

M - A N D, M - O R, N O T 演算を用いた多値論理関数の
簡単化に関する一考察

山口大学工学部 長尾敬之 (Takayuki Nagao)

高浪五男 (Itsuo Takanami)

井上克司 (Katushi Inoue)

1. はじめに

論理レベルを多値化することによって IC ピン数や配線数の減少、動作の高速化、特に直列伝送の高速化、等が期待できる。しかし、二値論理の場合のような統一的な論理系が確立されていないことなどが 実用化に際しての問題点となっている。

本報告では、Lukasiewiczにより示された演算（文献（1）に従い、M-AND, M-OR, NOT演算と呼ぶ）を基本演算とする多値論理関数の簡単化の一手法を提案する。本方法は、文献（1）、（2）に示された標準形およびCube表現を用いて簡単化を行うものであるが、文献（2）に示された方法に比較して特殊な場合を除き、更に簡単化することが出来る。

2. 準備

$L = \{0, 1, 2, 3, \dots, p-1\}$, p 値変数 $x_i, x_j \in L$ ($1 \leq i, j \leq n$) とし、 p 値 n 変数論理関数 $f(x_1, x_2, \dots, x_n)$ を $L^n \rightarrow L$ と定義する。

【定義 1】 p 値変数 $x_i, x_j \in L$ に対して M -AND 演算 $x_i \cdot x_j$ を以下の様に定義する。

$$x_i \cdot x_j = \begin{cases} 0 & ; x_i + x_j \leq p-1 \\ x_i + x_j - (p-1) & ; x_i + x_j > p-1 \end{cases} \quad (1)$$

【定義 2】 p 値変数 $x_i, x_j \in L$ に対して M -OR 演算 $x_i \vee x_j$ を以下の様に定義する。

$$x_i \vee x_j = \begin{cases} p-1 & ; x_i + x_j \geq p-1 \\ x_i + x_j & ; x_i + x_j < p-1 \end{cases} \quad (2)$$

【定義 3】 p 値変数 $x_i \in L$ に対して NOT 演算 $\bar{x}_i$ を以下の様に定義する。

$$\bar{x}_i = (p-1) - x_i \quad (3)$$

ただし、 $+$, $-$ は、通常の算術演算を表す。

上記の M -AND, M -OR, NOT 演算は、交換律、結合律、相補律、ベキ等律、ド・モルガンの法則を満足する。

また、 $a \cdot x_i$ の m 次の M -OR の意味での和

$$\overbrace{(a_1 x_1) \vee (a_2 x_2) \vee \cdots \vee (a_m x_m)}^{m=}$$

を $(a_i x_i)_m$ であらわす。定義 1, 2 より $(p x)_{p-1} = p-1$ である。

3. p 値論理関数の標準形

文献 (1)、(2) より、p 値論理関数の標準形は、次式の様になる。

$$f(x_1, x_2, \dots, x_n) = \sum f(\alpha) \prod ((p - a_i) x_i)_{p-1} ((a_i + 1) \bar{x}_i)_{p-1} \quad (4)$$

ただし、 $\alpha = (a_1, a_2, \dots, a_n) \in L^n$ 、 $1 \leq i \leq n$ であり、 $\sum$ はすべての α についての M-OR の意味での総和を、 $\prod$ は M-AND の意味での乗積を表す。

また、 $a_i \leq a_{i'}$ とすると

$$((p - a_i) x_i)_{p-1} ((a_{i'} + 1) \bar{x}_i)_{p-1} = \begin{cases} p-1 & ; a_i \leq x_i \leq a_{i'} \\ 0 & ; \text{それ以外するとき} \end{cases} \quad (5)$$

が成り立つ。式 (5) は $a_i = a_{i'}$ のとき、式 (4) の積項に一致する。

4. Cube 表現

簡単化を行う際、関数を効率的に表現する方法として、文献 (2) に示されているような Cube 表現を用いる。Cube 表現は積項

$$\prod ((p-a_i)x_i)^{p-1}((a_i'+1)\bar{x}_i)^{p-1} \quad (6)$$

に 1 対 1 に対応しており、次のように定義される。

【定義 4】 関数値 k ($1 \leq k \leq p-1$) を持つ項の積項

$$\prod ((p-a_i)x_i)^{p-1}((a_i'+1)\bar{x}_i)^{p-1}$$

に対して $\text{Cube } \mathbb{C}_k$ は

$$\mathbb{C}_k = \mathbb{C}_1 \mathbb{C}_2 \cdots \mathbb{C}_i \cdots \mathbb{C}_n \quad (1 \leq i \leq n)$$

$\mathbb{C}_i$ を座標 i のベクトルと呼ぶ。

ここで

$$\mathbb{C}_i = (C_{i0}, C_{i1}, \dots, C_{ij}, \dots, C_{i,p-1}) \quad (0 \leq j \leq p-1)$$

であり、 C_{ij} は、 $a_i \leq j \leq a_i'$ のとき 1 を、それ以外るとき 0 をとる。

次に、Cube 間で 次のような OR 演算、成分間の算術的和を定義する。

【定義 5】 Cube 間の OR 演算 $\mathbb{C} \cup \mathbb{C}'$ 、成分間の算術和 $\mathbb{C} + \mathbb{C}'$ を以下のように定義する。

$$\begin{aligned} \mathbb{C} \cup \mathbb{C}' &= \mathbb{C}_1 \mathbb{C}_2 \cdots \mathbb{C}_i \cdots \mathbb{C}_n \cup \mathbb{C}_1' \mathbb{C}_2' \cdots \mathbb{C}_i' \cdots \mathbb{C}_n' \\ &= (\mathbb{C}_1 \cup \mathbb{C}_1') \cdots (\mathbb{C}_i \cup \mathbb{C}_i') \cdots (\mathbb{C}_n \cup \mathbb{C}_n') \end{aligned}$$

ただし、

$$\begin{aligned} \mathbb{C}_i \cup \mathbb{C}_i' &= (\max(C_{i0}, C_{i0}'), \max(C_{i1}, C_{i1}'), \dots, \max \\ & (C_{i,p-1}, C_{i,p-1}')) \end{aligned}$$

とする。

$$\begin{aligned} \mathbb{C} + \mathbb{C}' &= \mathbb{C}_1 \mathbb{C}_2 \cdots \mathbb{C}_i \cdots \mathbb{C}_n + \mathbb{C}_1' \mathbb{C}_2' \cdots \mathbb{C}_i' \cdots \mathbb{C}_n' \\ &= (\mathbb{C}_1 + \mathbb{C}_1') \cdots (\mathbb{C}_i + \mathbb{C}_i') \cdots (\mathbb{C}_n + \mathbb{C}_n') \end{aligned}$$

ただし、 $+$ は通常の算術的な加算を意味する。

[例 1] Fig.1に示す真理値表により与えられた4

値2変数関数の場合、1の値を持つセルに対応するCube表現は、

(1000)(0010)、2の値を持つセルに対応するCube表現は (01

00)(0100)、3の値を持つセルに対応するCube表現は (0010)

(0100)となる。また、2の値を持つセルに対応するCubeと 3

の値を持つセルに対応するCubeとのOR演算は、(0110)(0100)

となり、算術和は(0110)(0200)となる。

5. 簡単化の方法

本方法では、式(4)の標準形の各積項をCubeによって表現

し、Cube間の隣接関係を調べることによって簡単化を行う。

隣接関係は、Cube間のOR演算、算術和を求め、その結果によ

って判断する。具体的には、以下のような条件を満たす場合

に、2つのCube(即ちそれらのCubeによって表現される項)は

隣接しているという。

条件1: 2つのCube間のOR演算によって得られたCubeの座標 i のベクトル $\mathbb{C}_i$ の成分に $\dots 10\dots 01\dots$ のような配列が存在しないこと。

条件2: $\mathbb{C}_i + \mathbb{C}_{i'} \in \{0, 1\}^p$ を満たす $i (1 \leq i \leq n)$ が唯一つ存在し、 $\mathbb{C}_i + \mathbb{C}_{j'} \in \{0, 2\}^p$ ($1 \leq j \leq n, i \neq j$) が成立すること。以下に例を示す。

[例2] Fig.1に示した関数の場合、

$$\mathbb{C}_1 \cup \mathbb{C}_2 = (1100)(0110)$$

$$\mathbb{C}_1 + \mathbb{C}_2 = (1100)(0110)$$

従って、条件2を満たさないため1の項と2の項は、隣接していない。また、

$$\mathbb{C}_2 \cup \mathbb{C}_3 = (0110)(0100)$$

$$\mathbb{C}_2 + \mathbb{C}_3 = (0110)(0200)$$

上記の条件1, 2を満たすため2の項と3の項は、隣接している。

式(4)の標準形の積項 $\prod ((p-a_i)x_i)^{p-1} ((a_i+1)\bar{x}_i)^{p-1}$ は式(5)より、 $\alpha = (a_1, a_2, \dots, a_n) \in L^n$ によって示されるセルが0以外の値を持っている場合に $p-1$ となり、そのセルの示す値が0のときには、0となる。従って、式(4)に示す標準形は、 α によって示されたセルが0以外の値を示すときには

$$\sum f(\alpha) \prod ((p-a_i)a_i)^{p-1} ((a_i+1)\bar{a}_i)^{p-1}$$

$$= 0 \vee 0 \vee \dots \vee f(\alpha) \prod ((p-a_i)x_i)^{p-1} ((a_i+1)\bar{a}_i)^{p-1} \vee$$

... $\vee 0$

$$= f(\alpha) \prod ((p-a_i)a_i)_{p-1} ((a_i+1)a_i)_{p-1} \quad (7)$$

$$= f(\alpha)$$

ここで、 $f(\alpha)=p-1$ のときには 式(7)は M-OR の定義より

$$\begin{aligned} & (p-1) \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \\ &= 1 \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \\ & \quad \vee (p-2) \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \\ &= 1 \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \\ & \quad \vee 1 \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \vee \dots \\ & \quad \vee 1 \prod ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} \quad (8) \end{aligned}$$

と書き換えることが出来る。

また、M-ANDの定義より $(p-1)x=x$ となるので積項のなかの $(p-a_i)$ や (a_i+1) が $p-1$ となるときには 定数 $p-1$ を除去することができる。従って、論理関数の表現に使用する定数を減らすことができ、実際に論理回路を設計する際、論理ゲートへの定数入力も減少させることができるため、配線数の減少を図ることができる。

例えば、 $a_i=1$ のとき、

$$\begin{aligned} ((p-a_i)x_i)_{p-1} ((a_i+1)\bar{x}_i)_{p-1} &= ((p-1)x_i)_{p-1} ((2\bar{x}_i)_{p-1} \\ &= (x_i)_{p-1} (2x_i)_{p-1} \end{aligned}$$

となる。この場合、得られた結果は、元の積項に比較して

定数 $p-1$ が省略できていることがわかる。

更に、式(8)より、重複する同じ形の Cube が n 個存在する場合、それらをまとめて $n \times$ (対応する積項) と表現することによって論理関数中の積項の個数を減少させることができる。Fig.1の関数の2のセルは、Cube表現では $(0100)(0100), (0100)(0100)$ と2つの Cube で表現されるが積項によって表現する場合 本方法では、

$$\begin{aligned} & 2(3x_1) \circledast (2\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \\ & = 2(x_1) \circledast (2\bar{x}_1) \circledast (x_2) \circledast (2\bar{x}_2) \circledast \end{aligned}$$

と、表現され、文献(2)の表現

$$\begin{aligned} & (x_1) \circledast (2\bar{x}_1) \circledast (x_2) \circledast (2\bar{x}_2) \circledast \\ & \vee (x_1) \circledast (2\bar{x}_1) \circledast (x_2) \circledast (2\bar{x}_2) \circledast \end{aligned}$$

より、積項の個数を減少させることが出来る。

以上のことから、標準形で与えられた論理関数の積項の個数の減少、定数の減少を行い、簡単化された関数を得る。

次に、具体的な手順を示す。

1st stage: 与えられた p 値論理関数を式(4)に従って展開する。

2nd stage: 1st stage で得られた標準形の各項のうち $f(\alpha) = p-1$ となる項を式(8)に従って展開する。

3rd stage: 各積項を定義4に従ってCube表現する。 $r=1$ と

する。

4th stage: Cubeを $f(\alpha) < p-2$ のCubeと $f(\alpha) = p-2$ のCube

を区別して縦に並べて書き、第 r 列とする。

5th stage: 第 r 列の $f(\alpha) < p-2$ のCube同志 と $f(\alpha) = p-2$

のCube同志でそれぞれ隣接関係を調べる。隣接するCubeが存

在しない場合は、7th stageへ行く。隣接するCubeが存在する

ときにはOR演算によって得られたCubeを第 $(r+1)$ 列として $f(\alpha)$

$< p-2$ のCubeと $f(\alpha) = p-2$ のCubeを区別して縦に並べて書く。

このとき第 r 列の2つのCubeにマークをしておく。以上の操作

を隣接するCubeが存在しなくなるまで行う。

6th stage: $r \leftarrow r+1$ とし、4th stageへ行く。

7th stage: マークされていないCubeを式(6)に従って積

項に展開する。

8th stage: 7th stageで得られた積項を式(4)に従って展

開する。このとき、式(4)の $f(\alpha)$ の値は、 $f(\alpha) = p-2$ の場合に

は、 $p-2$ となり、それ以外の場合には、同一のCubeの個数と

なる。ここで、同一のCubeとは、それぞれのCubeの座標 i のベ

クトルの成分の0,1の配列のパターンが等しいものをいう。

□

以上の処理を行うことによって論理関数の簡単化を行うことが出来る。

6. 例題

Fig. 2に示すような4値2変数論理関数 $f(x_1, x_2)$ について考える。

$$\begin{aligned} \text{1st stageより: } f(x_1, x_2) = & 1(2x_1) \circledast (3\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \\ & \vee 2(4x_1) \circledast (1\bar{x}_1) \circledast (2x_2) \circledast (3\bar{x}_2) \circledast \\ & \vee 2(3x_1) \circledast (2\bar{x}_1) \circledast (2x_2) \circledast (3\bar{x}_2) \circledast \\ & \vee 3(3x_1) \circledast (2\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \end{aligned}$$

$$\begin{aligned} \text{2st stageより: } f(x_1, x_2) = & 1(2x_1) \circledast (3\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \\ & \vee 2(4x_1) \circledast (1\bar{x}_1) \circledast (2x_2) \circledast (3\bar{x}_2) \circledast \\ & \vee 2(3x_1) \circledast (2\bar{x}_1) \circledast (2x_2) \circledast (3\bar{x}_2) \circledast \\ & \vee 1(3x_1) \circledast (2\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \\ & \vee 2(3x_1) \circledast (2\bar{x}_1) \circledast (3x_2) \circledast (2\bar{x}_2) \circledast \end{aligned}$$

3rd, 4th stageより: 各項のCube表現は、Tab. 1の第1列のようになる。

4th, 6th stageより: 第1列の(0010)(0100)と(0100)(0100)、(1000)(0010)と(0100)(0010)はそれぞれ隣接しているためCube間のOR演算の結果を第2列に書く。第2列には隣接するCubeは存在しない。

7th, 8th stageより: $f(x_1, x_2) = 1(3x_1)_3(3\bar{x}_1)_3(3x_2)_3(2\bar{x}_2)_3$

$$\vee 2(4x_1)_3(2x_1)_3(3x_2)_3(2x_2)_3$$

$$\vee 2(3x_1)_3(2\bar{x}_1)_3(3x_2)_3(2\bar{x}_2)_3$$

$$= 1(x_1)_3(\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3$$

$$\vee 2(2\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3$$

$$\vee 2(x_1)_3(2\bar{x}_1)_3(x_2)_3(2\bar{x}_2)_3$$

□

この例の場合、標準形に比べて、項の数も減少し、また式中の定数も、標準形が1, 2, 3, 4を使用するのに対して、1と2だけと減少している。従って、明らかに簡単化されている。

7. むすび

本報告では Lukasiewiczの示した論理和、論理積、否定演算を基本演算とするp値論理関数の簡単化の1手法について示した。本方法は、文献(2)に示されている方法に比較しても明らかに、より簡単化を行うことが出来る。また、論理関数のなかで使用する定数が減少しているということは、実際の配線の際に有利であろうと思われる。

しかし、簡単化の際、Cubeの並べ方によって結果が異なることが考えられるため簡単化に最も効果的な隣接項を探す方法

などが望まれる。また、減少させることができる定数の数が p の値に関係なく一定であるため、 p が大きくなると、あまり大きなメリットとはならないことなども今後の課題である。

参考文献

- (1) 藤田、佐藤、小高 "M-AND, M-OR, 及び NOT 演算による p 値論理関数の標準形" 信学論(D) J67-D No.10 , pp.1266-1267
- (2) 藤田、佐藤、小高 " p 値論理関数の一簡単化 " 信学論(D) J69-D No.3 pp.477-479

$x_2 \backslash x_1$	0	1	2	3
0	0	0	1	0
1	0	2	0	0
2	0	3	0	0
3	0	0	0	0

Fig.1 4値2変数論理関数の例

$x_2 \backslash x_1$	0	1	2	3
0	0	0	2	0
1	0	3	2	0
2	0	1	0	0
3	0	0	0	0

Fig.2 1値2変数論理関数の例

Tab.1 Fig.2の関数のCube表現とCubeの数の減少

	第 1 列	第 2 列
$f(\alpha)=1$	* (0010)(0100) * (0100)(0100)	(0110)(0100)
$f(\alpha)=2$	* (1000)(0010) * (0100)(0010) (0100)(0100)	(1100)(0010)