HOKKAIDO UNIVERSITY

Title	d互換ブロックからなる置換ネットワークとその互換アルゴリズムについて
Author(s)	呉, 敦; 栗原, 正仁; 加地, 郁夫
Citation	北海道大學工學部研究報告, 126, 35-45
Issue Date	1985-05-31
Doc URL	https://hdl.handle.net/2115/41934
Type	departmental bulletin paper
File Information	126_35-46.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

北海道大学工学部研究報告
第126号（昭和60年）

Bulletin of the Faculty of Engineering,
Hokkaido University, No. 126 (1985)

d互換ブロックからなる置換ネットワークとその互換アルゴリズムについて

呉　　敦　栗原正仁　加地郁夫

（昭和59年12月27日受理）

A Permutation Network Consisting of d-Exchange Blocks and Its Exchanging Algorithm

Dun WU, Masahito KURIHARA and Ikuo KAJI

(Received December 27, 1984)

Abstract

The construction of a permutation network capable of exchanging arbitrarily two connections between its input/output terminals is described. The network consists of "d-exchange blocks" which are the sub-permutation networks that allow exchanging of any arlitrary two routes without changing the states of the other d-2 routes. The exchanging algorithm is given in terms of bipartite graphs.

The 4-exchange block is actually constructed, and the control unit is designed based on the state space and the state-transition rules to assure successive state transitions.

1. まえがき

並列処理システムを左右する置換ネットワークについては，これまでさまざまな提案がされており[1)~3)]，それらの経路設定アルゴリズムも導かれている。経路設定した置換ネットワークにおいて，任意の2つの入出力対応関係の互換を効率良く行うことは実用上重要であるが，これに関する研究はこれまでのところ報告されていない。本研究ではこのための置換ネットワークの構成と互換アルゴリズムを考察している。

N本の入力端子とN本の出力端子の間で任意の一対一対応の接続が可能なネットワークを置換ネットワークと呼ぶ（図1を参照）。置換ネットワークに対して，第i番目の入力をu_i，第j番目の出力をv_jとする（$1 \leq i, j \leq N$）。それらの置換を

$$\pi_N = \begin{pmatrix} u_1, & u_2, & \cdots, & u_N \\ v_1, & v_2, & \cdots, & v_N \end{pmatrix}$$

で表わす。π_Nの上の互換を$\tau=(v_i\ v_j)$とすると，τとπ_Nの積π_N^*は次式で表わされる。

$$\pi_N^* = \tau\pi_N$$

$$= (v_i\ v_j)\begin{pmatrix} u_1, & \cdots, & u_i, & \cdots, & u_j, & \cdots, & u_N \\ v_1, & \cdots, & v_i, & \cdots, & v_j, & \cdots, & v_N \end{pmatrix}$$

電気工学科　系統工学講座

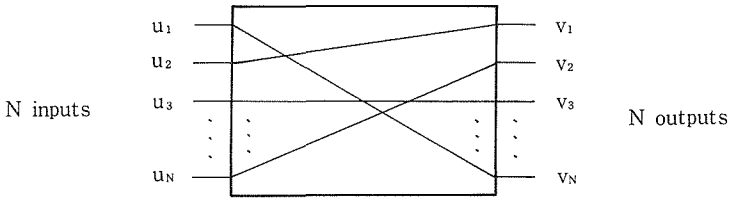


図1　置換ネットワーク

$$=\begin{pmatrix} u_1, \cdots, u_i, \cdots, u_j, \cdots, u_N \\ v_1, \cdots, v_j, \cdots, v_i, \cdots, v_N \end{pmatrix}$$

以下，こういう置換と互換を効率良く行う置換ネットワークの構成及びその上の互換アルゴリズムをグラフ理論に基づいて述べる。置換ネットワークの経路設定アルゴリズムの詳細については参考文献4)～7)等を参照されたい。

2. ネットワークの構成と諸定義

2.1 ネットワークの構成

入力 $N=d^k$，出力 $N=d^k$ とする時に，本ネットワークはBenesネットワークの構成法[3)~5)]に基づいて，$2d^{k-1}$ 個の互換性能を強化したd互換ブロック（dEB）とd個の d^{k-1} 入力 d^{k-1} 出力置換ネットワーク（d^{k-1} ネットワーク）で構成される。d^{k-1} ネットワークもdEBを用いて再帰的に構成する（図2）。ここで，dEBはd入力，d出力をもつ互換機能が強化された置換ネットワークであって，特にその上の互換に対する操作を行う時に，互換に関与しない他の残りの入出力間のd-2本の経路上のスイッチの状態が不変であるようなものをいう。すなわち，d-2本の経路上の物理的な接続は一瞬も切断されないようなものである。この構成法による $N=d^k$ ネットワークに含まれるdEBの個数を $F(k)$ とすると，

$F(1)=1$

$F(k)=2d^{k-1}+dF(k-1)$

より，$F(k)=d^{k-1}(2k-1)$ となる。

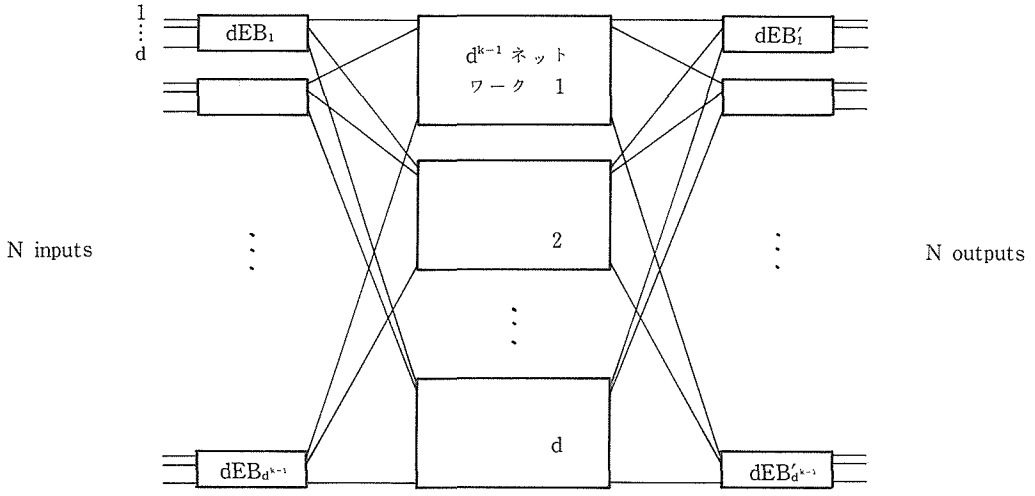


図2　$N=d^k$ ネットワークの構成

2.2 諸定義

〔定義1〕 置換ネットワークを構成するスイッチ群をある状態に設定することにおいて得られる入力 u_i から出力 v_j への導通路を経路 L_{ij} で表わす。

$N=d^k$ ネットワークに対しては，入出力置換関係は以下で述べる (d, d^{k-1}) 2部グラフで表現できる[3]。(d, d^{k-1}) 2部グラフは $2d^{k-1}$ 個のスーパーノードをもち，さらに各スーパーノードがd個のノードを有し，かつ各ノードの次数が1である正則2部グラフである(図3を参照)。(d, d^{k-1}) 2部グラフの左のスーパーノードiに入力側の dEB_i を対応させ，右のスーパーノードjに出力側の dEB_j' を対応させ，枝 $(a, b)_{1\,a\in i,\ b\in j}$ に dEB_i に含まれる入力 u_a から dEB_j' に含まれる出力 v_b への経路 L_{ab} を対応させる。

〔定義2〕 (d, d^{k-1}) 2部グラフの彩色問題において，$N=d^k$ ネットワークに対する (d, d^{k-1}) 2部グラフQはd色で彩色可能である[3),6),8)]。このd色を $c_1, c_2, \cdots, c_d$ とする(図3)。さらに，$d=lh$ の場合には，(d, d^{k-1}) 2部グラフQをh個の互いに独立な (l, d^{k-1}) 2部グラフに分解できる。こう分解したグラフを $G(c_1, c_2, \cdots, c_d)$ とする。こういう分解は唯一でない。特に，$d=2h$ の場合には二色 c_i, c_j に対するh個の分解された $G(c_j, c_j)$ を得る。明らかに，これらの $G(c_i, c_j)$ は常に幾つかの単純閉路で構成される。

彩色された (d, d^{k-1}) 2部グラフの枝の色 c_i については，色 c_i に関する対応する経路がすべて図2の第i番目の d^{k-1} ネットワークを通っていることを意味する(i=1, 2, …d)。以下では，経路の色とはその経路に対応する枝の色を意味することにする。

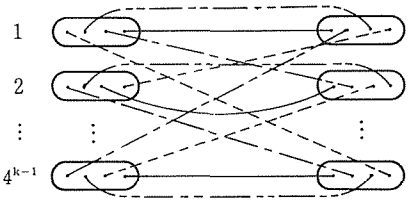
〔定義3〕 グラフの2つのノードxとyの間の最短距離を $D(x, y)$ とする。枝の長さを距離の単位とする。

$D(x, w)+D(z, y)\geq D(x, y)\geq 0$　　$w=z$，またはwとzが同じスーパーノードの上にある，

$D(x, y)=\infty$　　パスx, …, yが存在しない時，

$D(x, y)=1$　　枝 (x, y) が存在する時。

この最短距離 $D(x, y)$ は，置換 π_N から互換によって置換 π_N^* が確立する時，ネットワークに対してどの大きさの変更範囲が必要であるかを知るための重要な尺度である。



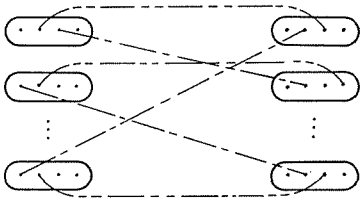
C_1 : ——— ——

C_2 : ——— ———

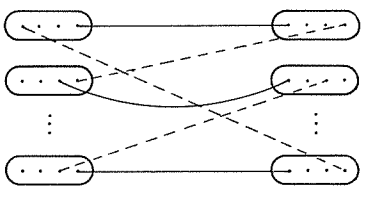
C_3 : ———————

C_4 : - - - - -

(a) $\pi_{N=4^k}$ に対する $(4, 4^{k-1})$ 2部グラフQ



(b) $(2, 4^{k-1})$ 2部グラフ $G(C_1, C_2)$



(c) $(2, 4^{k-1})$ 2部グラフ $G(C_3, C_4)$

図3　$(4, 4^{k-1})$ 2部グラフの分解 (d=4)

〔定義 4〕 ネットワークに対して互換操作を行う時，スイッチ切換によって生ずる経路の変更の数を P とする。P とネットワークの経路の総数 N との比率を波及率 R で表わす。

$R=P/N$。

3. 互換アルゴリズム

入力 $N=d^k$，出力 $N=d^k$ の図 2 に示したネットワークに対して，任意の互換操作を行うアルゴリズムの基本的な考え方を述べる。すなわち，互換 $\tau=(v_a\ v_b)$によって影響を受ける経路 L_{ga} と L_{fb} に着目すると，次の 3 つのケースに分類できる。

〈ケース 1〉 経路 L_{ga}，L_{fb} が d^k ネットワークの入力側または出力側にある同じ dEB を通っている場合，スイッチ切換操作はこの dEB 内に局部化される。置換 π_N^*（$=\tau\pi_N$）はこの dEB だけで実現できる。L_{ga} と L_{fb} 以外の経路は切断されない。また，経路 L_{ga} と L_{fb} もこの dEB 以外のスイッチの接続を変更する必要がない。(d, d^{k-1}) 2 部グラフを用いて説明すると，枝 (g, a) と枝 (f, b) がある同じスーパーノードに接続していることになる。$D(g, f)=0$ または $D(a, b)=0$ である。この時，波及率は 0 になる（P=0）。

〈ケース 2〉 経路 L_{ga} の色 c_i と経路 L_{fb} の色 c_j が異なる場合，さらに 3 つの場合に分ける。すなわち，(d, d^{k-1}) 2 部グラフを〔定義 2〕に従って，色 c_i と色 c_j に分解した部分グラフ $G(c_i, c_j)$ において，

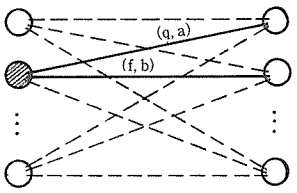


図 4　ケース 1

i）　$D(g, f)=2$ または $D(a, b)=2$ となる場合，

ii）　枝 (g, a) または枝 (f, b) は長さ 2 の単純閉路の上にある場合，この時，$D(g, f)=D(a, b)=\infty$，

iii）　i）と ii）以外の場合である。

i）　分解した $G(c_i, c_j)$ において，枝 (g, a) と枝 (f, b) は同じ単純閉路の上にあり，かつ $D(g, f)=2$ または $D(a, b)=2$ となる場合には，$D(g, f)=2$ または $D(a, b)=2$ に関するパスを p とすると〔例えば，図 5 の場合，$D(a, b)=2$，$p=(a, d, y, x, e, b)$〕，パス p のノード a，b とノード g，f との接続関係を互換によって変更すると，ノード a, b, g, f 以外の接続関係は不変である。その結果，得られるグラフを色 c_i，c_j で再彩色する。ただし，この際パス p 及び新しく生じた枝 (f, a) と (g, b) 以外の枝の色は変えずに再彩色するものとする。

再彩色した後のパス p を p′ とする。パス p′ 上のスーパーノード（図 5 のスーパーノード v_2, u_2, v_3）に対応する入出力側の dEB のスイッチを互換操作する（経路の色をかえることに対応する）。枝 (g, b) に対応する経路 L_{gb} 及びそれと同じ色のパス p′ 上の枝〔図 5 の (d, y)〕に対応する経路との互換操作は色 c_i に対応する d^{k-1} ネットワーク i での互換問題に帰着される。また，同様に経路 L_{fa} と同じ色のパス p′ 上の枝〔図 5 の (x, e)〕に対応する経路との互換操作は色 c_j に対応する d^{k-1} ネットワーク j での互換問題に帰着される。図 5 に示す色 c_1 と色 c_2 に対応する二つの d^{k-1} ネットワークでの互換を行うことは置換群で次の式で表現される。

$$c_1:\quad \pi_{d^{k-1}}=\begin{pmatrix}u_1, u_2, u_3, \cdots\\ v_2, v_3, v_4, \cdots\end{pmatrix}\longrightarrow \pi^*_{d^{k-1}}=\begin{pmatrix}u_1, u_2, u_3, \cdots\\ v_3, v_2, v_4, \cdots\end{pmatrix},$$

$$c_2:\quad \pi_{d^{k-1}}=\begin{pmatrix}u_1, u_2, u_3, \cdots\\ v_1, v_2, v_3, \cdots\end{pmatrix}\longrightarrow \pi^*_{d^{k-1}}=\begin{pmatrix}u_1, u_2, u_3, \cdots\\ v_1, v_3, v_2, \cdots\end{pmatrix}。$$

ここで，スーパーノード u_i，v_j はそれぞれ d^{k-1} ネットワークの入力側 dEB_i と出力側の dEB_j に

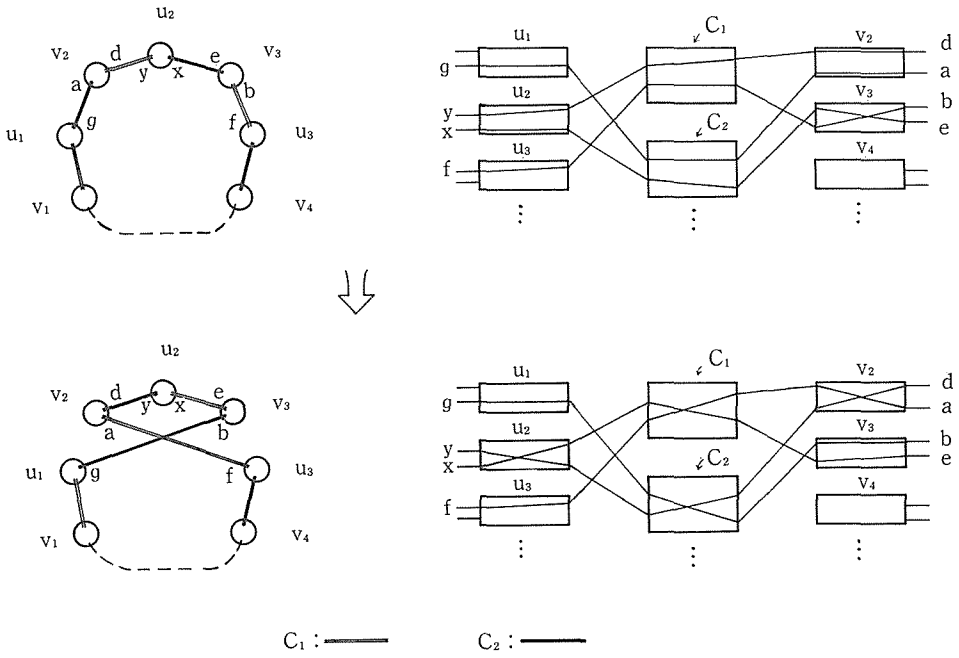


図5　ケース2の(i)

対応する。

ii)　経路 L_{ga} と経路 L_{fb} に対応する枝(g, a)と枝(f, b)が異なる単純閉路の上にあり，かつそのうち少なくとも一つの単純閉路の長さが2に等しい場合である〔図6の場合，枝(g, a)と枝(x, d)での単純閉路である〕。この場合，二つの異なる単純閉路は互換によって一つの単純閉路になる。これを色 c_i, c_j で再彩色する。ただし，この際に長さ2の単純路上の枝及び新しく生じた枝(g, b)と(f, a)以外の枝の色は変えずに再彩色するものとする。ノードa, b, g, f以外のノードの接続関係は不変である。

その結果，長さ2の単純閉路上のスーパーノード(図6の u_1, v_1)に対応する入出力側の2つのdEBのスイッチを互換操作する。さらに，生じた枝(g, b)と(f, a)に対応する経路 L_{gb} と経路 L_{fb} との互換操作は枝(g, b)と(f, a)の色(図6では c_2)に対応する d^{k-1} ネットワークのみの互換問題に帰着される。図6に示す色 c_1 と色 c_2 に対応する二つの d^{k-1} ネットワークにおいて，その互換を行うことは置換群で次の式で表現される。明らかに，色 c_1 に対応する置換群は不変である。

$$c_1 : \quad \pi_{d^{k-1}} = \begin{pmatrix} u_1, u_2, \cdots \\ v_1, \cdots \end{pmatrix} \longrightarrow \pi^*_{d^{k-1}} = \begin{pmatrix} u_1, u_2, \cdots \\ v_1, \cdots \end{pmatrix},$$

$$c_2 : \quad \pi_{d^{k-1}} = \begin{pmatrix} u_1, u_2, \cdots \\ v_1, v_2, \cdots \end{pmatrix} \longrightarrow \pi^*_{d^{k-1}} = \begin{pmatrix} u_1, u_2, \cdots \\ v_2, v_1, \cdots \end{pmatrix}。$$

iii)　この場合，互換によって新しく生じたG′(c_i, c_j)を色 c_i, c_j で再彩色する。その結果は再彩色されたG′(c_i, c_j)によって d^k ネットワークの入出力側のdEBのスチッチを互換操作し，色 c_i, c_j

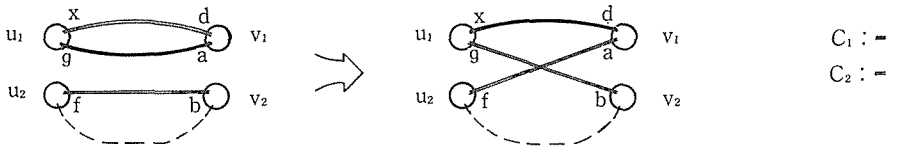


図6　ケース2の(ii)

に対応する二つの d^{k-1} ネットワークを再経路設定する。この時，d^k ネットワークの他の d-2 色の経路は切断されないので，波及率 $R=2(1/d-1/N)$ になる（$P=2(d^{k-1}-2)$]。

〈ケース 3〉　経路 L_{ga} と経路 L_{fb} が同色の場合には，この色に対応する d^{k-1} ネットワークでの互換問題に帰着する。入出力側の dEB のスイッチ操作は不要である。

4. 4 互換ブロック（4 EB）の実現

4. 1　4EB の構成

ネットワークを経路設定することは，すでての dEB の状態を決めることを意味する。こういう経路を設定した dEB に対して，任意の二つの入出力間の互換操作を行うと同時に他の d-2 本の経路を切断しないように実現したい。このため，4EB を図 7 のように考えた。これは 8 個の 2 ユニットからなり，$\binom{4}{2}=6$ 通りのうちの任意の互換を一つの 2 ユニットだけの状態変化で実現できる。このことについて述べる。

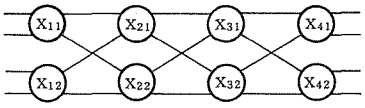
4EB の基本素子である 2 ユニットは図 7(b)に示すような 2 つの状態(reset-state と set-state)を有するスイッチである。この 2 ユニットの状態を表わすのに 1 ビットを使用すれば，4EB の状態は 8 ビットのベクトル A で表現できる。

$$A=\{a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, a_{32}, a_{41}, a_{42}\}$$

$$a_{ij}=\begin{cases}1 & x_{ij}:\text{set}-\text{state},\\ 0 & x_{ij}:\text{reset}-\text{state}。\end{cases}$$

2 ユニット x_{k1}, x_{k2} で構成される部分回路を 4EB の第 k 段(S_k)と呼ぶ。

4EB の 4 本の経路の中の任意の 2 つを，1 つの 2 ユニットのみの状態反転で互換したい。そのためには，4EB 中でどの 2 本の経路も，少なくとも 1 つの同一の 2 ユニットを経由していることが必要であり，かつ十分である。例えば，図 8 のような状態のときには，これは不可能である。さ



（a）4 EB の構成

（b）2 ユニットの 2 つの状態

図 7　4EB の構成と 2 ユニット

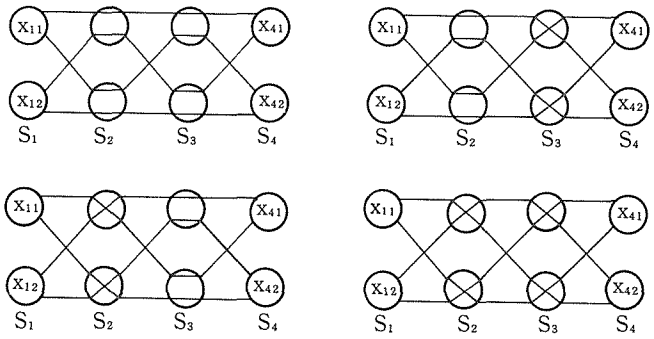


図 8

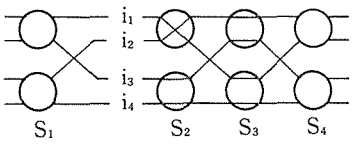
らに，互換操作後の4EBの状態もまたこの性質を満たしていることが必要である。これらの条件を満たす状態の集合は2^8通りのうち192通りに限定される。これらは第2段S_2と第3段S_3のスイッチの状態に着目して(表1のように)12通りの場合に分類できる。その分類に従って，その一覧を図9に示す。この図で$(i_2, i_3) : x_{31} \rightarrow$Case 2は経路$i_2$と$i_3$を互換するには，スイッチ$x_{31}$を反転すればよく，その結果はCase 2になることを表わしている。その他についても意味は同様である。さらに，互換によりこれらの12通りのスイッチ•パターンの間でおこる状態推移の様子を図10に示した。枝の上に示したラベル$(i_2\, i_3) : x_{31}$は上と同様の意味である。すなわち，枝(2, 1)上のラベル$(i_2\, i_3) : x_{31}$はCase 2またはCase 1内の状態のもとで入力値$(i_2\, i_3)$を与えたとき，出力としてスイッチ•ポジションx_{31}が得られることを表わしている。

第1表

Case	x_{21}	x_{22}	x_{31}	x_{32}
1	1	0	0	0
2	1	0	1	0
3	1	0	0	1
4	1	0	1	1
5	0	1	0	0
6	0	1	1	0
7	0	1	0	1
8	0	1	1	1
9	0	0	1	0
10	0	0	0	1
11	1	0	0	0
12	1	1	0	1

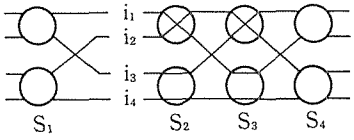
4. 2　4EBの互換ハードウェアの構成

Case 1



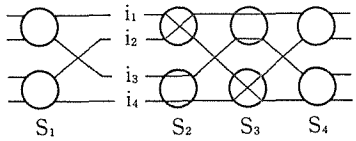
$(i_1\, i_3) : x_{11} \rightarrow$ Case 1 ;
$(i_2\, i_4) : x_{12} \rightarrow$ Case 1 ;
$(i_1\, i_2) : x_{41} \rightarrow$ Case 1 ;
$(i_3\, i_4) : x_{42} \rightarrow$ Case 1 ;
$(i_2\, i_3) : x_{31} \rightarrow$ Case 2 ;
$(i_1\, i_4) : x_{32} \rightarrow$ Case 3 。

Case 2



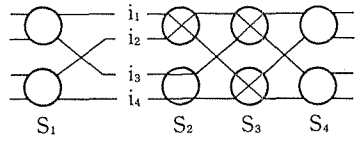
$(i_1\, i_3) : x_{11} \rightarrow$ Case 2 ;
$(i_2\, i_4) : x_{12} \rightarrow$ Case 2 ;
$(i_1\, i_2) : x_{21} \rightarrow$ Case 9 ;
$(i_3\, i_4) : x_{22} \rightarrow$ Case11 ;
$(i_2\, i_3) : x_{31} \rightarrow$ Case 1 ;
$(i_1\, i_4) : x_{32} \rightarrow$ Case 4 。

Case 3



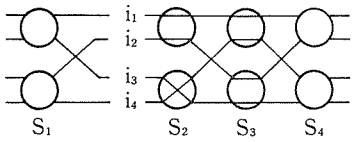
$(i_1\, i_3) : x_{11} \rightarrow$ Case 3 ;
$(i_2\, i_4) : x_{12} \rightarrow$ Case 3 ;
$(i_1\, i_2) : x_{21} \rightarrow$ Case10 ;
$(i_3\, i_4) : x_{22} \rightarrow$ Case12 ;
$(i_2\, i_3) : x_{31} \rightarrow$ Case 4 ;
$(i_1\, i_4) : x_{32} \rightarrow$ Case 1 。

Case 4



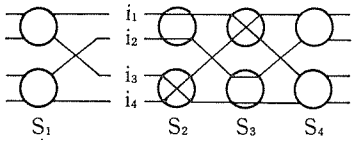
$(i_1\, i_3) : x_{11} \rightarrow$ Case 4 ;
$(i_2\, i_4) : x_{12} \rightarrow$ Case 4 ;
$(i_1\, i_2) : x_{42} \rightarrow$ Case 4 ;
$(i_3\, i_4) : x_{41} \rightarrow$ Case 4 ;
$(i_2\, i_3) : x_{31} \rightarrow$ Case 3 ;
$(i_1\, i_4) : x_{32} \rightarrow$ Case 2 。

Case 5



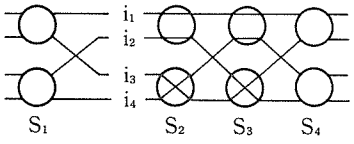
$(i_1\ i_3) : x_{11} \rightarrow$ Case 5 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case 5 ;
$(i_1\ i_2) : x_{41} \rightarrow$ Case 5 ;
$(i_3\ i_4) : x_{42} \rightarrow$ Case 5 ;
$(i_2\ i_3) : x_{32} \rightarrow$ Case 7 ;
$(i_1\ i_4) : x_{31} \rightarrow$ Case 6 。

Case 6



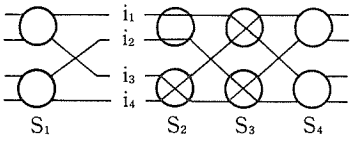
$(i_1\ i_3) : x_{11} \rightarrow$ Case 6 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case 6 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case11 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case 9 ;
$(i_2\ i_3) : x_{32} \rightarrow$ Case 8 ;
$(i_1\ i_4) : x_{31} \rightarrow$ Case 5 。

Case 7



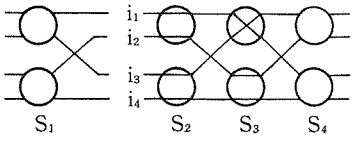
$(i_1\ i_3) : x_{11} \rightarrow$ Case 7 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case 7 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case12 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case10 ;
$(i_2\ i_3) : x_{32} \rightarrow$ Case 5 ;
$(i_1\ i_4) : x_{31} \rightarrow$ Case 8 。

Case 8



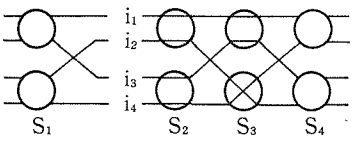
$(i_1\ i_3) : x_{11} \rightarrow$ Case 8 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case 8 ;
$(i_1\ i_2) : x_{42} \rightarrow$ Case 8 ;
$(i_3\ i_4) : x_{41} \rightarrow$ Case 8 ;
$(i_2\ i_3) : x_{32} \rightarrow$ Case 6 ;
$(i_1\ i_4) : x_{31} \rightarrow$ Case 7 。

Case 9



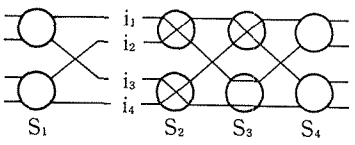
$(i_1\ i_3) : x_{11} \rightarrow$ Case 9 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case 9 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case 2 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case 6 ;
$(i_2\ i_3) : x_{41} \rightarrow$ Case 9 ;
$(i_1\ i_4) : x_{42} \rightarrow$ Case 9 。

Case 10



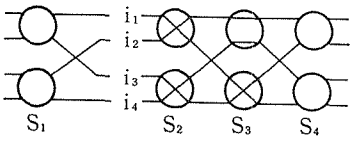
$(i_1\ i_3) : x_{11} \rightarrow$ Case10 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case10 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case 3 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case 7 ;
$(i_2\ i_3) : x_{42} \rightarrow$ Case10 ;
$(i_1\ i_4) : x_{41} \rightarrow$ Case10 。

Case 11



$(i_1\ i_3) : x_{11} \rightarrow$ Case11 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case11 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case 6 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case 2 ;
$(i_2\ i_3) : x_{42} \rightarrow$ Case11 ;
$(i_1\ i_4) : x_{41} \rightarrow$ Case11 。

Case 12



$(i_1\ i_3) : x_{11} \rightarrow$ Case12 ;
$(i_2\ i_4) : x_{12} \rightarrow$ Case12 ;
$(i_1\ i_2) : x_{21} \rightarrow$ Case 7 ;
$(i_3\ i_4) : x_{22} \rightarrow$ Case 3 ;
$(i_2\ i_3) : x_{41} \rightarrow$ Case12 ;
$(i_1\ i_4) : x_{42} \rightarrow$ Case12 。

図9

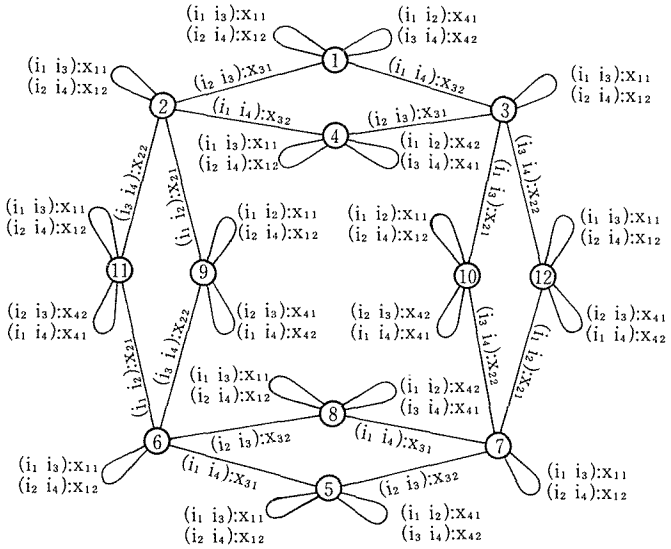


図10　状態推移グラフ

前述の原理に基いて，ハードウェア構成するためには，次の構成要素が必要である。

（ｉ）　状態記憶レジスタ

（ii）　互換要求信号（実入力）を内部入力信号に直す回路

（iii）　状態推移を実現する組合せ回路と遅延回路および出力回路

（ｉ）　状態記憶用レジスタとしては，図10の状態推移グラフに従って状態推移を行わさせるために，状態値を8ビットレジスタ $A=\{a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, a_{32}, a_{41}, a_{42}\}$ を用いる

（ii）　4項目に関する互換 τ を4ビット・ベクトル $U=(u_1, u_2, u_3, u_4)$ で表示する。一般に，$\tau=(v_i\ v_j)$ は $u_i=u_j=1$，その他0で表わす。このベクトルを互換要求または実入力と言う。Aの先頭の2ビット a_{11}, a_{12} から状態推移グラフ（図10）の入力信号 $1=(i_1, i_2, i_3, i_4)$ を導出する回路は表2で定義されるブール関数を実現する組合せ回路である。ただし，$i_\mu=i_\nu=1$ その他0はオートマト

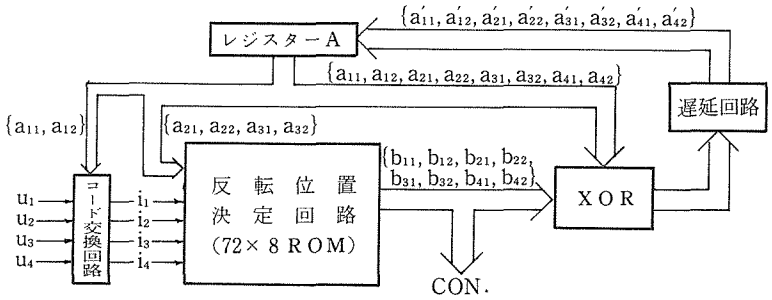


図11　ハードウェアの構成

表2　$\vec{u} \rightarrow \vec{I}$ の交換表

u_1 u_2 u_3 u_4	a_{11} a_{12} 0 0	0 1	1 0	1 1
1 1 0 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
1 0 1 0	1 1 0 0	1 0 0 1	0 1 1 0	0 0 1 1
1 0 0 1	1 0 0 1	1 1 0 0	0 0 1 1	0 1 1 0
0 1 1 0	0 1 1 0	0 0 1 1	1 1 0 0	1 0 0 1
0 1 0 1	0 0 1 1	0 1 1 0	1 0 0 1	1 1 0 0
0 0 1 1	0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1

表3：

入力(address)								出力(data)							
a_{21}	a_{22}	a_{31}	a_{32}	i_1	i_2	i_3	i_4	b_{11}	b_{12}	b_{21}	b_{22}	b_{31}	b_{32}	b_{41}	b_{42}
1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
1	1	0	0												
1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1

ンへの入力が $(i_\mu\ i_\nu)$ であることを意味する。

(iii)　図9のCase情報はAレジスタの $(a_{21}, a_{22}, a_{31}, a_{32})$ ビット位置に格納されている。この情報とベクトル $1=(i_1, i_2, i_3, i_4)$ から反転する2ユニットを決定する情報ベクトル $B=(b_{11}, b_{12}, b_{21}, b_{22}, b_{31}, b_{32}, b_{41}, b_{42})$ を表3の真理値に基いて組合せ回路で実現する。但し，Bは1カ所だけ1をもつベクトルで，$b_{ij}=1$ は2ユニット x_{ij} を反転することを表わす。これがスイッチへの制御信号となる。さらに，レジスタAの内容とベクトルBの内容のXORをとることによって，新しいAの値が得られる。(Bによって定まる1カ所だけのビットを反転することに注意)

5. むすび

経路設定した置換ネットワークに対して，ネットワークの構成とその上の互換アルゴリズムの

研究を行なった。即ち，本研究で以下の結果を得た。

1) 任意の2つの入出力接続関係の互換を効率良く行うために，d互換ブロック(dEB)からなる置換ネットワークの構成と互換アルゴリズムを考察した。

2) 4EB互換ブロック（4EB）を具体的に考察し，その互換操作のハードウェア制御方式を示した。

今後の研究課題としては，ネットワークのスイッチの数が増えないように，さらに低い波及率をもつ置換ネットワークの開発とさらに効率良い互換アルゴリズムの研究である。

参考文献

1) Broomell, G. and Heath, J.: Computing Surveys, **15** (1983), 2, p. 95～133.
2) Waksman, A.: J. ACM, **15** (1968), 1, p. 159～163.
3) 五十嵐善英，桜沢庄治：情報処理，**25** (1984), 1, p. 13～18.
4) Lev, G. Pipperger, N. and Valiant, L. G.: IEEE Trans. on Computers, **C-30** (1981), 2, p. 93～100.
5) D. Nassimi and S. Sahni: IEEE Trans. on computers, **C-30** (1982), 2, p. 148～154.
6) Sakurazawa, S. and Igarashi, Y.: Trans. IECE Japan, **E64** (1981), 4, p. 235～242.
7) D. Nassimi and S. Sahni: IEEE Trans. on Computers, **C-30** (1981), 5, p. 332～340.
8) H. N. Gabow.: Int. J. Comput. Inform. Sci., **5** (1976), p. 343～355.