設計自動化 30－3
（1986. 1. 21）

ＦＰＬＡのワンカット行畳み込み

井口 幸洋 ・ 向殿 政男

（明治大学・工学部）

ＰＬＡ(Programmable Logic Array)の面積削減法に，隙間を利用してＰＬＡの行や列を畳み込む方法がある 1)-12)．その１つとして，著者らは，行畳み込みにおいて切断点を縦１列に揃えたワンカット行畳み込みアルゴリズムを提案した 11)．

畳み込み手法は，従来，マスクＰＬＡ向きの手法であったが，本報告では，ワンカット行畳み込みをフィールドでも可能にしたＦＰＬＡ(Field PLA) の構造を提案する．更に，その為の一畳み込みアルゴリズムについて簡単な実験を行なったので，その結果についても述べる．

§１．はじめに

組合せ論理関数をＬＳＩ上で実現するのに適した方法にＰＬＡ(Programmable Logic Array)を用いるものがある．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，通常，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 14)-16)，入力デコーダを用いるもの 17)，出力アクティブレベルの選択 17)，禁止入力パターンのDon't care化 18)，畳み込み 1)-12) 等が提案されている．

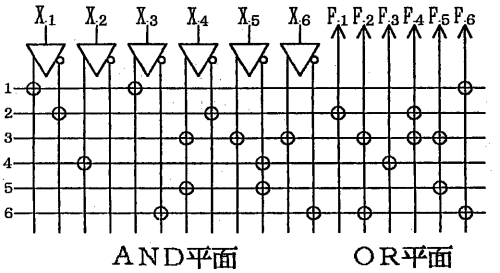
畳み込みとは，ＰＬＡの隙間を利用して行や列を畳み込み，トポロジー的に面積を縮小する方法である 1)．行畳み込みは，互いに列を共有しない２つの行（積項線）を切断点を用いて１本の行に配置する畳み込み法である．筆者らは，行畳み込みにおいて切断点を縦１列に揃えたワンカット行畳み込みアルゴリズムを提案した 11)．

畳み込み手法は，マスクＰＬＡ向きの手法であったが，本報告では，ワンカット行畳み込みをフィールドでも可能にしたＦＰＬＡ (Field PLA)の構造を提案する．更に，その為の一畳み込みアルゴリズムについて簡単な実験を行なったので，その結果についても述べる．本稿では，２章でワンカット行畳み込みについて述べ，３章で畳み込み可能なＦＰＬＡの構造，プログラム方法について論じる．次に，４章で最適なワンカット行畳み込みを求めるアルゴリズムについて紹介し，５章では実験例を示す．最後に問題点，これからの課題について述べる．

§２．ＰＬＡ行畳み込み

2.1 ＰＬＡのモデルについて

畳み込みを施す前のＰＬＡの例を図１に示す．左側の垂直線は入力線であり，入力とその否定がＡＮＤ平面に入り，そこで水平線により積項を形成する．そして，ＯＲ平面で各出力に関して必要な積項のＯＲがとられ，右側の垂直線の出力線に出力される．尚，○印は，そこにトランジスタが存在することを示している．



$F_1=\overline{X_1}\,\overline{X_4}$, $F_2=X_4X_5X_6+\overline{X_3}\,\overline{X_6}$, $F_3=X_2\overline{X_5}$, $F_4=\overline{X_1}\,\overline{X_4}+X_4X_5X_6$
$F_5=X_4X_5X_6+X_4\overline{X_5}$, $F_6=X_1X_3+\overline{X_3}\,\overline{X_6}$

図１．ＰＬＡの例

2.2 ＰＬＡワンカット行畳み込み

行畳み込みは，共通部分のない２つの積項線を見いだし，途中に切断点を入れることにより，２つの行対を１本の行に配置し，ＰＬＡの面積を削減する方法である．

畳み込みに於いて，切断点は，実現すべき論理関数や畳み込みアルゴリズムによってさまざまな位置に入る可能性がある．本論文で扱うワンカット行畳み込みは，畳み込み後のＰＬＡの構造をOR-AND-OR (AND-OR-AND)に固定し，切断点の存在を縦１列にのみ制限した行畳み込みで

ある 11)．図１のＰＬＡにワンカット(OR-AND-OR) 行畳み込みを施した例を図２に示す（以後，本稿ではOR-AND-OR 構造を取り扱う）．

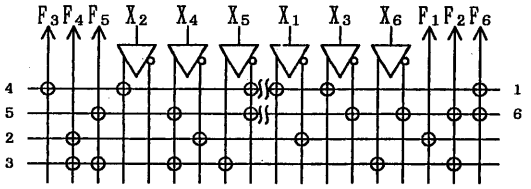


図２．図１のＰＬＡにワンカット行畳み込みを施した例

§３．ＦＰＬＡの畳み込み

２章で述べたように従来の方法による畳み込み後のＰＬＡでは，切断点は，さまざまな位置に入る．ワンカット行畳み込みに於いても，切断点は縦１列に入るが，それが何列目に入るか，又，左側の出力線，右側の出力線の個数はいくつになるかは，その論理関数や畳み込みアルゴリズムに依存する．よって，畳み込み手法は，マスクＰＬＡ向きの面積削減法であり，ＦＰＬＡに適した手法ではない．

ＦＰＬＡに於いては，マスクＰＬＡのように面積を縮小するのが目的ではなく，１つのチップでなるべく多くの論理関数を実現するのが目的となる．同一チップ内の限られたシリコン面積で，より多くの積項を実現できるとすれば，それは，有効なことである．例えば，２０積項を実現できるシリコン面積のＦＰＬＡがある時，実現しなければならない関数の積項数が２５個であったとする．今，このＦＰＬＡが畳み込み可能で，５行畳み込めれば，２チップの代りに１チップで実現する事ができる．

本章では，畳み込み可能なＦＰＬＡの構造を提案する．

まず，ヒューズ型のＦＰＬＡの構造を図３に示そう．ＦＰＬＡでは，ユーザーが，必要なダイオード及びトランジスタのみを残し，不要な部分のヒューズを溶断することによって，必要な論理関数を実現する．

ワンカット行畳み込みでは，切断点の存在が縦１列のみに許されている．この切断点の位置をさらにＡＮＤ平面内の中央に制限することを考える．図３のＦＰＬＡでまずOR-AND-OR 構造にし，ＡＮＤ平面内の積項線の中央にヒューズを設けたＦＰＬＡを提案する（図４参照）．すると，このＦＰＬＡでは，ワンカット行畳み込みがＡＮＤ平面中央のヒューズを溶断することで可能となる．

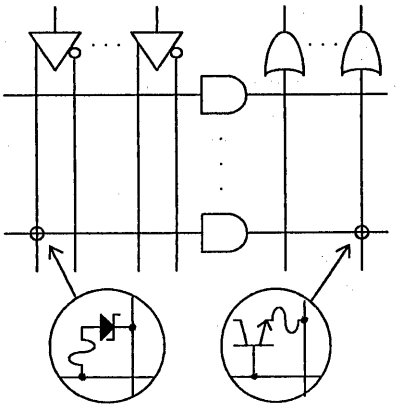


図３．ＦＰＬＡの構造例

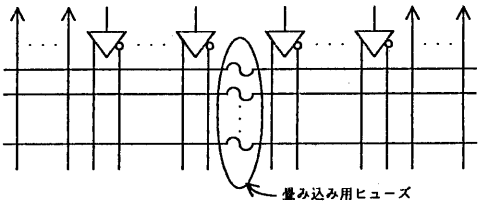


図４．ヒューズをＡＮＤ平面内に配したＰＬＡ

次に，このＦＰＬＡの畳み込み用ヒューズのプログラム法を考えてみよう．ｉ行目の畳み込み用ヒューズの溶断は，ｉ行目のヒューズをはさんで左と右との間でプログラム用の電流を流すことで行なうが，付加回路が必要である．通常のＦＰＬＡのプログラム用回路に変更を加える程度で実現できないか検討してみよう．

ＦＰＬＡに於いて不要なヒューズを溶断する方法は次の通りである．ｉ行ｊ列目の交点のヒューズを切断するには，ｎ行のうちからただ１つｉ行を，ｍ列の内からｊ列目のみを選び，プログラム用の電流を流すことで行なう．畳み込みヒューズ切断用に列を１本追加してみよう（図５参照）．

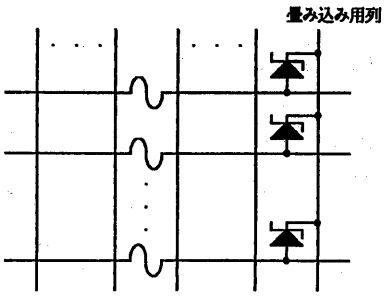


図５．ＦＰＬＡの畳み込み用ヒューズの溶断

ｉ行目の畳み込みヒューズを溶断する為には，畳み込み用の列（ｋ列）とｉ行目を選択し，プログラム用の電流を流すことで行なえばよい．尚，ＦＰＬＡの運用時には，ｋ列目はハイレベルにすれば，その存在は無視できる．

ここで，畳み込みヒューズ付きＦＰＬＡのプログラムの手順を示そう．

ＦＰＬＡプログラム手順

与えられたＰＬＡパターンに対して

１．４章で述べる畳み込みアルゴリズムを用いて畳み込みを求める．

２．１で得られた畳み込みをもとに，ＦＰＬＡの列と行にそれぞれ入出力と積項を割り当てる．

３．各交点にあるヒューズで不要なものを溶断する．

４．畳み込みヒューズを溶断することで畳み込みを得る．

§４．畳み込みアルゴリズム

本章では，ワンカット畳み込みの切断点が縦１列である特性を利用した畳み込みアルゴリズムを提案する．次に，３章のＦＰＬＡへの利用の為の変更点について述べる．

以下簡単の為にＰＬＡの構造を次のように簡略化して図示することとする（図６参照）．つまり入力線において，肯定と，否定の両方を併せて図示する．入力部に，２ビットデコーダを用いた場合も同様に扱う．尚，これは、ある入力線の肯定側と否定側の間に，切断点を許さないワンカット行畳み込みだから可能なのである．

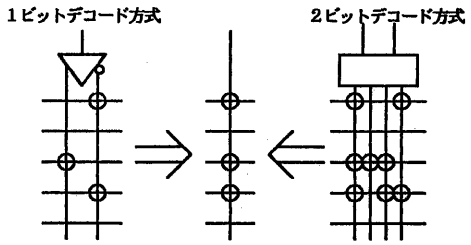


図６．畳み込み時の列の取り扱い方

4.1 従来の行畳み込みアルゴリズム

通常の行畳み込みでのアルゴリズムについて簡単に紹介しよう．詳細については文献５）を参照されたい．

まず，ＰＬＡパターンに対して，行を頂点とするグラフを描き，互いに畳み込むことが不可能な頂点同志を無向辺で結ぶ．次に，畳み込みを表わす有向辺（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わす）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われ一巡する道）を作らぬように行なわねばならない５）．

このように従来のアルゴリズムでは，ある行を選び，次にこ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付け加えるという方法をとっている．文献５）の OR-AND-OR構造での行畳み込みでは，次に畳み込む行の対を選ぶ時，行につながる無向辺の個数と，その行を選んだことによる次の行の対の選択への影響とを考慮した発見的手法を用いている．

4.2 ワンカット行畳み込みアルゴリズム

ワンカット行畳み込みにおいては，切断点は，縦１列である．畳み込み後においては，切断点の位置に対して左側と右側の列のそれぞれのグループ内では，列は自由に入れ換え可能であり，各列は切断点に対して左及び右に拘束されるだけという性質を持っている．ここで提案するアルゴリズムは，この性質を利用し，列を切断点に対し，左又は右に分けていくという方法であり，分枝限定法を用いて最適解を求めている．

まず，列を左，右へと決定していく時に，行はどのように左（右）に畳み込まれるか，又は，畳み込み不能になっていくかを例を用いて述べてみよう．

今，図７ａにワンカット行畳み込みを施してみよう．例えば，列１を左に選ぶ．すると，列１に○印を持つ行１，３及び４は，切断点より左側の列を持つため，畳み込み後，右側に畳み込まれることはない．

左（右）側の列と，その列に○印を持つ行にそれぞれ□（△）印を付けてみよう（図７ｂ参照）．ここで，行３は左側の列のみから構成されるので，左側に畳み込まれる．行３に畳み込み可能の印＊を付けて削除する．

次に例えば，列２を右側に選んでみよう（図７ｃ参照）．すると，行５は，右側の列のみで構成されているので右側に畳み込まれるから，＊印を付け削除する．行４は，左側の列と右側の列両方から構成されているので明らかに畳み込めないから削除する（図７ｄ参照）．

次に列３を左にとると，行１は左に畳み込まれ，最後に列４を右にとると，行２は右側に畳み込まれる（図７ｅ参照）．

以上で，列１と３を左に，列２と４を右に配置したとき，行１と３が左側に，行２と５が右側に畳み込まれることがわかった．畳み込みの結果を図７ｆに示す．

以下にいくつかのことばを定義しよう．

定義１．ある列の集合を左（右）に選んだとき，その各列に○印を持つ行を左（右）候補行と呼び，その行の○印すべてが左（右）列のみから構成されているとき，それを左（右）行と呼ぶ．更に，ある行が，左列と右列の両方から構成されているとき，その行は切断点の両側の列から構成されているので畳み込み不能であり，その行をデリート行と呼ぶ．

本畳み込み法では，各列を順次左又は右に選択していく．畳み込みを終了した後，各列は，左又は右の列に分離され，各行は，左行，右行又はデリート行に分離される．この時，畳み込める個数は，左行と右行の要素の個数のミニマムである．

次に，畳み込みを施す時に取り扱わねばならないＰＬＡの規模を小さくすることを考えてみよう．まず，列に関する支配関係を定義する．

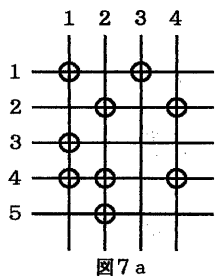


図７ａ

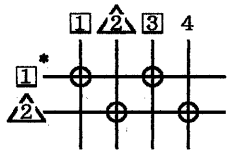


図７ｄ

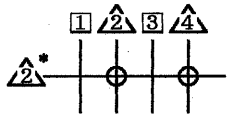


図７ｅ

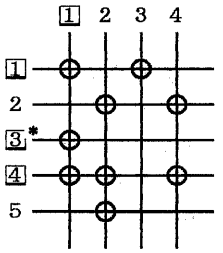


図７ｂ

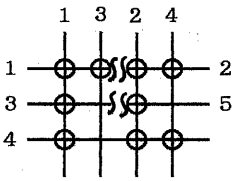


図７ｆ

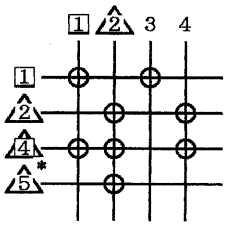


図７ｃ

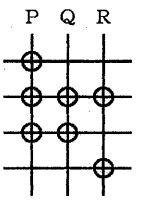


図８．列の支配関係

定義２．列Ｑが○印を持つ行に，列Ｐも○印を持つとき，列Ｐは列Ｑを支配するという．

例１．図８において，列Ｐは列Ｑを支配している．列Ｐと列Ｒ，及び，列Ｑと列Ｒの間には支配関係はない．

この支配関係を利用して，扱わねばならぬ列数を減らすことが可能であり，次の定理が成り立つ．

定理１．列Ｐを左（右）側に選択したとき，列Ｐに支配される列Ｑを左（右）側に選択しても，畳み込み可能な行数に影響は与えない．

（証明）今，列Ｐを左（右）側に選んだとすると，列Ｐに○印を持つ行は，すべて左（右）候補行となる．一度，左（右）候補行となった行は，左（右）に畳み込まれるか，もしくは畳み込み不可能な行である．なぜなら，行を構成するすべての列が左（右）列なら左（右）行とな

り，１つでも右（左）列が含まれれば畳み込み不能となるからである．よって，列Ｐを左（右）に選択後は，列Ｐに支配される列Ｑを左（右）に選択しても行には影響を与えない．（証明終）

この定理１は，最適解で列Ｐが左（右）側に入っているとすると，列Ｐに支配される列Ｑは，左（右）側に入っていても良いということを意味している．そこで，畳み込みを施す前にＰＬＡの互いに支配関係のある列で，支配されている列を省略すると，扱わねばならぬ列数を減らすことができる．

以下に，最適なワンカット行畳み込みを分枝限定法を用いて求める方法を述べる．

探索木

すべての場合をつくすには，図９のような探索木を用いれば良い．つまり，各枝を列に対応させ，その列を左又は右に選ぶことを各々，左，右に分枝させるのに対応させる．各頂点では，その状態での左行，右行，及びデリート行の集合を記憶させる．葉に至ったとき，行畳み込みが一つ求まることになる．

今，図９で，点線部は，無意味な探索を意味する．つまり，すべての列を左に畳み込むことは無意味であり，更に最初に列を左に選ぶか右に選ぶかは対称なので，一方のみを行なえば十分であり半分は省略できる．

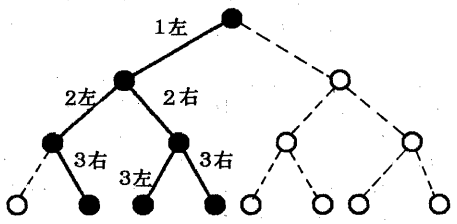


図9．列数3の時の探索木

次に，分枝限定操作について述べよう．

分枝限定操作

ある頂点に注目しよう．そこまでで求まっている最大の畳み込み数をＦmax とし，現在の左行，右行及び，未決定の行の個数をそれぞれＮｌ，Ｎｒ，Ｎｕとする．それから後，分枝していって求め得る最大の畳み込み数Ｆ’は，次の式で求まる．

｜Ｎｌ-Ｎｒ｜＜Ｎｕ の時，

$$F' = \lfloor (Nl+Nr+Nu)/2 \rfloor ,$$

｜Ｎｌ-Ｎｒ｜≧Ｎｕ の時，

$$F' = \min(Nl, Nr) + Nu .$$

よって，Ｆ’がＦmax に対しＦ’＞Ｆmax のときは分枝し，そうでないときは分枝しない（限定操作）ことにする．

分枝限定法を用いるときには，最初になるべく良い解を得ることが，探索量を減らすことにつながる．ここでは，ある頂点で分枝していくときに，以下の発見的手法を用いている．

発見的手法

次に選ぶ列は，残っている中で一番○印の多い列を選ぶ．分枝する方向は，現在迄に選んだ列の左候補行及び右候補行に対してその列を左（右）に選択することで，右（左）候補行から削除される行数の少ない方を選ぶ．もしそれでも同じときは，左列と右列の個数がなるべく等しくなるように分枝する方を選ぶ．

以上の方法をまとめ，次に示す．

アルゴリズム

１．列に関しての支配関係を調べ，被支配列を削除する．

２．一番○印の多い列を左に選択する．

３．以下，次々と分枝していく．各頂点に対しては，前記の限定操作及び発見的手法に従って分枝操作を行なう．

ＦＰＬＡ向きへの変更

以上のアルゴリズムは，一般的なＰＬＡに対するワンカット行畳み込み法であった．ここでは，ＦＰＬＡ向けへの変更点を述べる．

畳み込み前のＦＰＬＡの構造（図４参照）で，ヒューズに対して左側の入力列数及び出力列数をＣlin，Ｃlout，右側のをＣrin，Ｃrout，積項数をＣｔとおく（以下簡単の為Ｃlin ＝Ｃrin ＝Ｃin，Ｃlout＝Ｃrout＝Ｃout とする）．

与えられたＰＬＡパターンの入力数をＣ’in，出力数をＣ’out，積項数をＣ’tとすると，このパターンを先のＦＰＬＡで実現可能な為には，Ｃ’in ≦Ｃin＊２，Ｃ’out≦Ｃout ＊２であり，更にＣ’t－Ｃｔ 個畳み込まねばならない（Ｃ’ｔ－Ｃｔ≦０の時は畳み込みは不必要）．

畳み込み候補となりうる積項は，その入力文字数及び出力数が，各々Ｃin，Ｃout 以下でな

ければならない．よって，それ以外の積項に対応する行はデリート行とする．

さて，分枝操作のときであるが，その頂点での左列及び右列の個数は，ＦＰＬＡの入力列及び出力列数Ｃin，Ｃout 以下でなければならない．よって，方一方がＣin又はＣout と等しくなったら，もう一方のみに分枝していくしかない．　すべてをつくしたときに，畳み込み数がＣ't－Ｃt 未満ならば，このＦＰＬＡでは，その論理関数は実現不可能である．

ＦＰＬＡにおいては，最適解が必要なのではなく，１チップに納めることが目的なので，探索途中で最大畳み込みが，Ｃ't－Ｃt 以上になった所で脱出することができる．

以上の変更で，本章のワンカット畳み込みアルゴリズムはＦＰＬＡ向きのものとなった．

§５．実験結果

ＦＰＬＡは，フィールドで書き込む事を考慮し，１６ｂｉｔのパーソナルコンピュータ上で，４章のアルゴリズムについて簡単な実験を行なった．尚，使用言語にはＰＡＳＣＡＬを用いた．

表１に結果を示す．ＰＬＡ１は，図１のＰＬＡに畳み込みを施した例，ＰＬＡ２は，ＢＣＤから７セグメントへのデコーダ，ＰＬＡ３は，デコーダ，ＰＬＡ４は８ｂｉｔのＡＬＵ 9），ＰＬＡ５は，８ｂｉｔのＡｄｄｅｒ 9）である．これらに対して，発見的手法を用いた通常のOR-AND-OR 畳み込み 6），6)の行の対の選択規則はそのままでワンカット行畳み込みに変更した畳み込み法，本畳み込み法の３つの方法を適用したときの畳み込めた行数とＣＰＵ時間とを求めた．なお，ＰＬＡ３の畳み込みを施す前と後のパターンを図１０ａ及び図１０ｂに示す．

本稿で提案した畳み込み法は，表１で示した程度のＰＬＡに対しては，発見的手法で準最適解を求めた場合と同程度の時間で最適解を求められる．

§６．まとめ

マスクＰＬＡ向きの手法である畳み込みを，ＦＰＬＡにも応用できるように，切断点の位置に制約を加えたワンカット畳み込みアルゴリズムを紹介し，ＡＮＤ平面中央に畳み込み用ヒューズを設けたＦＰＬＡを提案した．

今後の課題としては，畳み込み用ヒューズのプログラム用の付加回路の大きさを見積り，多くの実際のＰＬＡパターンに対し畳み込む実験を行なうことである．

謝辞　ＰＬＡパターンを提供していただきました大阪大学の笹尾勤先生に深謝致します．

参考文献

1) R.A.Wood, "A high density programmable logic array chip", IEEE Trans. on Conputers, vol c-28, No. 9, pp. 602-608, Sep. 1979.

2) G.D.Hachtel, Sangivanni-Vincentelli, A.R. Newton, "Some results in optimal PLA folding" ICCC, pp. 1023-1027, 1980.

3) G.D.Hachtel, A.R.Newton, A.L.Sangiovanni-Vincentelli, "An algorithm for optimal PLA folding", IEEE Trans. on CAD of Int.Circ. and Syst., vol. 1, vol. 2 , pp63-77, April 1982.

4) J.R.Egan, C.L.Liu, "Optimal bipartite folding of PLA", 19th DAC, pp. 141-146, 1982.

5) G.D.Hachtel, "Techniques for programmable logic array folding", 19th DAC, pp. 147-155.

6) M.Luby, et al., "Some theoretical results on the optimal PLA folding probrem", ICCC, pp. 165-170, Oct 1982

7) G.D.Micheli, et al., "PLEASURE: A computer program for Simple/multiple constrained/unconstrained folding of programmable logic array", 20th DAC, pp. 530-537, 1983.

8) W.Lui, D.E.Atkihs, "Bound on the saved area ratio due to PLA folding", 20th DAC, pp. 538-544, 1983.

9) Y.S.Kuo, C.Chen, T.C.Hu, "A heuristic algorithm for PLA block folding" , 22nd DAC, pp. 744-747, 1985.

10) T.P.Moore, A.J.de Geus, "Simulated annealing controlled by rule-based expert system", ICCAD-85, pp. 200-202, Nov. 1985.

11) 井口，向殿，”ＰＬＡの畳み込みにおける一手法”情報処理学会春季全国大会6H-4, 1985.

12) 井口，向殿，”インバータを用いたＰＬＡ畳み込み”，情報処理学会，設計自動化研究会資料２６－３，1985.

13) C.Mead, L.Conway, "Introduction to VLSI Systems", Addison-Wesley, 1980.

14) S.J.Hong, "MINI : A heuristic approach for logic minimization", IBM J. Res. Deve., pp. 443-458, 1974.

15) R.K.Brayton, et al., "A comparison of logic minimization strategies using ESPRESSO : an APL program package for partitioned logic minimalization", ISCAS-82, pp. 42-48, 1982.

16) M.R.Dagenais, et al.,"The McBOOLE logic minimizer",22nd DAC,pp.667-673,1985.
17) T.Sasao, "Input variable assignment and output phase optimization of PLA's", IEEE Trans.Comput.,Vol.C-33,No.10,pp.879-894,1984
18) 島，宇野，久保，河崎，”ＰＬＡ面積縮小の為の一手法”情報処理学会春季全国大会6H-3,1985.
19) 南谷，”ＰＬＡの使い方”産報出版，電子科学シリーズ，No.80,1978.
20) 三上，”ＰＬＡ活用のための新しい設計手法”電子科学,Vol.35,No.2,pp.13-77,1985.

				OR-AND-OR 6)		ONE-CUT 6)		本畳み込み法		
	入力列数	出力列数	積項数	畳み込み行数	CPU[秒]	畳み込み行数	CPU[秒]	畳み込み行数	CPU[秒]	
ＰＬＡ１	6	6	6	2	０．３	2	０．３	2	０．５	図１のＰＬＡ
ＰＬＡ２	4	7	１３	3	０．９	3	０．７	3	２．０	BCD to 7Seg.
ＰＬＡ３	9	２３	３５	１６	１６．	１６	１４．	１６	５．７	Decoder
ＰＬＡ４	１０	２３	５３	２０	４５．	１８	２７．	１８	２３．	8 bit ALU
ＰＬＡ５	8	１８	２２	7	４．７	5	２．５	8	５．２	8 bit Adder

表１．実験結果

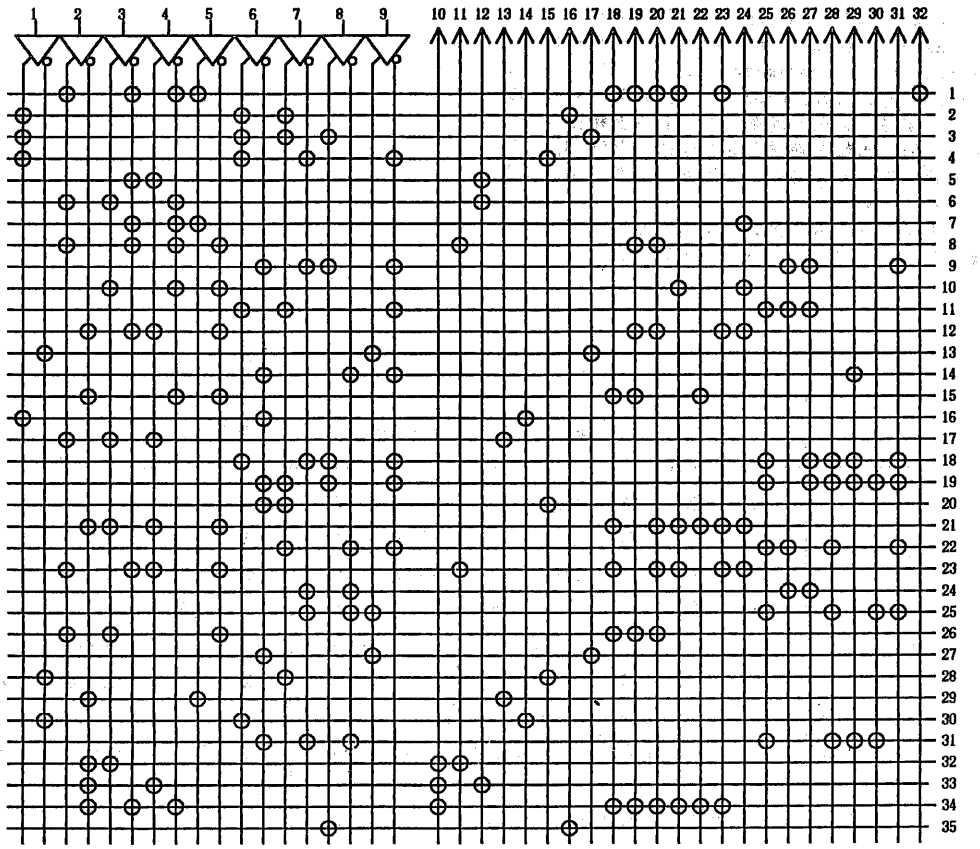

1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35

図１０ａ

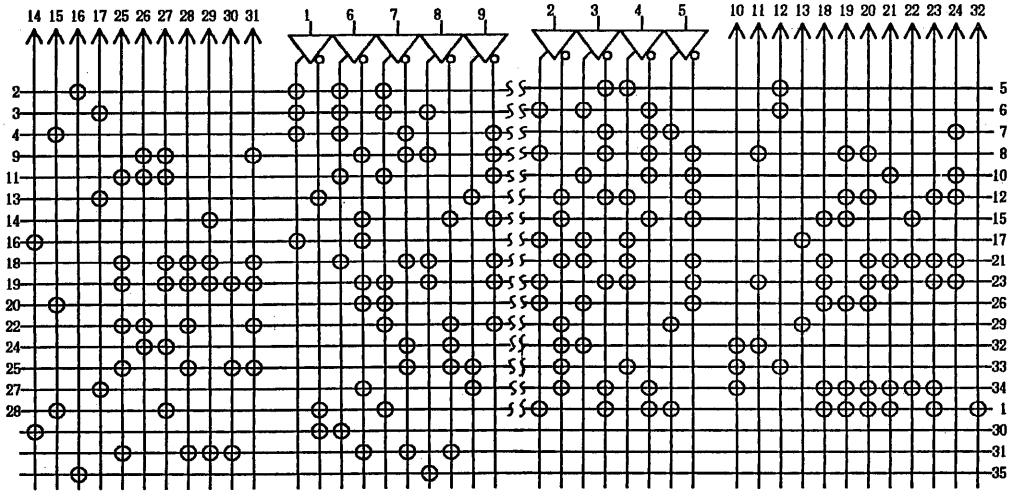

14 15 16 17 25 26 27 28 29 30 31
1 6 7 8 9
2 3 4 5
10 11 12 13 18 19 20 21 22 23 24 32
2 3 4 9 11 13 14 16 18 19 20 22 24 25 27 28
5 6 7 8 10 12 15 17 21 23 26 29 32 33 34 1 30 31 35

図１０ｂ