設計自動化 35－2
（1986. 12. 16）

ＰＬＡ畳み込みアルゴリズム

畳み込みを考慮した論理式の簡単化

井口 幸洋・向殿 政男
明治大学 工学部

ＰＬＡ(Programmble Logic Array)の面積を削減するために，従来は組合せ論理関数の簡単化を行い，その結果に対して畳み込みをそれぞれ個別に行っている．本報告では，畳み込み後のＰＬＡの面積を最適にするには，畳み込みを考慮して論理関数の簡単化を行わなければならないことを例を用いて示す．まず論理関数のコストが悪いにもかかわらず畳み込み後のコストは良くなる例を示し，行畳み込み後にＰＬＡのコストが最小となる論理関数の簡単化方法を提案する．この方法は行畳み込みを論理関数の最小被覆問題を解きながら求める方法である．次に，論理関数のコストは同じでも畳み込み後のコストが異なる例を示し，畳み込みの時に畳み込みの候補となる個数が増えるように，簡単化の最後の過程で方向を変更する方法を提案する．

A PLA Folding Algorithm

Yukihiro IGUCHI and Masao MUKAIDONO
Faculty of Engineering, Meiji University
Higashimita, Tamaku, Kawasaki 214, Japan

For the PLA area reduction, hitherto the logic minimization and the folding processes have been done individually. This paper insists that the logic minimization process has to be taken into consideration with the folding process simultaneously. We give an example in which the cost of the logic minimization is not optimum but after folding it is optimum. We propose a new area reduction method in which the logic minimization and the folding are taken into account simultaneously to obtain the optimum row folded PLA, although it is feasible only for small variables. For large variables, we describe a method which reshpes the covers in the minimization process to increase the number of candidates for the folding.

１．はじめに

組合せ論理関数をＬＳＩ上で実現する方法にＰＬＡ(Programmable Logic Array)を用いる方法がある[1]．しかし，ＰＬＡを用いて論理関数をそのままＬＳＩ上に実現すると一般に隙間が多く，シリコン面積の損失や信号遅延の点から得策ではない．そこで，面積削減の方法が種々提案されている．例えば，多出力論理関数の簡単化を行なうもの[3)4)]，入力デコーダを用いるもの[5]，出力アクティブレベルの選択[5]，畳み込み[6)-10)]等が提案されている．

畳み込みとはＰＬＡの隙間を利用して行や列を畳み込み，面積を縮小する方法である[6]．行畳み込みは，互いに列（入力線，出力線）を共有しない２つの行（積項線）を切断点を用いて１本の行に配置する畳み込み法である．従来はＰＬＡの面積を縮小するために，論理設計時に組合せ論理関数の簡単化を行い，その結果に対して実装設計時に畳み込みをそれぞれ個別に施している．

本報告では，畳み込み後のＰＬＡの面積をより小さくするためには，畳み込みを考慮して論理関数の簡単化を行わなければならないことを例を用いて示す．３章では，論理関数のコストが最小でないにもかかわらず畳み込み後のＰＬＡの面積が，論理関数のコストが最小のものを畳み込んだ結果よりも小さくなる例を示す．そして，行畳み込み後にＰＬＡのコストが最小となる論理関数の簡単化法を述べる．この方法は論理関数の最小被覆問題を解きながら行畳み込みを求める方法である．４章では，論理関数の簡単化後のコストは同じでも畳み込み後のコストが異なる例を示し，畳み込み時に畳み込みの候補となる線対の個数を増やすように論理関数の簡単化の最後の過程で被覆の方向を変更する方法を述べる．

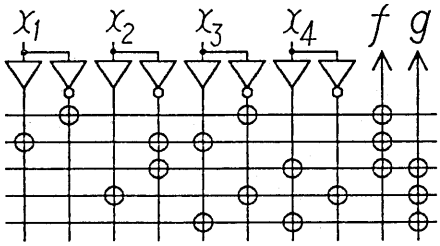
２．ＰＬＡの面積縮小

本章では，代表的な行畳み込みアルゴリズムについて概説し，次にＰＬＡの面積削減の過程について述べる．

2.1 従来の行畳み込みアルゴリズム

本節では代表的な行畳み込みアルゴリズムについて概説する．詳細については文献7)を参照されたい．

畳み込みを施す前のＰＬＡの例を図１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．



$$f = \overline{x}_1 \overline{x}_3 + x_1 \overline{x}_2 x_3 + \overline{x}_2 x_4$$
$$g = \overline{x}_2 x_4 + x_2 \overline{x}_3 \overline{x}_4 + x_3 x_4$$

図１　ＰＬＡの例

畳み込みを得るには，ＰＬＡパターンに対して行を頂点とし，互いに畳み込むことが不可能な頂点同志を無向辺で結んだ行交差グラフを描く．次に，畳み込みを表す有向辺（始点の頂点が左側に，終点の頂点が右側に畳み込まれることを表す）を付け加えていく．この時，有向辺の付け加えによって交互循環（無向辺と有向辺が交互に現れ一巡する道）を作らぬように行わねばならない[7]．このようにこのアルゴリズムでは，ある行を選び，次にその行と畳み込む相手を捜し，それまでの畳み込みによって生じた列の並び換えに矛盾しない時新しい畳み込みを付け加えるという方法をとっている．なお，ここで取り扱う行畳み込みでは入力線とその否定との間には切断点を設けないこととする．よって，以後入力線とその否定とを１本に併合した簡略化した図を用いることとする．

［例１］図２(a)のＰＬＡの例に行畳み込みを施してみる．

(1) 行交差グラフを構成する(図２(b)参照)． 行を頂点とするグラフを描き，互いに分離していない

頂点同士（互いに畳み込むことが不可能な行同志）を無向辺で結ぶ．

(2) 図２(c)に示すように畳み込みを表す有向辺(始点の頂点が左側に，終点の頂点が右側に畳み込まれることを表す）を交互循環（有向辺－無向辺－有向辺－・・・－無向辺で一巡する道）ができないように無向辺で結ばれていない頂点間に付け加える．

ａからｂに有向辺を付け加える．成功！

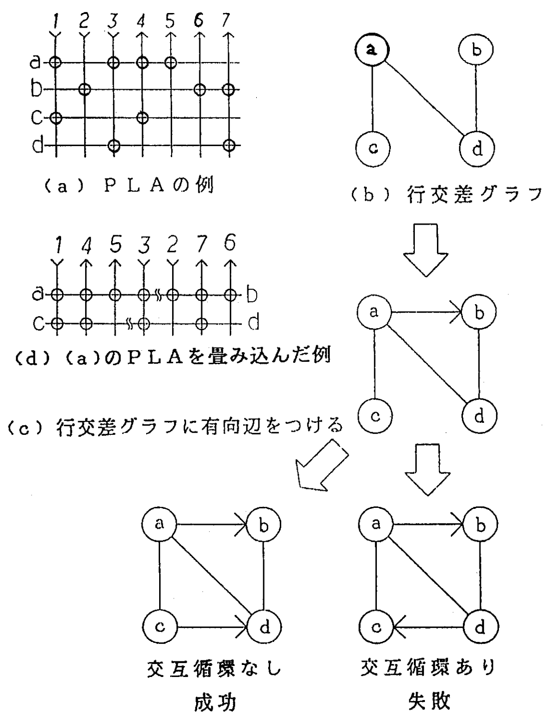
ｄからｃに有向辺を付け加えてみる．すると，

ａ→ｂ－ｄ→ｃ－ａという交互循環ができる．失敗！

ｄからｃの有向辺を取り去る．

ｃからｄへの有向辺を付け加えてみる．この時，交互循環はできない．成功！　これ以上有向辺を付け加えることはできない．よって，このＰＬＡでは行ａを行ｂの左側に，行ｃを行ｄの左側に畳み込むことができることがわかった．

(3)　(2)で得られた畳み込みを図２(d)に示す．



（ａ）ＰＬＡの例

（ｂ）行交差グラフ

（ｄ）（ａ)のＰＬＡを畳み込んだ例

（ｃ）行交差グラフに有向辺をつける

交互循環なし　成功

交互循環あり　失敗

図２．Ｈａｃｈｔｅｌの行畳み込みアルゴリズム

最適な行畳み込みを求める問題は極めて複雑である[12]．通常は上記のようにＰＬＡ内部の接続を無向グラフで表現し，それに発見的な方法で互いに畳み込む行を選び，畳み込みを表す有向グラフを交互循環経路を作らないように付け加えるという問題に帰着されている[7,8,10]．

2.2 論理関数簡単化と畳み込み

通常のＰＬＡの面積削減過程を図３に示す[11]．実現すべき組合せ論理関数を表す論理記述に対し，論理関数の簡単化を施す．これはＰＬＡで実現可能である．この結果に対して畳み込みが施され，畳み込まれたＰＬＡが得られる．このように，論理関数の簡単化と畳み込み過程が直列につながっている時，次のような未解決な問題「論理関数の簡単化と畳み込みとはどのようにかかわっているのか」がある[11]．

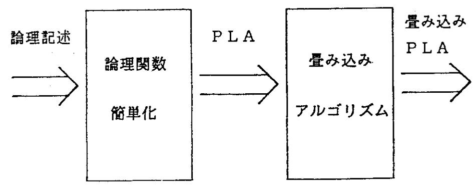


図３．ＰＬＡの面積削減過程

３．最適ＰＬＡ

図３のようにＰＬＡの面積を削減するとき，論理関数が最小のコスト（積項数が最小）のものに対して，論理関数のコストが最小でないにもかかわらず前者よりも行畳み込み後のコスト（ＰＬＡの面積：列数＊行数）が低い場合があることを示す．そして，行畳み込み後のコストが最小となる畳み込みを論理関数の最小被覆問題を解きながら求める方法を提案する．

［例２］６入力３出力論理関数を考える．出力f_2，f_3のカルノー図を図４(a)に示す．このときf_1を(b)のように被覆した場合，ＰＬＡの行数は９であり（(d)参照），(c)のように被覆した場合，ＰＬＡの行数は１０（(e)参照）である．(b)の被覆では，行畳み込みは不可能なのでＰＬＡの行数は９行．(c)では，２行畳み込め，(f)に示すように１０行から

８行に減っている．このように，論理関数のコストが最小でないにもかかわらず畳み込み後のコストが，論理関数のコスト最小のものを畳み込んだものより良い場合がある．

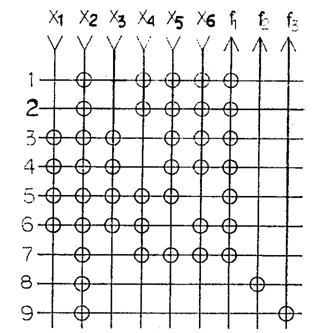
ＰＬＡの最適面積削減アルゴリズム

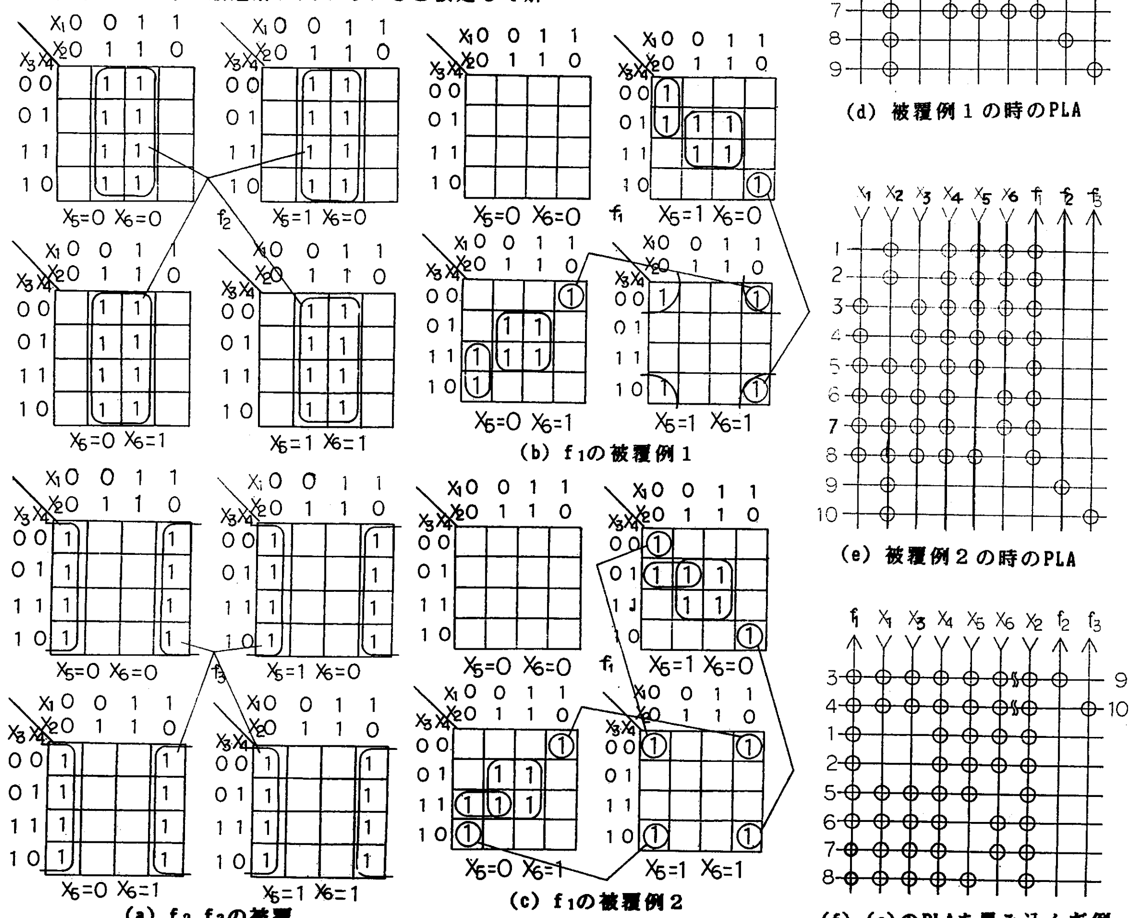
1)すべての多出力主項を求める．

2)縦に主項，横に最小項を並べた主項表を書く．
この時，主項間で互いに畳み込み可能な対を矢印で結び同一行に付け加える（矢印の左側を切断点より左に，右側を右に畳み込むことを表す）．ただし，多出力主項の出力部分は，他の主項と畳み込み可能なように出力部のすべての組合せに展開する（例３参照）．

3)行交差グラフを描く．

4)最小被覆問題[13]を解く． ただし主項表の畳み込み対はその対が最適解に入っていると仮定して解く場合と入っていないと仮定した場合とについて場合わけして解く．畳み込み対を最終解に入れたと考えたときは，行交差グラフでその対に対応した頂点に有向辺を付け加えて，交互循環が生じないときのみ処理を続行する．そして，その畳み込み対に矛盾する畳み込み対をすべて主項表から削除する．また，行が除去される場合にはその主項に対応する頂点とそれにつながる辺すべてを除去する．


(d) 被覆例１の時のPLA


(a) f₂,f₃の被覆

(b) f₁の被覆例１

(c) f₁の被覆例２

(e) 被覆例２の時のPLA

(f) (e)のPLAを畳み込んだ例

図４

［例３］図５のカルノーマップで示される４入力２出力の論理関数の時，主項表の縦の列に並ぶのは以下の通り，

1-0-/1-, -0-1/1-, --0-/-1, 10--/-1, -1-1/-1,
1--1/-1, 1-0-/11, 10-1/11, -001/11,
1-0-/1- → -1-1/-1, -1-1/-1 → 1-0-/1-,
-0-1/1- → --0-/-1, --0-/-1 → -0-1/1-,
-0-1/1- → 1-0-/-1, 1-0-/-1 → -0-1/1-,
10-1/1- → --0-/-1, --0-/-1 → 10-1/1-.

上記アルゴリズムは，最小被覆と行畳み込みを同時に求める方法であって，最適行畳み込みＰＬＡを得られるが，多くの手数を必要とする．次章では，積項の被覆の方向を変更することで行畳み込みの可能性を増やす方法を述べる．

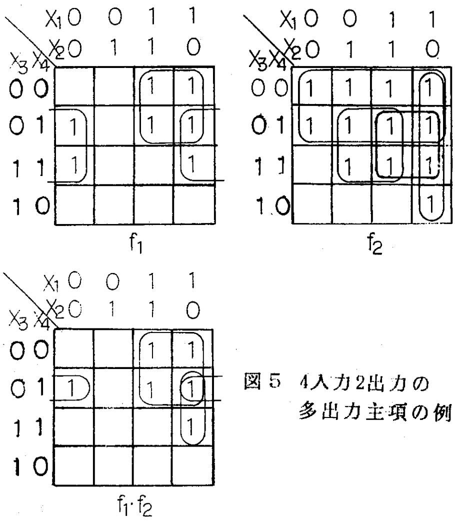


図５　4入力2出力の多出力主項の例

４．行畳み込みの可能性を増やす方法

本章で述べる方法は，図３に示される従来のＰＬＡの面積削減過程の論理関数の簡単化の後で，積項の被覆の方向を変更することによって，次につづく行畳み込みにおいて畳み込みの可能性を増やす方法である．この為には畳み込み候補の数を増やせば良く，行交差グラフで積項数を増やすことなく畳み込み不能を表す無向辺の数を減らせばよい．

［例４］５入力２出力の論理関数において簡単化の被覆が最初，図６(a) のカルノー図で表されているとする．この時，行畳み込みは不可能．今，ｆ₂の被覆を(b)のようにかえると畳み込み可能な候補が増え無向辺の個数が４個減る．(c)に畳み込み可能な頂点同志を点線で結んだ図を示す．この場合，２行畳み込める．畳み込んだ結果を(d)に示す．

この方法では，積項数を増やすことはないという条件の為に，４章の例２のように積項数は多くなるが，畳み込み後に解が良くなるような場合は求めることはできない．しかし，従来の論理関数の簡単化法と畳み込み法に手を加えることなく，積項の被覆を変えるだけという簡単な操作を付け加えるだけで畳み込み数を増やす可能性があるという特徴がある．

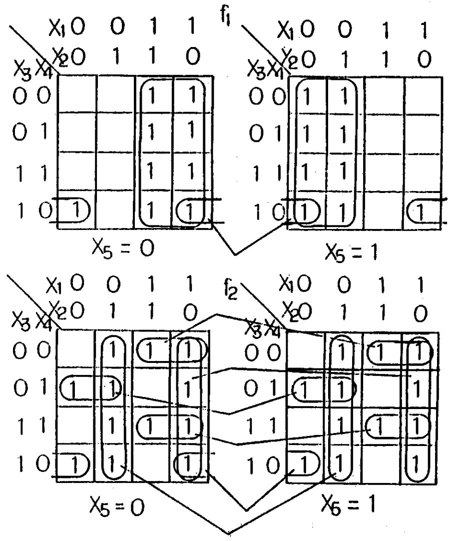


図６ (a)

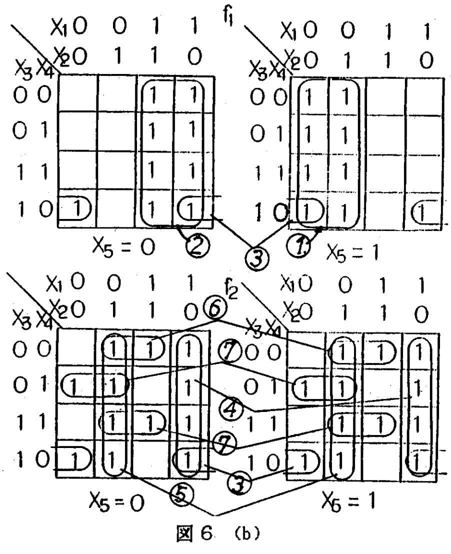


図６ (b)

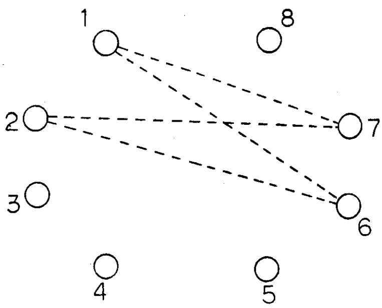


図６ (c)

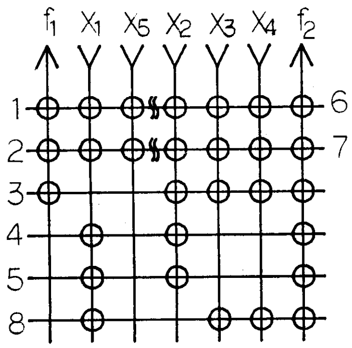


図６ (d)被覆例(b)の時のPLAを
畳み込んだ例

５．まとめ

ＰＬＡの面積削減を行うとき，畳み込みを考慮しながら論理関数を簡単化することによってＰＬＡの面積をより減らすことが可能なことを示した．３章で提案した面積縮小アルゴリズムについては最適な行畳み込みＰＬＡを得られるが，すべての素項を取り扱わねばならないのと最適な行畳み込みを求めるのはＮＰ－完全である[12]ので小規模な問題向きである．これに対し４章の方法は，積項の被覆の方向を変更することで畳み込みの可能性を増やす方法であるので，通常の論理簡単化プログラムに簡単に付加することが可能であり，中規模以上の問題向きである．

今後の課題は，ＰＬＡで実現される実際の論理関数について４章の方法を適用し，畳み込みを考慮しないで得たものに対しどれだけ改善されたかの評価実験を行うことである．

参考文献

1) Mead, C. and Conway, L.:Introduction to VLSI Systems, Addison-Wesley (1980).
2) McClusky, E.J. : Minimization of Boolean functions, Bell Syst. Tech. J., 35, 5, pp.1417-1444 (Nov. 1956).
3) Hong, S.J.: MINI : A heuristic Approach for Logic Minimization,IBM J. Res. Deve., Vol.18, No.5, pp.443-458 (1974).
4) Brayton, R.K., et al.:LOGIC MINIMIZATION ALGORITHMS FOR VLSI SYNTHESIS, Kluwer Academic Publishers (1984).
5) Sasao, T.: Input Variable Assignment and Output Phase Optimization of PLA's, IEEE Trans. Comput., Vol.C-33, No.10, pp.879-894 (1984).
6) Wood, R.A.: A High Density Programmable Logic Array Chip, IEEE Trans. on Conput., Vol.C-28, No.9, pp.602-608 (1979).
7) Hachtel, G.D. et al.,: Some Results in Optimal PLA Folding, ICCC,pp.1023-1027 (1980).
8) Hachtel, G.D. et al.,; An Algorithm for Optimal PLA Folding, IEEE Trans. on CAD of Int. Circ. Syst., Vol.1, No.2, pp.63-77 (1982).
9) Egan,J.R. and Liu,C.L.:Optimal Bipartite Folding of PLA,19th DAC,pp.141-146(1982).
10) Hachtel,G.D.:Techniques for Programmable Logic Array Folding,19th DAC,pp.147-155 (1982).
11) Lui, W. and Atkins, D.E. : Bound on the Saved Area Ratio due to PLA Folding,20th DAC, pp.538-544 (1983).
12) Luby, N., et al.: Some theoretical results on the optimal PLA folding problem", ICCC, pp.165-170 (1982).
13) 向殿，笹尾：スイッチング理論入門，朝倉書店 (1984).
14) 笹尾勤：ＰＬＡの作り方・使い方，日刊工業新聞社 (1986).