2003－ＳＬＤＭ－110　(1)
2003／5／15

PCA 可変部への組合わせ回路のマッピング手法の検討

稲森　稔†　永見　康一†　名古屋　彰†

† NTT 未来ねっと研究所　〒239–0847 神奈川県横須賀市光の丘 1-1
E-mail: †{inamori.minoru,nagami.kouichi,nagoya.akira}@lab.ntt.co.jp

あらまし　本論文では、PCA 可変部に組合わせ回路をマッピングする手法について提案する。PCA は動的再構成機能を有する汎用計算機アーキテクチャであり、再構成可能な部分 (可変部) と再構成を支援する部分 (組み込み部) から構成される。我々は、PCA 用の機能設計支援システムとして、可変部に有限状態機械をマッピングする手法を既に提案している。今回は、組合わせ回路部に注目し、面積最小化を目指してマッピングする手法を検討した。いくつかの回路をマッピングすることによりその有効性を確認した。

キーワード　組合わせ回路、マッピング、LUT

A Method of Mapping Combinational Logic into PCA Plastic Part

Minoru INAMORI†, Kouichi NAGAMI†, and Akira NAGOYA†

† NTT Network Innovation Laboratories　1-1 Hikarinooka Yokosuka-shi, 239–0847 Japan
E-mail: †{inamori.minoru,nagami.kouichi,nagoya.akira}@lab.ntt.co.jp

Abstract　This paper proposes a method of mapping combinational logic into PCA plastic parts. The PCA is a general-purpose computer architecture based on the dynamic reconfiguration of functions. It consists of a two-dimensional array of PCA cells, each of which consists of a plastic part and a built-in part. The plastic part is a programmable logic, while the built-in part supports the dynamic reconfiguration. We have already developed a method of mapping finite stats machine into PCA plastic part as a part of logic synthesis system for PCA. Here, we create a method to minimize the combinational circuit size. Some benchmark suits are mapped to estimate the efficiency. Experimental results show that the method can effectively reduce circuit size.

Key words　Combinational Logic, Mapping, LUT

1. はじめに

我々は、動的再構成機能を有する汎用計算機アーキテクチャである Plastic Cell Architecture(PCA) を提案している [1]。PCA は、可変部 (Plastic Part) と組み込み部 (Built-in Part) の組 (PCA セル) の二次元アレイ構造をとる。可変部は、機能をプログラマブルに構成可能な領域であり、一般の FPGA と同様に処理の実行を担う。本稿では、機能の単位をオブジェクトと呼ぶ。オブジェクトは、任意個の可変部に構成可能である。一方、組み込み部は、オブジェクト間通信、オブジェクトの生成と消滅のための機能を持つ。既に、PCA に基づく LSI(PCA-1) [2] 上で、アプリケーションの実装の検討が行われている。図 1に PCA-1 の構成を示す。

可変部は、内部に四個の 4 入力 1 出力の LUT(Look-up Table) を持つ基本セル (Basic Cell) の二次元構造をとる。可変部でのオブジェクトの構成は、コストの面から最小の基本セル数が望ましい。我々は、PCA 用論理合成システムの開発の一環としてハードウェア記述言語にて記述された有限状態機械を可変部の回路情報に変換する手法を既に提案している [3] [4]。これらの手法は、テンプレートベースのマッピング手法であり、有限状態機械が与えられると、状態レジスタと入出力信号の位置はパラメトリックに決定される。したがって、そのマッピングに要する基本セル数は、組合わせ回路のマッピング能力に依存する。そこで、今回、組合わせ回路のマッピングにおいて、基本セル数最小を目指す手法を検討したので報告する。

本稿の構成を以下に示す。第 2 章では、既に提案している有限状態機械のマッピングにおける組合わせ回路のマッピング手法について説明し、その後に今回提案するシャノン展開に基づく手法を提案する。第 3 章では、本手法の有効性を検証するために行った実験とその検討結果について述べる。最後に第 4 章でまとめる。

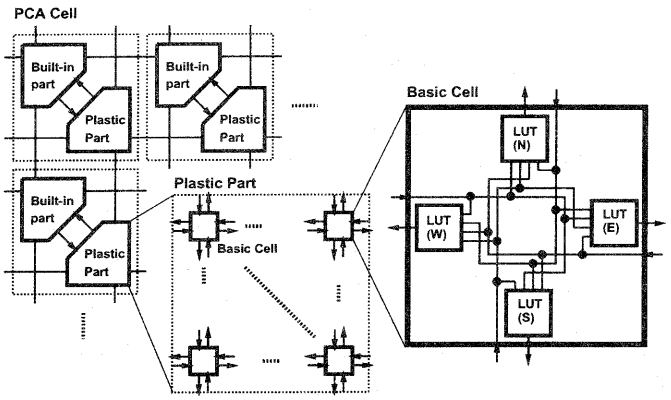

図1 PCA-1 の構成

Fig. 1 Construction of PCA-1

2. 組合わせ回路のマッピング

2.1 従来のマッピング手法

可変部は LUT の二次元アレイ構造をとるため、組合わせ回路の配置の自由度が極めて高い。そこで、既に提案している PCA 可変部への有限状態機械のマッピング手法 [3] では、テンプレートベースの手法をとっている。図 2(1) は、その組合わせ回路部分を示したものであり、PLA を用いた場合の組合わせ回路のマッピングを表している。下二行が AND 平面であり、東西から入力変数が入力する。AND 平面内の東西の LUT は、入力変数を東西に伝搬させるために使用され、北側の LUT は、積項を構成するために使用される。一方、上一行は OR 平面であり、東西から論理関数が出力する。OR 平面内の東西の LUT は、論理関数を構成するために使用される。本テンプレートでは、AND 平面、OR 平面とも南側の LUT は使用されない。

PLA に基づくマッピングでは、一列の基本セルに一つの積項をマッピングする。論理関数のマッピングに必要な基本セル列数を削減するため、本テンプレートの AND 平面において、一列の基本セルに複数の積項をマッピングすることを考え、筆者らは Maitra Term 法 [5] を提案している [3]。Maitra Term 法の処理の概略を図 2(1) を用いて示す。

（1） 論理関数 z が、入力変数 $i2$ または $i3$ に依存するか調べる。依存する場合、$i2$ と $i3$ によって表される部分論理 $map0$ を北に伝搬させる。依存しない場合は、北には何も伝搬させない。

（2） 論理関数 z が、$map0$、入力変数 $i0$ または $i1$ に依存するか調べる。依存する場合、その論理関数 $map1$ を北に伝搬させる。ここで、$map1$ の依存変数集合 $sup(map1)$ には、$map0$ が含まれていなければならない。依存しない場合は、$map0$ を北に伝搬させる。

（3） 論理関数 z から、マッピングされた論理関数 $map0$ を除去する。未マッピングの論理関数があれば、(1) に戻る。なければ処理を終了する。論理関数 z のマッピングには、この処理を繰り返す回数だけの基本セル列数を要する。

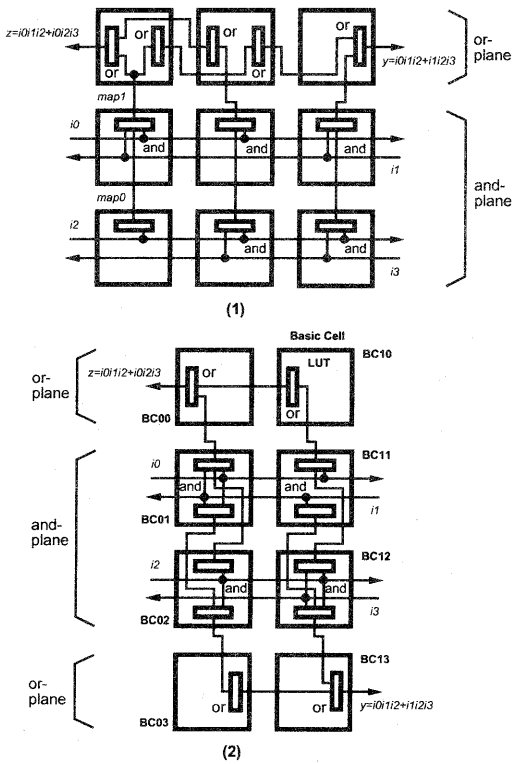

図2 組合わせ回路部分のテンプレート

Fig. 2 Templates for combinational logic

2.2 シャノン展開に基づく手法

今回、組合わせ回路のマッピングのためのテンプレートとして、図 2(2) を採用する。AND 平面では、東西に入力変数を

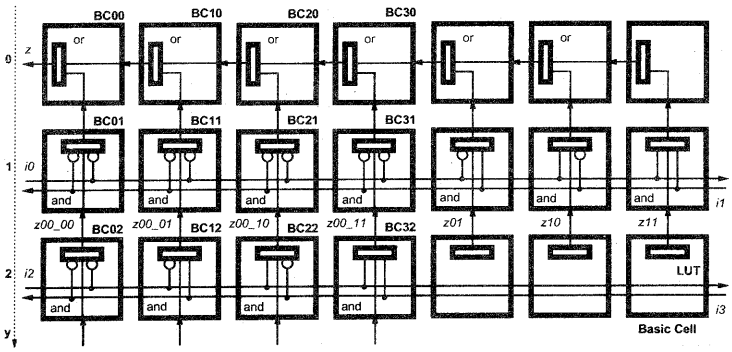

図3　シャノン展開法の手順

Fig. 3　Procedure of Shannon's expansion method

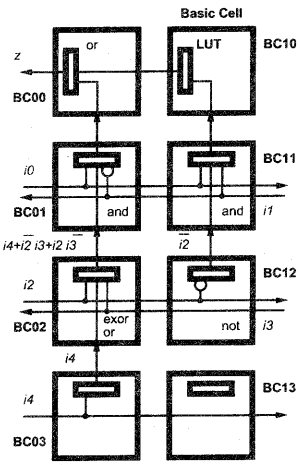

図4　シャノン展開法でのマッピング例

Fig. 4　Example of Shannon's expansion method

伝搬し、南北に論理関数を構成する。上下のOR平面では、東西に論理関数を出力する。論理関数を北側だけでなく南側にも構成することによって、LUTの使用効率を高める。

Maitra Term法は、処理の手順がボトムアップ的であり、論理関数のマッピングに要する基本セル列数の見通しが悪い。そこで、論理関数をトップダウン的にマッピングしていくことを考え、シャノン展開法を提案する。図3の論理関数zを例に、シャノン展開法の基本的な手順を示す。

- 論理関数zに対して、入力変数$i0$と$i1$(y=1の基本セルへの入力変数)にそれぞれ0と1を代入したときの論理関数(コファクター: $z_{00}, z_{01}, z_{10}, z_{11}$)を求める(シャノン展開)。
- コファクターが一般の論理関数(注1)である場合、それらはy=2の基本セルの北側のLUTの出力信号でなければならない。よって、y=1の基本セルの北側のLUTには、$i0$と$i1$に代入された値に応じて否定をつけた信号ならびに南からの入力信号との論理積を設定する。
- 各コファクターに、y=2以降の基本セルへの入力変数について再帰的にシャノン展開を適用して、基本セルの北側のLUTに値を設定する。
- y=0の基本セル行は、各基本セル列で構成される論理の論理和を演算して、論理関数zを構成する。

シャノン展開するための入力変数を含まない積項が論理関数に含まれる場合、複数のコファクターに同一の積項が含まれる。したがって、再帰的にシャノン展開を単純に続けると、マッピングに要する基本セル列数が増大する場合がある。また、コファクターが恒真関数であるとき、またはコファクター同士が否定の関係にあるときも、単純にシャノン展開を続けると、基本セル列数が増大する。そこで、以下のヒューリスティックを導入する。

- 複数のコファクターに同一の積項が存在する場合、
  - 同一の積項をコファクターから抽出して、それらを重畳してマッピングする
  - 通常通り、コファクターごとにマッピングする

場合の基本セル列数を比較し、列数の少ない側を採用する。

- コファクターが恒真関数の場合、一般の論理関数であるコファクターと重畳してマッピングする。
- 否定の関係にあるコファクターが存在する場合、重畳してマッピングする。

(例) 式(1)に示される論理関数zをシャノン展開法に基づきマッピングする例を示す。図4に示される基本セル列を考える。まず、論理関数zを$i0$と$i1$についてシャノン展開し、4つのコファクターを得る。

$$z = i0\cdot\overline{i1}\cdot i4 + i0\cdot\overline{i1}\cdot i2\cdot\overline{i3} + i0\cdot\overline{i2}\cdot i3 + i0\cdot i1\cdot\overline{i2}; \quad (1)$$

$$z_{00} = 0;$$

$$z_{01} = 0;$$

$$z_{10} = i4 + i2\cdot\overline{i3} + \overline{i2}\cdot i3;$$

$$z_{11} = \overline{i2}\cdot i3 + \overline{i2}$$

$$= \overline{i2};$$

(注1)：恒真関数(1)でも恒偽関数(0)でもないこと

z_{10} をマッピングするために、BC01 の北側の LUT には、$i0 \cdot \overline{i1}$ と BC02 の北側の LUT からの出力の論理積を割り当てる。$z_{10}(= z')$ を $i2$ と $i3$ についてシャノン展開することにより、

$$
\begin{aligned}
z'_{00} &= i4; \\
z'_{01} &= i4 + 1 \\
&= 1; \\
z'_{10} &= i4 + 1; \\
&= 1; \\
z'_{11} &= i4;
\end{aligned}
$$

を得る。$i4$ が全てのコファクターに含まれ、かつ z'_{01} と z'_{10} が恒真関数なので、BC02 の北側の LUT では、$i2$ と $i3$ の排他的論理和と BC03 の北側からの出力の論理和をとる。z_{11} も同様にシャノン展開を繰り返すことにより、図 4に示されるように、LUT に論理が割り当てられる。 □

2.3 変数の順序を考慮するシャノン展開法

図 4において、各基本セルの北側の LUT から出力すべき論理関数は、入力変数の順序に依存する。そのため、良い順序を発見できると、論理関数のマッピングに必要な基本セル列数は減少する。また、論理関数を構成させる方向を変えると、入力変数の順序が逆になるので、マッピングに必要な基本セル列数は変わる。

2.3.1 入力変数の順序付け

最適な入力変数の順序を得るために、その組合わせの数を探索する。以下では、入力変数の数が偶数 (2n 個) として説明する。入力変数の数が奇数のときは、半端となる入力変数を基本セル行の最下段に置く。

（1） 2n 個の入力変数を 2 個ずつ n 組に分ける。この組合わせの数は、次の通り。

$$
{}_{2n}C_2 \cdot {}_{2n-2}C_2 \cdot \cdots \cdot {}_2C_2 = \frac{(2n)!}{2^n}
$$

（2） 上記で求めた組を順次入れ替えることによって、入力変数の順序を得る。その順列の数は、$\frac{n!}{2}$ である。2 で割る意味は、対称性から逆の順序は考えないということである。

(例) 『$(i0, i1)$, $(i2, i3)$, $(i4, i5)$』 と 『$(i4, i5)$, $(i2, i3)$, $(i0, i1)$』 の順序は、どちらか一方だけ考慮する。 □

しがって、全体の組合わせの数は、次の通り。

$$
\frac{(2n)! \cdot n!}{2^{n+1}}
$$

2.3.2 論理関数を構成する方向の決定

2.3.1章に書かれた手順で入力変数の順序を順次生成し、論理関数の構成方向を南北に振りながら、論理関数群のマッピングに要する基本セル列数を評価する。本手法では、分枝限定法を用いて、最適な入力変数の順序ならびに論理関数の構成方向を得る。論理関数の数を m とすると、構成方向を南北に振り分ける場合の数は、2^m 通りである。

論理関数の Bdd 表現において、同じ基本セルに入力する二変数ごとにシャノン展開を適用して 4 つのコファクターを求めながら恒真関数へのパスを数え、それを分枝限定法でのマッピングに要する基本セル列数の評価値とする。パスを数えるときのヒューリスティックは、以下の通り。

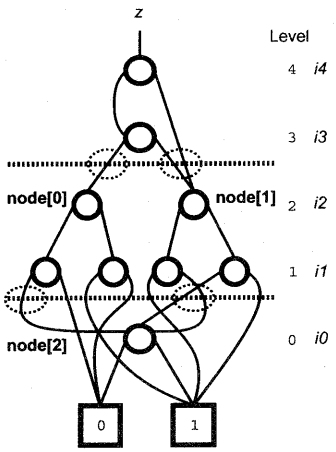

図 5 マッピングに要する基本セル列数の見積もり法
Fig. 5 CutSet Node of Bdd

- 4 つのコファクターに恒真関数と一般の論理関数が含まれる場合、恒真関数は一般の論理関数と重畳してマッピングできるので、ここでの恒真関数はパスの数には含めない。
- 否定の関係にあるコファクターが存在する場合、それらは重畳してマッピングできるので、一つコファクターについて処理を続ける。

(例) 図 5の Bdd で表現される論理関数 z をマッピングするのに必要な基本セル列数の評価値を得る手順を示す。

（1） 論理関数 z を入力変数 $i4, i3$ についてシャノン展開して、node[0] と node[1] のコファクターを得る。node[0] と node[1] は、否定の関係ではないので、両者とも引き続きシャノン展開を続ける。

（2） node[0] を入力変数 $i2, i1$ についてシャノン展開すると、node[2]、恒真関数、恒偽関数のコファクターを得る。恒真関数は、node[2] と重畳できるので、ここでの恒真関数へのパスは数えない。

- node[2] を入力変数 $i0$ についてシャノン展開すると、恒真関数と恒偽関数が得られる。よって、恒真関数までのパスを一つ数える。

（3） node[1] を入力変数 $i2, i1$ についてシャノン展開すると、node[2]、恒真関数が得られる。恒真関数は、node[2] と重畳できるので、ここでの恒真関数へのパスは数えない。

- node[2] を入力変数 $i0$ についてシャノン展開すると、恒真関数と恒偽関数が得られる。よって、恒真関数までのパスを一つ数える。

（4） 論理関数 z は、基本セル二列で表現できる。 □

2.3.3 論理関数の構成方向の決定

各論理関数の構成方向を南北に振りながら、分枝限定法を用

```
enum DirKind {South,North,Dir};

bool OptimizeDir(int index,int north,
int south,int** ref,DirKind* dir,···)
{
  if (現在の構成方向のもとでの基本セル列
    数が、従来の最善値よりも悪い)
    return(false);

  if (論理関数を全て調べ終わった){
    従来の最善値よりも良い結果なので、
      - 入力変数の順序
      - 論理関数群の構成方向
      - 論理関数群のマッピングに必要な基本セル列数
    を最善値として保持する。
    return(true);
  }
  else {
    bool mark[Dir];

    dir[index]=North;
        //index 番目の出力関数を北向きとする。
    mark[North]=OptimizeDir(index+1,
      north+ref[North][index],south,ref,dir,···);

    dir[index]=South;
        // index 番目の出力関数を南向きとする。
    mark[South]=OptimizeDir(index+1,
      north,south+ref[South][index],ref,dir,···);

    return((mark[North]==true ||
      mark[South]==true) ? true : false);
  }
}
```

図6　分枝限定法手順

Fig. 6　Procedure of branch and bound

いて最小の基本セル列数となる入力変数の順序と論理関数の構成方向を求める。図6にその手順を示す。図6内の変数の意味は、以下の通り。

index　注目している論理関数の番号。初期値0。

north　注目している入力変数の並びにおいて、北側に構成される論理関数群のマッピングに必要な基本セル列数。初期値0。

south　同じく、南側に構成される論理関数群のマッピングに必要な基本セル列数。初期値0。

ref[方向][論理関数の番号]　番号で指定される論理関数を指定される方向に構成させる際、マッピングに必要な基本セル列数の評価値。入力変数を順序付けした後、あらかじめ求めておく。

dir[論理関数の番号]　番号で指定される論理関数の構成方向。

3. 実験結果

本手法の有効性を示すために、PC(Pentium III, 966MHz, FreeBSD4.6)上にC++(g++ ver.2.95.3)を用いてプロトタイプシステムを作成し、MCNC FSM benchmark suitsの組合わせ回路部分をマッピングした。本システムは、slifまたはpla表現を用いて記述された組合わせ回路をPCA可変部の構成情報に変換する。

Maitra Term法と順序付け無しのシャノン展開法では、入力記述に記載された順序で、入力変数の順序と出力関数の南北への構成方向を決定する。実験結果を表1に示す。縦×横は、マッピング結果を矩形で被覆したときの基本セルの数である。LUT段数は、最大のLUT段数を示す。備考欄に、いくつかの回路について順序付けありのシャノン展開法の処理時間を示す。

Maitra Term法と順序付けなしのシャノン展開法を比較すると、面積で94%、LUT段数で95%、処理時間は15%となっている。4つのコファクターに効果的にヒューリスティックが適用できる場合に、シャノン展開法はMaitra Term法よりも小さな面積で論理関数をマッピングできる。2.2章の式(1)において、$\overline{i2}\cdot i3$は、本来z_{10}とz_{11}に含まれるので、z_{10}とz_{11}から

$$i0\cdot\overline{i1}\cdot\overline{i2}\cdot i3 + i0\cdot i1\cdot\overline{i2}\cdot i3 = i0\cdot\overline{i2}\cdot i3$$

が得られ、もとの式(1)に存在する形の積項となる。しかし、z_{11}は、論理最小化のため$\overline{i2}$となり、これが$i0\cdot i1\cdot\overline{i2}\cdot i3$を包含しているので、この式の変換は直接は見えない。式(1)では、z_{10}とz_{11}がそれぞれ一列の基本セルにマッピングされるので、二列の基本セルで表現可能である。一方、Maitra Term法では、$i4, i2, i3$の順番で入力変数が見えるので、

$$z = i0\cdot\overline{i1}(i4 + i2\cdot\overline{i3}) + i0\cdot\overline{i2}\cdot i3 + i0\cdot i1\cdot\overline{i2};$$

の代数的な因数分解が得られ、三列の基本セルで表現される。ブール式の特徴を活用することにより、

$$\begin{aligned} z &= i0\cdot\overline{i1}(i4 + i2\cdot\overline{i3}) + i0\cdot\overline{i2}\cdot i3 + i0\cdot i1\cdot\overline{i2} \\ &= i0\cdot\overline{i1}(i4 + i2\cdot\overline{i3}) + (i0\cdot i1\cdot\overline{i2}\cdot i3 + i0\cdot\overline{i1}\cdot\overline{i2}\cdot i3) + i0\cdot i1\cdot\overline{i2} \\ &= i0\cdot\overline{i1}(i4 + i2\cdot\overline{i3} + \overline{i2}\cdot i3) + i0\cdot i1\cdot\overline{i2}; \end{aligned}$$

の変換が得られれば、シャノン展開法と同じ結果となるが、このような変換は一般に難しい。Maitra Term法では、論理の包含関係をもとにLUTに割り当てる論理を決めていくので、処理時間を要する。シャノン展開法では、各基本セルへの入力変数に0または1を代入することによって、LUTに割り当てる論理が得られるので、処理時間は短い。

変数の順序づけありのシャノン展開法は、順序づけを考慮しない場合と比較して、面積で77%、LUT段数で77%になる。ただし、解の探索空間が、入力変数の数については、その階乗の乗算できいてくるので、入力変数の数8(30240通り)では、長大な時間を要する。出力である論理関数の数については、入力変数の数の同じex2とs27とで処理時間に大きな差(約60倍)が見られる。それぞれの最悪ケースでの探索数は、2^7=128通りと2^4=16通りなので、60倍の差はない。s27で

表1 実験結果

Table 1 Experimental results

回路名	入力変数	出力関数	状態レジスタ	Maitra T. 法 順序付けなし 縦×横	Maitra T. 法 順序付けなし LUT段数	シャノン展開法 順序付けなし 縦×横	シャノン展開法 順序付けなし LUT段数	シャノン展開法 順序付けあり 縦×横	シャノン展開法 順序付けあり LUT段数	備考
bbara	4	2	4	8× 13	21	8× 13	22	8× 10	20	6805.9 秒
bbtas	2	2	3	6× 6	10	6× 4	8	6× 3	7	
beecount	3	4	3	7× 13	27	7× 13	27	7× 8	14	
dk14	3	5	3	7× 24	45	7× 21	43	7× 20	27	
dk15	3	5	2	7× 16	28	7× 13	19	7× 11	16	
dk17	2	3	3	7× 15	23	7× 15	23	7× 10	15	
dk27	1	2	3	5× 8	11	5× 7	10	5× 5	7	
dk512	1	3	4	7× 16	28	7× 17	30	7× 10	18	
ex2	2	2	5	8× 28	52	8× 27	49	8× 24	44	3350.0 秒
ex3	2	2	4	7× 17	29	7× 17	28	6× 11	19	
ex5	2	2	4	7× 12	24	7× 11	23	7× 13	27	
ex7	2	2	4	7× 19	35	7× 19	34	7× 12	20	
lion	2	1	2	4× 3	5	4× 3	5	4× 3	5	
s27	4	1	3	6× 9	13	6× 8	12	7× 4	11	52.8 秒
shiftreg	3	3	3	4× 7	9	4× 6	9	4× 5	9	
tav	4	4	2	7× 6	10	7× 6	10	6× 4	8	
train11	2	1	4	6× 8	12	6× 8	12	6× 7	11	
train4	2	1	2	4× 3	5	4× 2	5	4× 2	5	
横の合計				223	387	210	369	162	283	
処理時間				46.0 秒		7.0 秒		10258.2 秒		(10208.7 秒)

は、分枝限定法での枝刈りが効率的に働いていると考えられる。

4. ま と め

本稿では、組合わせ回路を PCA の可変部にマッピングする手法について述べた。シャノン展開法を導入することにより、従来の Maitra Term 法よりも小さい面積に論理回路をマッピングできる。さらに、入力変数の順序付けならびに出力関数の構成方向を考慮することにより、順序付けなしのときよりもさらに小さな面積に論理回路をマッピングできる。ただし、変数の順序付けは、探索空間が広いので、現在の分枝限定法では長大な時間を要する。今後は、よい枝刈り法を検討して、処理時間の削減を目指す。

文 献

[1] K.Nagami, K.Oguri, T.Shiozawa, H.Ito, and R.Konishi. Plastic Cell Architecture: A Scalable Device Architecture for General-Purpose Reconfigurable Computing. *IEICE Transactions on Electronics*, Vol. E81-C, No. 9, pp. 1431–1437, September, 1998.

[2] 伊藤 秀之 小西 隆介 中田 広 小栗 清 永見 康一 塩澤 恒道 Norbert Imlig 稲森稔 名古屋彰. 動的再構成可能論理 LSI - PCA-1. 信学技報 ICD2000-47, pp. 9–16, 2000.

[3] M.Inamori, H.Nakada, R.Konishi, A.Nagoya, and K.Oguri. A Method of Mapping Finite State Machine into PCA Plastic Parts. *IEICE Transaction on Fundamentals*, Vol. E85-A, No. 4, pp. 804–810, 2002.

[4] M.Inamori, R.Konishi, and A.Nagoya. A New Approach to Mapping Finite State Machine into PCA Plastic Parts. *Proc. of SASIMI 2001*, pp. 178–185, 2001.

[5] K.K.Maitra. Cascaded Switching Networks of Two-Input Flexible Cells. *IRE Trans. Electron. Comput.*, Vol. EC-11, pp. 136–143, 1962.