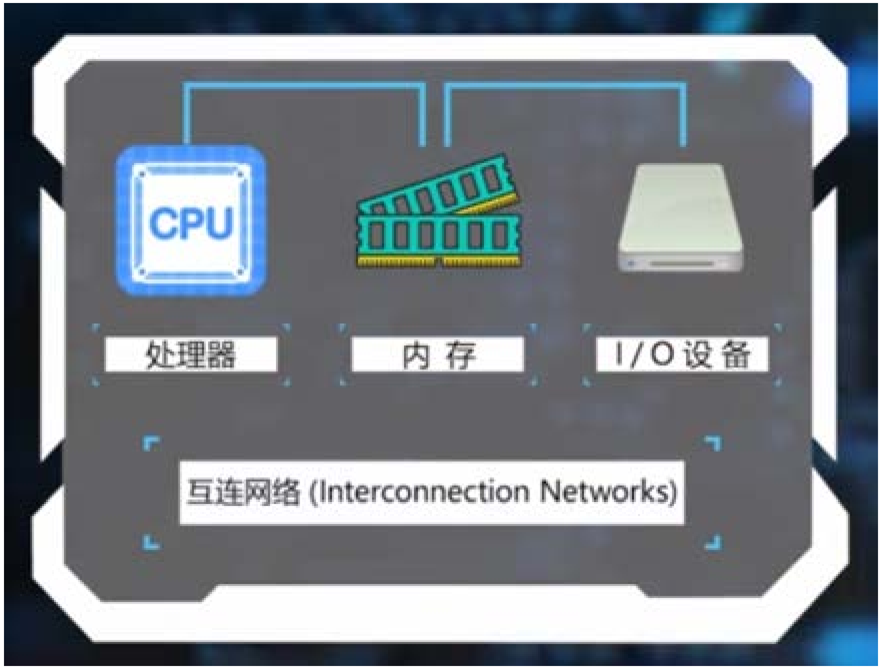
计算机系统结构

第9章 互连网络

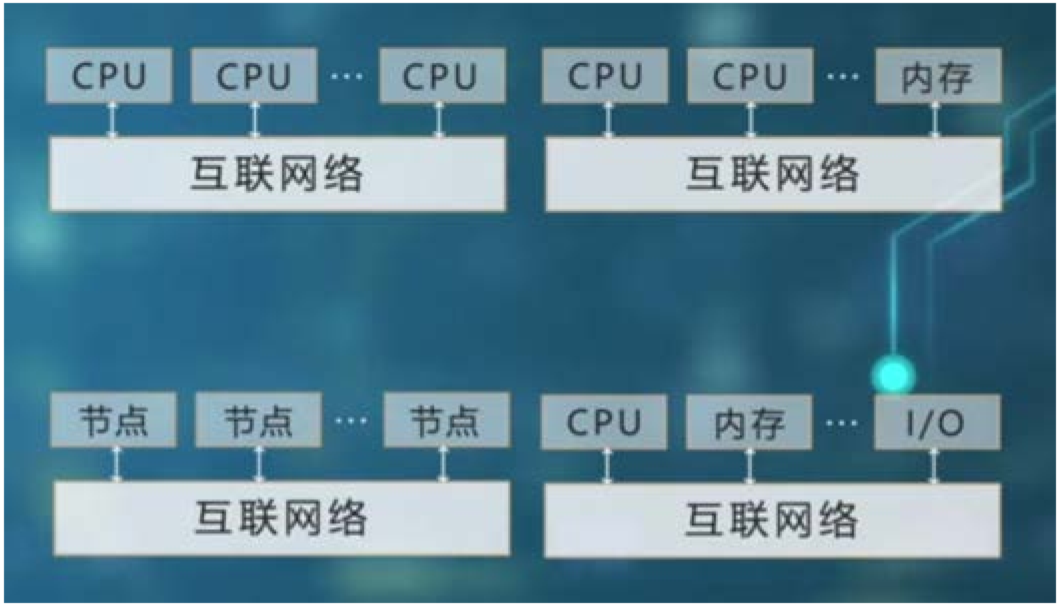
目录

9.1 互连网络基本概念



互连网络连接

- CPU内多核之间
- CPU和内存之间
- 计算机节点之间
- CPU处理器之间
- 内存和内存之间
- 网络和网络之间



互连网络连接

互连网络连接

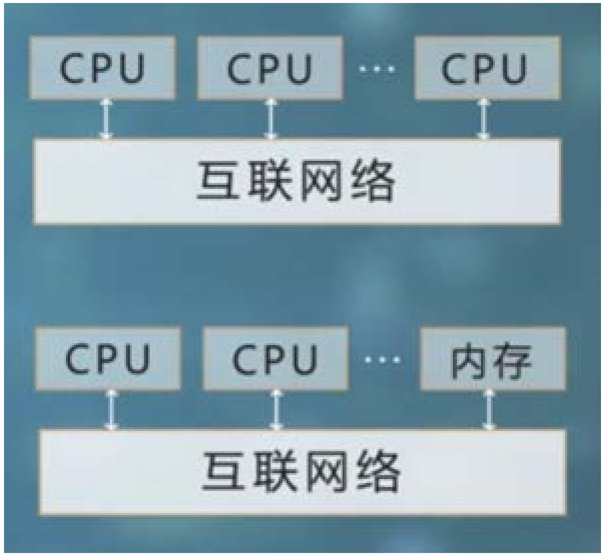
主要设计目标：

在最少传输延迟（成本，能耗等）约束内， 传输尽可能多的数据，避免成为系统的瓶颈。

互连网络连接

》高速互连网络

- CPU与CPU间
- CPU与内存间



网络延迟时间：

几十个或几百个时钟周期之内（纳秒量级）

9.1 互连网络基本概念

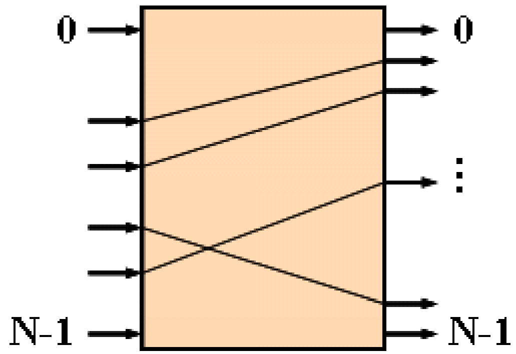
互连网络（Interconnection Network）由网络元件按照一定的拓扑结构和控制方式构成的网络，用来实现计算机系统中部件之间的相互连接。

- 三大要素：

  网络元件、互连结构、控制方式
- 结点：处理器、存储模块或其它设备。
- 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（Mapping）。
  - ❑ 互连结构 是静态连接拓扑
  - ❑ 控制方式 是基于静态拓扑结构的动态传输机制
- SIMD计算机和MIMD计算机的关键组成部分。

互连网络(IN) 的特性

*特性1：同时实现多个端口对的互连及通信。



*特性2：有多种并行端口对的互连方式.

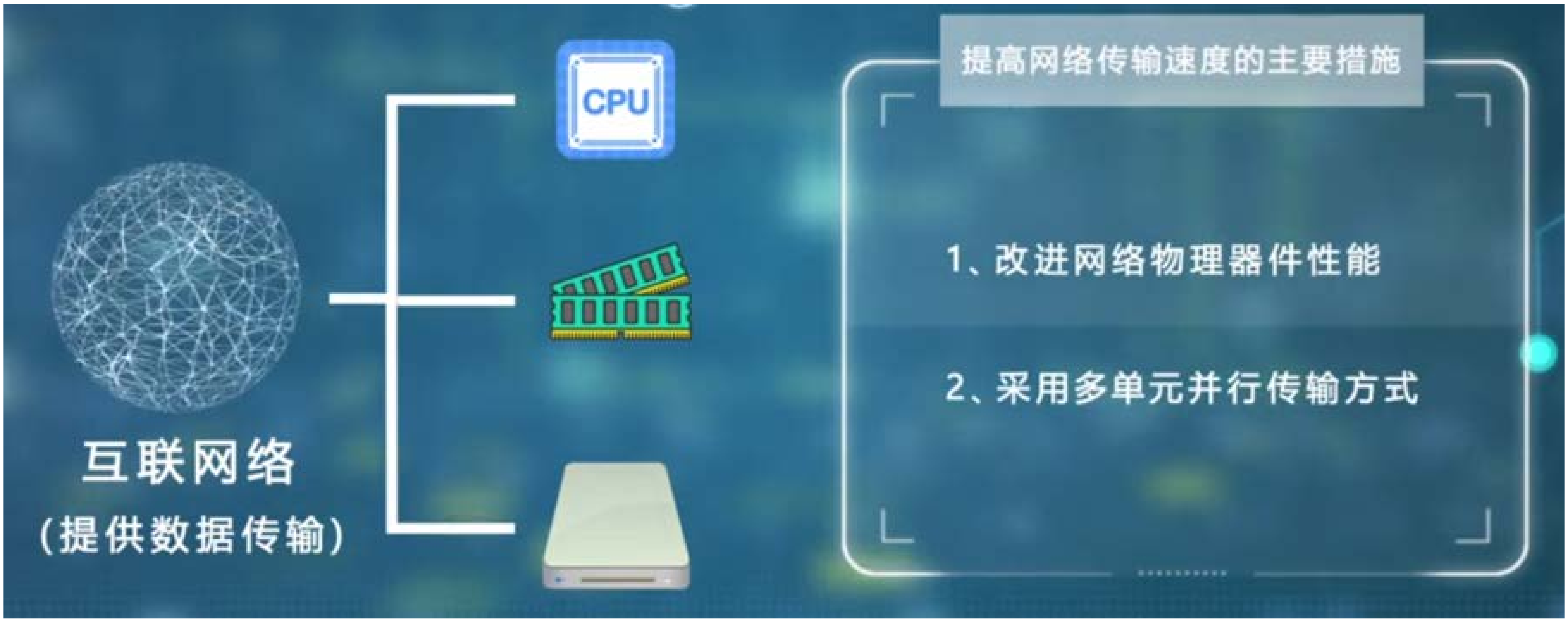
理论上有**N!种**端口对互连**排列**方式

*互连网络与总线比较：

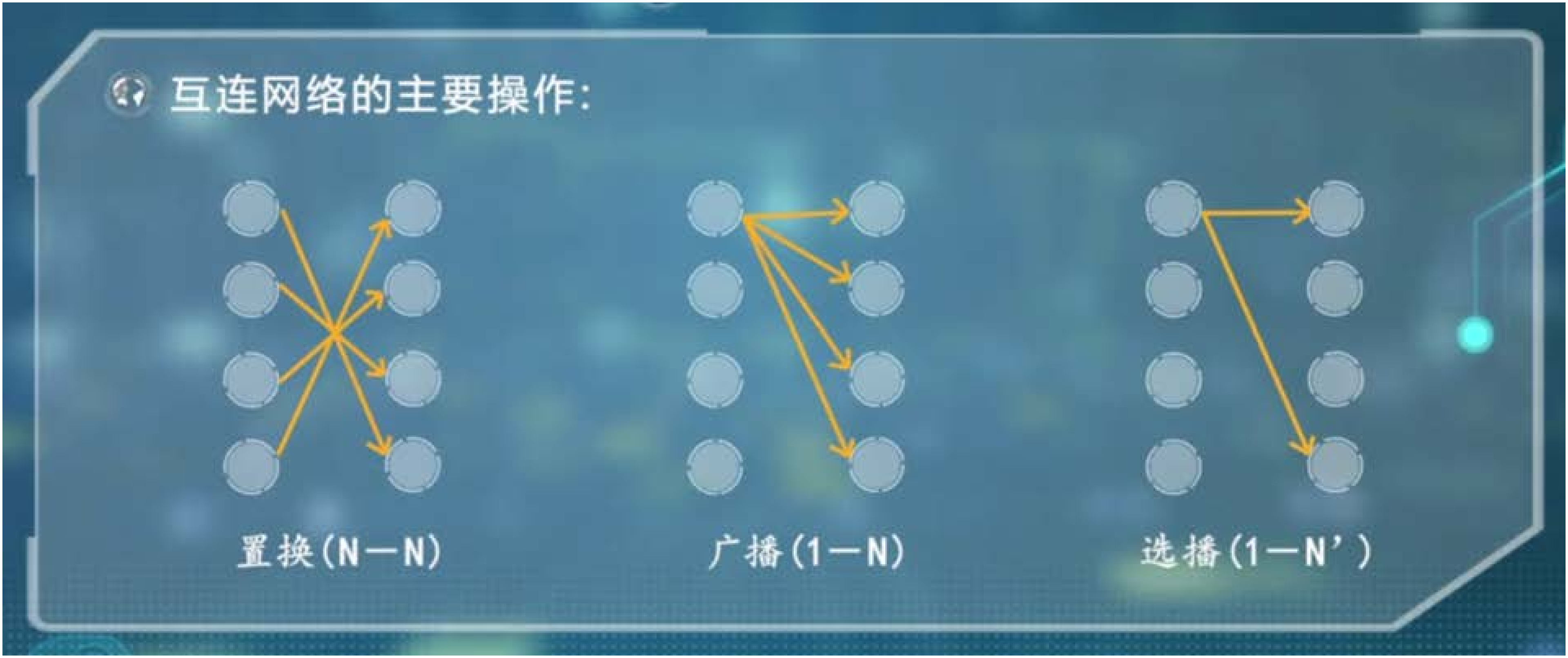
互连网络 ：强调多个节点对之间的互连及通信。

总　　线 ：多个设备或单元共享公共通道(分时)

ICN的目的与作用



ICN的目的与作用--互联网络的主要操作



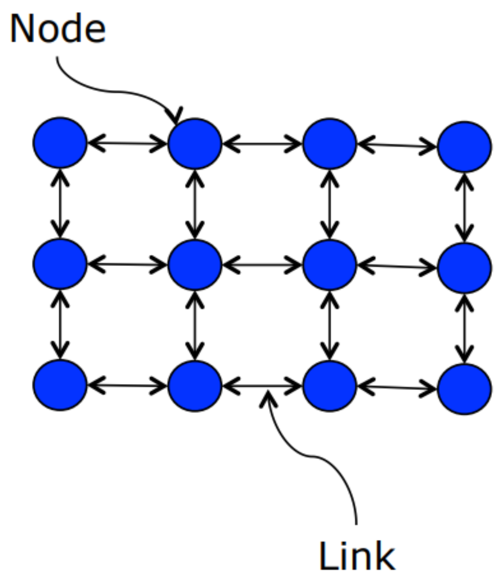
N个结点之间
同时传播消息

一个结点向N个结点
同时传播消息

一个结点向特定
多个结点传播消息

9.1.2 互连网络的结构参数

1. 网络通常是用有向边或无向边连接有限个结点（Node）的图来表示。



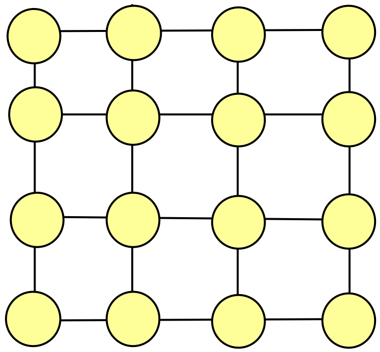
2. 互连网络的主要特性参数有：
   - 网络规模N：网络中结点的个数。
     表示该网络所能连接的部件的数量。
   - 结点度d：与结点相连接的边数（通道数），包括入度和出度。
     - 进入结点的边数叫入度。
     - 从结点出来的边数叫出度。

- 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。

$$D = \max_{i,j} D_{ij}$$

网络直径应当尽可能地小。

- 等分宽度b（Bisection Width）：把由N个结点构成的网络切成结点数(近似)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×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。

9.1.3 互连网络的性能指标

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

通信时延=软件开销+通道时延+选路时延+竞争时延

- 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。

通信时延=软件开销+通道时延+选路时延+竞争时延

2. 网络时延

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（Aggregate Bandwidth）

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：（40MB/s×512）/2＝10.24GB/s

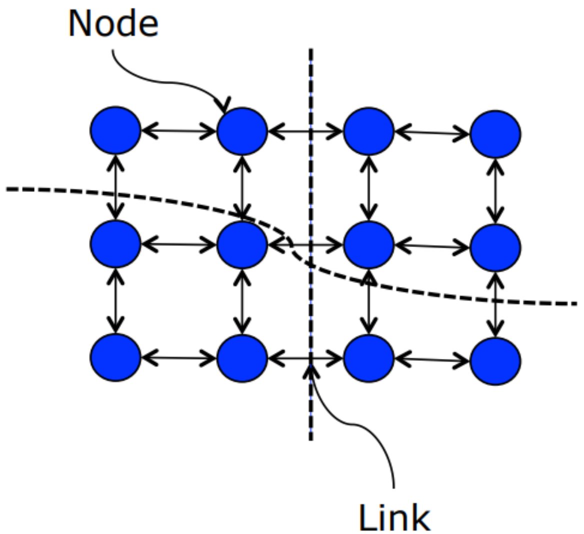
5. 等分带宽（Bisection Bandwidth）

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。

等分宽度b　　单位是通道数（link）

线等分宽度　　单位是位（bit）

等分带宽　　单位是bps（bit per second）



9.2 互连函数

9.2.1 互连函数

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。（置换函数或排列函数）

变量x：输入（设x=0，1，…，N—1）

函数f(x)：输出

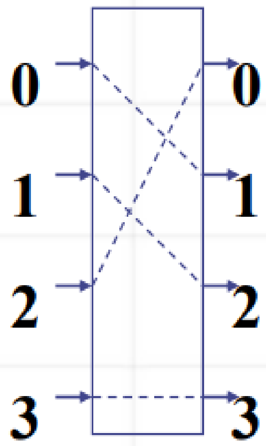
通过数学表达式建立输入端号与输出端号的连接关系。即在互连函数f的作用下，输入端x连接到输出端f(x)。

9.2 互连函数

互连函数有多种表示方式，如下例所示：

$$\begin{cases} f(0)=1 \\ f(1)=2 \\ f(2)=0 \\ f(3)=3 \end{cases}$$

a.枚举法



b.开关状态图

$$f=\begin{vmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 3 \end{vmatrix}$$

c.列表法

$$f=(0,1,2)(3)$$

d.循环函数

一个网络通过开关切换可以形成多个映射关系，所以要用“互连函数族”来定义一个网络。

9.2 互连函数

- 互连函数f(x)有时可以采用循环表示

  即： $(x_0\ x_1\ x_2\ \ldots\ x_{j-1})$

  表示： $f(x_0)=x_1$， $f(x_1)=x_2$， …， $f(x_{j-1})=x_0$

  j称为该**循环的长度**。

- 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：

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

9.2.2 几种基本的互连函数

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。

$$I（x_{n-1}x_{n-2}\ldots x_1x_0）=x_{n-1}x_{n-2}\ldots x_1x_0$$

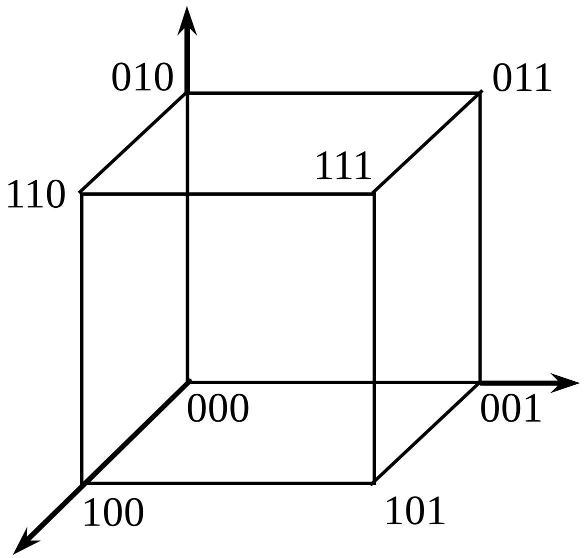
2. 交换函数

   - 交换函数（Exchange）：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。

$$E\left(x_{n-1}x_{n-2}\cdots x_{k+1}x_k x_{k-1}\cdots x_1x_0\right)=x_{n-1}x_{n-2}\cdots x_{k+1}\overline{x}_k x_{k-1}\cdots x_1x_0$$

9.2 互连函数

- 交换函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。
- 它共有$n=\log_2 N$种互连函数。（N为结点个数）
- 当$N=8$时，$n=3$，可得到常用的立方体互连函数：



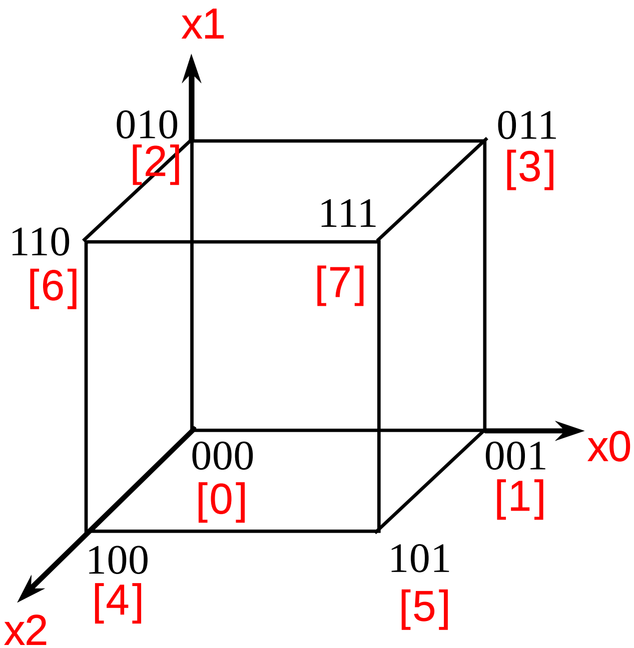
8结点的立方体网络

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

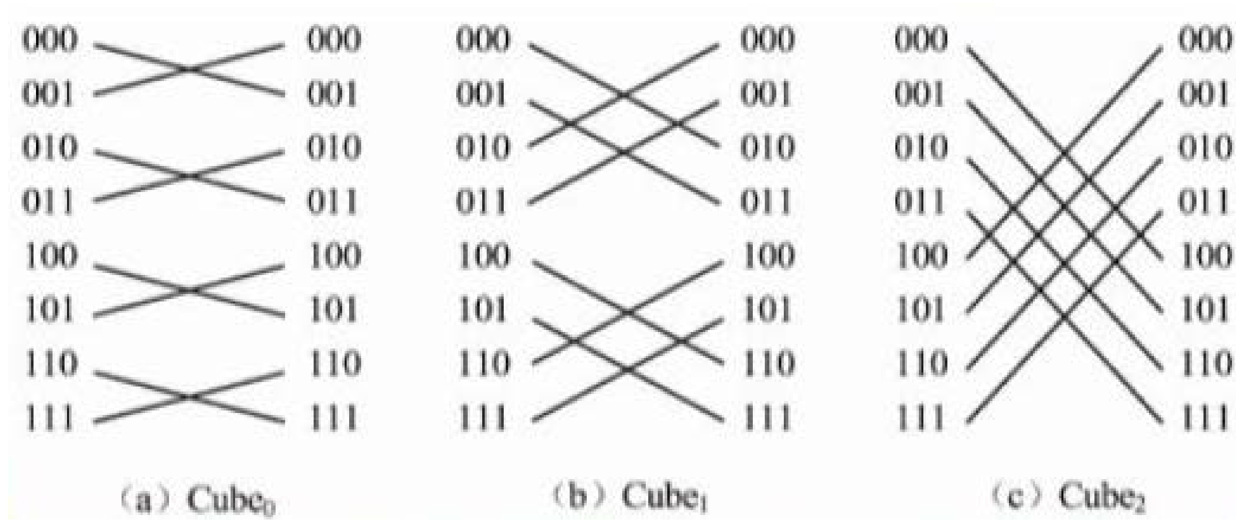
9.2 互连函数

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

- 变换图形



8结点的立方体网络



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

(0 1) (2 3) (4 5) (6 7) (0 2)(1 3) (4 6) (5 7) (0 4)(1 5) (2 6) (3 7)

N=8 的立方体交换函数

3. 均匀洗牌函数

- **均匀洗牌函数（shuffle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）

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

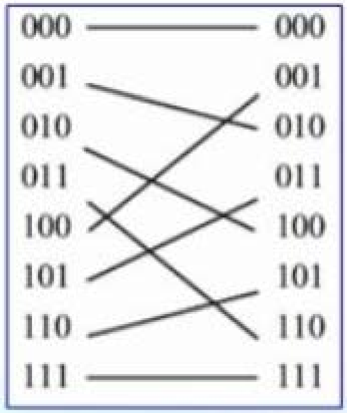
  - 即把输入端的二进制编号循环左移一位。

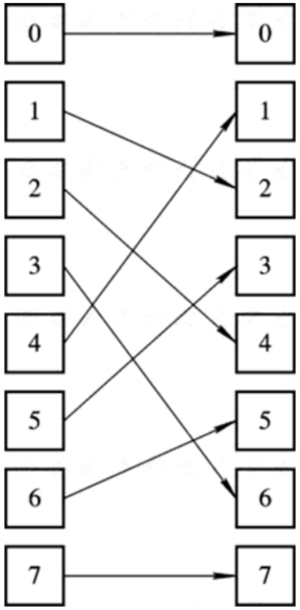
0 (000)-> 0 (000)　　1 (001)->2 (010)

2 (010)-> 4 (100)　　3 (011)->6 (110)

4 (100)-> 1 (001)　　5 (101)->3 (011)

6 (110)->5 (101)　　7 (111) ->7 (111)





8个处理单元间的全混连接

子函数和超函数

- 互连函数（设为s）的第k个子函数：把s作用于输入端的二进制编号的低k位。
- 互连函数（设为s）的第k个超函数：把s作用于输入端的二进制编号的高k位。

例如：对于均匀洗牌函数

第k个子函数：

$$\sigma_{(k)}(x_{n-1}\cdots x_k \mid x_{k-1}x_{k-2}\cdots x_0)=x_{n-1}\cdots x_k \mid x_{k-2}\cdots x_0x_{k-1}$$

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：

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

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

9.2 互连函数

- 逆函数

  对于任意一种函数f(x)，如果存在g(x)，使得

  g(f(x))=I(x)　　　➔ f(x)×g(x)=I(x) （不准确）

  反函数定义：y=f(x)，$f^{-1}(y)=x$

  则称g(x)是f(x)的逆函数，记为$f^{-1}(x)$。

  $f^{-1}(x)= g(x)$

- 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

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

  - 逆均匀洗牌是均匀洗牌的逆函数

9.2 互连函数

当N=8时，有：

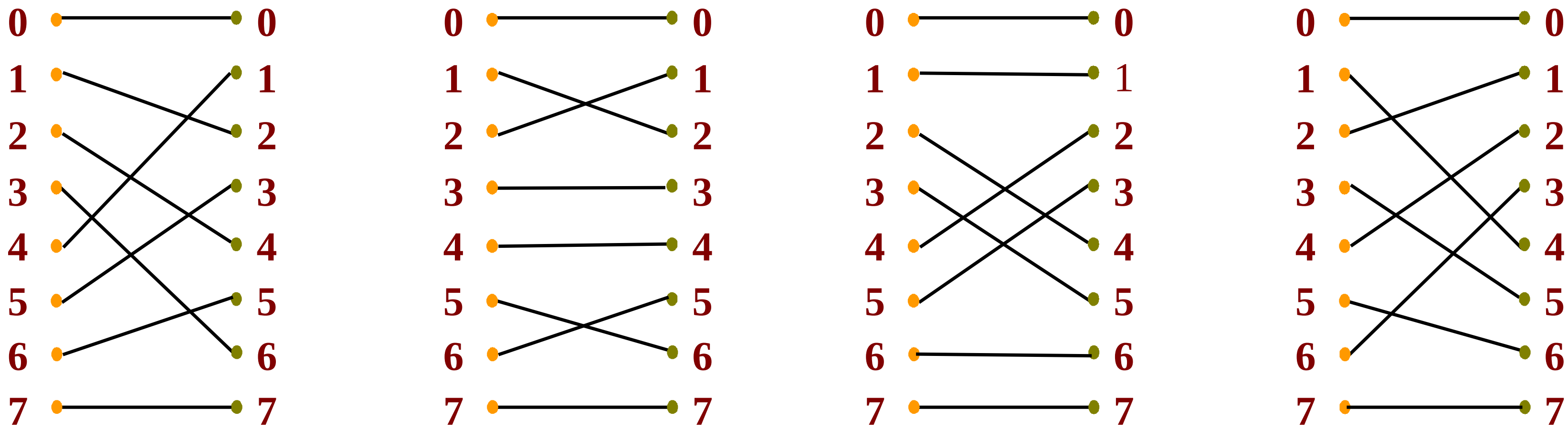
$$\sigma(x_2x_1x_0) = x_1x_0x_2$$

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

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

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

- N=8 的均匀洗牌和逆均匀洗牌函数



(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}

N=8 的均匀洗牌函数

9.2 互连函数

- 当N=8时，有：

$$\sigma(x_2x_1x_0) = x_1x_0x_2$$

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

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

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

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

性质2： $\sigma_{(1)}(X) = \sigma^{(1)}(X) = X$

特性：

- 当经过n次全混后，全部N个处理单元便又恢复到最初的排列次序。
- 在多次全混的过程中，除了编号为全“0”和全“1”的处理单元外，各个处理单元都遇到了与其他多个处理单元连接的机会(不是所有单元)。

混洗交换单级网络

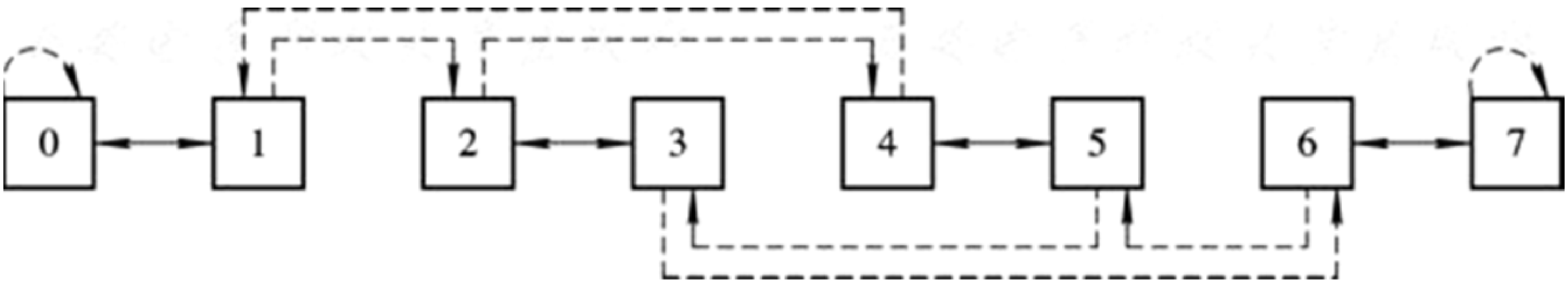
选讲

混洗交换单级网络=全混(Perfect Shuffle)+交换(Exchange)。

交换(Exchange)

引入原因：

由于单纯的全混互连网络不能实现二进制编号为全“0”和全“1”的处理单元与其他处理单元的连接，因此**还需增加Cube0交换函数**



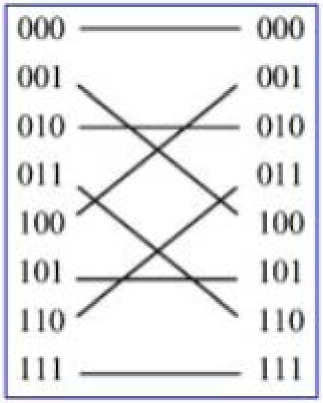
N=8时全混交换互联网络连接图

实线表示交换，虚线表示全混。

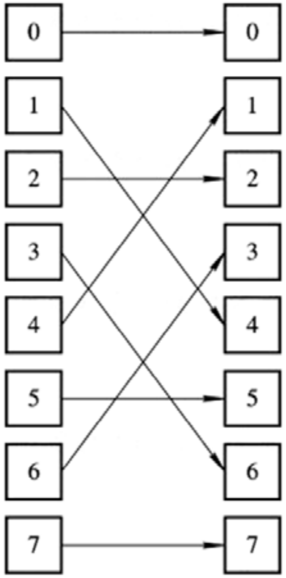
特点：在混洗交换网络中，最远的两个入、出端号是全“0”和全“1”，它们的连接，需要经过n次交换和n-1次混洗，所以混洗交换网络的最大距离为2n-1。

4. 碟式函数 (β)

➤ 蝶式（Butterfly）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。



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



0 (000)-> 0 (000)	**1 (001)->4 (100)**
2 (010)-> 2 (010)	**3 (011)->6 (110)**
4 (100)-> 1 (001)	**5 (101)->5 (101)**
6 (110)-> 3 (011)	**7 (111) ->7 (111)**

(0)(2)(1 4) (3 6)(5)(7)

8个处理单元间的蝶形单级互连

- **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

$$\beta_{(k)}(x_{n-1}\cdots x_kx_{k-1}x_{k-2}\cdots x_1x_0)=x_{n-1}\cdots x_kx_0x_{k-2}\cdots x_1x_{k-1}$$

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

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

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。

- 蝶式变换与交换变换的多级组合是构成多级立方体网络的基础。

9.2 互连函数

- 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。

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

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

- 第k个子函数

$$\rho_{(k)}(x_{n-1}\cdots x_kx_{k-1}x_{k-2}\cdots x_1x_0)=x_{n-1}\cdots x_kx_0x_1\cdots x_{k-2}x_{k-1}$$

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。

- 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。

9.1 互连函数

- 下列等式成立

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

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

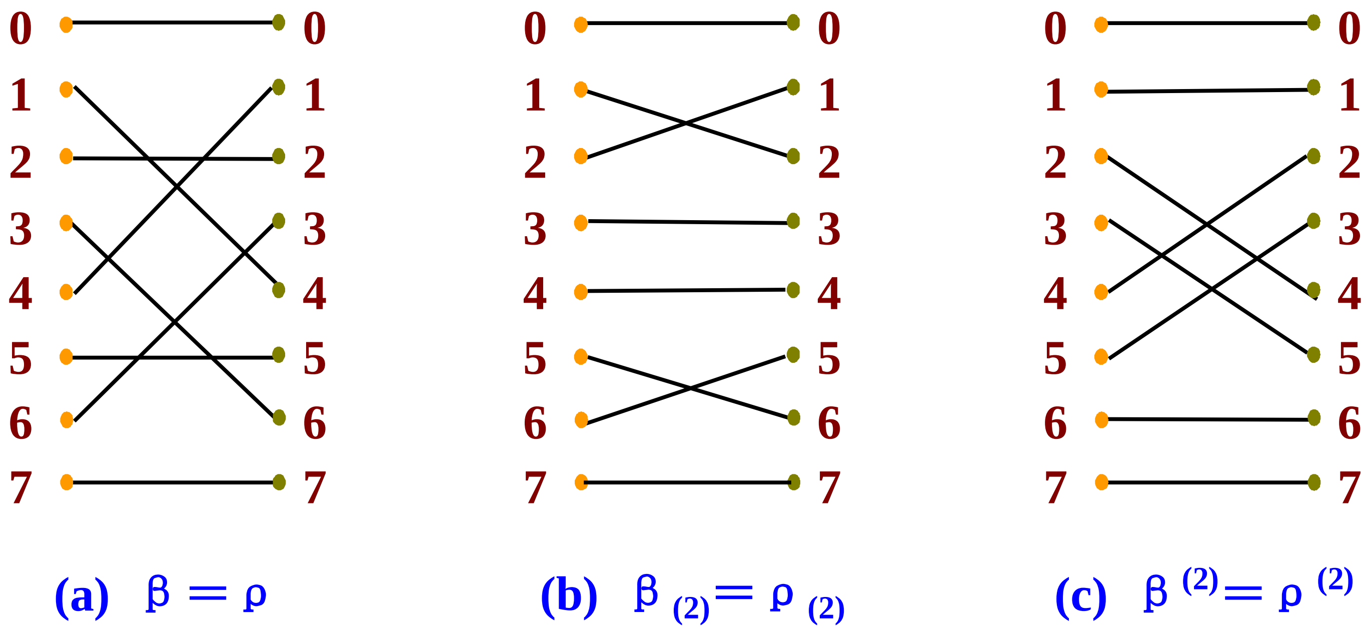
- 当N=8时，有：

$$\rho(x_2x_1x_0)=x_0x_1x_2$$

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

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

9.1 互连函数

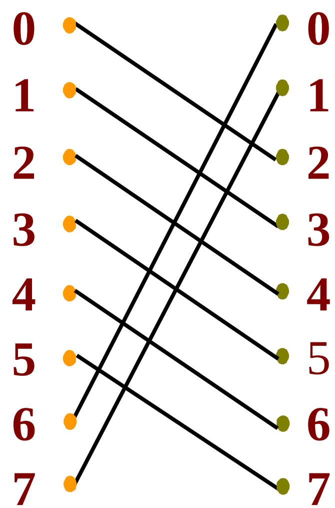


N=8 的蝶式函数和反位序函数

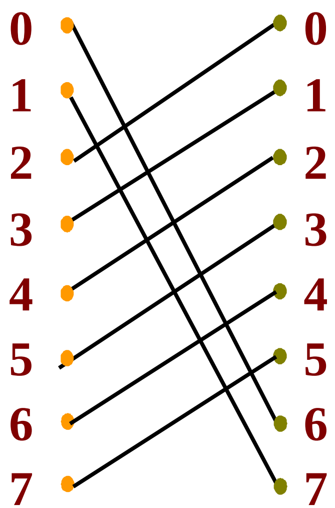
6. 移数函数

- **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

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



(a) 左移移数函数 k＝2



(b) 右移移数函数 k＝2

7. 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。
- 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

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

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

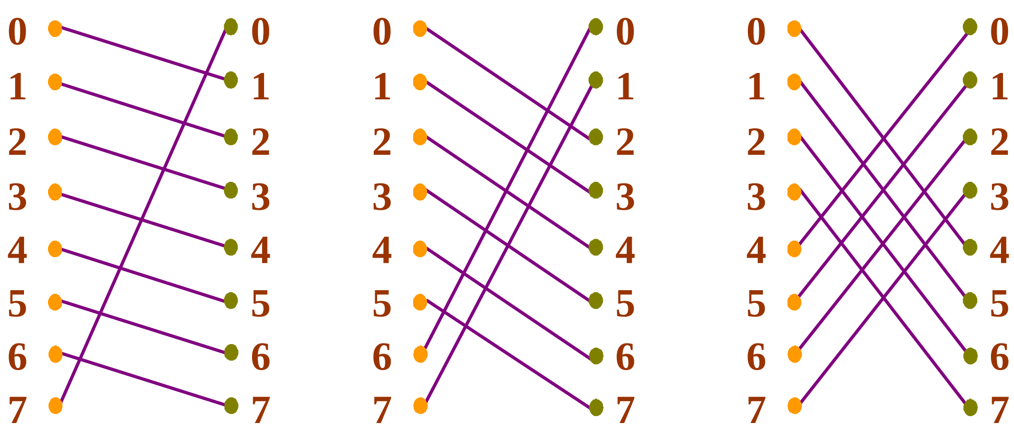
其中：

$0 \leqslant x \leqslant N-1$，$0 \leqslant i \leqslant n-1$，$n = \log_2 N$，N为结点数。

- PM2I互连网络共有2n个互连函数。

9.1 互连函数

- 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）



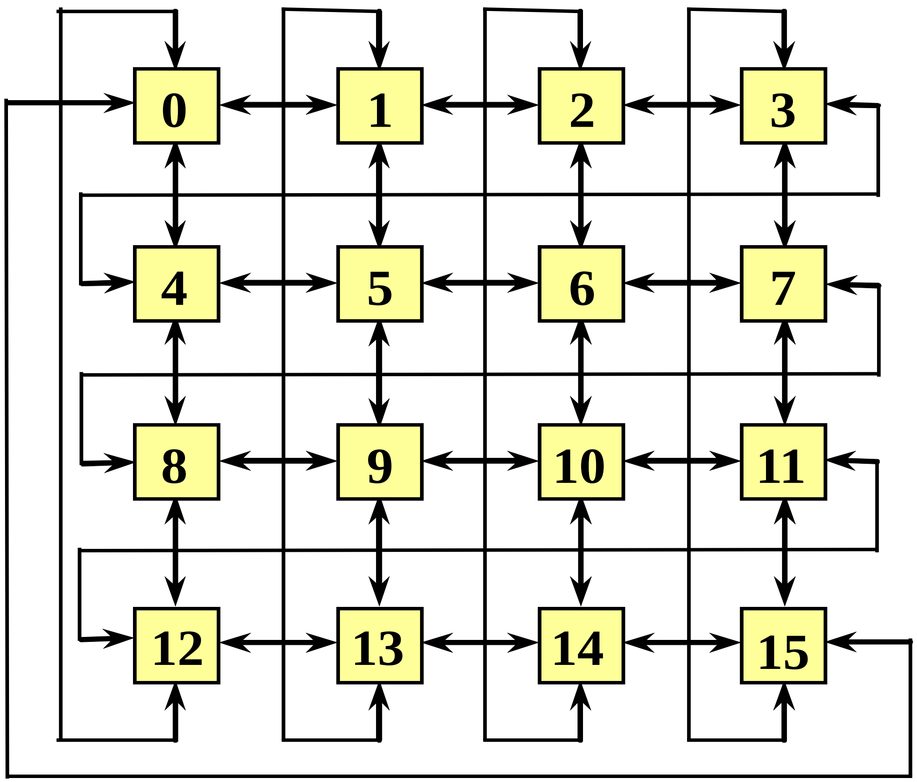
(a) $PM2_{+0}$　(b) $PM2_{+1}$　(c) $PM2_{+2}$

N=8 的PM2I函数

9.2 互连函数

- 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 。



用移数函数构成ILLIAC Ⅳ 阵列机的互连网络

例题解析

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）σ

（5）σ（σ）

时，分别给出与第13号处理器所连接的处理器号。

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

令$PM2_{+3}$（j）= j+2^3 mod 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 mod 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 mod 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。

9.2 互连函数

（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。

9.3 静态互连网络

互连网络通常可以分为两大类：

1. 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。

2. 动态互连网络

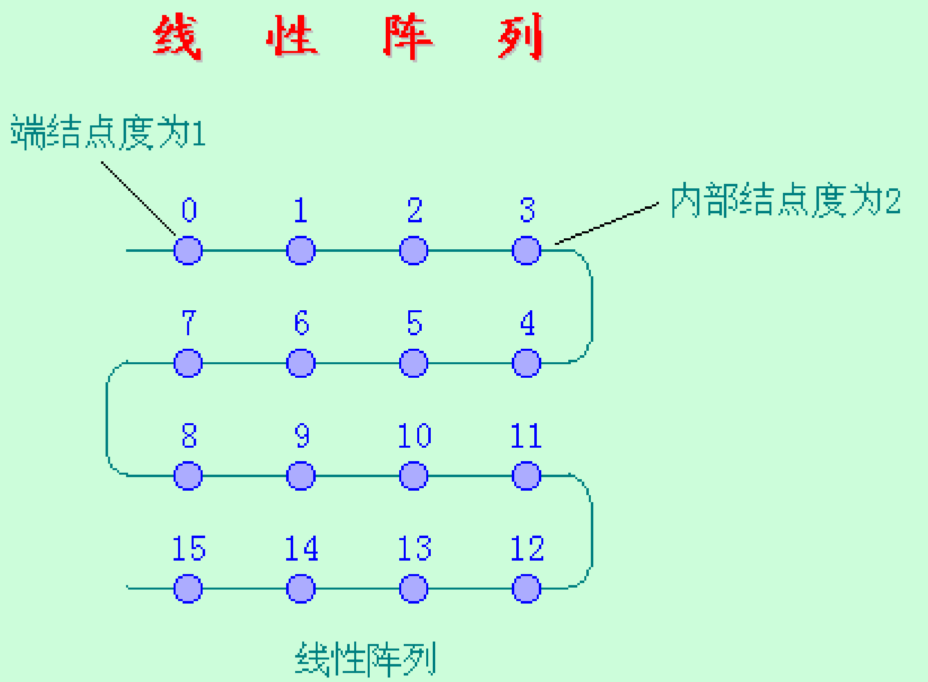
   由交换开关构成、可按运行程序的要求动态地改变连接状态的网络。

下面介绍几种静态互连网络。

（其中：N表示结点的个数）

9.3.1 1、2维互联函数

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。



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

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

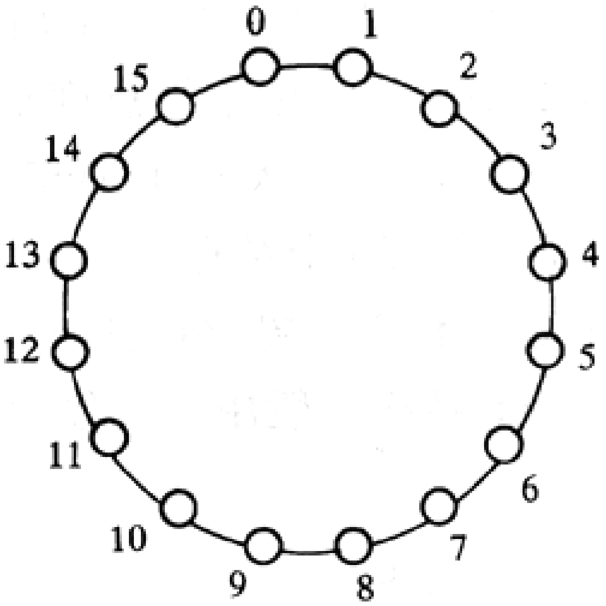
- 端结点的度：1
- 其余结点的度：2
- 链路数：N-1
- 直径：N-1
- 等分宽度b=1
- 不对称（？）

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。

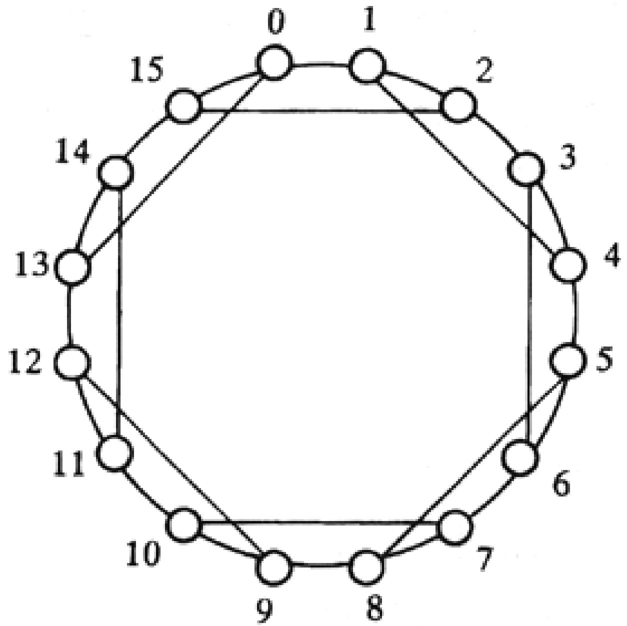
- 对称
- 结点的度：2
- 链路数：N
- 双向环的直径：⌊N/2⌋
- 单向环的直径：N
- 环的等分宽度b=2



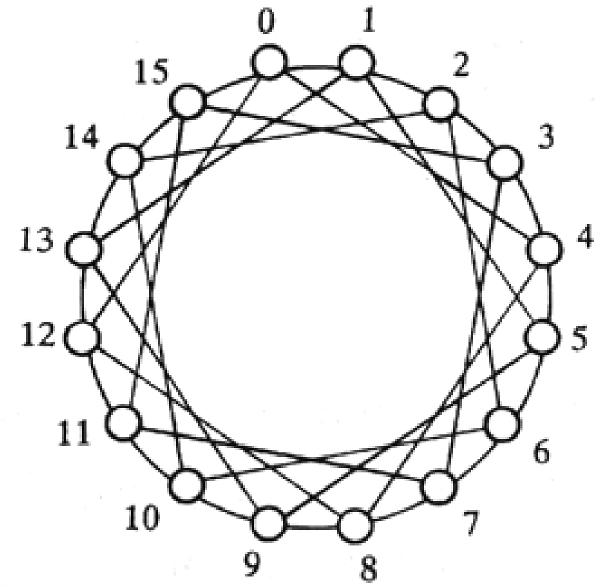
(b)环

- 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



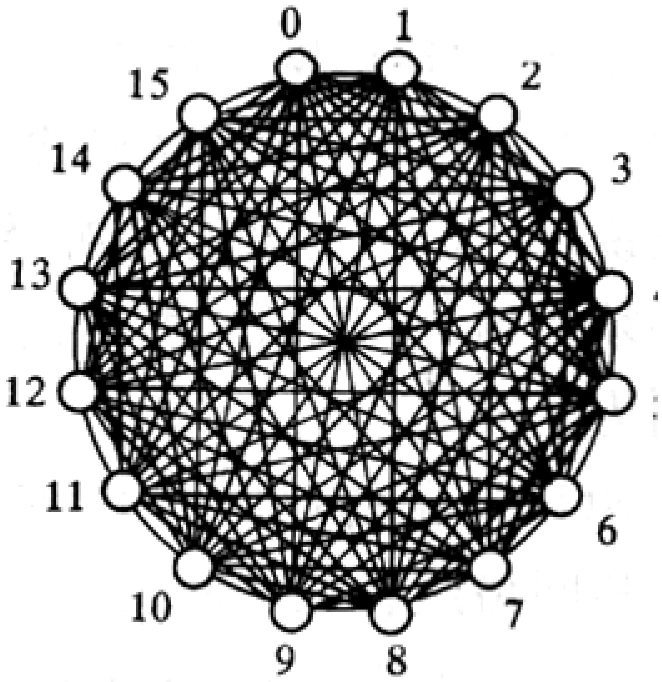
(d)度为 4 的带弦环(与 Illiac 网相同)

图为（16个结点）带弦双向环

- 结点度为3：链路数为24，直径4，度为3，不对称，等分宽度为4。
- 结点度为4：链路数为32，直径3，度为4，对称，等分宽度为10。

9.3 静态互连网络

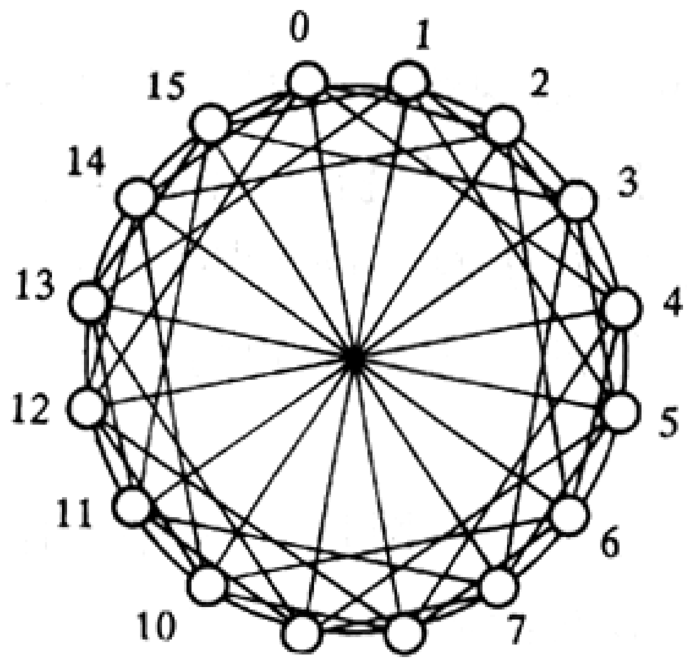
- 全连接网络
  - 结点度：N-1=15
  - 链路数：N(N-1)/2
  - 直径最短，为1
  - 等分宽度**b=(N/2)*(N/2)=64**



(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加链而构成。



N=16　结点度：7　直径：2

(e)循环移数网络

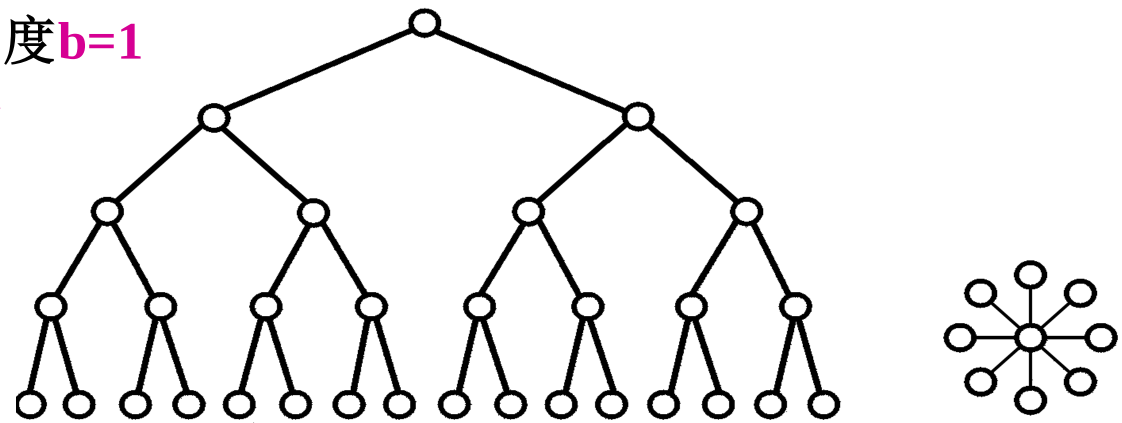
- 一般地，如果｜j−i｜=2^r（r=0,1,2,…,n-1，n=log2N），则结点i与结点j连接。
  - 结点度：2n−1
  - 直径：n/2
  - 网络规模$N=2^n$

4. 树形（Trees）和星形

- 树形：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。

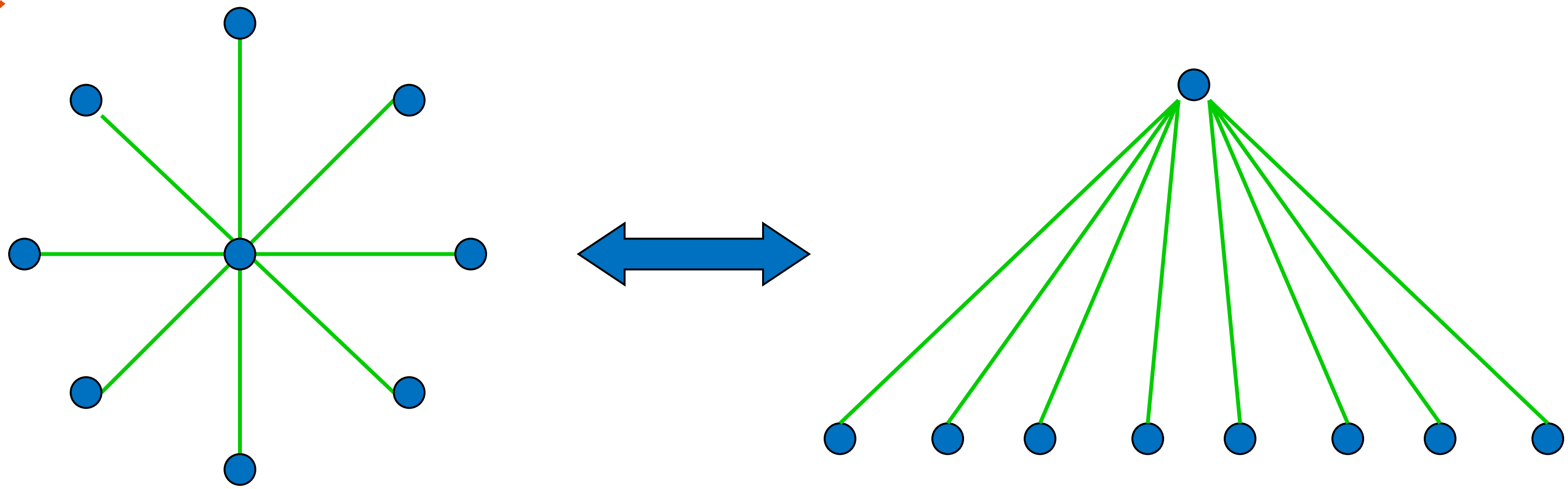
  - 最大结点度：3
  - 直径：2(k-1)
  - 等分宽度b=1
  - 不对称



(a) 二叉树　　(b) 星形

- 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。

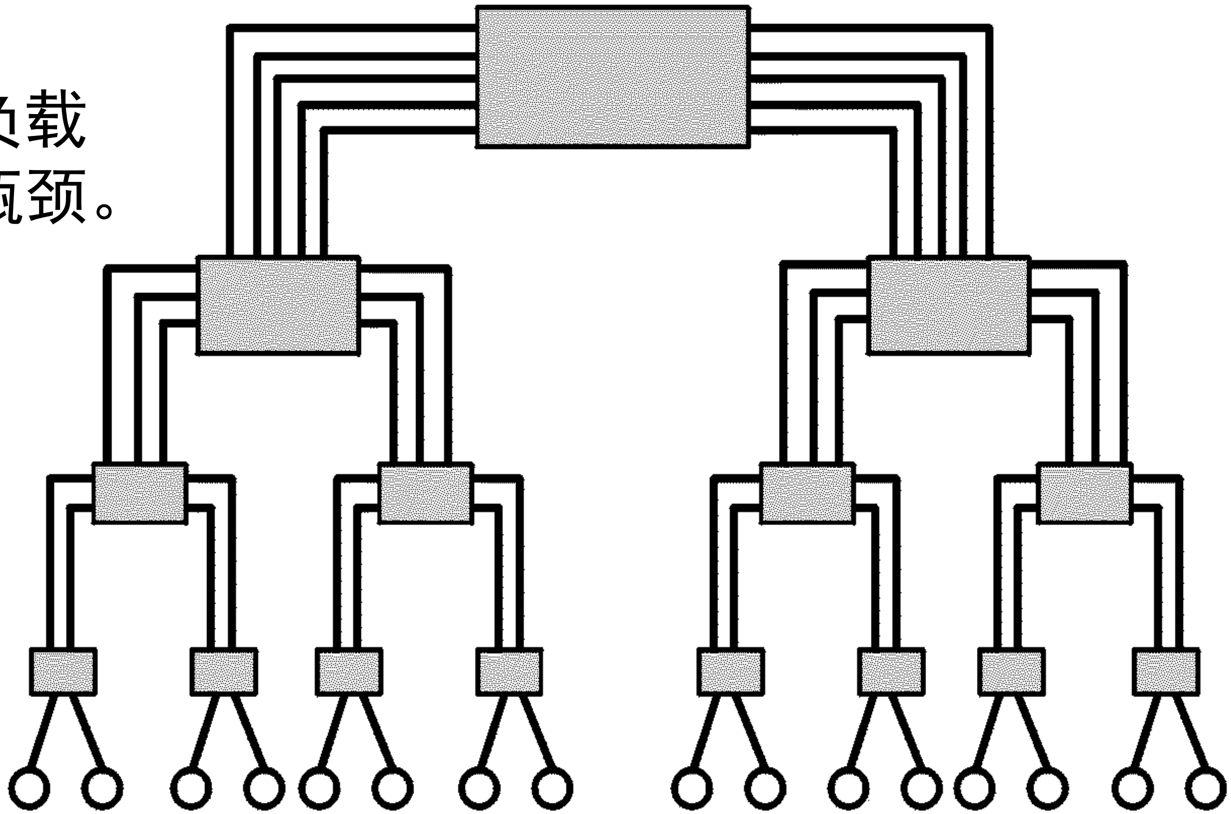
星形



- 星形：实际上是一种二层数（如右图）。
  - 结点度较高，为N－1。
  - N-1条链路。
  - 非对称（ ？ ）
  - 直径较小，是一常数2。等分宽度**b=⌊N/2⌋**
  - 可靠性比较差，只要中心结点出故障，整个系统就会瘫痪。

5. 胖树形（Flat Tree）

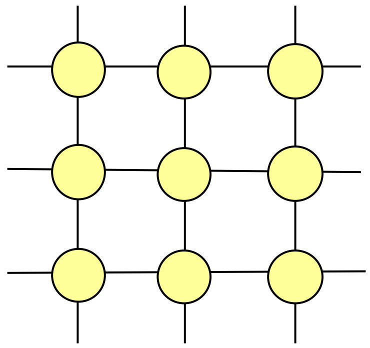
避免根部负载
过重成为瓶颈。



(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×n的2维网格形网络
    - 链路数：2N-2n
    - 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）
    - 非对称
    - 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)。

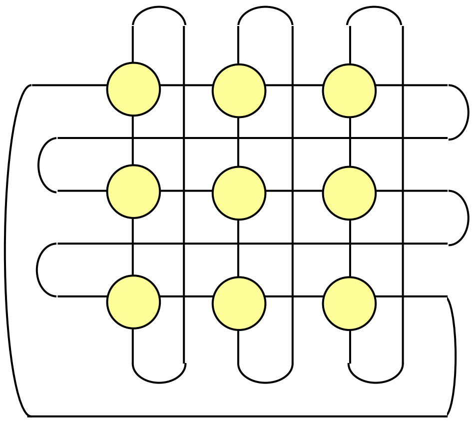


（a）网格形

9.3 静态互连网络

- 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。
  - 网络结构扩充性好，易在VLSI芯片上实现。
  - 一个规模为n×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**

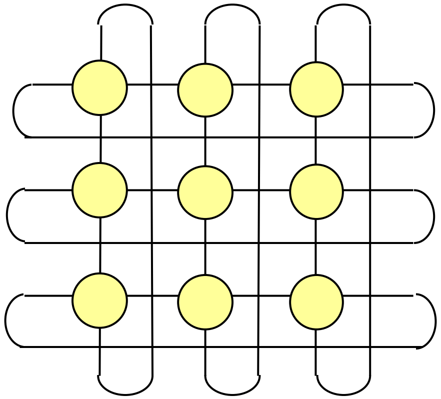


（b）Illiac网

9.3 静态互连网络

➢ 环网形（Torus）

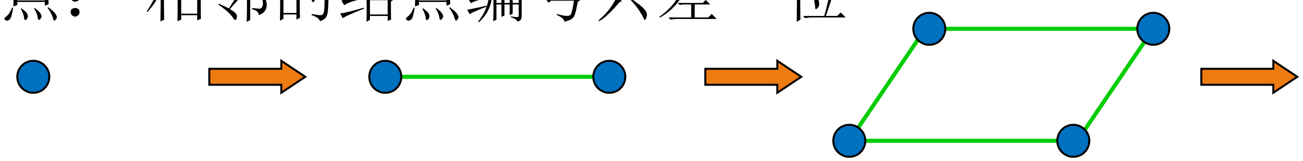
- 可看作是直径更短的另一种网格。
- 把2维网格形网络的每一行的两个端结点连接起来，把每一列的两个端结点也连接起来。
- 将环形和网格形组合在一起，并能向高维扩展。
- 一个n×n的环网形网
  - 结点度：4
  - 网络直径：$2\times\lfloor n/2 \rfloor$
  - 等分宽度**b=2n**



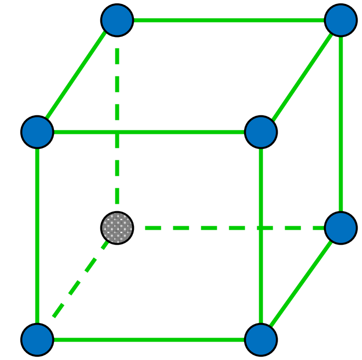
（c）环网形

7. 超立方体（HyperCubic）

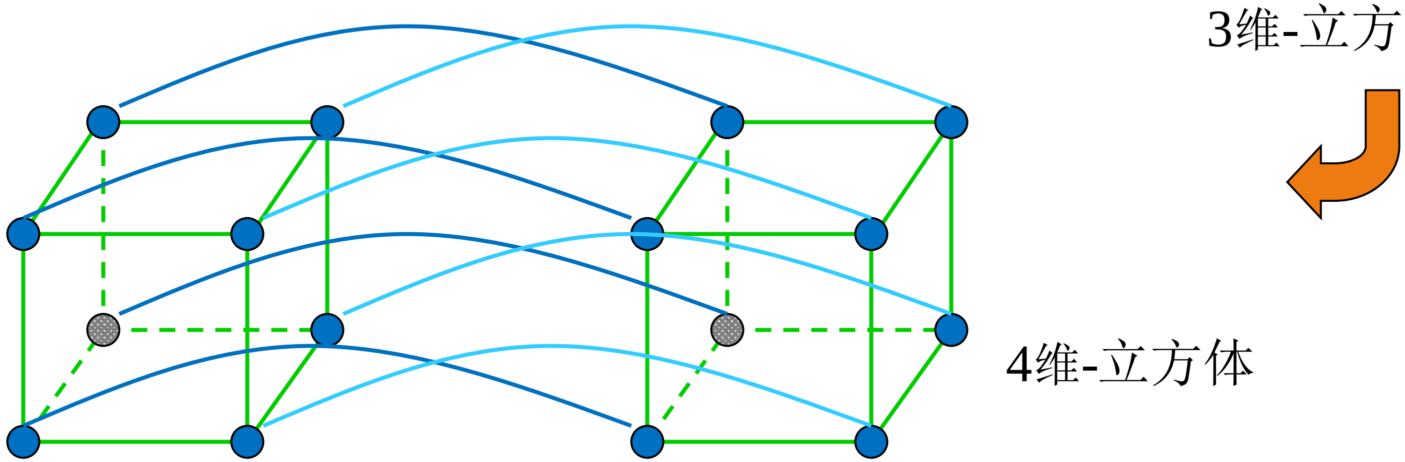
特点： 相邻的结点编号只差一位



0维-立方体　　1维-立方体　　2维-立方体



3维-立方体

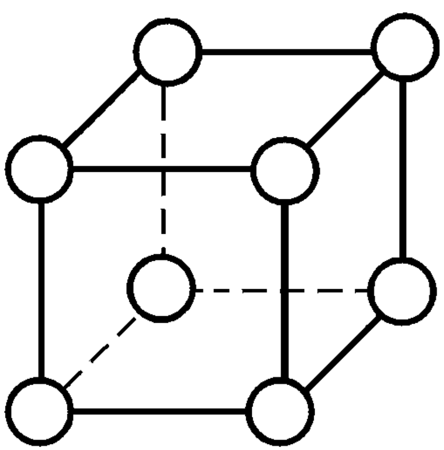


4维-立方体

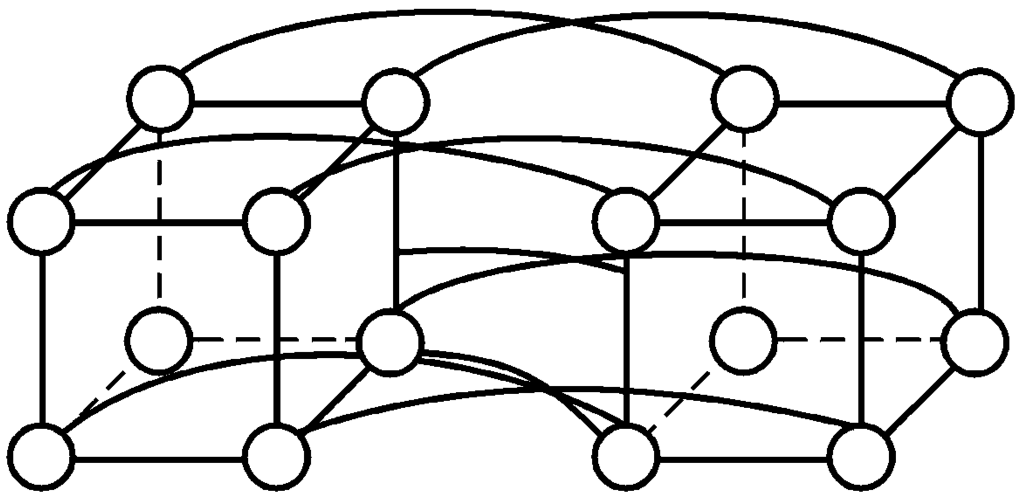
- n-立方体由$N = 2^n$个结点构成。直径为n，结点度为n，对称。结点度随维数线性增加。

7. 超立方体

- 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 。



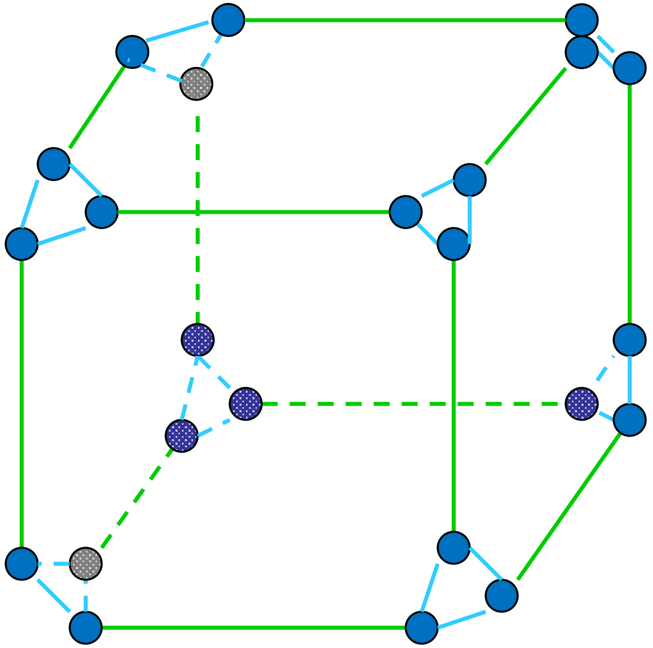
(a) 3–立方体



(b) 由 2 个 3– 立方体组成的 4– 立方体

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3-立方体

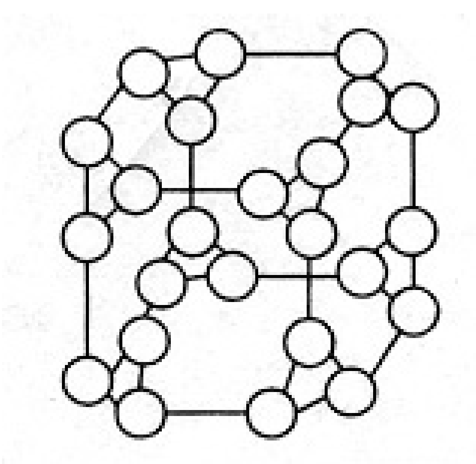
– 带环n-立方体，由$N = 2^n$个结点环构成。

– 每个结点环是有n个结点的环，结点总数为$n\ 2^n$个。

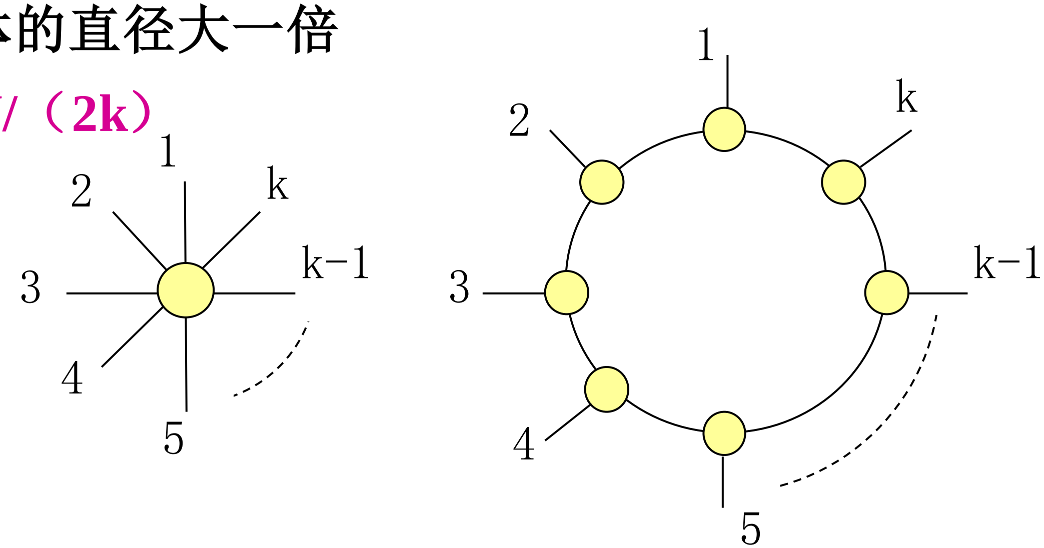
– 直径通常为2n，结点度为3，对称。

8. 带环立方体（简称3-CCC）

- 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）



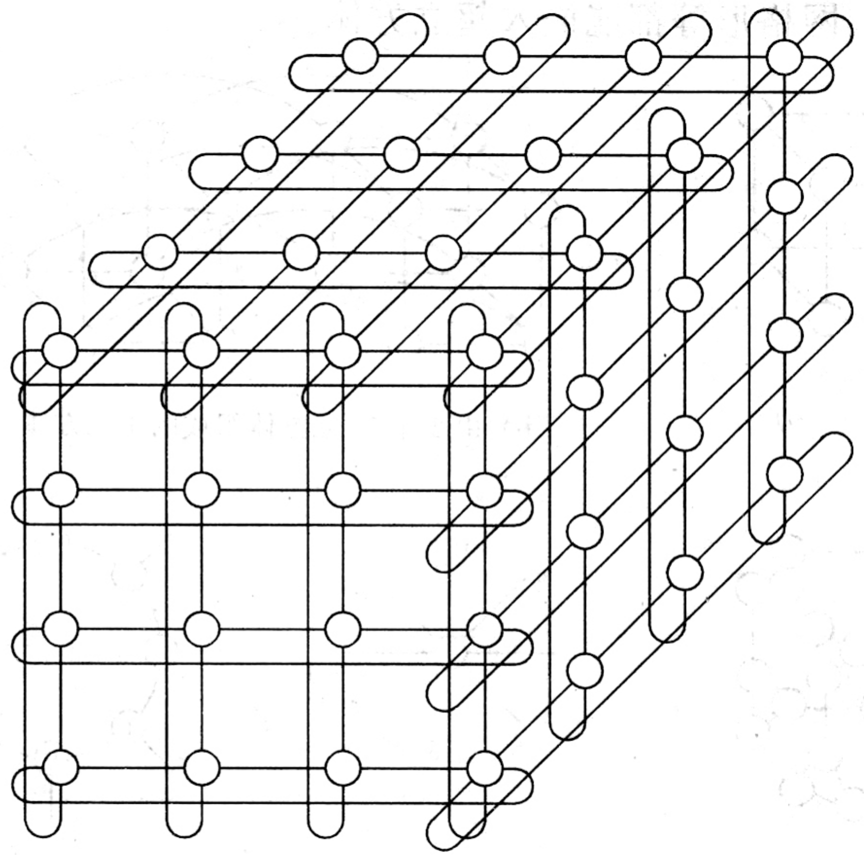
(a) 带环3-立方体



(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. k元n-立方体网络

4元3-立方体网络的拓扑结构



4元3-立方体网络

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）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。

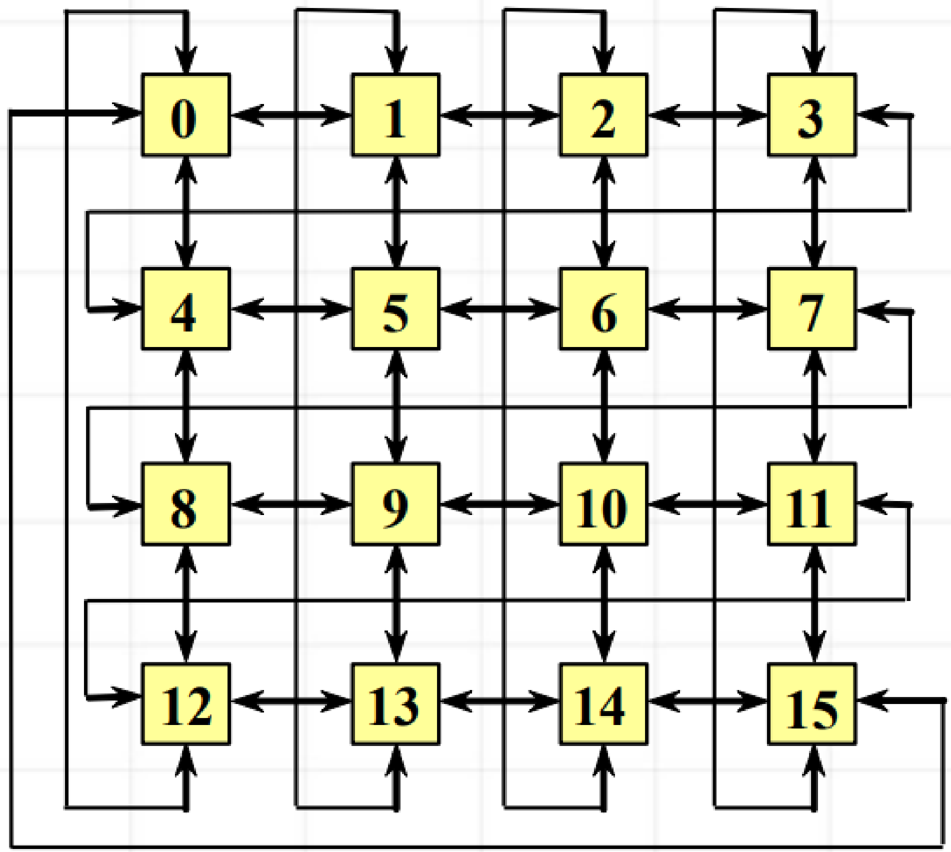
  $\mathbf{N=k^n}$， （ $k=\sqrt[n]{N}$， $\mathbf{n=\log_k N}$）

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

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×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个结点，n=[$\log_2 N$]（维数）
CCC	3	2k-1+[k/2]	3N/2	N/(2k)	是	N=k×2^k结点 环长k≥3
k元n-立方体	2n	n[k/2]	nN	$2k^{n-1}$	是	N=k^n个结点

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≤i≤15）与其他处理器直接互连的一般表达式。



用移数函数构成ILLIAC Ⅳ 阵列机的互连网络

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：

（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）

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：

$$PM2_{+2}(X) = (X + 2^2) \bmod N = (X + 4) \bmod 16$$

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：

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

9.3 静态互连网络

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：

$(i \pm 1) \bmod 16$

$(i \pm 4) \bmod 16$

小结

9.1　[互连网络基本概念]

结点度、网络直径、等分宽度

9.2　[互连函数]

几种基本的互联函数

9.3　[静态互连网络]

几种典型的静态互联网络

9.4　[动态互连网络]

9.5　[消息传递机制]

小结

1. 恒等置换

$$I(x_{n-1}x_{n-2}\ldots x_1x_0)=x_{n-1}x_{n-2}\ldots x_1x_0$$

2. 交换置换 Exchange

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

立方体置换

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

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

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

3. 混洗置换 Shuffle

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

逆混洗置换

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

4. 蝶式置换 Butterfly

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

小结

5. 反位序置换 Reversal

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

6. 移数置换

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

7. PM2I置换

$PM2_{+i}(x) ＝ x＋2^i \bmod N$

$PM2_{-i}(x) ＝ x－2^i \bmod N$