第6章
并行处理机与互联网络

郑宏 副教授
计算机学院
北京理工大学

学习内容

- 6.1 并行处理机原理
- 6.2 阵列处理机的并行算法
- 6.3 互连网络基本概念
- 6.4 互连网络的种类
- 6.5 并行存储器的无冲突访问

6.1 并行处理机原理

已经学习的并行性的一些知识。包括：

- 并行性包括同时性和并发性
- 开发并行性的途径主要有三个：
  - 时间重叠
  - 资源重复
  - 资源共享

6.1 并行处理机原理

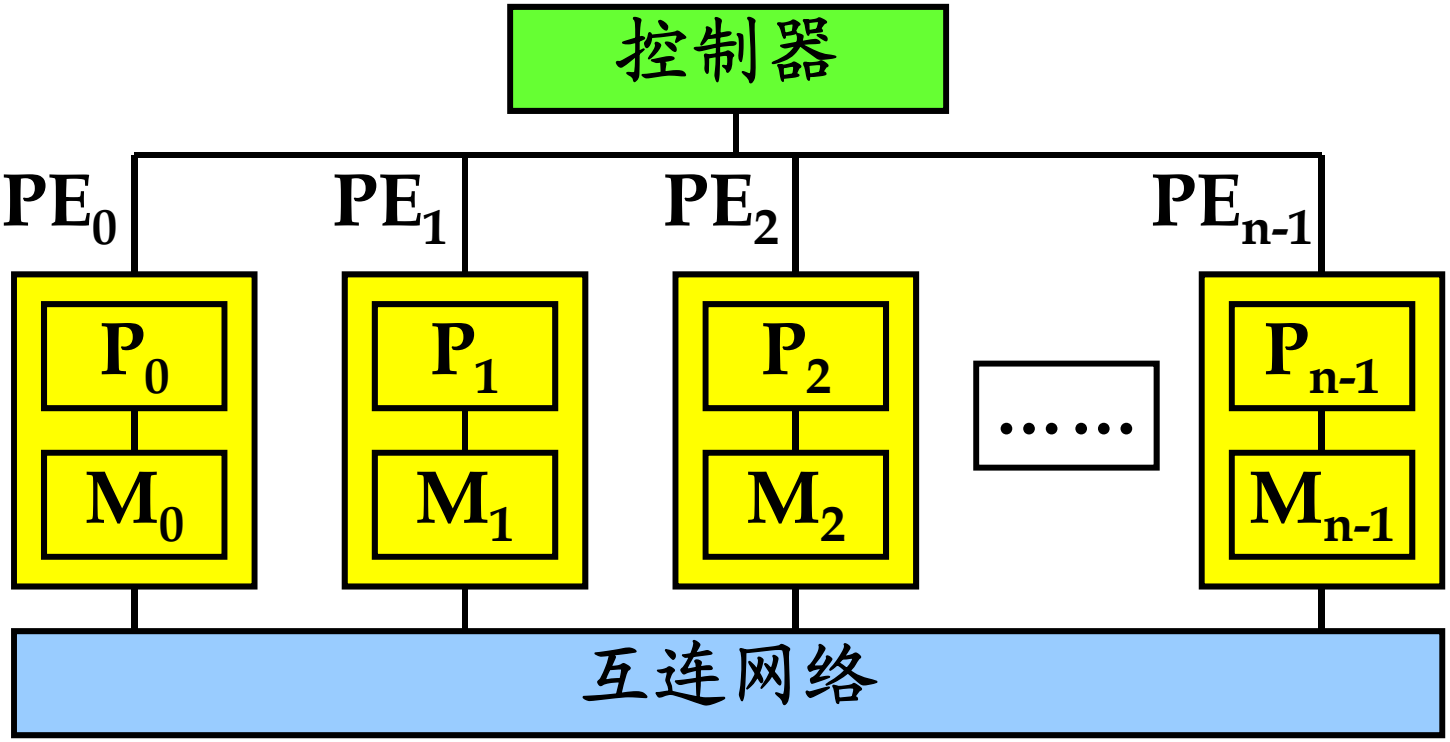
- 流水线主要采取时间重叠技术提高并行性，但主要局限在功能部件上，并行性的发挥有限。
- 为实现更高一级的并行，必须摆脱单处理机的限制，发展各种不同耦合度的并行处理计算机结构。
- 并行处理机是并行处理计算机中的一种重要结构，它主要通过资源重复实现同时性并行。

6.1.1 并行处理机定义及特点

- 并行处理机也称为阵列处理机。
- 定义：

  重复设置多个同样的处理单元（PE），并将它们按照一定方式互相连接，在统一的控制部件（CU）作用下，各自对分配的数据并行地完成同一条指令所规定的操作，实现操作级并行的SIMD计算机。

6.1.1 并行处理机定义及特点



H J Siegel提出的并行处理机模型

由五个部分组成：

一个控制器CU，多个处理单元PE，多个存储器模块M， 一个互连网络ICN， 一台输入输出处理机IOP

6.1.1 并行处理机定义及特点

- 从CU看，指令是串行执行的。
- 从PE看，数据是并行处理的。
- PE
  - 不带指令控制部件的算术运算部件
  - 使用按地址访问的随机存贮器
- 按照佛林分类法，它属于SIMD计算机。

6.1.1 并行处理机定义及特点

- 特点：
  - (1)速度快，而且潜力大。
  - (2)模块性好，生产和维护方便。
  - (3)可靠性高，容易实现容错和重构。
  - (4)效率低（与流水线处理机、向量处理机等比较）。通常作为专用计算机，因此，在很大程度上依赖于并行算法。

6.1.1 并行处理机定义及特点

- 特点：
  - (5)依赖于互连网络和并行算法。互连网络决定了PE之间的连接模式，也决定了并行处理机能够适应的算法。
  - (6)需要有一台高性能的标量处理机。使向量计算与标量计算平衡。

6.1.2 阵列处理机基本构形与特点

- 根据存贮器组成方式的不同，阵列处理有两种基本构形：
  - 分布式存贮器的阵列处理机
  - 集中式共享存贮器的阵列处理机

1. 分布式存贮器的阵列处理机

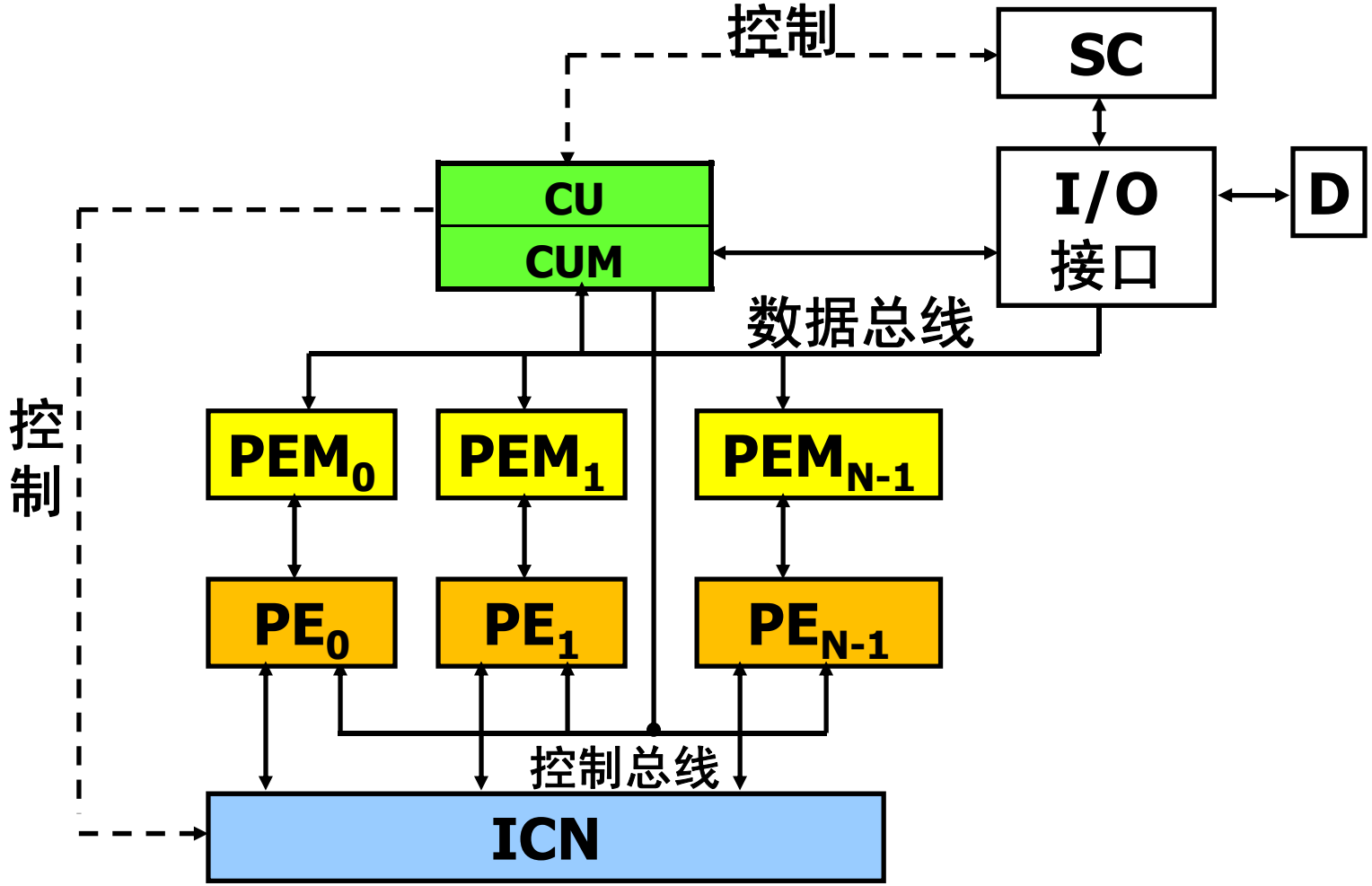


图 具有分布式存贮器的阵列处理机构形

1. 分布式存贮器的阵列处理机

- 每个PE有自己的局部存贮器PEM；
- PEM只能被本PE访问；
- 整个系统在CU控制下运行；
- 所有指令都在CU中译码。译码后把适合于并行处理的向量类指令“播送”给各个PE，让处于“活跃”的那些PE并行地执行。因此CU中的指令基本上是单指令流；

1. 分布式存贮器的阵列处理机

- CU也可以采用流水线工作方式，进一步让多条向量指令在时间上重叠执行；
- 为了有效地对向量数据进行高速处理，要求能把数据合理地预分配到各个PEM中；
- PE之间通过互连网络ICN来交换数据。

1. 分布式存贮器的阵列处理机

- 这种构形是**SIMD的主流**
  - 比较容易构成**MPP**（**Massively Parallel Processor**），几十万个**PE**。
  - 必须依靠并行算法来提高**PE**的利用率。因此，应用领域很有限。
- 典型的机器有：
  - **60**年代研制、**1972**年生产的**ILLIAC** Ⅳ
  - **1979**年研制成功的巨型并行处理机**MPP**
  - **1980**年生产的分布式阵列处理机**DAP**等

2. 集中式共享存贮器的阵列处理机

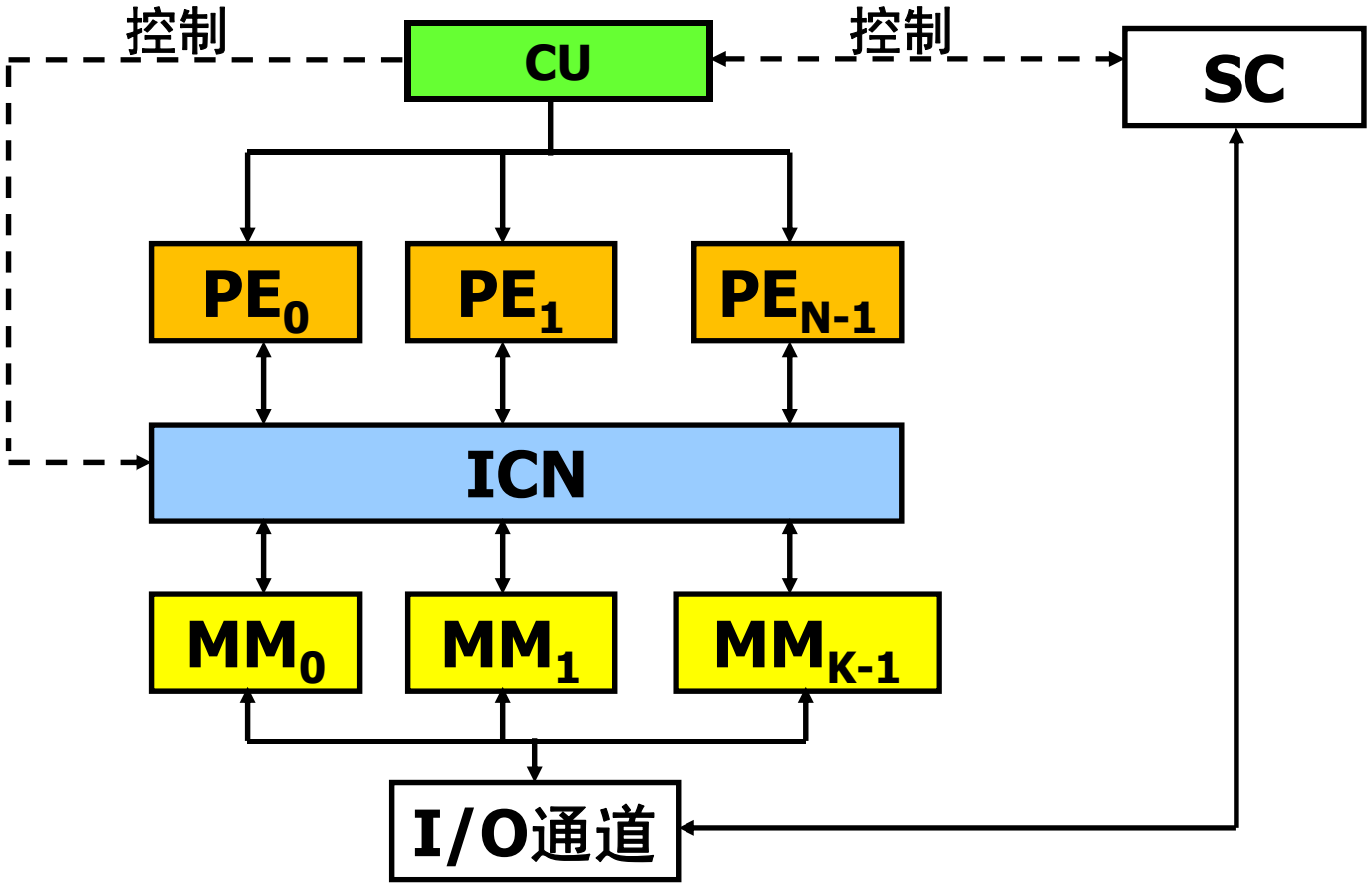


图 集中式共享存贮器的阵列处理机构形

2. 集中式共享存贮器的阵列处理机

- 与分布式存贮器的阵列处理机相比，有两处不同 (1/2)：
  - (1)系统存贮器由K个存贮体构成，经互连网络ICN为全部N个PE所共享。为使各个PE对长度为N向量的各个元素同时处理，存贮体数 K 应等于或大于 PE 数 N，即 K >= N。为避免访存冲突，需要合适的算法将数据合理地分配到各个存贮体中。

2. 集中式共享存贮器的并行处理机

- 与分布式存贮器的并行处理机相比，有两处不同 (2/2)：
  - (2)ICN的作用不同。ICN用于在PE和MM之间构建数据通路，使各个PE可以高速、灵活、动态地与不同的存贮体相连。因此有时被称为对准网络。
- 典型的机器有：
  - BSP(16个PE通过一个16×17的对准网络访问17个共享存储器模块)等。

3. 阵列处理机特点

- (1)阵列机以单指令流多数据流方式工作。
- (2)阵列机采用资源重复方法引入空间因素，即在系统中设置多个相同的处理单元来实现并行性，这与利用时间重叠的向量流水处理机是不一样的。此外，阵列机利用并行性中的同时性，所有处理单元必须同时进行相同的操作。
- (3)它使用简单而规整的ICN来确定PE之间的连接模式。ICN限定了并行处理机适用的解题算法的类型，也对整个系统的性能产生明显影响。因此，ICN是设计重点。

3. 阵列处理机特点

- (4)阵列机是以某一类算法（如图像处理）为背景的专用计算机。这是由于阵列机中通常都采用简单、规整的互连网络来实现处理单元间的连接操作，从而限定了其所适用的求解算法类别。
- (5)阵列机的研究必须与并行算法的研究密切结合，以使其求解算法的适应性更强一些，应用面更广一些。

3. 阵列处理机特点

- (6)从处理单元来看，由于结构都相同，因而可将阵列机看成是一个同构型并行机。但其控制器实质上是一个标量处理机，而为了完成I/O操作及操作系统管理，尚需一个前端机，因此，实际的阵列机系统是由上述三部分构成的一个异构型多处理机系统。

6.2 阵列处理机的并行性算法

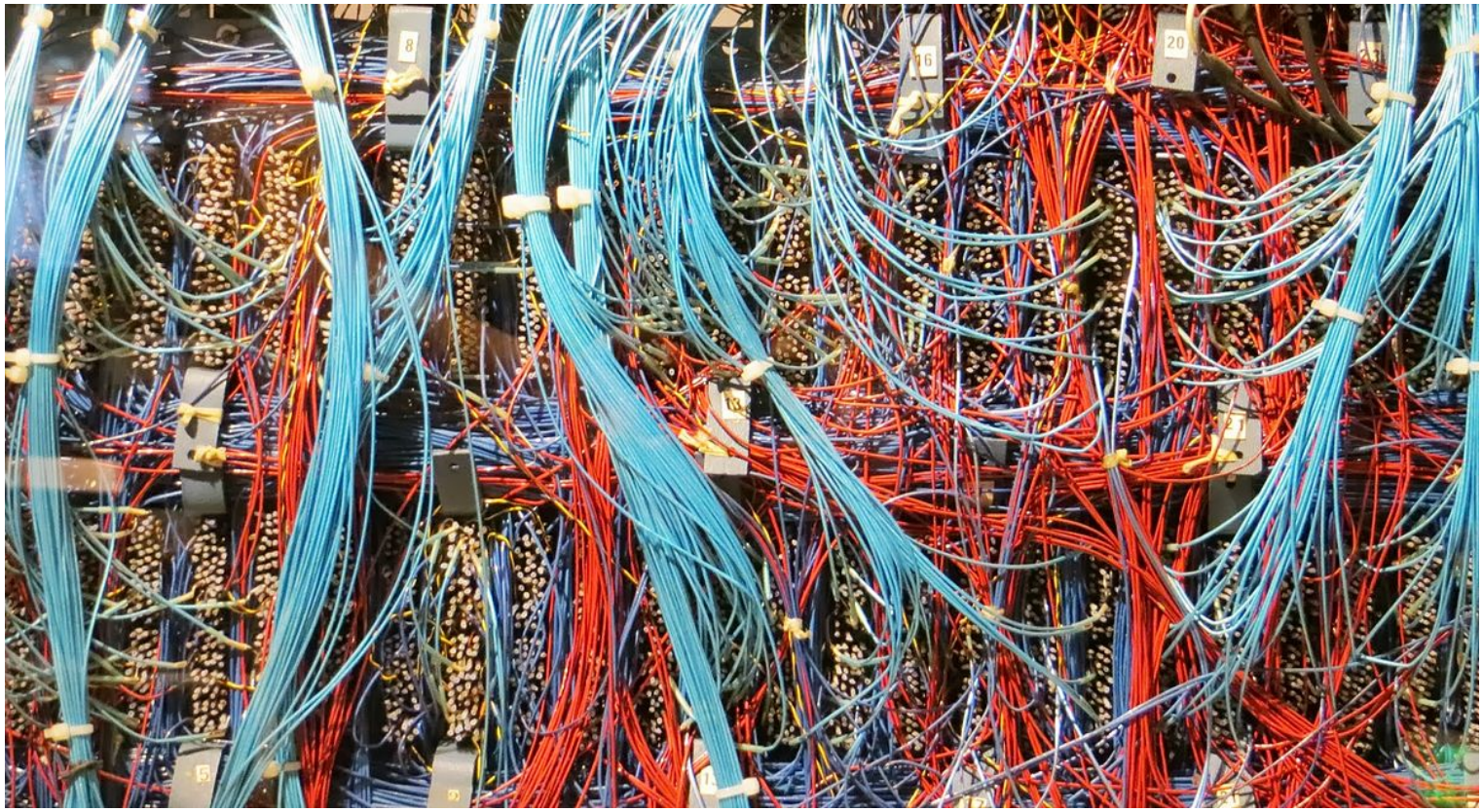
- 下面以 **ILLIAC IV** 为例介绍阵列处理机的并行性算法。
- **ILLIAC IV** 是最先采用SIMD结构的并行机。
- 由美国宝来(Burroughs)公司和伊利诺依大学于1965年开始研制，1975年实际投入运行。

1. ILLIAC IV系统组成

- ILLIAC IV系统由两大部分组成：
  - ILLIAC IV阵列
  - ILLIAC IV输入/输出系统
- 实际上，它是由3种类型处理机组成的多机系统：
  - (1) 专门对付数组运算的处理单元阵列 (processing element array)；
  - (2) 阵列控制器 (array control unit)，它既是处理单元阵列的控制部分，又可以视为一台相对独立的小型标量处理机；
  - (3) 一台标准的Burroughs B6700计算机，担负ILLIAC IV输入/输出系统和操作系统管理功能。

1. ILLIAC IV系统组成

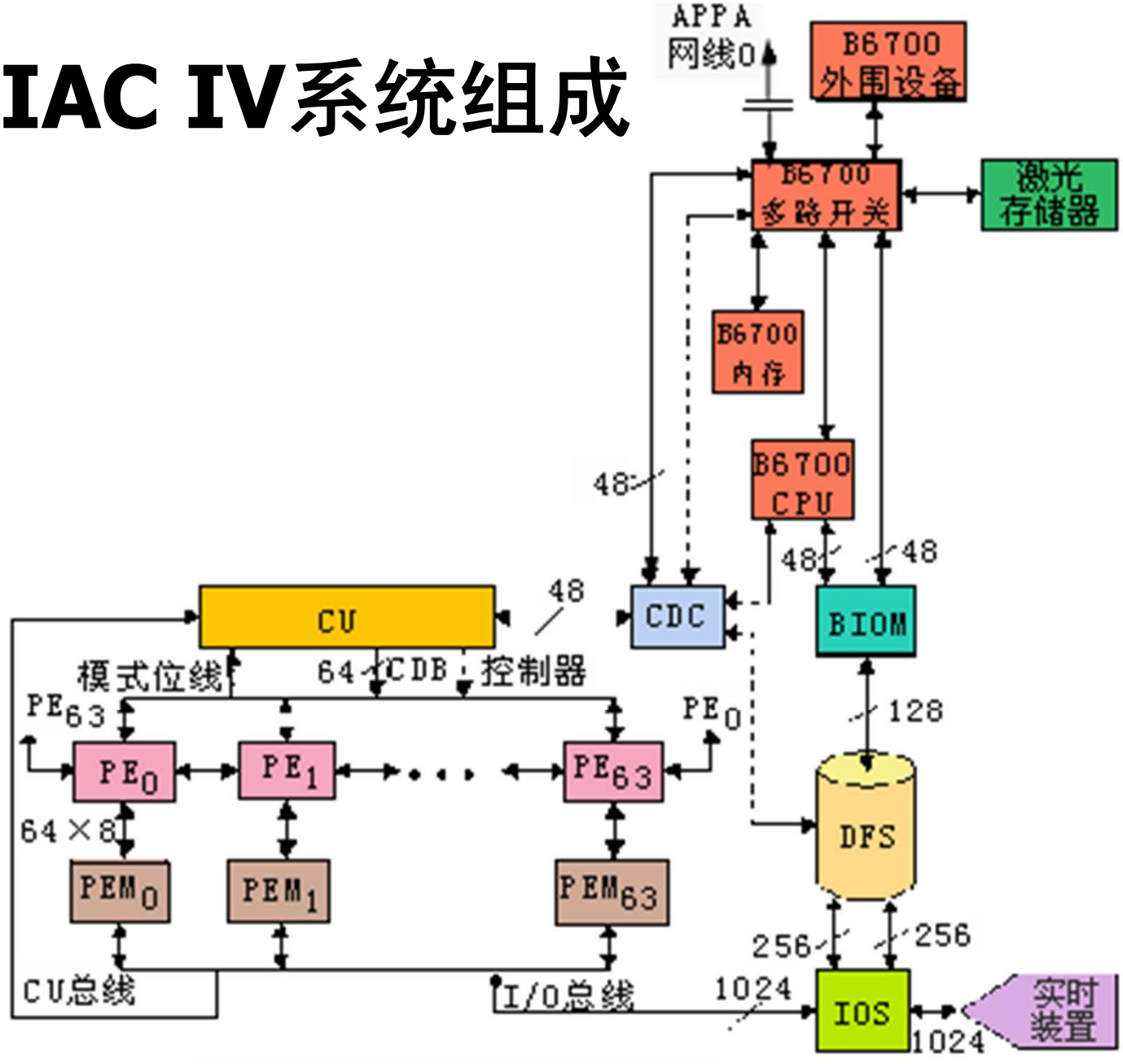
- 处理单元阵列 (PU)



1. ILLIAC IV系统组成

- 处理单元阵列 (PU)

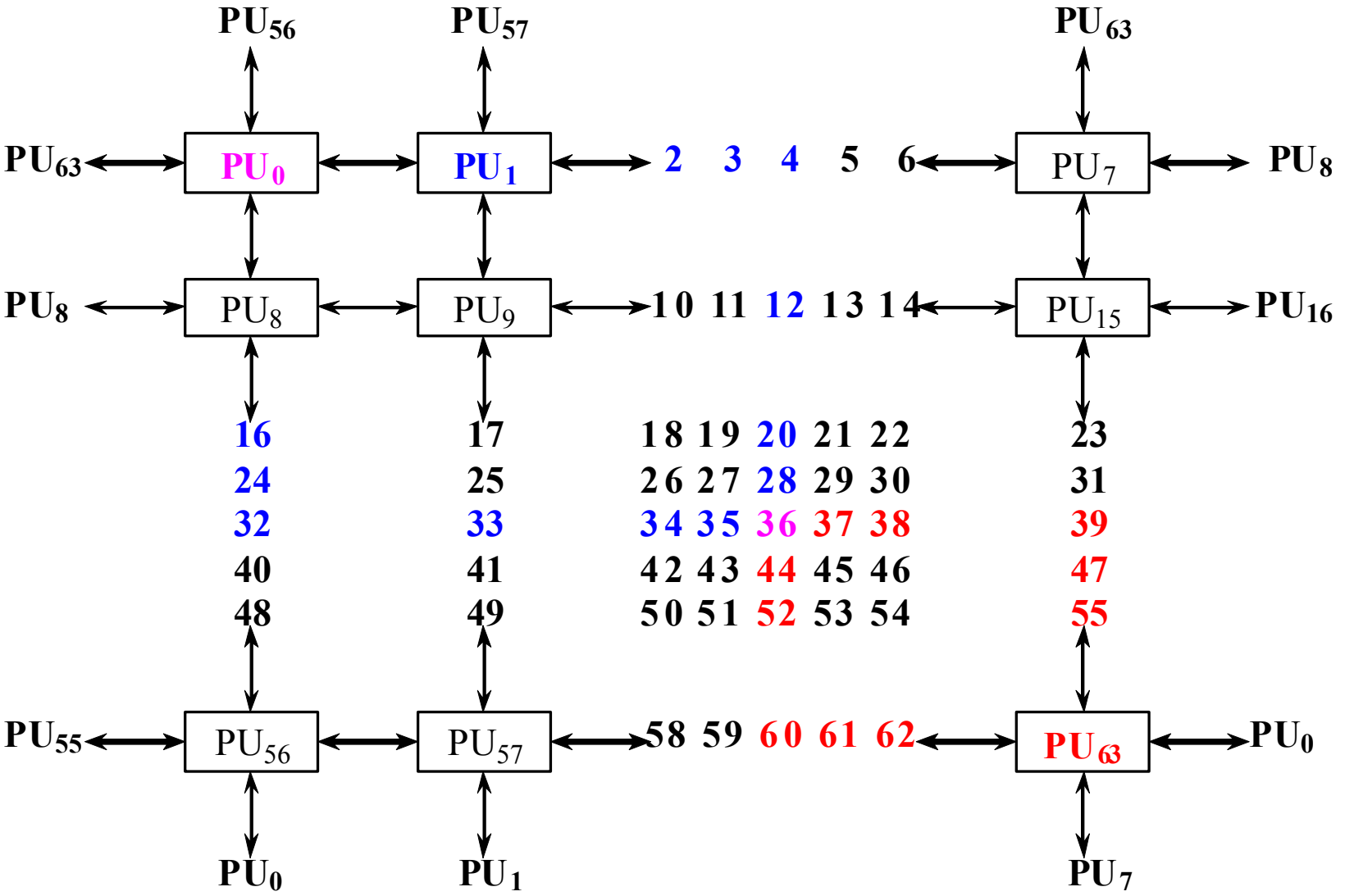
1. ILLIAC IV系统组成



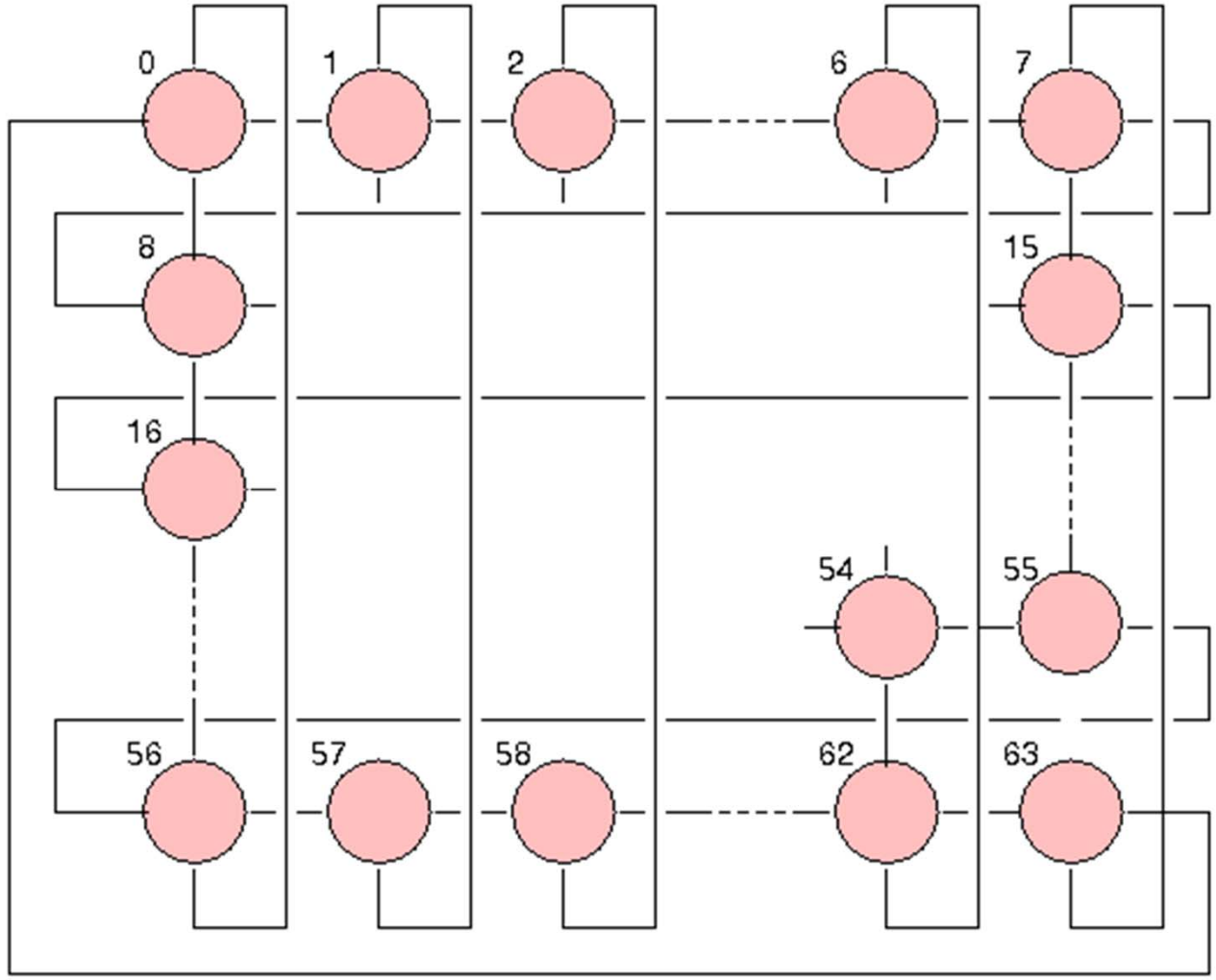
ILLIAC IV阵列

- 由 8×8=64 个PU组成。每个PU由处理部件PE和它的局部存储器PEM组成。
- 每一个PU_i 只和它的东、西、南、北四个近邻直接连接。{PU_{i+1} mod 64、PU_{i-1} mod 64、PU_{i+8} mod 64、PU_{i-8}mod 64}
- 南北方向上同一列的PU连成一个环，东西方向上构成一个闭合螺线。

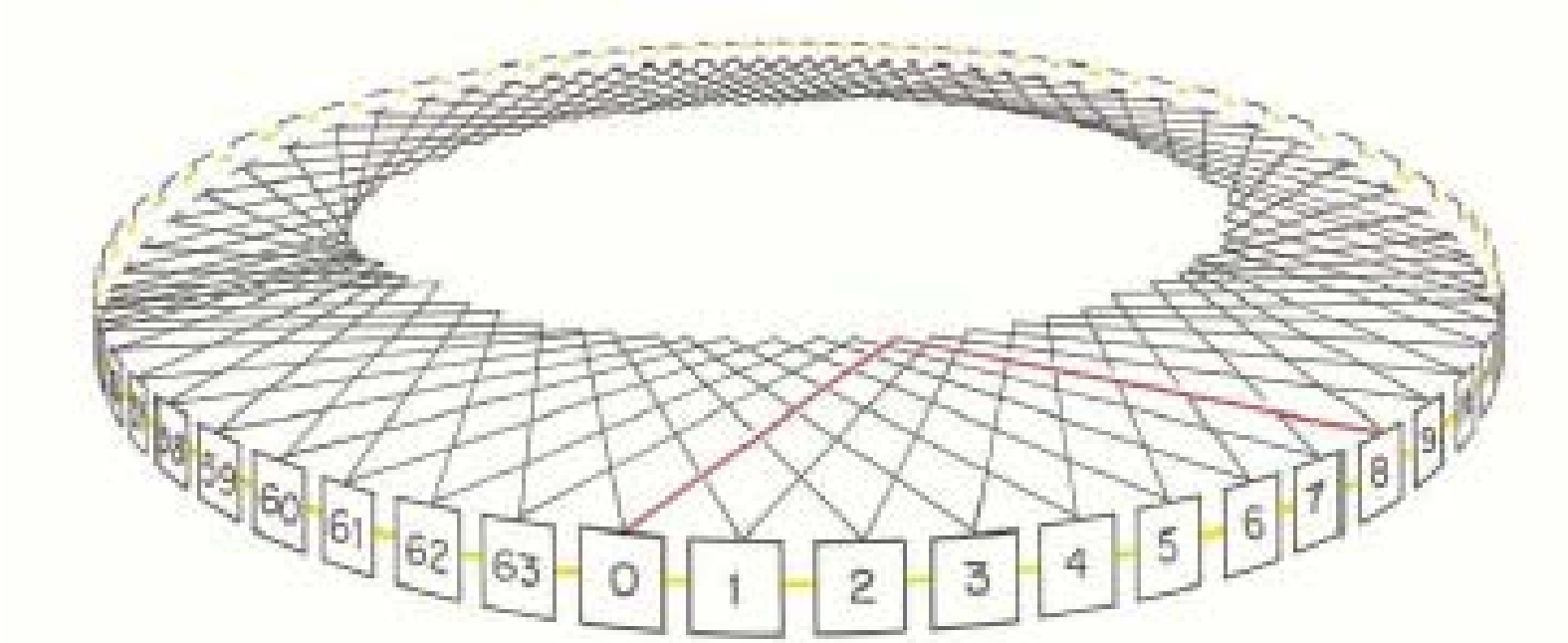
ILLIAC IV阵列


PU56
PU57
PU63
PU63
PU0
PU1
2 3 4 5 6
PU7
PU8
PU8
PU8
PU9
10 11 12 13 14
PU15
PU16
16
17
18 19 20 21 22
23
24
25
26 27 28 29 30
31
32
33
34 35 36 37 38
39
40
41
42 43 44 45 46
47
48
49
50 51 52 53 54
55
PU55
PU56
PU57
58 59 60 61 62
PU63
PU0
PU0
PU1
PU7

ILLIAC IV阵列



ILLIAC IV阵列



The above figure shows this end-around routing connection schematically. In addition to the neighbor-to neighbor linkages which form the PE's into a ring, there are connections of PE's eight apart such that data can leapfrog intermediate PE's when the distance to be covered is large.

ILLIAC IV阵列

采用闭合螺线最短距离不超过7步，而普通网格最短距离不超过8步。

- 例如：从 PU_0 到 PU_{36} 的距离
  - 采用普通网格必须 8 步：

$PU_0 \rightarrow PU_1 \rightarrow PU_2 \rightarrow PU_3 \rightarrow PU_4 \rightarrow PU_{12} \rightarrow PU_{20} \rightarrow PU_{28} \rightarrow PU_{36}$ 或

$PU_0 \rightarrow PU_8 \rightarrow PU_{16} \rightarrow PU_{24} \rightarrow PU_{32} \rightarrow PU_{33} \rightarrow PU_{34} \rightarrow PU_{35} \rightarrow PU_{36}$ 或

……

（等于8步的很多，大于8步的更多）

ILLIAC IV阵列

- 例如：从 PU_0 到 PU_{36} 的距离
  - 采用闭合螺旋线，只需要 7 步：

    $PU_0 \rightarrow PU_{63} \rightarrow PU_{62} \rightarrow PU_{61} \rightarrow PU_{60} \rightarrow PU_{52} \rightarrow PU_{44} \rightarrow PU_{36}$ 或

    $PU_0 \rightarrow PU_{63} \rightarrow PU_{55} \rightarrow PU_{47} \rightarrow PU_{39} \rightarrow PU_{38} \rightarrow PU_{37} \rightarrow PU_{36}$ 或

    ……

推广到一般的 n×n 个单元组成的二维阵列中，任意两个处理单元之间的最短距离不会超过(n-1)步。

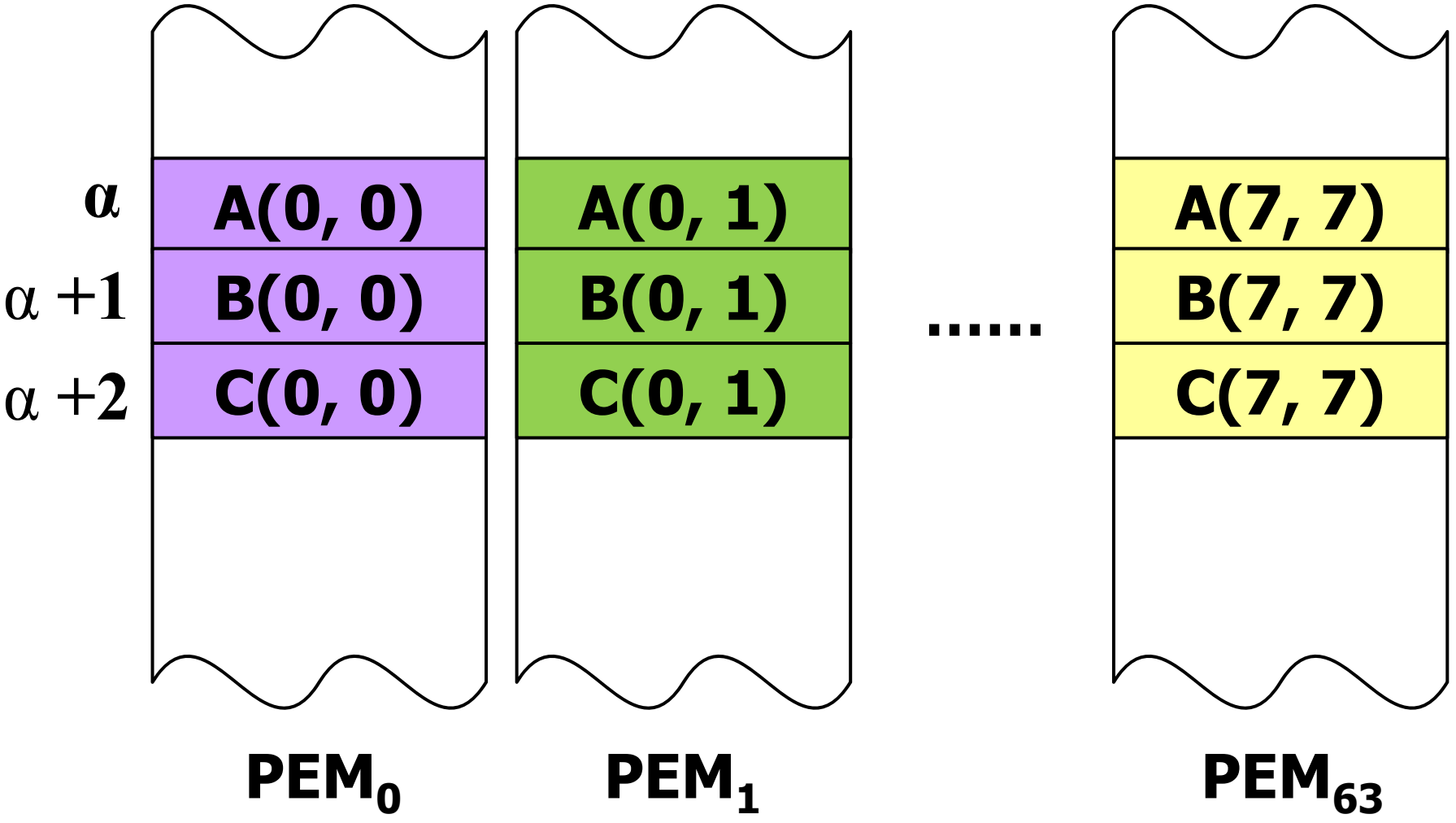
2. 阵列处理机并行算法举例

- 要发挥SIMD计算机的效率，特别依赖于并行算法。并行算法的一个关键问题是要提高向量化的程度。
- 在设计并行算法时，要特别注意数据在多个存储模块之间的分布，要解决好访问存储器的冲突问题。
- 互连网络并不能提供所有处理单元之间的连接，因此，并行算法要充分利用互连网络的结构。

(1) 矩阵加

- 在阵列处理机上解决矩阵加法问题是最简单的一维情形。
- 假定两个 8×8 的矩阵A和B相加，则只需把A和B居于相似位置的一对分量存放在同一PEM内，且令A的分量在全部 64 个PEM中存放的单元地址均为α， B的分量单元地址码均为α+1，而C＝A＋B的结果分量均放在地址码为α+2的单元内。

(1) 矩阵加



(1) 矩阵加

- 只需用3条 ILLIAC IV 的汇编指令就可一次实现矩阵相加。

```
LDA  ALPHA      ;  全部(α)由PEM_i 送PE_i 的累加器RGA_i
ADRN ALPHA+1    ;  全部(α+1)与(RGA_i)进行浮点规舍加，  结果送RGA_i
STA  ALPHA+2    ;  全部(RGA_i)由PE_i送PEM_i的α+2单元
```

其中，0 ≤ i ≤ 63

全并行
Sp = 64

(2) 矩阵乘

- 矩阵乘是二维数组运算，故它比矩阵加要复杂一些。
- 设A、B 和C 为3个 8×8 的二维矩阵。若给定A和B，则为计算C=A*B的 64 个分量，可用下列公式

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

其中， $0 \le i \le 7$ 且 $0 \le j \le 7$

(2) 矩阵乘

- 如果在SISD计算机上求解这个问题，则可执行下列FORTRAN程序：

```
DO 10 I=0,7
   DO 10 J=0, 7
   C(I,J)=0
   DO 10 K=0, 7
10 C(I,J)=C(I,J)+A(I,K)*B(K,J)
```

需要经过I、J、K三重循环完成。每重循环执行 8 次，总共需要512次乘、加的时间，此外每次还应包括执行循环控制、判别等其他操作需花费的时间。

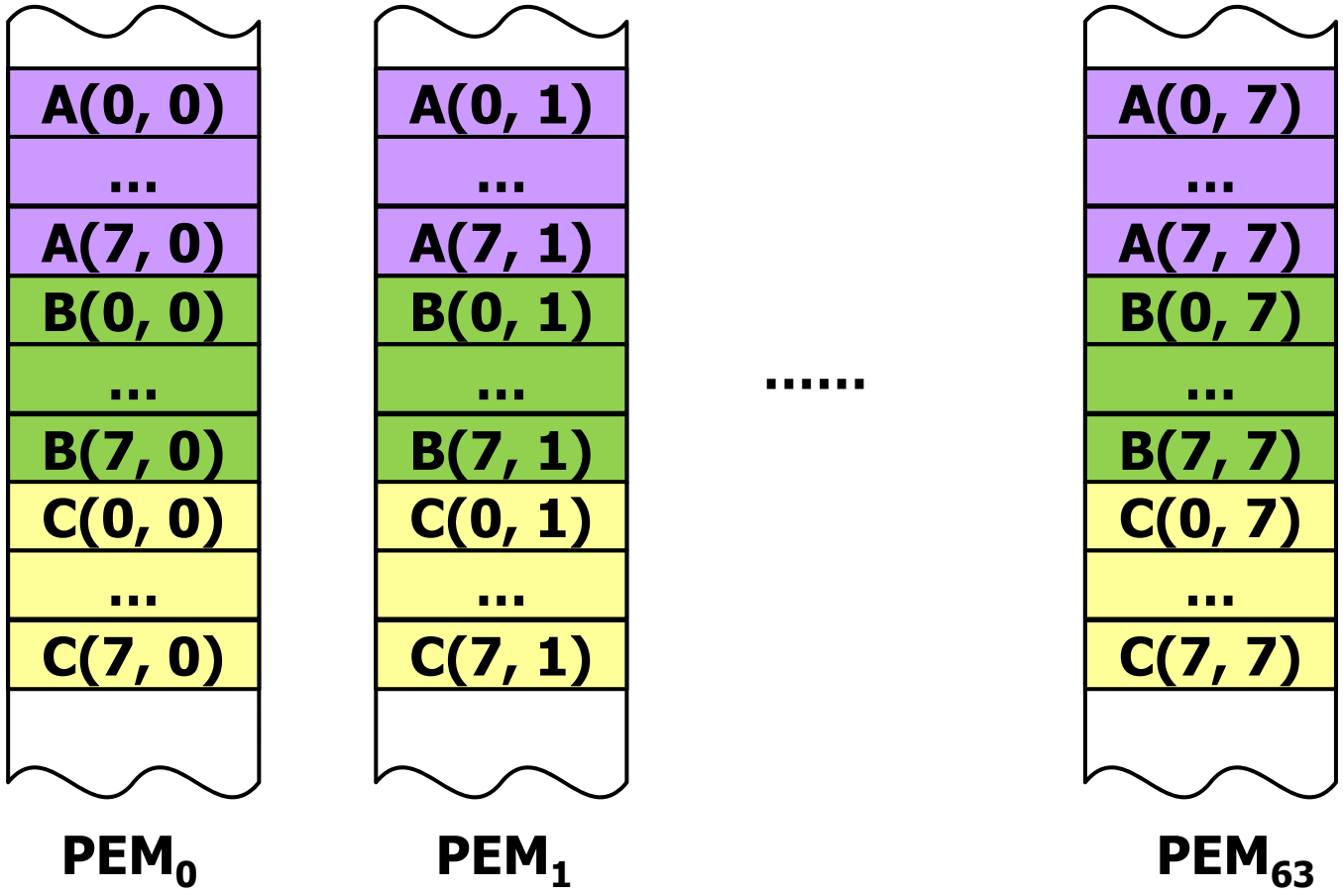
(2) 矩阵乘

- 如果在SIMD计算机上求解这个问题，如果利用8个处理部件并行计算，则J循环只需一次即可完成，I、K循环照旧。这样，便可使速度提高到8倍，即缩短为64次加乘时间。

```
DO 10 I=0,7
   C(I,J)=0
   DO 10 K=0, 7
10 C(I,J)=C(I,J)+A(I,K)*B(K,J)
```

(2) 矩阵乘

- 局部存储器中的数据分布如下：



(2) 矩阵乘

PE0：c00＝a00b00＋a01b10＋a02b20……＋a07b70
PE1：c01＝a00b01＋a01b11＋a02b21……＋a07b71

……

PE7：c07＝a00b07＋a01b17＋a02b27……＋a07b77
PE0：c10＝a10b00＋a11b10＋a12b20……＋a17b70
PE1：c11＝a10b01＋a11b11＋a12b21……＋a17b71

……

PE7：c17＝a10b07＋a11b17＋a12b27……＋a17b77

……

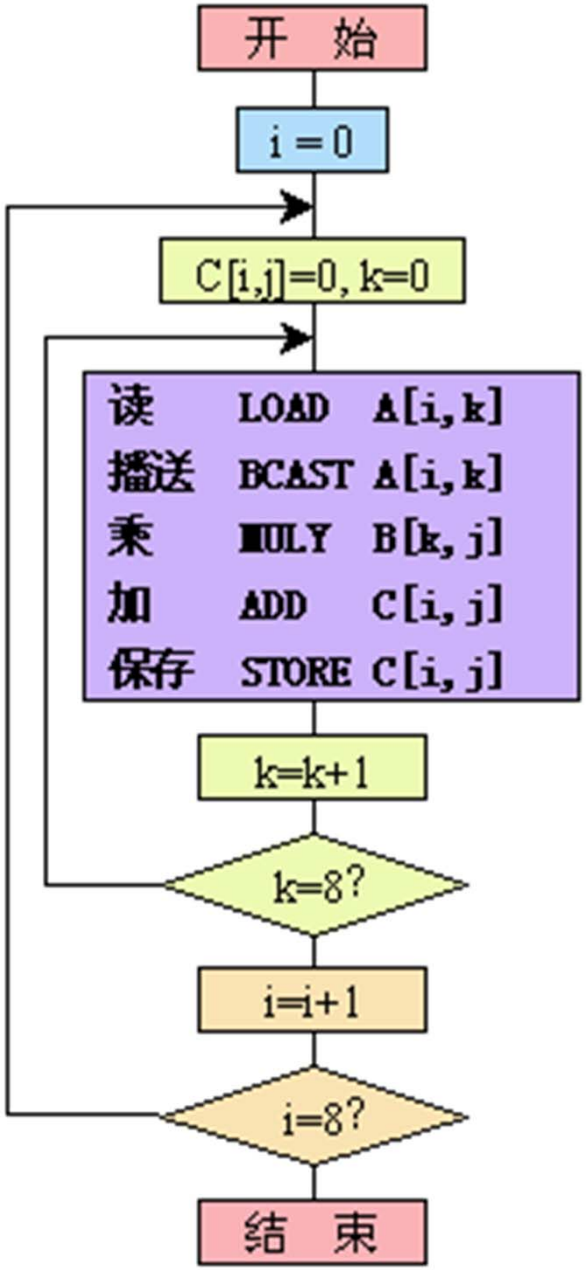
PE0：c70＝a70b00＋a71b10＋a72b20……＋a77b70
PE1：c71＝a70b01＋a71b11＋a72b21……＋a77b71

……

PE7：c77＝a70b07＋a71b17＋a72b27……＋a77b77

(2) 矩阵乘

矩阵乘程序流程图



(3) 累加和

- 把N个数的顺序相加变为并行相加。
- 在**SISD**计算机上可写成下列**FORTRAN**程序：

```
C=0
DO 10 I=0, 7
10 C=C+A(I)
```

这是一个串行程序，需要8次加法时间。

(3) 累加和

- 在**SIMD**计算机上采用**成对递归相加**的算法，则只需$\log_2 8=3$ 次加法时间就够了。

```
DO 10 I=0, log2N-1
10 A＝A+SRL(A, 2**I)
; 把A向量逻辑右移2^I个PE
```

只需做$\log_2 N$次加法。

(3) 累加和

- 首先，将原始数据A(I)， 0≤I≤7，存放在8个PEM的 α 单元中，然后按照下面的步骤求累加和：
- 第1步： 置全部PE为活动状态；
- 第2步：全部A(I)，0≤I≤7，从PE的 α 单元读到相应PE的累加寄存器RGA中；
- 第3步：令K＝0；

(3) 累加和

- **第4步：** 全部PE的(RGA)转送到传送寄存器RGR；
- **第5步：** 全部PE的(RGR)经过互连网络向右传送 2^k 步距；
- **第6步：** 令 $j=2^k-1$；
- **第7步：** 置 PE_0 至 PE_j 为不活动状态；
- **第8步：** 处于活动状态的PE执行(RGA)=(RGA)+(RGR)操作；

(3) 累加和

- 第9步：k:=k+1；
- 第10步：若k<3，则转回第四步，否则继续往下执行；
- 第11步：置全部PE为活动状态；
- 第12步：全部PE的(RGA)存入相应PEM的 $\alpha+1$单元中。

(3) 累加

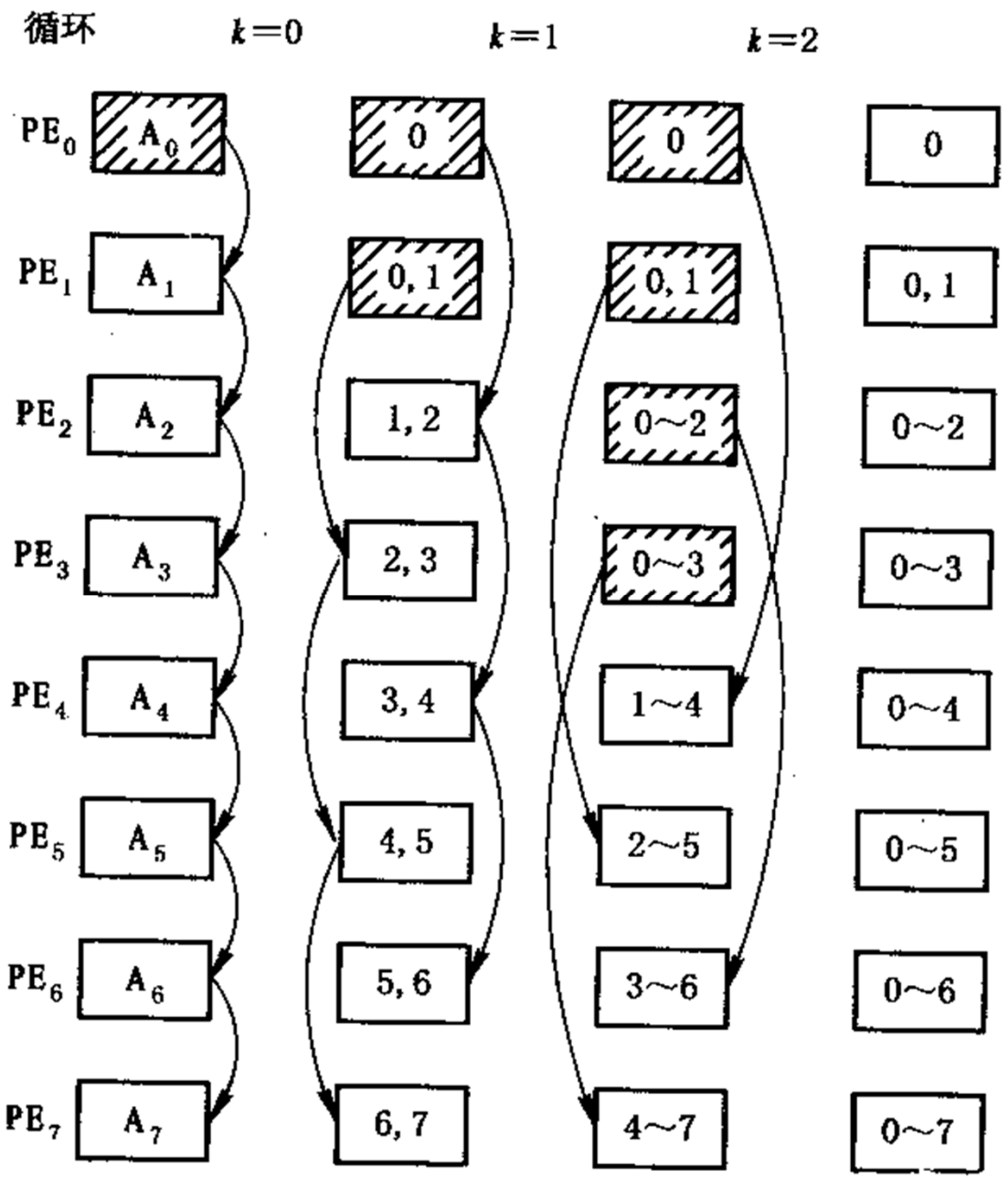
图 阵列处理机上累加和计算过程的示意图

(3) 累加

图 阵列处理机上累加和计算过程的示意图

(3) 累加和

采用SIMD并行算法，运算速度提高：加速比为$N/\log_2 N$倍，运算次数增加：从N次增加到$N\cdot\log_2 N$次，效率降低：实际效率为$1/\log_2 N$。



(3) 累加和

- 这种方法也称为级联求和或递归求和。
- 与流水线中采用的方法相同，它利用结合律来提高并行度。
- 可以利用结合律求解的问题还有：求最大数，求最小数，a与b进行异或运算，a与b进行逻辑或运算，a与b进行逻辑与运算。求a与b的点积，等等。

SIMD与向量计算机的区别

- 在向量计算机中，同一条指令可以连续地处理一个数据组，而这些数据是以流水线的方式通过处理机，即采用一种时间复用的方式，
- 在SIMD计算机中，通过大量处理单元对向量中包含的各分量同时进行处理，也就是说，SIMD是通过硬件资源的重复设置来实现并行运算的，即采用空间复用方式。

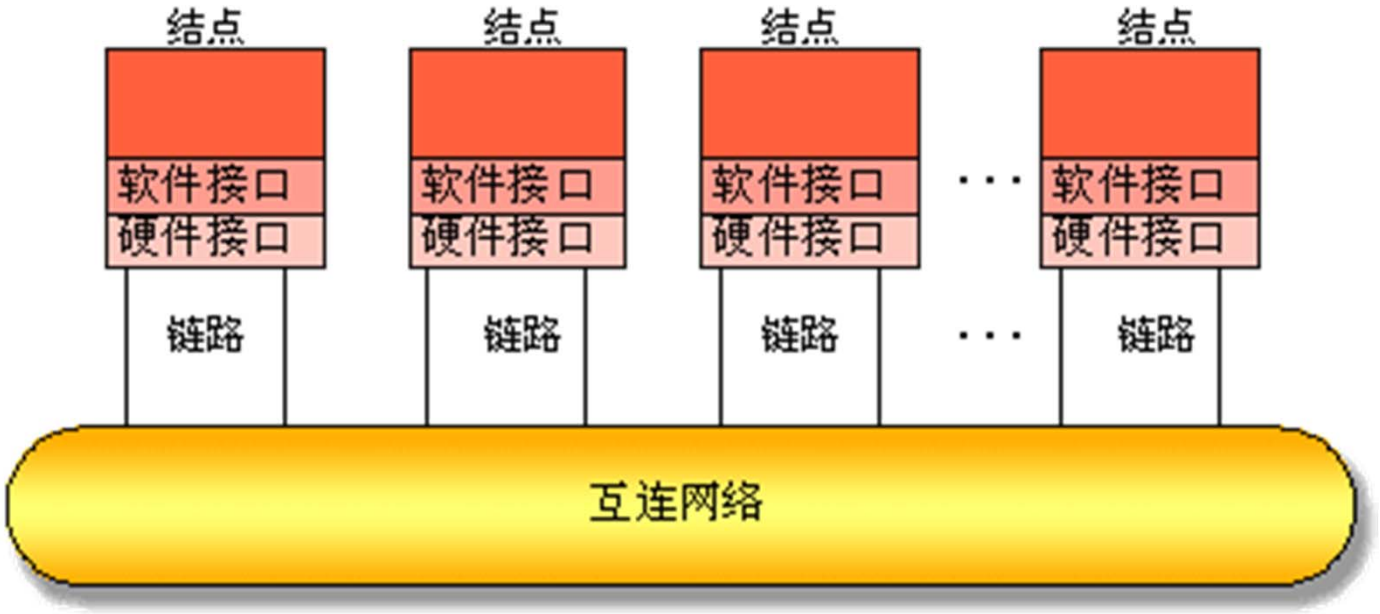
系统结构	并行性 开发途径	运算 速度	设备利用率	向量长度对运算时间的影响	向量长度对算法的影响
向量流水计算机	时间重叠	较慢	设备少，利用率高	线性增长	在一定范围内无影响
向量并行计算机	资源重复	较快	设备多，利用率低	在一定范围内无关	密切相关

6.3 互连网络的基本概念

- 在SIMD计算机中，PE之间以及PE与存储体之间都要通过互连网络（ICN）进行信息交换
- ICN限定了并行处理机适用的解题算法的类型，也对整个系统的性能产生明显影响
  - ICN结构复杂性：反映成本
  - ICN结构灵活性：反映性能
- 因此，ICN是设计重点

6.3.1 互连网络的设计目标与互连函数

- **互连网络（ICN）：** 互连网络是一种由开关元件按照一定的拓扑结构和控制方式构成的网络，用来实现计算机系统内部多个处理机或多个功能部件之间的相互连接。



6.3.1 互连网络的设计目标与互连函数

- 互连网络已成为并行处理系统的核心组成部分。
- 互连网络对整个计算机系统的性能价格比有着决定性的影响。

图注：IPMN（处理机－存储器网络）
PION（处理机－I/O网络）
IPCN （处理机之间通信网络）
P（处理机）
C（高速缓冲存储器）
SM（共享存储器）
LM（本地存储器）

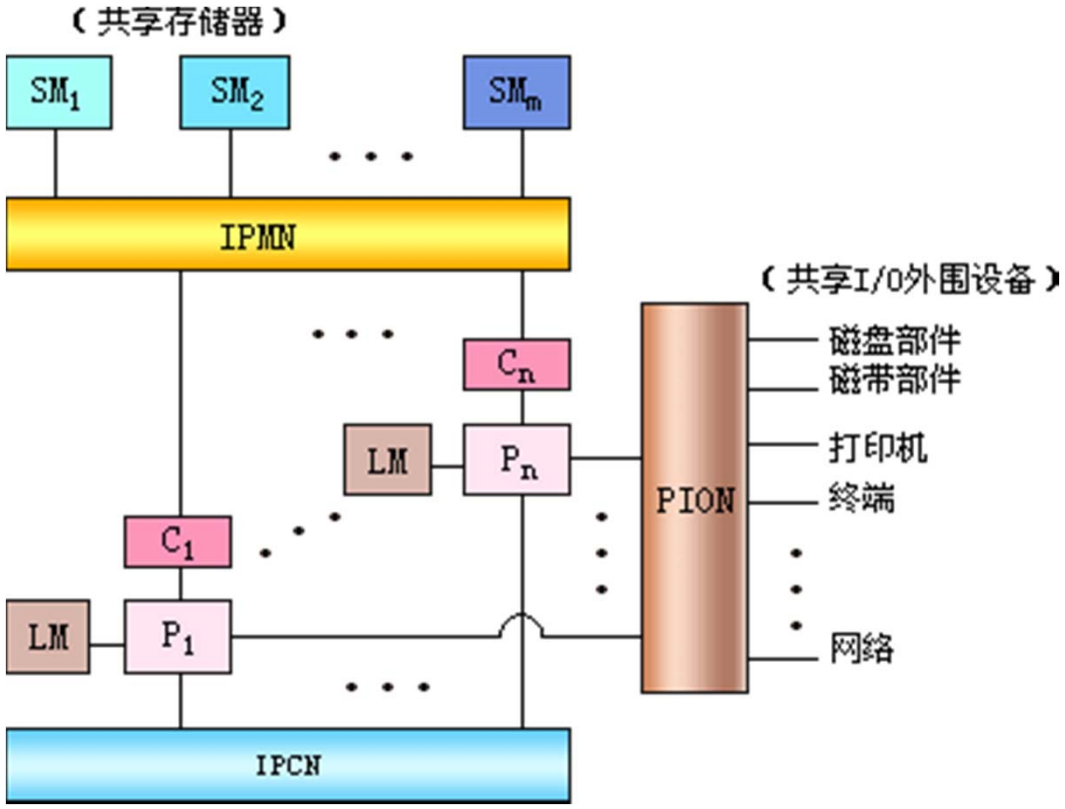


图 具有本地存储器、私有高速缓存、共享存储器和共享外围设备的多处理机系统互连结构

6.3.1 互连网络的设计目标与互连函数

- 目前，建造多达214~216个PE的阵列处理机已经成为现实。
- 若让任意2个PE之间都有直接通路，则ICN的连线数量多得无法实现【$O(n^2)$】。
- 因此，采取让PE、MM之间只有有限的几种直连方式，经过一步或少数几步的传送来实现任意两个PE之间信息传送是一种可行的方法。

互连网络的设计目标

- 结构不要复杂，以降低成本；
- 互连要灵活，以满足算法和应用的需要；
- PE之间传送信息的步数尽可能少，以提高速度性能；
- 可以用一系列规整单一的基本构件或其组合实现，或经多次通过或经多级连接来实现复杂的互连，模块性好，便于用VLSI实现，并满足系统的可扩充性。

1. 互连网络的表示方法

- 为了在输入结点与输出结点之间建立对应关系，互连网络有**3种表示方法：**
- (1) 互连函数表示法
- (2) 图形表示法
- (3) 输入输出对应表示法

互连函数表示法

- 为了反映不同互连网络的连接特性，可以用一组互连函数来定义互连网络。
- **互连函数：**

  如果将互连网络的N个输入端和N个输出端分别用整数0，1，…，N-1来表示，则互连函数表示相互连接的输出端号和输入端号之间的一一对应关系。或者说，存在互连函数 f，在它的作用下，输入 i 应与输出 f(i) 相连，0 ≤ i ≤ N-1。

互连函数表示法

- 函数表示法用x表示输入端变量，用f(x)表示互连函数。x还常用二进制形式来表示，写成$x_{n-1}x_{n-2}\cdots x_1x_0$。互连函数则对应地表示为$f(x_{n-1}x_{n-2}\cdots x_1x_0)$。例如：

二进制：$f(x_{n-1}\cdots x_1x_0)=x_0x_{n-2}\cdots x_1x_{n-1}$

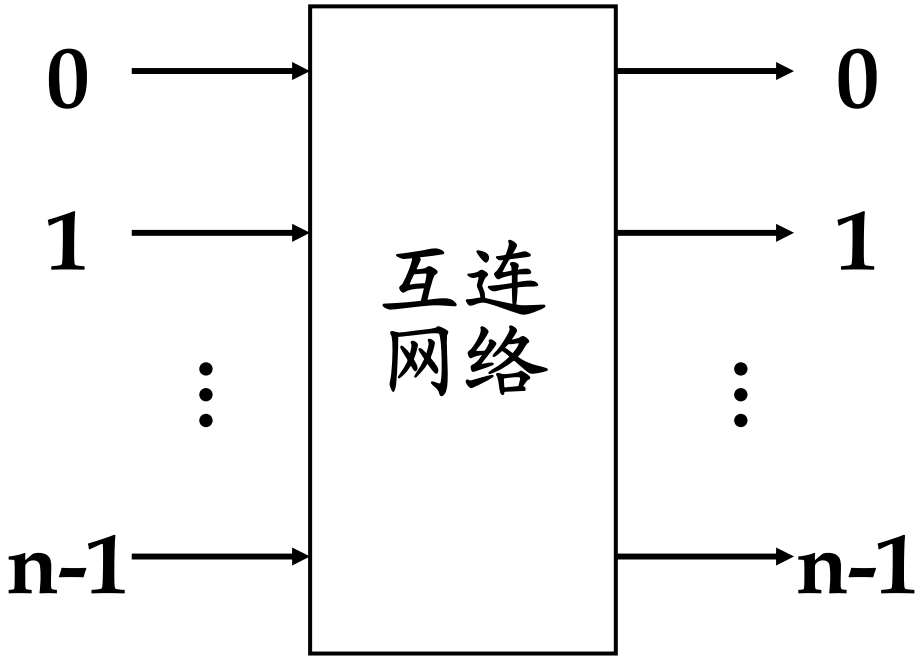
十进制：f(入端号j) ＝出端号i

$i, j=0, 1, 2, \ldots N-1$

为了体现连接上的内在规律，常把入端 x 和出端 f(x) 用二进制数编码，从二者的二进制数编码上找出规律。

图形表示法

- 直观



输入输出对应表示法

输入: 0 1 2 3 4 5 6 7

输出: 1 0 3 2 5 4 7 6

$$\begin{pmatrix} 0 & 1 & \cdots\cdots & N-1 \\ f(0) & f(1) & \cdots\cdots & f(N-1) \end{pmatrix}$$

2. 互连网络特性

- 互连网络的主要特性有(1/3)：
  - (1) 网络规模：网络中结点的个数
  - (2) 结点度：与结点相连接的边数称为结点度。包括入度和出度。进入结点的边数叫入度，从结点出来的边数则叫出度

    结点度 = 入度 + 出度
  - (3) 距离：两个结点之间相连的最少边数

2. 互连网络特性

- 互连网络的主要特性有(2/3):
  - (4) 网络直径：网络中任意两个结点间距离的最大值。用结点间的连接边数表示
  - (5) 结点间的线长：两个结点之间连线的长度。用米、公里等表示
  - (6) 等分宽度：某一网络被切成相等的两半时，沿切口的最小边数称为该网络的等分宽度

2. 互连网络特性

- 互连网络的主要特性有(3/3)：
  - (7) 对称性：从任何结点看到拓扑结构都是一样的网络称为对称网络。对称网络比较易实现，编程也较容易。

3. 互连网络的传输性能参数

- 一台机器发送消息给另一台机器时，发送方的步骤如下：
  - (1) 用户程序把要发送的数据拷贝到操作系统的缓冲区。
  - (2) 操作系统把缓冲区中的数据打包，并发送到网络接口部件。
  - (3) 网络接口硬件开始发送消息。

3. 互连网络的传输性能参数

- 数据包的接收步骤如下：
  - (1) 把数据包从网络接口部件拷贝到操作系统缓冲区。
  - (2) 检查收到的数据包，如果正确，给发送方发回答信号。
  - (3) 把接收到的数据拷贝到用户地址空间。
- 发送方接收到回答信号后，释放系统缓冲区。

3. 互连网络的传输性能参数

- 互连网络的性能参数: (1/2)
  - (1) 频带宽度(Bandwidth)：互连网络传输信息的最大速率。
  - (2) 传输时间(Transmission time)：等于消息长度除以频宽。
  - (3) 飞行时间(Time of flight)：第一位信息到达接收方所花费的时间。
  - (4) 传输时延(Transport latency)：等于飞行时间与传输时间之和。

3. 互连网络的传输性能参数

- 互连网络的性能参数: (1/2)
  - (5) 发送方开销(Sender overhead)：处理器把消息放到互连网络的时间。
  - (6) 接收方开销(Receiver overhead)：处理器把消息从网络取出来的时间。
- 一个消息的总时延可以用下面公式表示

  总时延＝发送方开销＋飞行时间＋消息长度/频宽＋接收方开销

3. 互连网络的传输性能参数

- 例：

  假设一个网络的频宽为10Mb/s，发送方开销为230μs，接收方开销为270 μs。如果两台机器相距100米，现在要发送一个1000字节的消息给另一台机器，试计算总时延。如果两台机器相距1000公里，那么总时延为多大？

3. 互连网络的传输性能参数

- 解：

  光的速度为299792.5km/s，信号在导体中传递速度大约是光速的50%，相距100米时总时延为：

$$T = \text{发送方开销} + \text{飞行时间} + \frac{\text{消息长度}}{\text{频宽}} + \text{接收方开销}$$

$$= 230\,\mu s + \frac{0.1Km \times 10^6}{0.5 \times 299792.5}\,\mu s + \frac{1000 \times 8\text{位}}{10\text{兆位/秒}} + 270\,\mu s$$

$$= 230\,\mu s + 0.67\,\mu s + 800\,\mu s + 270\,\mu s = 1301\,\mu s$$

3. 互连网络的传输性能参数

- 解：
  相距1000公里时总时延为：

$$T = 230\mu s + \frac{1000km \times 10^6}{0.5 \times 299792.5}\mu s + \frac{1000 \times 8}{10}\mu s + 270\mu s$$
$$= 230\mu s + 6671\mu s + 800\mu s + 270\mu s \quad = 7971\mu s$$

6.3.2 设计互连网络时应考虑的问题

- 确定PE之间通信的互连网络的因素包括：
  - 操作方式
  - 控制策略
  - 交换方法
  - 网络拓扑结构等

1. 操作方式

- **3种方式：**
  - 同步
  - 异步
  - 同步/异步组合
- 现在的阵列处理机均采用同步操作方式，让所有PE按时钟同步。
- 多处理机一般采用异步或同步/异步组合方式

2. 控制策略

- 典型的ICN由开关单元和互连线路组成。
- 互连通路的路径选择通过设置开关单元的状态来控制。
- 开关单元的状态的设置有两种控制策略：
  - 集中控制策略
  - 分布控制策略
- 多数SIMD的ICN采用集中控制策略。

3. 交换方法

- 主要有3种：
  - 线路交换
  - 包交换
  - 线路/包交换
- 线路交换需要在源和目的之间建立实际的连接线路，适合大批数据传输。
- 包交换将数据置于包内传输，不需要建立实际的连接线路，适合短数据传输。
- SIMD的ICN一般采用线路交换。
- 多处理机和计算机网络一般采用包交换。

4. 网络拓扑结构

- 网络拓扑结构指互连网络输入、输入端可以实现连接的模式。
- 有两种：
  - 静态
  - 动态：单级 和 多级
- 在静态连接模式中，两个PE之间的连接是固定的，不能重新配置成与其他PE连接。
- 在动态连接模式中，PE之间的连接可以通过设置开关单元的状态重新配置。

6.3.2 设计互连网络时应考虑的问题

操作方式

- 同步：SIMD采用
- 异步：多处理机采用
- 同步异步组合：多处理机采用

控制方式

- 集中式：SIMD采用。集中控制全部开关
- 分布式：多处理机采用？

6.3.2 设计互连网络时应考虑的问题

交换方式

- 线路交换：SIMD采用硬联线路交换，大批数据传输
- 包交换：多处理机和计算机网络采用
- 线路/包交换组合：ATM

拓扑结构

- 静态：连接固定，灵活性差，适应性差
- 动态：
  - 单级：有限几种连接，需多次循环使用
  - 多级：多个单级网路串联而成，实现任意连接
    多级网络循环使用，实现复杂连接

6.4 互连网络的种类

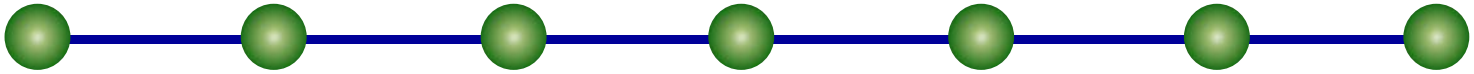
- 种类很多，分类方法也很多。
- 以互连特性为特征，可分为如下几类：
  - 静态互连网络
  - 循环互连网络
  - 多级互连网络
  - 全排列互连网络
  - 全交叉开关网络
  - 动态互连网络

6.4.1 静态互连网络

- 在各结点之间有固定的连接通路，在运行过程中不能改变的网络结构。
- 网络拓扑结构有：
  - 一维线形
  - 二维环形，星形，树形，胖树形，网格形，脉动阵列形
  - 三维的弦环形，立方体形，环立方体形等
  - 三维以上的有超立方体等

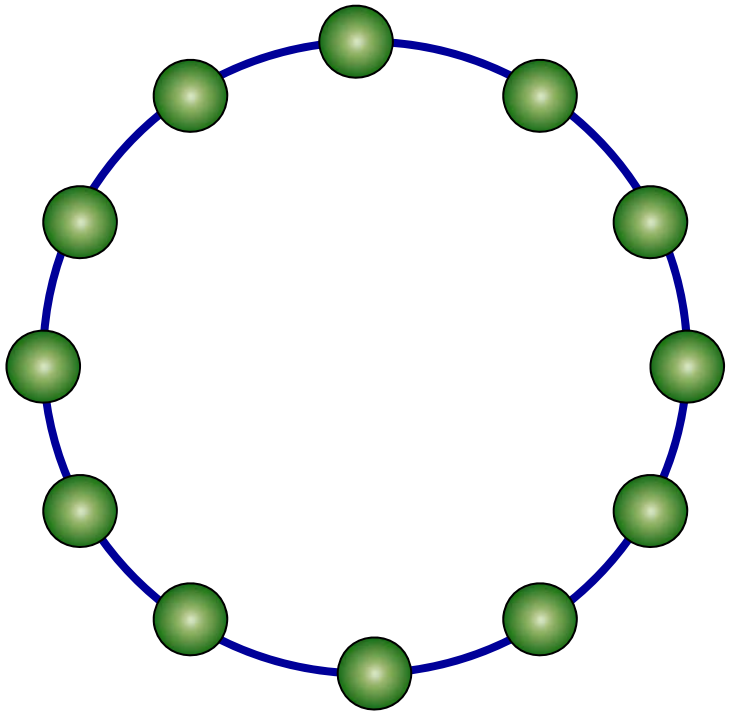
6.4.1 静态互连网络

- 一维线性阵列
  - 对N个结点的线性阵列，有N-1条链路，度为2，不对称，直径为N-1（任意两点之间距离的最大值），等分宽度为1。
  - N很大时，通信效率很低。



6.4.1 静态互连网络

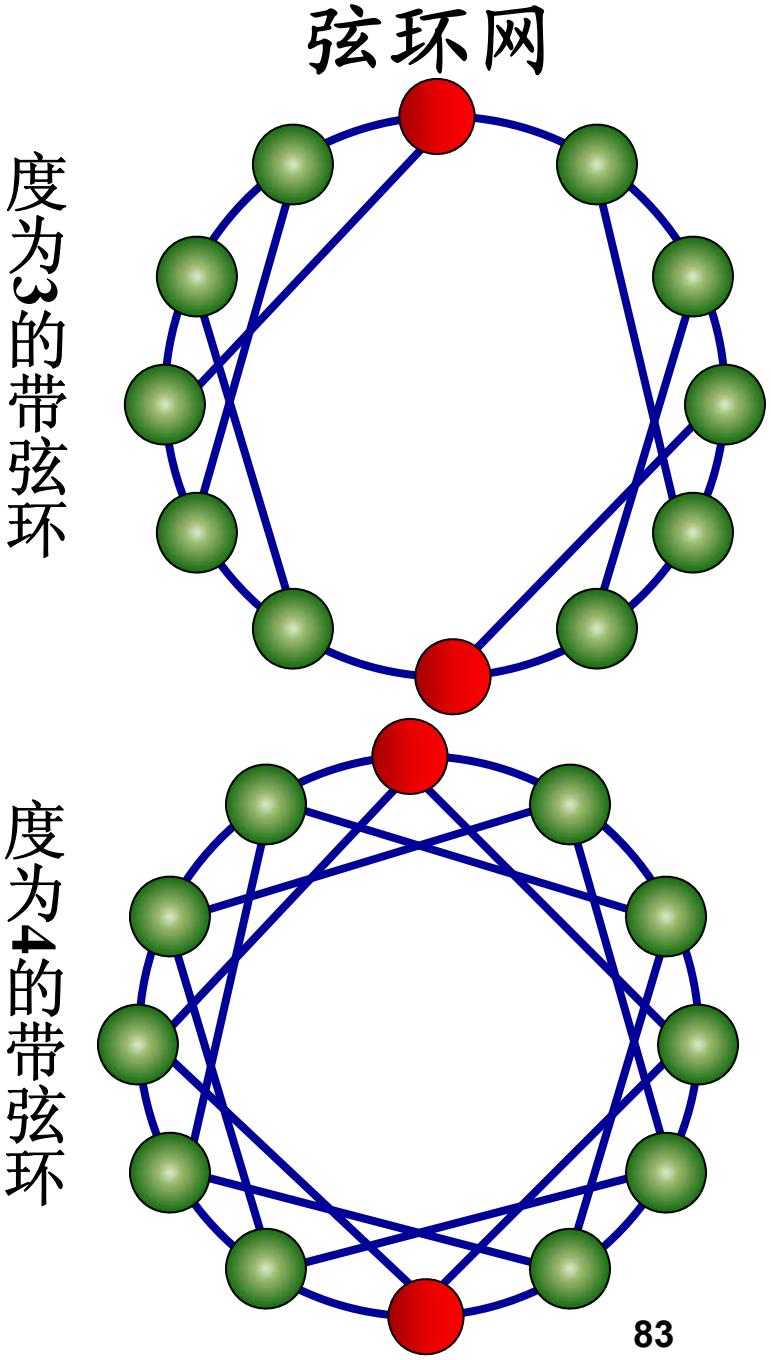
- 环形
  - 对N个结点的环，考虑相邻结点数据传送方向。
  - 双向环：链路数为N，直径⌊N/2⌋，度为2，对称，等分宽度为2。
  - 单向环：链路数为N，直径N-1，度为2，对称，等分宽度为2。



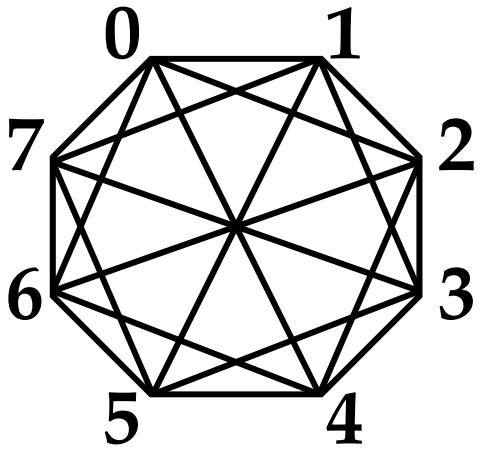
度为2的环

6.4.1 静态互连网络

- 带弦环（弦环网）
  - 对图中12个结点的带弦双向环：
    - **结点度为3**：链路数为18，直径4（比如红色结点），度为3，不对称，等分宽度为2。
    - **结点度为4**：链路数为24，直径3（比如红色结点），度为4，对称，等分宽度为8。



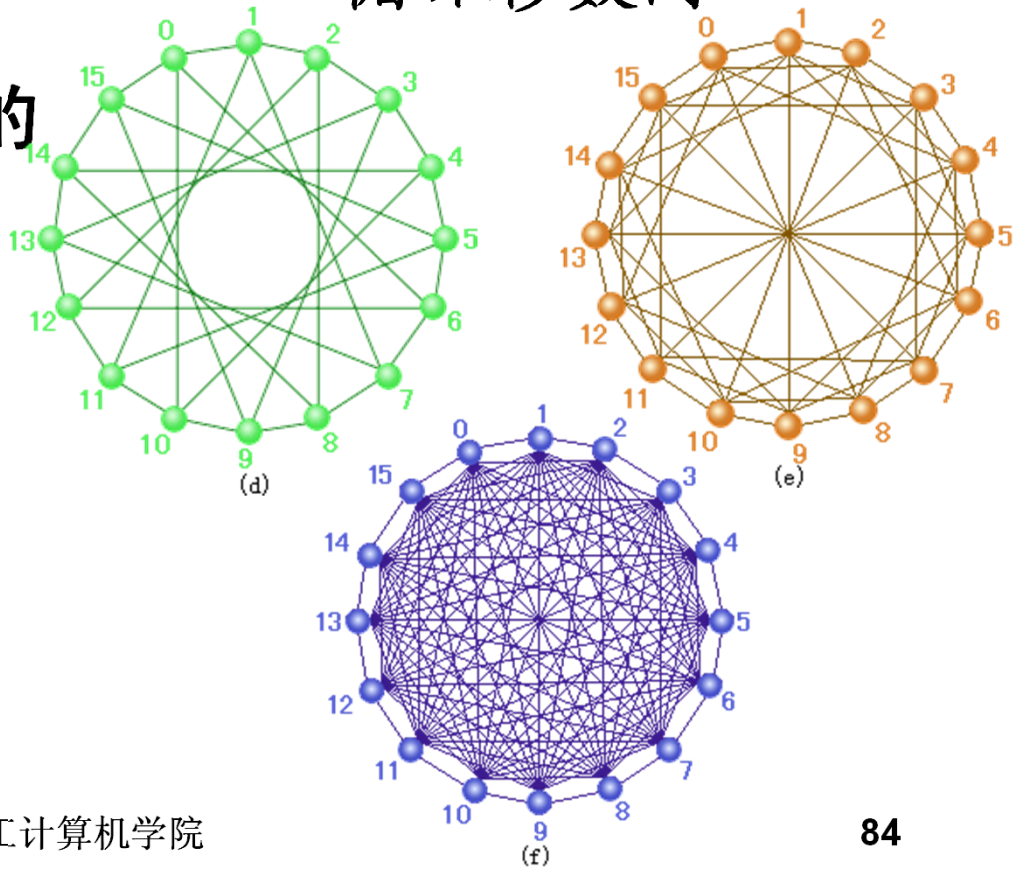
6.4.1 静态互连网络



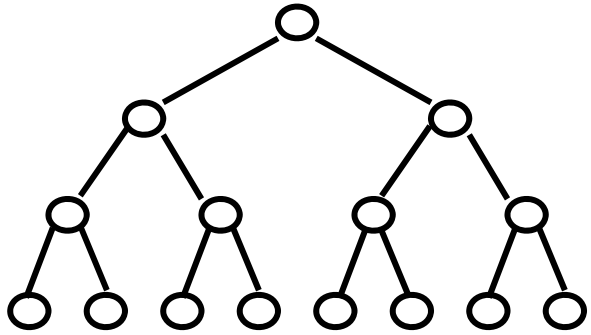
循环移数网

循环移数网络

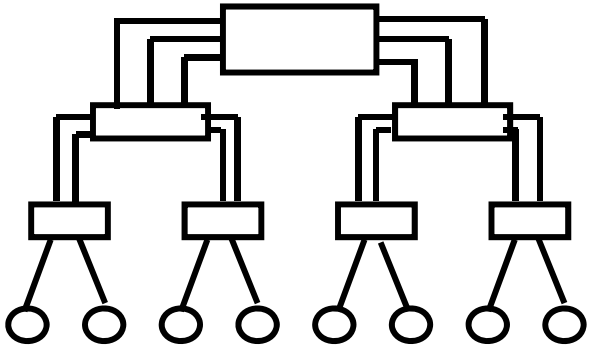
- 也是一种环形网。
- 它将环上每个结点与其距离为2的整数幂的结点之间连接构成。
- 循环移数网的结点度为2n-1，直径为⌈n/2⌉。



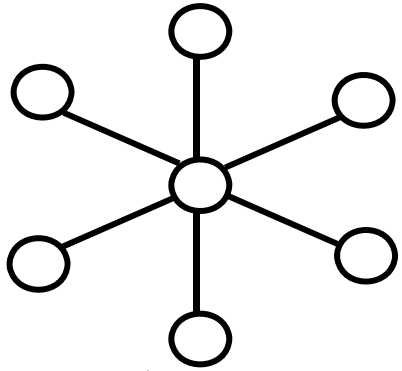
6.4.1 静态互连网络



二叉树网



二叉胖树网

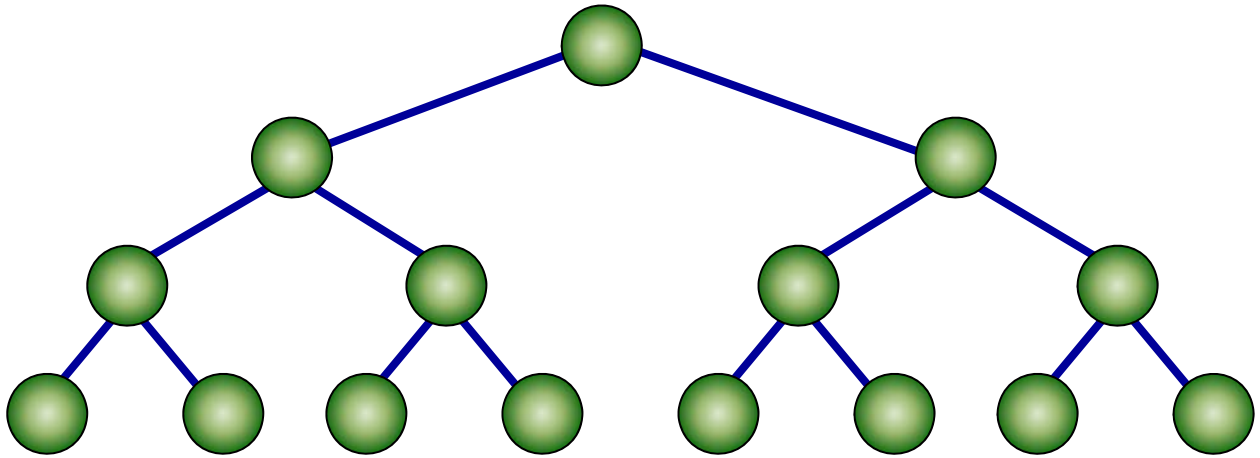


星形网

树形网和星形网

6.4.1 静态互连网络

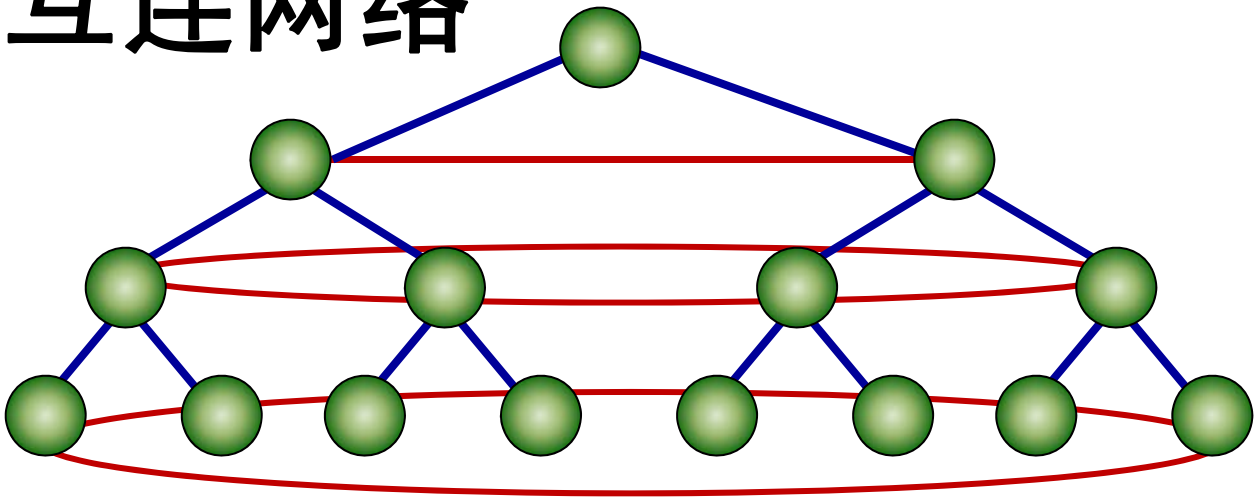
- 一棵K层完全二叉树应有$N = 2^K - 1$个结点，最大结点度为3，直径为$2(K-1)$（即右边任意一个叶子结点到左边任意一个叶子结点）。不对称，等分宽度为1。



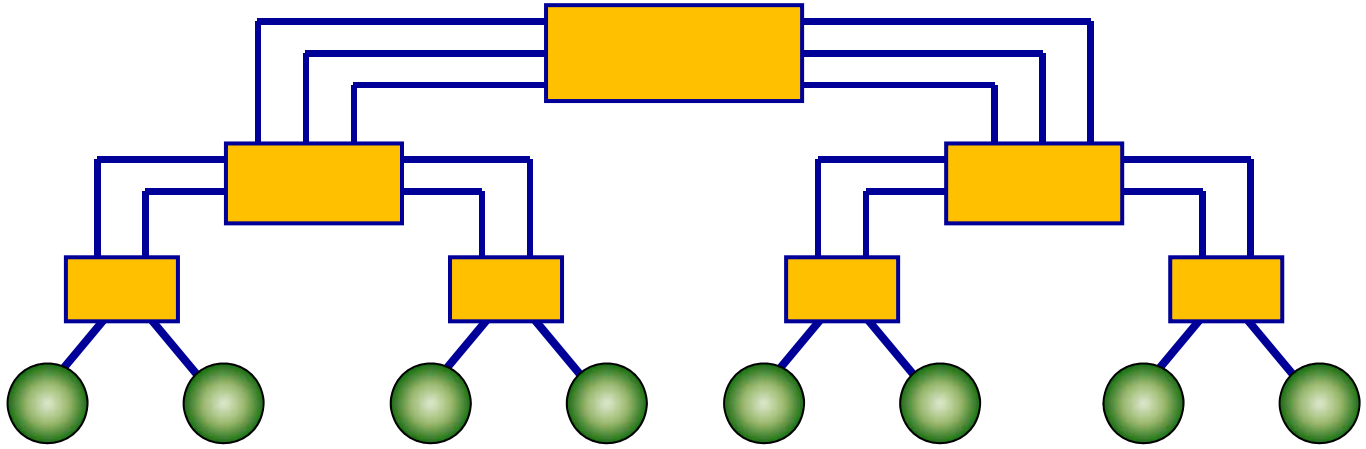
4层的二叉树

6.4.1 静态互连网络

- 树形的扩展



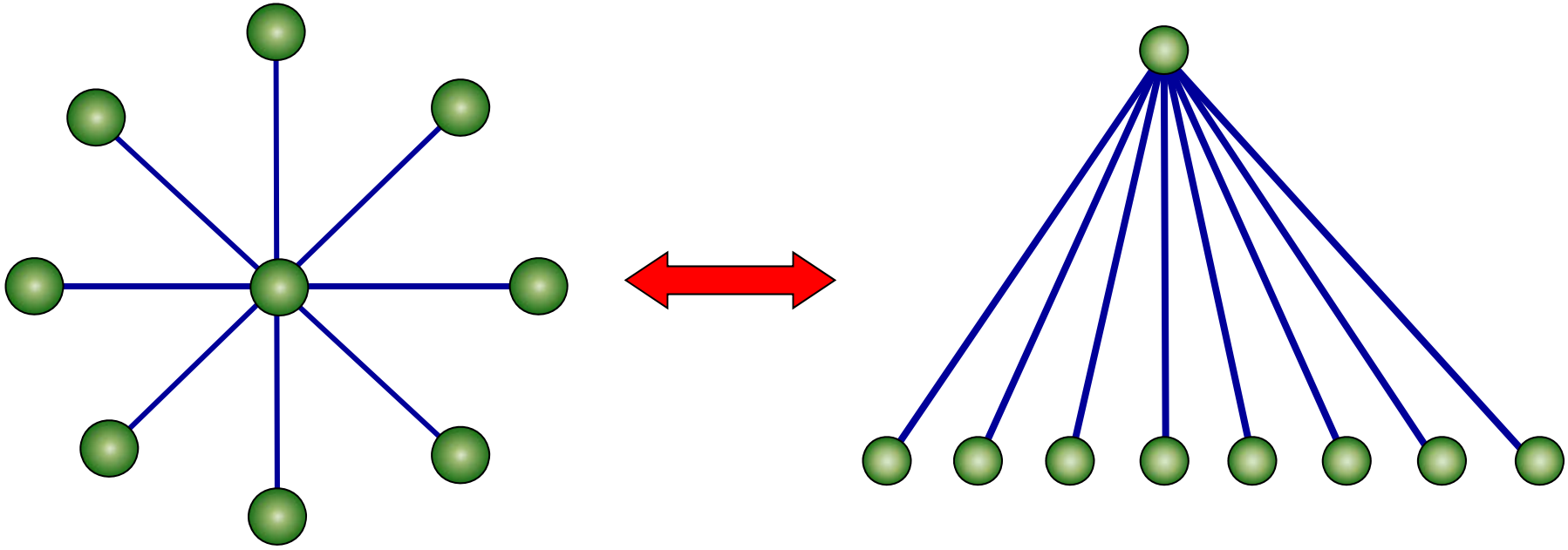
带环树



二叉胖树

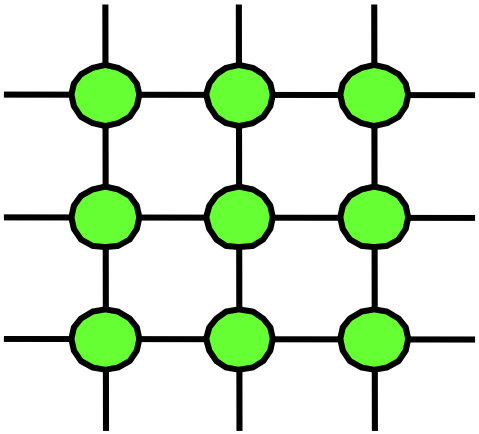
6.4.1 静态互连网络

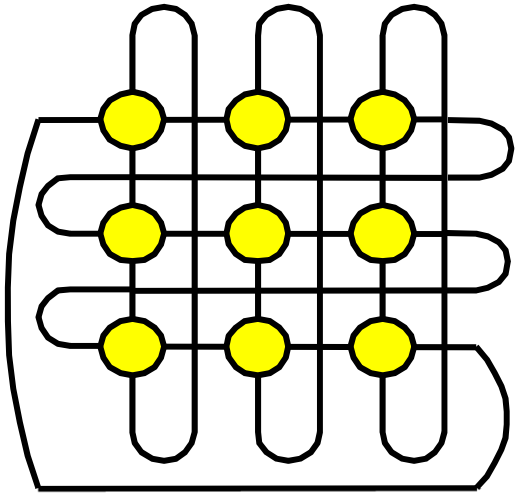
- 星形实际上是一种二层树（如右图）。有N个结点的星形网络，有N - 1条链路，直径为2，最大结点度为N - 1，非对称，等分宽度为1。



6.4.1 静态互连网络

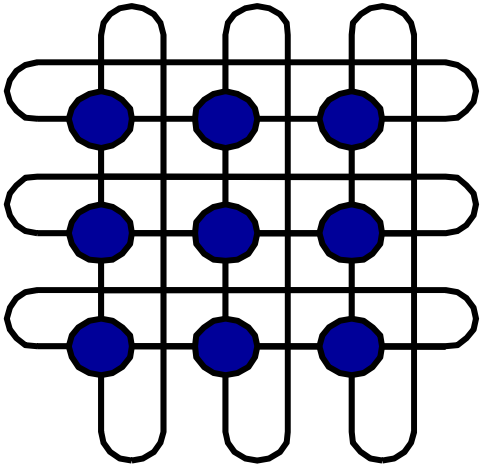
网格型

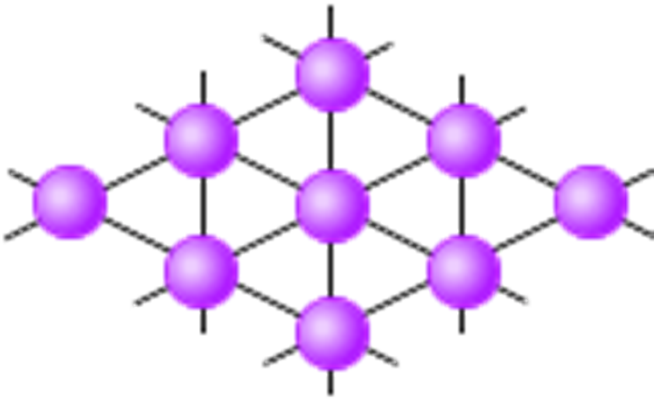




ILLIAC网

环型网



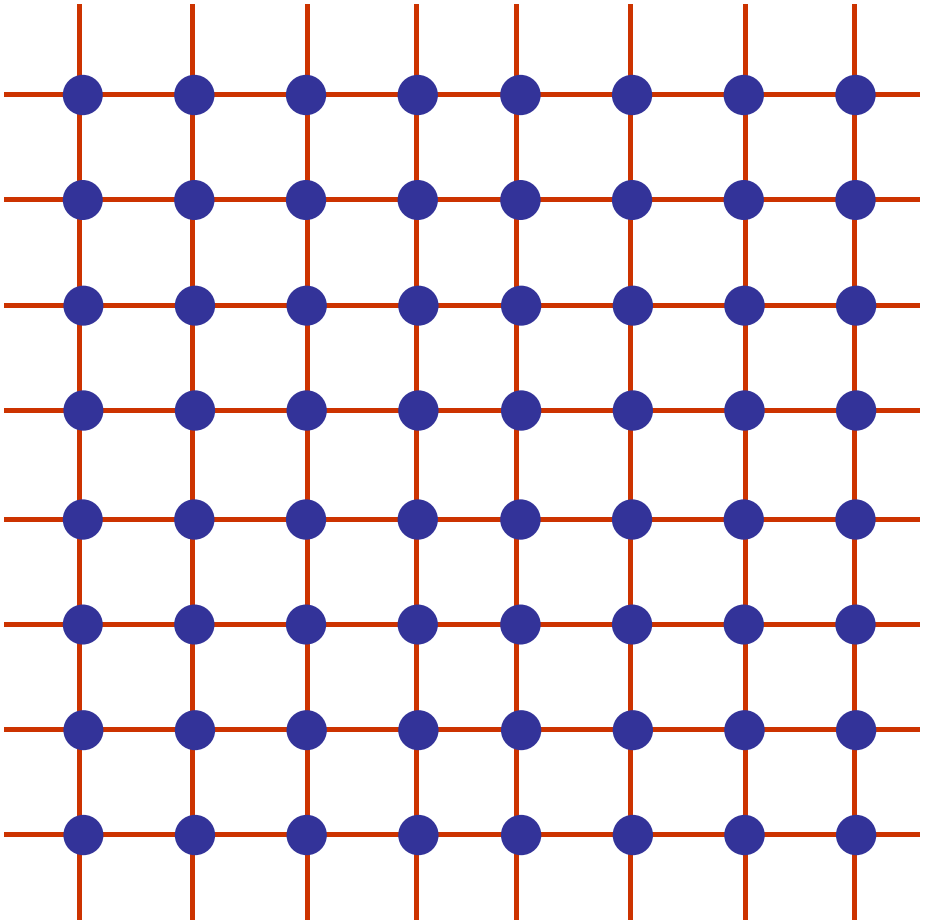


脉动式
阵列

网格形网

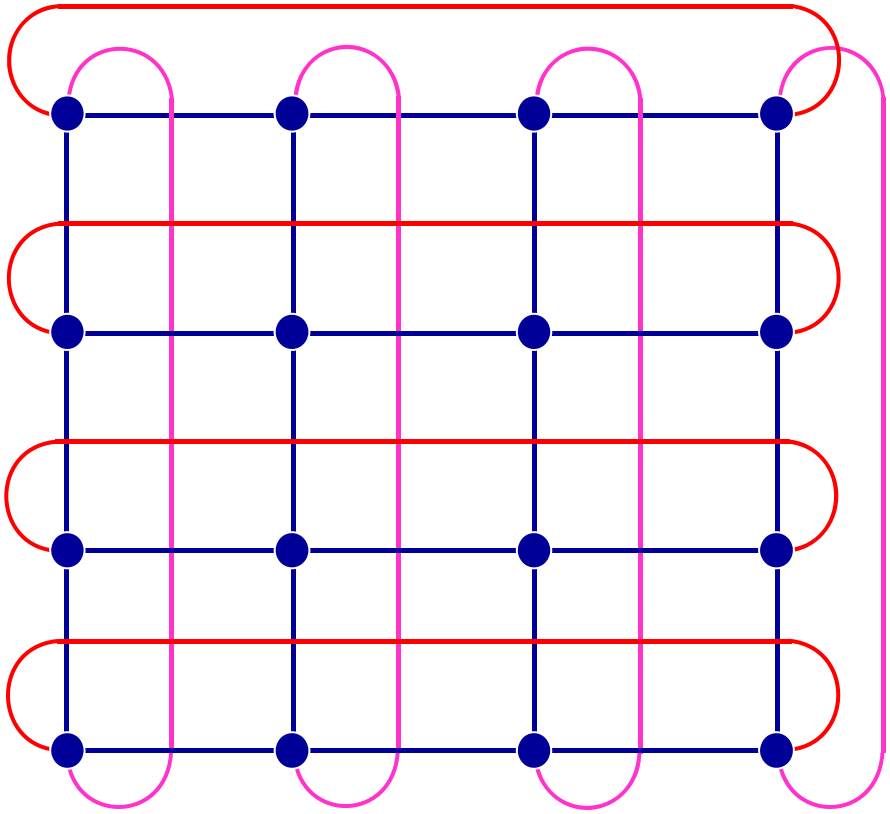
6.4.1 静态互连网络

- 网格形
  - 有**N**个结点的**r×r**网，有**2N - 2r**条链路，直径为**2(r-1)**，结点度为**4**，非对称，等分宽度为**r**。
  - 其中 $r = \sqrt{N}$



6.4.1 静态互连网络

- 二维环网形
  - 有**N**个结点的**r×r**网，有**2N**条链路，直径为**2⌊r/2⌋**，结点度为**4**，对称。
  - 其中 $r = \sqrt{N}$

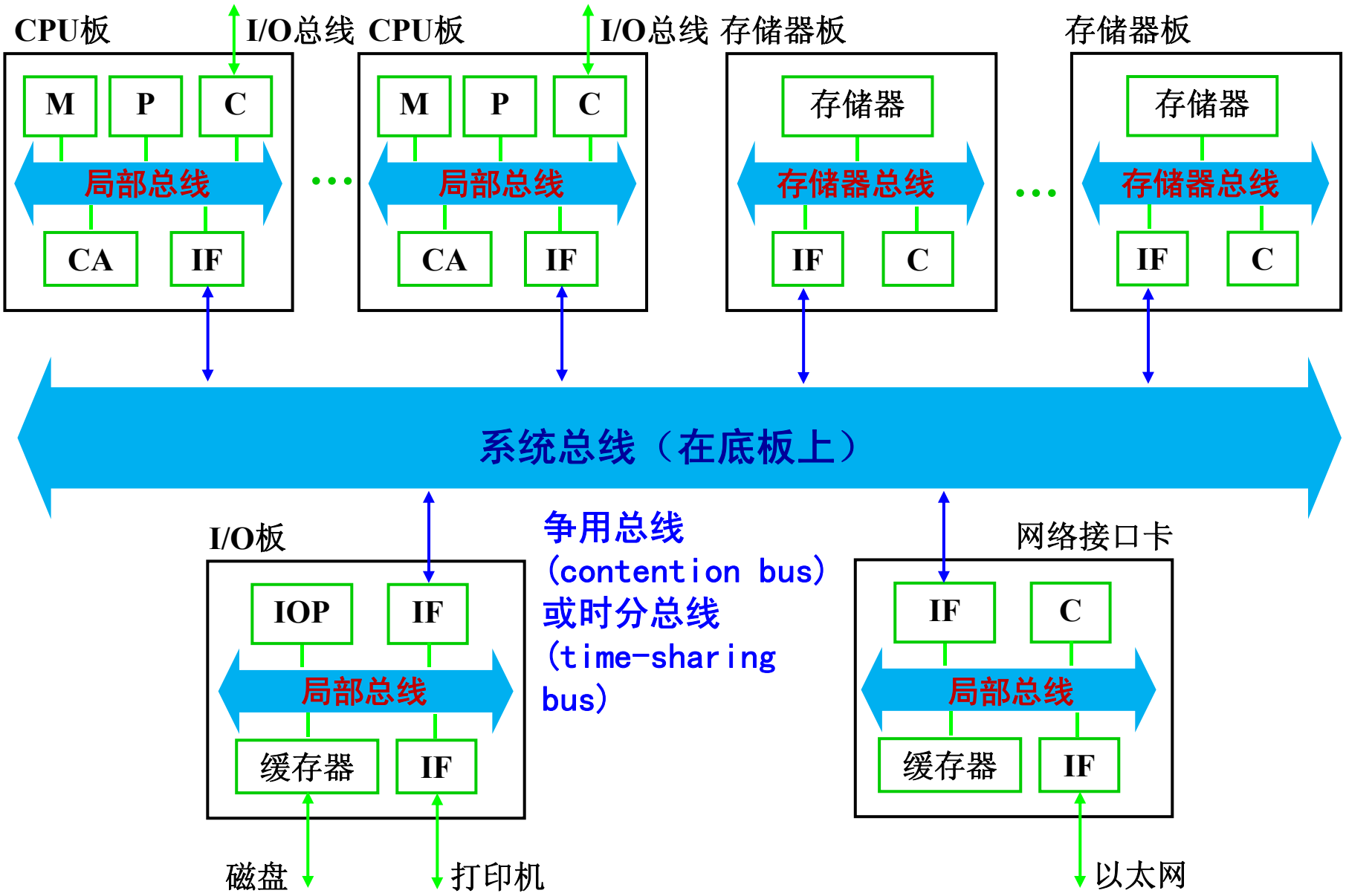


动态互连网络

- 动态网络中的连接不固定，在程序执行过程中可根据需要改变。
- 网络的开关元件有源，链路可通过设置这些开关的状态来重构。
- 只有在网络边界上的开关元件才能与处理机相连。
- 动态网络主要有：总线、交叉开关、循环网络，多级交换网络。

动态互连网络

- 总线（Bus）
  - 总线实际上是连接处理器、存储器和I/O等外围设备的一组导线和插座。
  - 它在某一时刻只能用于一对源和目的之间传输数据。
  - 当有多对源和目的请求使用总线时，要进行总线仲裁。当CPU数目较多时对总线争用严重（<=32个）。



IF:专用逻辑接口 C:专用控制器 P:处理器 M:局部存储器 CA:高速缓存 IOP:I/O处理器

动态互连网络

- 线性阵列与总线的区别
  - **线性阵列：** 允许不同的源结点和目的结点对**并发**使用系统的不同部分。
  - **总线：** 通过切换与其相连的许多结点来实现**时分**特性，同一时刻只有一对结点在传送数据。

动态互连网络

- 交叉开关（**Crossbar Switcher**）
  - 交叉开关是一种高带宽网络，它可以在输入端和输出端之间建立动态连接
  - 在每个输入端和输出端的交叉点上都有交叉点开关。该开关可以根据需要置为“开”或“关”状态，从而使不同的输入端和输出端导通
  - 交叉开关的硬件复杂性为n^2数量级，造价昂贵。但是其带宽和寻径性能在这三种动态网络中最好
  - 如果网络规模小，它是一种理想的选择（**<=64**个）

交叉开关

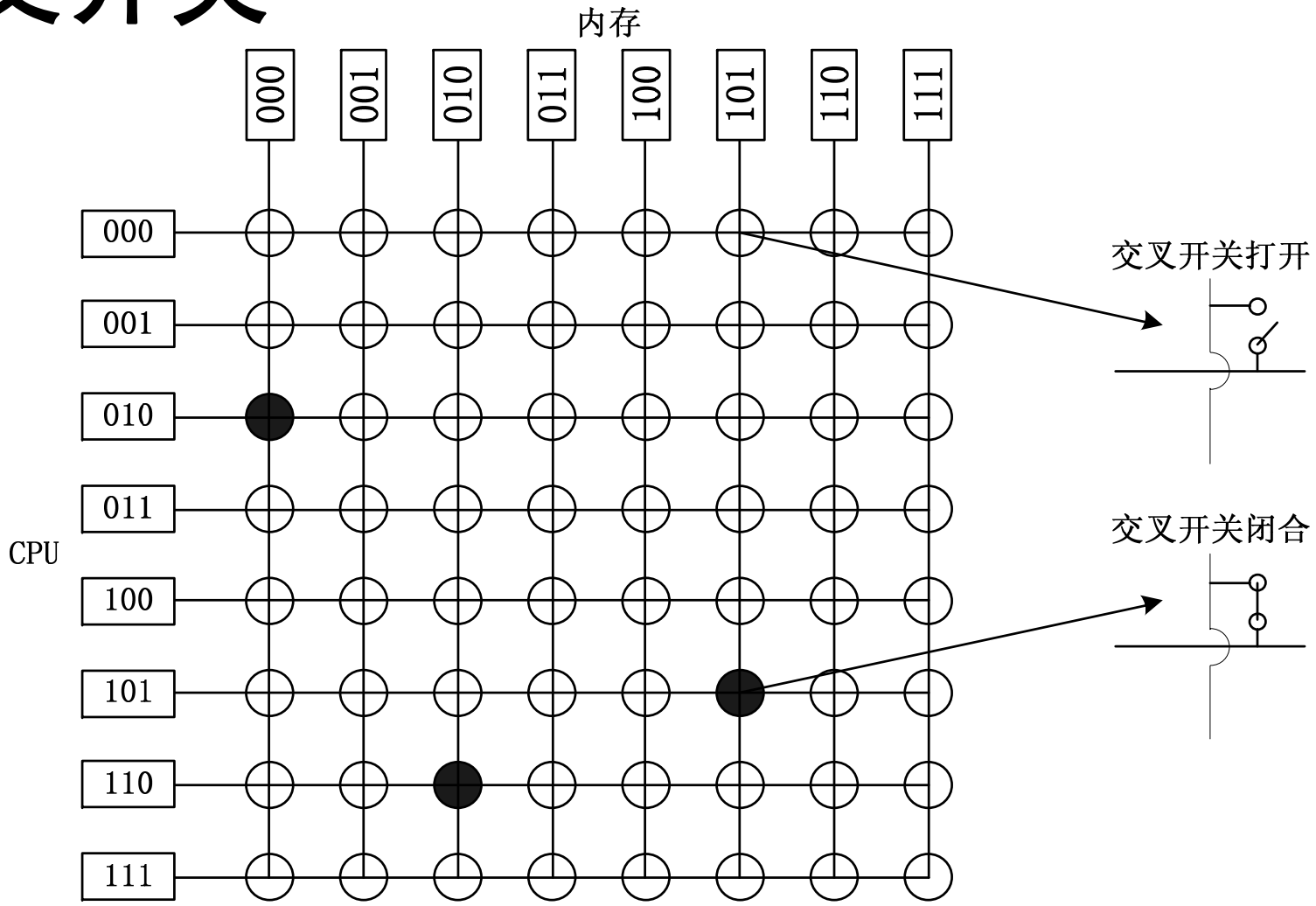


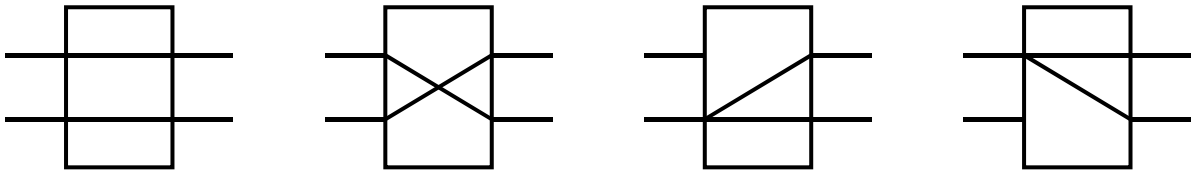
图 8×8的交叉开关

动态互连网络

- 循环网络
  - 将同一套单级互连网络循环使用，组成循环互连网络。
  - 经过多次循环，实现PE之间的连接。

动态互连网络

- 多级交换网络
  - 开关单元：a个输入a个输出的开关单元记作a×a的开关单元，其中，a是2的整数倍。常见的有2×2、 4×4、8×8等。
  - 根据开关单元功能的多少， 2×2又可以分为两功能和四功能开关。

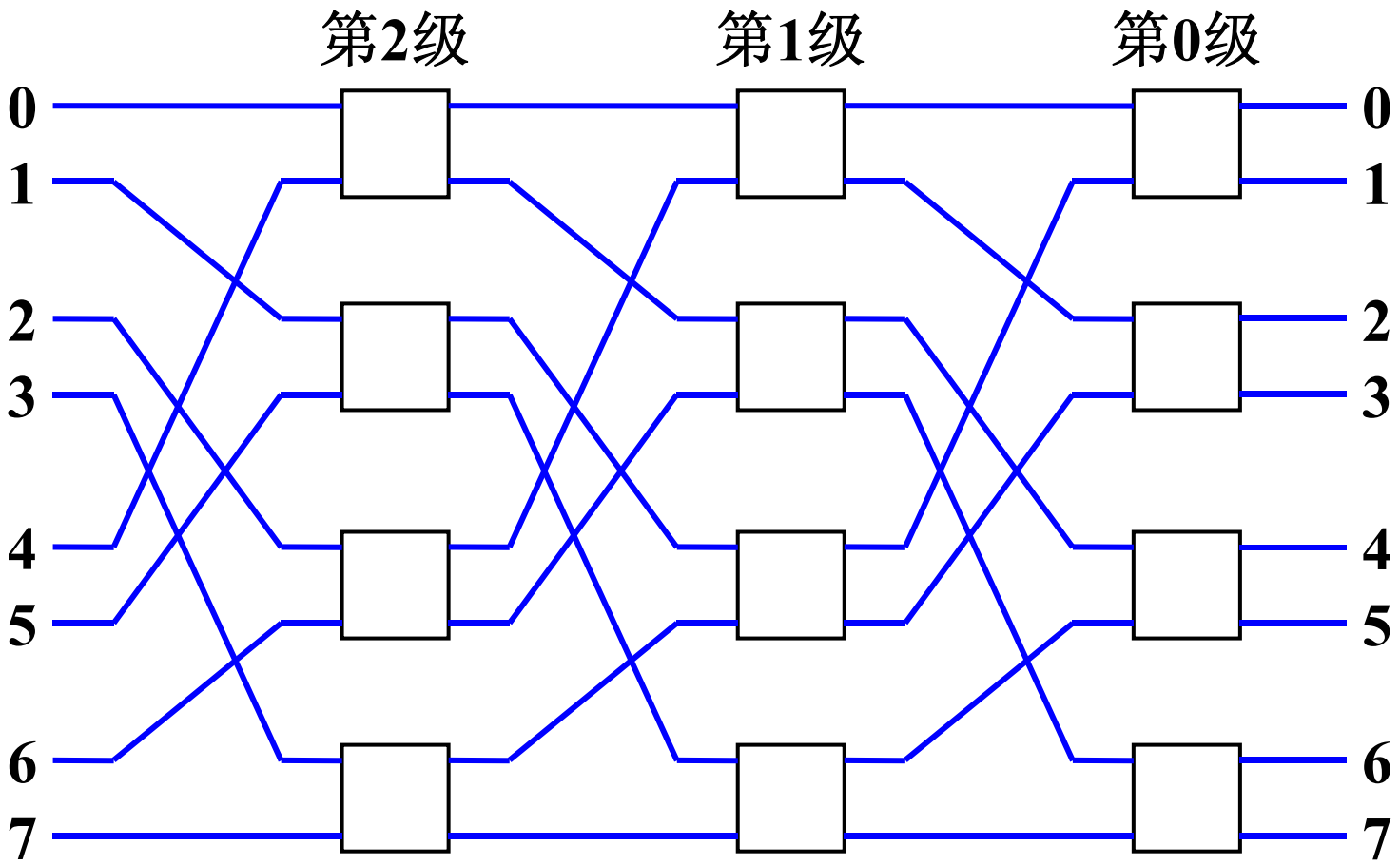


动态互连网络

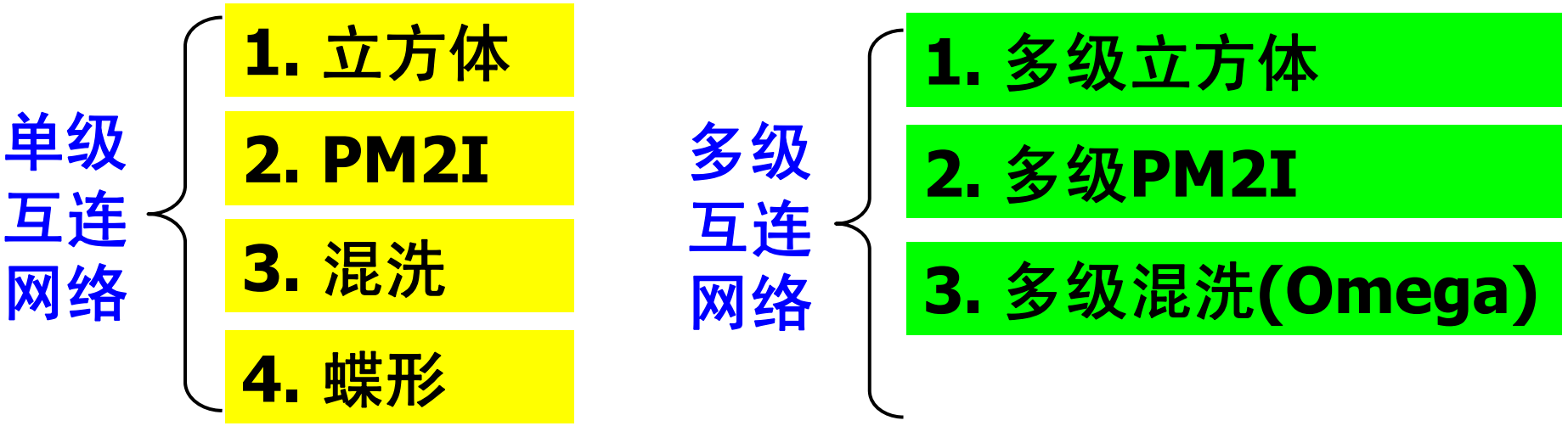
- 多级交换网络
  - 级间互连模式
    - 混洗、蝶式、纵横开关及立方体连结等
  - 控制方式
    - 级控制：每级只有一个控制信号
    - 单元控制：每个开关一个控制信号
    - 部分级控制：几个开关合用一个控制信号
  - 多级交换网络是总线和交叉开关的折衷。它的主要优点在于采用模块结构，可扩展性好（>64）

MIMD和SIMD计算机都使用多级网络

Ω多级交换网络



动态互连网络

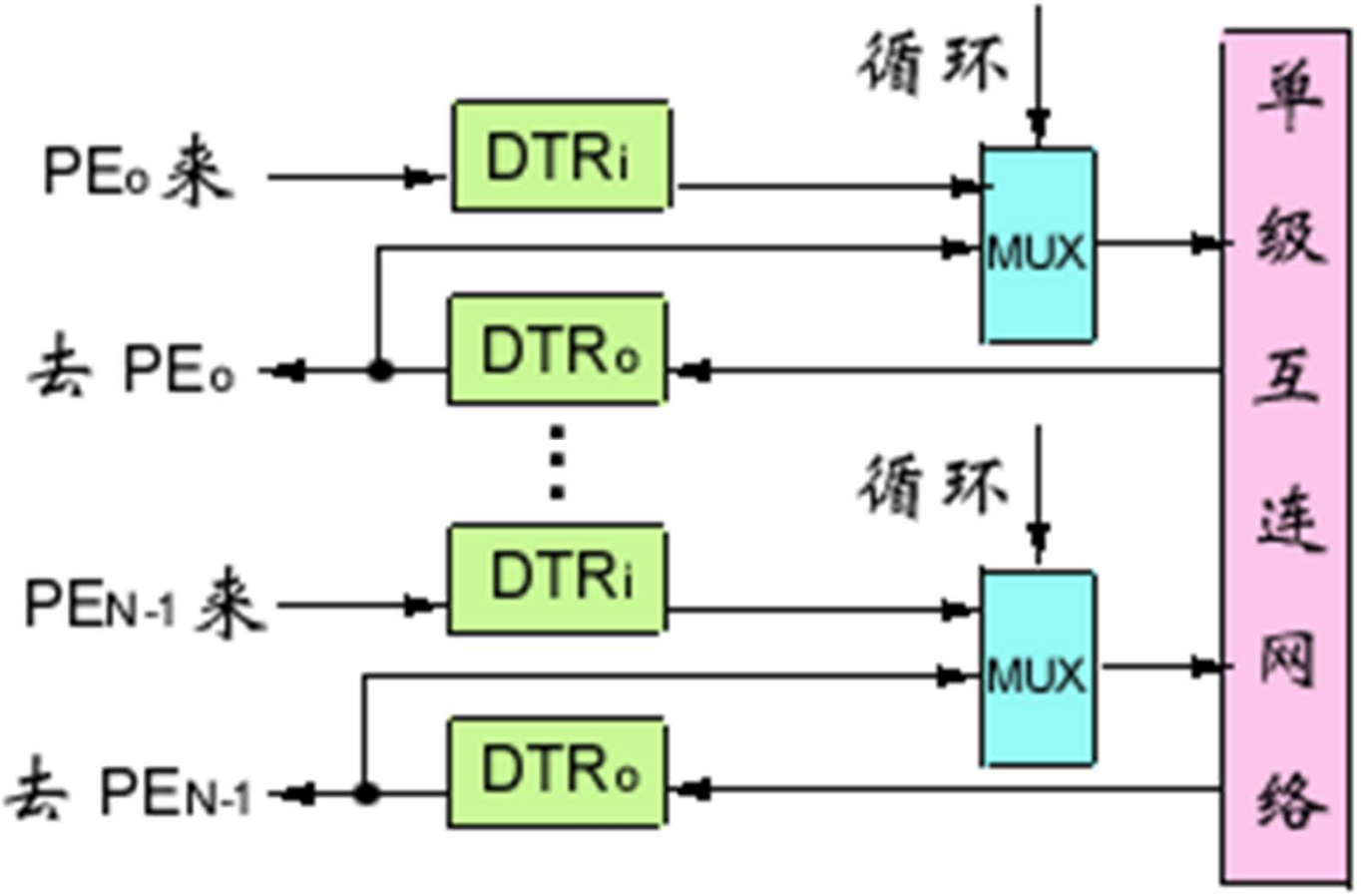

单级
互连
网络
1. 立方体
2. PM2I
3. 混洗
4. 蝶形
多级
互连
网络
1. 多级立方体
2. 多级PM2I
3. 多级混洗(Omega)

6.4.2 循环互连网络

- 单级互连网络只能实现有限几种基本连接，并不能实现任意处理器之间的互连
- 有两种解决办法：
  - 循环互连网：多次重复使用同一个单级互连网络
  - 多级互连网：将多套相同单级互连网络连接起来
- 前一种方法是牺牲时间换取设备
- 后一种方法是以设备换取时间

6.4.2 循环互连网络

循环互连网络与多级互连网络相比，节省了重复的设备，但加长了通过时间。



循环互连网络的模型

6.2.3 影响互联网络的因素

- 循环网络与多级网络的比较

循环网络	多级网络
节省设备	增加设备和成本
通过时间加长	缩短了通过时间，提高了速度
对网络控制部分要求高	可以灵活组合单级网络，形成具有不用特性和连接模式的多级网络

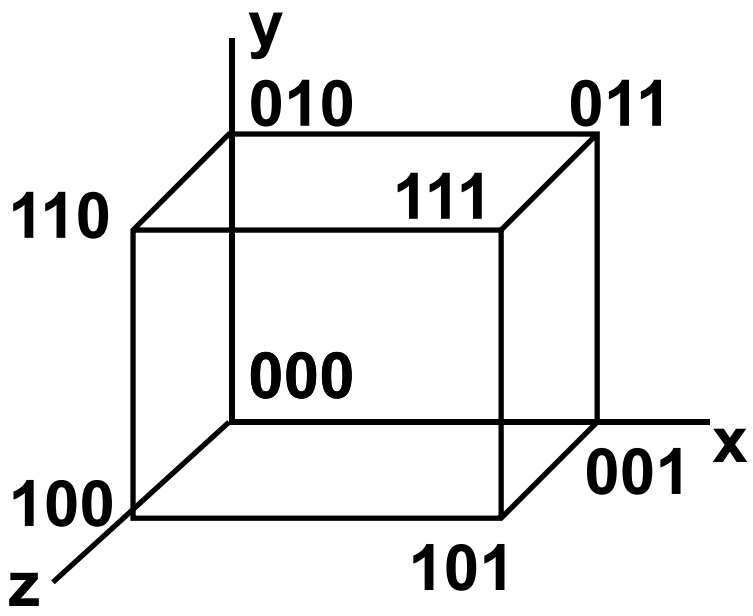
- 目前，绝大多数阵列处理机都采用多级网络或多级循环网络

6.4.3 基本的单级互连网络

- 介绍 4 种基本的单级互连网络：
  - 立方体 单级互连网络
  - **PM2I** 单级互连网络
  - 混洗交换 单级互连网络
  - 蝶形 单级互连网络

1. 立方体单级互连网络

- 立方体单级互连网络源于立方体结构

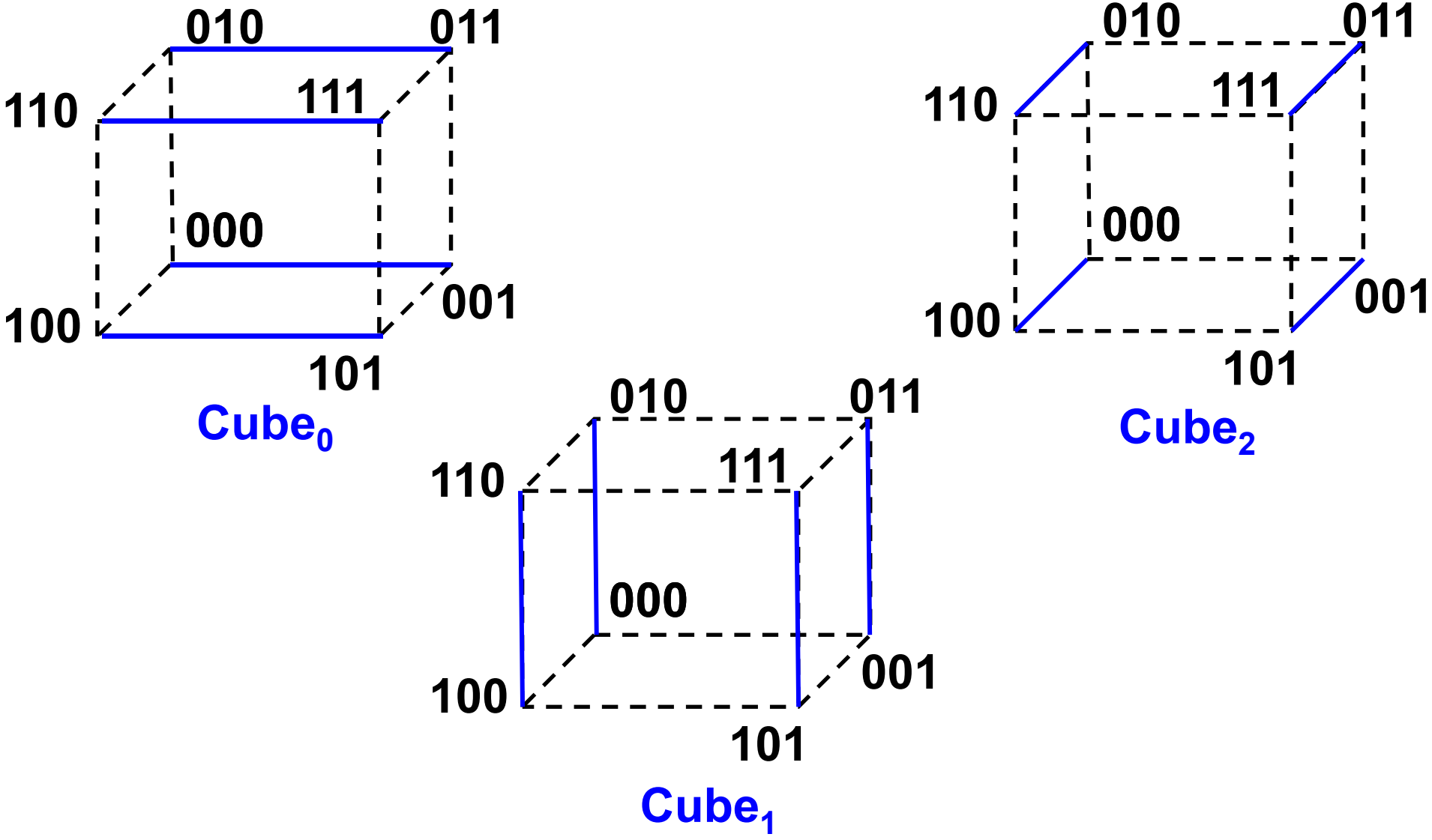


1. 立方体单级互连网络

- 每个顶点代表一个PE，共有8个PE
- 每个PE只能直接连接到与其二进制编码的某一位取反的其他3个PE上
- 立方体单级互连网络有3个互连函数：
  - $Cube_0$ ： $Cube_0(P_2P_1P_0) = P_2P_1\overline{P_0}$
  - $Cube_1$ ： $Cube_1(P_2P_1P_0) = P_2\overline{P_1}P_0$
  - $Cube_2$ ： $Cube_2(P_2P_1P_0) = \overline{P_2}P_1P_0$

如 010 只能连到 000、 011、110，不能直接连到对角线上的 001、100、101、111

1. 立方体单级互连网络


010
011
110
111
000
001
100
101
Cube0
010
011
110
111
000
001
100
101
Cube2
010
011
110
111
000
001
100
101
Cube1

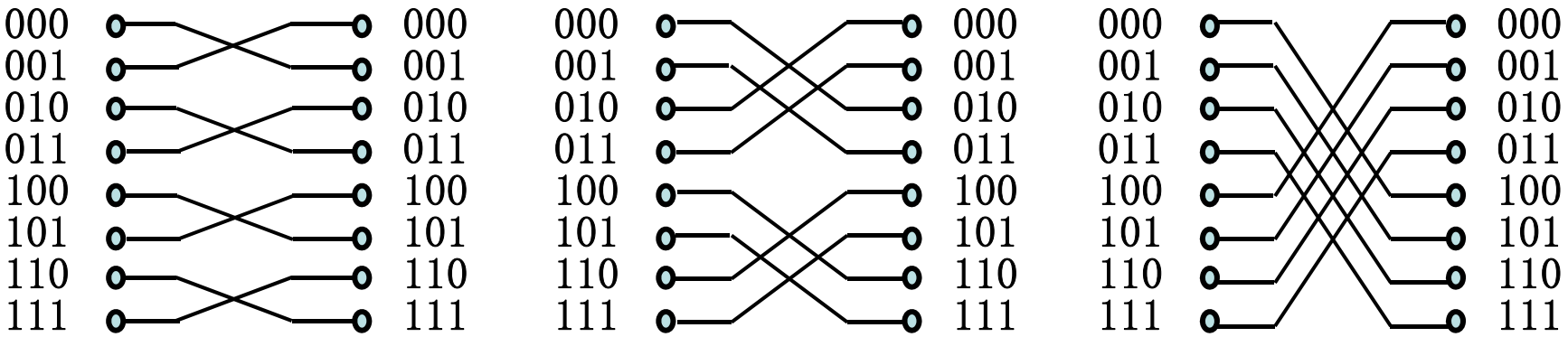
1. 立方体单级互连网络

- 推广到n维情况。N个节点的立方体单级互连网络共有$n=\log_2 N$种互连函数，即：

$$Cube_i(P_{n-1}\ldots P_i\ldots P_1P_0)=P_{n-1}\ldots\overline{P_i}\ldots P_1P_0$$

- 当n>3时，称之为超级立方体网络
- 显然，超级立方体网络最大距离为n，即反复使用单级网络，最多经过n次就可以实现任意一对入端出端的连接，而且任意两个节点之间至少有n条不同的路径，容错性很强

1. 立方体单级互连网络

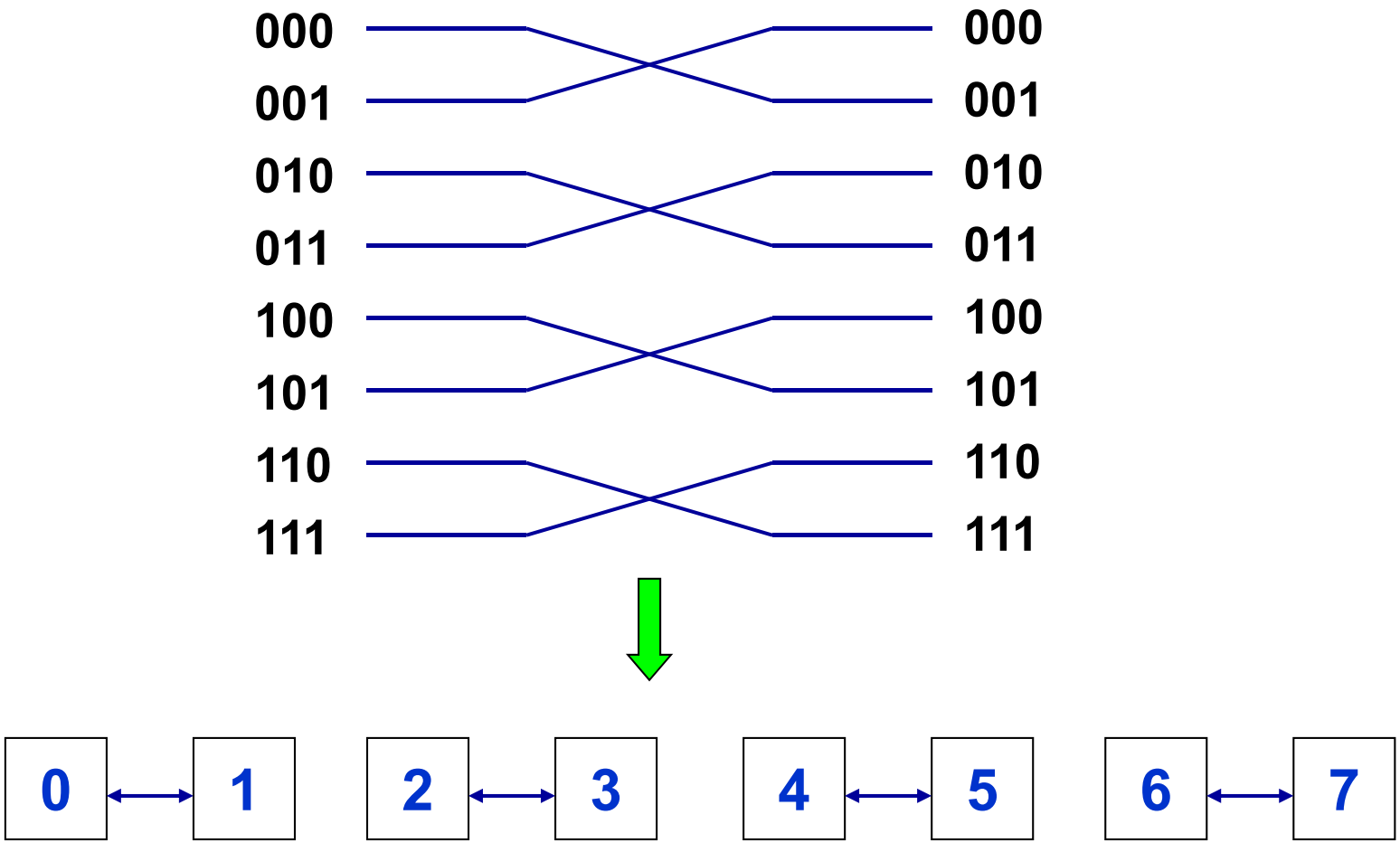


N=8 $Cube_0$置换互连　　N=8 $Cube_1$置换互连　　N=8 $Cube_2$置换互连

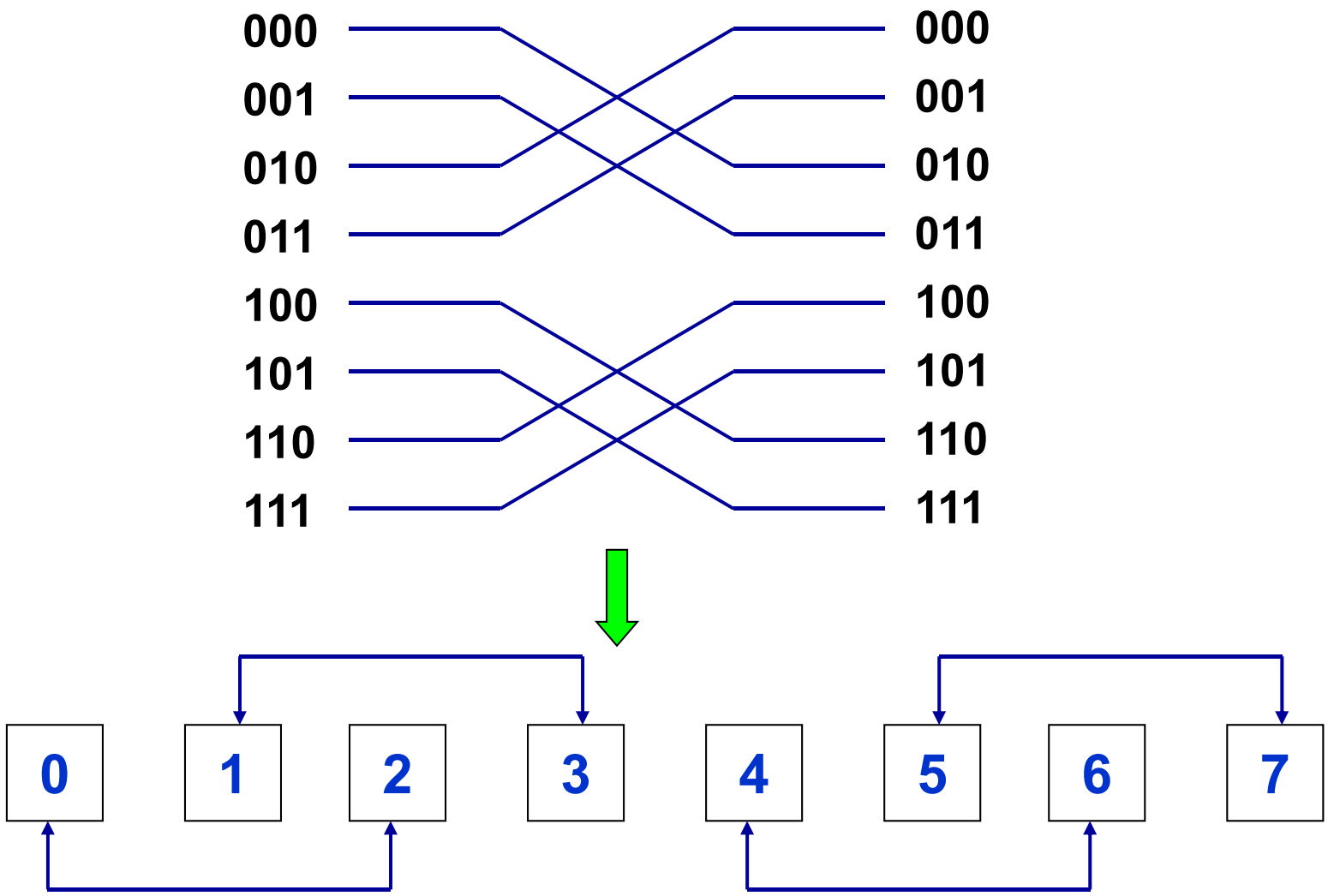
Cube0 : (0,1) (2,3) (4,5) (6,7)
Cube1 : (0,2) (1,3) (4,6) (7,7)
Cube2 : (0,4) (1,5) (2,6) (3,7)

- **互连特性：互连函数可逆，有n种 ($Cube_0$，…，$Cube_{n-1}$) 变换**

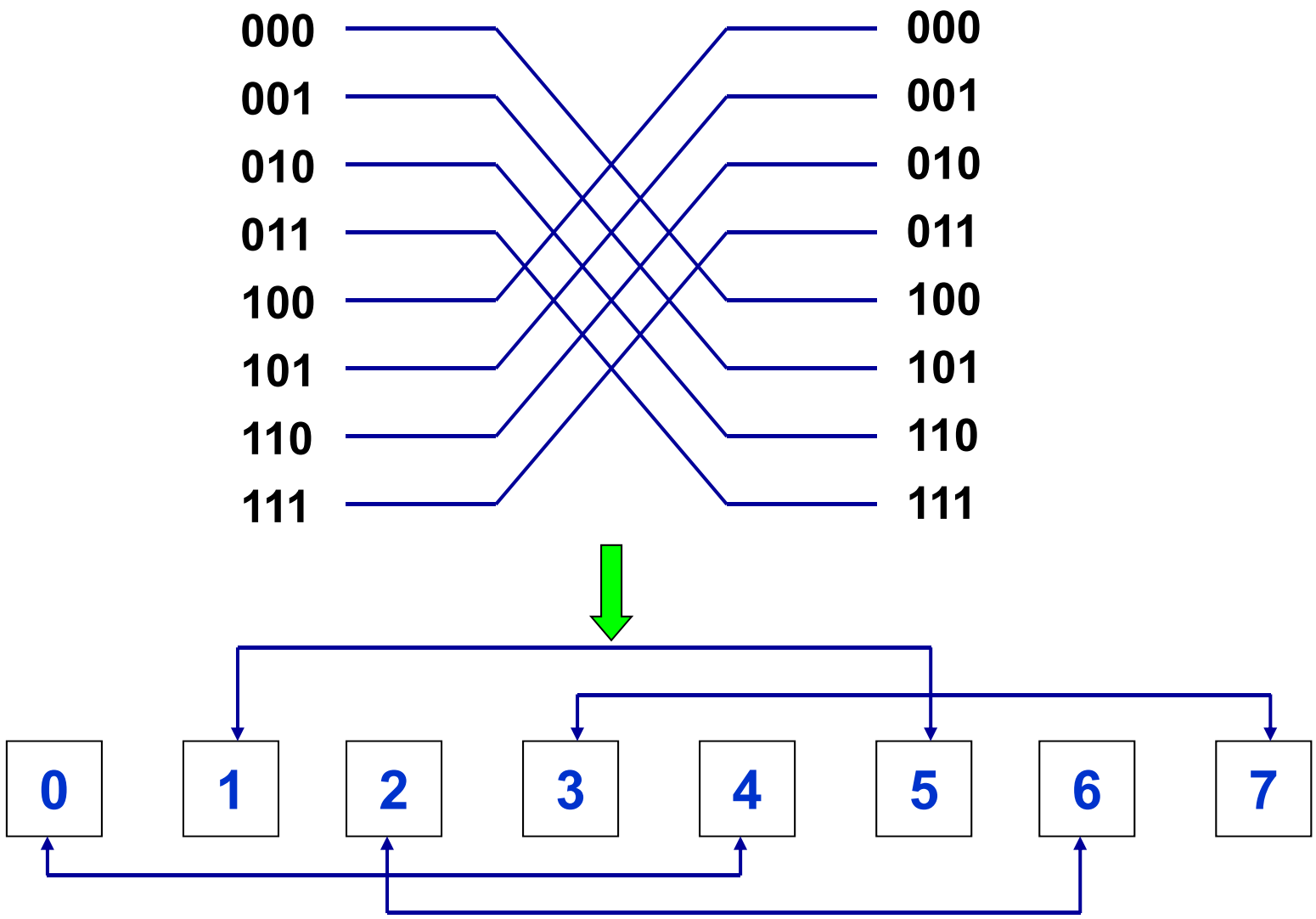
$Cube_0$: $cube_0(X_2X_1X_0) = (X_2X_1\overline{X_0})$



$Cube_1$: $cube_1(X_2X_1X_0) = (X_2\overline{X_1}X_0)$



$Cube_2$: $cube_2(X_2X_1X_0) = (\overline{X_2}X_1X_0)$



2. PM2I单级互连网络

- **PM2I是Plus-Minus 2（加减2）单级网络的简称，能实现j号PE直接与j±2号PE连接，即**

$$PM2_{+i}(j) = j + 2^i \bmod N$$

$$PM2_{-i}(j) = j + 2^i \bmod N$$

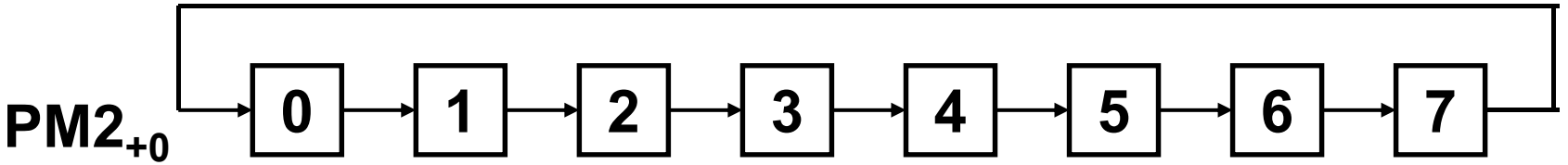
$$\text{其中} 0 \le j \le N, \quad 0 \le i \le n, \quad n = \log_2 N$$

- **因此，它共有2n个互连函数**

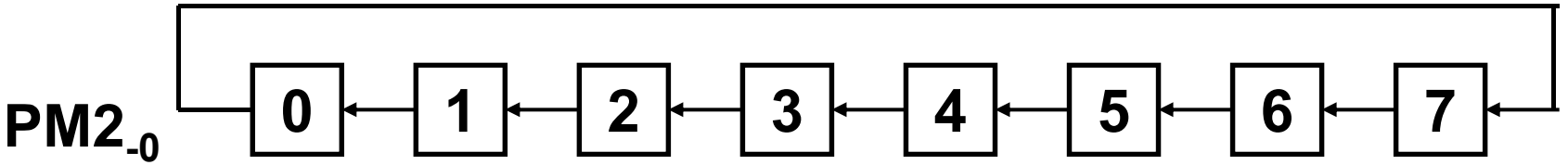
2. PM2I单级互连网络

- 由于总存在 $PM2_{+(n-1)} = PM2_{-(n-1)}$，所以PM2实际共有2n-1个互连函数
- 对于N=8的情况，n=3，共有5种互连函数：

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

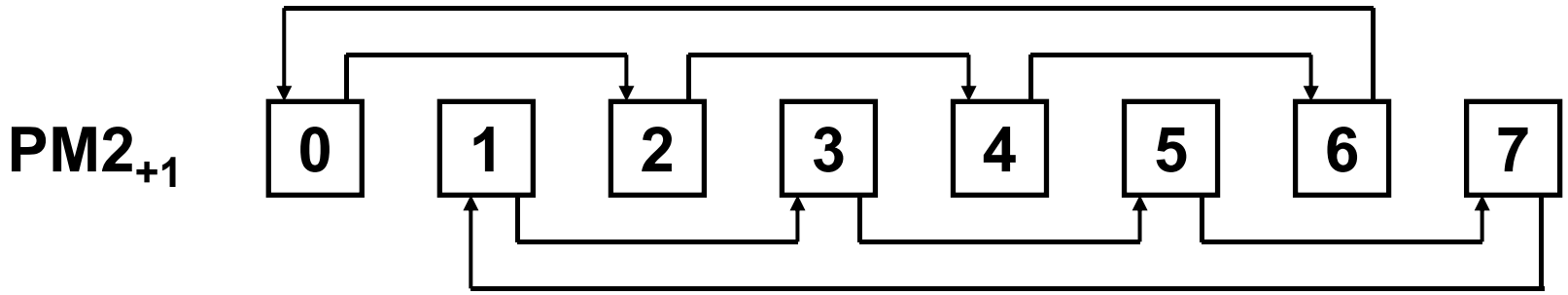


$PM2_{-0}$: (7 6 5 4 3 2 1 0)

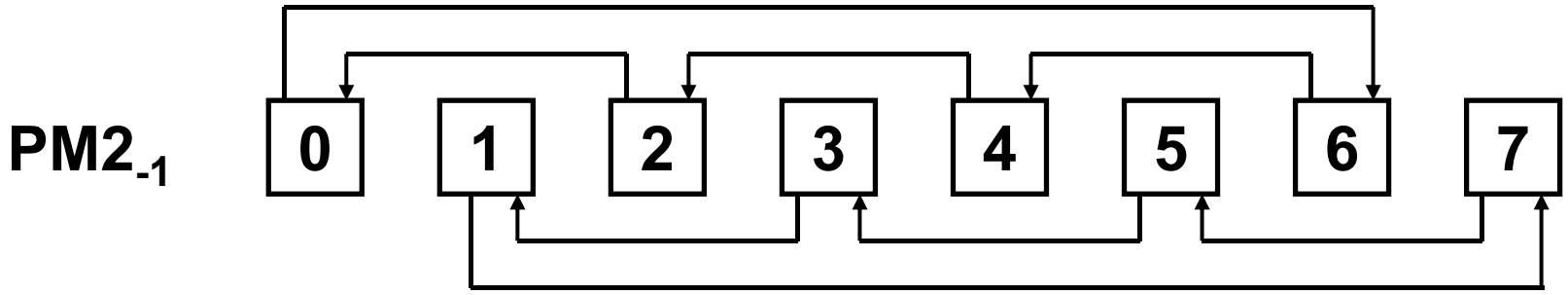


2. PM2I单级互连网络

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

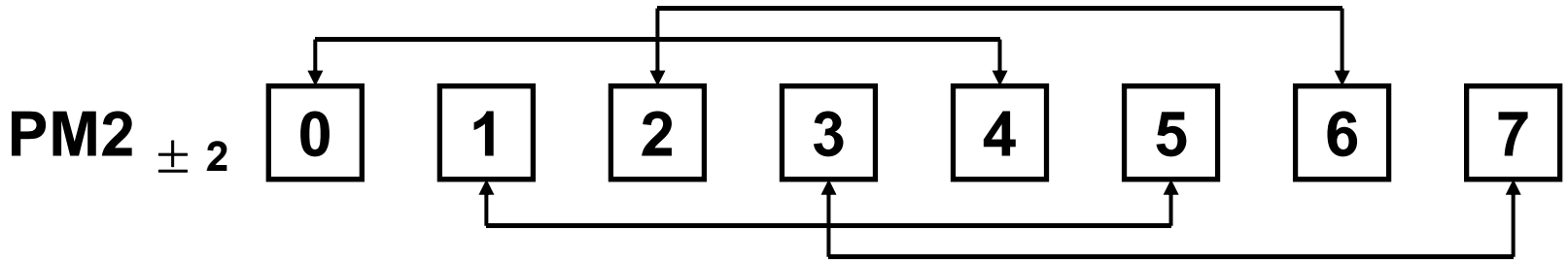


$PM2_{-1}$: (6 4 2 0)(7 5 3 1)



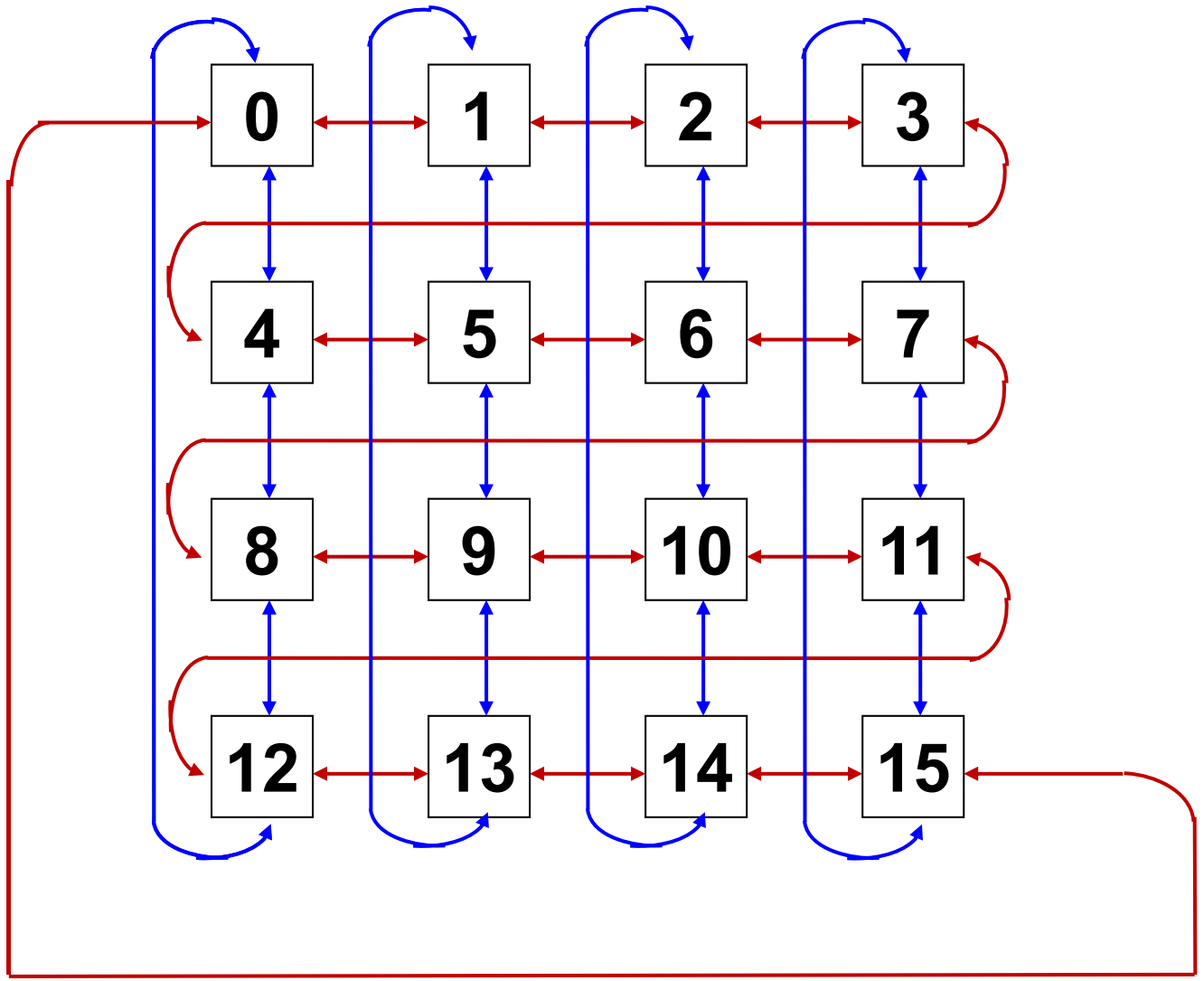
2. PM2I单级互连网络

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



2. PM2I单级互连网络

- 单向环网或双向环网可以看成是PM2I网络的特例，它仅使用了其中的$PM2_{+0}$、$PM2_{-0}$或$PM2_{\pm 0}$互连函数。
- 不难看出， ILLIAC IV 处理单元的互连也是PM2I互连网络的特例，只采用了其中的$PM2_{\pm 0}$和 $PM2_{\pm (n/2)}$（即$PM2_{\pm 3}$）4 个互连函数。



上面的网络可以用4个PM2I函数表示

$PM2_{+0}$: (0 1 2 ... 15)

$PM2_{-0}$: (15 14 13 ... 0)

$PM2_{\pm2}$: (0 4) (1 5) (2 6) (3 7)
(4 8) (5 9) (6 10) (7 11)
(8 12) (9 13) (10 14) (11 15)
(12 0) (13 1) (14 2) (15 3)

2. PM2I单级互连网络

- PM2I比立方体单级网络灵活。
  - 例如：0可以直接连接到1、2、4、6、7，而在立方体单级网络中，0只能直接连接到1、2、4
- PM2I单级网络的最大距离为[n/2]。
- 对于n=3，最多使用两次，既可以实现任意一对入端出端的连接

3. 混洗交换单级互连网络

- 混洗交换(shuffle-Exchange)单级互连网络包含两个互连函数：
  - 全混(Perfect Shuffle)
  - 交换(Exchange)
- 先混洗，再交换

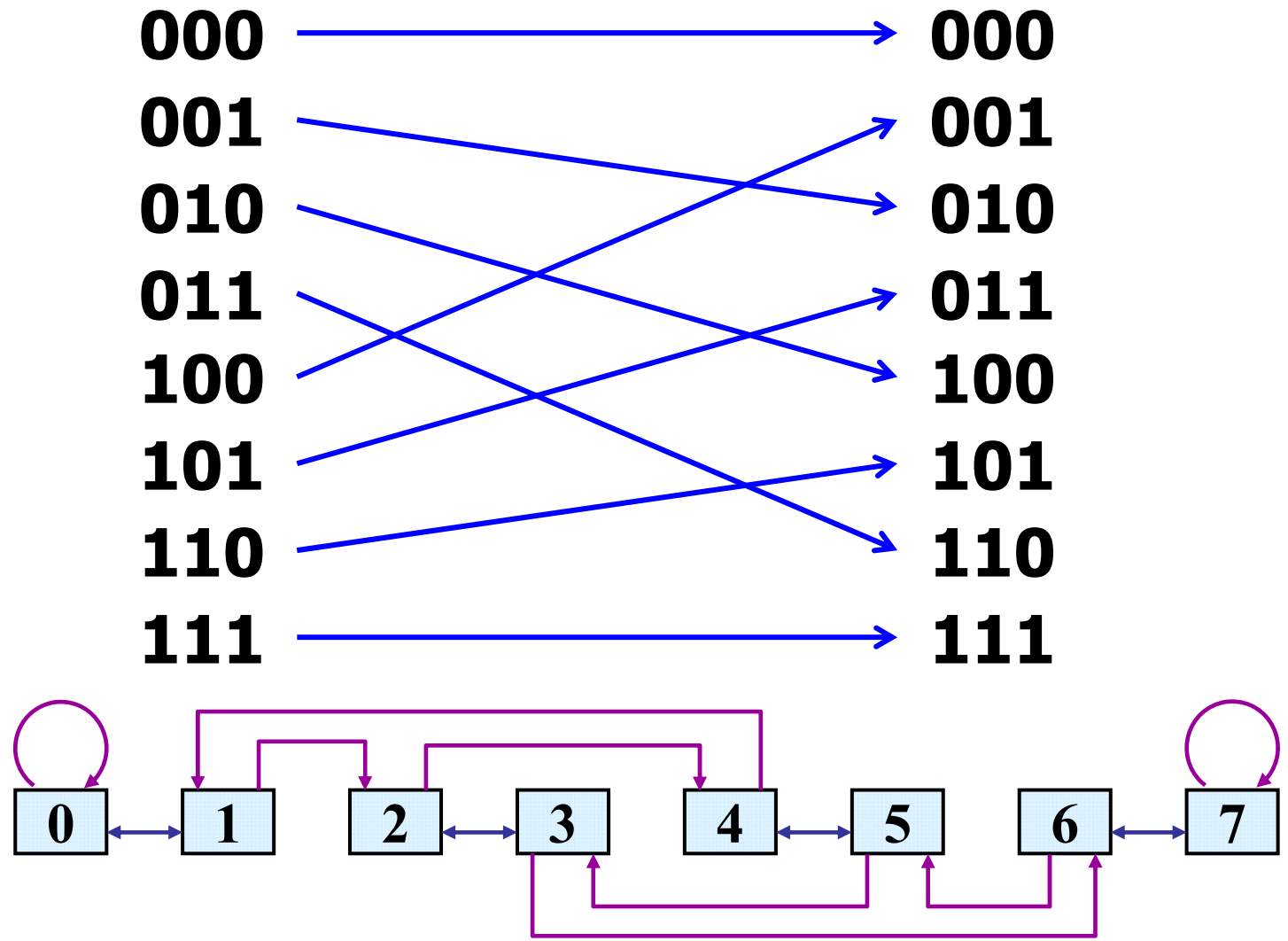
3. 混洗交换单级互连网络

- 全混的连接规律是：
  - 把全部按编码顺序排列的 PE 从中间分为两半，前一半和后一半在连接至输出端时正好一一隔开，如同洗扑克牌一样
  - 其互连函数为（左移）：

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

其中 $n=\log_2 N$

3. 混洗交换单级互连网络



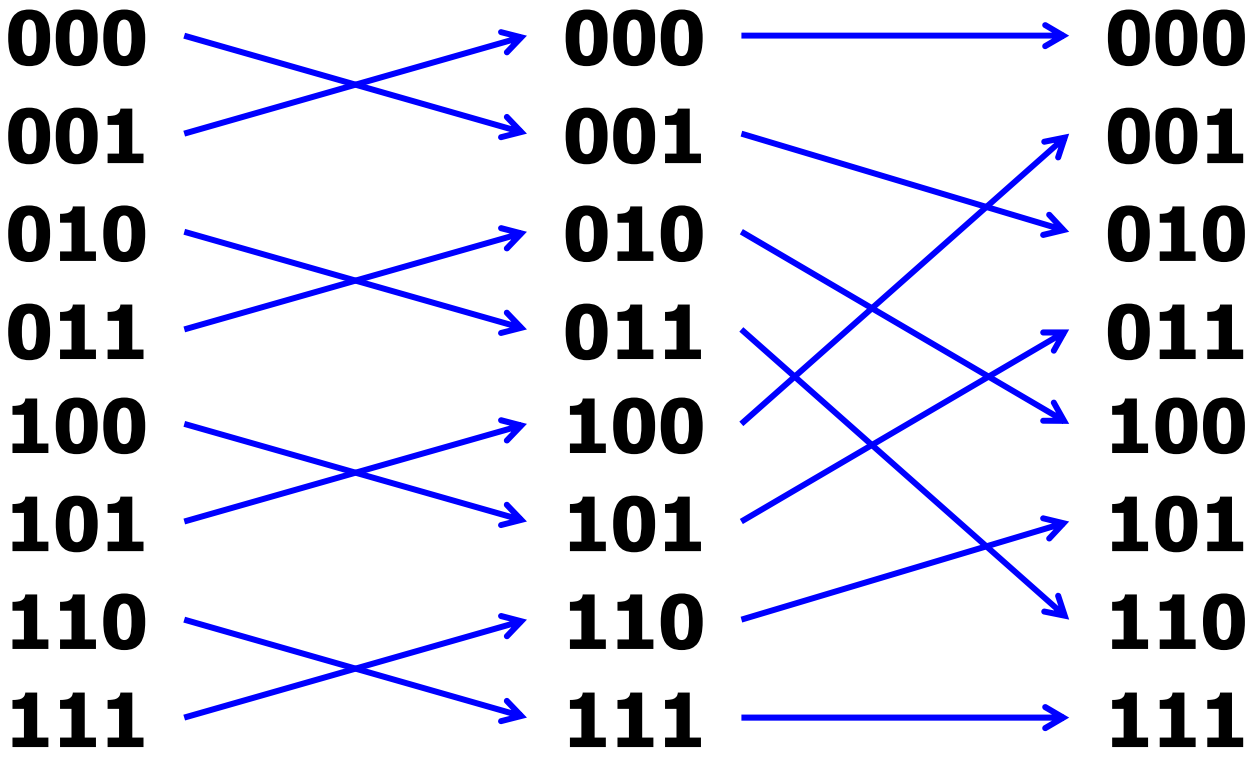
3. 混洗交换单级互连网络

- **Shuffle函数有两个重要特征(1/2):**
  - **Shuffle函数是不可逆的**
    - 若把入端当出端，出端当入端，则原网络变成另外一个网络
  - **如果把它再做一次Shuffle函数变换，得到一组新的编码，即$P_{n-3}...P_1P_0P_{n-1}P_{n-2}$。每全混一次，新的最高位就被移至最低位。**
    - 经过n次后，全部N个PE又恢复到最初的排列。
    - 在多次全混过程中，除了全0和全1 PE外，每个PE都有与其他PE连接的机会。

3. 混洗交换单级互连网络

- 为实现全0和全1 PE与其他PE的任意连接，还必须增加$Cube_0$交换函数，这样就得到了全混交换单级网络，也称为均匀混洗
- 全混交换单级网络的最大距离为 2n-1

3. 混洗交换单级互连网络



$Cube_0$ 函数　　混洗函数

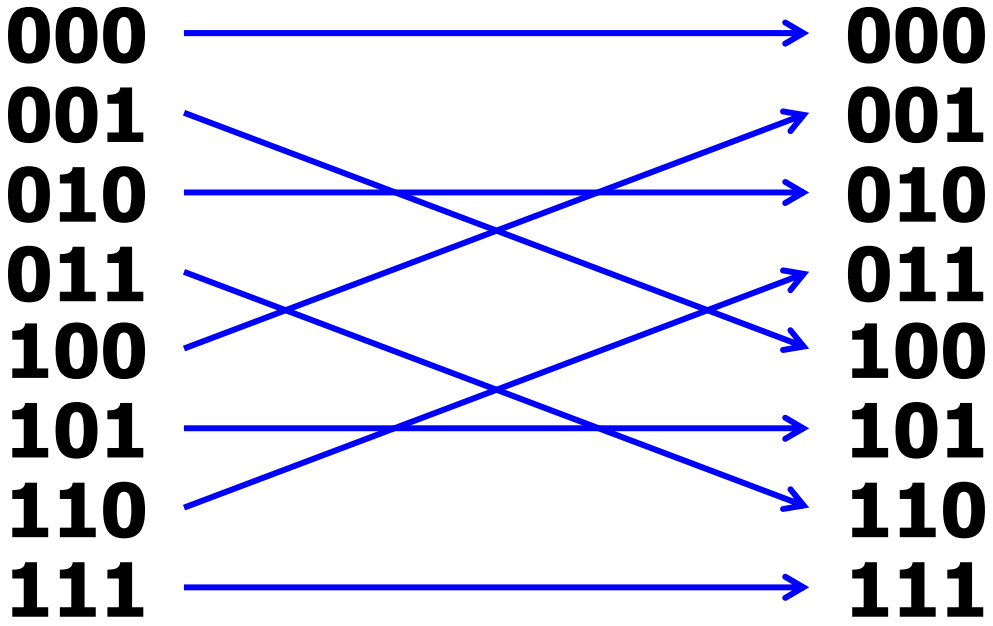
全混交换单级网络，也称为均匀混洗

4. 蝶形单级互连网络

- 互连函数

$$Butterfly(P_{n-1}P_{n-2}\cdots P_1P_0)=P_0P_{n-2}\cdots P_1P_{n-1}$$

- 将二进制编码的最高位和最低位交换



例：编号为0、1、…、15的16个处理器用单级互连网络互连。当互连函数分别为：

1)$Cube_3$

2)$PM2+_3$

3)$PM2-_0$

4)Shuffle

5)Shuffle(Shuffle)

时，第13号处理器各连至哪一个处理器？

[分析] 编号为0、1、…、15的16个处理器，其号可用4位二进制码$P_3P_2P_1P_0$表示。第13号处理器二进制编号为1101。

$Cube_3$实现将处理器号为$P_3P_2P_1P_0$与处理器号为$\overline{P_3}P_2P_1P_0$的数据进行交换。

$PM2+_3$实现将j号处理器数据送至第（$j+2^3 \bmod 16$）处理器上。

$PM2-_0$实现将j号处理器数据送至第（$j-2^0 \bmod 16$）处理器上。

Shuffle实现将处理器号为$P_3P_2P_1P_0$的信息送至处理器号为$P_2P_1P_0P_3$的处理器上。

Shuffle（Shuffle）实现将处理器号为$P_3P_2P_1P_0$的信息送至处理器号为$P_1P_0P_3P_2$的处理器上。

[解答]

1）第1101处理器连至0101处理器号上，即第5号处理器上。

2）第13号处理器数据送至第（$13+2^3 \bmod 16$）处理器上，即第5号处理器上。

3）第13号处理器数据送至第（$13-2^0 \bmod 16$）处理器上，即第12号处理器上。

4）第1101处理器连至1011处理器号上，即第11号处理器上。

5）第1101处理器连至0111处理器号上，即第7号处理器上。

6.4.4 多级互连网络

- 能够实现结点到结点之间的任意互连是互连网络的一种基本功能。
- 多级互连网络采用多个相同的或不同的互连网络直接连接起来。属于组合逻辑线路，一个时钟周期就能够实现任意结点到结点之间的互连。

1. 多级互连网络关键技术

- 多级互连网络采用的关键技术：
  - (1) 交换开关
  - (2) 交换开关之间的拓扑连接
  - (3) 对交换开关的控制方式

交换开关

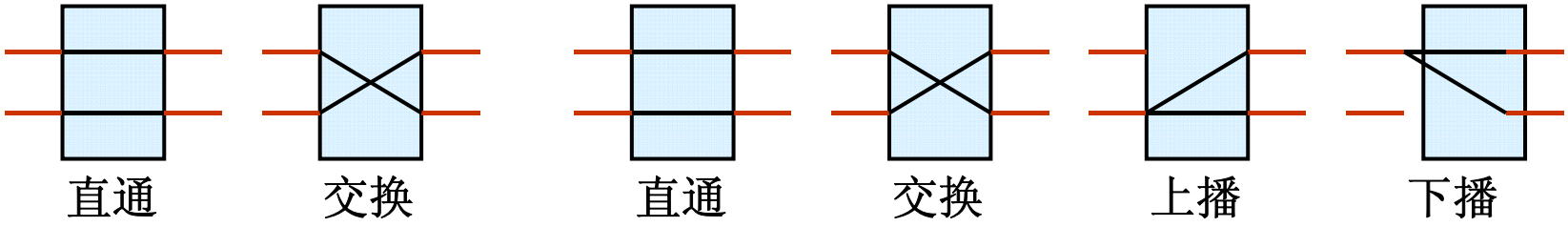
- 一个a×b交换开关有a个输入和b个输出

  最常用的二元开关：a=b=2

- 每个输入可与一个或多个输出相连，但是在输出端必须避免发生冲突（多个输入同时连接到一个输出）。
- 一对一和一对多映射是容许的；但不容许有多对一映射（冲突）。
- 只容许一对一映射时称为置换连接，称这种开关为n×n交叉开关。

交换开关

- 具有直通和交换两种功能的交换开关称为二功能开关，或交换开关。用 1位控制信号控制。
- 具有所有四种功能的交换开关称为四功能开关，用 2 位控制信号控制。



(a) 1位控制信号　　(b) 2位控制信号

直通实现恒等置换

交换开关

- 不允许交叉开关的2个入端同时连接到同一个出端 – 会发生信息传送冲突
- 允许 $i_{入}$ 连 $j_{入}$， $i_{出}$ 连 $j_{出}$， 称此为返回
  - 用于实现入端与入端相连，出端与出端相连，从而将N个入端和N个出端的网络变为2N个处理单元的网络

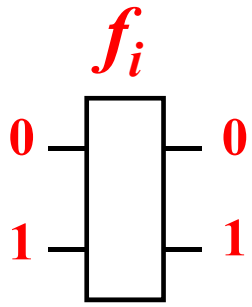
交换开关之间的拓扑连接

- 前一级交换开关的输出端与后一级交换开关的输入端之间的连接模式称为拓扑结构
- 通常采用前面介绍的互连函数实现拓扑结构
- 实际上，从结点的输出到第一级交换开关的输入，以及从最后一级交换开关的输出到结点的输入也可以采用拓扑结构连接

交换开关的不同控制方式

- 在多级互连网络中有多级交换开关，每一级又有多个交换开关。通常有三种控制方式：
  - (1) 级控制：同一级交换开关使用同一个控制信号控制。
  - (2) 单元级控制：每个交换开关分别控制。
  - (3) 部分级控制：例如，第i级使用i+1个控制信号控制 ($0 \le i \le n-1$)。
- 同一个多级互连网络分别常用三种不同的控制方式，可以构成三种不同的互连网络

交换开关的不同控制方式



按级控制和交换置换(1/2):

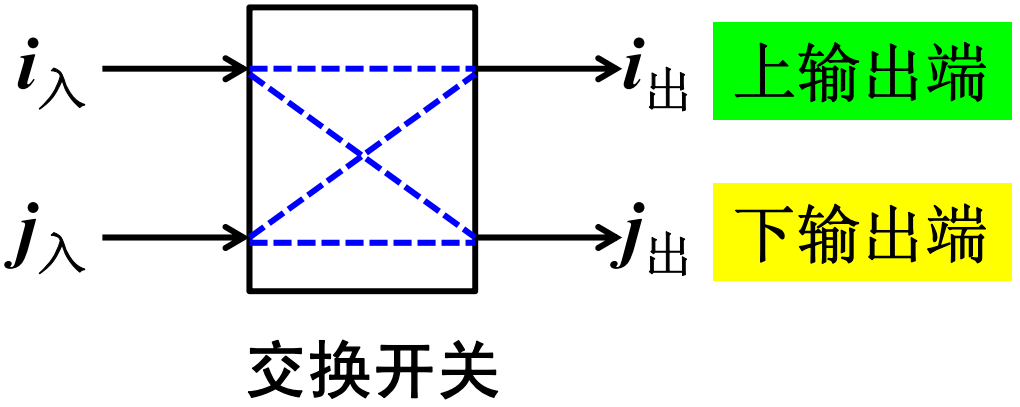
- 在右上图有一个开关控制示意图。假定开关的两个输入端分别标注以“0”和“1”，同样也给输出端标上“0”和“1”的标注。
- 在直送方式下，0→0，1→1；
- 在交叉方式下，则有0→1，1→0。
- 如果将这种传输情况看作二进制运算，那末控制信号 f_i 就是参与逻辑运算的一个变量，其逻辑关系可以表示为：

$$E(x_i) = x_i \oplus f_i$$

f_i =0时，表示直送； f_i = 1 时，表示交叉。

交换开关的不同控制方式

- 按级控制和交换设置(2/2)：
  - 采用单元控制方式时，控制信号为0时，2×2开关的输入端与上输出端连接；为1时，输入端与下输出端连接。



- 例子：
  - UIUC的Cedar
  - IBM的RP3
  - NYU的UltraComputer

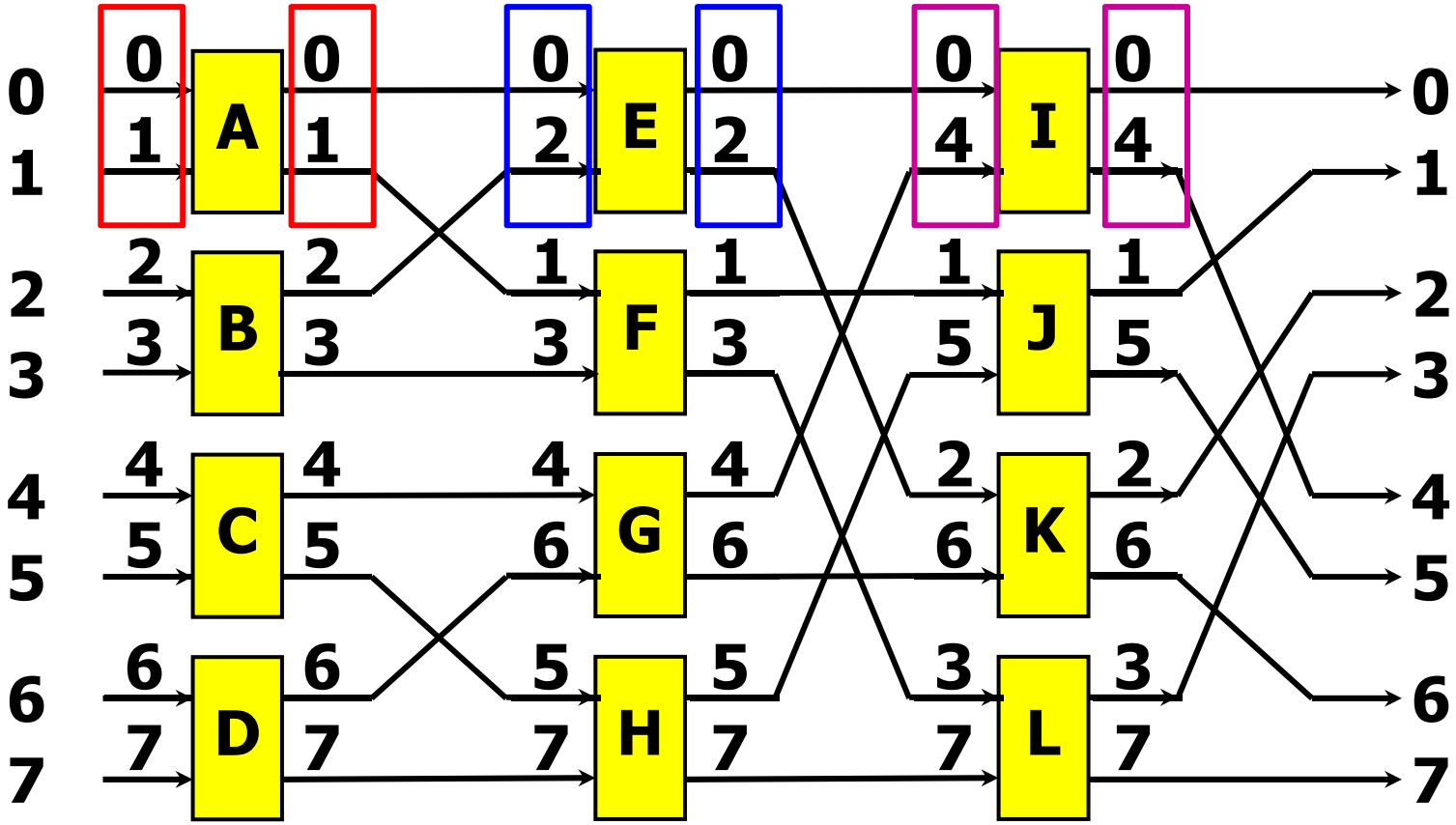
2. 多级立方体网络

- 单级立方体网络的最大距离为n。即重复n次使用单级互连网络，可以实现任意两个结点的连接。
- 因此，用n个单级立方体网络相串联，让每一级实现一种互连函数$Cube_i$，其中i=0，1，... n-1，则同样能实现任意两个结点的连接。

2. 多级立方体网络

- 应用在巨型机STARAN中
- 采用2x2 二功能开关，立方体拓扑结构
- 画法：
  - 计算立方体网络的级数 $n=\log_2 N$
  - 每级有 N/2 个二功能交叉开关。整个网络开关数 $(N/2)\log_2 N$
  - 第 i 级的两个入端和出端按交换函数$Cube_i$ 配对编号
  - 将各级交叉开关的相同编号的各端相连

2. 多级立方体



N=8的多级立方体网络

2. 多级立方体网络

- 当A、B、C、D四个开关交换，其余直通时实现 $\mathbf{Cube_0}$ 互连函数。
- 当E、F、G、H四个开关交换，其余直通时实现 $\mathbf{Cube_1}$ 互连函数。
- 当I、J、K、L四个开关交换，其余直通时实现 $\mathbf{Cube_2}$ 互连函数。

2. 多级立方体网络

- 根据控制方式的不同，多级立方体网络有STARAN网络、间接二进制 n方体网络等
  - 两者仅在控制方式上不同，在其他方面都是一样的。
  - 都采用二功能（直送和交换）的2×2开关。
  - 当第i级（0≤i≤n-1）交换开关处于交换状态时，实现的是$Cube_i$互连函数。

2. 多级立方体网络

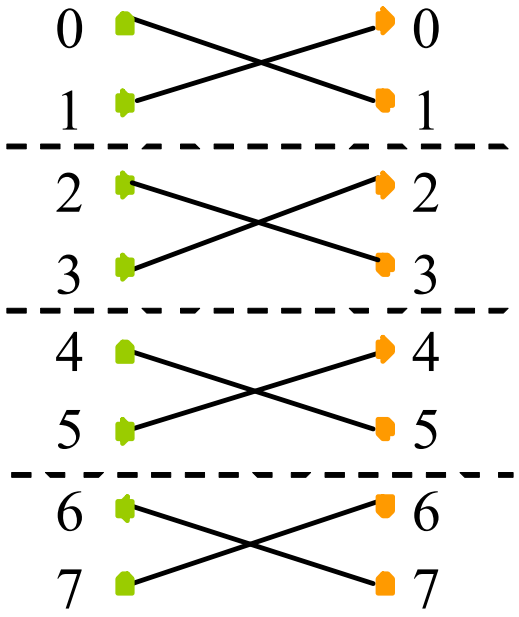
- STARAN网络是多级互连网络中已经获得成功应用的一种，它因实际用于巨型相联处理机STARAN而得名
- STARAN网络开关控制方式有2种：级控方式和组控方式
  - 采用级控制可以构成STARAN交换网
  - 采用部分级控制可以构成STARAN移数网

2. 多级立方体网络

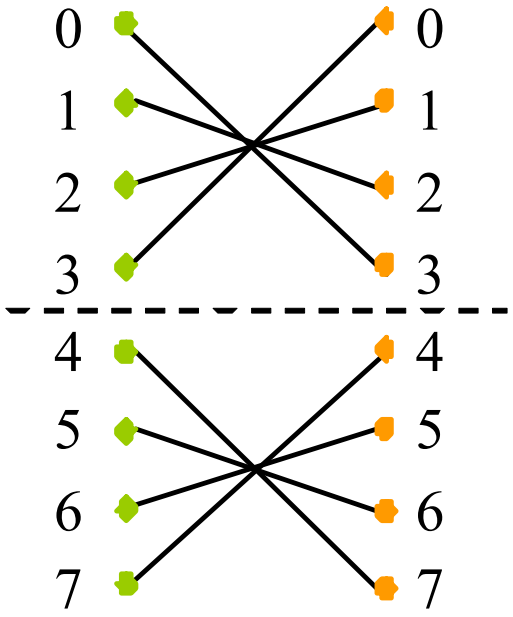
- **STARAN网络用作交换网络时，采用级控制，实现的是交换函数。所谓交换(Flip)函数，是将一组元素首尾对称地进行交换：**

  如果一组元素包含有 2^s 个，则它是将所有第k个元素与第$(2^s - (k+1))$个元素互相交换。

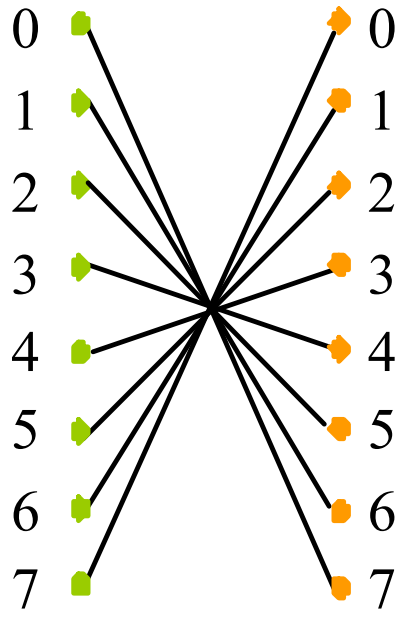
2. 多级立方体网络：交换网络



(a) 4 组 2 元交换



(b) 2 组 4 元交换



(c) 1 组 8 元交换

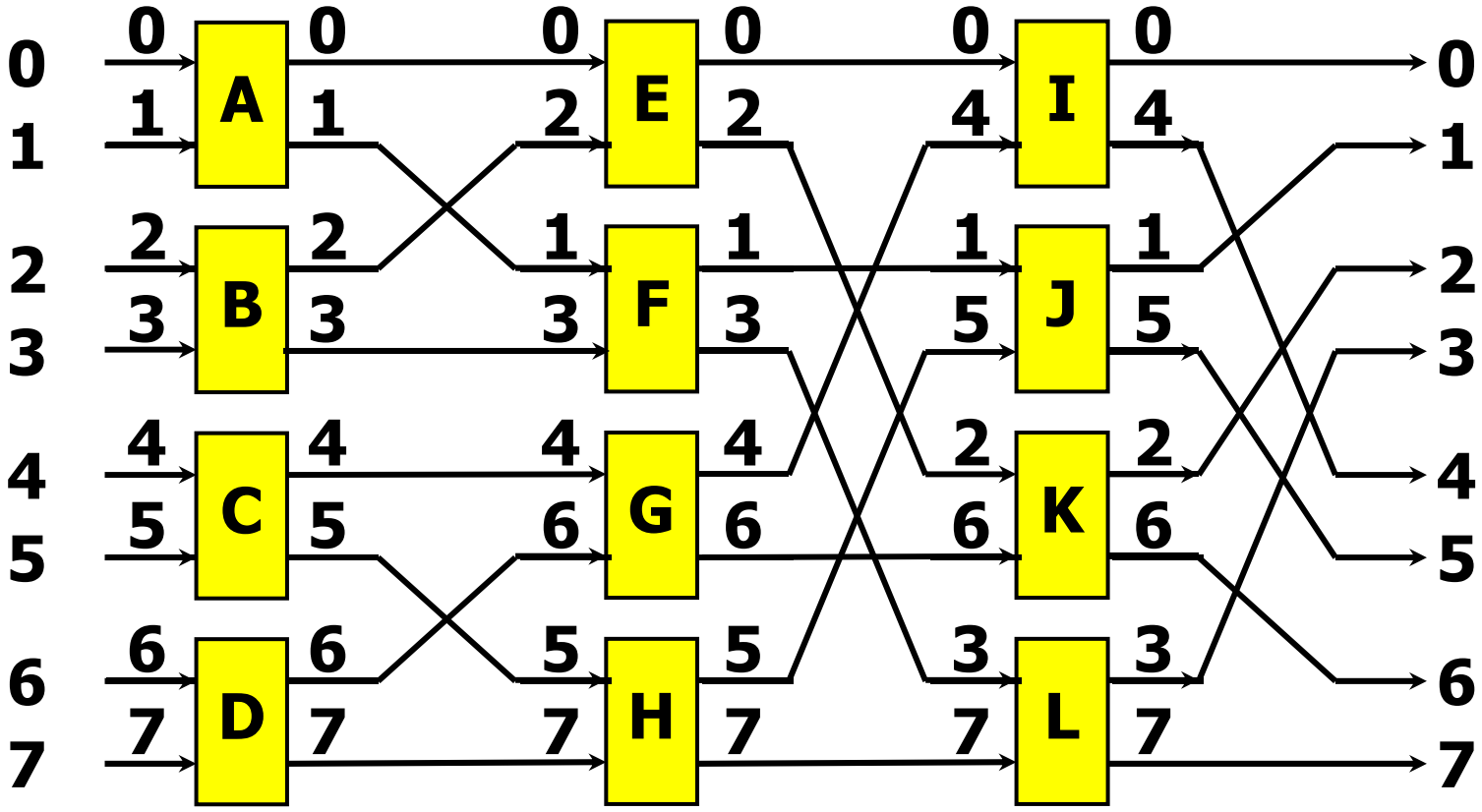
8个元素的基本交换图形

2. 多级立方体网络：交换网络

- 三级STARAN网络交换网络就是将$Cube_0$、$Cube_1$、$Cube_2$三种互连函数的单级立方体网络串联起来，采用不同的级控制信号可以实现任何一个入端到任何一个出端的之间连接。
- 设$K_2k_1k_0$为控制信号，k_i（i=0，1，2）为第i级的级控制信号。
  - 该控制信号为"1"时，该级所有的交换单元同时处于交换状态，出端编号是从入端编号在相应位代码上变反；
  - 当控制信号为"0"时，该级所有处理单元同时处于直连状态，代码不变。

置换为：E(Xn-1Xn-2…X1X0)=(Xn-1 ⊕ fn-1, Xn-2 ⊕ fn-2,…, X1 ⊕ f1, X0 ⊕ f0)

2. 多级立方体网络：交换网络



N=8的STARAN网络

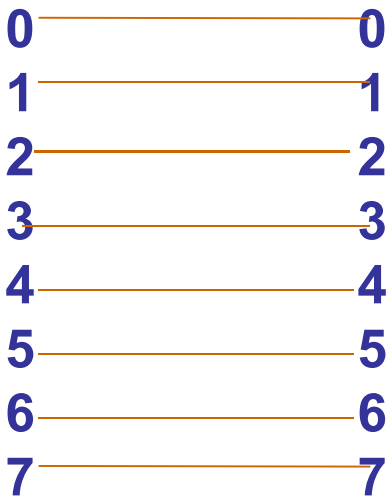
ki 实现Cubei 置换

		级控制信号（$k_2k_1k_0$）							
		000	001	010	011	100	101	110	111
入端	0	0	1	2	3	4	5	6	7
	1	1	0	3	2	5	4	7	6
	2	2	3	0	1	6	7	4	5
	3	3	2	1	0	7	6	5	4
	4	4	5	6	7	0	1	2	3
	5	5	4	7	6	1	0	3	2
	6	6	7	4	5	2	3	0	1
	7	7	6	5	4	3	2	1	0
功能		i	$Cube_0$	$Cube_1$	$Cube_0$ + $Cube_1$	$Cube_2$	$Cube_0$ + $Cube_2$	$Cube_1$ + $Cube_2$	$Cube_0$ + $Cube_1$ + $Cube_2$

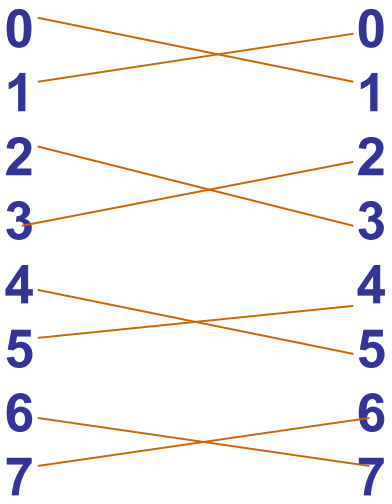
置换为：$E(Xn-1Xn-2\ldots X1X0)=(Xn-1 \oplus fn-1, Xn-2 \oplus fn-2,\ldots, X1 \oplus f1, X0 \oplus f0)$

ki 实现Cubei 置换

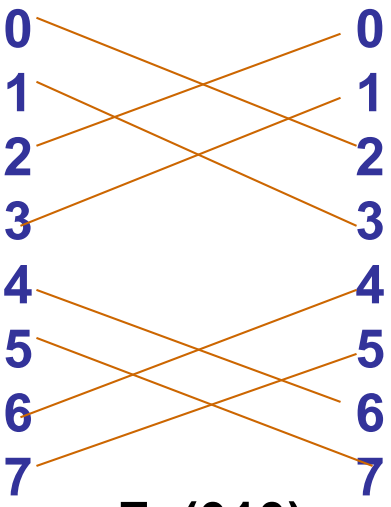
级控制信号 $k_2k_1k_0$	连接的输出端号序列（入端号序列：01234567）	实现的分组交换	实现的互连函数
000	0 1 2 3 4 5 6 7	恒等	I
001	1 0 3 2 5 4 7 6	4组2元交换	$Cube_0$
010	2 3 0 1 6 7 4 5	4组2元交换＋ 2组4元交换	$Cube_1$
011	3 2 1 0 7 6 5 4	2组4元交换	$Cube_0$＋$Cube_1$
100	4 5 6 7 0 1 2 3	2组4元交换＋ 1组8元交换	$Cube_2$
101	5 4 7 6 1 0 3 2	4组2元交换＋ 2组4元交换＋ 1组8元交换	$Cube_0$＋$Cube_2$
110	6 7 4 5 2 3 0 1	4组2元交换＋ 1组8元交换	$Cube_1$＋$Cube_2$
111	7 6 5 4 3 2 1 0	1组8元交换	$Cube_0$＋$Cube_1$＋$Cube_2$



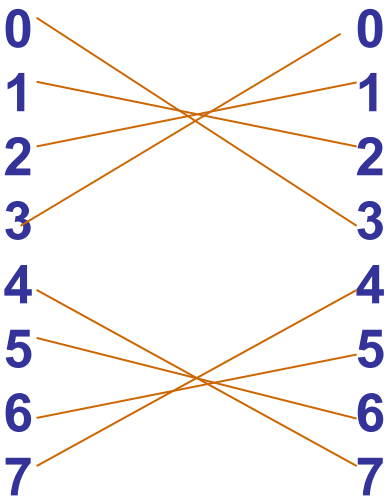
F=(000)



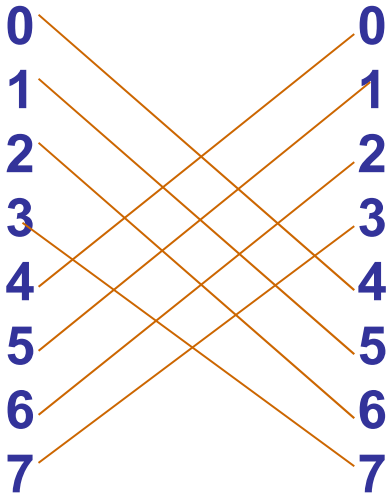
F=(001)



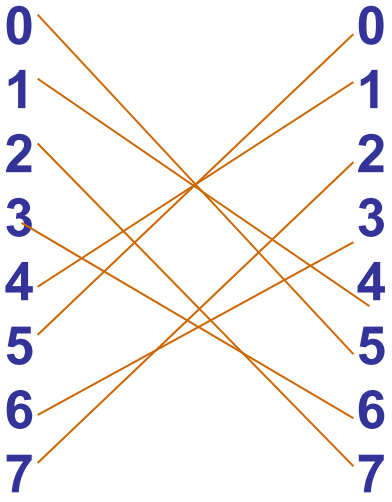
F=(010)



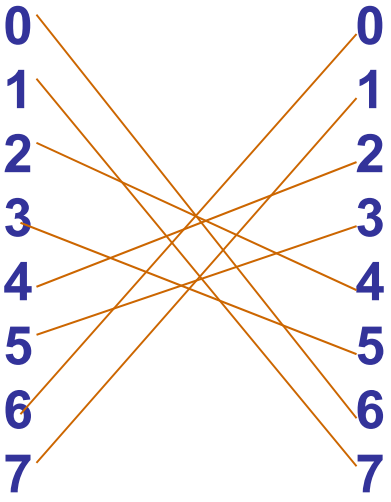
F=(011)



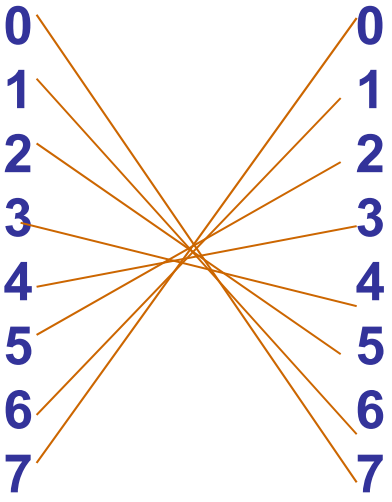
F=(100)



F=(101)



F=(110)



F=(111)

ki 实现Cubei 置换

2. 多级立方体：移数网络

- 当STARAN网络用作移数网络时，采用部分级控制（组控方式）。

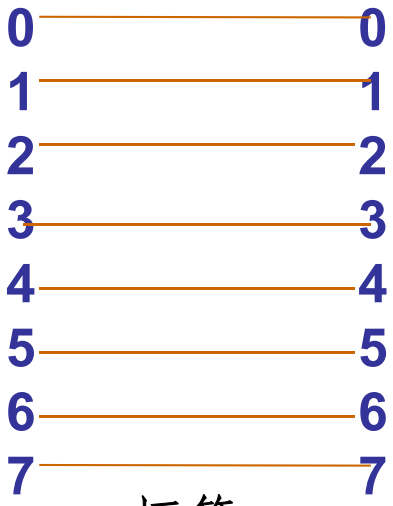
组控方式是指：将第i级的N/2个2×2的2功能开关分成i+1组，每组用一个位信号控制该组开关的状态。

对于级数是$n=\log_2 N$的N×N网络，从第0级至第n-1级各开关级所需的位信号数分别为1,2,...,n个，
因此，共需n(n+1)/2个位信号组成二进制控制向量F。

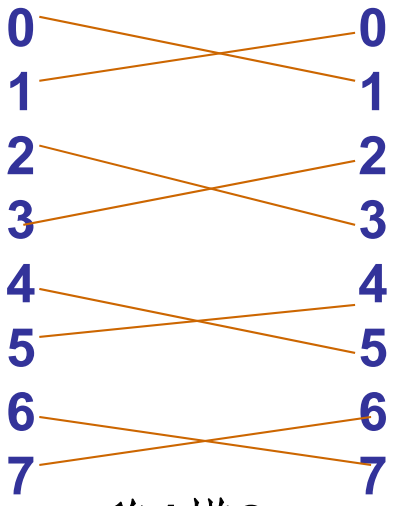
2. 多级立方体：移数网络

- 控制信号的分组和控制结果。

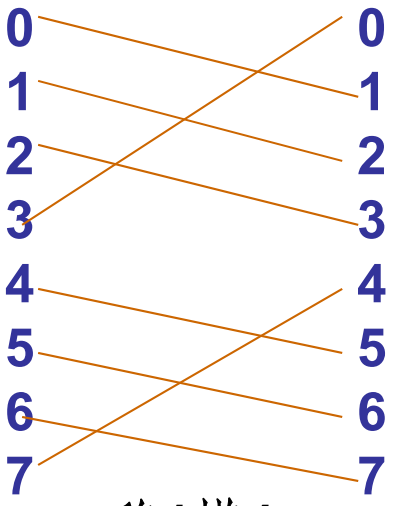
部分级控制信号						连接的输出端号序列（入端号序列：01234567）	所实现的移数功能
第0级	第1级		第2级				
A B C D	E G	F H	I	J	K L		
1	1	0	1	0	0	1 2 3 4 5 6 7 0	移1 mod 8
0	1	1	1	1	0	2 3 4 5 6 7 0 1	移2 mod 8
0	0	0	1	1	1	4 5 6 7 0 1 2 3	移4 mod 8
1	1	0	0	0	0	1 2 3 0 5 6 7 4	移1 mod 4
0	1	1	0	0	0	2 3 0 1 6 7 4 5	移2 mod 4
1	0	0	0	0	0	1 0 3 2 5 4 7 6	移1 mod 2
0	0	0	0	0	0	0 1 2 3 4 5 6 7	不移 全等



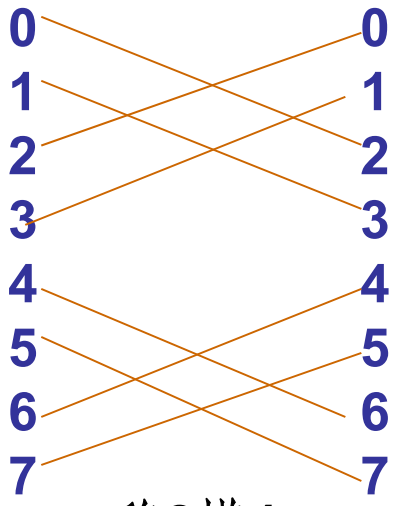
恒等



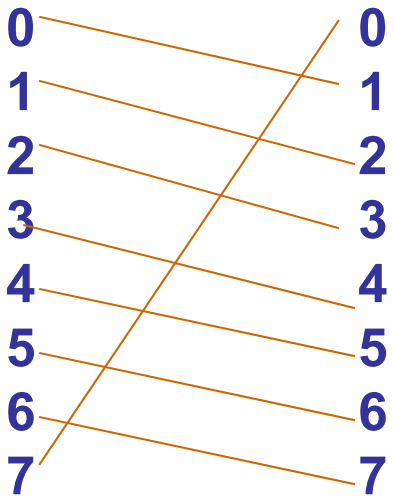
移1模2



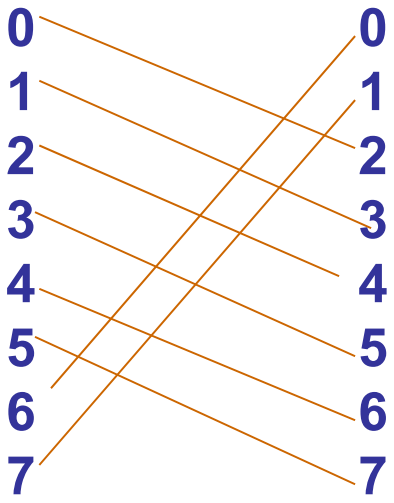
移1模4



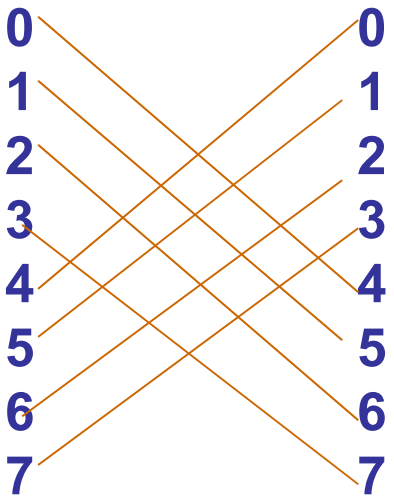
移2模4



移1模8



移2模8



移4模8

题目： 编号分别为0,1,2,...,F的16个处理器之间要求按下列配对通信：(B,1), (8,2), (7,D), (6,C), (E,4), (A,0), (9,3), (5,F)。试选择互连网络类型、控制方式，并画出该互连网络的拓扑结构和各级交换开关状态图。

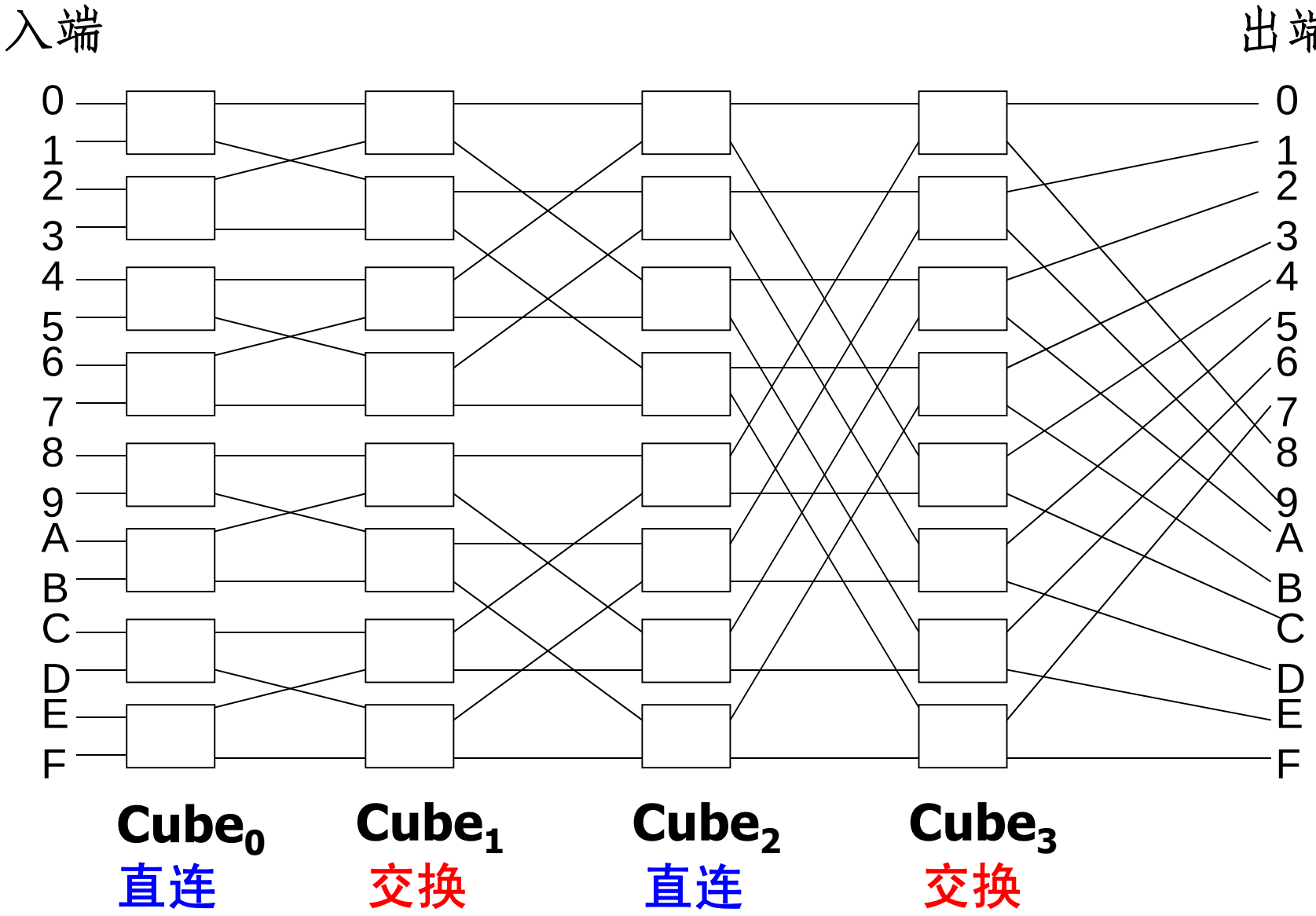
[分析]： 要求配对通讯的处理器号用二进制表示如下：

(B, 1)是(1011, 0001)
(8, 2)是(1000, 0010)
(7, D)是(0111, 1101)
(6, C)是(0110, 1100)
(E, 4)是(1110, 0100)
(A, 0)是(1010, 0000)
(9, 3)是(1001, 0011)
(5, F)是(0101, 1111)

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ \oplus\ 0\ 0\ 0\ 1 \\ \hline 1\ 0\ 1\ 0 \\ k_3\ k_2\ k_1\ k_0 \end{array}$$

用二进制表示互连函数就是：

$Cube(P_3P_2P_1P_0)= \overline{P}_3P_2\overline{P}_1P_0$


入端
出端
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
Cube0
Cube1
Cube2
Cube3
直连
交换
直连
交换

题目：并行处理机有16个处理器，要实现相当于先4组4元交换，然后是两组8元交换，再次是一组16元交换的交换函数功能，请写出此时各处理器之间所实现之互连函数的一般式；画出相应多级网络拓扑结构图，标出各级交换开关的状态。

[分析]：输入端号为：

|0 1 2 3|4 5 6 7|8 9 A B|C D E F|

经4组4元交换后为：

|3 2 1 0|7 6 5 4|B A 9 8|F E D C|

分成2组后为：

|3 2 1 0 7 6 5 4|B A 9 8 F E D C|

然后经2组8元交换后为：

|4 5 6 7 0 1 2 3|C D E F 8 9 A B|

再经1组16元变换后为：

|B A 9 8 F E D C 3 2 1 0 7 6 5 4|

用二进制表示互连函数就是：

$Cube(P_3P_2P_1P_0)= \overline{P}_3P_2\overline{P}_1\overline{P}_0$

$$\begin{array}{r} 0\ 0\ 0\ 1 \\ \oplus\ 1\ 0\ 1\ 0 \\ \hline 1\ 0\ 1\ 1 \\ k_3\ k_2\ k_1\ k_0 \end{array}$$

最后，可得出配对互连的是：

(0,B), (1,A), (2,9), (3,8), (4,F), (5,E), (6,D), (7,C)

3. 多级混洗交换网络(Ω网)

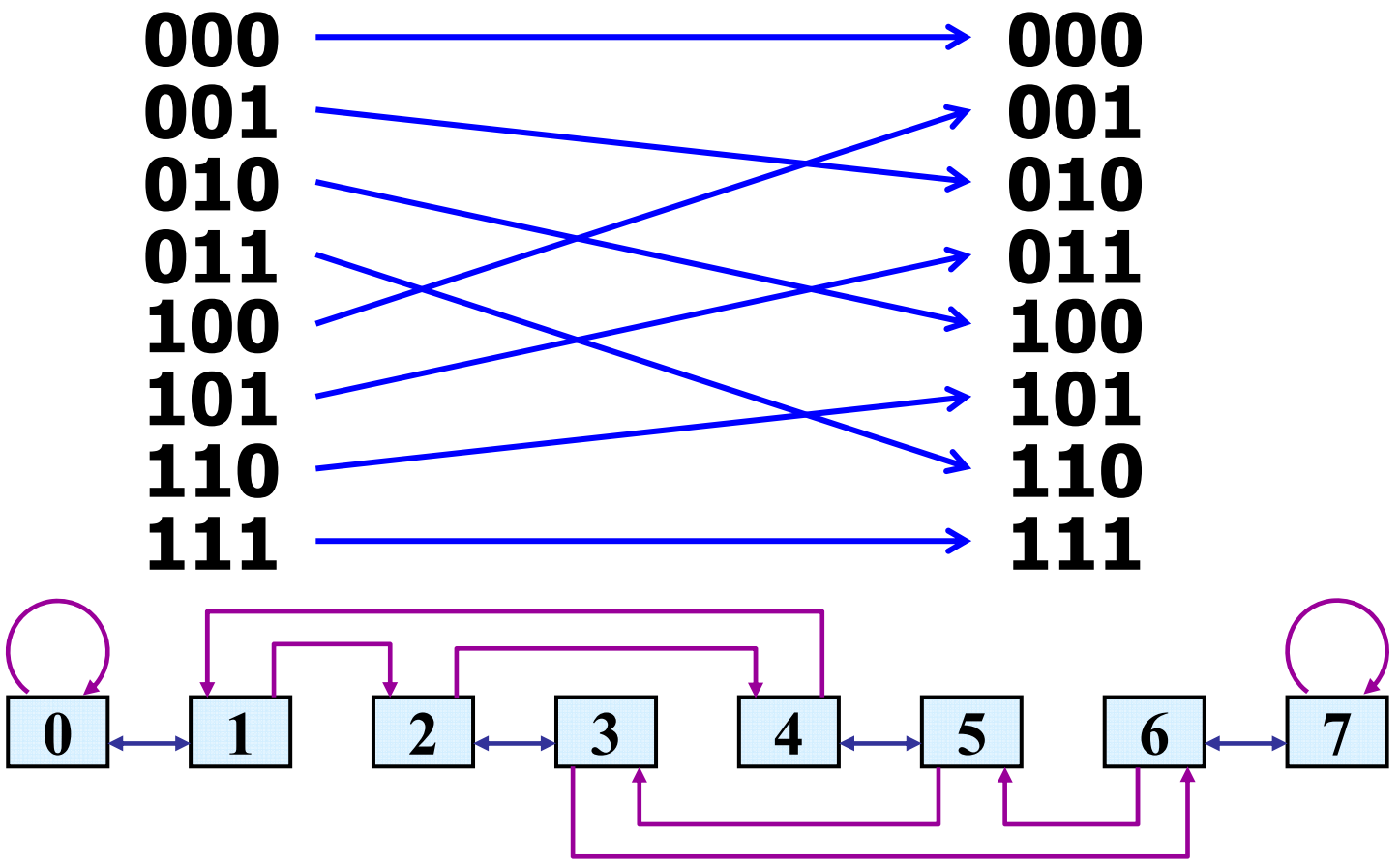
- 采用全混洗函数和交换函数，又称为Omega网络。
- 特点：
  - N个输入的Omega网络有$\log_2 N$级，每级有N/2个2×2的四功能交换开关
  - 每级的拓扑结构相同
  - 采用单元控制
  - 能够实现任意一个输入端到任意一个输出端的连接，但不能同时实现多个输入端到多个输出端的连接。
  - 能够实现从任意一个输入端到所有输出端的广播。

3. 多级混洗交换网络(Ω网)

- 画法：
  - 1. 计算级数 $n=\log_2 N$
  - 2. 每级都是N个端的全混洗（均匀混洗）连接，之后加1个四功能交换开关
  - 3. 每级有N/2个 2×2 的四功能交换开关，每级的编号从输入到输出依次定为 n-1, n-2, … i, … 1, 0
  - 4. 将各级交换开关同一编号的各端用线连接起来

混洗交换单级互连网络

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



5. 多级混洗交换网络(Ω网)

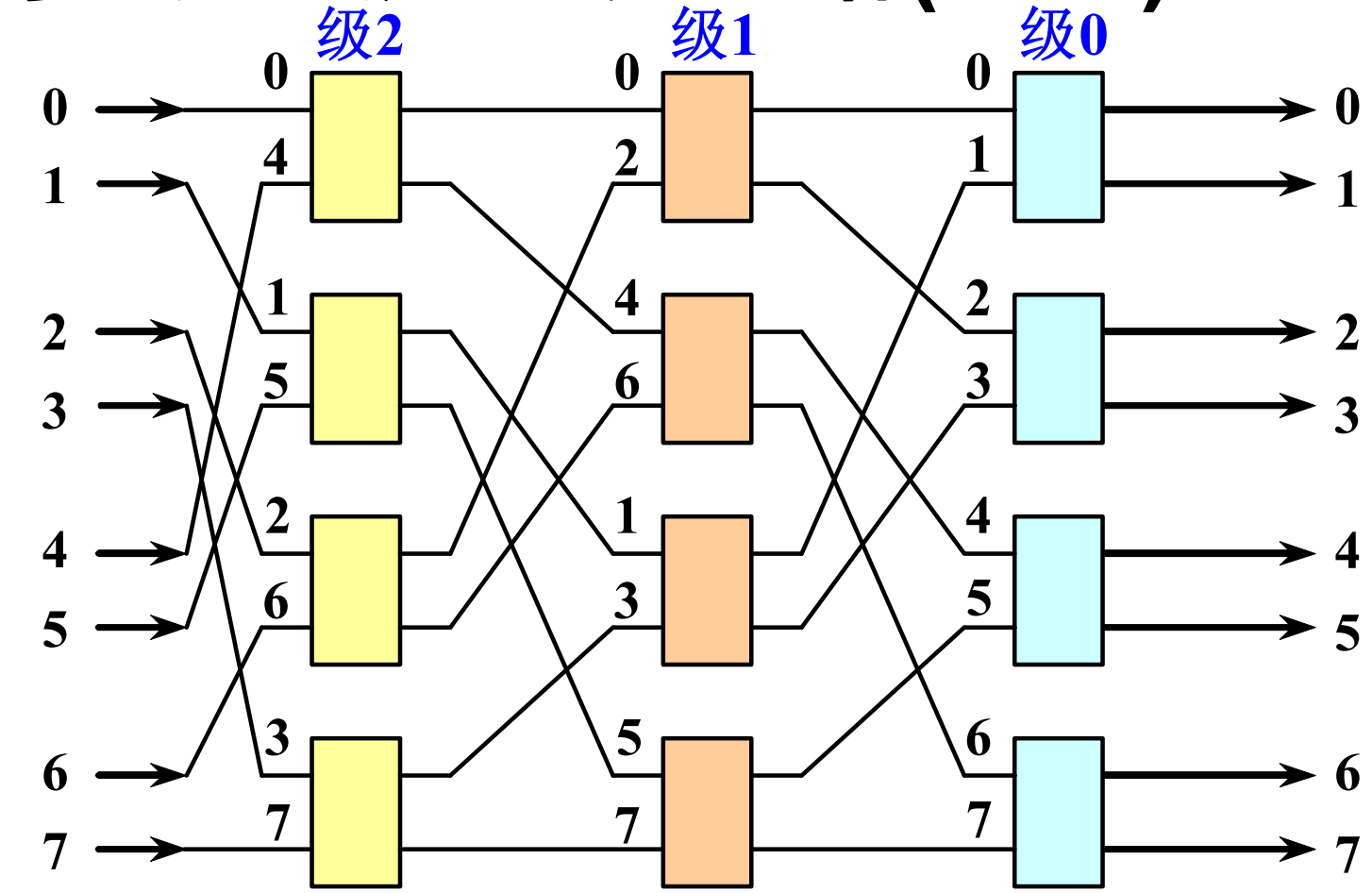


图 N=8多级混洗交换网络

5. 多级混洗交换网络(Ω网)

另一种画法

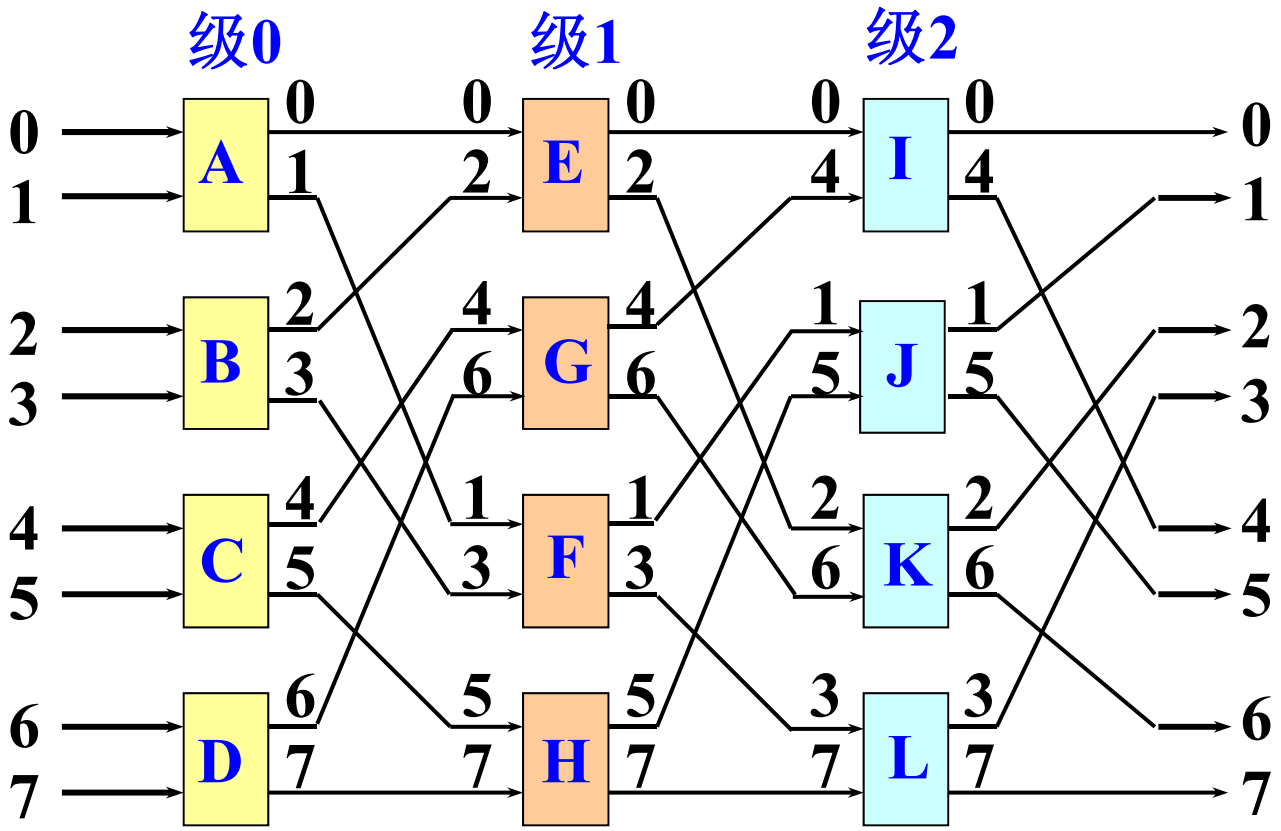


图 N=8多级混洗交换网络

5. 多级混洗交换网络(Ω网)

级控制方式：

- 级控制信号 k_i 是参与逻辑运算的一个变量，其逻辑关系可以表示为：

$$E(x_i) = x_i \oplus k_i$$

k_i =0时，表示直送； k_i =1时，表示交叉。

多级混洗—交换网络寻径算法（路由算法）

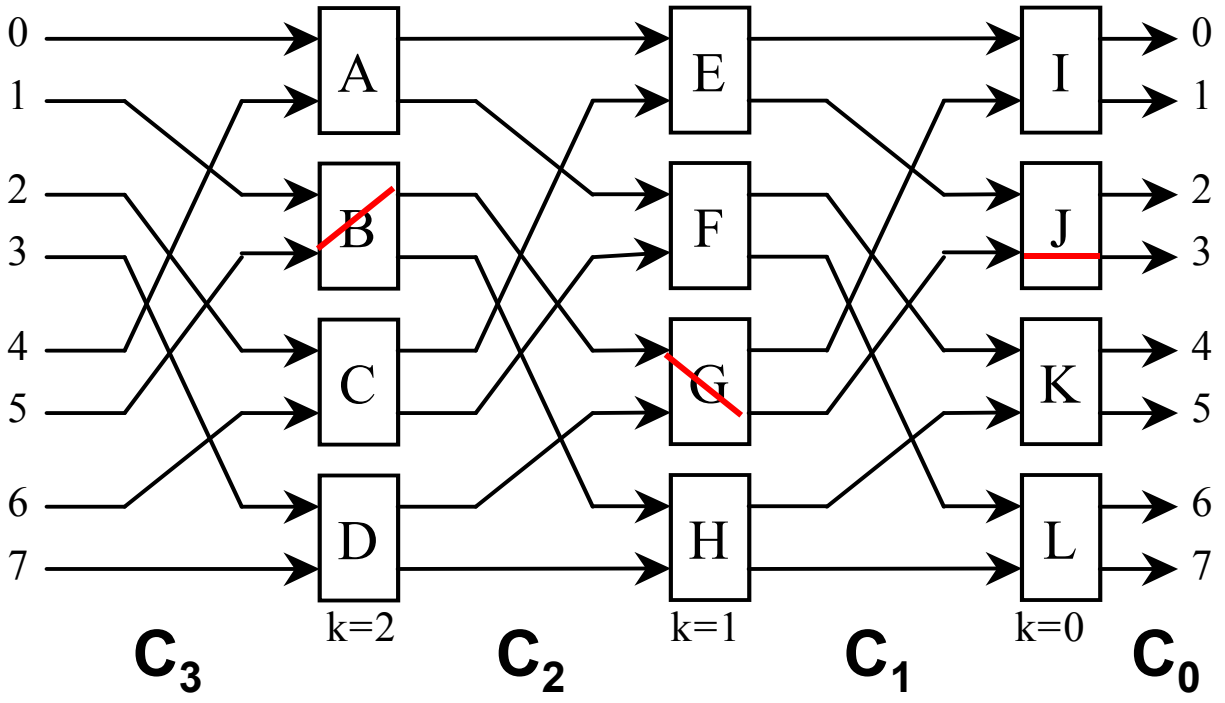
目的： 根据给定的输入/输出对应关系，确定各开关的状态。

名称： 源-目的地址异或法

操作： 将任一个输入地址与它要到达的输出地址作异或运算，其结果的第i位控制数据到达的第i级开关，**0** 表示“直连”，**1** 表示“交换”。（例如给定传输：**101B→011B**）

级控制：

根据二者异或结果为**110B**，于是从**101B**（**5**）号输入端开始，把它遇到的第**2**级开关置为“交换”，第**1**级开关置为“交换”，第**0**级开关置为“直连”。

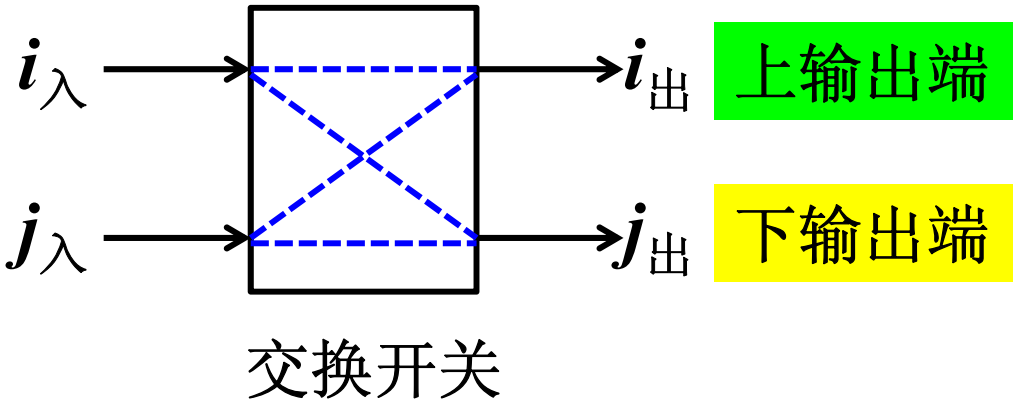


根据目的地址 011？

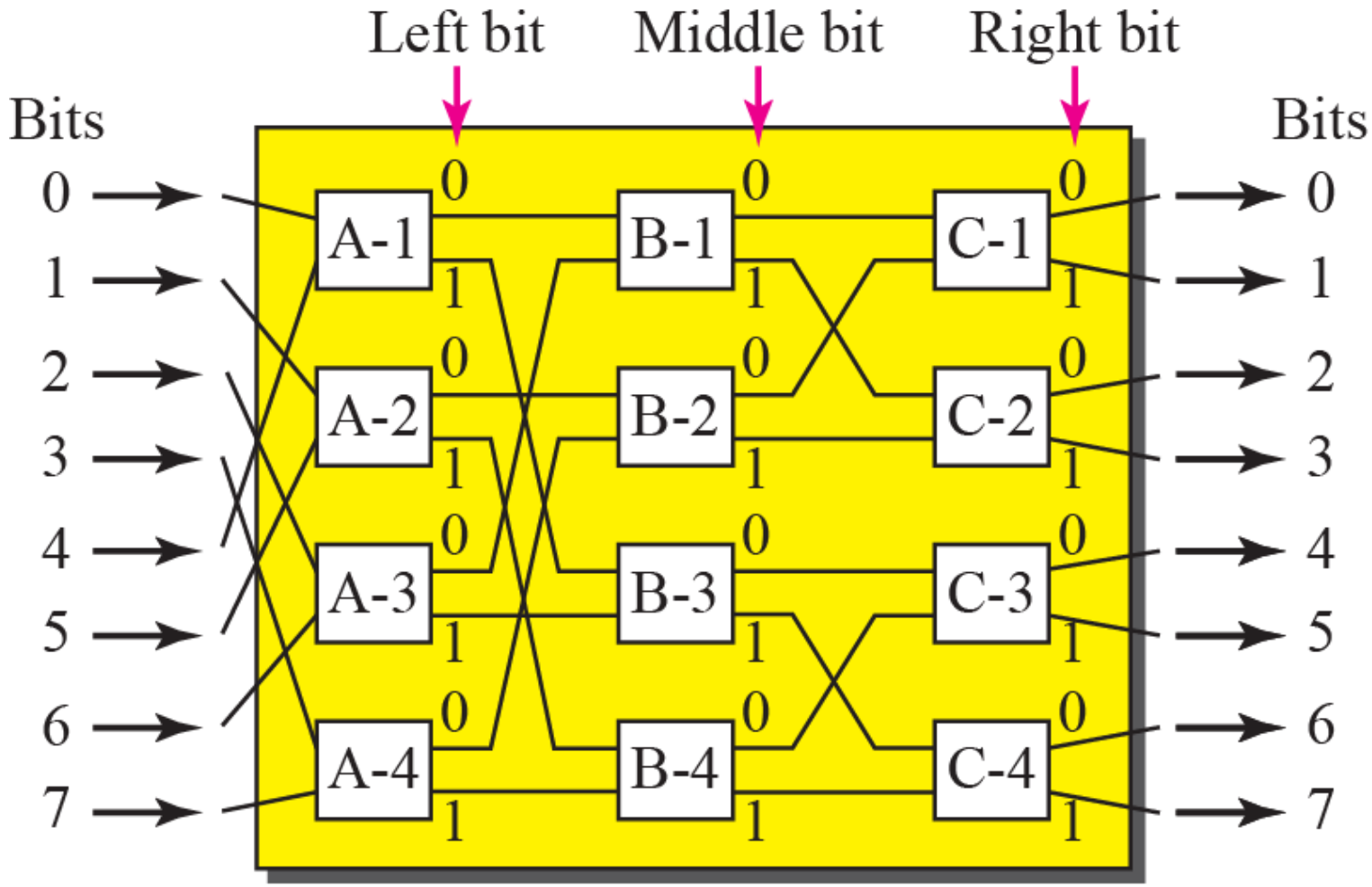
5. 多级混洗交换网络(Ω网)

■ 单元控制方式：

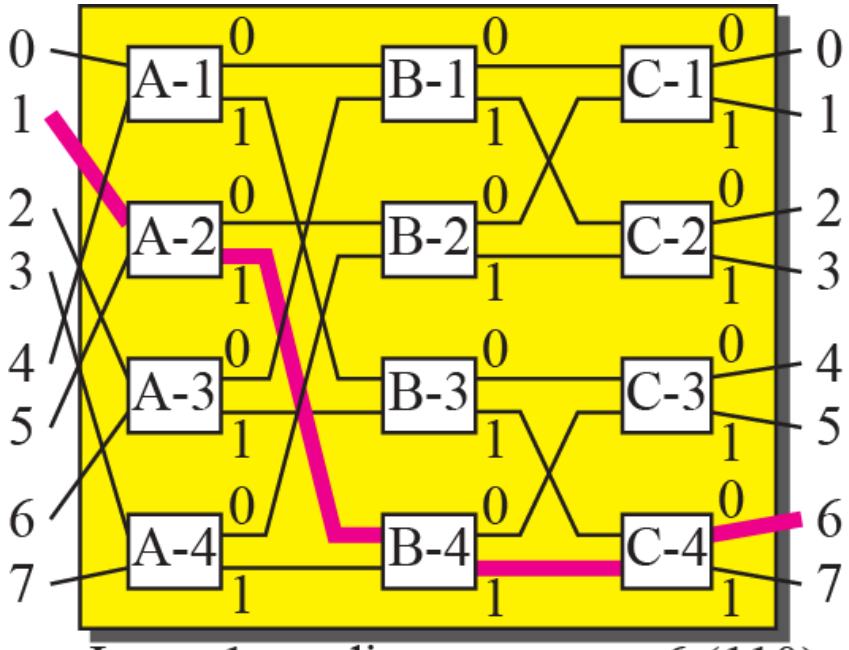
- 采用单元控制方式时，当目的地址编码从高位开始的第i位为0时，第i级的2×2开关的输入端与上输出端连接（$i_{入}$和$j_{入}$ → $i_{出}$），否则输入端与下输出端连接（$i_{入}$和$j_{入}$ → $j_{出}$）。



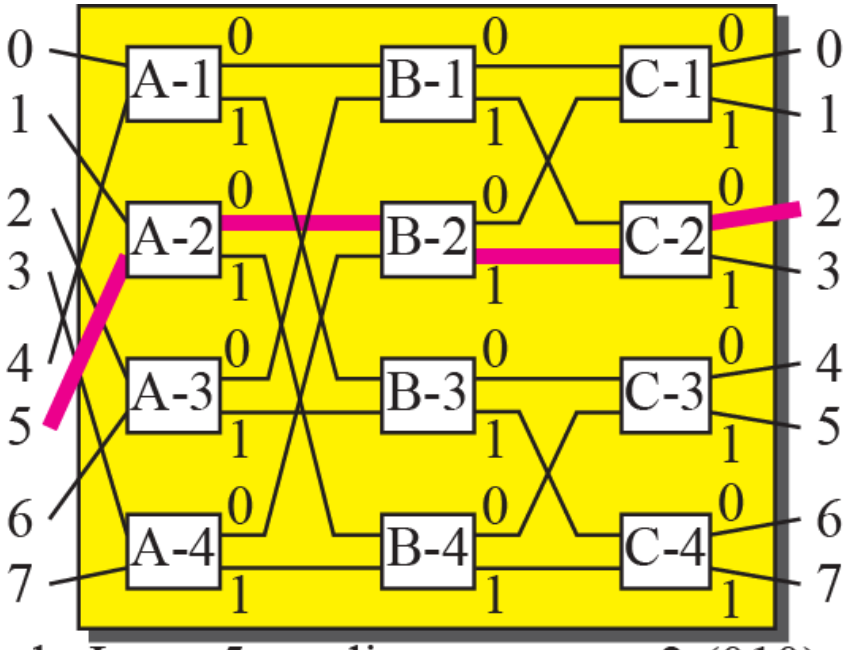
例：某Ω网


Left bit
Middle bit
Right bit
Bits
Bits
0
1
2
3
4
5
6
7
A-1
A-2
A-3
A-4
B-1
B-2
B-3
B-4
C-1
C-2
C-3
C-4

例：



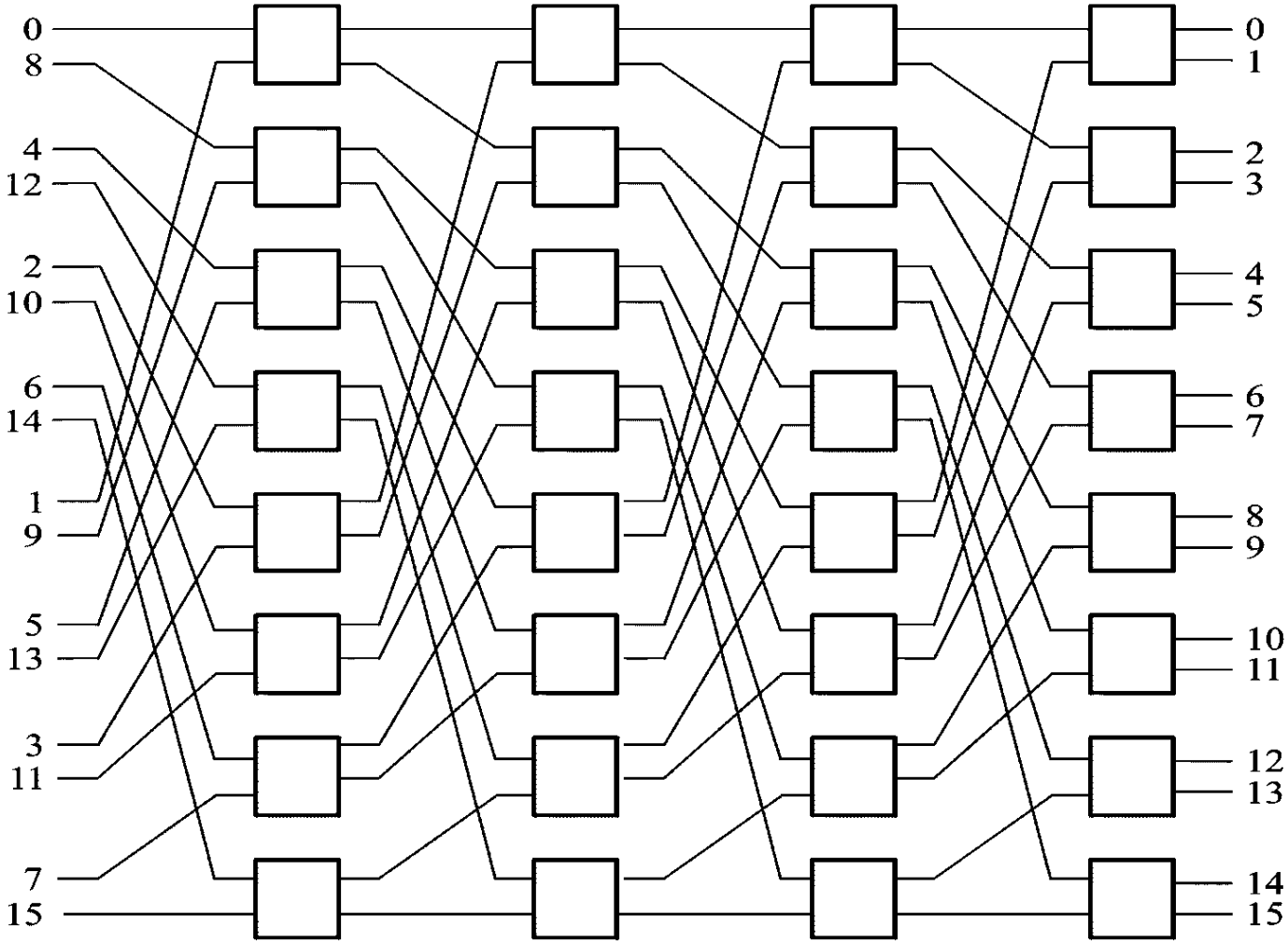
a. Input 1 sending to output 6 (110)



b. Input 5 sending to output 2 (010)

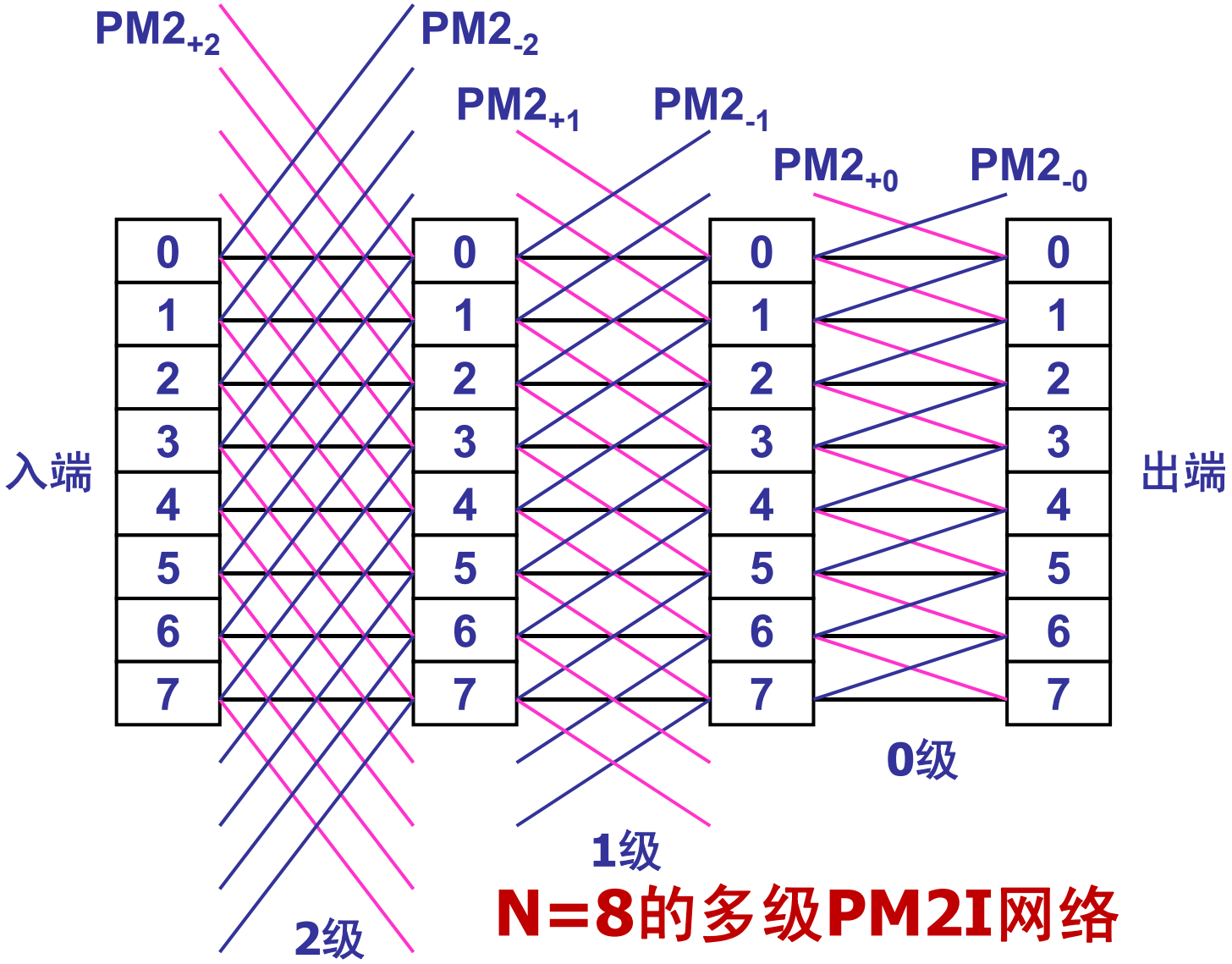
根据目的地址控制开关

一个16×16 Omega网络



(e) 16×16O mega 网络

6. 多级PM2I交换网络

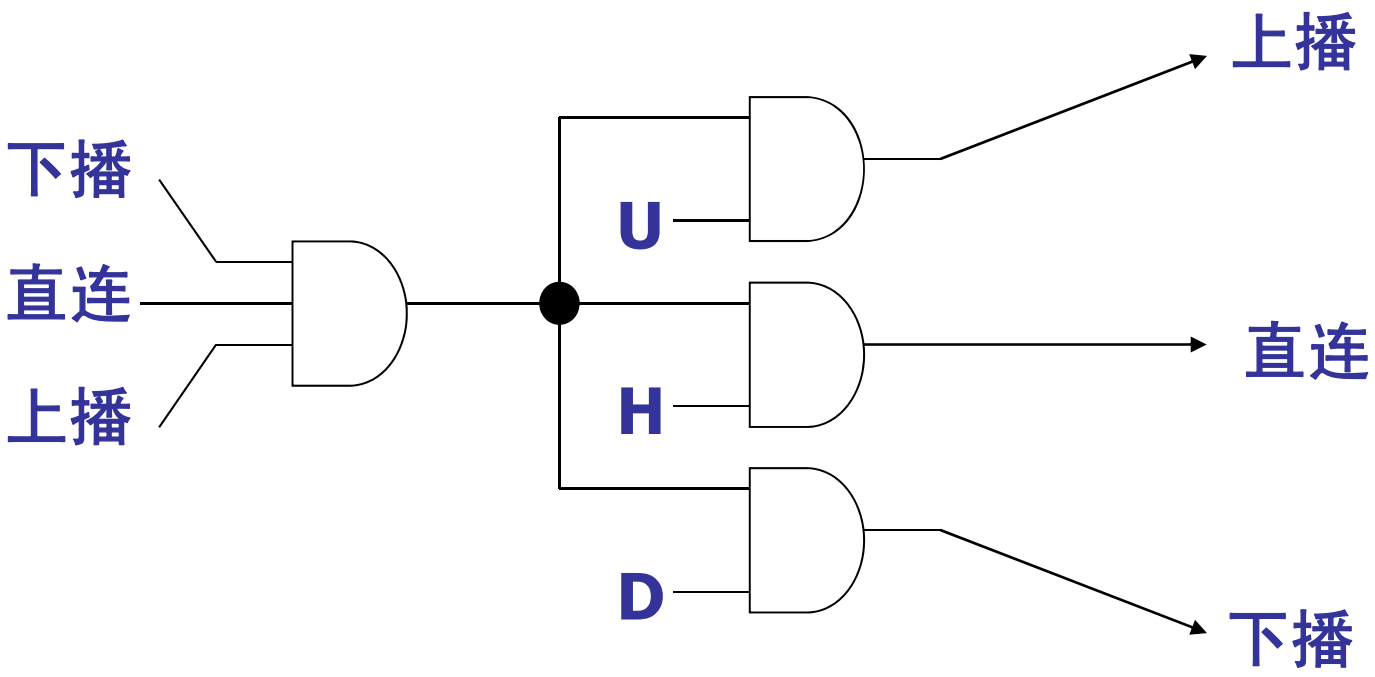


N=8的多级PM2I网络

6. 多级PM2I交换网络

1) 就i级而言（0<=i<=n-1,0<=i<=2），每个输入端j（0<=j<=2）都是三根线分别输出端（$j-2^i$）mod N，j和（$j+2^i$）mod N，图中分别用蓝线、黑线和红线表示。
2) 第0级完成的是$PM2_{\pm 0}$，第1级完成的是$PM2_{\pm 1}$，第2级完成的是$PM2_{\pm 2}$。
3) 从一个单元到另一个单元的路由有多条路径可供选择，如从6号处理单元到4号处理单元，可以经过6-6-4-4，也可6-2-4-4，显然，在多级网络中提供了冗余通路，有利于提高系统的可靠性。
4) 当各级处理单元均采用单元控制时，这种多级PM+I网络成为了强化数据交换网络(augmented data manipulator，简称ADM网络)

6. 多级PM2I交换网络



多级PM2I网络中各级各单元的单元控制原理图

五种多级互连网络比较

		交换开关	控制方式	拓扑结构
二进制立方体网络	STARAN交换网络	二功能交换单元	级控制	单级立方体网络
	STARAN移数网络	二功能交换单元	部分级控制	单级立方体网络
	间接二进制n方体网络	二功能交换单元	单元控制	单级立方体网络
Omega网络		四功能交换单元	单元控制	单级混洗交换网络
强化数据变换网络（ADM网络）		多功能交换单元	单元控制	单级PM2I网络

题：对于采用级控制的三级立方体网络，当第i级(0<=i<=2)为直连状态时，不能实现哪些结点之间的通信？为什么？反之，当第i级为交换状态呢？

[分析]：三级立方体网络的入出结点的二进制编号为$P_2P_1P_0$。

当第i级(0<=i<=2)为直连状态时，反映出能够实现互连的入、出端号的二进制中的Pi位不能变反，其他的Pj(Ii>j)可以不变，也可以变反。

当第i级(0<=i<=2)为交换状态时，反映出能够实现互连的入、出端号的二进制中的Pi位必须变反，其他的Pj(Ii>j)可以变反，也可以不变。

[解]：

当第i级(0<=i<=2)为直连状态时，不能实现入、出端号处理器的二进制码的编号中第Pi位取反的处理器之间的连接，因为Pi位控制第i级开关只能直连。反之，当第i级(0<=i<=2)为交换状态时，不能实现入、出端处理器号的二进制中的Pi位相同的处理器之间连接。

例如，当第0级为直连时，只能实现000、010、100、110号处理器之间进行数据传送，不能与001、011、101、111号处理器中任一处理器之间数据传送。

6.4.5 全排列网络

- 如果互连网络是从 *N* 个入端到 *N* 个出端的一到一的映射，就可以把它看成是对此 *N* 个端的重新排列。因此，互连网络的功能实际上就是用新排列来置换 *N* 个入端原有的排列。
- 循环互连网络和多级互连网络不能实现同时多对入端与出端之间的互连。因为会发生数据通路冲突。
  - 多级立方体：0→5 与 1→7 在开关A发生冲突
  - N=8的多级混洗交换网络：0→6和4→7，3→0和5→1，3→0和7→3，5→0和7→1等有冲突

6.4.5 全排列网络

- 要同时实现两对或多对入端与出端之间的连接时，有可能因争用数据传送路径而发生冲突。我们称具有这类性质的互连网络为阻塞式网络(Blocking Network)。
- 反之，不具有这类性质的互连网络为非阻塞式网络，或称为全排列网络。非阻塞式网络连接的灵活性好，但连线多，控制复杂， 成本高。

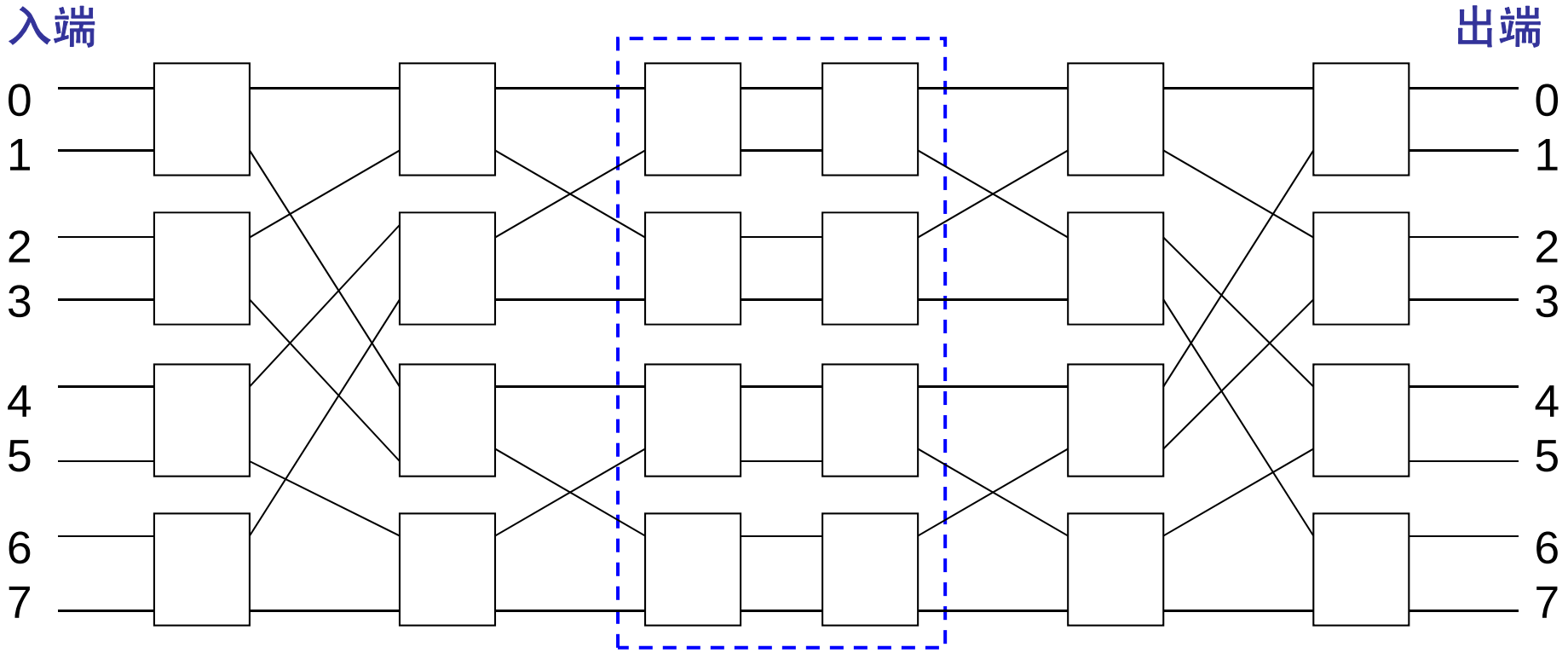
6.4.5 全排列网络

- 解决方法：采用多个多级互连网络连接
- 原理：N个结点的全排列为N！。
- 只要对这个多级互连网络通行两次，每次通行时，让各开关处于不同状态，就可以满足对N个端子的全部N!种排列。
- 因为此时，全部开关的总状态数可有$N^{N/2}\cdot N^{N/2}=N^N$种，足以满足N!种不同排列的开关状态要求。

6.4.5 全排列网络

- 这种只要经过重新排列已有入出端对的连接，就可以完成所有可能的入出端间的连接而不发生冲突的互连网络，称为可重排列网络(Rearrangeable Network)。
- 实现时，可以在上述任何一种基本多级互连网络的出端设置锁存器，使数据在时间上顺序通过两次，这实际上就是循环互连网络的实现思路。

多级BENES可重排网络（非阻塞网络）



N=8的BENES多级可重排网络

特点：

1) BENES网络=基准网络 + 逆基准网络
2) 开关级数$2\log_2 N-1$，每一级有N/2个2×2的功能开关。
3) BENES网络置换表达式：$E\sigma^{-1}_{(n-1)}E\sigma^{-1}_{(n-2)}\cdots E\sigma^{-1}_{(-1)}E\sigma_{(1)}\cdots E\sigma_{(n)}E$
4) 当N=8时，有$2\log_2 N-1=5$级，每级8/2=4个开关，单元控制，可实现置换数$2^{5\times4}=2^{20}=1M$，开关组合8!=40320，所以多级BENES网络无阻塞网络

动态互连网络的比较

网络特性	总线系统	多级网络	交叉开关
单位数据传送的最小时延	恒定	$O(\log_k n)$	恒定
每台处理机的带宽	O(w/n)至O(w)	O(w)至O(nw)	O(w)至O(nw)
连线复杂性	O(w)	$O(nw\log_k n)$	$O(n^2w)$
开关复杂性	O(n)	$O(n\log_k n)$	$O(n^2)$
连接特性和寻径性能	一次只能一对一	只要网络不阻塞，可实现某些置换和广播	全置换，一次一个
典型计算机	Symmetry S1，Encore Multimax	BBNTC-2000 IBM RP3	Cray Y-MP/816 Fujitsu VPP 500
说明	总线上假定有n台处理机；总线宽度为w位	n×n MIN采用k×k开关，其线宽为w位	假定n×n交叉开关的线宽为w位

6.5 并行存储器的无冲突访问

1、访问需求

并行存取分量，不同分量步长不一致(如按行/列/对角线)

2、存在问题

存储器宽度＜向量长度，易产生访存冲突(分量步长不同)

3、解决方法

⑴采用多体交叉存储器：存储体数＞PE数，使PE可并行访问

⑵对向量进行分组操作：使MEM宽度＞向量长度

⑶选择适当存储体数(m)：使访问无冲突

对一维向量：顺序存放，访问步长（跑步模式）与 m 互质

m＝质数，便于与PE数(常为偶数)互质

一维数组

- 对地址连续的元素访问，低位交叉存放这些元素在各个体中；
- 若以间距=2的跑步模式访问，会因访存冲突，降低存储器的带宽；
- 解决方法：将存储体分体数取成质数，且与跑步距离互质，可实现无冲突访问；

一维数组

存储体号

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}

存储体分体数=4

以间距=2的跑步模式访问，元素0与2不冲突，但与4、8冲突。

一维数组

存储体号

0	1	2	3	4
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}	a_{13}	a_{14}

存储体分体数=5

∵存储体分体数=5

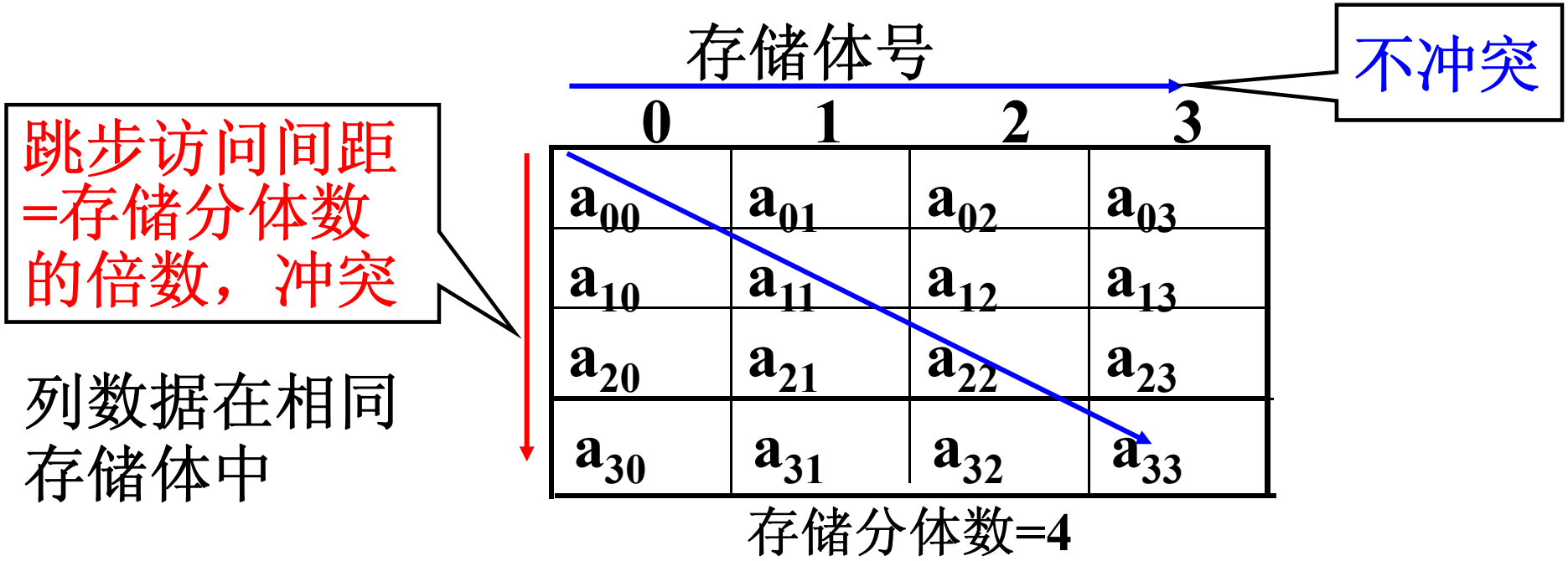
∴只要跳步访问间距 ≠ 5 的倍数，则不会发生访存冲突

多维数组

例如：

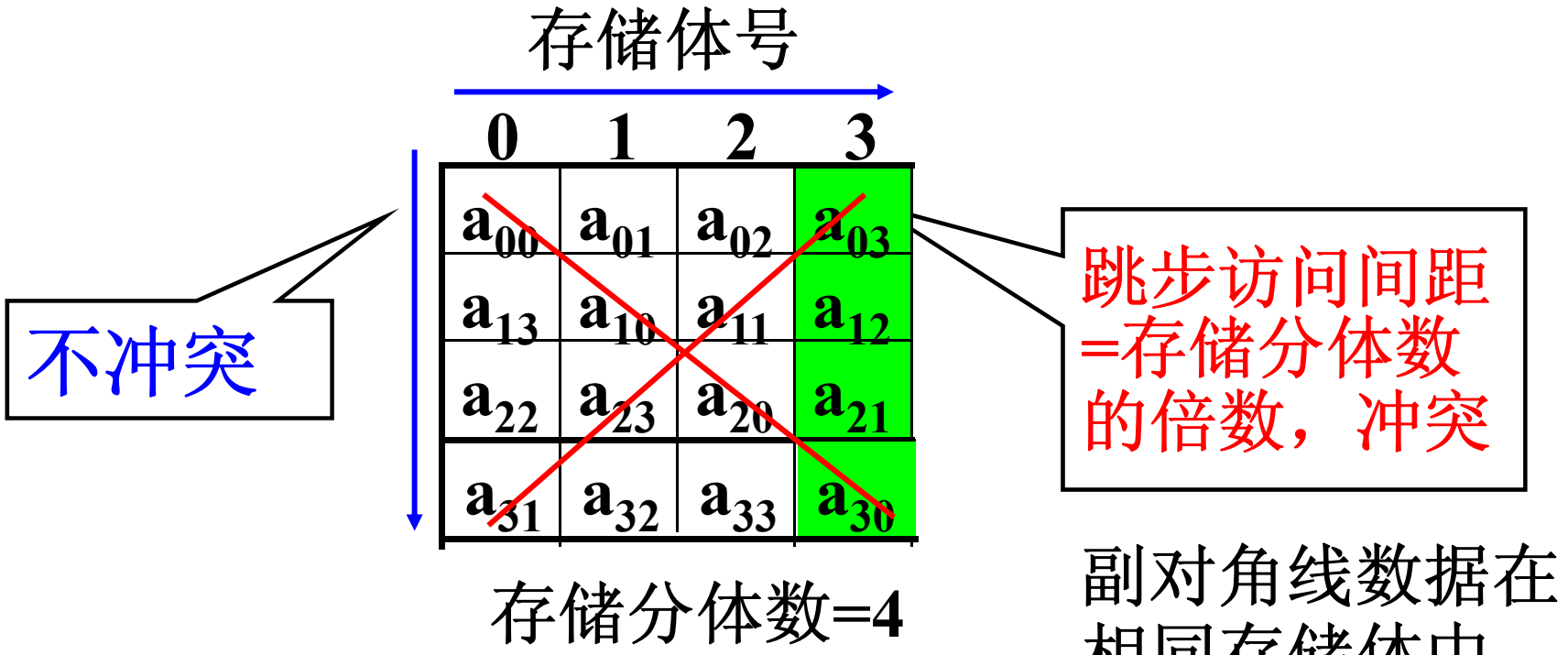
$$4\text{x}4\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

（1）直接按行存放



多维数组

（2）错位存放　行/列采用不同的错开距离



无冲突方法

错位存放，行/列/对角线数据在不同体中

- 存储体数 M > 每次访问的向量元素N，且为质数，行、列等方向上采用不相同的错开距离(d_1、d_2)。当模块数(存储分体数)=m，当$m=2^{2P}+1$（P>0），实现的充分条件：

$$d_1=2^{2P}，d_2=1。$$

例：BSP中，存储模块数m=17，运算单元 n=16（m>n），P=2（ $m=2^{2P}+1$ ），d_1= 4（$d_1=2^{2P}$），d_2=1，可实现各种模式的无冲突访问。

例：

4*5

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

错位存放：

存储模块号j=a mod m

一维线性地址空间

块内地址 i=⌊a/n⌋

	0	1	2	3	4	5	6
0	a_{00}	a_{10}	a_{20}	a_{30}	a_{01}	a_{11}	*
1	a_{31}	a_{02}	a_{12}	a_{22}	a_{32}	*	a_{21}
2	a_{23}	a_{33}	a_{04}	a_{14}	*	a_{03}	a_{13}
3				* 未用	a_{24}	a_{34}	
			*				

存储模块损失

步长≠7的倍数，不发生冲突

m（存储体模块数）=7， n（运算单元数）=6

本章重点

- 并行处理机的基本构形
- 并行处理机的特点
- **ILLIAC Ⅳ处理单元的互连**
- 阵列处理机的算法
- **基本的单级互连网络（立方体、 PM2I、混洗交换）**
- **多级互连网络（多级立方体、多级混洗交换、多级PM2I）**
- 并行存贮器的无冲突访问

第6章 作业6