

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](#)
- 9.2 [互连函数](#)
- 9.3 [静态互连网络](#)
- 9.4 [动态互连网络](#)
- 9.5 [消息传递机制](#)

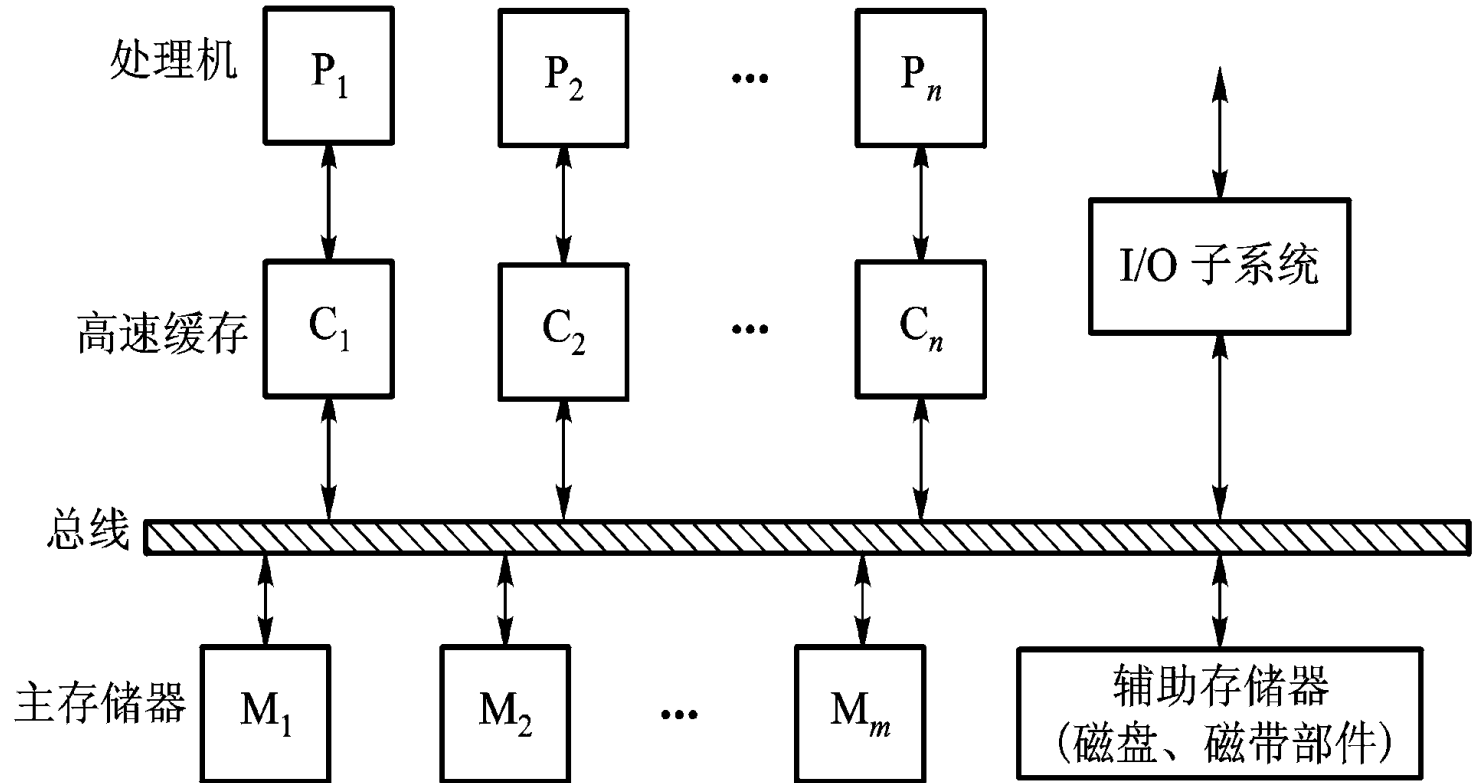
9.4 动态互连网络

9.4.1 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。
 - 多个请求时，仲裁逻辑将总线分配给一个请求，并按分时原则轮流为多个请求服务。
 - 多个功能模块之间的争用总线（Contention Bus）或时分总线（Time-sharing Bus）
 - 特点
 - 结构简单、实现成本低、带宽较窄

9.4 动态互连网络

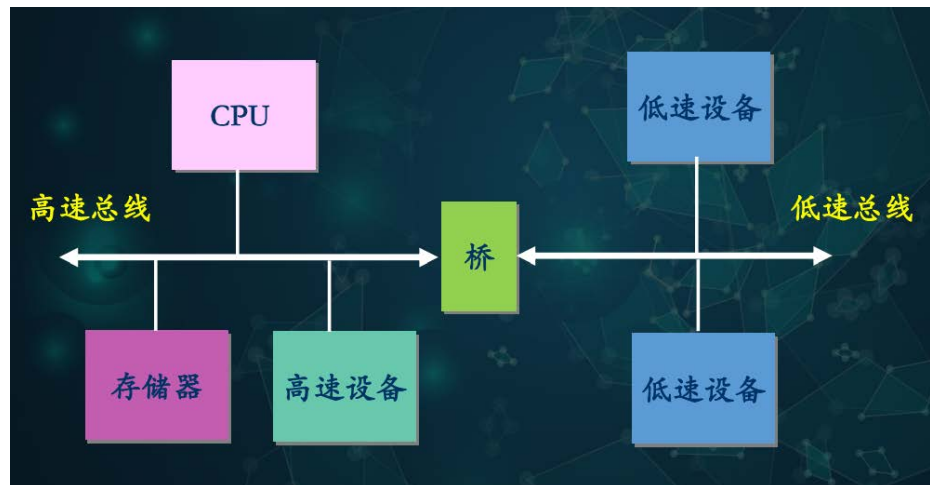
2. 一种由总线连接的多处理机系统



- 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。

9.4 动态互连网络

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总线连接。

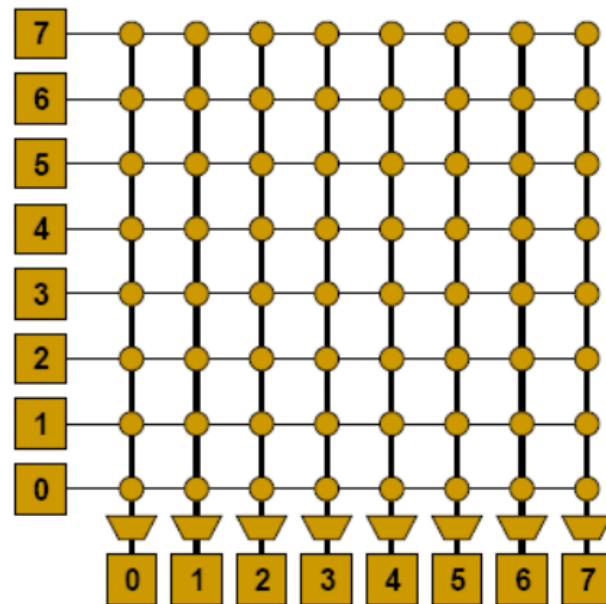


9.4 动态互连网络

9.4.2 交叉开关网络 (Crossbar)

1. 单级开关网络

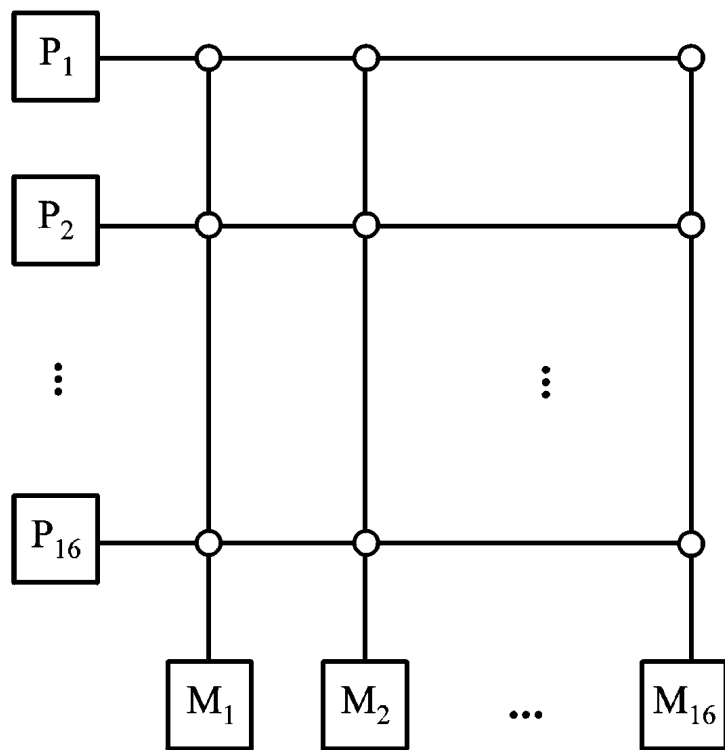
- 交叉点开关能在对偶（源、目的）之间形成动态连接，同时实现多个对偶之间的无阻塞连接。
- 开关可以根据程序的要求动态地设置为“开”或“关”。
- 带宽和互连特性最好。
- 一个 $n \times n$ 的交叉开关网络，可以无阻塞地实现 $n!$ 种置换。
- 对一个 $n \times n$ 的交叉开关网络来说，需要 n^2 套交叉点开关以及大量的连线。
 - 当 n 很大时，交叉开关网络所需要的硬件数量非常巨大。



9.4 动态互连网络

2. C. mmp 多处理机的互连结构 (Carnegie Mellon 多处理机)

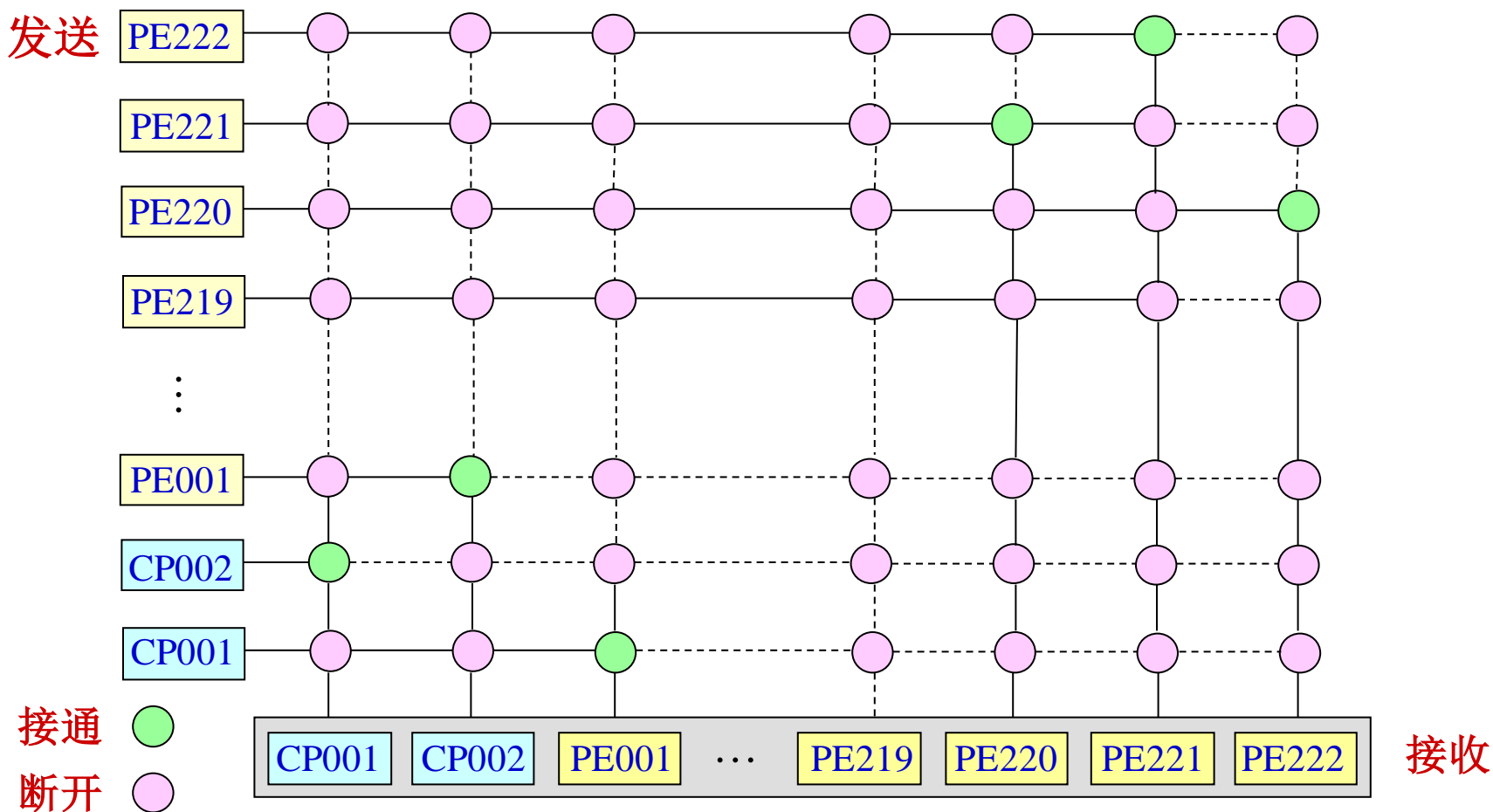
- 用 16×16 的交叉开关网络把16台PDP-11处理机与16个存储模块连在一起
- 最多可同时实现16台处理机对16个不同存储模块的并行访问



- 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 动态互连网络

3. Fujitsu向量并行处理机VPP500的交换开关网络 (224×224)

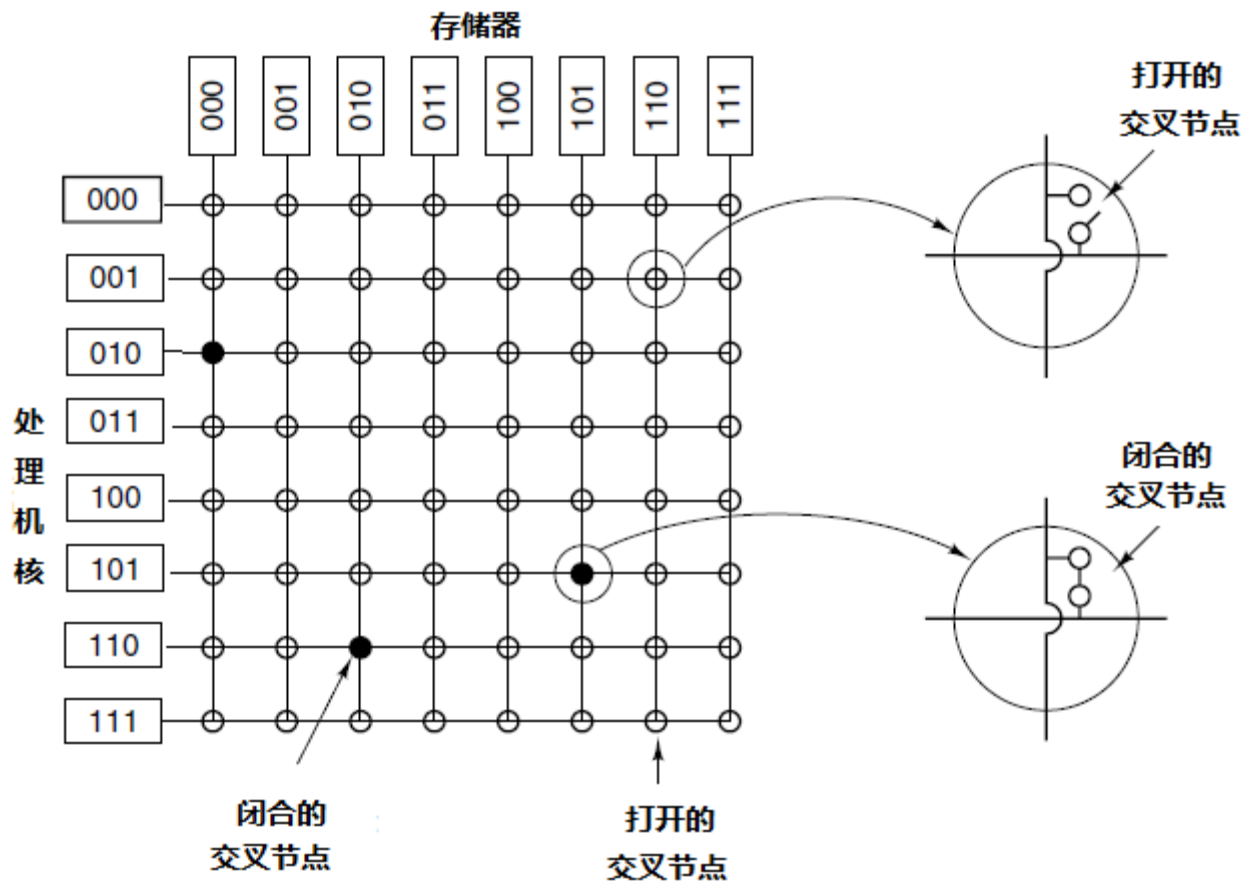


PE: 带存储器的处理机 CP: 控制处理机

每一行和每一列只能接通一个交叉点开关

9.4 动态互连网络

4. AMD公司的速龙（Athlon）X2双核处理器就是采用交叉开关来控制核心与外部通信的典型实例



交叉开关网络特点：

† 一对一（p2p）连接；

† $n \times n$ 网络一次可以实现 n 对连接（互不产生交叉）。

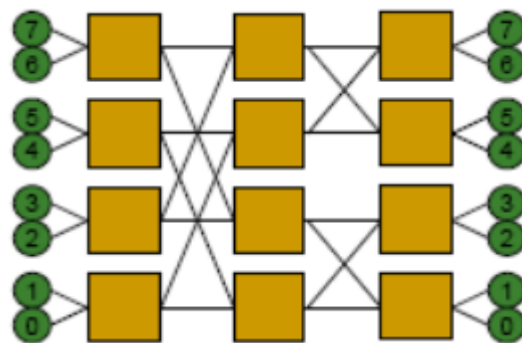
连接8个处理器核和8个内存模块的交叉开关结构

9.4 动态互连网络

9.4.3 多级互连网络

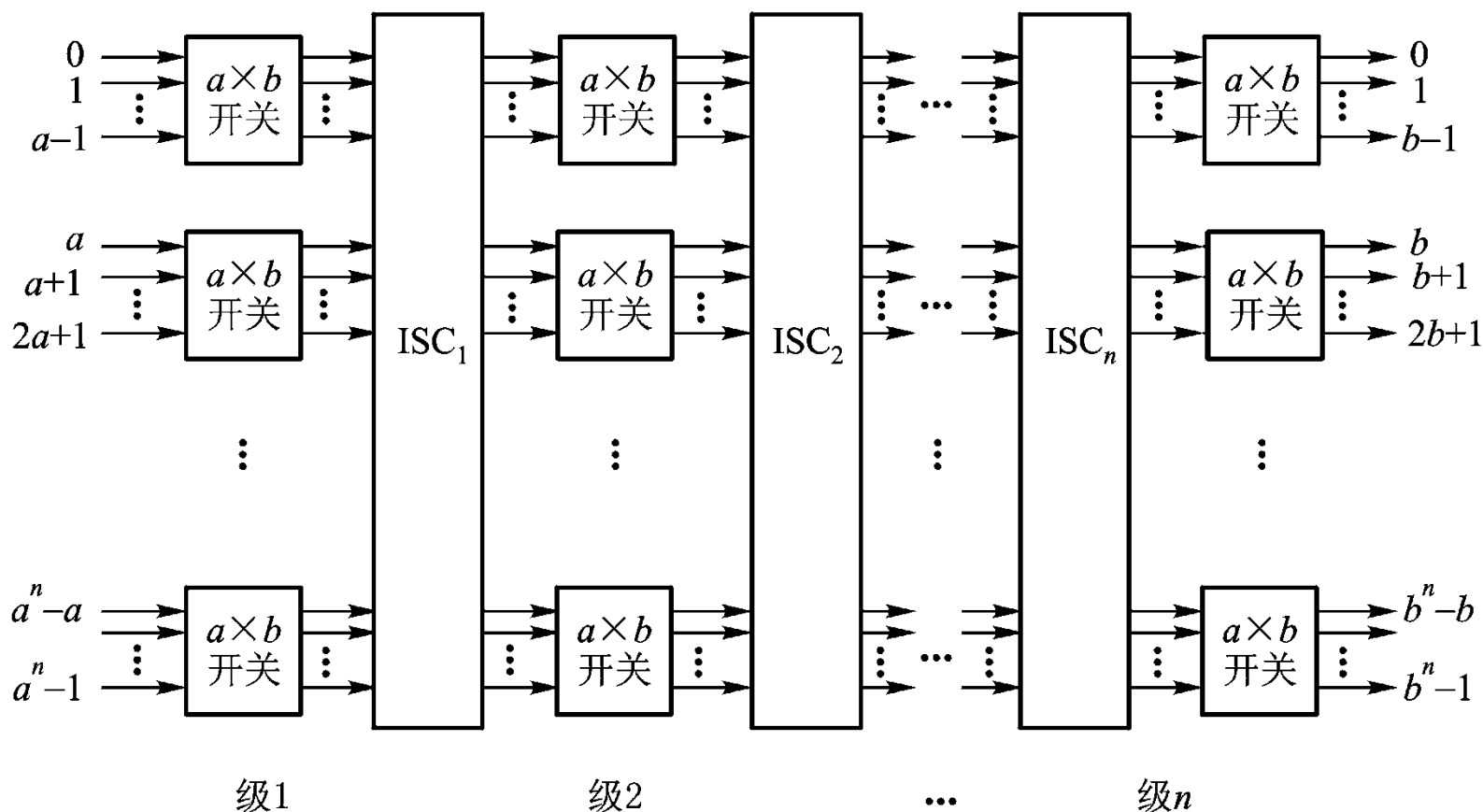
1. 多级互连网络的构成 (Multistage Interconnection Network)

- MIMD和SIMD计算机都采用多级互连网络MIN
- 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 $a \times b$ 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 经常选为相同的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 ($ISC_1, ISC_2, \dots, ISC_n$)



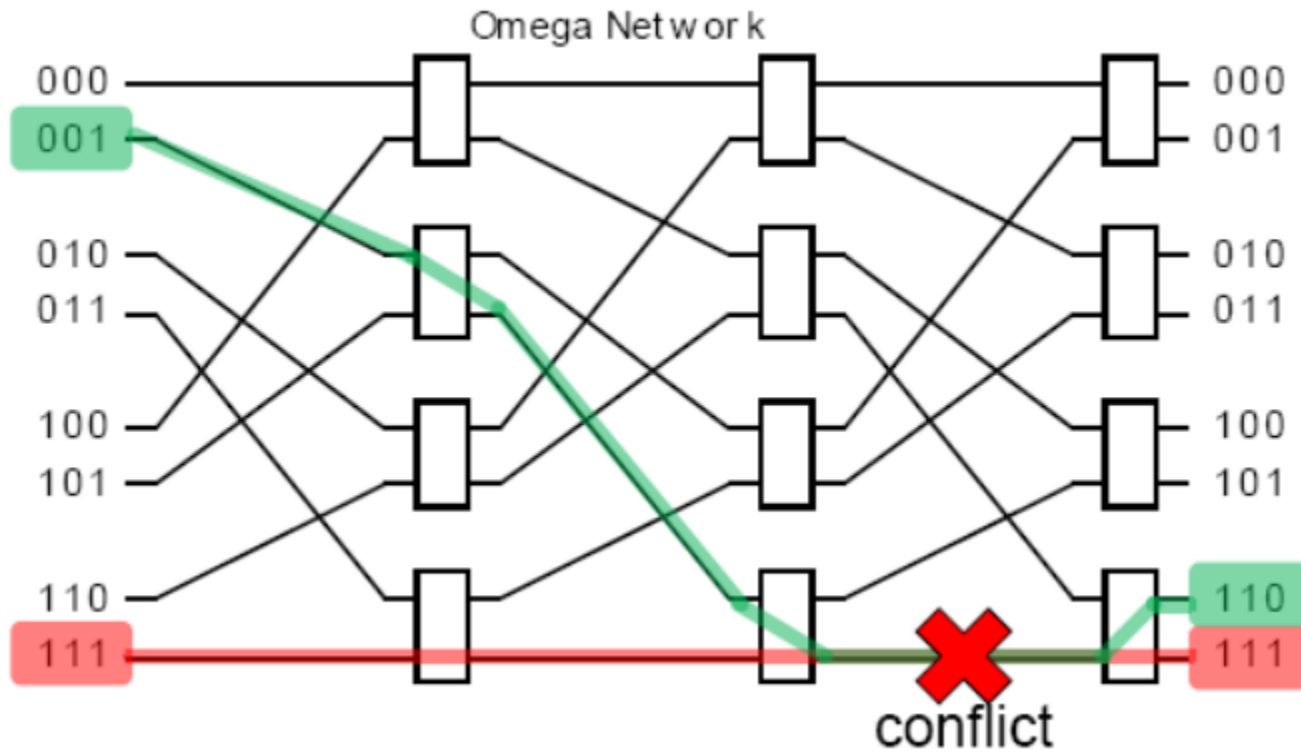
9.4 动态互连网络

➤ 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。

9.4 动态互连网络



Q: Blocking or non-blocking?

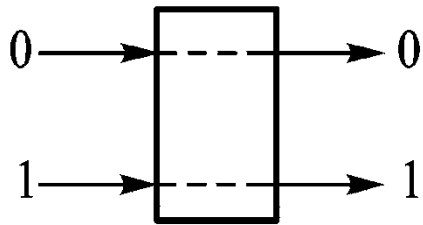
开关模块每个输入可以一个或多个输出相连，但是在输出端不许发生冲突。
即允许一对一和一对多映射，不允许多对一映射。

*与函数正好相反，如 $f(x)=x^2$ 。

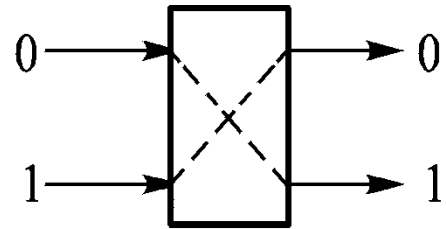
9.4 动态互连网络

最简单的开关模块： 2×2 开关

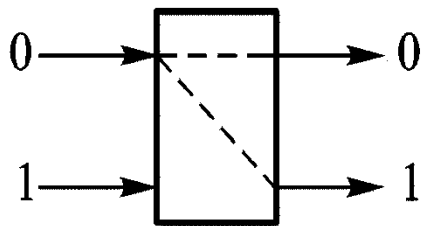
- 二元交换开关的四种基本接通状态：“直连”、“交换”、“上播”和“下播”
- 在进行数据置换时只能使用前2种。



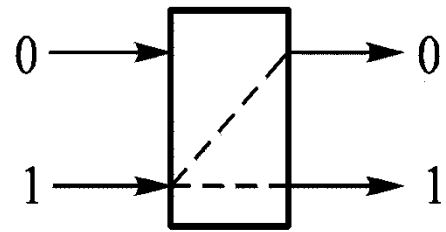
(a) 直送



(b) 交叉



(c) 上播



(d) 下播

9.4 动态互连网络

➤ 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	$n!$

9.4 动态互连网络

- 各种多级互连网络的**区别**在于所用开关模块、控制方式和级间互连模式的不同。
 - **控制方式**：对各个开关模块进行控制的方式。
 - **级控制**：每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**：几个开关合用一个控制信号。
 - 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

9.4 动态互连网络

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。
 - 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 动态互连网络

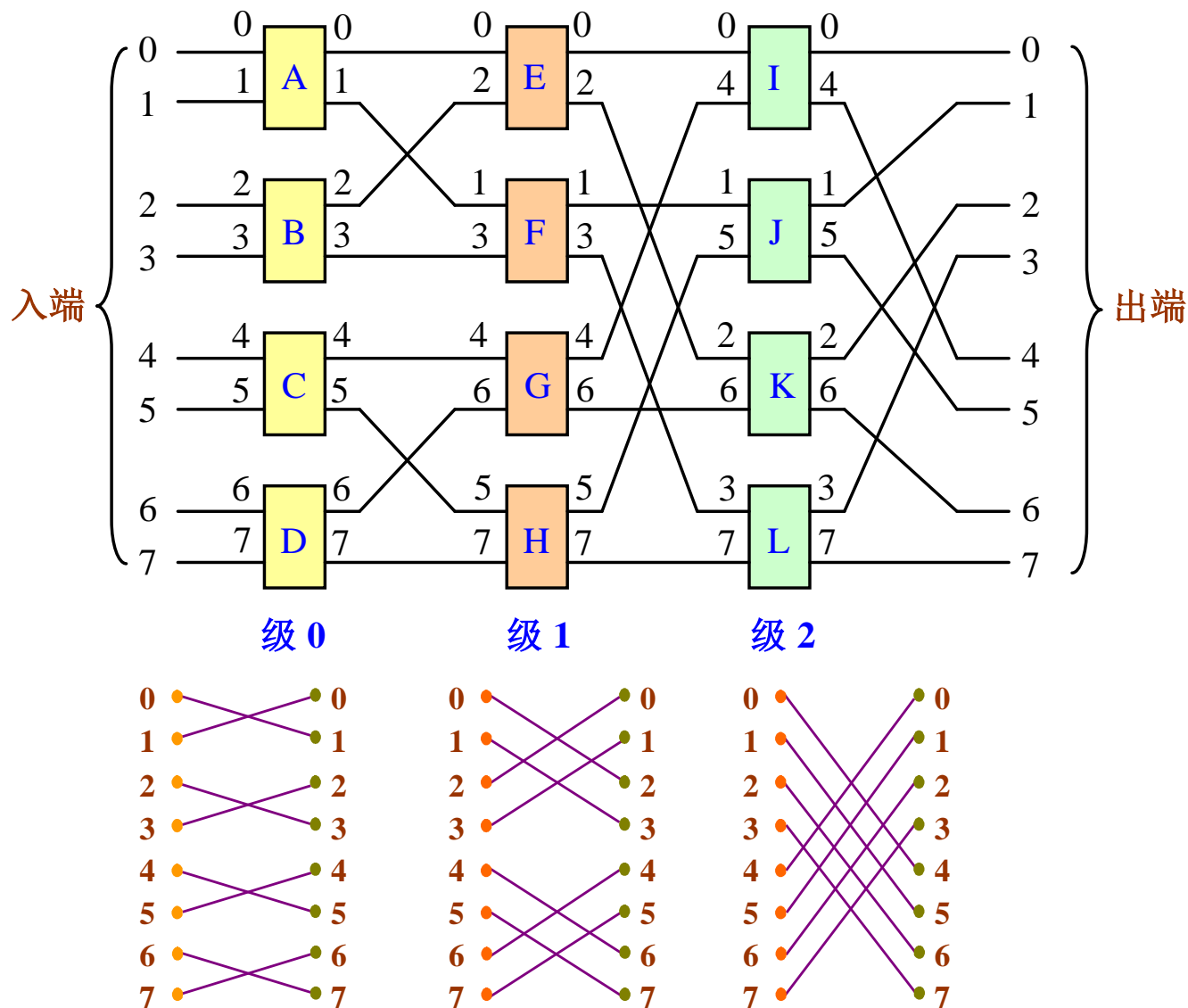
一个8个入端的

STARAN网络

(多级立方体网络)

采用 2×2 开关

当第 i 级开关处于交换状态时，
实现的是 Cube_i 互
联函数。



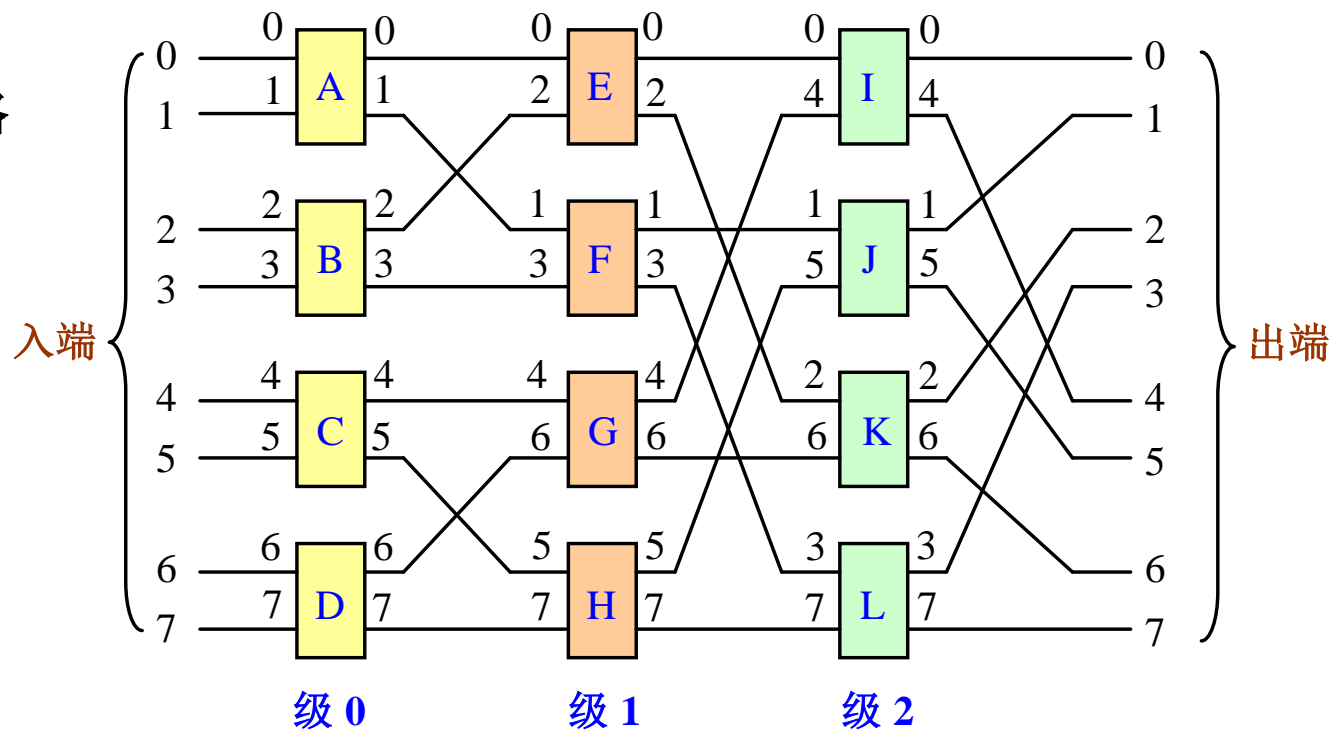
(a) Cube_0 交换函数

(b) Cube_1 交换函数

(c) Cube_2 交换函数

9.4 动态互连网络

一个8个入端的多级立方体网络



多级立方体
网络

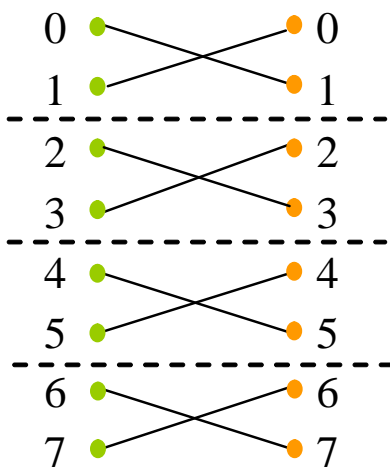
- STARAN网络采用级控制和部分级控制。
 - ❑ 采用级控制时，所实现的是交换功能；
 - ❑ 采用部分级控制时，则能实现移数功能。
- 间接二进制n方体网络则采用单元控制。
 - ❑ 具有更大的灵活性。

9.4 动态互连网络

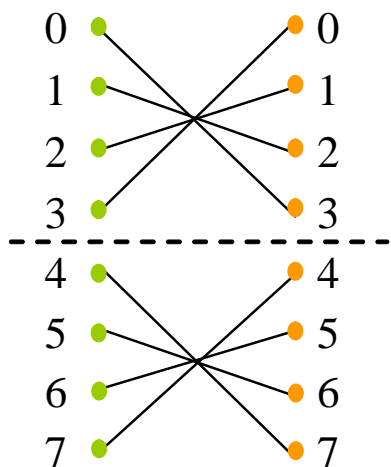
交换

将有序的一组元素头尾对称地进行交换。

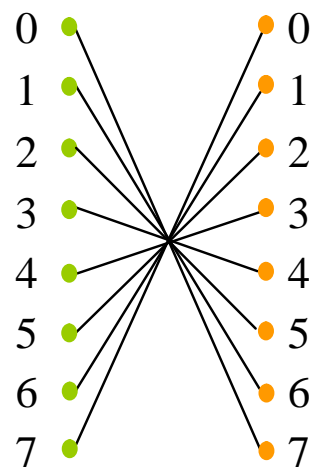
例如：对于由8个元素构成的组，各种基本交换的图形：



(a) 4 组 2 元交换



(b) 2 组 4 元交换



(c) 1 组 8 元交换
(全交换/镜像交换)

8个元素的基本交换图形

9.4 动态互连网络

- 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。

$$110 = 001 \oplus 111$$

9.4 动态互连网络

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	$\text{Cube}_0 + \text{Cube}_1$
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	$\text{Cube}_0 + \text{Cube}_2$
110	6 7 4 5 2 3 0 1	4组2元交换+ 1组8元交换	$\text{Cube}_1 + \text{Cube}_2$
111	7 6 5 4 3 2 1 0	1组8元交换	$\text{Cube}_0 + \text{Cube}_1 + \text{Cube}_2$

例: $110 = 001 \oplus 111$

9.4 动态互连网络

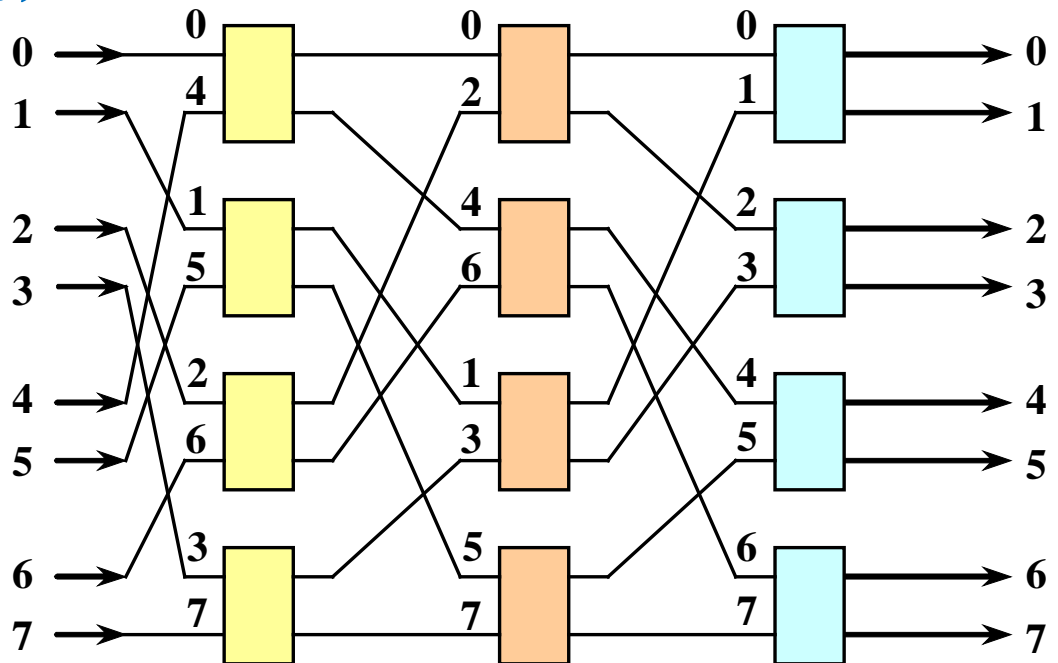
- 当STARAN网络用作移数网络时，采用部分级控制，控制信号的分组和控制结果。

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 动态互连网络

3. Omega网络（多级混洗—交换网络）

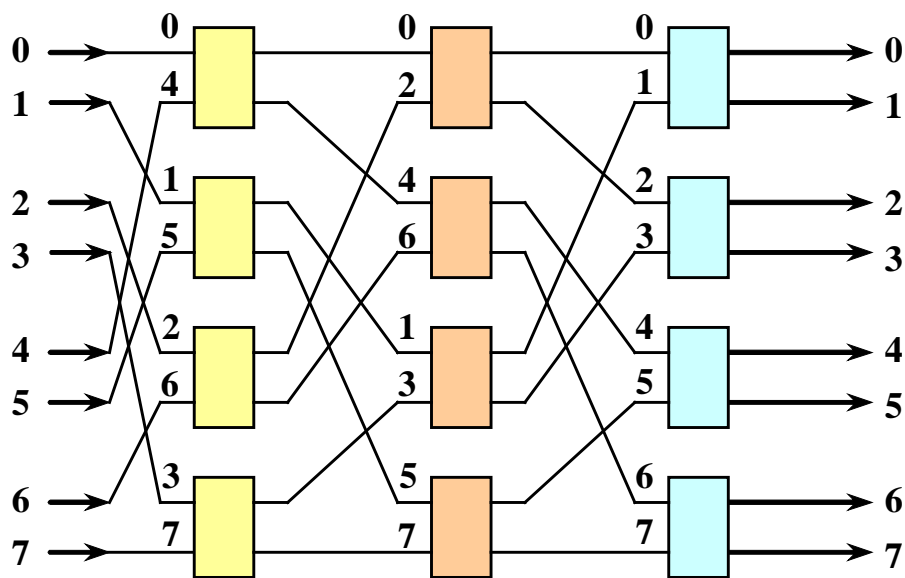
- 一个 8×8 的Omega网络
 - 每级由4个4功能的 2×2 开关构成
 - 级间互连采用均匀洗牌连接方式
 - 单独一级混洗拓扑线路（ISC）可完成一次数据混洗（shuffle）
 - 单独一列二元交换开关在处于“交换”状态时可完成一次交换操作（Cube0）



9.4 动态互连网络

➤ 一个 N 输入的Omega网络

- 有 $\log_2 N$ 级，每级用 $N/2$ 个 2×2 开关模块，共需要 $(N \log_2 N)/2$ 个开关。
- 每个开关模块均采用单元控制方式。
- 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。

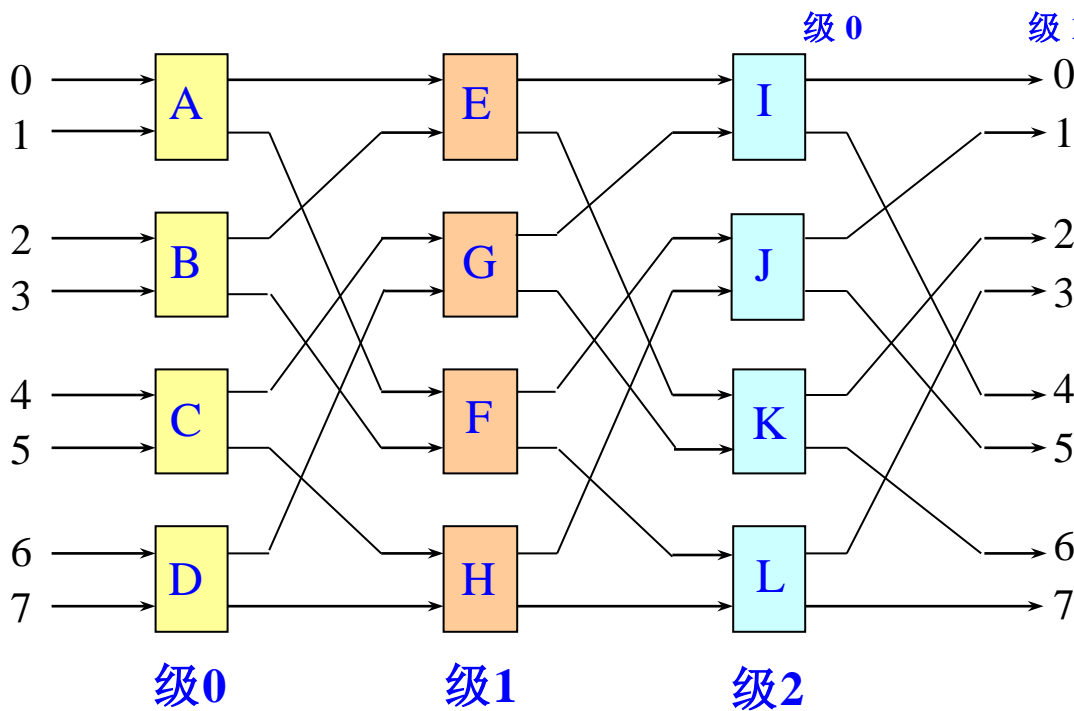
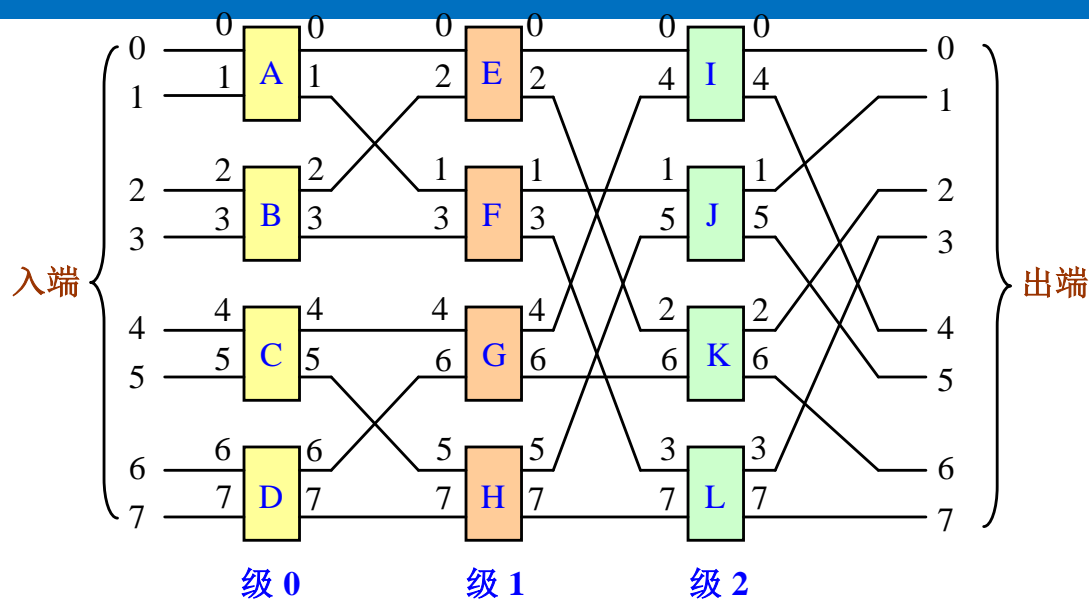


Omega网络

9.4 动态互连网络

➤ N=8的多级立方体

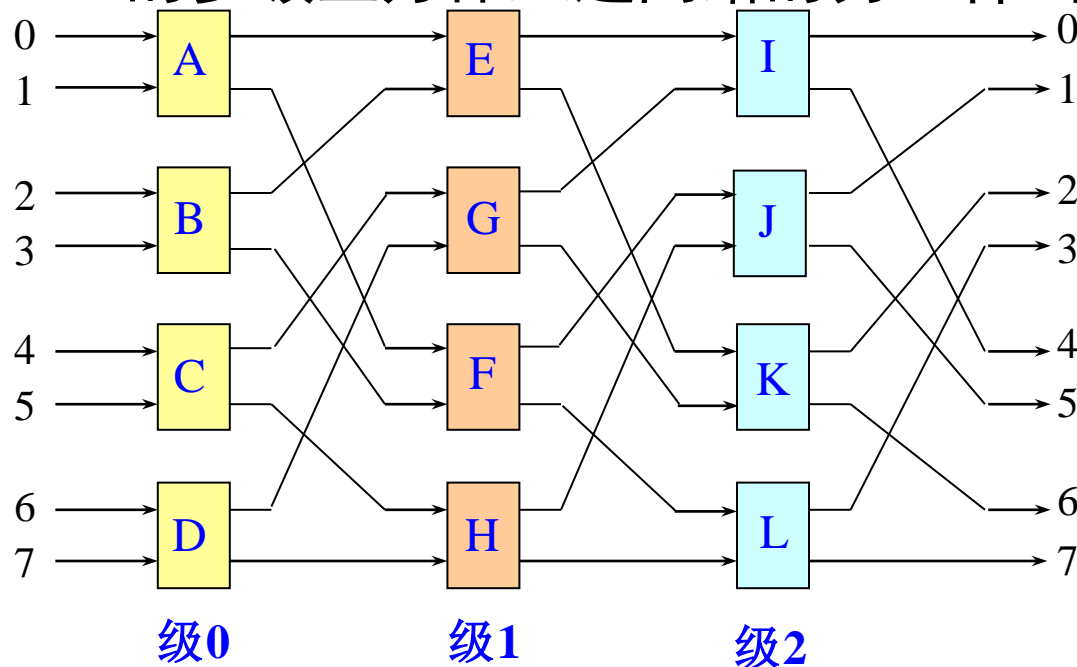
互连网络的另一种画法



上面的多级立方体网络中的开关 **F** 和 **G** 对调（保持原来的链接）

9.4 动态互连网络

➤ N=8的多级立方体互连网络的另一种画法



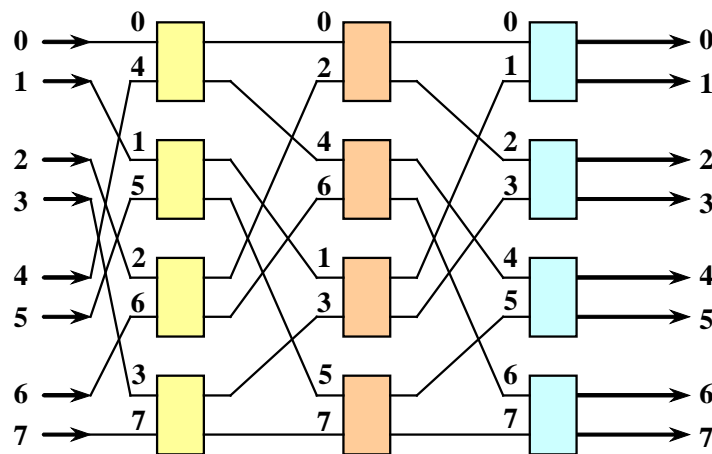
前面的多级立方体网络中的开关**F**和**G**对调（保持原来的链接）



如果Omega网络也采用2功能交换开关，那么Omega网络和间接二进制n方体网络就互为逆网络。

互逆网络：如果把出端当入端，入端当出端，则原网络变为另一个互连网络，称此网络为原网络的逆互连网络。

Omega网络



9.4 动态互连网络

9.4.4 动态互连网络的比较

成本高，复杂度高，性能高

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

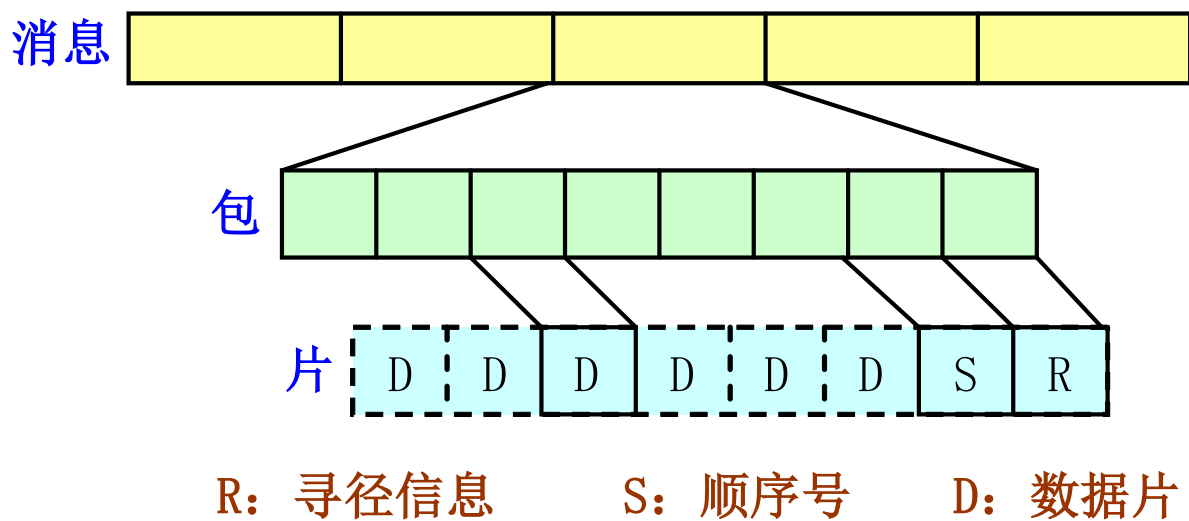
当源结点和目的结点之间没有直接的连接时，消息需要经过中间的结点进行传递。**寻径**就是用来实现这种传递的通信方法和算法。有的称之为**路由**。

9.5.1 消息寻径方案

1. 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（Packet）**”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 消息传递机制



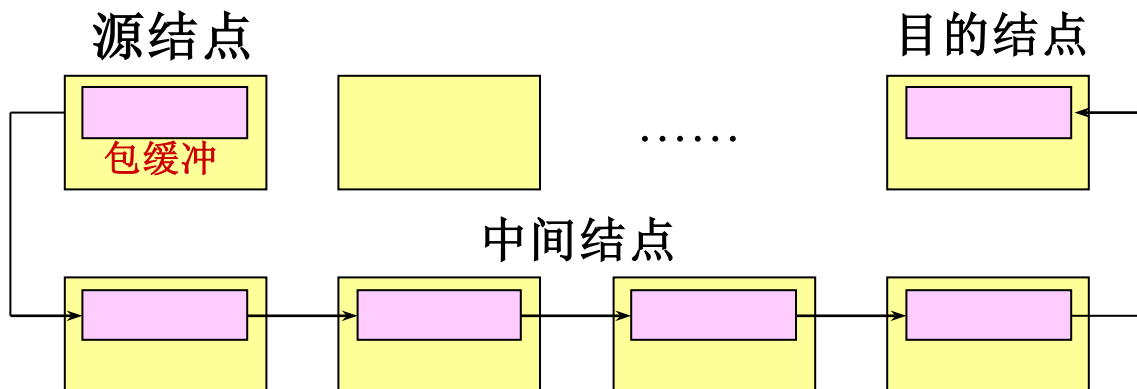
消息、包和片的格式

9.5 消息传递机制

- **包**：包含寻径所需目的地址的基本单位。
 - 一条消息（**Message**）中的各个包都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（Flit）**”。
 - 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 消息传递机制

2. 四种寻径方式



消息寻径方式两大类：◎线路交换/电路交换（Circuit Switching）

——以消息粒度运行

◎包交换/分组交换（Packet Switching）

（存储转发、虚拟直通、虫蚀）

——以包、片粒度分配

9.5 消息传递机制

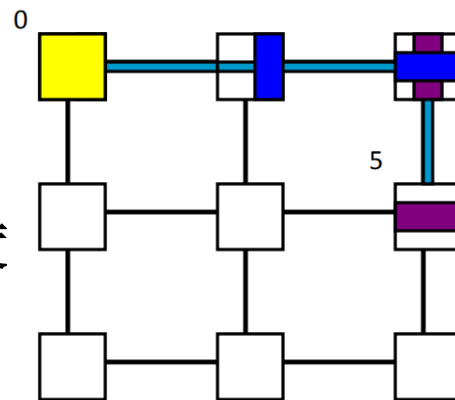
➤ **线路交换（电路交换）**：在线路交换方式下，在传递一个信息之前，需要先建立一条从源结点到目的结点的物理通路，然后再传递信息。

□ 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（Probe）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。

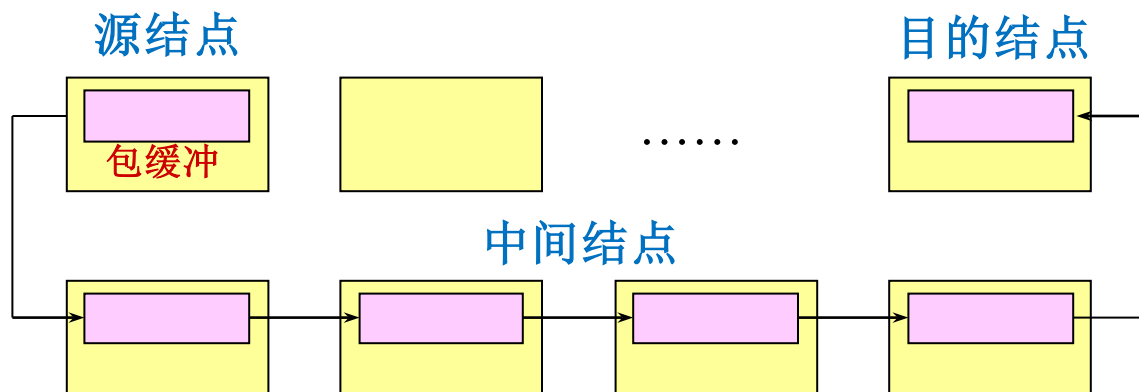
9.5 消息传递机制

➤ **存储转发（Store and Forward）**：最简单的分组交换方式。

- 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；*时延大，与结点距离成正比



(a) 存储转发

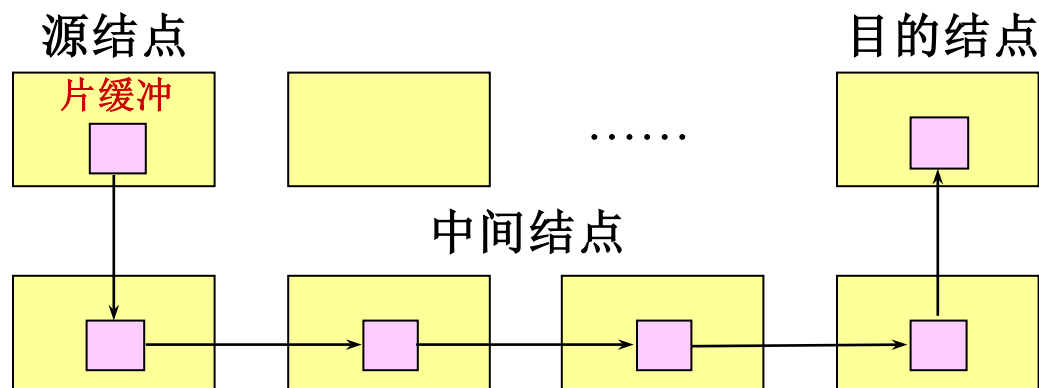
9.5 消息传递机制

- **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 消息传递机制

- **虫蚀方式 (Wormhole)**：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。



(b) 虫蚀方式

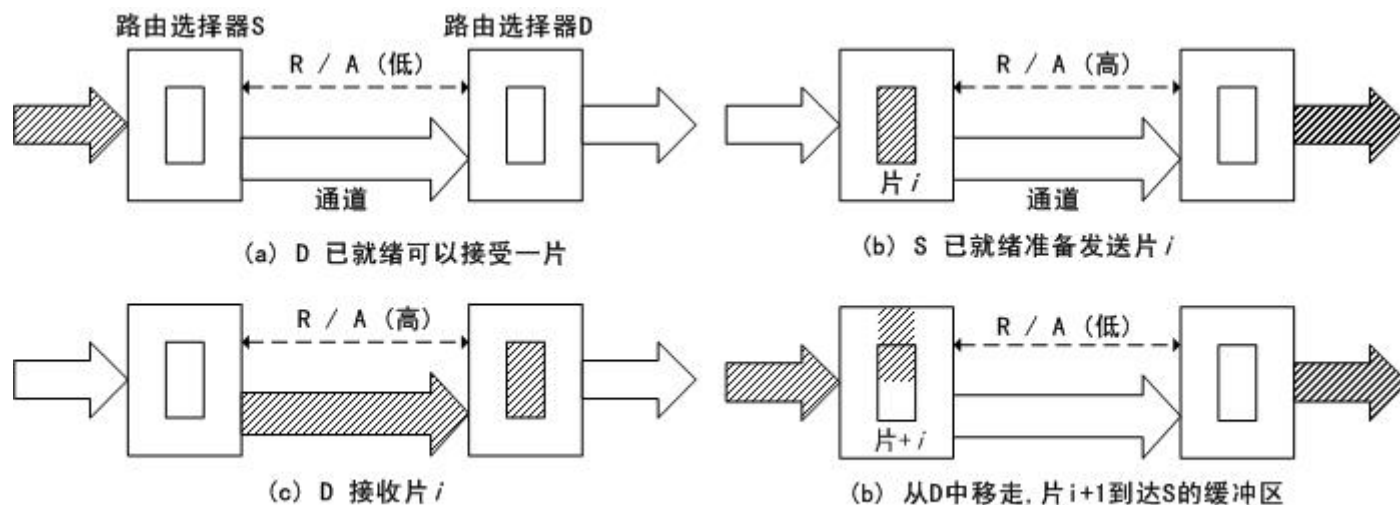
9.5 消息传递机制

- 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。
- 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。
- 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, $L \gg L_f \times D$ 。

9.5 消息传递机制



两个虫蚀寻经器之间的握手协议

□ 优点

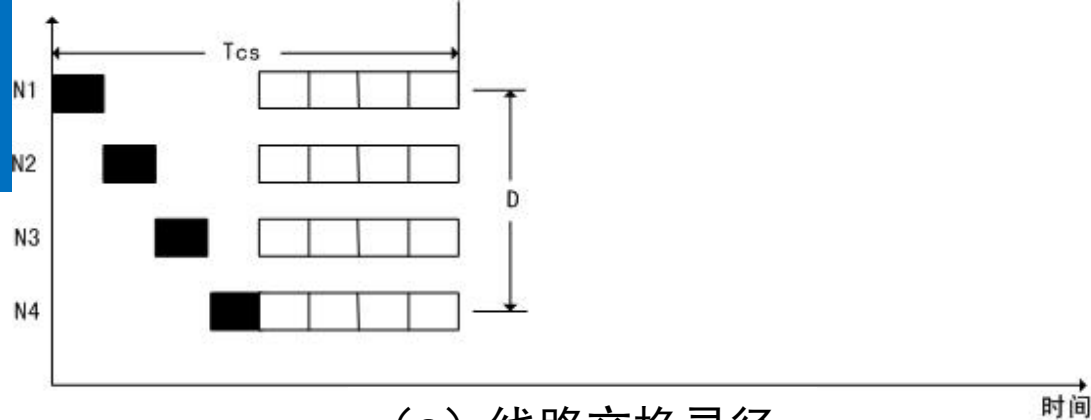
- 每个节点的缓冲器较小，易于VLSI实现；
- 有较小的网络传输延迟；
- 通道共享性好，利用率高；
- 易于实现选播和广播通信模式。

□ 缺点

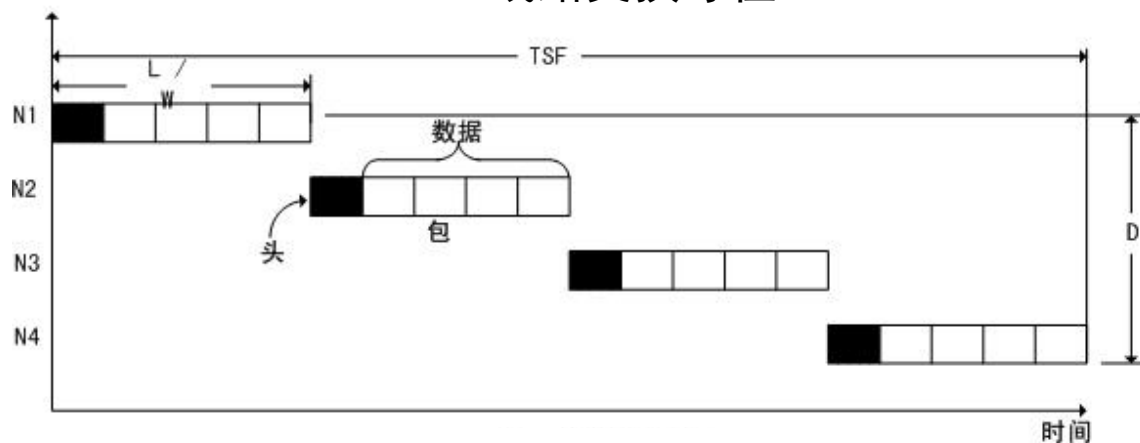
- 链路利用率低：当包的一片被阻塞时，整个包的所有片都将被阻塞在所在结点，占用了结点资源。

9.5 消息传递机制

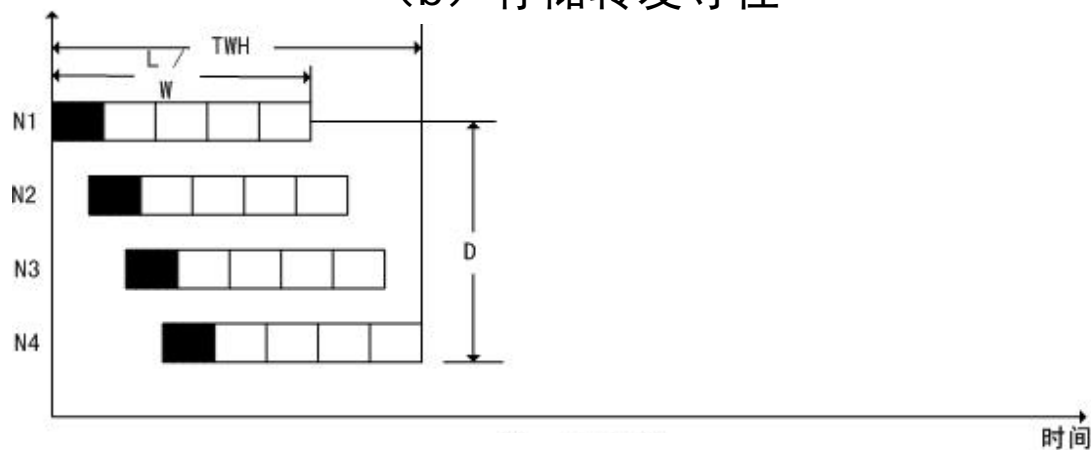
几种寻径方式的时空图



(a) 线路交换寻径



(b) 存储转发寻径

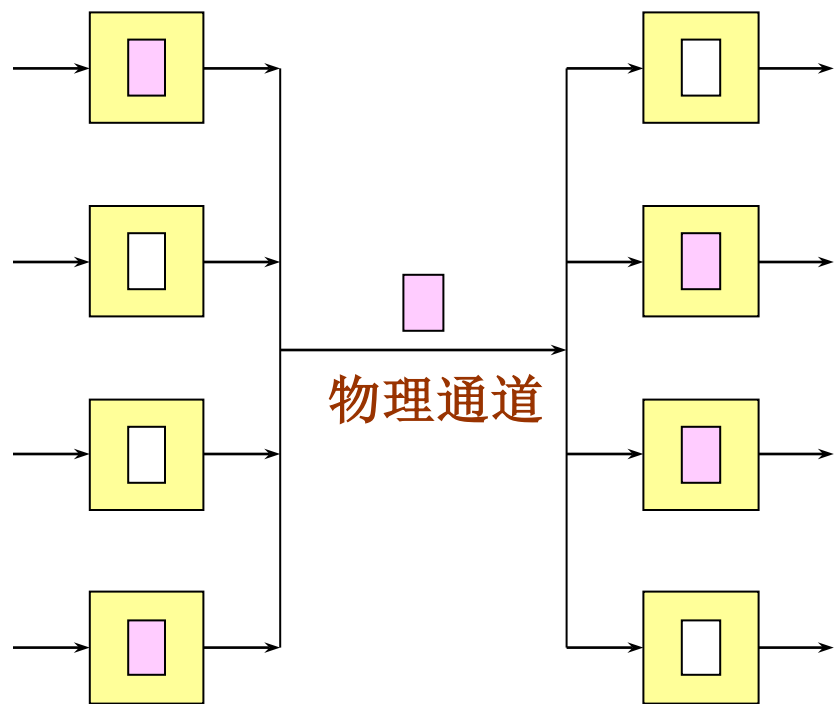


(c) 虫蚀寻径

9.5 消息传递机制

9.5.2 死锁与虚拟通道

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。

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 消息传递机制

2. 避免死锁

- 缓冲区或通道上的循环等待会引起死锁。

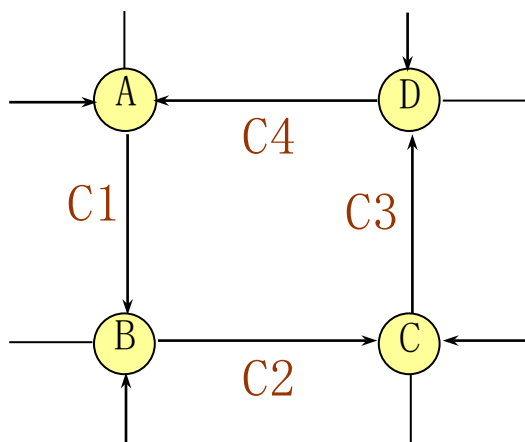
例如：图（a）：出现循环的通道相关而产生死锁

图（b）：利用虚拟通道方法避免死锁，可以增加两条虚拟通道V3和V4。

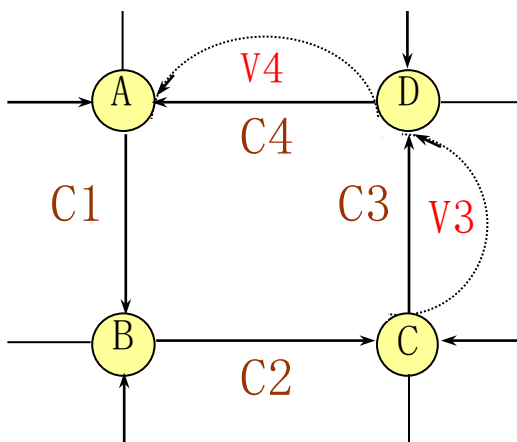
图（c）：避免了死锁

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。

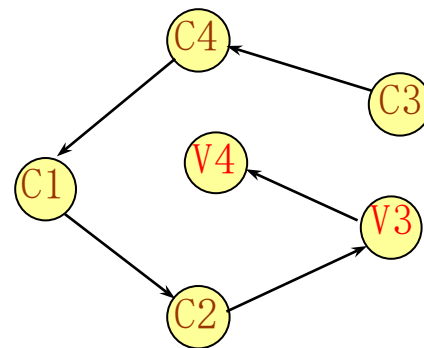
为此，当实现数目很大的虚拟通道时需要用高速的多路选择开关。



(a) 通道死锁



(b) 增加虚拟通道



(c) 利用虚拟通道后的通道相关图

9.5 消息传递机制

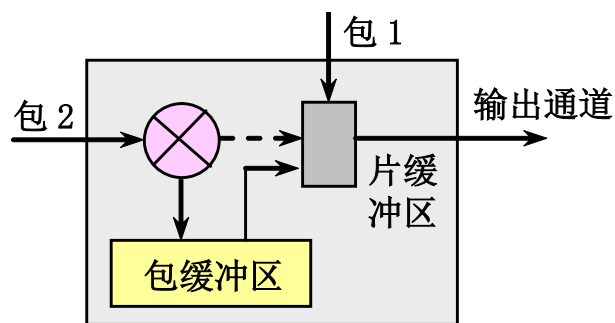
9.5.3 流控制策略

1. 包冲突的解决

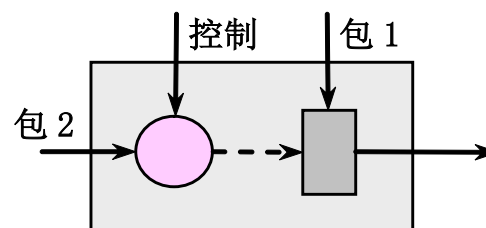
- 为了通过通道在两个相邻结点之间传送一个片，要同时具备3个条件：
 - 源缓冲区已存有该片；
 - 通道已分配好；
 - 接收缓冲区准备接收该片。
- 当两个包到达同一个结点时，它们可能都在请求同一个接收缓冲器或者同一个输出通道，这时必须对两个问题进行仲裁。
 - 把通道分配给哪个包？
 - 如何处理被通道拒绝的包？

9.5 消息传递机制

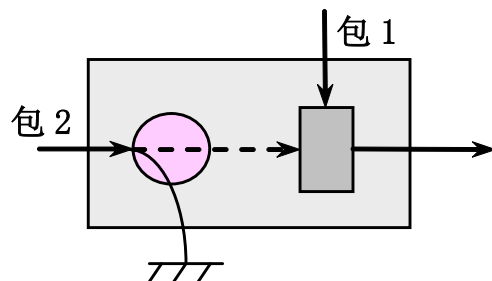
➤ 4种解决方案



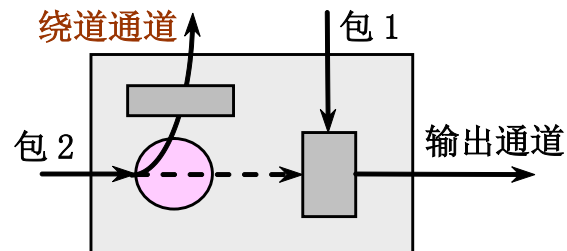
(a) 用缓冲实现虚拟通道



(b) 阻塞流控制



(c) 扬弃并重发



(d) 阻塞后绕道

9.5 消息传递机制

- 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。
- 阻塞第二个包
- 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。
- 绕道（**Detour**）
 - 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。

9.5 消息传递机制

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 消息传递机制

(1) 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) 。

9.5 消息传递机制

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)

(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)

是一条东-北路径；

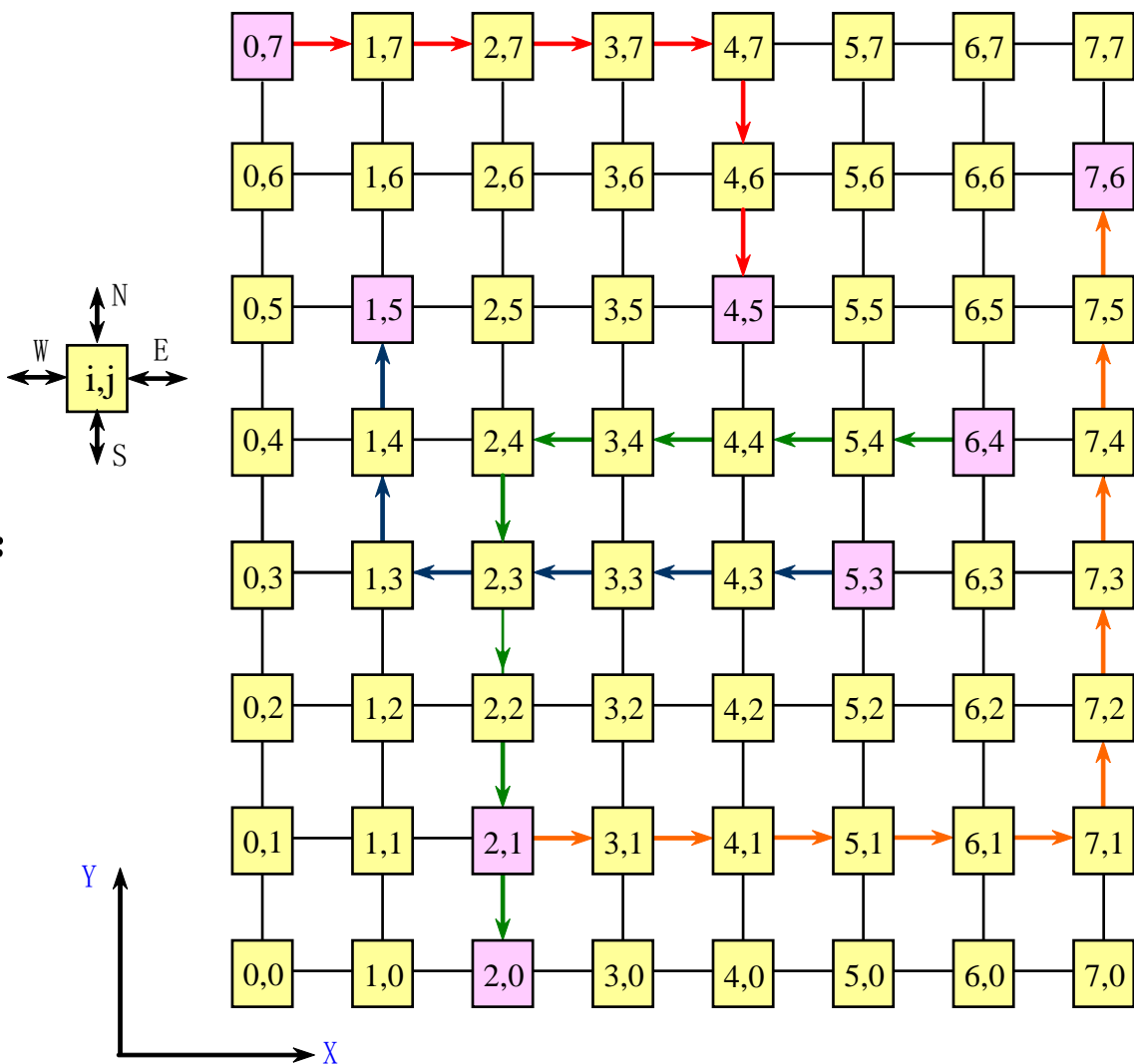
(0, 7) 到 (4, 5)

是一条东-南路径；

(6, 4) 到 (2, 0)

是一条西-南路径；

(5, 3) 到 (1, 5) 是一条西-北路径。



9.5 消息传递机制

(2) 超立方体E-cube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 $b = b_{n-1}b_{n-2}\dots b_1b_0$ 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。

9.5 消息传递机制

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：

- 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。

\oplus ：异或（XOR）

9.5 消息传递机制

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 $r = (r_4 r_3 r_2 r_1) = 0110 \oplus 1101 = 1011$

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 $v \oplus 2^{2-1} = 0111 \oplus 0010 = 0101$

$i=i+1=3$

9.5 消息传递机制

第4步: $r_3=0$, 所以跳过一步

$$i=i+1=4$$

第5步: $r_4=1$

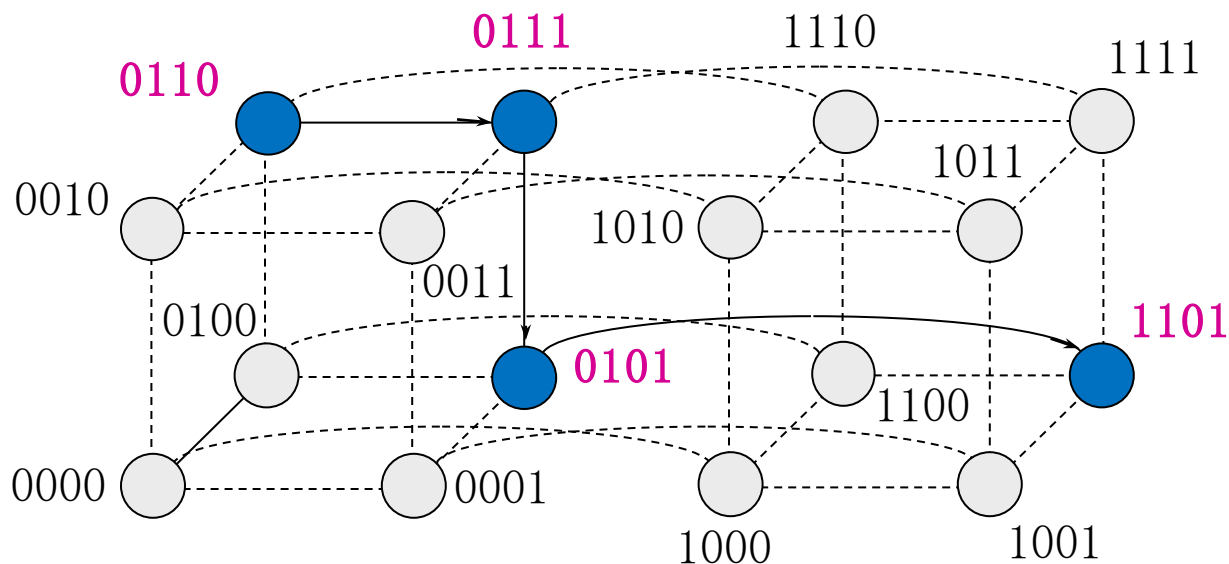
所以从 $v=0101$ 寻径到 $v \oplus 2^{4-1} = 0101 \oplus 1000 = 1101$, 结束。

因此路径为: **0110**→**0111**→**0101**→**1101**

源: $s=0110$

目的: $d=1101$

路径: $0110 \rightarrow 0111$
 $\rightarrow 0101 \rightarrow 1101$



9.5 消息传递机制

9.5.4 选播和广播寻径算法

1. 多计算机网络中会出现以下4种通信模式：

- **单播（Unicast）**：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（Multicast）**：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（Broadcast）**：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（Conference）**：对应于多到多的通信情况。

9.5 消息传递机制

- 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。

2. 以网格网络为例，讨论选播和广播。

在 3×4 网格上实现的选播寻径

源结点：S

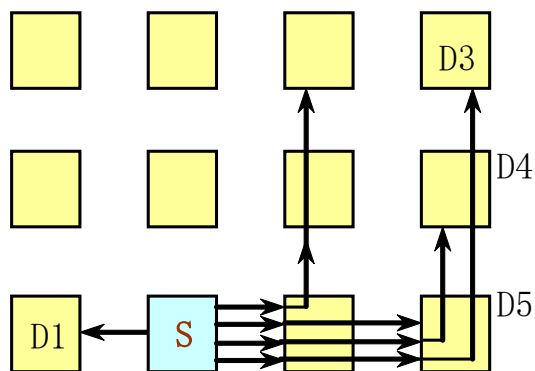
传送一个包到标号为 D_i 的5个目的结点

$(i=1, 2, \dots, 5)$

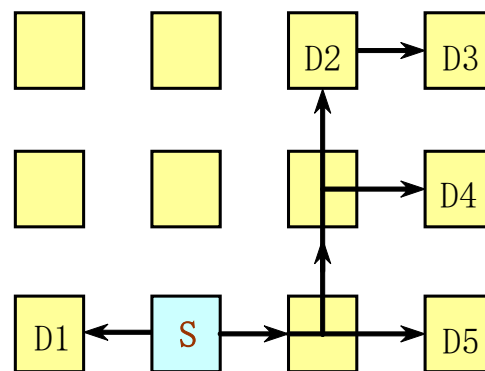
9.5 消息传递机制

- 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，则用图（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

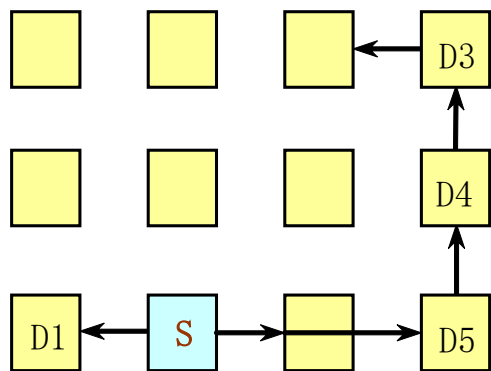
9.5 消息传递机制



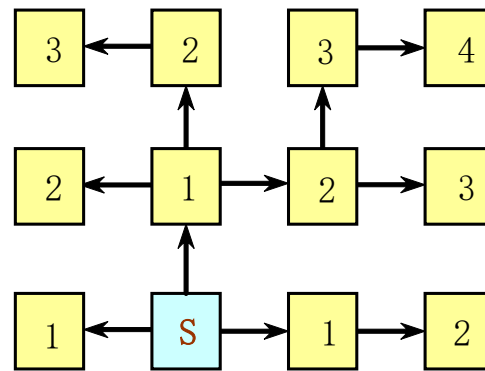
(a) 5 次单播，流量为 13，距离为 4



(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

本章小结

- 互连网络的定义、功能、描述工具、拓扑结构、性能参数、寻径方法、流控制策略等问题。
- 重点是：(1) 静态网络，动态网络；(2) 寻径方法；(3) 消息传递方式。
- 互连网络可反映网络输入数组和输出数组之间的对应排列关系。常用的互连函数有交换、均匀洗牌、蝶式函数、反位序函数、移数函数等。
- 互连网络分为静态互连网络和动态互连网络。
- 静态网络是指处理结点间有着固定连接的一类网络，程序执行期间，这种连接保持不变。
- 静态网络的特性参数有网络规模、结点度、距离、网络直径、等分长度、对称性等。

本章小结

- 动态网络是用交换开关构成的，可按执行程序的要求动态改变连接状态，通常分为**总线网络、多级互连网络、交叉开关网络**三类。
- 互连网络通信中常用的消息传递方式：
线路交换、存储转发、虚拟直通、虫蚀方式
- 为了避免产生死锁，可以利用虚拟通道，并使用有效的寻径算法。X-Y寻径法、E-立方路由算法是确定性算法。
- 多计算机网络中有**单播、选播、广播、会议**四种通信模式，分别对应一对一、一对多、一对全体、多对多四种通信情况

逆函数、互逆网络

小结

1. 恒等置换(可逆)

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

2. 交换置换(可逆)—Exchange

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

$$\text{立方体置换(可逆)} \quad \text{Cube}_0(x_2x_1x_0) = x_2x_1\bar{x}_0$$

$$\text{Cube}_1(x_2x_1x_0) = x_2\bar{x}_1x_0$$

3. 混洗置换 Shuffle

$$\text{Cube}_2(x_2x_1x_0) = \bar{x}_2x_1x_0$$

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

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(可逆)—Butterfly

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

5. 反位序置换(可逆)-Reversal

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

6. 移数置换

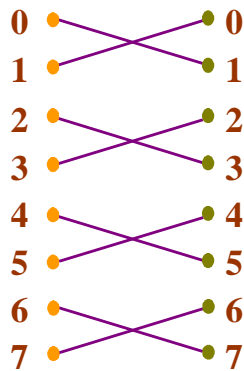
$$\alpha(x) = (x \pm k) \bmod N \quad 1 \leq x \leq N-1, \quad 1 \leq k \leq N-1$$

7. PM2I置换

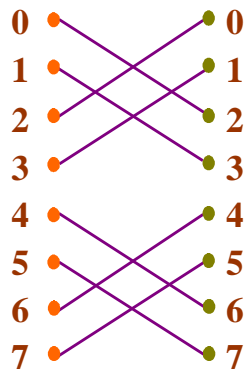
$$\text{PM2}_{+i}(x) = x + 2^i \bmod N$$

$$\text{PM2}_{-i}(x) = x - 2^i \bmod N$$

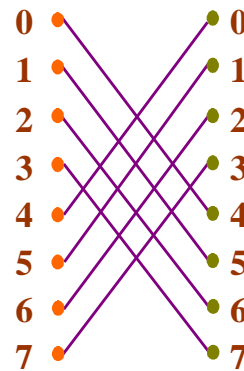
恒等函数、交换函数、立方体函数、蝶式函数



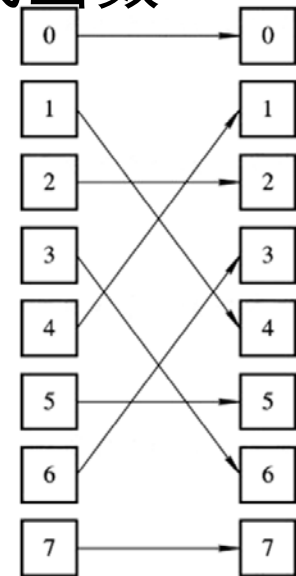
(a) $Cube_0$ 交换函数



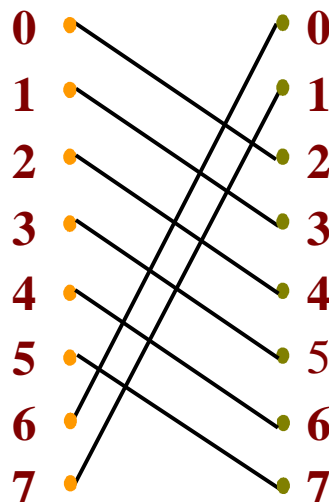
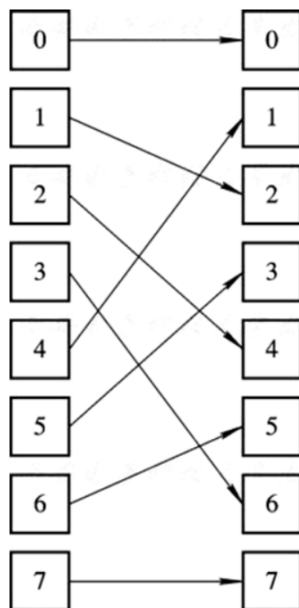
(b) $Cube_1$ 交换函数



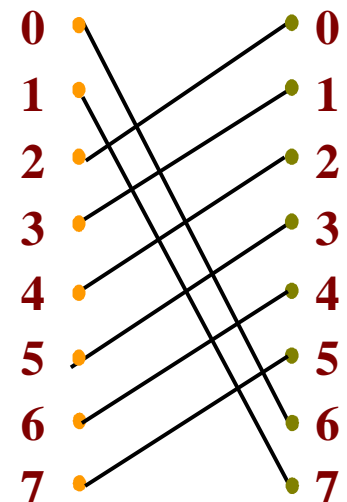
(c) $Cube_2$ 交换函数



均匀洗牌函数、反位序函数、移数函数、PM2I



(a) 左移移数函数 $k=2$



(b) 右移移数函数 $k=2$

恒等函数、交换函数、立方体函数、蝶式函数

双向:

$$f(f(x)) = x \qquad f^{-1}(x) = f(x)$$

逆函数就是自己

均匀洗牌函数、反位序函数、移数函数、PM2I函数

单向:

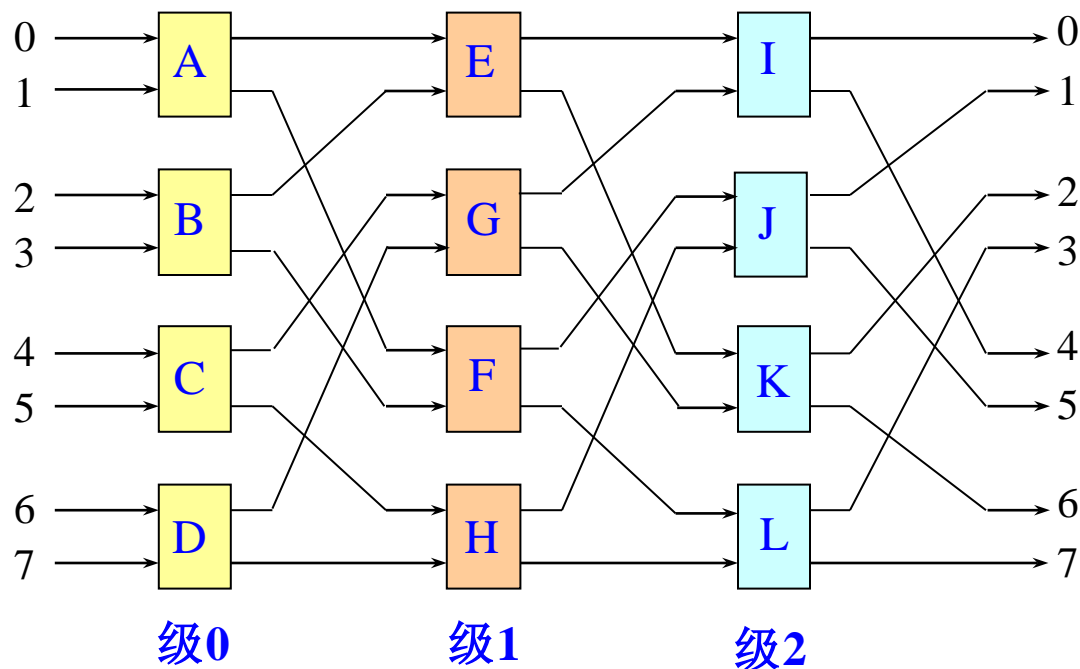
$$f(f(x)) \neq x$$

逆函数:
(反函数)

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

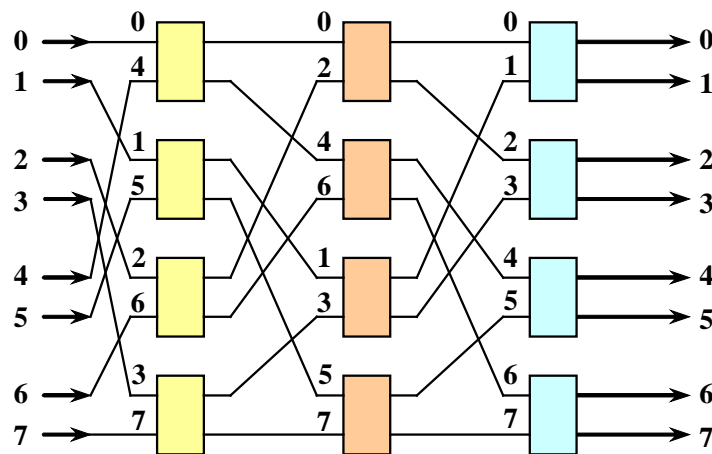
互逆网络：如果把出端当入端，入端当出端，则原网络变为另一个互连网络，称此网络为原网络的逆互连网络。

➤ N=8的多级立方体互连网络的另一种画法 ($f(x)$)



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

Omega网络 ($g(x)$)



如果Omega网络也采用2功能交换开关，那么Omega网络和间接二进制n方体网络就互为逆网络。