一条指令包含取指令、译码、执行、存结果4个子过程，每个子过程经过时间为 Δt 。常规的标量流水线单处理机是在每个期间解释完一条指令，如图5-15所示。执行完12条指令共需15 Δt 。称这种流水机的度 $m=1$ 。

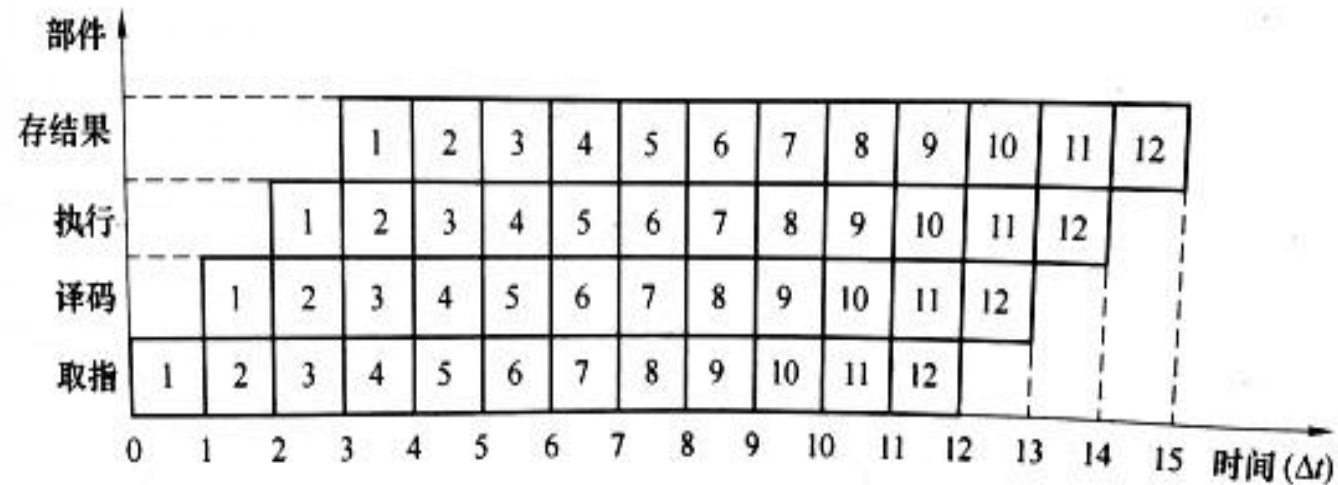


图 5-15 常规（度 $m=1$ ）的标量流水时空图

5.3.1 超标量处理机（一定要会画）

超标量处理机采用多指令流水线，每个同时流出 m 条指令（称为度 m ）。

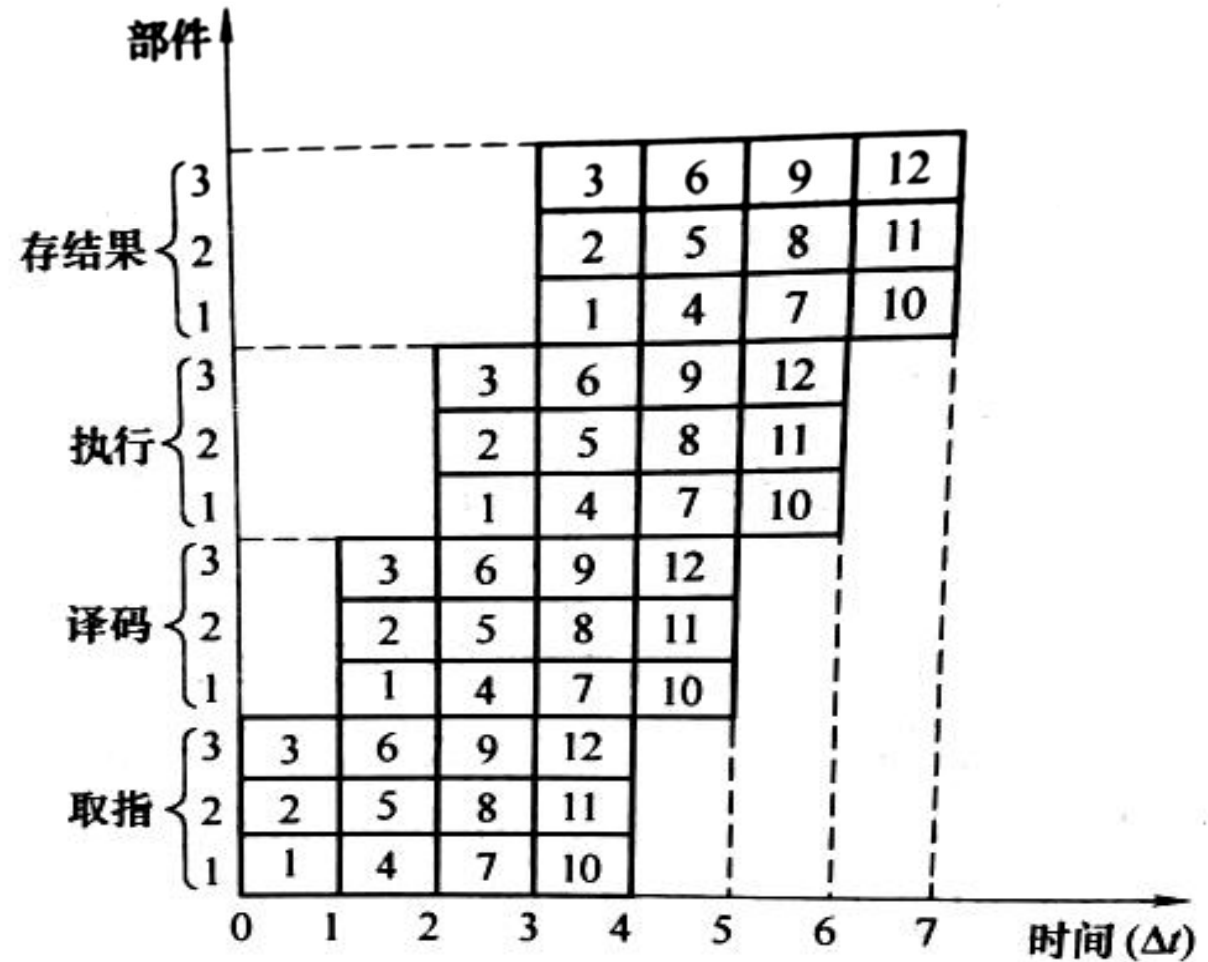


图5-16 度 $m=3$ 的超标量处理机时空图



那么意气风发地
走在成功的道路上

真题练练手

- 1、流水处理机“先写后读”相关的解决方法包括推后（ ）的读和设置（ ）。1910



那么意气风发地
走在成功的道路上

真题练练手

1、流水处理机“先写后读”相关的解决方法包括推后（ ）的读和设置（ ）。1910

答案：后续指令对相关单元 相关直接通路



那么意气风发地
走在成功的道路上

真题练练手

4、在流水机器中，全局性相关是指（） 1304

A:指令相关

B:先读后写相关

C:先写后读相关

D:由转移指令引起的相关



那么意气风发地
走在成功的道路上

真题练练手

4、在流水机器中，全局性相关是指（） 1304

A:指令相关

B:先读后写相关

C:先写后读相关

D:由转移指令引起的相关

答案：



那么意气风发地
走在成功的道路上

真题练练手

5、任务在流水线中流动顺序的安排和控制有（ ）方式和（ ）方式。

1804



那么意气风发地
走在成功的道路上

真题练练手

5、任务在流水线中流动顺序的安排和控制有（ ）方式和（ ）方式。

1804

答案：顺序流动 异步流动

第6章 向量处理机

第6章 向量处理机

●	向量的流水处理与向量流水处理机	★
●	阵列处理机的原理	★
●	SIMD计算机的互连网络	★ ☆
●	共享主存构形的阵列处理机中并行存储器的无冲突访问	★ ☆
●	脉动阵列流水处理机	★

6. 1向量的流水处理与向量流水处理机

本节主要内容：

向量处理的3种方式

向量的流水处理方式

向量指令集安并行、链接、串行的条件

分析向量指令间的情况，计算最少时钟数

6.1.1 向量的处理和向量的流水处理（填空）

【例】计算 $D = A \times (B + C)$ ，其中A、B、C、D都是有N个元素的向量，应该采用什么方式处理才能充分发挥流水线的效能？



6.1.1 向量的处理和向量的流水处理（填空）

【例】计算 $D = A \times (B + C)$ ，其中A、B、C、D都是有N个元素的向量，应该采用什么方式处理才能充分发挥流水线的效能？

如果采用逐个求D向量元素的方法，即访存取 a_i 、 b_i 、 c_i 元素求 d_i ，再取 a_{i+1} 、 b_{i+1} 、 c_{i+1} ，求 d_{i+1} ，这种处理方式称为**横向（水平）**处理方式。

这时只有采用对整个向量按相同操作都执行完之后再转去执行别的操作，才能较好地发挥流水处理的效能。

结论：向量横向处理是向量的处理方式，但不是向量的流水处理方式；而**向量纵向处理**和**分组纵横处理**既是向量的处理方式，也是向量的流水处理方式。

6.1.3通过并行、链接提高性能

【1510真题】求向量 $D=A \times (B+C)$ ，向量为浮点数，各向量元素个数均为 N ，参照CRAY—1方式分解为3条向量指令：

① $V3 \leftarrow$ 存储器；访存取 A 送入 $V3$ 寄存器组

② $V2 \leftarrow V0+V1$ ； $B+C \rightarrow K$

③ $V4 \leftarrow V2 \times V3$ ； $K \times A \rightarrow D$ 当采用下列3种方式工作时，各需多少拍才能得到全部结果？

(1)①、②和③串行执行。

(2)①和②并行执行后，再执行③。

(3)采用链接技术。

6.1.3通过并行、链接提高性能

在该向量运算中用到浮点加和浮点乘运算,CRAY-1 计算浮点加需要 6 拍,计算浮点乘需要 7 拍,数据存入寄存器需要 1 拍。

(1) ①、②和③串行执行所需要的时间为

$$7 + N + 7 + N + 8 + N = 22 + 3N(\text{拍})$$

(2) ①和②并行执行后,再执行③所需要的时间为

$$\begin{Bmatrix} 7 + N \\ 7 + N \end{Bmatrix} + 8 + N = 15 + 2N(\text{拍})$$

(3) 采用链接技术所需要的时间为

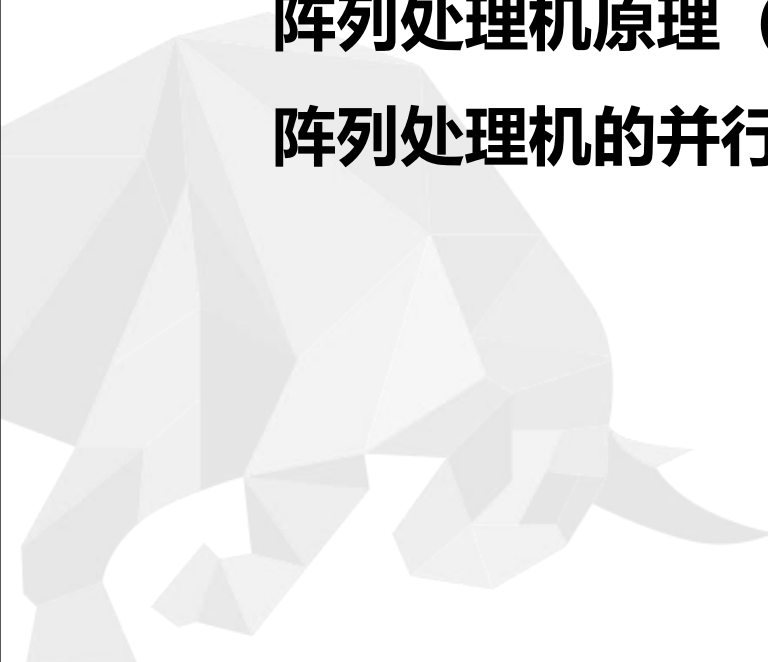
$$\begin{Bmatrix} 1 + 6 + 1 \\ 1 + 6 + 1 \end{Bmatrix} + 8 + N = 16 + N(\text{拍})$$

6.2阵列处理机的原理

本节主要内容：

阵列处理机原理（识记）

阵列处理机的并行算法（领会）



6.2. 1阵列处理机的构形和特点

1.阵列处理机的构形

阵列处理机有两种构形，差别主要在于存储器的组成方式和互连网络的作用不同。

构形1图6-4所示是采用**分布式**存储器阵列处理机的构形。

为了高速有效地处理向量数据，这种构形要求能把数据合理地预分配到各个处理单元的局部存储器中，使各处理单元 PE_i 主要用自己的 PEM_i 中的数据运算。

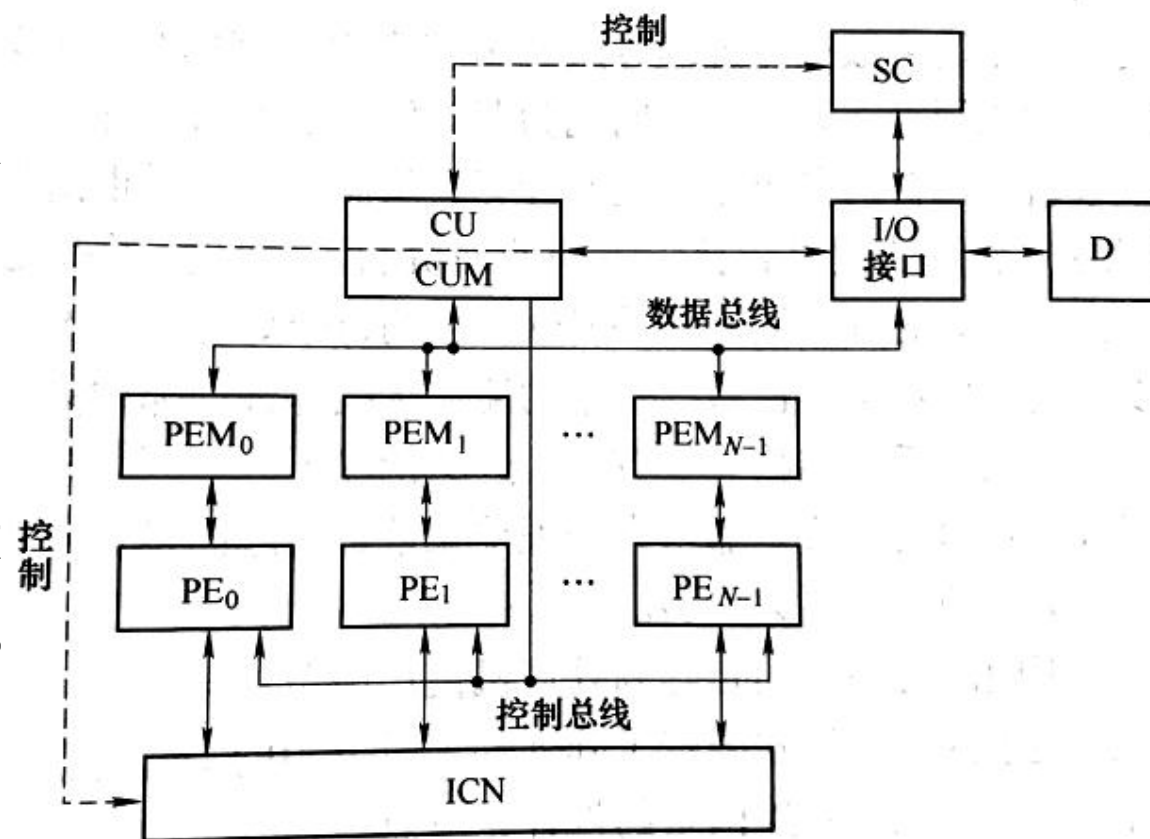
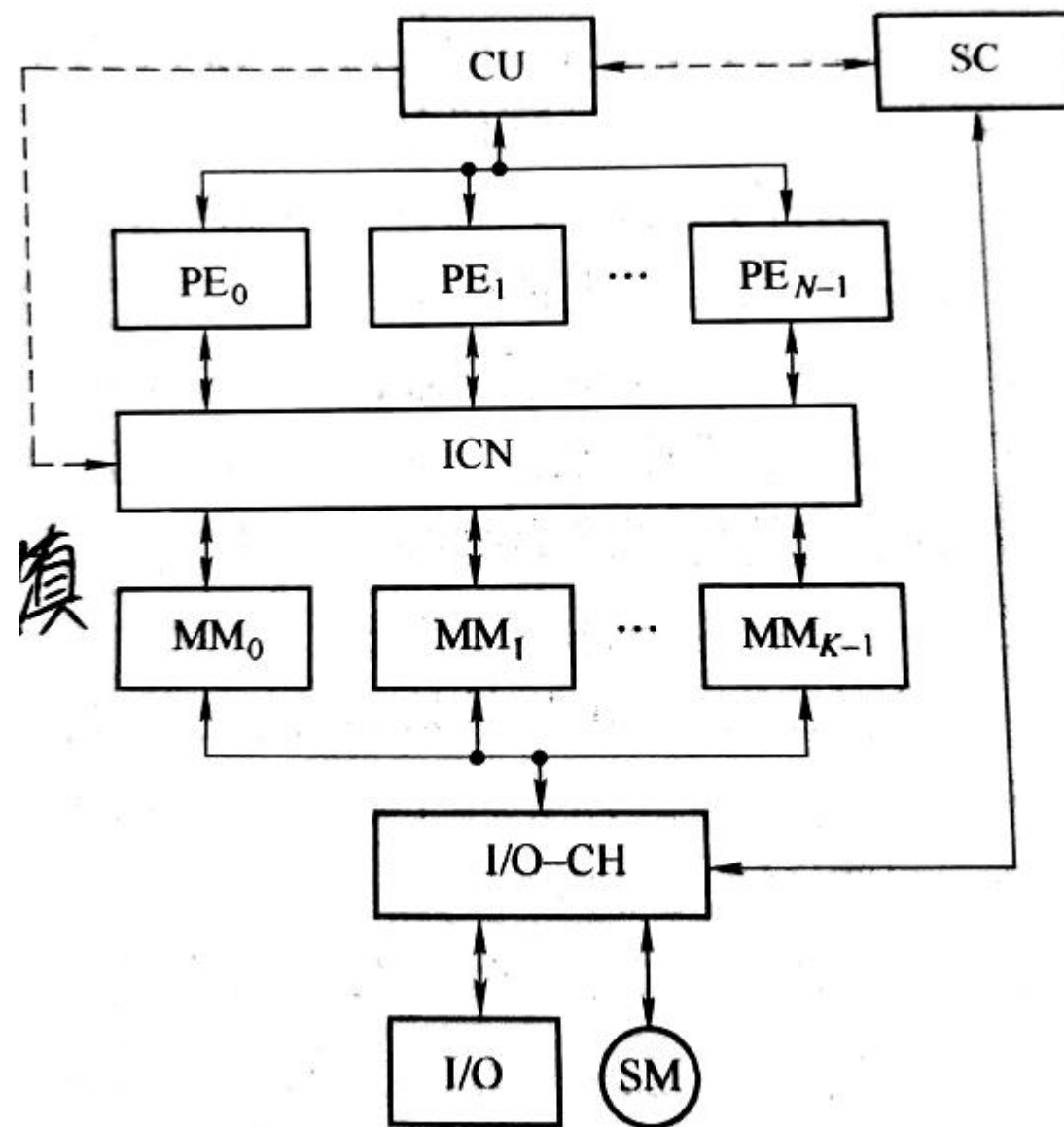


图 6-4 具有分布式存储器的阵列处理机构形

6.2. 1阵列处理机的构形和特点

1.阵列处理机的构形

构形2 是采用**集中式**共享存储器的阵列处理机构形。



6.2. 1 阵列处理机的构形和特点（单选、填空）

阵列处理机按存储器的组织方式不同，分为两种不同的基本构形，一种是采用**分布式**存储器的阵列处理机构形，另一种是具有**集中式**共享存储器的阵列处理机构形。

与同样擅长于向量处理的流水线处理机相比，阵列处理机利用的是**资源重复**，而不是时间重叠。利用的是并行性中的**同时性**，而不是并发性。

6.2.2 ILLIAC IV的处理单元阵列结构（简单了解）

采用这种构形的阵列处理机是

SIMD的主流。

典型的机器有ILLIAC IV、MPP、
DAP、CM-2、MP-1、DAP600
系列等。

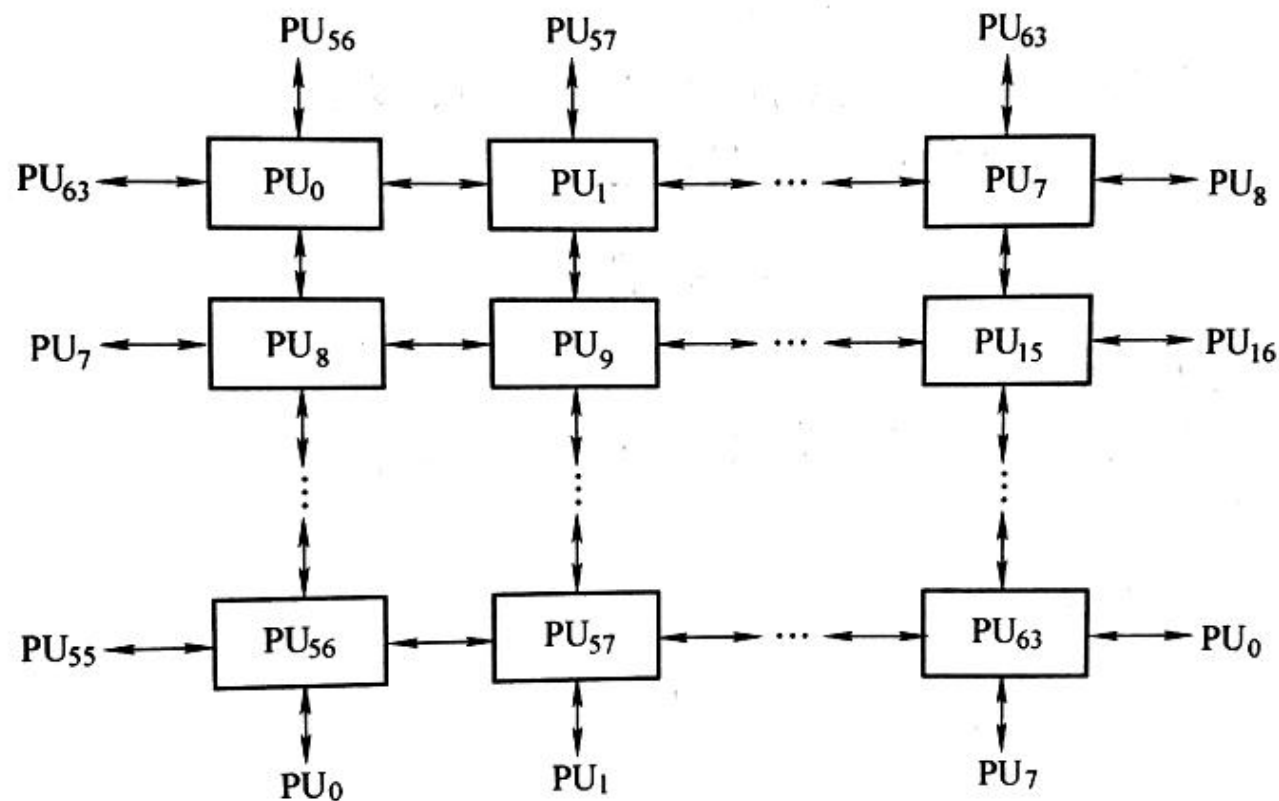


图 6-6 ILLIAC IV处理单元的阵列结构



那么意气风发地
走在成功的道路上

真题练练手

1、分布式存储器阵列处理机属于（ ） 1910

A:SIMD系统

B:SISD系统

C:MISD系统

D:MIMD系统



那么意气风发地
走在成功的道路上

真题练练手

1、分布式存储器阵列处理机属于（ ） 1910

A:SIMD系统

B:SISD系统

C:MISD系统

D:MIMD系统

答案：A



那么意气风发地
走在成功的道路上

真题练练手

3、下列属于阵列机所采用的技术的是 () 1810

A:时间重叠

B:资源重复

C:并发性

D:资源共享



那么意气风发地
走在成功的道路上

真题练练手

3、下列属于阵列机所采用的技术的是（） 1810

A:时间重叠

B:资源重复

C:并发性

D:资源共享

答案：



那么意气风发地
走在成功的道路上

真题练练手

4、与阵列处理机相比，流水线处理机利用的是（ ）方式而不是（ ）方式的并行技术。1910



那么意气风发地
走在成功的道路上

真题练练手

4、与阵列处理机相比，流水线处理机利用的是（ ）方式而不是（ ）方式的并行技术。1910

答案： 时间重叠 资源重复

6.2.3 ILLIAC IV的并行算法举例

1. 矩阵加

阵列处理机解决矩阵加是最简单的一维情况。

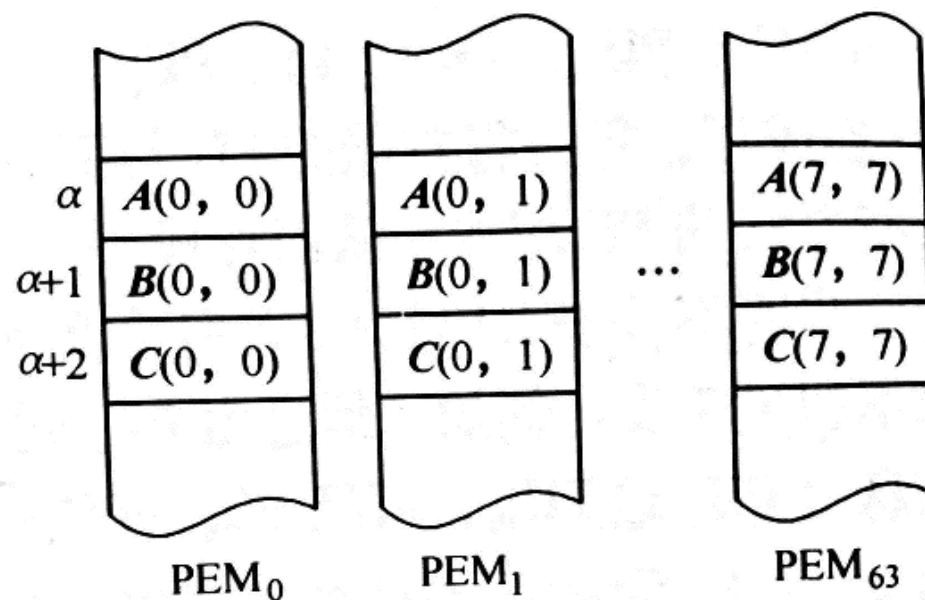


图 6-7 矩阵相加的存储器分配举例

6.2.3 ILLIAC IV的并行算法举例

2.矩阵乘

矩阵乘是二维数组运算，比矩阵加要复杂。设A、B和C为3个8×8的二维矩阵，给定A和B，计算C=A×B的64个分量可用公式

$$c_{ij} = \sum_{k=0}^7 a_{ik} b_{kj}$$

其中， $0 \leq i \leq 7$ 且 $0 \leq j \leq 7$ 。

6.2.3 ILLIAC IV的并行算法举例

3.累加和

这是一个将N个数的顺序相加转为并行相加的问题。为得到各项累加的部分和与最后的总和，要用到处理单元中的活跃标志位。只有处于活跃状态的处理单元才能执行相应的操作。

为叙述方便，取 $N=8$ ，即有8个数 $A(I)$ 顺序累加，其中 $0 \leq I \leq 7$ 。

在SISD计算机上可以编写下列FORTRAN程序：

```
C=0
```

```
DO 10 I=0,7
```

```
10 C=C+A(I)
```

6.2.3 ILLIAC IV的并行算法举例

3.累加和

这是一个将N个数的顺序相加转为并行相加的问题。为得到各项累加的部分和与最后的总和，要用到处理单元中的活跃标志位。只有处于活跃状态的处理单元才能执行相应的操作。

为叙述方便，取 $N=8$ ，即有8个数 $A(I)$ 顺序累加，其中 $0 \leq I \leq 7$ 。

在SISD计算机上可以编写下列FORTRAN程序：

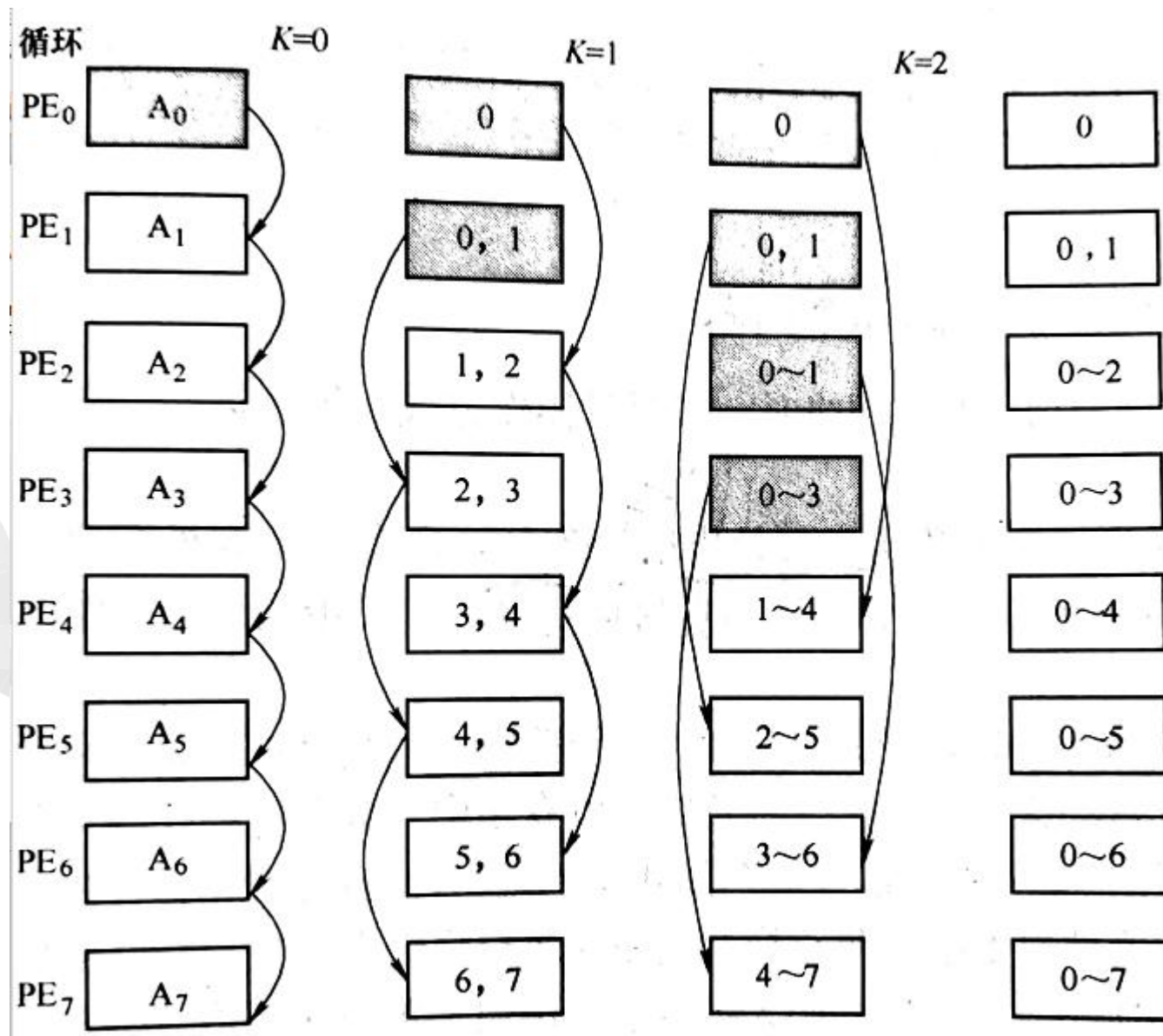
```
C=0
```

```
DO 10 I=0,7
```

```
10 C=C+A(I)
```

6.2.3 ILLIAC IV的并行算法举例

3.累加和



6.2.3 ILLIAC IV的并行算法举例

3.累加和

- (1) 置全部 PE_i 为活跃状态, $0 \leq i \leq 7$
- (2) 置全部 $A(i)$ 从 PE_i 的 α 单元读到相应 PE_i 的累加寄存器 RGA_i 中, $0 \leq i \leq 7$
- (3) 令 $K = 0$;
- (4) 将全部 PE_i 的 (RGA_i) 转送到传送寄存器 RGR_i , $0 \leq i \leq 7$
- (5) 将全部 PE_i 的 (RGR_i) 经过互连网络各右传送 2^K 步距, $0 \leq i \leq 7$
- (6) 令 $j = 2^K - 1$;
- (7) 置 $PE_0 \sim PE_j$ 为不活跃状态;
- (8) 处理活跃状态的所有 PE_i 执行 $(RGA_i) := (RGA_i) + (RGR_i)$, $j < i \leq 7$
- (9) $K := K + 1$;
- (10) 若 $K < 3$ 则转回(4) ;
- (11) 置全部 PE_i 为活跃状态, $0 \leq i \leq 7$
- (12) 将全部 PE_i 的累加寄存器内容 (RGA_i) 存入相应 PE_i 的 $\alpha + 1$ 单元中, $0 \leq i \leq 7$

6.3 SIMD计算机的互连网络

本节主要内容：

SIMD计算机的互连网络

SIMD计算机的循环和多级网络的想法和3个参数

多级互连网络的设计

阻塞式网络

全排列网络

6.3.1 互连网络的设计目标与互连函数

SIMD系统的互连网络的设计目标是：（简答）

- 1) 结构不要过分复杂，以降低**成本**；
- 2) 互连要**灵活**，以满足算法和应用的需要；
- 3) 处理单元间信息交换所需的传送步数要尽可能少，以提高**速度**性能；
- 4) 能用规整单一的基本构件组合而成，或者经多次通过或者经多级连接来实现复杂的互连，使模块性好，以便于用VLSI实现并满足系统的可**扩充**性。

（口诀：简洁、规整、让系统有好又快又便宜）

6.3.2互连网络应抉择的几个问题（单选）

操作方式有同步、异步及同步与异步组合3种。

交换方法主要有线路交换、包交换及线路与包交换组合3种。

网络的拓扑结构指的是互连网络入、出端可以连接的模式，有静态和动态两种。

动态网络有单级和多级两类。



那么意气风发地
走在成功的道路上

真题练练手

1、互连网络操作方式的种类有 () 1704

A:2种

B:3种

C:4种

D:5种

答案： B

6.3.3基本的单级互连网络

1.立方体单级网络

立方体 (Cube)单级网络的名称来源于图6-12所示的三维立方体结构。立方体的每个顶点 (网络的结点) 代表一个处理单元, 共有8个处理单元, 用zyx三位二进制码编号。它所能实现的入、出端连接如同立方体各顶点间能实现的互连一样, 即每个处理单元只能直接连到其二进制编号的某一位取反的其他3个处理单元上。

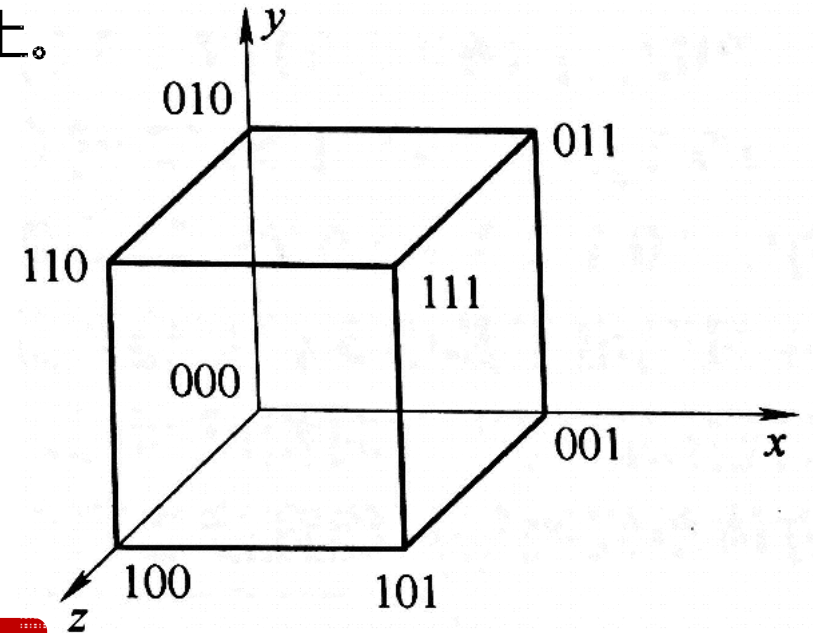
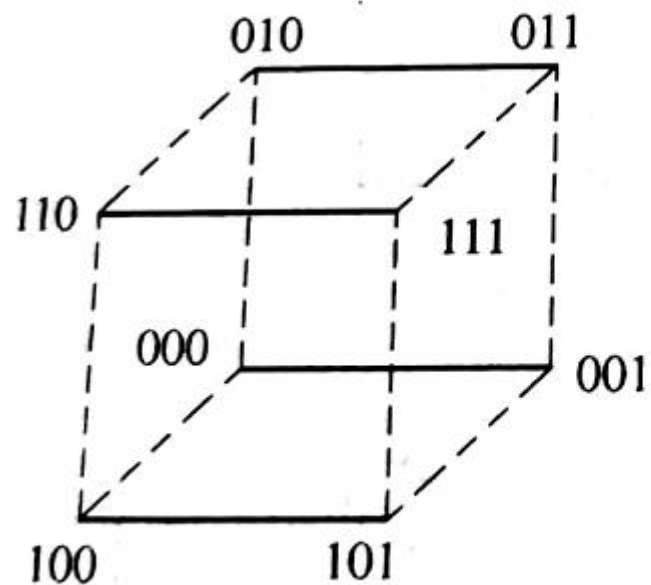


图6-12 三维立方体结构

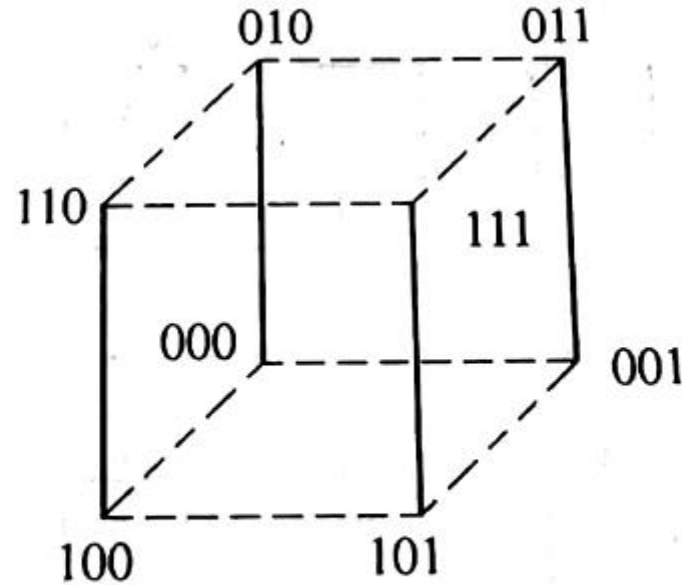
6.3.3基本的单级互连网络

1.立方体单级网络

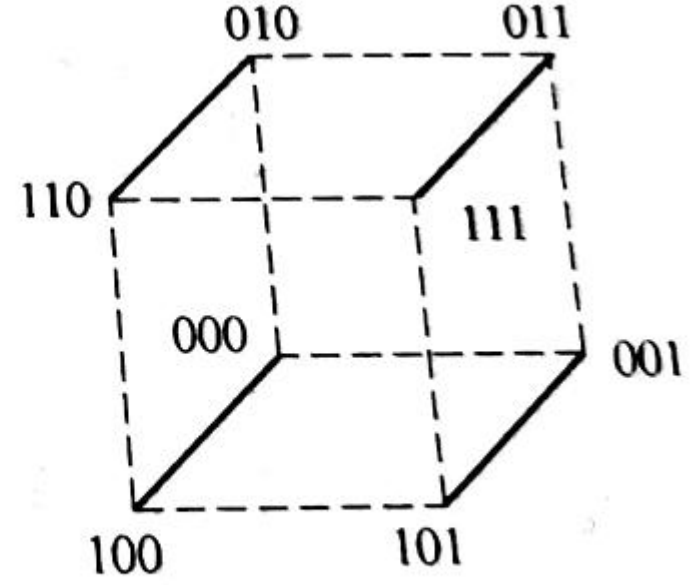
例如, 010只能连到000、011、110,不能直接连到对角线上的001、100、101、111。所以, 三维的立方体单级网络有3种互连函数: Cube0、Cube1和Cube2, 其连接方式如图6-13中的实线所示。



a)



b)

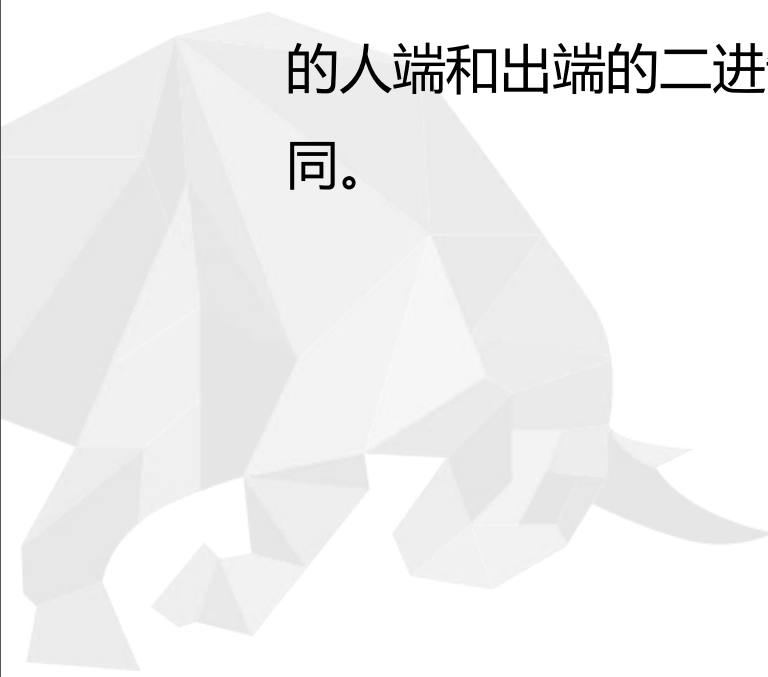


c)

6.3.3基本的单级互连网络

1.立方体单级网络

三维的立方体单级网络有3种互连函数：Cube0、Cube1和Cube2，Cube i 函数表示相连的人端和出端的二进制编号只在右起第 i 位 ($i=0, 1, 2$)上0、1互反，其余各位代码都相同。



6.3.3基本的单级互连网络

1.立方体单级网络

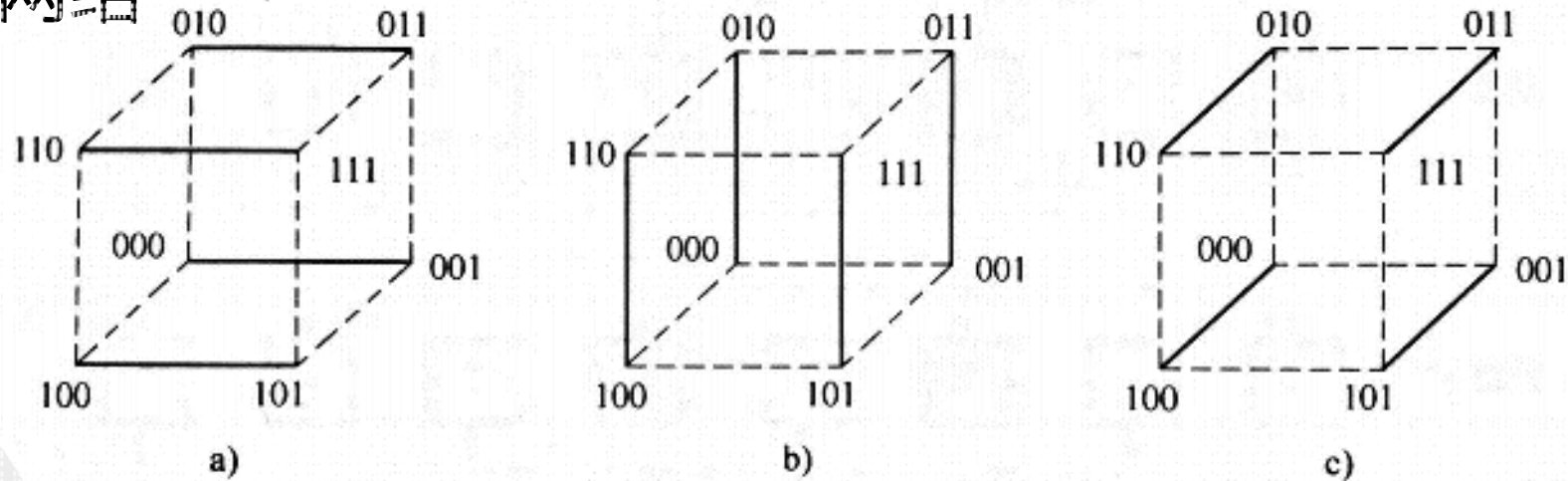


图 6-13 立方体单级网络连接示意

a) $Cube_0$ b) $Cube_1$ c) $Cube_2$

推广到 n 维时, N 个结点的立方体单级网络共有 $n = \log_2 N$ 种互连函数, 即

$$Cube_i (P_{n-1} \cdots P_i \cdots P_1 P_0) = P_{n-1} \cdots \bar{P}_i \cdots P_1 P_0$$

式中, P_i 为入端标号二进制码的第 i 位, 且 $0 \leq i \leq n-1$ 。当维数 $n > 3$ 时, 称为超立方体 (HyperCube) 网络。

6.3.3基本的单级互连网络

2.PM2I单级网络（综合）

PM2I单级网络是“加减 2^i ” (Plus-Minus 2^i)单级网络的简称。能实现与 j 号处理单元直接相连的是号为 $j \pm 2^i$ 的处理单元，即

$$\begin{cases} \text{PM2}_{+i}(j) = j + 2^i \mod N \\ \text{PM2}_{-i}(j) = j - 2^i \mod N \end{cases}$$

式中， $0 \leq j \leq N-1$ ， $0 \leq i \leq n-1$ ， $n = \log_2 N$ 。它共有 $2n$ 个互连函数。由于 $\text{PM2}_{+(n-1)} = \text{PM2}_{-(n-1)}$ ，因此PM2I互连网络只有 $2n-1$ 种互连函数是不同的。对于 $N=8$ 的三维PM2I互连网络的互连函数，有 PM2_{+0} 、 PM2_{-0} 、 PM2_{+1} 、 PM2_{-1} 和 $\text{PM2}_{\pm 2}$ 等5个不同的互连函数，它们分别为

6.3.3基本的单级互连网络

2.PM2I单级网络（综合）

$PM2_{+0}: (0\ 1\ 2\ 3\ 4\ 5\ 6\ 7)$

$PM2_{-0}: (7\ 6\ 5\ 4\ 3\ 2\ 1\ 0)$

$PM2_{+1}: (0\ 2\ 4\ 6)(1\ 3\ 5\ 7)$

$PM2_{-1}: (6\ 4\ 2\ 0)(7\ 5\ 3\ 1)$

$PM2_{+2}: (0\ 4)(1\ 5)(2\ 6)(3\ 7)$

其中， $(0\ 1\ 2\ 3\ 4\ 5\ 6\ 7)$ 表示 0 连到 1，与此同时，1 连到 2，2 连到 3， \dots ，7 连到 0。图 6-14 只画出了其中 3 种互连函数的情况， $PM2_{-0}$ 和 $PM2_{-1}$ 的连接与 $PM2_{+0}$ 和 $PM2_{+1}$ 的差别只是连接的箭头方向相反而已。可见在 PM2I 中，0 可以直接连到 1、2、4、6、7 上，比立方体单级网络只能直接连到 1、2、4 的要灵活。

6.3.3基本的单级互连网络

2.PM2I单级网络（综合）

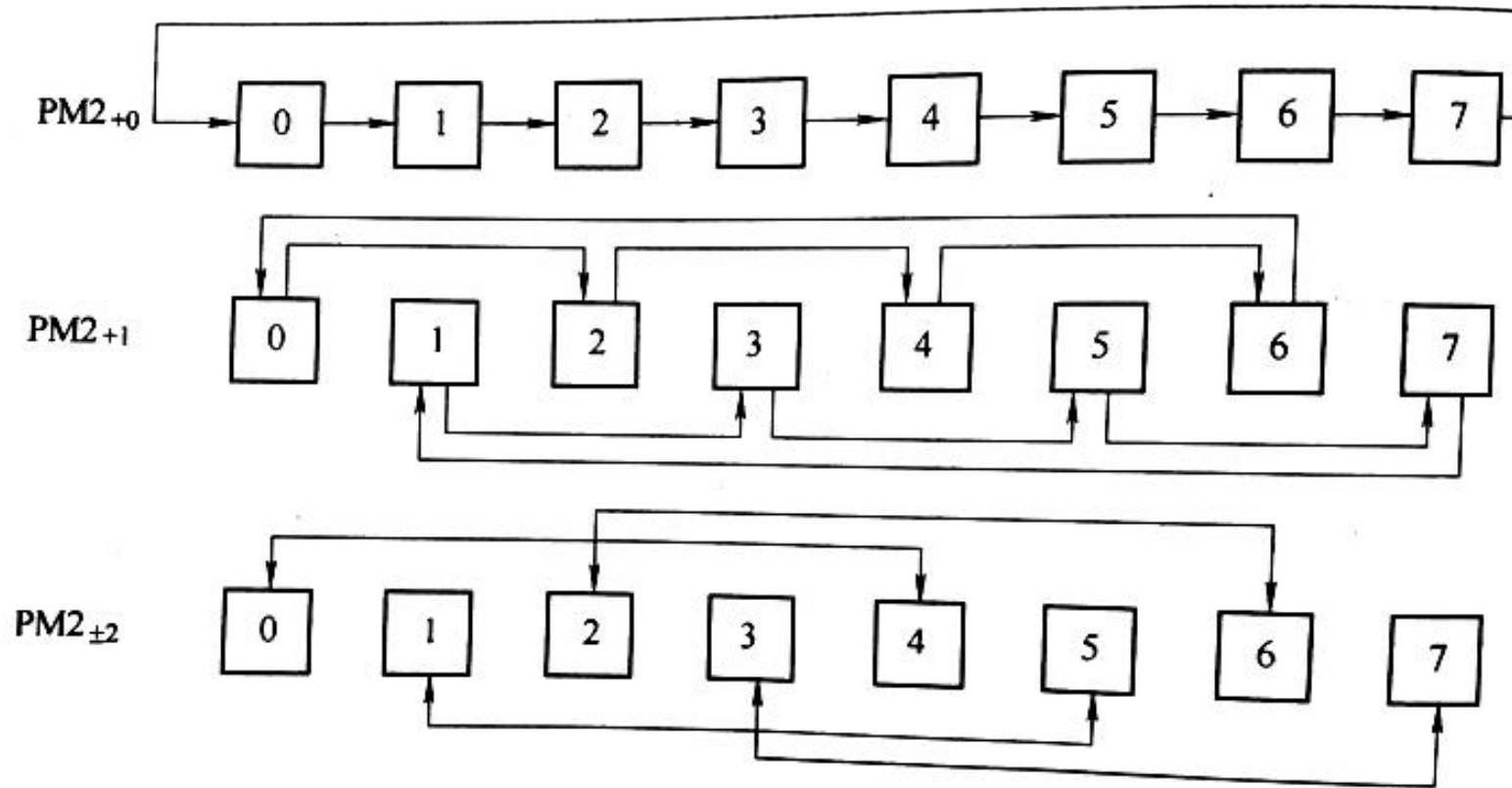


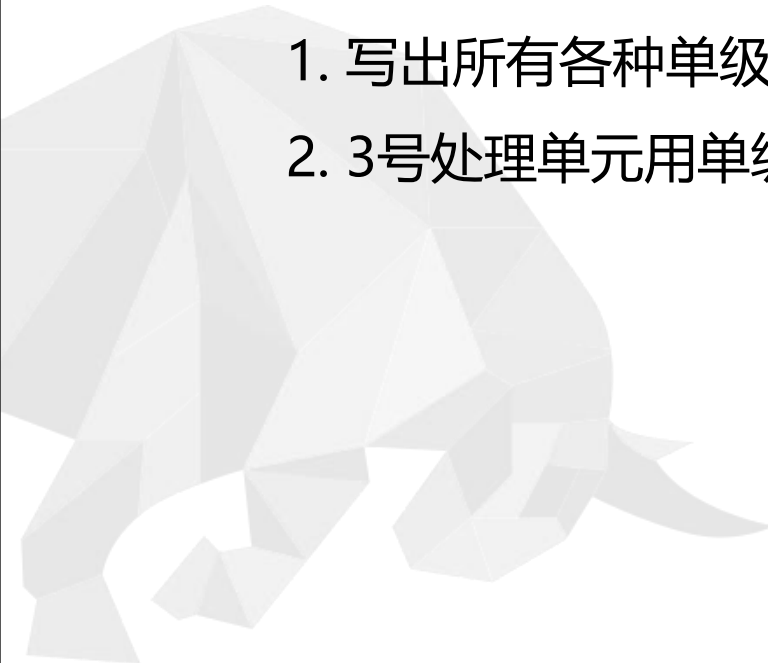
图 6-14 PM2I 互连网络的部分连接图

6.3.3基本的单级互连网络

2.PM2I单级网络

【1704真题】实现16个处理单元互连的PM2I单级网络。

1. 写出所有各种单级PM2I互连函数的一般式。
2. 3号处理单元用单级PM2I网络可以将数据商接传送到哪些处理单元上？



6.3.3基本的单级互连网络

2.PM2I单级网络

【1704真题】实现16个处理单元互连的PM2I单级网络。

1. 写出所有各种单级PM2I互连函数的一般式。
 2. 3号处理单元用单级PM2I网络可以将数据直接传送到哪些处理单元上？
- 2、3号处理单元可直接送到
1、2、4、5、7、11、15号处理单元上。

所有互连函数一般式：

$$PM2_{+0}(j) = j + 1 \mod 16$$

$$PM2_{-0}(j) = j - 1 \mod 16$$

$$PM2_{+1}(j) = j + 2 \mod 16$$

$$PM2_{-1}(j) = j - 2 \mod 16$$

$$PM2_{+2}(j) = j + 4 \mod 16$$

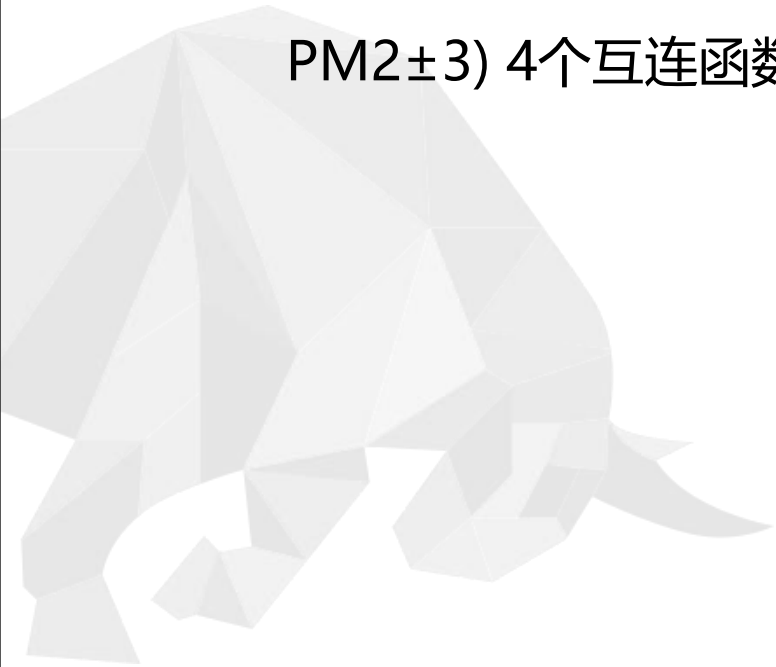
$$PM2_{-2}(j) = j - 4 \mod 16$$

$$PM2_{+3}(j) = j \pm 8 \mod 16$$

6.3.3基本的单级互连网络

2.PM2I单级网络（单选）

ILLIACIV处理单元的互连也是PM2I的特例，采用了其中的 $PM2 \pm 0$ 和 $PM2 \pm n/2$ (即 $PM2 \pm 3$) 4个互连函数。



6.3.3基本的单级互连网络

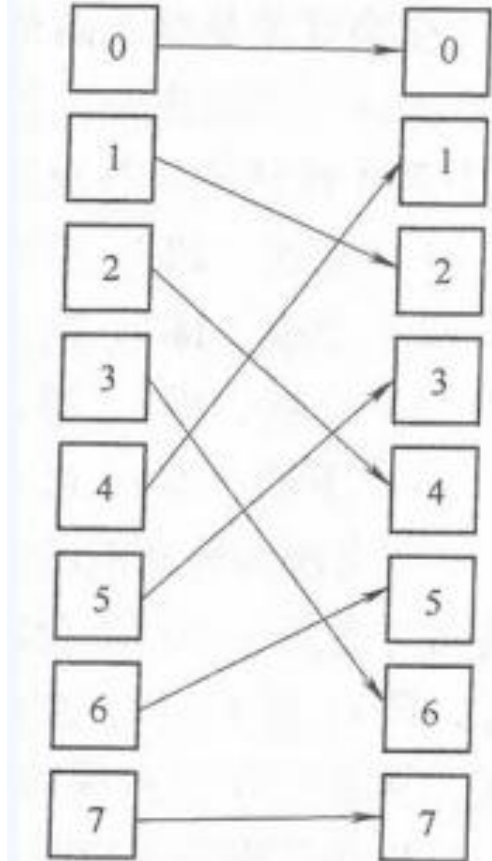
3.混洗交换单级网络（单选）

混洗交换单级（Shuffle-Exchange)网络包含两个互连函数，一个是**全混**（Perfect Shuffle),另一个是**交换**（Exchange)。

用互连函数表示为

$$\text{Shuffle}(P_{n-1}P_{n-2}\cdots P_1P_0) = P_{n-2}\cdots P_1P_0P_{n-1}$$

式中， $n = \log_2 N$ ； $P_{n-1}P_{n-2}\cdots P_1P_0$ 为入端编号的二进制码。



6.3.3基本的单级互连网络

3.混洗交换单级网络（单选）

混洗交换单级（Shuffle-Exchange)网络包含两个互连函数，一个是全混（Perfect Shuffle),另一个是交换（Exchange)。

用互连函数表示为

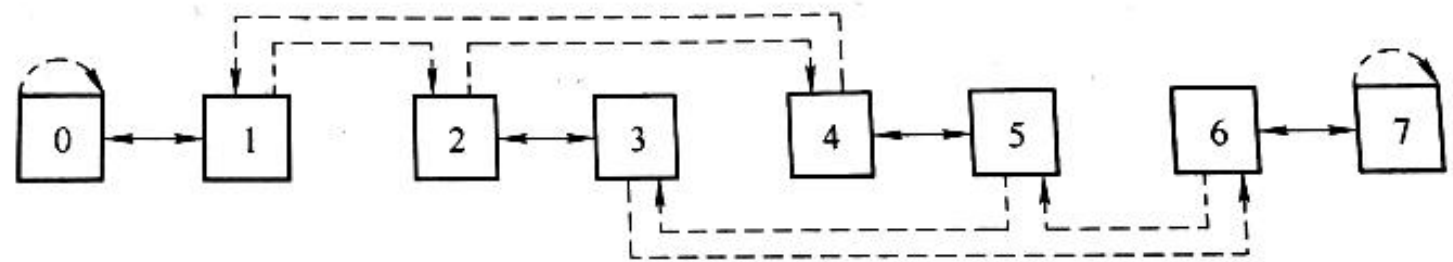


图 6-16 $N=8$ 时全混交换单级网络连接图

$$\text{Shuffle}(P_{n-1}P_{n-2}\cdots P_1P_0) = P_{n-2}\cdots P_1P_0P_{n-1}$$

式中, $n = \log_2 N$; $P_{n-1}P_{n-2}\cdots P_1P_0$ 为入端编号的二进制码。

6.3.3基本的单级互连网络

3.混洗交换单级网络（填空）

在混洗交换网络中，最远的两个入、出端号是全“0”和全“1”，它们的连接需要次交换和 $n - 1$ 次混洗，所以其最大距离为 $n - 1$ 。



6.3.3基本的单级互连网络

4.蝶形单级网络（综合）

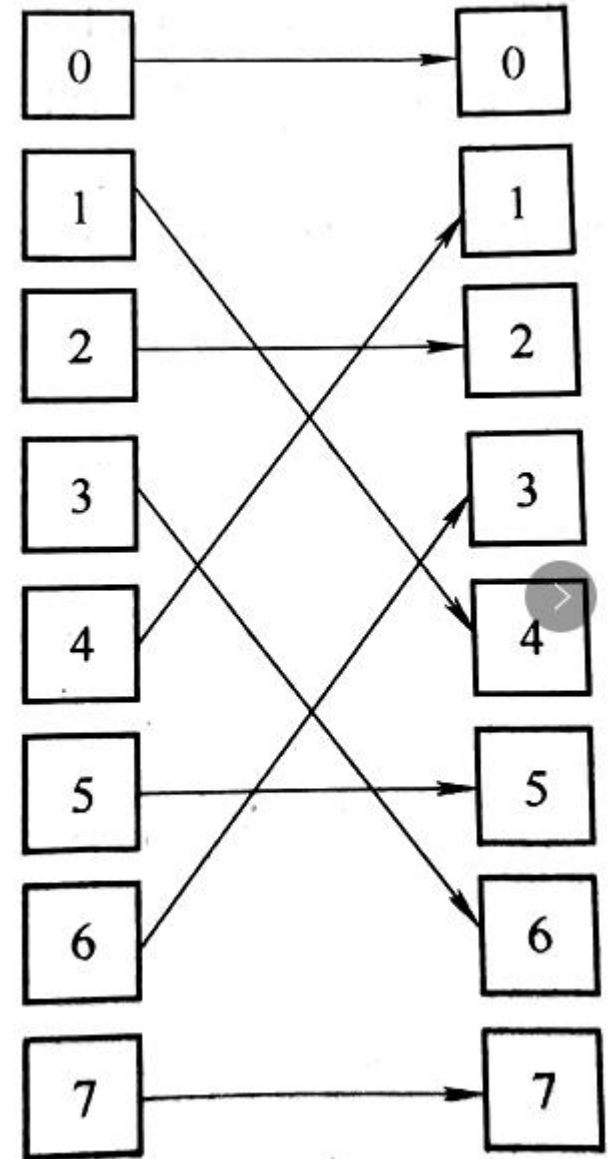
蝶形单级网络（Butterfly）的互连函数为

$$\text{Butterfly}(P_{n-1} P_{n-2} \dots P_1 P_0) = P_0 P_{n-2} \dots P_1 P_{n-1}$$

即将二进制地址的最高位和最低位相互交换位置。

图6-17所示为8个处理单元之间用蝶形单级互连网络互连的情况。它实现的是

$0 \rightarrow 0, 1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 6, 4 \rightarrow 1, 5 \rightarrow 5, 6 \rightarrow 3, 7 \rightarrow 7$ 的同时连接。





那么意气风发地
走在成功的道路上

真题练练手

1、编号为0,1,...,15的16个处理器,当互连网络函数为Cube3时,13号处理器连接到的处理器的号数是 () 1904

A:3

B:4

C:5

D:6



那么意气风发地
走在成功的道路上

真题练练手

1、编号为0,1,...,15的16个处理器,当互连网络函数为Cube3时,13号处理器连接到的处理器的号数是 () 1904

A:3

B:4

C:5

D:6

答案： C



那么意气风发地
走在成功的道路上

真题练练手

2、N个结点的混洗交换网络中，最远的两个入、出端的二进制编号是（ ）
和（ ），其最大距离为 $2n-1$ 。 0707



那么意气风发地
走在成功的道路上

真题练练手

2、N个结点的混洗交换网络中，最远的两个入、出端的二进制编号是（ ）
和（ ），其最大距离为 $2n-1$ 。 0707

答案：全“0” 全“1”



那么意气风发地
走在成功的道路上

真题练练手

3、编号为0,1,...,15的16个处理器,当互连网络函数为Shuffle时,9号处理器连接到的处理器的号数是 () 1810 1804

A:0

B:1

C:2

D:3



那么意气风发地
走在成功的道路上

真题练练手

3、编号为0,1,...,15的16个处理器,当互连网络函数为Shuffle时,9号处理器连接到的处理器的号数是 () 1810 1804

A:0

B:1

C:2

D:3

答案: D

6.3.4基本的多级互连网络（填空）

不同的多级互连网络，在所用的交换开关、拓扑结构和控制方式上各有不同。

交换开关是具有两个入端和两个出端的交换单元，用作各种多级互连网络的基本构件。无论入端或出端，如果令居于上方的都用 i 表示，居于下方的都用 j 表示，则可以定义下列4种开关状态或连接方式：

- 1) 直连，即 $i_{\text{入}}$ 连 $i_{\text{出}}$ ， $j_{\text{入}}$ 连 $j_{\text{出}}$ 。
- 2) 交换，即 $i_{\text{入}}$ 连 $j_{\text{出}}$ ， $j_{\text{入}}$ 连 $i_{\text{出}}$ 。
- 3) 上播，即 $i_{\text{入}}$ 连 $i_{\text{出}}$ 和 $j_{\text{出}}$ ， $j_{\text{入}}$ 悬空。
- 4) 下播，即 $j_{\text{入}}$ 连 $i_{\text{出}}$ 和 $j_{\text{出}}$ ， $i_{\text{入}}$ 悬空。

6.3.4基本的多级互连网络

(简单了解)

控制方式是对各个交换开关进行控制的方式，以多级立方体网络为例，它可以有3种：

- 1) **级控制**——同一级的所有开关只用一个控制信号控制，同时只能处于同一种状态。
- 2) **单元控制**——每一个开关都由自己独立的控制信号控制，可各自处于不同的状态。
- 3) **部分级控制**——第 i 级的所有开关分别用 $i+1$ 个信号控制， $0 \leq i \leq n-1$ ， n 为级数。利用上述交换开关、拓扑结构和控制方式3个参量，可以描述各种多级互连网络的结构。

6.3.4基本的多级互连网络

1.多级立方体网络（单选、填空）

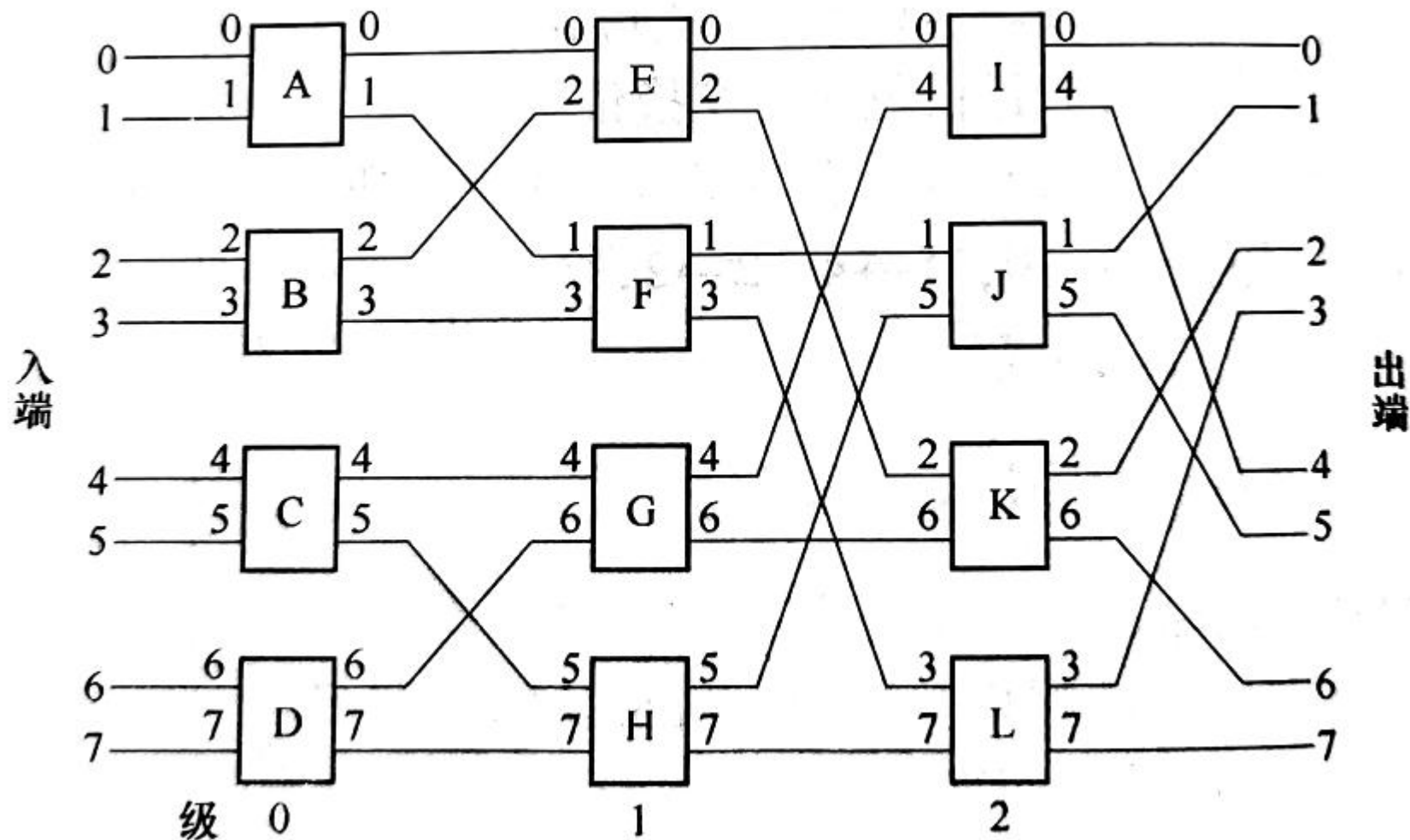
多级立方体网络有STARAN网络、间接二进制n方体网络等。

STARAN网络用作交换网络时，采用级控制，实现的是交换函数。



6.3.4 基本的多级互连网络

1. 多级立方体网络（单选、填空）



6.3.4 基本的多级互连网络

2. 多级混洗交换网络

多级混洗交换网络又称omega网络,

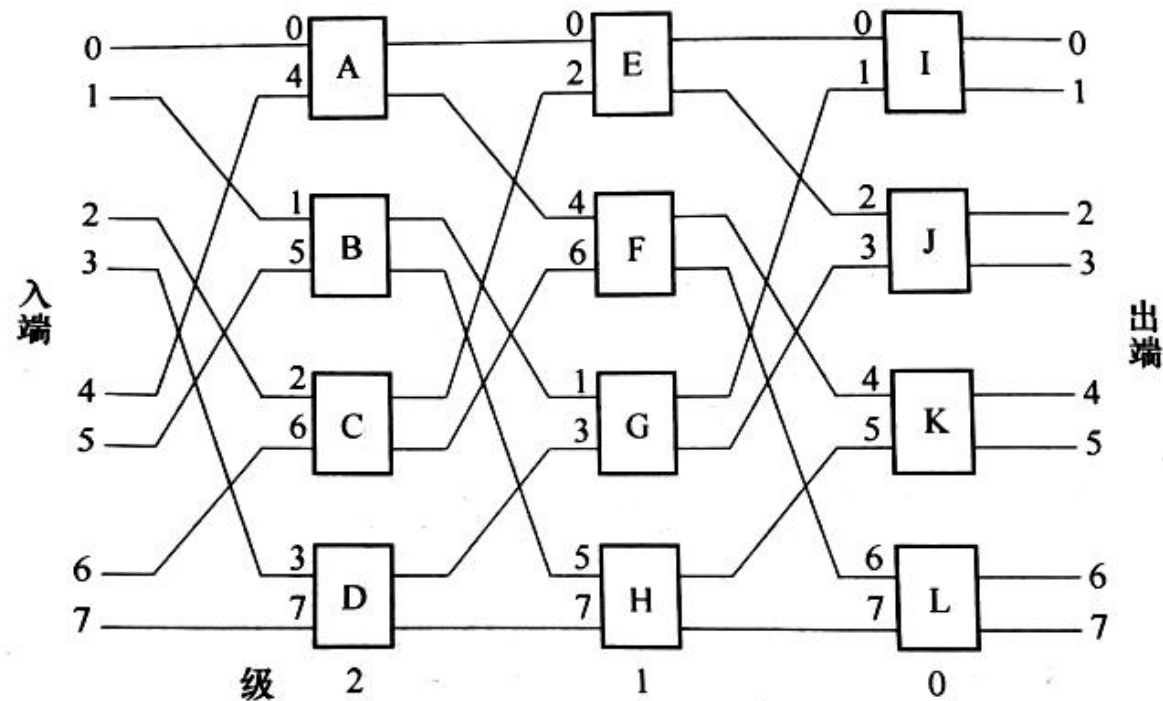
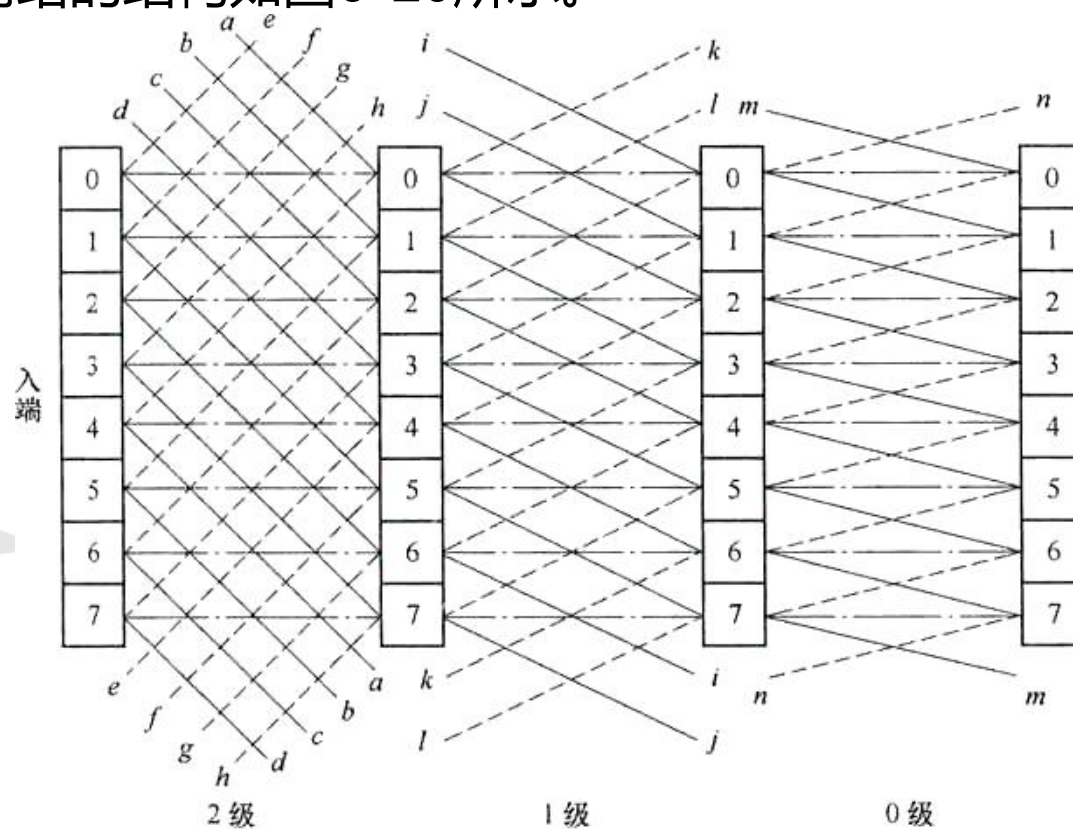


图 6-19 $N=8$ 的多级混洗交换网络

6.3.4基本的多级互连网络

3.多级PM2I网络

N=8的多级PM2I网络的结构如图6-20所示。

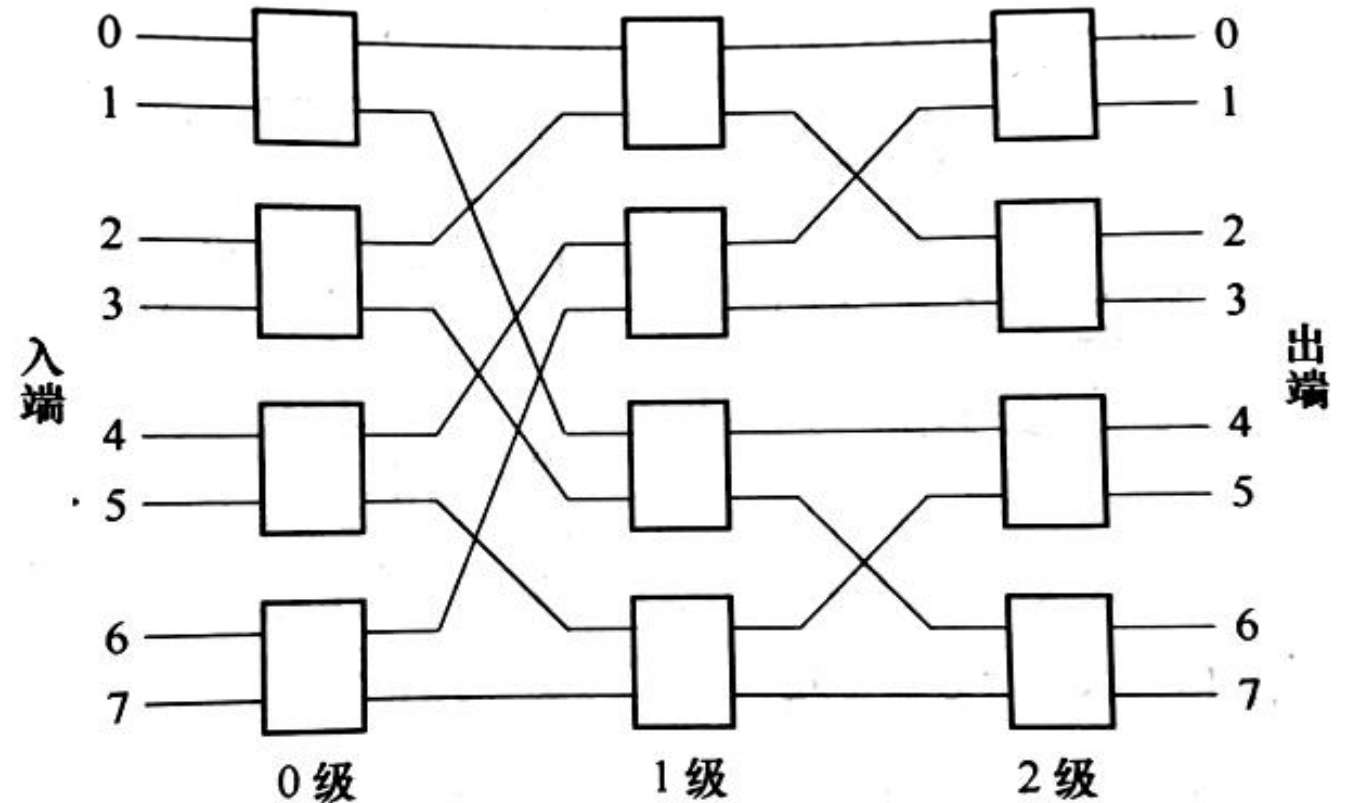


6.3.4 基本的多级互连网络

4. 基准网络

图6-21所示是 $N=8$ 的基准网络。

基准网络在多级网络中可作为中间介质，模拟一种网络的拓扑和功能。



6.3.4 基本的多级互连网络

5. 多级交叉开关网络

多级交叉开关 (CLOS)网络是一种非阻塞式网络，图6-22给出了一个三级交叉开关网络的结构。

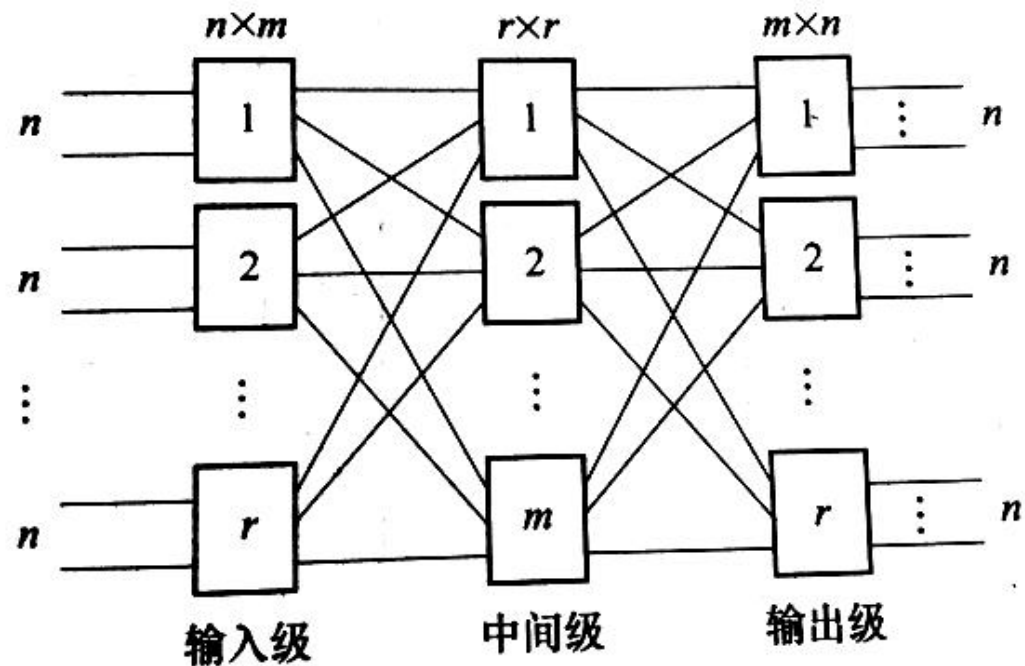


图 6-22 三级交叉开关网络的结构

6.3.4 基本的多级互连网络

5. 多级交叉开关网络

多级交叉开关 (CLOS)网络是一种非阻塞式网络, 图6-22给出了一个三级交叉开关网络的结构。

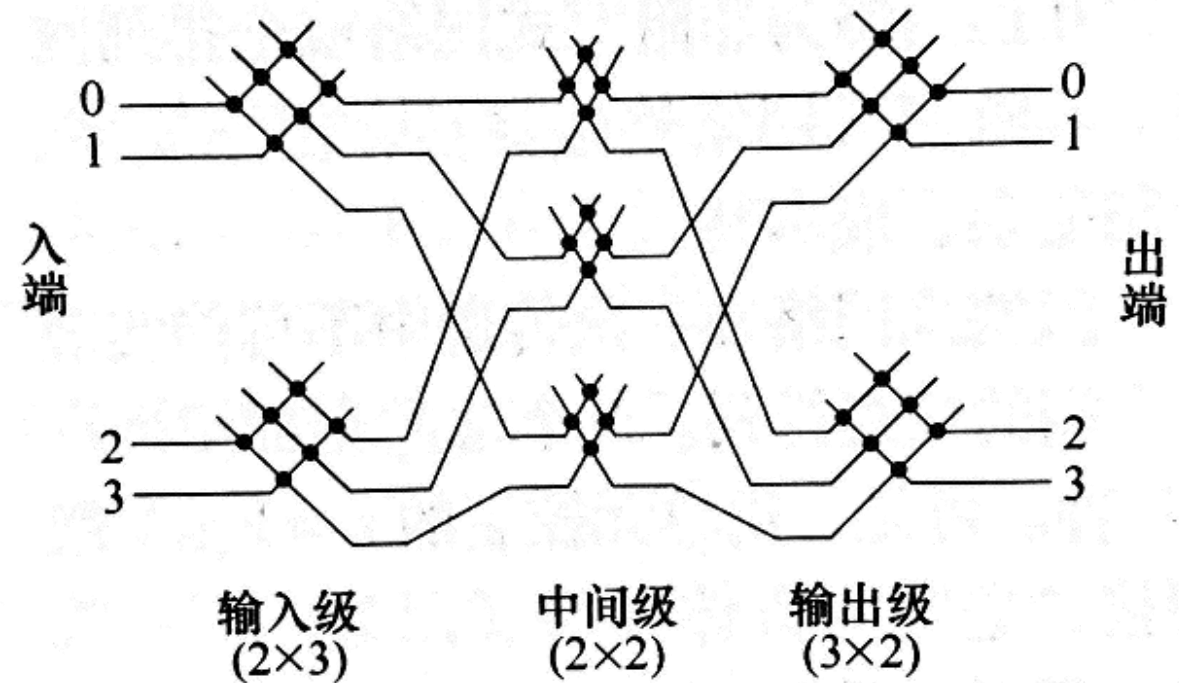


图 6-23 $N(3, 2, 2)$ 多级交叉
开关网络

6.3.4 基本的多级互连网络

6. 多级蝶式网络

图6-24所示是由16个 8×8 交叉开关作为基本构件组成的二级蝶式网络，级间采用8路混洗，构成了 64×64 的蝶式互连。

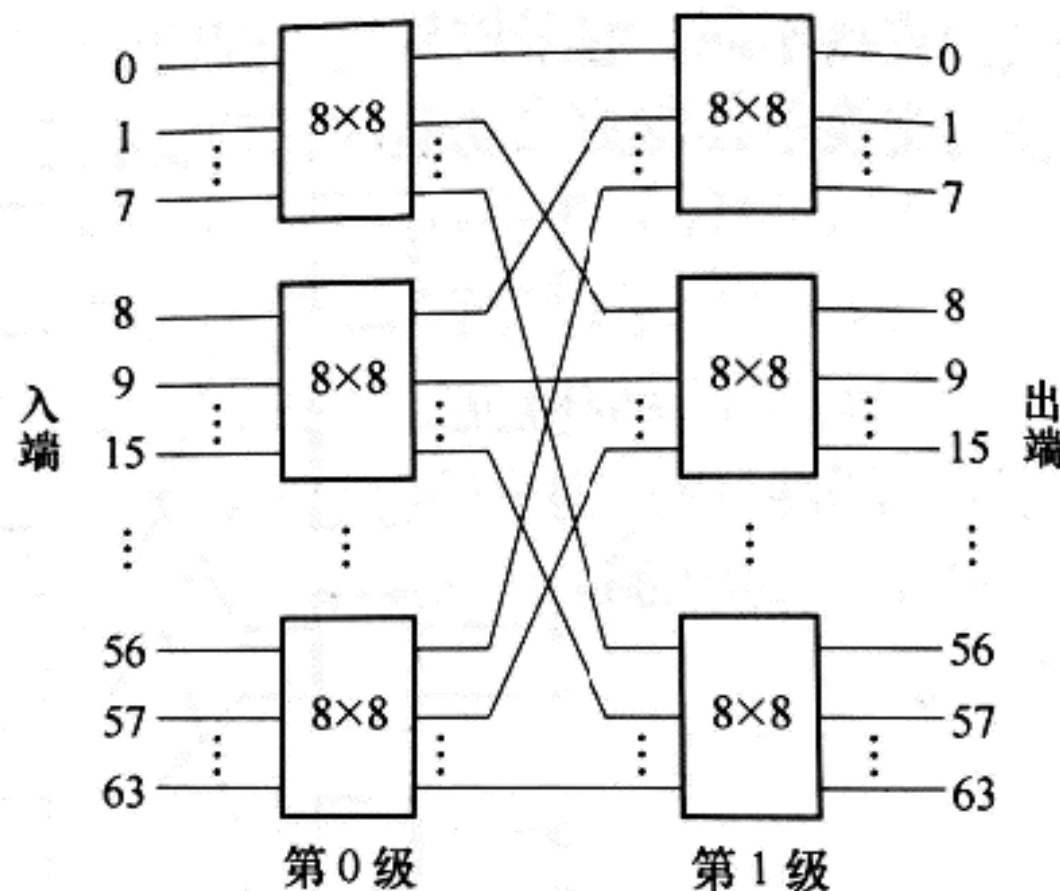


图 6-24 用 8×8 交叉开关构造的

6.3.5全排列网络（简单了解）

该网络至少有两个以上的通道能满足一对结点的互连要求，即数据寻径不唯一，有较多的冗余，这有利于选择合适的路径传送，可靠性、灵活性较好。

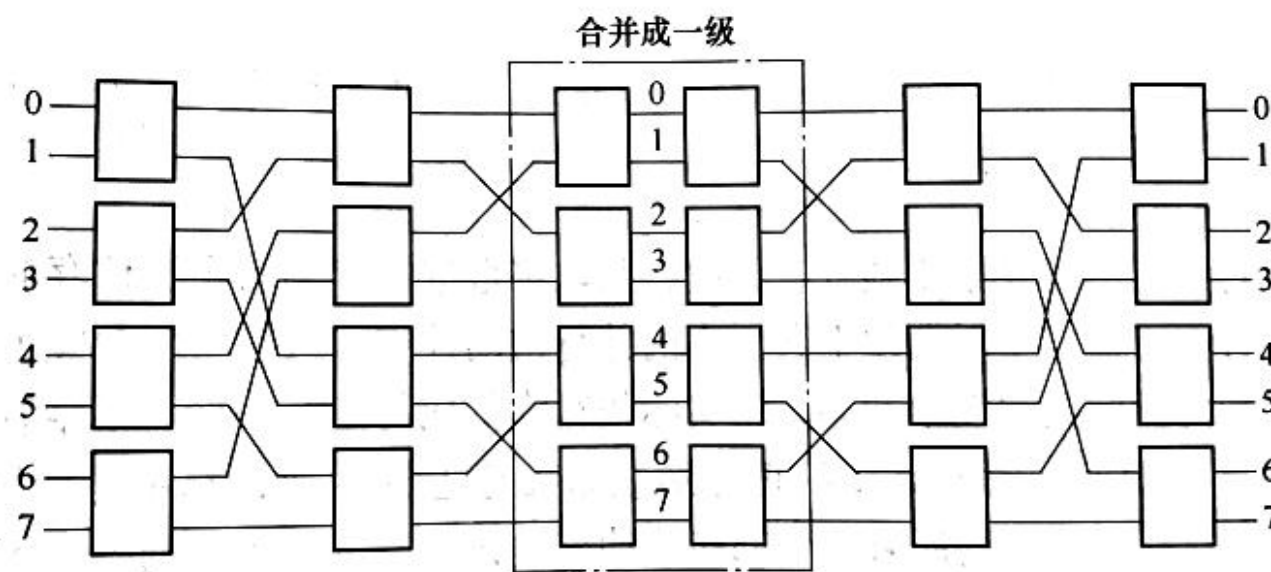


图 6-26 多级全排列网络举例（Benes 网络）



那么意气风发地
走在成功的道路上

真题练练手

1、不同的多级互连网络，在所用的（ ）、拓扑结构和（ ）上各有不同。0804 1304



那么意气风发地
走在成功的道路上

真题练练手

1、不同的多级互连网络，在所用的（ ）、拓扑结构和（ ）上各有不同。0804 1304

答案：交换开关 控制方式



那么意气风发地
走在成功的道路上

真题练练手

2、多级立方体对各个交换开关的控制方式有级控制、（ ）和（ ）3种。1410



那么意气风发地
走在成功的道路上

真题练练手

2、多级立方体对各个交换开关的控制方式有级控制、（ ）和（ ）3种。1410

答案：单元控制 部分级控制

6. 4共享主存构形的阵列处理机中并行存储器的无冲突访问

情况1

对一维数组而言，假定并行存储器分体数 m 为4,交叉存放一维数组 a_0, a_1, a_2, \dots ，如图6-27所示。

存储体体号			
0	1	2	3
a_0	a_1	a_2	a_3
a_4	a_5	a_6	a_7
a_8	a_9	a_{10}	a_{11}
a_{12}	.	.	.

图 6-27 一维数组的存储
($m = 4$)

6. 4共享主存构形的阵列处理机中并行存储器的无冲突访问情况2

对于二维数组（结论也适用于多维数组）而言，假设主存有 m 个分体并行，从中访问有 n 个元素的数组子集。这 n 个元素的变址跳距对于二维数组的行、列、主对角线、次对角线都是不一样的，但要求都能实现无冲突访问。

存储体体号			
0	1	2	3
a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

图 6-28 4×4 数组的直接按行存储
($m = n = 4$)

6. 4共享主存构形的阵列处理机中并行存储器的无冲突访问

(填空)

为了能使行或列的各元素都能并行访问，采取将数据在存储器中**错位**存放，如图6-29 所示。但是该方案可造成主对角线上各元素的并行访问冲突，致使实际频宽下降一半；次对角线上各元素的访问则都发生冲突，使实际频宽降低成与串行一样。

存储体体号			
0	1	2	3
a_{00}	a_{01}	a_{02}	a_{03}
a_{13}	a_{10}	a_{11}	a_{12}
a_{22}	a_{23}	a_{20}	a_{21}
a_{31}	a_{32}	a_{33}	a_{30}

图 6-29 4×4 数组一种错位存放的方案
($m = n = 4, \delta_1 = \delta_2 = 1$)

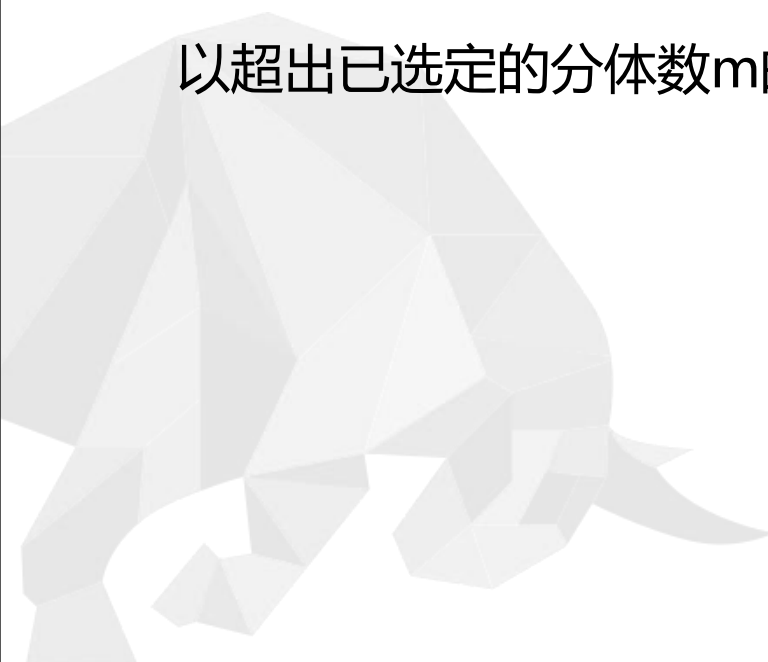
存储体体号				
0	1	2	3	4
a_{00}	a_{01}	a_{02}	a_{03}	
a_{13}		a_{10}	a_{11}	a_{12}
a_{21}	a_{22}	a_{23}		a_{20}
	a_{30}	a_{31}	a_{32}	a_{33}

图 6-30 4×4 数组错位存放的例子
($m = 5, n = 4, \delta_1 = 2, \delta_2 = 1$)

6. 4共享主存构形的阵列处理机中并行存储器的无冲突访问

情况3

并行存储器中存放的数组大小是不固定的，多维数组各维的元素个数也不一定相等，它们还可以超出已选定的分体数 m 的值。





那么意气风发地
走在成功的道路上

真题练练手

1、将二维数组中各元素在存储器中（ ）存放可以使行或列的各元素都能并行访问，但会造成（ ）上各元素的并行访问冲突。1604



那么意气风发地
走在成功的道路上

真题练练手

1、将二维数组中各元素在存储器中（ ）存放可以使行或列的各元素都能并行访问，但会造成（ ）上各元素的并行访问冲突。1604

答案：错位 主对角线

6.5脉动阵列流水处理机

本节主要内容：

脉动阵列流水处理机的工作原理

通用脉动阵列结构的实现方法



6.5.1 脉动阵列结构的原理

(填空)

脉动阵列结构是由一组处理单元（PE）构成的阵列。

运算时数据在阵列结构的各个处理单元间沿各自的方向**同步**向前推进，就像血液受心脏有节奏地搏动在各条血管中间同步向前流动一样。因此，形象地称其为脉动阵列结构。实际上，为了执行多种计算，脉动型系统内的**输入数据流**和**结果数据流**可以在多个不同方向上以不同速度向前搏动。

6.5.1 脉动阵列结构的原理

(简答)

脉动阵列结构具有如下一些特点：

- 1) 结构简单、规整，模块化强，**可扩充性**好，非常适合用超大规模集成电路实现。
- 2) PE间数据通信距离短、规则，使数据流和控制流的设计、同步控制等均**简单规整**。
- 3) 脉动阵列中所有PE能同时运算，具有极高的计算**并行性**，可通过流水获得很高的运算效率和吞吐率。输入数据能被多个处理单元重复使用，大大减轻了阵列与外界的 I/O 通信量，降低了对系统主存和 I/O 系统频宽的要求。
- 4) 脉动阵列结构的构形与特定计算任务和算法密切相关，具有某种**专用性**，限制了应用范围，这对 VLSI 是不利的。

6.5.2 通用脉动阵列结构

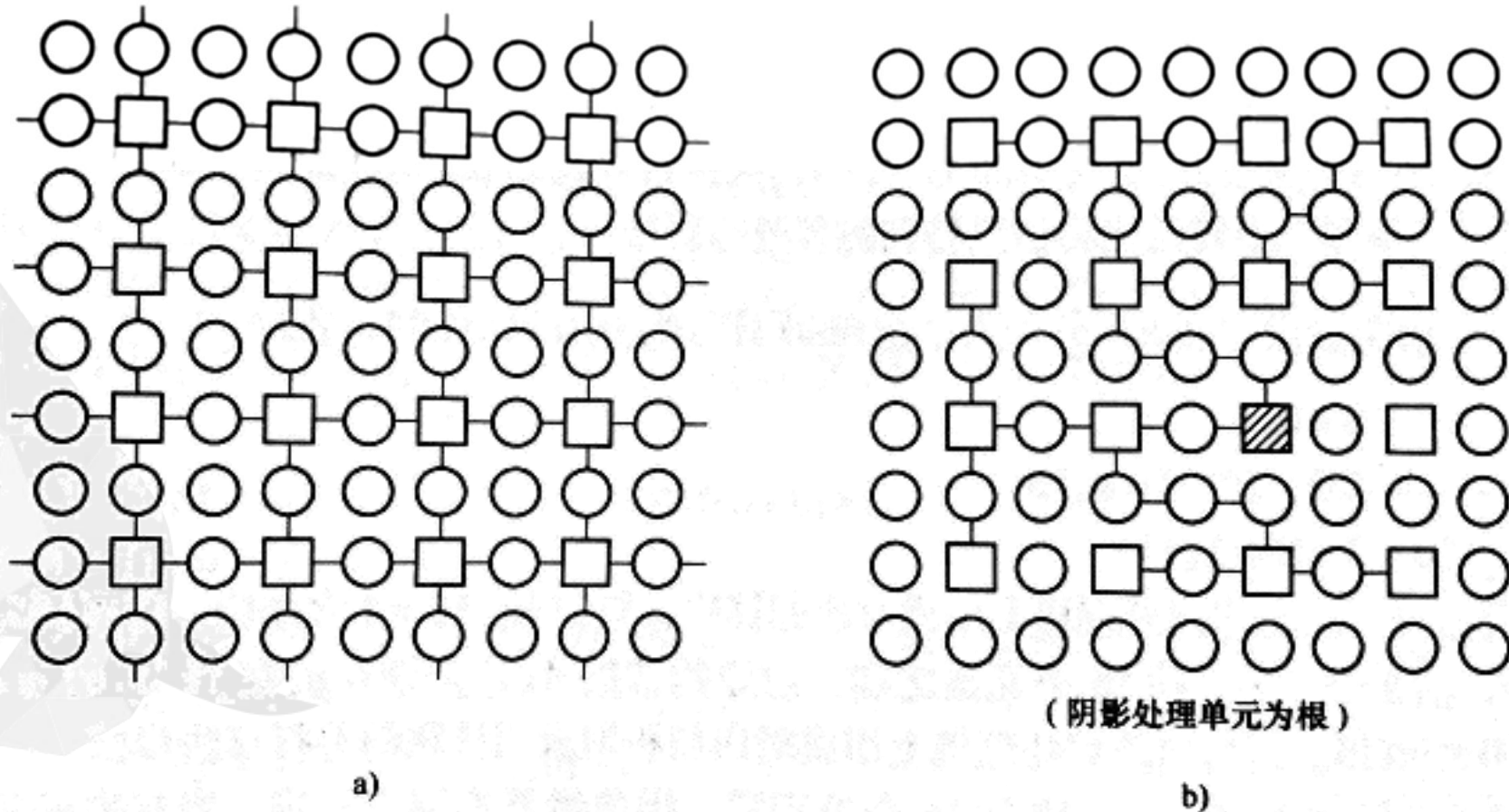


图 6-34 可编程脉动阵列结构

a) 控制开关按正方形阵列结构互连 b) 控制开关按二叉树形阵列结构互连



那么意气风发地
走在成功的道路上

真题练练手

- 1、脉动阵列结构是由一组处理单元PE构成的阵列，阵列中的输入数据流和（ ）数据流可各自沿多个方向（ ）地向前推进。1304



那么意气风发地
走在成功的道路上

真题练练手

1、脉动阵列结构是由一组处理单元PE构成的阵列，阵列中的输入数据流和（ ）数据流可各自沿多个方向（ ）地向前推进。1304

答案：结果 同步



第7章 多处理机

第7章 多处理机

●	多处理机的概念、问题和硬件	★
●	紧耦合多处理机多cache的一致性问题	★
●	多处理机的并行性和性能	★ ★
●	多处理机的操作系统	
●	多处理机的发展	

7.1 多处理机的概念、问题和硬件结构

本节主要内容：

多处理机的定义和并行性等级

多处理机与阵列处理机的对比

多处理机的主要技术问题

7.1.1 多处理机的基本概念和要解决的技术问题

(填空)

多处理机是指有两台以上的处理机，共享I/O子系统，机间经共享主存或高速通信网络通信，在统一操作系统控制下，协同求解大而复杂问题的计算机系统。

7.1.1多处理机的基本概念和要解决的技术问题

使用多处理机的目的（填空、简答）

第一个目的是通过多台处理机对多个**作业**、**任务**进行并行执行来提高解题速度，从而提高系统的整体性能；
第二个目的是使用**冗余**的多个处理机通过重新组织来提高系统的**可靠性**、**适应性**和**可用性**。（一适两可）

7.1.1 多处理机的基本概念和要解决的技术问题

(单选、填空)

多处理机可以有同构型、异构型和分布型3种。



7.1.1 多处理机的基本概念和要解决的技术问题

多处理机是属于多指令流多数据流的系统。它与单指令流多数据流的阵列处理机相比，有很大的差别。其差别主要来源于并行性的等级不同。阵列处理机主要是针对向量、数组处理，实现向量指令操作级的并行，是开发并行性中的同时性。（资源重复）多处理机实现的则是更高一级的作业或任务间的并行，是开发并行性中的并发性。

7.1.1 多处理机的基本概念和要解决的技术问题

(简答)

- 1) 在**硬件**结构上，它的多个处理机要用多个指令部件分别控制，通过共享主存或机间互连网络实现异步通信；
- 2) 在**算法**上，不限于向量、数组处理，还要挖掘和实现更多通用算法中隐含的并行性；
- 3) 在**系统管理**上，要更多地依靠操作系统等软件手段，有效地解决资源分析和管理工作，特别是任务分配、处理机调度、进程的同步和通信等问题。

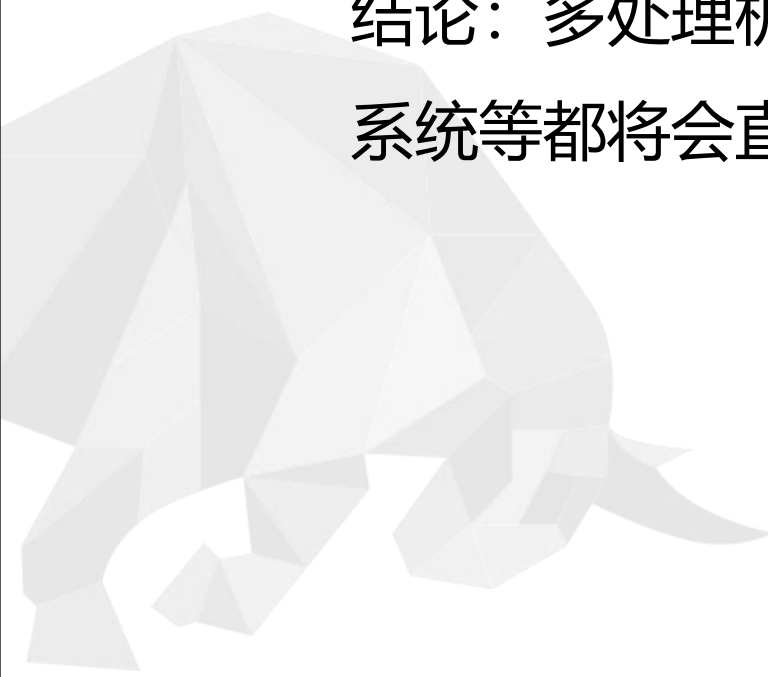
7.1.1多处理机的基本概念和要解决的技术问题

(填空)

多处理机执行并发任务所需的处理机的机数是不固定的。各处理机进入或退出任务的时间及所需资源的变化比较大。必须研究如何较好地解决动态的资源分配和任务调度，让各处理机的**负荷**尽可能均衡，并要防止**死锁**。

7.1.1 多处理机的基本概念和要解决的技术问题

结论：多处理机的结构、机间互连、并行算法、并行语言、编译、操作系统等都将会直接影响到系统的性能和效率。



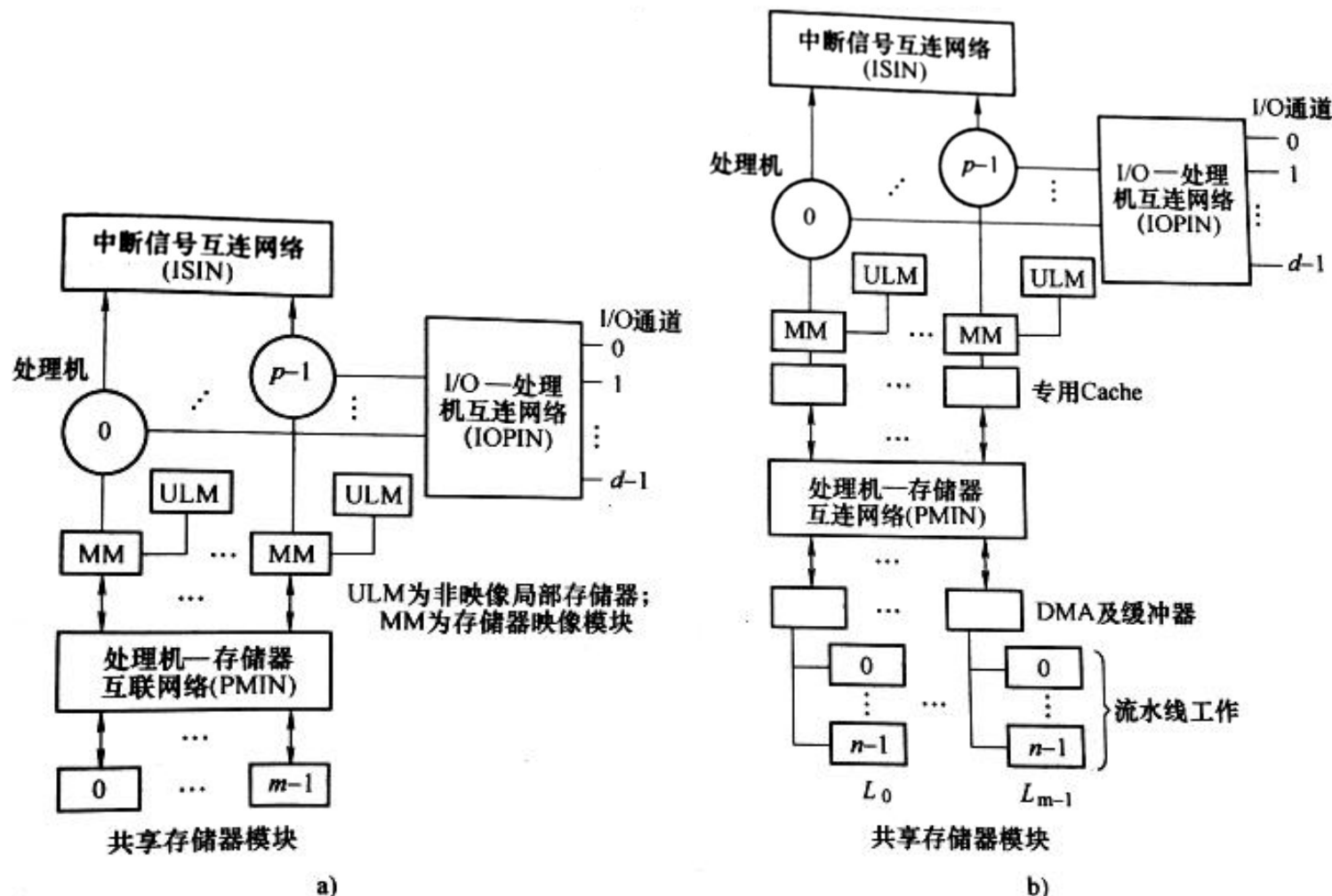
7.1.2多处理机的硬件结构

1.紧耦合和松耦合（单选）

(1)紧耦合多处理机

紧耦合多处理机是通过**共享主存**实现处理机间通信的，其通信速率受限于主存频宽。

1) 层次型 2) 非层次型



7.1.2多处理机的硬件结构

1.紧耦合和松耦合（单选）

(2)松耦合多处理机

不同处理机间或者通过**通道互连实现通信**，以共享某些外围设备；或者**通过消息传送系统**来交换信息，这时各台处理机可带有自己的外围设备。

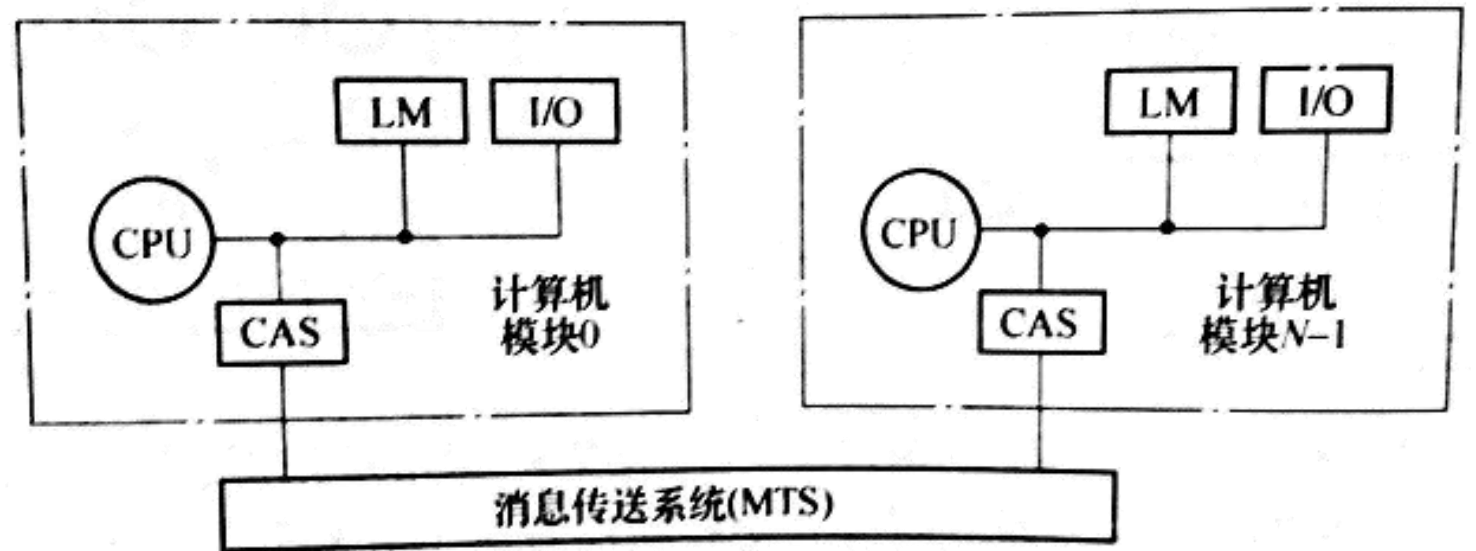


图 7-4 通过消息传输系统连接的松耦合多处理机结构



那么意气风发地
走在成功的道路上

真题练练手

1、紧耦合多处理机系统的机间通信是通过（ ） 1910

A:共享总线实现

B:共享Cache实现

C:共享虚拟存储器实现

D:共享主存实现



那么意气风发地
走在成功的道路上

真题练练手

1、紧耦合多处理机系统的机间通信是通过（ ） 1910

A:共享总线实现

B:共享Cache实现

C:共享虚拟存储器实现

D:共享主存实现

答案： D



那么意气风发地
走在成功的道路上

真题练练手

2、通过通道互连实现通信,或通过消息传送系统交换信息的计算机系统是 ()

1904

A:向量处理机

B:紧耦合多处理机

C:松耦合多处理机

D:标量处理机



那么意气风发地
走在成功的道路上

真题练练手

2、通过通道互连实现通信,或通过消息传送系统交换信息的计算机系统是 ()

1904

A:向量处理机

B:紧耦合多处理机

C:松耦合多处理机

D:标量处理机

答案：C



那么意气风发地
走在成功的道路上

真题练练手

3、多端口存储器适合于连接 () 0707 0704

A:松耦合多处理机

B:紧耦合多处理机

C:机数很多的多处理机

D:机数可变的的处理机



那么意气风发地
走在成功的道路上

真题练练手

3、多端口存储器适合于连接 () 0707 0704

A:松耦合多处理机

B:紧耦合多处理机

C:机数很多的多处理机

D:机数可变的的处理机

答案：A



那么意气风发地
走在成功的道路上

真题练练手

4、紧耦合多处理机实现多处理机机间通信靠的是共享（ ）。1304



那么意气风发地
走在成功的道路上

真题练练手

4、紧耦合多处理机实现多处理机机间通信靠的是共享（ ）。1304

答案：主存

7.1.2多处理机的硬件结构

2.机间互连形式（单选、填空）

多处理机的互连一般采用**总线**、**环形互连**、**交叉开关**、**多端口存储器**或**蠕虫穿洞**寻径网络等几种形式。

随着技术的发展，当处理机的机数较多时，也有类似SIMD的多级网络。

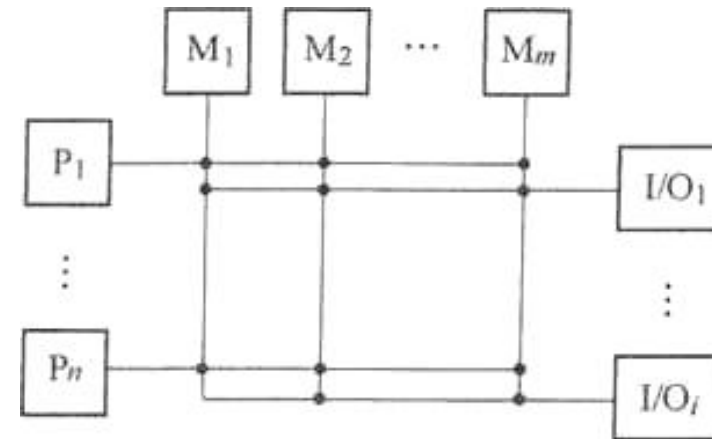
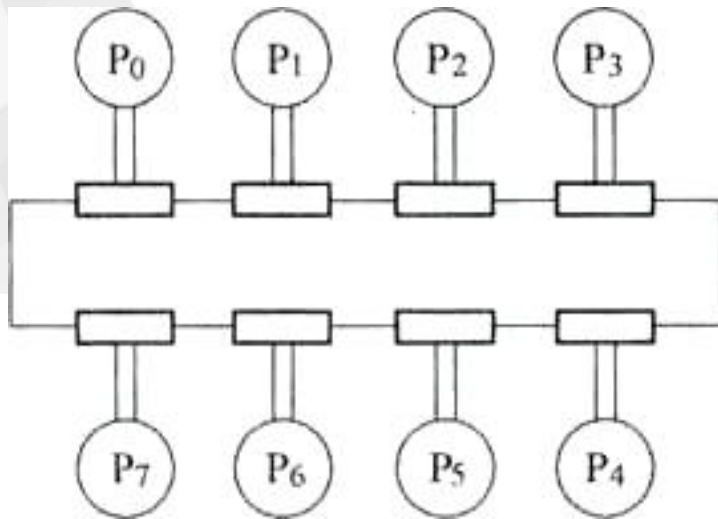
7.1.2 多处理机的硬件结构

2. 机间互连形式

(1) 总线形式

(2) 环形互连形式

(3) 交叉开关形式



7.1.2多处理机的硬件结构

2.机间互连形式

(4)多端口存储器形式

(5)蠕虫穿洞寻径网络

(6)开关枢纽结构形式

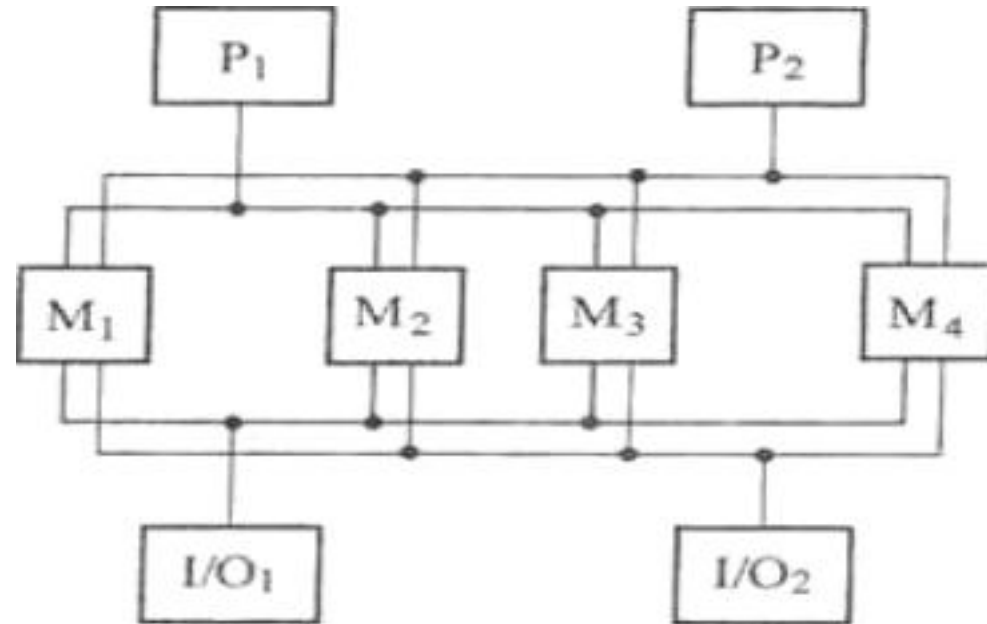


图7-8 四端口存储形式的结构



那么意气风发地
走在成功的道路上

真题练练手

1、多处理机的互连形式一般有总线形式、环形互连形式、（ ）形式、多端口存储器形式和（ ）形式等几种。0804 0807 0907



那么意气风发地
走在成功的道路上

真题练练手

1、多处理机的互连形式一般有总线形式、环形互连形式、（ ）形式、多端口存储器形式和（ ）形式等几种。0804 0807 0907

答案：交叉开关 开关枢纽结构



那么意气风发地
走在成功的道路上

真题练练手

2、多处理机的互连一般采用总线、（ ）、（ ）或多端口存储器等几种形式。1410



那么意气风发地
走在成功的道路上

真题练练手

2、多处理机的互连一般采用总线、（ ）、（ ）或多端口存储器等几种形式。1410

答案：环形互连 交叉开关



那么意气风发地
走在成功的道路上

真题练练手

3、机间互连的多端口存储器形式适合应用于（）1504

- A:紧耦合多处理机系统
- B:机数很多的多处理机系统
- C:机数较少的多处理机系统
- D:松耦合多处理机系统



那么意气风发地
走在成功的道路上

真题练练手

3、机间互连的多端口存储器形式适合应用于（）1504

A:紧耦合多处理机系统

B:机数很多的多处理机系统

C:机数较少的多处理机系统

D:松耦合多处理机系统

答案：C

7.1.2多处理机的硬件结构

3.存储器的组织（单选）

多处理机的主存一般都采用由多个模块构成的并行存储器。

为减少访问主存冲突，采用的方式一般是：**并行多体交叉主存系统**



7.1.2 多处理机的硬件结构

3. 存储器的组织

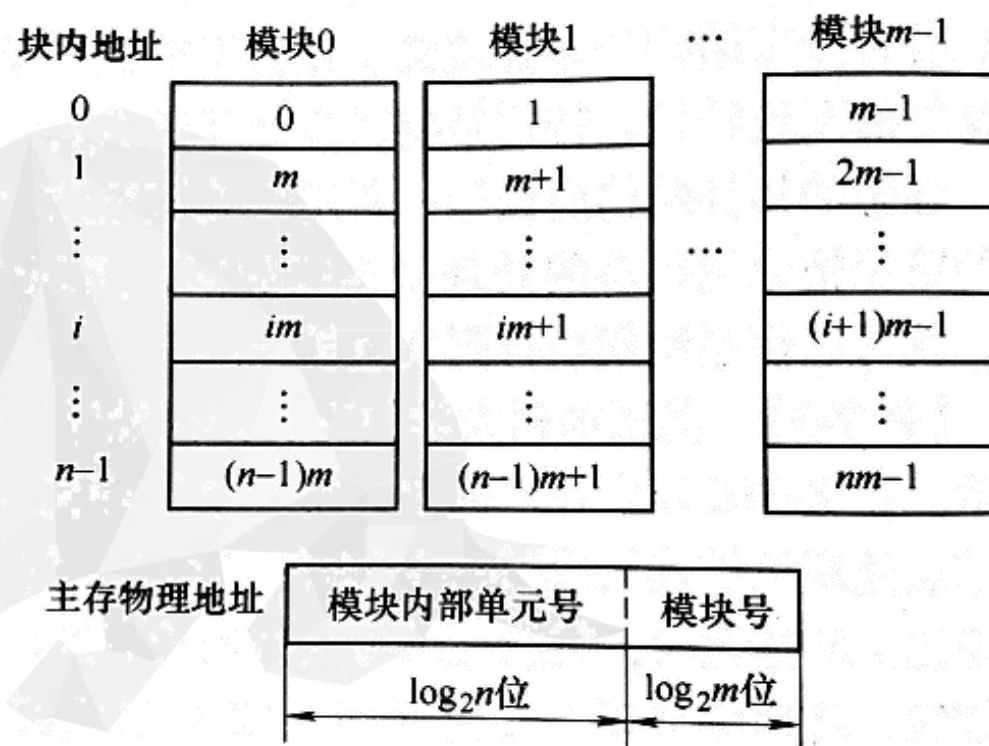
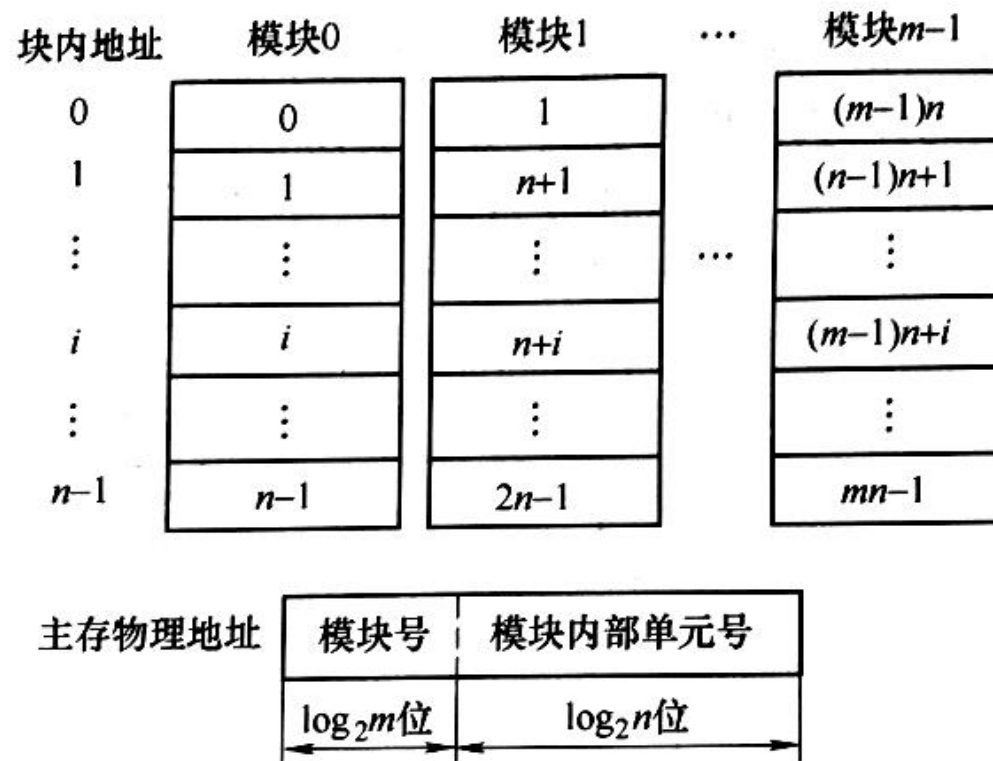
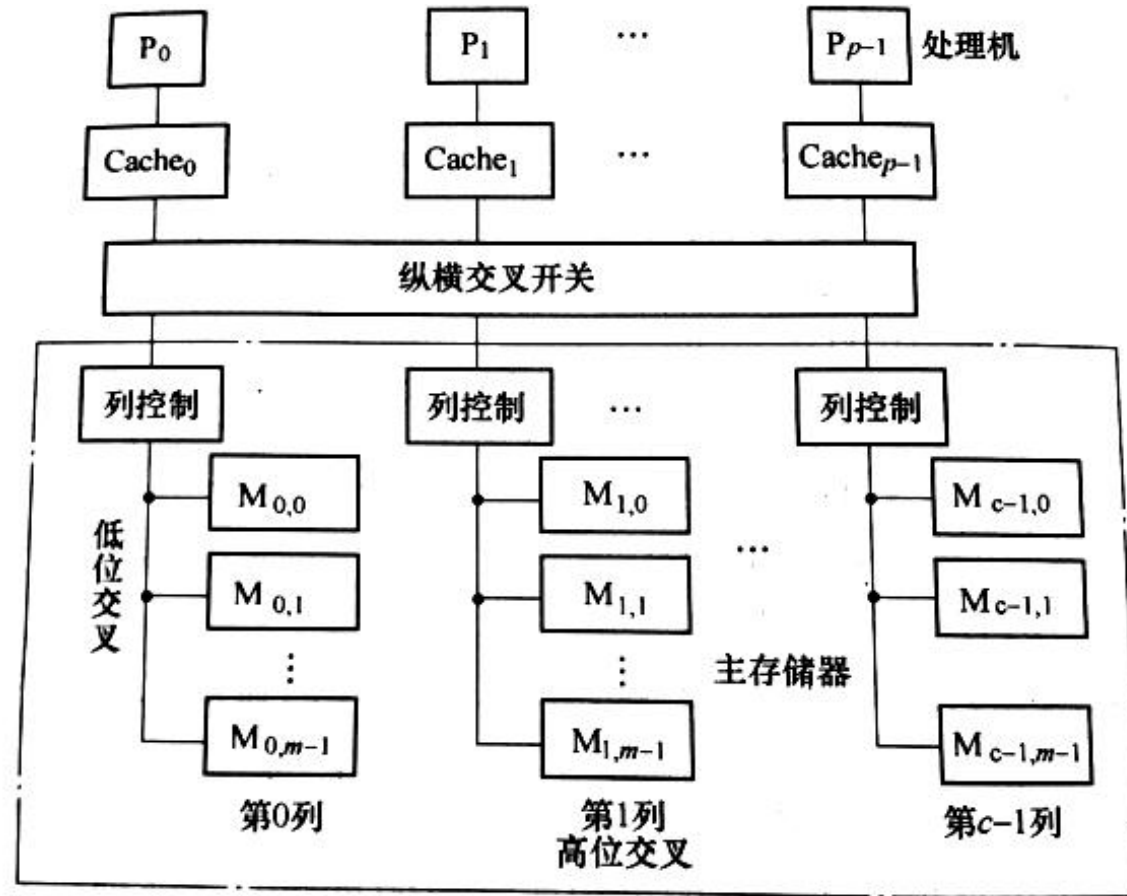


图7-9 m 个模块的地位交叉编址



7.1.2多处理机的硬件结构

3.存储器的组织



7.2 紧耦合多处理机多cache的一致性问题

本节主要内容：

多cache的一致性问题

解决多cache一致性的办法



7.2紧耦合多处理机多Cache的一致性问题

7.2.1多Cache的一致性问题的产生（简单了解）

由于每个处理机都有自己的专用Cache,当主存中同一个信息块在多个Cache中都有时，会出现多个Cache之间的相应信息块的内容不一致的问题。

7.2紧耦合多处理机多Cache的一致性问题

7.2.2多Cache的一致性问题解决办法（简答1910）

1.解决**进程迁移**引起的多Cache不一致性

禁止进程迁移或者在进程挂起时靠硬件方法，将改写过的信息块强制写回主存相应位置；

2.以**硬件**为基础实现多Cache的一致性

监视Cache协议法和目录表法；

3.以**软件**为基础实现多Cache的一致性

不把一些公用的可写数据存入Cache中。

7. 3多处理机的概念、问题和硬件结构

本节主要内容：

并行算法的研究思路

给出表达式，画出串行运算树和并行运算树，计算 P 、 T_1 、 T_p 、 S_p 和 E_p

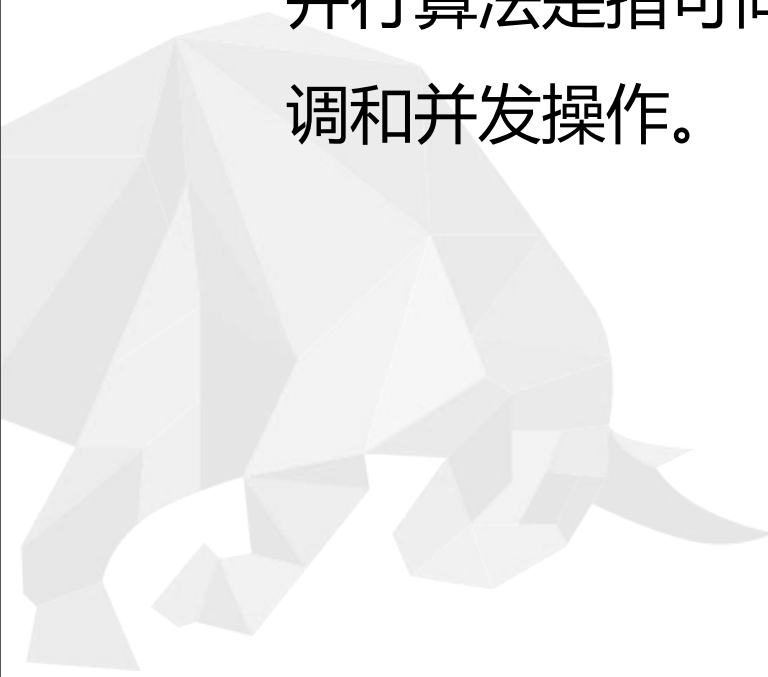
给出程序中的语句或指令，分析并行情况

给出计算式或高级语言源程序，加FORK、JOIN、GOTO语句

7.3.1 并行算法

1. 并行算法的定义和分类（简单了解）

并行算法是指可同时执行的多个进程的集合，各进程可相互作用、协调和并发操作。



7.3.1 并行算法

1. 并行算法的定义和分类（填空）

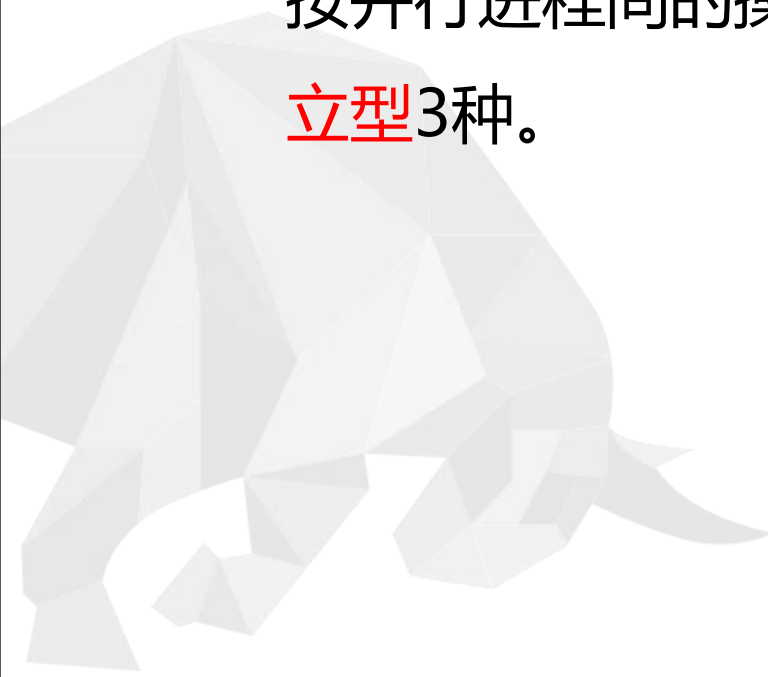
按运算基本对象，并行算法可分为数值型的和非数值型两类。



7.3.1 并行算法

1. 并行算法的定义和分类（单选、填空）

按并行进程间的操作顺序不同，并行算法又分为同步型、异步型和独立型3种。



7.3.1 并行算法

1. 并行算法的定义和分类（单选）

根据各处理机计算任务的大小（即任务粒度）不同，并行算法又分为细粒度、中粒度和粗粒度3种。

细粒度并行算法一般指**向量或循环级**的并行。

中粒度并行算法一般指**较大的循环级**并行，并确保这种并行的好处可以补偿因并行带来的额外开销。

粗粒度并行算法则一般是指**子任务级**的并行。



那么意气风发地
走在成功的道路上

真题练练手

1、按运算基本对象，并行算法可分为（ ）型和（ ）型两类。1904



那么意气风发地
走在成功的道路上

真题练练手

1、按运算基本对象，并行算法可分为（ ）型和（ ）型两类。1904

答案：数值型 非数值型

7.3.1 并行算法

2. 多处理机并行算法的研究思路

为了评价所提出的并行算法的性能效率，用 P 表示可并行处理的处理机机数；

用 T_p 表示 P 台处理机运算的级数，即树高；

用多处理机的加速比 S_p ，表示单处理机顺序运算的级数 T_1 与 P 台处理机并行运算的级数 T_p 之比；

用 E_p 表示 P 台处理机的设备利用率（效率），可见， $S_p \geq 1$ 时，会使 $E_p \leq 1$ ，即运算的加速总是伴随着效率的下降。

7.3.1 并行算法

2. 多处理机并行算法的研究思路

【1704真题】由霍纳法则给定的表达式如下: $E = a(bc + d(ef + g(h + ij)))$

利用减少树高的办法来加速运算,要求:

- 1、画出树形流程图;
- 2、确定 T_p 、 P 、 S_p 、 E_p 的值。

7.3.1 并行算法

2. 多处理机并行算法的研究思路

【1704真题】由霍纳法则给定的表达式如下: $E = a(bc + d(ef + g(h + ij)))$

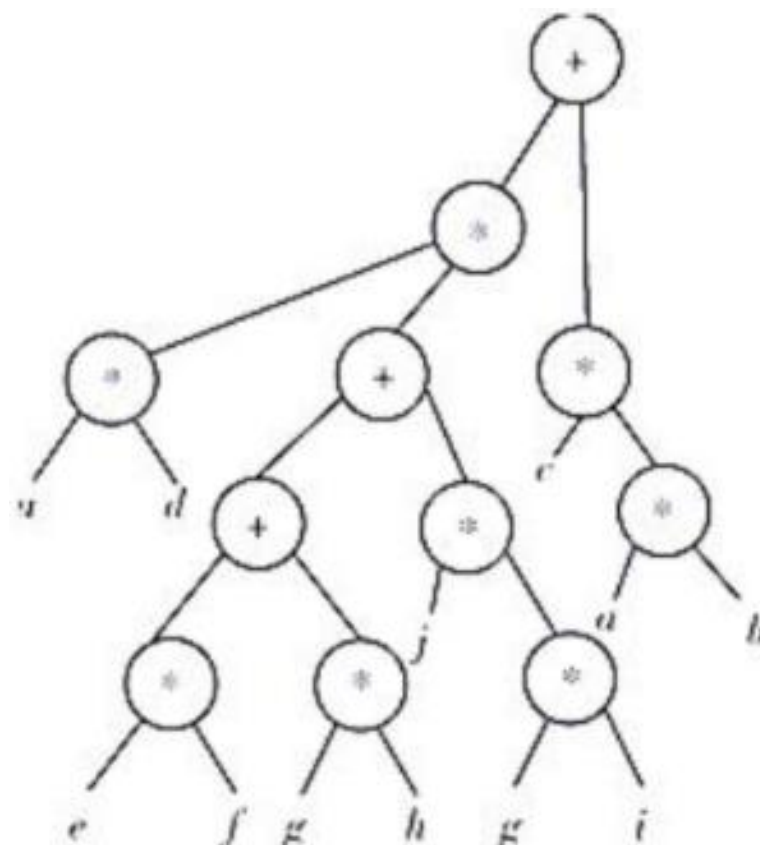
利用减少树高的办法来加速运算,要求:

1、画出树形流程图;

2、确定 T_p 、 P 、 S_p 、 E_p 的值。

1、若处理机为单处理机, $T_1 = 9$, 改成以下形式:
 $E = abc + ad(ef + gh + gij)$ 。树形流程图如图所示。

2、 $T_p = 5$ 、 $P = 3$ 、 $S_p = T_1/T_p = 9/5$ 、 $E_p = 3/5$ 。



7.3.2程序并行性的分析

1.数据相关

如果 P_i 的左部变量在 P_j 的右部变量集内，且 P_j 必须取出 P_i 运算的结果来作为操数，就称 P_j “数据相关”于 P_i 。

$$P_i \quad A = B + D$$

$$P_j \quad C = A * E$$

相当于流水中发生的“先写后读”相关。顺序串行运行的正确结果应当是

$$P_i \quad A_{\text{新}} = B_{\text{原}} + D_{\text{原}}$$

$$P_j \quad C_{\text{新}} = A_{\text{新}} * E_{\text{原}} = (B_{\text{原}} + D_{\text{原}}) * E_{\text{原}}$$

7.3.2程序并行性的分析

1.数据相关

如果让 P_i 和 P_j 并行， P_j 的 C 新成了 $A_{原} * E_{原}$ ，显然不是应有的结果，因此， P_i 和 P_j 是不能并行的。如果将 P_i 和 P_j 的执行顺序颠倒，交换串行，即先执行 P_j ，而后再执行 P_i ，很明显同样也得不到应有的正确结果。如果能够交换串行，就可以让空闲处理机先去执行 P_j ，从而有利于宏观上提高各个程序段间的并行，加快作业执行的速度，改进系统的运行效率。然而，有一种特殊情形，即当 P_i 和 P_j 服从交换律时，如

$$P_i \quad A = 2 * A$$

$$P_j \quad A = 3 * A$$

虽不能并行执行，却允许它们交换串行。最终 $A_{新} = 6 * A_{原}$ 和顺序执行的结果一致。

7.3.2程序并行性的分析

2.数据反相关

如果 P_j 的左部变量在 P_i 的右部变量集内，且当 P_i 未取用其变量的值之前，是不允许被 P_j 所改变的，就称 P_j “数据反相关”于 P_i 。例如

$$P_i \quad C = A + E$$

$$P_j \quad A = B + D$$

7.3.2程序并行性的分析

2.数据反相关

相当于流水中发生的“先读后写”相关。顺序串行运行的正确结果应是

$$P_i \quad C_{\text{新}} = A_{\text{原}} + E_{\text{原}}$$

$$P_j \quad A_{\text{新}} = B_{\text{原}} + D_{\text{原}}$$

可见，当 P_i 与 P_j 并行时，只要硬件上能保证 P_i 对相关单元 A 先读出，就能得到正确的结果。然而，若将 P_i 和 P_j 交换串行就成了

7.3.2程序并行性的分析

2.数据反相关

可见，当 P_i 与 P_j 并行时，只要硬件上能保证 P_i 对相关单元 A 先读出，就能得到正确的结果。然而，若将 P_i 和 P_j 交换串行就成了

$$P_j \quad A_{\text{新}} = B_{\text{原}} + D_{\text{原}}$$

$$P_i \quad C_{\text{新}} = A_{\text{新}} + E_{\text{原}} = B_{\text{原}} + D_{\text{原}} + E_{\text{原}}$$

而发生错误，所以是不能交换串行的。

7.3.2程序并行性的分析

3.数据输出相关

如果 P_i 的左部变量也是 P_j 的左部变量，且 P_j 存入其算得的值必须在 P_i 存入之后，则称 P_j “数据输出相关”于 P_i 。

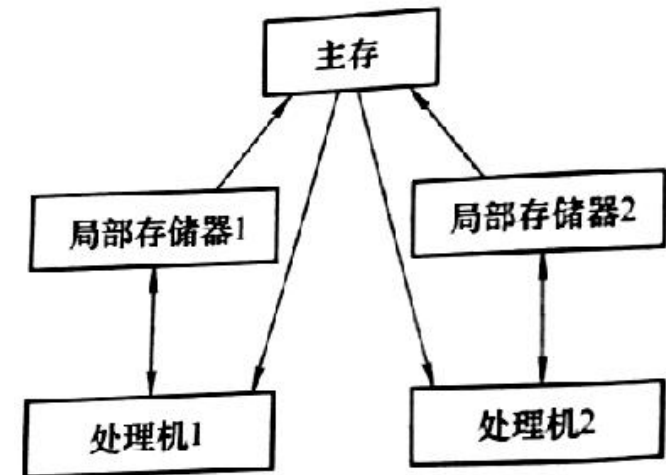


图7-13 能保证读写次序的多处理机结构

7.3.2程序并行性的分析

3.数据输出相关（单选）

结论：两个程序段之间若有先写后读的数据相关，不能并行，只在特殊情况下可以交换串行；若有先读后写的数据反相关，可以并行执行，但必须保证其写入共享主存时的先读后写次序，不能交换串行；若有写-写的数据输出相关，可以并行执行，但同样需保证其写入的先后次序，不能交换串行；若同时有先写后读和先读后写两种相关，以交换数据为目的时，必须并行执行，且读、写要完全同步，不许顺序串行和交换串行；若没有任何相关或仅有源数据相同时，可以并行、顺序串行和交换串行。

7.3.3 并行语言与并行编译

FORK语句的形式为FORK m，其中m为新进程开始的标号。执行FORK m语句时，派生出标号为m开始的新进程，具体为：准备好这个新进程启动和执行所必需的信息；

如果是共享主存，则产生存储器指针、映像函数和访问权数据；将空闲的处理机分配给派生的新进程，如果没有空闲处理机，则让它们排队等待；继续在原处理机上执行FORK语句的原进程。

7.3.3 并行语言与并行编译

与FORK语句相配合，作为每个并发进程的终端语句JOIN的形式为JOIN n ，其中 n 为并发进程的个数。JOIN语句附有一个计数器，其初始值为0。每当执行JOIN n 语句时，计数器的值加1，并与 n 比较。若比较相等，表明这是执行中的第 n 个并发进程经过JOIN 语句，于是允许该进程通过JOIN语句，将计数器清0,并在其处理机上继续执行后续语句；若比较不等，计数器的值仍小于 n ，表明此进程不是并发进程的最后一个，可让现在执行JOIN语句的这个进程先结束，把它所占用的处理机释放出来，分配给正在排队等待 其他任务。如果没有排队等待的任务，就让该处理机空闲。

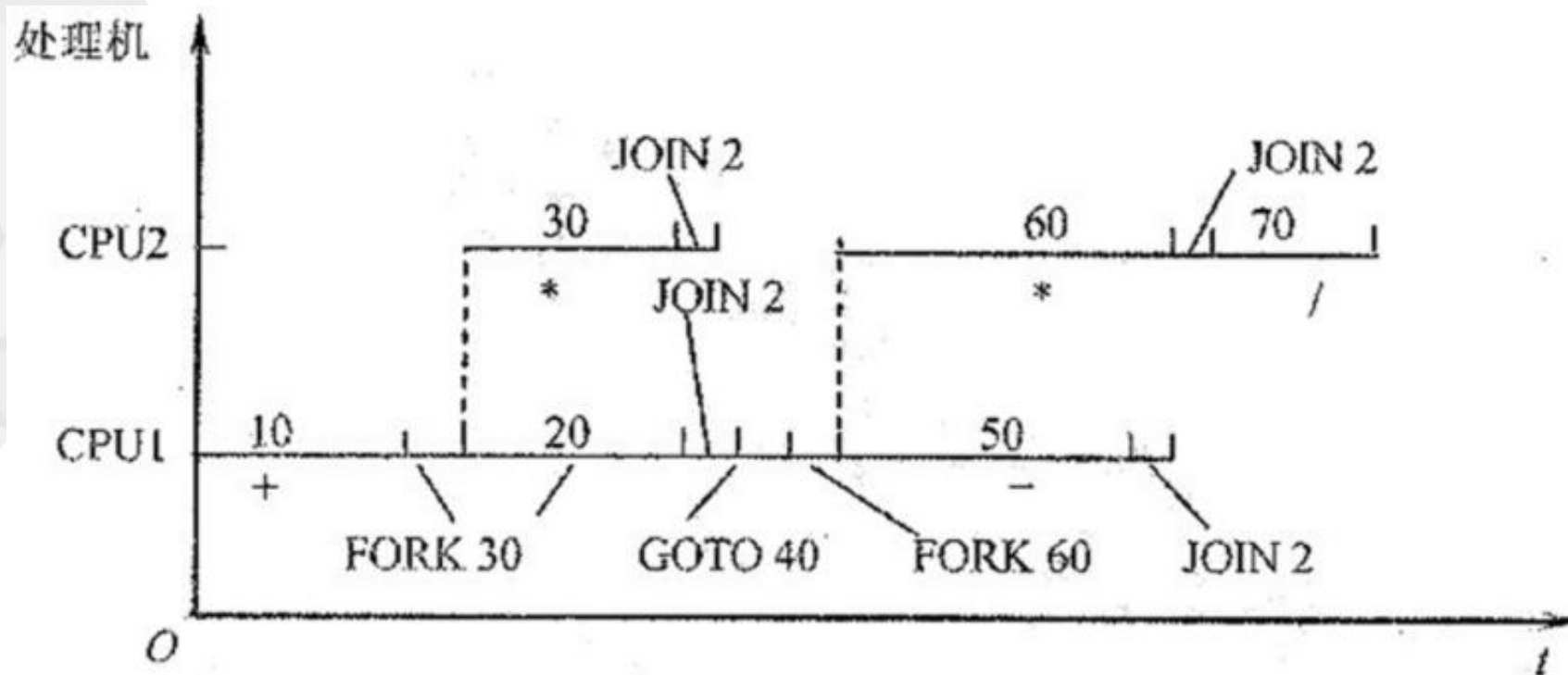
7.3.3 并行语言与并行编译

【1604真题】有以下FORK、JOIN写成的在多台处理机上并行执行的程序：1. 假设现为两台处理机,除法速度最慢,加、减法速度最快,画出该程序在两台处理机上运行时的资源时间图。

```
10 U = A + B
   FORK 30
20 V = U/B
   JOIN 2
   GOTO 40
30 W = A * U
   JOIN 2
40 FORK 60
50 X = W - V
   JOIN 2
   GOTO 70
60 Y = W * U
   JOIN 2
70 Z = X/Y
```

7.3.3 并行语言与并行编译

【1604真题】有以下FORK、JOIN写成的在多台处理机上并行执行的程序：1. 假设现为两台处理机,除法速度最慢,加、减法速度最快,画出该程序在两台处理机上运行时的资源时间图。



7.3.4多处理机的性能

任务粒度 (Task Granularity)的大小会显著影响多处理机的性能和效率。

任务粒度过小，辅助开销大，系统效率低；

任务粒度过大，并行度低，性能不会很高。

7.4 多处理机的操作系统

本节主要内容：

主从型操作系统的定义、特点和适用场合

各自独立型操作系统的定义、特点和应用场合

浮动型操作系统的定义、特点和应用场合

7.4多处理机的操作系统

分类：（填空）

多处理机操作系统有3类。它们是主从型（Master-Slave Configuration）、各自独立型（Separate Supervisor）及浮动型（Floating Supervisor）。

7.4多处理机的操作系统

7.4.1主从型操作系统

1.优点（简答）

主从型操作系统的结构比较简单；整个管理程序只在一个处理机上运行，除非某些需递归调用或多重调用的公用程序，一般都不必是可再入的；只有一个处理机访问执行表，不存在系统管理控制表格的访问冲突和阻塞，简化了管理控制的实现。所有这些均使操作系统能最大限度地利用已有的单处理机多道程序分时操作系统的成果，只需要对它稍加扩充即可。因此，实现起来简单、经济、方便，是目前大多数多处理机操作系统所采用的方式。

7.4多处理机的操作系统

7.4.1主从型操作系统

2.缺点（简答）

对主处理机的可靠性要求很高。一旦发生故障，很容易使整个系统瘫痪，这时必须由操作员干预才行。如果主处理机不是设计成专用的，操作员可用其他处理机作为新的主处理机来重新启动系统。整个系统显得不够灵活，同时要求主处理机必须能快速执行其管理功能，提前等待请求，以便及时为从处理机分配任务，否则将使从处理机因长时间空闲而显著降低系统的效率。即使主处理机是专门的控制处理机，如果负荷过重，也会影响整个系统的性能。特别是当大部分任务都很短时，由于频繁地要求主处理机完成大量的管理性操作，系统效率将会显著降低。

7.4多处理机的操作系统

7.4.2各自独立型操作系统

1.优点（简答）

很适应分布处理的模块化结构特点，减少对大型控制专用处理机的需求；某个处理机发生故障，不会引起整个系统瘫痪，有较高的可靠性；每台处理机都有其专用控制表格，使访问系统表格的冲突较少，也不会有许多公用的执行表，同时控制进程和用户进程一起进行调度，能取得较高的系统效率。

7.4多处理机的操作系统

7.4.2各自独立型操作系统

2.缺点（简答）

实现复杂。尽管每台处理机都有自己的专用控制表格，但仍有一些共享表格，会增加共享表格的访问冲突，导致进程调度的复杂性和开销的加大。某台处理机一旦发生故障，要想恢复和重新执行未完成的工作较困难。每台处理机都有自己专用的输入/输出设备和文件，使整个系统的输入/输出结构变换需要操作员干预。各处理机负荷的平衡比较困难。各台处理机需有局部存储器存放管理程序副本，降低了存储器的利用率。

7.4多处理机的操作系统

7.4.3浮动型操作系统

1.优点（简答——没考过）

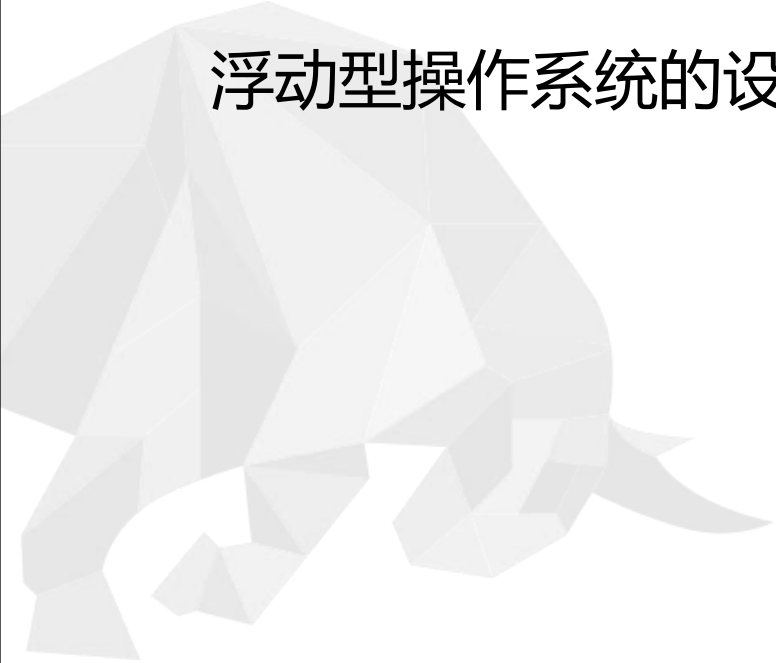
各类资源可以较好地做到负荷平衡。一些像I/O中断等非专门的操作可交由在某段时间最闲的处理机去执行。它在硬件结构和可靠性上具有分布控制的优点，而在操作系统的复杂性和经济性上则接近于主从型。如果操作系统设计得好，将不受处理机机数多少的影响，因而具有很高的灵活性。

7.4多处理机的操作系统

7.4.3浮动型操作系统

2.缺点（简答——没考过）

浮动型操作系统的设计最为困难。



7.5 多处理机的发展

本节主要内容：

多处理机发展的几种形式

大规模并行处理机MPP和机群系统的特点



7.5多处理机的发展

- 1) 分布式共享存储器多处理机
- 2) 对称多处理机
- 3) 多向量多处理机
- 4) 并行向量处理机
- 5) 大规模并行处理机 (MPP)
- 6) 机群系统

7.5多处理机的发展

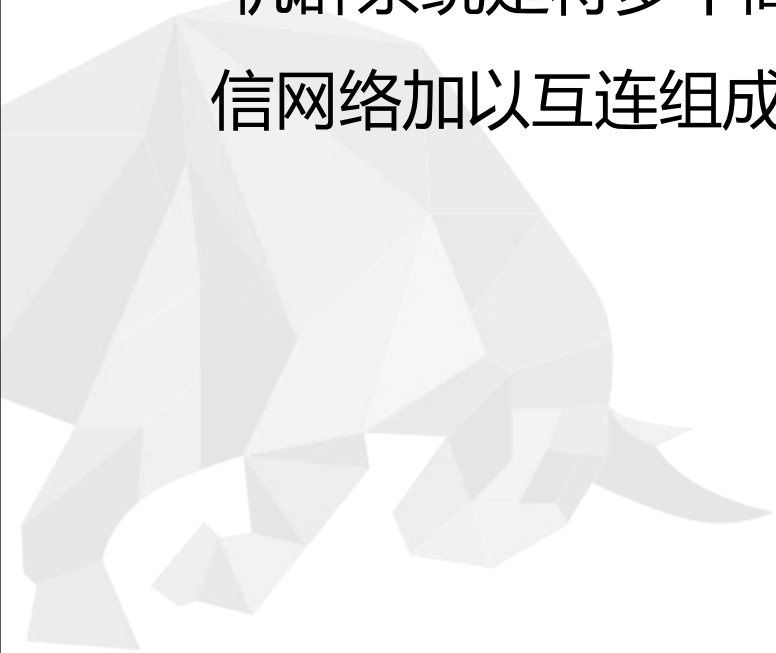
计算机系统的3T性能目标（补充知识点——填空）

- 1 TFLOPS的计算能力、
- 1 Tbyte的主存容量
- 1 Tbyte / s的I / O带宽。

7.5多处理机的发展

机群系统（简单了解）

机群系统是将多个高性能的工作站或高档微型计算机，使用高速的通信网络加以互连组成的系统。



7.5多处理机的发展

机群系统比起传统的并行处理系统有如下明显的优点：（单选、简答）

- 1) 系统有高的性能价格比。
- 2) 系统的开发周期短。
- 3) 系统的可扩展性好。
- 4) 系统的资源利用率高。
- 5) 用户投资风险小。
- 6) 用户编程方便。

（口诀：简单好用价格低）



那么意气风发地
走在成功的道路上

真题练练手

2、与传统的并行处理系统相比，机群系统所具有的特点不包括()1204

A:用户编程方便

B:系统的开发周期长

C:系统的资源利用率高

D:系统的可扩展性好



那么意气风发地
走在成功的道路上

真题练练手

2、与传统的并行处理系统相比，机群系统所具有的特点不包括()1204

A:用户编程方便

B:系统的开发周期长

C:系统的资源利用率高

D:系统的可扩展性好



那么意气风发地
走在成功的道路上

真题练练手

3、计算机系统的3T性能目标是（ ）的计算能力、（ ）的主存容量和1 Tbyte / s的I / O带宽。 0604



那么意气风发地
走在成功的道路上

真题练练手

3、计算机系统的3T性能目标是（ ）的计算能力、（ ）的主存容量和1 Tbyte / s的I / O带宽。 0604

答案：1 TFLOPS 1 Tbyte

第8章 数据流计算机和归约机

第8章 数据流计算机和归约机



数据流计算机



归约机

8.1 数据流计算机

本节主要内容：

工作原理

构形

特点

近年来的发展

8.1.1 数据驱动的概念

Von Neumann型计算机的基本特点是在程序计数器集中控制下，顺次地执行指令，因此它是以**控制流**（Control Flow）方式工作的。

8.1.1 数据驱动的概念

(填空)

数据驱动的数据流方式中，数据令牌是一种表示某一**操作数**或**参数**已准备就绪的标志。

8.1.1 数据驱动的概念

控制驱动的控制流方式的特点是：（简答）

通过访问共享存储单元让数据在指令之间传递；指令执行的顺序性隐含于控制流中，但却可以显式地使用专门的控制操作符来实现并行处理；指令执行的顺序受程序计数器控制，换句话说，是受控制令牌所支配的。数据驱动的数据流方式则不同，它没有通常的共享变量的概念，即没有共享存储数据的概念；指令执行序只受指令中数据相关性的制约；数据是以数据令牌方式直接在指令之间传递的。

8.1.1 数据驱动的概念

需求驱动计算模型

数据驱动计算，其操作是按输入数据可用性决定的次序进行的。

需求驱动计算，其操作则按数据需求所决定的次序进行。

8.1.1 数据驱动的概念

(填空)

从语义上讲，数据流是基于异步性 (Asynchrony) 和函数性 (Functionality) 的一种计算模型。

8.1.3数据流计算机的结构

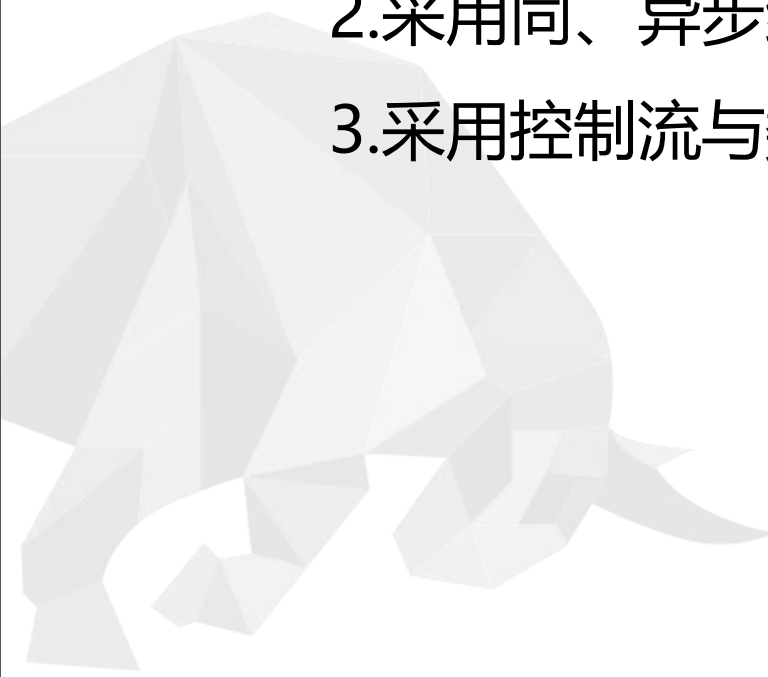
(填空)

根据对数据令牌处理的方式不同，可以把数据流计算机的结构分成静态和动态两类。



8.1.5数据流计算机的进展

- 1.采用提高并行度等级的数据流计算机
- 2.采用同、异步结合的数据流计算机
- 3.采用控制流与数据流相结合的数据流计算机



8.2归约机

(填空)

归约机和数据流计算机一样，都是基于**数据流**的计算模型，只是其采用的驱动方式不同。

数据流计算机采用**数据驱动**，执行的操作序列取决于输入数据的可用性；

归约机则是**需求驱动**，执行的操作序列取决于对数据的需求，对数据的需求又来源于函数式程序设计语言对表达式的归约（Reduction）。

8.2归约机（简答）

- 1)归约机应当是面向函数式语言，或以函数式语言为机器语言的非Neumann型机器，其内部结构应不同于Neumann型机器。
- 2)具有大容量物理存储器并采用大虚存容量的虚拟存储器，具备高效的动态存储分配和管理的软、硬件支持，满足归约机对动态存储分配及所需存储空间大的要求。
- 3)处理部分应当是一种有多个处理器或多个处理机并行的结构形式，以发挥函数式程序并行处理的特长。
- 4)采用适合于函数式程序运行的多处理器（机）互连的结构，最好采用树形方式的互连结构或多层次复合的互连结构形式。
- 5)为减少进程调度及进程间的通信开销，尽量把运行进程的结点机紧靠该进程所需用的数据安排，并使运行时需相互通信的进程所占用的处理机也靠近，让各处理机的负荷平衡。



那么意气风发地
走在成功的道路上

真题练练手

1、数据驱动的数据流方式中，数据令牌是一种表示某一（ ）或（ ）已准备就绪的标志。1904



那么意气风发地
走在成功的道路上

真题练练手

1、数据驱动的数据流方式中，数据令牌是一种表示某一（ ）或（ ）已准备就绪的标志。1904

答案：操作数 参数



尚德机构

▶ 答疑时间 ◀





尚德机构

▶ THANK YOU ◀

