

计算机系统结构

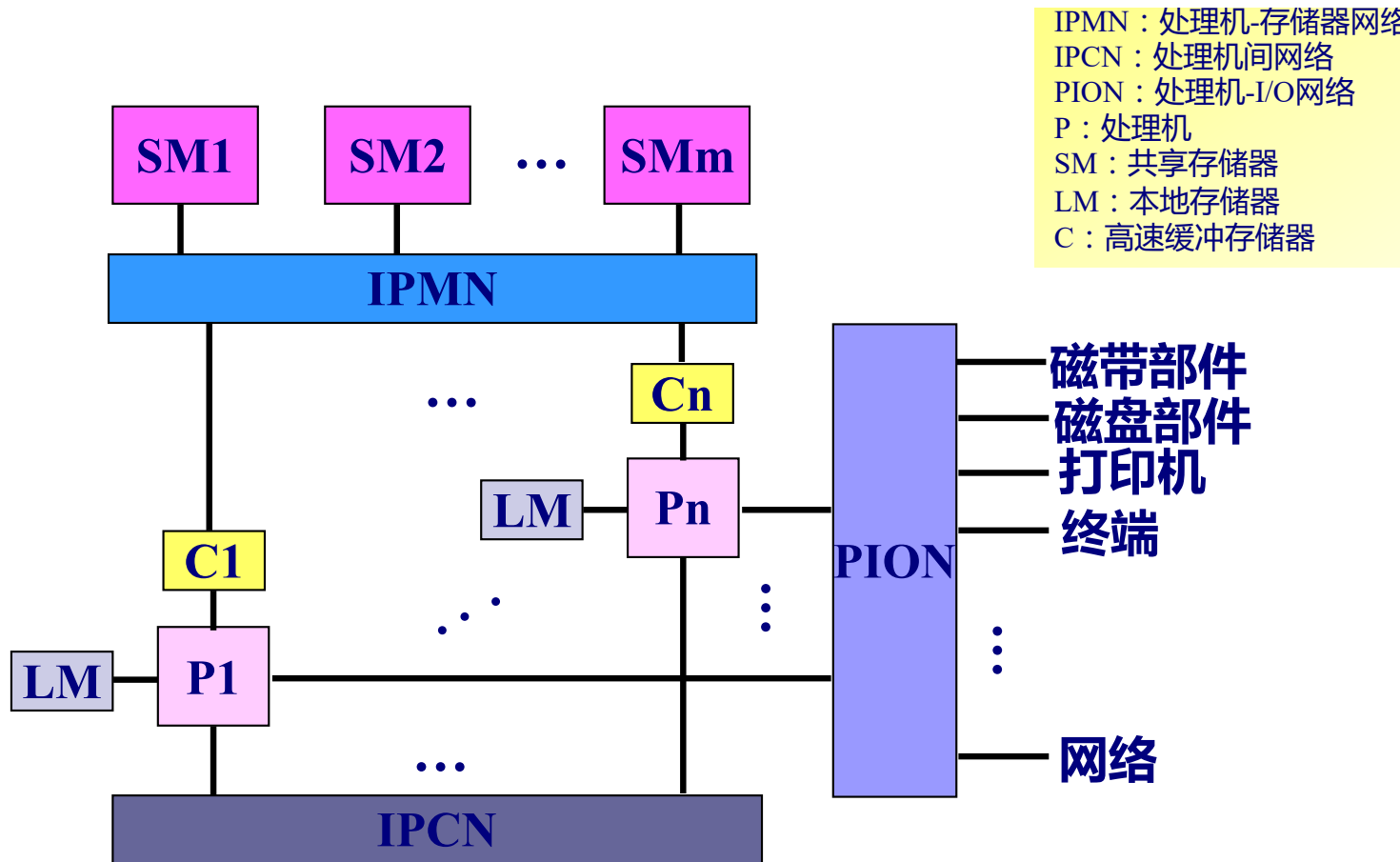
第9章 互连网络

2023年春季学期

互连网络：开关元件按一定拓扑结构和控制方式构成的网络以实现计算机系统内部多个处理机或多个功能部件间的相互连接

- 在拓扑上，互连网络为输入结点到输出结点之间的一组互连或映象。
- SIMD计算机和MIMD计算机的关键组成部分。
- 三大要素：互连结构、开关元件、控制方式。

一般多处理机系统的互连结构



- 9. 1 [互连函数](#)
- 9. 2 [互连网络的结构参数与性能指标](#)
- 9. 3 [静态互连网络](#)
- 9. 4 [动态互连网络](#)
- 9. 5 [消息传递机制](#)

9.1 互连函数

9.1.1 互连函数

第 6 页

- 变量 x : 输入 (设 $x=0, 1, \dots, N-1$) \rightarrow 输出 : 函数 $f(x)$
- 互连函数通过数学表达式建立输入端号与输出端号的连接关系, 即在互连函数 f 的作用下, 输入端 x 连接到输出端 $f(x)$.
 - 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
 - 互连函数 $f(x)$ 有时可以采用循环表示, 即: $(x_0 \ x_1 \ x_2 \ \dots \ x_{j-1})$, j 称为该循环的长度, 则表示: $f(x_0)=x_1, f(x_1)=x_2, \dots, f(x_{j-1})=x_0$
 - 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

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

9.1.2 几种基本的互连函数

第 7 页

介绍几种常用的基本互连函数及其主要特征。

1. 恒等函数

- 恒等函数：实现同号输入端和输出端之间的连接。

$$I(X_{n-1}X_{n-2}\cdots X_1X_0)=X_{n-1}X_{n-2}\cdots X_1X_0$$

2. 交换函数

- 交换函数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。

$$E(x_{n-1}x_{n-2}\cdots x_{k+1}x_kx_{k-1}\cdots x_1x_0)=x_{n-1}x_{n-2}\cdots x_{k+1}\bar{x}_kx_{k-1}\cdots x_1x_0$$

- 主要用于构造立方体互连网络和各种超立方体互连网络。
- 它共有 $n = \log_2 N$ 种互连函数，N为结点个数

9.1.2 几种基本的互连函数

第 8 页

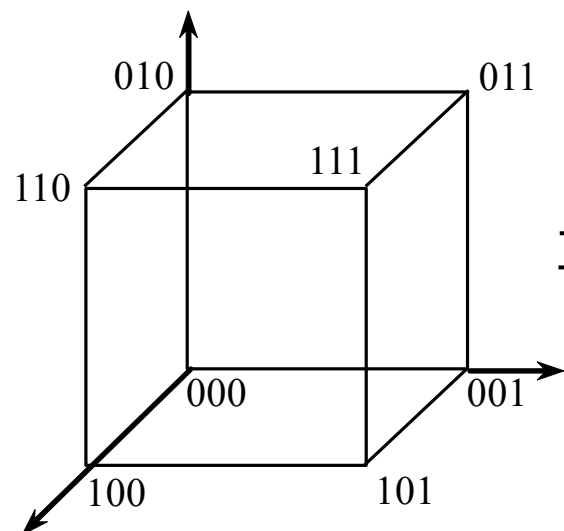
2. 交换(方体)函数

■ 当 $N = 8$ 时， $n = 3$ ，可得到常用的立方体互连函数：

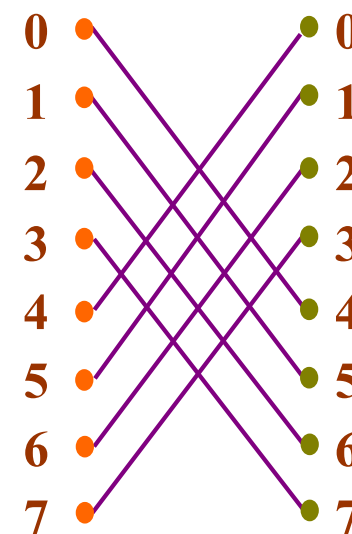
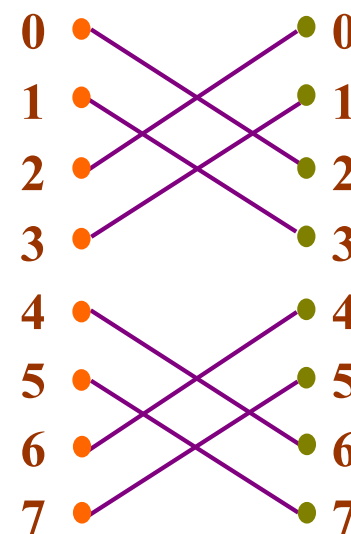
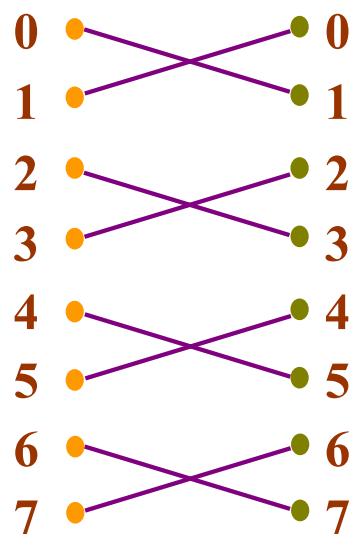
$$Cube_0(x_2x_1x_0) = x_2x_1\bar{x}_0$$

$$Cube_1(x_2x_1x_0) = x_2\bar{x}_1x_0$$

$$Cube_2(x_2x_1x_0) = \bar{x}_2x_1x_0$$



立方体网络



(a) $Cube_0$ 交换函数 (b) $Cube_1$ 交换函数 (c) $Cube_2$ 交换函数

$N=8$ 的立方体交换函数的变换图形

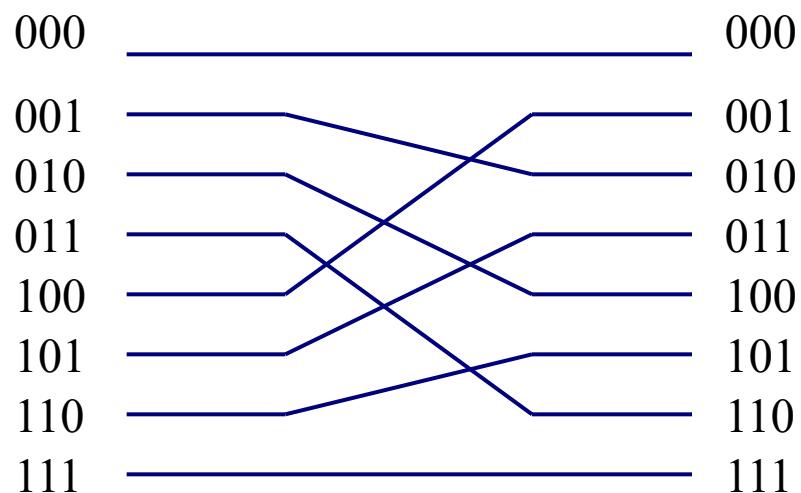
9.1.2 几种基本的互连函数

第 9 页

3. 均匀洗牌函数

- 均匀洗牌函数：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（置换），函数关系如下：

$$\sigma(x_{n-1}x_{n-2} \cdots x_1x_0) = x_{n-2}x_{n-3} \cdots x_1x_0x_{n-1} \quad \text{即把输入端的二进制编号循环左移一位}$$



9.1.2 几种基本的互连函数

第 10 页

例如：对于均匀洗牌函数

第k个子函数： $\sigma_{(k)}(x_{n-1} \dots x_k \mid x_{k-1} x_{k-2} \dots x_0) = x_{n-1} \dots x_k \mid x_{k-2} \dots x_0 x_{k-1}$

即把输入端的二进制编号中的低k位循环左移一位。

第k个超函数： $\sigma^{(k)}(x_{n-1} x_{n-2} \dots x_{n-k} \mid x_{n-k-1} \dots x_1 x_0) = x_{n-2} \dots x_{n-k} x_{n-1} \mid x_{n-k-1} \dots x_1 x_0$

即把输入端的二进制编号中的高k位循环左移一位。

下列等式成立：

$$\sigma^{(n)}(X) = \sigma_{(n)}(X) = \sigma(X)$$

$$\sigma^{(1)}(X) = \sigma_{(1)}(X) = X$$

对于任意一种函数 $f(x)$ ，如果存在 $g(x)$ ，使得 $f(x) \times g(x) = I(x)$ ，则称 $g(x)$ 是 $f(x)$ 的逆函数，记为 $f^{-1}(x) = g(x)$ 。

9.1.2 几种基本的互连函数

第 11 页

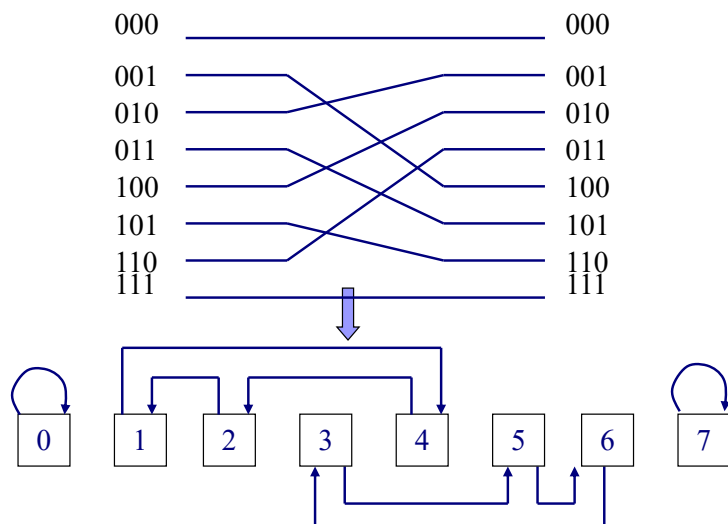
逆均匀洗牌函数：将输入端的二进制编号循环右移一位而得到所连接的输出端编号。

■ 互连函数

$$\sigma^{-1}(x_{n-1}x_{n-2} \cdots x_1x_0) = x_0x_{n-1}x_{n-2} \cdots x_1$$

■ 逆均匀洗牌是均匀洗牌的逆函数

■ 当N=8时，有：



$$\sigma(x_2x_1x_0) = x_1x_0x_2$$

$$\sigma_{(2)}(x_2x_1x_0) = x_2x_0x_1$$

$$\sigma^{(2)}(x_2x_1x_0) = x_1x_2x_0$$

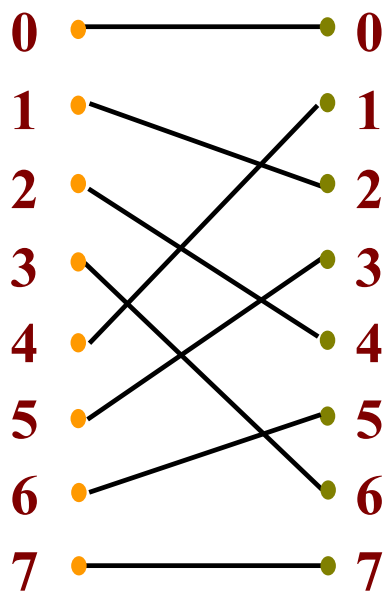
$$\sigma^{-1}(x_2x_1x_0) = x_0x_2x_1$$

9.1.2 几种基本的互连函数

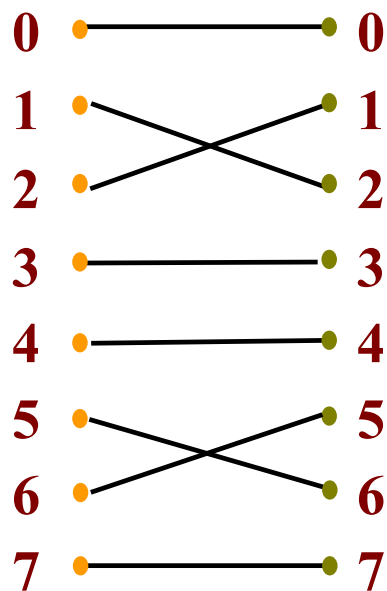
第 12 页

■ N=8 的均匀洗牌和逆均匀洗牌函数

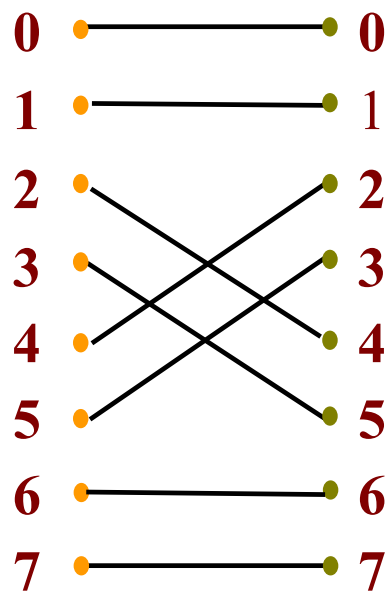
$$\sigma(x_2x_1x_0) = x_1x_0x_2$$



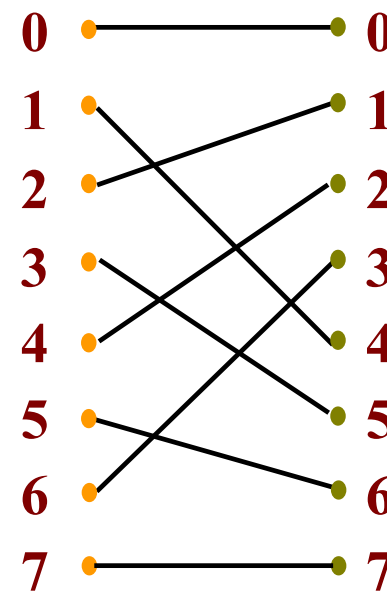
$$\sigma_{(2)}(x_2x_1x_0) = x_2x_0x_1$$



$$\sigma^{(2)}(x_2x_1x_0) = x_1x_2x_0$$



$$\sigma^{-1}(x_2x_1x_0) = x_0x_2x_1$$



(a) 均匀洗牌函数 σ

(b) 子洗牌函数 $\sigma_{(2)}$

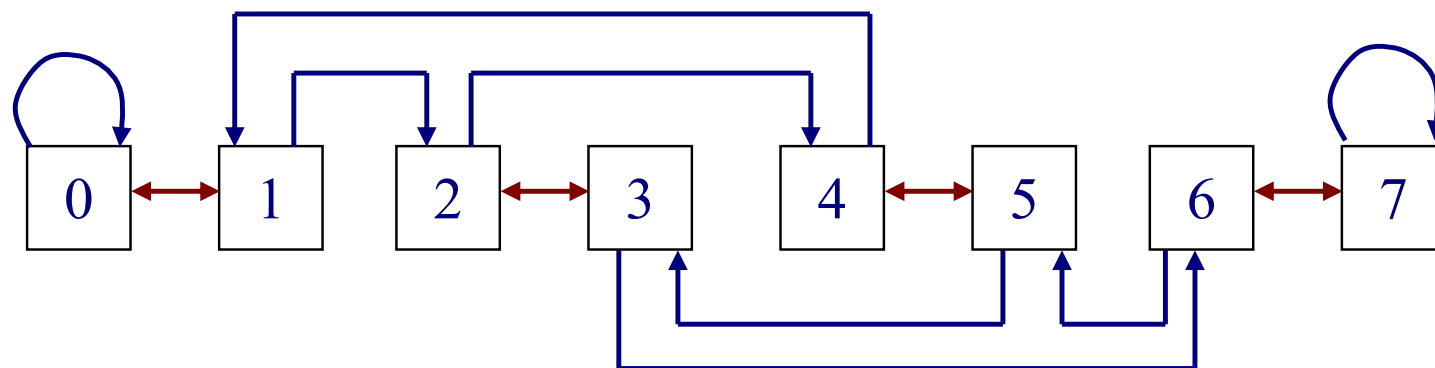
(c) 超洗牌函数 $\sigma^{(2)}$

(d) 逆均匀洗牌函数 σ^{-1}

均匀洗牌函数和逆均匀洗牌函数的连接图互为镜像，与交换开关多级组合可构成Omega网络和逆Omega网络。

混洗交换函数(Shuffle-Exchange)

均匀洗牌函数与 Cube_0 函数的组合



: 洗牌



: Cube_0

9.1.2 几种基本的互连函数

第 14 页

4. 蝶式函数

- **蝶式互连函数**：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。

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

- **第k个子函数**

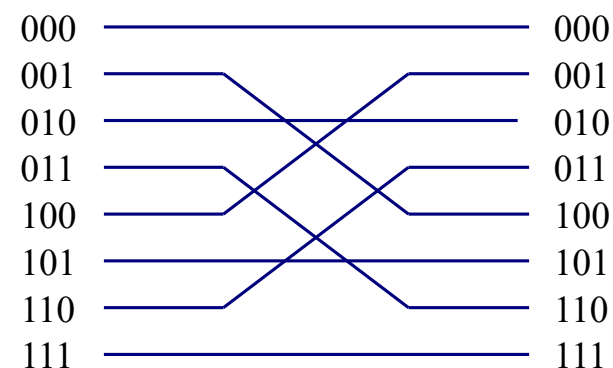
$$\beta_{(k)}(x_{n-1} \cdots x_k x_{k-1} x_{k-2} \cdots x_1 x_0) = x_{n-1} \cdots x_k x_0 x_{k-2} \cdots x_1 x_{k-1}$$

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。

- **第k个超函数**

$$\beta^{(k)}(x_{n-1} x_{n-2} \cdots x_{n-k+1} x_{n-k} x_{n-k-1} \cdots x_1 x_0) = x_{n-k} x_{n-2} \cdots x_{n-k+1} x_{n-1} x_{n-k-1} \cdots x_1 x_0$$

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。



9.1.2 几种基本的互连函数

第 15 页

- 下列等式成立

$$\beta^{(n)}(X) = \beta_{(n)}(X) = \beta(X)$$

$$\beta^{(1)}(X) = \beta_{(1)}(X) = X$$

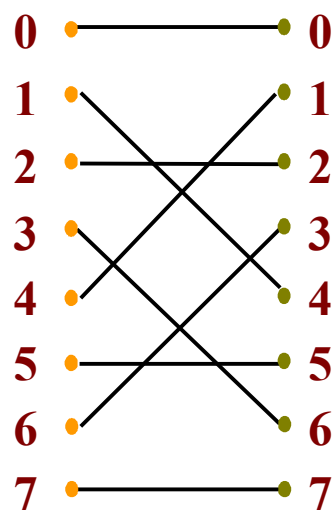
- 当 $N=8$ 时，有：

$$\beta(x_2x_1x_0) = x_0x_1x_2$$

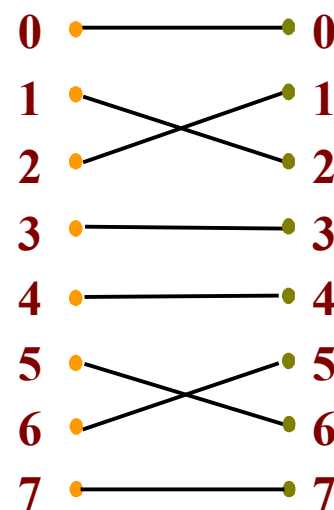
$$\beta_{(2)}(x_2x_1x_0) = x_2x_0x_1$$

$$\beta^{(2)}(x_2x_1x_0) = x_1x_2x_0$$

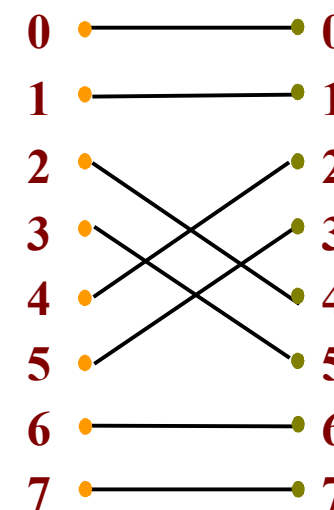
- 蝶式变换与交换变换的多级组合可作为构成方体多级网络的基础。



(a) $\beta = \rho$



(b) $\beta_{(2)} = \rho_{(2)}$



(c) $\beta^{(2)} = \rho^{(2)}$

$N=8$ 的蝶式函数和反位序函数

9.1.2 几种基本的互连函数

第 16 页

5. 反位序函数

- 反位序函数：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。

➤ 互连函数

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

- 第k个子函数

$$\rho_{(k)}(x_{n-1}\cdots x_k \mathbf{x_{k-1}x_{k-2}\cdots x_1x_0}) = x_{n-1}\cdots x_k \mathbf{x_0x_1\cdots x_{k-2}x_{k-1}}$$

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。

5. 反位序函数

- 第k个超函数

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

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。

- 下列等式成立

$$\rho^{(n)}(X) = \rho_{(n)}(X) = \rho(X)$$

$$\rho^{(1)}(X) = \rho_{(1)}(X) = X$$

- 当N=8时，有：

$$\rho(x_2x_1x_0) = x_0x_1x_2$$

$$\rho_{(2)}(x_2x_1x_0) = x_2x_0x_1$$

$$\rho^{(2)}(x_2x_1x_0) = x_1x_2x_0$$

9.1.2 几种基本的互连函数

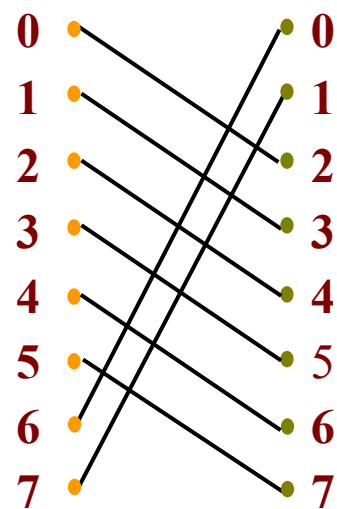
第 18 页

6. 移数函数

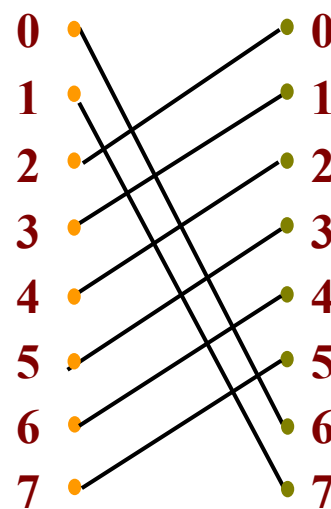
- 移数函数：将各输入端都错开一定的位置（模N）后连到输出端。

➤ 函数式

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



(a) 左移移数函数 $k=2$



(b) 右移移数函数 $k=2$

9.1.2 几种基本的互连函数

第 19 页

7. PM2I函数

- P和M分别表示加和减，2I表示 2^i 。
 - 该函数又称为“加減 2^i ”函数。
- PM2I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：

$$0 \leq x \leq N - 1, 0 \leq i \leq n - 1, n = \log_2 N, N \text{为结点数。}$$

- PM2I互连网络共有 $2n$ 个互连函数。

9.1.2 几种基本的互连函数

第 20 页

■ 当 $N = 8$ 时，有6个PM2I函数：

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

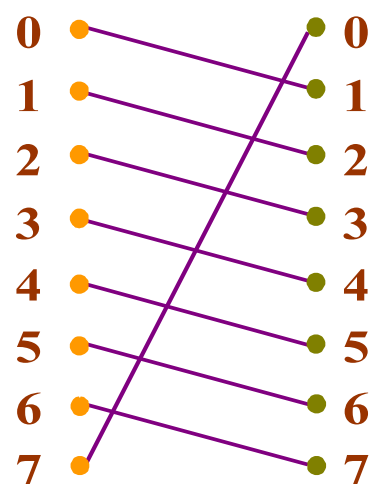
➤ $PM2_{-0} : (7\ 6\ 5\ 4\ 3\ 2\ 1\ 0)$

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

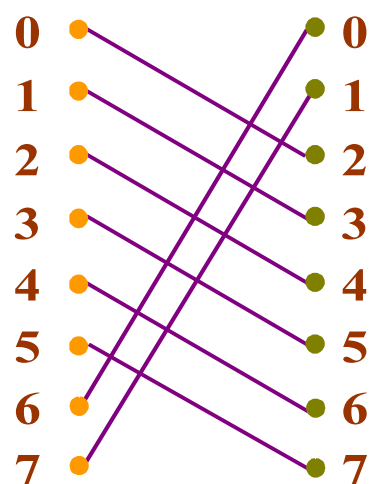
➤ $PM2_{-1} : (6\ 4\ 2\ 0)\ (7\ 5\ 3\ 1)$

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

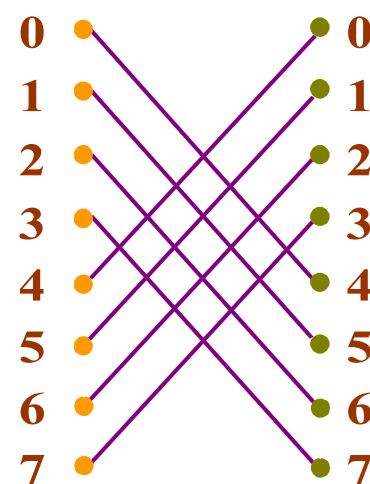
➤ $PM2_{-2} : (4\ 0)\ (5\ 1)\ (6\ 2)\ (7\ 3)$



(a) $PM2_{+0}$



(b) $PM2_{+1}$



(c) $PM2_{+2}$

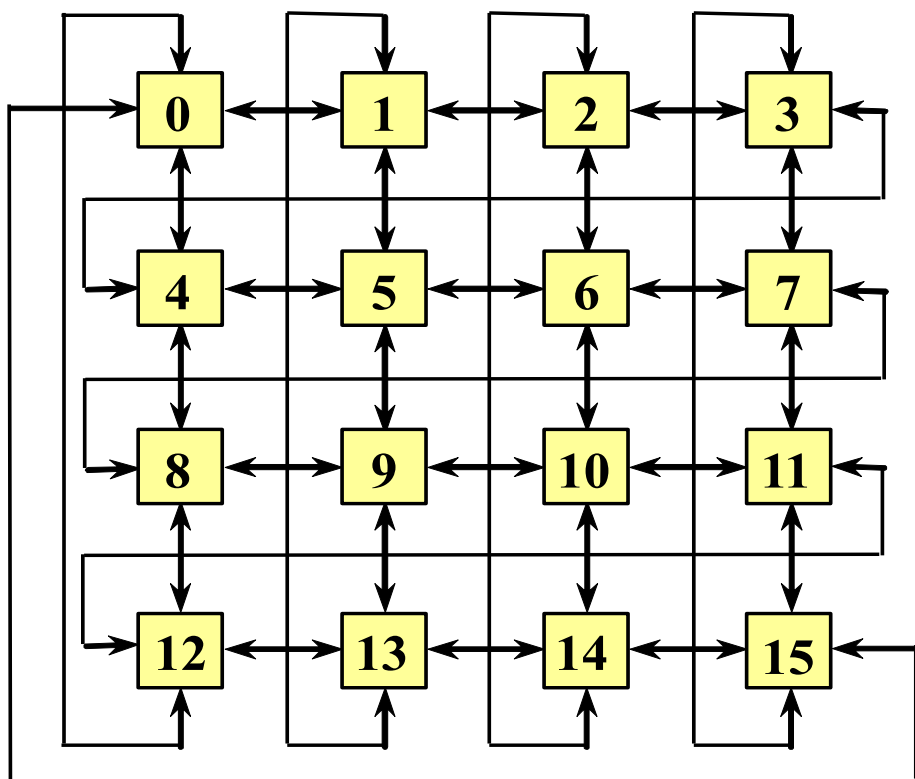
$N=8$ 的PM2I函数

9.1.2 几种基本的互连函数

第 21 页

■ 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。



用移数函数构成ILLIAC IV 阵列机的互连网络

9.1.2 几种基本的互连函数

第 22 页

例9.1 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) 洗牌函数 σ

(5) 复合函数 $\sigma(\sigma)$

时，分别给出与第13号处理器所连接的处理器号。

9.1.2 几种基本的互连函数

第 23 页

解：(1) 由 $Cube_3(x_3x_2x_1x_0) = \bar{x}_3x_2x_1x_0$,

得 $Cube_3(1101) = 0101$, 即处理器13连接到处理器5。

令 $Cube_3(x_3x_2x_1x_0) = 1101$, 得 $x_3x_2x_1x_0 = 0101$, 故与处理器13相连的是处理器5。

所以处理器13与处理器5双向互连。

(2) 由 $PM2_{+3} = j + 2^3 \bmod 16$, 得 $PM2_{+3}(13) = 13 + 2^3 = 5$, 即处理器13连接到处理器5。

令 $PM2_{+3}(j) = j + 2^3 \bmod 16 = 13$, 得 $j = 5$, 故与处理器13相连的是处理器5。

所以处理器13与处理器5双向互连。

(3) 由 $PM2_{-0}(j) = j - 2^0 \bmod 16$, 得 $PM2_{-0}(13) = 13 - 2^0 = 12$, 即处理器13连接到处理器12。

令 $PM2_{-0}(j) = j - 2^0 \bmod 16 = 13$, 得 $j = 14$, 故与处理器13相连的是处理器14。

所以处理器13连至处理器12 , 而处理器14连至处理器13。

9.1.2 几种基本的互连函数

第 24 页

(4) 由 $\sigma(x_3x_2x_1x_0) = x_2x_1x_0x_3$, 得 $\sigma(1101) = 1011$, 即处理器13连接到处理器11。

令 $\sigma(x_3x_2x_1x_0) = 1101$, 得 $x_3x_2x_1x_0 = 1110$, 故与处理器13相连的是处理器14。

所以处理器13连至处理器11 , 而处理器14连至处理器13。

(5) 由 $\sigma(\sigma(x_3x_2x_1x_0)) = x_1x_0x_3x_2$, 得 $\sigma(\sigma(1101)) = 0111$, 即处理器13连接到处理器7。

令 $\sigma(\sigma(x_3x_2x_1x_0)) = 1101$, 得 $x_3x_2x_1x_0 = 0111$, 故与处理器13相连的是处理器7。

所以处理器13与处理器7双向互连。

9.2 互连网络的结构参数与性能指标

9.2.1 互连网络的结构参数

第 26 页

1. 网络通常是用有向边或无向边连接有限个结点的图来表示。
2. 互连网络的主要特性参数有：
 - 网络规模 N ：网络中结点的个数。表示该网络所能连接的部件的数量。
 - 结点度 d ：与结点相连接的边数（通道数），包括入度和出度。
 - ✓ 进入结点的边数叫入度。
 - ✓ 从结点出来的边数叫出度。
 - 结点距离：对于网络中的任意两个结点，从一个结点出发到另一个结点终止所需要跨越的边数的最小值。
 - 网络直径 D ：网络中任意两个结点之间距离的最大值，网络直径应当尽可能地小。

9.2.1 互连网络的结构参数

第 27 页

- **等分宽度 b** ：把由 N 个结点构成的网络切成结点数相同（ $N/2$ ）的两半，在各种切法中，沿切口边数的最小值。

线等分宽度： $B = b \times w$

其中： w 为通道宽度（用位表示），该参数主要反映了网络最大流量。

- **对称性**：从任何结点看到的拓扑结构都是相同的网络称为对称网络。
对称网络比较容易实现，编程也比较容易。

9.2.2 互连网络的性能指标

第 28 页

评估互连网络性能的两个基本指标：时延和带宽

1. 通信时延

指从源结点到目的结点传送一条消息所需的总时间，它由以下4部分构成：

- **软件开销**：在源结点和目的结点用于收发消息的软件所需的执行时间。
 - ✓ 主要取决于两端端结点处理消息的软件内核。
- **通道时延**：通过通道传送消息所花的时间。
 - ✓ 通路时延 = 消息长度/通道带宽
 - ✓ 通常由瓶颈链路的通道带宽决定。
- **选路时延**：消息在传送路径上所需的一系列选路决策所需的时间开销
(与传送路径上的结点数成正比)
- **竞争时延**：多个消息同时在网络中传送时，会发生争用网络资源的冲突。为避免或解决争用冲突所需的时间就是竞争时延
(很难预测，它取决于网络的传输状态)

2. 网络时延

指通道时延与选路时延的和，它是由网络硬件特征决定的，与程序行为和网络传输状态无关。

3. 端口带宽

- 对于互连网络中的任意一个端口来说，其端口带宽是指单位时间内从该端口传送到其他端口的最大信息量。
 - 在对称网络中，端口带宽与端口位置无关。网络的端口带宽与各端口的端口带宽相同。
 - 非对称网络的端口带宽则是指所有端口带宽的最小值。

4. 聚集带宽

- 网络从一半结点到另一半结点，单位时间内能够传送的最大信息量。

例如，HPS是一种对称网络

网络规模N的上限：512

端口带宽：40MB/s

HPS的聚集带宽： $(40\text{MB/s} \times 512) / 2 = 10.24\text{GB/s}$

5. 等分带宽

与等分宽度对应的切平面中，所有边合起来单位时间所能传送的最大信息量

9.3 静态互连网络

9.3 静态互连网络

第 32 页

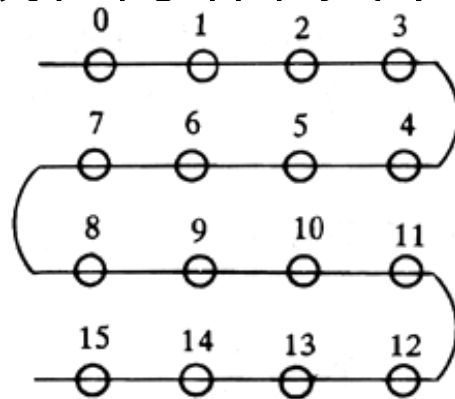
互连网络通常可以分为两大类：

- **静态互连网络**：各结点之间有固定的连接通路、且在运行中不能改变的网络。
- **动态互连网络**：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。

下面介绍几种静态互连网络（其中： N 表示结点的个数）

1. **线性阵列** 一种一维的线性网络，其中 N 个结点用 $N-1$ 个链路连成一行。

- 端结点的度：1
- 其余结点的度：2
- 直径： $N - 1$
- 等分宽度 $b=1$



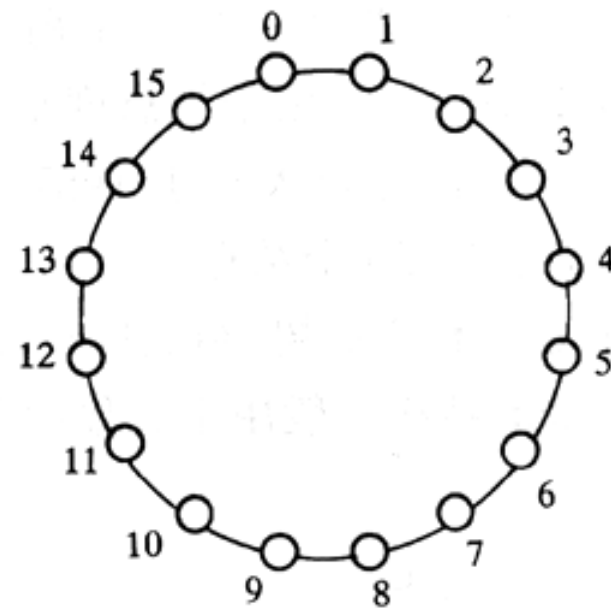
(a) 线性阵列

线性阵列与总线的区别：

- **总线**是通过切换与其连接的许多结点来实现**时分特性**的，独占
- **线性阵列**允许不同的源结点和目的结点对并行地使用其不同的部分

2. 环和带弦环

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

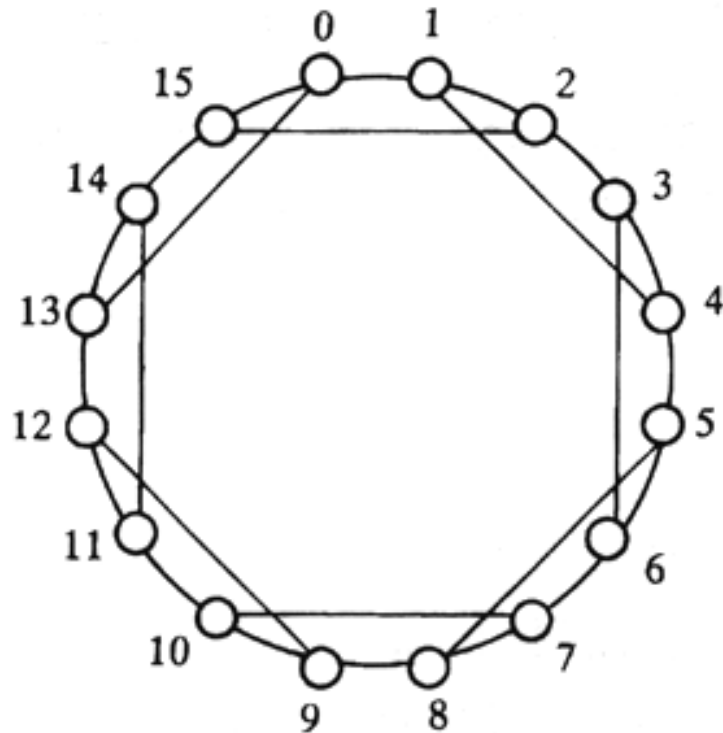


(b)环

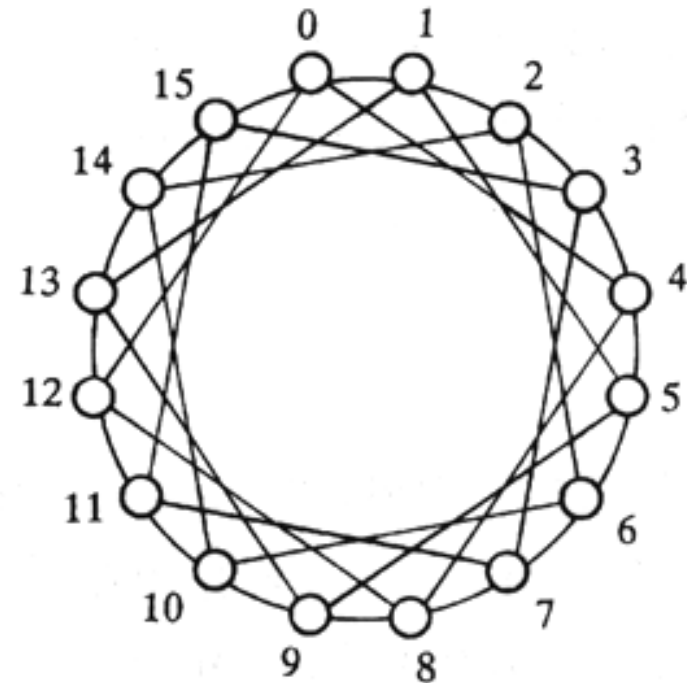
9.3 静态互连网络

第 34 页

- 带弦环：增加的链路愈多，结点度愈高，网络直径就愈小。



(c)度为 3 的带弦环



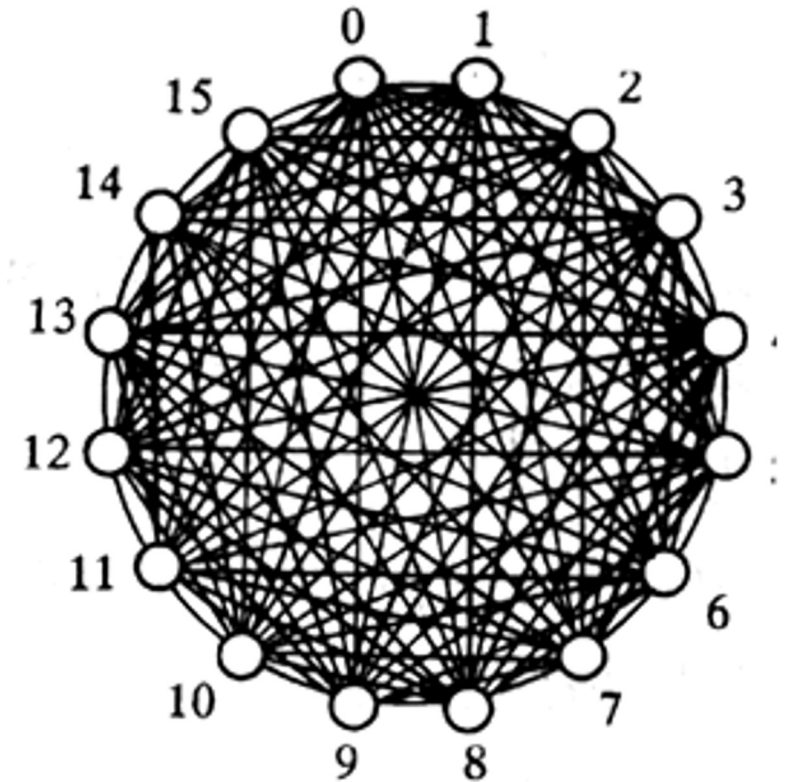
(d)度为 4 的带弦环(与 Illiac 网相同)

9.3 静态互连网络

第 35 页

■ 全连接网络

- 结点度：15
- 直径最短，为1。



(f) 全连接

3. 循环移数网络

- 通过在环上每个结点到所有与其距离为2的整数幂的结点之间都增加一条附

$N=16$

- 结点度：7

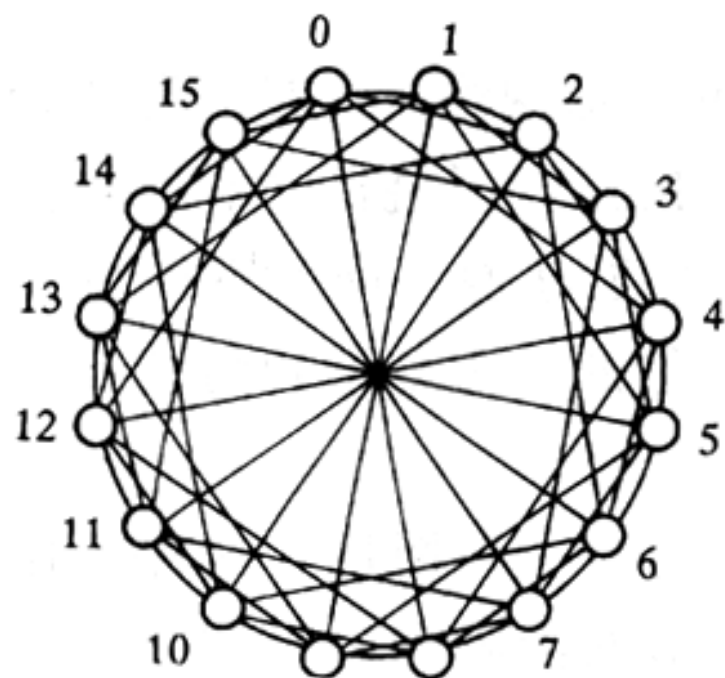
- 直径：2

一般地，如果 $|j-i| = 2^r$ ($r=0, 1, 2, \dots, n-1$, $n=\log_2 N$)，则结点*i*与结点*j*连接。

- 结点度： $2n - 1$

- 直径： $n/2$

- 网络规模 $N=2^n$



(e) 循环移数网络

4. 树形和星形

■ 一棵5层31个结点的二叉树

一般说来，一棵k层完全平衡的二叉树有 $N=2^k-1$ 个结点。

➤ 最大结点度：3

➤ 直径： $2(k-1)$

➤ 等分宽度 $b=1$

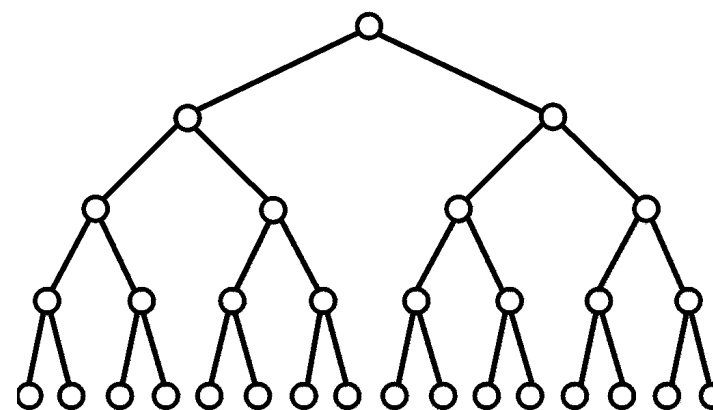
■ 星形

➤ 结点度较高，为 $N-1$ 。

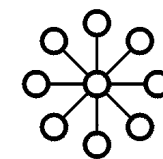
➤ 直径较小，是一常数2

➤ 等分宽度 $b=\lfloor N/2 \rfloor$

➤ 可靠性比较差，只要中心结点出故障，整个系统就会瘫痪。



(a) 二叉树



(b) 星形

5. 胖树形

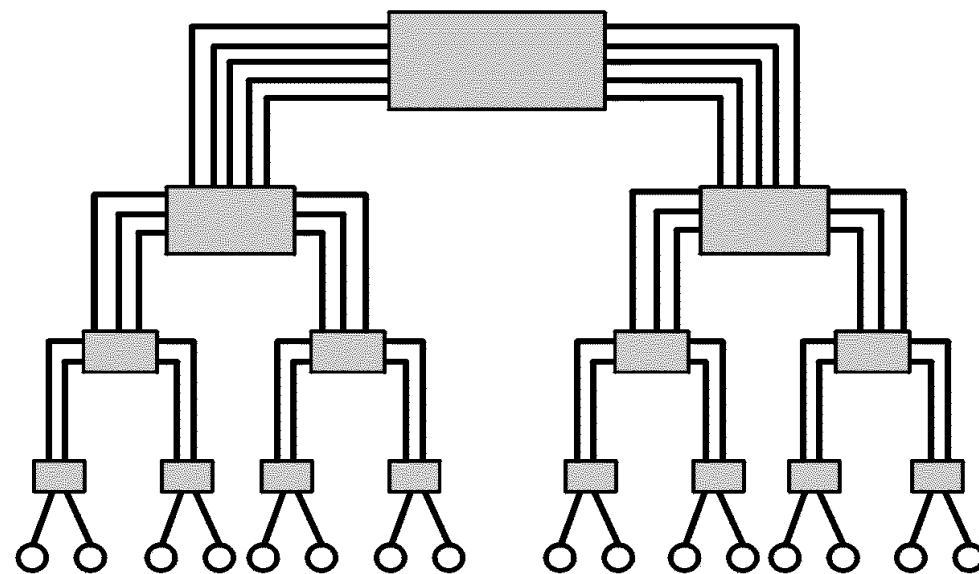
■ 一棵4层15个结点的二叉树

一般说来，一棵 k 层完全平衡的二叉树有 $N=2^k-1$ 个结点。

➤ 最大结点度：3

➤ 直径： $2(k-1)$

➤ 等分宽度 $b=1$



(c) 二叉胖树

6. 网格形和环网形

■ 网格形

➤ 一个 3×3 的网格形网络

➤ 一个规模为 $N=n \times n$ 的2维网格形网络

✓ 内部结点的度 $d=4$

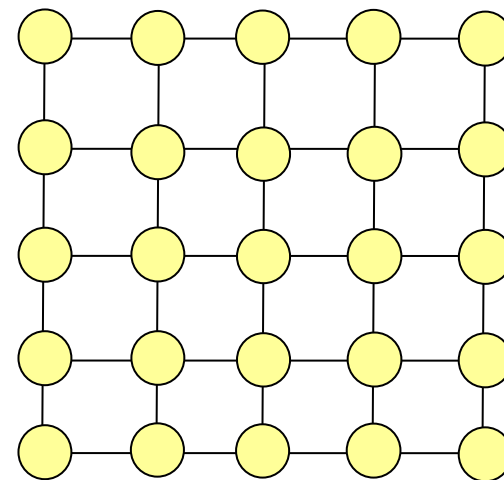
✓ 边结点的度 $d=3$

✓ 角结点的度 $d=2$

✓ 网络直径 $D=2(n-1)$

✓ 等分宽度 $b=n$

➤ 一个由 $N=n^k$ 个结点构成的 k 维网格形网络（每维 n 个结点）的内部结点度 $d=2k$ ，网络直径 $D=k(n-1)$ 。

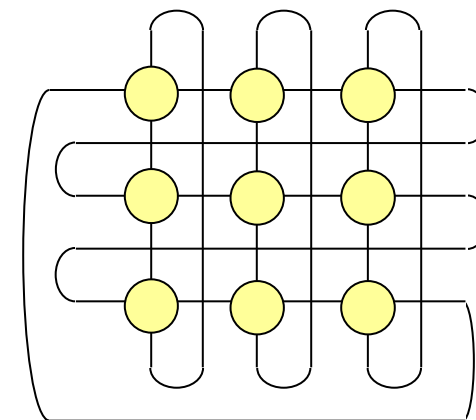


(a) 网格形

6. 网格形和环网形

■ Illiac网络

- 名称来源于采用了这种网络的Illiac IV计算机
- 把2维网格形网络的每一列的两个端结点连接起来，再把每一行的尾结点与下一行的头结点连接起来，并把最后一行的尾结点与第一行的头结点连接起来。
- 一个规模为 $n \times n$ 的Illiac网络
 - ✓所有结点的度 $d=4$
 - ✓网络直径 $D=n-1$ ，Illiac网络的直径只有纯网格形网络直径的一半。
 - ✓等分宽度： $2n$

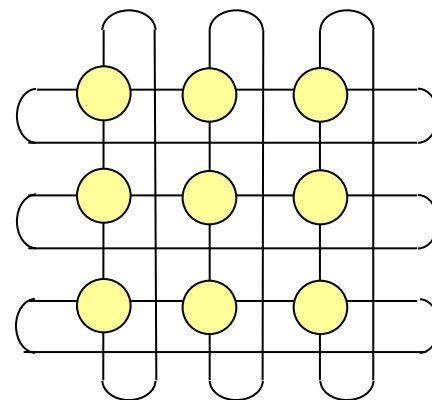


(b) Illiac网

6. 网格形和环网形

■ 环网形

- 可看作是直径更短的另一种网格。
- 把2维网格形网络的每一行的两个端结点连接起来，把每一列的两个端结点也连接起来。
- 将环形和网格形组合在一起，并能向高维扩展。
- 一个 $n \times n$ 的环网形网
 - ✓ 结点度：4
 - ✓ 网络直径： $2 \times \lfloor n/2 \rfloor$
 - ✓ 等分宽度 $b=2n$

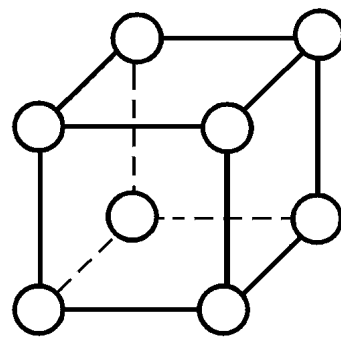


(c) 环网形

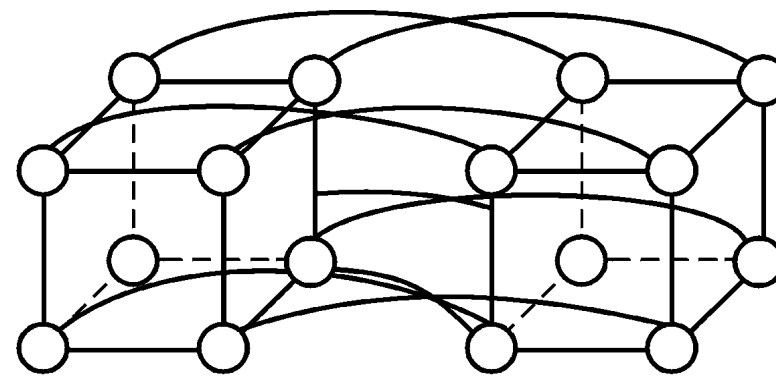
7. 超立方体

- 一种二元 n -立方体结构
- 一般来说，一个二元 n -立方体由 $N=2^n$ 个结点组成，它们分布在 n 维上，每维有两个结点。

例 8个结点的3维立方体
4维立方体



(a) 3-立方体

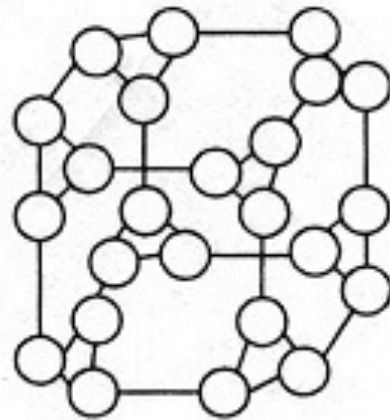


(b) 由 2 个 3-立方体组成的 4-立方体

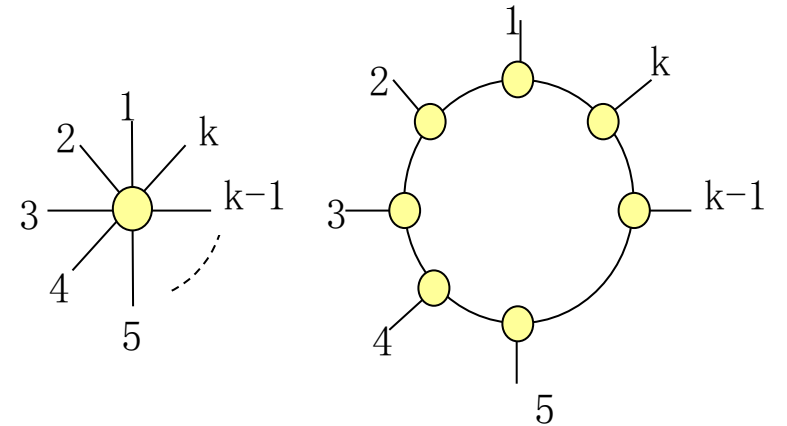
- 为实现一个 n -立方体，只要把两个 $(n-1)$ 立方体中相对应的结点用链路连接起来即可。共需要 2^{n-1} 条链路。
- n -立方体中结点的度都是 n ，直径也是 n ，等分宽度为 $b=N/2$ 。

8. 带环立方体（简称3-CCC）

- 把3-立方体的每个结点换成一个由3个结点构成的环而形成的。
- 带环k-立方体（简称k-CCC）
 - k-立方体的变形，它是通过用k个结点构成的环取代k-立方体中的每个结点而形成的。
 - 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 $b = N / (2k)$



(a) 带环3-立方体



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

9.3 静态互连网络

第 44 页

9. k元n-立方体网络

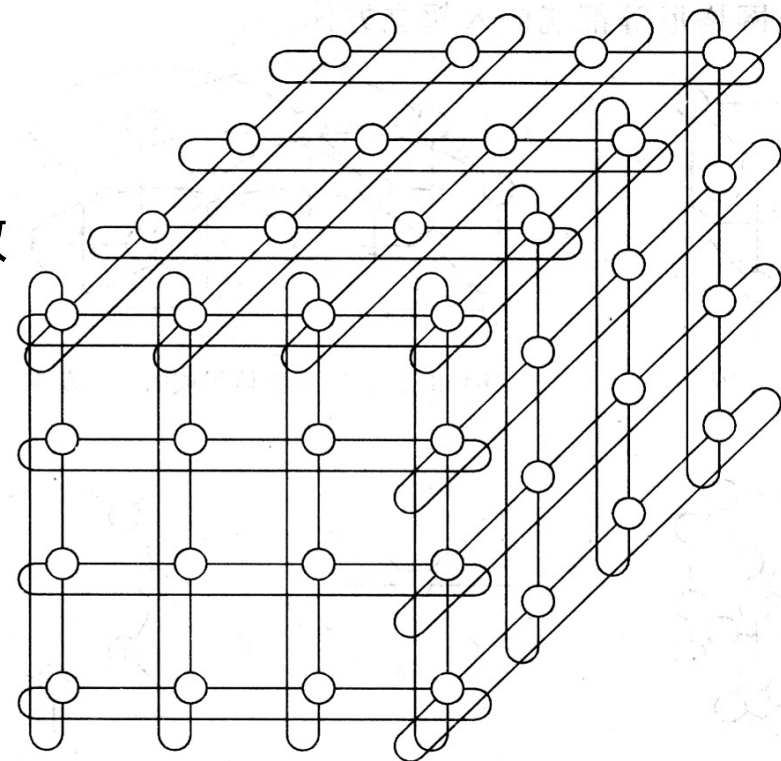
- 环形、网格、环网形、二元n-立方体（超立方体）和Omega网络都是k元n-立方体网络系列的拓扑同构体。
- 在k元n-立方体网络中，参数n是立方体的维数，k是基数即每一维上的结点个数。

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

- k元n-立方体的结点可以用基数为k的n位地址A= $a_1a_2\dots a_n$ 来表示。

其中 a_i 表示该结点在第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

9.3 静态互连网络

第 45 页

静态互连网络特征一览表

网络类型	结点度d	网络直径D	链路数l	等分宽度B	对称性	网络规格说明
线线阵列	2	$N-1$	$N-1$	1	非	N个结点
环形	2	$[N/2]$	N	2	是	N个结点
全连接	$N-1$	1	$N(N-1)/2$	$(N/2)^2$	是	N个结点
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个结点
2D网格	4	$2(r-1)$	$2N-2r$	r	非	$r \times r$ 网格, $r = \sqrt{N}$
Illiacc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	$r \times r$ 环网, $r = \sqrt{N}$
超立方体	n	n	$nN/2$	$N/2$	是	N个结点, $n=[\log_2 N]$ (维数)
CCC	3	$2k-1+[k/2]$	$3N/2$	$N/(2k)$	是	$N=k \times 2^k$ 结点 环长 $k \geq 3$
k元n-立方体	$2n$	$n[k/2]$	nN	$2k^{n-1}$	是	$N=k^n$ 个结点

9.3 静态互连网络

第 46 页

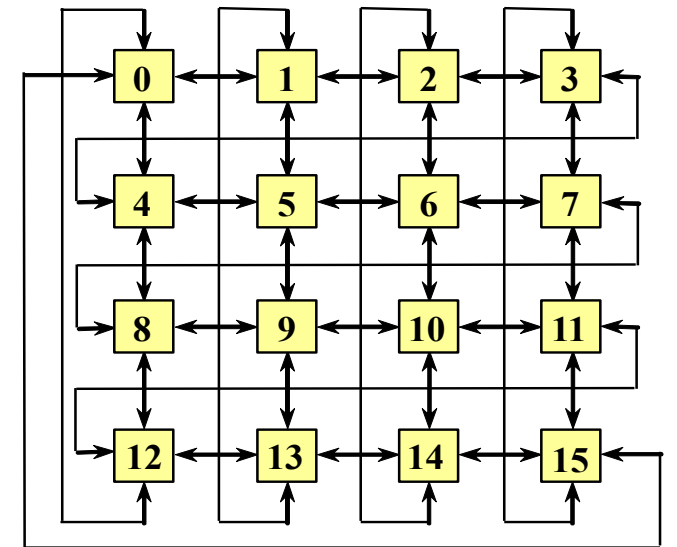
例9.2 已知有16台个处理器用Illiac网络互连，写出Illiac网络的互连函数，给出表示任何一个处理器 PU_i ($0 \leq i \leq 15$) 与其他处理器直接互连的一般表达式。

解：Illiac网络连接的结点数 $N=16$ ，组成 4×4 的阵列。每一列的4个处理器互连为一个双向环，第1列～第4列的双向环可分别用循环互连函数表示为：

$$\begin{array}{ll} (0 \ 4 \ 8 \ 12) & (12 \ 8 \ 4 \ 0) \\ (1 \ 5 \ 9 \ 13) & (13 \ 9 \ 5 \ 1) \\ (2 \ 6 \ 10 \ 14) & (14 \ 10 \ 6 \ 2) \\ (3 \ 7 \ 11 \ 15) & (15 \ 11 \ 7 \ 3) \end{array}$$

其中，传送方向为顺时针的4个单向环的循环互连函数可表示为：

$$PM_{2+2}(X) = (X + 2^2) \bmod N = (X + 4) \bmod 16$$



9.3 静态互连网络

第 47 页

传送方向为逆时针的4个单向环的循环互连函数可表示为：

$$PM2_{-2}(X) = (X - 2_2) \bmod N = (X - 4) \bmod 16$$

16个处理器由Illiac网络的水平螺线互连为一个双向环，用循环互连函数表示为：

(0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

(15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)

其中，传送方向为顺时针的单向环的循环互连函数可表示为：

$$PM2_{+0}(X) = (X + 2_0) \bmod N = (X + 1) \bmod 16$$

传送方向为逆时针的单向环的循环互连函数可表示为：

$$PM2_{-0}(X) = (X - 2_0) \bmod N = (X - 1) \bmod 16$$

所以， $N=16$ 的Illiac网络的互连函数有4个： $PM2_{\pm 0}(X)$ 和 $PM2_{\pm 2}(X)$

由互连函数可得任何一个处理器 i 直接与下述4个处理器双向互连：

$$i \pm 1 \bmod 16 \quad i \pm 4 \bmod 16$$

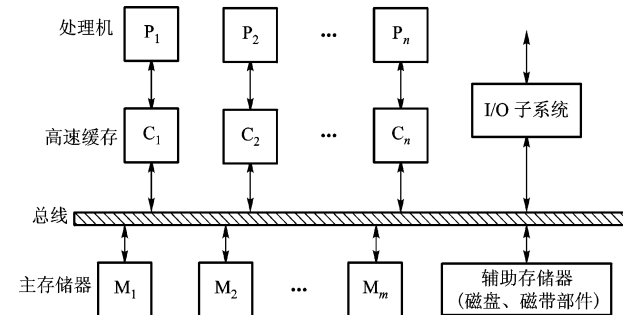
9.4 动态互连网络

9.4.1 总线网络

第 49 页

1. 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。

- 每一次总线只能用于一个源（主部件）到一个或多个目的（从部件）之间的数据传送。
- 多个功能模块之间的争用总线或时分总线
- 特点
 - 结构简单、实现成本低、带宽较窄



2. 一种由总线连接的多处理机系统

- 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

总线的分类

1. 片内总线（在芯片内部）
2. 系统总线（各个部件之间的信息传输线）
 - 数据总线：双向 与机器字长、存储字长有关
 - 地址总线：单向 与存储地址、I/O地址有关
 - 控制总线：有出 有入（从CPU的角度来说的），如存储器读、写，总线允许，终端确认；中断请求，总线请求
3. 通讯总线用于计算机系统之间或计算机系统与其他系统（如控制仪表、移动通信等）之间的通信

3. 解决总线带宽较窄问题：采用多总线或多层次的总线

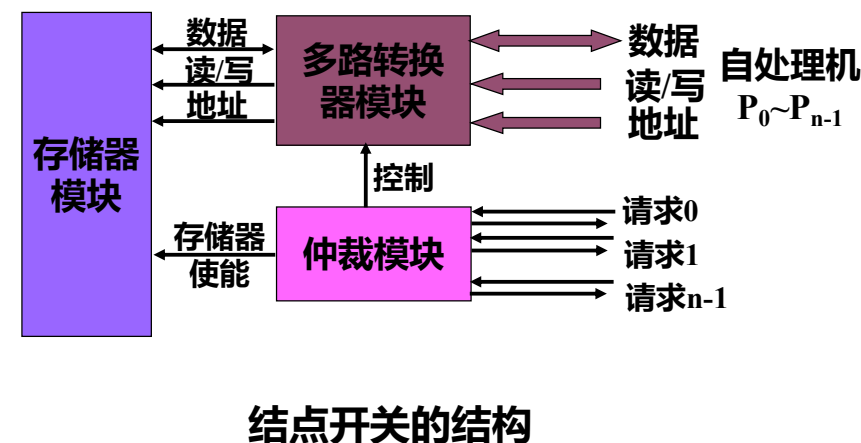
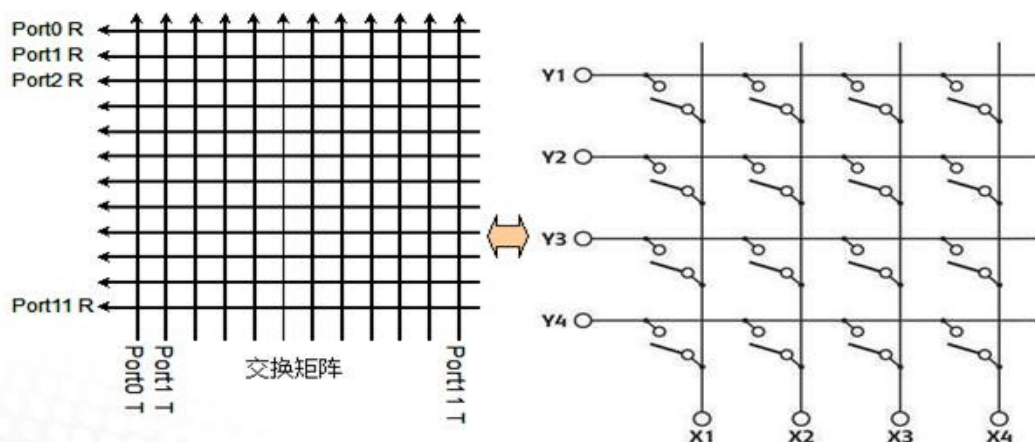
- 多总线是设置多条总线，有两种做法：
 - 为不同的功能设置专门的总线
 - 重复设置相同功能的总线
- 多层次的总线是按层次的架构设置速度不同的总线，使得不同速度的模块有比较适合的总线连接。

9.4.2 交叉开关网络

第 51 页

1. 单级开关网络

- 交叉点开关能在对偶（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。

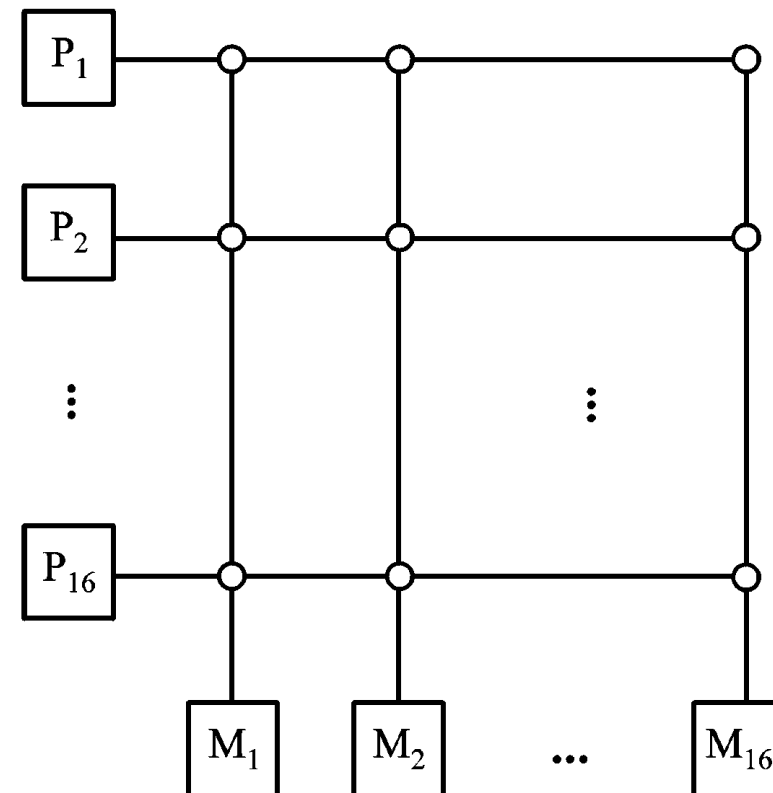


9.4.2 交叉开关网络

第 52 页

2. 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。**

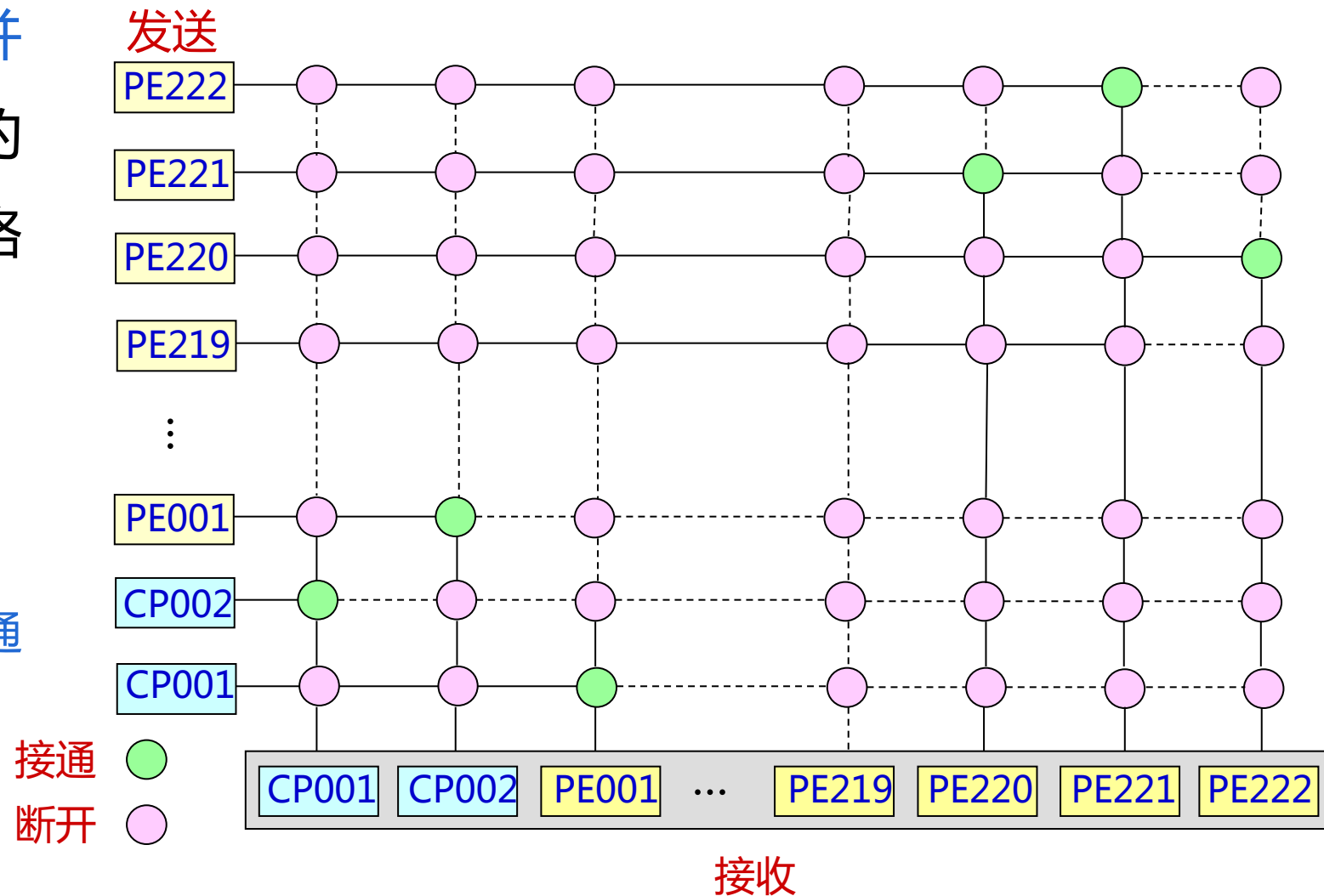


9.4.2 交叉开关网络

第 53 页

3. 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



9.4.3 多级互连网络

第 54 页

1. 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

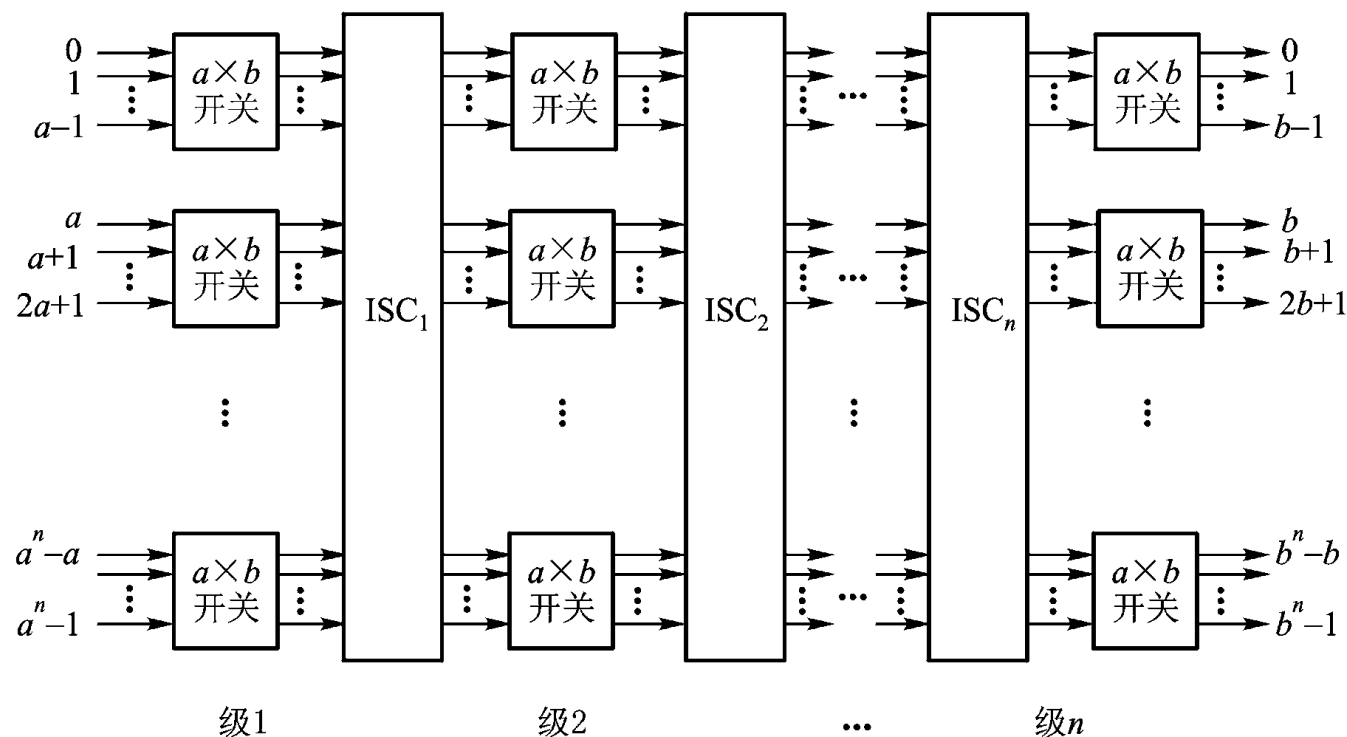
➤ 由 $a \times b$ 开关模块和级间连接构成的通用多级互连网络结构

➤ 每一级都用了多个 $a \times b$ 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 $a = b = 2^k, k \geq 1$ 。

➤ 相邻各级开关之间都有固定的级间连接



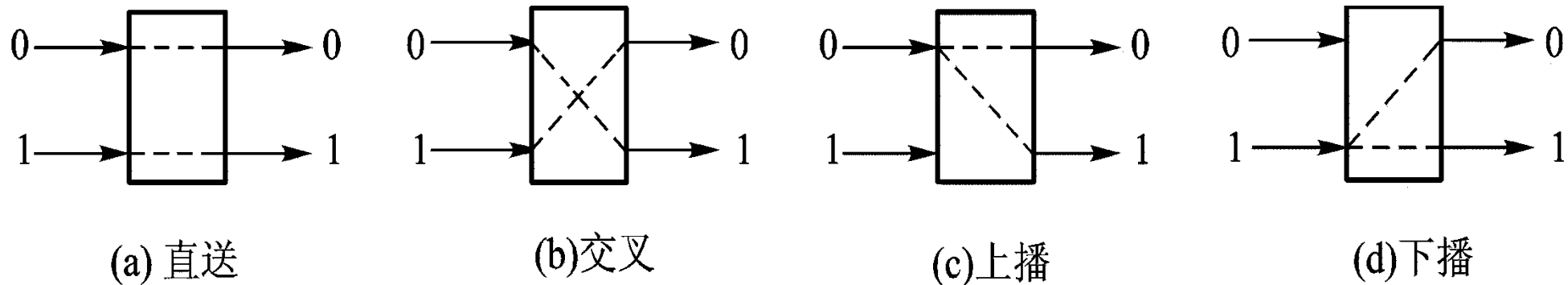
9.4.3 多级互连网络

第 55 页

■ 几种常用的开关模块

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
$n \times n$	n^n	$n!$

■ 最简单的开关模块：2×2开关



9.4.3 多级互连网络

第 56 页

- 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

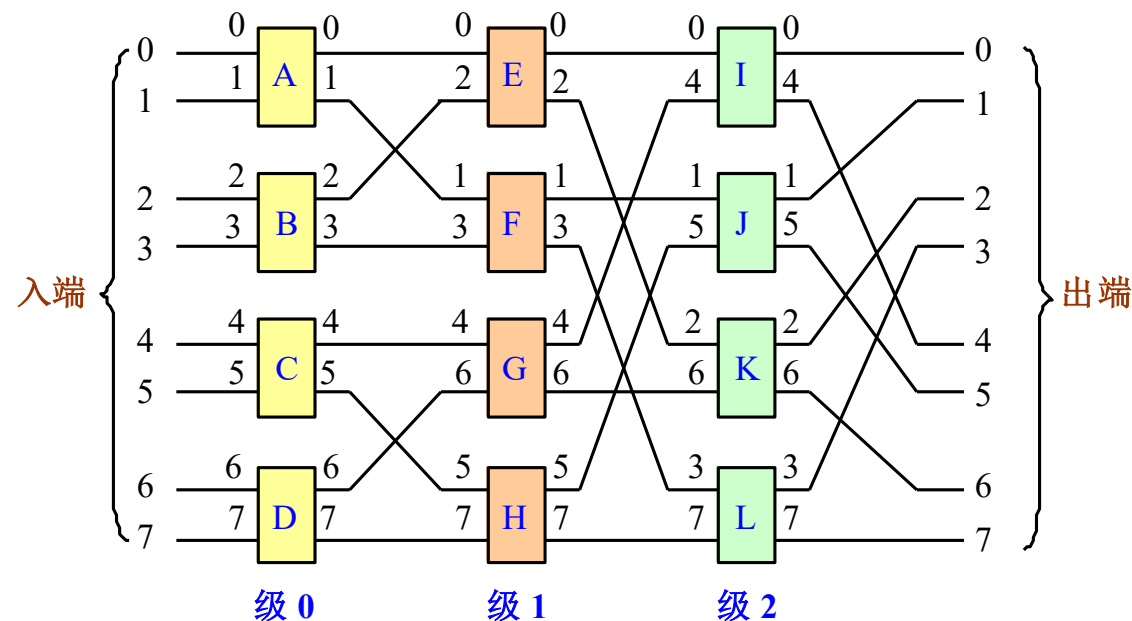
2. 多级立方体网络

- 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（ $0 \leq i \leq n-1$ ）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。

9.4.3 多级互连网络

第 58 页

- 一个8个入端的多级立方体网络
- STARAN网络采用级控制和部分级控制
 - 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



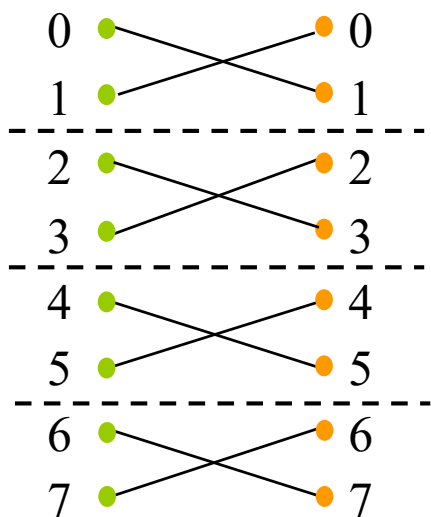
9.4.3 多级互连网络

第 59 页

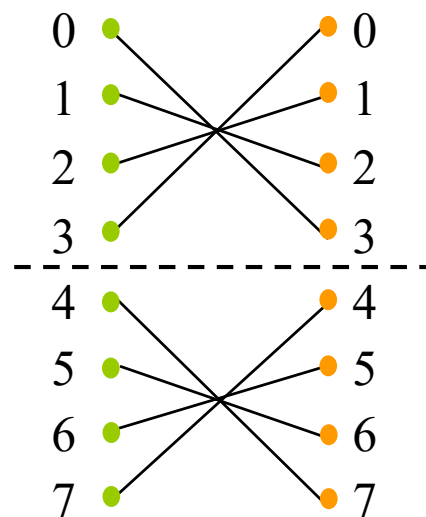
■ 交换

- 将有序的一组元素头尾对称地进行交换。

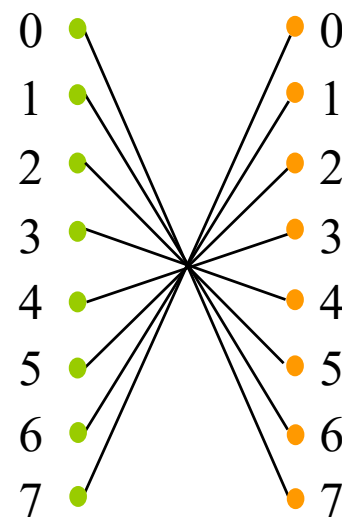
例如：对于由8个元素构成的组，各种基本交换的图形：



(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

9.4.3 多级互连网络

第 60 页

- 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，
 - 表中下面的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。（ $k_2k_1k_0$ ：控制信号， k_i （ $i=0, 1, 2$ ）为第 i 级的级控制信号。）

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

9.4.3 多级互连网络

■ 当STARAN网络用作移数网络时，采用部分级控制，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

9.4.3 多级互连网络

第 62 页

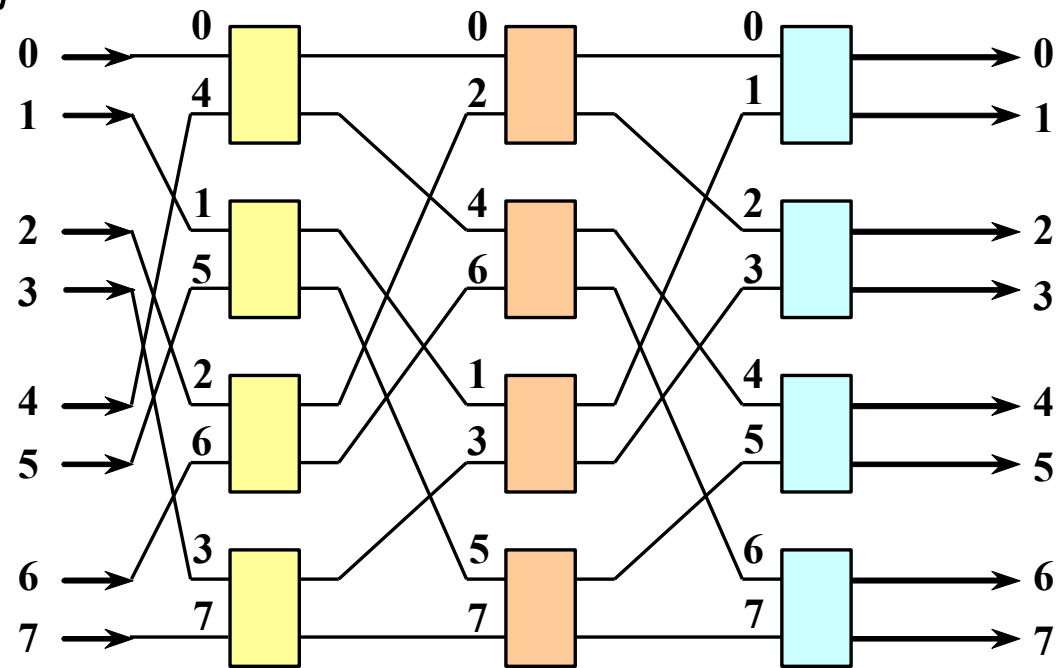
3. Omega网络

- 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+恒等置换连接方式

- 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。
- 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它连接。
- 是阻塞网络：并不是所有的置换在 Ω 网中一次通过便可以实现。出现冲突时，可以采用几次通过的方法来解决冲突。



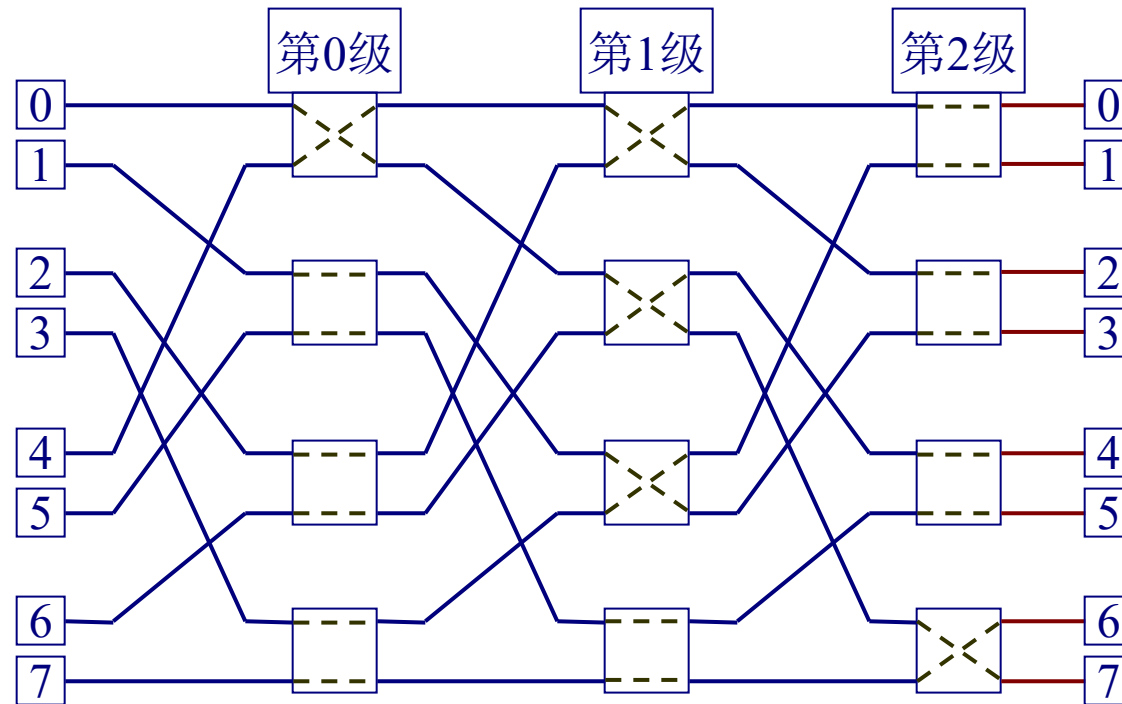
9.4.3 多级互连网络

第 63 页

3. Omega网络

无阻塞的实现置换

$$\pi_1 = (0\ 7\ 6\ 4\ 2)\ (1\ 3)\ (5)$$

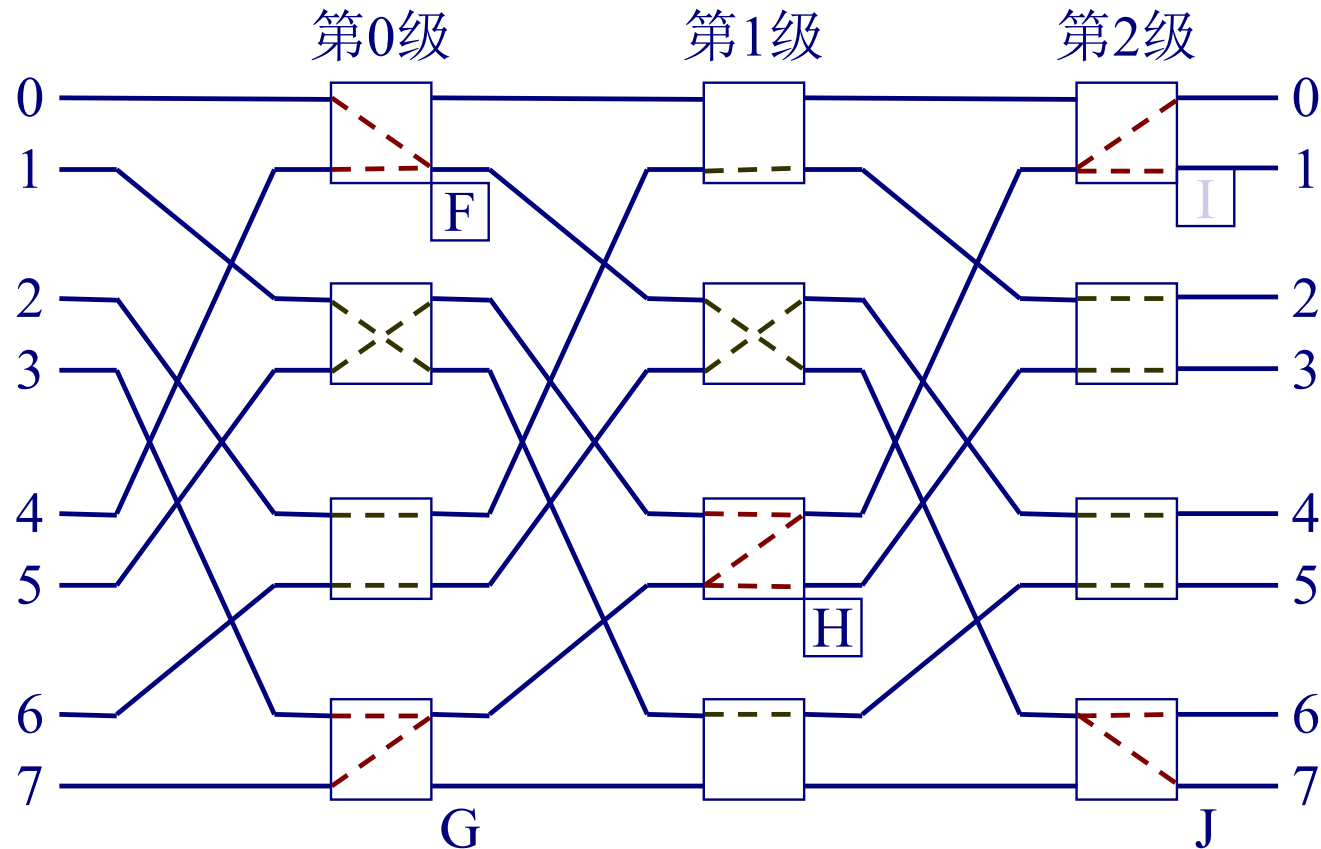


9.4.3 多级互连网络

第 64 页

3. Omega网络

置换 $\pi_2 = (0\ 6\ 4\ 7\ 3)\ (1\ 5)\ (2)$ 阻塞

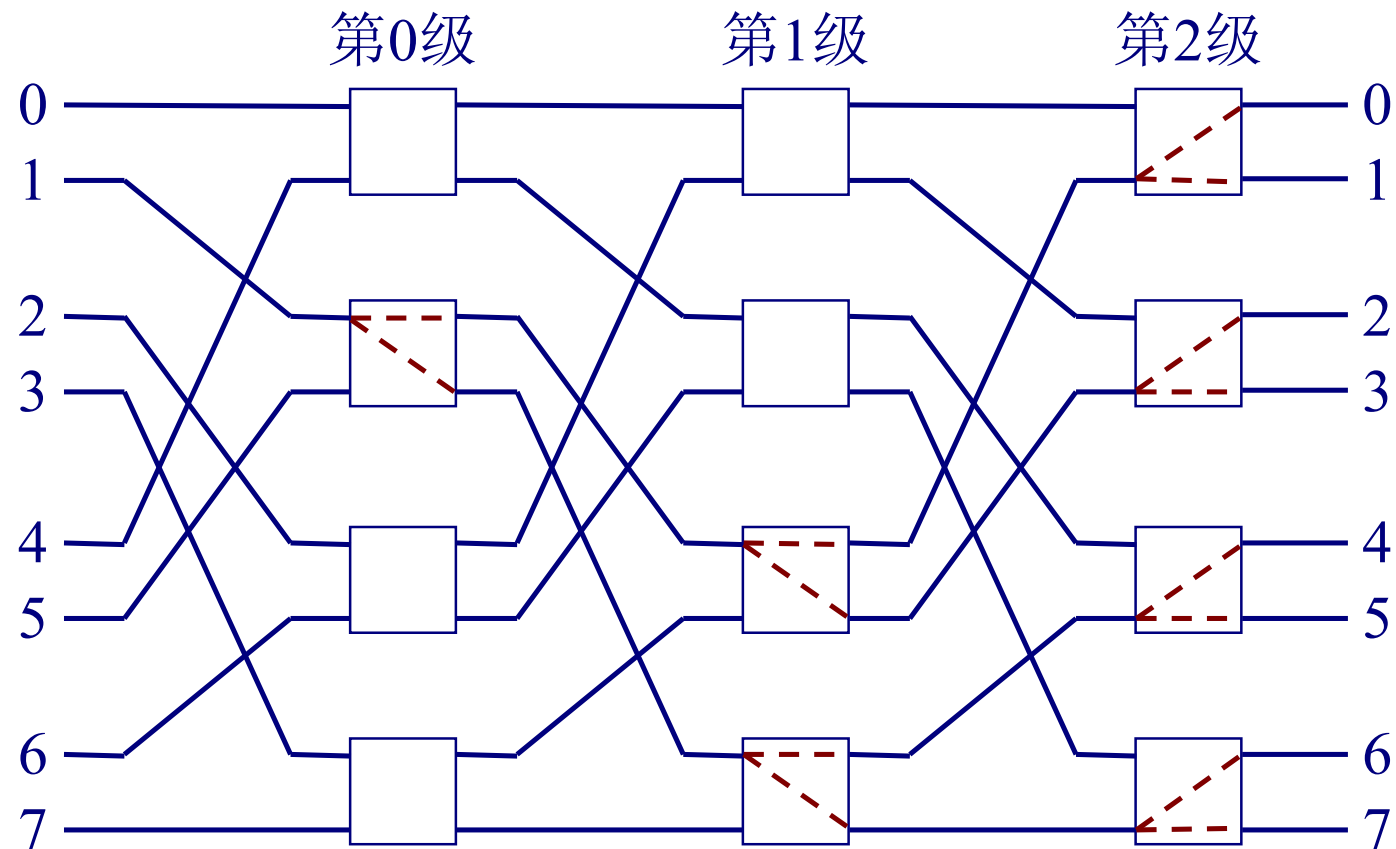


9.4.3 多级互连网络

第 65 页

3. Omega网络

实现广播功能

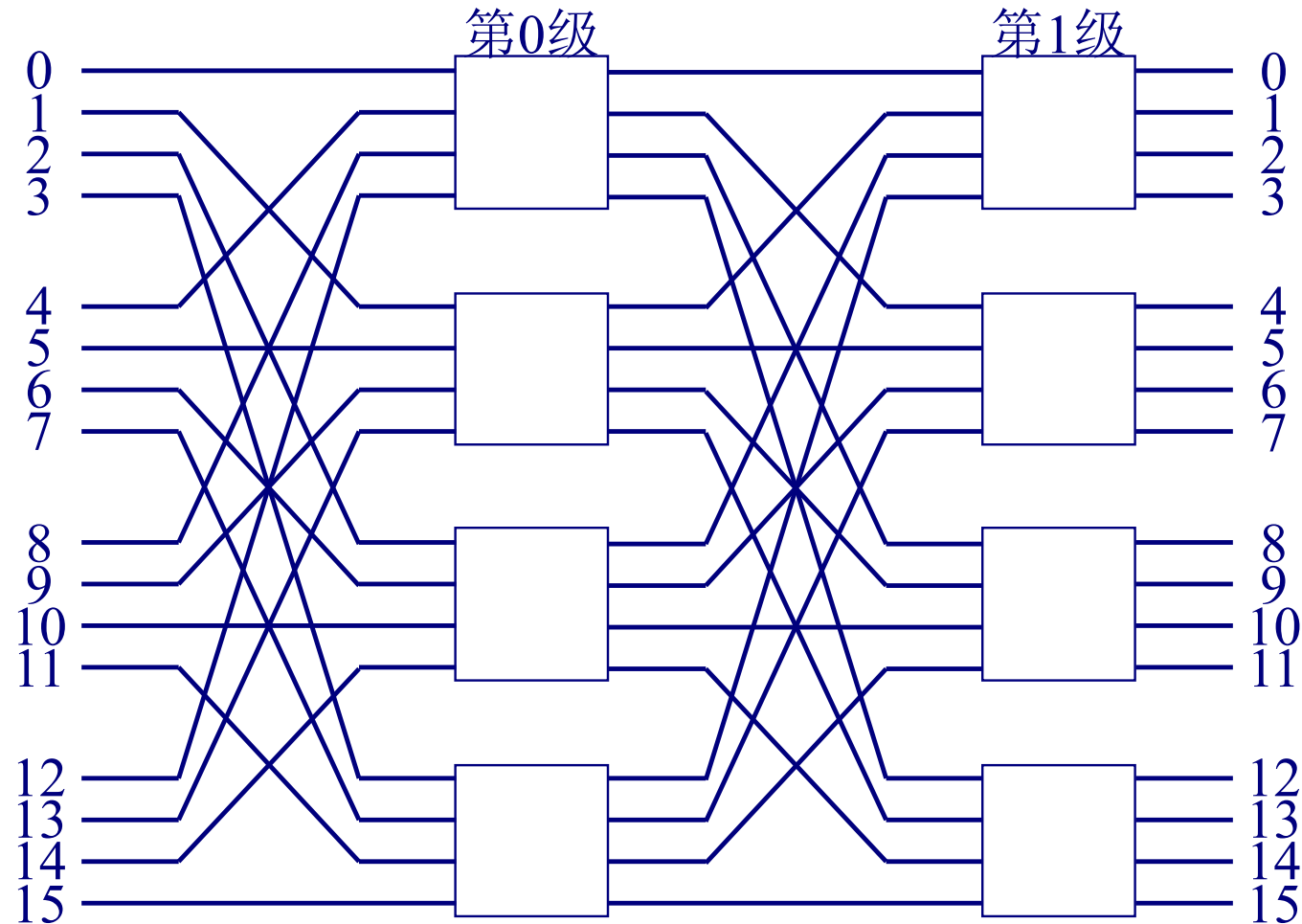


9.4.3 多级互连网络

第 66 页

3. 4×4 开关构成的Omega网络

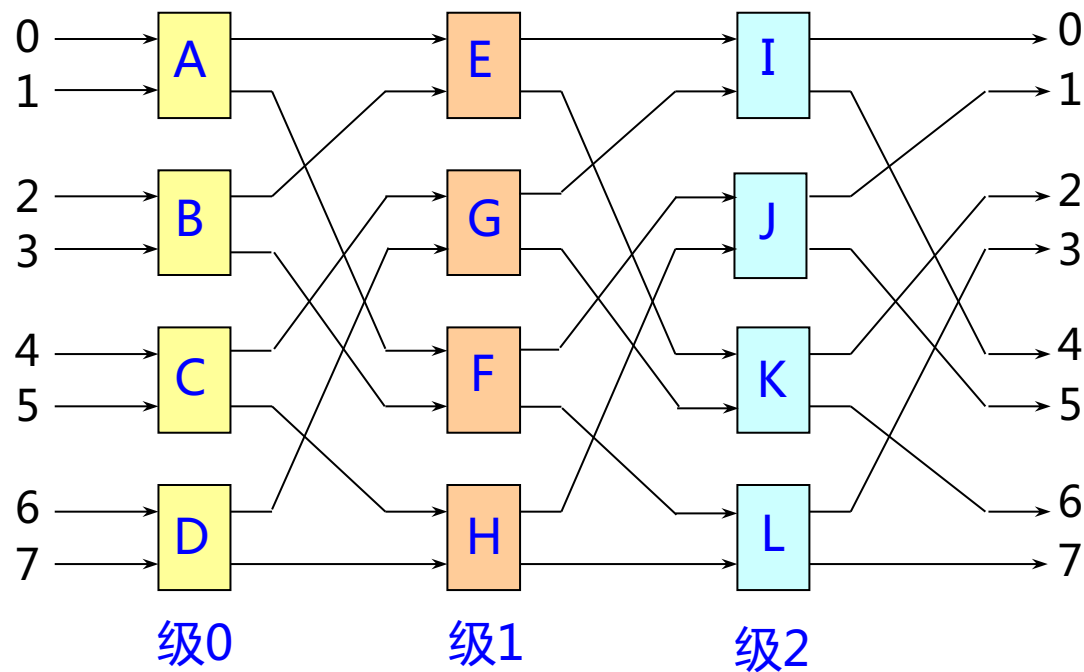
如16输入4路洗牌：网路级数为 $\log_4 16 = 2$



9.4.3 多级互连网络

第 67 页

- N=8的多级立方体互连网络的另一种画法



9.4.4 动态互连网络的比较

第 68 页

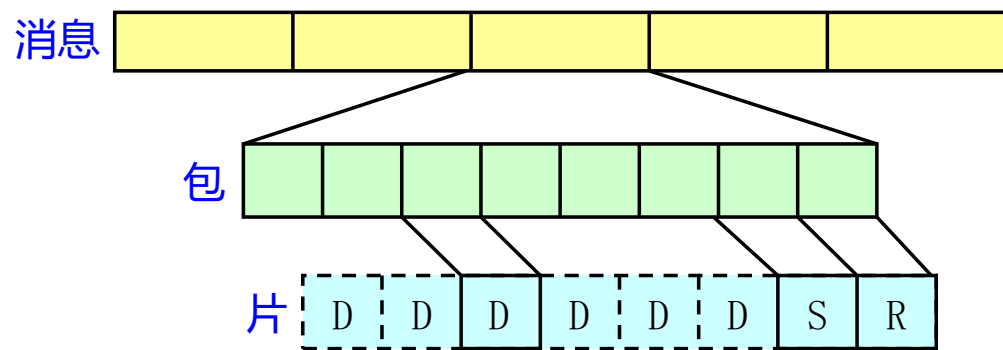
网络特性	总线系统	多级网络	交叉开关
单位数据传送的最小时延	恒定	$O(\log_k n)$	恒定
每台处理机的带宽	$O(w/n)$ 至 $O(w)$	$O(w)$ 至 $O(nw)$	$O(w)$ 至 $O(nw)$
连线复杂性	$O(w)$	$O(nw \log_k n)$	$O(n^2 w)$
开关复杂性	$O(n)$	$O(n \log_k n)$	$O(n^2)$
连接特性和寻径性能	一次只能一对一	只要网络不阻塞， 可实现某些置换和广播	全置换， 一次一个
典型计算机	Symmetry S1 , Encore Multimax	BBNTC-2000 IBM RP3	Cray Y-MP/816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

9.5 消息传递机制

9.5.1 消息寻径方案

第 70 页

- 当源结点和目的结点之间没有直接的连接时，消息需要经过中间的结点进行传递。
- 寻径就是用来实现这种传递的通信方法和算法，有的称之为路由。



消息、包和片的格式

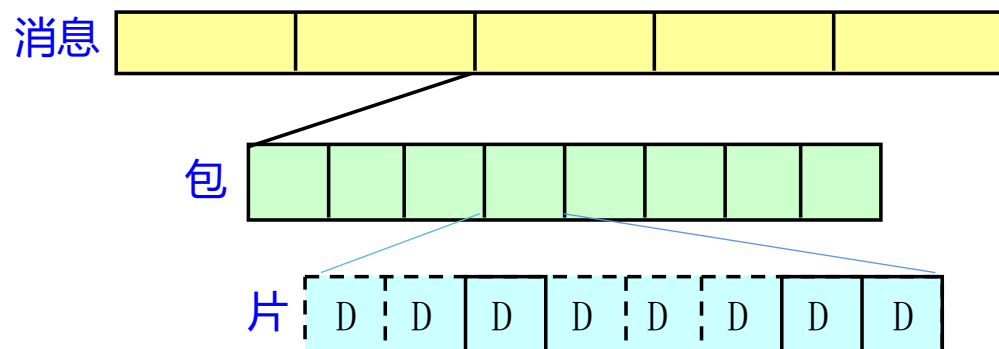
R : 寻径信息 S : 顺序号 D : 数据片

■ 消息的格式

- **消息**：结点之间进行通信的逻辑单位
 - ✓ 由若干个“包”组成
 - ✓ 包的长度是固定的，一条消息中所包含的包的个数是可变的，消息的长度是不定长的。

9.5.1 消息寻径方案

第 71 页



消息、包和片的格式

➤包：包含寻径所需目的地址的基本单位。

- ✓一条消息中的各个包依次被分配一个序号以便这些包到达目的结点后能重新组装出消息。
- ✓包可以进一步分成一些更小的固定长度的单位，称为“片”。寻径信息和包序列号形成头片，其余的是数据片。
- ✓包的长度主要是由寻径方案和网络的具体实现所决定的，典型的长度是64 ~ 512位不等
- ✓片的长度经常是受网络大小的影响

9.5.1 消息寻径方案

第 72 页

2. 四种寻径方式

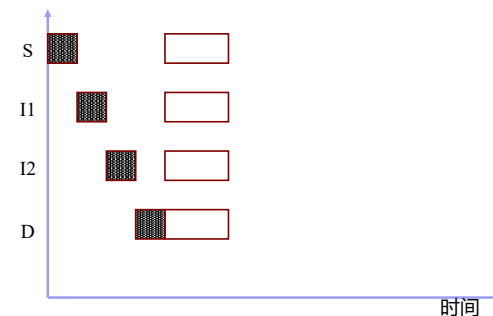
- **线路交换**：在线路交换方式下，在传递一个信息之前，需要先建立一条从源结点到目的结点的物理通路，然后再传递信息。

➤ 传输时延 T

$$T = \frac{L + L_t \times (D + 1)}{B}$$

其中：

- ✓ L ：信息包的长度（位数）
- ✓ L_t ：建立路径所需的小信息包的长度
- ✓ D ：经过的中间结点个数
- ✓ B ：带宽



- **优点**：传输带宽较大，平均传输时延较小，而且使用的缓冲区小。
 - 适合于具有动态和突发性的大规模并行处理数据的传送。
- **缺点**：需要频繁地建立源结点到目的结点的物理通路，时间开销会很大。

9.5.1 消息寻径方案

第 73 页

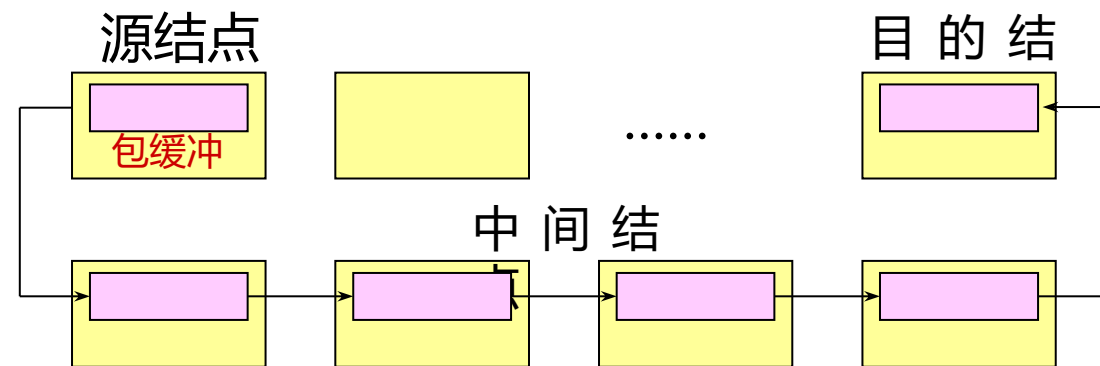
■ 存储转发：最简单的分组交换方式。

- 包是信息传递的基本单位。包从源结点经过一系列中间结点到达目的结点。
- 要求：所经过的每个中间结点都要设置一个包缓冲器，用于保存所传递的包。当一个包到达某个中间结点时，该结点先把这个包全部存储起来，然后在出口链路可用、而且下一个结点的包缓冲器也可用的情况下，传递给下一个结点。
- 网络的时延与源和目的地之间的距离（跳数）成正比

$$T_{SF} = \frac{L}{B}(D + 1)$$

➤ 缺点

- ✓ 包缓冲区大，不利于VLSI实现；
- ✓ 网络时延大，与结点距离成正比。



(a) 存储转发

9.5.1 消息寻径方案

第 74 页

■ **虚拟直通**：对存储转发方式的一种改进，减少了网络时延。

➤ **基本思想**：没有必要等到信息包全部放入缓冲器后再作路由选择，只要接收到用作寻径的包头，就可作出判断。

✓ 如果结点的输出链路空闲，信息包可以不必存储在该结点的缓冲器中，而是立即传送到下一个结点。

✓ 如果整条链路都空闲，包就可以立即直达目的结点。

✓ 在输出链路不空闲时，要用缓冲器进行存储。

➤ **通信时延**

$$T = \frac{L + L_h \times (D + 1)}{B} \approx \frac{L}{B}$$

✓ L_h ：信息包寻径头部的长度

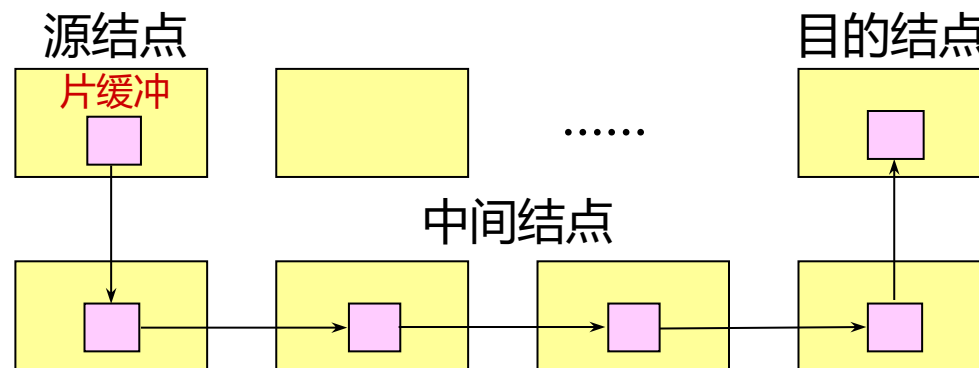
✓ 一般来说， $L \gg L_h \times (D + 1)$ ，所以 $T \approx L/B$ 。

➤ 当出现寻径阻塞时，虚拟直通方式需要将整个信息包全部存储在寻径结点中，要求每个结点都有足够大的缓冲区，不利于VLSI的实现

9.5.1 消息寻径方案

第 75 页

- **虫蚀方式**：把信息包“切割”成更小的单位——“片”，而且使信息包中各片的传送按流水方式进行。
 - 可以减少结点中缓冲器的容量，缩短传送延迟时间。
 - 在新型的多计算机系统中得到了广泛的应用。
 - 处理的最小信息单位是“片”。当一个结点把头片送到下一个结点后，那么接下来就可以把后面的各个片也依次送出。一个结点一旦开始传送一个包中的头片后，这个结点就必须等待这个包的所有片都送出去后，才能传送其他包。不同包的片不能混合在一起传送。
 - 虫蚀方式与虚拟直通的不同之处
 - ✓ 当输出通路忙时，结点是把一个片存储到缓冲器中。
 - ✓ 由于片的大小比包小很多，所以能有效地减少缓冲器的容量，使得它易于用VLSI实现。



(b) 虫蚀方式

9.5.1 消息寻径方案

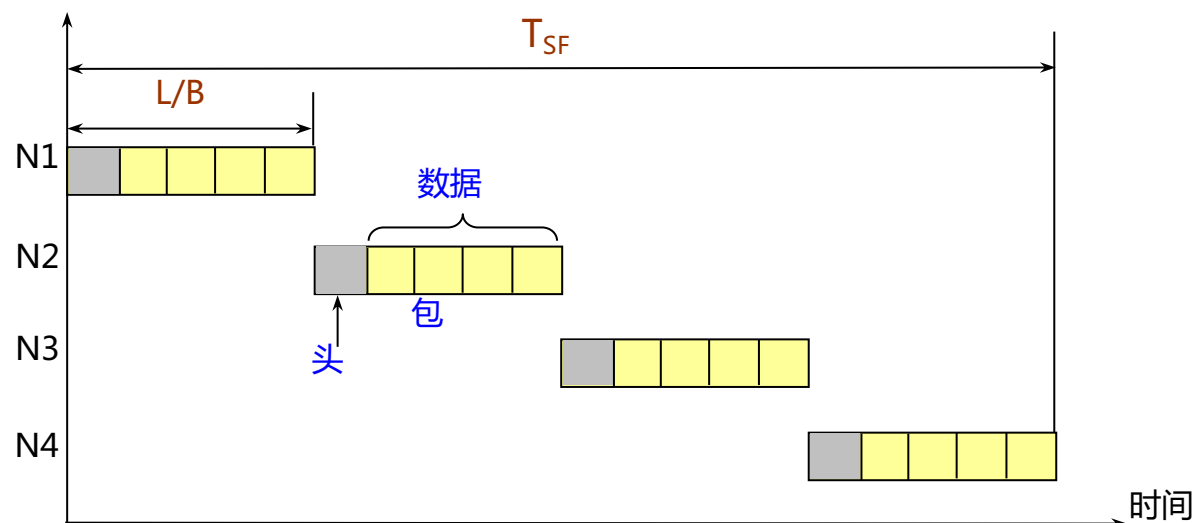
第 76 页

➤ 通信时延

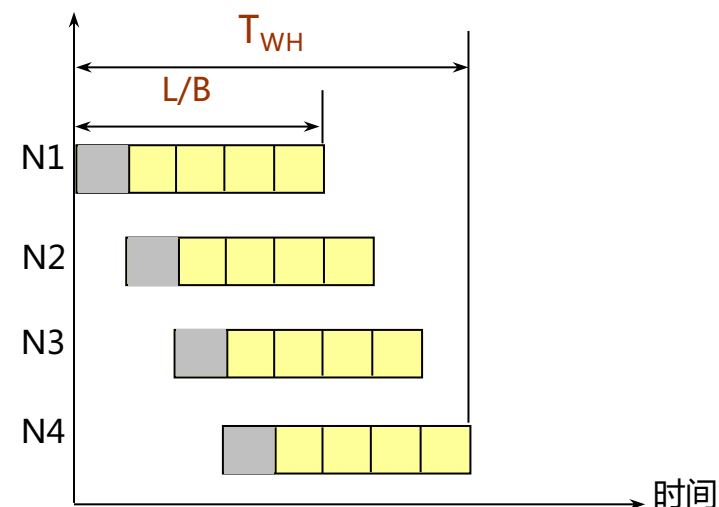
$$T_{WH} = T_f \times D + \frac{L}{B} = \frac{L + L_f \times D}{B} \approx \frac{L}{B}$$

L_f : “片” 的长度

T_f : 片经过一个结点所需时间, $L \gg L_f \times D$ 。



(a) 存储转发



(b) 虫蚀方式

存储转发与虫蚀方式的时间比较

■优点

- 每个节点的缓冲器较小，易于VLSI实现；
- 有较小的网络传输延迟；
- 通道共享性好，利用率高；
- 易于实现选播和广播通信模式。

■缺点

- 当消息的一片被阻塞时，整个消息的所有片都将被阻塞在所在结点，占用了结点资源。

9.5.2 死锁与虚拟直通

第 78 页

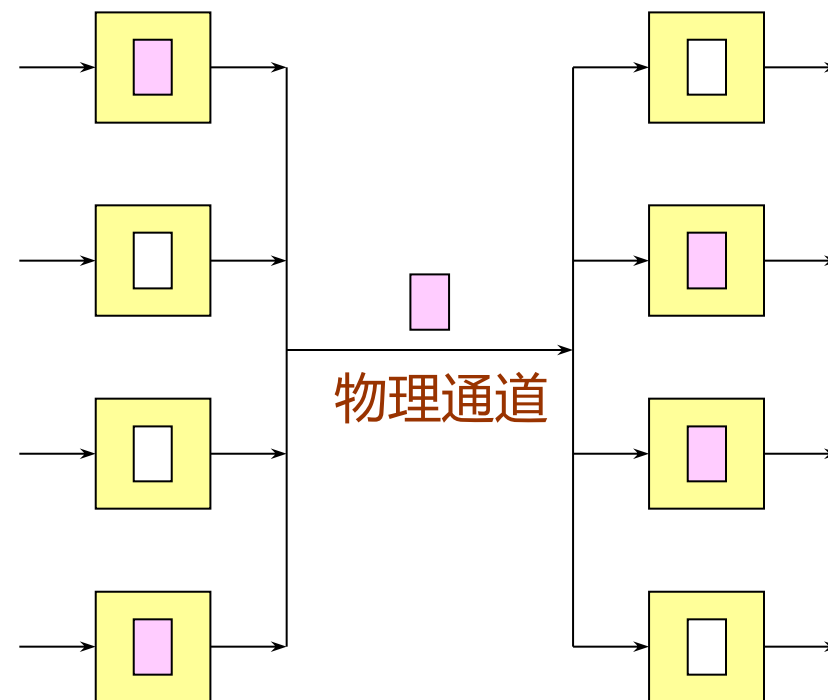
1. 虚拟通道：两个结点间的逻辑链接，它由源结点的片缓冲区、结点间的物理通道以及接收结点的片缓冲区组成。

- 4条虚拟通道共享一条物理通道

- 源结点和接收结点各有4个片缓冲区。
- 当物理通道分配给某对缓冲区时，这一对的源缓冲区和接收缓冲区就形成了一条虚拟通道。
- 物理通道是由所有的虚拟通道分时地共享。

- 虚拟通道也可以用双向通道实现。把两条单向通道组合在一起可以构成一条双向通道。

- 增加了利用率，使通道的带宽加倍。



源结点中的片缓冲区

目的结点中的片缓冲区

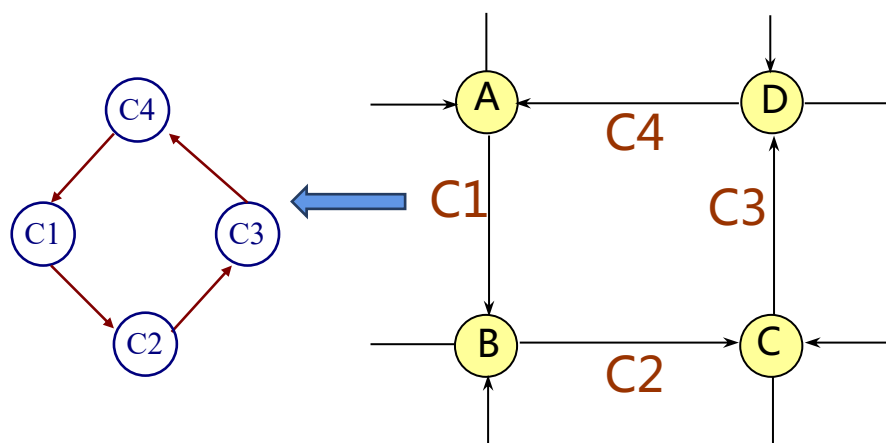
4条虚拟通道以片传递为基础分时地共享一条物理通道

9.5.2 死锁与虚拟直通

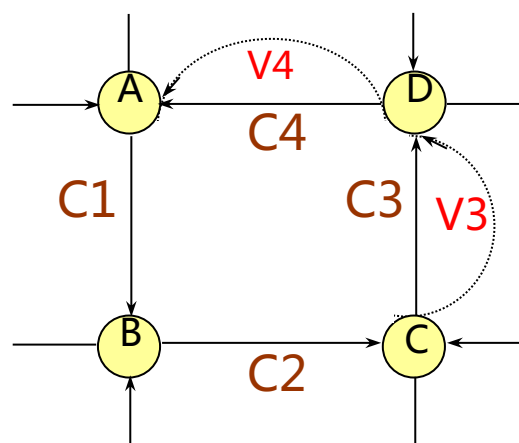
第 79 页

2. 避免死锁

- 缓冲区或通道上的循环等待会引起死锁。

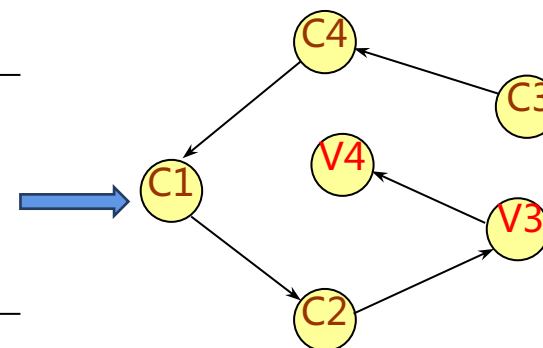


(a) 通道死锁



(b) 增加虚拟通道

利用虚拟通道减少死锁



(c) 利用虚拟通道后的通道相关图

例如：图（a）：出现循环的通道相关而产生死锁；图（b）：利用虚拟通道方法可以避免这个死锁，可以增加两条虚拟通道V3和V4；图（c）：避免了死锁，增加虚拟通道可能会使每个请求可用的有效通道带宽降低。因此，当实现数目很大的虚拟通道时需要用高速的多路选择开关。

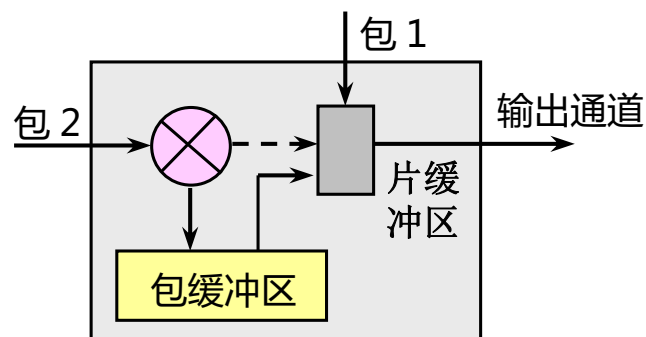
1. 包冲突的解决

- 为了通过通道在两个相邻结点之间传送一个片，要同时具备3个条件：
 - 源缓冲区已存有该片；
 - 通道已分配好；
 - 接收缓冲区准备接收该片。
- 当两个包到达同一个结点时，它们可能都在请求同一个接收缓冲器或者同一个输出通道，这时必须对两个问题进行仲裁。
 - 把通道分配给哪个包？
 - 如何处理被通道拒绝的包？

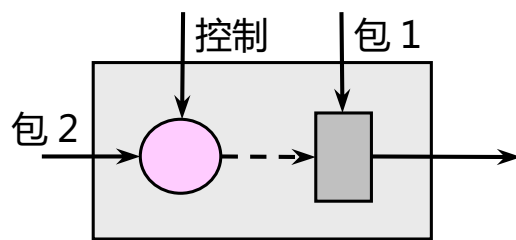
9.5.3 流控制策略

第 81 页

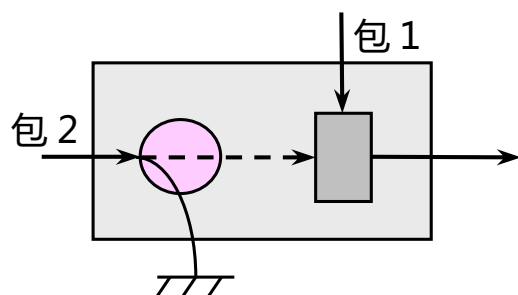
■ 4种解决方案



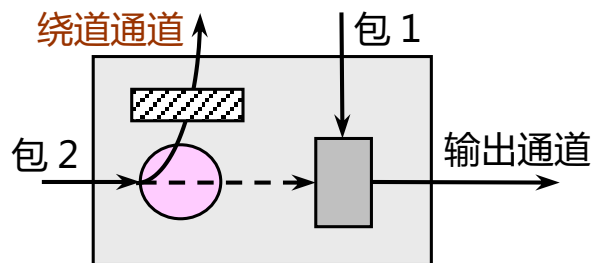
(a) 用缓冲实现虚拟通道



(b) 阻塞流控制



(c) 丢弃并重发



(d) 阻塞后绕道

- ① 把第二个包暂存在缓冲区
 - ✓ **优点:**不会浪费已经分配了的资源，但它要求结点中有一个足够大的缓冲器来存放整个信息包。
- ② 阻塞第二个包
- ③ 丢弃第二个包
 - ✓ 有可能会造成严重的资源浪费，而且要求重新进行被丢弃包的传输与确认。
- ④ 绕道
 - ✓ 在包寻径方面提供了更多的灵活性，但为了到达目的结点，可能要花费超过实际需要的通道资源，造成浪费。

2. 确定性寻径和自适应寻径

- **确定性寻径**：通信路径完全由源结点地址和目的地址来决定，也就是说，寻径路径是预先唯一地确定好了的，而与网络的状况无关。
- **自适应寻径**：通信的通路每一次都要根据资源或者网络的情况来选择。
 - 可以避开拥挤的或者有故障的结点，使网络的利用率得到改进。
- **两种确定性寻径算法**
 - 都是建立在维序概念之上的
 - 对于一个多维网来说，维序寻径要求对后继通道的选择是按照各维的顺序来进行的。
 - 对于二维的网格网络来说，这种寻径方法被称为**X-Y**寻径。
 - ✓ 先沿X维方向进行寻径，然后再沿Y维方向寻找路径。
 - 对于超立方体来说，这种寻径方法被称为E-cube寻径。

9.5.3 流控制策略

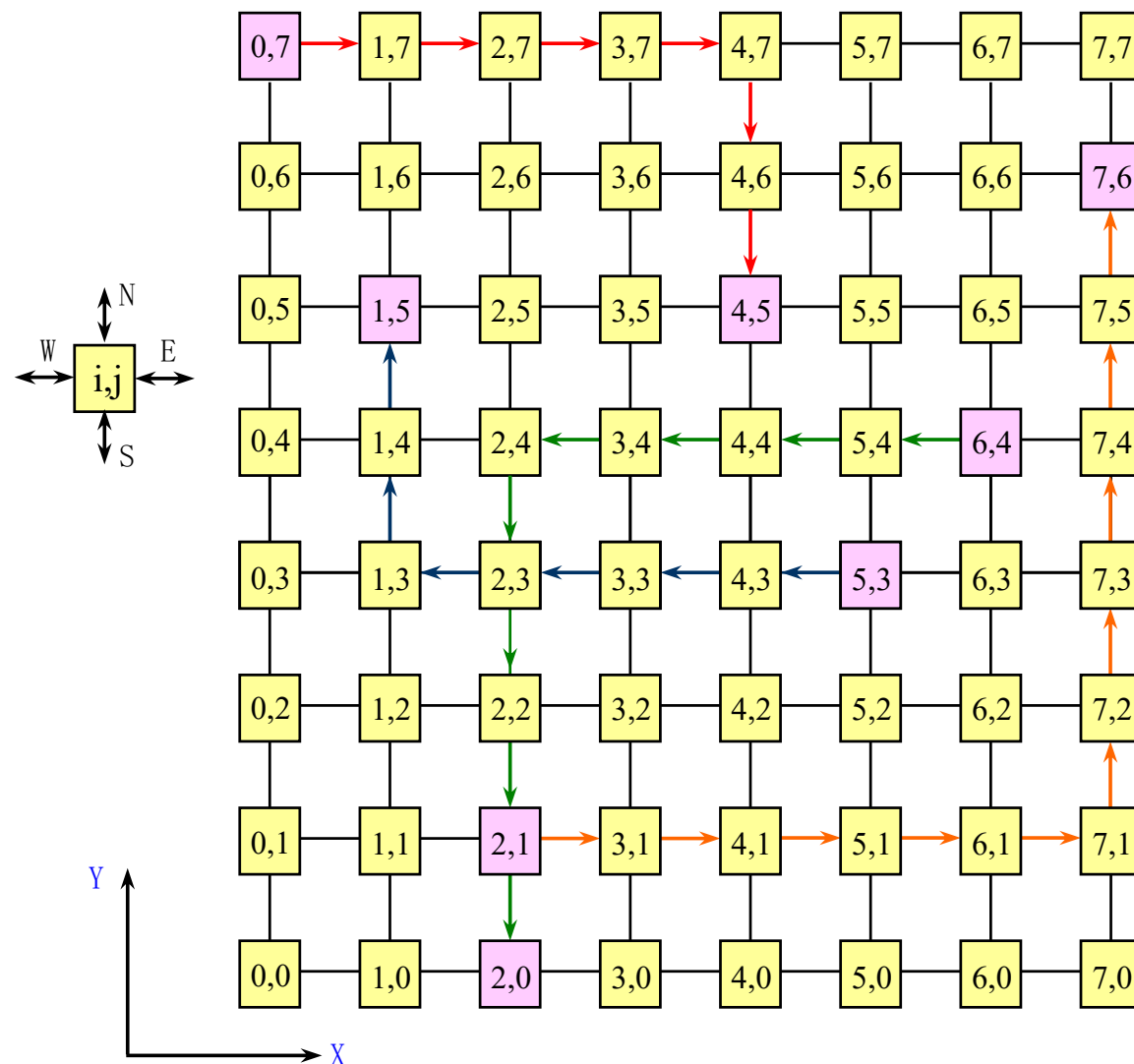
第 83 页

■ 二维网格网络的X-Y寻径

- 任意一个源结点： $s = (x_1, y_1)$
- 任意一个目的结点： $d = (x_2, y_2)$
- 从 s 出发，先沿X轴方向前进，直到找到 d 所在的列 x_2 ；
- 然后再沿Y轴方向前进，直到找到目标结点 (x_2, y_2) 。

例9.3 对于图所示的二维网格，确定以下4组“源结点-目的结点”所需要的路径。

- (2, 1) 到 (7, 6) (一条东-北路径)
- (0, 7) 到 (4, 5) (一条东-南路径)
- (6, 4) 到 (2, 0) (一条西-南路径)
- (5, 3) 到 (1, 5) (一条西-北路径)



9.5.3 流控制策略

第 84 页

考虑一个由 $N = 2^n$ 个结点构成的 n 方体，每个结点的编号是形为 $b = b_{n-1}b_{n-2}\dots b_1b_0$ 的二进制编码。设：源结点 $s = s_{n-1}s_{n-2}\dots s_1s_0$ ，目的结点 $d = d_{n-1}d_{n-2}\dots d_1d_0$ ，现在要确定一条从 s 到 d 的步数最少的路径，将这个 n 方体的各维表示成 $i = 1, 2, \dots, n$ ，其中第 i 维对应于结点地址中的第 $i - 1$ 位。

设 $v = v_{n-1}v_{n-2}\dots v_1v_0$ 是路径中的任一结点，路径可以根据以下算法唯一地确定：

① 计算方向位 $r_i = s_{i-1} \oplus d_{i-1}$ ，其中 $i = 1, 2, \dots, n$ 。

令 $v = s$ ， $i = 1$ ，反复执行以下步骤：

② 如果 $r_i = 1$ ，则从当前结点 v 寻径到下一结点；否则，就跳过这一步。

③ $i \leftarrow i + 1$ 。如果 $i \leq n$ ，则转第②步，否则退出。

9.5.3 流控制策略

第 85 页

例9.4 假设有一个N=16个结点的4方体，每个结点的二进制编码如图所示。请寻找一条从结点0110到1101的距离最短的路径。

解 $s=0110$, $d=1101$

第1步：计算方向位 $(r_4 r_3 r_2 r_1) =$

$0110 \oplus 1101 = 1011$, 令 $v=s=0110$, $i=1$

第2步： $r_1 = 1$, 所以从 $v=0110$ 寻径到

$v \oplus 2^0 = 0110 \oplus 0001 = 0111$, $i=i+1=2$

第3步： $r_2 = 1$, 所以从 $v = 0111$ 寻径到

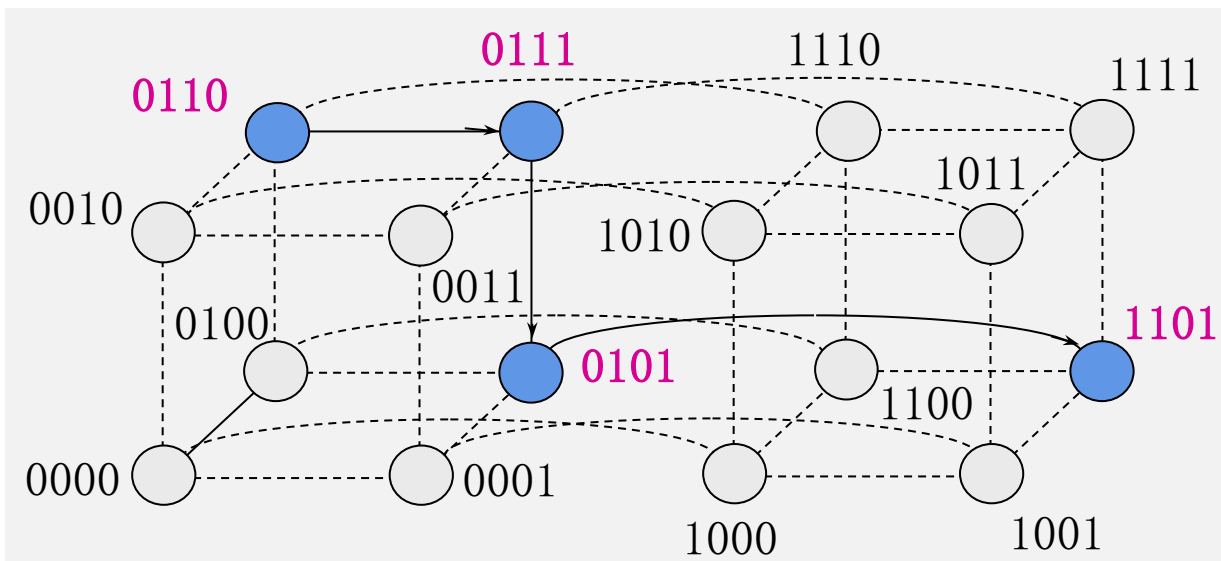
$v \oplus 2^{2-1} = 0111 \oplus 0010 = 0101$, $i=i+1=3$

第4步： $r_3 = 0$, 所以跳过一步

$i=i+1=4$

第5步： $r_4 = 1$, 所以从 $v = 0101$ 寻径到 $v \oplus 2^{4-1} = 0101 \oplus 1000 = 1101$, 结束。

因此路径为： $0110 \rightarrow 0111 \rightarrow 0101 \rightarrow 1101$



9.5.3 流控制策略

第 86 页

1. 多计算机网络中会出现以下4种通信模式：

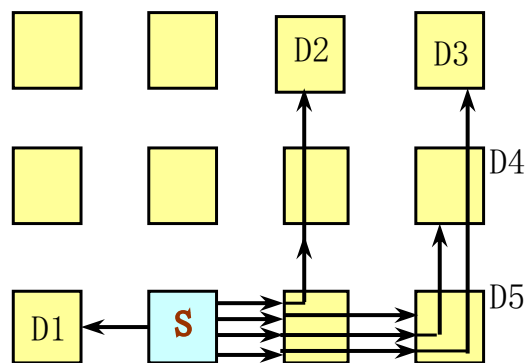
- **单播**：对应于一对一的通信情况，即一个源结点发送消息到一个目的结点。
 - **选播**：对应于一到多的通信情况，即一个源结点发送同一消息到多个目的结点。
 - **广播**：对应于一到全体的通信情况，即一个源结点发送同一消息到全部结点。
 - **会议**：对应于多到多的通信情况。
-
- **通道流量**和**通信时延**是常用的两个参数。
 - 通道流量可用传输有关消息所使用的通道数来表示。
 - 通信时延则用包的最大传输时间来表示。
 - 优化的寻径网络应能以最小流量和最小时延实现相关的通信模式。

9.5.3 流控制策略

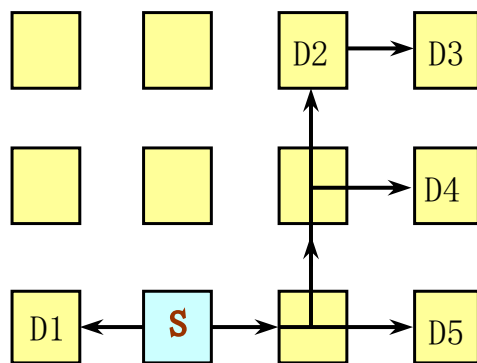
第 87 页

2. 以网格网络为例，讨论选播和广播。

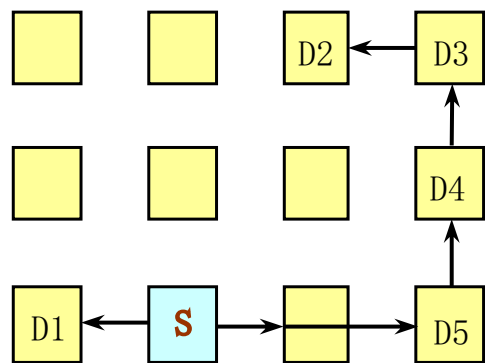
在 3×4 网格上实现选播寻径：源结点：S，传送一个包到标号为 D_i 的5个目的结点（ $i = 1 \dots 5$ ）



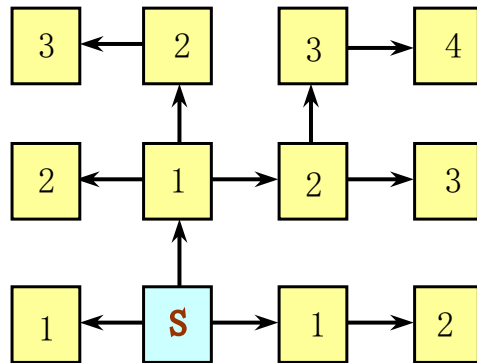
(a) 5次单播，流量为13，距离为4



(b) 流量为7，距离为4的选播方式



(c) 流量为6，距离5的选播方式



(d) 通过树结构广播给所有结点

- 图(a)：目的结点为5个的选播可以用5次单播来实现X-Y寻径的流量需要用 $1 + 3 + 4 + 3 + 2 = 13$ 条通道。

到D3的路径最长，所以时延是4。

- 图(b)和(c)：给出了两种选播寻径模式，流量分别为7和6。

✓在虫蚀网络中，用图(c)的选播寻径模式比较好

✓在存储转发网络中，用图(b)的寻径模式比较好，时延较短。

- 图(d)：使用一棵4层的生成树可以把一个包从结点S广播到所有的网络结点产生的时延和流量最小

- 1、互连函数：表示方法、恒等函数、交换函数、均匀洗牌函数、逆均匀洗牌函数、碟式函数、反位序函数、移数函数、PM2I函数、阵列计算机等
- 2、互连网络的结构参数与性能指标：网络规模、结点度、网络直径、等分宽度、通信时延、网络时延、端口带宽、等分带宽等
- 3、静态互连网络：线性阵列、环和带弦环、全连接网络、循环移数网络、树形和星形、胖树形、网格形和环网形、超立方体等
- 4、动态互连网络：总线、开关、级控制、单元控制、部分级控制、交换等
- 5、消息传递机制：路由、消息、包、寻径方式、线路交换、存储转发、虚拟直通、虫蚀方式、避免死锁、包冲突解决等。

每章作业

第 89 页

第9章习题9.9、9.10、9.11、9.12