

## PLAにおけるハザードの考察

井口 幸洋 ・ 向殿 政男  
明治大学 ・ 工学部

PLA(Programmable Logic Array)は、多出力組合せ論理関数を実現するためにVLSI内部で多用されている。本稿では、PLAにおけるハザードについて考察する。組合せ論理回路におけるハザードについては、3値論理を用いたハザードの検出などの多くの研究がなされている。PLAにおける論理ハザードは、その入力変化に対応する積項をPLAに付加することで除去できることが知られており、PLA用の論理合成ツールには、ハザード除去のために冗長な積項を付け加える機能を持つものもある。本稿では、PLAの構造をもとに、信号遅延を積極的に利用してハザードを回避する考え方を述べる。これは、(1)論理関数を変えずにPLA内部の行及び列を並べ換える事ができる、(2)PLA内部での配線長が長くなればなるほど、また、駆動すべきトランジスタの個数が多くなればなるほど信号遅延は大きくなる、ということを利用して、この考え方を利用して、論理ハザードだけでなく関数ハザードも回避することができる。

## A Consideration on Hazards in PLA

Yukihiro IGUCHI and Masao MUKAIDONO

Faculty of Engineering, Meiji University  
Tama-ku, Kawasaki, Kanagawa, 214 Japan

PLA(Programmable Logic Array) is a very effective tool for implementing multiple/output combinational logic circuits in VLSI. In this paper we consider hazards in PLA. The hazards in combinational logic circuits have been studied by many researchers, e.g. for the detecting hazards by handling ternary logic. In PLA, it's known that a logic hazard is removable by adding a redundant cube to PLA. Some PLA synthesis tools have the function for adding redundant cubes. In this paper, We propose the new idea of removing the hazards by using propagation delay. This idea is based on the following facts, (1) rows and/or columns in PLA can be exchanged without changing the logic function, (2) propagation delay is depend on the length of wire and the number of transistors. It's able to remove not only logic hazards but also function hazards.

1. はじめに

集積回路技術の発達により、デジタルシステムのVLSI化が進展している。これに伴い回路や機能の大規模化・複雑化が進み、設計時間やコストの増大の原因となっている。さらに、人手のみによる設計はもはや不可能となっており、計算機支援設計が必須となっている[1]。計算機支援設計向きの設計手法として、一般に構造化設計手法が用いられている。これにより設計期間の短縮や回路検証の容易化等がはかられており、PLA(Programmable Logic Array)等の規則正しい構造の素子が、構造化設計手法に基づいてVLSI内で多用されるようになってきている[2]。

本稿では、PLAで発生するハザードについて検討する。論理回路網に発生するハザードについては数多くの研究が行われている[3]-[9]。ここでは、まず、あいまいさを取り扱うのに適したB-3値論理関数[10]とそれを用いてハザードの発生の可能性を検出する方法[8][9]を概説する。続いて、PLAにおけるハザードについて述べる。従来PLAでは、積項を付け加えることでハザードを除去する方法がとられている[11]。ここでは、積項線と入出力線をPLAの行及び列に割り当てる順番を考慮することで信号遅延を意識的に起こさせてハザードを回避する方法を提案する。

2. B-3 値論理関数[10]

本章では、あいまいさを取り扱うのに適したB-3値論理関数について概説する。詳細は文献[10]を参照されたい。

B-3 値論理は、2 値論理で定義されている演算AND(・), OR(∨), NOT(¯)を次のように3 値V = {0, 1/2, 1} に拡張定義した論理体系である(表1参照)。

$$\begin{aligned}
 x \cdot y &= \min(x, y) \\
 x \vee y &= \max(x, y) \\
 \bar{x} &= 1 - x
 \end{aligned}$$

x	y	0	1/2	1
0	0	0	0	0
1/2	0	1/2	1/2	1/2
1	0	1/2	1	1

x	y	0	1/2	1
0	0	0	1/2	1
1/2	1/2	1/2	1/2	1
1	1	1	1	1

AND:  $x \cdot y$

OR:  $x \vee y$

x	y	0	1/2	1
0	0	0	1/2	1
1/2	1/2	1/2	1/2	1
1	1	1	1	1

NOT:  $\bar{x}$

表1. B-3 値論理のAND,OR,NOTの真理値表

B-3 値論理では、2 値論理で成立する可換法則、結合法則、吸収法則、分配法則、べき等法則、ド・モルガンの法則は成立するが、

$$\text{相補法則: } x \cdot \bar{x} = 0, \quad x \vee \bar{x} = 1$$

が成立しないのが特徴的である。

どんな論理回路に対しても、その回路に対応するB-3 値論理式を求めることができる。更に、その論理式を加法標準形にも乗法標準形にも展開することができる。しかし、相補法則が成立しないので、ある変数  $x_i$  とその否定  $\bar{x}_i$  が項の中に同時に現れることがある。ここで変数  $x_i$  とその否定  $\bar{x}_i$  を文字という。単和項(積)項は、変数とその否定とが同時に現れないような和(積)項である。相補和(積)項は少なくとも1つの変数について、変数とその否定とが同時に現れる和(積)項である。相補項はクリーネの法則を用いて、すべての変数が現れる相補最大(小)項に展開することができる。

クリーネの法則

$$\begin{aligned}
 x \vee \bar{x} &= x \vee \bar{x} \vee y \cdot \bar{y} \\
 &= (x \vee \bar{x} \vee y) \cdot (x \vee \bar{x} \vee \bar{y}) \\
 x \cdot \bar{x} &= x \cdot \bar{x} \cdot (y \vee \bar{y}) \\
 &= x \cdot \bar{x} \cdot y \vee x \cdot \bar{x} \cdot \bar{y}
 \end{aligned}$$

が任意の  $x, y$  について成立する。

次式のように論理式が表された場合、その式は、主乗(加)法標準形であるという。

$$F = T_1 \cdot T_2 \cdot \dots \cdot T_n$$

$$(F = T_1 \vee T_2 \vee \dots \vee T_n)$$

但し、 $T_i$  は単和(積)項か、または相補最大(小)項であり、すべての  $i, j (i \neq j)$  について  $T_i \not\leq T_j$  を満足する。任意のB-3 値論理式は主乗(加)法標準形に一意的に展開できる。なお、これ以降は簡単のために、AND演算を表す  $\cdot$  を省略して表記することとする。

【例1】図1の回路のB-3 値論理式を求め、それを主加法標準形及び主乘法標準形に変形してみよう。

まず、図1のB-3 値論理式  $f$  は

$$f = x \bar{y} \vee y (\bar{y} \vee z) \quad \dots(1)$$

(1)式を主加法標準形に変形すると

$$\begin{aligned}
 f &= x \bar{y} \vee y \bar{y} \vee y z \\
 &= x \bar{y} \vee y z \vee (x \vee \bar{x}) y \bar{y} (z \vee \bar{z}) \\
 &= x \bar{y} \vee y z \vee x y \bar{y} z \vee x y \bar{y} \bar{z} \\
 &\quad \vee \bar{x} y \bar{y} z \vee \bar{x} y \bar{y} \bar{z} \\
 &= x \bar{y} \vee y z \vee \bar{x} y \bar{y} \bar{z} \quad \dots(2)
 \end{aligned}$$

また、(1)式を主乘法標準形に変形すると

$$\begin{aligned}
 f &= x \bar{y} \vee y \bar{y} \vee y z \\
 &= (x \vee y)(x \vee y \vee z)(x \vee \bar{y} \vee y)
 \end{aligned}$$

$$\begin{aligned}
& (x \vee \bar{y} \vee z)(\bar{y} \vee y)(\bar{y} \vee y \vee z) \\
& (\bar{y} \vee y)(\bar{y} \vee z) \\
& = (x \vee y)(\bar{y} \vee z)(y \vee \bar{y}) \\
& = (x \vee y)(\bar{y} \vee z)(x \vee y \vee \bar{y} \vee z) \\
& (x \vee y \vee \bar{y} \vee z)(\bar{x} \vee y \vee \bar{y} \vee z) \\
& (\bar{x} \vee y \vee \bar{y} \vee z) \\
& = (x \vee y)(\bar{y} \vee z)(\bar{x} \vee y \vee \bar{y} \vee z)
\end{aligned}$$

となる。

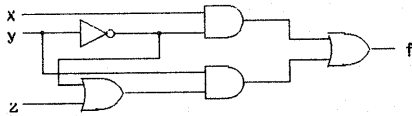


図1 論理回路の例

【定義1】[10]ベクトル  $\mathbf{a} = (a_1, a_2, \dots, a_n) \in V^n$  は、次のとき相補最大(小)項  $M$  と一致しているという。

- $a_i = 0 (1)$  iff  $M$  に  $x_i$  が存在し、  
 $\bar{x}_i$  が存在する。
- $a_i = 1 (0)$  iff  $M$  に  $\bar{x}_i$  が存在し、  
 $x_i$  が存在する。
- $a_i = 1/2$  iff  $M$  に  $x_i$  と  $\bar{x}_i$  の両方が存在する。

【例2】

$x \vee y \vee \bar{y} \vee \bar{z}$  は  $(x, y, z) = (0, 1/2, 1)$   
 $x \cdot y \cdot \bar{y} \cdot \bar{z}$  は  $(x, y, z) = (1, 1/2, 0)$   
 にそれぞれ対応する。

### 3. B-3 値論理を用いたハザードの検出[8][9]

本章では、B-3 値論理を用いて論理回路のハザードを検出する方法[8][9]を概説する。

組合せ論理回路において、ゲートや信号線上での信号遅延のため出力に過渡的な誤りを生じる。これをハザードという。

【例3】図2の回路で  $y = 1$  で、 $x$  が0から1に変化したとき、経路Aでの遅延がBでの遅延より大きいとき、出力  $f$  は  $1 \rightarrow 0 \rightarrow 1$  と変化する。この過渡的誤りを1ハザードという(図3(a)参照)。これに対して、入力の変化前と後での定常状態の出力が0で、過渡的に1パルスが生じるハザードを0ハザードという(図3(b)参照)。

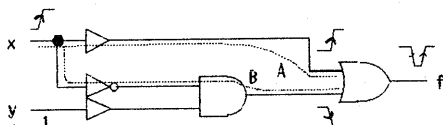


図2 1ハザードの生じる回路の例



図3 1ハザードと0ハザード

ハザードは、図2のように信号変化の前と後の定常状態での信号値が同じ静的ハザードと、前と後の値が異なる動的ハザードとがある。本稿では、静的ハザードのみ取り扱うこととする。B-3 値論理を用いてハザードを検出することが可能であり、次の定理1, 2が証明されている[8]。ここではB-3 値論理での真理値1/2が”信号変化”を表している。

【定理1】[8] 論理回路に対応するB-3 値論理式  $f$  に対し、それを主乗法標準形に変形する。この中に相補最大項があれば、それに対応する信号変化で論理1ハザードが存在する。

【例4】図2の回路のB-3 値論理式  $f$  は、

$$\begin{aligned}
f & = x \vee \bar{x} y \\
& = (x \vee \bar{x})(x \vee y) \\
& = (x \vee \bar{x} \vee \bar{y})(x \vee \bar{x} \vee y)(x \vee y) \\
& = (x \vee \bar{x} \vee \bar{y})(x \vee y)
\end{aligned}$$

よって、 $(x, y) = (1/2, 1)$ 、つまり  $y = 1$  で  $x$  の値が1から0、または0から1に変化したとき、論理ハザードが存在する。

ここで、”ハザードが存在する”というのは、必ずその入力変化でハザードが出るということではなく、ハザードが生じる可能性があるということ、つまり回路に適切な遅延があるときハザードが起きるということに注意されたい。

【定理2】[8] 論理回路に対応するB-3 値論理関数  $f$  に対し、それを主加法標準形に変形する。この中に相補最小項があるときのみ論理0ハザードが存在する。とくに相補最小項に対応する信号変化で論理0ハザードが存在する。

以上の定理を用いることで、1入力だけでなく多入力の入力変化で生じるすべての論理ハザードを検出できる[8]。

### 4. PLAにおけるハザード

PLA (Programmable Logic Array) は、2段のAND-OR論理式をそのまま実現できる。構造化設計手法向きであり、設計変更が容易、デザインルールチェックの省力化をはかれる等の優れた特徴がある。この為、VLSI内部で組合せ論理回路を実現

するのにも多用されている[2].

#### 4.1 PLAの構造

PLAの構造例を図4に示す。左側の列は入力線である。入力とその否定がAND平面に入り、行方向の積項線で論理積がとられる。そして、OR平面で各出力に関して必要な論理積の和がとられ、右側の列の出力線に出力される。なお、○印は、そこにトランジスタが存在することを示している。

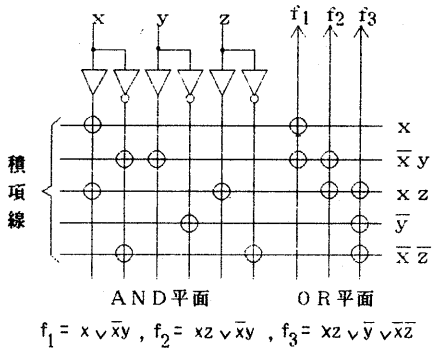


図4 PLAの構造例

#### 4.2 PLAのハザード

2段AND-OR論理回路では、論理1ハザードのみ存在し、論理0ハザードは存在しない。

各出力 $f_i$ は加法標準形

$f_i = \alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_m$  (但し $\alpha_j$ は単積項)で表現される。

この時、意味のある論理式では、各 $\alpha_j$ は単積項のみで、相補項は存在しないので、定理2より論理0ハザードは存在しない。つまり、PLAでは、論理1ハザードのみが存在する。

【例5】図4の回路の論理1ハザードを検出してみよう。各出力についての論理式を主乗法標準形に展開する。

$$\begin{aligned} f_1 &= x \vee \bar{x}y \\ &= (x \vee \bar{x})(x \vee y) \\ &= (x \vee \bar{x} \vee \bar{y} \vee z)(x \vee \bar{x} \vee \bar{y} \vee \bar{z})(x \vee y) \end{aligned}$$

入力変化(1/2, 1, 1)と(1/2, 1, 0)で論理1ハザードが存在する。

$$\begin{aligned} f_2 &= xz \vee \bar{x}y \\ &= (x \vee \bar{x})(x \vee y)(\bar{x} \vee z)(y \vee z) \\ &= (x \vee \bar{x} \vee \bar{y} \vee \bar{z})(x \vee y)(\bar{x} \vee z)(y \vee z) \end{aligned}$$

入力変化(1/2, 1, 1)で論理1ハザードが存在する。

$$\begin{aligned} f_3 &= xz \vee \bar{y} \vee \bar{x}z \\ &= (x \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y} \vee z) \end{aligned}$$

論理1ハザードは生じない。

ここでPLAにハザードが生じたときの問題点を考えてみよう。PLAの出力に順序回路(例えば、フリップフロップやラッチの入力)がつながっている場合、これが決定的な回路の誤動作を引き起こす可能性がある。この時ハザードを除去しなければならない。一方、前記の計算によりハザードが生じる可能性があってもかまわない場合がある。以下に列挙してみよう。

- (a) 出力の次段以降に組合せ回路がつながっている場合。
- (b) ある入力変化で出力に1ハザードが存在する可能性があっても、その入力変化は実際の回路ではありえない場合。
- (c) 入力変化に立ち上がり、立ち下がりの両方があるが、どちらか一方の変化のときは、ハザードが生じても出力には影響をあたえない場合。

以上のように、PLA上にある論理関数が実現されている場合、それに対応するB-3値論理式を主乗法標準形に変形することでハザードの存在とその条件がわかる。しかし、そのすべてのハザードが問題になるのではなく、実際には、その一部のみを除去すればよい。

#### 4.4 ハザードの除去

(1)ハザードの除去(1入力変化)

1入力変化によるハザードは、すべて論理ハザードなので、積項を追加することで容易に除去できる[11].

【例6】図4の出力 $f_2$ で実現される論理式 $f_2$ は、 $f_2 = \bar{x}y \vee xz$

である。これは図5のカルノー図で示される。

例5より入力変化(1/2, 1, 1)で1ハザードが生じる。そこでこの入力変化に対応する単積項 $yz$ (点線のループ)を付け加えて、 $f_2 = \bar{x}y \vee xz \vee yz$ とすることで1ハザードを除去できる。

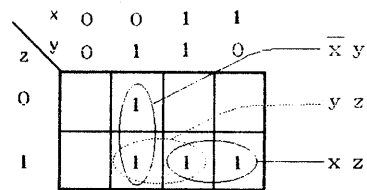


図5  $f_2$ のカルノー図

(2)ハザードの除去(2入力以上の変化)

2入力以上の変化の場合、上記の方法ではハザードを除去できない場合がある。

【例7】図5でのカルノー図の論理関数を

$$\text{論理式 } f_2 = \bar{x}y \vee xz \vee yz$$

で表した時、入力変化(1,1,1)→(0,1,0)の時を考えてみよう。この時、点線のように(1,1,1)→(1,1,0)→(0,1,0)と入力が変わる差でxよりもzの方が大きくなる変化するとyzという単積項を付け加えたにもかかわらず関数1ハザードが生じる。

この場合、適当に時間遅れを挿入して、入力の変化の方向を(1,1,1)→(0,1,1)→(0,1,0)にしなければハザードは回避できない。

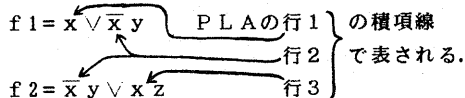
PLAにおいては、従来ハザードの除去といえば積項を付加することで、論理1ハザードのみを除去していた。そこで、次節ではPLAに適当な時間遅れを挿入してハザードを回避する方法を考えてみよう。

#### 4.5 PLAにおける信号遅延とハザードの回避

PLAでは、そのPLAがどのプロセスで作られるかによってその遅延の大きさが変化するが、信号線の長さが長くなればなるほど、また、駆動すべきトランジスタが多くなればなるほど信号遅延が大きくなると思われる。例えば、入力ラインにポリシリコンが用いられているとき、寄生容量を持つ配線の抵抗が、配線の長さの2乗に比例する固有の遅れをもたらす[12]。

【例8】図4のPLAで、 $f_1$ 、 $f_2$ のハザードのうち $y = z = 1$ で $x$ が $1 \rightarrow 0$ に変化する時のハザードのみを除去すると仮定する

( $y = z = 1$ のとき $x$ は $0 \rightarrow 1$ という入力変化はないとする)。



今、 $f_1$ について、 $x$ の立ち下がりでのハザードを起こさないためには、文字 $x$ を含んでいて、 $\bar{x}$ 及び $\bar{y}$ の文字を含まない積項線(行1)での信号遅延が、文字 $\bar{x}$ を含んでいて、 $\bar{y}$ 及び $\bar{z}$ の文字を含まない積項線(行2)での信号遅延より大きいとき、ハザードは生じない。上記の条件を出力 $f_1$ については、

$$\text{遅延(行1)} > \text{遅延(行2)}$$

で表そう。続いて、 $f_2$ については同様に、

$$\text{遅延(行3)} > \text{遅延(行2)}$$

で表せる。

信号遅延を調整するには、ランダムロジックではゲートの挿入などを行うことで実現できる。PLAでの調整法には次のような方法が考えられる。

(1) 行の並べかえ(総配線長の相違による調整)

(2) 論理関数を変化させないで駆動すべきトランジスタを付加する。

(3) 入力線の並べかえ(総配線長の相違による調整)。

このうち、1入力変化によるハザードを除去するには、(1)、(2)のみ利用できるが、2入力以上の変化では(3)も利用できる。

それでは、以上3つの方法を例を用いながら説明しよう。

#### 行の並べかえによる調整

図6で、ある積項を行1に配置した場合と行 $m$ に配置した場合で、 $2(m-1)$ 行分総配線長が異なる。この場合、信号遅延は

$$\text{遅延(行1)} < \text{遅延(行m)}$$

と考えられる。これを利用し、信号遅延をより小さくしたい行をPLAの上部に位置させることで、ハザードを除去することが可能である。

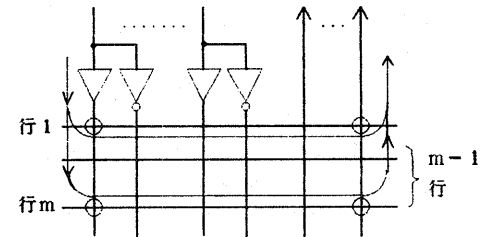


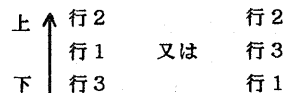
図6 総配線長による遅延の相違

【例9】例8でハザードの除去の条件は、

$$\text{遅延(行2)} < \text{遅延(行1)},$$

$$\text{遅延(行2)} < \text{遅延(行3)}.$$

であった。この2条件より、



の順に積項線を並べかえることで遅延を調整できる。上記の全ての条件が満足できるかどうかは、グラフを用いて判定することができる。例えば、以下のよう条件

$$\text{遅延(行2)} < \text{遅延(行1)} \quad \text{--- 条件1}$$

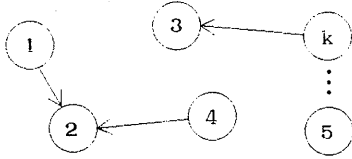
$$\text{遅延(行2)} < \text{遅延(行4)} \quad \text{--- 条件2}$$

⋮

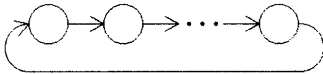
$$\text{遅延(行3)} < \text{遅延(行k)} \quad \text{--- 条件m}$$

があったとする。

まず、条件式に出現した行の番号を節点とするグラフを描く。次に、 $m$ 個の各条件毎に、不等号の大きい方の番号の節点から、小さい方の節点に向かって有向辺を結んでゆく。



もし、次のように循環が生じた場合は、上記の全ての条件を同時に満足することはできない。



以上の方法で大規模なPLAで各積項線の信号遅延に差を付けたいときは、その積項線と積項線との間にハザードに無関係な積項線を配置してその差を広げることが必要と思われる。

#### 駆動すべきトランジスタの付加

【例9】 例8で行3が実現している単積項  $xz$  は、 $f_1$  には現れていない。しかし、

$$f_1 = x \vee \bar{x}y$$

であるから、 $xz$  を加えても論理関数は変化しない。よって、 $f_1 = x \vee \bar{x}y \vee xz$

として、行3が駆動すべきトランジスタの個数を増やすこともできる。また、単積項の文字数を増やしても新しいハザードを増やすことなく、また、論理関数も変化させないことがある。これらの方法は、駆動すべきトランジスタの個数を増やして遅延を付け加えるという考えに基づいている。

#### 入力線の並べかえ

2入力以上の同時変化での関数ハザードを入力線の並べかえで回避してみよう。

【例10】 図5のカルノーマップで、 $(1,1,1) \rightarrow (0,1,0)$  と入力に変化するとき、入力線  $x$ 、 $z$  の位置を入れ換えることにより、行1の方が0から1の変化は、行3の1から0への変化より早く起こると考えられ、1ハザードを回避できる(図7参照)。

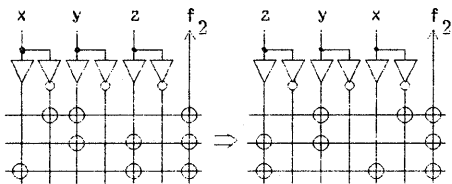


図7 入力線の並びかえ

この他に、PLAに付加回路をつけ、遅延を設定できる等のハザード回避法も考えられる。しかし、信号遅延は、総配線長とトランジスタの数、その他にも他のトランジスタの状態、プロセス等、種々の

要素が絡み合っている。この方法を実際に適用するには、ディスプレイ上でインタラクティブに行や列の入れ替えを実行し、その時は、SPICE等のシミュレータを用いながらハザードを除去する必要があると思われる。

#### 5. まとめ

PLAにおけるハザードについて考察した。PLA内部では、行や列を自由に入れ換えることが可能であることを利用し、信号遅延を調整してハザードを回避する考え方を提案した。

#### 参考文献

- [1] 樹下: 論理装置のCAD, 情報処理双書5, 情報処理学会(1981).
- [2] 笹尾: PLAの作り方使い方, 日刊工業新聞社(1986).
- [3] D.A. Huffman: The design and use of hazard-free switching networks, J.ACM, 4, 1, p.47 (Jan. 1957).
- [4] E.J. McClusky: Transients in combinational logic circuits, in Redundancy Technique of Computing System, Spartan Book, Washington D. C., p.9 (1962).
- [5] M. Yoeli and S. Rinon: Application of ternary algebra to the static hazards J. ACM, 11, 1, p.84 (Jan. 1964).
- [6] E. B. Eichelberger: Hazard detection in combinational and sequential switching circuits, IBM J., 9, 2, p.90 (1965).
- [7] 杉野, 他: 多入力変数の変化による静的ハザードの3値論理を用いた一検出法, 信学誌, 50, 6, p.997 (1967).
- [8] 向殿: 組合せ回路におけるハザードのB-3値論理を用いた考察, 信学論, Vol.61-D, No.9(1978).
- [9] M. Mukaidono: Fundamental properties of logic simulation based on three-valued logic, IEEE Workshop on Languages for Automation (1987).
- [10] 向殿: B-3値論理関数について, 信学論(D), 55-D, 6, p.355 (1972).
- [11] 室賀: 論理設計とスイッチング理論, 共立出版(1981).
- [12] ヒックス編: カスタムIC設計技術, 総研出版(1986).