第7章 互连网络

肖梦白
xiaomb@sdu.edu.cn
https://xiaomengbai.github.io

7.1 互连网络的基本概念

7.2 互连网络的种类

7.3 静态互连网络

7.4 动态互连网络

7.5 阵列处理机

7.6 脉动处理机

7.7 消息传递机制

7.1 互连网络的基本概念

7.1.1 互连网络的作用

7.1.2 互连函数

7.1.3 互连网络的特性

7.1.4 互连网络的传输性能参数

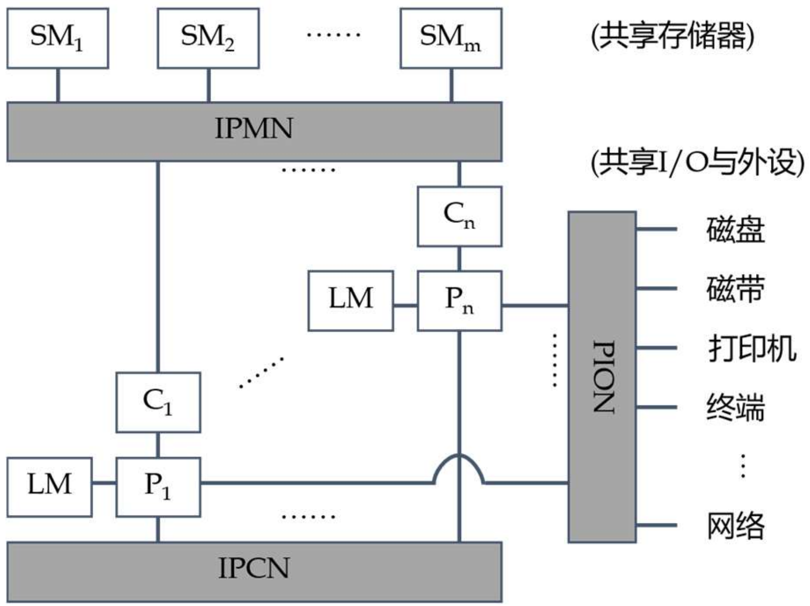
7.1.5 互连网络的种类

7.1.1 互连网络的作用

- 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中节点之间的相互连接。
- 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

7.1.1 互连网络的作用

- 每台处理机P_i与自己的本地存储器（LM）和私有高速缓存（C_i）相连
- 多处理机-存储器互联网络IPMN与共享存储器模块（SM）相连
- 处理机通过处理机-I/O网络IPON访问共享的I/O与外围设备
- 处理机之间通过处理机间通信网络IPCN进行通信



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一般处理机系统的互连结构

- 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）

7.1.2 互连函数

- 为反映互连网络的连接特性，每种互连网络可用一组互连函数来描述
- 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，即在互连函数f的作用下，输入端x连接到输出端$f(x)$。

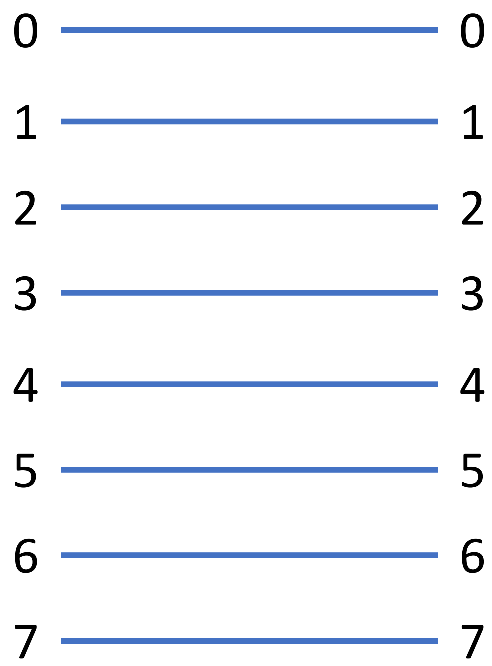
- 当互连网络用来实现处理器与处理器之间的数据变换时，互连函数反映了网络输入数组和输出数组之间对应的置换关系或排列关系，有时也称为置换函数或排列函数
- 设$n=\log_2 N$，则可以用n位二进制来表示N个输入端和输出端的二进制地址，互连函数表示为：

$$f(x_{n-1}x_{n-2}\cdots x_1x_0)$$

7.1.2 互连函数

➢ <u>恒等函数</u>：相同编号的输入端与输出端一一对应互连所实现的置换

$$I(x_{n-1}x_{n-2}\cdots x_0)=x_{n-1}x_{n-2}\cdots x_0$$



$N=8$的恒等置换

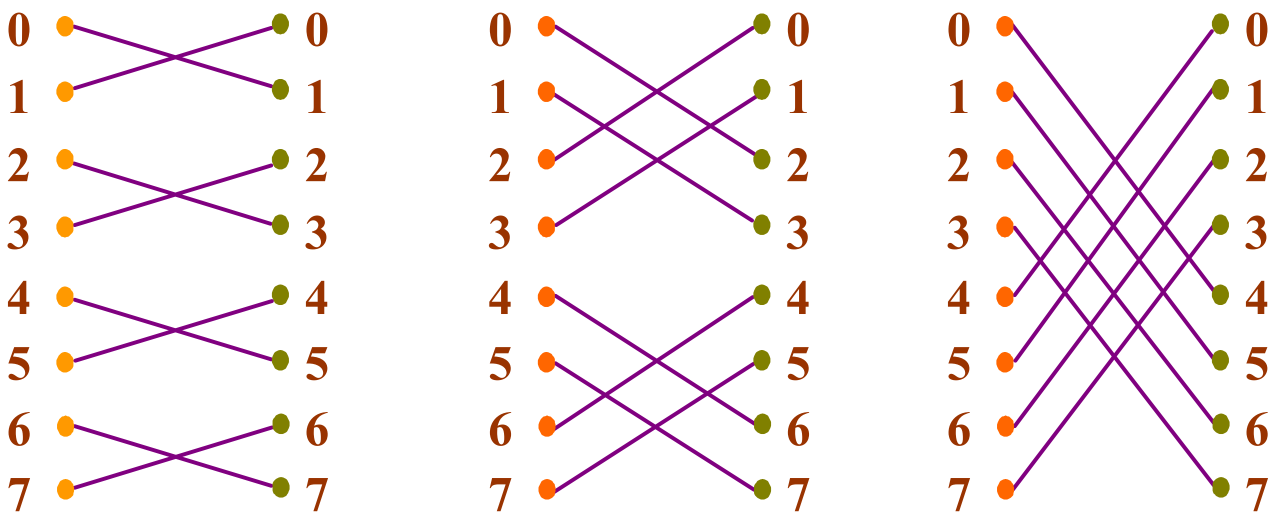
7.1.2 互连函数

➢ <u>交换函数</u>：实现二进制地址编码中第k位互反的输入端与输出端之间的连接

$$C_k(x_{n-1}x_{n-2}\cdots x_1x_0) = x_{n-1}x_{n-2}\cdots\overline{x_k}\cdots x_0$$

- 它共有$n = \log_2 N$种互连函数（N为节点个数）

010 (2) → 011 (3)
011 (3) → 010 (2)



010 (2) → 110 (6)
011 (3) → 111 (7)

(a) Cube$_0$ 交换函数　**(b) Cube$_1$ 交换函数**　**(c) Cube$_2$ 交换函数**

010 (2) → 000 (0)
011 (3) → 001 (1)

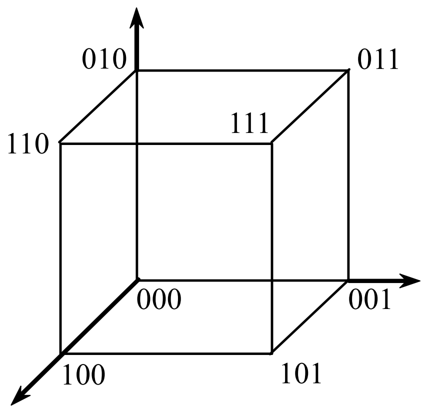
7.1.2 互连函数

- <u>交换函数</u>：实现二进制地址编码中第0位互反的输入端与输出端之间的连接

$$C_k(x_{n-1}x_{n-2}\cdots x_1x_0) = x_{n-1}x_{n-2}\cdots\overline{x_k}\cdots x_0$$

- 它共有$n=\log_2 N$种互连函数（N为节点个数）
- 主要用于构造立方体互连网络和各种超立方体互连网络。
- 当N＝8时，n＝3，可得到常用的立方体互连函数：

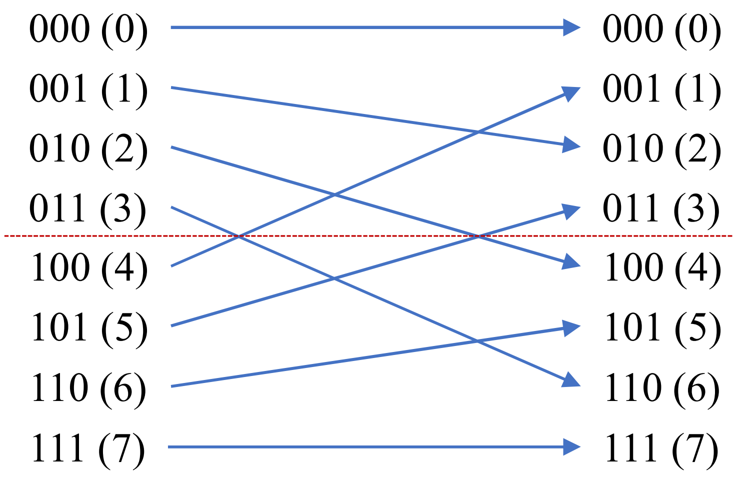
$$\begin{aligned}Cube_0(x_2x_1x_0) &= x_2x_1\bar{x}_0\\ Cube_1(x_2x_1x_0) &= x_2\bar{x}_1x_0\\ Cube_2(x_2x_1x_0) &= \bar{x}_2x_1x_0\end{aligned}$$



7.1.2 互连函数

➢ <u>均匀洗牌函数</u>（shuffle）：将输入端分成数目相等的两半，前一半和后一半按类似均匀混洗扑克牌的方式交叉地连接到输出端（输出端相当于混洗的结果，也称为混洗函数或混洗置换 ）

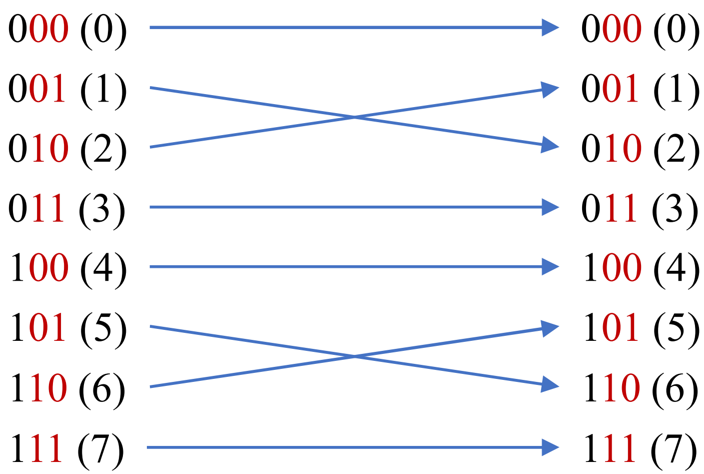
$$\sigma(x_{n-1}x_{n-2}\cdots x_1x_0) = x_{n-2}x_{n-3}\cdots x_1x_0x_{n-1}$$



7.1.2 互连函数

- <u>子洗牌</u>（子混洗，subshuffle）：即把输入端的二进制编号中的低k位循环左移一位

$$\sigma_{(k)}(x_{n-1}x_{n-2}\cdots x_k x_{k-1}x_{k-2}\cdots x_1x_0)$$
$$= x_{n-1}x_{n-2}\cdots x_k x_{k-2}\cdots x_1x_0x_{k-1}$$

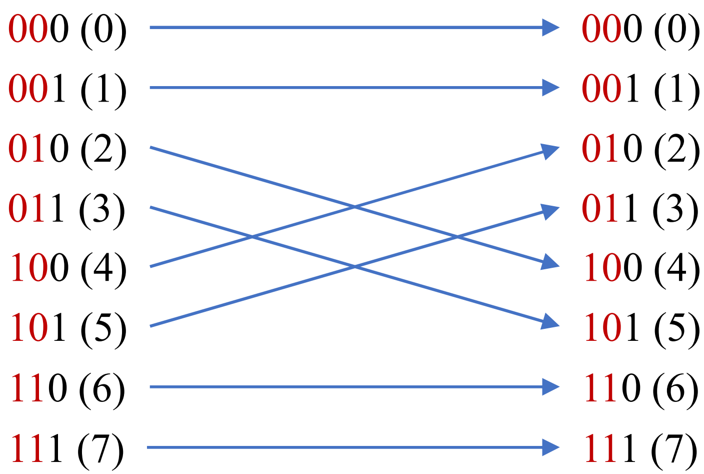


k=2

7.1.2 互连函数

- 超洗牌（超混洗, supershuffle）：即把输入端的二进制编号中的高k位循环左移一位

$$\sigma^{(k)}(x_{n-1}x_{n-2}\cdots x_{n-k}x_{n-k-1}\cdots x_1x_0)$$
$$= x_{n-2}\cdots x_{n-k}x_{n-1}x_{n-k-1}\cdots x_1x_0$$

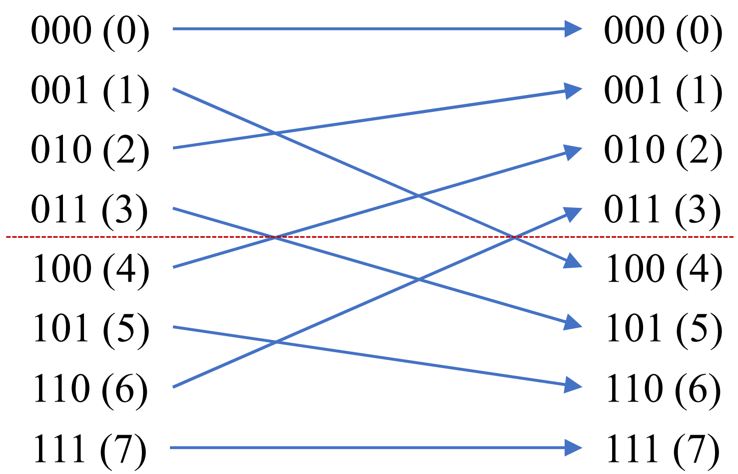


k=2

7.1.2 互连函数

➢ <u>逆混洗</u>：即把输入端的二进制编号循环右移一位

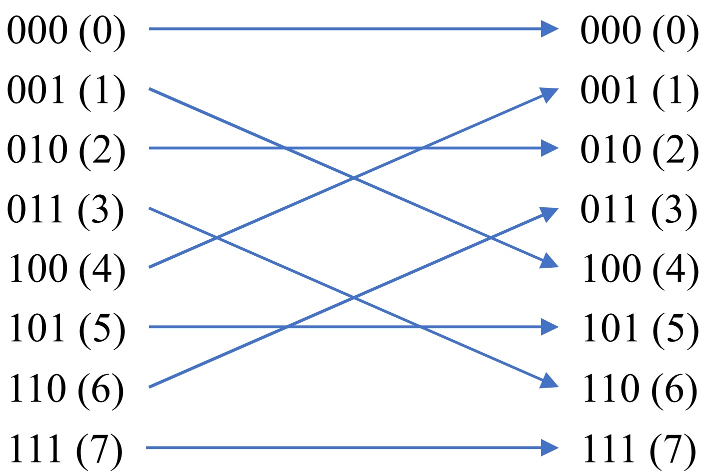
$$\sigma^{(k)}(x_{n-1}x_{n-2}\cdots x_1x_0)$$
$$= x_0x_{n-1}\cdots x_2x_1$$



7.1.2 互连函数

- 蝶式互连函数(Butterfly)：把输入端的二进制编号的最高位与最低位互换位置，便得到了输出端的编号。

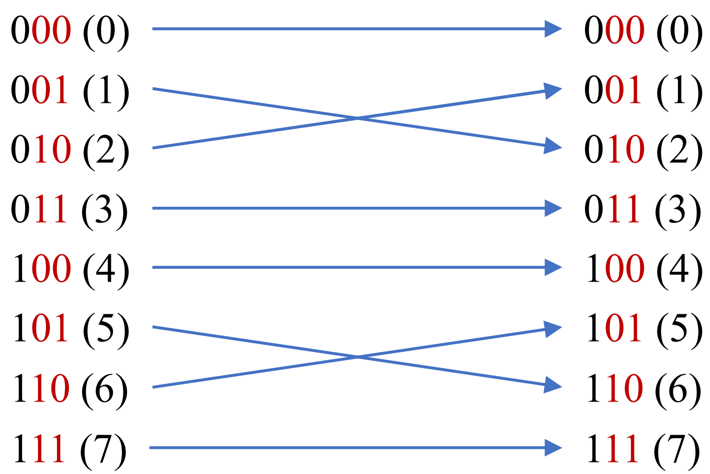
$$\beta(x_{n-1}x_{n-2}\cdots x_1x_0) = x_0x_{n-2}\cdots x_1x_{n-1}$$



7.1.2 互连函数

➢ <u>子蝶式</u>（subbutterfly）：把输入端的二进制编号的低k位中的最高位与最低位互换。

$$\beta_{(k)}(x_{n-1}\cdots x_k x_{k-1} x_{k-2}\cdots x_1 x_0) = x_{n-1}\cdots x_k x_0 x_{k-2}\cdots x_1 x_{k-1}$$



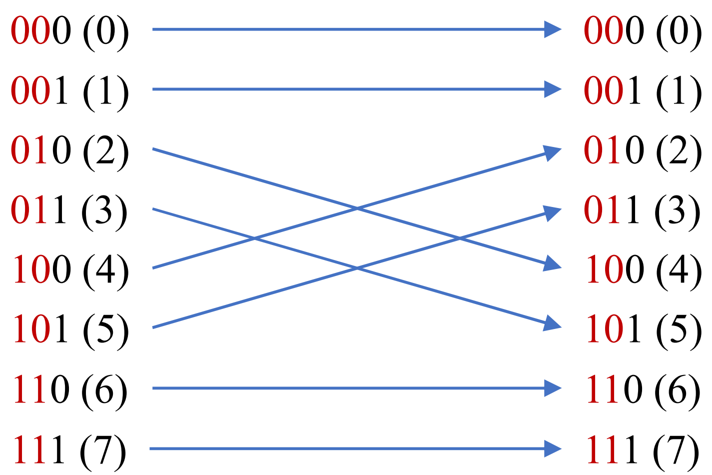
k=2

7.1.2 互连函数

➢ 超蝶式（superbutterfly）：把输入端的二进制编号的高k位中的最高位与最低位互换。

$$\beta^{(k)}(x_{n-1}x_{n-2}\cdots x_{n-k+1}x_{n-k}x_{n-k-1}\cdots x_1x_0)$$

$$= x_{n-k}x_{n-2}\cdots x_{n-k+1}x_{n-1}x_{n-k-1}\cdots x_1x_0$$

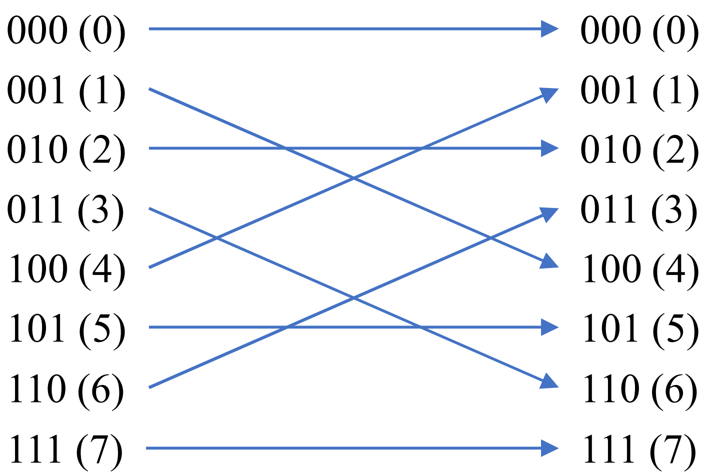


k=2

7.1.2 互连函数

- <u>反位序函数</u>（Bit Reversal）：将输入端二进制编号的位序颠倒过来求得相应输出端的编号。

$$\rho(x_{n-1}x_{n-2}\cdots x_1x_0) = x_0x_1\cdots x_{n-2}x_{n-1}$$



7.1.2 互连函数

- 子反位序：即把输入端的二进制编号的低k位中各位的次序颠倒过来

$$\rho_{(k)}(x_{n-1}\cdots x_k x_{k-1}x_{k-2}\cdots x_1x_0) = x_{n-1}\cdots x_k x_0 x_1\cdots x_{k-2}x_{k-1}$$

- 超反位序：即把输入端的二进制编号的高 k 位中各位的次序颠倒过来。

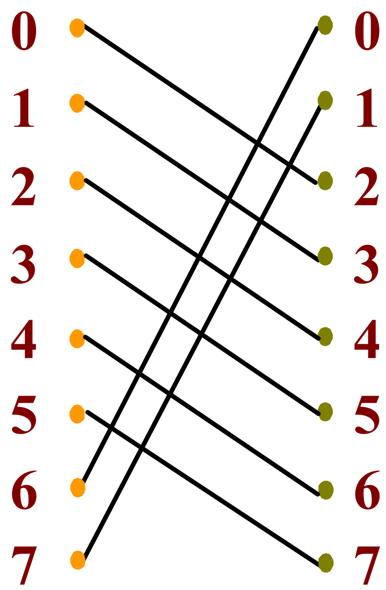
$$\rho^{(k)}(x_{n-1}x_{n-2}\cdots x_{n-k+1}x_{n-k}x_{n-k-1}\cdots x_1x_0)$$

$$= x_{n-k}x_{n-k+1}\cdots x_{n-2}x_{n-1}x_{n-k-1}\cdots x_1x_0$$

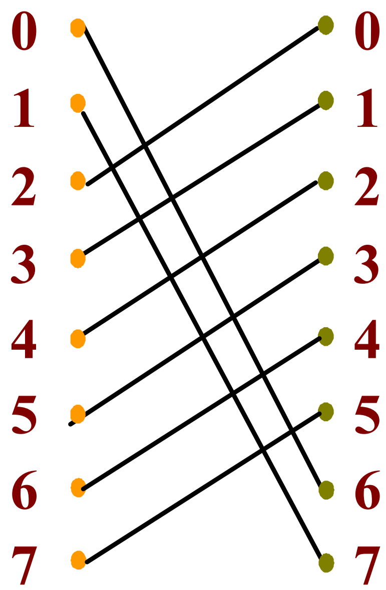
7.1.2 互连函数

➢ <u>移数函数</u>：将各输入端都错开一定的位置（模N）后连到输出端。

$$\alpha(x) = (x \pm k) \bmod N \quad 1 \le x \le N-1，\ 1 \le k \le N-1$$



(a) 左移移数函数 k=2



(b) 右移移数函数 k=2

7.1.2 互连函数

➢ PM2I函数：P和M分别表示加和减，$2I$表示2^i。

- 该函数又称为“加减2^{i}”函数。
- 一种移数函数，将各输入端都错开一定的位置（模N）后连到输出端。
- 互连函数

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

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

其中：$0 \le x \le N-1$，$0 \le i \le n-1$，$n = \log_2 N$，N为节点数。

- PM2I互连网络共有$2n$个互连函数。

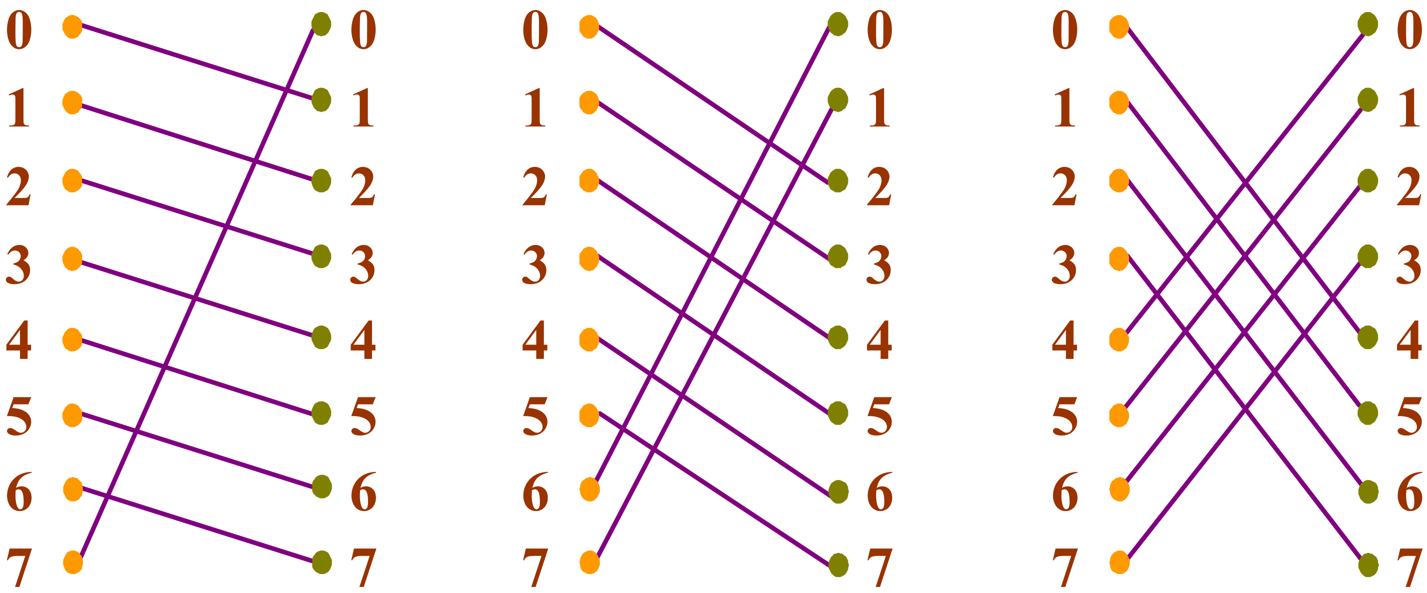
7.1.2 互连函数

➢ PM2I函数：当N＝8时，有6个PM2I函数：

- $PM2_{+0}$ ：（0 1 2 3 4 5 6 7）　$x + 2^0 \bmod 8$
- $PM2_{-0}$ ：（7 6 5 4 3 2 1 0）　$x - 2^0 \bmod 8$
- $PM2_{+1}$ ：（0 2 4 6 ）（1 3 5 7）　$x + 2^1 \bmod 8$
- $PM2_{-1}$ ：（6 4 2 0）（7 5 3 1）　$x - 2^1 \bmod 8$
- $PM2_{+2}$ ：（0 4）（1 5）（2 6）（3 7）　$x + 2^2 \bmod 8$
- $PM2_{-2}$ ：（4 0）（5 1）（6 2）（7 3）　$x - 2^2 \bmod 8$

$PM2_{+0}$ ： （0 1 2 3 4 5 6 7）

$PM2_{+2}$ ： （0 4） （1 5） （2 6） （3 7）

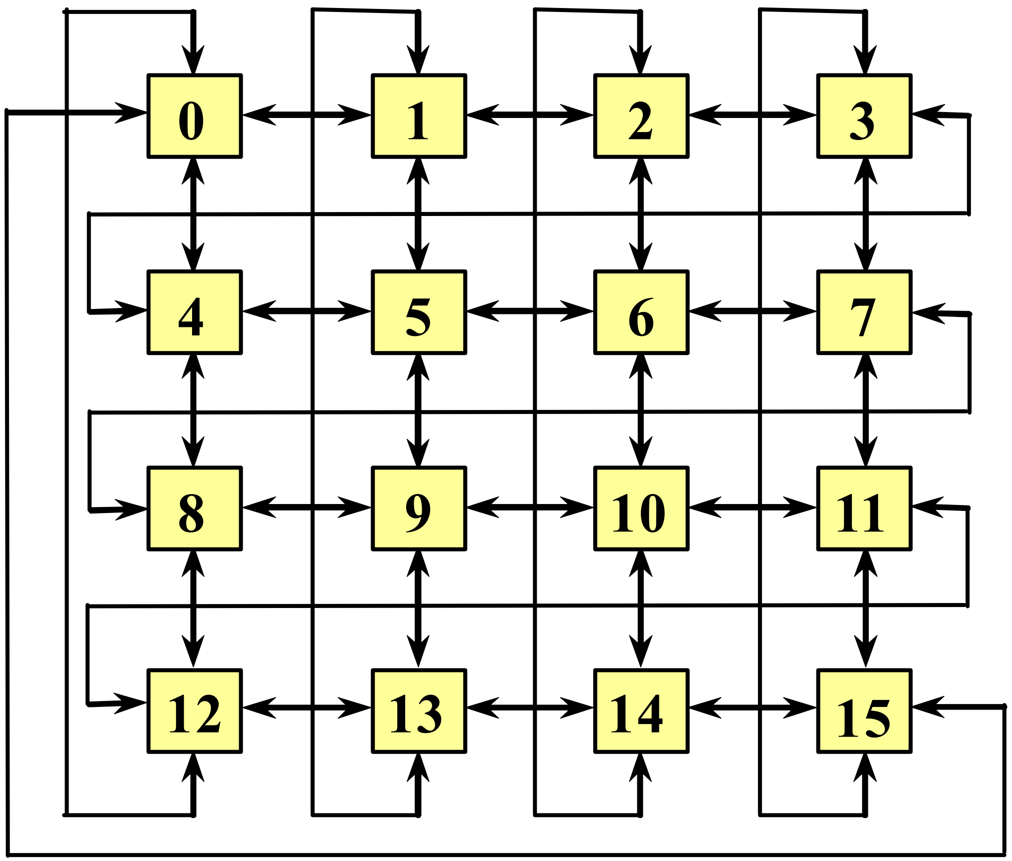


(a) $PM2_{+0}$　　**(b) $PM2_{+1}$**　　**(c) $PM2_{+2}$**

$PM2_{+1}$ ： （0 2 4 6 ） （1 3 5 7）

N=8 的PM2I函数

➢ 阵列计算机ILLIAC Ⅳ采用$PM2_{\pm 0}$和$PM2_{\pm n/2}$构成其互连网络，实现各处理单元之间的上下左右互连 。



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

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

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

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

用移数函数构成ILLIAC Ⅳ 阵列机的互连网络

现有16个处理器，编号分别为0，1，…，15，用一个N=16的互连网络互连。处理器i的输出通道连接互连网络的输入端i，处理器i的输入通道连接互连网络的输出端i。当该互连网络实现的互连函数分别为：

（1）$Cube_3$

（2）$PM2_{+3}$

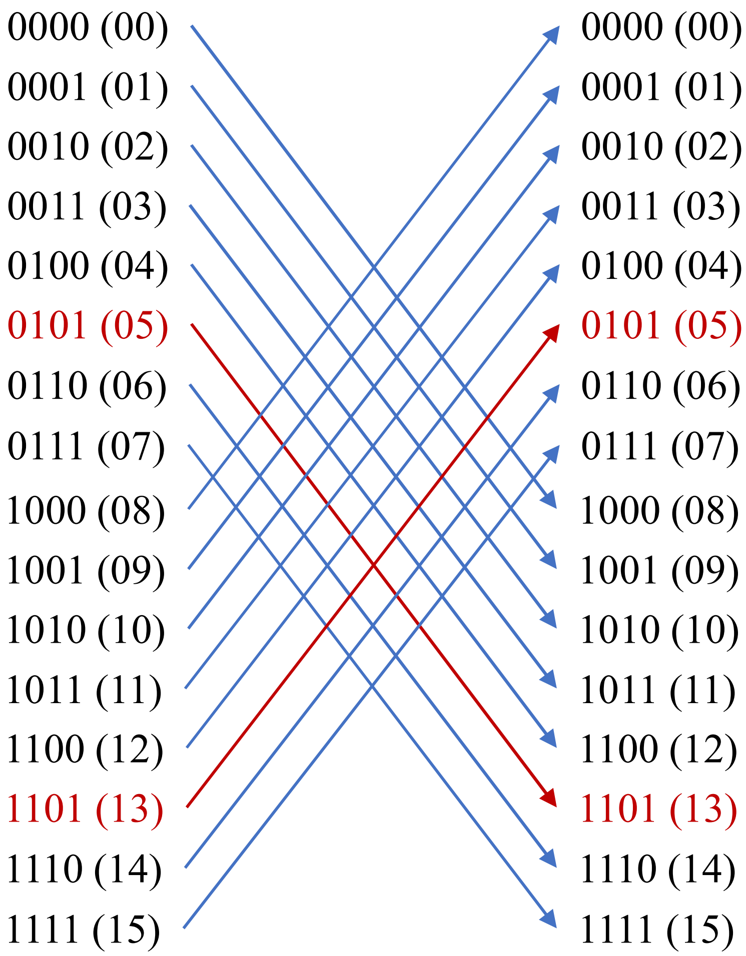
（3）$PM2_{-0}$

（4）σ

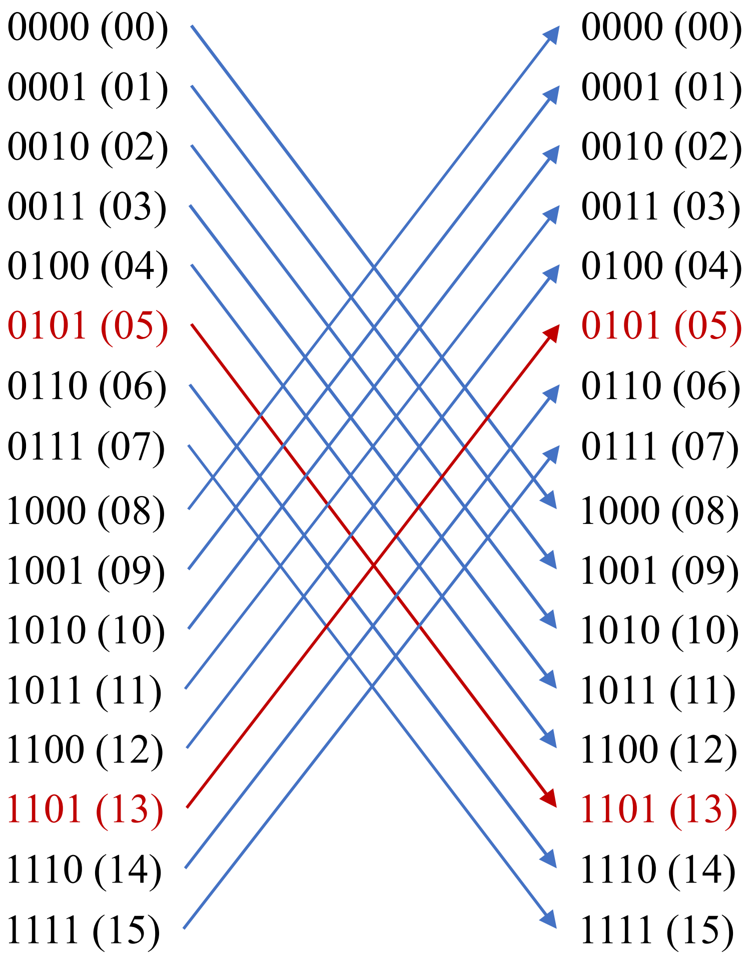
（5）σ（σ）

分别给出与第13号处理器所连接的处理器号。

$$Cube_3(x_3x_2x_1x_0) = \bar{x}_3x_2x_1x_0$$



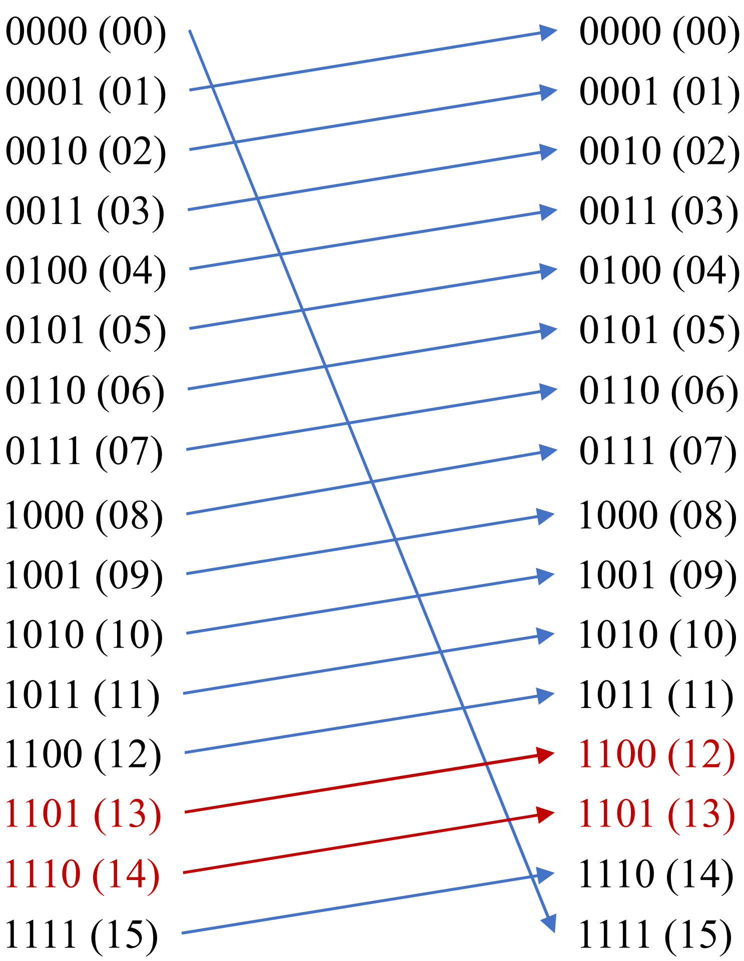
第5号处理器



$$PM2_{+3} = x + 2^3 \bmod 16$$

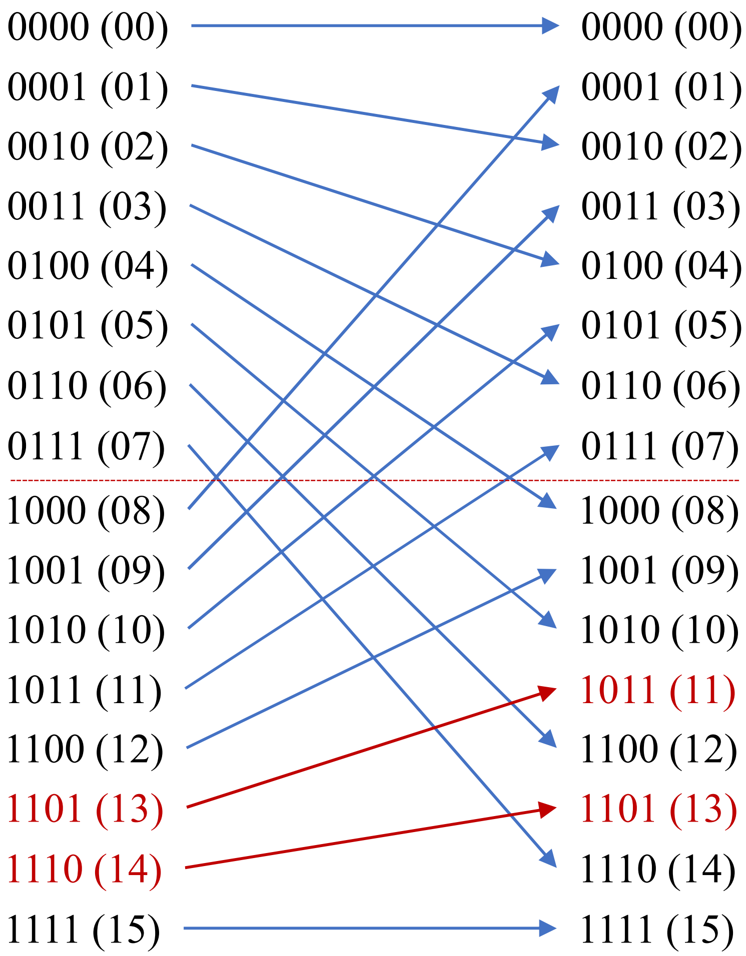
第5号处理器

$$PM2_{-0} = x - 2^0 \bmod 16$$



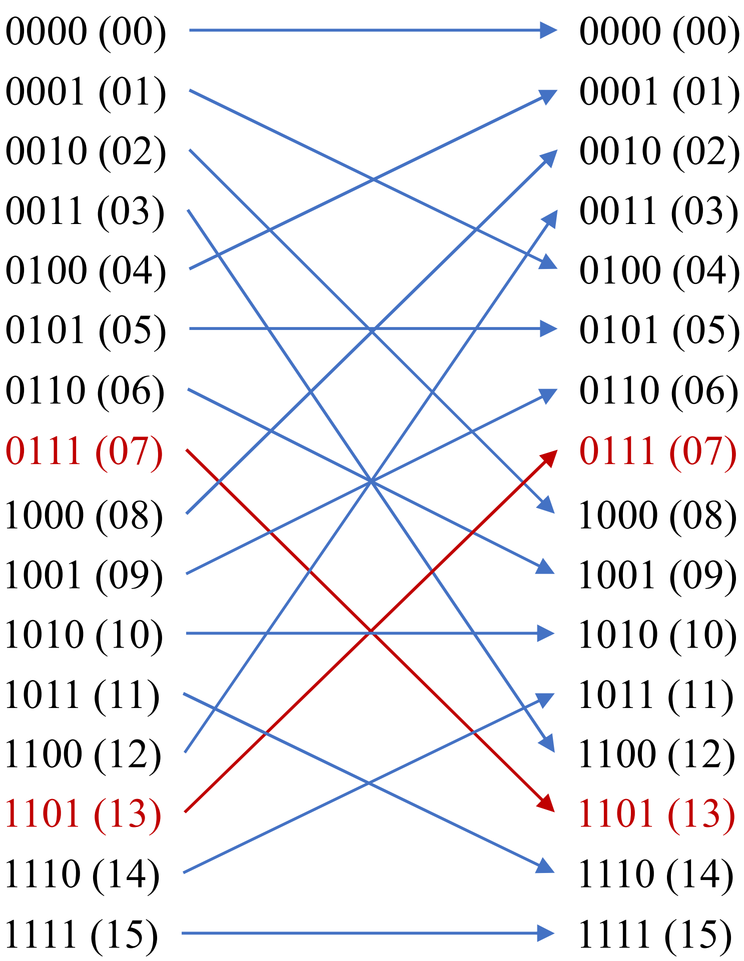
第12, 14号处理器

$$\sigma(x_3x_2x_1x_0) = x_2x_1x_0x_3$$



第11, 14号处理器

$$\sigma(\sigma(x_3x_2x_1x_0)) = x_1x_0x_3x_2$$



第07号处理器

7.1.3 互连网络的特性

互连网络通常是用有向边或无向边连接有限个结点的组成，主要特性有：

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，进入结点的边数叫入度，从结点出来的边数则叫出度
3） 距离：两个结点之间相连的最少边数
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） 等分宽度：把由N个节点构成的网络切成节点数相同（N/2）的两半，在各种切法中，沿切口边数的最小值
6） 结点间的线长：两个节点间线的长度（影响信号的时延、时钟扭斜和对功率的需要）
7） 对称性：从任何节点看到的拓扑结构都是相同的网络称为对称网络

7.1.4 互连网络的传输性能参数

- 发送方的步骤如下：

(1) 用户程序把要发送的数据拷贝到系统缓冲区。

(2) 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。

- 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系统缓冲区。

(2) 检查收到的数据包，如果正确，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（若发送方的超时计数器已超时，则重新发送消息）

7.1.4 互连网络的传输性能参数

➢互连网络的主要性能参数：

1） 频带宽度（Bandwidth)：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。

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。

7.1.4 互连网络的传输性能参数

- 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

- 例：假设一个网络的频宽为10Mb/S，发送方开销为230us，接收方开销为270us。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?

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：

$$T = \text{发送方开销} + \text{飞行时间} + \frac{\text{消息长度}}{\text{频宽}} + \text{接收方开销}$$
$$= 230\mu s + \frac{0.1 \times 10^6}{0.5 \times 299792.5Km/s} + \frac{1000 \times 8\text{位} \times 10^6}{10\text{兆位/秒}} + 270\mu s$$
$$= 230\mu s + 0.67\mu s + 800\mu s + 270\mu s = 1301\mu s$$

相距1000公里时的总时延为：

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

7.2 互连网络的种类

- 互连网络通常可以分为两大类：
  - <u>静态互连网络</u>：各节点之间有固定的连接通路、且在运行中不能改变的网络。
  - <u>动态互连网络</u>：由交换开关构成、可按运行程序的要求动态地改变连接状态的网络。

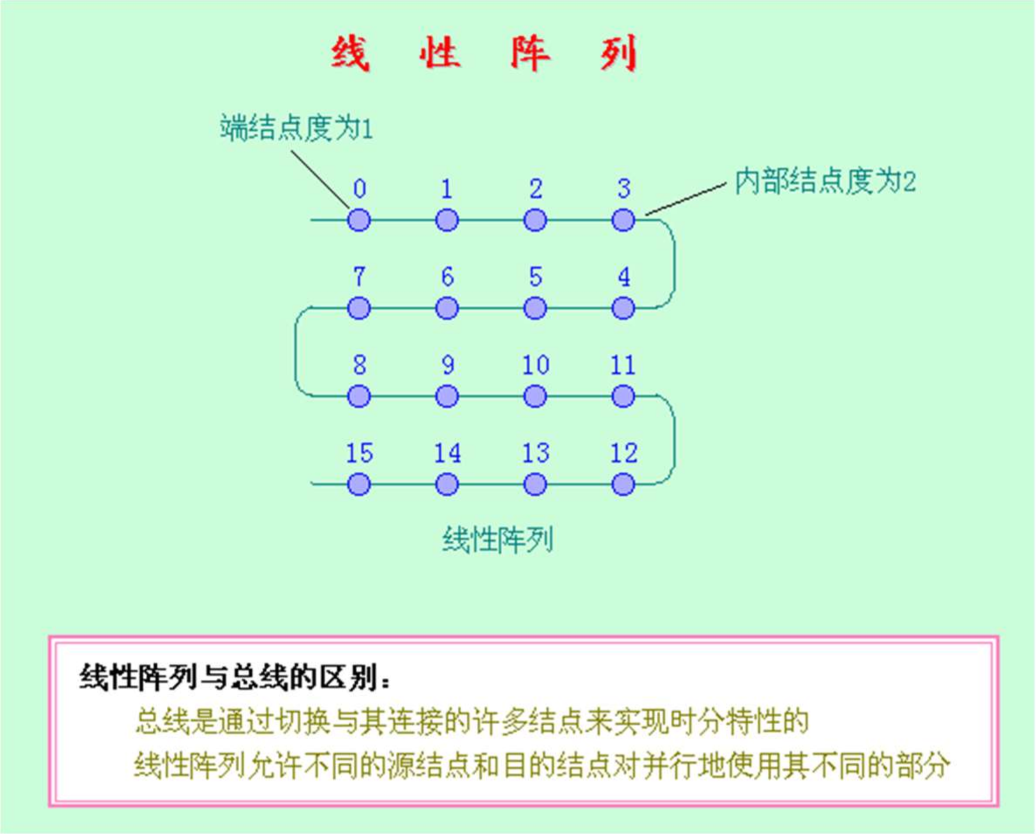
7.3 静态互连网络

- 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。
- 一般静态互连网络不能实现任意结点到结点之间的互连。
- 一维的有线性阵列结构；二维的有环形、星形、树形、网格形等；三维的有立方体等；三维以上的有超立方体等。

7.3 静态互连网络

1.　<u>线性阵列</u>：一种一维的线性网络，其中N个节点用N-1个链路连成一行

- 端节点的度：1
- 其余节点的度：2
- 直径：N－1
- 等分宽度b=1

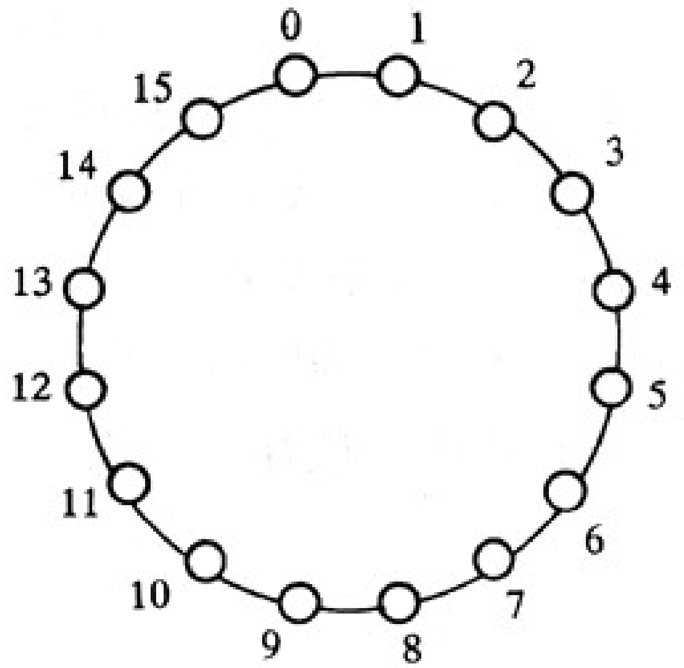


7.3 静态互连网络

2. <u>环和带弦环</u>

➢环：用一条附加链路将线性阵列的两个端点连接起来而构成。可以单向工作，也可以双向工作

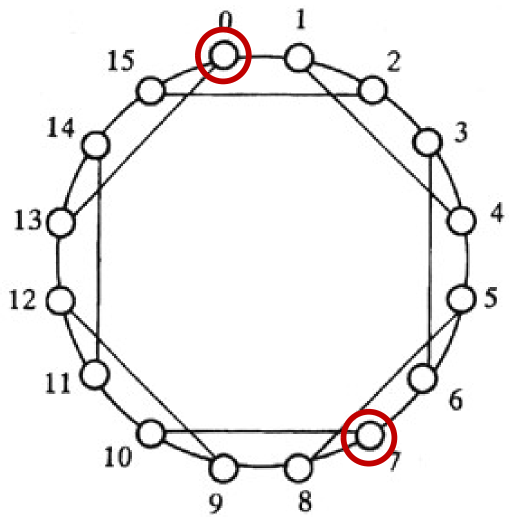
- 对称
- 节点的度：2
- 双向环的直径：N/2
- 单向环的直径：N
- 环的等分宽度b=2



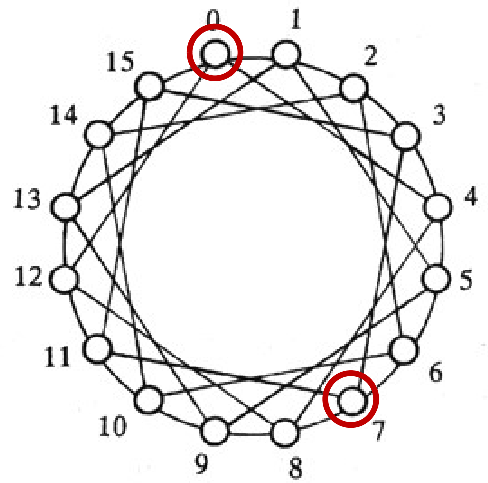
7.3 静态互连网络

2. 环和带弦环

➢ 带弦环：分别增加一条或两条附加链路可以得到如下的带弦环，增加的链路愈多，节点度愈高，网络直径就愈小



节点度3，直径5



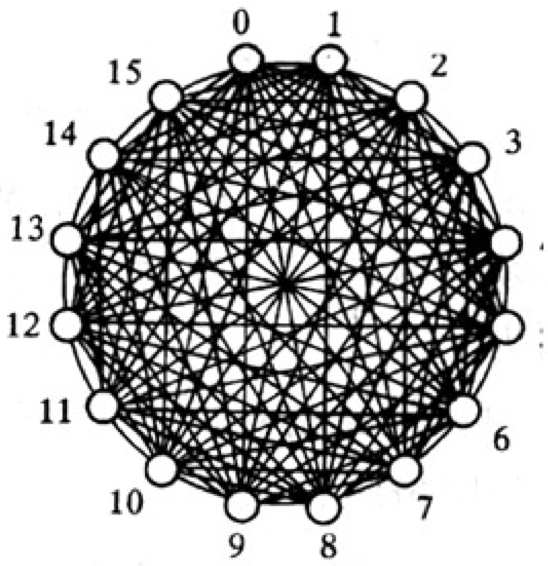
节点度4，直径3

7.3 静态互连网络

2.　环和带弦环

➢带弦环：分别增加一条或两条附加链路可以得到如下的带弦环，增加的链路愈多，节点度愈高，网络直径就愈小

- 节点度：15
- 直径最短，为1。

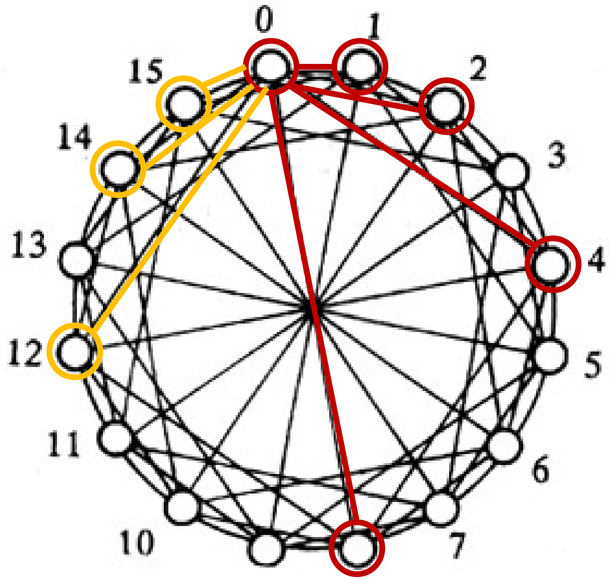


全连接

7.3 静态互连网络

3. 循环移数网络

➢通过在环上每个节点到所有与其距离为2的整数幂的节点之间都增加一条附加链而构成。



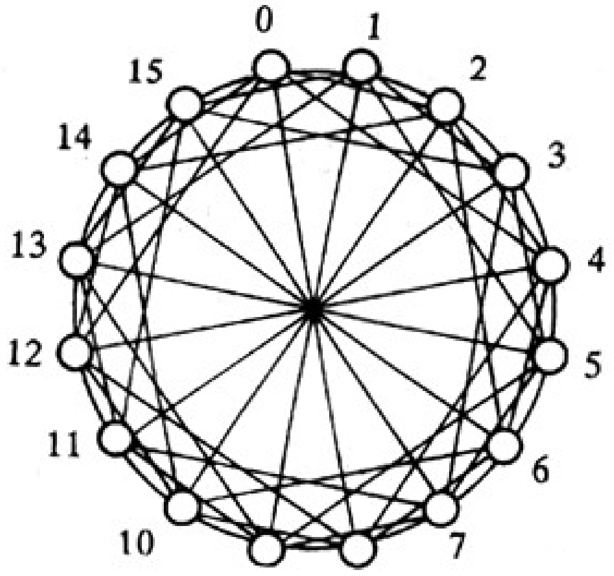
(e)循环移数网络

- 循环移数网络的连接特性与结点度较低的任何带弦环相比有了改进
- 对$N = 16$的情况，循环移数网络的结点度为7，直径为2
- 复杂性比全连接网络要低得多

7.3 静态互连网络

3. 循环移数网络

➢ 通过在环上每个节点到所有与其距离为2的整数幂的节点之间都增加一条附加链而构成。



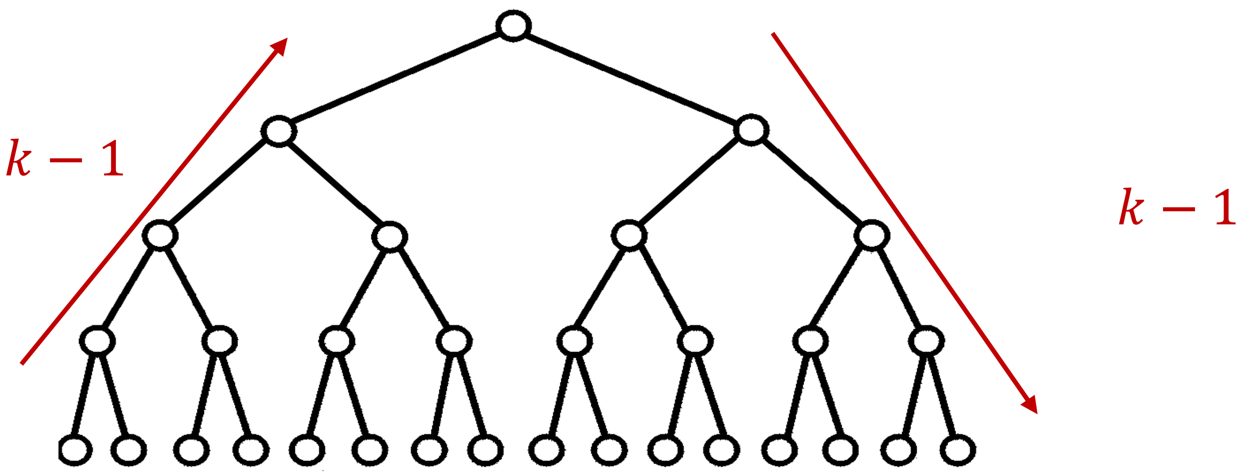
(e)循环移数网络

- 如果$|j-i|=2^r$（$r=0,1,\cdots,n-1$），则节点i与节点j连接。
- 节点度：$2n-1$
- 直径：$\frac{n}{2}$
- 网络规模$N=2^n$

7.3 静态互连网络

4. 树形和星形

- 树形：一颗k层完全平衡的二叉树应有$N = 2^k - 1$个结点，最大结点度是3，直径是$2(k-1)$，等分宽度$b = 1$
- 由于结点度是常数，因此二叉树是一种可扩展的系统结构，但其直径相当长

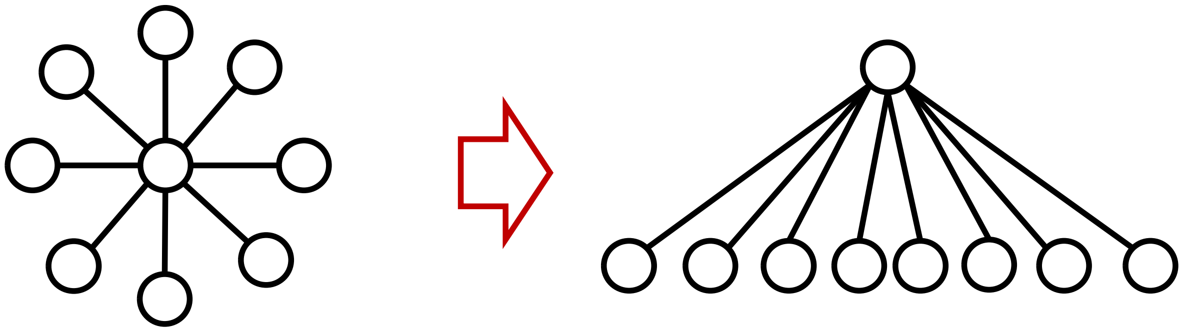


(a) 二叉树

7.3 静态互连网络

4. 树形和星形

- 星形：是一种2层树，结点度较高，为$d = N - 1$；直径较小，为常数2；等分宽度为$b = \lfloor N/2 \rfloor$
- 可靠性较低，只要中心结点出故障，整个系统就会瘫痪

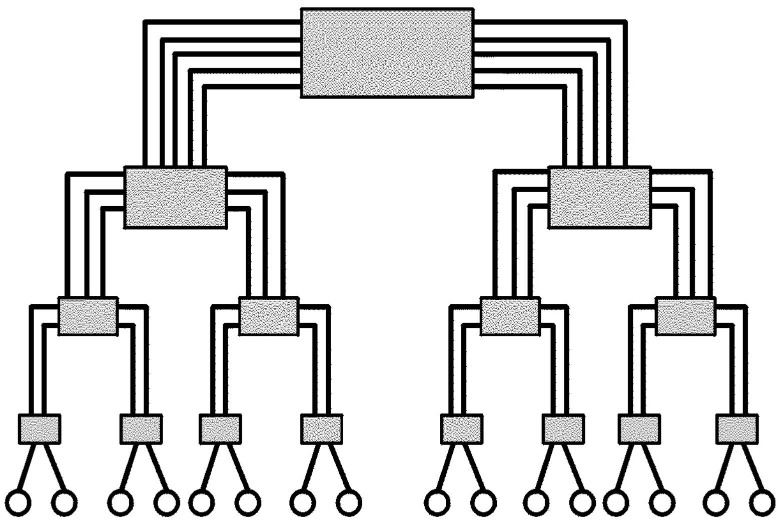


（b）星形

7.3 静态互连网络

5. 胖形树

- 胖树的通道宽度从叶结点往根结点上行方向逐渐增宽
- 传统二叉树的主要问题之一就是通向根结点的瓶颈问题，这是因为根部的通信最忙，而胖树的提出使问题得到了缓解



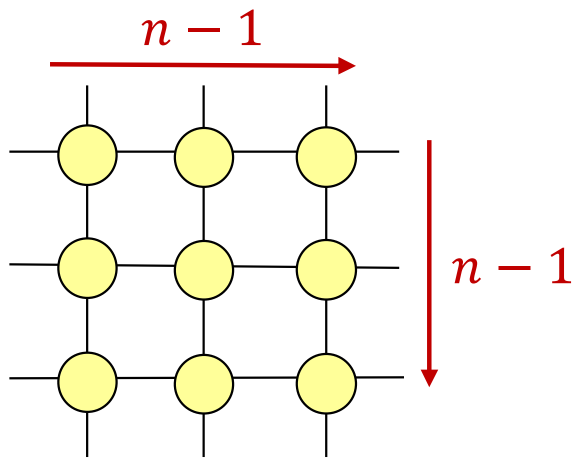
(c) 二叉胖树

7.3 静态互连网络

6. 网格形与环网形

➢ 网格形

- 一个规模为$N = n \times n$的2维网格形网络
  - 内部节点的度$d = 4$
  - 边节点的度$d = 3$
  - 角节点的度$d = 2$
  - 网络直径$D = 2(n-1)$
  - 等分宽度$b = n$
- 一个由$N = n^k$个节点构成的k维网格形网络（每维n个节点）的内部节点度$d = 2k$，网络直径$D = k(n-1)$



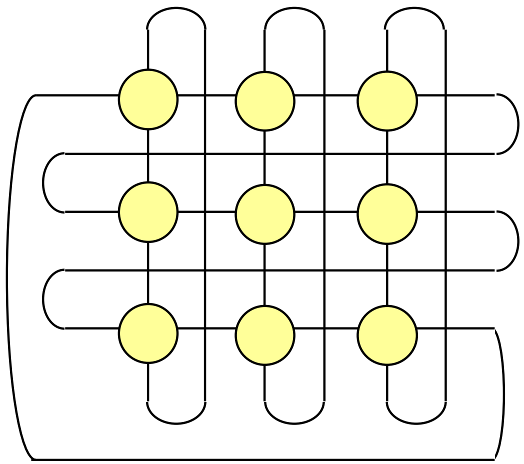
（a）网格形

7.3 静态互连网络

6. 网格形与环网形

- Illiac网络：名称来源于采用这种网络的Illiac IV计算机
  - 把二维网格形网络的每一列的两个端节点连接起来，再把每一行的尾节点与下一行的头节点连接起来，并把最后一行的尾节点与第一行的头节点连接起来
  - 一个规模为$n \times n$的Illiac网络
    - 所有结点的度$d = 4$
    - 网络直径$D = n - 1$
    - 等分宽度$b = 2n$

n=2 D=1;
n=3 D=2;
n=4 D=3;



（b）Illiac网

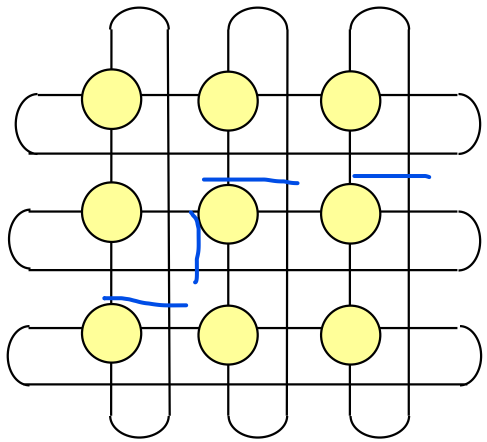
7.3 静态互连网络

6. 网格形与环网形

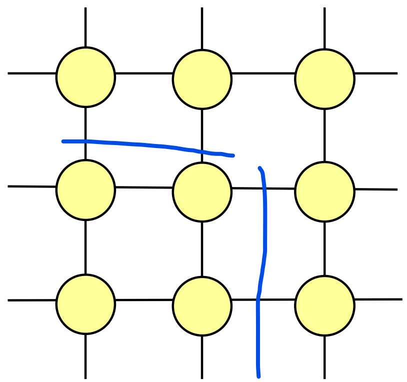
- 环网形
  - 把2维网格形网络的每一行的两个端节点连接起来，把每一列的两个端节点也连接起来
  - 将环形和网格形组合在一起，并能够向高维扩展
  - 一个规模为$n \times n$的环网形网络
    - 结点的度$d = 4$
    - 网络直径$D = 2 \times \lfloor n/2 \rfloor$
    - 等分宽度$b = 2n$

$\left\lfloor \frac{3}{2} \right\rfloor$

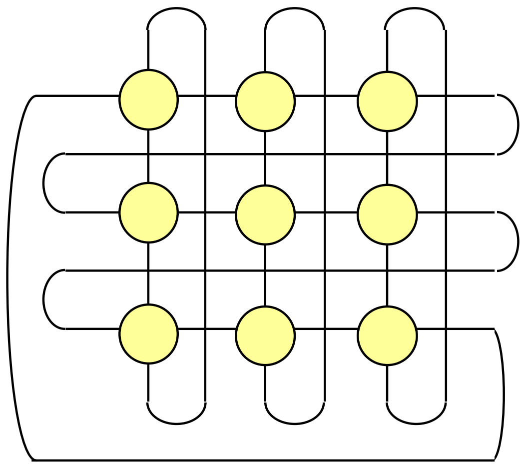
n=2 D=2;
n=3 D=2;
n=4 D=4;



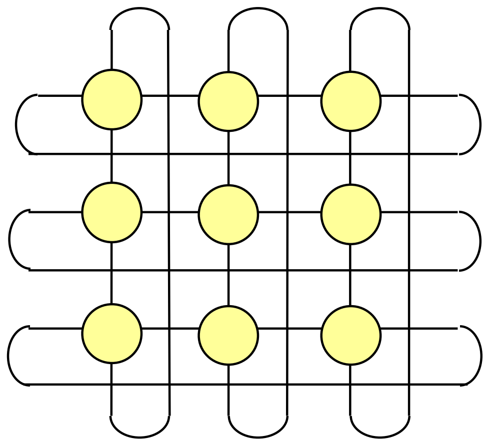
（c）环网形



（a）网格形



（b）Illiac网

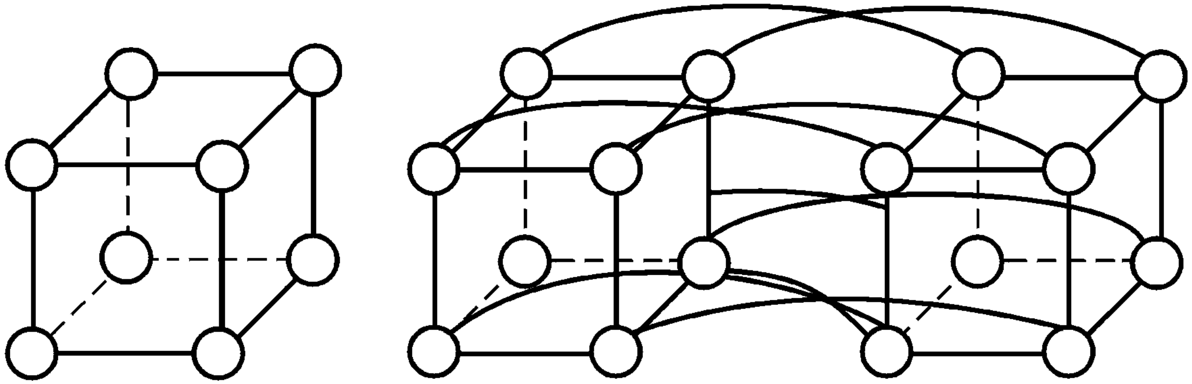


（c）环网形

7.3 静态互连网络

7. 超立方体

- 一种二元n-立方体结构，由$N = 2^n$个节点组成，它们分布在n维上，每维有两个节点
- 为实现一个n-立方体，只要把两个$n-1$立方体中相对应的节点用链路连接起来即可。共需要2^{n-1}条链路
- n-立方体中节点的度都是n，直径也是n，等分宽度为$b = N/2$

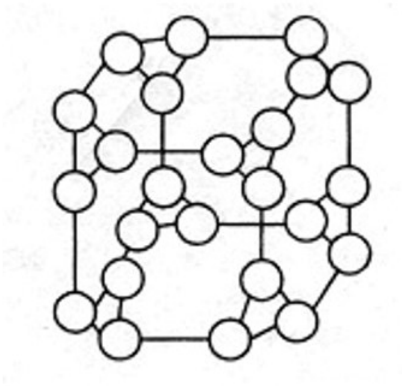


(a) 3–立方体　　(b) 由 2 个 3– 立方体组成的 4– 立方体

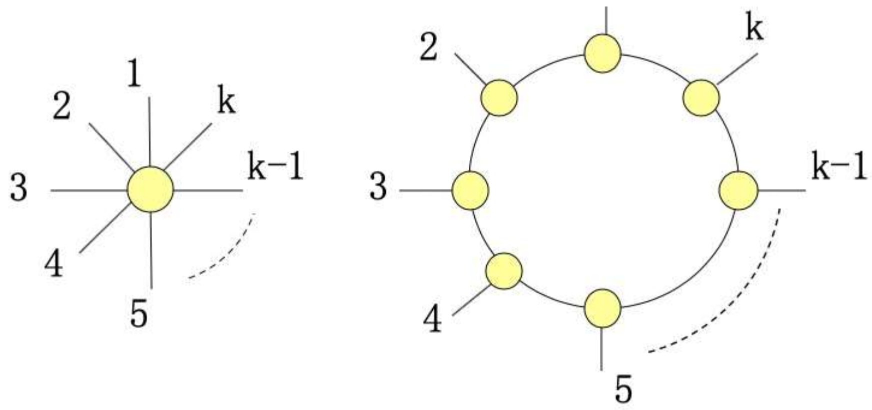
7.3 静态互连网络

8. 带环立方体

- 带环k-立方体（简称k -CCC），是k -立方体的变形，通过用k个节点构成的环取代k -立方体中的每个节点而形成的。
- 网络规模为$N = k \times 2^k$；网络直径为$D = 2k - 1 + \lfloor k/2 \rfloor$，比$k$ -立方体的直径大一倍；等分宽度为$b = \frac{N}{2k}$



(c)把3-立方体的每个节点换成一个由3个节点构成的环，形成带环3-立方体



(b) 将k-立方体的每个节点用由k个节点的环来代替，组成带环k-立方体组成

7.3 静态互连网络

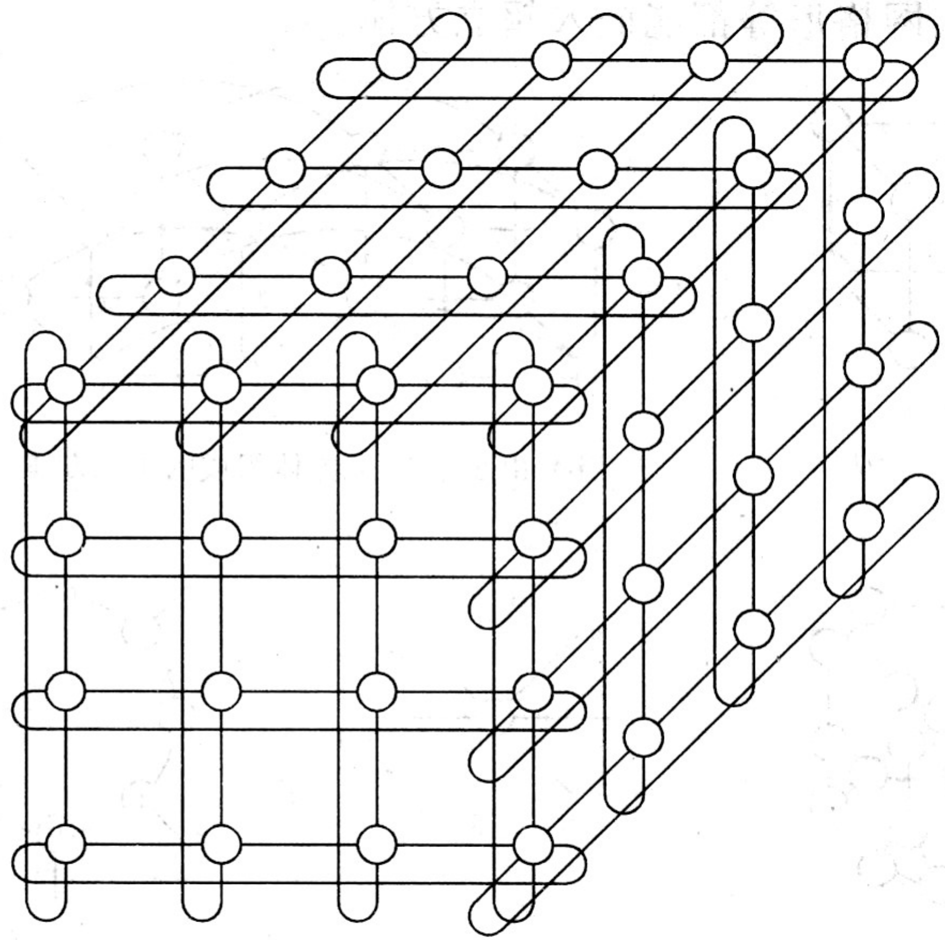
9. ***k*元*n*-立方体网络**

➢ 环形、网格、环网形、二元n-立方体（超立方体）都是k元n-立方体网络系列的拓扑同构体。

➢ 在k元n-立方体网络中，参数n是立方体的维数， k是基数，即每一维上的节点个数。

$$N = k^n, \left(k = \sqrt[n]{N}, n = \log_k N\right)$$

- k元n-立方体的节点可以用基数为k的n位地址$A = a_1 a_2 \ldots an$来表示。
- 其中a_i表示该节点在第i维上的位置
- 通常把低维k元n-立方体称为环网，而把高维k元n-立方体称为超立方体。



4元3-立方体网络

静态互连网络特征一览表

网络类型	节点度d	网络直径D	链路数l	等分宽度B	对称性	网络规格说明
线性阵列	2	N-1	N-1	1	非	N个节点
环形	2	[N/2]	N	2	是	N个节点
全连接	N-1	1	N(N-1)/2	$(N/2)^2$	是	N个节点
二叉树	3	2(h-1)	N-1	1	非	树高$h=[\log_2 N]$
星形	N-1	2	N-1	[N/2]	非	N个节点
2D网格	4	2(r-1)	2N-2r	r	非	r×r网格，$r=\sqrt{N}$
Illiac网	4	r-1	2N	2r	非	与 $r=\sqrt{N}$ 的带弦环等效
2D环网	4	2[r/2]	2N	2r	是	r×r环网，$r=\sqrt{N}$
超立方体	n	n	nN/2	N/2	是	N个节点，$n=[\log_2 N]$（维数）
CCC	3	2k-1+[k/2]	3N/2	N/(2k)	是	$N=k\times 2^k$节点 环长$k\geq 3$
k元n-立方体	2n	n[k/2]	nN	$2k^{n-1}$	是	$N=k^n$个节点

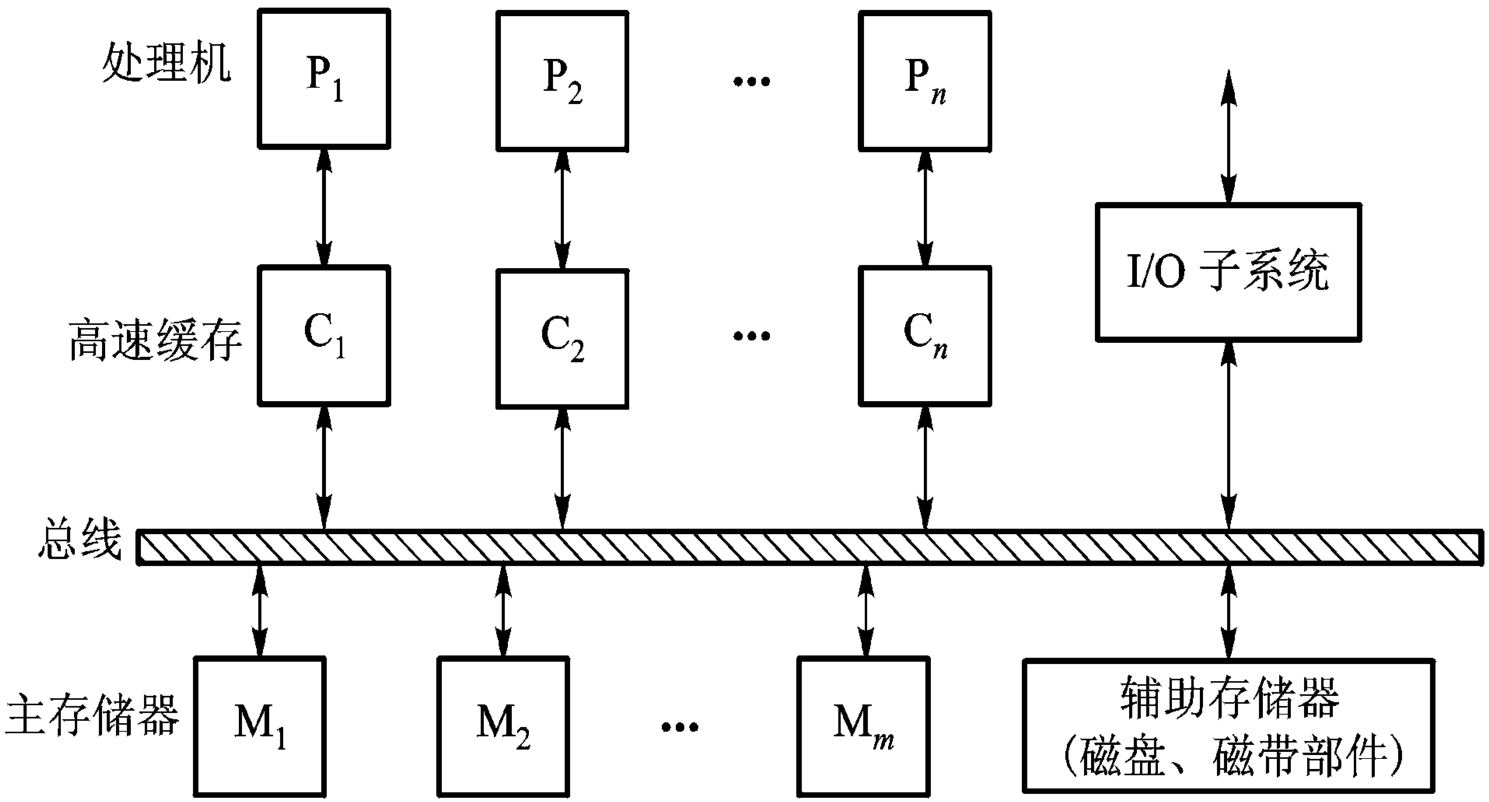
7.4 动态互连网络

7.4.1 总线网络

- 由一组导线和插座构成，经常被用来实现计算机系统中处理机模块、存储模块和外围设备等之间的互连。
  - 总线只能用于一个源（主部件）到一个或多个目的（从部件）之间一回处理一次业务
  - 多个功能模块之间的争用总线（Contention Bus）或时分总线(Time-Sharing Bus)

7.4 动态互连网络

7.4.1 总线网络



一种由总线连接的多处理机系统

7.4 动态互连网络

7.4.1 总线网络

- 特点
    - 结构简单、实现成本低、带宽较窄
    - 系统总线在处理机、I/O子系统、主存储器以及辅助存储设备（磁盘、磁带机等）之间提供了一条公用通路。
    - 系统总线通常设置在印刷电路板底板上。处理器板、存储器板和设备接口板都通过插座或电缆插入底板。

7.4 动态互连网络

7.4.1 总线网络

➢ 解决总线带宽较窄问题：采用多总线或多层次的总线

- 多总线是设置多条总线，有两种做法：1）为不同的功能设置专门的总线；2）重复设置相同功能的总线
- 多层次的总线是按层次的架构设置速度不同的总线，使得不同速度的模块有比较适合的总线连接。

7.4 动态互连网络

7.4.2 交叉开关网络

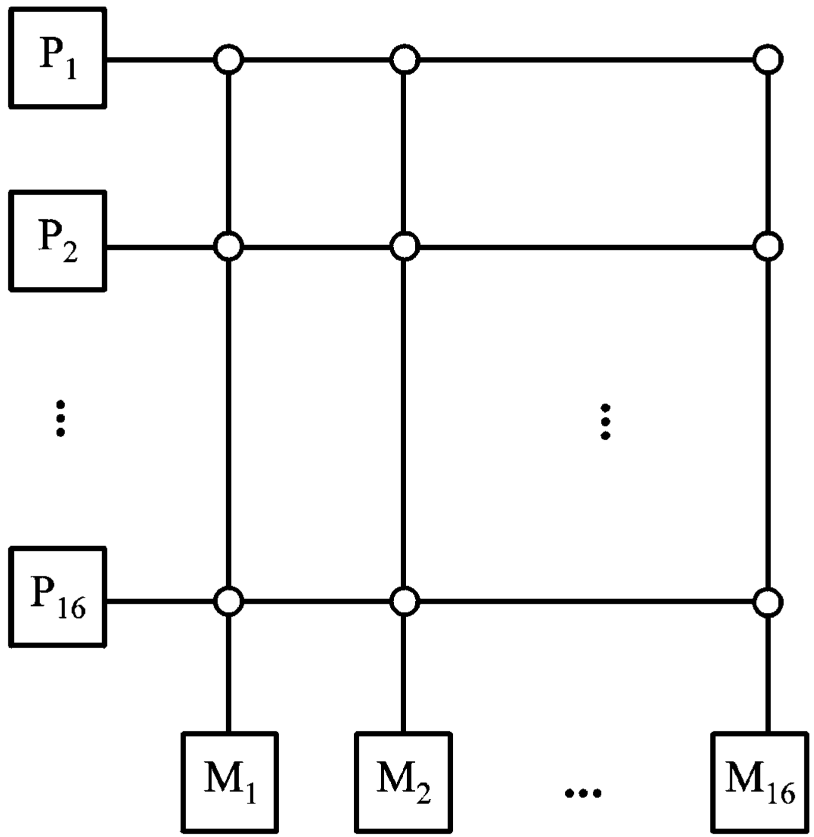
➢ 单级开关网络

- 交叉开关能在对偶（源、目的）之间形成动态连接，同时实现多个对偶之间的无阻塞连接。
- 带宽和互连特性最好。
- 一个$n \times n$的交叉开关网络，可以无阻塞地实现$n!$种置换。
- 对一个$n \times n$的交叉开关网络来说，需要n^2套交叉点开关以及大量的连线。
- 当n很大时，交叉开关网络所需要的硬件数量非常巨大。

7.4 动态互连网络

7.4.2 交叉开关网络

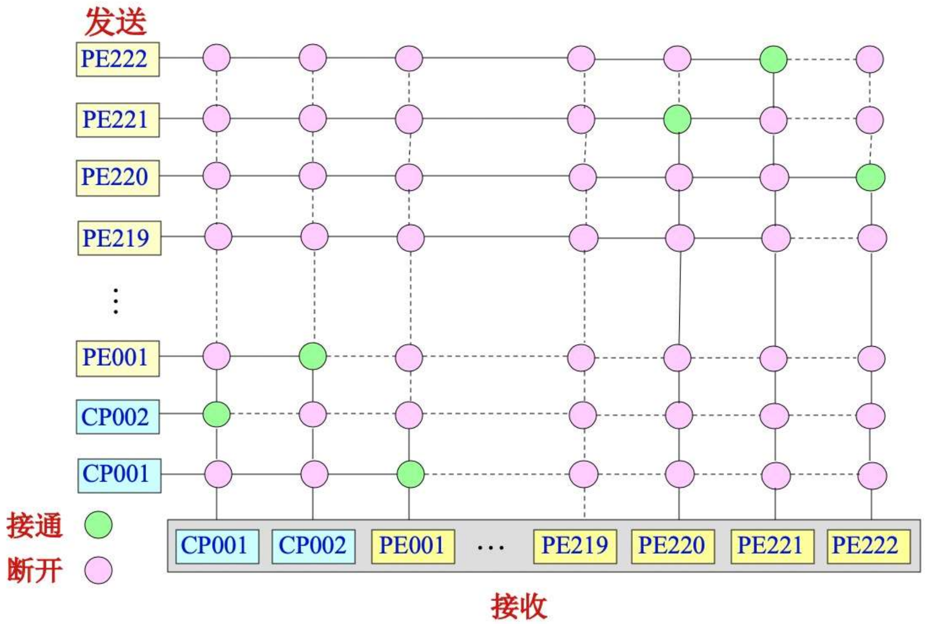
- C.mmp多处理机的互连结构
  - 用16×16的交叉开关网络把16台PDP-11处理机与16个存储模块连在一起
  - 最多可同时实现16台处理机对16个不同存储模块的并行访问
  - 每个存储模块一次只能满足一台处理机的请求
  - 当多个请求要同时访问同一存储模块时，交叉开关就必须分解所发生的冲突，每一列只能接通一个交叉点开关。
  - 为了支持并行（或交叉）存储器访问，可以在同一行中接通几个交叉点开关。



7.4 动态互连网络

7.4.2 交叉开关网络

- Fujitsu公司制造的向量并行处理机VPP500所采用的大型交换开关网络（224×224）
  - PE：带存储器的处理机
  - CP：控制处理机
  - 每一行和每一列只能接通一个交叉点开关



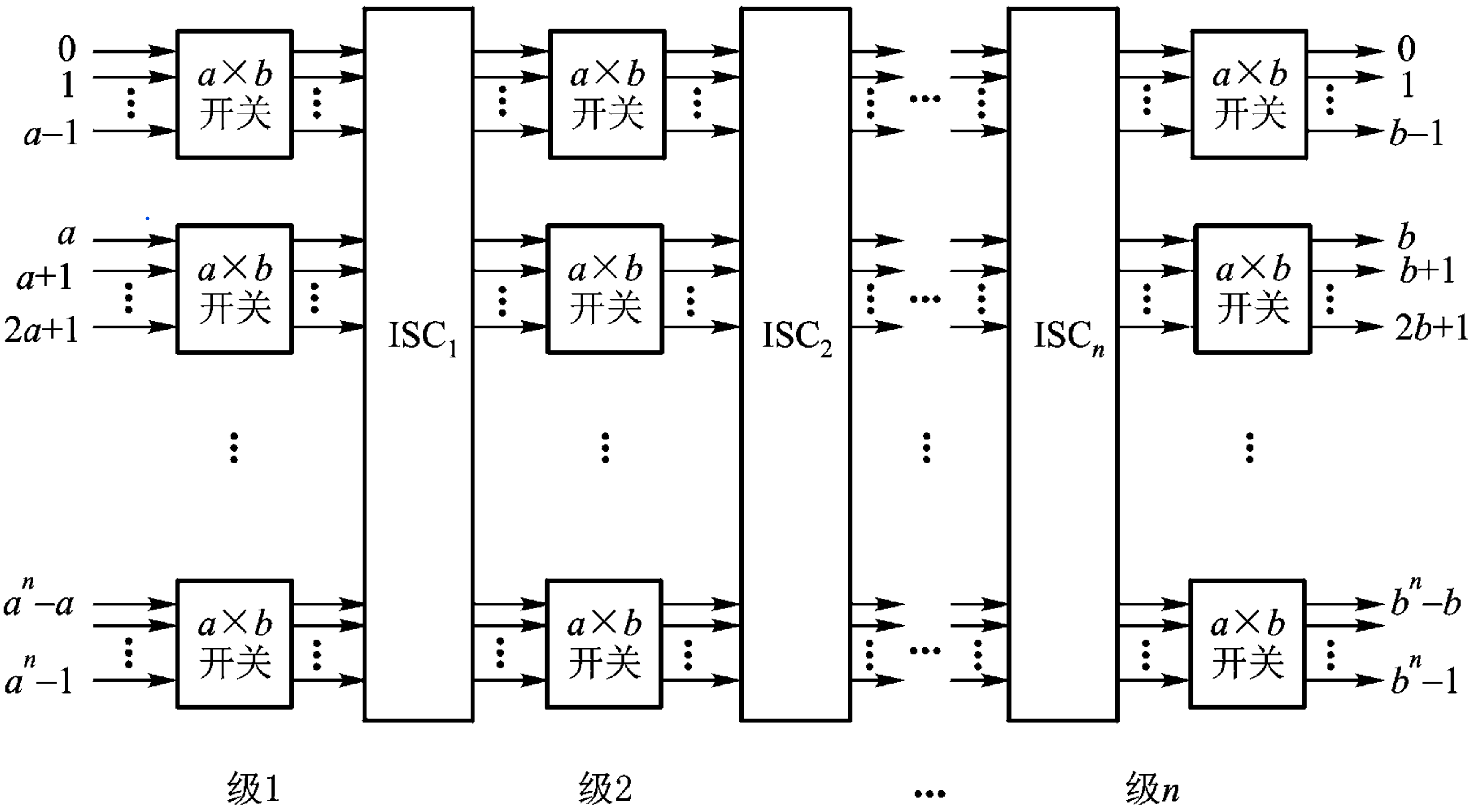
7.4 动态互连网络

7.4.3 多级互连网络

- 多级互连网络的构成
  - MIMD 和 SIMD 计 算 机 都 采 用 多 级 互 连 网 络 MIN （ Multistage Interconnection Network）
  - 一种通用的多级互连网络
    - 由a×b开关模块和级间连接构成的通用多级互连网络结构
    - 每一级都用了多个a×b开关
      - a个输入和b个输出
      - 在理论上，a和b不一定相等，然而实际上a和b经常选为2的整数幂，即$a=b=2^k$，k≥1。
    - 相邻各级开关之间都有固定的级间连接(ISC)

7.4 动态互连网络

7.4.3 多级互连网络



7.4 动态互连网络

7.4.3 多级互连网络

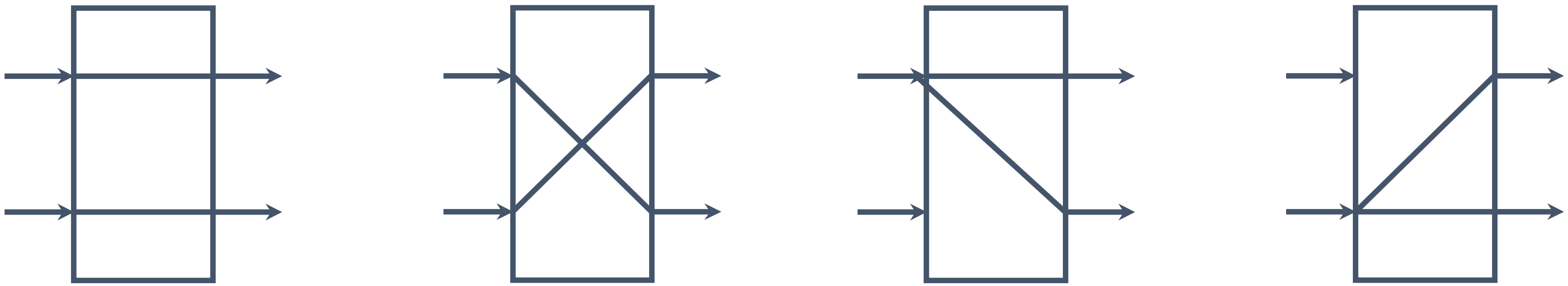
- 几种常用的开关模块
- 每个输入可与一个或多个输出相连，但是在输出端不许发生冲突

模块大小	合法状态	置换连接
2×2	4	2
4×4	256	24
8×8	16 777 216	40 320
n×n	n^n	n!

7.4 动态互连网络

7.4.3 多级互连网络

- 最简单的开关模块：2×2开关



直连　　交换　　上播　　下播

7.4 动态互连网络

7.4.3 多级互连网络

- 各种多级互连网络的区别在于所用开关模块、控制方式和级间互连模式的不同。
  - 控制方式：对各个开关模块进行控制的方式。
    - 级控制：每一级的所有开关只用一个控制信号控制，只能同时处于同一种状态。
    - 单元控制：每一个开关都有一个独立的控制信号，可各自处于不同的状态。
    - 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。
  - 常用的级间互连模式：均匀洗牌、蝶式、多路洗牌、纵横交叉、立方体连接等

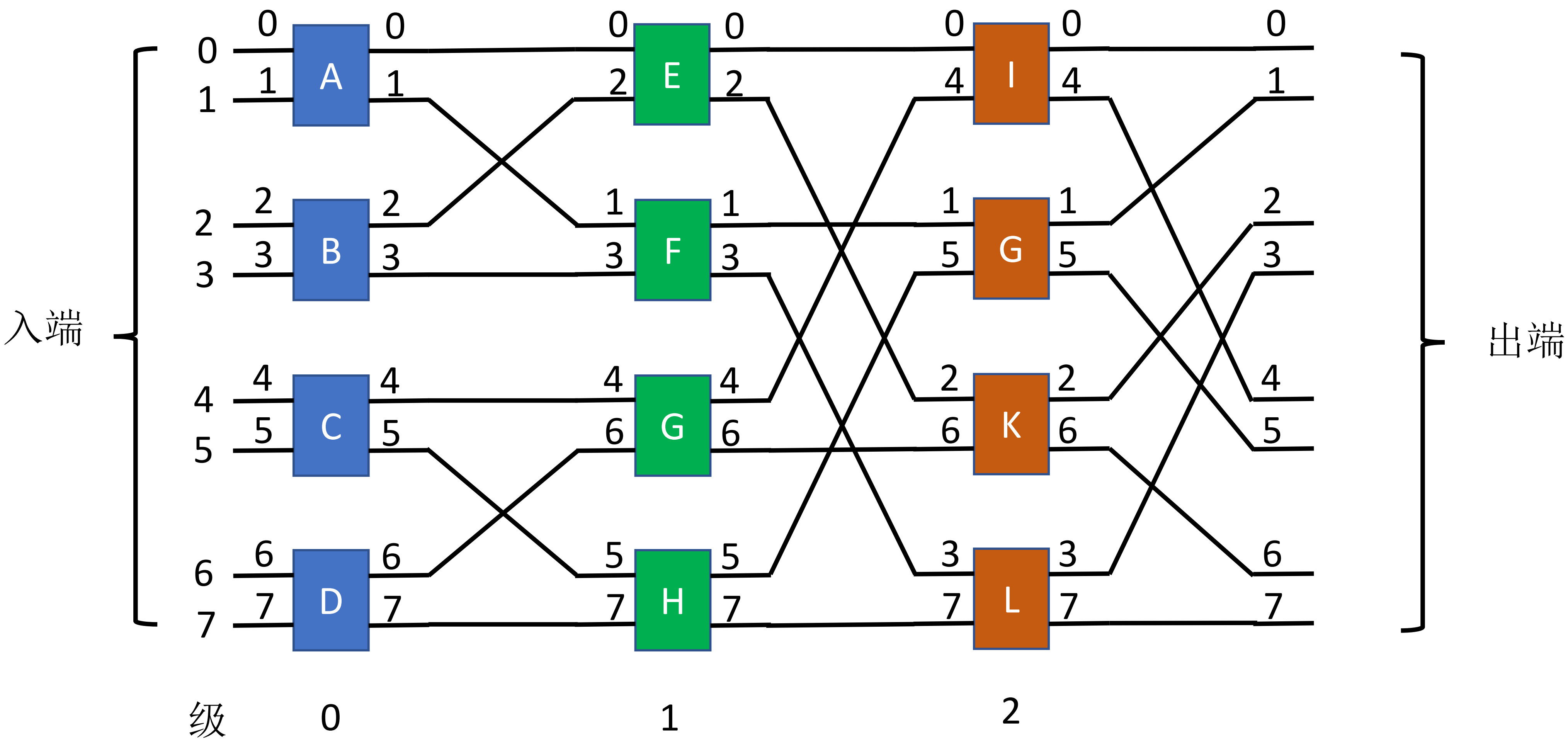
7.4.3 多级互连网络

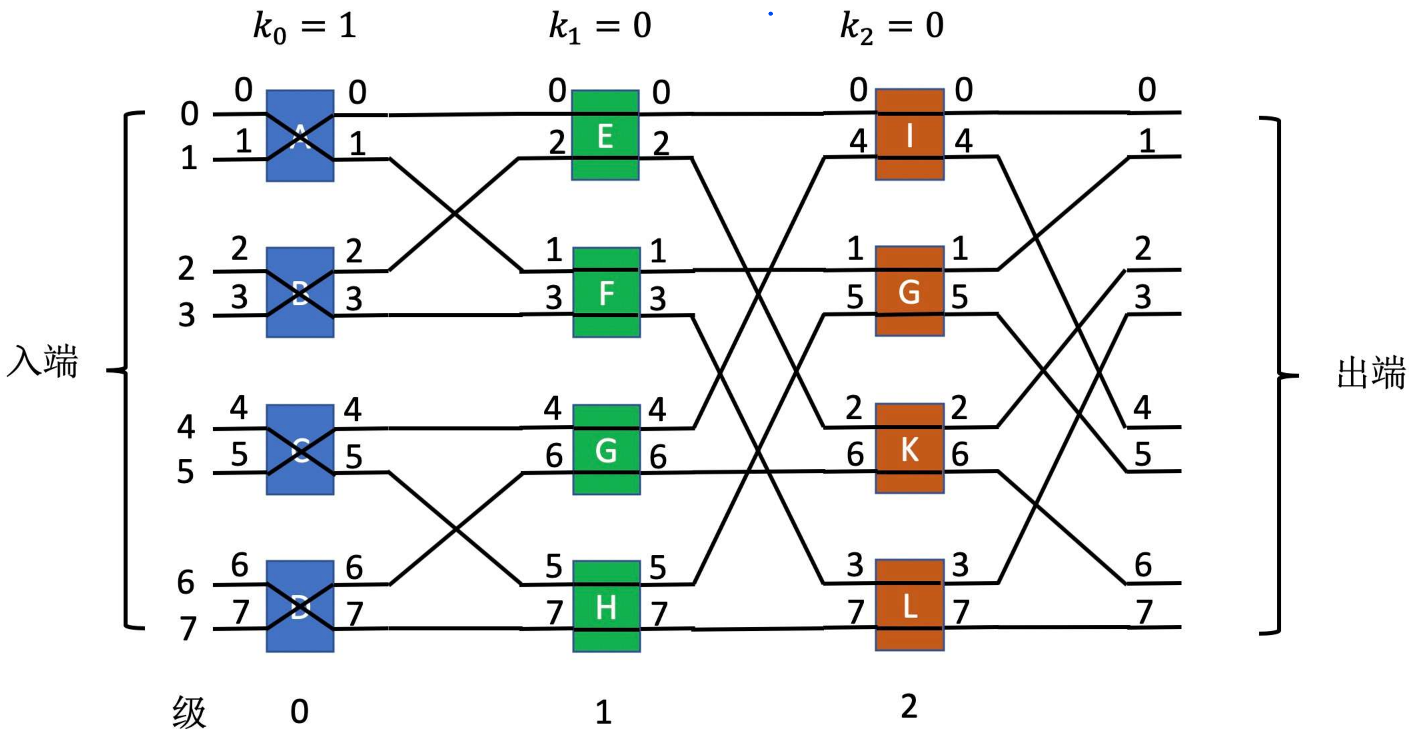
1 多级立方体网络

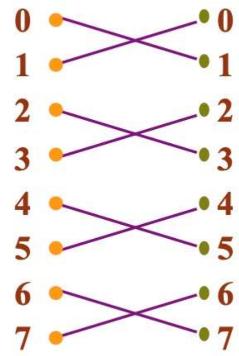
- 多级立方体网络包括STARAN网络和间接二进制$\boldsymbol{n}$方体网络等。
  - 两者仅在控制方式上不同：STRAN网络采用级控制（称交换网络）和部分级控制（其中可实现移数功能的称移数网络），而间接二进制n方体网络采用单元控制，具有更大的灵活性。
  - 共同点：都采用二功能（直送和交换）的2×2开关；当第i级（$0\le i\le n-1$）交换开关处于交换状态时，实现的是$Cube_i$互连函数。
- 一个N输入的多级立方体网络有$\log_2 N$级，每级用$N/2$个2×2开关模块，共需要$\log_2 N\times N/2$个开关。

7.4.3 多级互连网络

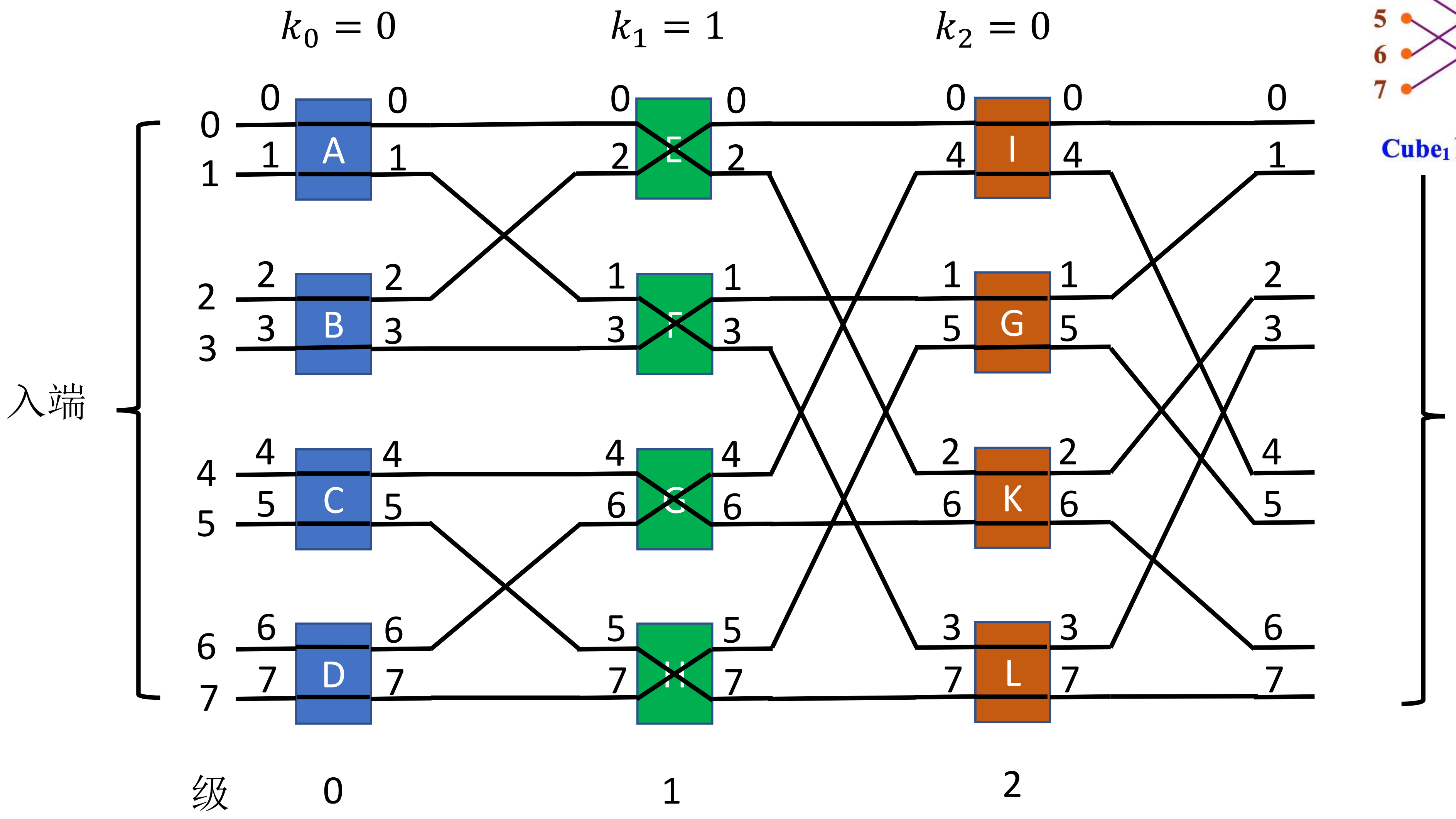
1 多级立方体网络



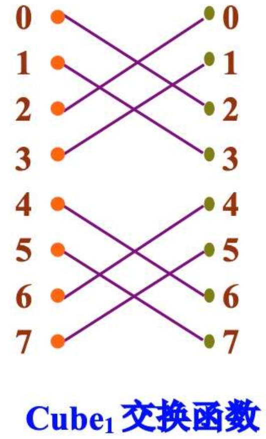




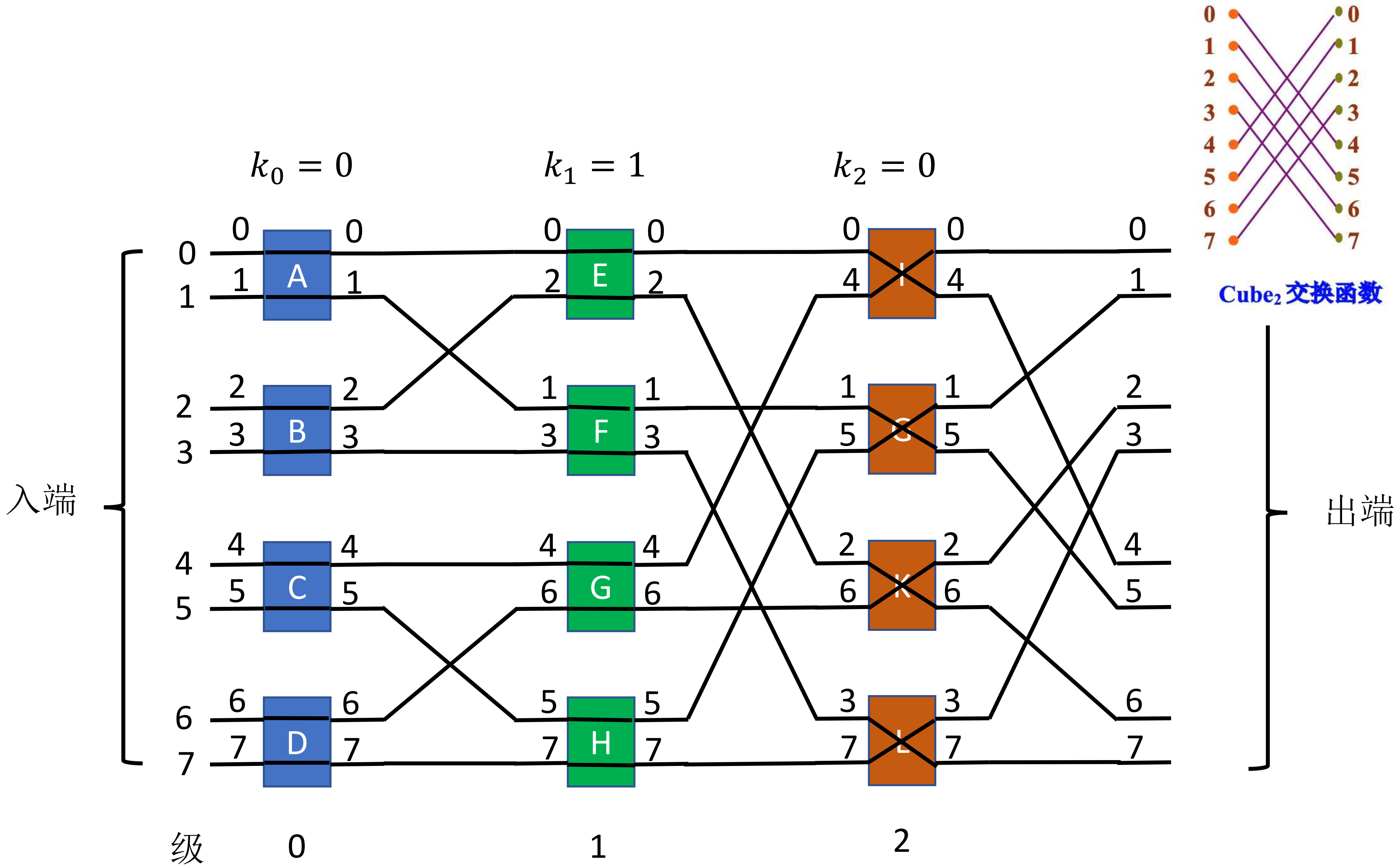
Cube₀交换函数

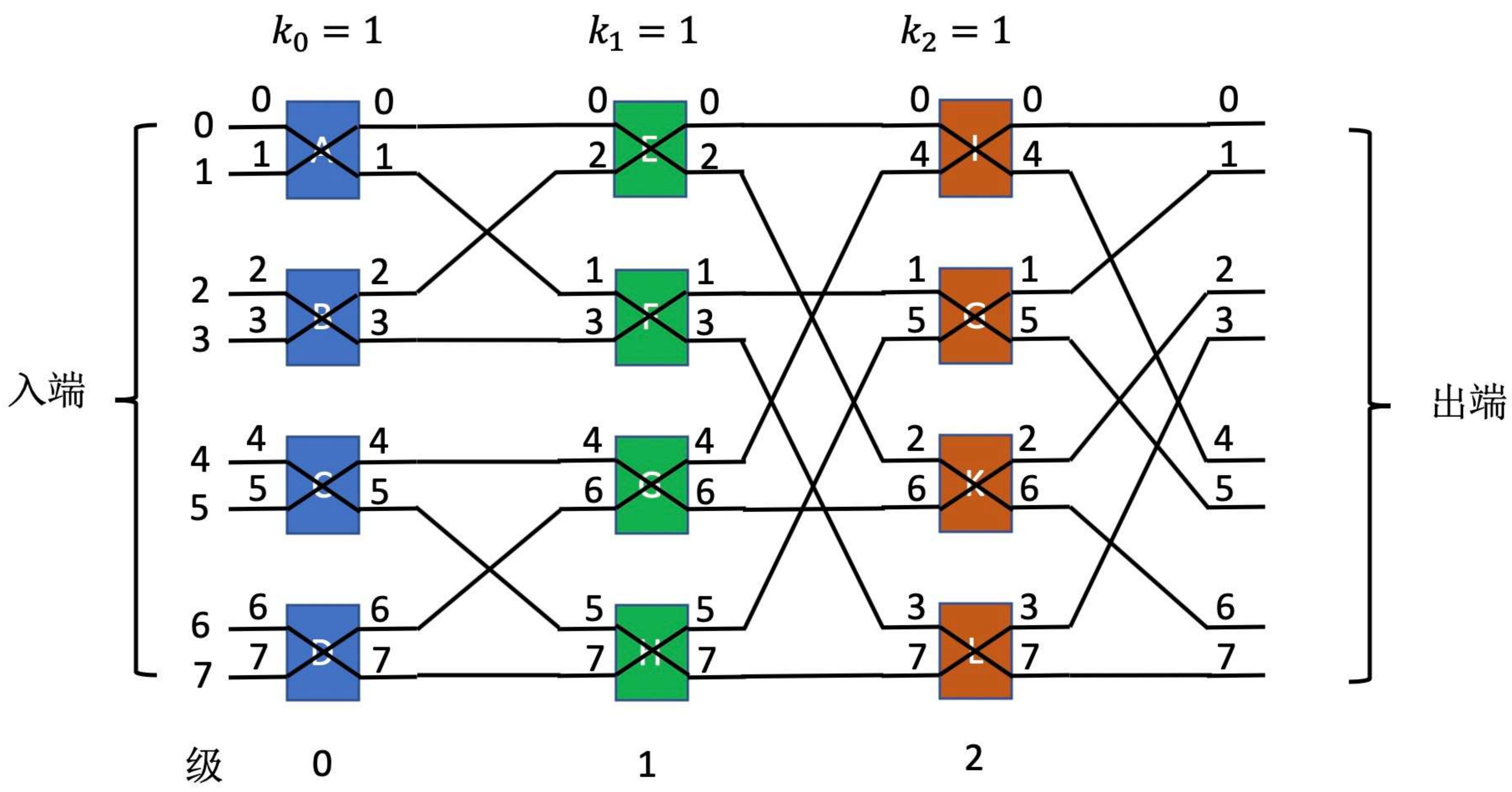


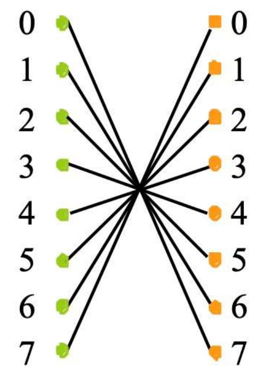
$k_0=0$
$k_1=1$
$k_2=0$
A
B
C
D
E
F
G
H
I
J
K
L
入端
出端
级
0
1
2



Cube1 交换函数


$k_0 = 0$
$k_1 = 1$
$k_2 = 0$
入端
出端
A
B
C
D
E
F
G
H
I
J
K
L
级
0
1
2
Cube₂交换函数


$k_0 = 1$
$k_1 = 1$
$k_2 = 1$
入端
出端
级
0
1
2



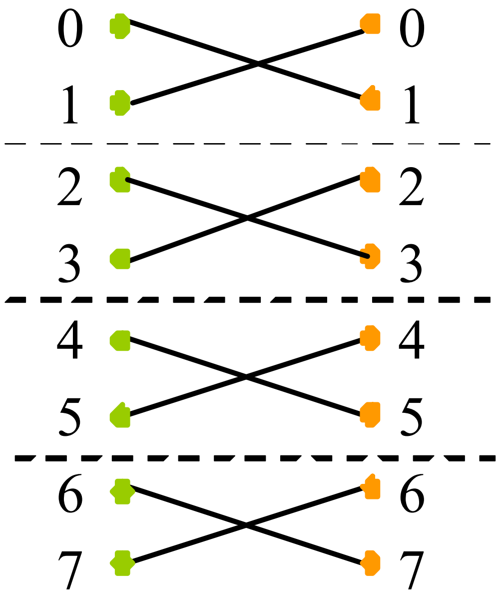
7.4.3 多级互连网络

1 多级立方体网络

- STARAN网络采用<u>级控制</u>和部分级控制。
  - 采用级控制时，所实现的是交换功能；
  - 采用部分级控制时，则能实现移数功能。
- 间接二进制n方体网络则采用单元控制。
  - 具有更大的灵活性。
- 交换
  - 将有序的一组元素头尾对称地进行交换。

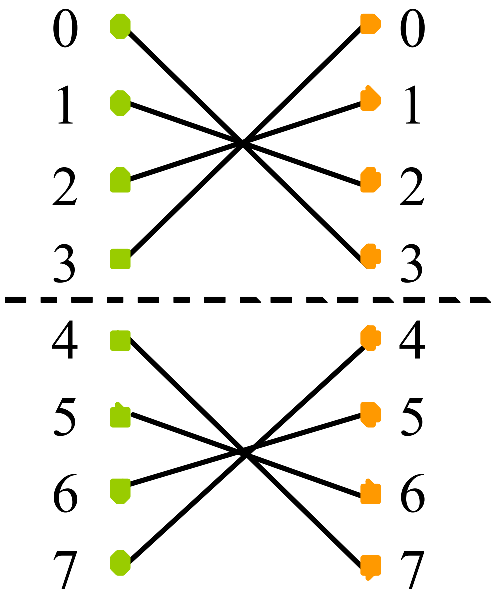
7.4.3 多级互连网络

1 多级立方体网络



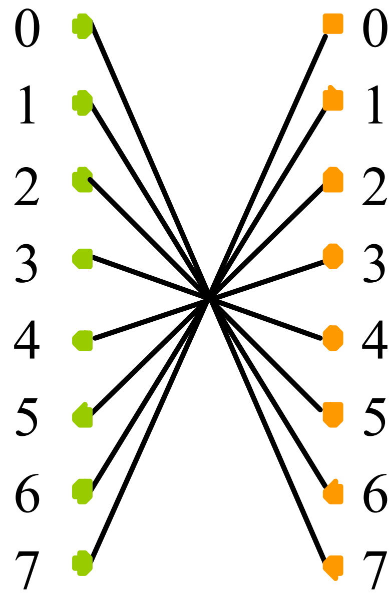
(a) 4 组 2 元交换

第一级交换



(b) 2 组 4 元交换

第一、二级交换



(c) 1 组 8 元交换

第一、二、三级交换

8个元素的基本交换图形

7.4.3 多级互连网络

1 多级立方体网络

- 3级STARAN网络在各种级控制信号的情况下所实现的入出端连接以及所实现的交换函数和功能。
  - $k_2k_1k_0$：控制信号，k_i（i=0，1，2）为第i级的级控制信号。
  - 从表中可以看出：下面的4行中每一行所实现的功能可以从级控制信号为其反码的一行中所实现的功能加上1组8元变换来获得。
  - 例如：级控制信号为110所实现的功能是其反码001所实现的4组2元交换再加上1组8元交换来获得。

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	$Cube_0$
001 ^ 011 010	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	$Cube_0$＋$Cube_1$
011 ^ 111 100	4 5 6 7 0 1 2 3	2组4元交换＋ 1组8元交换	$Cube_2$
101	001 ^ 011 ^ 111 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	$Cube_0$＋$Cube_2$
001 ^ 111 110	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	$Cube_0$＋$Cube_1$＋ $Cube_2$

➢ 当STARAN网络用作移数网络时，采用<u>部分级控制</u>

控制信号的分组和控制结果。

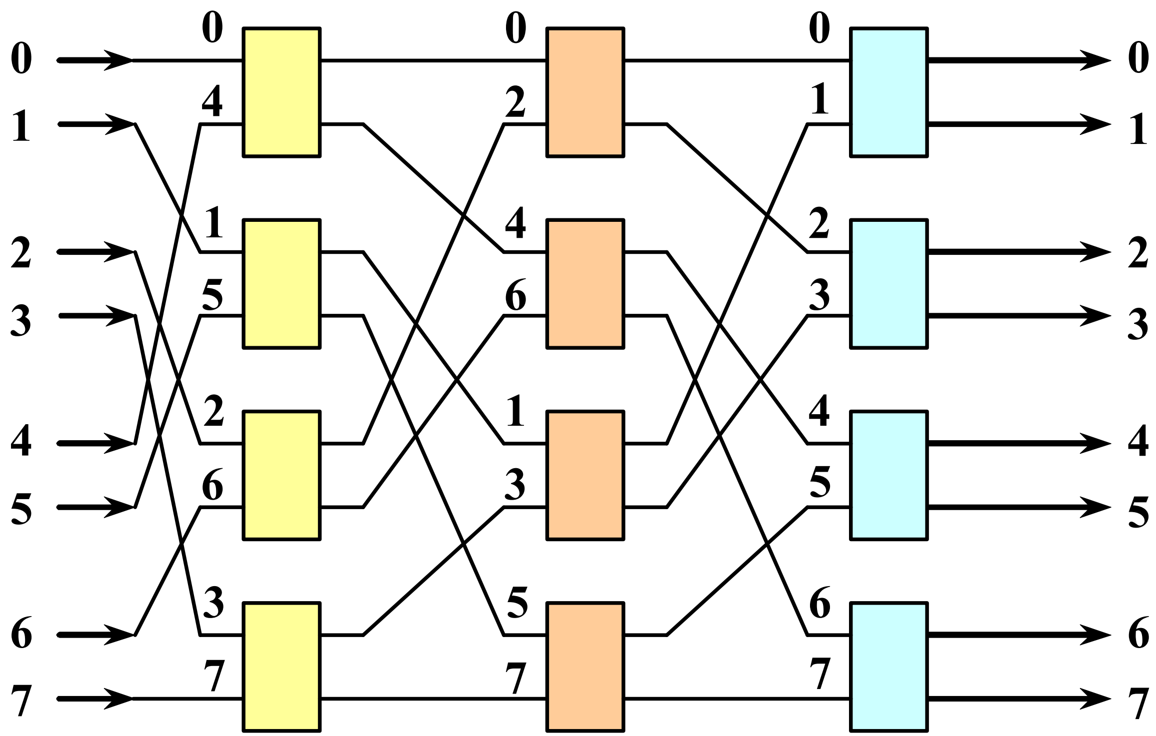
部分级控制信号						连接的输出端号序列（入端号序列：01234567）	所实现的移数功能
第0级	第1级		第2级				
A B C D	E G	F H	I	J	K L		
1	1	0	1	0	0	1 2 3 4 5 6 7 0	移1 mod 8
0	1	1	1	1	0	2 3 4 5 6 7 0 1	移2 mod 8
0	0	0	1	1	1	4 5 6 7 0 1 2 3	移4 mod 8
1	1	0	0	0	0	1 2 3 0 5 6 7 4	移1 mod 4
0	1	1	0	0	0	2 3 0 1 6 7 4 5	移2 mod 4
1	0	0	0	0	0	1 0 3 2 5 4 7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

<u>部分级控制</u>：第i级的所有开关分别用$i+1$个信号控制，$0 \le i \le n-1$，n为级数。

7.4.3 多级互连网络

2 Omega网络（多级混洗交换网络）

- 一个8×8的Omega网络
  - 每级由4个4功能的2×2开关构成
  - 级间互连采用均匀洗牌连接方式



7.4.3 多级互连网络

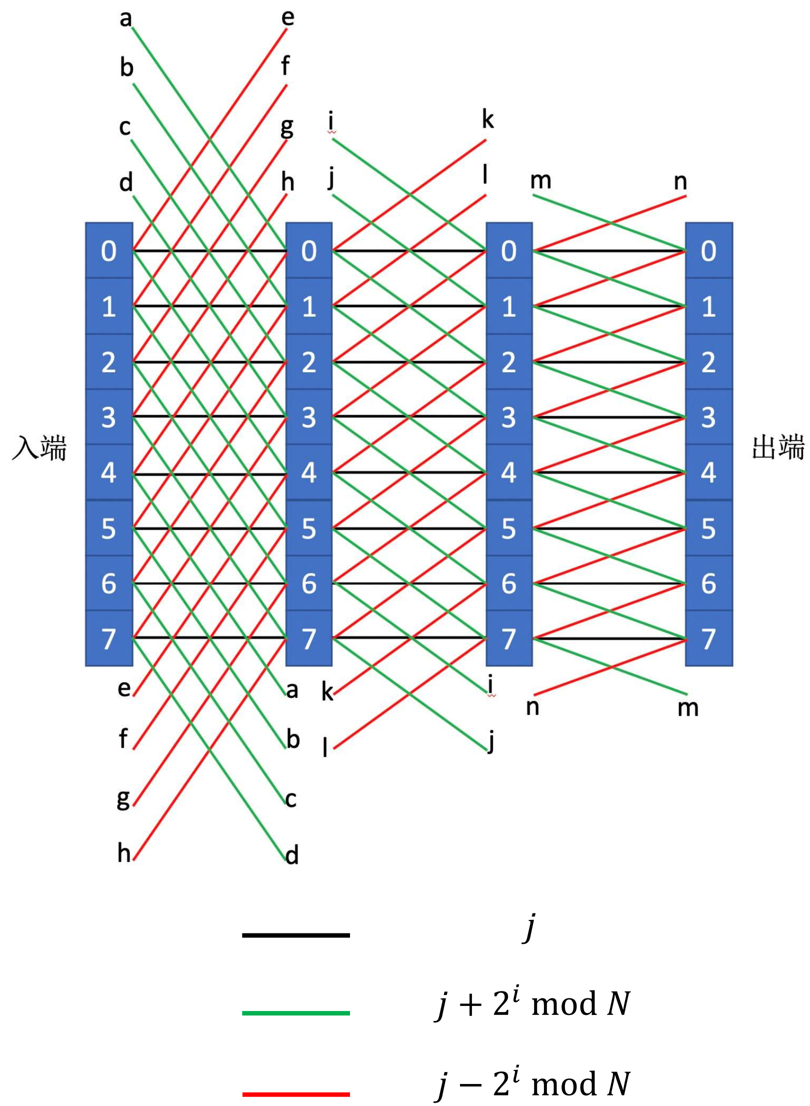
2 Omega网络（多级混洗交换网络）

- 一个N输入的Omega网络
  - 有$\log_2 N$级，每级用N/2个2×2开关模块，共需要$N\log_2 N/2$个开关。
  - 每个开关模块均采用单元控制方式。
  - 直连、交换、上播、下播
  - 不同的开关状态组合可实现各种置换、广播或从输入到输出的其他连接。

7.4.3 多级互连网络

3 多级PM2I网络

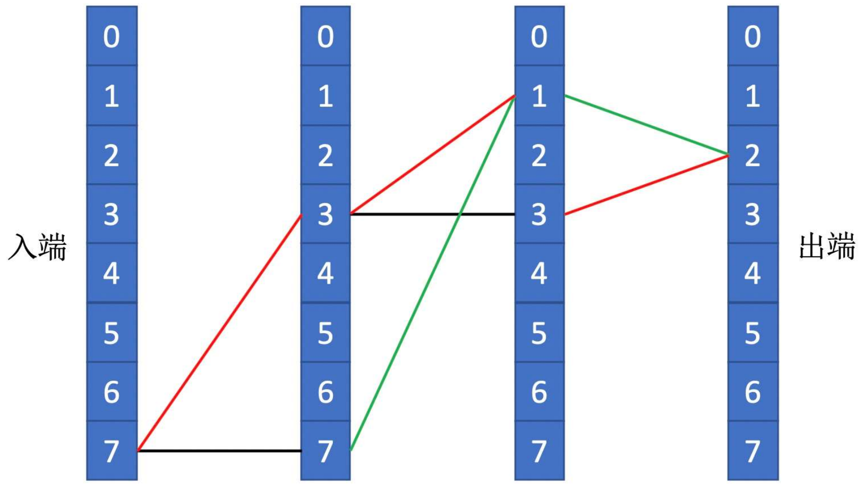
- 包含n级单元连接，每一级都把前后两列各$N=2^n$个单元按PM2I拓扑相互连接
- 第i级（$0 \le i \le n-1$）中的每一个入单元j都有三根连接线分别通往出单元j、$j+2^i \bmod N$、和$j-2^i \bmod N$



7.4.3 多级互连网络

3 多级PM2I网络

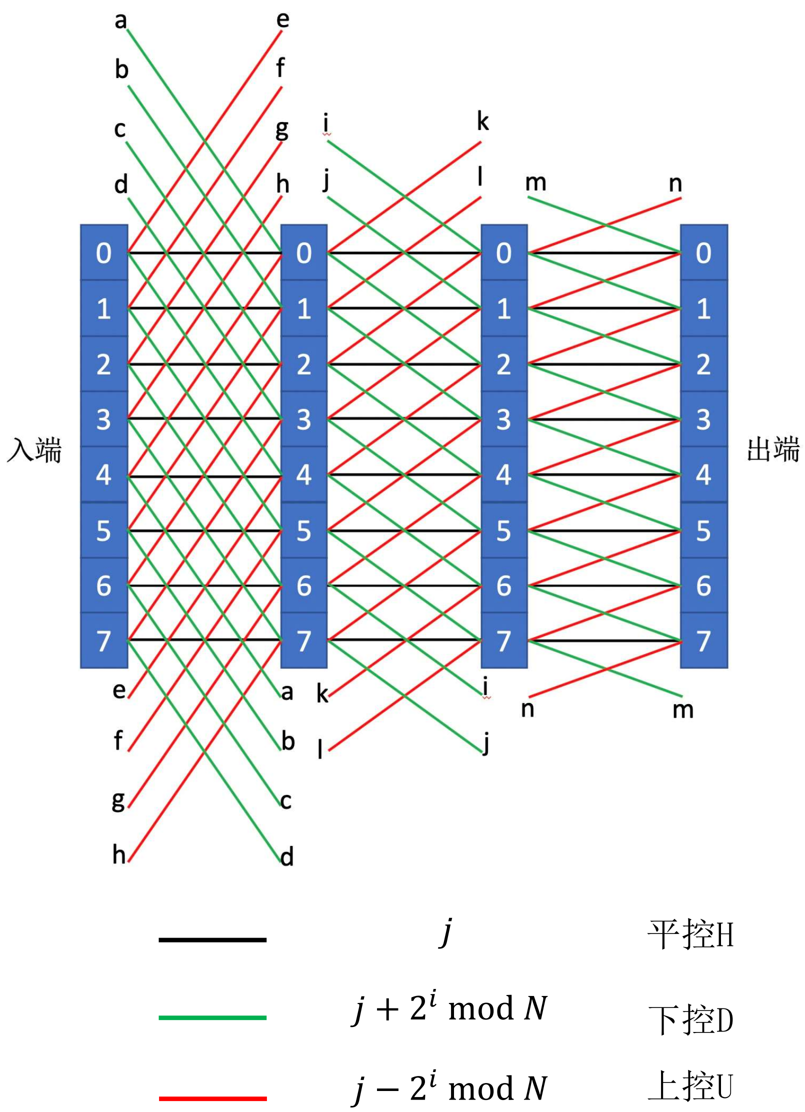
- 包含n级单元连接，每一级都把前后两列各$N=2^n$个单元按PM2I拓扑相互连接
- 第i级（$0 \le i \le n-1$）中的每一个入单元j都有三根连接线分别通往出单元j、$j+2^i \bmod N$、和$j-2^i \bmod N$
- 可以提供冗余路径，便于电路集成，提高可靠性



7.4.3 多级互连网络

3 多级PM2I网络

- 包含n级单元连接，每一级都把前后两列各$N=2^n$个单元按PM2I拓扑相互连接
- 第i级（ $0 \le i \le n-1$ ）中的每一个入单元j都有三根连接线分别通往出单元j、$j+2^i \bmod N$、和$j-2^i \bmod N$
- 可以提供冗余路径，便于电路集成，提高可靠性
- 控制三种连接线的信号分别称为<u>平控H</u>、<u>下控D</u>和<u>上控U</u>
  - <u>数据变换网络</u>（Data Manipulator, DM）:对于第i级，让H_1^i, D_1^i, U_1^i控制第i位为0的那些入单元，而让H_2^i, D_2^i, U_2^i控制第i位为1的那些入单元
  - <u>强化数据变换网络</u>（Augmented Data Manipulator, ADM）：采用单元控制增强对各级单元控制的灵活性，让每一个单元都有自己独立的控制信号，但是由于控制线较多，成本较高



7.4.3 多级互连网络

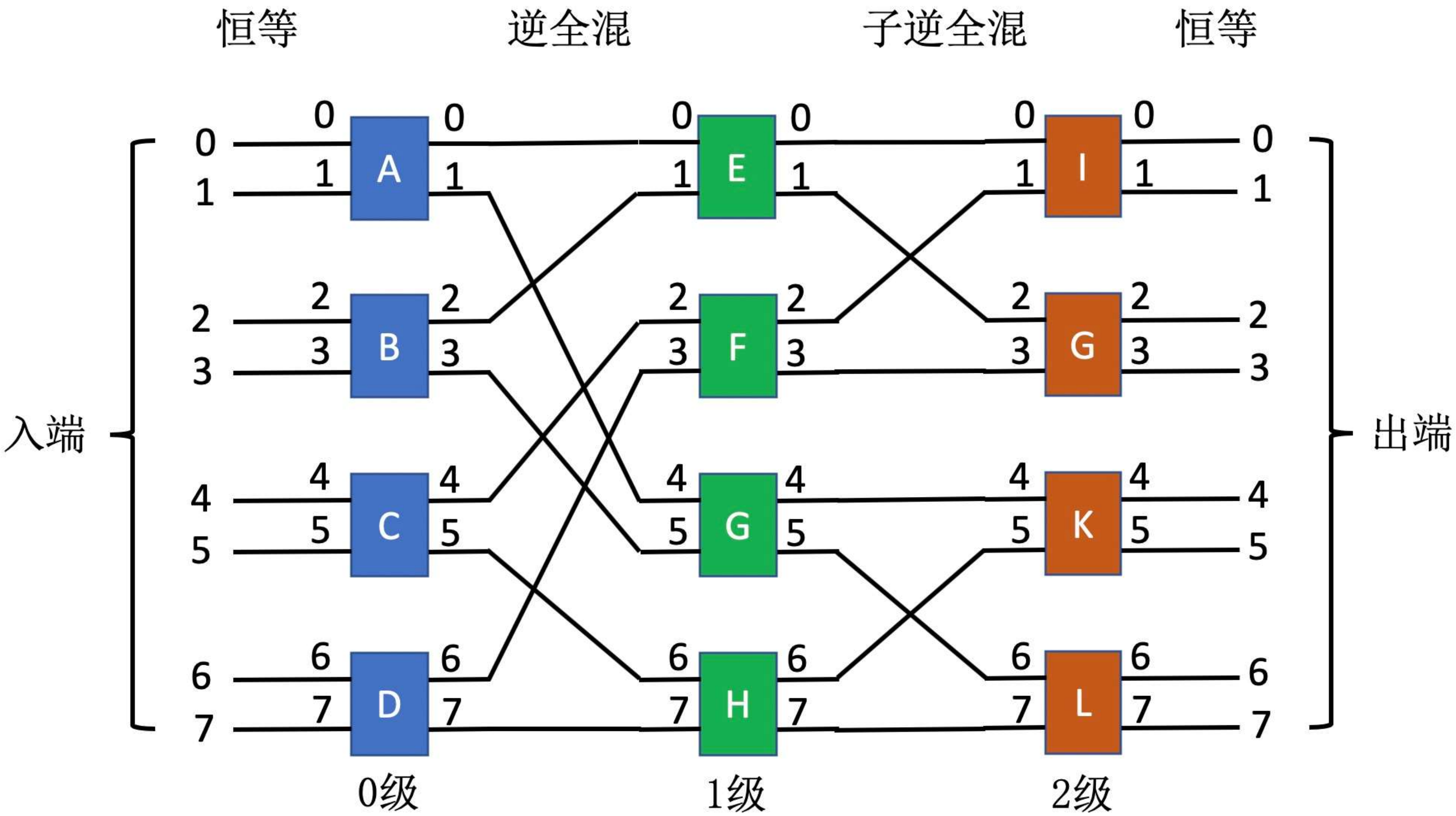
比较

- 灵活性由低到高：
  - 级控制立方体<部分级控制立方体<间接二进制立方体<Omega<ADM
- 复杂性和成本由低到高：
  - 级控制立方体<部分级控制立方体<间接二进制立方体<Omega<ADM
- 不同用途
  - STARAN网络和Omega网络都是为了进行存储器和处理单元之间的数据变换
  - 间接二进制n方体网络是为了连接成微处理器阵列

7.4.3 多级互连网络

4 基准网络

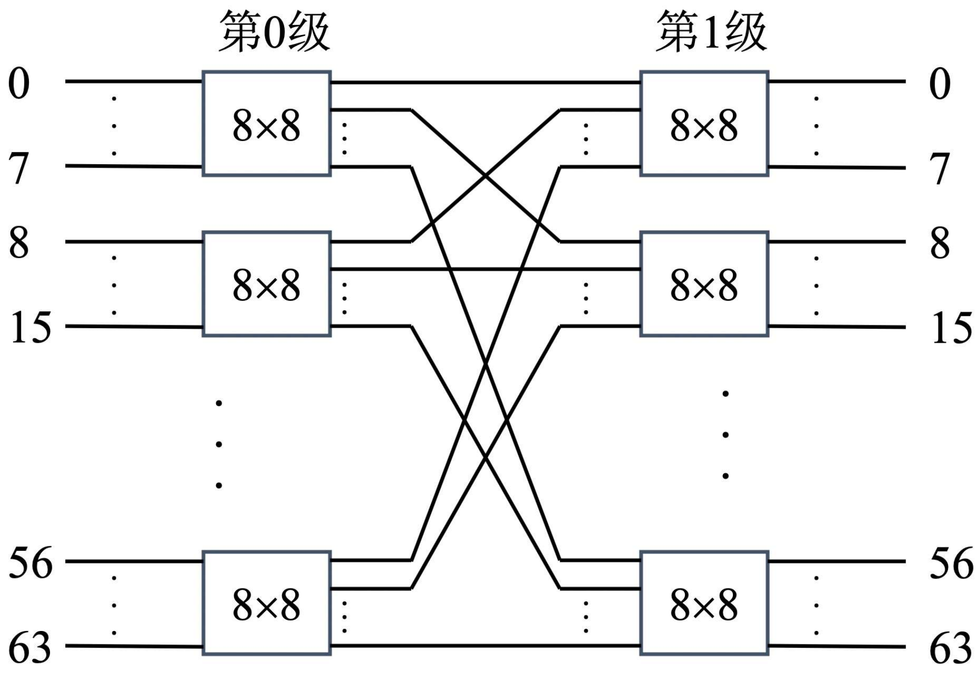
➢ 所用交换单元均为二功能，采取单元控制



7.4.3 多级互连网络

6 多级蝶式网络

- 蝶式网络的开关不允许广播功能，它实际上是Omega网的一个子集
- 两级64×64的蝶式网络如下图所示：它采用16个8×8交叉开关构成，两级间采用8路洗牌连接。



7.4.3 多级互连网络

7 全排列网络

- 非阻塞型网络：所有入端、出端的连接均不发生冲突的网络，又称非阻塞型网络(全排列网络)
  - $N_{入} \rightarrow N_{出}$有N!种排列
  - 灵活性好，连线多，控制复杂，成本高
- 阻塞型网络（Blocking Network）：同时实现两对或多对入端与出端之间连接时，都有可能因争用数据传送路径而发生冲突。
  - STARAN等网络属于阻塞型网络

7.4.4 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
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