第 3 章　互联网络技术

互联网络是高度并行计算机系统中的关键部件。本章介绍互联网络的基本概念、描述工具、互连函数和性能参数等，讨论静态互联网络的结构与特性，分析动态互联网络的互连形式和几种多级互联网络结构特点、控制与寻径方式，阐述交叉开关的设计技术和互联网络消息传递控制策略与机制。

3.1　互联网络的基本概念

3.1.1　互联网络及其组成与特征

3.1.1.1　互联网络及其组成

互联网络是一种由开关元件按照一定的拓扑结构和控制方式构成的网络，用来实现计算机系统内部多个处理机或多个功能部件之间的相互连接及信息交换。所有的互联网络都是由链路、网络接口电路和交叉开关 3 部分组成（共享介质的互联网络不使用交叉开关），其中交叉开关是核心。

1. 链路

链路（Link）也称为通道或电缆，是用来将计算机系统中两个硬件进行物理连接。一条链路可连接两个交叉开关或一个交叉开关和一台处理机或一个功能部件的网络接口。目前链路的介质一般是铜线或光纤，铜线链路较便宜，但长度有限；光纤价格贵，但长度可以很长，带宽也很高。链路的主要逻辑特性包括长度、宽度和驱动时钟。因此，链路除可从使用的介质分类外，还可从逻辑特性上划分。

从长度来看，有短链路和长链路之分；一条短链路在任何时刻仅包含一个逻辑信号，而一长短链路在任何时刻允许传输一串逻辑信号，如同一条传输线。从宽度来看，有串行链路和并行链路之分；串行链路（窄链路）只有一位信号线，各种信息以多路分时复用的方式共享单信号线；并行链路（宽链路）有多位信号线，各种信息可并行传输。从驱动时钟来看，有同步时钟链路和异步时钟链路之分；同步时钟链路是指链路两端的结点使用相同的时钟；异步时钟链路是指通过嵌入时钟编码，链路两端的结点可使用不同的时钟。

2. 网络接口电路

网络接口电路（Network Interface Circuitry，NIC）也称为网卡，是用来将计算机系统中结点（一台处理机或一个功能部件）连接到网络上。网络接口必须能够处理结点与网络之间的双向传输，其功能主要包括消息包格式化、路由通路选择、一致性检查、流量与错误控制等。因此，网络接口的成本由端口规模、存储容量、处理与控制能力等决定。

网络接口的体系结构取决于网络和结点，在同一网络中不同的结点，也可能需要不同的网络接口。典型的网络接口电路包括嵌入式处理机、输入输出缓冲器、控制存储器和控制逻辑电路，其复杂性一般高于交叉开关。

3. 交叉开关

交叉开关（Switch）也称为路由器，是用来建立结点对之间连接的开关阵列。交叉开关包

括输入输出端口、结点开关阵列及其控制逻辑，如图 3-1 所示为一个 4 输入 4 输出的交叉开关。结点开关阵列可在程序控制下接通和断开，以同时建立 n 个输入和 n 个输出间的连接（n 为交叉开关输入端口或输出端口数，可称为度）。每个输入端口内有接收器、输入缓冲器，用于处理到达的消息包；每个输出端口内有发送器和输出缓冲器，用于把数据信号传送到链路上。

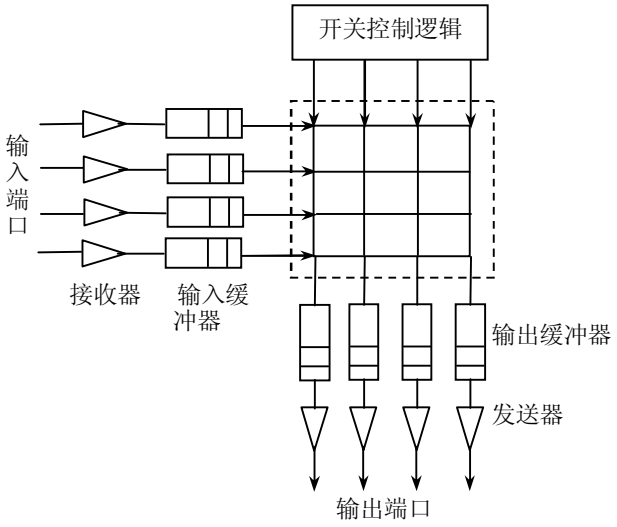


图 3-1　交叉开关的基本结构

3.1.1.2　互联网络功能及其划分

随着各个领域对高性能计算机的要求越来越高，多处理机系统和多计算机系统的规模越来越大，处理机之间或处理单元与存储模块之间的通信要求和难度越来越突出。所以互联网络已成为并行处理系统的核心组成部分，它对并行处理系统的性能起着决定性的作用。互联网络在多处理机系统中的位置和功能作用如图 3-2 所示。例如，多处理机系统中每台处理机 P_i 与自己的本地存储器 LM 和私有高速缓存储器 C_i 可直接相连，但应通过多处理机—存储器互联网络（IPMN）与共享存储器模块 SM 相连，通过多处理机—I/O 网络（PION）访问共享的 I/O 和外围设备，多处理机之间通过处理机间通信网络（IPCN）进行通信。

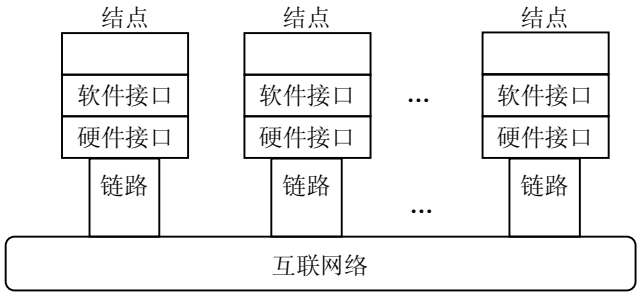


图 3-2　互联网络在系统中的功能作用

根据互联网络连接的结点距离可划分出系统域网络 SAN（3～25m）、局域网络 LAN（500～2000 m）、城域网络 MAN（≥25km）和广域网络 WAN（全球），它们间的作用关系如图 3-3 所示。系统域网络带宽要求为 100Mb/s～100Gb/s，主要有总线、交叉开关与多级交叉开关等网络和 SCI、光纤通道、HiPPI 与 Myrinet 等技术；局域网络带宽要求为 10Mb/s～10Gb/s，主要有以太网、HiPPI、光纤通道和 FDDI 等网络技术；城域网络带宽要求为 20Mb/s～5Gb/s，主

要有光纤通道、FDDI 和 ATM 等网络技术；广域网络带宽要求为 20Mb/s～0.55Gb/s，主要有光纤通道和 ATM 等网络技术。

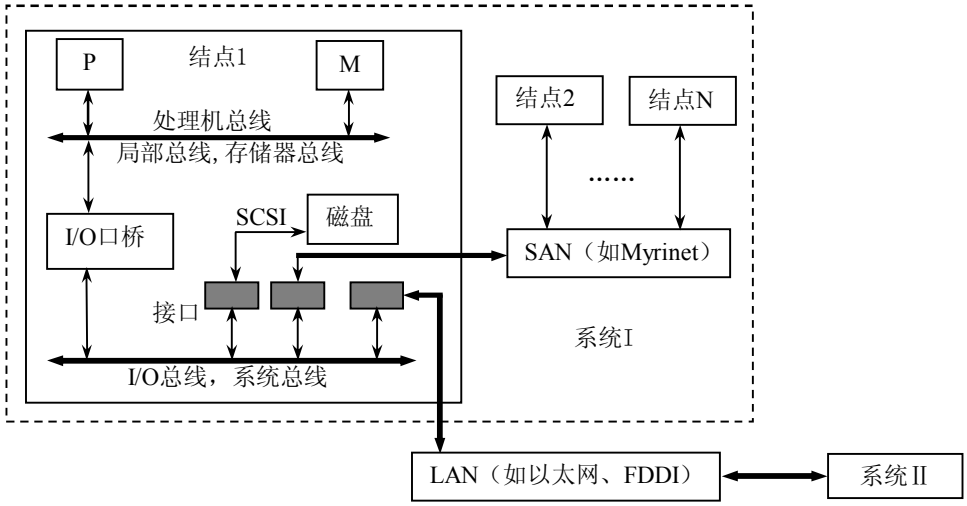


图 3-3　各种网络之间的作用关系

3.1.1.3　互联网络的基本特征

互联网络的基本特征主要包括定时方式、交换方法、控制策略和拓扑结构等几个方面。

1. 定时方式

互联网络的定时方式有同步和异步两种。同步系统使用一个集中的统一时钟，它可以把数据同时播送到各结点中，或者使各结点同时与相邻结点进行通信。异步系统没有时钟，各结点根据各自的需要进行通信。阵列处理机为典型的同步系统，多机系统一般都为异步系统。

2. 交换方法

互联网络的交换方法有线路交换（Circuit Switching）和分组交换（Packet Switching）两种。在线路交换中，源结点和目的结点的物理通路在整个数据传递期间一直保持连接。在分组交换中，把要传递的数据分成许多包，这些包分别送入互联网络，各个包可以通过不同的路径传送到目的结点，并不存在一个固定的实际连接的物理通路。

3. 控制策略

互联网络的控制策略有集中式和分散式两种。集中控制有一个全局的控制器接收所有的通信请求，并设置互联网络中相应开关的实际连接的物理通路。而分散控制对通信请求处理和设置互联网络中相应开关实际连接的物理通路是由分布在各个功能部件中的控制逻辑分散地进行通信实现的。

4. 拓扑结构

互联网络的拓扑结构有静态拓扑结构和动态拓扑结构两种。在静态拓扑结构中，各结点间的物理通路是专用链路，且固定连接，不能重新组合。在动态拓扑结构中，各结点间的物理通路可以通过设置互联网络的开关重新组合，链路连接不固定。互联网络的链路是连接网络中相邻结点所用的通信线路，是网络的相邻结点间进行数据信息传送时所使用的通路。

3.1.2　互联网络的描述工具

为了反映不同互联网络的连接特性，在输入结点和输出结点间建立相应的对应关系，通

常采用 3 种方法来描述。

3.1.2.1 图形表示法

图形表示法是把互联网络中输入输出的对应关系用连线图来表示。该方法虽然直观，但比较烦琐，且难以体现内在规律，因此，一般结合另外两种表示法一起使用。

3.1.2.2 对应表示法

对应表示法是把互联网络中输入输出的对应关系表示为$\begin{bmatrix} 0 & 1 & \cdots & N-1 \\ f(0) & f(1) & \cdots & f(N-1) \end{bmatrix}$，即 0 变换 f(0)，1 变换为 f(1)，…，N−1 变换为 f(N−1)。例如，$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 2 & 4 & 6 & 1 & 3 & 5 & 7 \end{pmatrix}$则表示输入结点 0、1、…、7 分别对应连接输出结点 0、2、…、7。

3.1.2.3 函数表示法

函数表示法是把互联网络中输入输出的变换关系通过数学表达式表示，若用 x 表示输入端变量，则用函数 $f(x)$ 表示输出端变量，函数 $f(x)$ 称为互连函数。由于一个结点在一般情况下既可作输入端，也可作输出端，所以通常认为输入端数与输出端数是相等的。如果互联网络将 N 个结点连接，则该互联网络有 N 个输入结点和 N 个输出结点，即有 N 个输入变量和 N 个输出变量。输入端与输出端的变量通常用对应的二进制数的结点编号来表示，则互连函数与对应表示法一样，表示了输入端与输出端之间的一一对应关系。若 x 为 n 位二进制数，$n=\log N$，则互连函数一般写成 $f(x_{n-1},x_{n-2},\cdots,x_1,x_0)$。

当互联网络用来实现处理器与处理器之间的数据交换时，互连函数也反映了网络输入数组与输出数组间对应的置换关系或排列关系，所以互连函数有时也称为置换函数或排列函数。

有一种特殊的互连函数 $f(x)$ 称为循环互连函数，它表示的对应关系为：$f(x_0)=x_1$，$f(x_1)=x_2$，…，$f(x_j)=x_0$，则可以把循环互连函数表示为 $(x_0,x_1,\cdots,x_j)$，即互联网络的入端号 x_0 连至出端号 x_1，入端号 x_1 连至出端号 x_2，…，入端号 x_j 连至出端号 x_0，$j+1$ 称为循环长度。

3.1.3 常用的基本互连函数

3.1.3.1 恒等置换

相同编号的输入端与输出端一一对应互连所实现的置换称为恒等置换。其表达式为：

$$I(x_{n-1},x_{n-2},\cdots,x_1,x_0)=x_{n-1},x_{n-2},\cdots,x_1,x_0$$

等式左边括号内的 $x_{n-1},x_{n-2},\cdots,x_1,x_0$ 和等式右边的 $x_{n-1},x_{n-2},\cdots,x_1,x_0$ 均为网络输入端和输出端的二进制地址编号。这种恒等置换实现的输入端与输出端的连接如图 3-4 所示，图中左部为输入端，右部为输出端。

5.1.3.2 交换置换

交换置换是实现二进制地址编号中第 0 位值不同的输入端和输出端之间的连接。其表达式为：

$$E(x_{n-1},x_{n-2},\cdots,x_1,x_0)=x_{n-1},x_{n-2},\cdots,x_1,\overline{x}_0$$

它所实现的输入端与输出端的互连图形如图 3-5 所示。

3.1.3.3 方体置换（Cube）

方体置换是实现二进制地址编号中第 k 位值不同的输入端和输出端之间的连接。其表达式为：

$$C(x_{n-1},x_{n-2},\cdots,x_{k+1},x_k,x_{k-1},\cdots,x_1,x_0)=x_{n-1},x_{n-2},\cdots,x_{k+1},\overline{x}_k,x_{k-1},\cdots,x_1,x_0$$

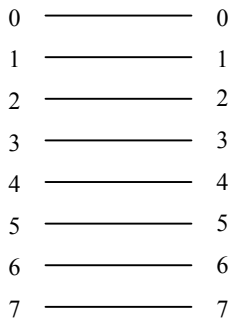


图 3-4　N=8 时恒等置换

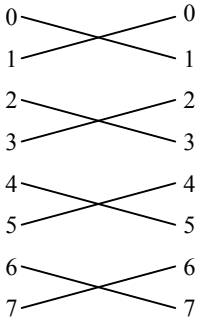
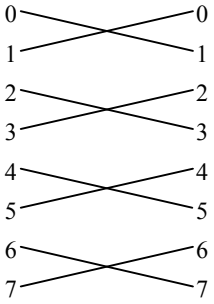
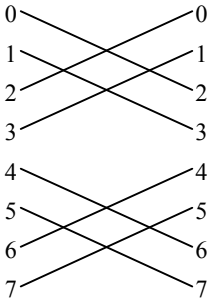


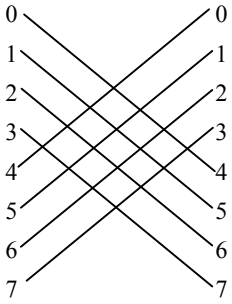
图 3-5　N=8 时交换置换



（a）C_0 方体置换



（b）C_1 方体置换



（c）C_2 方体置换

图 3-6　N=8 时方体置换

显然，对于 N 个输入输出结点的互联网络，可以有 $n=\log N$ 个方体置换 C_0、C_1、…、C_{n-1}。以 N=8 为例，则有 $n=\log_2 8=3$ 个方体置换，分别为：

$$C_0(x_2,x_1,x_0)=x_2,x_1,\overline{x}_0 \qquad C_1(x_2,x_1,x_0)=x_2,\overline{x}_1,x_0 \qquad C_2(x_2,x_1,x_0)=\overline{x}_2,x_1,x_0$$

其互连图形如图 3-6 所示，其中 C_0 即为交换置换。

3.1.3.4　均匀洗牌置换（Shuffle）

均匀洗牌置换是将输入端分成数目相等的两半，前一半和后一半按序一个隔一个地从头至尾依次与输出端相连。由于类似洗扑克牌，将整副扑克牌分成相等的两叠，理想时一张隔一张均匀搭配。其实质是将输入端二进制地址循环左移一位即得到对应输出端的二进制地址。其函数关系可表示为：

$$\sigma(x_{n-1},x_{n-2},\cdots,x_1,x_0)=x_{n-2},x_{n-3},\cdots,x_0,x_{n-1}$$

对于 N=8 个输入输出结点的互联网络均匀洗牌的互连图形如图 3-7（a）所示。

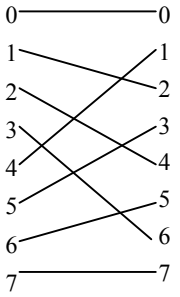
此外，还可分别定义子洗牌（Subshuffle）$\sigma_{(k)}$ 和超洗牌（Supershuffle）$\sigma^{(k)}$ 如下：

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

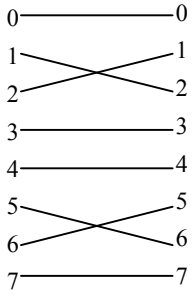
$$\sigma^{(k)}(x_{n-1},x_{n-2},\cdots,x_{n-k},x_{n-k-1},x_{n-k-2},\cdots,x_1,x_0)=x_{n-2},x_{n-3},\cdots,x_{n-k},x_{n-k-1},x_{n-1},x_{n-k-2},\cdots,x_0,x_k$$

显然下列等式成立：$\sigma^{(n-1)}(x)=\sigma_{(n-1)}(x)=\sigma(x)$　　$\sigma^{(0)}(x)=\sigma_{(0)}(x)=x$

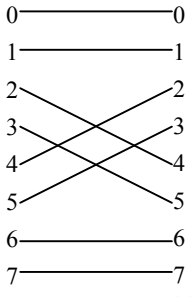
对于 N=8 个输入输出结点的互联网络子洗牌置换 $\sigma_{(2)}$ 和超洗牌置换 $\sigma^{(2)}$ 的互连图形如图 3-7（b）、（c）所示。从图中可以看出，子洗牌是将整组数据分成若干个小组，对每个子组完成均匀洗牌变换，超洗牌仍对整组数据进行均匀洗牌变换，但增加了数据变换宽度。



（a）均匀洗牌置换σ



（b）子洗牌置换$\sigma_{(2)}$



（c）超洗牌置换$\sigma^{(2)}$

图 3-7　N=8 时的均匀洗牌置换

逆均匀洗牌是均匀洗牌的逆函数，其函数表达式为：

$$\sigma^{-1}(x_{n-1},x_{n-2},\cdots,x_1,x_0)=x_0,x_{n-1},\cdots,x_2,x_1$$

它是将输入端二进制地址编号循环右移一位即得到相应的输出端地址。对于 N=8 个输入输出结点的互联网络的互连图形如图 3-8 所示。

以均匀洗牌和逆均匀洗牌代表的链路与以交换代表的开关多级组合起来可构成Ω和逆Ω网络。σ 函数在实现多项式求值、矩阵转置和 FFT 等并行排序方面得到广泛应用。

另外还有 q 洗牌函数，q 洗牌函数的表达式为：

$$S_{qr}(i)=(qi+i/r)\bmod qr$$

其中，q 和 r 是正整数，q×r=N，0≤i≤qr-1。其理意义是：将 q×r 张牌分成 q 组，每组 r 张，洗牌时将第一组的第一张牌放在第一个位置，再取第二组的第一张牌放在第二个位置，……直至取 q 组的第一张牌放在第 q 个位置上之后，再取第一组的第二张牌放在第 q+1 个位置上，这个过程一直进行到把各组的牌全部取完为止。对于 N=8 个输入输出结点的互联网络，若 q=2，r=4，则互连图形如图 3-9 所示。

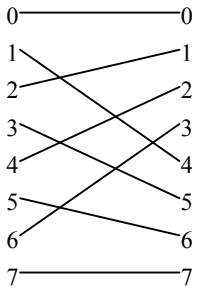


图 3-8　N=8 时的逆均匀洗牌置换

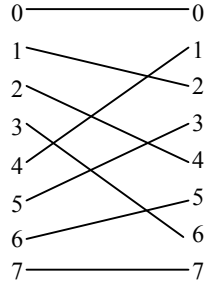


图 3-9　N=8、q=2 时的 q 洗牌置换

3.1.3.5　蝶式置换（Butterfly）

蝶式置换是将输入端二进制地址的最高位和最低位互换位置即可得相应输出端的地址。其函数表达如下：

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

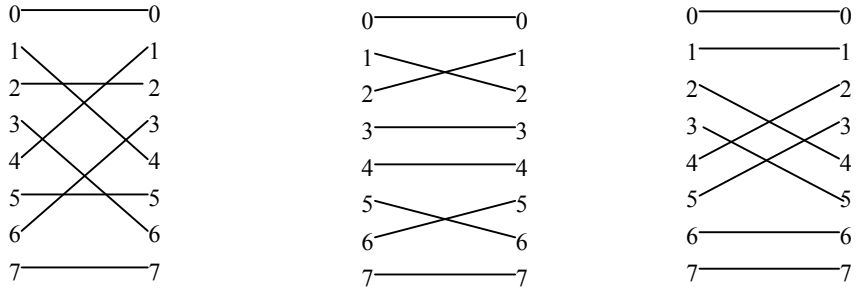
同样，可定义子蝶式$\beta_{(k)}$和超蝶式$\beta^{(k)}$如下：

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

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

显然下列等式成立：$\beta^{(n-1)}(x)=\beta_{(n-1)}(x)=\beta(x)$　　$\beta^{(0)}(x)=\beta_{(0)}(x)=x$

对于 N=8 个输入输出结点的互联网络β、$\beta_{(2)}$和$\beta^{(2)}$的互连图形如图 3-10 所示。蝶式与子蝶式是构成方体多级网络的基础。



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

图 3-10　N=8 时的蝶式置换和位序颠倒置换

3.1.3.6　位序颠倒置换

位序颠倒置换是将输入端二进制地址的位序颠倒过来求得相应输出端的地址。其表达式为：

$$\rho(x_{n-1},x_{n-2},\cdots,x_1,x_0)=x_0,x_1,\cdots,x_{n-2},x_{n-1}$$

同样，也可以定义子位序颠倒置换和超位序颠倒置换：

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

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

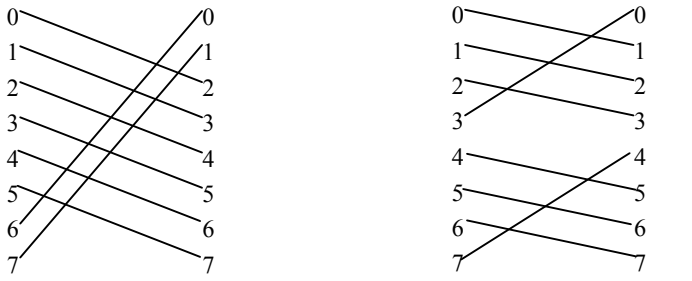
对于 N=8 个输入输出结点的互联网络ρ、$\rho_{(2)}$和$\rho^{(2)}$的互连图形如图 3-10 所示。这时正好ρ=β，$\rho_{(2)}=\beta_{(2)}$，$\rho^{(2)}=\beta^{(2)}$。但要注意，不要因为这些特殊情况下的ρ和β的关系而错认为β和ρ是一样的。

3.1.3.7　移数置换

移数置换是将输入端编号循环移动一定的位置得出输出端编号。其表达式如下：

$$\alpha(x)=(x+k)\bmod N，\quad 0\leqslant x\leqslant N-1$$

k 为常数，指移动的位置值。对于 N=8 个输入输出结点且 k=2 的互联网络α的互连图形如图 3-11（a）所示。



（a）移数置换 k=2　（b）组内移数置换 k=1、r=2

图 3-11　N =8 时移数置换

也可以将整个输入端编号分成若干个组，在组内进行循环移数置换，这种组内循环移数的表达式如下：

$$\alpha(x)_{(N-1):2^r} = ((x)_{(N-1):2^r} + k) \bmod 2^r + 2^r \qquad \alpha(x)_{(2^r-1):0} = ((x)_{(2^r-1):0} + k) \bmod 2^r$$

其中下标(N-1):2^r和(2^r-1):0 分别指从 N-1 结点到 2^r结点和从 2^r-1 结点到 0 结点，r=logM，M 为组内结点数。对于 N=8 个输入输出结点、k=2、r=2 的互联网络α的互连图形如图 3-11（b）所示。

移数置换可以用循环互连函数表示，图 3-11（a）和图 3-11（b）所示的循环互连函数为：

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

3.1.3.8　加减 2I 置换

加减 2I 置换使输入端编号 x 同输出端编号 $x \pm 2^i$ 相连。其表达式为：

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

其中 0≤x≤N-1，0≤i≤(n-1)，$n=\log_2 N$。对于 N=8 个输入输出结点的互联网络 PM2I 的互连图形如图 3-12 所示。从图可知，它实际上也是一种移数置换。

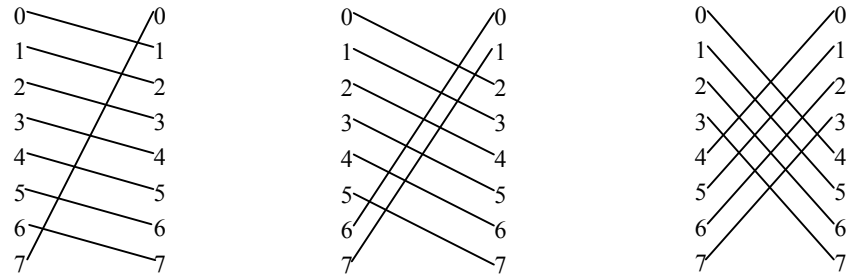


图 3-12　N=8 时的 PM2I 置换

加减 2I 置换也可以用循环互连函数表示。对于 N=8 的 PM2I 置换，共有 2n=6 个互连函数，分别用循环互连函数表示为：

$PM2_{+0}$=(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)

3.1.4　互联网络结构特性和传输性能参数

3.1.4.1　互联网络的结构特性参数

互联网络的拓扑结构可用有向边或无向边连接有限个结点的图来表示。利用图的有关参数能定义出互联网络拓扑结构的若干特性参数。互联网络的结构特性参数可分为物理结构和逻辑特性两个方面。

1. 物理结构参数

（1）网络规模。互联网络中的结点数称为网络规模，它表示该网络所能连接的部件个数。

（2）结点度。互联网络中某一结点相连接的边（即链路或通道）数称为该结点的结点度，用 d 表示。进入结点的边数称为入度，从结点出来的边数称为出度，结点度为入度与出度之和。

（3）结点距离。互联网络中两个结点之间相连的最少边数称为这两个结点的结点距离。

（4）网络直径。互联网络中任意两个结点之间距离的最大值称为网络直径。从通信的观点来看，网络直径应当尽可能小。

（5）结点线长。互联网络中两个结点之间连接线的长度称为这两个结点的结点线长，它会影响信号的时延等性能特性。

2. 逻辑特性参数

（1）网络等分宽度。当某一互联网络被切分成相等的两半时，沿切口的最小边（通道）数称为通道等分宽度，又称对剖宽度，用 b 表示。相应的切口称为对剖平面（一组连线）。而线等分宽度 B=b×ω，ω为通道宽度（用位表示）。

（2）网络对称性。从任何结点看，互联网络的拓扑结构都是相同的，则称该互联网络具有对称性，该互联网络称为对称网络。对称网络容易实现，对编程的支持良好。

（3）网络可扩展性。网络可扩展性是指在互联网络拓扑性能保持不变的情况下，可扩充结点的能力。

3.1.4.2　互联网络的传输性能参数

两台计算机连接的最简单的网络如图 3-13 所示，它们都有一个先进先出的数据队列，一台计算机向另一台计算机发送消息。

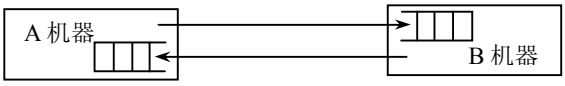


图 3-13　连接两台机器的简单网络

发送方的步骤为：首先应用程序把要发送的数据拷贝送到操作系统缓冲区，然后操作系统根据要发送的数据计算出检查和。把它放在消息中，并启动超时计数器；最后操作系统把缓冲区中的数据送到网络接口硬件，并通知硬件开始发送消息。

接收方的步骤为：首先把数据从网络接口硬件拷贝到操作系统缓冲区；然后根据接收到的数据计算出检查和，若与发送过来的检查和匹配，则发送一个应答信号给发送方（发送方接收到应答信号就会释放缓冲区），否则删除该消息，发送方在超时计数器超时后重发该消息。最后若检查和匹配，则把接收到的数据拷贝到用户地址空间并启动应用程序继续执行。

因此，互联网络的传输性能参数可分为时延和带宽两个方面，其中时延性能参数如图 3-14 所示。

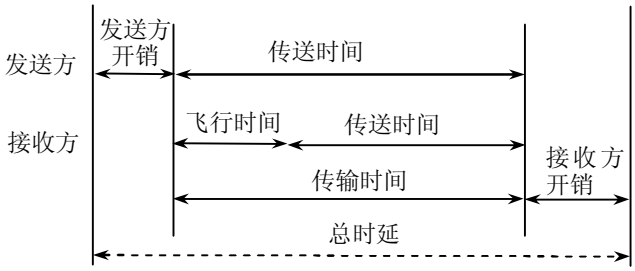


图 3-14　互联网络时延传输性能参数

1. 时延性能参数

由图 3-14 可知一个消息在互联网络上传输的总时延为：

总时延=发送方开销+飞行时间+传送时间+接收方开销。

（1）发送方开销（Sender Overhead）。处理器把消息放到互联网络的时间称为发送方开销，包括软件和硬件所花费的时间。

（2）接收方开销（Receiver Overhead）。处理器把到达的消息从互联网络取出来的时间称为接收方开销，也包括软件和硬件所花费的时间。

（3）飞行时间（Time of Flight）。飞行时间是指发送方开始发送消息至第一位信息到达接

收方所花费的时间，它包括由于网络中转发或其他硬件所花费的时间。

（4）传送时间（Transmission Time）。传送时间是指消息通过网络的时间，它等于消息长度除以网络频宽。

（5）传输时间（Transport Latency）。传输时间是指消息在互联网络上传输所花费的时间，它等于飞行时间和传送时间之和。

2. 带宽性能参数

（1）端口带宽。互联网络中任一端口到另一端口传输信息的最大速率称为端口带宽，单位为 MB/s。对称网络的端口带宽与端口位置无关，非对称网络的端口带宽是所有端口带宽的最小值。

（2）聚集带宽。互联网络中从一半结点到另外一半结点传输信息的最大速率称为聚集带宽，单位为 MB/s。聚集带宽=端口带宽×结点数/2。例如，每个端口带宽为 10MB/s，那么 512 个结点的聚集带宽为(10×512)/2≈2.5GB/s。

（3）对剖带宽。互联网络中对剖平面上传输信息的最大速率称为对剖带宽，单位为 MB/s。

（4）网络频宽（Bandwidth）。网络频宽泛指消息进入网络频宽网络后，互联网络传输信息的最大速率，单位为 MB/s。

3.1.5 互联网络的分类

根据互联网络的拓扑结构和特性，可将互联网络分成 5 大类，如图 3-15 所示。

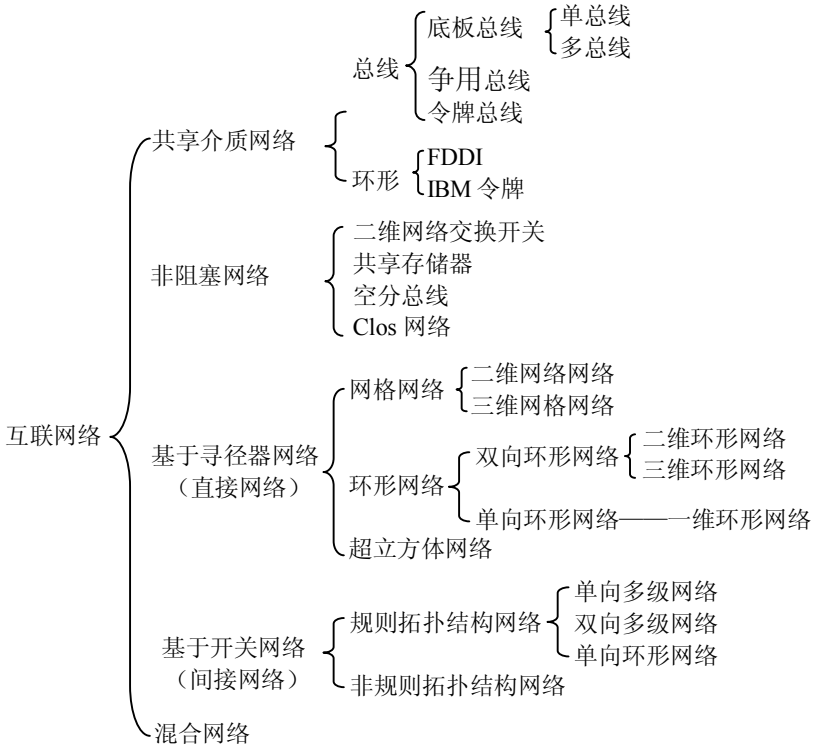


图 3-15 互联网络的分类

3.1.5.1 共享介质网络

共享介质网络是指在同一时间都只允许一个结点进行发送或接收，它又分总线结构和环

形结构。其中总线结构包括底板总线、争用总线和令牌总线，底板总线又包括单总线和多总钱等。而环形结构主要有 FDDI 和 IBM 令牌环。

3.1.5.2　非阻塞网络

非阻塞网络是指任何输入输出结点对之间总可以建立连接通路，消息通信不会阻塞。设计非阻塞网络的方法有多种，如采用交叉开关、空分总线和共享存储器技术等。非阻塞网络包括二维网络交换开关、共享存储器网络、空分总线和 Clos 网络。

3.1.5.3　直接网络

直接网络是指结点间直接连接，消息在传递途中经过的路径由开关元件事先固定接通，因此也称为静态网络，或称为基于寻径器网络。一般地，相邻的两个结点通过一对相反方向的单向通道连接或通过一个双向通道连接，用一个双向通道连接时，必须有一个仲裁协议来决定使用通道的是哪一侧。直接网络可分为 3 类，即网格网络、环形网络和超立方体的网络。其中网格网络分为二维网格网络和三维网格网络，环形网络分为双向环形网和单向环形网。双向环形网又分为二维的和三维的环形网，单向环形网是一维的环形网。

3.1.5.4　间接网络

间接网络是指结点不是通过直接相连的通道进行消息通信，而是通过网络的可控制开关机构进行连接的。由于每一个结点有一个网络适配器连接到网络的开关上，因此也称为基于开关的网络。开关的互连方式决定了网络的拓扑结构，大多数间接网络采用由多级开关组成的多级开关互联网络。间接网络按拓扑结构可分为规则拓扑结构和不规则拓扑结构，其中规则拓扑结构又分为单向多级网络、双向多级网络和单向环网络。

3.1.5.5　混合网络

混合网络是指一个互联网络中混合了两种以上的网络。

特别地根据网络各结点间的通路在运行中是否可改变的原则，可将互联网络分为静态互联网络和动态互联网络两大类。

【例 3.1】 由 16 个处理单元组成的 Illiac IV 阵列处理机采用的互联网络如图 3-16 所示，该互联网络采用的是哪一种互连函数。

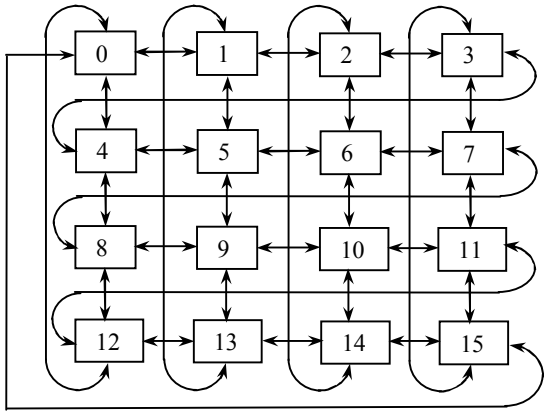


图 3-16　Illiac Ⅳ阵列处理机采用的互联网络

解　横向处理单元的连接有两种，一是 0→1→2→…→14→15→0，即是(0　1　2　…　14　15)；另一是 15→14→…→2→1→0→15，即是(15　14　…　2　1　0)。互连函数分别为 $PM2_{+0}$ 和 $PM2_{-0}$。

纵向处理单元的连接有两种，一是 0→4→8→12→0、1→5→9→13→1、2→6→10→14→2、3→7→11→15→3，即是(0 4 8 12)、(1 5 9 13)、(2 6 10 14)、(3 7 11 15)；另一是 12→8→4→0→12、13→9→5→1→13、14→10→6→2→14、15→11→7→3→15，即是(12 8 4 0)、(13 9 5 1)、(14 10 6 2)、(15 11 7 3)。互连函数分别为 $PM2_{+2}$ 和 $PM2_{-2}$。

【例 3.2】 设 16 个处理器编号分别为 0、1、…、15，用单级互联网络连接，当互连函数分别为（1）$Cube_3$ 、（2）PM_{+3} 、（3）Shuffle(Shuffle)时，第 13 号处理器分别与哪一个处理器相连？

解 （1）因为 $Cube_3(X_3X_2X_1X_0)=\overline{X}_3X_2X_1X_0$　　所以 13 → $Cube_3(1101)=0101$ → 5

（2）因为 $PM_{+3} = X + 2^3 \text{ MOD } N$　　所以 13 →$PM_{+3}(13)= 5$

（3）因为 $Shuffle(Shuffle(X_3X_2X_1X_0))= Shuffle(X_2X_1X_0X_3)= X_1X_0X_3X_2$

所以 13 → $Shuffle(Shuffle(1101))= Shuffle(1011)= 0111$ → 7

【例 3.3】 假设一个互联网络的频宽为 10Mb/s，发送方和接收方开销分别等于 230μs 和 270μs。如果两台机器相距 100m，现在要一台机器发送一个 1000 字节的消息给另外一台机器，试计算总时延。如果两台机器相距 1000km，总时延是多少。

解 光的速度为 299792.5km/s，信号在导体中传输的速度大约是光速度的 50%，从而可计算出飞行时间。

相距 100m 总时延 T=发送方开销+飞行时间+传送时间+接收方开销

= 230μs+0.1km/(0.5×299792.5km/s)+(1000×8)/10Mb/s×270μs

= 1301μs

相距 1000km 总时延 T=发送方开销+飞行时间+传送时间+接收方开销

= 230μs+1000 km/(0.5×299792.5km/s)+(1000×8)/10Mb/s×270μs

= 7971μs

3.2 静态互联网络

3.2.1 静态互联网络及类型

3.2.1.1 什么是静态互联网络

静态互联网络是指在各结点间有专用的连接通路，且在运行中不能改变的网络。网络中的每一个开关元件固定地建立结点之间的连接，直接实现结点之间的通信。这种网络一旦构成就是固定不变的，比较适合构成通信模式可预测的并行处理系统和分布计算机系统。

3.2.1.2 静态互联网络的种类

静态互联网络可以用维数来分类，所谓 n 维是指将它们画在 n 维空间上各条链路不会相交。一维的静态互联网络有线性阵列结构；二维的有环形、星形、树形和网格形等；三维的有带弦环形网络、循环移数网络、全连接和立方体网络及其变形等；三维以上的有超立方体等。

3.2.2 静态互联网络的结构

3.2.2.1 一维网络

一维网络又称线性阵列，是互联网络中拓扑结构最简单的，如图 3-17 所示是 N 个结点用

N-1 条链路连成一行，内部结点度为 2，端结点度为 1，直径为 N-1，等分宽度为 1，结构不对称。应当注意，它与总线结构是有区别的，总线结构是通过时分切换使多对结点分时进行通信，而线性阵列允许不同的结点对并发使用系统的不同部分（通道）。线性阵列的 N 较大时，直径比较大，通信效率比较低，且直径随 N 线性增大，因此当 N 比较大时，一般不使用线性阵列的拓扑结构。在 N 很小时，实现线性阵列是相当经济。

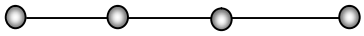
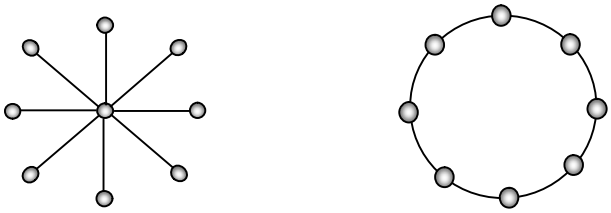


图 3-17　一维网络的拓扑结构

3.2.2.2　二维网络

二维网络拓扑结构容易在 VLSI 芯片上实现，且可扩充性比较好，从而得到广泛应用，如 Intel 公司生产的 Paragon 和 Touchstone Delta 等多处理机系统都使用了二维网络。二维网络主要有四种，即星形、环形、树形和网格形。

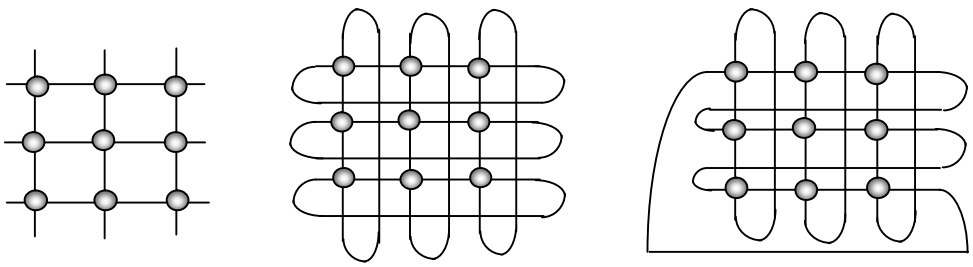
星形和环形二维网络的拓扑结构如图 3-18 所示。星形二维网络是一种二层树，结点度较高为 N–1，直径较小为常数 2，主要用于集中监控系统中。环形二维网络是将线性阵列网络的两个端点用附加链路连接起来，是对称的，结点度为常数 2；环形二维网络可单向工作，也可双向工作；单向环的直径是 N，双向环的直径是 N/2；适合于流水线工作使用。



（a）星形二维网　　（b）环形二维网

图 3-18　星形和环形二维网络的拓扑结构

树形和网格形二维网络会有许多变形。网格形的典型结构是 N=r×r 格式，每格点上有一个结点，如图 3-19（a）所示，内部结点度为常数 4，边结点度为常数 2 或 3，网络直径为 2(r–1)，r=$\sqrt{N}$，是不对称网络。网格形的变形主要有如图 3-19（b）和（c）所示的环形网和 Illiac 网。环形网是在网格形网络的基础上，沿每行每列有环形连接，一般一个 N=r×r 二元环网的结点度为常数 4，网络直径为 2「r/2」，比网格形的减少二分之一，是对称网络。一般一个 N=r×rIlliac 网的网络直径为 r–1，也比网格形的减少二分之一，结点度为常数 4，主要用于阵列处理机上。

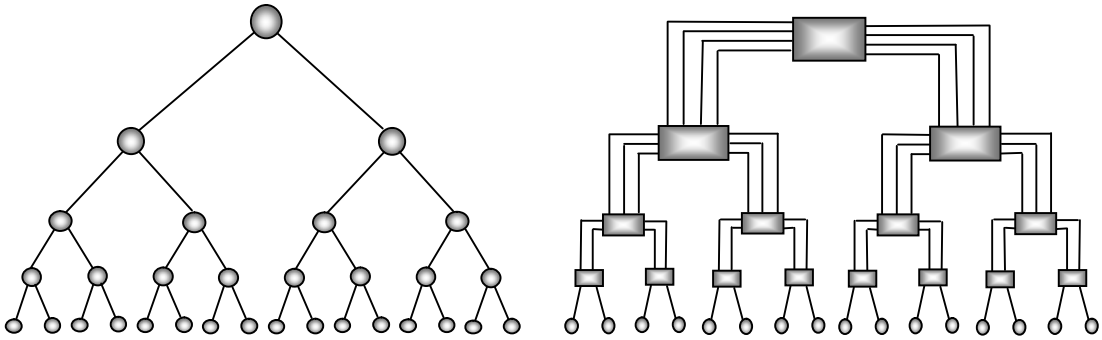


（a）网格形网络　　（b）环形网格网络　　（c）Illiac 网

图 3-19　网格形二维网络及其变形的拓扑结构

树形二维网络主要形式有完全平衡的二叉树和二叉胖树（Fat Tree），如图 3-20 所示。一

棵 r 层的完全平衡二叉树应有 $N=2^{r-1}$ 个结点，最大结点度为常数 3，网络直径很长为 2(r−1)，是一种具有良好扩展性的网络。二叉胖树的通道宽度从叶结点往根结点上行方向逐渐增宽，缓解了完全平衡二叉树根结点通信忙的问题，也更像真实的树。



（a）完全平衡二叉树　　（b）二叉胖树

图 3-20　树形二维网络的拓扑结构

3.2.2.3　三维网络

三维网络的拓扑结构主要有带弦环形网络、循环移数网络、全连接和立方体网络及其变形。

带弦环形网络是环形二维网络的变形，是将环形网的结点度提高，以降低网络直径，提高越多，网络直径越小，如图 3-21 所示是将环形网的结点度 2 增加到 3 和 4，即结点间分别增加一条和两条附加链路，使网络直径由 4 减少到 3 和 2。

（a）度为 3 的带弦环　　（b）度为 4 的带弦环

图 3-21　带弦环形三维网络的拓扑结构

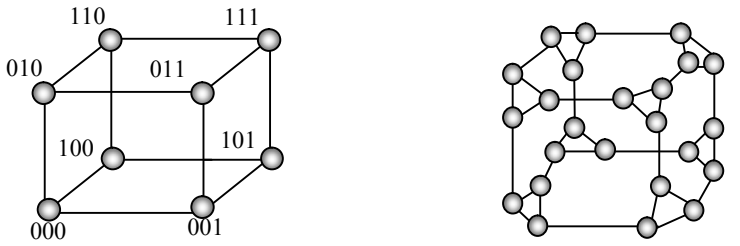
循环移数网络也是环形二维网络的变形，是将环形网上每个结点到与其距离为 2 整数幂的结点增加一条附加链而构成的，它的连接特性和结点度较低的带弦环形网络相比有了改进，但复杂性仍比全连接网络（如图 3-22（b）所示）低得多。如图 3-22（a）所示的循环移数网络的结点度为 5，网络直径为 2。

（a）循环移数网络　　（b）全连接网络

图 3-22　循环移数与全连接三维网络的拓扑结构

立方体网络是典型的三维网络，如图 3-23（a）所示。立方体网络的结点数为 8，结点度

和网络直径均为常数 3。立方体网络的变形是带环立方体网络（3-CCC），如图 3-23（b）所示。是用一个 3 个结点环代替立方体网络角结点（顶角）。带环立方体网络的结点数为 24，结点度为常数 3，网络直径为常数 6。



（a）三维方体网络　　（b）带环三维方体网络

图 3-23　立方体及其变形三维网络的拓扑结构

3.2.2.4　r 维网络

r 维网络的典型结构是 r 维方体（r-cube 或 r-CCC）网络或超立方体网络及其变形结构，该网络结构得到广泛应用。如 Intel 公司生产的 iPSC 系统则采用了方体网络，它的结点数可从 18 到 128 个。

一个 r 维方体网络有 $N=2^r$ 个结点，度数是 r，直径是 r，它是线性阵列拓扑和全连接拓扑之间的一个折中。当然在 N 很大时，度数 r 将很大，硬件成本也很大。特别地 r 维方体网络可很容易扩展为 r+1 维立方体，即只要把两个 r 维立方体对应点用链路连接起来，共要连接 2^r 条链路，如图 3-24 所示的四维方体网络即是通过将两个三维方体网络的相应结点互连组成。

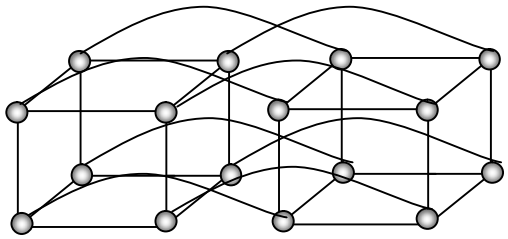


图 3-24　四维方体网络的拓扑结构

r 维方体网络的变形是带环超立方体网络，它是用一个 r 个结点环代替 r 维方体网络角结点（顶角），这样结点数为 $r\times 2^r$,，网络直径为 2r，但结点度与 r 无关，均为常数 3。

特别地立方体和超立方体的拓扑网络有一个特性，即相邻结点的二进制编号仅差一位，而两个结点间的距离正好等于这两个结点二进制编号间不同的位数。如图 3-23（a）中的 000 结点接到 001、010 和 100 结点。而在 001 结点上的信息要送到 111 结点时，可以先从 101 结点走，也可以先从 011 结点走，但不能从 000 走，因 001 同 111 的第三位二进制数已经相同了，如果走 000 就要增加距离。这种拓扑结构的优点之一是在两个结点之间总存在 r 条不同的路径可走，因而容错性就好。例如，从 001 结点到 111 结点（距离为 2）有 3 条路径：

001 ⟶ 011 ⟶ 111

001 ⟶ 101 ⟶ 111

001 ⟶ 000 ⟶ 100 ⟶ 110 ⟶ 111

可以看出，在距离小于 r（例中 r =3）的两个结点间，各条路径的长度并不全是相等的。

3.2.3 静态互联网络特性的比较

如表 3-1 所示汇总了静态互联网络的重要特性。大多数网络的结点度 d 都小于 4，这是比较理想的。全连接网络和星形网络的结点度都太高。超立方体的结点度随 $\log_2 N$ 增大而增大，当 N 值很大时，其结点度也太高。较高的结点度要求为结点提供较多的通信通道。

表 3-1 静态互联网络特性比较一览表

网络类型	结点度 d	网络直径 D	链路数	等分宽度 b	对称性	网络规格评注
线性阵列	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(r−1)	N−1	1	非	树高 r=⌈$\log_2 N$⌋
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 网络，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 网络，r=$\sqrt{N}$
超立方体	r	r	rN/2	N/2	是	N 个结点，r=$\log_2 N$（维）
CCC	3	2r−1+⌈r/2⌋	3N/2	N/(2r)	是	N=r×2^r 结点，环长 r≥3

网络直径 D 的变化范围很大，但随着硬件寻径技术的发展虫蚀寻径已不是一个严重的问题，因为任意两结点间的通信延迟在虫蚀寻径这种高度流水线操作下几乎是固定不变的。链路数会影响网络价格，等分宽度 b 将影响网络的带宽，对称性会影响网络的可扩展性和寻径效率。客观地说，网络的总价格随 d 和 l 增大而上升。且根据以上分析，环形、网格和 CCC 都具备一定的条件，用来建造未来的大规模并行处理（MPP）系统。

3.3 动态互联网络

3.3.1 动态互联网络及其互连形式

动态互联网络可通过设置有源开关，根据需要借助控制信号对连接通路重新组合，实现所要求的通信模式。动态互联网络的形式主要有总线、交叉开关和多级交叉开关等 3 种类型。

3.3.2 总线互联网络

3.3.2.1 总线互联网络及其特点

总线互联网络是指用一组导线和插座将处理机、存储模块和各种外围设备互连起来，实现功能部件间的数据通信。当总线上的各模块需要通信时，功能部件发出申请，由总线仲裁逻辑对多个请求进行仲裁，进行总线服务分配。总线只用于源和目的部件之间确立关系后处理一次业务。总线上各模块是通过争用或时分方式获得总线服务的，所以总线被称为争用总线或分时总线。总线互联网络与其他两种动态互联网络相比，主要特点有以下几个方面：

（1）功能部件间信息传输的带宽低。计算机中多个功能部件共享总线，采用分时方式实

现数据交换，即同一时刻只能有一个功能部件发送信息。

（2）计算机组装方便，扩展性好。功能部件间交换信息的总线标准化，使得功能部件间连接的接口标准化，与总线标准相匹配的功能部件都可挂在总线上成为计算机的一个部分。

（3）计算机体系结构简单，成本低。功能部件间的连接关系直观，可简化体系结构、硬件与软件的设计，减轻软件调试，缩短软硬件研制周期。

（4）有很多可用的工业标准，如 IEEE 总线标准。

总线连接的多处理机系统的系统总线为多个处理机、I/O 子系统、主存的多个存储模块和辅助存储设备之间提供了一条公用通信通路，每台处理机或 I/O 设备可产生访问存储器的请求，存储器或外围设备则响应该请求。

3.3.2.2　总线互联网络的基本技术

总线互联网络的基本技术主要有总线仲裁、中断处理、Cache 一致性协议和总线事务的处理等。其中总线仲裁最为重要，一般用硬件实现，常用的算法主要有 4 种：一是静态优先级算法，即各功能部件的优先级是固定的，通常以串行链上的物理位置来决定；二是固定时间片算法，即将总线可用的带宽分成固定的时间片，且按循环方式顺序分配于各功能部件；三是动态优先级算法，即各功能部件的优先级可以调整，使它们都有机会使用总线，如采用 LRU 算法等；四是先来先服务算法，即按接收到的请求顺序分配总线的使用。

3.3.3　交叉开关互联网络

3.3.3.1　交叉开关互联网络及其特点

交叉开关（Crossbar）互联网络是利用一组纵横交错的开关阵列，把各功能部件互连起来，实现功能部件间的数据通信。开关阵列中的开关可由程序控制动态设置其处于“开”与“关”状态，而提供结点对之间的动态连接。交叉开关互联网络实际是多总线向总线数量增加方向发展的极端情况，总线数等于全部相连的模块数，从而大大加宽了互连传输频带，提高了系统的效率。如图 3-25 所示把横向的 S 个处理机及 I 个 I/O 设备与纵向的 N 个存储器模块连接起来，总线数等于全部相连的模块数（N+I+S），且 N≥I+S，使 S 个处理机和 I 个 I/O 设备都能分到一套总线与 N 个存储器模块中的某个相连。

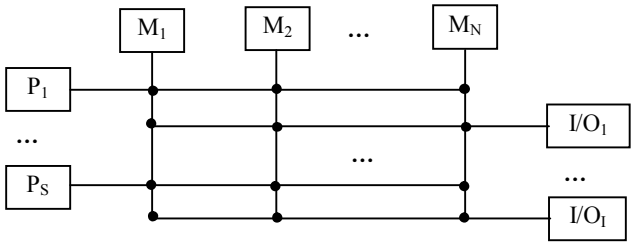


图 3-25　交叉开关阵列互连形式

交叉开关实际是一种单级的互联网络，采用无阻塞的形式实现输入输出端的连接。但在数据传送过程中仍然会有端口冲突的情况，原因可能有多个输入端的数据分组转发到同一个输出端。交叉开关与总线相比，交叉开关采用按空间分配的机制，而总线互联网络采用按时间分配的机制。

3.3.3.2　结点开关的结构模型

图 3-25 所示的交叉开关中，每一个交叉点都表示一套开关，即结点开关。结点开关不仅

要有多路转接逻辑，还要具有处理访问存储器冲突的仲裁硬件，加上总线本身有一定的宽度，使得整个交叉开关阵列相当复杂。

C.mmp 的 16×16 处理机－存储器模块的交叉开关阵列中一个结点开关的结构模型如图 3-26 所示，它主要是由仲裁模块和多路转换模块两部分组成。16 个处理机都可以为请求访问某一个存储器模块而给相应的结点开关的仲裁模块发出请求信号，由仲裁模块按一定的算法响应具有最高优先级的请求，且返回一个应答信号。该处理机或 I/O 设备接到应答信号后，就经多路转换模块开始访问相应的存储器模块。多路转换器是一个 16 选 1 的多路选择器，它受仲裁模块控制，为仲裁模块确定的处理机同存储器模块之间建立连接，进行数据、地址和读/写信息的传送。美国的 C.mmp 和 S-1 系统都是采用交叉开关互连的多处理机系统，它们都包含 16 台处理机。

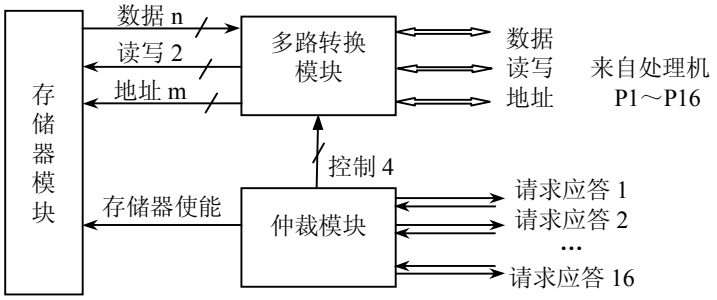


图 3-26　交叉开关中结点开关的结构模型

3.3.3.3　交叉开关模块

一个 a×b 交叉开关模块有 a 个输入端和 b 个输出端。理论上 a 和 b 不一定相等，但实际上经常选 a=b，且为 2 的整数幂，即：$a=b=2^k$，$k\geqslant 1$。常用为 2×2、4×4 和 8×8 的开关模块。交叉开关模块的每个输入端可与一个或多个输出端相连，而且容许一对一或一对多的连接或映射，但不允许有多对一的连接，因为多个输入端同时争用一个输出端的冲突会导致通过这个开关传送信息被阻塞。

只允许一对一映射的 n×n 的交叉开关模块，有 n 个输入端和 n 个输出端，结点开关数为 n^2，输入端与输出端的合法状态（连接模式）为 n^n，可实现的连接或置换为 n!。例如，4×4 的交叉开关，含有 16 个结点开关，合法的状态 256，可实现的连接为 24。

3.3.4　多级交叉开关互联网络

3.3.4.1　多级交叉开关互联网络的基本概念

交叉开关互联网络是一种单级的互联网络，输入端的数据经过一个开关元件就被输出，而交叉开关阵列是非常复杂的。当纵向和横向的总线数都为 n 时，交叉开关阵列的所有交叉点的设备量是 $O(n^2)$。n 很大时，其成本可能会超过连接的 2n 个部件（包括 n 个处理机和 I/O 设备、n 个存储器模块）的成本，因此，采用交叉开关的多处理机一般 n≤16，少数有 n=32。

由于大规模交叉开关的复杂性，人们一直在寻求改进交叉开关结构的各种方式。改进的基本思想是：通过采用多个较小规模的交叉开关的“串连”和“并连”来构成一个多级交叉开关互联网络，以取代单个的大规模交叉开关。例如，用 8 个 4×4 交叉开关模块可组成一个 16×16 的二级交叉开关网络，每一级有 4 个 4×4 交叉开关组成，两级间采用某种固定连接。8

个 4×4 交叉开关模块组成的多级交叉开关互联网络的结点开关数是 128 个，而单一的 16×16 的交叉开关阵列的开关结点是 256 个，前者结点开关设备量仅为后者的一半。

多级交叉开关互联网络是把重复设置的多套动态单级交叉开关网络串并联起来，级间串联的交叉开关之间采用固定的级间连接模式，同级的交叉开关之间相互独立，通过动态控制各级上的交叉开关的结点开关状态来实现多级交叉开关互联网络的输入端和输出端之间所需的连接。多级交叉开关互联网络一般简称为多级互联网络。许多 MIMD 和 SIMD 计算机都使用多级交叉开关互连。

3.3.4.2　多级交叉开关互联网络的结构模型

多级互联网络结构模型如图 3-27 所示，它需要用 3 个参数来描述，即开关模块、级间连接（ISC）模式和控制方式。

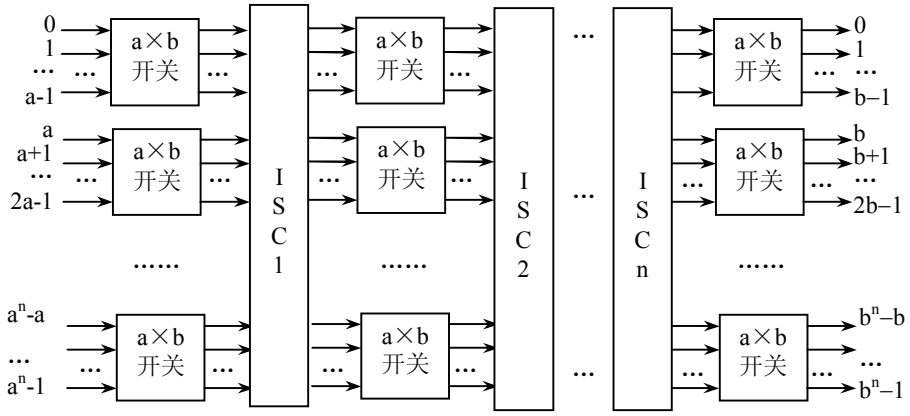


图 3-27　多级交叉开关互联网络的结构模型

1. 开关模块

在多级交叉开关互联网络中一般采用最简单的 2×2 开关模块，2×2 开关模块有 4 种合法的工作状态，即直送、交叉、上播和下播，如图 3-28 所示，但它有两种类型。一是只有“直送”和“交叉”两种工作状态的开关，则称为二功能交叉开关；另一是具有 4 种合法的工作状态的开关，则称为四功能交叉开关。

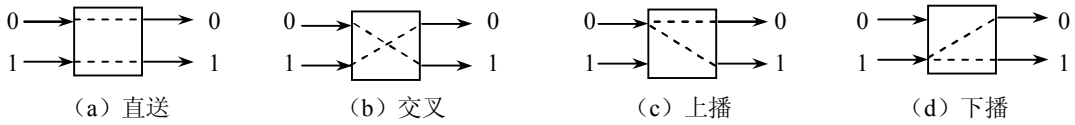


图 3-28　2×2 开关模块的 4 种合法状态

2. 级间连接模式

级间连接模式（ISC）是指多级交叉开关互联网络中上一级开关模块的输出端和下一级开关模块的输入端相互连接的模式。级间连接是固定的，可以用互连函数表示级间连接模式。常用的级间连接模式有均匀洗牌、蝶式、多路洗牌、纵横交叉和立方体连接等。

3. 控制方式

为了使各级交叉开关的输入端和输出端建立所需的连接，可通过控制信号动态控制开关模块的工作状态来实现，即通过对开关模块的状态控制来实现对多级互联网络要求实现的互连，这称为互联网络拓扑结构的可动态重构，其控制的方式有以下 3 种：

（1）级控制。对同一级的所有交叉开关只用一个控制信号进行控制，使同一级的所有交叉开关同时处于同一工作状态。

（2）组控制。对同一级的所有交叉开关分组控制，第 i 级的所有交叉开关分别用 i+1 个信号进行控制，即分为 i+1 组，同一组上交叉开关同时处于同一工作状态，0≤i≤n−1，n 为级数。

（3）单元控制。每一个交叉开关模块都有自己单独的控制信号进行控制，使各个交叉开关可以处于不同工作状态。

3.3.4.3 多级交叉开关互联网络的种类

虽然众多的多级交叉开关互联网络在结构模型上可以用图 3-24 来表示，但在开关模块、级间连接（ISC）模式和控制方式上各有不同，从而形成各种不同的多级互联网络。多级交叉开关互联网络可分为阻塞网、可重排非阻塞网和非阻塞网 3 种类型。

阻塞网络是指一对以上输入端和输出端可同时实现互连的网络中，可能发生两个或两个以上的输入端对输出端的连接要求产生路径争用冲突。各种阻塞网络可实现一些典型互连函数表示的连接，但不能实现任意的互连函数表示的连接。由于阻塞网所用开关数量少，延时也不长，路径控制较简单，能实现并行处理中许多常用的互连函数，所以在实际系统中使用广泛。有代表性的阻塞网有Ω网络、STARAN 网络、间接二进制 n 方体网络、基准网络、δ 网络、数据变换网络等。

可重排非阻塞网络是指如果改变开关状态，一方面重新安排现有连接的通路，另一方面为新连接安排通路，满足一个新的端点对的连接请求，从而就可实现无阻塞的任意端点对的连接，即可实现任意的互连函数。有代表性的可重排非阻塞网络有可重排 Clos 网络、Benes 二进制置换网络等。

非阻塞网络是指不必改变原来的开关状态就可满足任意输入端和输出端之间的连接请求。它与可重排非阻塞网是不同的，可重排非阻塞网要通过改变原来的开关状态来改变连接的路径，才能满足新的连接请求。因此，非阻塞网是连接能力最强的多级互联网络。交叉开关网络属于单级非阻塞网，对称和非对称多级 Clos 网络属于多级非阻塞网。

特别地所谓互联网络实现了某种互连函数是指该互连函数表示的连接关系在该互联网络中可同时建立而不会产生路径争用冲突的现象。

3.3.5 动态互联网络特性的比较

总线、交叉开关、多级交叉开关等 3 种动态互联网络的主要特性如表 3-2 所示。其中在总线中，ω为总线上数据通路的宽度，n 为总线上连接的分接头数；在交叉开关中，ω为交叉开关设计上的通路宽度，n 为交叉开关结点的行数和列数；在多级交叉开关中，ω为多级交叉开关设计上的最小链路宽度，n 为多级交叉开关的级数，k 为 k×k 的交叉开关模块。

1. 硬件复杂性

用连线和交换开关表示复杂性。

总线互连的成本最低。连线的复杂性主要由总线设计的数据通路的宽度和地址线的宽度决定。256 根数据线和 42 根地址线代表了当今总线设计的水平。地址线可被隐蔽而成为数据通路的一部分，例如，256 位的 Futurebsu 总线中，64 根线由地址和数据总线共享。总线开关的复杂性由在总线上连接的分接头数 n 决定，这也受到只能以小数目的处理器、存储器和 I/O 板连接到总线上的限制。设ω为总线上数据通路的宽度，总线互连的硬件复杂性随 n 和ω两者

线性增加，可用函数 O(n+ω)表示。

交叉开关是最昂贵的，因为它的硬件复杂性随 $n^2\omega$的乘积增大。对于相同的数据通路宽度，n×n 交叉开关的价格几乎是总线互连的 n^2 倍。

一个 n 输入的多级交叉开关，硬件复杂性的函数为 $O(n\omega\log_k n)$，其中 $n\log_k n$ 相应于所使用的交叉开关的数目。多级交叉开关硬件复杂性位于总线和交叉开关网络这两种极端的情况之间。对于相同的通路宽度，可以粗略地估计多级交叉开关价格比交叉开关便宜 $n/\log_k n$ 倍。

表 3-2　动态互联网络特性比较一览表

网络特性	总线	交叉开关	多级交叉开关
单位数据传输的最小时延	恒定	恒定	$O(\log n)$
每台处理机的带宽	$O(\omega/n)$ 到 $O(\omega)$	$O(\omega)$ 到 $O(n\omega)$	$O(\omega)$ 到 $O(n\omega)$
连线的复杂性	$O(\omega)$	$O(n^2\omega)$	$O(n\omega\log_k n)$
开关的复杂性	$O(n)$	$O(n^2)$	$O(n\log_k n)$
连接特性和寻径性能	一次只能一对一	全置换，一次一个	只要不阻塞，可实现某些置换和广播

2. 每台处理器带宽

总线由 n 个功能部件分时共享，因此 n 个处理器竞争总线带宽。假设相同的时钟频率为 f，在 3 种互联网络中每单位数据传输都仅需一个时钟周期，那么总线的每个处理器带宽在函数 $O(\omega f/n)$和 $O(\omega f)$范围内变化。而交叉开关和多级交叉开关具有较宽的处理器带宽，该带宽随函数 $O(\omega f)$线性变化。

总线只需较少时间（通常是 1 或 2 周期）来传输单位数据片，而多级交叉开关需要多个时钟周期经过多级开关传输一个数据片。因此，总线带宽并不比多级交叉开关带宽低很多。在所有情况下，交叉开关具有最高的处理器带宽，因为它用较短时延（也是 1 或 2 周期）与输入输出端口实现的无冲突连接。

3. 聚集带宽

Gigaplane 总线的聚集带宽为 2.67GB/s=21.36Gbps，假设 n=24 个处理器共享总线带宽，那么每个处理器的带宽降低为 21.36Gb/s/24=0.89Gb/s。GIGAswitch 交叉开关聚集带宽为 3.4 Gb/s，意味着 GIGAswitch 潜在的每个处理器带宽比 Gigaplane 总线高 3.8 倍。IBM HPS 多级交叉开关在最大配置（n=512 个端口）时，具有的聚集带宽为 10.24GB/s=81.92 Gb/s。显然，多级交叉开关比总线或交叉开关互连具有更好的扩展性。

总之，多级交叉开关是总线和交叉开关之间的折中，每台处理器带宽与交叉开关相同，硬件复杂性介于总线和交叉开关之间。多级交叉开关的最主要优点是采用模块结构，可扩展性较好，但时延随级数增加而对数上升。

3.4　常用的多级交叉开关动态互联网络

3.4.1　Ω多级动态网络（Omega 网络）

3.4.1.1　Ω网络的结构及其特点

Ω网络又称为多级洗牌置换网络，若它的输入端或输出端数为 N，则Ω网络的交叉开关级

数为 $n=\log_2 N$，每级有 N/2 个交叉开关，故Ω网络的交叉开关数为$(N/2)\log_2 N$。Ω网络的结构特点是：

（1）采用 2×2 的 4 功能交叉开关，4 个功能为直送、交叉、上播、下播。

（2）各级交叉开关的级号编排是从网络输入端到输出端，依次为 K_{n-1}、…、K_1、K_0。

（3）级间连接从网络输入端到输出端依次分别表示为 C_n、…、C_1、C_0，其中，C_1～C_n都是均匀洗牌置换函数，C_0是恒等置换函数。

因此，Ω网络的输入端对输出端的互连函数表达式为：

$$\Omega(n)=\sigma_n H_{n-1}\sigma_{n-1}H_{n-2}\cdots\sigma_1 H_0 I_0=(\sigma H)^n$$

其中：H_i是 K_i级交叉开关在单元控制下实现的置换函数（$0\leqslant i\leqslant n-1$），$\sigma_j$是 C_j级级间连接模式实现的均匀洗牌置换函数（$1\leqslant j\leqslant n$），I_0 是 C_0 级级间连接模式实现的恒等置换函数。故可称Ω网络为多级洗牌置换网络，N=8 的Ω网络的结构如图 3-29 所示。

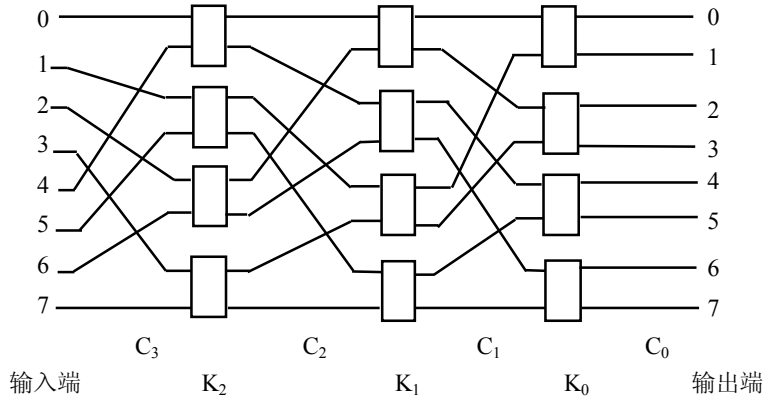


图 3-29　N=8 的Ω网络结构

3.4.1.2　Ω网络的开关控制与寻径算法

Ω网络对交叉开关状态采用单元控制方式来获得所需要的输入端到输出端的连接路径，交叉开关单元控制采用终端标记寻径算法。所谓终端标记寻径算法的含义是指：以终端的二进制地址 D 中的各位作为控制信号，来控制从源端到终端所经过路径上的各级交叉开关的工作状态，实现源端到终端的连接来保证数据正确传送。终端标记寻径算法具体如下：

设Ω网络输入端二进制地址编号为 $S=s_{n-1}s_{n-2}\cdots s_1 s_0$，输出端二进制地址编号为 $D=d_{n-1}d_{n-2}\cdots d_1 d_0$。从输入端 S 开始，第 i 级 K_i 交叉开关状态由终端地址 D 的相应二进制数位 d_i 控制。若 $d_i=0$，则 K_i 级上对应交叉开关的输入端与上输出端相连；若 $d_i=1$，则 K_i 级上对应交叉开关的输入端与下输出端相连。

如图 3-30 所示，当源端地址为 S =010，终端地址为 D=110 时，连接源端 010 的 K_2 级的交叉开关因终端地址 D =110 的 $d_2=1$，故使该交叉开关输入端与下输出端相连。同样，因 $d_1=1$，故 k_1 级的对应交叉开关的输入端与下输出端相连；因 $d_0=0$，故 K_0 级的对应交叉开关的输入端与上输出端相连。从而完成了网络源端 S=010 到终端 D=110 的连接。

对Ω网络的源端集合到终端集合的连接，都可用终端标记法来控制交叉开关的工作状态。但由于终端标记法使每一个源端终端对的连接路径是唯一的，因此不能保证不发生争用交叉开

关状态的冲突。例如要实现(000,000)和(100,010)两对同时连接，就会发生 K_2 级交叉开关的冲突，如图 3-30 所示，即Ω网络是一种阻塞网。

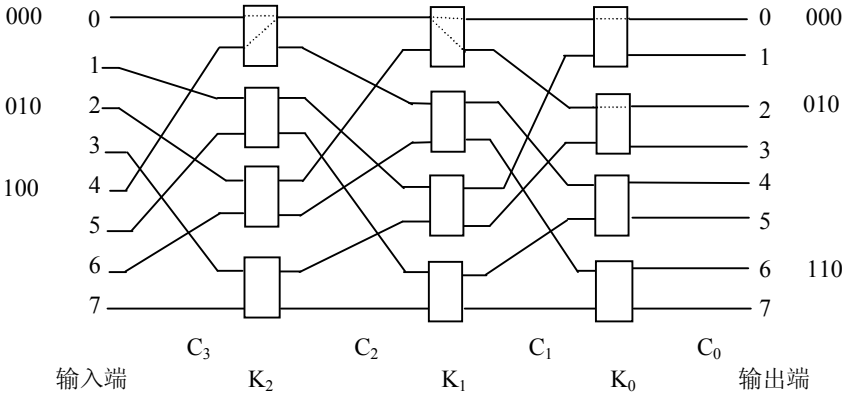


图 3-30　N=8 的Ω网络发生争用交叉开关冲突

3.4.1.3　Ω网络可实现的互连函数

（1）恒等置换。

f(x)=x　　其中 0≤x<N

（2）按 c 序散播加上距离为 d 的移数置换。

f(x)=cx+d　　其中 0≤x<N，c 为奇数

（3）a 序向量的收集加上距离为 b 的移数置换。

f(ax+b)=x　　其中 0≤x<N，a 为奇数

在数组中，Ω网络可完成按行、列、对角线和子块等无冲突的访问，因此Ω网络的应用极广泛。但不能用Ω网络来实现源端集合到终端集合的均匀洗牌、蝶式和位序颠倒等置换，因为这些置换连接要求会发生争用开关的冲突。

3.4.2　STARAN 多级动态网络

3.4.2.1　STARAN 网络的结构及其特点

若STARAN 网络的输入端或输出端数为 N，则 STARAN 网络的交叉开关级数为 $n=\log_2 N$，每级有 N/2 个交叉开关，故 STARAN 网络的交叉开关数为 $(N/2)\log_2 N$。STARAN 网络的结构特点是：

（1）采用 2×2 的 2 功能交叉开关，两个功能为直送和交叉。

（2）各级交叉开关的级号编排从网络的输入端到输出端，依次将为 K_0、K_1、…、K_{n-1}。

（3）级间连接模式从网络输入端到输出端依次表示为 C_0、C_1、…、C_n，其中，C_0 是恒等置换，C_1～C_n 都是逆洗牌置换。

因此，STARAN 网络的输入端对输出端的互连函数表达式为：

$$STARAN(n)=I_0H_0\sigma_1^{-1}H_1\sigma_2^{-1}\cdots H_{n-1}\sigma_n^{-1}=(H\sigma^{-1})^n$$

其中：H_i 是 K_i 级交叉开关级在级控制或组控制下实现的置换函数（$0\leqslant i\leqslant n-1$），$\sigma_j^{-1}$ 是 C_j 级级间连接模式实现的逆洗牌置换函数（$1\leqslant j\leqslant n$），I_0 是 C_0 级级间连接模式实现的恒等置换函数。N=8 的 STARAN 网络的结构如图 3-31 所示。

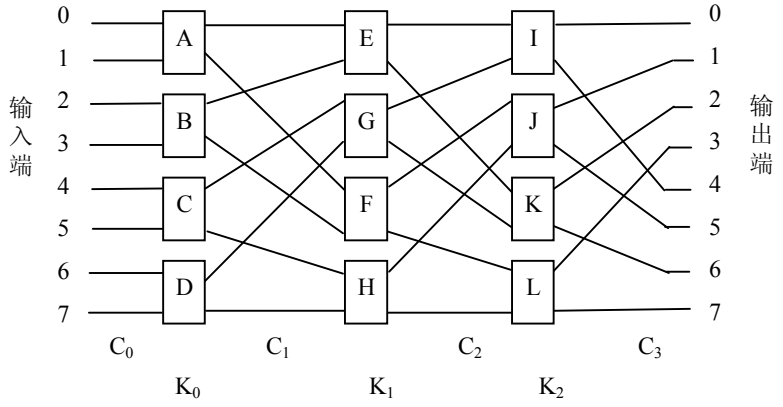


图 3-31　N=8 的 STARAN 网络结构

3.4.2.2　STARAN 网络的开关控制

STARAN 网络的交叉开关控制方式有两种：一种是级控制，可实现方体置换，故级控制的 STARAN 网络又被称为方体网络；另一种是组控制，可实现移数置换，故组控制的 STARAN 网络又被称为移数网络。

1. 级控制

在级控制下，对于一个两功能交叉开关，只需一个控制位信号 f 就可控制交叉开关的工作状态。交叉开关输出 V(x)与输入 x 的连接和控制位 f 的关系可表示为：

$$V(x)=x \oplus f$$

若 f=0，则 V(x)=x，开关为直送连接；若 f=1，则 V(x)=$\overline{x}$，交叉开关为交叉连接。

对于级控制，可以用二进制向量 F=($f_{n-1}f_{n-2}\cdots f_1f_0$)表示网络的控制信号，F 的位分量 f_i 就是交叉开关级 K_i 的所有交叉开关的控制位信号。这样 STARAN 网络各级交叉开关实现的互连函数为：

$$V(x_{n-1}x_{n-2}\cdots x_1x_0)=(x_{n-1} \oplus f_{n-1},x_{n-2} \oplus f_{n-2},\cdots,x_1 \oplus f_1,x_0 \oplus f_0)$$

2. 组控制

STARAN 网络的组控制是指将第 i 级的 N/2 个 2×2 的 2 功能开关分成 i+1 组，每组用一个位信号控制交叉开关的工作状态。对于一个两功能交叉开关，级数是 $n=\log_2 N$ 的 N×N 的 STARAN 网络，从第 0 级至第 n−1 级所需的控制信号位数分别为 1、2、…、n 个，因此，共需 n(n+1)/2 个位组成二进制控制向量 F。对于 N=8 的网络，需要 6 个位信号组成控制向量 F，表示为 F=(f_{23} f_{22} f_{21} $f_{12}$$f_{11}$ f_0)。交叉开关输出 V(x)与输入 x 的连接和控制位 f 的关系与级控制相同。实现移数置换的互连函数为：

$$\alpha(x)=(x+2^m) \bmod 2^p$$

其中，p 和 m 都是整数，且 $0 \leqslant m < p \leqslant n$。

3.4.2.3　STARAN 网络可实现的互连函数

1. 级控制方式的方体置换

在级控制方式下，含有 n 级的 STARAN 网络需要的二进制控制信号向量 F 为 n 位，可取 $N=2^n$ 个不同组合值，从而可使 N×N 的 STARAN 网络实现 N 种置换。N=8 的 STARAN 网络分别在 8 种级控信号（000～111）控制下，实现的网络输入端到输出端的连接如表 3-3 所示。

表 3-3　3 级 STARAN 网络入出端连接及实现的方体函数功能（f_i 为 k_i 级控制信号）

	级控制信号（f_2 f_1 f_0）							
	000	001	010	011	100	101	110	111
入端号 0	0	1	2	3	4	5	6	7
入端号 1	1	0	3	2	5	4	7	6
入端号 2	2	3	0	1	6	7	4	5
入端号 3	3	2	1	0	7	6	5	4
入端号 4	4	5	6	7	0	1	2	3
入端号 5	5	4	7	6	1	0	3	2
入端号 6	6	7	4	5	2	3	0	1
入端号 7	7	6	5	4	3	2	1	0
执行方体函数功能	恒等	4 组 2 元	4 组 2 元 + 2 组 4 元	2 组 4 元	2 组 4 元 + 1 组 8 元	4 组 2 元 + 2 组 4 元 + 1 组 8 元	4 组 2 元 + 1 组 8 元	1 组 8 元
	i	$Cube_0$	$Cube_1$	$Cube_0$+ $Cube_1$	$Cube_2$	$Cube_0$+ $Cube_2$	$Cube_1$+ $Cube_2$	$Cube_0$+ $Cube_1$+ $Cube_2$

由表 3-3 可见，除 F=(000)实现恒等置换外，其他 7 种级控信号实现的置换是分组方体置换。例如，级控信号 F=(010)实现的置换可看成是：将输入端号序列[0　1　2　3　4　5　6　7]先分成 4 组[0　1]、[2　3]、[4　5]、[6　7]，组内 2 元交换后为[1　0] 、[3　2]、[5　4]、[7　6]，排列序列为[1　0　3　2　5　4　7　6]；再分成两组[1　0　3　2] [5　4　7　6]，组内 4 元交换后为[2　3　0　1][6　7　4　5]，得到输入端序列按序连接的输出端序列为[2　3　0　1　6　7　4　5]。因此，把级控制方式下的 STARAN 网络称为方体网络。

N=8 的 STARAN 网络实现的方体置换的图形表示如图 3-32 所示。把图 3-32 中 F=(001)、(010)和(100)的 3 个图形表示与图 3-6 所示的 N =8 的方体置换图形表示比较，可以看出：F=(f_2 f_1 f_0)=(001)的方体置换就是 $Cube_0$ 置换，F =(010)的方体置换就是 $Cube_1$ 置换，F =(100) 方体置换就是 $Cube_2$ 置换。即 f_i =1 时，实现 $Cube_i$ 置换，例如当级控信号为 F=(011)时，3 级 STARAN 网络实现的方体置换为：

$$C(x_2x_1x_0) = Cube_1(Cube_0(x_2x_1x_0)) = \overline{x}_2\overline{x}_1\overline{x}_0$$

简记为 $Cube_0$+$Cube_1$。实际上同样有：

$$C(x_2x_1x_0) = (x_2 \oplus f_2, x_1 \oplus f_1, x_0 \oplus f_0) = (x_2 \oplus 0, x_1 \oplus 1, x_0 \oplus 1) = \overline{x}_2\overline{x}_1\overline{x}_0$$

2. 组控制方式的移数置换

在级控制方式下，含有 n 级的 STARAN 网络需要的二进制控制信号向量 F 为 n(n+1)/2 位，但仅可取$(n^2+n+2)/2$ 不同的组合值（n(n+1)/2 二进制数有 $2^{n(n+1)/2}$ 个不同的组合值，但仅有$(n^2+n+2)/2$ 不同的组合值可使网络不会发生争用交叉开关冲突），即一个 $N\times N=2^n\times 2^n$ 的 STARAN 网络能实现$(n^2+n+2)/2$ 种置换。例如 N=8 的 STARAN 网络，需要的二进制控制信号向量 F 为 6 位，分别在 7 种不同的组合值的组控信号控制下能实现 7 种移数置换，实现的网络输入端到输出端的连接如表 3-4 所示。K_0 级只有一个位信号 f_0 控制 K_0 级的开关 A、B、C、D；K_1 级有两个位信号 f_{11} 和 f_{12}，f_{11} 控制开关 E 和 G，f_{12} 控制开关 F 和 H；K_2 级有 3 个位信

号 f_{21}、f_{22} 和 f_{23}，f_{21} 控制开关 I，f_{22} 控制开关 J，f_{23} 控制开关 K 和 L。实现的 7 种移数置换的图形表示如图 3-33 所示。

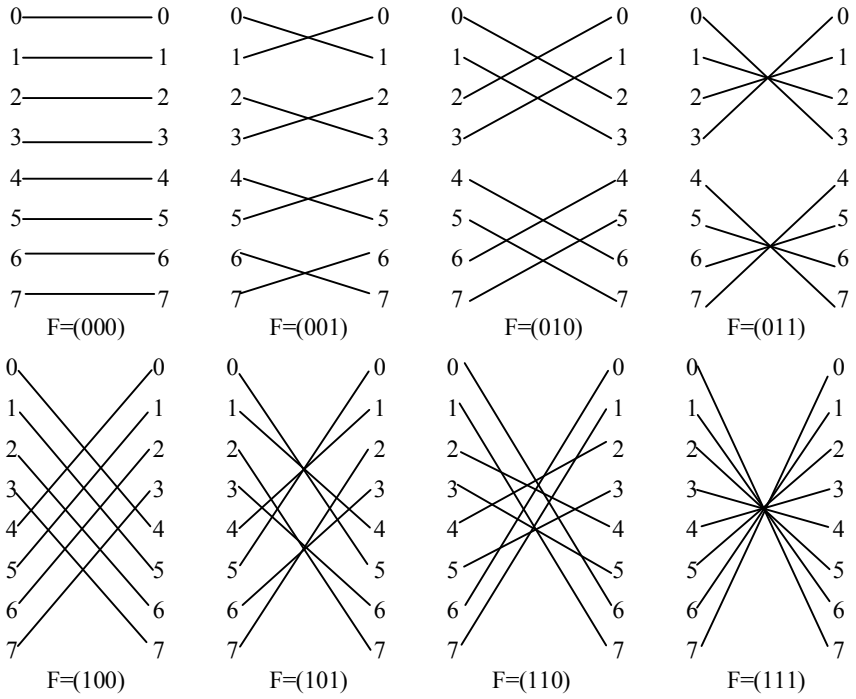


图 3-32　N =8 的 STARAN 网络实现的交换置换

表 3-4　3 级 STARAN 网络入出端连接及实现的的移数函数功能

组控制信号										
	2 级	f_{23}	K，L	0	0	1	0	0	0	0
		f_{22}	J	0	1	1	0	0	0	0
		f_{21}	I	1	1	1	0	0	0	0
	1 级	f_{12}	F，H	0	1	0	0	1	0	0
		f_{11}	E，G	1	1	0	1	1	0	0
	0 级	f_0	A，B，C，D	1	0	0	1	0	1	0
入端号			0	1	2	4	1	2	1	0
			1	2	3	5	2	3	0	1
			2	3	4	6	3	0	3	2
			3	4	5	7	0	1	2	3
			4	5	6	0	5	6	5	4
			5	6	7	1	6	7	4	5
			6	7	0	2	7	4	7	6
			7	0	1	3	4	5	6	7
执行的移数功能				移 1 mod 8	移 2 mod 8	移 4 mod 8	移 1 mod 4	移 2 mod 4	移 1 mod 2	不移恒等

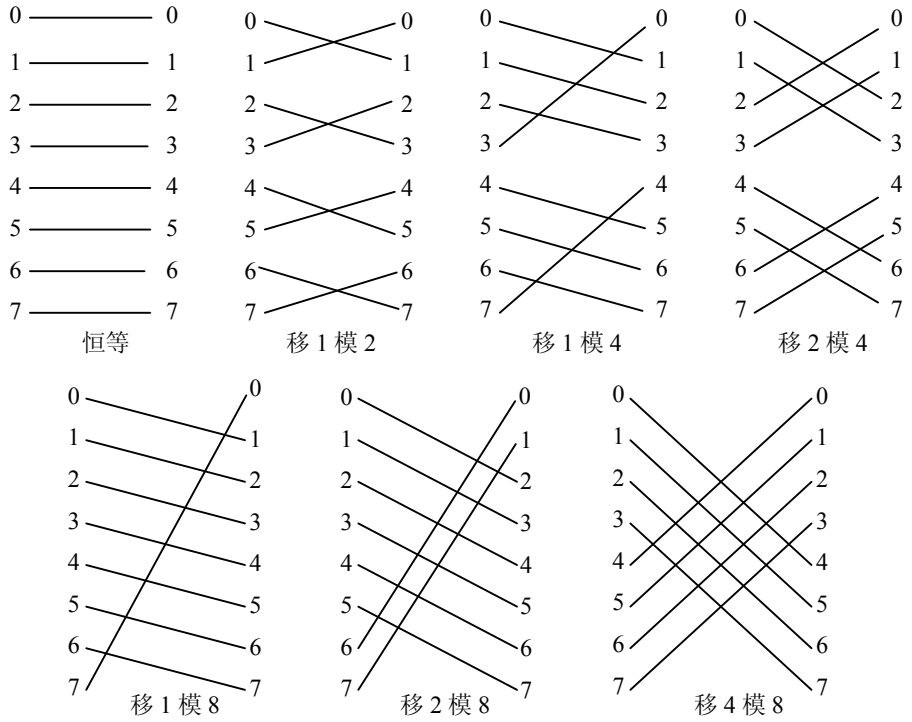


图 3-33　N=8 的 STARAN 网实现的移数置换

3.4.3　间接二进制 n 方体多级动态网络

3.4.3.1　n 方体网络的结构及其特点

若间接二进制 n 方体网络的输入端或输出端数为 N，则网络级数为 $n=\log_2 N$，每级有 N /2 个交叉开关，故 n 方体网络的开关数为$(N/2)\log_2 N$。间接二进制 n 方体网络的结构特点是：

（1）采用 2×2 的 2 功能交叉开关，两个功能为直送和交叉。

（2）各级交叉开关级的级号编排从网络输入端到输出端，依次为 K_0、K_1、…、K_{n-1}。

（3）间连接模式从网络输入端到输出端依次表示为 C_0、C_1、…、C_n，其中，C_0 是恒等置换，C_1～C_{n-1} 都是子蝶式置换，C_n 是逆洗牌置换。

因此，间接二进制 n 方体网络的输入端对输出端的互连函数表达式为：

$$\text{文体}(n)=I_0H_0\beta_{(1)}H_1\cdots\beta_{(n-1)}H_{n-1}\sigma_n^{-1}$$

其中，H_i 是 K_i 级交叉开关在单元控制方式下实现的置换函数（$0\leqslant i\leqslant n-1$），$\beta_j$ 是 C_j 级级间连接模式实现的子蝶式置换函数（$1\leqslant j\leqslant n-1$），$\sigma_n^{-1}$ 是最后一级级间连接模式的逆洗牌置换函数，I_0 是 C_0 级级间连接模式实现的恒等置换函数。N=8 的间接二进制 n 方体网络的结构如图 3-34 所示。

3.4.3.2　n 方体网络的开关控制与寻径算法

间接二进制 n 方体网络的交叉开关控制采用单元控制方式。网络共有 $n=(N/2)\log_2 N$ 个开关，每个开关有两种状态，因此，网络有 2^m 个不同状态，即网络能实现 2^n 种置换连接。单元控制方式可以采用前述Ω网络应用的终端标记法，也可以采用下述的网络输入端与输出端二进制地址按位加的控制方法。

设输入端二进制地址为 $S=s_{n-1}s_{n-2}\cdots s_1s_0$，输出端二进制地址为 $D=d_{n-1}d_{n-2}\cdots d_1d_0$，则 $s_i\oplus d_i$

的值决定了开关级 K_i 上相应交叉开关的状态。若 $s_i \oplus d_i=0$，则 K_i 上相应交叉开关为直送状态；若 $s_i \oplus d_i=1$，则 K_i 上相应交叉开关为交叉状态。例如，当源端地址为 S=100 和终端地址为 D=010 时，从输入端 4 到输出端 2 的路径上的 K_0 级相应开关为直送，K_1 级和 K_2 级的相应开关为交叉。若同时有从输入端 7 到输出端 6 的连接要求，则会发生争用开关冲突。所以间接二进制 n 方体网络是阻塞网。

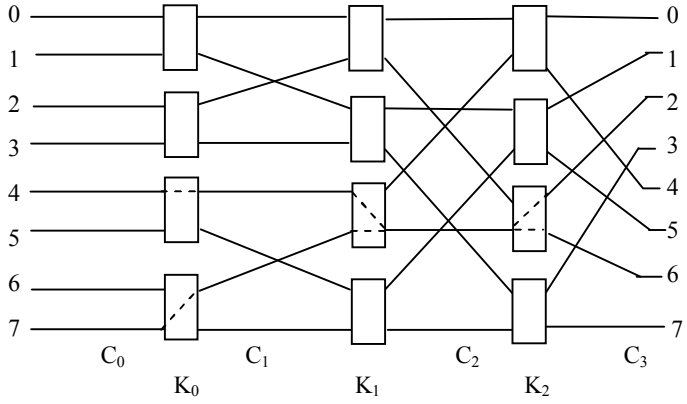


图 3-34　N=8 的间接二进制 n 方体网络结构

3.4.3.3　n 方体网络可实现的互连函数

间接二进制 n 方体网络的连接能力很强，可实现多种常用的函数置换，例如，移数置换、子移数置换、P 序置接、逆 P 序置换、P 序加移数置换和交换置换等。

3.4.4　δ多级动态网络

3.4.4.1　δ网络结构及其特点

δ（Delta）网络的一般结构形式为如图 3-35 所示的 $a^n \times b^n$ 结构，网络级为 n。δ网络的结构特点是：

（1）采用 a×b 的交叉开关，各级交叉开关的级号编排从网络的输入端到输出端，依次将为 K_0、K_1…、K_{n-1}。

（2）级间连接模式从网络输入端到输出端依次表示为 C_0、C_1、…、C_n，级间连接模式为 q 洗牌置换函数。

（3）每一级的交叉开关数为：第一交叉开关级 K_1 的交叉开关数是 a^{n-1} 个，最后交叉开关级 K_{n-1} 的交叉开关数是 b^{n-1} 个，中间交叉开关级 K_i 的交叉开关数是 $a^{n-i}b^{i-1}$ 个。

（4）每一级交叉开关的输入端与输出端数为：第一交叉开关级 K_1 的输入端数是 $a^{n-1} \times a=a^n$，输出端数是 $a^{n-1} \times b=a^{n-1}b$；最后交叉开关级 K_{n-1} 的输入端数是 $b^{n-1} \times a=ab^{n-1}$，输出端数是 $b^{n-1} \times b=b^n$；中间交叉开关级 K_i 的输入端数是 $a^{n-i}b^{i-1} \times a^1=a^{n-i+1}b^{i-1}$，输出端数是 $a^{n-i}b^{i-1} \times b=a^{n-i}b^i$；中间交叉开关级 K_{i+1} 的输入端数是 $a^{n-i-1}b^i \times a^1=a^{n-i}b^i$，输出端数是 $a^{n-i-1}b^i \times b=a^{n-i}b^{i+1}$。显然，$K_i$ 级交叉开关的输出端数与 K_{i+1} 级交叉开关的输入端数相等。

（5）级间连接为：K_1 级和 K_2 级的级间连接是将 K_1 级的 $a^{n-1}b$ 个输出端依序分为 $q_1=a^{n-1}$ 组，每组有 $r=a^{n-1}b/a^{n-1}=b$ 个输出端，按 q 洗牌分配连接到 K_2 级的 $a^{n-1}b$ 个输入端位置；K_{n-1} 级和 K_n 级的级间连接是将 K_{n-1} 级的 ab^{n-2} 个输出端依序分为 $q_{n-1}=ab^{n-3}$ 组，每组有 $r=ab^{n-2}/ab^{n-3}=b$ 个输出端，按 q 洗牌分配连接到 K_n 级的 ab^{n-2} 个输入端位置；K_i 级和 K_{i+1} 级的

级间连接是将 K_i 级的 $a^{n-i}b^i$ 个输出端依序分为 $q_i=a^{n-i}b^{i-1}$ 组，每组有 $r=a^{n-i}b^i/a^{n-i}b^{i-1}=b$ 个输出端，按 q 洗牌分配连接到 K_{i+1} 级的 $a^{n-i}b^i$ 个输入端位置。由 r 恒为 b 可见，实现上就是把一个交叉开关级的 b 个输出端作为一组，对这一级交叉开关的所有输出端进行 q 洗牌，分配到下一级交叉开关的输入端位置。

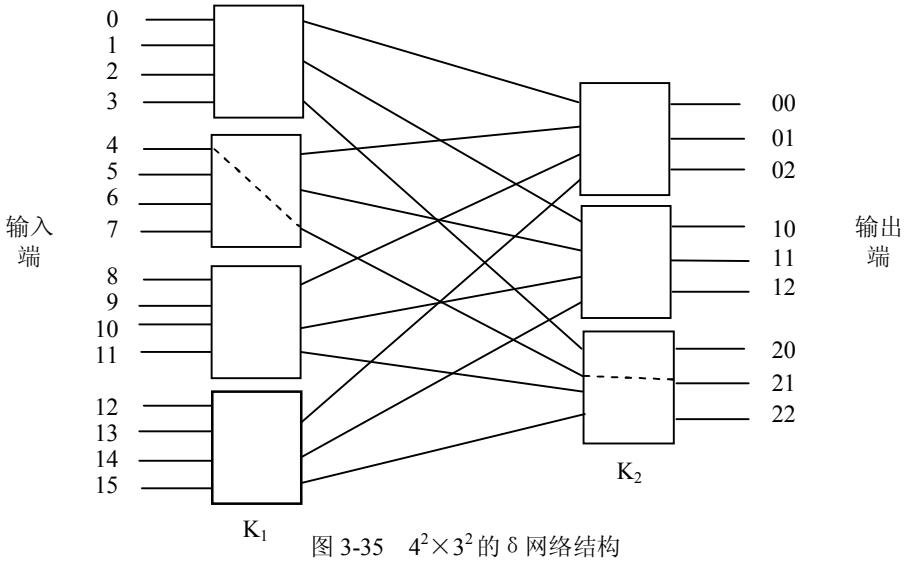


图 3-35　$4^2\times3^2$ 的 δ 网络结构

3.4.4.2　δ网络的开关控制与寻径算法

δ网络的 a×b 交叉开关采用终端标记法来确定源端与终端的连接路径，但终端地址 D 不是二进制数字，而是以 b 为基数的 b 进制数字。若终端地址表示为 $D=(d_{n-1}d_{n-2}\cdots d_1d_0)_b$，由 d_i 控制第（n–i）级上的交叉开关，1≤i≤n。例如，一个 $4^2\times3^2$ 的δ网络如图 3-32 所示。该δ网络的 n=2、a=4、b=3，采用 4×3 的交叉开关。若要从输入端 $S=(100)_2$ 连到输出端 $D=(21)_3$，则由终端标记 d_i 控制第（n–i）级的交叉开关。由 i=0，有 d_0 的“1”去控制 K_2 级的相应交叉开关；由 i=1，有 d_1 的“2”去控制 K_1 级的相应交叉开关。所以，K_1 级上的交叉开关是输入端与输出端 2 相连，K_2 级上的交叉开关是输入端与输出端 1 相连，最后到达终端$(21)_3$。

3.4.5　DM 多级动态网络

3.4.5.1　DM 网络的结构及其特点

数据变换网络（Data Manipulator，DM）是用于实现数据的排列、重复和间隔等变换的网络。一个 N×N 的数据变换网络由 n+1 级交叉开关级和 n 级级间连接构成，$n=\log_2 N$，每个交叉开关级有 N 个开关，则数据变换网络的交叉开关数为 $N(\log_2 N+1)$。N=8 的数据变换网络的结构如图 3-36 所示。数据变换网络的结构特点有：

（1）交叉开关级的级号编排从网络输出端到输入端，依次为 K_0、K_1、…、K_n。

（2）网络输入端的第 n 交叉开关级的每个开关有一个输入端和 3 个输出端，网络输出端的第 0 级交叉开关级的每个开关有 3 个输入端和一个输出端，中间各交叉开关级的每个开关都有 3 个输入端和 3 个输出端。

（3）交叉开关级之间的级间连接模式都是 PM2I 置换。即中间交叉开关级第 i 级（0<i<n）的交叉开关 j（0≤j≤N–1）的 3 个输入端分别连接前一级第 i+1 级的 $j-2^i \bmod N$、j 和 $j+2^i \bmod N$ 交叉开关，3 个输出端分别连接后一级第 i–1 级的 $j-2^{i-1} \bmod N$、j 和 $j+2^{i-1} \bmod N$ 交叉开关。

例如，第 i=2 级交叉开关 j=3 的 3 个输入端分别连接前一级第 i+1=3 级的 $j-2^i \bmod N = 3-2^2 \bmod 8=7$、j=3、$j+2^i \bmod N=3+2^2 \bmod 8=7$ 交叉开关，3 个输出端分别连接后一级第 i−1=1 级的 $j-2^{i-1} \bmod N=3-2^1 \bmod 8=1$、j=3 和 $j+2^{i-1} \bmod N=3+2^1 \bmod 8=5$ 交叉开关。第 n 级开关级的 N 个输入端就是 DM 网络的 N 个输入端，第 0 级开关的 N 个输出端就是 DM 网络的 N 个输出端。

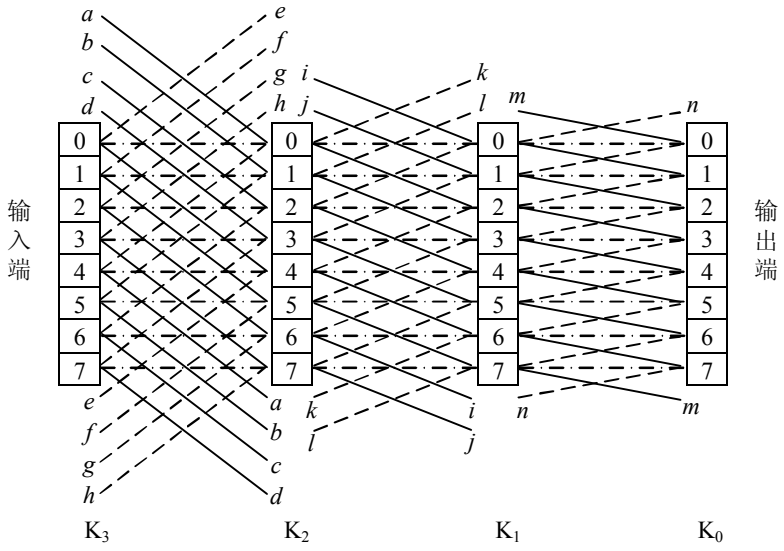


图 3-36　N=8 的数据变换网络结构

（4）数据变换网络的一个重要特点是为连接要求提供了冗余线路。一个 N=8 的数据变换网络，若要求实现由输入端 7 到输出端 2 的连接，则可以经 7→3→3→2 路径，也可以经 7→7→1→2 路径，或者经 7→3→1→2 路径等。所有的源端到终端的连接要求都可以有多条路径来实现，这有利于避免冲突，有利于提高网络可靠性，便于集成化。

3.4.5.2　DM 网络的开关控制与寻径算法

DM 网络交叉开关的控制采用的是组控方式。由于每个交叉开关有 3 个输出端可分别连接后级的 3 个交叉开关，因此，需要用 3 个控制信号来控制一个交叉开关与后级的哪一个交叉开关连接。这 3 个控制信号分别称为平控 H、下控 D 和上控 U。对开关级的交叉开关的分组为：对于第 i 交叉开关级的交叉开关，由 H_1^i、D_1^i 和 U_1^i 控制交叉开关的二进制编号第 i 位为 0 的哪些交叉开关，由 H_2^i、D_2^i 和 U_2^i 控制交叉开关的二进制编号第 i 位为 1 的哪些交叉开关。例如，第 1 交叉开关级的交叉开关，由 H_1^1、D_1^1 和 U_1^1 控制交叉开关的二进制编号第 1 位为 0 的 000、001、010、011 四个交叉开关，由 H_2^1、D_2^1 和 U_2^1 控制交叉开关的二进制编号第 1 位为 1 的 100、101、110、111 四个交叉开关。

还有一种交叉开关的控制是采用单元控制的强化数据变换网络（Augmented Data Manipulator，ADM），每一个开关都有自己的控制信号 H、D 和 U。ADM 的拓扑结构和控制方式使 ADM 完全可以模仿多级立方体网络和Ω网络的 4 功能交换开关和实现它们的连接功能。

3.4.6　基准多级动态网络

3.4.6.1　基准网络的结构及其特点

若基准网络的输入端或输出端数为 N，则网络的交叉开关级数为 $n=\log_2 N$，每级有 N/2 个交叉开关，故基准网络的交叉开关数为 $(N/2)\log_2 N$。基准网络的结构特点是：

（1）采用 2×2 的 2 功能交叉开关，两个功能为直送和交叉。

（2）各级交叉开关的级号编排是从网络输入端到输出端，依次为 K_0、K_1、…、K_{n-1}。

（3）级间连接从网络输入端到输出端依次分别表示为 C_0、C_1、…、C_n，其中，C_0 和 C_n 是恒等置换，C_1 是逆均匀选牌置换，C_2～C_{n-1} 都是子逆均匀洗牌置换。

因此，基准网络的输入端对输出端的互连函数表达式为：

$$B(n) = I_0 H_0 \sigma_1^{-1} H_1 \sigma_{(2)}^{-1} \cdots \sigma_{(n-1)}^{-1} H_{n-1} I_n$$

其中，H_i 是 K_i 级交叉开关在单元控制方式下实现的置换函数（$0 \leqslant i \leqslant n-1$），$\sigma_{(j)}^{-1}$ 是 $C_{(j)}$级级间连接模式实现的子逆均匀洗牌置换函数（$2 \leqslant j \leqslant n-1$），$\sigma_1^{-1}$ 是 C_1 级级间连接模式的逆均匀洗牌置换函数，I_0 和 I_n 是 C_0 和 C_n 级级间连接模式实现的恒等置换函数。N=8 的基准网络的结构如图 3-37 所示。

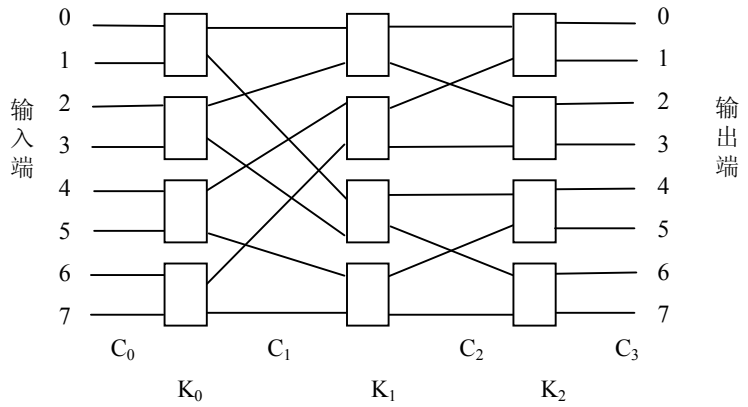


图 3-37　N=8 的基准网络结构

3.4.6.2　基准网络的开关控制与寻径算法

基准网络的交叉开关控制及其寻径算法与Ω网络相同，对交叉开关状态采用单元控制方式来获得所需要的输入端到输出端的连接路径，交叉开关单元控制采用终端标记寻径算法。

基准网络是研究多级互联网络的拓扑等价和功能等价的中间介质。一次通过基准网络可实现位序颠倒置换，二次通过基准网络可实现任意置换。

3.4.7　可重排 3 级 Clos 网络

3 级 Clos 网络的 3 级交叉开关的级号编排是从网络输入端到输出端，依次为 K_0、K_1、K_2。交叉开关级 K_0、K_1、K_2 采用的交叉开关分别为 n×m、r×r、m×n，输入级 K_0 有 r 个 n×m 交叉开关，输出级 K_2 有 r 个 m×n 交叉开关，中间级 K_1 有 m 个 r×r 交叉开关。显然，3 级 Clos 网络的输入端数与输出端数相等，输入有 N=r×n 个，输出 N=n×r 个。

级间连接从网络输入端到输出端依次分别表示为 C_0、C_1、C_2、C_3，其中 C_0 和 C_3 是恒等置换，C_1 和 C_2 都是 q 均匀洗牌置换。级间连接 C_1 是把输入级 K_0 的输出端数 r×m 分成 q=r 组（每组有 m 个输出端），与中间级 K_1 的输入端数 m×r，以 q 均匀洗牌置换实现；级间连接 C_2 是把中间级 K_1 的输出端数 m×r 分成 q=m 组（每组有 r 个输出端），与输出级 K_3 的输入端数 r×m，也以 q 均匀洗牌置换实现。

3 级 Clos 网络的一般结构如图 3-38 所示。3 级 Clos 网络需要用 m、n、r 三个参数来描述，m 是输入级所用交叉开关的输出端数或输出级所用交叉开关的输入端数，n 是输入级所用交叉

开关的输入端数或输出级所用交叉开关的输出端数，r 是中间级所用交叉开关的输入端数和输出端数。3 级 Clos 网络可记为 C(m,n,r)。当 m=n=r 时，即为 3 级可重排 Clos 网络。

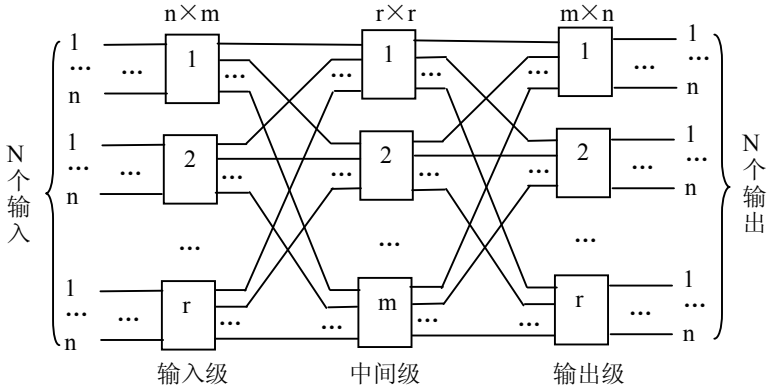


图 3-38　3 级 Clos 网络的一般结构

为讨论方便，以 m=n=r=2 的 3 级可重排 Clos 网络为例说明可重排网络的可重排原理。如图 3-39 所示，如果要实现$\begin{pmatrix}1&2&3&4\\4&2&1&3\end{pmatrix}$的置换，若先按图 3-39（a）所示的连接路径连接好 1→4 和 3→1 的路径后，就无法实现 2→2 和 4→3 的连接了。因此需要再次控制开关的状态来重新安排连接路径。可重排为如图 3-39（b）或（c）所示的状态，以实现要求的置换。在复杂的网络中，重排次数可能大于 1。

当 m≥2n–1 时，3 级 Clos 网络 C(m,n,r)就是一个非阻塞网络，例如，C(3,2,2) 就是一个非阻塞网络。C(3,2,2) 3 级 Clos 网络的每一级都有 12 个交叉点，即共要有 36 个结点开关。如果采用单级交叉开关连接，由于输入和输出结点各有 4 个，共需要 4^2=16 个交叉点，即共要有 16 个结点开关，显然要更经济一些。但当输入端数较大时，3 级 Clos 网络所需要的结点开关就会小于 N^2 个，例如，当 N=36 时，3 级 Clos 网络仅需要 1188 个结点开关，而单级交叉开关则需要 36^2=1296 个结点开关。因此，3 级 Clos 网络在 N 大时作为非阻塞网络更就有优势，另外 3 级 Clos 网络是由若干个小规模的交叉开关组成，在工程实现上也比较容易。

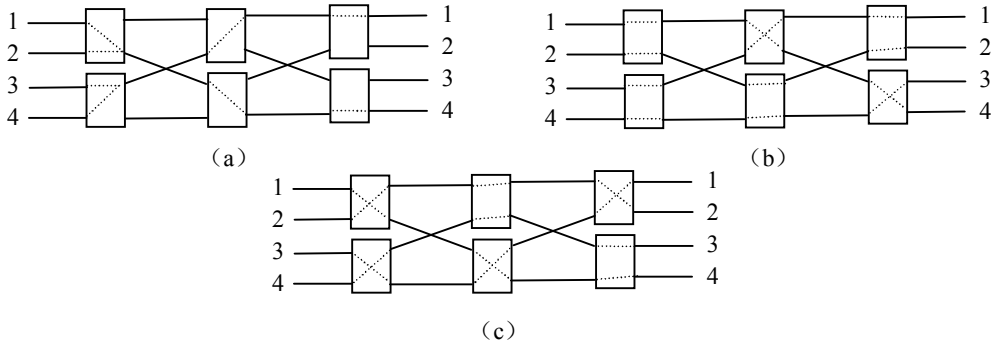


图 3-39　m=n=r=2 的 3 级可重排 Clos 网络

3.4.8　Benes 二进制置换网络

3.4.8.1　Benes 网络的结构及其特点

若 Benes 二进制置换网络的输入端或输出端数为 N，则交叉开关级数为 $n=2\log_2 N-1$，每

级有 N /2 个交叉开关，故 Benes 二进制置换网络的交叉开关数为 $N/(2\log_2 N-1)$。Benes 二进制置换网络的结构特点是：

（1）采用 2×2 的 2 功能交叉开关，两个功能为直送和交叉。

（2）各级交叉开关的级号编排是从网络输入端到输出端，依次为 K_0、K_1、…、K_{n-1}。

（3）级间连接从网络输入端到输出端依次分别表示为 C_0、C_1、…、C_n，它是将两个基准网络进行背对背互连，就可构成一个基本 Benes 二进制置换网络。

因此，Benes 二进制置换网络的输入端对输出端的互连函数表达式为：

$$\beta(n)=I_0H_0\sigma_1^{-1}H_1\sigma_{(2)}^{-1}\cdots\sigma_{(n-2)}^{-1}H_{n-2}\sigma_{n-1}^{-1}H_{n-1}I_n$$

其中，H_i 是 K_i 级交叉开关在单元控制方式下实现的置换函数（$0\leqslant i\leqslant n-1$），$\sigma_{(j)}^{-1}$ 是 $C_{(j)}$ 级级间连接模式实现的子逆均匀洗牌置换函数（$2\leqslant j\leqslant n-2$），$\sigma_1^{-1}$ 和 σ_{n-1}^{-1} 是 C_1 和 C_n 级级间连接模式的逆均匀洗牌置换函数，I_0 和 I_n 是 C_0 和 C_n 级级间连接模式实现的恒等置换函数。N=8 的 Benes 二进制置换网络结构如图 3-40 所示（K_2 与 K_3 为同一级）。

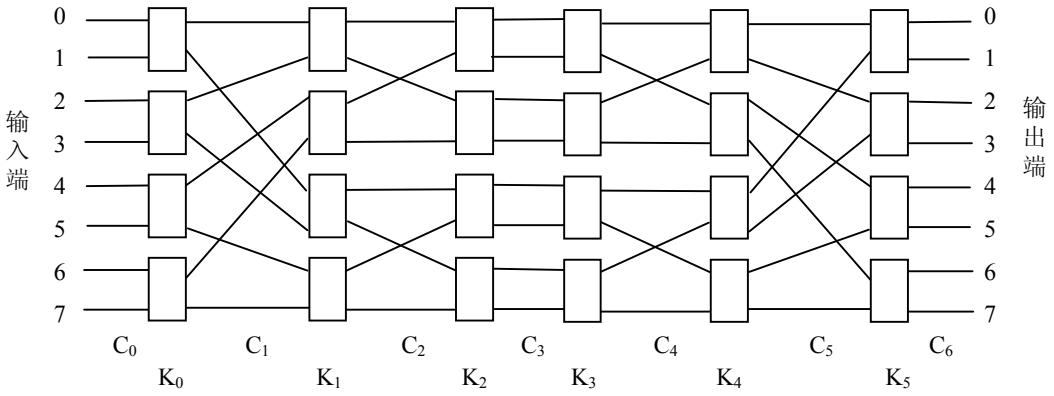


图 3-40　N=8 的 Benes 二进制置换网络结构

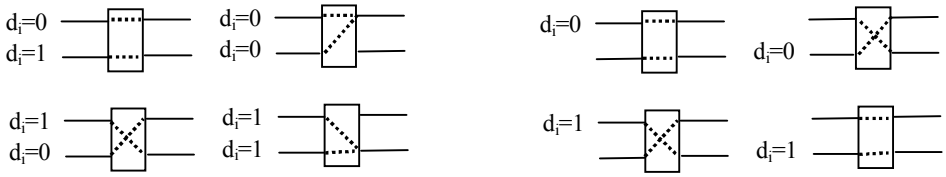
3.4.8.2　Benes 网络的开关控制与寻径算法

Benes 二进制置换网络对交叉开关状态采用单元控制方式来获得所需要的输入端到输出端的连接路径，交叉开关单元控制采用算法有：循环控制法、改进的终端标记法、方阵分解法、并行设置法、递归分治法及 a×b 交换开关的一般控制等。现介绍改进的终端标记算法。

改进的终端标记法是对在Ω网络中介绍的终端标记法的改进。两种方法的不同之处如图 3-41 所示。终端标记法规定第 i 级交叉开关的状态用终端地址 D 的第 i 位 d_i 来控制。若 $d_i=0$，则相应交叉开关的输入端连接该交叉开关的上输出端；若 $d_i=1$，则相应交叉开关的输入端连接该交叉开关的下输出端。如果同一个交叉开关的 2 个输入端由终端地址 D 的 d_i 位的值提出的连接要求，一个输入端是 $d_i=1$，另一个输入端是 $d_i=0$，那么，这个交叉开关不会发生连接冲突。如果 2 个输入端的 d_i 都是 0，那么，这个交叉开关会发生 2 个输入端同时要求连接上输出端的连接冲突。如果 2 个输入端的 d_i 都是 1，那么，这个交叉开关会发生 2 个输入端同时要求连接下输出端的连接冲突。如图 3-41（a）所示。

改进的终端标记法只用一个输入端的控制信号 d_i 来控制交叉开关状态，另一个输入端的连接要求服从此状态。可以用交叉开关的上输入端的 d_i 来控制（上控法），也可以用交叉开关的下输入端的 d_i 来控制（下控法），但是，在一次置换中只能用一种控制方案。在改进的终端标记控制算法中规定：若用交叉开关的上输入端的 d_i 来控制，则当 $d_i=0$ 时，交叉开关是直送

状态；当 d_i=1 时，交叉开关是交叉状态。若用交叉开关的下输入端的 d_i 来控制，则当 d_i=0 时，交叉开关是交叉状态；当 d_i=1 时，交叉开关是直送状态，如图 3-41（b）所示。改进的终端标记法虽然使连接的灵活性受到限制，但确有效地防止了交叉开关中冲突的发生，而连接灵活性的限制则通过交叉开关级来弥补。Benes 二进制置换网络的交叉开关级增加了一倍，某一种连接中一个交叉开关被动连接的那个输入端经过级间互连的置换，通常还会有不少于 n–1 次作为主动控制的机会，从而能够准确地连通到指定的输出端。



（a）终端标记法　　（b）改进的终端标记法

图 3-41　2×2 的 2 功能交叉开关控制的比较

另外，Benes 二进制置换网络交叉开关级数是终端二进制数地址编号的一倍，因此终端二进制数地址编号中的一位 d_i 应控制 Benes 二进制置换网络中 K_i 和 K_{n-1-i} 级上的两个交叉开关。

例如，用改进的终端标记法在 Benes 网络上实现位序颠倒置换，$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 4 & 2 & 6 & 1 & 5 & 3 & 7 \end{pmatrix}$

如图 3-42 所示。若网络输入端 3 要连接输出端 6，即源端地址 S=011 由位序颠倒置换连接到终端地址 D=110，采用改进的终端标记控制算法中的上输入端 d_i 控制。那么，由 K_0 级的交叉开关 2 的上输入端 d_0=0，使交叉开关 2 为直送状态，再由级间连接 σ^{-1} 置换到 K_1 级的交叉开关 7。交叉开关 7 的上输入端 d_1=0（因为交叉开关 7 的上输入端网络输入端 1 到输出端 4 的连接，而 4 的二进制数是 100，即 d_1=0），使交叉开关 7 为直送状态。依次类推到 K_2 级的交叉开关 12 为交叉状态，K_3 级的交叉开关 16 为直送状态，K_4 级的交叉开关 20 为直送状态和 K_5 级的交叉开关 24 为交叉状态，从而完成从网络输入端 3 到输出端 6 的连接。

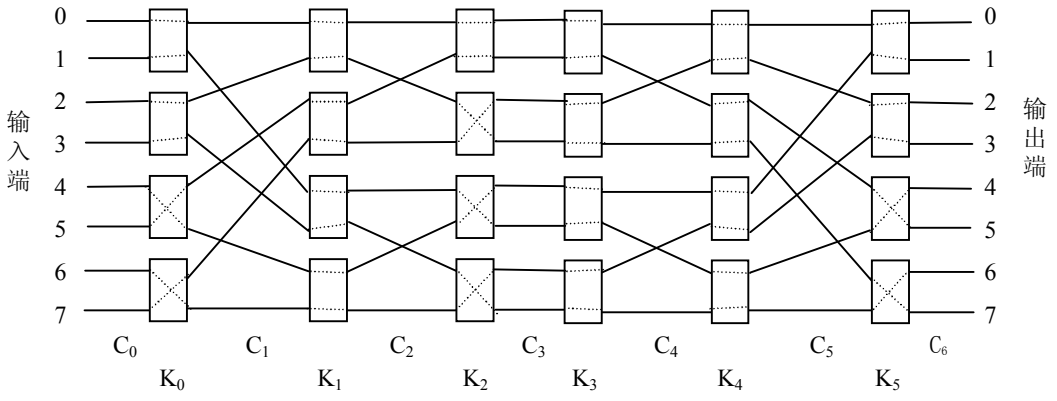


图 3-42　改进的终端标记法实现位序颠倒置换（上控法）

3.4.8.3　Benes 网络可实现的互连函数

Benes 二进制置换网络至少有两个以上的连接来满足同一结点对的互连要求，以避免可能

发生的阻塞冲突。Benes 二进制置换网络不但可以满足输入输出对间所有可能的 N!置换连接，还有多余的通路冗余量。因此，Benes 二进制置换网络属于可重排非阻塞网络，当发生阻塞冲突时，可以通过重新设置交叉开关的状态来加以避免。

特别地，N 个输入端与 N 个输出端之间连接的可能排列有 N!种，多级交叉开关动态互联网络要实现无阻塞，其所包含的交叉开关状态也应至少要有 N!种组合。但当多级交叉开关动态互联网络采用 2×2 的交叉开关时，在输入与输出之间交叉开关的级数一般只有 $\log_2 N$，每一级交叉开关数为 N/2，交叉开关的总数为 $(N/2)\log_2 N$，那么交叉开关状态只有 $2^{\left(\frac{N}{2}\times\log_2 N\right)}=(\sqrt{N})^n$ 种组合，小于 N!，必然有一些连接会被阻塞。多级交叉开关动态互联网络要实现无阻塞，就需要增加交叉开关的总数。

3.5　互联网络的消息传递

在多处理机系统中通过互联网络进行消息传递时需要专门的硬件和软件支持，互联网络的拓扑结构的选择与系统策略有关，不同拓扑结构的寻径策略是不一样的。例如，拓扑结构在很大程度上决定了各类不同寻径方案的可用性及是否可以采用自适应寻径。

3.5.1　消息传递的格式与方式

3.5.1.1　消息传递的格式

消息是结点间数据交换或通信的逻辑单位，但它所包含的数据量或消息的长度是可变的。消息传递的典型格式是将消息分组，每一组称为一个消息包。显然，消息是由任意数目的长度固定的消息包组成，如图 3-43 所示。

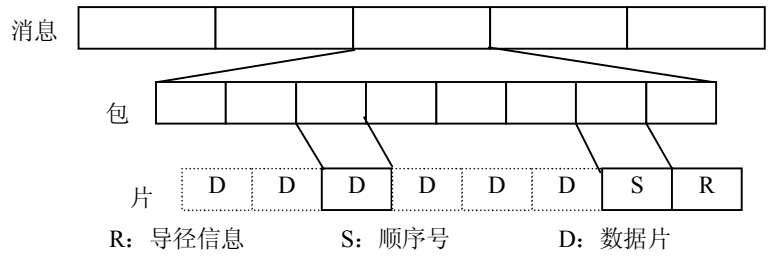


图 3-43　消息传递的信息单位：消息、包和片的格式

消息包是包含寻径目的地址的基本单位，是消息传送的最小单位。由于不同的消息包可能异步地达到目的结点，因此每个包需要一个序号，以便把传送的消息重新装配起来。另外。可以进一步把消息包分成长度固定的信息片，传递信息的目的地址和序号形成报头片，其余的信息片是数据片。信息包的长度取决于传递方式和网络的实现方法，典型信息包的长度为 64～512 位。包和片的大小还与通道频宽、寻径器设计以及网络密度等有关。

3.5.1.2　消息传递的方式

消息传递方式可以分为两大类：线路交换和包交换，其中包交换又包括存储转发、虚拟直通和虫蚀 3 种。

1. 线路交换（Circuit Switch）传递

线路交换传递是指在传送一个消息之前，先建立一条从源结点到目的结点的物理通路，

然后再传送消息，如图 3-44 所示。因此，线路交换传递方式需要提前预订整个路径及其所需要的开关端口，路径和端口一旦预订成功，消息包就可全速地由源结点流向目的结点。线路交换传递的传输时延包括路径建立时间和数据交换时间，即有：

$$T_{CS}=(L_t\times D + L)/B$$

式中：L_t 为建立路径所需的小信息包的长度，L 为信息包的长度，D 为源和目的之间的距离（经过的中间结点数），B 为线路频宽。

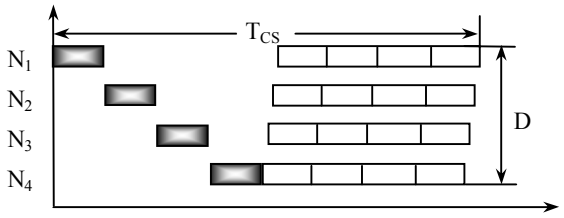


图 3-44　线路交换传递的时空图

线路交换方式的优点是在包传递过程中，能实现无竞争、无干扰地全速传递。其缺点是需要提前预留资源，使用效率低。在多处理机中，往往需要频繁地传递小信息包，则需要频繁地建立源结点到目的结点的物理通路，导致开销很大。

2. 存储转发（Store and Forward）传递

存储转发传递是指在网络中，当一个包到达一个中间结点时，先被存入结点的包缓冲区，当所要求的输出通道和接收结点的包缓冲区可使用时，再将它传送给下一个结点，如图 3-45 所示。显然，包是信息流的基本单位，每个结点有一个包缓冲区，包从源结点经过一系列中间结点到达目的结点。存储转发传递的传输时延是每个存储转发结点所花费时间的和，即有：

$$T_{CS}= (D + 1)\times(T_d + L/B)$$

式中：T_d 为每个存储转发结点因处理和排队等待所造成的时延。若 $T_d=0$，则最小的存储转发传递的的传输时延为：

$$T_{CS}= (D + 1)\times L/B$$

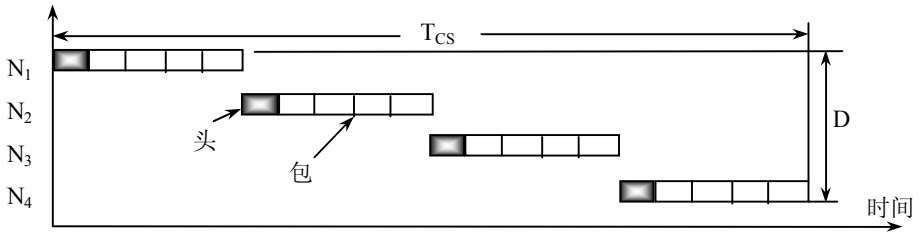


图 3-45　存储转发传递的时空图

存储转发传递不需要提前预留资源，链路的使用效率高。其缺点是传输时延与源和目的之间的距离成正比，传输时延大；另外，为避免多个消息包向同一个结点传递时造成包丢失，每个结点需要较大的包缓冲区。

3. 虚拟直通（Virtual Cut Through）传递

虚拟直通传递方式是为减少存储转发传递的时延，没有必要等到整个消息包全部到达缓冲后再作路由选择，只要包含路由选择的头片到达后即可判断，如图 3-46 所示。虚拟直通传递的传输时延也是头片在每个存储转发结点所花费时间总时间与数据交换时间的和，即有：

$$T = L/B+(L_h/B+T_d)\times(D+1)$$

式中：L_h 是消息包中消息头片的长度。一般有 $L>>L_h\times(D+1)$，则最小的虚拟直通传递的传输时延为：

$$T = L/B + T_d(D+1)$$

可以看出：在 $T_d=0$ 时，传输时延与结点数无关。

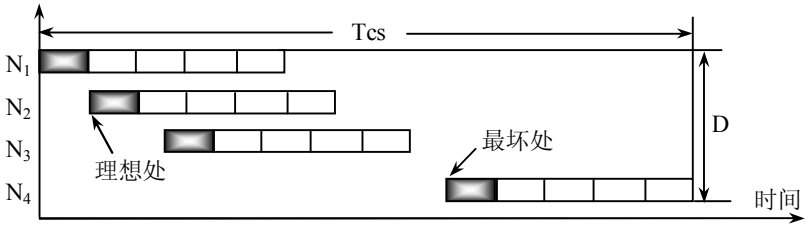


图 3-46　虚拟直通传递的时空图

虚拟直通传递的前提是链路通畅不阻塞，最理想的情况是每个结点都如同结点 1 处，这时传输时延最小。当出现结点阻塞时，虚拟直通传递的数据片也需要存储，因此，虚拟直通传递时，每个结点仍需要包缓冲区。最坏的情况是每个结点都如同结点 3 处，这时传输时延最大，与存储转发传递一样。

目前有一些计算机系统采用虚拟直通传递方式。

4. 虫蚀（Wormhole）传递

虫蚀传递是把消息包进一步分成更小的片，与结点相连的硬件寻径器中有片缓冲区，消息从源结点传送到目的结点要经过一系列寻径器，同一个消息包中所有的片以流水方式在各寻径器中顺序地传送，如图 5-47 所示。由于只有消息包中的头片含有目的地址，用头片直接建立一条从源结点到目的结点的路径，则所有数据片必须紧跟头片。因此，不同的消息包可以交替地传送，但不同消息包的片不能交叉，否则它们可能被送到错误的目的地。

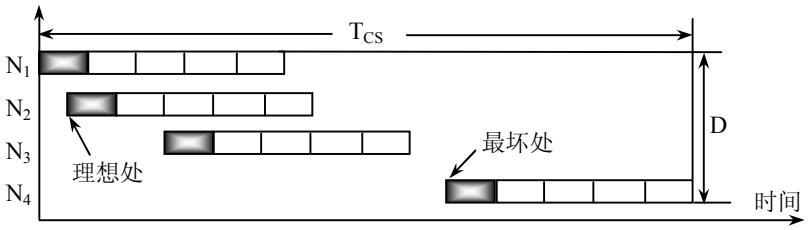


图 3-47　虫蚀传递的时空图

用消息包的头片开辟出一条从输入链路到输出链路的路径，消息包中的片按头片开辟的路径以流水方式在网络中向前“蠕动”。整个消息包就如同一条蠕虫，每个片相当于虫的一个“节”，“蠕动”是以“节”为单位顺序地向前爬行的。当消息包的头片到达一个 A 结点后，A 结点寻径器根据头片的传递消息立即作出路由选择。如果所选择的通道空闲且所选择的结点 B 的片缓冲区可用，那么这个头片就不必等待，直接通过结点 A 传向下一个结点 B，随后的其他数据片会跟着相应地向前“蠕动”一步。当消息的尾片向前“蠕动”一步之后，它刚才所占有的结点就被放弃了。如果所选择的通道忙或结点的片缓冲区不可用，那么头片就必须在该结点的片缓冲区中等待，直到两者都可用为止，其他数据片也在原来的结点上等待。此时，被阻塞的消息不从网络中移去，片也不放弃它占有的结点和链路。虫蚀传递的传输时延为：

$$T = L/B+L_h(D + 1)/B+T_d$$

一般有 $L>>L_h\times(D+1)$，则最小的虫蚀传递的传输时延为：

$$T=L/B+T_d$$

可以看出：T_d 是否为 0，传输时延都与结点数无关。

虫蚀传递的优点是：一是各结点不需要大的包缓冲区，只需要很小的片缓冲区；二是消息包中的片以流水方式传送，利用时间并行性减少每个结点因处理和排队等待所造成的时延，传输距离对传输时延影响很小，在最理想的情况下虚拟直通传递与虫蚀传递的传输时延虽然相等为 L/B，但虫蚀传递与实际更为接近；三是链路的共享性好、利用率高，新链路建立与旧链路的释放是同时进行的，即一旦建立了一条新的链路，另一条旧的链路就释放；四是允许寻径器复制消息包的片并从多个输出链路输出，容易实现选播与广播。因此，新型的多计算机系统很多采用虫蚀传递方式。

虫蚀传递的缺点是当消息包的一个片被阻塞在某一结点时，整个消息包的所有片都将被阻塞，而且要占用结点资源。

3.5.2 路由选择及其方法

3.5.2.1 什么是路由选择

互联网络的基本功能就是要在多处理机或多计算机或功能部件的各个结点间实现高效率的通信，当结点间没有直接的链路相连接时，信息就要通过中间结点进行传递。这时可能存在多条路径，为了充分利用互联网络的可用带宽，尽量减少传送延迟和避免死锁，就要选择一条合适的路径。

路由选择即是通路选择或路径选择，它是指用来实现选择经中间结点传递信息功能的通信方法或算法，有时简称寻径。路由选择的基本操作就是监控输入端口进来的信息包，并为每个信息包选择一个输出端口

3.5.2.2 路由选择的方法

在互联网络中的两个结点之间往往有多条物理通路，当两结点要通信时，希望通过的路径最短最合理地满足互联网络要求。另外，也可能发生几对结点间传递信息时都要通过某一个或某几个中间结点的情况，即路径选择冲突。这就是路由选择要解决的问题，解决的方法有两种：确定性方法和自适应方法。

1. 确定性方法

确定性方法是指路由完全由源地址和目的地址决定，即一对源地址和目的地址只有一条通路可选。显然这条通路的路径最短但不一定最合理地满足互联网络要求。该方法的算法简单、实现方便，但当发生冲突或路径有故障时，无法改变通路。对于系统内的互联网络，由于每隔几个时钟周期，交叉开关就要为所有输入的信息包进行选路，因此，路由选择算法要尽可能简单和快速。自适应路由选择的算法很复杂，在系统内的互联网络中通常不使用。目前，交叉开关一般采用的是确定性方法。确定性方法主要有 3 种，即算术选路法、源选路法和查表选路法。对于确定性选路，无论选路路径上是否有链路出现阻塞，消息包都将沿着确定的选择的路径进行传输。

算术选路法是指如果所有消息的选路路径由消息的源地址和目的地址完全确定，与网络当前负载情况无关。例如维序选路就是算术选路法的最短选路法。

源选路法是指源结点为消息建立一个头部，其中包含选路路径上经过的所有交叉开关的

输出端口，消息路径的各个交叉开关简单地从消息包头中取出端口号并将消息传递到相应的通道。源选路法可以采用相对简单的交叉开关设计（更少的控制状态），也不需要采用复杂的算术单元来实现对任意网络拓扑的支持，因此，具有通用性。缺点是可能使得消息包头过大，而且长度不固定。源选路法用于 MIT Parc 和 Arctic 路由器、Meiko CS-2 以及 Myrinet 上。

查表选路法是指每个交叉开关维护一张选路表 R，而消息包的头部包含一个选路域 I，以 I 为索引查选路表，就可以得到输出端口 O=R[I]。查表选路法的缺点是选路表可能很大，要求结点间的选路路径相对稳定，具有很好的通用性，适用于局域网和广域网的选路，但对于选路路径相对灵活的并行计算机网络来说并不合适。查表选路法用于 HiPPI 和 ATM 交换开关中。

2. 自适应方法

自适应方法是指路由通路每次都要根据资源和网络状态来选择。显然该方法可以避开拥挤的或有故障的结点，使网络的利用率得到改进、吞吐率得到很大提高。同一拓扑结构的互联网络有很多自适应路由选择实施方案。

自适应寻径的链路选择是由结点上的寻径器根据寻径中碰到的流量动态决定的。如果所希望的输出端口之一被阻塞或失效，寻径器可以选择一个替代链路送出数据包。最小自适应寻径仅沿着到达目的结点的最短路径引导数据包，每一次寻径都必须缩短到达目的结点的距离。允许使用所有最短路径的自适应算法称为完全自适应算法，否则就是部分自适应的。非最小自适应寻径的一种做法是：寻径器从不缓冲数据包，如果一个以上的数据包指向同一个输出链路，寻径器只将其中的一个包送往其目的结点，而将其他的包传送到别的链路，而不管经过这些链路传送后是否会离目的结点更近。

如果选路算法只允许选择一条路径，一方面单个连接的失效就会使网络断开，导致消息传送的失败；另一方面将给网络带来大量竞争，多对结点对会使用相同的链路，多个通信只能顺序进行。自适应选路算法是放宽对选路算法的限制，采用多条路径的选路，允许结点对之间存在多条合法路径，一方面当某条链路失效时，可以绕开故障点进行消息传送，提高容错能力；另一方面可以在可用链路上更广泛地分布流量，将网络负载可以分布到多个通道上，从而提高网络的利用率。例如若有 4 个数据包从不同的源结点向不同的目的结点传送，当采用确定性寻径方法时，都被强迫沿同一路径前进，就可能会在某一路径上产生通信的瓶颈，而其他最短路径上的链路却闲置未用。而自适应寻径方法则可以根据情况使用其他的链路。但由于自适应选路灵活且具有动态性，因此容易产生死锁。

自适应选路可以与算术选路法、源选路法和查表选路法 3 种选路方法结合而形成多种方式的自适应寻径。与源选路法结合则有对基于源结点的寻径，源结点可以简单地在多条合法路径中挑选传送路径，并根据选择建立信息包的头片，无需改变交叉开关设计。与查表选路法结合则有基于表驱动的寻径，可以通过为多条路径建立寻径表项来完成，通过在查找寻径表可以了解链路的情况，自动选择合适的链路来传送消息。与算术选路法结合则有基于运算的寻径，需要给数据包头附加一些额外的控制信息，并由寻径器解释这些信息，以选择最佳的传送路径。

自适应选路并没有被当前的并行计算机广泛使用。Cray T3E 在立方体上实现了最短自适应选路，Ncube/3 则在超立方体上实现最短自适应选路。自适应选路的缺点是增加了交叉开关的复杂性，因此也会降低它的速度。在网络负载的饱和点附近，简单的确定选路由于对带宽要求低，性能也会比自适应选路算法性能好。

3.5.2.3　寻径效率

一般来说，描述网络寻径效率常用的两个参数是通道流量和通信时延。网络通道流量可

用传输有关消息所使用的通道（链路）数来表示，它反映了网络中的通信负载情况。网络通信时延用消息包的最长传输时间来表示，它反映了网络中消息的传输速度。寻径效率不但与网络中的物理链路、缓冲器和寻径器等通信资源有关，还与网络的拥塞和死锁问题密切相关。

设计与优化的寻径网络应能以最小流量和最小时延实现有关通信模式，但这两个参数并不是毫不相关的。达到最小流量的同时不一定能达到最小时延，相反的情况也是如此。因此，它与大多数目标优化的算法类似，它只能是在这两个优化目标之间折衷。而且，在使用不同的交换技术时，其优化的侧重点也不尽相同。例如，在存储转发互联网络中，时延优化是最重要的问题；而在虫蚀互联网络中，流量优化对通信效率的影响则具有重要的意义。

3.5.3 算术选路算法

算术选路法在系统内的互联网络中应用最为广泛，它既不会像源选路法那样使信息包的包头很大，降低信息包的有效传输率；也不需要像查表选路法那样占用大量结点资源——缓冲区，在互联网络规模很大时，查表速度慢。

对于系统内的互联网络，拓扑结构一般都是规则的，算术选路法就可满足路由选择的需要。其中最典型的是维序路由选择算法，即根据通信链路的坐标维来决定信息包如何流过相继的链路。维序路由选择算法用于二维网格网络时就称为 X-Y 选路算法，用于超立方体网络时就称为 E——立方选路算法。

3.5.3.1 X-Y 寻径算法

X-Y 寻径算法主要用于二维互联网络，是指先沿 X 维方向确定路径，后沿 Y 维方向确定路径。假设信息包从任一源结点 S=(X_1,Y_1)到任一目的结点 D=(X_2,Y_2)，则一般是从 S 开始，先沿 X 方向前进直到 D 所在的第 X_2 列为止，后沿 Y 方向前进到 D。采用 X–Y 寻径算法寻径不会出现死锁现象。

X-Y 寻径算法包含 4 种寻径模式，分别与东—北、东—南、西—北、西—南的路径相对应。如图 3-48 所示为在 16 个结点的二维网格网络进行 X-Y 寻径，图中有 4 对“源—目”的结点对，分别对应 4 种寻径模式。从结点(0,0)到结点(2,1)需要一条东—北路径，从结点(0,3)到结点(1,2)需要一条东—南路径，从结点(3,3)到结点(2,2)需要一条西—南路径，从结点(1,1)到结点(0,2)需要一条西—北路径。

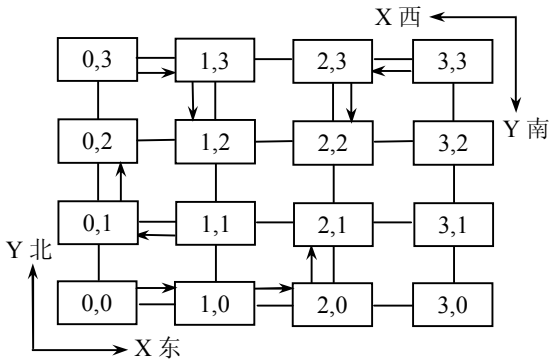


图 3-48 16 个结点的二维网格网络的 X-Y 寻径

X-Y 寻径算法可以在源和目的结点之间建立一条距离最短的路径，但有时为了减少网络流量和避免死锁，只能通过该算法得到非最短路径。由于 X-Y 寻径算法不会产生死锁，因此

可用于存储转发和虫蚀寻径网络，但对于环形网络，采用该算法则得不到最短路径。

3.5.3.2　E-立方寻径算法

E-立方寻径算法主要用于超立方体互联网络。假设有一个 $N=2^n$ 个结点的 n 方体，源结点的地址为 $S=S_{n-1}\cdots S_1S_0$，目的结点的地址为 $D=D_{n-1}\cdots D_1D_0$，$V=V_{n-1}\cdots V_1V_0$ 是路径中的任一点的地址。将 n 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，选择一条 S 到 D 最短路径的算法为：

（1）计算方向位 $r_i= S_{i-1} \oplus D_{i-1}$，其中 i=1,2,…,n。

（2）i=1，V=S。

（3）如果 $r_i=1$，则从当前结点 V 寻径到下一结点 $V=V \oplus 2^{i-1}$；如果 $r_i=0$，则跳过这一步。

（4）i←i+1，如果 i≤n，则转第三步，否则退出。

E-立方寻径算法可以在源和目的结点之间建立多条距离最短的路径。在如图 3-49 所示的 3-立方体中，当结点 011 向结点 110 发送消息时，一种寻径方案是首先沿第 0 维方向把消息送至结点 010，再沿第 2 维方向把消息送至结点 110；而另一种寻径方案是先沿第 2 维方向把消息送到结点 111，再沿第 0 维方向把消息送到结点 110。以上两条路径都是最短路径，判别的依据是路径所经过的最少链路数量与两结点间的海明距离相同，都为 2。假设源结点与目的结点间的海明距离为 h，则最短路径有 h！条。例如，结点 000 与结点 111 之间的海明距离为 3，则它们之间共有 3×2×1 条路径，分别是 000→001→011→111，000→001→101→111，000→010→011→111，000→010→110→111，000→100→101→111 和 000→100→110→111。

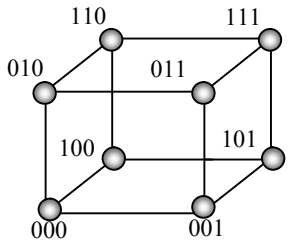


图 3-49　3-立方体网络的 E-立方寻径

除了最短路径之外，还有多条非最短路径。这样一来，n-立方体网络的可选路径就会有很多，因此它的可靠性是比较高的，若某个或某些结点出现故障，剩下的结点仍可以完成源-目的结点间的通信。对于其他拓扑结构的网络，有些也可能蕴含在 n-立方体中，因此也可以利用 E-立方体寻径算法进行消息寻径。

3.5.4　虚拟通道

虫蚀传递方式中的通信链路实际上由许多源与目的对共享，从共享物理通道可以引出虚拟通道的概念。虚拟通道（Virtual Channel）是指两个结点间的逻辑链，它由源结点的片缓冲区、结点间的物理通道以及接收点的片缓冲区组成。如图 3-50 所示是 4 条虚拟通道共享一条物理通道。源结点和目的结点各有 4 个片缓冲区，当物理通道分配给某对缓冲区时，这对源缓冲区和目的缓冲区便形成了一条虚拟通道。源缓冲区存放等待使用通道的片，目的缓冲区存放由通道刚刚传过来的片，而物理通道（电缆或光纤）是它们之间的通信媒介。

显然，物理通道由所有的虚拟通道分时共享。虚拟通道除有关的片缓冲区和通道外，还需要用某些通道状态来表示不同的虚拟通道。虚拟通道可能会使每个请求可用的有效通道频宽

降低，确定虚拟通道数目时，需要综合考虑吞吐量和通信时延。实现数目很大的虚拟通道需要用高速的多路选择开关。

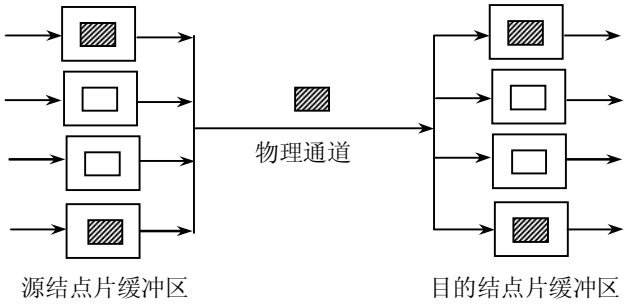


图 3-50　4 条虚拟通道以片传递为基础分时地共享一条物理通道

虚拟通道有单向或双向之分，两条单向通道组合在一起可以构成一条双向通道。双向通道不仅可以增加利用率，还可以使通道的频宽加倍，但双向通道的仲裁要复杂，因而增加了延迟和成本。

3.5.5　死锁

3.5.5.1　死锁及其类型

广义上的死锁是指消息包等待一个不可能发生的事件，例如，当互联网络中所有结点的消息队列已满，每条消息都在等待其他消息释放资源，那么就没有消息可以到达目的地。

根据死锁是否可以解除来分，可分为无限延期及活锁。无限延期是指发生在消息包等待一个可能出现但永不会发生的事件，主要存在一个公平性问题，它是不可能解除的；而活锁是指发生在消息包网络传输中却无法达到目的地，活锁只会出现在自适应的非最短路径选路算法中，它是可以解除的。活锁又有迎面死锁和选路死锁之分，通常所说的死锁指的是活锁。

在两个结点之间，当它们都试图向对方发送消息包，并且在收到对方发出的消息包之前就进行发送时，就可能发生迎面死锁（Hend on）。在使用同步发送和接收的应用程序中，迎面死锁是常见的。另外如果网络中采用半双工的通道或者交叉开关控制器无法在双向通道上同时发送和接收消息包，迎面死锁也可能会出现。

当互联网络中多个消息竞争系统资源时就可能会发生选路死锁（Routing Deadlock）。如图 3-51 所示是若干条消息在网络中传输，每条消息由一些数据片构成。而每条通道可看作与一定数量的系统缓冲区资源（包括通道目的的交叉开关的输入缓冲区、通道源交叉开关的输出缓冲区）相关。在如图 3-51 所示中，每个消息包都将左转，但 4 个通道相关的缓冲区都已被占用。在申请到新的缓冲区之前，每条消息都不会释放已经占用的缓冲区。也就是互联网络中的 4 个交叉开关，每个交叉开关有 4 个输入端口和 4 个输出端口。四条消息包每个都占用了一交叉开关的一个输入端口、一个输出端口和另外一个交叉开关上的一个输入端口，因此每个消息包在左转时都再需要一个输出端口。如果没有信包愿意释放它的输出端口，那么就进入了死锁。

显然，无论活锁是如何产生的，解除的办法就是允许结点在无法发送时仍然可以接收消息。一个可靠的互联网络要免于活锁，就要求结点即使无法发送消息包，也应能将无用的消息从结点中剔除。

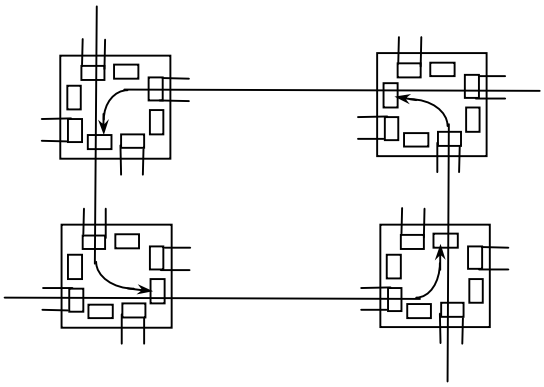
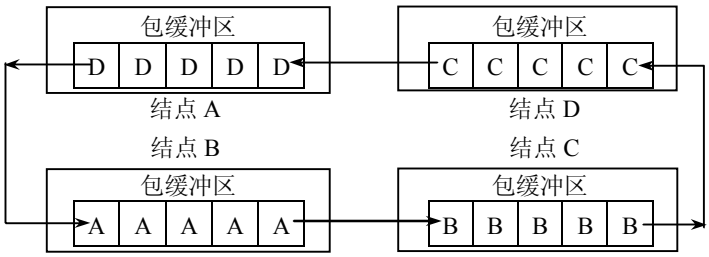


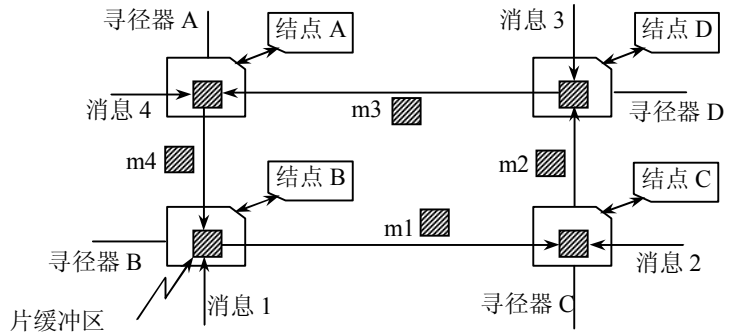
图 3-51　4 个交叉开关互联网络的选路死锁

3.5.5.2　死锁产生原因

死锁是由于在缓冲区或通道上的循环等待而产生的，如图 3-52 所示。在图 3-52（a）所示的存储转发网络中 4 个消息包分别占用了 4 个结点的 4 个缓冲区，每个消息包又都在等待其他消息包释放资源，那么就没有消息包可以到达目的地，导致循环等待而出现缓冲区死锁的情况。除非抛弃某个消息包，否则死锁不会解除。在图 3-52（b）所示的采用虫蚀传递的网络中，4 个消息沿 4 条通道同时传送，4 个消息的 4 个片同时占用了 4 条通道而产生通道死锁。如果循环中没有一条通道被释放，则死锁状态将持续下去。



（a）采用存储转发寻径的 4 个结点之间出现缓冲区选路死锁



（b）采用虫蚀寻径的 4 个结点之间出现通道死锁（带阴影方块是片缓冲区）

图 3-52　缓冲区或通道上的循环等待引起的死锁

显然，无论采用存储转发还是虫蚀传递方式，都可能会发生死锁。然而，由于虫蚀传递将一个消息包分解成许多数据片序列分布到多个数据片缓冲区中，所以造成死锁的概率要大一些。

3.5.4.3　死锁的解除与避免技术

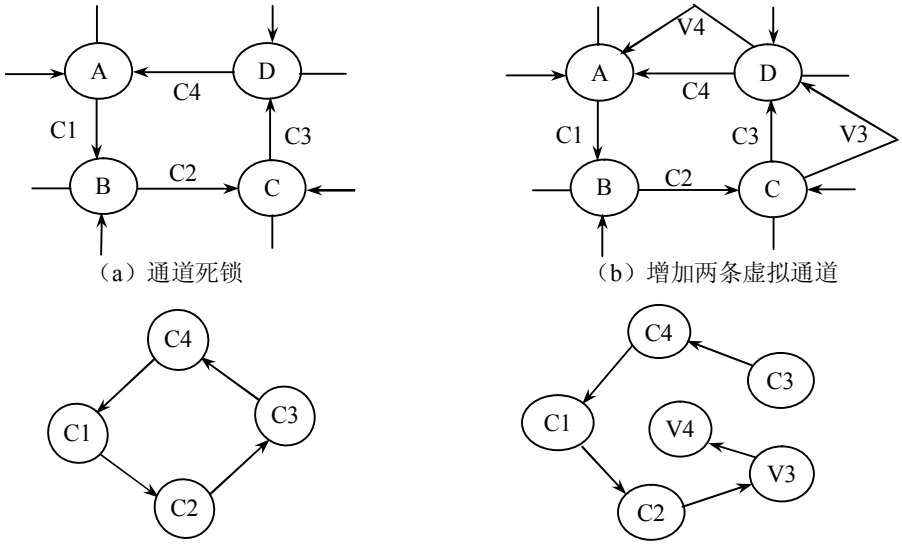
由于死锁是由循环等待引起的，其原因是由于消息在互联网络通道移动时造成资源共享

的需求大于现有资源。例如，在迎面死锁时两结点之间都要向对方发送消息，但都必须在得到对方发出的消息之前进行发送，由于传输的物理通道共享，两个结点之间抢占物理链路而导致死锁。所以避免死锁的基本要求是通道相关图上不出现圈，实现的简单算法是为每个通道资源分配一个数字（资源号），在进行分配时，按照通道资源号递增（或递减）的顺序进行分配。

死锁的解除与避免技术主要有虚拟通道和转弯选路。

1. 虚拟通道技术

利用虚拟通道，增加资源破坏现有资源循环来解除死锁，如图 3-53 所示是通道的相关图。在通道相关图中，结点表示通道，带方向的箭头用来表示通道之间的相关依赖关系。



（a）通道死锁　（b）增加两条虚拟通道

（c）通道死锁的通道相关图　（d）利用虚拟通道后的通道相关图

图 3-53　利用虚拟通道解除死锁

利用虚拟通道解除死锁的基本思想是允许结点在无法发送时仍可接收消息。如图 3-53（a）所示出现循环的通道相关而产生死锁，图 3-53（c）为相应的通道相关图。于是可增加两条虚拟通道 V_3、V_4，如图 3-53（b）所示。利用虚拟通道将通道相关循环链变成螺旋线来解除死锁，图 3-53（d）为相应的通道相关图。

虚拟通道在虫蚀选路网络中避免死锁的实现方法是为每个物理通道提供多个缓冲区，并将缓冲区劈开，构成一组虚拟通道，如图 3-54 所示。虚拟通道并不要求增加网络中的物理连接和开关的数目，它需要在交叉开关中添加更多的选择器和多路（复用）器，以允许多个虚拟通道共享物理通道。虚拟通道使实现自适应寻径更加经济和灵活。

例如在网格网络中，同一维的所有链接都可以使用虚拟通道，在此基础上构建虚拟网络来避免死锁，如图 3-55 和图 3-56 所示。

图 3-55（a）是一个采用 X-Y 寻径的二维网格网络，在 Y 维上使用了两对虚拟通道，在 X 维上使用了一对虚拟通道。图 3-55（b）中的虚拟网络可以用来避免消息在向西传输时出现的死锁，因为所有向东的 X 通道都没有使用。同样，图 3-55（c）的虚拟网络使用另一组 X 方向虚拟通道来支持只向东的传输。如果在不同的时刻分别使用两个虚拟网络，就可以避免死锁。

图 3-56（a）是在 X 维和 Y 维方向各有两条虚拟通道的二维网格网络，这些虚拟通道可以用来生成 4 个虚拟网络。西和北方向的通信可使用图 3-56（b）所示的虚拟网络，类似地，其

他 3 个方向的通信也可以构造另外 3 个虚拟网络，如图 3-56（c）、（d）和（e）所示。由于在任何一个虚拟网络中都不会出现环路，因此在这些网络上实现 X-Y 寻径方法时，完全可以避免死锁。

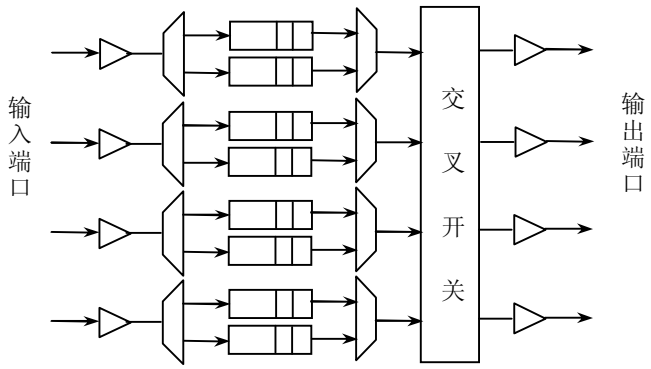
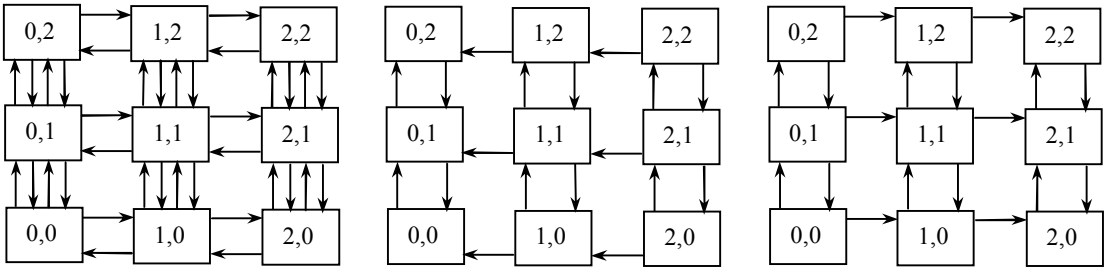
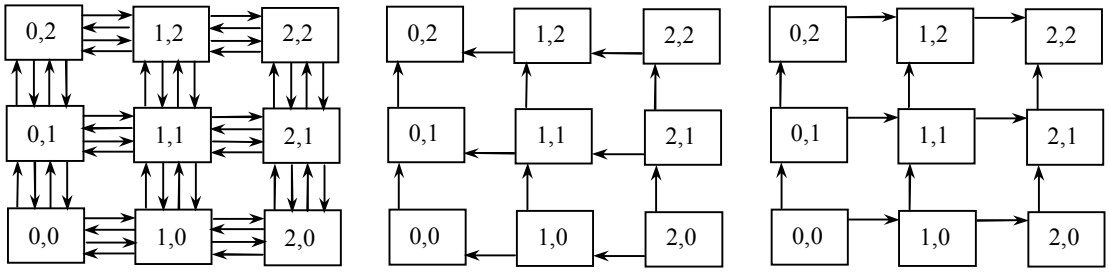


图 3-54　缓冲区劈开而构成一组虚拟通道

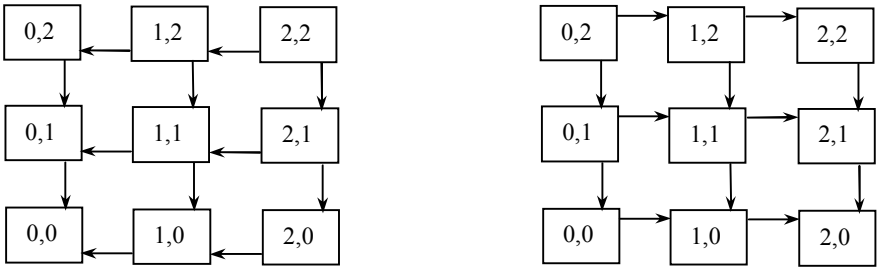


（a）多对虚拟通道的网格网络　　（b）向西传输虚拟网络　　（c）向东传输虚拟网络

图 3-55　利用虚拟通道避免 X-Y 寻径单双通道二维网络的死锁



（a）多对虚拟通道的网格网络　　（b）向西向北传输虚拟网络　　（c）向东向北传输虚拟网络



（d）向西向南传输虚拟网络　　（e）向东向南传输虚拟网络

图 3-56　利用虚拟通道避免 X-Y 寻径双通道二维网络的死锁

如果相邻结点之间的两对通道都是物理通道，那么 4 个虚拟网络中的任何两个都可以同时使用而不会产生冲突。如果相邻结点之间的双虚拟通道只能共享一对物理通道，那么只有（b）和（c）或（c）和（d）可以同时使用，而如（b）和（c）、（b）和（d）、（c）和（e）以及（d）和（e）等其他组合都由于缺少通道而不能同时存在。

2. 转弯选路技术

由于死锁环的形成实质是消息包寻径转弯不当而导致的，因此如果链路间没有环相关，就不会发生死锁。例如，蝶式置换是无环的，条件自然满足；对于树和胖树网络，只要向上的通道和向下的通道互不相关即可。无死锁选路算法网络并不意味着会免于死锁，但是只要网络接口总是可以接收消息，即使无法发送消息，也不会出现死锁。

在许多网络拓扑中，链路按维分组，因此在消息寻径时，从一维转到另一维会产生一个转弯。不转弯而改变方向可以认为是一个 180° 的转弯。另外，物理通道被分成虚拟通道后，在同一维同一方面上从一个虚拟通道转到另一个虚拟通道可以认为是一个 0° 的转弯。由于转弯可以合并成环，因此转弯选路（Turn-model Routing）的基本思想是禁止最小数量的转弯，从而防止环的出现。也就是说，只要禁止足够的转弯并打开所有的环，就可以防止死锁。

用来实现 n 维网格和 n-立方体的自适应寻径算法防止死锁可分为以下 6 个步骤：

（1）根据消息在链路内寻径的方向将链路分类。

（2）识别一个方向和另一个方面之间出现的转弯。

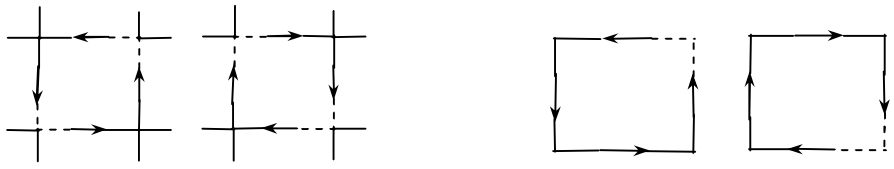
（3）识别转弯可能形成的简单环。

（4）在每个环中禁止一个转弯。

（5）在 n-立方体中，在不引入新环的前提下，尽可能多地合并涉及环绕链路的转弯。

（6）在不引入新环的前提下，加入 0° 和 180° 的转弯，如果在一个方向上有多个链路并且是非最小寻径算法，这些转弯是需要的。

对于二维网络，有 8 种可能的“转弯”和两个可能的抽象环（简单的圈），如图 3-57（a）所示。二维网孔中的维序选路（即 X-Y 选路）通过禁止 8 种可能转弯中的 4 种（图 3-57（a）中虚“转弯”线是非法的，而实“转弯”线是合法的）来防止死锁。4 种合法的转弯不会形成环，但也不允许任何自适应寻径。



（a）二维网络的抽象环及 X-Y 寻径允许的 4 种转弯　　（b）西向优先允许的 6 种转弯

图 3-57　二维网络的转弯选路

对于二维网格来说，实际上，禁止少于四种的转弯也能防止环的形成，如只需禁止两种转弯。如图 3-57（b）所示的实线箭头表示使用西向优先（不允许转向-X 方向）寻径算法时允许的 6 种转弯。西向优先寻径算法为：如果需要的话，先向西寻径，然后向南、向东、向北。可以看到，图 3-57（b）中禁止的两个转弯都是向西转弯，这说明，如果消息包需要向西传，那么它必须一开始就向西传送。如图 3-58 所示给出了西向优先算法的 3 条路径，标记为不可用的链路要么出现故障，要么被别的数据包占用。3 条路径中有一条为最短路径，其他两条由于在寻径时绕开了不可用的链路，而成为非最短路径。西向优先算法由于避免了环而成为无死

锁的寻径算法。对于最小路径寻径，如果目的结点在源结点的右侧（东侧），则寻径算法是完全自适应的。如果允许非最小路径寻径，则寻径算法是部分自适应的。

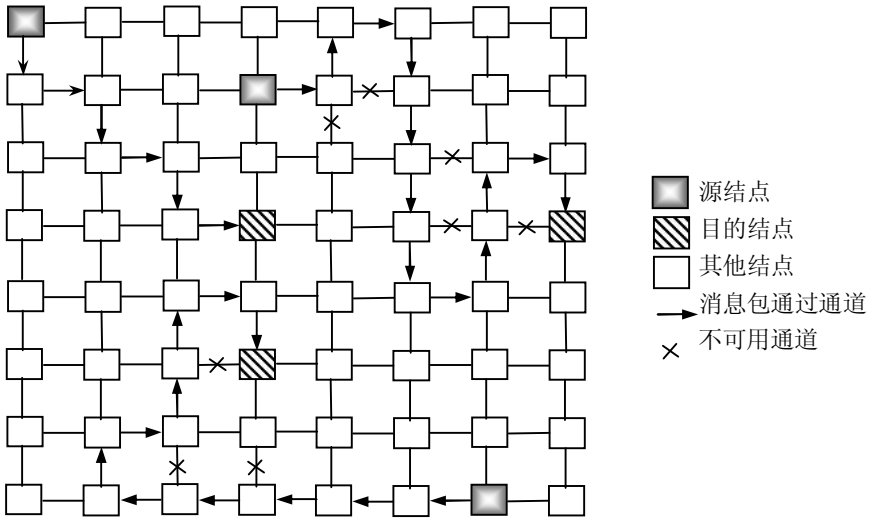
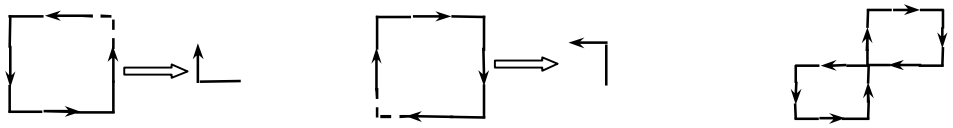


图 3-58　8×8 二维网络中西向优先的转弯选路

为了防止出现环，还有其他的方法来选择 6 种转弯，但被禁止的两种转弯不能任意选择。如果像图 3-57 那样禁止转弯的话，仍可能出现死链。如图 3-59（a）所示说明了剩下的 3 个方向的左转弯与被禁止的向右转弯是等价的，如图 3-59（b）所示说明了剩下的 3 个向右转弯与被禁止的向左转弯也是等价的。如图 3-59（c）所示描述了由 6 种未被禁止的转弯而产生的环。禁止两种转弯的方法有 16 种，其中有 12 种能防止死锁，如果考虑对称性，只有 3 种是独立的。这 3 种组合分别对应西向优先、北向最后和负向优先 3 个寻径算法。北向最后寻径算法不允许从北向东和从北向西转弯（即不允许来自+Y 方向的转弯），负向优先寻径算法不允许从北向西和从东向南转弯（即禁止从负方向转向正方向）。



（a）3 个左转等价禁止的右转　（b）3 个右转等价禁止的左转　（c）6 种未禁止的转弯而产生的环

图 3-59　二维网络中允许 6 种转弯的选择

转弯选路除了用于二维网格外，还可以用于 n 维网格、n-立方体网络的部分自适应寻径算法。将转弯选路用于 n-立方体网络的自适应寻径算法称为 P-立方寻径算法。令 n-立方体中的源结点 $S=s_{n-1}s_{n-2}\cdots s_0$，目的结点 $D=d_{n-1}d_{n-2}\cdots d_0$，集合 E 由所有 S 和 D 有差别的维数组成，E 的大小等于 S 和 D 之间的海明距离。如果 $s_i \neq d_i$，$i \in E$，E 将被分成两个不相交的子集 E_0 和 E_1；如果 $s_i=0$ 且 $d_i=1$，则 $i \in E_0$；如果 $s_j=1$ 且 $d_j=0$，则 $j \in E_1$。P-立方寻径的基本概念就是将寻径选择分成两个阶段。在第一个阶段，包在 E_0 中以任意维序寻径；而在第二阶段，包在 E_1 中以任意维序寻径。如果 E_0 为空，则包可以在 E_1 的任意维中寻径。

3.5.6　流量控制策略

死锁产生的根源是消息在互联网络传输中，所需求的资源大于现有资源。利用虚拟通道

解除死锁是在网络中缓冲区还未完全满时是可以实现的，若缓冲区资源完全利用尽的话，虚拟通道对于死锁的解除也无能为力。利用转弯选路可以避免死锁，但在网络资源一定时，若流量太大则会产生拥堵，使传输时延很大。而无论是死锁还是拥堵其产生的原因是网络中传输的流量太大，因此，需要对互联网络中传输的流量加以控制，即当网络中有多个数据流需要同时使用共享网络资源时，就需要有一种流量控制机制来控制这些数据流。所谓流量控制（Flow Control）策略是指对互联网络上的消息包的流量和路径进行控制，从而避免死锁和拥堵的方法。路径控制在前面已单独做过介绍，在此仅需介绍流量控制。

实际上，在所有网络以及网络的多个层次中都需要进行流量控制，但由于系统内互联网络的一些特点，使得其流量控制与局域网和广域网中的流量控制有很大区别。例如，在并行计算机中，可能在很短的时间内产生大量的并发数据流，并且对网络传输的可靠性要求很高。在此，主要考虑系统内互联网络的流量控制问题。

流量控制机制是一个调节器，当输入速率与输出速率不匹配时，才要进行提示。因此，在数据流传输很平缓的理想情况下，只要有足够的网络资源，就不需要进行流量控制。

3.5.6.1 包冲突及其解决方法

包冲突是指当多个包在某个结点为竞争缓冲区或通道资源而发生的现象，对包冲突必须确定某种方法解决和避免。一般来说，是通过控制网络的流量来解决这一问题。包冲突的解决方法主要包括两个问题：一是通道分配给哪一个包；二是没有分配到通道的包做什么。这里有4 种方法，如图 3-60 所示。

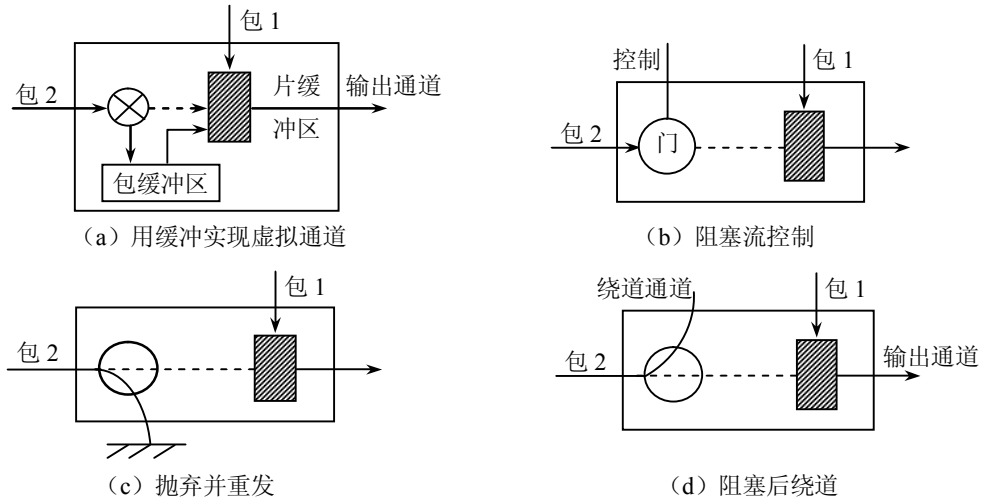


图 3-60 解决两个包请求同一条输出通道发生冲突时的流控制

（1）虚拟通道缓冲法。如图 3-60（a）所示，包 1 和包 2 争用一个通道，包 1 分配到片缓冲区，包 2 被拒绝进入通道的片缓冲区而暂时存放在另一个包缓冲区，当通道可以使用时再传送该包缓冲区中的包 2。该方法不会浪费已分配的资源，但要增加一个能存放整个包的缓冲区。

（2）阻塞流控制法。如图 3-60（b）所示，在虫蚀传递中出现冲突时采用该方法，包 1 继续传输，包 2 被阻塞不再前进，但并没有被抛弃。

（3）抛弃并重发法。如图 3-60（c）所示，在出现冲突时，包 1 继续传输，包 2 被抛弃，并且源结点重新发送包 2。

（4）绕道传送法。如图 3-60（d）所示，当出现冲突时，包 1 继续传输，包 2 选择另一

路径绕道传输。

实现上，某些互联网络中通常综合这些方法的优点，采用混合法。

3.5.6.2　链路层流量控制

数据从一个结点的输出端口通过链路传输到另一个结点的输入端口，数据可能存储在一个锁存器、队列或者一块内存缓冲区内，链路也可能是长的或短的、宽的或窄的、同步或异步的。问题在于目的结点的输入端口存储区域可能被填满，这就要求数据保存在源结点的存储区域中，直到目的结点的存储区域变得可用。这样，就有可能造成源结点的存储区域也被填满。链路层流量控制的实现主要依赖于链路设计，其基本思想是：目的结点向源结点提供反馈信息，指示是否能继续接收链路上传来的数据；源结点保持数据，一直到目的结点显示它能继续接收数据。对于不同的链路，流量控制在实现上有所不同。

在长度短且带宽宽的链路中，通过链路的传输很像一个机器内部寄存器间的数据传输，只不过扩展了一组控制信号，如图 3-61 所示，也就是把源和目的寄存器用满－空位进行扩展。如果源是满、目的是空，则发生传输。传输发生后，目的变满、源变空。如果交叉开关采用同步操作（如在 Cray T3D、IBM SP2、TMC CM-5 和 MIT J-machine 中），流量控制要确定在每个时钟周期内是否要进行传输（采用边缘触发是很容易实现的）。如果交叉开关是异步操作的，则非常像自定时的寄存器传输；当源满时，发出一个请求信号，准备开始传输；目的接到请求信号时，就从输入端口接收数据，当接收到数据后，就发出一个确认信号。对于长度短且带宽窄的链路，操作相似，只不过是一次请求/确认握手信号传输一串位片而已。

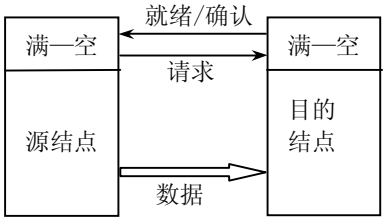


图 3-61　链路层的握手

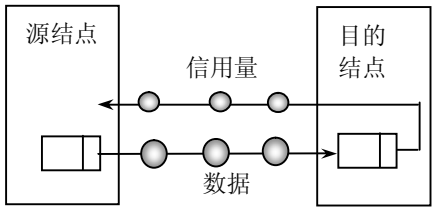


图 3-62　长链路传输数据片和信用量

对于长链路，则采用信用量机制。实质上，请求/确认握手可以看成是在源和目的之间传输单个令牌（Token）或信用量（Credit）。当目的结点释放输入缓冲区时，就把令牌传给源结点，源结点就使用这些信用量来发送数据片。对于长链路，只要对该信用量机制加以扩展，使得整个与链路传播延迟相关的流水线能被充满。如图 3-62 所示，传输一个确认信号的几个时钟周期期间，可以同时传输一些信用量信息。最简单的基于信用量的流量控制是让源结点保持目的结点的输入缓冲区中空表项的数目。在结点中设置一个计数器，计数器被初始化为输入缓冲区的大小，当发送一个数据片时，就减 1 计数。当计数器减为 0 时，就阻塞发送。当目的结点从输入缓冲区中取出一个数据片时，就将信用量还给源结点，源结点就增 1 计数。这样，输入缓冲区就不会溢出。该方法对于宽链路更为有效，因为宽链路有专门的控制线来传递确认信号。对于窄链路，需要多路复用确认信号进行反向通道（Opposite Going Channel）。另外，通过传输更大块的信用量，可以减少每个数据片的确认信号，但在丢失信用量令牌的情况下，不是很健壮。

基于信用量的链路层流量控制机制实际上是把目的结点输入缓冲区看成是一个具有低水位标记和高水位标记的水箱。当缓冲区的数据量低于低标记时，就向源结点发一个“发送”信

号；当高于高标记时，就向源结点发一个“停止”信号。

特别地，在调制解调器、处理器－网络接口等中，也应用信用量控制机制。

另外，在网络规模很大时，还要对端到端的流量进行控制。但对于系统内的互联网络，网络规模一般是有限的，即端到端路径中包含的链路不会很多，因此，通过链路层流量控制则可有效地实现端到端流量控制。

3.5.7 选播和广播寻径

3.5.7.1 互联网络的通信模式

在多计算机系统的互联网络中，通信模式包括单播、选播、广播和会议 4 种。

单播模式对应于一对一的通信情况，即一个源结点发送消息到达一个目的结点。选播模式对应于一到多的通信情况，即一个源结点发送同一个消息到多个目的结点。广播模式对应于一到全体的通信情况，即一个源结点发送同一个消息到全部结点。会议模式对应于多到多的通信情况。

单播模式在系统内的互联网络中，应用最为广泛，前面所介绍的寻径算法都是针对单播模式的。会议模式极其复杂，目前还没有很有效的寻径算法。

3.5.7.2 选播和广播寻径

无论是选播还是广播，都可以通过多次单播来实现，但其通信时延和通道流量都会比较大，即寻径效率不高。

例如，要在 3×4 网格上实现的选播寻径，即从源结点 S 传送一个消息包到 5 个目的结点 D_1～D_5。方法之一是采用 5 次单播的 X-Y 寻径算法来实现，如图 3-63（a）所示。总的通道流量为 1+3+4+3+2=13 条链路，通信时延为 4，即从 S 到 D_3 的路径长度。方法之二是采用选播模式，在一个中间结点上复制所传送的消息包，然而把该消息包的多个备份送到目的结点，从而可以减少通道流量。对此又有两种不同方法，分别如图 3-63（b）和（c）所示。图 3-63（b）中方法的通道流量为 7、通信时延为 4，因此更适合用于对时延要求较高的存储转发网络；而图 3-63（c）中方法的通道流量为 6、通信时延为 5，因此适合用于注重通道流量的虫蚀寻径网络。

如果要把一个包从 S 广播到所有其他的网格结点，则可以由所有网络结点构造一棵 4 层的生成树，使得到达树的第 i 层上结点的时延为 i，如图 3-63（d）所示，结点中的数字表示树的层号，采用该方法产生的通信时延和通道流量都能达到最小。

立方体互联网络是一种常用的通信网络，所以以它为例来讨论选播和广播寻径算法。

n-立方体网络通信的选播算法可通过构造贪婪选播树来实现，其基本思想是向那些可达到最多剩余目的结点的维方向发送包。贪婪选播算法所需的链路数与多次单播或广播树相比要少。

例如，图 3-64 是一棵贪婪选播树，源结点为 0101，现要发送消息包到 7 个目的结点 1100、0111、1010、1110、1011、1000 和 0010。从 0101 开始，由维二方向可以到达两个目的结点，由维四方向可以达到 5 个目的结点，因此，第一层所用的链路是 0101→0111 和 0101→1101。从结点 1101，由维 2 方向可以到达 3 个目的结点，由维一方向可以到达 4 个目的结点，因此，第二层所用的链路是 1101→1111、1101→1100 和 0111→0110。同理，第三层所用的链路是 1111→1110、1111→1011、1100→1000 和 0110→0010。第四层所用的链路是 1110→1010。

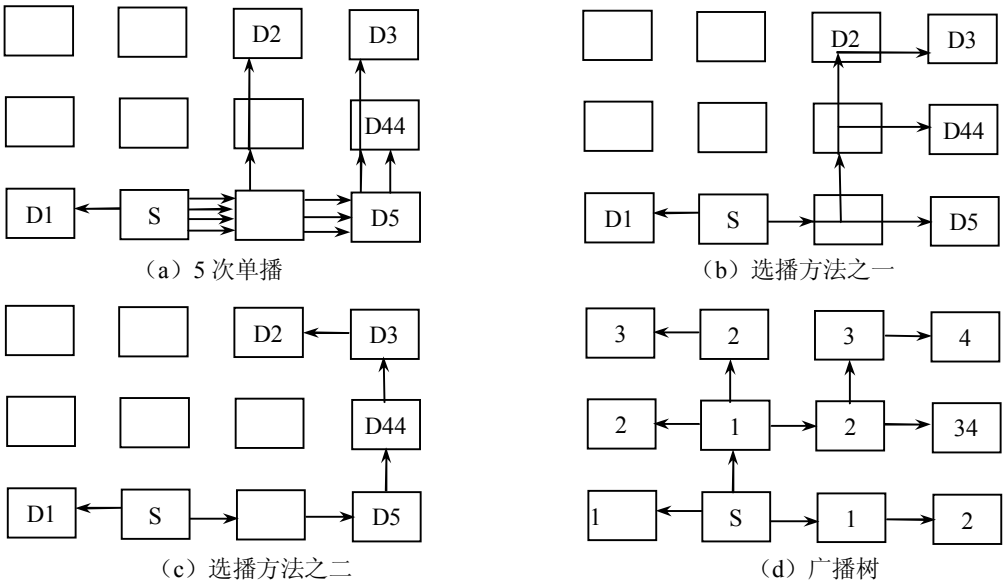


图 3-63　3×4 网格上单播、选播和广播的比较

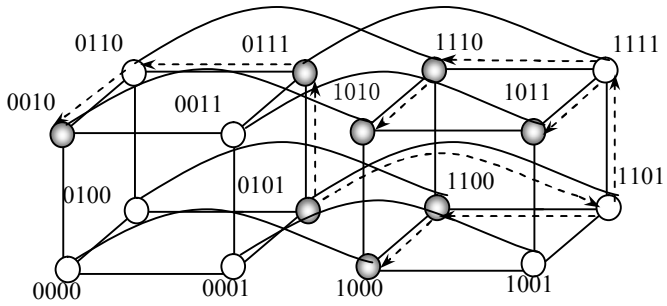


图 3-64　4-立方体寻径的贪婪选播树

在扩充选播树时，首先应该比较所有各维方向的可达性，然后选择某些维使剩余目的结点的集合最小。如果两维之间有连线，那么选择其中任何一维都可以。因此，所生成的树不是唯一的。

而 n-立方体网络通信的广播算法，可通过类似图 3-63（d）所示的生成树来实现。树中的根结点（源结点）到达所有结点的通信时延不超过 n。例如，一个根结点为 0000 的 4-立方体广播树，不仅到达所有其他结点的通信时延不超过 4，而且其广播通信的总流量也最小。

需要说明的是，为了避免在中间结点增加缓冲区，选播树或广播树中同一层的所有输出链路必须在传输向前推进一层之前处于就绪状态；而且在虫蚀寻径网络中实现选播操作时，每个结点的寻径器应具有复制片缓冲区数据的能力。

【例 3.4】 如图 3-65 所示为一四维超立方体，即 n=4。现设源结点的地址为 S=0110，目的结点的地址为 D=1101，请用 E-立方选路算法为 S 与 D 之间选择一条最短路径。

解　方向位向量 $R = S \oplus D = 0110 \oplus 1101 = 1011$，$V = S = 0110$

r1=1，$V = V \oplus 2^{i-1} = 0110 \oplus 0001 = 0111$；

r2=1，$V = V \oplus 2^{i-1} = 0111 \oplus 0010 = 0101$；

r3=1，跳过；

r4=1，$V = V \oplus 2^{i-1} = 0101 \oplus 1000 = 1101$（目的结点）。

所以，S 与 D 之间的最短路径为：S=0110→0111→0101→D=1101。

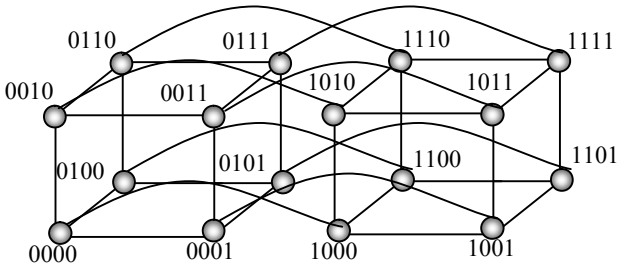


图 3-65　四维超立方体网络

习题三

3.1　解释下列名词术语。

互连函数	循环互连函数	网络规模	结点度
结点距离	网络直径	网络等分宽度	网络对称性
网络可扩展性	端口带宽	聚集带宽	对剖带宽
共享介质网络	非阻塞网络	直接网络	间接网络
级间连接	阻塞网	可重排非阻塞网	非阻塞网
终端标记寻径	消息	消息包	确定性寻径
算术选路法	源选路法	查表选路法	通道流量
自适应寻径	通道相关图	西向优先	北向最后
负向优先	流量控制策略	单播	选播
广播	会议	线路交换	存储转发
虚拟直通	虫蚀传递		

3.2　什么是互联网络？它一般由哪几部分组成？

3.3　什么是交叉开关？它一般由哪几部分组成？

3.4　什么是链路？简述它的分类。

3.5　根据连接的结点距离来看，互联网络可分为哪几类？从拓扑结构和特征来看，又可分为哪几类？

3.6　互联网络的基本特征包括哪几个方面？可用哪些方法来描述？

3.7　什么是通信时延？它由哪几部分组成？

3.8　什么是静态互联网络？它是如何分类的？

3.9　静态互联网络中的二维网络、三维网络各主要有哪几种？

3.10　什么是动态互联网络？动态互联网络的形式有哪几种？试比较它们间的差异。

3.11　交叉开关允许的映射有哪些？只允许一对一映射的 n×n 的交叉开关至多可实现的连接或置换为多少？

3.12　什么是多级交叉开关互联网络？它可分为哪几种？为什么通常都采用该种网络？

3.13　2×2 交叉开关有哪几种工作状态？它有哪两种类型？

3.14　在多级交叉开关互联网络中，交叉开关的控制方式有哪几种？

3.15　简述消息传递的格式与方式，并对各种传递方式加以比较，分别给出通信时延的计算公式。

3.16　什么是路由选择？路由选择有哪几种算法？

3.17　试比较确定性寻径与自适应寻径的优缺点。

3.18　什么是虚拟通道？它一般由哪几部分组成？

3.19　简述广义死锁含义？它是如何分类的？

3.20　什么是死锁？简述死锁形成的原因。

3.21　死锁解除与避免技术方法有哪些？简述它们的原理思想。

3.22　在二维网络中，哪几种转弯是允许的？哪几种转弯是禁止的？但至少应禁止哪两种转弯也能防止环的形成？

3.23　什么是包冲突？解决包冲突的方法有哪些？

3.24　简述链路层流量控制的基本思想。

3.25　互联网络的通信模式有哪几种？简述贪婪选播树选播算法的基本思想。

3.26　设 16 个处理器编号分别为 0、1、…、15，要用单级互联网络，当互连函数分别为：

（1）$Cube_3(Cube_1)$　（2）PM_{+2}　（3）PM_{-3}　（4）Shuffle

（5）Butterfly(Butterfly)　（6）$\sigma_{(2)}$　（7）$\sigma^{(3)}$　（8）σ^{-1}

（9）$\beta_{(1)}$　（10）$\beta^{(3)}$　（11）$\rho(\rho)$　（12）$\rho_{(0)}$

（13）$\rho^{(2)}$

时，第 13 号处理器分别与哪一个处理器相连?

3.27　设 PM2I 网络有 8 个结点，请画出 $PM_{\pm 0}$、$PM_{\pm 1}$ 和 $PM_{\pm 2}$ 互联网络的连接图。

3.28　要求用直径最小的三维网、六维二元超立方体和带环立方体（CCC）设计一台由 64 个结点组成的多计算机直接连接网络，令 d、D、L 分别为网络结点度、直径和链路数，且用 $(d\times D\times l)^{-1}$ 来衡量其性能，按性能排出 3 种网络的顺序。

3.29　设 E 为交换函数、S 为均匀洗牌函数、PM2I 为移数函数，且入出端编号用十进制数表示，现有 32 台处理机。

（1）用 E_0 和 S 构成均匀洗牌交换网（每步只能用一次 E_0 和 S），网络直径是多少？从 5 号处理机发数据到 7 号处理机，最短路径要经过几步？列出要经过的处理机编号。

（2）采用移数函数构成互联网络，网络直径是多少？结点度是多少？与 2 号处理机距离最远的是几号处理机？

3.30　在有 16 个处理器的均匀洗牌网络中，若要使第 0 号处理器与第 15 号处理器相连，需要经过多少次均匀洗牌和交换置换。

3.31　N=16 的互联网络入出端编号分别为 0～15，若实现的互连关系可用互连函数 $f(x_3\ x_2\ x_1\ x_0) = x_0\ x_1\ x_2\ x_3$ 表示，该互连函数是否是循环互连函数？如是，请写出循环互连函数表示法表示。

3.32　画出 16 台处理器仿 IlliacⅣ的模式进行互连的连接图，列出 PE_0 分别只经一步、二步和三步传送就能将信息传送到的各处理器。给出任何一台处理器 PU_i（$0\leqslant i\leqslant 15$）与其他处理器直接互连的一般表达式。

3.33　对于采用级控制方式的三级 STARAN 网络，当第 i 级开关（$0\leqslant i\leqslant 2$）为直送状态

时，不能实现哪些结点之间的通信?为什么？当第 i 级开关为交叉状态时，不能实现哪些结点之间的通信?

3.34　在编号分别为 0、1、2、…、9、A、B、…、F 的 16 个处理器之间，要求按下列配对通信：(B,1)、(8,2)、(7,D)、(6,C)、(E,4)、(A,0)、(9,3)、(5,F)。试选择所用互联网络类型和控制方式，并画出该互联网络的拓扑结构和各级的交换开关状态图。

3.35　画出编号分别为 0、1、…、9、A、B、…、F，共 16 个处理器之间实现 STARAN 网络，当采用级控制信号为 1100 时，9、A、B、…、F 号处理器连向哪个处理器?

3.36　在一并行处理机中，用 STARAN 网络连接 16 个处理器，要实现相当于先 4 组 4 元交换，然后是 2 组 8 元交换，再次是 1 组 16 元交换的交换函数功能，请写出此时各处理器之间所实现的互连函数的一般式，画出相应多级互联网络拓扑结构图，标出各级交换开关的状态。

3.37　假定 8×8 矩阵 $A=(a_{ij})$顺序存放在存储器的 64 个单元中，用什么样的单级互联网络可实现对该矩阵的转置变换？总共需要传送多少步？

3.38　假定有 128 个处理器，采用 PM2I 多级网络完成某种变换，若 i=2 的一级损坏，今拟用 $Cube_i$ 网络代替损坏的这一级，试说明最多要几级?

3.39　用单级方体网络模仿 N=16 的单级 PM2I（i=0）网络，最差情况下要用几次单级循环传送？

3.40　分别使用方体单级网络和均匀洗牌单级网络将一个处理器的数据播送到所有的 2^n 个处理器中去，问各需要循环传送多少步？其中假设单级网络每步只能进行一种变换，方体单级网络第 i 步完成 $Cube_i$ 的变换传送。

3.41　写出 N=8 的蝶式置换的互连函数，如采用 Omega 网络，则需几次通过才能完成此变换？画出 Omega 网络实现此变换的控制状态图。

3.42　具有 $N=2^n$ 个输入端的 Omega 网络，采用单元控制。

（1）N 个输入总共应有多少种不同的排列?

（2）该 Omega 网络通过一次可以实现的置换总共可有多少种是不同的?

（3）假设 N=8，计算出一次通过能实现的置换数占全部排列的百分比。

3.43　画出 N=8 的 n 方体网络，标出采用单元控制同时实现 0→3、1→7、2→4、3→0、4→2、5→6、6→1、7→5 的传送时各交叉开关的状态，说明为什么不会发生阻塞?

3.44　设 N 个输入端的 Omega 网络，它的每个交叉开关单元都是独立控制的。

（1）给定任意一个源—终端（S-D）对，其连接通路可用终端地址唯一控制。现不用终端地址 D 作为寻径标记，而定义 $T=S\oplus D$ 作为寻径标记，试说明可以单独用 T 来确定连接通路。用 T 作为寻径标记的优点是什么？

（2）Omega 网络能实现播送（一源到多目的）功能，若目的处理器数为 2 的幂，试给出一简单的寻径算法以完成这一功能。

3.45　一个 N=8 的 Omega 网络连接 8 台处理器（P_0～P_7），如果处理器 P_6 要把数据播送到 P_0～P_4，处理器 P_3 要把数据播送到 P_5～P_7，那么 Omega 网络能否为它们的播送要求实现连接？画出实现播送的 Omega 网络的交叉开关状态图。

3.46　分别画出 4×9 的一级交叉开关以及用两级 2×3 的交叉开关组成的 4×9 的 Delta 网络，比较一下交叉开关设备量的多少。

3.47　如图 3-66 所示是一个 $2^3\times2^3$ 的 Delta 网络。

（1）问该网络在任何处理机和任何存储器模块之间是否都有一个通路？

（2）令 $d_2d_1d_0$ 是二进制编号为 $P_2P_1P_0$ 的某处理机所要访问的存储模块号的二进制编码，网络中第 0、1、2 级的控制信号分别为 x_0、x_1、x_2，其中第 i 级控制信号 x_i 为 0 时，控制为直连，x_i 为 1 时控制为交叉连接。根据某处理机 $P_2P_1P_0$ 的给出的访存模块号 $d_2d_1d_0$，为了将网络通路建立起来，请写出控制信号 x_0、x_1、x_2 与 $d_0d_1d_2$ 及 $P_0P_1P_2$ 的逻辑关系式。

（3）若 0 号处理机访问 2 号存储模块的同时，4 号处理机要访问 4 号存储模块，6 号处理机要访问 3 号存储模块，问是否发生阻塞？

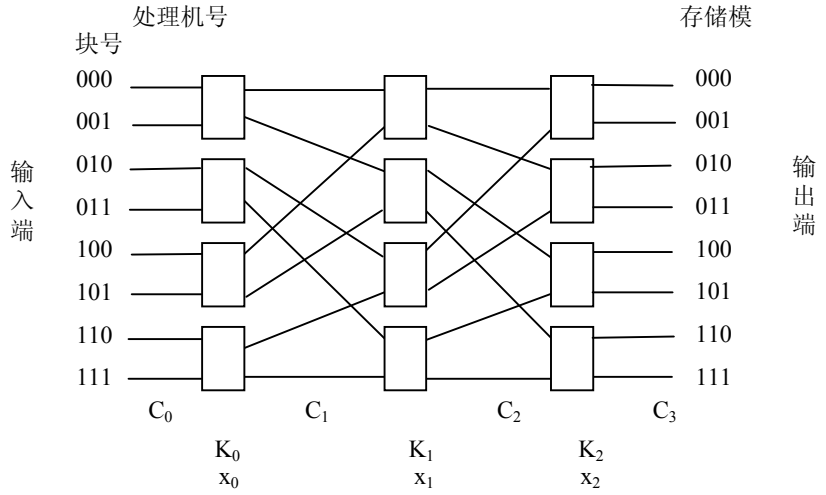


图 3-66　Delta 网络

3.48　画出用 4×4 交叉开关组成一个 3 级的 16×16 交叉开关网络，其设备量比单级 16×16 的交叉开关节省多少设备？举例说明在输入和输出之间存在着较多的冗余路径。

3.49　令 $2^m \times 2^m$ 矩阵 A 以行主方式存放在主存储器中，试证明在对 A 进行 m 次完全均匀洗牌变换后可获得转置矩阵 A^T。

3.50　（1）画出 2×2 交叉开关构成的 16 个输入端的 Omega 网络。

（2）结点 1011 传送消息给结点 0101，同时结点 0111 传送信息给结点 1001，画出完成这一寻径的交叉开关设置。这种情况会出现阻塞吗？

（3）试计算这个 Omega 网络一次通过实现的置换个数，一次通过实现的置换个数占全部置换的百分比为多少？

（4）这个网络实现任意一个置换最多的通过次数是多少？

3.51　根据推理分析或反例证明下列命题的正确性。

（1）采用虫蚀传递的超立方体多计算机系统，相邻结点之间有一对方向相反的单向通道，试证明在该系统上实现 E 立方体传递不会死锁。

（2）试证明在二维网络上实现 X-Y 传递不会死锁。

（3）试证明在三维网络（k 元 n 方体）上实现 E 立方体传递不会死锁。

3.52　在一个 8×8 的网络上，源结点是(3,5)，目的结点是(1,1)，(1,2)、(1,6)、(2,1)、(4,1)、(5,5)、(5,7)、(6,1)、(7,1)、(7,5)。

（1）确定一条优化的选播路径。

（2）确定计算机中最优（结点间距离最短）的寻径路径。

3.53　试确定下列网格计算机和超立方体多计算机中的最优寻径路径。

（1）假设有一个 64 个结点的超立方体网络，根据 E 立方体寻径算法，画出从结点 101101 发送消息给结点 011010 的路径，并标出这条路径上的所有中间结点。

（2）在一个 8×8 网格上，根据下面的条件确定两条优化的选播路径，源结点是(3,5)10 个目的结点是(1,1)、(1,2)、(1,6)、(2,1)、(4,1)、(5,5)、(5,7)、(6,1)、(7,1)、(7,5)。

①第一条选播路径应使通道数最少。

②第二条选播路径应使从源结点到每个目的结点的距离最短。

（3）假设有一个 16 个结点的超立方体网络，源结点是 1010，9 个目的结点是 0000、0001、0011、0101、0111、1111、1101、1001。根据贪婪算法，尽可能地使用流量较少的通道，确定一条较优的选播路径，使从源结点到所有目的结点的距离最短。

3.54　对于 16 个结点的二维网格网络见图 3-48，采用 X-Y 维序寻径算法，标出从结点(3,0)到结点(0,3)、从结点(0,0)到结点(3,3)的路径。

3.55　对于 4 方体网络见图 3-65，从结点 0000 到结点 1111，有多少条最短路径？为什么？用 E-立方维序寻径算法找出其中一条最短路径。