設計自動化 62－26
(1992. 5.28)

ニューラルネットワークを用いた論理式の簡単化

泉 健憲* 若林 哲** 檀 良*

*法政大学工学部 **東芝マイクロエレクトロニクス株式会社

概要

本稿では論理式の簡単化に関する新しい手法を提案する．通常，コンピュータ上での論理式の簡単化は機械向けのアルゴリズムを用いている．しかし，我々はこのようなアルゴリズムを用いることなく，より人間的なアルゴリズムを用いて論理式の簡単化を行う研究を行ってきた．今回提案する手法はカルノー図とニューラルネットワークを組み合わせることによって，より人間的なアルゴリズムでコンピュータ上で論理式の簡単化を行うものである．はじめに，カルノー図より主項となり得るパターンをデータ化する．つぎに，その中から与えられた問題に関するパターンのみを考慮し，各項についてその項が含まれているパターンを抽出する．そして，それをニューラルネットワークを構成するユニットに対応づけることによってカルノー図とニューラルネットワークの組み合わせを行った．本手法によって簡単化を行える変数の個数はデータベースに依存しており，現在，４，５変数に対応している．本手法を用いて簡単化を行った結果，最適解あるいはそれに近い解が得られた．

A method for simplifying logical expression using Karnaugh map and Neural Network

Takenori IZUMI* Satoshi WAKABAYASHI** Ryo DANG*

*College of Engineering, Hosei University **Toshiba Microelectronics Corp.

Abstract

This paper proposes a new method for simplifying logical expressions using the Karnaugh map and a neural network. Conventional methods for simplifying logical expressions using a computer are generally based on a machine-oriented heuristic algorithm. However, our method is using a more human-oriented algorithm.

First, the patterns corresponded to prime implicants are databased. Then the patterns related to each term are extracted from the patterns concerned with the problem, and matched to the units of a Hopfield network. The number of variables depends on the size of the database, and at present the method is applicable to 4 to 5 variables. The method provides the optimum solution or a solution close to optimum.

1．はじめに

ＩＣからＬＳＩ，ＶＬＳＩへと近年の半導体技術の進歩は非常にめざましく，素子の微細化，加工技術の進歩によって１チップ当たりの素子数は非常に増加している．それに伴い開発コストと開発期間が増加し，１チップ当たりに占める設計コストの比率が高まっている．このような状況で，設計コストの比率を減少させるために様々な自動化が行われいている．今回，我々は論理設計の段階で行われる論理式の簡単化に着目した（図１）．

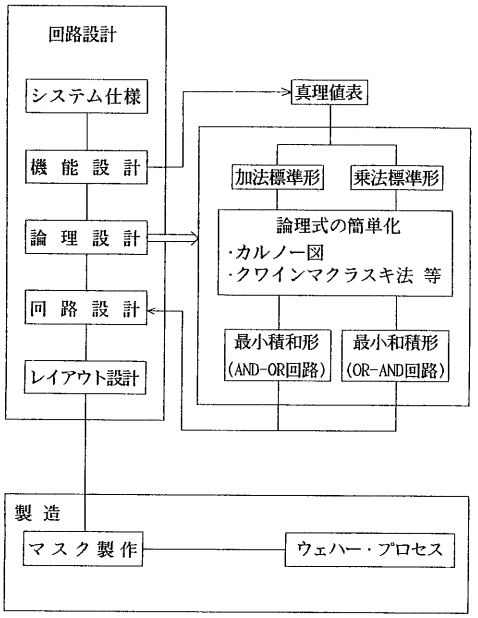


図１　ＬＳＩ設計行程

論理式の簡単化は論理合成において最も重要な技術であり，論理合成に費やされる時間のうち，論理式の簡単化が占める時間の割合は非常に大きい．さらに，論理式の簡単化はＰＬＡの簡単化に直接結びついており，論理式の簡単化を行うことによって，積項線数，接続等が減少し，それにともないチップ面積，消費電力，遅延時間などを減少することができる．

論理式の簡単化を行う代表的な手法として，ブール代数の公式，カルノー図，クワイン・マクラスキ法等を用いる方法があげられる．

カルノー図は与えられた論理式からただちにカルノー図をつくることができる．また，図上の'１'の配列パターンより容易に主項が求められ，必須項も簡単に求めることができる．しかしながら，入力変数の個数に限界があり，６個以上の変数では図が複雑になりすぎて実用にならないという欠点を持っている．また，ブール代数の公式を用いる場合も入力変数の増加にともない，それ以上の簡単化ができるか否かの限界がわかりにくくなる．さらに，公式をよく理解し，その活用に習熟しているか否かによるところが大きい．また，最初の変形のやり方によって結果が異なり，途中の見通しが良くないという欠点を持っている．このように，ブール代数の公式やカルノー図を用いる方法は人間の直感力等に依存する点が大きく，人手向きの手法であるといえる．

一方，クワイン・マクラスキ法は手順が確定しているが，作業量が多くなる．したがって，コンピュータ処理向きであるといえる．しかし，論理式の簡単化がＮＰ（Nondeterministic Polynominal-time）完全問題であるため，入力変数の増加に対して加速度的に主項数が増加し，ドント･ケアや多出力に対する対応まで考慮すると，コンピュータの記憶容量と計算時間が指数関数的に大きくなってしまい使用不能となる．そこで，ヒューリスティック（heuristic）なアルゴリズムがいくつか開発されている[1]．

ヒューリスティックなアルゴリズムでは必ずしも最小解が求まるわけではない．しかし，実際の応用では必ずしも最小解の必要はなく，問題の規模が大きいときには通常このようなアルゴリズムが用いられている．

我々はこのようなヒューリスティックなアルゴリズムを用いずに，他の手法によってコンピュータ上で論理式の簡単化を行う研究を行ってきた．今回我々が提案する手法は，より人間的なアルゴリズムによって論理式の簡単化を行うものである．

人間による簡単化のアルゴリズムは，コンピュータを用いて簡単化を行うアルゴリズムとは根本的に異なっている．コンピュータが直列集中的に処理を行うのに対して，人間の場合は並列分散的に処理を行っているという違いがある．そこで，このような並列分散型の処理を行うための手法としてニューラルネットワークがあげられる．ニューラルネットワークは人間の神経細胞のモデルを用いて人間に近い高度な並列分散型の情報処理を実現しようとするものである．

本手法ではニューラルネットワークに着目し，ニューラルネットワークのこの分野への導入の可能性を考えるという意味も含めて，ニューラルネットワークとカルノー図を組み合わせることによって，より人間的なアルゴリズムで論理式の簡単化を行う手法を提案する．以下に本手法について記す．

2．論理式の簡単化とニューラルネットワーク

2．1アルゴリズム

論理の簡単化はドント･ケアや多出力に対する対応まで考慮すると作業量が非常に大きくなる．特に共有項を最初から考慮にいれた多出力論理の最小解を得るアルゴリズムは相当複雑で，手間がかかる．しかし，その手間ほど実際の回路規模が小さくなるとは限らない．したがって，現在，本手法では多入力１出力，ドント･ケア無しの組み合わせ回路を対象としている．

本手法の流れをカルノー図を用いた人手による簡単化と比較したものを図２に示す．

はじめに真理値表より論理式を求める．次に求めた論理式をカルノー図へ変換する．次の段階として，得られたカルノー図より解となる組み合わせを選択する．人手による簡単化の場合，ただちにその組み合わせを選択す

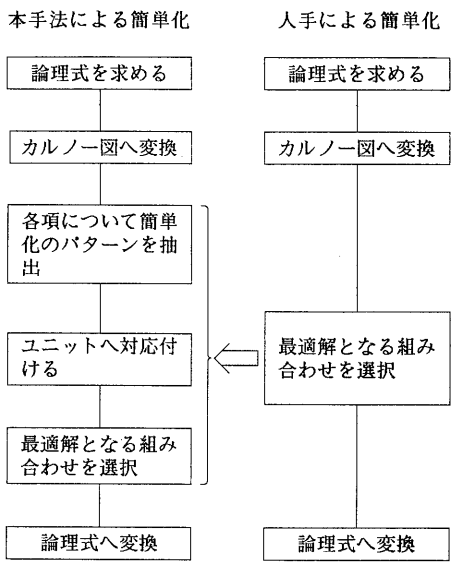


図２　本手法の流れと人手による簡単化

ることができる．しかし，コンピュータ処理の場合，何らかの手順を踏む必要がある．本手法ではこの部分にニューラルネットワークを導入した．今回用いたニューラルネットワークは図３に示すような相互結合型の構造を持つホップフィールド型ネットワーク[2]である．図では表現できなかったが，実際にはネットワークを構成する一つのユニットは他のすべてのユニットと結合している．近年，このモデルを用いた最適化問題近似解法[3]が発表され注目されている．本手法では，この最適化問題近似解法を応用した．

最適解となる組み合わせを選択するために，まず，得られたカルノー図より，論理式の各項について簡単化可能なパターンの抽出を行う．つまり，カルノー図の作業では，簡単化可能なパターンの抽出部分は主項を求め，抽出したパターンから解となる組み合わせを求める部分では主項の選択を行っていることになる．

次に，抽出したパターンの一つ一つをホップフィールドネットワークを構成するユニットに対応づけ，ネットワークを動作させることによって解となる組み合わせを選択する．このようにして選択されたパターンにしたがって簡単化を行い，簡単化された論理式を得る．

以上が本手法の簡単な流れである．次節で詳細について記す．

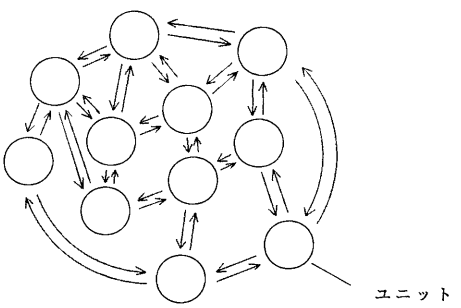


図３　相互結合型ネットワーク

２．２カルノー図とホップフィールドネットワークの結合

図４に示すような，ａ，ｂ，ｃ，ｄを入力とし，ｆを出力とする多入力１出力の組み合わせ回路の真理値表をサンプルとして用いる[4]．この真理値表より得られた論理式は式(1)となる．

a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

図４　サンプル

$$f=\bar{a}\bar{b}\bar{c}d+\bar{a}b\bar{c}d+\bar{a}bc\bar{d}+a\bar{b}\bar{c}d+a\bar{b}cd+ab\bar{c}\bar{d}+abc\bar{d} \quad (1)$$

$$f=0001+0101+0110+1001+1011+1100+1101 \quad (2)$$

はじめに，真理値表より得られた論理式を入力形式に変換する．変換した論理式は式(2)となる．

ここでは変数の位置が固定してあり，特定の桁が特定の変数を表している．そして，ある変数が真であるならば'１'，偽であるならば'０'とした．例えば，第１項目の'０００１'はａ，ｂ，ｃは偽，ｄは真であることを示している．このようにして得られた入力データをカルノー図へ変換すると図５のようになる．次にこのカルノー図より論理式の各項について簡単化可能なパターンを抽出する．簡単化可能なパターンとは，得られたカルノー図上で主項になり得るパターンのことを意味する．このパターンの抽出は，カルノー図上で主項となるパターン

cd \ ab	00	01	11	10
00			1	
01	1	1	1	1
11				1
10		1		

図５　カルノー図

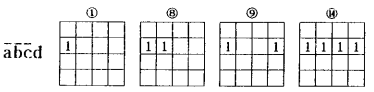


図６　第１項のパターン

をデータ化し，入力データとのパターンマッチングによって行っている．４変数の場合，全部で８０通りのパターンが存在する．その中から与えられた問題に関するパターンのみを考慮し，各項についてその項が含まれているパターンを抽出する．例えば，式（１）の第１項に着目すると，図６に示すように，その項が含まれている４つのパターンが抽出される．同様にして他のすべての項について簡単化のパターンを抽出したものが図７である．

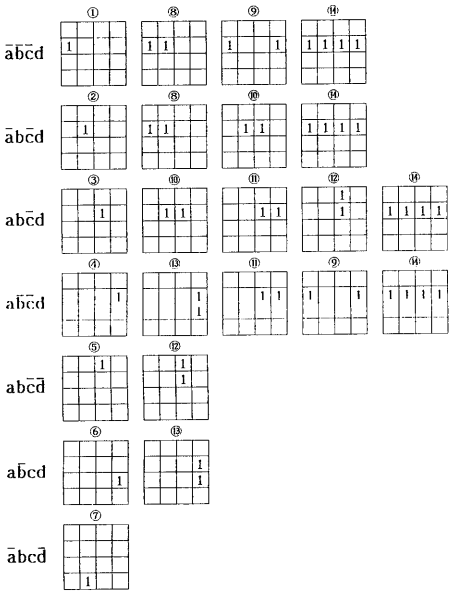


図７　全項のパターン

このようにして抽出したパターンをホップフィールドネットワークを構成するユニットと対応させるために，抽出した各パターンに番号を付け，それをユニットの番号と対応させる．ただし，同じパターンが複数の項に含まれている場合は同じ番号を付ける．

このような対応付けより，図８に示す３つの集合が得られる．集合Ａ(n)，Ｘｉ，Ｎｉはそれぞれ，論理式の各項のすべての簡単化のパターン，ネットワークが持つすべてのユニット，各ユニットに対応するパターンにしたがって簡単化された場合の１つの項における変数の個数を表している．例えば，図７に示した対応付けにおいてパターン１４が選択された場合，その項の変数の数は４から簡単化後には２となるので，集合Ｎｉの１４番目の数が２となっている．

２．３目的関数の導出

以上のようにして，カルノー図とホップフィールドネットワークを組み合わせた．次に，この組み合わせを基に最小解を得るための制約条件をつくる．

$A(\bar{a}\bar{b}\bar{c}d) = \{1, 8, 9, 14\}$

$A(\bar{a}b\bar{c}d) = \{2, 8, 10, 14\}$

$A(ab\bar{c}d) = \{3, 10, 11, 12, 14\}$

$A(a\bar{b}\bar{c}d) = \{4, 9, 11, 13, 14\}$

$A(ab\bar{c}\bar{d}) = \{5, 12\}$

$A(a\bar{b}cd) = \{6, 13\}$

$A(\bar{a}bc\bar{d}) = \{7\}$

$X_i = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
$\downarrow\ \downarrow\ \downarrow\ \cdots$
$N_i = \{4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 2\}$

図８　集合Ａ(n)，Ｘｉ，Ｎｉ

論理式の簡単化を行った場合，簡単化前と簡単化後の論理式は同じ機能を表現していなければならない．つまり，簡単化後の論理式には簡単化前の論理式が含まれていることになる．したがって，簡単化前のある１つの項に着目した場合，その項に含まれている簡単化可能なパターンの中から少なくとも１つは選択しなければならないことを意味する．さらに，簡単化可能なパターンを選択する場合，どのパターンを選択しても良いのではなく，なるべく簡略化された論理式になるようなパターンを選択する必要がある．また，今回の対応付けの場合，ユニットは'選択'あるいは'非選択'の２つの状態のみである．したがって，ユニットの出力を'１'あるいは'０'の２値とし，'１'ならば選択，'０'ならば非選択とする．

以上の制約条件をまとめると次のようになる．

(a) １つの項について，簡単化可能なパターンを少なくとも１つは選択する．
(b) １つの項における変数の個数をなるべく少なくする．
(c) ユニットの出力を'１'あるいは'０'とする．

次に，この制約条件をネットワークに対応させるように定式化し目的関数を作る．

ホップフィールドネットワークを最適化問題に応用する場合，その問題の特徴をユニットの出力の２次式で表現し，それを最小にすることによって解を導く．これが目的関数となる．つまり，上で述べた３つの条件を満たすときに最小となるような関数を作る．

制約条件(a)　は，ある１つの項に着目した場合，その項に属するユニットの集合Ａ(n)におけるユニットの出力の総和が１以上となればよい．言い替えれば，一つも選択されていない，つまり，その項におけるユニットの出力の総和が０でないときに式の値が最小となればよい．したがって，はじめに集合Ａ(n)におけるユニットの出力の総和をとる．つぎに１つも選択されなかった場合に式の値が最小となるのを防ぐために－１を加える．そして，目的関数がユニットの出力の二次形式で表現されるという条件を満たすために全体を２乗する．これをすべての項について総和をとったものが式(3)となる．

$$\phi_1=\frac{1}{2}C_1\sum_{n=0}^{Term}\left(\sum_{Xi\in An}X_i-1\right)^2 \quad (3)$$

制約条件(b)は各項における簡単化後の変数の数を少なくするように選択を行うのだから，ユニットが選択された場合のみにN_iが有効となるようにする．したがって，ユニットの出力X_iとN_iの積とする．また，簡単化前の変数の個数より増えることはないので，用いている変数の個数，Vを引く．このままでは，ユニットの出力がすべて0であると最小となってしまうので，これを避けるために，また，目的関数となる条件を満たすために全体にX_iを掛ける．これをすべての項について行うと式(4)を得る．

$$\phi_2=\frac{1}{2}C_2\sum_{n=0}^{Term}\left(\sum_{Xi\in An}X_i(X_iN_i-V)\right) \quad (4)$$

制約条件(c)をみたすために，まず，ユニットの出力が1のときに式の値が最小となるように1からユニットの出力X^iを引く．同時に，ユニットの出力が0のときにも式の値が最小となるように，全体にユニットの出力X^iを掛ける．これをすべてのユニットに出力に対して行うと式(5)になる．

$$\phi_3=\frac{1}{2}C_3\sum_{i=0}^{Unit}X_i(1-X_i) \quad (5)$$

ここで，1/2は各ユニットが相互結合であることを表している．また，C_1，C_2，C_3は係数．

ここでいう係数とは，ネットワークをうまく収束させるためのパラメータである．これは解を得るために非常に重要な役割を果たしており，このパラメータによっては最小解が得られない場合もあり得る（最悪の場合は解が得られない場合もある）．

ホップフィールドネットワークを用いる場合のパラメータ決定方法は明確な定義がなく，経験的な値を用いることが多い．したがって，本手法では実験を繰り返すことによって得られたパラメータを用いている．しかし，異なる論理式の簡単化を行う毎に別のパラメータが必要となるのではなく，現状では変数の個数ごとに最適なパラメータを見つけることによって，2回目以降は同じパラメータを用いて簡単化を行うことができる．

以上のようにして得られた各々の目的関数を最小にすることによって制約条件を満たすことができる．つまり，問題全体としてみれば，それぞれの目的関数の和が最小となればよい．したがって，最終的な目的関数は式(6)となる．

$$\phi=\phi_1+\phi_2+\phi_3 \quad (6)$$

2．4 ネットワークの動作

目的関数が最小となるようにネットワークを動作させるために，目的関数以外に以下の3つの関数を用いる．

$$E=-\frac{1}{2}\sum_{i,j}W_{ij}X_iX_j+\sum_i h_iX_i \quad (7)$$

$$u_i=\sum_j W_{ij}X_j+h_i \quad (8)$$

$$X_i=\frac{1}{2}\left(1+\tanh\left(\frac{u_i}{0.5}\right)\right) \quad (9)$$

式(7)はホップフィールドネットワークが持つエネルギーを表すエネルギー関数，式(8)は各ユニットの内部状態を決定する内部状態関数，式(9)X_iはユニットの出力を決定する出力関数である．出力関数にはシグモイド関数を用いた．ここで，W_{ij}はユニット間の結合の強さを表す結合加重，X_i，X_jはユニットの出力，h_iはユニットが持つしきい値である．

はじめに，結合荷重およびしきい値を求める．これらは，目的関数とネットワークが持つエネルギー関数から係数比較法によって求められる．

求めた結合荷重およびしきい値を式(8)に代入し，さらに，そこで得られた内部状態を式(9)に代入してユニットの出力を決定する．

このような関数を用いてユニットの状態を変化させ，ネットワークを動作させると，ネットワークのエネルギーは減少する．そして，ある一定のところまで行くとエネルギーの変化がなくなる．本手法ではこの点を収束点とした．つまり，このときのユニットの状態が解を表している．

3．実験結果

以上のアルゴリズムにしたがってコンピュータ上でシミュレーションを行った結果を図9に示す．

```
(d)
Energy=-10.75 loop=1
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 0  0  0  0  0  0  1  0  1  0  0  1  1  0
Energy=-15.50 loop=2
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 0  1  0  0  0  0  1  0  0  0  0  1  1  1
Energy=-23.25 loop=3
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 0  0  0  0  0  0  1  0  0  0  0  1  1  1
Energy=-23.25 loop=4
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 0  0  0  0  0  0  1  0  0  0  0  1  1  1

(e)
Eureka! A solution is obtained.
C1= 0.50 C2= 4.00 C3= 5.00
Energy= -23.25000  loop=4
 1  0  0  0  1  1  1  1  0  1  1  0  0  0
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 0  0  0  0  0  0  1  0  0  0  0  1  1  1

(f)
f=0001+0101+0110+1001+1011+1100+1101
f=0110+110-+10-1+--01
```

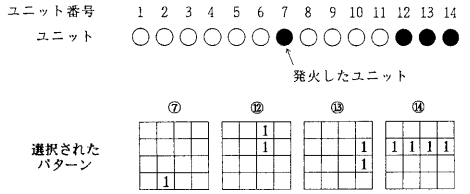
図9　出力結果

(d)の部分はユニットの状態の遷移を示している．ここで，Energyはネットワークのエネルギー，loopはネットワークの状態遷移回数，1から14までの数字はユニット番号を表し，その下の数字がそのユニットの出力を表している．これより，ネットワークのエネルギーが徐々に減少し，4ループ目に収束していることがわかる．(e)の部分は最終的なユニットの状態を示している．今回用

いたパラメータは$C_1=0.50$，$C_2=4.00$，$C_3=5.00$であり，エネルギーが-23.25の時に４ループで収束している．エネルギーの値は特に意味はない．つぎの３行の数字は上から，ユニットの初期状態，ユニット番号，最終的なユニットの出力結果を表している．この場合，７，１２，１３，１４のユニットが選択されたことを示している．(f)の部分は簡単化前の論理式と簡単化後の論理式を示している．ここで，'０'は否定，'１'は肯定，'－'はその部分に対応する変数は用いられていないことを意味する．例えば，出力結果の第４項目はａ，ｂの変数は用いられず，ｃが偽，ｄが真であることを示している．

図１０は図９の出力結果を視覚的にわかりやすく書き直したものである．

簡単化前　$f=\bar{a}\bar{b}\bar{c}d+\bar{a}b\bar{c}d+\bar{a}bcd+a\bar{b}\bar{c}d+a\bar{b}cd+abc\bar{d}+ab\bar{c}d$　(12)



簡単化後　$f=\bar{a}bcd+ab\bar{c}+a\bar{b}d+\bar{c}d$　(13)

図１０　出力結果の内容

サンプルの場合，７，１２，１３，１４のユニットの出力が'１'となっているので，これに対応した７，１２，１３，１４のパターンが選択されたことを意味する．そして，このパターンにしたがって論理式の簡単化が行われ，式(12)の論理式は式(13)のように簡単化される．また，図１１に示したように人手によって簡単化を行った場合も同様な結果を得ることができ，本手法による結果が正しいことは明かである．また，本手法によって簡単化を行える変数の個数はデータベースに依存しており，現在，５変数まで対応している．今後，データベースを拡大することによってさらに多くの変数に対応可能である．

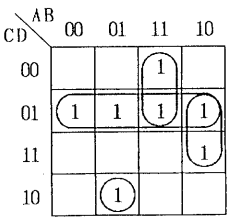


図１１　人手による簡単化の結果

４．まとめ

本稿では論理式の簡単化を行う一手法として，ニューラルネットワークとカルノー図を組み合わせる手法を提案した．これによって，ヒューリスティックなアルゴリズムを用いず，より人間的な手法で論理式の簡単化を行うことができた．また，本手法で用いたニューラルネットワークは本来並列計算を基本としているため，本手法を並列計算させることは容易であり，問題の規模が増加した場合，その効果は大きいと予想できる．さらに，本手法を拡張することによってＰＬＡ等の簡単化に対応できると考えられる．

以上のことを考慮して考察すると，ニューラルネットワークのこの分野への導入は多くの可能性を秘めており，さらに様々な応用が考えられ，非常に興味深いものであるといえる．

今後の課題として，いくつかの問題点を上げる．

はじめに，カルノー図のデータ化に関する問題があげられる．現在，各項の簡単化のパターンを抽出するために，カルノー図上での簡単化可能なパターンをすべてデータとして与えている．したがって，変数の増加に伴いその数が膨大になる．そこで全パターンをデータ化するのではなくカルノー図の構成をデータとして与えることによってこの点の解消を行う予定である．

次に解の精度の問題が上げられる．さらにネットワークの収束率を高め，最小解あるいは最小解に近い解が得られるように，ユニットの初期状態の決定方法，制約条件の改良，新しい制約条件の導入，パラメータ決定方法等を考慮する必要がある．

謝辞

本手法の研究･開発に当たり，御協力頂いた本研究室の王　益新氏，市川貴士氏に感謝致します．

参考文献

[1]今井正治，杉山尚志，" 超LSI設計ｼﾘｺﾝｺﾝﾊﾟｲﾚｰｼｮﾝ" ，ｻｲｴﾝｽﾌｫｰﾗﾑ，pp.63-95，1988

[2]J.J.Hopfield,"Neural Networks and Physical Systems with Emergent Collective Computational Abilities",Proc.Natl.Acad.Sci.USA.79,pp.2554-2558,1982

[3]J.J.Hopfield,D.W.Tank,"Neural Computation of Decisions in Optimization Problems",Biol.Cybern.52,pp.141-152,1985

[4]三上廉司，" ASIC時代の論理設計" ，電波新聞社，pp.119，1988