

水平线（进线）和垂直线（出线）的每个相交位置上是一个交叉点（crosspoint）。交叉点是一个以电子方式开关的小开关，具体取决于水平线和垂直线是否需要连接。在图8-3a中我们看到有三个交叉点同时闭合，允许（CPU，存储器）对（010，000）、（101，101）和（110，010）同时连接。其他的连接也是可能的。事实上，组合的数量等于象棋盘上8个棋子安全放置方式的数量（8皇后问题）。

交叉开关最好的一个特性是它是一个非阻塞网络，即不会因有些交叉点或连线已经被占据了而拒绝连接（假设存储器模块自身是可用的）。而且并不需要预先的规划。即使已经设置了7个任意的连接，还有可能把剩余的CPU连接到剩余的存储器上。

当然，当两个CPU同时试图访问同一个模块的时候，对内存的争夺还是可能的。不过，通过将内存分为 n 个单元，与图8-2的模型相比，这样的争夺概率可以降至 $1/n$ 。

交叉开关最差的一个特性是，交叉点的数量以 n^2 方式增长。若有1000个CPU和1000个存储器我们就需要一百万个交叉点。这样大数量的交叉开关是不可行的。不过，无论如何对于中等规模的系统而言，交叉开关的设计是可用的。

3. 使用多级交换的UMA 多处理机

有一种完全不同的、基于简单 2×2 开关的多处理机设计，参见图8-4a。这个开关有两个输入和两个输出。到达任意一个输入线的消息可以被交换至任意一个输出线上。就我们的目标而言，消息可由四个部分组成，参见图8-4b。Module（模块）域指明使用哪个存储器。Address（地址）域指定在模块中的地址。Opcode（操作码）给定了操作，如READ或WRITE。最后，在可选的Value（值）域中可包含一个操作数，比如一个要被WRITE写入的32位字。该开关检查Module域并利用它确定消息是应该送给X还是发送给Y。

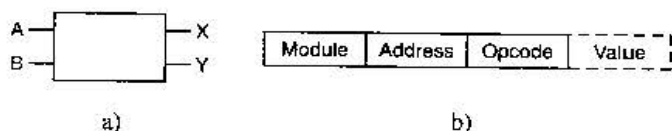


图8-4 a) 一个带有A和B两个输入线以及X和Y两个输出线的 2×2 的开关；b) 消息格式

这个 2×2 开关可有多种使用方式，用以构建大型的多级交换网络（Adams 等人，1987；Bhuyan 等人，1989；Kuman和Reddy，1987）。有一种是简单经济的Omega网络，见图8-5。这里采用了12个开关，把8个CPU连接到8个存储器上。推而广之，对于 n 个CPU和 n 个存储器，我们将需要 $\log_2 n$ 级，每级 $n/2$ 个开关，总数为 $(n/2) \log_2 n$ 个开关，比 n^2 个交叉点要好得多，特别是当 n 值很大时。

Omega网络的接线模式常被称作全混洗（perfect shuffle），因为每一级信号的混合就像把一副牌分成两半，然后再把牌一张张混合起来。接着看看Omega网络是如何工作的，假设CPU 011打算从存储器模块110读取一个字。CPU发送READ消息给开关1D，它在Module域包含110。1D开关取110的首位（最左位）并用它进行路由处理。0路由到上端输出，而1的路由到下端，由于该位为1，所以消息通过低端输出被路由到2D。

所有的第二级开关，包括2D，取用第二个比特位进行路由。这一位还是1，所以消息通过低端输出转发到3D。在这里对第三位进行测试，结果发现是0。于是，消息送往上端输出，并达到所期望的存储器110。该消息的路径在图8-5中山字母a标出。

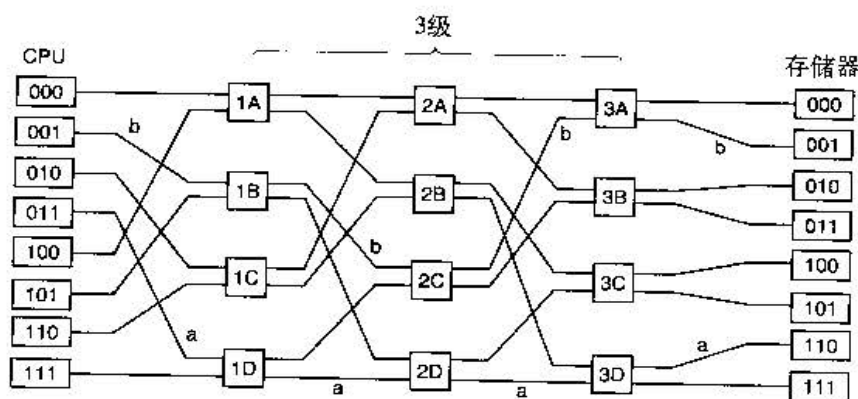


图8-5 Omega交换网络