

# 多値多変数回路の 一致検出関数による合成

並木淳台 平山 博  
(早大理工学部)

1. 関数の定義とその性質
2.  $\phi(i)$  多値変数の隣接性
3. 隣接性と回路網の簡略化との関係
4. 多値多変数回路網合成法
5. 関数  $F$  の多値関数の出力の扱い
6. 置換素の導入
7. 多値NAND数と回路網レベル数について

## 多値多変数回路の一致検出関数による合成

並木 淳治 平山 博

(早大理工学部)

概要 各多値論理素子を見ていくと多値比較器が一般的である。従って展開関数として一致検出関数の(1)が適合している。そこでまずそれらの関数の定義と性質を述べ  $\phi$  の符号空間の隣接性を調べ それと回路網簡略化との関係を明らかにし その上で回路網合成法を述べ さらに各符号空間の状態が担っている多値出力の扱いによって様々な回路網の簡略が成されることを示す。また必要NAND数と回路網レベル数との関係にも付記して終る。

## 1. 関数の定義とその性質

多値変数のとり値として  $0, 1, 2, \dots, M$  の整数を扱う。

$$A + B = \max(A, B) \quad A \cdot B = \min(A, B) \quad \bar{A} = M - A$$

•, + の優先順位は • が優先する。#, - は一般代数の和差を表わし、定数を規則的に変更するのに用いる。まず

$x, y$  の2変数入力関数  $f(x; y, 0, M)$  を以下の様に定義する。

$$f(x; y, 0, M) \equiv \begin{cases} x > y & ; 0 \\ \text{other} & ; M \end{cases} \quad \bar{f} = \begin{cases} ; M \\ ; 0 \end{cases}$$

関数  $f$  の性質を以下(I)~(III)まで記す。

$$(I) f(0; x, 0, M) = f(x; M, 0, M) = M$$

$$\Rightarrow \frac{f(0; x \ 0, M)}{f(x; M \ 0, M)} \cdot f(x; y \ 0, M) = f(x; y \ 0, M)$$

$$(II) \overline{f}(x; y \ 0, M) = f(x; y \ M, 0)$$

$$(III) \overline{f}(A; x \ 0, M) = f(x; A^{-1} \ 0, M) \quad \text{----- (a)}$$

$$\overline{f}(x; A \ 0, M) = f(A^{-1}; x \ 0, M) \quad \text{----- (b)}$$

次に基底関数  $\phi(i)$  を以下の様子上定義する。

$$\phi(i)(x) \equiv \begin{cases} x=i & ; M \\ x \neq i & ; 0 \end{cases} \quad \overline{\phi(i)(x)} = \begin{cases} ; 0 \\ ; M \end{cases}$$

関数  $\phi(i)$  の性質を以下 (N) ~ (XI) まで記す。

$$(N) \phi(i)(x) = f(i; x \ 0, M) \cdot f(x; i \ 0, M)$$

$$\begin{aligned} \overline{\phi(i)(x)} &= \overline{f(i; x \ 0, M)} + \overline{f(x; i \ 0, M)} \\ &= f(x; i^{-1} \ 0, M) + f(i^{-1}; x \ 0, M) \end{aligned}$$

$$(V) \phi(i)(x) \cdot \phi(j)(x) = 0 \quad i \neq j \quad \text{----- (a)}$$

$$\overline{\phi(i)(x)} + \overline{\phi(j)(x)} = M \quad \text{----- (b)}$$

$$(VI) \sum_{i=0}^M \phi(i)(x) = M \quad \text{----- (a)}$$

$$\prod_{i=0}^M \overline{\phi(i)(x)} = \prod_{i=0}^M \phi(i)(x) = 0 \quad \text{----- (b)}$$

$$(VII) \prod_{\substack{j \text{ 互換素} \\ j \neq i}} \overline{\phi(j)(x)} = \sum_{\substack{j \text{ 互換素} \\ j \neq i}} \phi(j)(x)$$

$$(VIII) P_i = P_i + P_i \cdot P_j \quad P_i \text{ は置換素と言われれ内容を示す。}$$

$$P \equiv \begin{pmatrix} \lambda \text{力}; 0, 1, 2, \dots, M \\ \text{出力}; A_0, A_1, A_2, \dots, A_M \end{pmatrix} \quad \lambda_i \text{ は } 0 \sim M \text{ の任意の値}$$

$$(IX) \prod_i (\overline{\phi(\omega)\{x\}} + P_i) = \sum_j P_j \cdot \phi(\omega)\{x\}$$

$$(X) \phi(\omega)\{x\} \cdot \overline{\phi(\omega)\{x\}} = \phi(\omega)\{x\}$$

$$(XI) (\phi(\omega)\{x\} + \phi(\omega)\{x\}) \cdot (\phi(\omega)\{x\} + \phi(\omega)\{x\}) \\ = \phi(\omega)\{x\} + \phi(\omega)\{x\} \cdot \phi(\omega)\{x\}$$

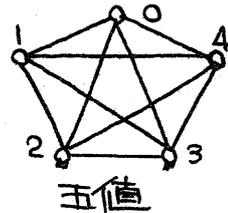
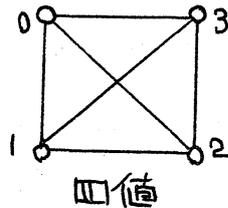
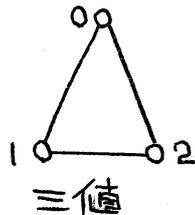
## 2. $\phi(\omega)\{x\}$ 多値変数の隣接性

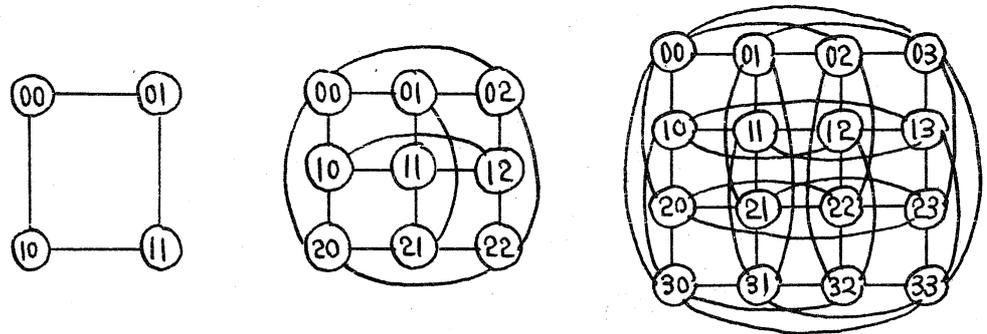
多値変数  $x$  は  $\phi(\omega)\{x\}$  の形で展開されるので  $\{x\}$  のかわりに  $\phi(\omega)\{x\}$  を考える。

$$\phi(\omega)\{x\} \equiv \begin{cases} x=i & ; M \\ x \neq i & ; 0 \end{cases} \quad \overline{\phi(\omega)\{x\}} \equiv \begin{cases} ; 0 \\ ; M \end{cases}$$

であるので  $x=i$  以外の  $0 \sim M$  の全ての値は  $i$  に対して等距離にある。よって一つの変数が  $n$  個の値を取ってもそれらの間の距離はそれによつて増すことはない。今現われたい距離を一距離と言うことにする。そして多値変数組の隣接性とはこのことを言うのが適当であろう。だから  $0 \sim i \sim M$  の値を取る変数  $x$  のある値  $i$  に隣接関係を持つ値の数は  $M$  個ある。例えば  $\phi(\omega)\{x\}$  5 値  $(0, 1, 2, 3, 4)$  について言うと  $2$  に隣接する値は  $1, 3$  だけではなく  $2$  以外の値、すなわち  $0, 1, 3, 4$  の 4 個であることに注意していただきたい。そこで隣接関係をグラフ表示すると以下の様になる。

多値変数  $x$   
の  
隣接関係  
<図 1 a>





二値 三値 四値  
多値2変数組の隣接性 <図1 b>

これらのグラフよりも わかる様に多値2変数のある状態  $(i_1, j_1)$  から他の状態  $(i_2, j_2)$  への距離は最大2距離であり距離の増大は変数の数にのみ関係することがかかる。よって以下のことが言える。

$\phi(x)$  の多値  $n$  変数組の取りうる状態間の各距離の最長値は多値の程度には無関係で 変数の数  $n$  にのみ関係し その最長距離は  $n$  である。

### 3. 隣接性と回路網の簡略化との関係

回路網の簡略化として ニニでは演算素子の数と それらに出入する Fan in, Fan out の数を問題にしている。今  $(a, b)$  を  $a \cap b$  の状態を表示するものとすると 各々が1距離にある各状態は以下の様に書ける。  $(\phi(x) \cap x_0)$  を以下便宜上  $\phi(x)$  と書く。  
 $(k, \phi(0) \cap x_0), (k, \phi(1) \cap x_0), (k, \phi(2) \cap x_0), \dots \dots \dots (k, \phi(M) \cap x_0)$   
 ニニで  $k$  は  $\phi(x)$  以外の変数の表わす ある状態を示している。  
 これらの状態のうち同一出力を担っているものは別けておく

必要がないので 二れらを一緒にする。今  $(k, \phi(1)), (k, \phi(2))$  がそうであるとすると各々が一距離にある各状態の組は一組減、て以下の様になる。

$(k, \phi(0)), (k, \begin{bmatrix} \phi(1) \\ \phi(2) \end{bmatrix}), (k, \phi(3)), \dots (k, \phi(i)), \dots (k, \phi(M))$   
 そにて ある状態を表わしている  $(a, b)$  が一演算素子に対応しているとするに一素子減、をことにする。ところが  $\begin{bmatrix} \phi(1) \\ \phi(2) \end{bmatrix}$  はどうなっているのであろうか。  $\begin{bmatrix} c \\ d \end{bmatrix}$  はこの場合  $C \cup d$  の状態を表わしているので  $(a, b)$  とは明らかに同一のレベルでは考えられない。そにて (Ⅳ)式により  $\begin{bmatrix} \phi(1) \\ \phi(2) \end{bmatrix}$  を  $(\overline{\phi(0)}, \overline{\phi(3)}, \overline{\phi(4)}, \dots \overline{\phi(M)})$  で表わせば  $(a, b)$  と同一のレベルで扱えることになり各状態は以下の様になる。

$(k, \phi(0)), (k, \overline{\phi(0)}, \overline{\phi(3)}, \overline{\phi(4)}, \dots \overline{\phi(M)}), \dots (k, \phi(i)), \dots (k, \phi(M))$   
 よって  $F_{an\ in}$  を増すことにより  $\begin{bmatrix} \phi(1) \\ \phi(2) \end{bmatrix}$  を表示する演算素子の増加をさせられる。この様に隣接関係にある状態のうち同一出力を担うものが存在するならば それは  $F_{an\ in}$  を増すことにより演算素子を減らすことができる。

今度は同一出力を持つ状態が  $\{(M+1)/2\}$  個以上ある場合はどうであらうか。前同様 演算素子はもちろん その分だけ減る訳である。さて  $F_{an\ in}$  の方はどうであらうか。今例として  $(0, 1, 2, 3, 4)$  の5値において  $(k, \phi(1)), (k, \phi(2)), (k, \phi(4))$  が同一出力を持つとすると それらは一つにまとめられ

て  $(k, \begin{bmatrix} \phi(1) \\ \phi(2) \\ \phi(4) \end{bmatrix})$  とする。すると各々は  $(k, \phi(0))$ ,  $(k, \begin{bmatrix} \phi(0) \\ \phi(2) \\ \phi(4) \end{bmatrix})$ ,  $(k, \phi(3))$  となり演算素子は5個から3個になる。さらに  $(k, \begin{bmatrix} \phi(1) \\ \phi(2) \\ \phi(4) \end{bmatrix})$  の Fan in は  $\begin{bmatrix} \phi(0) \\ \phi(2) \\ \phi(4) \end{bmatrix}$  が  $(\phi(0), \phi(3))$  とまり3本から2本に減らすことができる。よって以下の様と言える。

隣接関係にある状態のうち 同一出力を持つものが  $m$  個あって  $m < \frac{M+1}{2}$  ならば演算素子は  $(m-1)$  個減るが同一出力の状態を担う素子への Fan in は増す。  
 $m > \frac{M+1}{2}$  である場合は 演算素子が  $(m-1)$  個減りさらに同一出力の状態を担う素子への Fan in も減る。

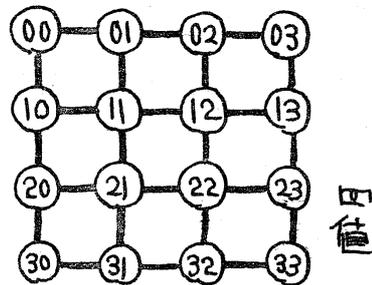
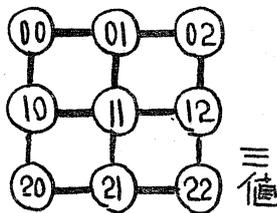
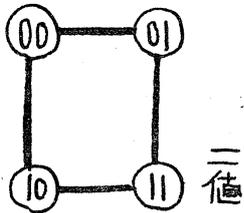
今2値について考えると  $(M+1)/2 = 1$  で  $m > 1$ , 3値について考えると  $m > 1.5$ .  $M$  の最小値は2であり上定理のうちの  $m > \frac{M+1}{2}$  を常に満足している。これより以下のことが言える。

2値, 3値に限って同一出力の隣接は 必ず回路を簡単にする。

上の理由によって多値と一こと言っても 3値とそれ以外とは この事情によって区別されるべきである。

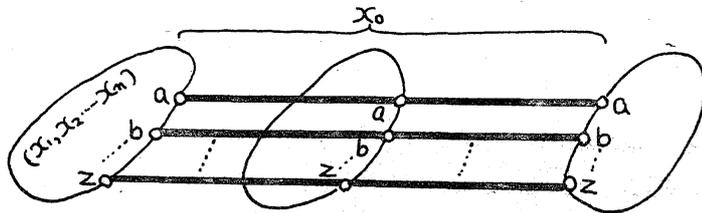
さて後々のために煩雑な〈図1〉の表示を改めて新しい表示を考えよう。そこで直線上の各状態はそれ自身を除いて各々一距離の状態であると考えることによって〈図2〉の様になる。ここで図形を構成する直線を以下の様に定義する。一直線とは  $n$  変数組状態において同一の  $(n-1)$  変数組状態を結ん

でいるものを言う。——



多値2変数組の隣接性表示<図2>

するとM値n変数はいつも以下の様に表記できる。



M値n変数組状態図における $P_{ex_0}$ <図4>

ここで  $a, b, \dots, z$  は  $x_0$  を除いた変数組のとり状態を示している。今現われたい直線部分は  $P_{ax_0}, P_{bx_0}, \dots, P_{zx_0}$  に対応している部分である。そして それらのうち出力が同一の順番で直線上に並んでいるものは共通置換素で置換えられ さらにそれらが(直線同志が)2距離以上はなれているものはより有用で それは新しい回路網簡略の余地を示している。

#### 4. 多値多変数回路網合成

まず以下の様な  $F, F'$  なる関数を考えてみよう。

$$(XV) F = f_M + (M-1) \cdot f_{M-1} + (M-2) \cdot f_{M-2} + \dots + 1 \cdot f_1 \quad \dots (a)$$

$$F' = ((M-1) + f'_{M-1})((M-2) + f'_{M-2}) \cdot \dots \cdot f'_0 \quad \dots (b)$$

ここで  $f_j$  は  $(x_0, x_1, \dots, x_n)$  の関数で  $j$  の出力領域を表わして  
いて各変数の定める状態がその領域外ならば 0  $j$  領域内  
である場合に  $M$  の値をとるものとし  $f'_j$  も同様  $j$  の出力領  
域を表わしていて各変数の定める状態がその領域外ならば  $M$   
を  $j$  領域内である場合に 0 の値をとるものとしよう。すると  
 $F, F'$  は任意の多値多変数関数を表現することになる。

ここで  $f_j, f'_j$  を実現するためにこれらの関数を次の様な形  
で書いてみよう。

$$(XVII) \quad f_j = f_{j1} + f_{j2} + f_{j3} + \dots + f_{jq} + \dots \quad \text{---(a)}$$

$$f'_j = f'_{j1} \cdot f'_{j2} \cdot f'_{j3} \cdot \dots \cdot f'_{jq} \cdot \dots \quad \text{---(b)}$$

ここで  $f_{jq}$  は  $j$  の値をとる一状態を  $(x_0, x_1, \dots, x_n)$  の関数とし  
て表現しており その状態に  $(x_0, x_1, \dots, x_n)$  が落ちれば  $M$  の値  
を取り その他の状態に落ちれば 0 の値をとることとし  $f'_{jq}$   
も同様にその定める  $j$  領域に  $(x_0, x_1, \dots, x_n)$  が落ちれば 0 の  
値を取り その他の状態に落ちれば  $M$  の値を取ることにしよ  
う。すると  $f_j, f'_j$  はある  $j$  値の全領域を表現し 之に  $(x_0, x_1, \dots, x_n)$  の取る状態が含まれるか含まれないかを判別でき  
る関数になる訳である。次に  $S, S'$  なる以下の様な関数を  
考えてみる。

$$(VII) S = \phi(P)\{x_0\} \cdot \phi(Q)\{x_1\} \cdot \phi(R)\{x_2\} \cdot \dots \quad \dots (a)$$

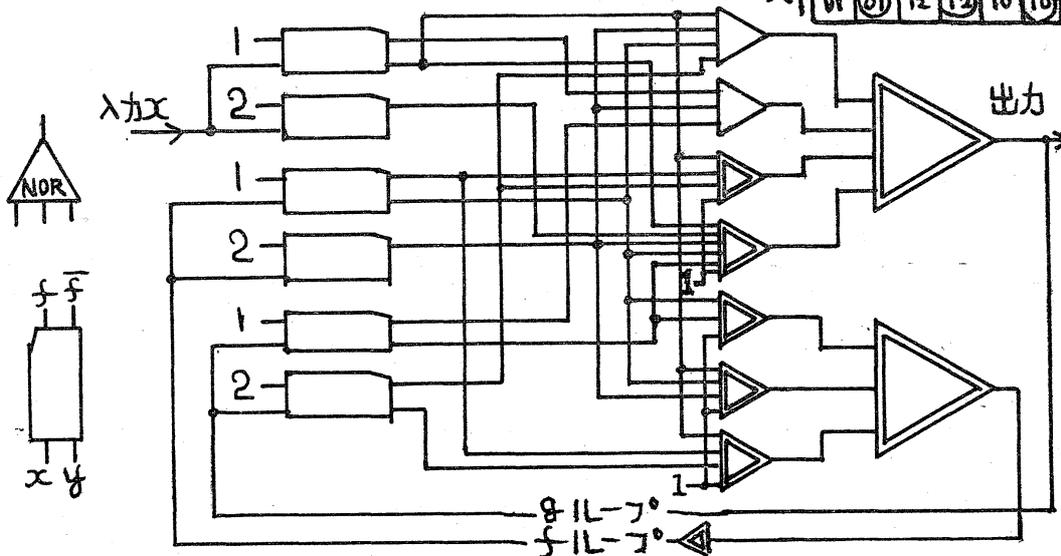
$$S' = \overline{\phi(P)\{x_0\}} + \overline{\phi(Q)\{x_1\}} + \overline{\phi(R)\{x_2\}} + \dots \quad \dots (b)$$

さて両方の  $\phi(i)\{x_i\}$  による多項式についてその出力を考えてみる。  
 $S$  は  $(x_0, x_1, \dots, x_n)$  が各々  $(P, Q, R, \dots)$  の時  $M$  の値をとり、その他の時は  $0$  の値をとる。  
 $S'$  については  $(x_0, x_1, \dots, x_n)$  が各々  $(P, Q, R, \dots)$  の時  $M$  の値をとり、その他の時は  $0$  の値をとる。  
 よって  $S \rightarrow f_i, S' \rightarrow f'_i$  に対応し、 $f_i, f'_i$  は  $\phi(i)\{x_i\}$  によって実現されることになる。

(VII)式による回路網簡略はきわめて明解で(V)~(XI)の  $\phi(i)\{x_i\}$  の性質を利用して各  $f_i$  を簡単にすればよいのである。

例) 入力  $x$  に  $0, 1, 0, 1, \dots$  を、出力に  $0-1-2-2-1-0$  を出す順序回路を実現してみる。

5-8	00	01	02	12	11	10
$x_0$	00	02	02	11	11	00
$x_1$	01	01	12	12	10	10



< 図 5 >

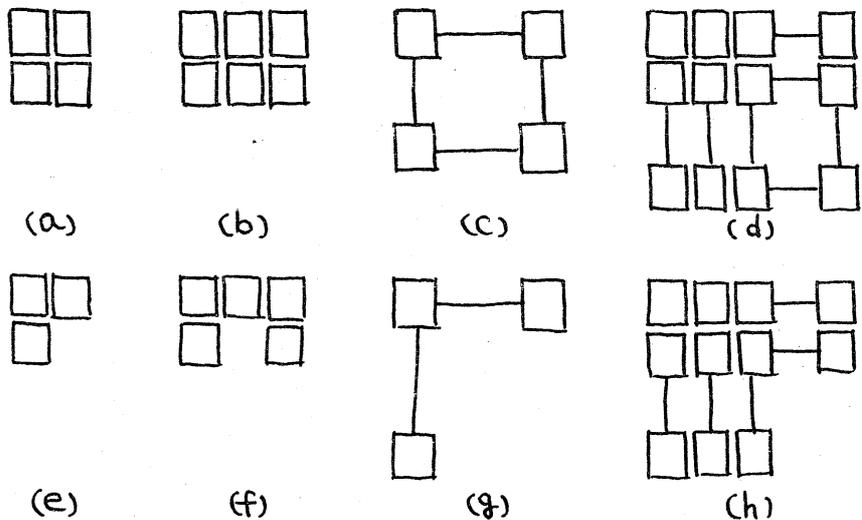
## 5 関数群の多値関数の出力の扱ひ

### 5.1 $\phi(x)$ の符号空間の1-gate領域

まず一つのゲートで表現できる領域とは何かについて記すと3.で明らかにしな様に以下の様になる。

ある領域を表現するのに それに係る全ての変数が単一レベルで書き表わせば それは一つのゲートで構成される。

この領域を1-gate領域と呼ぶことにする。ここでこれらの領域がどんなものを概念を明かにするため 任意の2変数が占める平面において示そう。



(a)~(d)は1-gate領域 (e)~(h)は前者の領域から一部分を取去ったもので 2-gate領域になる。これらからわかる様に1-gate領域は以下の様に言うことができる。

ある領域の一部分を一距離だけ移動することによ

って  $N$ 変数の作る符号空間の内  $N_s$ 変数に対して  
どの2変数を取っても その平面で写形に写る様な  
領域は その  $N_s$ 変数に対して 1-gate領域である。

そこで与えられた多値多変数関数による表を これらの  
1-gate領域パターンによって より少ない数で覆うことが  
回路網簡略化の方向である。

## 5.2 $F_i$ と $F_j$ ( $F_i$ と $F_j$ ) の包含関係

$j > i$  の関係において  $F_i$  に  $F_j$  の任意の元  $f_k$  を加  
えても (M)式において等価である。  $\Rightarrow F_i$  が  $F_j$  の一  
部または全てを包含することが許される。

同様に  $F_j$  に  $F_i$  の任意の元を加えることもでき  
る。

これは

$$F_i = f_{i1} + f_{i2} + f_{i3} + \dots + f_{ig} + \dots$$

$$F_j = f_{j1} + f_{j2} + f_{j3} + \dots + f_{jg} + \dots$$

であるとする。例えば  $F_i$  に  $F_j$  の  $f_{j1}, f_{j2}, f_{j3}$  を加  
えて

$$F_i^* = (f_{j1} + f_{j2} + f_{j3}) + f_{i1} + f_{i2} + f_{i3} + \dots + f_{ig} + \dots$$

としても  $i \cdot F_i + j \cdot F_j$  において  $F_i$  と  $F_i^*$  は等価である  
ということである。この場合  $F_i$  が  $F_j$  を包含しうると言うことに  
し  $F_i \text{ sbm } F_j$  と書くことにする。0~4の五値においては

$\mathcal{F}_1 \text{ sbm } \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$  かつ  $\mathcal{F}_2 \text{ sbm } \mathcal{F}_3, \mathcal{F}_4$  として  $\mathcal{F}_3 \text{ sbm } \mathcal{F}_4$   
 かつ  $\mathcal{F}_4 \text{ sbm } \mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_1$  かつ  $\mathcal{F}_3 \text{ sbm } \mathcal{F}_2, \mathcal{F}_1$  として  $\mathcal{F}_2 \text{ sbm } \mathcal{F}_1$   
 と言うことになる。

(証明)  $\mathcal{F}_i$  の元  $b_{i1}, b_{i2}, b_{i3}, \dots$  を  $\mathcal{F}_i$  へ加えたとす  
 ると  $\lambda \cdot \mathcal{F}_i^*$  は

$$\lambda \cdot \mathcal{F}_i^* = \lambda \cdot (b_{i1} + b_{i2} + b_{i3} + \dots + b_{iq} + \dots + b_{j1} + b_{j2} + b_{j3} + \dots)$$

(XVI)式において  $\lambda \cdot \mathcal{F}_i + \mu \cdot \mathcal{F}_j$  は

$$\lambda \cdot \mathcal{F}_i^* + \mu \cdot \mathcal{F}_j = (\lambda + \mu) \cdot b_{j1} + (\lambda + \mu) \cdot b_{j2} + (\lambda + \mu) \cdot b_{j3} + \dots$$

$$+ \lambda \cdot (b_{i1} + b_{i2} + b_{i3} + \dots + b_{iq} + \dots) + \mu \cdot (\dots + b_{jq-1} + b_{jq} + b_{jq+1} + \dots)$$

$$\mu > \lambda \text{ より } (\lambda + \mu) = \mu \text{ よって}$$

$$\lambda \cdot \mathcal{F}_i^* + \mu \cdot \mathcal{F}_j = \lambda \cdot (b_{i1} + b_{i2} + b_{i3} + \dots + b_{iq} + \dots)$$

$$+ \mu \cdot (b_{j1} + b_{j2} + b_{j3} + \dots + b_{jq} + \dots)$$

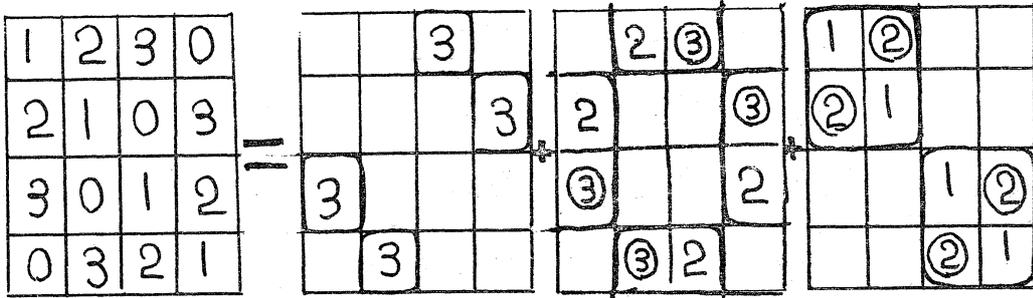
よって  $\mathcal{F}_i$  と  $\mathcal{F}_i^*$  は等価である。

終り

この包含関係は  $\mathcal{F}_i = \mathcal{F}_j$  の任意の  $b_{jq}$  を含ませることを許したもので  $\mathcal{F}_i$  全領域を含ませなくても良いことが特徴である。この包含関係によって  $\lambda$  のゲート領域に  $\mu$  のある元を加えることにより 1-gate 領域にして処理すること加えて回路網簡略に極めて有効である。

問題は  $\mathcal{F}_m, \mathcal{F}_n$  は包含するものかなく  $\mathcal{F}_m, \mathcal{F}_n$  は一つと言う様に  $\mu$  の値によって包含する相手の数が異なり それによる回路網簡略化に偏りが存在することである。

例) 包含関係による回路網簡略の実際



$$F = 3 \cdot f_3 + 2 \cdot f_2 + 1 \cdot f_1$$

(4gates  $\rightarrow$  4gates)    (4gates  $\rightarrow$  2gates)    (4gates  $\rightarrow$  2gates)

上の例でも明らかな様に [3] に対して回路網の簡略化は包含関係によつては成されず。そこで今度は 二の偏りをなくすため各出力の大小による順序関係を認めず 異記号として取扱うことを考えよう。

### 5.3

今  $0 \sim M$  の出力値を  $i, j, k$  の三つのグループに分割することなく別けをとする。そして  $f_{ij}, f_{jk}$  を各々  $(i, j)$  と  $(j, k)$  の出力領域を表わす関数とする。すると以下の関係が成立する。

$$\left\{ \begin{array}{l} f_i = f_{ij} \cdot \overline{f_{jk}} \\ f_j = f_{ij} \cdot f_{jk} \\ f_k = \overline{f_{ij}} \cdot f_{jk} \\ (\overline{f_{ijk}} = \overline{f_{ij}} \cdot \overline{f_{jk}}) \end{array} \right.$$

同様に  $f_{ij}$   $f_{jk}$  を定義すると

$$\left\{ \begin{array}{l} f_i = f_{ij} + \overline{f_{jk}} \\ f_j = f_{ij} + f_{jk} \\ f_k = \overline{f_{ij}} + f_{jk} \\ (\overline{f_{ijk}} = \overline{f_{ij}} + \overline{f_{jk}}) \end{array} \right.$$

また

$$\overline{f_i} = f_{\bar{i}} = f_{jk}$$

$$\overline{f_j} = f_{\bar{j}} = f_{ik}$$

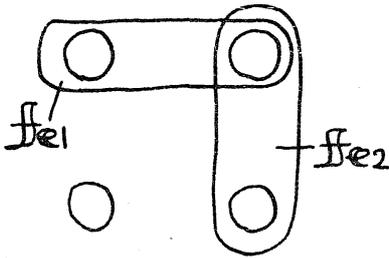
これらの演算等によって各  $i, j, k$  は単一出力領域を規定することかできる訳であり  $f_j, \overline{f_j}$  が構成される。

以下  $(ij)$   $(jk)$  等の  $f$  の表現する出力集合を  $\mathcal{E}$  と書くことにする。

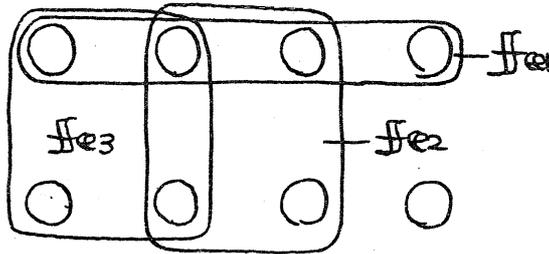
さて今まで  $m$  値の各領域を表現するのに基本的には  $(m-1)$  個の  $f_j$  を用いてきたが  $f_{\mathcal{E}}$  とその補元を用いることにより  $n$  個の  $f_{\mathcal{E}}$  によって  $2^n$  値の領域が識別される。よって

$m$  値の各領域を表現するのに必要な  $f_{\mathcal{E}}$  の数は  $m < 2^n$  を満足する最小の  $n$  の数である。

3~4 値に対しては〈図 6〉の 4 状態のいずれかに各出力値を 5~8 値に対しては〈図 7〉の 8 状態のいずれかに各出力を割当てる訳である。



〈図6〉



〈図7〉

例として0~4の五値に対して  $f_1, f_2, f_3, f_4$  の4つの  $f_i$  を用いる代わりに例えば  $f_{012}, f_{03}, f_{14}$  の3つをもって前者に代えることができる。

さて各  $f_{e1}, f_{e2}, \dots$  によって (XVI) 式を構成する際に新しくいくつかのゲートが必要になるかについて考えてみよう。問題は各  $f_{ei}$  の補元をどうして得るかである。

$$f_{ei} = f_{ij} \cdot \overline{f_{jk}} \quad i \rightarrow i$$

であるとする  $f_{ijk}$  は (XVI) 式によって以下の様に見える。

$$f_{ijk} = b_1 + b_2 + b_3 + \dots + b_q + \dots$$

$$\begin{aligned} \text{よって } f_{ei} &= f_{ij} \cdot \overline{(b_1 + b_2 + b_3 + \dots + b_q + \dots)} \\ &= f_{ij} \cdot \overline{b_1} \cdot \overline{b_2} \cdot \overline{b_3} \cdot \dots \cdot \overline{b_q} \cdot \dots \end{aligned}$$

ここで回路のレベルを考えに入れると  $\overline{f_{ijk}}$  は それを構成する各  $b_q$  を個別に  $f_{ij}$  との積をとればよく新たにゲートは必要としない。

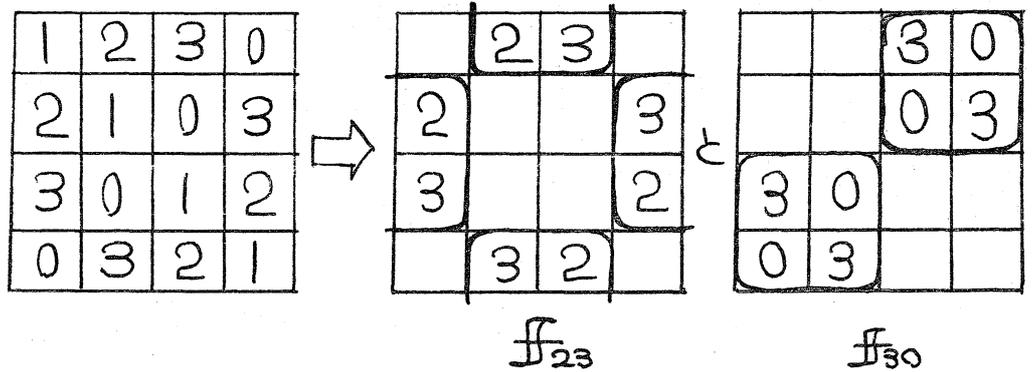
$f_{e1}, f_{e2}, \dots$  より  $f_i$  のかわりに (XVI) 式を構成する際にも同様

$$f_i = f_{ij} + \overline{b_1} + \overline{b_2} + \overline{b_3} + \dots + \overline{b_q} + \dots$$

よって回路網の簡略化に対しては  $e_1, e_2, e_3, \dots$  をどの様に選ぶかだけを考えれば良いことが保証される訳である。

例  $f_e$  による回路網簡略化の実際

前と同じ例題で見ると



$$\begin{cases} f_3 = f_{23} \cdot f_{30} \\ f_2 = f_{23} \cdot \overline{f_{30}} \\ f_1 = \overline{f_{23}} \cdot \overline{f_{30}} \end{cases}$$

$e_1 = (23)$   $e_2 = (30)$  と選ぶことにより、包含関係によつては簡略できなかった  $3$  も簡略でき、計 4-gates で回路網が構成される。

ここで前記の  $e_1 = (23)$  と決めたとすると、2, 3 の出力領域は  $f_{23}$  によって全て表現しなくてはならず、 $f_i \text{ sbm } f_j$  の時の様に都合の良いものだけを選ぶことはできない。ある部分でゲートを減らし、 $e$  も別の所で余計なゲートを新たに要求するかもしれないから、ここには  $e$  の割当てにおける

むずかしさがある。

5.2で示した包含関係は演算素子感度の大小によっているが一般的には  $0 \sim M$  の値が 整列集合の性質を持ってあつかわれていればその順序は値の大小、演算素子感度には無関係に決めることのできる。回路網の大小はこの順序関係に大きく左右される。文献(8), (9)により多値NAND回路網における一般的包含順序による展開式と最適順序関係を決めて回路を合成する自動設計について記した。詳図は(8), (9)に同じ  $f_3 \text{ sbm } f_1 \text{ sbm } f_2 \text{ sbm } f_0 \text{ sbm } f_4$  を含む順序を満足するF式を求めてみる。

$$F = 4 \cdot f_4 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot \bar{f}_1 \cdot \bar{f}_3 + 3 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot \bar{f}_1 \cdot f_3 + 2 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot f_2 \cdot \bar{f}_1 \cdot \bar{f}_3 + 1 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot f_1 \cdot \bar{f}_3$$

$$= 4 \cdot f_4 + 3 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot \bar{f}_1 \cdot f_3^* + 2 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot f_2^* + 1 \cdot \bar{f}_4 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot f_1^*$$

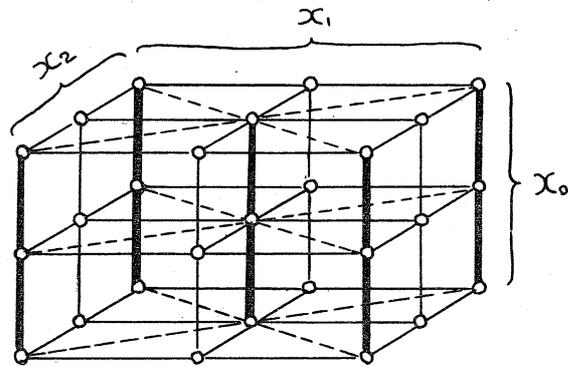
$$= 4 \cdot f_4 + 3 \cdot \bar{f}_0 \cdot \bar{f}_2 \cdot \bar{f}_1 \cdot f_3^* + 2 \cdot \bar{f}_0 \cdot f_2^* + 1 \cdot \bar{f}_0 \cdot f_1^*$$

これかやえられ包含関係を満足するF式である。

## 6. 置換素の導入

さて〈図2〉の表示によって3値3変数組状態図を書くと〈図8〉の様になる。  $x_1, x_2$  を付に注目すると点線で結ばれ各点の状態は各々2距離にある。だからもし点線で結ばれ状態に同一出力が存在してもそれは前記の理由により回路網を簡単にはしない。そこで今度は実線(太い線)で結ばれ5本の各状態を見ると  $x_1, x_2$  からは それぞれの棒状

態は2距離であるが  $x_0$  に対しては一距離である。実線で結ばれた5本の各状態を見ると それらは  $P_{Kx_0}$  の9本中の5本に対応していることがわかる。Kは  $x_1, x_2$  の状態を示す。もしこれらの内に同じ並ぶものものが存在すれば それらは  $x_0$  の制御する置換素の共用によって表現することかできる。それにより新しい回路網簡略の余地をみつけないことになる。



三値三変数組の状態図 <図8>

この図は 2変数組状態( $x_1$ と $x_2$ ) の対応点を第3変数 $x_0$  の変化が結んでいる<図4>に対応するものである。

置換素合成法による多値多変数回路網合成

今  $F = \phi(x) \{x\} \cdot f$  ----- (a)

$F' = \overline{\phi(x) \{x\}} + f$  ----- (b)

存在=つの関数を考えると Fは  $x=i$  の時に限り f なる出力を与え その他の時は0の出力とする。またF'は  $x=i$  の時に限り同様 f なる出力を与え その他の時はMの出力と

なる。そこで上二式を以下の様へ定義しなおす。

$$(XIX) \quad F = \sum_i \phi(\omega) \{x_i\} \cdot b_i \quad \text{----- (a)}$$

$$F' = \prod_i (\overline{\phi(\omega) \{x_i\}} + b_i) \quad \text{----- (b)}$$

すると各々の出力は

$$F, F' = \begin{pmatrix} x; & 0, 1, 2, \dots, M \\ \text{出力}; & b_0, b_1, b_2, \dots, b_m \end{pmatrix}$$

と各り  $M$  値のゲートを形成する。よって  $F, F'$  のいずれかを重ねていくことにより多変数回路を合成することかてきる。

そして  $F, F'$  の関数連鎖の最終段にはその多値多変数関数の取るべき値とある一変数  $x_0$  との置換素  $P_{Kx_0}$  が付いて回路は完成する訳である。

$$(XX) \quad P_{Kx_0} = \begin{pmatrix} x_0; & 0, 1, 2, \dots, M \\ \text{出力}; & A_0, A_1, A_2, \dots, A_m \end{pmatrix}$$

上式の  $P_{Kx_0}$  を  $K$  により枝分けされ  $x_0$  により制御される置換素と言う。  $A_i$  は  $0 \sim M$  の任意の値を取る定数。  $K$  は  $x_0$  を除く ( $x_1, x_2, \dots, x_n$ ) により表記される。そしてこれらの値により選択される置換素であることを示す。すると前記 (XIX) 式より

$$(XXI) \quad P_{Kx_0} = \left[ \sum_i A_i \cdot \phi(\omega) \{x_i\} \right]_{Kx_0} \text{----- (a)} = \left[ \prod_i (\overline{\phi(\omega) \{x_i\}} + A_i) \right]_{Kx_0} \text{----- (b)}$$

よって任意の  $M$  値  $n$  変数関数は (XIX), (XX) を用いて以下の様へ展

開される。

$$(XII) \quad T_{mm} = \sum_{i=0}^{m-1} (\phi(i)\{x_{m-1}\}) \cdot \sum_{j=0}^{m-1} (\phi(j)\{x_{m-2}\}) \cdot \dots \dots \dots \\ \dots \dots \dots \cdot \sum (\phi(k)\{x_i\}) \cdot \sum (\phi(i_0)\{x_0\} \cdot A_{k i_0}) \dots$$

$$T'_{mm} = \prod_{i=0}^{m-1} (\overline{\phi(i)\{x_{m-1}\}}) + \prod_{i=0}^{m-1} (\overline{\phi(j)\{x_{m-2}\}}) + \dots \dots \dots \\ \dots \dots \dots + \prod (\overline{\phi(k)\{x_i\}}) + \prod (\overline{\phi(i_0)\{x_0\}} + A_{k i_0}) \dots$$

上式は関数の内容の如何にかかわらず実現される。この置換素合成法は単独では冗長な回路網になってしまうが置換素の概念はある条件において有効な回路網簡略化に一役かう。まず  $\mathbb{P}_{Kx_0}$  の  $K$  により規定される状態を指定するのに多値ゲート連鎖を用いず  $\mathbb{F}$  により規定するのである。

(XVI) …… (a), (b) において  $\mathbb{F}_j, \mathbb{F}'_j$  は  $j$  出力領域を表わす関数であるがこれを今置換素  $\mathbb{P}_{Kx_0}$  で置換しようとする領域を表わす関数と考えると  $\mathbb{F}_j, \mathbb{F}'_j$  等は その置換素を制御する変数  $x_0$  から独立になる。で (XVI) 式 - (a), (b) は以下の様に書き換えられる。

$$(XIII) \quad F = \mathbb{F}_{x_0}(x_1, x_2, \dots, x_n) \cdot \mathbb{P}_{Kx_0} + \dots \dots \dots \\ + \mathbb{F}_M + (M-1) \cdot \mathbb{F}_{M-1} + (M-2) \cdot \mathbb{F}_{M-2} + \dots + 1 \cdot \mathbb{F}_1 \quad - (a)$$

$$F' = (\mathbb{F}'_{x_0} + \mathbb{P}_{Kx_0}) \cdot \dots \dots \dots \\ \cdot ((M-1) + \mathbb{F}'_{M-1}) \cdot ((M-2) + \mathbb{F}'_{M-2}) \cdot \dots \dots \cdot \mathbb{F}'_0 \quad - (b)$$

$f_{x_0}$ ,  $f'_{x_0}$  は  $\mathbb{R}_{x_0}$  の点める領域を表現している。

上式はさらに  $\mathbb{R}_{x_0} \in (M) \dots (a), (b)$  によって表示して書きなおすと

$$F = f_{x_0} \cdot (A \cdot \phi(a)\{x_0\} + B \cdot \phi(b)\{x_0\} + C \cdot \phi(c)\{x_0\} + \dots) \\ + f_M + (M-1) \cdot f_{M-1} + (M-2) \cdot f_{M-2} + \dots + 1 \cdot f_1$$

そこで  $A = M-1$ ,  $B = M-2$ ,  $C = M-3$  とすると上式は

$$(XXIV) \quad F = f_M + (M-1) \cdot \left( f_{M-1} + \begin{array}{l} \phi(a)\{x_0\} \cdot f_{x_0} \\ \phi(b)\{x_0\} \cdot f_{x_0} \\ \phi(c)\{x_0\} \cdot f_{x_0} \end{array} \right) \\ + (M-2) \cdot \left( f_{M-2} + \begin{array}{l} \phi(b)\{x_0\} \cdot f_{x_0} \\ \phi(c)\{x_0\} \cdot f_{x_0} \end{array} \right) \\ + (M-3) \cdot \left( f_{M-3} + \begin{array}{l} \phi(c)\{x_0\} \cdot f_{x_0} \end{array} \right) \\ + \dots + \dots \quad \dots (a)$$

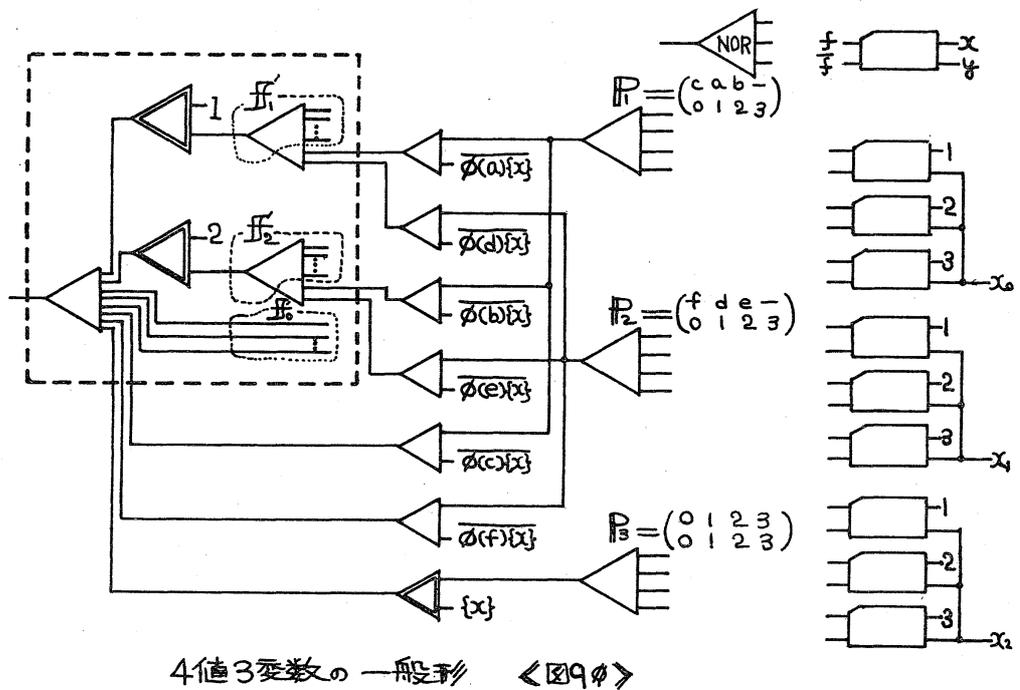
同様にして  $F'$  も以下の様に表示する。

$$F' = \left( f'_{x_0} + (A + \phi(a)\{x_0\})(B + \phi(b)\{x_0\})(C + \phi(c)\{x_0\}) \dots \right) \\ \cdot \left( (M-1) + f'_{M-1} \right) \cdot \left( (M-2) + f'_{M-2} \right) \cdot \left( (M-3) + f'_{M-3} \right) \cdot \dots \cdot f'_0 \\ = \left( (M-1) + f'_{M-1} \cdot \begin{array}{l} (\phi(a)\{x_0\} + f'_{x_0}) \\ (\phi(b)\{x_0\} + f'_{x_0}) \\ (\phi(c)\{x_0\} + f'_{x_0}) \end{array} \right) \\ \cdot \left( (M-2) + f'_{M-2} \cdot \begin{array}{l} (\phi(b)\{x_0\} + f'_{x_0}) \\ (\phi(c)\{x_0\} + f'_{x_0}) \end{array} \right) \\ \cdot \left( (M-3) + f'_{M-3} \cdot \begin{array}{l} (\phi(c)\{x_0\} + f'_{x_0}) \end{array} \right) \\ \cdot \dots \cdot f'_0 \quad \dots (b)$$

上2式(XXIV) ... (a), (b) の点線で囲んだ部分は変数  $\{x_0\}$  の多値ゲートを構成しているといわねがかる。しかもそのゲートを通過する関数は  $f_{x_0}$ ,  $f'_{x_0}$  でこれは0とMとしか値を取らないので0, Mの2値素子によく置換素を導入しなすかかかめらる多値

NANDでゲートを作らなくても良い。そして多値NANDは(XVI)式構成に必要素数より一般に増すことはない。

ニニ以下置換素導入による回路網の一般的形状を4値3変数の例をとって示してみよう。〈図9〉の $P_3$ の様ニ置換素が直接 $\{x\}$ の関数である場合は多値素子にしなくてはならず、多値素子の増す一例として上げろ。



### 7. 多値NAND数と回路網レベル数について

回路網を大きく分けると $f(x; y, 0, M)$ とNAND回路網とに分れている。ニニで各NANDを通過する情報を調べていくと全てに多値NANDを用いる必要が無いとは前記の通りである。さるに多値NANDを増すことによつて

回路網として段数を減らすこともできる。

まず(1)のみのみを扱う関数  $f_j$ ,  $f_j$  は 0, 1 の 2値 NAND で良いことか分かる。よって多値 NAND は (XVI) 式

$$F = f_M + (M-1) \cdot f_{M-1} + (M-2) \cdot f_{M-2} + \dots + 1 \cdot f_1$$

を構成する  $(M+1)$  個の良いことか分かる。  $F'$  についても同じである。しかしながら  $f_j$  と出力との間に 2 段の回路が直列に入っているのは速度等の点から言っても面白くない。そこで多値 NAND 最小の (XVI) 式を求めて 重みを  $f_j$  の各項に分配する回路にする。そこで  $f_j$  を (XVII) 式で書いて

$$\begin{aligned} F &= b_{M1} + b_{M2} + b_{M3} + \dots + b_{M8} + \dots \\ &\quad + (M-1) \cdot b_{(M-1)1} + (M-1) \cdot b_{(M-1)2} + (M-1) \cdot b_{(M-1)3} + \dots \\ &\quad + \dots + 1 \cdot b_{11} + 1 \cdot b_{12} + 1 \cdot b_{13} + \dots + 1 \cdot b_{18} + \dots \end{aligned}$$

とすることによって各  $b_{ij}$  は多値 NAND を必要とする様になる。またそれによって回路素子は減少し段数も 2 段下がっていることか分かる。また先頭の NAND の  $F_{an \text{ in}}$  が大巾に増大しているので問題になる場合もある。  $F'$  についても同様。

一般的には多値 NAND, 2 値 NAND 総数は上の変形によって前者は  $f_M$  を除く  $f_1, f_2, \dots, f_{M-1}$  関係する  $(M-1)$  個, そしてまた  $f_M$  を除く  $f_1, f_2, \dots, f_{M-1}$  で (XVI) 式を担っている  $(M-1)$  個の 2 値 NAND が不用になり計  $2(M-1)$  個減少している。しかし  $f_M$  にかかわる  $b_{ij}$  以外はすべて多値 NAND で

構成されなくてはならない。

よって多値NANDを用いる不経済性と回路網簡略化による高速化との関係でどちらの立場をより強調するかでその回路が決まるのである。

おわりに

一致検出関数の(1)(2)による回路網合成一般について記してきたが素子数に着目すると同一状態を識別する一般2値回路とコンパラブル素子数で実現される訳だが今後多値論理が2値論理に対して実際面でのXリットを調べることに興味深い。

### 参考文献

- (1) 三根 久, 長谷川利治, 島田良作; “三値NAND回路を用いた自動タイミンゲ非同期論理回路とその二進加算器における効果” 信学誌VOL 53-C NO.9 P652~659
- (2) 島田良作; “多値論理回路の簡単化” 信学誌VOL54C,  
“多値論理における置換と束縛算について” 70年当研究会報告集
- (3) 平山 博, 永井正大; “電流切換形三値論理回路”  
電算機研究会資料EC68-36
- (4) CHARLES M. ALLEN; DONALD D. GIVONE; “A Minimization Technique for Multiple-Valued Logic Systems”  
IEEE 1968 2月 TRANSACTIONS ON COMPUTERS, P182~

- (5) W.H. HANSON ; "Ternary Threshold Logic" IEEE  
TRANSACTION ON ELECTRONIC COMPUTERS , JUNE 1963 P191~197
- (6) 並木淳治, 平山 博 ; "多値多変数回路網合成とその  
最小化" 46年全国大会
- (7) 並木淳治, 平山 博 ; "複数合成法による多値多変数  
回路網の簡略化" 電算機研資EC71-4
- (8) 並木淳治 平山 博 "多値NAND回路の一般的包含関  
係" 電算機研資EC71-24
- (9) 並木淳治 "包含順序まで決める多値論理関数の自動最小化"  
電算機研資 2月 (1972年)

早大音研・柳川博文君図を書いてもらってありがとうございます。