計算機を用いた論理回路の自動設計

（第 1 報）

菅　原　英　一・堅固山　幸　治

（昭和51年10月31日受理）

Design Automation of Logical Circuit by Computer

(1st Report)

Eiichi Sugawara, Koji Kengoyama

1. 緒　言

論理回路の設計に関して現実的な問題は，複数の入力に対して所用の出力を得ようとするとき，いかに低価格に実現できるかであり，その実現のためには，与えられた論理式をできるだけ簡単なものとすべきことは異論のないところである。

本報告では，与えられた論理式を計算機を用いて一定の規則に従って簡単化し，その最も簡単な論理式が得られたら，X－Yプロッタによって記号化された論理回路を相乗項の和の形で図形出力しようとしたものであり，1出力の場合について一応の成果を得たので報告する。

2. 論理式の簡単化

2－1 簡単化の基準

最も簡単な論理式の定義はきわめて困難であるが，本報告ではその目安として以下のような簡単化の基準を与えることにした。

① 出力は相乗項の和の形式とするので，OR回路は必ず1個必要となる。

② AND回路の数が最少の論理式を最も簡単な論理式とする。

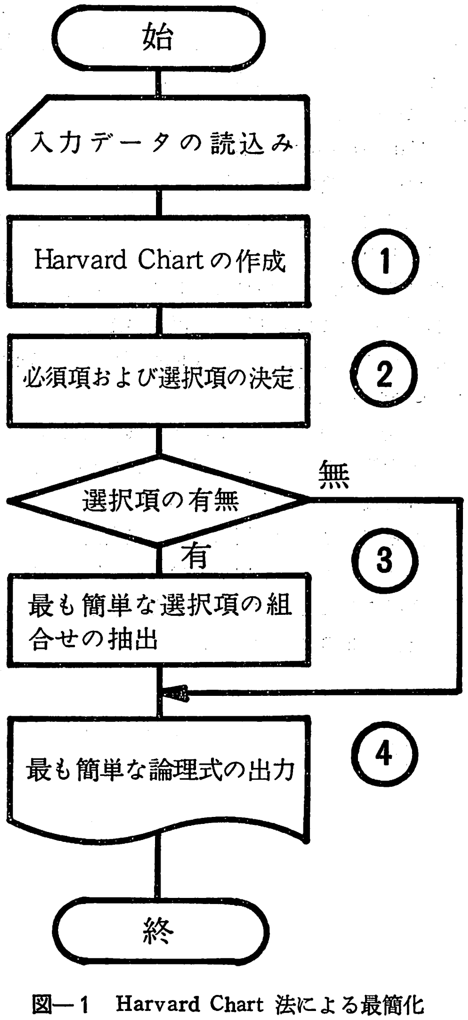
③ AND回路の数が等しいならば，各AND回路への入力数の総和が最少の論理式を最も簡単な論理式とする。

④ 論理式には否定形の論理変数も含まれているが，多くの場合は否定形の信号端子も同時に備えているので，ここではNOT回路の多少を簡単化の基準としては考慮しないことにする。

2－2 入力データの形式

例えば，論理関数Fが

$$F=(A+\overline{A}\cdot C)\cdot\overline{B}\cdot D+\overline{C+D}+B\cdot(A\cdot\overline{D}+\overline{A}\cdot\overline{C})+\overline{A}\cdot\overline{D}\cdot(B\cdot\overline{C}+\overline{B}\cdot C) \quad\cdots\cdots\cdots\cdots\cdots(1)$$



図－1　Harvard Chart 法による最簡化の処理過程

で与えられるとき，このままでは計算機への入力として取扱えないので，論理変数の否定形は〈　〉で囲むことにする。したがって，(1)式は計算機への入力データとし

ては次のように表わされる。

$$F=(A+<A>C)<B>D+<C+D>+B(A<D>+<A><C>)+<A><D>(B<C>+<B>C) \cdots\cdots (2)$$

2—3　HARVARD CHART法

論理式の簡単化の方法としては，真理値表による直観的な方法，論理代数を用いた Cut and Try 法，Veitch Map による方法，Karnaugh Map による方法，Quine-Mc Clusky 法および Harvard Chart 法等が知られている。本報告では簡単化の方法として Harvard Chart 法を採用し，Chart の処理を計算機で行なうことにした。

Harvard Chart 法による簡単化の手順をブロック・ダイヤグラムを用いて示すと図—1のようである。このブロック・ダイヤグラムに従って(1)式の論理関数（入力データとしては(2)式）を例にとって，その手順と途中結果を以下に述べる。

① Harvard Chart の作成

すべての論理変数の組合せに対応する2進数を10進数に置き換えて Chart を完成する。たとえば，否定形の変数には'0'，そうでない変数には'1'を対応させるので，$A\cdot\overline{B}\cdot C$に対しては'101'という2進数が得られ，これを10進数に置き換えて'5'とする。このようにして完成された Chart を表—1に示す。

② Chart の処理

ⓐ 表—1のＩＮＰＵＴの欄は真理値表による関数出力値であるから，ＩＮＰＵＴが'0'の行は与えられた論理関数に含まれていないことになる。したがって，この行は消去しても差支えない。この操作としてＩＮＰＵＴが'0'の各行の10進数を'*'で置き換えることにする。ここまでの途中結果を示すと表—2のようである。

	INPUT	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
m_0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m_1	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1
m_2	1	0	0	1	0	0	1	0	1	0	2	1	0	2	2	2
m_3	1	0	0	1	1	0	1	1	1	1	3	1	1	3	3	3
m_4	1	0	1	0	0	1	0	0	2	2	0	2	2	0	4	4
m_5	1	0	1	0	1	1	0	1	2	3	1	2	3	1	5	5
m_6	0	0	1	1	0	1	1	0	3	2	2	3	2	2	6	6
m_7	0	0	1	1	1	1	1	1	3	3	3	3	3	3	7	7
m_8	1	1	0	0	0	2	2	2	0	0	0	4	4	4	0	8
m_9	1	1	0	0	1	2	2	3	0	1	1	4	5	5	1	9
m_{10}	0	1	0	1	0	2	3	2	1	0	2	5	4	6	2	10
m_{11}	1	1	0	1	1	2	3	3	1	1	3	5	5	7	3	11
m_{12}	1	1	1	0	0	3	2	2	2	2	0	6	6	4	4	12
m_{13}	0	1	1	0	1	3	2	3	2	3	1	6	7	5	5	13
m_{14}	1	1	1	1	0	3	3	2	3	2	2	7	6	6	6	14
m_{15}	0	1	1	1	1	3	3	3	3	3	3	7	7	7	7	15

表—1　Harvard Chart

	INPUT	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
m_0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m_1	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_2	1	0	0	1	0	0	1	0	1	0	2	1	0	2	2	2
m_3	1	0	0	1	1	0	1	1	1	1	3	1	1	3	3	3
m_4	1	0	1	0	0	1	0	0	2	2	0	2	2	0	4	4
m_5	1	0	1	0	1	1	0	1	2	3	1	2	3	1	5	5
m_6	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_7	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_8	1	1	0	0	0	2	2	2	0	0	0	4	4	4	0	8
m_9	1	1	0	0	1	2	2	3	0	1	1	4	5	5	1	9
m_{10}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{11}	1	1	0	1	1	2	3	3	1	1	3	5	5	7	3	11
m_{12}	1	1	1	0	0	3	2	2	2	2	0	6	6	4	4	12
m_{13}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{14}	1	1	1	1	0	3	3	2	3	2	2	7	6	6	6	14
m_{15}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

表—2　Harvard Chart 処理 ⓐ

	INPUT	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
m_0	1	*	*	*	*	*	*	*	*	*	0	*	0	0	0	0
m_1	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_2	1	*	*	*	*	*	*	*	*	*	*	1	0	*	*	2
m_3	1	*	*	*	*	*	*	*	*	*	*	1	*	*	3	3
m_4	1	*	*	*	*	*	*	*	*	*	0	2	*	0	4	4
m_5	1	*	*	*	*	*	*	*	*	*	*	2	*	*	*	5
m_6	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_7	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_8	1	*	*	*	*	*	*	*	*	*	0	4	*	4	0	8
m_9	1	*	*	*	*	*	*	*	*	*	*	4	5	*	*	9
m_{10}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{11}	1	*	*	*	*	*	*	*	*	*	*	*	5	*	3	11
m_{12}	1	*	*	*	*	*	*	*	*	*	0	*	6	4	4	12
m_{13}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{14}	1	*	*	*	*	*	*	*	*	*	*	*	6	*	*	14
m_{15}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

表—3　Harvard Chart 処理 ⓑ

	INPUT	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
m_0	1	*	*	*	*	*	*	*	*	*	0	*	0	*	*	*
m_1	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_2	1	*	*	*	*	*	*	*	*	*	*	1	0	*	*	*
m_3	1	*	*	*	*	*	*	*	*	*	*	1	*	*	3	*
m_4	1	*	*	*	*	*	*	*	*	*	0	2	*	*	*	*
m_5	1	*	*	*	*	*	*	*	*	*	*	2	*	*	*	*
m_6	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_7	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_8	1	*	*	*	*	*	*	*	*	*	0	4	*	*	*	*
m_9	1	*	*	*	*	*	*	*	*	*	*	4	5	*	*	*
m_{10}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{11}	1	*	*	*	*	*	*	*	*	*	*	*	5	*	3	*
m_{12}	1	*	*	*	*	*	*	*	*	*	0	*	6	*	*	*
m_{13}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{14}	1	*	*	*	*	*	*	*	*	*	*	*	6	*	*	*
m_{15}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

表—4　Harvard Chart 処理 ⓒ

	INPUT	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
m_0	1	*	*	*	*	*	*	*	*	*	0	*	0	*	*	*
m_1	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_2	1	*	*	*	*	*	*	*	*	*	*	1	0	*	*	*
m_3	1	*	*	*	*	*	*	*	*	*	*	1	*	*	3	*
m_4	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_5	1	*	*	*	*	*	*	*	*	*	*	◎	*	*	*	*
m_6	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_7	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_8	1	*	*	*	*	*	*	*	*	*	0	4	*	*	*	*
m_9	1	*	*	*	*	*	*	*	*	*	*	4	5	*	*	*
m_{10}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{11}	1	*	*	*	*	*	*	*	*	*	*	*	5	*	3	*
m_{12}	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{13}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
m_{14}	1	*	*	*	*	*	*	*	*	*	*	*	◎	*	*	*
m_{15}	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

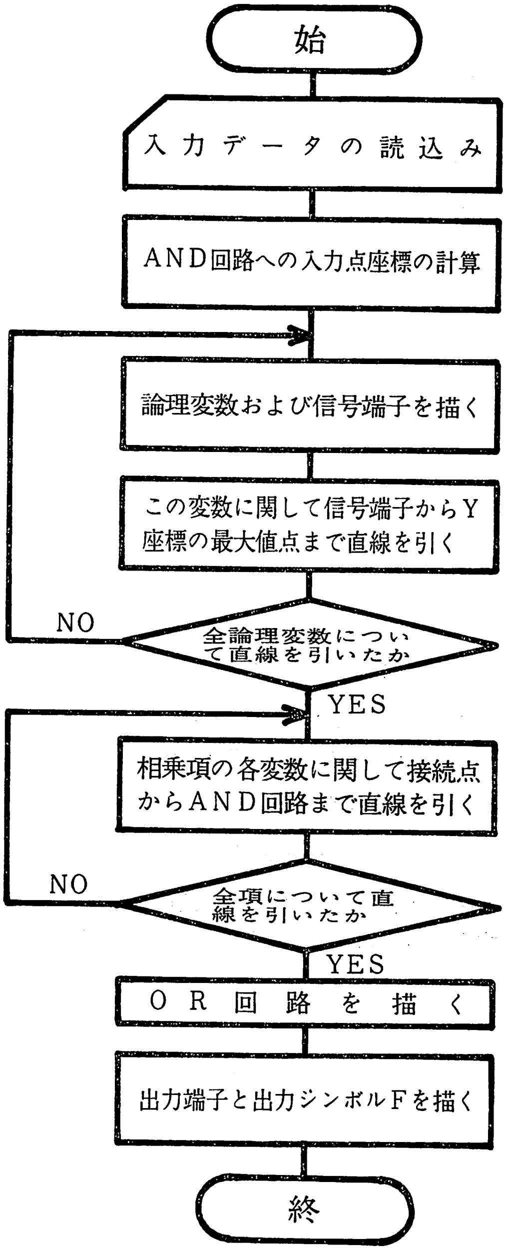
表―5　Harvard Chart 処理 ⓓ

ⓑ　次に各列を調べて，上の操作で'*'に置き換えられた欄の数と同じ数があったら，これは前に消去された論理変数と同じ変数を表わしているから同様に'*'で置き換えることにする。ここまでの途中結果を示すと表―3のようである。

ⓒ　次に各行を左端から調べて，消去されていない（'*'になっていない）項を見出し，その項の変数をすべて含む消去されていない項をすべて消去する。すなわち，表―3における m_0 行について調べると，消去されていない項はＣＤであるから，これを含むＡＣＤ，ＢＣＤ，ＡＢＣＤの３項を消去するためこれらの欄を前と同様に'*'で置き換えることにする。ここまでの途中結果を示すと表―4のようである。

ⓓ　各行について消去されていない項が唯一つあれば，これは必ず必要な項（必須項という）であり，この例では m_5 行目のＡＢＣ列の'2'および m_{14} 行目のＡＢＤ列の'6'がこれに相当する。すなわち，'2'および'6'を進表現に戻すと'10'および'110'となるから$\overline{A}B\overline{C}$および$AB\overline{D}$が必須項である。これら必須項についてはその欄を'◎'で置き換えることにする。このようにして消去されなかった項は選択項として残り，この組み合わせによって最も簡単な論理式が得られる。なお，上述の２項が必須項として取上げられると，これらと同列にある同値の項は選択項からは除去される。さらに，この除去された項と同行にある未消去の項も不要となるので除去される。除去の操作は前述同様'*'で置き換えることにすると，ここまでの結果は表―5のようになる。

③　選択項の組み合わせ

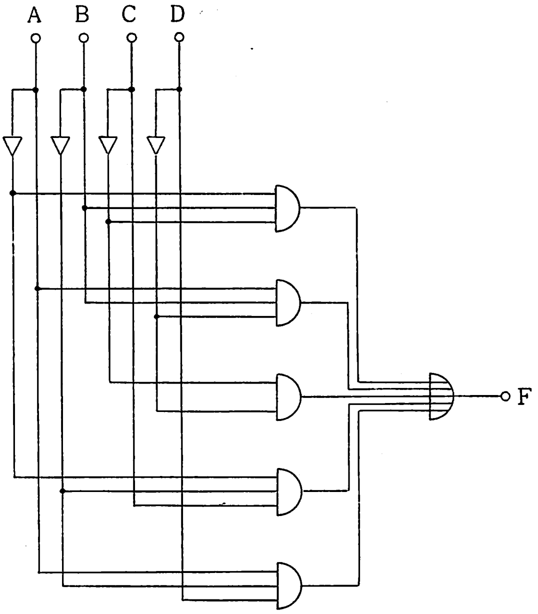


図―2　論理回路図の出力過程

前述までの操作で選択項がなければ必須項のみの和が最も簡単な論理式となるが，もし選択項あがる場合には，すべての組み合わせを調べて２―１の簡単化の基準に合った組み合わせを決定する。

④　出　　力

必須項および選択項の組み合わせが決まったならば，それは最も簡単な論理式であるから出力する。このような操作の結果，出力された(1)式の最も簡単な論理式は次のようになる。



図— 3　論理回路図の出力

$$F=\overline{A}\cdot B\cdot\overline{C}+A\cdot B\cdot\overline{D}+\overline{C}\cdot\overline{D}$$
$$+\overline{A}\cdot\overline{B}\cdot C+A\cdot\overline{B}\cdot D \cdots\cdots\cdots\cdots\cdots\cdots(3)$$

3.　論理回路の図形出力

3—1　入力データの変換

2—3で得られた出力データを論理回路図作成の入力とするため，以下のようにデータの変換を行なう。すなわち，(3)式の4変数の論理式の各項を(4)式のように対応する2進表現で置き換えることにする。

F＝010␣＋11␣0＋␣␣00＋001␣＋10␣1……(4)

なお，上式での␣は空白を意味する。

3—2　X—Yプロッタによる図形出力

図—2は(4)式をデータとして入力した場合のX—Yプロッタによる図形出力の処理をブロック・ダイヤグラムで示したものであり，その結果出力された論理回路図が図—3である。

4.　結　　言

(1)式のような論理関数が与えられた場合，これを最簡化の処理によって最も簡単な関数とし，さらに記号化された論理回路として図形出力するという初期の目的は一応達成された。しかし，これは1出力についてのみ達成されたということで，今後は多出力の場合についても最簡化から図形出力という処理が可能となるようプログラムの拡張を考えている。

参 考 文 献

1)　当麻喜弘：ディジタル回路の論理設計入門
2)　佐々木次郎，中野馨：トランジスタ・ディジタル計算機の設理設計
3)　Montgomery Phister, Jr.：ディジタル計算機の論理設計