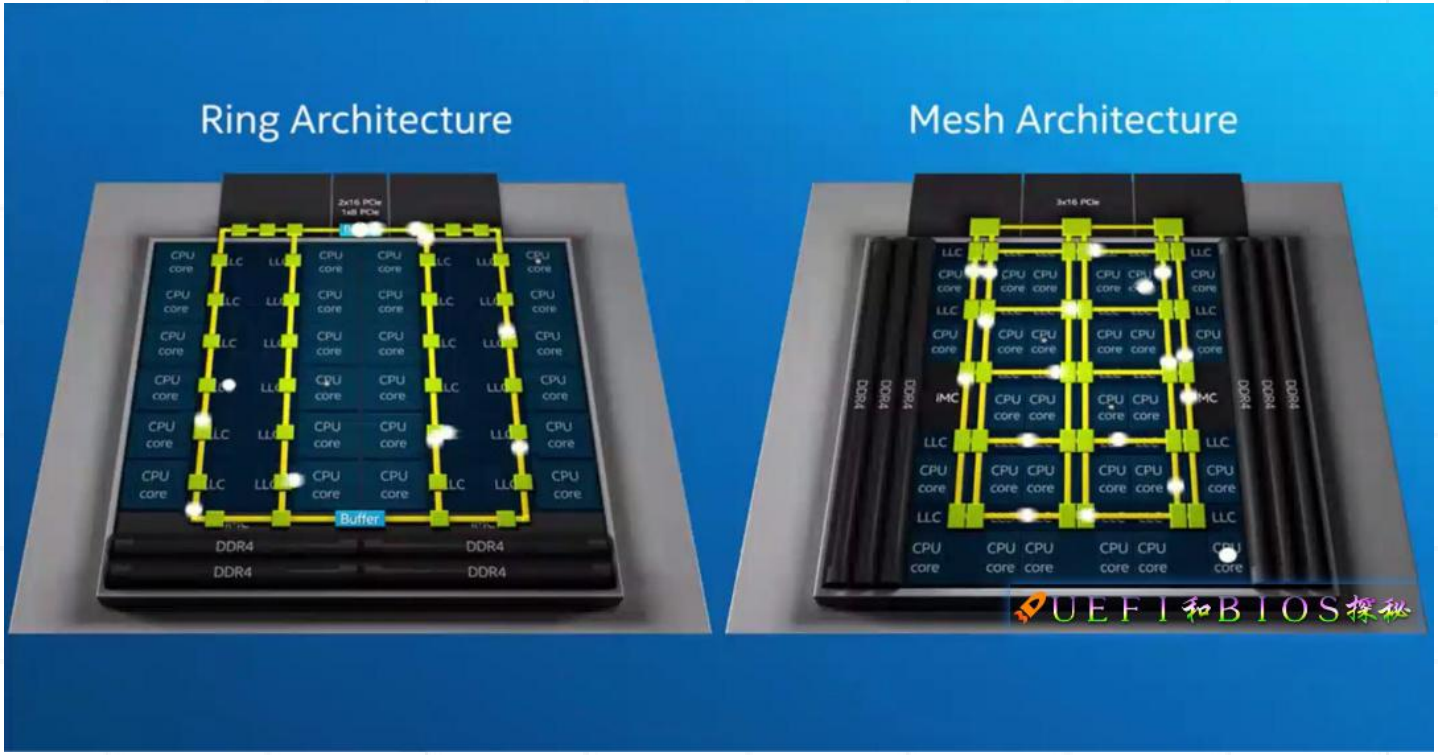
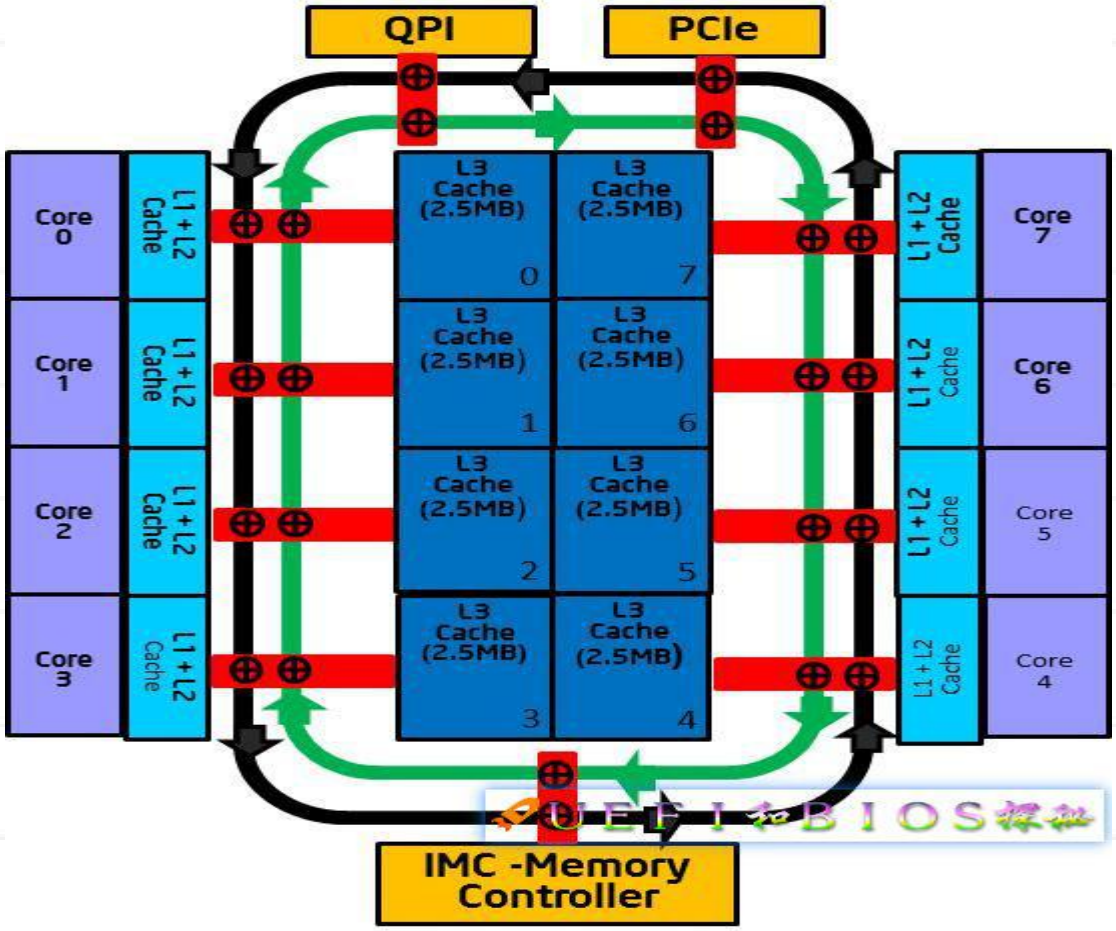
Intel CPU从Ring Bus到Mesh网络



Intel Nehalem EP/EX 的Ring Bus网络



1.双环设计可以保证任何两个ring stop之间距离不超过Ring Stop总数的一半，延迟较低。

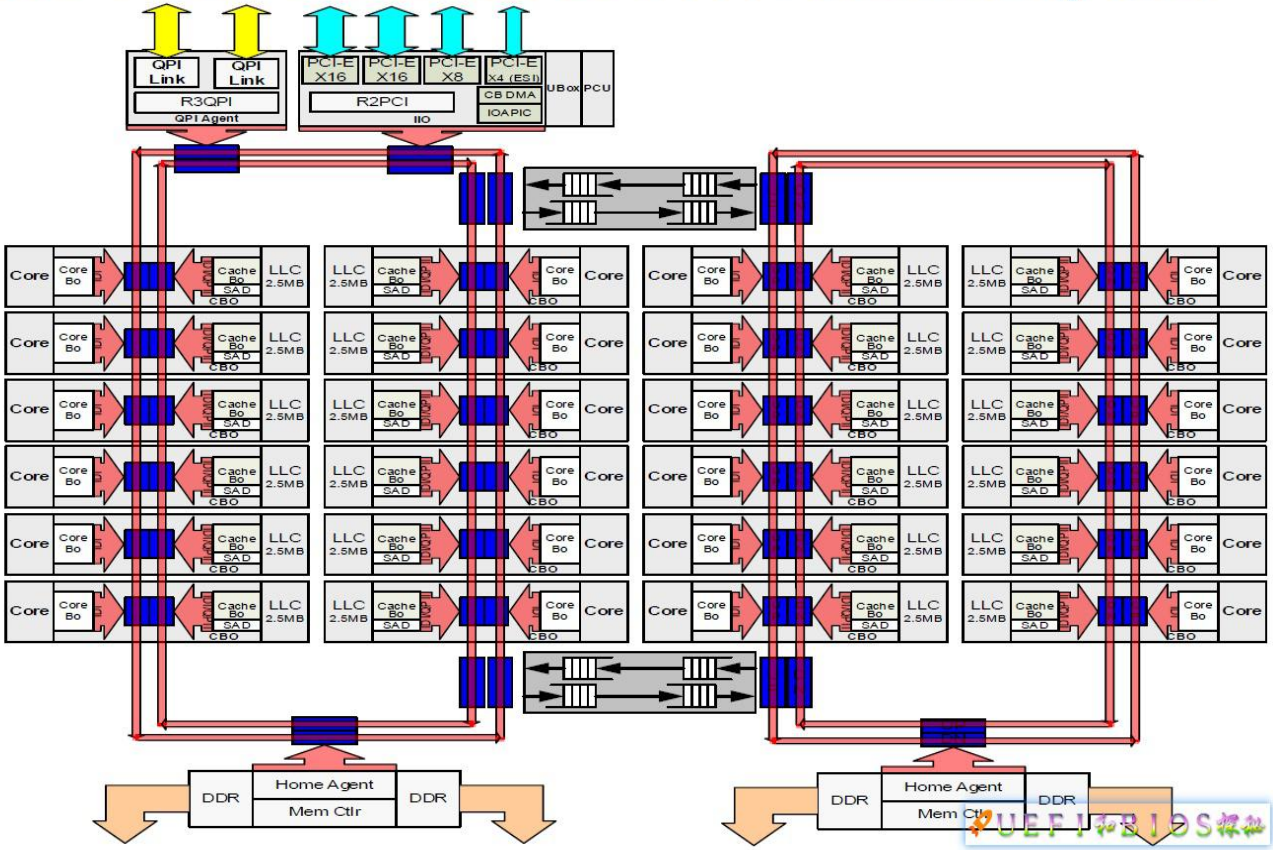
2.各个模块之间交互方便，不需要Core中转。这样一些高级加速技术，如DCA（Direct Cache Access), Crystal Beach等等应运而生。

3.高速的ring bus保证了性能的极大提高。Core to Core latency 只有60ns左右，而带宽则高达数百G(记得Nehalem是192GB/s).

4.方便灵活。增加一个Core，只要在Ring上面加个新的ring stop就好，不用像以前一样考虑复杂的互联问题。

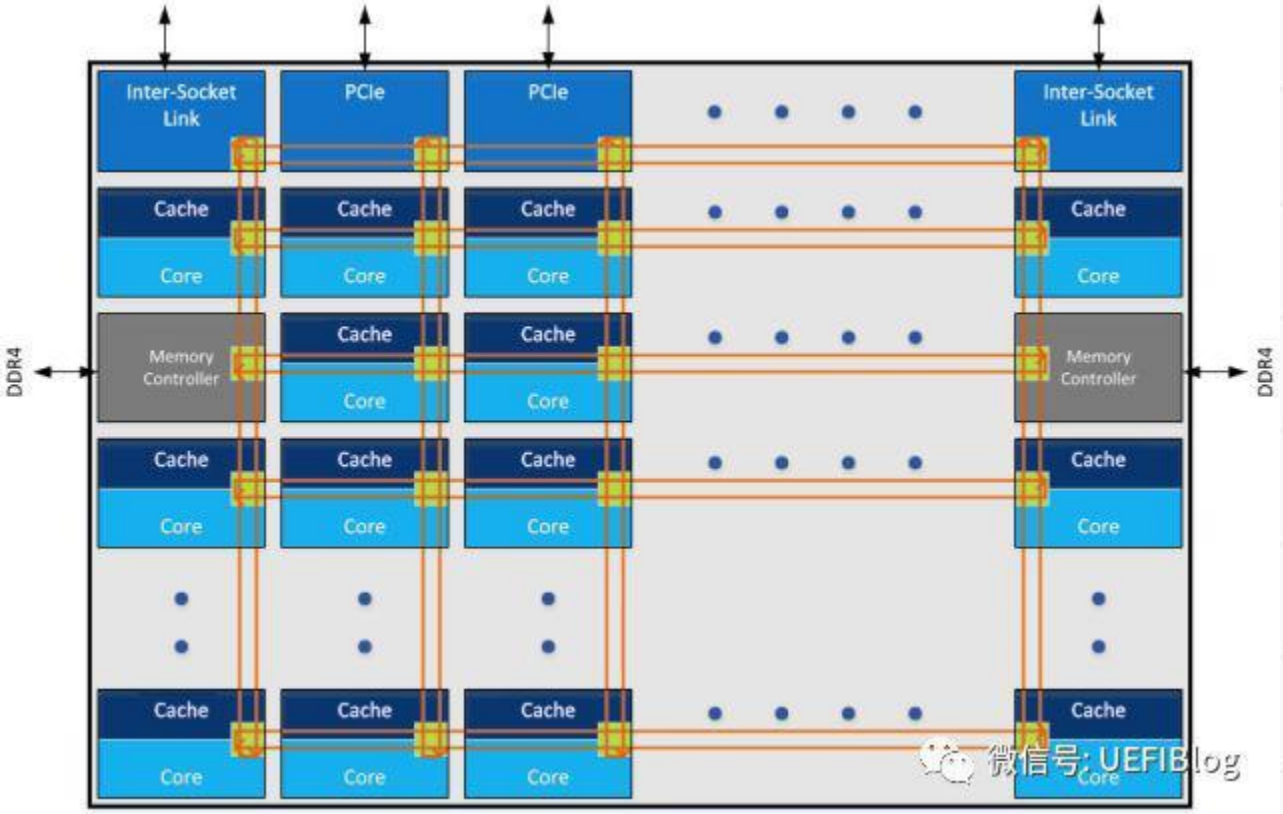
Intel Xeon 志强

Intel® Xeon® Processor E5 v4 Product Family HCC



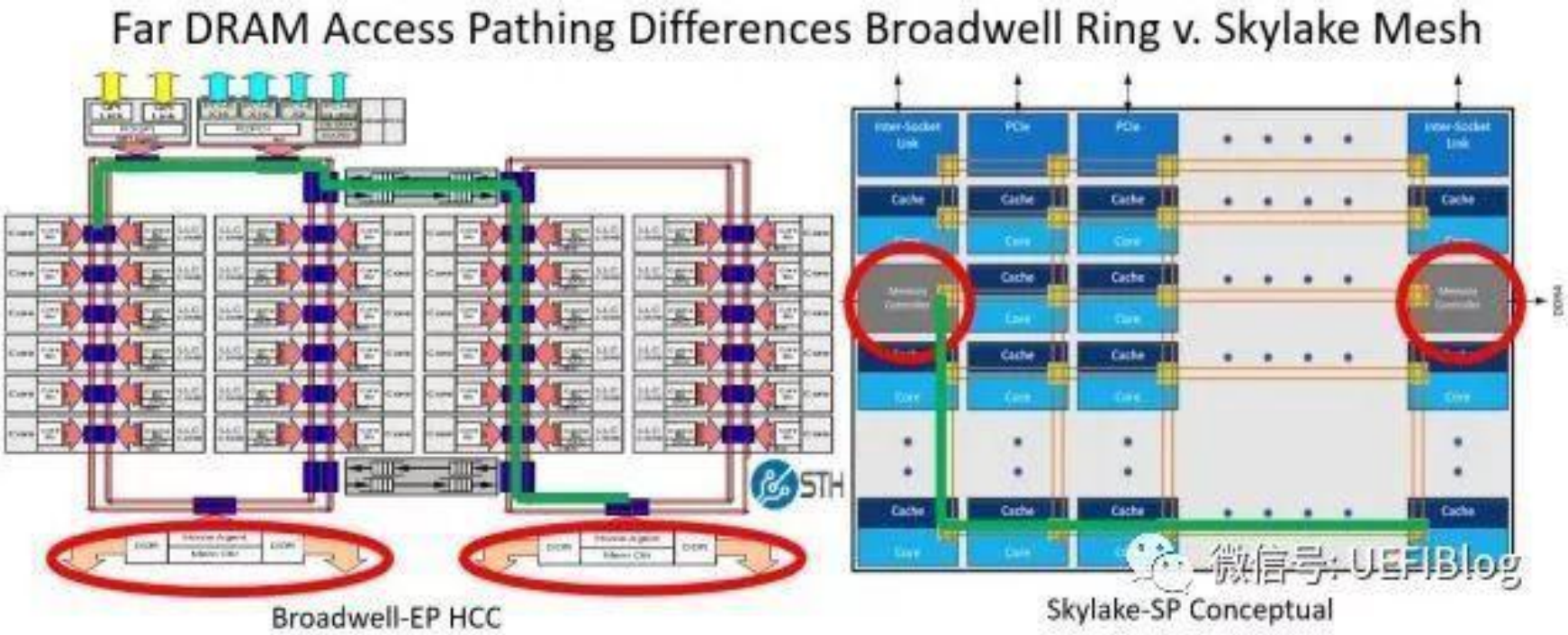
1.在至强HCC(High Core Count, 核很多版)版本中，又加入了一个ring bus。两个ring bus各接12个Core，将延迟控制在可控的范围内。俩个Ring Bus直接用两个双向Pipe Line连接，保证通讯顺畅。于此同时由于Ring 0中的模块访问Ring 1中的模块延迟明显高于本Ring，亲缘度不同，所以两个Ring分属于不同的NUMA（Non-Uniform Memory Access Architecture）node

Intel在Skylake和Knight Landing中Mesh



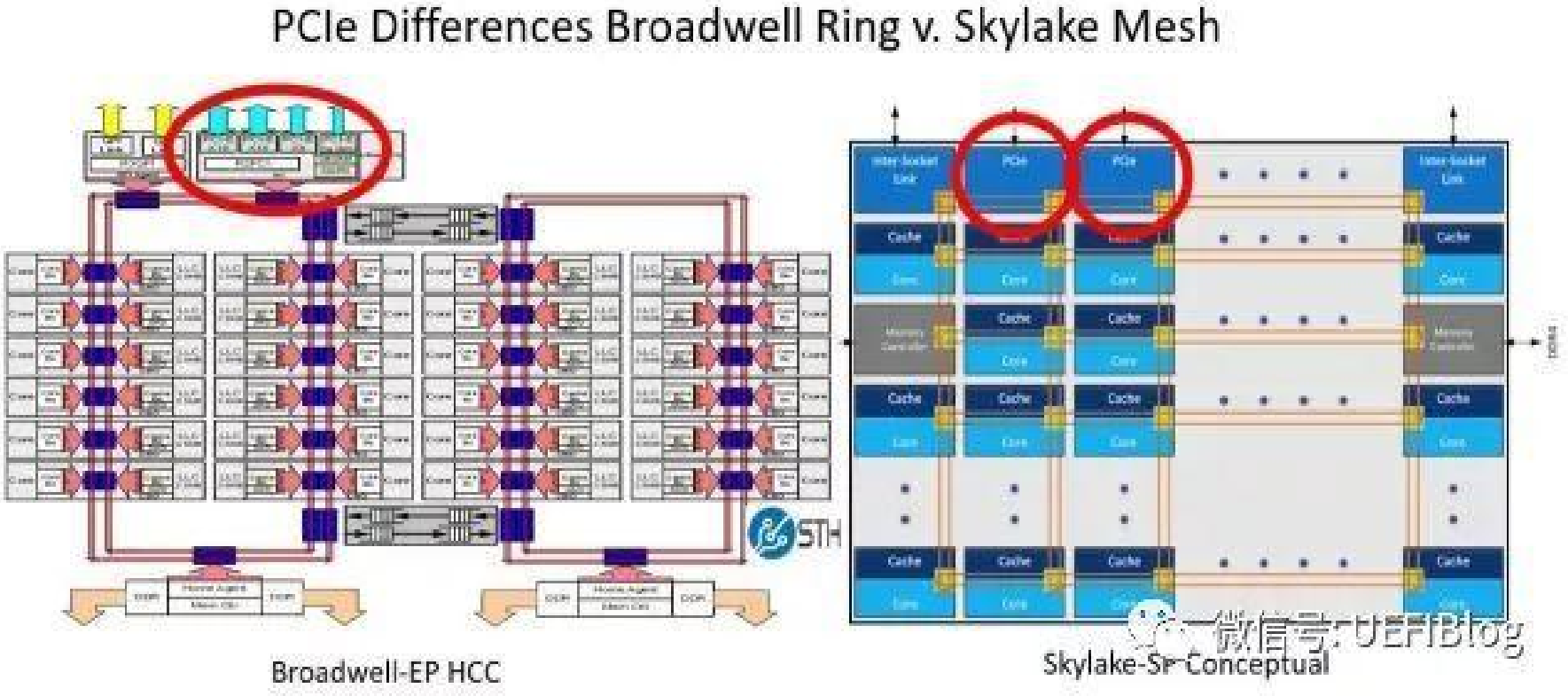
1.首先当然是灵活性。新的模块或者节点在Mesh中增加十分方便，它带来的延迟不是像ring bus一样线性增加，而是非线性的。从而可以容纳更多的内核。
2.设计弹性很好，不需要1.5 ring和2ring的委曲求全。
3.双向mesh网络减小了两个node之间的延迟。过去两个node之间通讯，最坏要绕过半个ring。而mesh整体node之间距离大大缩减。
4.外部延迟大大缩短

RAM延迟大大缩短



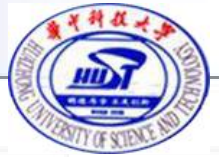
左边的是ring bus，从一个ring里面访问另一个ring里面的内存控制器。最坏情况下是那条绿线，拐了一个大圈才到达内存控制器，需要310个cycle。而在Mesh网络中则路径缩短很多。

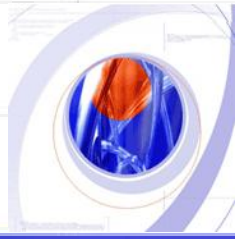
IO延迟缩短



*Mesh*网络带来了这么多好处，那么缺点有没有呢？它网格化设计带来复杂性的增加，从而对*Die*的大小带来了负面影响。

第9章 互连网络（ICN）

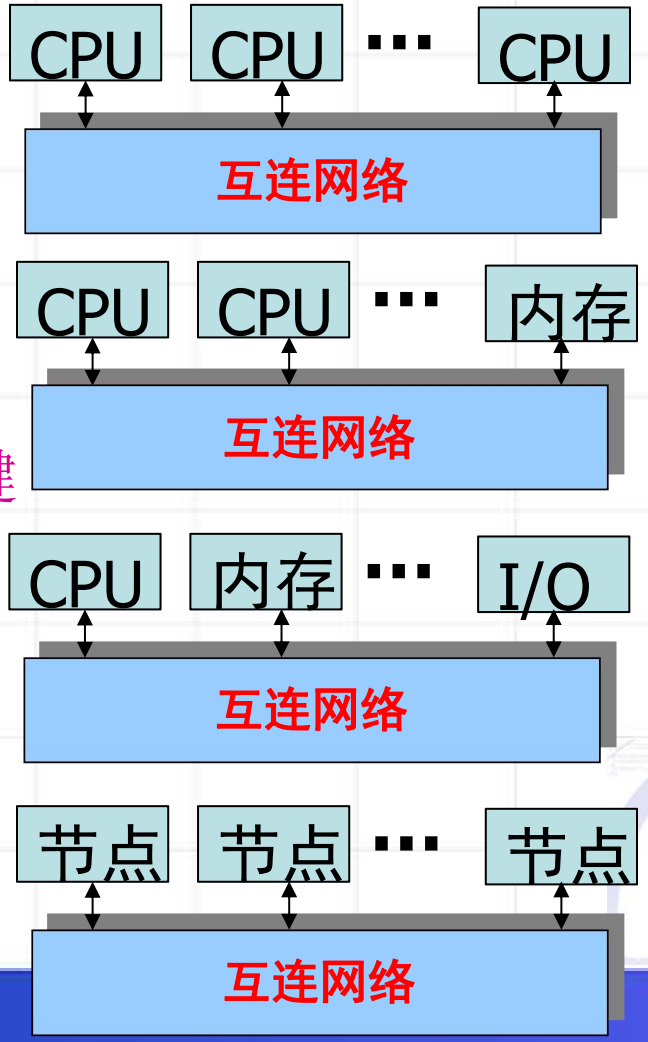




互连网络

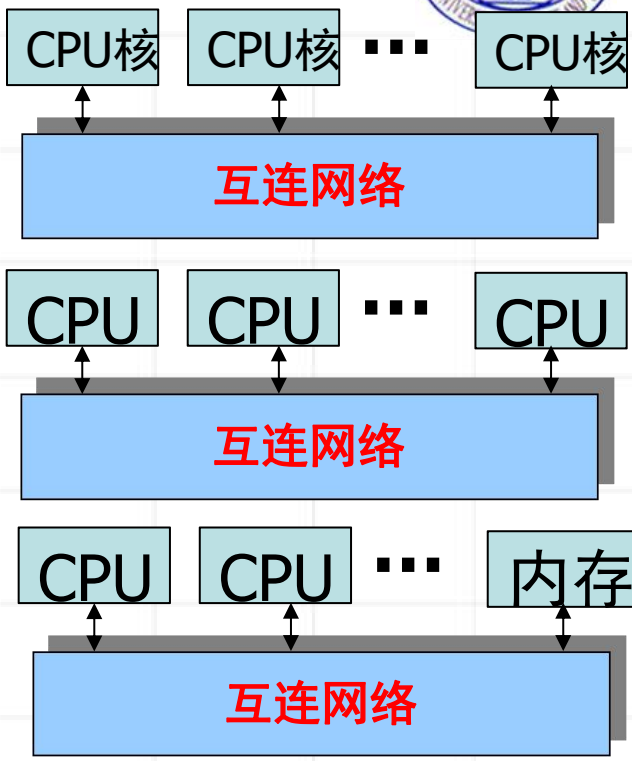
互连网络是计算机部件、计算机节点或计算机系统之间的连接

CPU内多个核之间
CPU之间
CPU和内存之间
内存和内存之间
计算机节点之间
网络和网络之间
是SIMD计算机和MIMD计算机的关键组成部分，也是理解互连网络能够更好的设计和评价计算机系统
互连网络的直接设计目标是：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



第9章 互连网络（ICN）

1. 高速互连网络（<100s时钟周期）
   - 片上网络On-chip networks (OCNs)
     - 芯片内部功能模块之间（例如 多核）
   - 系统网System area networks (SANs)
     - 芯片之间
     - IO单元之多路处理器、处理器和内存
     - 间



2. 本章内容
   - 互连网络基本概念，互连网络的结构参数与性能指标
   - 互连函数
   - 静态互连网络
   - 动态互连网络

7.1.1 互连网络概念

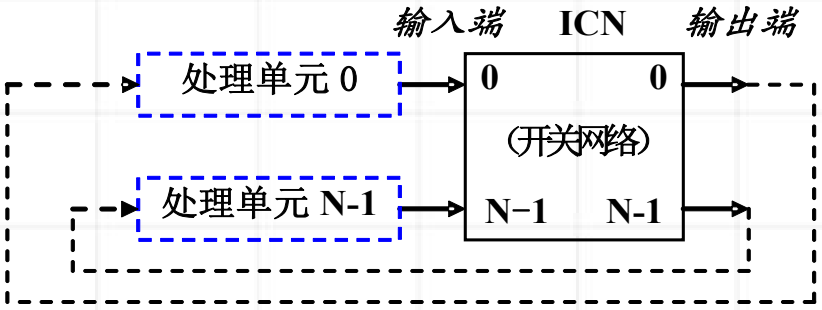
片上/系统高速互连网络是一种由**网络元件**按照**一定的拓扑结构**和**控制方式**构成的网络，用来实现计算机系统中部件之间的高速连接。

- 3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
  - **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**

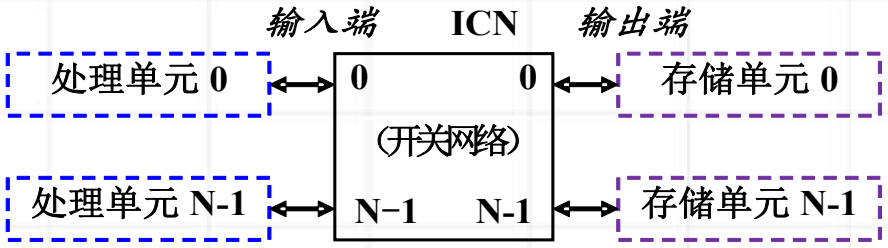
7.1.1 互连网络概念

ICN目的与作用

(1)互连网络与处理单元的连接模型



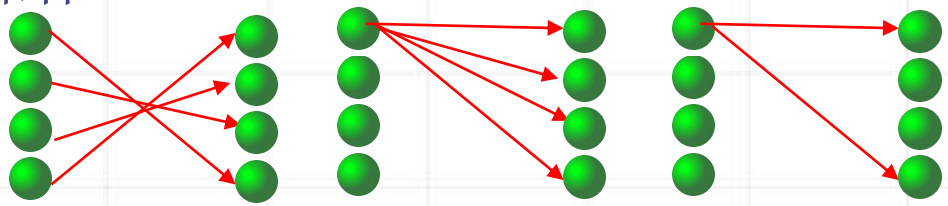
(a) 处理单元/处理单元的连接



(b) 处理单元/存储单元的连接

(2)互连网络的主要操作：

置换(N—N)



广播(1 — N)

选播(1 — N')

7.1.2 互连网络的参数和指标

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。

7.1.2 互连网络的参数和指标

- 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：网络中任意两结点间距离的最大值。

  网络直径应当尽可能地小。
- 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×w，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。

7.1.2 互连网络的参数和指标

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。

=软件开销+通道时延+选路时延+竞争时延

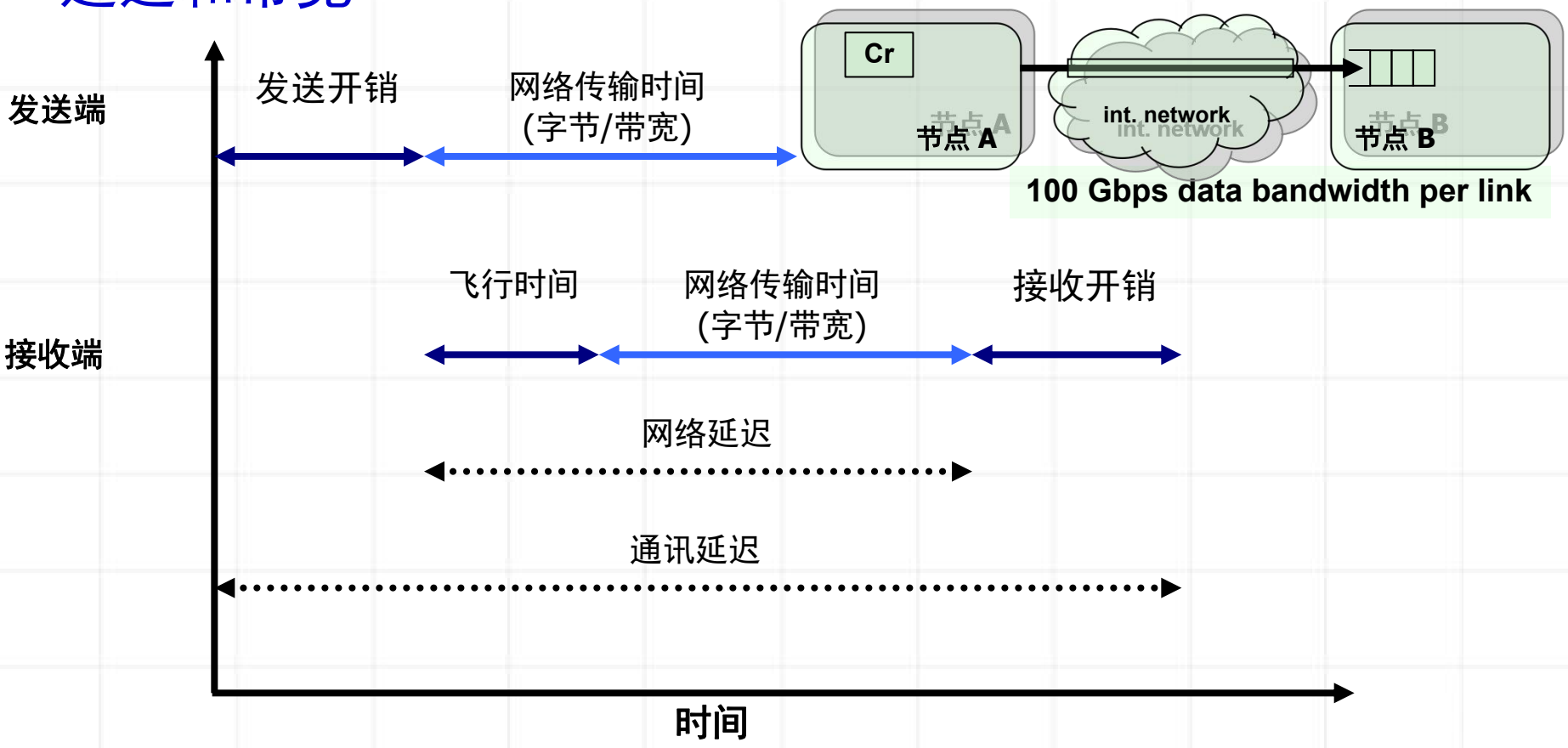
2. 网络时延

通道时延与选路时延的和。

- 由网络硬件特征决定，与程序行为和网络传输状态无关。

7.1.2 互连网络的参数和指标

延迟和带宽



$$通信延迟 = 发送开销 + 飞行时间 + \frac{数据包大小}{带宽} + 接收开销$$

例子：Latency = 5 ns + 0.025 ns + 50 ns + 5 ns = 60.025 ns

7.1.2 互连网络的参数和指标

3. 端口带宽

- 对于互连网络中任意一个端口来说，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。
  - ◆ 对称网络
  - ◆ 非对称网络

4. 聚集带宽

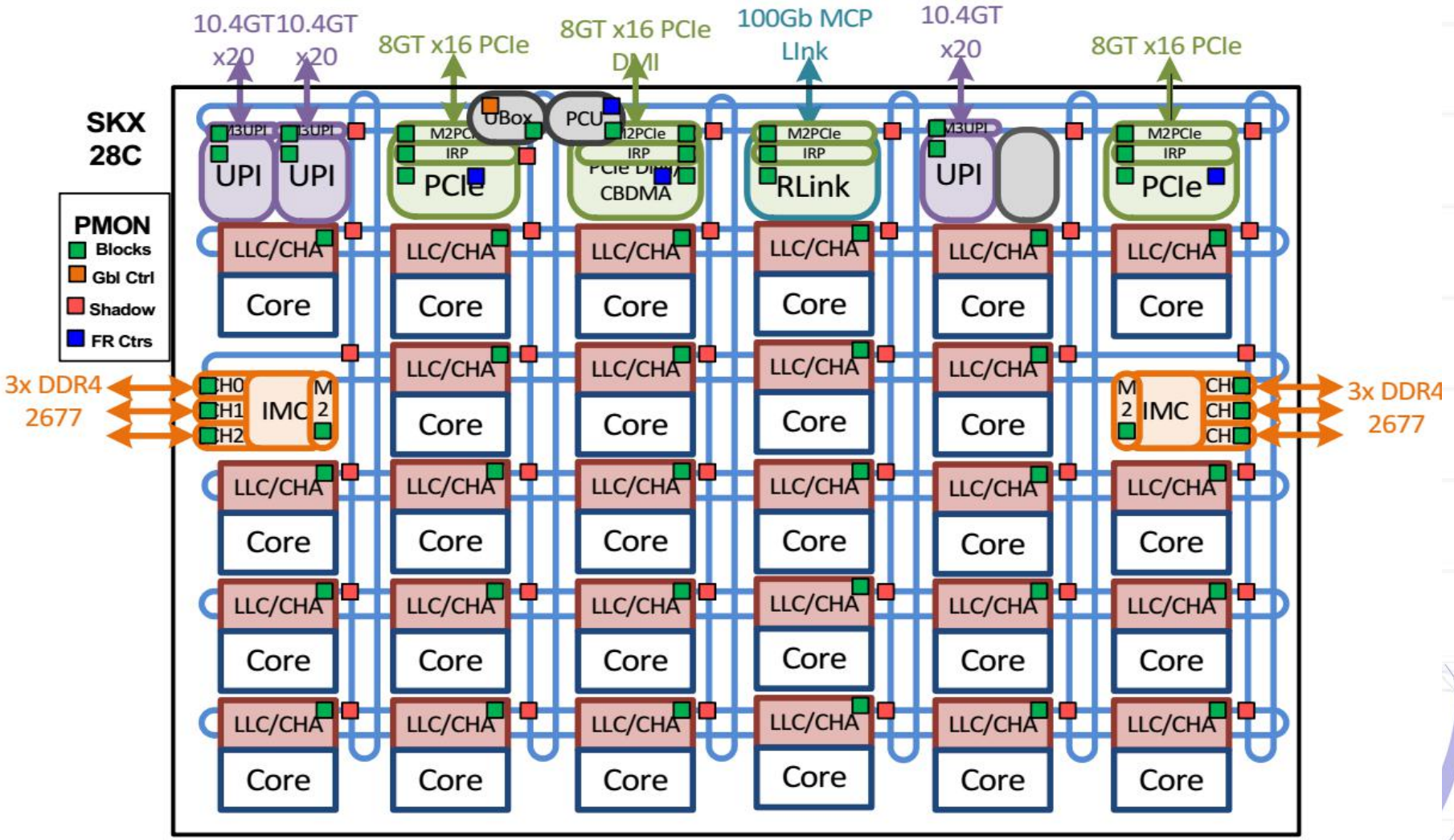
例如，HPS是一种对称网络

- **网络规模N的上限：512**
- **端口带宽：40MB/s**

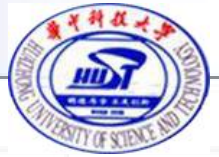
HPS的聚集带宽：（40MB/s×512）/2＝10.24GB/s

7.1.2 互连网络的参数和指标

Intel® Xeon® 处理器内部互连结构



第7章 互连网络（ICN）

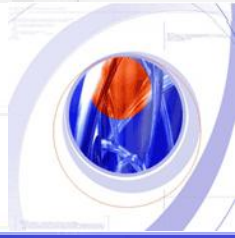


7.1 互连网络基本概念

7.2 互连函数

7.3 静态互连网络

7.4 动态互连网络



7.2.1 互连函数

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

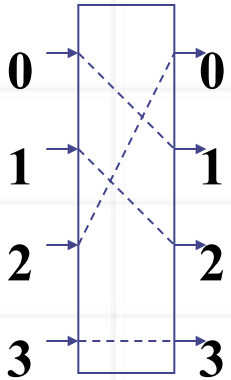
通过数学表达式描述输入端号与输出端号的连接关系。即在互连函数f的作用下，输入端x连接到输出端f(x)。

7.2 互连函数

互连函数有多种表示方式，如下例所示：

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

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。

7.2 互连函数

- 互连函数f(x) 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：

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

7.2.2 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。

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

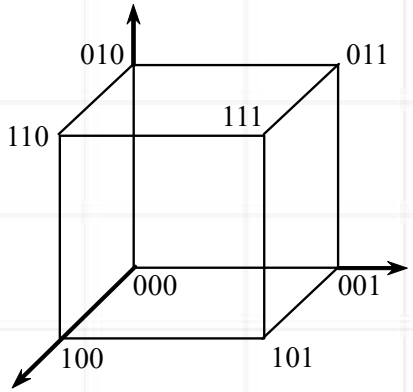
2. 交换函数

- 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。

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

7.2 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：



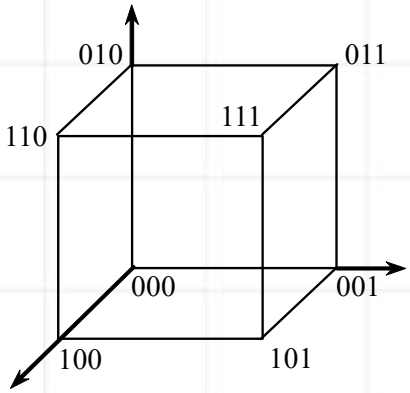
8结点的立方体网络

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

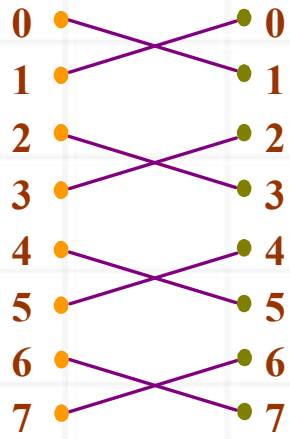
$$Cube_1(x_2x_1x_0)=x_2\bar{x}_1x_0$$

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

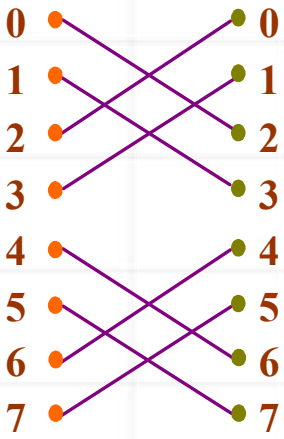
7.2 互连函数



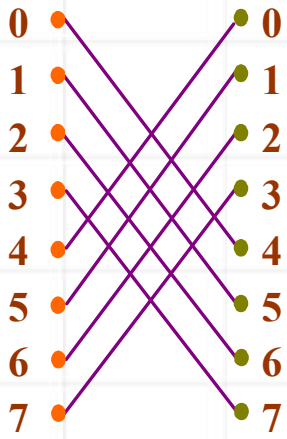
8结点的立方体网络



(a) $Cube_0$ 函数



(b) $Cube_1$ 函数



(c) $Cube_2$ 函数

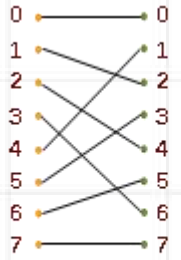
N=8 的立方体交换函数

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），**shuffle函数**

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

  - 即把输入端的二进制编号循环左移一位。



(a) 均匀洗牌函数σ

7.2 互连函数

- 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。

7.2 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

$$f(x)\times g(x)=I(x)$$

则称g(x)是f(x)的逆函数，记为$f^{-1}(x)$。

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

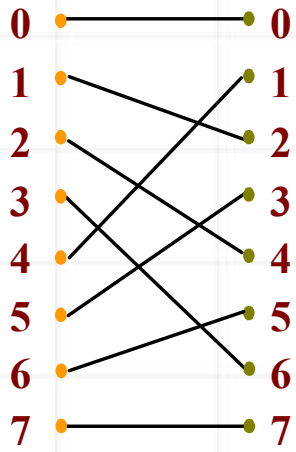
- 逆均匀洗牌函数：将输入端的二进制编号循环右移一位而得到所连接的输出端编号。
  - 互连函数

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

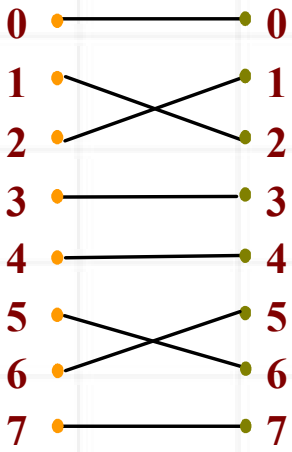
  - 逆均匀洗牌是均匀洗牌的逆函数

7.2 互连函数

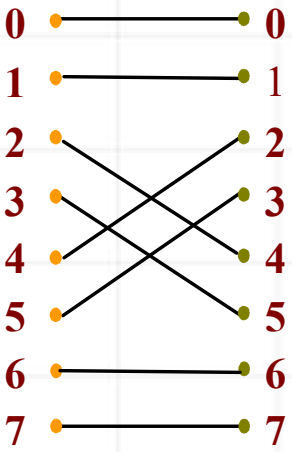
- N=8 的均匀洗牌和逆均匀洗牌函数



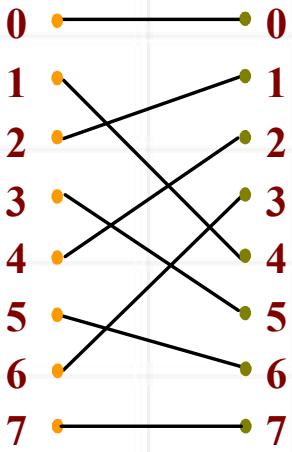
(a) 均匀洗牌函数 σ



(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

– 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：$shuffle^n(j) = j$

性质2：$shuffle_n(j) = j$

4. 碟式函数 (β)

- 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。

- 第k个子函数

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

- 第k个超函数

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

- 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。

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。

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

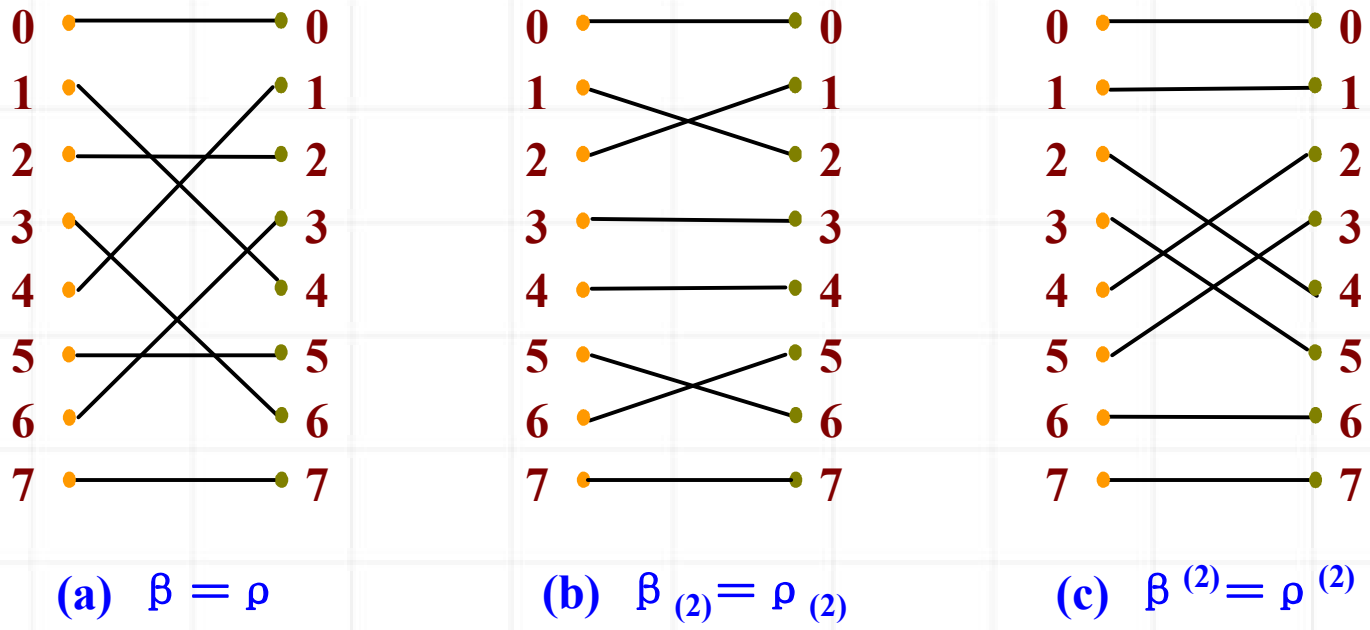
- 第k个子函数

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

- 第k个超函数

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

7.2 互连函数

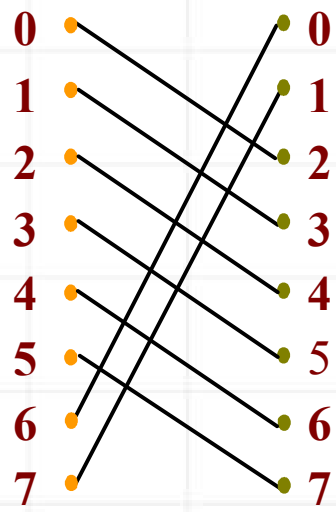


(a) $\beta = \rho$　　(b) $\beta_{(2)} = \rho_{(2)}$　　(c) $\beta^{(2)} = \rho^{(2)}$

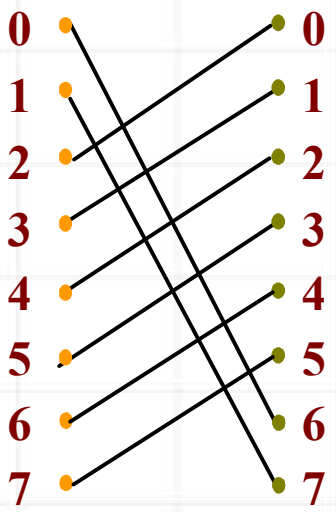
N=8 的蝶式函数和反位序函数

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



(a) 左移移数函数 k＝2



(b) 右移移数函数 k＝2

7. PM2I函数 （加减2^i函数）

- P和M分别表示加和减，2I表示2^i。
- 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。

7.2 互连函数

- 当N＝8时，有6个PM2I函数：

$PM2_{+0}$ ： (0 1 2 3 4 5 6 7)

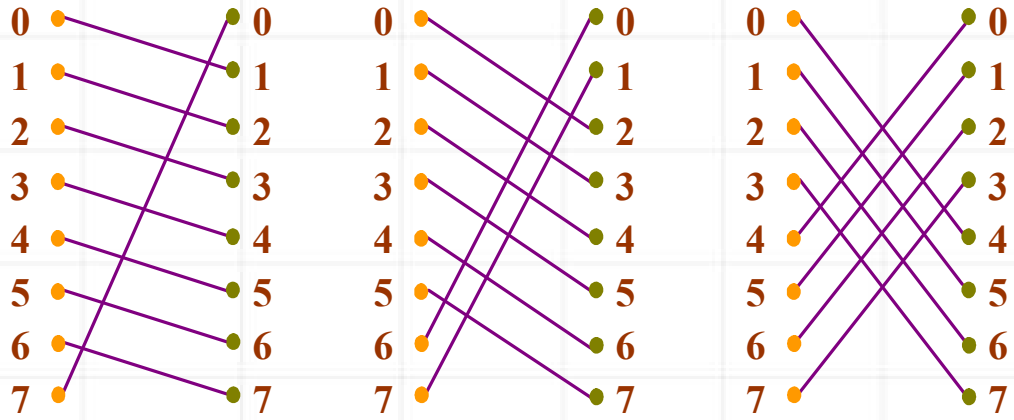
$PM2_{+1}$ ： (0 2 4 6)(1 3 5 7)

$PM2_{+2}$ ： (0 4)(1 5)(6)(3 7)

$PM2_{-0}$ ： (7 6 5 4 3 2 1 0)

$PM2_{-1}$ ： (6 4 2 0)(7 5 3 1)

$PM2_{-2}$ ： (4 0)(5 1)(6 2)(7 3)



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

N=8 的PM2I函数

例题解析

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 $Cube_3(x_3x_2x_1x_0)=\overline{x}_3x_2x_1x_0$，

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 = 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。

（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。

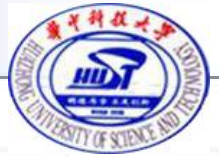
所以处理器13连至处理器11，而处理器14连至处理器13。

（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。

第7章 互连网络（ICN）

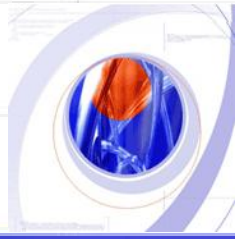


7.1 互连网络基本概念

7.2 互连函数

7.3 静态互连网络

7.4 动态互连网络



7.3 静态互连网络

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

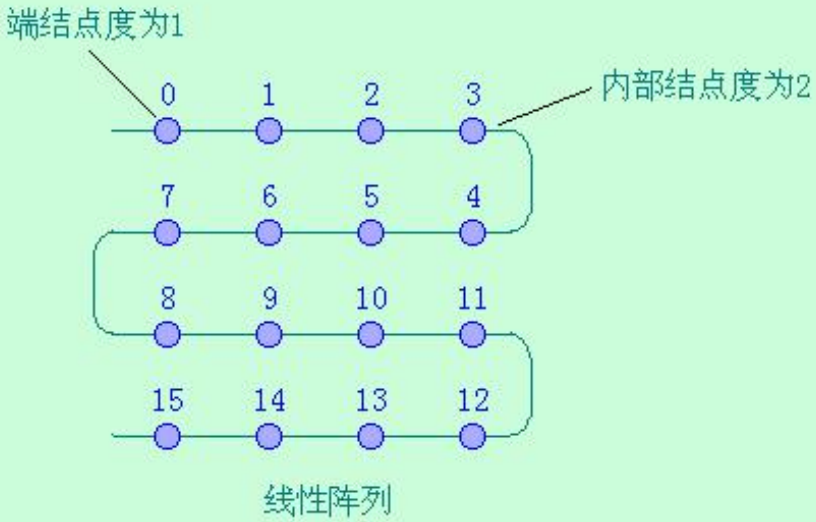
  由交换开关构成、可按运行程序的要求动态地改变连接状态的网络。

下面介绍几种静态互连网络。

（其中：N表示结点个数）

7.3.1 1、2维互连函数

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。



端结点的度：1

其余结点的度：2

直径：N－1

等分宽度b=1

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

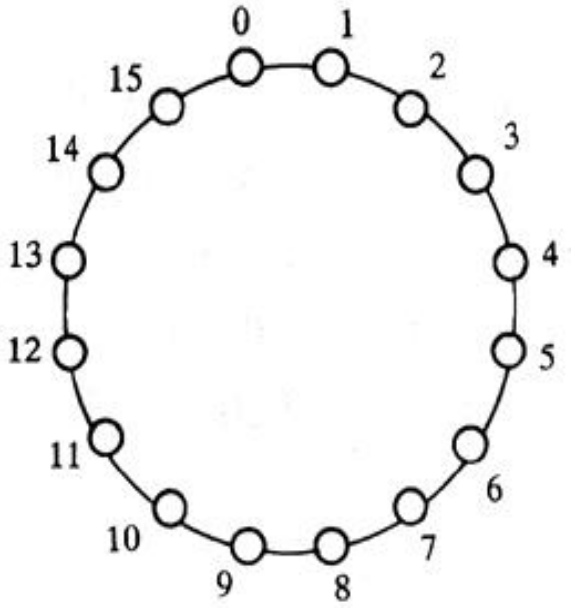
线性阵列允许不同的源结点和目的结点对并行地使用其不同的部分

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。

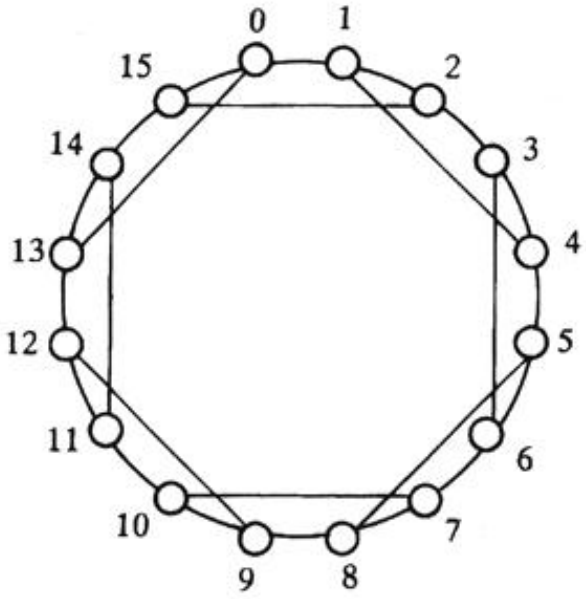
- 对称
- 结点的度：2
- 双向环的直径：N/2
- 单向环的直径：N-1
- 环的等分宽度b=2



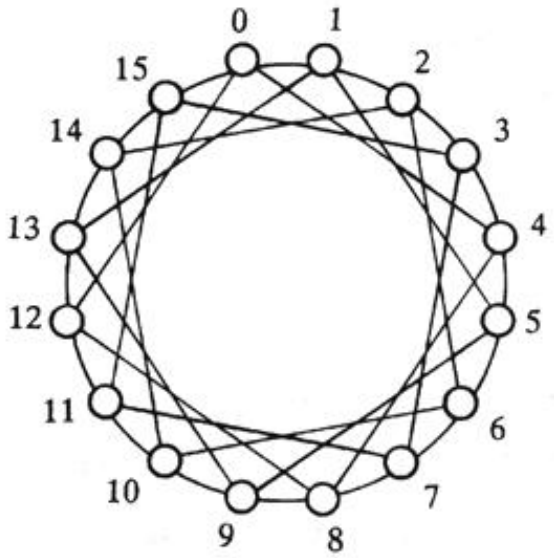
(b)环

7.3.1 1、2维互连函数

带弦环　增加的链路愈多，结点度愈高，网络直径就愈小。



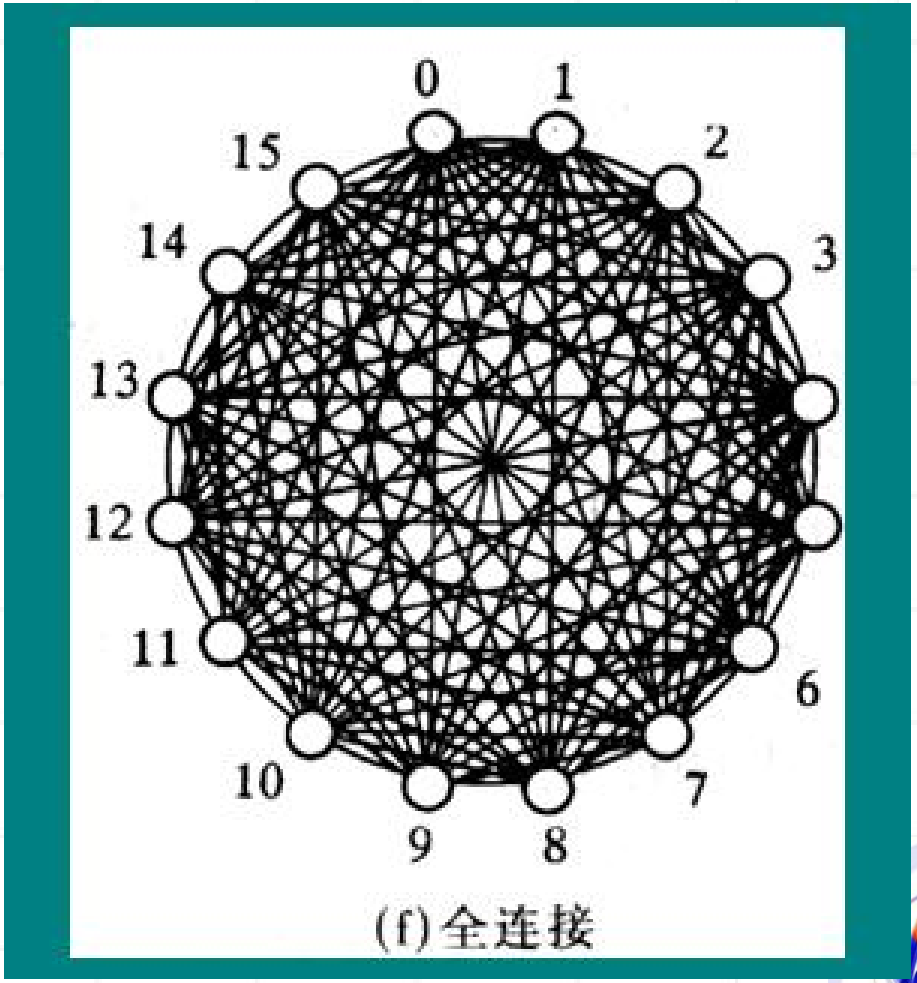
(c)度为 3 的带弦环



(d)度为 4 的带弦环(与 Illiac 网相同)

7.3.1 1、2维互连函数

- 全连接网络
  - 结点度：15
  - 直径为1。

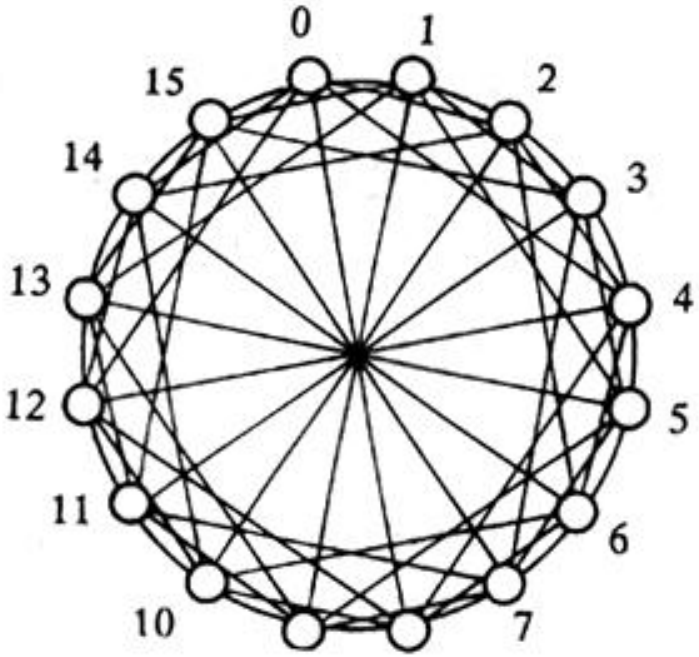


(f)全连接

7.3.1 1、2维互连函数

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

1. 一般地，如果 | j−i | $=2^r$

（r=0, 1, 2, …, n−1，n=$\log_2 N$），

则结点i与结点j连接。

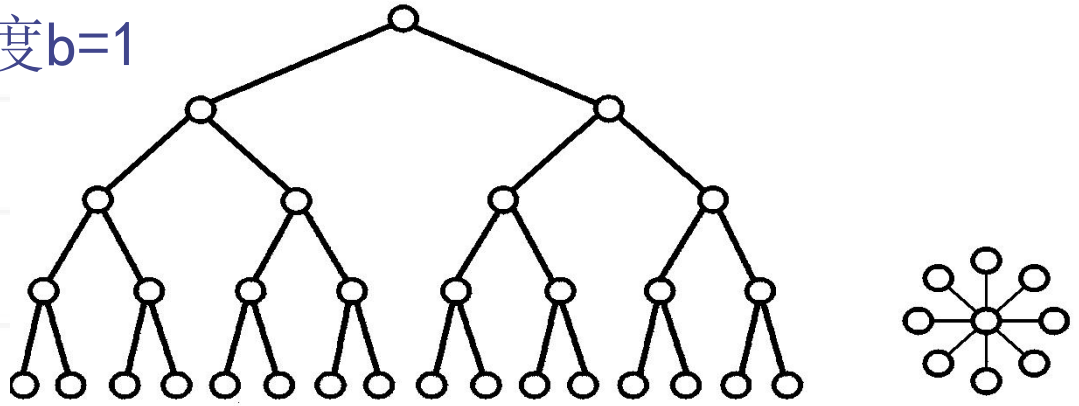
- 结点度：2n－1
- 直径：n/2
- 网络规模$N=2^n$

4. 树形和星形

树形:一棵5层31个结点的二叉树

一般，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

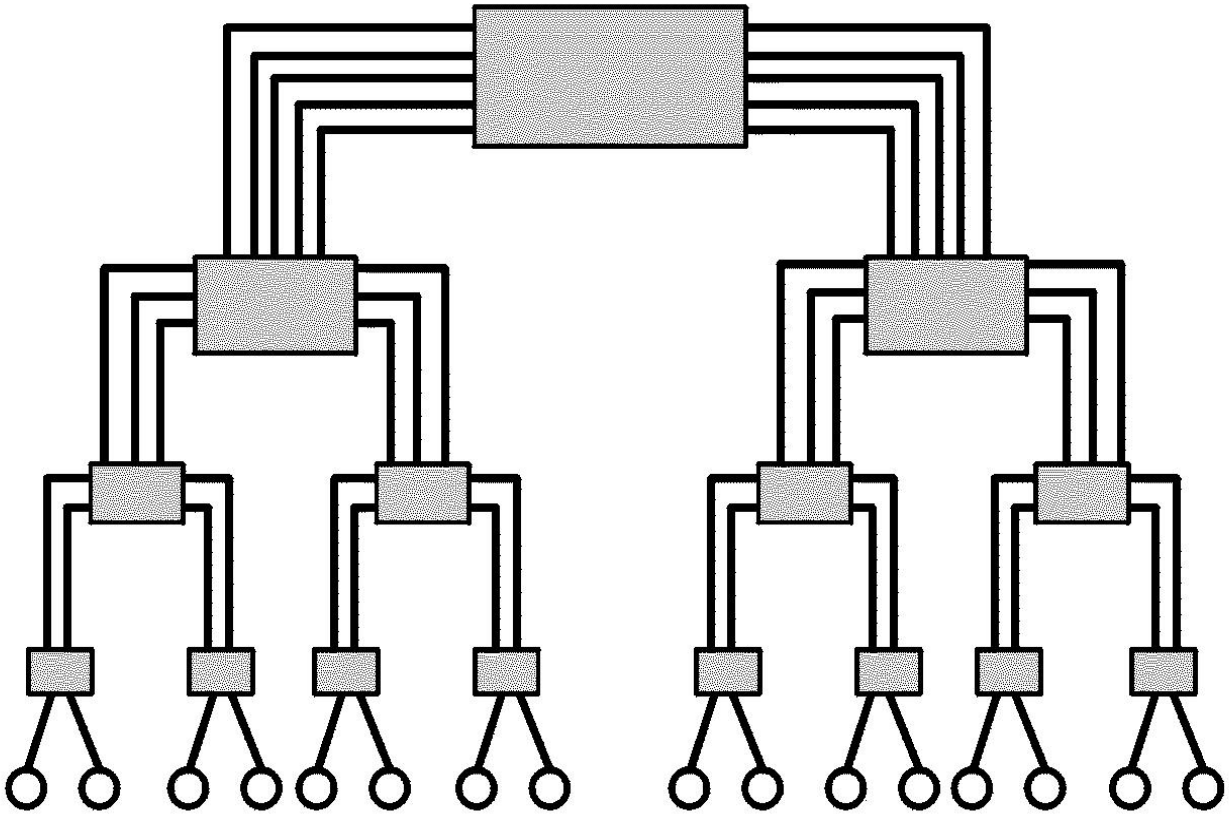


(a) 二叉树　　(b) 星形

星形:

- 结点度较高，为N－1。
- 直径较小，是一常数2。等分宽度b=⌊N/2⌋
- 可靠性较差，中心结点出故障，整个系统就会瘫痪。

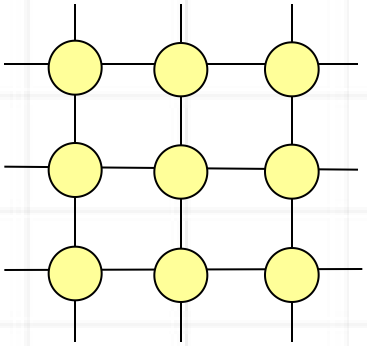
5. 胖树形



(c) 二叉胖树

6. 网格形和环网形

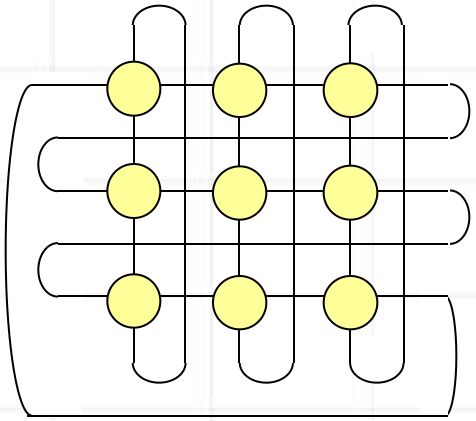
➢ 网格形



（a）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) 。

7.3.1 1、2维互连函数



Illiac网络

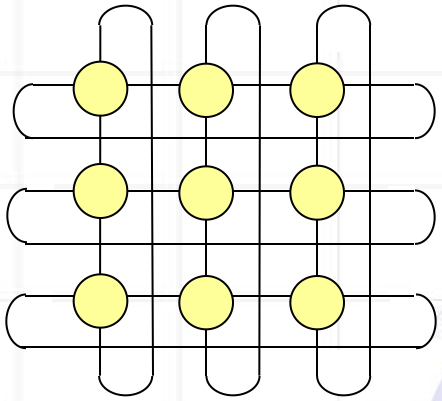
- 名称来源于采用了这种网络的Illiac IV计算机
- 把2维网格形网络的每一列的两个端结点连接起来，再把每一行的尾结点与下一行的头结点连接起来，并把最后一行的尾结点与第一行的头结点连接起来。
- 一个规模为n×n的Illiac网络
  - 所有结点的度d=4
  - 网络直径D=n-1

    Illiac网络的直径只有纯网格形网络直径的一半。
  - 等分宽度：2n

7.3.1 1、2维互连函数

环网形

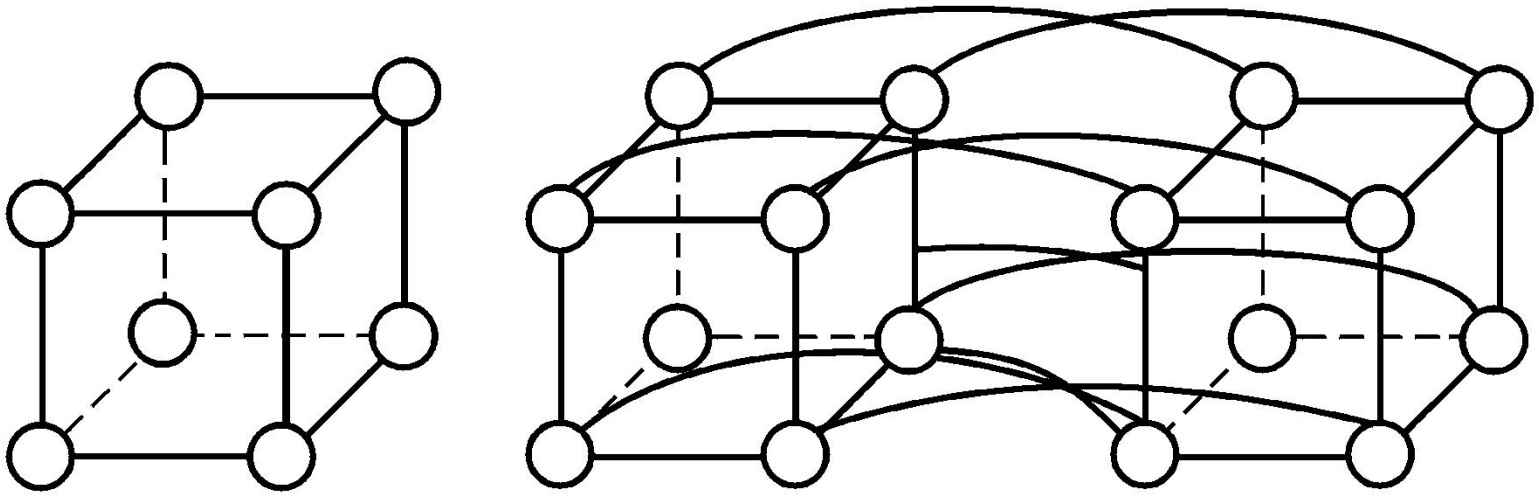
- 可看作是直径更短的另一种网格。
- 把2维网格形网络的每一行的两个端结点连接起来，把每一列的两个端结点也连接起来。
- 将环形和网格形组合在一起，并能向高维扩展。
- 一个n×n的环网形网
  - 结点度：4
  - 网络直径：$2\times\lfloor n/2\rfloor$
  - 等分宽度b=2n



（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。



(a) 3–立方体　　(b) 由 2 个 3– 立方体组成的 4– 立方体

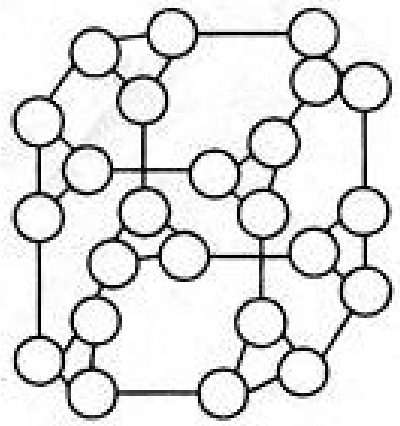
7.3.2 超立方体和寻径

7. 超立方体

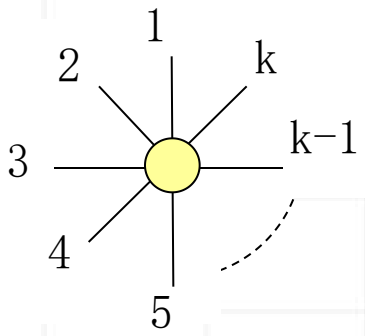
- 为实现一个n-立方体，只要把两个（n—1）立方体中相对应的结点用链路连接起来即可。共需要2^{n-1}条链路。
- n-立方体中结点的度都是n，直径也是n，等分宽度为b=N/2 。

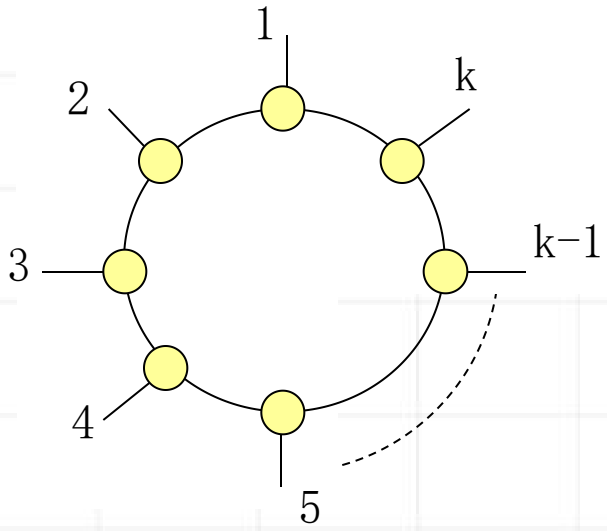
8. 带环立方体 （简称3-CCC）

- 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。



(a) 带环3-立方体





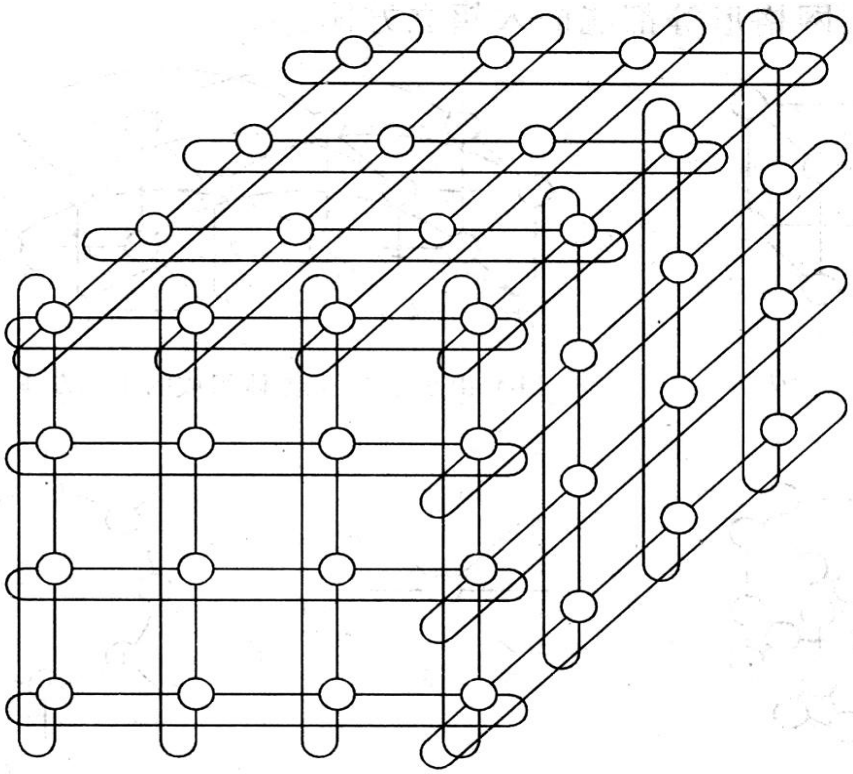
(b) 将k-立方体的每个结点用由k个结点的环来代替，组成带环k-立方体

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$b=N/（2k）$

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。

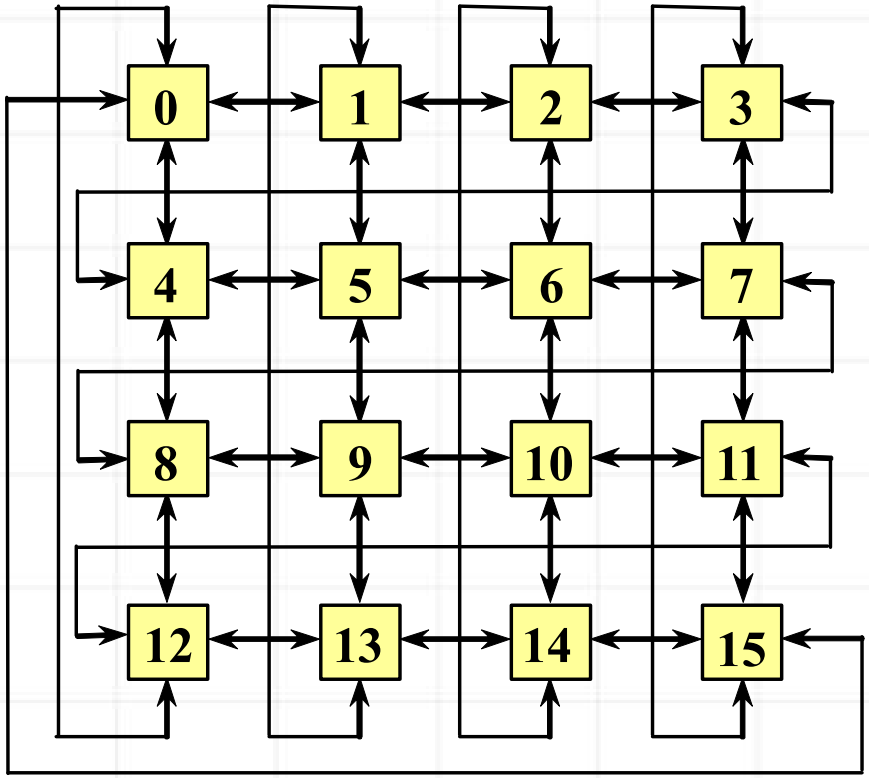
  $\mathbf{N=k^n}$，（ $k=\sqrt[n]{N}$ ，$\mathbf{n=\log_k N}$）
- k元n-立方体的结点可以用基数为k的n位地址$A=a_1a_2\ldots a_n$来表示。
  - **其中a_i表示该结点在第i维上的位置**
- 通常把低维k元n-立方体称为环网，而把高维k元n-立方体称为超立方体。

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，
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

7.3.2 超立方体和寻径

例 已知有16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）

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：

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

7.3.2 超立方体和寻径

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）

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$PM2_{\pm 2}(X)$

基于静态互连网络的寻径机制

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
  - 二维网格网络的**X-Y**寻径
  - **N**元超方体的**E-cube**寻径
- 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。
  - 具有拥塞控制，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
  - 例如：**TCP/IP**网络

二维网格网络的X-Y寻径

- 二维网格网络的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

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）。

二维网格网络的X-Y寻径

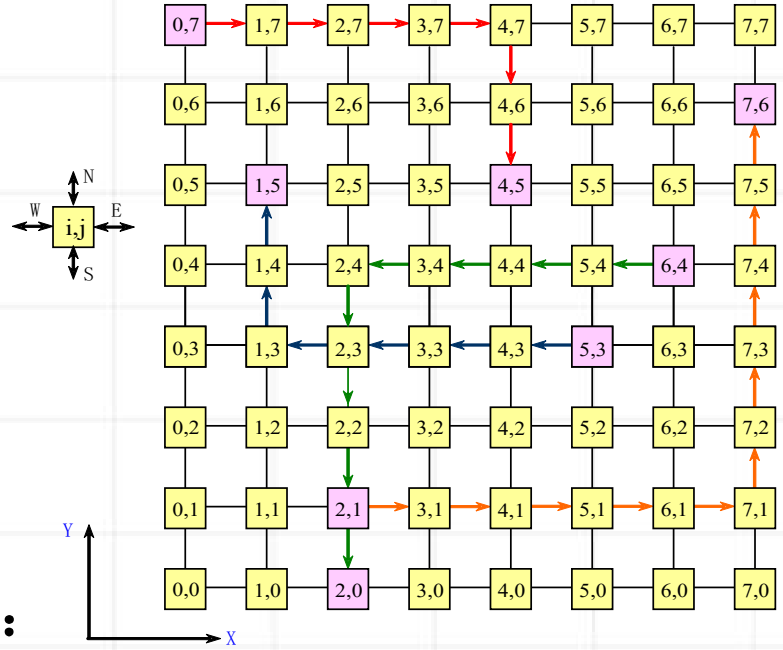
对于图所示的二维网格，确定以下4组“源结点-目的结点”所需要的路经。

（2，1）到（7，6）

（0，7）到（4，5）

（6，4）到（2，0）

（5，3）到（1，5）



解

所需要的路径如图所示。其中：

（2，1）到（7，6）需要用到的是一条东-北路径；

（0，7）到（4，5）需要用到的是一条东-南路径；

（6，4）到（2，0）需要用到的是一条西-南路径；

（5，3）到（1，5）需要用到的是一条西-北路径。

超立方体寻径

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$b=b_{n-1}b_{n-2}\cdots b_1b_0$的二进制编码。

设：源结点$s=s_{n-1}s_{n-2}\cdots s_1s_0$

目的结点$d=d_{n-1}d_{n-2}\cdots d_1d_0$

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，…，n，其中第i维对应于结点地址中的第i－1位。

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

超立方体寻径

① 计算方向位 $r_i = s_{i-1} \oplus d_{i-1}$，其中i=1，2，…，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←i+1。如果i≤n，则转第②步，否则退出。

超立方体寻径

例 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（$r_4r_3r_2r_1$）= $0110 \oplus 1101 = 1011$

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

超立方体寻径

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。

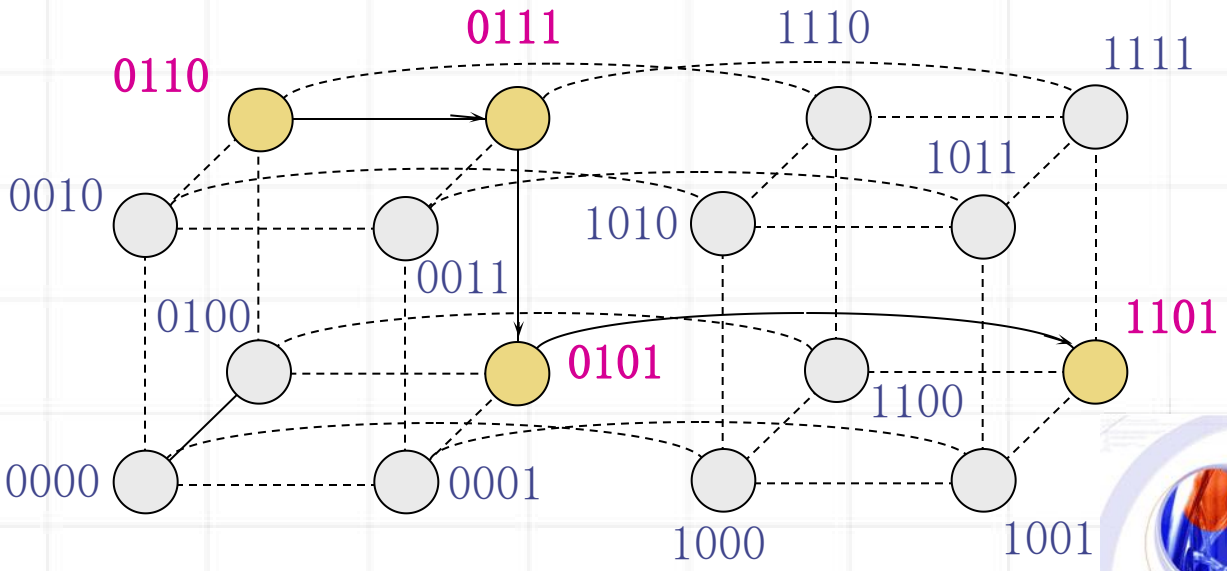
因此路径为：0110→0111→0101→1101

源：s=0110

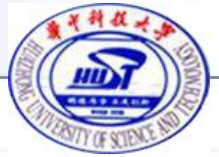
目的：d=1101

路径： 0110→0111

→0101→1101



第7章 互连网络（ICN）

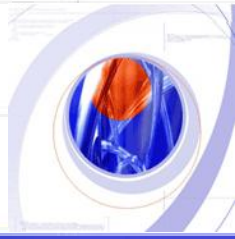


7.1 互连网络基本概念

7.2 互连函数

7.3 静态互连网络

7.4 动态互连网络

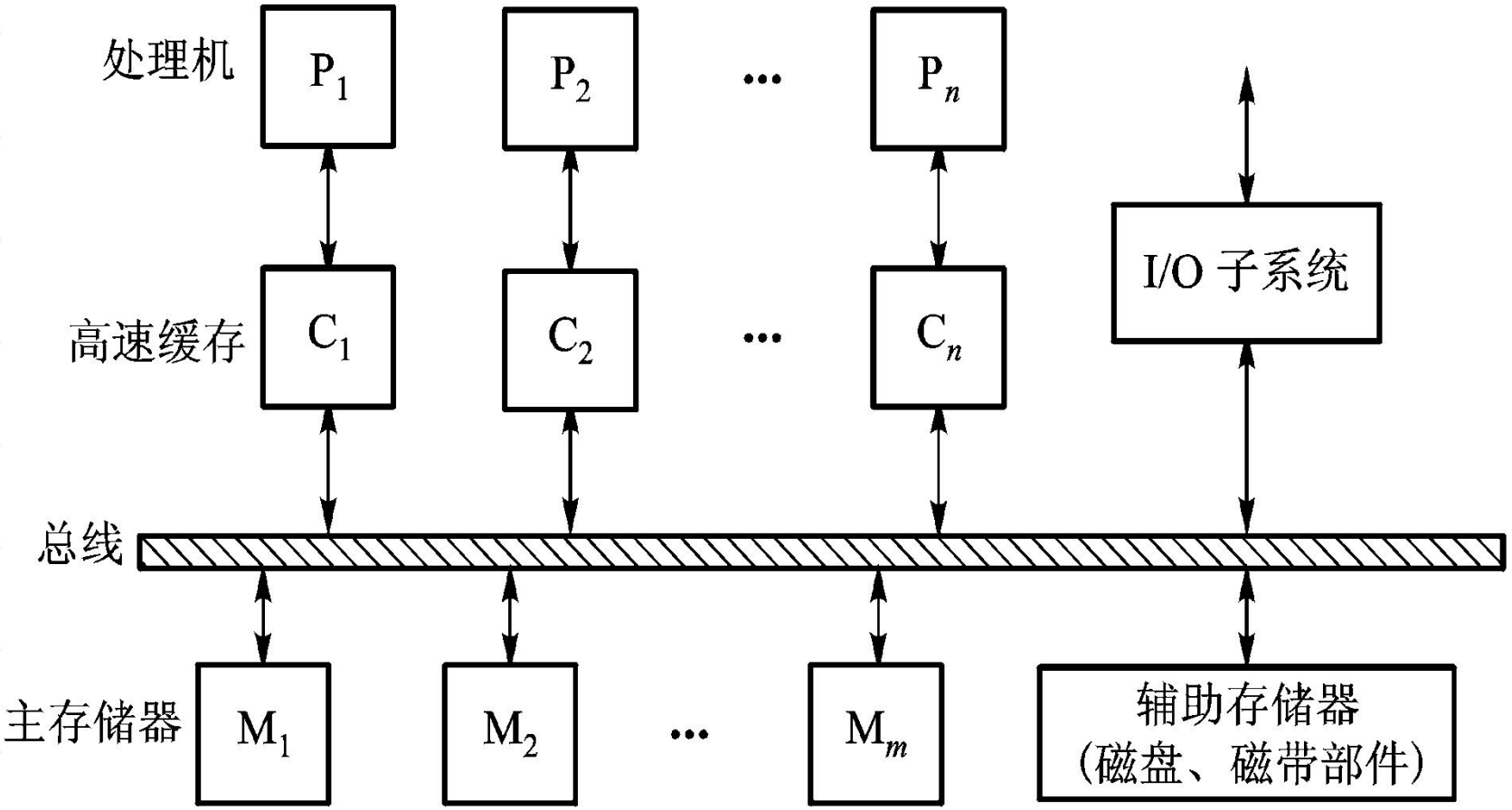


7.4 动态互连网络

一. 总线网络

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



3. 解决总线带宽较窄问题：采用多总线或多层次的总线

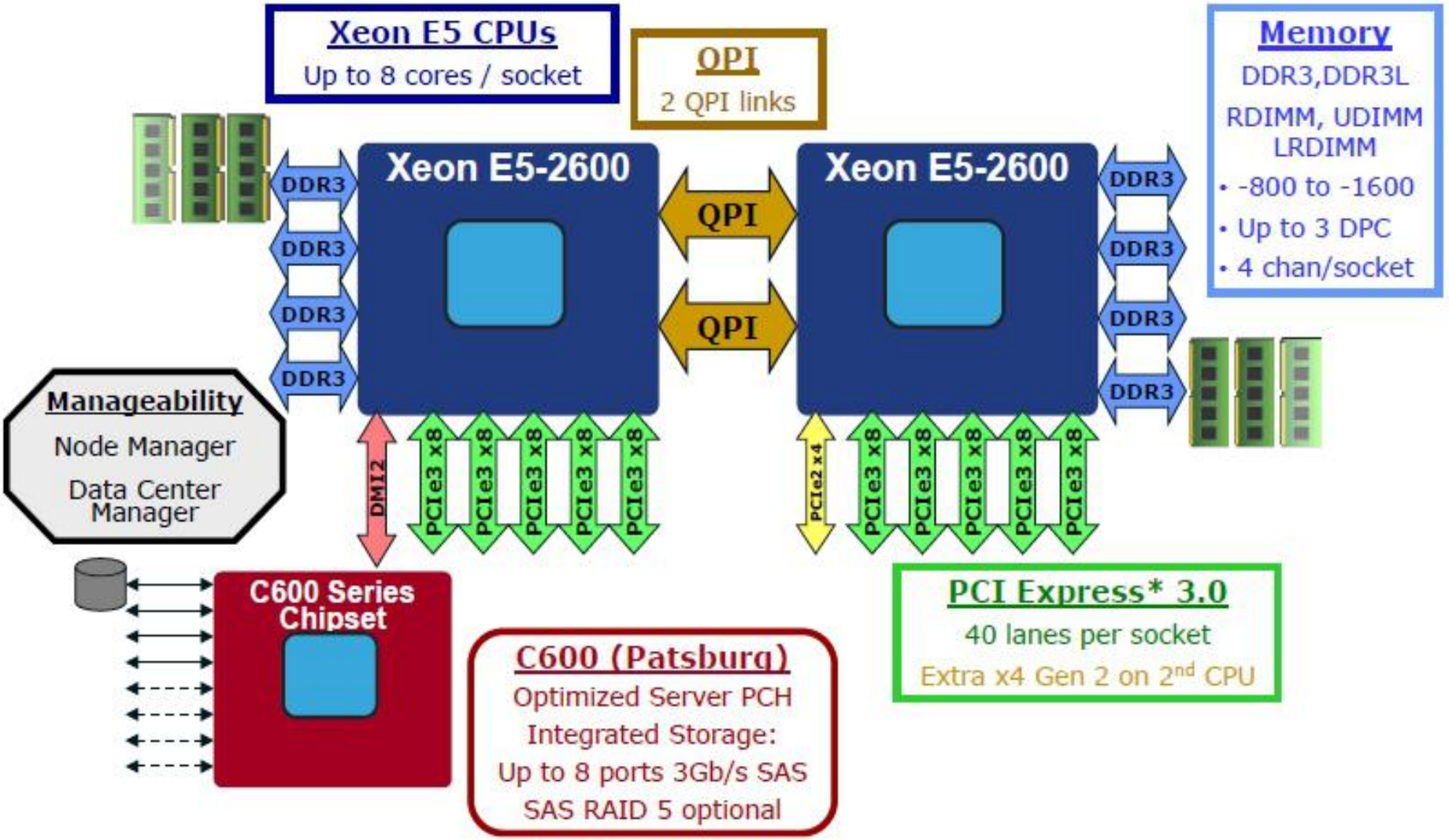
- 多总线是设置多条总线

  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。

实例：Intel Server Platform

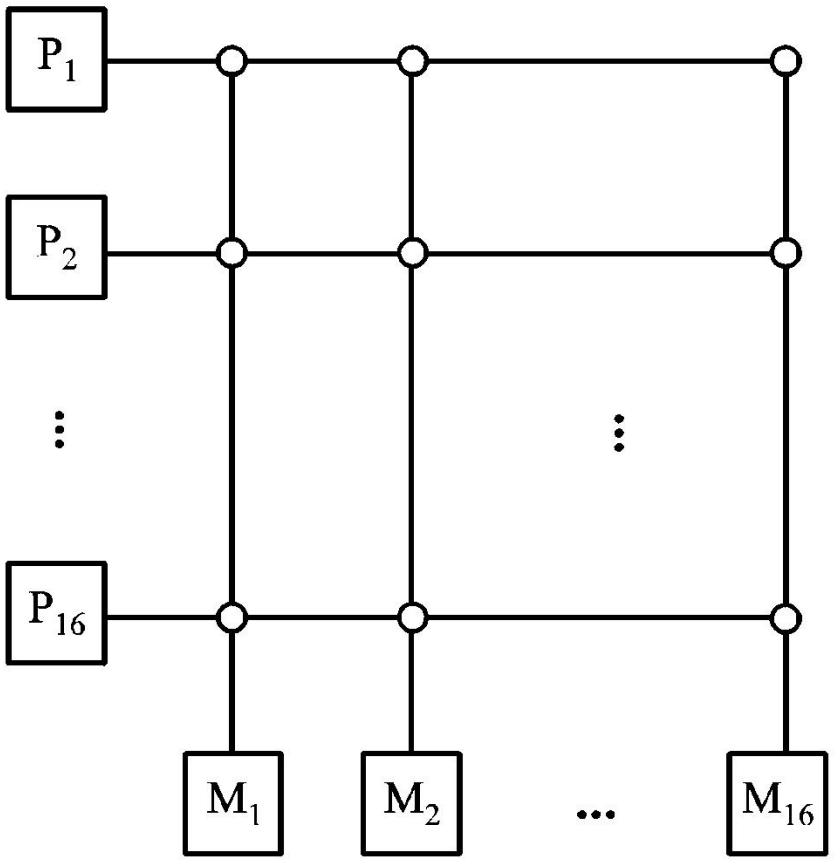


二. 交叉开关网络

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×n的交叉开关网络，可以无阻塞地实现n!种置换。
- 对一个n×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。

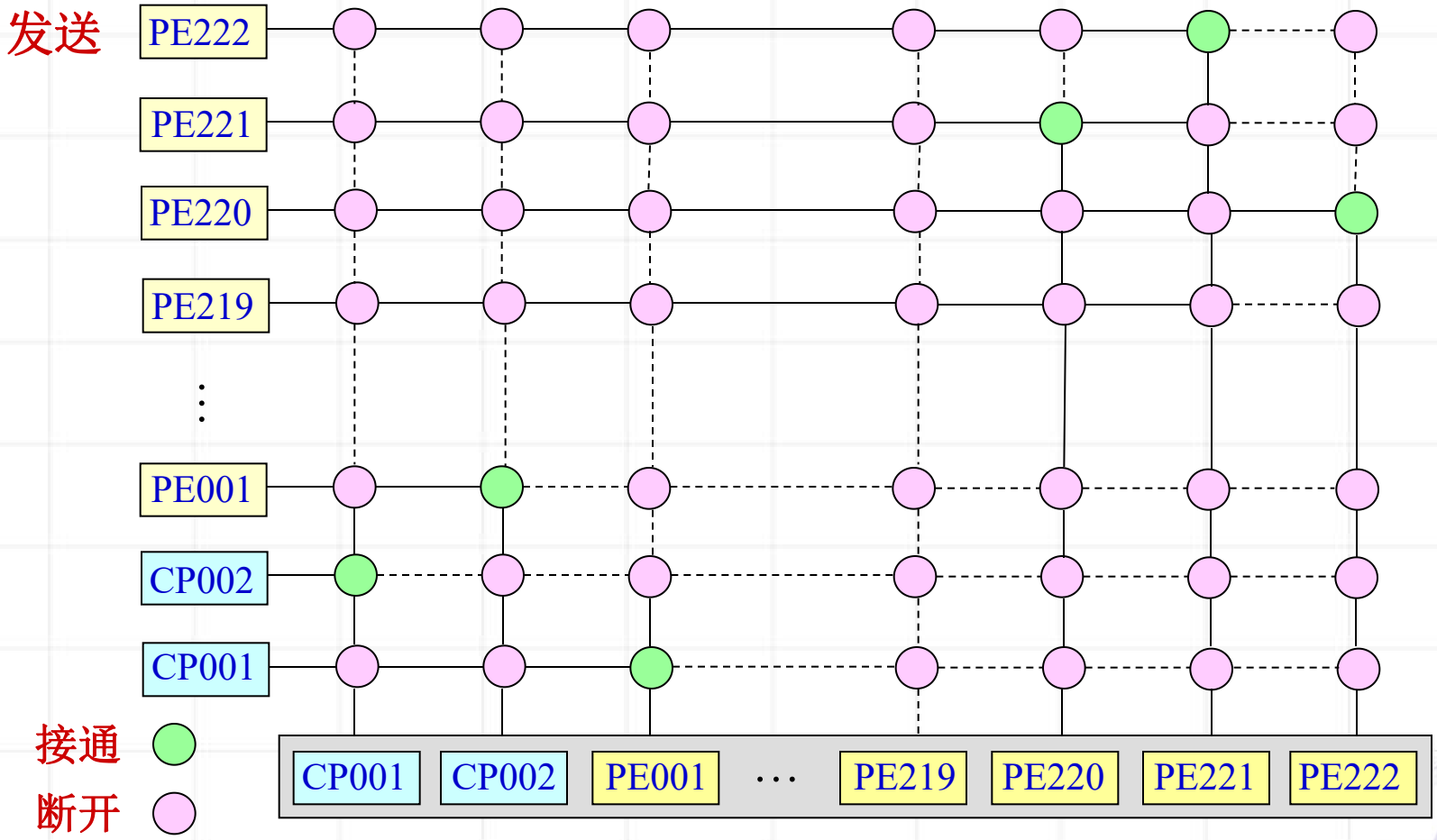
2. C.mmp多处理机的互连结构



- 16台PDP-11处理机与16个存储模块互连
- 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

7.4.1总线和交叉开关

3. Fujitsu向量并行处理机VPP500的开关网络（224×224）



PE：带存储器的处理机　　CP：控制处理机

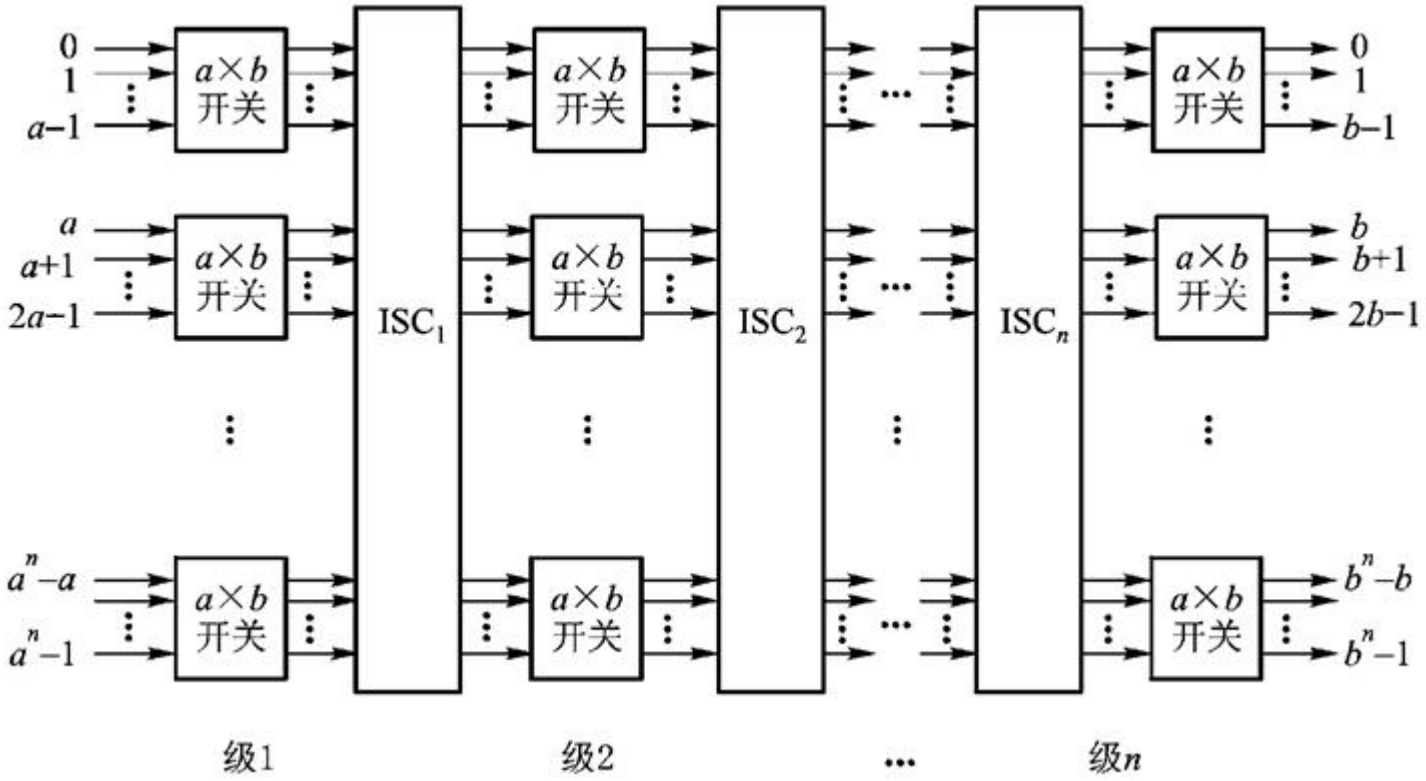
7.4.2 多级互连网络

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）
- MIN使用多级开关，使得数据在一次通过网络的过程中可以实现的置换种类更多。
- 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， 通常$a=b=2^k$，$k \geqslant 1$。
  - 相邻各级开关之间都有固定的级间连接

7.4.2 动态多级互连网络

- 通常在N个结点的网络中，多级ICN由n级构成（$n = \log_2 N$）。



- 经典的多级互连网有多级立方体网、多级混洗交换网和多级PM2I网。

7.4.2 动态多级互连网络

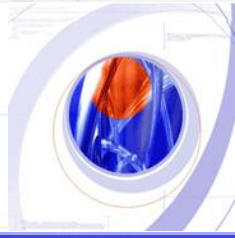
最简单的开关模块：2×2开关

- 二元交换开关的四种基本接通状态：“直连”、“交换”、“上播”和“下播”
- 在进行数据置换时只能使用前2种。

(a)直连　(b)交换　(c)上播　(d)下播

- 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：对各个开关模块的控制方式。
    - 级控制：每一级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≤i≤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

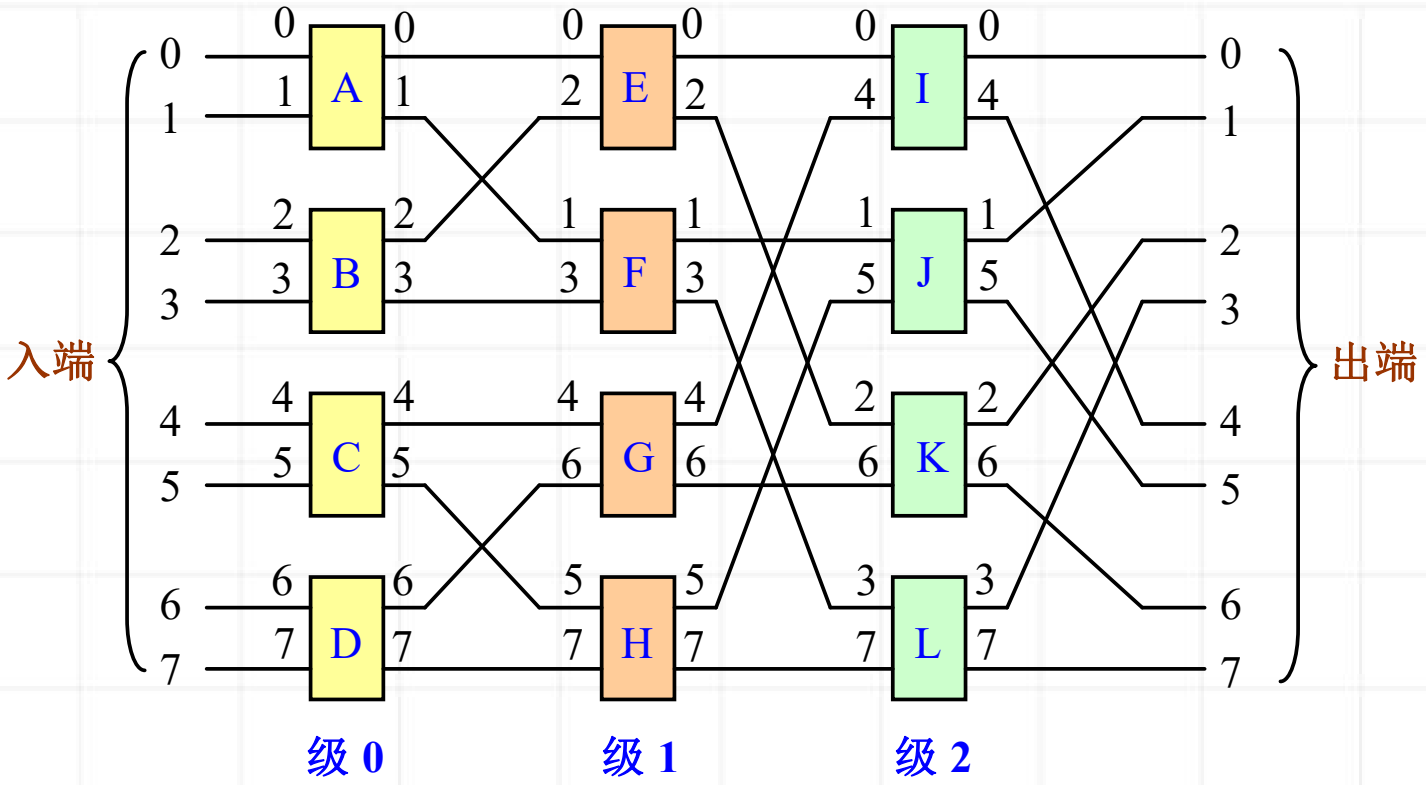


7.4.2 动态多级互连网络

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≤i≤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。
- 一个8个入端的多级立方体网络

7.4.2 动态多级互连网络



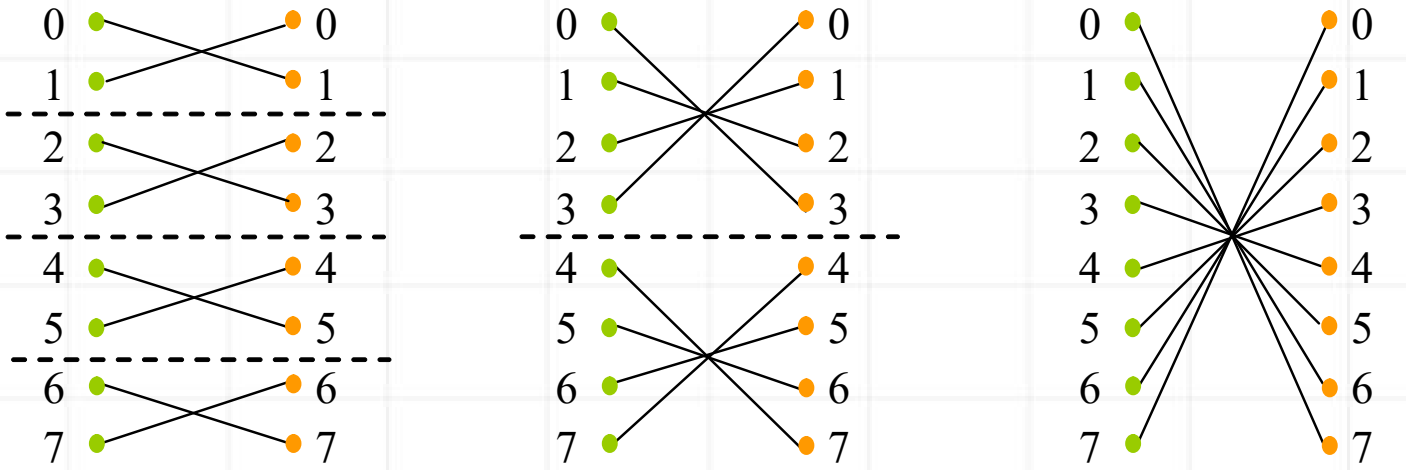
1. 间接二进制n方体网络采用单元控制。具有更大灵活性。
2. STARAN网络采用级控制和部分级控制。
   - 采用级控制时，实现交换功能；
   - 采用部分级控制时，实现移数功能。

7.4.2 动态多级互连网络

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

7.4.2 动态多级互连网络

- 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。

其中：

- $K_2k_1k_0$：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。

7.4.2 动态多级互连网络

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	$Cube_0$＋$Cube_1$＋ $Cube_2$

7.4.2 动态多级互连网络

- 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

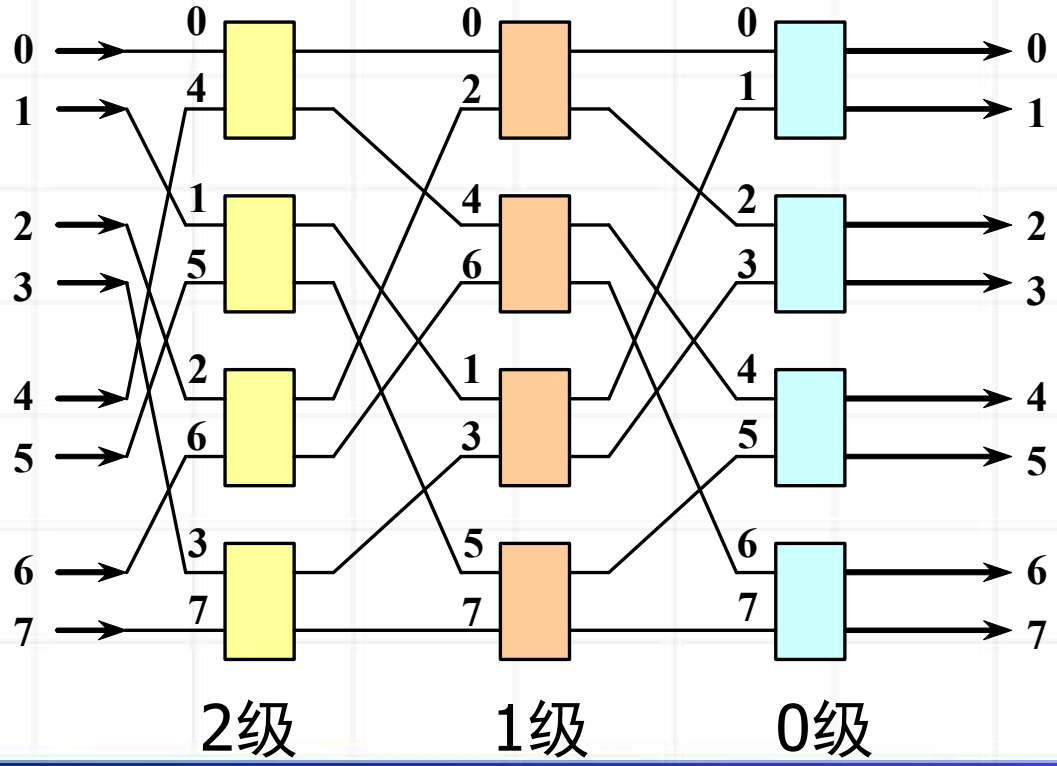
7.4.3 Omega网络和寻径

3. Omega网络（多级混洗—交换网络）

- N输入Omega网络的构成
  - 有$\log_2 N$级，每级$N/2$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。
  - 级间互连采用均匀洗牌连接方式
  - 每一级包含一个无条件混洗拓扑线路和一列可控的二元交换开关，前后重复，便于制造。
  - 各级编号是**n-1，……，0**，即按降序排列

7.4.3 Omega网络和寻径

- 8×8的Omega网络
  - 单独一级混洗拓扑线路可完成一次数据混洗（shuffle），
  - 单独一列二元交换开关在处于“交换”状态时可完成一次交换操作（Cube0）

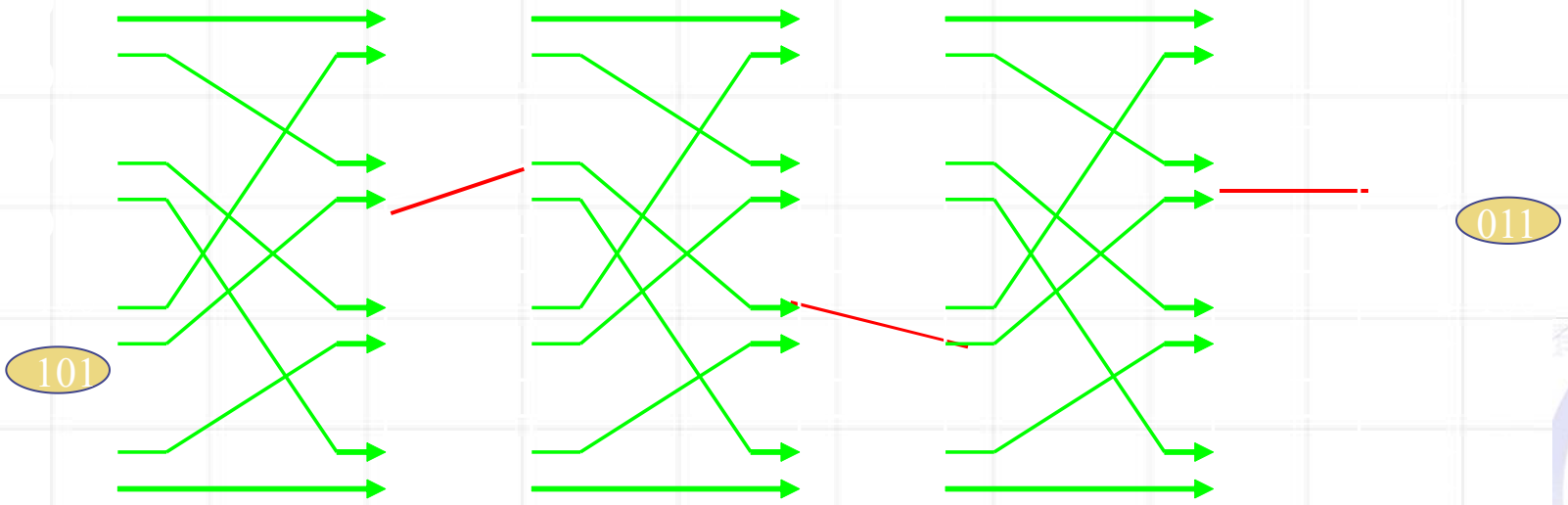


7.4.3 Omega网络和寻径

多级混洗—交换网络寻径算法

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biti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，二者异或结果为110B，于是从101B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”。如下图红线所示。



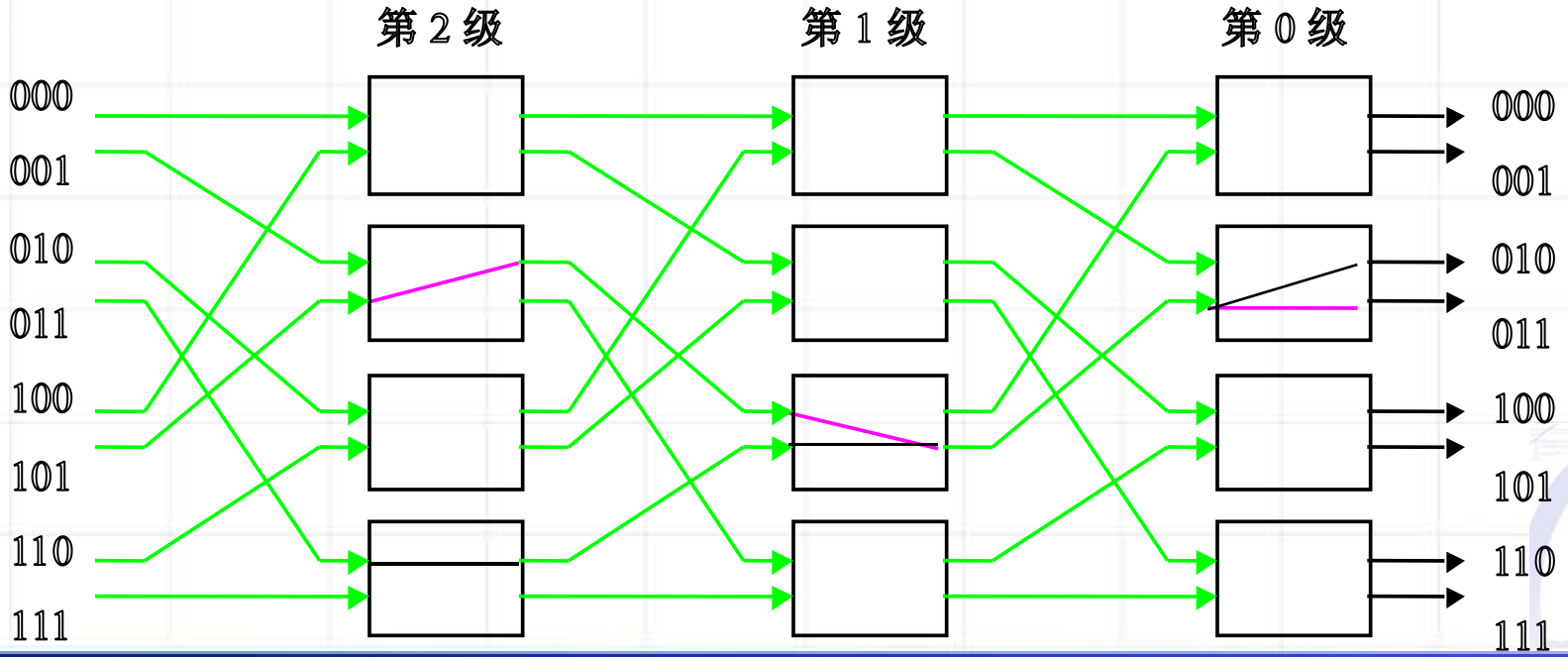
7.4.3 Omega网络和寻径

Omega网寻径冲突

- N输入互连网络，可能的置换数总共N!个
- N输入的Omega网，一次通过可实现$N^{N/2}$个 → 阻塞网络

给定传输101B→011B，二者异或结果为110B，路径红线所示。

给定传输011B→010B，二者异或结果为001B，路径黑线所示。



本章习题

- 9.9,9.13