可重构比特置换网络配置信息提取算法研究 *

高 飞[1], 李红燕[2], 戴紫彬[1], 张永福[1]
(1. 解放军信息工程大学 电子技术学院,郑州 450004; 2. 上海第二工业大学, 上海 201209)

摘 要: 比特置换单元由比特置换网络和配置信息组成,基于 Benes 网络实现可重构比特置换网络,并改进和实现了两种配置信息提取算法,即二分法和并行算法。这两种方法能有效控制 Benes 网络中各开关元件的状态,实现各个待置换的比特在网络中非阻塞正确选路,其各有特点,在应用中可根据实际需要加以选择。

关键词: 配置信息; Benes 网络; 二分法; 并行算法
中图分类号: TP309 **文献标志码**: A **文章编号**: 1001-3695(2010)05-1867-04
doi:10.3969/j.issn.1001-3695.2010.05.074

Research on reconfigurable bit permutation network configuration information extraction

GAO Fei[1], LI Hong-yan[2], DAI Zi-bin[1], ZHANG Yong-fu[1]
(1. *School of Electronic Technology, PLA Information Engineering University, Zhengzhou* 450004, *China*; 2. *Shanghai Second Polytechnic University, Shanghai* 201209, *China*)

Abstract: The bit permutation unit consists of bit permutation network and configuration information. This paper brought out reconfigurable bit permutation network based on the Benes network, improved and implemened two kinds of configuration information extraction algorithms, namely, the dichotomy and parallel algorithms. These two methods with different characteristics can effectively control the state of each switch element in Benes network to achieve the non-blocking correct routing of the various bits to be replaced in the network. They can be chosen according to actual needs in the application.

Key words: configurable information; Benes network; dichotomy; parallel algorithm

0 引言

在密码学中,比特置换提供了字节操作所无法实现的混乱扩散等功能。置换作为扩散的首要手段,在密码算法中得到了广泛应用。从计算机体系结构的角度来说,随着多媒体和信息安全技术的发展,快速比特置换将成为面向字节的处理器的一个重要发展方向[1]。

本文借鉴通信互联网络及其排序算法方面的研究方法和成果,来进行密码微处理器中比特置换单元的构造,基于 Benes网络的可重构比特置换网络架构,重点改进并实现了两种配置信息提取算法。

1 配置信息

在专用密码微处理器上实现的比特置换单元,其结构如图 1 所示。主要由两部分电路构成,即比特置换网络和配置信息寄存电路。比特置换网络是整个单元的核心,完成比特置换运算,能实现 $N!$ 种 N 比特位宽的任意置换。比特置换网络根据不同的配置信息完成不同的置换,配置信息寄存电路对配置信息进行暂存,并完成对比特置换网络的静态配置。

一个比特置换网络应该实现输入、输出之间所有可能的变换,即 N-N 置换单元的 N 个输入中的任何一个能够实现 N 个输出中的任何一个。因此,比特置换网络所需要的可控编码的宽度和它所能实现的选择变换的个数可由下述定理给出:

定理 1 对于 $N\times N$ 比特置换,需要从 $N!$ 个结果中挑选出来一个作为结果,因此至少需要 $\log_2(N!)$ 比特作为置换信息指定[1~3]。

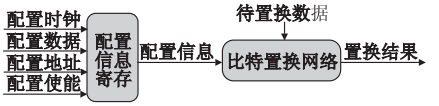

图1 比特置换单元结构

定理 2 设一个 $N\times N$ 置换单元能够实现其输入到输出的所有选择变换,即其 N 个输出中的任何一个能够选择 N 个输入中的任何一个,则该置换单元需要 $N\log_2 N$ 位[4]可控编码。

定理 1 给出了一个 $N\times N$ 置换单元置换操作所需要的最少置换信息,定理 2 则给出了一种可知置换网络可控编码的位数。一般情况下,一个 $N\times N$ 置换单元所需的可控编码介于两者之间。表 1 给出了当 $N=32$、64、128 时,可控编码的取值范围。

表 1 比特置换的可控编码

N	min	max
32	118	160
64	296	384
128	717	896

可见,一般情况下,比特置换操作的控制信息较多,直接由

收稿日期: 2009-09-21; **修回日期**: 2009-12-15 **基金项目**: 国家"863"计划资助项目(2008AA01Z103)

作者简介:高飞(1973-),男,河南新乡人,工程师,博士研究生,主要研究方向为信息安全、密码工程(gslsxd@126.com);李红燕(1975-),女,讲师,博士,主要研究方向为计算机应用;戴紫彬(1965-),男,教授,博导,主要研究方向为密码工程;张永福(1942-),男,教授,博导,主要研究方向为信息安全.

指令给出配置信息困难,为此,必须采用静态重构技术,即设计专门配置寄存器,指令使用前首先进行静态配置,配置信息算法由软件实现,用户指定置换的目标序列,软件即可为用户生成相应的置换网络配置信息,配置完成后,配置信息通过配置指令注入到比特置换网络中。

2　基于 Benes 网络的比特置换网络

自从 1965 年 Benes[5] 提出了经典的交换网络理论以来,国外的许多学者对互联网络进行了大量的研究。由于交换网络与密码学中的置换网络功能非常类似,可以借鉴通信交换网络的研究方法,结合密码处理中置换操作的特点,实现比特置换单元中比特置换网络架构。

基于 $N\times N$ 任意比特置换要求,比特置换网络应该具有无阻塞的特性。根据互联网络的阻塞特点,一般将其分为阻塞网、可重排非阻塞网、非阻塞网。非阻塞网虽然理想,但是它的级联级数以及相关的硬件设施非常复杂,所以在比特置换系统的设计中,没有采用这种网络而是采用可重排非阻塞网。

Benes 二进制置换网络,简称 Benes 网络,是一种可重排非阻塞网,能实现输入端到输出端的所有置换。Benes 网络的结构在文献[6]中有详细的描述。这里不再赘述。8 输入的 Benes 网络结构如图 2 所示。

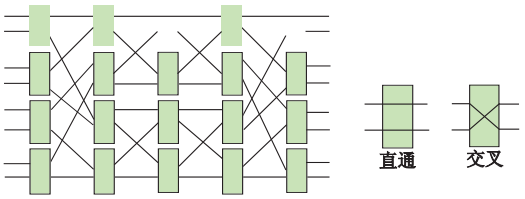

图2　8×8的Benes网络及其开关元件状态

Benes 网络具有以下一些很好的特性可以用于解决任意比特置换的问题:

a)Benes 网络由 Butterfly 网络和逆 Butterfly 网络连接而成,通过开关元件状态的改变可以实现 $N\times N$ 的任意交换,因此可用来构造实现任意置换操作的比特置换单元。

b)Benes 网络可以被拆分成不同的级。可以逐级利用和控制这个网络。规模为 N 的 Benes 网络是由 $2\log_2 N-1$ 级开关元件构成,每级开关元件总数是 $N/2$ 个。

c)Benes 网络中每一个开关元件实现 2×2 的比特置换,需要一个比特来配置,决定其状态(交叉或直通),每一级的 $N/2$ 个开关元件可以由 $N/2$ 个比特来配置。因此,为确定数据在网络中的路径,一个 $N\times N$ 的 Benes 网络需要 $N\log_2 N-N/2$ 比特配置信息。

d)Benes 网络是一种递归结构,可以用较小的 Benes 网络构成较大的 Benes 网络,较大的 Benes 网络可以拆分成为较小的 Benes 网络。

基于 Benes 网络实现 $N\times N$ 比特置换操作,这种网络的特点使其可以根据各个开关元件的配置信息,实时改变各级开关元件的状态,实现 $N!$ 种置换中指定的一种。此时,电路由 $2\log_2 N-1$ 列开关元件及其间的连线构成,每列开关元件有 $N/2$ 个,每个开关元件由两个 2 选 1 的数据选择器构成。共计需要 $2N\log_2 N-N$ 个数据选择器。

3　配置信息提取

由于 Benes 网络是一种可重构的网络,如何控制网络中各个开关元件的状态,进而决定各个待置换的比特如何在网络中选路,而不至于发生阻塞,就需要有一种正确简洁的配置信息提取(寻径)算法。借鉴互联网络排序算法的研究成果,笔者改进并实现了两种配置信息提取算法分别介绍。

3.1　二分法(dichotomy algorithm)

利用二分算法确定每级开关元件的状态。对于每个 2×2 开关元件,本文定义 $2i$ 为上输入端或上输出端,$2i+1$ 为下输入端或下输出端,如图 3 所示,并且定义同一开关元件中,两输入互斥,两输出互斥,而任意属于不同开关元件的输入或输出不互斥。

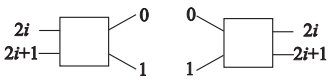

图3　开关元件

根据实际的输入和输出互斥对,得到一个二分图,即上下两个 $N/2\times N/2$ 子网 $B_0(n-1)$ 和 $B_1(n-1)$。其中 $n=\log_2 N$,并确定两互斥对点集 X 和 Y,如图 4 所示。

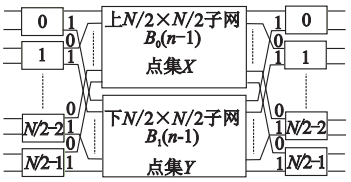

图4　二分图及互斥对点集X和Y

其中,X 中的元素连接左边一级开关元件的上输出端和右边一级开关元件的上输入端,即 $2i$;Y 中的元素连接左边一级开关元件的下输出端和右边一级开关元件的下输入端,即 $2i+1$。这样确定出左、右两级开关元件的状态。根据这两列开关元件状态把待置换的比特通过级间连线送往左边下一级和右边下一级。由此得到新的互斥对,由互斥对确定新的 X、Y 集合,进而确定出两列开关元件的状态。循环这个过程,直到所有的开关元件状态都确定完毕。

具体算法描述如下:

a)输入待置换的比特序列 $0,1,2,3,\cdots,N-1$;输出 $\Pi(0),\Pi(1),\Pi(2),\Pi(3),\cdots,\Pi(N-1)$。其中 Π 为置换变换关系。

b)输入互斥对。$O_1=\{0,1\},O_2=\{2,3\},\cdots,O_{N/2}=\{N-2,N-1\}$。

c)输出互斥对。$P_1=\{\Pi(0),\Pi(1)\},P_2=\{\Pi(2),\Pi(3)\},\cdots,P_{N/2}=\{\Pi(N-2),\Pi(N-1)\}$;

d)确定互斥点集 X、Y,分别包含 $N/2$ 个元素,满足条件 $|X\cap P_i|=|X\cap O_i|=1$,$|Y\cap P_i|=|Y\cap O_i|=1$。其中 $1\leqslant i\leqslant N/2$。

具体操作如下:任选一个 $P_i(1\leqslant i\leqslant N/2)$,在 P_i 中选择一个元素 a 作为 X 中的一个元素,P_i 中除去 a 之外的另一个元素 b 放入 Y 中;找出包含 a 的 $O_j(1\leqslant j\leqslant N/2)$,则 O_j 中除去 a 之外的另一个元素 c 放入 Y 中;找出包含 c 的 $P_k(1\leqslant k\leqslant N/2)$,则 P_k 中除去 c 之外的另一个元素 d 放入 X 中;依次对每个 P_i 作筛选,如此循环,可得 X 和 Y。

e)X、Y 用来确定位置左右对称的两级开关元件的状态(交叉或直通)。对于左边的一级 2×2 开关元件,X 中的元素对应连接开关元件的上输出端,Y 中的元素对应连接开关元件的下输出端;对于右边的一级 2×2 开关元件,X 中的元素对应连接开关元件的上输入端,Y 中的元素对应连接开关元件的下输入端。X、Y 中左右两边元素的前后顺序与其在左右两级开关元件中的相应顺序一致。由此左右两级开关元件状态得到确定。

f)根据 e)确定的两列开关元件,把待置换的比特通过开关元件经过级间连线送到中间两级开关元件的输入和输出。由此得到新的互斥对。重复步骤 d)和 e)。直到所有开关元件的状态都被确定。

举例如下:

用 8×8 的 Benes 网络实现 8×8 的置换。输入信号序列 0,1,2,3,4,5,6,7;输出 7,2,0,4,3,1,6,5。

1)$\Pi(0)=7,\Pi(1)=2,\Pi(2)=0,\Pi(3)=4,\Pi(4)=3,\Pi(5)=1,\Pi(6)=6,\Pi(7)=5$;

2)输入互斥对:$O_1=\{0,1\},O_2=\{2,3\},O_3=\{4,5\},O_4=\{6,7\}$;

3)输出互斥对:$P_1=\{7,2\},P_2=\{0,4\},P_3=\{3,1\},P_4=\{6,5\}$;

4)构造 X、Y,每组均包含四个元素,$X=\{0,3,5,7\},Y=\{1,2,4,6\}$;

5)得到网络中最左和最右两级开关元件状态,如图 5 所示。

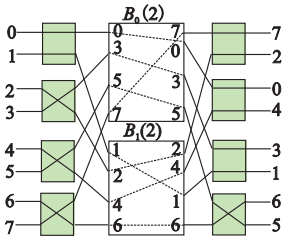

图5　最左和最右两级开关元件状态

6)根据上下两个 4×4 子网 $B_0(2)$ 和 $B_1(2)$,得到新的互斥对;重复步骤 4),5),得到中间两级开关元件状态;最后得到最中间一级开关元件状态,如图 6 所示。

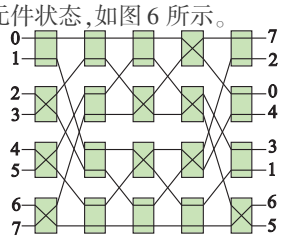

图6　二分法实现在8×8Benes网络上的比特置换

3.2　并行算法(*parallel routing algorithm*)

令布尔变量 a_i 表示第 i 个输入开关元件的状态,$a_i=0$ 表示第 i 个输入开关元件状态为直通,$a_i=1$ 表示第 i 个输入开关元件状态为交叉,如图 7 所示。

其中,上输入端 $2i$ 始终与 a_i 的值相同,下输入端 $2i+1$ 始终与 a_i 的值相反,所以可以表示为

$$2i=a_i \quad 2i+1=a_i \tag{1}$$

布尔变量 b_i 表示第 i 个输出开关元件的状态,$b_i=0$ 表示第 i 个输出开关元件状态为直通,$b_i=1$ 表示第 i 个输出开关元件状态为交叉,如图 8 所示。

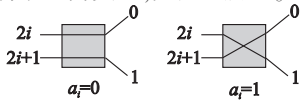

图7　输入开关元件状态

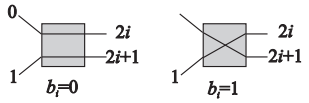

图8　输出开关元件状态

其中,上输出端 $2i$ 始终与 b_i 的值相同,下输出端 $2i+1$ 始终与 b_i 的值相反,所以可以表示为

$$2i=b_i \quad 2i+1=b_i \tag{2}$$

令 α 和 β 分别表示某一输入和输出。其中 $\alpha: O\rightarrow\{0,1\}$,$\beta: P\rightarrow\{0,1\}$,$\Pi$ 为置换变换关系,则有

$$\alpha(k)=\beta(\Pi(k)) \quad k=0,1,2,\cdots,N-1$$

又式(1)(2)得出

$$\alpha(k)=\overline{\alpha}(k+1) \quad k=0,2,4,\cdots,N-2$$

所以　　$\beta(\Pi(k))=\alpha(k)=\overline{\alpha}(k+1)=\overline{\beta}(\Pi(k+1))$

其中:$k=0,2,4,\cdots,N$。

输入、输出序列的路由比特可以分别表示为如下形式:

$$\alpha(k)=\begin{cases} a_{k/2} & k\text{ 为偶数} \\ \overline{a}_{(k-1)/2} & k\text{ 为奇数} \end{cases}$$

$$\beta(k)=\begin{cases} b_{k/2} & k\text{ 为偶数} \\ \overline{b}_{(k-1)/2} & k\text{ 为奇数} \end{cases}$$

其中:$k=0,1,2,\cdots,N-1$。

具体算法描述如下:

a)输入待置换的比特序列 0,1,2,3,…,$N-1$;输出 $\Pi(0),\Pi(1),\Pi(2),\Pi(3),\cdots,\Pi(N-1)$。其中 Π 为置换变换关系。

b)输入互斥对。$O_1=\{0,1\},O_2=\{2,3\},\cdots,O_{N/2}=\{N-2,N-1\}$。

c)输出互斥对。$P_1=\{\Pi(0),\Pi(1)\},P_2=\{\Pi(2),\Pi(3)\},\cdots,P_{N/2}=\{\Pi(N-2),\Pi(N-1)\}$。

d)用布尔表达式 $\alpha(k)$、$\beta(\Pi(k))$ 表示输入、输出置换变换关系,即输入、输出序列分别表示成 a_i、$\overline{a}_i$ 和 b_i、$\overline{b}_i$。其中 $0\leqslant i\leqslant N/2-1$。

e)根据输入、输出置换变换关系,消去 a_i、$\overline{a}_i$,得到仅由 b_i、$\overline{b}_i$ 组成的布尔表达式。

f)设 $C=\{b_0,b_1,b_2,\cdots,b_{N/2-1}\}$,将 C 分成若干个等价类,满足条件:如果,当且仅当 b_i 和 b_j($b_i,b_j\in C$)互相依赖,那么 b_i 和 b_j 属于一类;否则,b_i 和 b_j 分属两个不同类。

g)令每个类中索引号最小的 b_k 取值为 0,依据同一类中 b_i 和 b_j 的相互关系确定每个 b_i($0\leqslant i\leqslant N/2-1$)相应的值为 0 或 1。根据输入、输出置换变换关系,确定每个 a_i($0\leqslant i\leqslant N/2-1$)相应的值为 0 或 1。从而确定左边和右边两级中每个开关元件的状态,a_i 表示左边一级各开关元件状态,b_i 表示右边一级各开关元件状态。其中 0 表示直通,1 表示交叉。

h)将输入、输出的比特序列表示成二进制代码形式,并表示成置换变换关系矩阵 Π。

i)将输入、输出的比特序列的二进制代码的最右一位由相应的布尔表达式 $\alpha(k)$、$\beta(\Pi(k))$ 替代,得到置换变换关系矩阵 Π^{ut1}。

j)将 Π^{ut1} 中的二进制代码循环右移一位,得到新的矩阵 Π^{ut2},并将 $\alpha(k)$、$\beta(\Pi(k))$ 相应值 0 或 1 带入。

k)根据 Π^{ut2} 中输入序列最左边一位的值是 0 或者是 1,将 Π^{ut2} 分为相应的两个置换变换关系矩阵 Π_0 和 Π_1,并去除其中所有二进制代码序列的最左边一位。这里 Π_0 和 Π_1 分别为上下两个 $N/2\times N/2$ 子网 $B_0(n-1)$ 和 $B_1(n-1)$ 中(图 4)的置换变换关系矩阵。如此循环,即可确定所有各级开关元件的状态。

举例如下:

仍用上例中的 8×8 的 Benes 网络实现 8×8 的置换。输入信号序列 0,1,2,3,4,5,6,7;输出 7,2,0,4,3,1,6,5。

1)$\Pi(0)=7,\Pi(1)=2,\Pi(2)=0,\Pi(3)=4,\Pi(4)=3,\Pi(5)=1,\Pi(6)=6,\Pi(7)=5$;

2)输入互斥对:$O_1=\{0,1\},O_2=\{2,3\},O_3=\{4,5\},O_4=\{6,7\}$;

3)输出互斥对:$P_1=\{7,2\}$,$P_2=\{0,4\}$,$P_3=\{3,1\}$,$P_4=\{6,5\}$;

4)用布尔表达式 $\alpha(k)$、$\beta(\prod(k))$表示输入、输出置换变换关系,即 $O_1=\{a_0,\bar{a}_0\}$,$O_2=\{a_1,\bar{a}_1\}$,$O_3=\{a_2,\bar{a}_2\}$,$O_4=\{a_3,\bar{a}_3\}$;$P_1=\{\bar{b}_3,b_1\}$,$P_2=\{b_0,b_2\}$,$P_3=\{\bar{b}_1\bar{b}_0\}$,$P_4=\{b_3,\bar{b}_2\}$;

5)由 $a_0=\bar{b}_3$,$\bar{a}_0=b_1$,得到 $b_3=b_1$;

由 $a_1=b_0$,$\bar{a}_1=b_2$,得到 $b_2=\bar{b}_0$;

由 $a_2=\bar{b}_1$,$\bar{a}_2=\bar{b}_0$,得到 $b_1=\bar{b}_0$;

由 $a_3=b_3$,$\bar{a}_3=\bar{b}_2$,得到 $b_3=b_2$;

6)由 5 可知,b_0、b_1、b_2、b_3 属于同一类;

7)令 $b_0=0$,则 $b_1=1$,$b_2=1$,$b_3=1$;$a_0=0$,$a_1=0$,$a_2=0$,$a_3=1$;从而得到最左边和最右边两级各开关元件的状态;

8)置换变换关系矩阵为

$$\Pi=\begin{pmatrix}0&1&2&3&4&5&6&7\\7&2&0&4&3&1&6&5\end{pmatrix}$$

二进制代码表示置换变换关系矩阵为

$$\Pi=\begin{pmatrix}000&001&010&011&100&101&110&111\\111&010&000&100&011&001&110&101\end{pmatrix}$$

9)最右一位由相应的布尔表达式 $\alpha(k)$、$\beta(\prod(k))$替代,得到置换变换关系矩阵为

$$\Pi^{out1}=\begin{pmatrix}00a_0&00\bar{a}_0&01a_1&01\bar{a}_1&10a_2&10\bar{a}_2&11a_3&11\bar{a}_3\\11\bar{b}_3&01b_1&00b_0&10b_2&01\bar{b}_1&00\bar{b}_0&11b_3&10\bar{b}_2\end{pmatrix}$$

10)将Π^{out1}中的二进制代码循环右移一位,得到新的矩阵为

$$\Pi^{out2}=\begin{pmatrix}a_000&\bar{a}_000&a_101&\bar{a}_101&a_210&\bar{a}_210&a_311&\bar{a}_311\\\bar{b}_311&b_101&b_000&b_210&\bar{b}_100&\bar{b}_000&b_311&\bar{b}_210\end{pmatrix}$$

将 $\alpha(k)$、$\beta(\prod(k))$相应的值 0 或 1 带入,得到矩阵:

$$\Pi^{out2}=\begin{pmatrix}000&100&001&101&010&110&111&011\\011&101&000&110&001&100&111&010\end{pmatrix}$$

11)根据Π^{out2}中输入序列最左边一位的值是 0 或者是 1,将Π^{out2}分为相应的两个置换变换关系矩阵Π_0和Π_1,并去除其中所有二进制代码序列的最左边一位。得到

$$\Pi_0=\begin{pmatrix}00&01&10&11\\11&00&01&10\end{pmatrix}\quad\Pi_1=\begin{pmatrix}00&01&10&11\\01&10&00&11\end{pmatrix}$$

则左右两级各开关元件状态及上下子网 $B_0(2)$和 $B_1(2)$置换变换关系如图 9 所示。

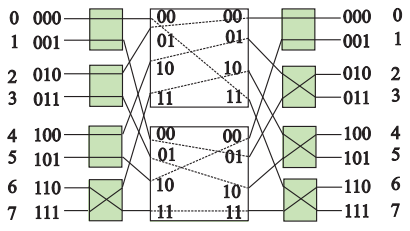

图9　左右两级各开关元件状态及上下子网 $B_0(2)$和$B_1(2)$置换变换关系

12)根据上下两个 4×4 子网 $B_0(2)$和 $B_1(2)$,得到新的互斥对;重复步骤 4)到 11),得到中间两级开关元件状态;最后得到最中间一级开关元件状态,如图 10 所示。

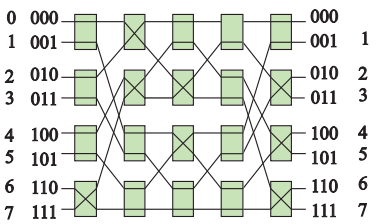

图10　并行算法实现在8×8Benes网络上的比特置换

3.3　算法分析

并行算法与二分法相比,具有网络结构简洁、变换效率高、易于控制等优点。而二分法具有算法简单、易于实现等特点。

在专用密码微处理器上实现比特置换单元时,用 VC 实现该两种算法,结果为 $2\log_2 N-1$ 列,$N/2$ 行的矩阵。其中:0 表示对应位置的开关元件状态为直通;1 表示对应位置开关元件状态为交叉。二分法和并行算法计算得到的控制信息矩阵分别由表 2 和 3 所示。

表 2　二分法实现控制信息矩阵

开关元件	左一级	左二级	中间级	右二级	右一级
1	0	0	0	1	0
2	1	0	1	1	0
3	1	0	1	1	0
4	1	0	0	0	1

表 3　并行算法实现控制信息矩阵

开关元件	左一级	左二级	中间级	右二级	右一级
1	0	1	0	0	0
2	0	1	1	0	1
3	0	1	1	0	1
4	1	0	0	0	1

4　结束语

本文基于 Benes 网络实现可重构比特置换网络,借鉴互连网络排序算法的研究成果,改进并实现了两种配置信息提取算法,即二分法和并行算法。这两种方法均能有效控制 Benes 网络中各开关元件的状态,实现各个待置换的比特在网络中正确选路,而不发生阻塞,且各有特点,在应用中可根据实际需要加以选择。

参考文献:

[1] SHI Z J. Bit permutation instructiong: architecture, implementation, and cryptographic properties[D]. New Jersey: Princeton University, 2004.

[2] ABRAMOWITZ M, STEGUN I A. Handbook of mathematical functions[S]. Washington DC: US Dept of Commerce and National Bureau of Standards, 1970.

[3] CORMEN T H, LEISERSON C E, RIVEST R L. Introduction to algorithms[M]. Cambridge: MIT Press, 1994.

[4] 曲英杰. 可重组密码逻辑的研究与设计[D]. 北京:北京科技大学,2004.

[5] BENES V E. Mathematical theory of connecting networks and telephone traffic[M]. New York: Academic Press,1965.

[6] ZHONG JI-ling. Upper bound analysis and routing in optical benes networks[D]. USA: Georgia State University University,2005.

[7] MCKEOWN N, MEKKITTIKUL A, ANANTHARAM V, *et al*. Achieving 100% throughput in an input-queued switch[J]. IEEE Trans on Commun, 1999, 47(8): 1260-1267.

[8] 金惠文. 现代交换原理[M]. 北京:电子工业出版社,2000.

[9] FENG T Y, SEO S W. A new routing algorithm for a class of rearrangeable networks[J]. IEEE Trans on Computers, 1994, 43(11):1270-1280.

[10] KOPPELMAN D M, A Y ORUC. A self-routing permutation network[J]. J Parallel Distrib Comput,1990, 10(2): 140-151.