

禁止用ループの使用による一線入力 NAND ゲート回路の生成の一手法†

後 藤 公 雄††

本論文では、一線入力 NAND ゲート回路の生成用として、従来のマップファクタリング法を改善した計算機向きの手法について提案している。本論文では、まず理論的にこのアルゴリズムの基となる原理を解明し、その原理にしたがって曖昧さを排除した回路生成の手順を繰り返し、カルノー図上の全セルを埋め尽くすまでこれを続ける。この場合許容ループと禁止用ループをカルノー図上に繰り返し設定するが、従来のマップファクタリング法のような試行錯誤的かつ発見的な手法ではなく、許容ループの選択範囲を一定の規則にしたがって明確に設定し、その中から最大かつすべての許容ループを求め、さらにこれらの許容ループ群をそれまでに求めたすべての禁止用ループの組合せで禁止して所望の禁止用ループを抽出するようにしている。このアルゴリズムを用いた LISP 言語プログラムによって 3 変数論理関数 P 同値類や 4 変数論理関数について回路生成の演算を行い、ゲート数については 90% 近くまで、段数については 70% 近くまで最小回路に一致する回路が得られた。また、結果が最小回路と一致しない関数にたいしては得られた結果を調べ、冗長結線を除去したり、ゲート出力の入力位置を移動したりする簡単な手法により最小化回路が得られた。

1. ま え が き

一線入力 NAND (または NOR) ゲート回路の設計法として種々の方法が考えられてきた。特に手計算向きの方法としてカルノー図を用いるマップファクタリング法^{1),2)}が知られているが、この手法は試行錯誤的かつ発見的な手法であって、人間の勤や習熟度に依存するところが多く、しかも回路の最小性を保証しない。最小回路が保証される計算機向きの手法に整数計画論理設計法³⁾があるが、この手法は複雑である。また最近、計算機向きの手法として、負関数ゲートを用いた多段フィードフォワード法⁴⁾が提案された。しかし、これは一線入力 NOR ゲート回路を対象としている。IC を用いて論理回路を実現する場合には NOR ゲートと同等以上に NAND ゲートが用いられる。

ここでは、一線入力 NAND ゲート回路のみの設計法としてマップファクタリング法を改善した計算機向きの手法を提案する⁵⁾。本論文の手法は従来のマップファクタリングの手法の持つ試行錯誤的かつ発見的な性質を排除し、明確に定められた規則に従って一定の手順を繰り返し、カルノー図上のセル領域を探索し尽くす方法である。この手法では探索を繰り返す度ごとに探索範囲を明確に限定し、しかも前回の探索で得られた結果を次回の探索に有効に利用するようにして曖昧

さを除いている。また、従来のマップファクタリング法では、解を何回も求め直し、それらの解の中から最良のものを見出すのにたいし、本論文の手法は、計算の各段階で全部の解を求めるようになっていく。

本論文では、アルゴリズムの基となる原理の理論的検討を行い、それらの原理に従ったアルゴリズムを確立している。さらにそのアルゴリズムにより LISP 言語によるプログラムを作成し、3 変数および 4 変数の論理関数について演算を実行し、得られた結果のゲート数、結線数、段数について最小回路⁶⁾と比較し、その一致の度合いを検討する。さらに一致しない結果を最小回路へ縮約できる手法⁷⁾についても検討している。

2. 禁止用ループ法に関する原理

本章では、マップファクタリング法^{1),2)}を改善して一線入力 NAND ゲート回路を実現する禁止用ループ法⁵⁾の基となる定義、規則および定理について述べる。

〔定義 1〕 最小項を構成するリテラルがすべて肯定変数のみのとき、この最小項を最大最小項と呼ぶ。

〔定理 1〕 n 変数の最大最小項の添字、すなわちカルノー図上のセル番号は $2^n - 1$ となる。

(証明) 自明。 (QED)

〔例 1〕 図 1 に示す 3 変数 a, b, c のカルノー図で最大最小項のセル番号は $7 (= 2^3 - 1)$ である。

〔定義 2〕 n 変数のカルノー図上で最大最小項に相当するセルを含む隣接する 2^j (ここで $0 \leq j \leq n$) 個のセルより成るループを許容ループと呼ぶ。

〔例 2〕 3 変数 a, b および c にたいするカルノー

† A Method for Generation of Single-Rail Input NAND Gate Circuits by Using Inhibiting Loops by KIMIO GOTO (Department of Computer Science, Faculty of Engineering, Kanagawa Institute of Technology).

†† 神奈川県工科大学工学部情報工学科

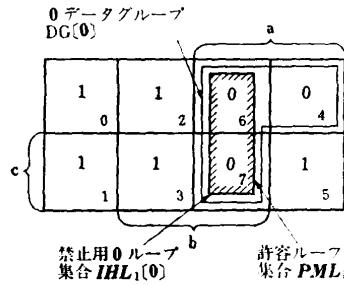


図 1 3変数カルノー図の一例 ($f = \Sigma(0, 1, 2, 3, 5)$ の第1回目の許容ループと禁止用ループの生成)
Fig. 1 An example of 3 variable Karnaugh Map.

図(図1)上の許容ループ全部の集合 PML は、その各要素(許容ループ)のセル番号の数列で表すと、

$$PML = \{7, 37, 57, 67, 1357, 2367, 4567, 01234567\}. \quad (1)$$

【定義3】許容ループの集合 PML の各要素(許容ループ)の持つ α セル全部を、これらの α セルを含む他のいくつかのループで除去して得られる $\bar{\alpha}$ セルのみの空でない集合を禁止用 $\bar{\alpha}$ ループと呼ぶ。ここで、 α は 0、または 1 である。なお、禁止用 0 ループと禁止用 1 ループを総称して禁止用ループと呼ぶ。

【例3】図1のカルノー図上で許容ループの集合を $\{1357, 2367\}$ のみに限定すると、0セルのみより成るループ $\{67\}$ によって、これらの許容ループのセルを除去して禁止用 1 ループの集合 $\{23, 135\}$ が得られる。

つぎに、本論文で扱う禁止用ループを用いた一線入力 NAND ゲート生成のための基本規則を示す。

【規則1】いま、2値の数字 0、1 のどちらかを α で表す。これまでに得られた禁止用 α ループの集合のセル集合 $IHL_{i-1}[\alpha]^c$ と全 $\bar{\alpha}$ セルの集合(以後これを $\bar{\alpha}$ データループと呼ぶ) $DG[\bar{\alpha}]$ とより成るカルノー図上の領域 RG_i の内部で最大数のセルから成る許容ループの集合 PML_i を作成し、これをそれまでに得られた禁止用ループの組合せで禁止することによって新たな禁止用 $\bar{\alpha}$ ループの集合 $IHL_i[\bar{\alpha}]$ を作る。これのセル集合 $IHL_i[\bar{\alpha}]^c$ と全 α セルの集合、すなわち α データグループ $DG[\alpha]$ とより成る新領域 RG_{i+1} を作り、同様にこの内部で最大数のセルから成る次の許容ループの集合 PML_{i+1} を作成し、これをそれまでに得られた禁止用ループの組合せで禁止して新たな

禁止用 α ループの集合 $IHL_{i+1}[\alpha]$ を作る。この手順を繰り返して最後に得られた禁止用 1 ループのみを用いて論理関数の一線入力 NAND ゲート回路を作る。ただし、最初に設定される許容ループの集合 PML_1 は禁止用 α ループの集合 $IHL_1[\alpha]$ そのものとする。つぎに規則 1 によって得られる諸定理を述べる。

【定理2】規則 1 によって何回目かに作られる許容ループの集合はそれよりも前に作られた許容ループの集合を完全にカバーする**。

(証明) 規則 1 により、一般に i 番目の許容ループの集合 PML_i の選択範囲 RG_i は、 $i-1$ 番目の許容ループの集合 PML_{i-1} に含まれるすべての禁止用 $\bar{\alpha}$ ループの集合のセル集合 $IHL_{i-1}[\bar{\alpha}]^c$ と α データグループ $DG[\alpha]$ との和集合である。すなわち、

$$RG_i = IHL_{i-1}[\bar{\alpha}]^c \cup DG[\alpha]. \quad (2)$$

ここで、規則 1 により禁止用 $\bar{\alpha}$ ループの集合 $IHL_{i-1}[\bar{\alpha}]$ は PML_{i-1} に含まれるすべての α セルを、それまでに作られた禁止用ループによって除去して得られる。このように除去される α セルの集合を $DEL_{i-1}[\alpha]$ とすると、

$$IHL_{i-1}[\bar{\alpha}]^c = PML_{i-1}^c - DEL_{i-1}[\alpha]. \quad (3)$$

式(2)、(3)および $DEL_{i-1}[\alpha] \subseteq PML_{i-1}^c$ かつ $DEL_{i-1}[\alpha] \subseteq DG[\alpha]$ の成立より、

$$RG_i = (PML_{i-1}^c - DEL_{i-1}[\alpha]) \cup DG[\alpha] = PML_{i-1}^c \cup (DG[\alpha] - DEL_{i-1}[\alpha]). \quad (4)$$

ここで、式(4)の右辺の PML_{i-1}^c と括弧 () 内は互いに独立であり、さらに規則 1 より、 PML_i は RG_i に含まれる最大の許容ループの集合であるから、 PML_i は PML_{i-1}^c 、および $DG[\alpha] - DEL_{i-1}[\alpha]$ から PML_i の生成に干渉しない最小の $\Delta_{i-1}DG[\alpha]$ を除いたセル集合より作られる。すなわち、

$$PML_i^c = PML_{i-1} \cup (DG[\alpha] - DEL_{i-1}[\alpha] - \Delta_{i-1}DG[\alpha]). \quad (5)$$

式(5)で右辺の PML_{i-1}^c と括弧 () 内とは互いに独立であるから、この括弧 () 内が空集合でない限り、

$$PML_i^c \supset PML_{i-1}^c. \quad (6)$$

$$\therefore PML_i \supset PML_{i-1}. \quad (7)$$

(QED)

【例4】図1より許容ループの集合 PML_1 は $\{67\}$ である。図2は図1と同じ関数 $f = \Sigma(0, 1, 2, 3, 5)$ にたいする第2回目と第3回目の許容ループと禁止用

* セル番号 a_{ij} (ここで $j = m_i, i = 1, 2, \dots, n$) から成る数列を要素とする集合 $X = \{a_{11}a_{12}\dots a_{1m_1}, a_{21}a_{22}\dots a_{2m_2}, \dots, a_{n1}a_{n2}\dots a_{nm_n}\}$ があるとき、各数列を構成するセル番号それぞれを要素とする集合 $XC = \{a_{11}, a_{12}, \dots, a_{1m_1}, a_{21}, a_{22}, \dots, a_{2m_2}, \dots, a_{n1}, a_{n2}, \dots, a_{nm_n}\}$ を X のセル集合と仮称する。

** 2つの集合 A, B があって $A \supset B$ のとき ($A \supseteq B$ ではないことに注意)、 A は B を完全にカバーすると仮称する。 $A \supset B$ となるためには $A^c \supset B^c$ ($A^c \supseteq B^c$ ではないことに注意) が成立せねばならない。

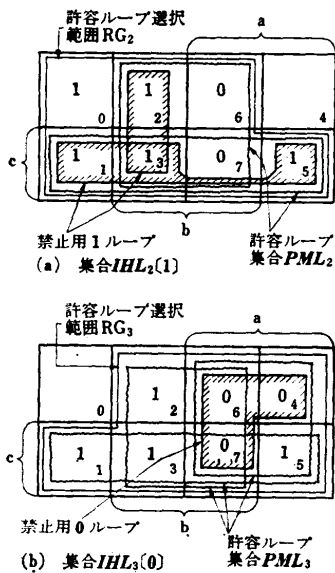


図2 $f = \sum(0, 1, 2, 3, 5)$ に対する許容ループと禁止用ループの生成 (a) 第2回目, (b) 第3回目
Fig. 2 Generation of permissible Loops and inhibiting Loops (a) The second time, (b) The third time.

ループのカルノー図上での生成をそれぞれ(a)と(b)に図示したものである。図2(a)より, $PML_2 = \{1357, 2367\}$ であるから, PML_1 は PML_2 に完全にカバーされる。

【定理3】 規則1によって作られる許容ループは, 最初に作られるものを除き, 0セルと1セルの両方を含む。

(証明) 規則1と式(2)で $i=2$ とすることにより,

$$PML_i = IHL_i[\bar{\alpha}], \quad (8)$$

$$PML_f \subset RG_2 = IHL_i[\bar{\alpha}]^c \cup DG[\alpha]. \quad (9)$$

これらの式と定理2より,

$$PML_f \subset PML_f \subset PML_f \cup DG[\alpha]. \quad (10)$$

式(10)より PML_2 は $\bar{\alpha}$ セルのみの PML_1 と α セルのみの $DG[\alpha]$ の一部とから作られる。すなわち PML_2 には0と1が混在する。定理2より $i \geq 3$ の PML_i はすべて PML_2 を完全にカバーするから, $i=1$ を除くすべての PML_i には0と1が混在する。(QED)

【例5】 図1で $PML_1 = \{67\}$ は0セルのみより成るが, 図2(a)の $PML_2 = \{1357, 2376\}$ で, セル1, 3, 5は1セル, セル7は0セルであり, またセル2, 3は1セル, セル6, 7は0セルである。

【定理4】 規則1に従って生成された禁止用 α ループは, つぎの2つの条件,

- (1) 空集合ではない,
 - (2) データグループ $DG[\alpha]$ に属している,
- を満足するときのみ採用できる。なお, α は0または1を表す(禁止用ループ採用条件定理)。

(証明) 規則1で条件(1)が成立しないときは禁止用ループとならないことは明らかである。また, 禁止用ループとして生成されたもので, (i) α ループを求めようとしたのに $\bar{\alpha}$ ループとなるもの, (ii) 1セルと0セルが混在するもの, および(iii)禁止したにもかかわらず, 禁止前の許容ループのまま残るものいずれか1つが満たされると, 禁止用ループとして採用できない。ここで, 定理3より許容ループは0セルと1セルの両方を含むから, (iii)は(ii)に統合でき, さらに(i)と(ii)は, ともに生成した禁止用 α ループが α データグループ $DG[\alpha]$ に含まれない場合に統合できる。したがって条件(2)が満足されないと禁止用 α ループとして採用できない。(QED)

【定理5】 規則1を繰り返して最後に得られる許容ループの集合は全セルを含む(終了条件定理)。

(証明) もし最後に得られる許容ループが全セルを含まないものとする, この許容ループを基にしてこれよりも大きな許容ループを作り得ることになり, これは最後に得られる許容ループという前提に反することより容易に証明できる。(QED)

【例6】 図2の第3回目を基にし第4回目に許容ループを求めると, $\{01234567\}$ が得られる。

3. アルゴリズムと計算例

前章の結果より一線入力 NAND ゲート回路生成のアルゴリズムを述べ, 計算例を検討する。

3.1 アルゴリズム

つぎの各ステップに分けてアルゴリズムを述べる。

【ステップ1】 n 変数の全許容ループを記憶する。

【ステップ2】 最大最小項のセルが α であれば状態記憶を α とする。ここで α は0または1を表す。

【ステップ3】 禁止用ループを生成済みか否か判定し, 生成済みでなければステップ4へ, 生成済みならステップ5へ移る。

【ステップ4】 このステップは禁止用ループ生成の初回である。規則1より, 状態記憶が α なら, α データグループ $DG[\alpha]$ に含まれる最大の許容ループの集合をステップ1の記憶内容から選択し, これを禁止用 α ループの集合とする。さらにステップ6に移る。

【ステップ5】 このステップは規則1に従う。

(1) ステップ3の判定で禁止用ループが作られていることがわかるので、状態記憶が α ならば前の段階で生成された禁止用 $\bar{\alpha}$ ループの集合のセル集合と α データグループ $DG[\alpha]$ との和集合 RG を生成する。

(2) この和集合 RG が全セルを含むか否かを調べ、含まなければ定理5の終了条件を満たさないから、この RG に含まれる最大の許容ループをステップ1の記憶内容からすべて選出し、(3)へ移る。 RG が全セルを含めば終了条件を満足するから、ステップ8に移る。

(3) (2)で求めた各許容ループをそれまでに求めた禁止用ループの組合せで禁止し、複数個の新たな禁止用ループの候補を作る。この新候補は、状態記憶が α ならば禁止用 α ループの候補となる。

[ステップ6] ステップ4で求めたもの、またはステップ5の候補の中で定理4の禁止用ループ採用条件を満足するものを新たな禁止用 α ループの集合として記憶し、さらにこれまで生成した禁止用ループの集合に追加記憶する。なお、この新禁止用ループの中にセル内容が同じものがあれば、それらの生成に用いた禁止用ループの使用回数の少ないものを選ぶ。

[ステップ7] 状態記憶内容 α を $\bar{\alpha}$ に反転させてステップ3に戻る。

[ステップ8] ステップ5の(2)で終了条件が満たされた場合、つぎの処理を行う。

(1) 得られた最終の禁止用ループの集合の要素の中で他のものに含まれるものは除去し、残った要素より成る禁止用ループの集合の部分集合を作る。

(2) これらの部分集合の中で要素数の最も少ない方から多い方に順に調べてゆき、部分集合のセル集合が一番最初に $DG[\alpha]$ と一致するものを採用する。

(3) 状態記憶 α が1、または0のときは、(2)で求めた部分集合の各要素の論理関数表現の否定を、それぞれさらに1回、または2回否定して解とする。

以上のアルゴリズムのフローチャートを図3に示す。

3.2 計算例

図1と2に示した3変数関数 $f = \Sigma(0, 1, 2, 3, 5)$ を例として前節のアルゴリズムの使用例を示す。

ステップ1で式(1)に示した3変数にたいする許容ループを記憶する。最大最小項セル(セル番号7)は0であるから、ステップ2で状態記憶を0とする。ステップ3で禁止用ループがまだ求まっていないことを判定し、ステップ4で状態記憶0に応じた初回の禁止

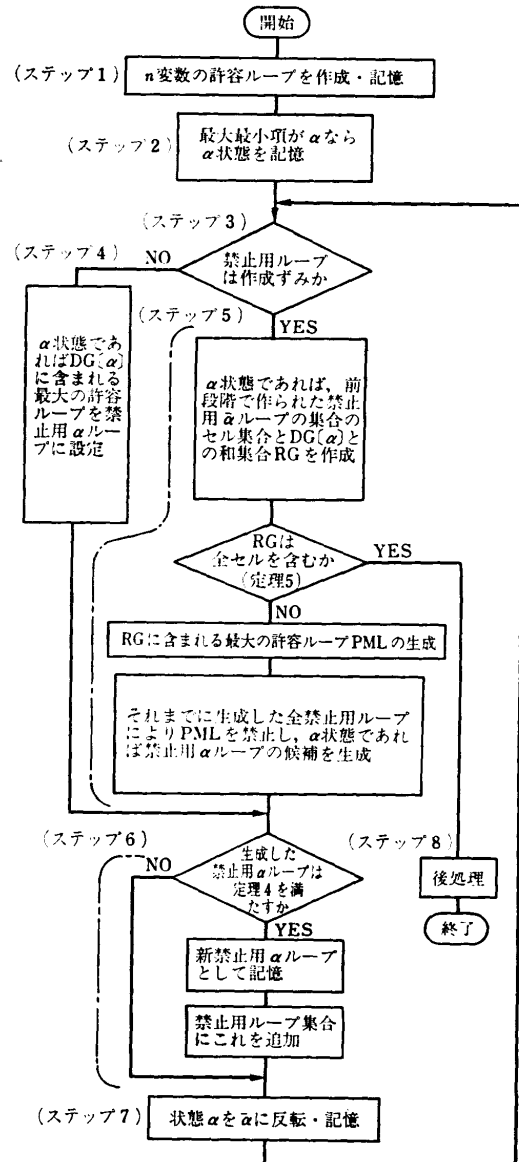


図3 本アルゴリズムのフローチャート
Fig. 3 Flow chart of this algorithm.

用0ループとして、図1に示す0データグループ $DG[0] = \{4, 6, 7\}$ に含まれる最大の許容ループである $PML_1 = \{67\}$ をステップ1の結果から選出する。ステップ6で、これはそのまま禁止用の0ループの集合 $IHL_1[0] = \{67\}$ となり、さらに禁止用ループの集合 $IHL = \{67\}$ としても記憶される。ステップ7で状態記憶を0から1に反転させ、ステップ3で、禁止用ループが生成済みであることを判定してステップ5に移る。状態記憶が1なので前段階で作成された禁止用0ループの集合のセル集合 $IHL_1[0]^c = \{6, 7\}$ と1デー

タグループ $DG[1]=\{0, 1, 2, 3, 5\}$ との和集合 $RG_2=\{0, 1, 2, 3, 5, 6, 7\}$ を求める。これを図2(a)に示す。この RG_2 はセル4を欠くから定理5の終了条件を満たさない。そこで GR_2 に含まれる最大の許容ループの集合をステップ1の内容から選出すると、 $PML_2=\{1357, 2367\}$ となる。この PML_2 の各要素をそれまでに得られた禁止用ループの集合 $IHL_1=\{67\}$ で禁止し、新たな禁止用1ループの集合 $IHL_2[1]=\{135, 23\}=\{1357(67), 2367(67)\}$ を得る。ステップ6でこの集合のセル集合 $IHL_2[1]^c=\{1, 2, 3, 5\}$ は $DG[1]=\{0, 1, 2, 3, 5\}$ に含まれ定理4を満足するから、 $IHL_2[1]$ は採用・記憶され、さらに禁止用ループ集合を更新し、 $IHL=\{67, 135, 23\}=\{67, 1357(67), 2367(67)\}$ として記憶される。ステップ7で状態記憶を1から0に反転させる。以後同様にステップ3に戻り、さらにステップ5で禁止用0ループ $IHL_3[0]$ を生成する。そのため、図2(b)に示すように、図2(a)の段階で生成された禁止用1ループ集合のセル集合 $IHL_2[1]^c=\{1, 2, 3, 5\}$ と $DG[0]=\{4, 6, 7\}$ との和集合 $RG_3=\{1, 2, 3, 4, 5, 6, 7\}$ に含まれる最大の許容ループの集合 $PML_3=\{1357, 2367, 4567\}$ を得る。この PML_3 の各要素をそれまでに得られた禁止用ループ $IHL=\{67,$

表1 $f=\Sigma(0, 1, 2, 3, 5)$ の最終回禁止用0ループ
Table 1 Inhibiting 0 Loops at last time for $f=\Sigma(0, 1, 2, 3, 5)$.

番号 No.	禁止用0ループ集合の要素		セル内容	定理5で 満足され ない条件 番号	採用○か 否×か
	禁止形態	セル構成			
1	1357(67)	135	1のみ	(2)	×
2	1357(135)	7	0のみ		◎
3	1357(23)	157	0, 1混在	(2)	×
4	1357(67)(135)	φ	なし	(1)	×
5	1357(67)(23)	15	1のみ	(2)	×
6	1357(135)(23)	7	0のみ		○
7	1357(67)(135)(23)	φ	なし	(1)	×
8	2367(67)	23	1のみ	(2)	×
9	2367(135)	267	0, 1混在	(2)	×
10	2367(23)	67	0のみ		◎
11	2367(67)(135)	2	1のみ	(2)	×
12	2367(67)(23)	φ	なし	(1)	×
13	2367(135)(23)	67	0のみ		○
14	2367(67)(135)(23)	φ	なし	(1)	×
15	4567(67)	45	0, 1混在	(2)	×
16	4567(135)	467	0のみ		◎
17	4567(23)	4567	0, 1混在	(2)	×
18	4567(67)(135)	4	0のみ		◎
19	4567(67)(23)	45	0, 1混在	(2)	×
20	4567(135)(23)	467	0のみ		○
21	4567(67)(135)(23)	4	0のみ		○

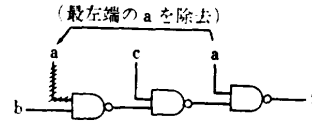


図4 $f=\Sigma(0, 1, 2, 3, 5)$ にたいする一線入力
NAND ゲート回路
Fig. 4 Single-rail input NAND gate network
for $f=\Sigma(0, 1, 2, 3, 5)$.

135, 23} の要素のすべての組合せで禁止して求まった新たな禁止用0ループ集合 $IHL_3[0]$ の各要素を表1に示す。ステップ6として表1で定理4の条件(1)と(2)を満足しない要素を除き、○印で示す No. 2, 6, 10, 13, 16, 18, 20 および 21 を得る。これらの中でセル構成が同じになるものについて禁止する回数の少ない方を残し、◎印で示す No. 2, 10, 16 および 18 を得る。つづいてステップ7で状態記憶を0から1に変え、ステップ3に戻り、さらにステップ5で $RG_4=\{0, 1, 2, 3, 4, 5, 6, 7\}$ を得る。これは定理5の終了条件を満たすから、ステップ8に移りつぎの後処理を行う。

- (1) 要素相互間の包含関係を調べると、No. 2, 10 および 18 はすべて No. 16 に含まれるので、No. 16, すなわち $467=4567(135)$ を採用する。
- (2) このNo. 16 のセル集合は $DG[0]=\{4, 6, 7\}$ と一致する。
- (3) ここで $467=4567(135)=4567(1357(67))$ が成立し、4567はaに、1357はcに、また67はabに相当するから、元の式 $f=\Sigma(0, 1, 2, 3, 5)$ は

$$f = \overline{acab} \tag{11}$$

で表される。この論理関数は図4の一線入力 NAND ゲート回路で実現される。

4. 演算結果と検討

本アルゴリズムに従い作成した LISP プログラムをミニコンピュータ $\mu VAX-II$ (DEC 製) 上で実行させた。その結果を述べ、その検討と改善法を示す。

4.1 演算結果

まず3変数関数の中で P 同値類の代数関数 68 個について生成した回路のゲート数、入力結線数、および段数を、L. Hellerman⁶⁾による一線入力3変数 NAND ゲートにたいする最小回路と比較した。その結果を表2に示す。またゲート数が7個以下、段数が3段以下の4変数関数の8事例について行った同様な評価を表3に示す⁴⁾。なお、これらの表では、各関数を最小項和と8進数表示とで表している。8進数表示に当って

表 2 3変数論理関数にたいする最小回路と演算結果の比較
Table 2 Comparison between minimal circuits and operating results.

8進表示	関数		(ゲート数, 結線数, 段数)		最小回路への縮約手法	8進表示	関数		(ゲート数, 結線数, 段数)		最小回路への縮約手法
	最小項の和	論理関数式	最小回路	本アルゴリズム			最小項の和	論理関数式	最小回路	本アルゴリズム	
177	0123456	$\bar{a}+\bar{b}+c$	1, 3, 11, 3, 1			207	0127	$\bar{a}\bar{b}+\bar{a}c+abc$	5, 10, 35, 11, 4		R2
200	7	abc	2, 4, 22, 4, 2			027	0124	$\bar{a}b+\bar{a}c+\bar{b}c$	5, 10, 35, 10, 3		
217	01237	$\bar{a}+bc$	2, 4, 22, 4, 2			276	123457	$\bar{a}b+\bar{a}\bar{b}+c$	5, 10, 35, 11, 4		R1
277	0123457	$\bar{a}+\bar{b}+c$	2, 4, 22, 4, 2			032	134	$\bar{a}c+\bar{a}\bar{b}c$	5, 10, 35, 11, 3		R1
352	13567	$ab+c$	3, 5, 23, 5, 2			202	17	$abc+\bar{a}\bar{b}c$	5, 10, 35, 11, 4		R1
357	0123567	$\bar{a}+b+c$	3, 5, 23, 5, 2			216	1237	$\bar{a}b+\bar{a}c+bc$	5, 10, 45, 10, 4		
052	135	$c\bar{a}+c\bar{b}$	3, 5, 33, 5, 3			274	23457	$\bar{a}\bar{b}+\bar{a}c+\bar{a}b$	5, 11, 45, 11, 4		
057	01235	$\bar{a}+\bar{b}c$	3, 5, 33, 6, 3		R1	076	12345	$\bar{a}b+\bar{a}\bar{b}+\bar{a}c$	5, 11, 35, 11, 3		
010	3	$\bar{a}bc$	3, 5, 33, 5, 3			030	34	$\bar{a}\bar{b}c+\bar{a}bc$	5, 11, 35, 12, 3		R1
250	357	$ac+bc$	3, 6, 23, 6, 2			006	12	$\bar{a}\bar{b}c+\bar{a}bc$	5, 11, 35, 11, 3		
376	1234567	$a+b+c$	4, 6, 24, 6, 2			232	1347	$\bar{a}c+bc+\bar{a}\bar{b}c$	5, 11, 46, 14, 4		R2, R1, R2'
037	01234	$\bar{a}+\bar{b}c$	4, 6, 34, 7, 3		R2	176	123456	$\bar{a}\bar{b}+\bar{b}c+\bar{a}c$	5, 12, 35, 12, 3		
007	012	$\bar{a}\bar{b}+\bar{a}c$	4, 6, 34, 6, 3			150	356	$\bar{a}\bar{b}c+\bar{a}\bar{b}c+abc$	5, 15, 35, 15, 3		
002	1	$\bar{a}bc$	4, 6, 34, 6, 3			211	037	$\bar{a}\bar{b}c+bc$	6, 10, 36, 12, 4		R2
256	12357	$\bar{a}b+c$	4, 6, 34, 7, 4		R1	353	013567	$\bar{a}\bar{b}+\bar{a}b+c$	6, 10, 36, 12, 4		R2
212	137	$\bar{a}c+bc$	4, 6, 44, 7, 4		R1	031	034	$\bar{a}bc+\bar{b}c$	6, 10, 36, 12, 5		R2, R1
254	2357	$\bar{a}b+ac$	4, 7, 37, 16, 4		R1, R2, R1'	053	0135	$\bar{a}\bar{b}+\bar{a}c+\bar{b}c$	6, 10, 36, 11, 5		R2
016	123	$\bar{a}b+\bar{a}c$	4, 7, 34, 7, 3			075	02345	$\bar{a}b+\bar{a}\bar{b}+\bar{a}c$	6, 10, 46, 12, 5		R1
056	1235	$\bar{a}b+\bar{b}c$	4, 8, 34, 8, 3			255	02357	$\bar{a}c+bc+ac$	6, 10, 46, 12, 4		R1, R2
350	3567	$\bar{a}b+ac+bc$	4, 9, 24, 9, 2			011	03	$\bar{a}bc+\bar{a}\bar{b}c$	6, 11, 46, 12, 5		R2
054	235	$\bar{a}b+\bar{a}\bar{b}c$	4, 9, 34, 9, 3			201	07	$abc+\bar{a}\bar{b}c$	6, 11, 36, 13, 4		R2
156	12356	$\bar{a}c+\bar{b}c+\bar{b}c$	4, 9, 34, 9, 3			275	023457	$\bar{a}\bar{b}+\bar{a}c+\bar{a}b+\bar{a}c$	6, 11, 46, 13, 4		R1, R2
157	012356	$\bar{a}+\bar{b}c+\bar{b}c$	4, 9, 34, 10, 3		R1	055	0235	$\bar{a}b+\bar{a}c+\bar{a}\bar{b}c$	6, 11, 47, 15, 5		R2, R1
152	1356	$\bar{a}c+\bar{b}c+abc$	4, 10, 34, 10, 3			251	0357	$\bar{a}\bar{b}c+ac+bc$	6, 12, 47, 15, 4		R1, R2, R2'
050	35	$\bar{a}\bar{b}c+\bar{a}\bar{b}c$	4, 10, 34, 10, 3			036	1234	$\bar{a}b+\bar{a}c+\bar{a}\bar{b}c$	6, 14, 36, 14, 3		
001	0	$\bar{a}\bar{b}c$	5, 7, 35, 7, 3			227	01247	$\bar{a}\bar{b}+\bar{a}c+\bar{b}c+abc$	6, 14, 36, 16, 4		R2
253	01357	$c+\bar{a}\bar{b}$	5, 7, 35, 8, 4		R2	206	127	$\bar{a}\bar{b}c+\bar{a}\bar{b}c+abc$	6, 15, 36, 16, 4		R2
013	013	$\bar{a}\bar{b}+\bar{a}c$	5, 7, 45, 8, 5		R1	236	12347	$\bar{a}c+bc+\bar{a}b+\bar{a}\bar{b}c$	6, 15, 47, 18, 4		R2, R1'
213	0137	$\bar{a}\bar{b}+bc$	5, 8, 35, 9, 4		R2	153	01356	$\bar{a}c+\bar{a}\bar{b}+\bar{b}c+abc$	7, 14, 47, 17, 5		R2, R1
033	0134	$\bar{a}c+\bar{b}c$	5, 8, 48, 16, 5		R1'	026	124	$\bar{a}\bar{b}c+\bar{a}\bar{b}c+\bar{a}\bar{b}c$	7, 15, 37, 18, 3		R1
237	012347	$bc+\bar{b}c+\bar{a}$	5, 9, 35, 11, 4		R2	051	035	$\bar{a}\bar{b}c+\bar{a}\bar{b}c+\bar{a}\bar{b}c$	7, 15, 48, 19, 5		
230	347	$\bar{a}\bar{b}c+bc$	5, 9, 35, 11, 4		R1	351	03567	$\bar{a}b+ac+bc+\bar{a}\bar{b}c$	7, 15, 48, 19, 4		
203	017	$abc+\bar{a}\bar{b}$	5, 9, 35, 10, 4		R2	151	0356	$\bar{a}\bar{b}c+\bar{a}\bar{b}c+abc+\bar{a}\bar{b}c$	7, 16, 59, 25, 5		比較的難解
233	01347	$\bar{a}\bar{b}+\bar{b}c+bc$	5, 9, 35, 10, 4		R2	226	1247	$abc+\bar{a}\bar{b}c+\bar{a}\bar{b}c+\bar{a}\bar{b}c$	7, 20, 48, 24, 4		

表 3 4変数論理関数にたいする最小回路と演算結果の比較
Table 3 Comparison between minimal circuits and operating results.

8進表示	関数		(ゲート数, 結線数, 段数)		最小回路への縮約手法
	最小項の和	論理関数式	最小回路	本アルゴリズム	
100000	15	$abcd$	2, 5, 2	2, 5, 2	
177752	1 3 5 6 7 8 9 10 11 12 13 14 15	$a+bc+d$	4, 7, 2	4, 7, 2	
164200	7 11 13 14 15	$abc+bcd+acd+abd$	5, 16, 2	5, 16, 2	
146124	2 4 6 10 11 14 15	$ac+\bar{a}\bar{b}\bar{d}+c\bar{d}$	5, 10, 4	6, 13, 4	比較的難解
177676	1 2 3 4 5 7 8 9 10 11 12 13 14 15	$a+d+(b\oplus c)$	6, 12, 3	6, 14, 4	R1
177576	1 2 3 4 5 6 8 9 10 11 12 13 14 15	$a+\bar{b}c+\bar{b}d+c\bar{d}$	6, 14, 3	6, 15, 4	R2
077776	1 2 3 4 5 6 7 8 9 10 11 12 13 14	$a\bar{c}+c\bar{d}+\bar{a}b+\bar{b}d$	6, 16, 3	6, 16, 3	
177127	0 1 2 4 6 9 10 11 12 13 14 15	$\bar{a}b+\bar{a}c+\bar{a}\bar{d}+\bar{b}c\bar{d}$	8, 18, 4	8, 20, 4	比較的難解

は、 n 変数の最小項の添字 $2^n-1 \sim 0$ を降順に並べ、最小項の存否によって対応する添字位置に1か0を記入し、これを8進数に変換している。表2より、このアルゴリズムによって得られた結果の中で、ゲート数が最小回路と一致するものは68個中58個(85%)、最小回路の平均ゲート数の2割増し、すなわち1個増しまで許すと66個(97%)となった。また結線数が最小回路と一致したものは28個、平均結線数の3割増し、すなわち3本増しまで許すと59個(85%)となった。つぎに段数が最小回路と一致したものは44個(65%)、平均段数の3割増し、すなわち1段増しまで許すと65個(96%)となった。演算に要したCPU時間は最小3sec、最大50secとなり、平均では12secであった。また、表3では、ゲート数については8例とも最小回路と一致し、段数とゲート数が一致しなかったものは3例のみであった。また、CPU時間は平均11secであった。

4.2 最小回路への改善法

表2, 3の結果でゲート数, 結線数, 段数のいずれかが最小にならなかったものにたいし、縮約を検討した。

[定理6] 多段縦続 NAND ゲート回路で、 k 段目の NAND ゲート G_A への入力 q (k 段目が並列 NAND ゲートを持つときは共通入力 q) と同じ入力 q を持つ NAND ゲート G_X が、 G_A より前段(出力側)に存在すれば、ゲート G_A への入力 q を除去できる(縮約定理1)。

[説明1] 図5(a)に示す多段縦続回路では、最後段(最左端)とそれよりも前段(右側)の NAND ゲートに同じ入力 q が存在するので、最後段の q を除去して図5(b)が得られる。この結果、結線数は減少する。

[例7] 図4に示す回路では最後段の入力線 a を除去でき、結線数は6から5に減少する。

[定理7] 多段縦続 NAND ゲート回路中で、ある段の l 個のゲート $G_{31}, G_{32}, \dots, G_{3l}$ はすべて並列にな

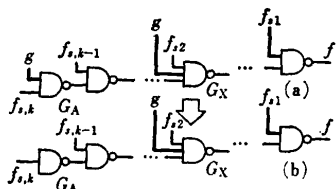


図5 定理6による縮約 (a) 縮約前, (b) 縮約後
Fig. 5 Reduction by Theorem 6 (a) Before reduction, (b) After reduction.

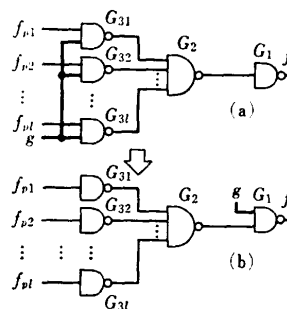


図6 定理7による縮約 (a) 縮約前, (b) 縮約後
Fig. 6 Reduction by Theorem 7 (a) Before reduction, (b) After reduction.

っており、しかもこれらの全ゲートに同一入力 q が接続されており、これらの全ゲートは前段の1個の NAND ゲート G_2 に接続され、さらに G_2 はその前段の NAND ゲート G_1 に接続されているとき、ゲート $G_{31}, G_{32}, \dots, G_{3l}$ から共通入力 q を除去してゲート G_1 の入力として q を追加できる(縮約定理2)。

[説明2] 定理7の内容を図6(a)および(b)に示す。結線数は大きく減少することがわかる。

[例8] 関数 $f = \Sigma(0, 3, 7)$ の回路を本手法で生成すると、図6(a)で $n=3, f_{p1}=a, f_{p2}=b, f_{p3}=c, q=\overline{bc}$ としたものと同一になる。これは定理7により図6(b)のように置換され、(ゲート数, 結線数, 段数)は(6, 12, 4)から(6, 10, 3)に縮約される。

これらの定理6および7を用いた縮約手法をそれぞれ縮約手法 R1 および R2 と呼ぶ。また、縮約手法 R1 とは対照的に前段のゲートへの入力 q があるとき、後段への入力 q を追加できる。この方法を縮約手法 R1' と呼ぶ。さらに、縮約手法 R2 とは対照的に前段ゲートへの入力 q を除去し、これをこのゲートの段を含み3段後の並列全ゲートへの共通入力として移すことができる。この方法を縮約手法 R2' と呼ぶ。

表2および3の実験結果には、得られた結果を最小回路に修正するのに用いた縮約手法が示してある。特に表2については、R1, R2, もしくは R1' のみを用いて縮約できる場合は40個中27個で、R1 と R2, または R2 と R1' を併用する場合は6個、これらの手法を3つ使いねばならない場合は3個、さらに縮約が比較的難解で別の手法を用いねばならなかったのが4個であった。これら4つの場合は、3変数間に排他的論理和の関係が成立するか、またはこれに近い関数の

場合であった。

5. 従来のマップファクタリング法との比較

本アルゴリズムによる方法（これを GK 法と仮称する）を従来のマップファクタリング法^{1),2)}（これを MF 法と仮称する）と比較すると、つぎのような点で異なっている。

(1) MF 法は試行錯誤的かつ発見的手法であるが、GK 法は確定的手法である。

(2) MF 法では、最初に選ばれる許容ループは、その個数、大きさ、種類について任意であり、しかもループ中に 0 セルと 1 セルを同時に含み得る。一方 GK 法では、0 セル、または 1 セルのどちらかのみを含む最大の許容ループである。

(3) MF 法では、2 回目以降に選択される許容ループについても、その個数、大きさ、種類とも任意である。一方 GK 法では許容ループの選択範囲を明確に限定し、その範囲内で最大のものを選んでいく。

(4) MF 法では選んだ 1 個の許容ループがそれ以前の禁止用 α ループによって禁止されて α セルのみより成るループが作られる。GK 法では選ばれた複数個の許容ループを、それまでに得られたすべての禁止用ループの組合せで禁止するようにしている。

(5) MF 法では、1 つの問題にたいして異なる出発点、すなわち異なる許容ループから出発していくつかの解を求め、それらの解を比較して最小解を求めるようにしている。これにたいし、GK 法では 1 つの問題にたいする演算は 1 回だけとし、その演算の中で最小化が達成されるよう配慮した。すなわち、

(a) 新禁止用ループの生成の都度、その論理関数形をも記憶し、セル構成が同一のものの中で、実際の回路のゲート数が少ない方を採用するよう配慮した。

(b) 最終的な禁止用 α ループ集合を決定する前に他の禁止用 α ループに含まれるものはすべて除去した。

(c) 最終的な禁止用 α ループ集合の部分集合の要素の少ない方から調べ、そのセル集合が α データグループ DG[α] と最初に一致するものを求める解とする。

なお、これら 3 つの手法の中で、(a) は複数解を単一解に限定したり、重複使用可能な関数の消失により最小化を阻害したりするので注意を要する。(c) は最小被覆解を求める 1 つの手法ではあるが、(b) での禁止用ループにたいする情報落ちを配慮すると、今後はむしろ (b) と (c) を最小被覆の複数解を求める手法として同時に処理する方が得策である。

このような若干の問題を残すが、従来の試行錯誤的な MF 法に比し、この GK 法は明確な規則に従った計算機向きの手法と言える。

6. む す び

本論文では、一線入力 NAND ゲート回路を生成するのに、従来手計算用として用いられていたマップファクタリング法を改善した計算機向きの手法を述べた。本論文のアルゴリズムでは、従来のような発見的かつ曖昧な設定法ではなく、許容ループを選択する範囲を一定の規則にしたがって設定し、この中から最大かつすべての許容ループを求め、これらの許容ループをそれまでに求まった禁止用ループで禁止する手法を繰り返す。このアルゴリズムを用いて LISP 言語によるプログラムを作成し、3 変数関数の P 同値類の 68 個と 4 変数関数 8 個について演算を実行し、ゲート数については 90% 近くまで、段数については 70% 近くまで最小回路に一致する回路が得られた。

最小被覆と複数解の導出法の改善、最小回路に達しなかったものへの縮約手法のプログラム化、CPU 時間の短縮については今後さらに検討する予定である。

謝辞 プログラム作成に尽力された谷水好彦（神田通信機）、佐藤薫夫（富士通 SSL）、山下義基（CSK）の三氏に謝意を表する。

参 考 文 献

- 1) Maley, G. A. and Earle, J.: *The Logic Design of Transistor Digital Computers*, John Wiley (1963).
- 2) Muroga, S.: *Logic Design and Switching Theory*, John Wiley (1972).
- 3) Muroga, S. and Ibaraki, T.: Design of Optimal Switching Networks by Integer Programming, *IEEE Trans. Comput.*, Vol. C-21, No. 6, pp. 573-582 (1972).
- 4) 魏, 清水: 多段フィードフォワード NOR 回路網の一設計法, 電子情報通信学会誌, Vol. J70-D, No. 2, pp. 325-334 (1987).
- 5) 後藤, 佐藤: 禁止用ループによる一線入力論理 NAND 回路の設計アルゴリズム, 昭和 62 年電子情報通信学会創立 70 周年記念総合全国大会論文集, 311 (1987).
- 6) Helleman, L.: A Catalog of Three-Variable Or-Invert and And-Invert Logical Circuits, *IEEE Trans. Electron. Comput.*, Vol. EC-12, No. 3, pp. 198-223 (1963).
- 7) 後藤: 一線入力 NAND ゲート回路の合成結果の簡単化, 昭和 63 年電子情報通信学会秋季全国

大会論文集, D-158 (1988).

(昭和 63 年 8 月 29 日受付)
(平成 元年 2 月 14 日採録)



後藤 公雄 (正会員)

大正 15 年生. 昭和 26 年東京大学第二工学部電気工学科卒業. 同年(株)日立製作所入社. 昭和 59 年同社退社. 同年畿徳工業大学工学部電気工学科教授. 昭和 63 年神奈川工科大学工学部情報工学科教授. 工学博士. これまで TV 中継装置, ドップラレーダ, 非同期多重通信方式等の開発, および非同期式順序回路の研究に従事. 現在, 計算機支援による論理設計, 計算機アーキテクチャ等の研究に従事し, 人工知能, 数値解析にも関心を持つ. 著書「パルス回路」(産業図書), 「詳解・デジタル IC 回路, 上, 下」(ラジオ技術社). 監訳書「高品質ソフトウェア」(近代科学社), 「フレンドリ・プログラミング」(近代科学社). 電子情報通信学会, 電気学会各会員.