

2入力論理セルを有するPLAのための論理合成手法

吉田 浩章[†] 山岡 寛明[†] 池田 誠[†] 浅田 邦博^{††}

[†] 東京大学 大学院 工学系研究科 電子工学専攻
^{††} 東京大学 大規模集積システム設計教育研究センター
〒 113-8656 東京都文京区本郷 7-3-1

E-mail: †{hiroaki,yamaoka,ikeda,asada}@silicon.u-tokyo.ac.jp

あらまし 本論文では、2入力論理セルを有するPLAのための論理合成手法を提案する。2入力論理セルを有するPLAは、ラッチ型センスアンプおよび寄生容量間の電荷分配を利用することによって高速動作および低消費電力を実現している。さらに、従来のAND/OR平面におけるAND/ORセルを任意の2入力論理セルに置き換えることが可能になっている。よって従来のPLAに比べて論理関数を効率的に実現することができる。提案手法は、新たな2入力論理関数抽出手法に基づいており、また多値論理合成やファクタリングといった既存のアルゴリズムを最大限利用しているため、簡単な実装および高速な計算を可能にしている。また提案手法の例題に対する計算機実験の結果を示す。

キーワード 論理合成, 多値論理, PLA

Logic Synthesis for PLA with 2-input Logic Elements

Hiroaki YOSHIDA[†], Hiroaki YAMAOKA[†], Makoto IKEDA[†], and Kunihiro ASADA^{††}

[†] Department of Electronic Engineering, University of Tokyo
^{††} VLSI Design and Education Center, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

E-mail: †{hiroaki,yamaoka,ikeda,asada}@silicon.u-tokyo.ac.jp

Abstract In this paper, we present a new logic synthesis method for PLA with 2-input logic elements. A PLA with 2-input logic elements can achieve low-power dissipation and high-speed operation by using latch sense-amplifiers and a charge sharing scheme. In addition, an arbitrary 2-input logic function is conveniently implemented in place of the conventional AND/OR planes. Therefore it can realize some classes of logic functions in a smaller circuit area. Since the proposed method makes full use of the existing multiple-valued logic minimization algorithms along with a new logic extraction technique for 2-input functions, it can be easily implemented and can handle practical circuits. The method has been implemented and the experimental results are presented.

Key words logic synthesis, multiple-valued logic, PLA

1. はじめに

近年、半導体プロセスの微細化により VLSI の大規模化が進むに従って設計は複雑さを増しており、短期間に信頼性のある回路を設計することは非常に難しくなっている。Programmable Logic Array(PLA)はその規則的な構造のため設計や面積・性能の見積もりが容易であることや、AND-ORの二段論理で構成されるため高速に動作する、などといった理由からしばしば利用されてきた。最近では IBM の 1-GHz のマイクロプロセッサのコントロール部分に使用されるなどといった例が報告されており [1], 再び注目を浴びている。また Khatri らによって PLA の多段接続を用いた低クロストークノイズを実現する VLSI 設計手法が提案されている [2].

しかし一方で、従来の PLA はセルベース方式といった多段論理を実現する方式に比べ大きな面積を必要とするといった欠点を持っており、そのため大規模な論理回路の実現には向いていない。このような背景から、論理関数を効率的に実現する PLA が提案されており、AND-OR-AND といった 3 段論理を実現する PLA や入力にデコーダを持った PLA [3] [4] などが提案されている。しかしこれらの PLA は従来の PLA に比べ動作速度が遅くなってしまうといった欠点を持っている。

2 入力論理セルを有する PLA はセンスアンプを用いた回路方式を採用することによって高速な動作を実現しており、また従来の PLA における AND セルや OR セルの代わりに任意の 2 入力論理関数セルを挿入することが可能になっている [5] [6]. そのため積和形の論理式だけでなく、より複雑な論理式をそのまま実現することが可能となっており、効率的に論理関数を実現できる。

本論文では、この 2 入力論理セルを有する PLA のための論理合成手法を提案する。本手法はデコーダを入力に有する PLA と同様、多値論理に基づいており、また既存のアルゴリズムを最大限に利用しているため簡単に実装することが可能である。また本手法は AND-XOR-OR 型 PLA の論理合成手法 [7] [8] の拡張であるとみることができる。

2. 2 入力論理セルを有する PLA

2 入力論理セルを有する PLA はラッチ型センスアンプと電荷分配を利用することによって特に多入力の場合に高速動作・低消費電力を実現している。また図 1 の LE に示すように AND/OR 平面における AND/OR セルを任意の 2 入力論理関数セルに置き換えることが可能である。図 2 に 2 入力論理セルを有する PLA の基本回路を示す。この図からわかるようにセルの置き換えはいくつかの配線をつなぎ変えることによって行われ、そのためほとんど面積や遅延に影響しない。AND 平面の出力は論理セルの出力の論理積であり、また同様に OR 平面の出力は論理セルの出力の論理和である。

2 入力デコーダを入力を持つ PLA は加算器などといった数学関数を非常に効率的に実現できることが知られている [3].

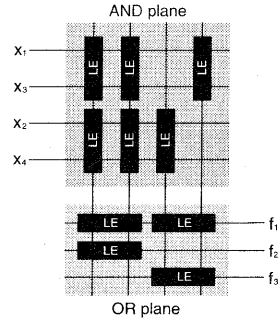


図 1 2 入力論理セルを有する PLA

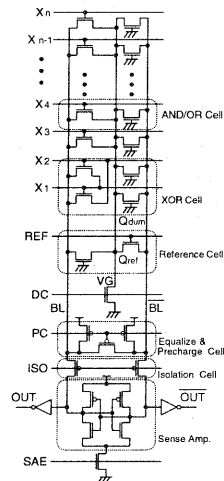


図 2 基本回路

また、そのための論理合成手法も開発されており、多値論理合成を用いた手法となっている。2 入力論理セルを有する PLA は図 3 に示すような AND-OR 平面間にも 2 入力デコーダを持った PLA と等価と見なすことが可能である。つまり、2 入力論理セルを有する PLA は 2 入力デコーダを入力を持つ PLA の拡張であり、より効率的に論理関数を実現することが可能であると思われる。

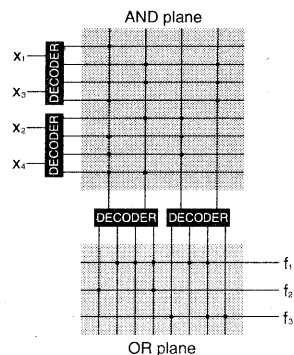


図 3 デコーダを有する PLA

3. 論理合成手法

3.1 用語の定義

多値論理に関する用語については文献[3][9]に詳しい定義について説明があるが、この節では本論文で使用する用語についてのみ簡単に説明する。

X_i を集合 $P_i = \{0, \dots, p_i - 1\}$ のいずれかの値をとる変数とする。このとき、リテラル $X_i^{S_i}$ を以下に示す論理関数と定義する。

$$X_i^{S_i} = \begin{cases} 0 & \text{if } X_i \notin S_i \\ 1 & \text{if } X_i \in S_i \end{cases}$$

ここで、 S_i は集合 P_i の部分集合である。リテラルの否定 $\overline{X_i^{S_i}}$ はリテラル $X_i^{\overline{S_i}}$ である。リテラルの論理積 $X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ を積項といい、積項の和を積和形論理式という。

積項 S と T のスーパーキューブ (supercube) は

$$X_1^{S_1 \cup T_1} X_2^{S_2 \cup T_2} \dots X_n^{S_n \cup T_n}$$

で求まる積項であり、 S と T を含む最小の積項である。同様に積和形論理式 F のスーパーキューブ $supercube(F)$ は F 中のすべての積項を含む最小の積項として定義される。

積項 S の積項 T に関するコファクタ S_T は

$$S_T = \begin{cases} 0 & \text{if } S_i \cap T_i = \emptyset \exists i \\ X_1^{S_1 \cup \overline{T_1}} X_2^{S_2 \cup \overline{T_2}} \dots X_n^{S_n \cup \overline{T_n}} & \text{otherwise} \end{cases}$$

で定義される。同様に積和形論理式 F の積項 T に関するコファクタ F_S は F 中のそれぞれの積項の S に関するコファクタの論理和である。

3.2 手法の概略

2入力デコーダを入力に持つ PLA と同様に、2入力論理セルを有する PLA における AND 平面の出力は4値論理の積項で表すことができる。すると、OR 平面の2入力論理セルの出力は2つの積項を入力とする任意の論理関数で表される。ここで2つの積項を S と T とすると、その論理関数は以下のように分類することができる。

- タイプ1: S (重み1)
- タイプ2: \overline{S} (重み1)
- タイプ3: $\overline{S} \cdot \overline{T}$ (重み2)
- タイプ4: $S \cdot \overline{T}$ (重み2)
- タイプ5: $S \oplus T$ もしくは $\overline{S \oplus T}$ (重み2)

図4 LE-term

これらの関数を LE-term と呼ぶ。2入力論理セルを有する PLA で実現可能な論理式は LE-term の和であるとおみることができる。重みはその関数を実現するために必要な AND 平面における積項の数を表している。ここで、上の分類に属さない関数 (例えば ST や $S+T$) は LE-term の和によって冗長なく表されることに注意したい。

本手法では、重みの和を最小にする、つまり必要な積項の数を最小にする LE-term の和の形の論理式を合成することを目的とする。論理合成の流れは、1) 与えられた積和形論理式に含まれる LE-term の抽出、2) 抽出した LE-term を考慮した多値論理合成、の2つのステップに分かれている。3.3節および3.4節では、この2つのステップについてそれぞれ詳しく説明する。

3.3 LE-term の抽出

本論理合成手法においてもっとも重要なステップは与えられた積和形論理式に含まれる LE-term の抽出である。本手法ではそれぞれのタイプの LE-term を異なる方法で抽出する。タイプ1の LE-term は与えられた論理関数の主項 (prime implicant) として求まるのであるので考慮しない。以下に残りのタイプの LE-term の抽出手法を示す。

タイプ2: 例えば、

$$\overline{X_1^{(1,2,3)} X_2^{(0,2)}} = X_1^{(0)} + X_2^{(1,3)}$$

のように積項の否定はリテラルの和とみなすことができる。よって、このタイプの LE-term は与えられた積和形論理式中から1つのリテラルからなる積項のみを取り出し、その否定を求めることで簡単に求めることができる。例えば、次の積和形論理式

$$F = X_1^{(0)} X_2^{(0,1)} X_3^{(0,2)} + X_1^{(0)} + X_2^{(0,1)} + X_3^{(0,2)}$$

は、タイプ2の LE-term を含む論理式

$$F = X_1^{(0)} X_2^{(0,1)} X_3^{(0,2)} + \overline{X_1^{(1,2,3)} X_2^{(2,3)} X_3^{(1,3)}}$$

に変形することができる。この論理式は2入力論理セルを有する PLA では2つの積項を必要とする。

タイプ3: 2つの積項の否定の積は、次のようにド・モルガンの定理から2つのリテラルの和の積に変形することができる。

$$\begin{aligned} & \overline{X_1^{(2,3)} X_2^{(2)}} \cdot \overline{X_1^{(0,1,3)} X_3^{(0,3)}} \\ &= (X_1^{(0,1)} + X_2^{(0,1,3)}) (X_1^{(2)} + X_3^{(1,2)}) \end{aligned}$$

よってこのタイプは、1) 与えられた積和形論理式中の2つ以下のリテラルからなる積項を取り出し、2) ファクタリングを行う、ことによって抽出できる。ファクタリングは文献[10]で提案されている手法を利用している。例えば、次の積和形論理式

$$F = X_1^{(0)} X_3^{(1,2)} + X_2^{(0,1,3)} X_3^{(1,2)} + X_1^{(1)}$$

についてファクタリングを行うことによって、次のようなタイプ3の LE-term からなる論理式を得る。

$$\begin{aligned} F &= (X_1^{(0,1)} + X_2^{(0,1,3)}) (X_1^{(2)} + X_3^{(1,2)}) \\ &= \overline{X_1^{(2,3)} X_2^{(2)}} \cdot \overline{X_1^{(0,1,3)} X_3^{(0,3)}} \end{aligned}$$

この論理式は2入力論理セルを有する PLA では2つの積項を必要とする。

タイプ4: タイプ4のLE-termの抽出は次の定理に基づいている。

定理 3.1 $S = X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ および T を積項, F を積和形論理式とすると,

$$\overline{F_S} \subseteq T \iff F \supseteq S \overline{T}$$

が成立する。ここで F_S は F の S に関するコファクタである。

証明 付録を参照

この定理より, 積項 S が与えられたとき $F \supseteq S \overline{T}$ を満たす積項 T を求めることが可能である。このようなLE-termは複数存在するが, 他のLE-termに含まれるようなLE-termは不必要であることから, 積項 T として $\overline{F_S}$ のスーパーキューブを用いることができる。例えば, 次の積和形論理式

$$F = X_1^{[0]} X_2^{[3]} X_3^{[0,1]} + X_1^{[0]} X_2^{[1]} + X_1^{[1,2]} X_2^{[1,3]}$$

において, 積項 S を $X_1^{[0,1,2]} X_2^{[1,3]}$ とする。このとき, $\overline{F_S}$ のスーパーキューブは次のように求まる。

$$\text{supercube}(\overline{F_S}) = X_1^{[0]} X_2^{[3]} X_3^{[2,3]}$$

よって F はタイプ4のLE-termを含む論理式

$$F = X_1^{[0,1,2]} X_2^{[1,3]} \cdot \overline{X_1^{[0]} X_2^{[3]} X_3^{[2,3]}}$$

で表される。この論理式は2入力論理セルを有するPLAでは2つの積項を必要とする。

この手法では積項 S を与える必要があるが, 可能な積項の数は変数の数を n とすると 2^n 存在し, そのすべてについて定理3.1の計算を行うことは現実的ではない。本手法では図5に示す手法によって生成された積項を積項 S として用いる。

Given: a sum-of-products F

Procedure cube.gen

$C = \{ \}$

for each product term $X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ in F

for each $1 \leq i \leq n$

for all subset T of $\overline{S_i}$

$C = C \cup \{X_1^{S_1} X_2^{S_2} \dots X_i^{S_i \cup T} \dots X_n^{S_n}\}$

end for

end for

end for

return C

end Procedure

図5 積項生成の手続き

これは以下のような考察に基づいている。まず先ほどのタイプ4のLE-termを再びド・モルガンの定理を用いて積和形に展開し, 最小化を行う。

$$X_1^{[0,1,2]} X_2^{[1,3]} \cdot \overline{X_1^{[0]} X_2^{[3]} X_3^{[2,3]}}$$

$$\begin{aligned} &= X_1^{[0,1,2]} X_2^{[1,3]} \cdot (\overline{X_1^{[0]} + X_2^{[3]} + X_3^{[2,3]}}) \\ &= X_1^{[1,2]} X_2^{[1,3]} + X_1^{[0,1,2]} X_2^{[1]} + X_1^{[0,1,2]} X_2^{[1,3]} X_3^{[0,1]} \\ &= X_1^{[1,2]} X_2^{[1,3]} + X_1^{[0]} X_2^{[1]} + X_1^{[0]} X_2^{[3]} X_3^{[0,1]} \end{aligned}$$

ここでの目的は最後の論理式から S の候補となるべき積項である $X_1^{[0,1,2]} X_2^{[1,3]}$ を見つけることである。最後の式の第1項である $X_1^{[1,2]} X_2^{[1,3]}$ は $X_1^{[0,1,2]} X_2^{[1,3]}$ と $\overline{X_1^{[0]}}$ の積に由来する。よって, 積項 $X_1^{[1,2]} X_2^{[1,3]}$ のリテラル $X_1^{[1,2]}$ について, このリテラルを含むリテラル $X_1^{[0,1,2]}$ に置き換える操作を行うことによって所望の積項が得ることができる。図5の手続きはこの操作を行っている。当然ながら, この手法では必ずしもすべての可能なLE-termを抽出することはできない。図5の手続きによって生成される積項の数は, 積和形論理式中の積項の数を m とすると, 多くとも $15mm$ であり, 定理3.1の計算の回数を減らすことができる。

タイプ5: タイプ5のLE-termもタイプ4の場合と同様に以下のような定理に基づいて抽出を行う。

定理 3.2 $S = X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ および T を積項, F を積和形論理式とすると,

$$\overline{F_S} \subseteq T \iff F \supseteq S \oplus T$$

が成立する。ここで $F_S, F_{X_k^{S_k}}$ はそれぞれ F の積項 S , リテラル $X_k^{S_k}$ に関するコファクタである。

証明 付録を参照

系 3.1 $S = X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ および T を積項, F を積和形論理式とすると,

$$\overline{F_{X_1^{S_1}} F_{X_2^{S_2}} \dots F_{X_n^{S_n}}} \subseteq T \iff F \supseteq S \oplus T$$

が成立する。ここで $F_S, F_{X_k^{S_k}}$ はそれぞれ F の積項 S , リテラル $X_k^{S_k}$ に関するコファクタである。

定理3.2の計算を行うとき, 積項 S はタイプ4の場合と同様に図5を用いて生成する。積項 S が与えられたときに, 定理3.1を満たす積項 T は複数存在するが, すべてを求めることは現実的ではないため, $\overline{F_S}$ を $F_{X_1^{S_1}} F_{X_2^{S_2}} \dots F_{X_n^{S_n}}$ をドントケアとして2段論理最小化を行い, その結果を積項 T としている。

3.4 LE-termを考慮した多値論理合成

本論理合成手法の最後のステップは, 前節で説明した手法で抽出したLE-termの和を求めることである。このときLE-termの重みを最小にするようなものを求める必要がある。その手続きを図6に示す。

図6では, 最終的な論理式を求めるためにブール除算の手法[11]を応用している。これを例を用いて説明する。次のような積和形論理式

$$F = X_1^{[0]} X_3^{[0,3]} + X_2^{[0,3]} X_3^{[2,3]} + X_1^{[2,3]} + X_3^{[1]}$$

Given: a sum-of-products F

Procedure LE-PLA

$X = \{\}$

Extract LE-terms of type 2 and type 3, and

put them into X .

$C = \text{cube_gen}(F)$

for each $S = X_1^{s_1} X_2^{s_2} \dots X_n^{s_n} \in C$

$T = \text{supercube}(F_S)$

$X = X \cup \{S\bar{T}\}$

if S satisfies $\bar{F}_S \subseteq F_{\bar{X}_1^{s_1}} F_{\bar{X}_2^{s_2}} \dots F_{\bar{X}_n^{s_n}}$ then

Simplify \bar{F}_S with $F_{\bar{X}_1^{s_1}} F_{\bar{X}_2^{s_2}} \dots F_{\bar{X}_n^{s_n}}$

as a don't care set and obtain T .

$X = X \cup \{S \oplus T\}$

end if

end for

$G = 0$

for each LE-term T_k in X

Create a new variable P_k to represent T_k .

$G = G + (T_k \oplus P_k)$

end for

Simplify F with G as a don't care set and obtain F_{LE} .

Replace P_k in F_{LE} with T_k .

return F_{LE}

end Procedure

図6 論理合成の手続き

において、以下の2つのLE-termが抽出されているとする。

$$\overline{X_1^{(0,1)} X_3^{(0,2,3)}} \quad (\text{タイプ2})$$

$$X_1^{(1,2,3)} \cdot \overline{X_1^{(0,1)} X_2^{(1,2)} X_3^{(2,3)}} \quad (\text{タイプ4})$$

ここでそれぞれのLE-termに対応する新たな変数 P_1, P_2 を用意し、次のようなドントケアを求める。

$$G = (X_1^{(0,1)} X_3^{(0,2,3)} \oplus P_1) + (X_1^{(1,2,3)} \cdot \overline{X_1^{(0,1)} X_2^{(1,2)} X_3^{(2,3)}} \oplus P_2)$$

この G をドントケアとして F の2段論理最小化を行うことにより、最終的な論理式を得る。

$$F_{LE} = X_1^{(0,2,3)} X_3^{(0,3)} + P_2 = X_1^{(0,2,3)} X_3^{(0,3)} + X_1^{(1,2,3)} \cdot \overline{X_1^{(0,1)} X_2^{(1,2)} X_3^{(2,3)}}$$

厳密に重みの和が最小である論理式は以下のようにして求まる。まず G をドントケアとして F のすべての主項を求める。次にそれぞれの主項に、対応する重みを割り当てる。これらの主項の重みの和を最小にする被覆を求める問題 (weighted covering problem) を解くことによって最終的な論理式を得ることができる。この手法では計算量が非常に大きくなってしまふ場合があるため、その場合には ESPRESSO-MV [9] などのヒューリスティック手法を用いることで効率的に解を求めることができる。この場合には必ずしも厳密解を得ることはできない。

3.5 LE-term の削減手法

前節で説明した論理合成手法では、抽出した LE-term の数が多い場合にはドントケアが非常に大きくなり、計算時間が大きくなってしまふ。したがって、抽出した LE-term の中からあらかじめ重要なものを取り出し、LE-term の数を削減する必要がある。本手法では、与えられた積和形論理式中の積項を自らの重みより多く含む LE-term のみをその候補としている。

4. 実験結果

3章で提案した論理合成手法を ESPRESSO-MV の一部として実装を行い、PLA ベンチマーク math 中の回路に対して実験を行った。回路に対してまず通常の2段論理最適化を適用し、次に ESPRESSO-MV に実装されているヒューリスティック [3] を用いて入力変数の割当てを行い多値論理合成を適用した。これを初期入力として本手法を適用した。その結果を表1に示す。表中で AND-OR, LE-AND-OR, AND-LE-OR, LE-AND-LE-OR はそれぞれ次のようになっている。

AND-OR: 通常の PLA。2値論理合成によって生成される。
LE-AND-OR: AND 平面のみに論理セルを持つ PLA。AND-OR に対して変数割当てを行った後、多値論理合成によって生成される。

AND-LE-OR: OR 平面のみに論理セルを持つ PLA。AND-OR に対して3章の手法を適用することによって生成される。

LE-AND-LE-OR: 両平面に論理セルを持つ PLA。LE-AND-OR に対して3章の手法を適用することによって生成される。

実験結果より、提案手法によってほとんどの回路について、2入力論理セルを有する PLA が4つの PLA の中で最も少ない積項数で論理関数を実現できたことがわかる。しかしながら PLA ベンチマークの indust や random 中の回路に対しては2入力デコーダを入力に有する PLA を含め、良好な結果を得ることができなかった。これは、2入力論理セルを有する PLA が特に数学関数の実現に向いていることを示しているものと思われる。

5. おわりに

本論文では、2入力論理セルを有する PLA のための論理合成手法を提案した。本手法は、多値論理合成やファクタリングといった既存の高性能なアルゴリズムを最大限利用することによって、簡単な実装および高速な計算を実現している。さらに、実験結果より従来の2入力デコーダを入力に有する PLA と比較してもより小さな積項数で論理関数を実現することができることを示した。

本手法では LE-term 間での積項の共有を考慮していないため、これを考慮することによってより小さな積項数からなる PLA を生成することが可能であると思われる。今後は、この LE-term 間での積項の共有を考慮したアルゴリズムの開発を行っていく予定である。

表1 実験結果

回路	AND-OR		LE-AND-OR		AND-LE-OR		LE-AND-LE-OR	
	積項数	時間 [秒]	積項数	時間 [秒]	積項数	時間 [秒]	積項数	時間 [秒]
Z5xpl	65	0.1	53	0.1	62	0.6	52	2.0
add6	355	0.5	37	0.1	325	93.2	37	1.5
addm4	200	0.4	109	0.3	193	4.9	99	13.4
adr4	75	0.1	17	0.0	69	0.7	17	0.1
dist	123	0.1	75	0.1	120	2.3	70	4.8
f51m	77	0.1	51	0.1	69	0.4	48	2.0
l8err	52	0.1	39	0.0	49	0.6	38	1.0
m181	42	0.1	30	0.1	40	0.2	28	14.6
mlp4	128	0.2	97	0.1	124	1.3	92	25.6
rd73	127	0.0	37	0.0	113	5.0	34	1.4
root	57	0.1	42	0.0	52	0.9	40	1.4
sqr6	49	0.0	42	0.0	49	0.1	40	0.3

付録: 定理 3.1 および 3.2 の証明

定理 3.1 および 3.2 を証明するためには、次の 2 つの補題を証明すれば十分である。

補題 5.1 S および T を積項, F を積和形論理式とすると,

$$F_S \supseteq \bar{T} \iff F \supseteq \bar{S}\bar{T}$$

が成り立つ。

証明 \implies : コファクタの定義より $F \supseteq SF_S \supseteq \bar{S}\bar{T}$

\impliedby : $F \supseteq \bar{S}\bar{T}$ の両辺の S に関するコファクタを求めることにより $F_S \supseteq \bar{T}$ □

補題 5.2 $S = X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ および T を積項, F を積和形論理式とすると,

$$F_{\bar{X}_1^{S_1} \bar{X}_2^{S_2} \dots \bar{X}_n^{S_n}} \supseteq T \iff F \supseteq \bar{S}T$$

が成り立つ。

証明 \implies : コファクタの性質より、任意の k について

$$F \supseteq \bar{X}_k^{S_k} F_{\bar{X}_k^{S_k}} \supseteq \bar{X}_k^{S_k} F_{\bar{X}_1^{S_1} \bar{X}_2^{S_2} \dots \bar{X}_n^{S_n}}$$

が成り立つ。よって

$$\begin{aligned} F &\supseteq (\bar{X}_1^{S_1} + \bar{X}_2^{S_2} + \dots + \bar{X}_n^{S_n}) F_{\bar{X}_1^{S_1} \bar{X}_2^{S_2} \dots \bar{X}_n^{S_n}} \\ &= \bar{S} F_{\bar{X}_1^{S_1} \bar{X}_2^{S_2} \dots \bar{X}_n^{S_n}} \supseteq \bar{S}T \end{aligned}$$

\impliedby : $S = X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ より,

$$F \supseteq \bar{S}T = \bar{X}_1^{S_1} T + \bar{X}_2^{S_2} T + \dots + \bar{X}_n^{S_n} T$$

が成り立つ。この両辺のコファクタを求めることにより、任意の k について $F_{\bar{X}_k^{S_k}} \supseteq T$ を得る。よって

$$F_{\bar{X}_1^{S_1} \bar{X}_2^{S_2} \dots \bar{X}_n^{S_n}} \supseteq T \quad \square$$

文 献

- [1] S. Posluszny, N. Aoki, D. Boerstler, J. Burns, S. Dhong, U. Ghoshal, P. Hofstee, D. LaPotin, K. Lee, D. Meltzer, H. Ngo, K. Nowka, J. Silberman, O. Takahashi, and I. Vo, "Design Methodology for a 1.0 GHz Microprocessor," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1998, pp. 17-23.
- [2] S. P. Khatri, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Cross-talk Immune VLSI Design using a Network of PLAs Embedded in a Regular Layout Fabric," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 412-418.
- [3] T. Sasao, "Input Variable Assignment and Output Phase Optimization of PLA's," *IEEE Trans. Computer*, vol. C-28, no. 9, pp. 879-894, Oct. 1984.
- [4] K.-C. Chen and S. Muroga, "Input Assignment Algorithm for Decoded-PLAs with Multi-input Decoders," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1988, pp. 474-477.
- [5] H. Yamaoka, M. Ikeda, and K. Asada, "A High-Speed PLA Using Array Logic Circuits with Latch Sense Amplifiers and a Charge Sharing Scheme," in *Proc. IEEE Asia South Pacific Design Automation Conf.*, Jan. 2001, pp. 3-4.
- [6] 山岡 寛明, "センスアンプを用いた配列型 CMOS 論理回路の研究," 修士論文, 東京大学大学院電子工学専攻, 2001 年.
- [7] H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic Synthesis for AND-XOR-OR type Sense-Amplifying PLA," *IEEE Int. Conf. VLSI Design & Asia South Pacific Design Automation Conf.*, Jan. 2002. (to appear)
- [8] 吉田, 山岡, 池田, 浅田, "排他的論理和を実現可能な 2 線式 PLA のための論理合成手法," DA シンポジウム, 2000 年 7 月.
- [9] R. L. Rudell and A. Sangiovanni-Vincentelli, "Multiple-Valued Minimization for PLA Optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 5, pp. 727-750, Sept. 1987.
- [10] L. Lavagno, S. Malik, R. K. Brayton, and A. Sangiovanni-Vincentelli, "MIS-MV: Optimization of Multi-level Logic with Multiple-valued Inputs," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 560-563.
- [11] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang, "MIS: A Multiple-Level Logic Optimization System," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 6, pp. 1062-1081, Nov. 1987.