

多値多しきい値関数との類似性に着目した ユニナリ関数の実現について

宮崎大学工学部 石塚興彦

1. ま え が き

ユニナリ (unary) 関数は 1 変数論理関数であり, r 値論理においては, r^r 個のユニナリ関数が存在する. このユニナリ関数はポスト代数を始め, 種々の多値論理代数系に取り入れられ, 多値論理における重要な関数となっている⁽¹⁾. ユーナリ関数の回路実現に関しては, 従来より, 3 値論理変数に対するもの^{(2), (3)} や, リテラル (literal) と呼ばれる多値論理変数を任意の 2 値の変数に変換した関数に対するもの⁽⁴⁾⁻⁽⁶⁾ などが提案されている.

本論文は, ユーナリ関数と多値多しきい値関数との類似性に着目し, 多値多しきい値回路網を用いて任意のユニナリ関数を実現する方法について述べるものである. まずユニナリ関数が多値多しきい値関数によって表現できることが示され

る。このことは多値多しきい値関数を実現できる素子又は回路網は、任意のユーナリ関数も実現できることを意味する。著者は既に電流動作形回路を用いたいくつかの多値多しきい値回路網の構成について発表してきた⁽⁷⁾。本論文では、これらを含めて、最近多値論理への適用が図られている種々のIC (MOS, ECL, I²L, CCD等)を用いたユーナリ関数の回路実現について検討する。

2. 準 備

2.1 多値多しきい値関数

次式(1)で定義される関数 $g_k^m(e)$ を m 値多しきい値関数(一般には多値多しきい値関数)と呼ぶ⁽⁸⁾。

$$g_k^m(e) = \begin{cases} r_0 & (e < t_0 \text{ のとき}) \\ r_j & (t_j \leq e < t_{j+1} \text{ のとき}) \\ r_k & (t_k \leq e \text{ のとき}) \end{cases} \quad (1)$$

但し、 e = アナログ入力 (excitation)

$$r_0, r_j, r_k \in L, L = \{0, 1, \dots, m-1\}$$

$$t_j = j \text{ 番目のしきい値, } t_j < t_{j+1}$$

$$j = 1, 2, \dots, k-1.$$

関数 $g_k^m(e)$ を図示すると、図1となる。

ここで、 m 値 n 変数論理関数 $f^m(\mathbf{x})$ を考える。但し、 $\mathbf{x} =$

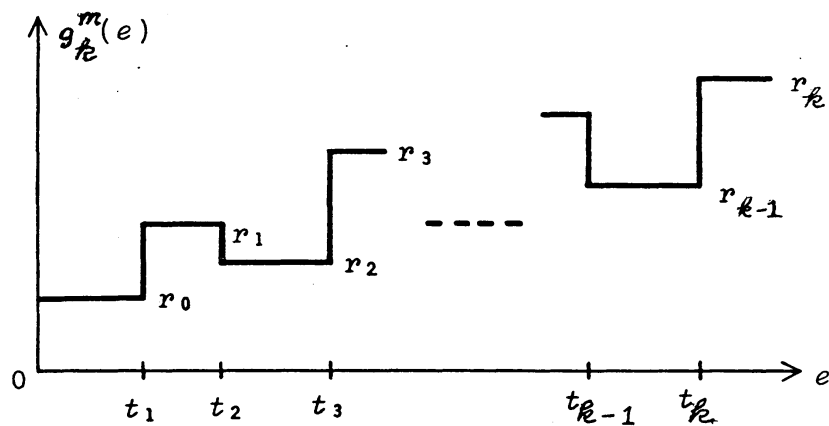


図1. m 値をとり値関数 $g_k^m(e)$.

$(x_1, x_2, \dots, x_n), x_i \in L, i = 1, 2, \dots, n$. いま関数の値が異なる \mathbf{x} の項 (Vertex) に対しては必ず e の値が異なるように, \mathbf{x} と e との間に対応をつけることができれば,

$$f^m(\mathbf{x}) = g_k^m(e) \quad (2)$$

と置くことができる。このことを, 多値論理関数の多値多しきい値関数への変換という。

入力ベクトル \mathbf{x} を上述の条件の下で e に対応づける機能と, 図1の入出力特性とを合せ持つ素子又は回路網によって, $f^m(\mathbf{x})$ は実現される。このような回路網を多値多しきい値回路網という。

2.2 しきい素子の回路実現

しきい素子は入力変数の重み付け, 重み付けされた変数の加算, その和に応じたしきい値判定と2値出力の発生を各機能

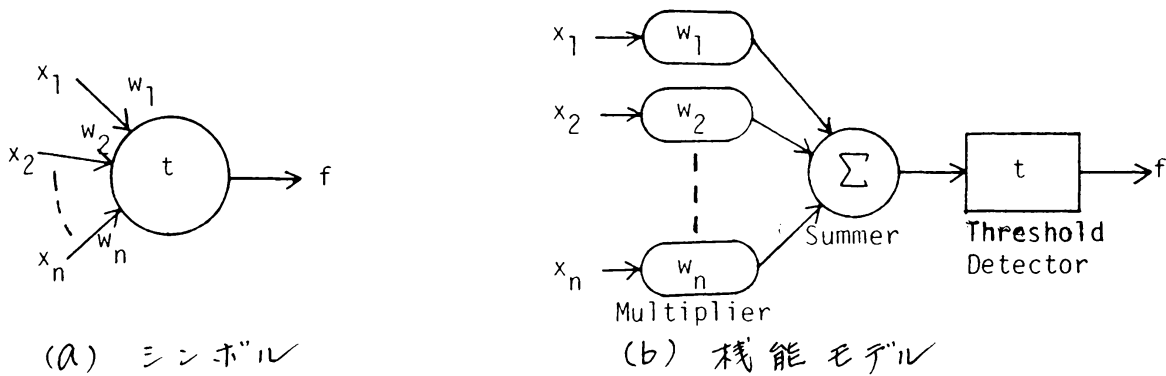


図 2. しきい素子のシンボルと機能モデル

能を有する素子と言えらる。このようにしきい素子を機能毎に分けたモデルを機能モデルと呼び、図 2 のように示す。電子回路を用いたしきい素子の回路実現においては、重み付け、加算及びしきい値判定はそれぞれ独立して構成することが可能である。

論理回路は、その入出力信号が、電圧、電流又は電荷のどの形で伝達されるかに従い、それぞれ電圧動作形、電流動作形、電荷動作形に分類される。電圧動作形回路としては、T²L や種々の MOS 回路があり、電流動作形回路には ECL や I²L が有名である。又 CCD は文字通り電荷動作形である。

しきい素子の機能モデルの観点に立てば、電圧動作形回路は、重み付け及び加算の機能をもたせることが困難である。これに対し、電流動作形回路では出力線を 1 本に接続すると、出力線中の信号の和を取り出すことが出来る。この作用を

WIRED-SUM と呼ぶことにする。これは電圧動作形回路における WIRED-OR や WIRED-AND に相当するものである。電荷動作形回路も電流動作形と同じ機能を持ち、異なった出力に含まれる電荷量の総和を得ることが出来る。

この WIRED-SUM は重み付けや加算の機能に都合のよい作用ではあるが、一方ファンアウトの減少をもたらす結果となる。すなわち、一旦 WIRED-SUM に使用された出力信号は他の入力信号として使用することができない。ファンアウトを増やすには、信号を複製する機能をもつ回路が必要となる。この信号複製回路は又、出力信号の整数倍の重み付けに利用することが出来る。

3. 多値多しきい値関数によるユニナリ関数の表現

いま、 r 値変数 x のユニナリ関数を $X(x)$ で示すと、 $X(x)$ は次のように定義できる。

$$X(x) = r_x \quad (3)$$

但し、 $X, x, r_x \in L'$, $x = 0, 1, \dots, r-1$, $L' \in \{0, 1, \dots, r-1\}$.

さて、変数 x の隣り合った値、 j と $j+1$ を考えると、一般性を失うことなく次式が成り立つ。

$$r_j = r_{j+1}$$

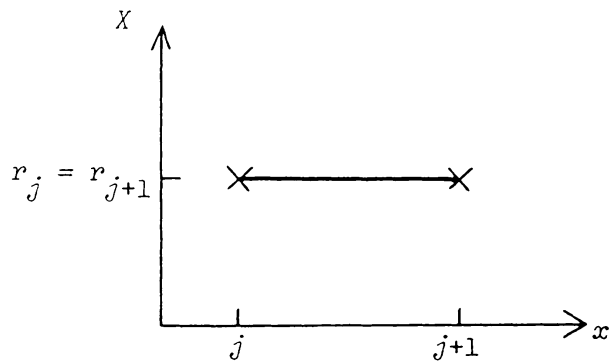
$$X(x) = r_j (= r_{j+1}) \quad (j \leq x \leq j+1)$$

$$r_j \neq r_{j+1}$$

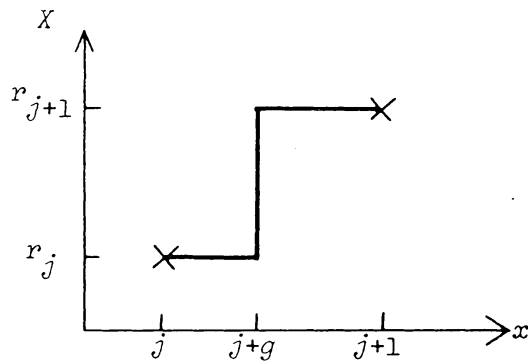
$$X(x) = \begin{cases} r_j & (j \leq x < j+g) \\ r_{j+1} & (j+g \leq x < j+1) \end{cases} \quad (4)$$

式(4)において g は $0 < g < 1$ なる値を適当に取ることが可能だが、ここでは便宜上 $g = 1/2$ と置く。

図3は x と X の関係を図示したものである。図において X は式(3)で定義された X の実を示し、実線は式(4)で定義された X の曲線(直線又は折線)を示している。この表現を



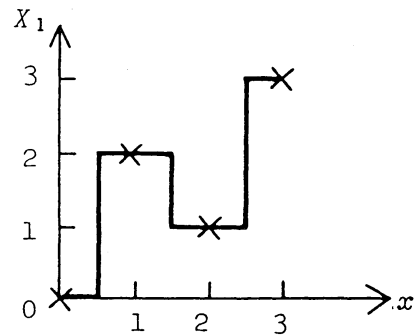
(a) $r_j = r_{j+1}$



(b) $r_j \neq r_{j+1}$

図3. ユーナリ関数 X の図的表現

x	X_1
0	0
1	2
2	1
3	3



(a) 真理値表

(b) 図的表現

図4. 4値変数ユーナリ関数の1例

用いると、例えば図4(a)の真理値表で示される4値変数のユーナリ関数 $X_1(x)$ は図4(b)のように図示できる。図1と図4を比較することにより、任意のユーナリ関数が多値多しきい値関数でもって表現されることわかる。すなわち、ユーナリ関数における変数 x を、多値多しきい値関数の変数 e に対応づければ、両者は等しいと言える。

ユーナリ関数を多値多しきい値関数に対応づけた場合、論理値は L の部分集合から成り、しきい値数は高々 $(r-1)$ となる。多値多しきい値関数は従来のしきい素子から成る回路網によって実現可能であるから⁽⁸⁾、任意のユーナリ関数もしきい素子を用いて実現可能といえる。

4. ユーナリ関数の分類

ユーナリ関数は対応する m 値 k しきい値関数の論理値の教 m としきい値の教 k とによって分類できる。 r 値変数のユーナリ関数に対し, m と k と r の間には次の関係が存在する。

$$m-1 \leq k \leq r-1 \quad (5)$$

これらの数はユーナリ関数を実現する回路の複雑さに影響を与える。 r が与えられれば, m と k の値の上限が式 (5) から定まり, 又, m と k の特定組合せに対し, ユーナリ関数が何個存在するかを求めることができる。表 1 は 4 値変数に対するユーナリ関数の分類を示す。

ユーナリ関数のうちで最も重要な関数は 2 値ユーナリ関数

表 1. 4 値変数ユーナリ関数の分類

m : ユーナリ関数の論理値教
 k : " のしきい値教
 n : このクラスに含まれる関数の個数

m	k	n	example functions
4	3	24	(0,1,2,3), (0,2,1,3), (3,1,2,0)
3	3	72	(0,1,0,2), (2,1,2,3), (0,2,0,3)
	2	72	(0,0,1,2), (2,1,1,3), (0,2,3,3)
2	3	12	(0,1,0,1), (2,1,2,1), (3,0,3,0)
	2	36	(0,0,1,0), (2,3,2,2), (1,3,3,1)
	1	36	(0,0,1,1), (2,2,2,3), (1,3,3,3)
1	0	4	(0,0,0,0), (1,1,1,1), (2,2,2,2)

($m = 2$ であり, $r = 2$ ではない) である。Post の単調ユーナリ関数とその相補関数は 2 値単しきい値関数に対応し, Post のディスジョイント関数及び Allen と Givone によって定義されたリテラルは, 2 値 2 しきい値関数に対応する⁽⁹⁾。ここで注意すべきことは, 2 値ユーナリ関数の 2 つの値は, 0 から $(r-1)$ までの任意の整数の中から選ばれるという点である。しかしながら, 2 値ユーナリ関数 f は, 通常の論理値 0 と 1 をもつ 2 値ユーナリ関数 f^b によって次のように書き表わされる。

$$f = (r_1 - r_2) \times f^b + r_2 \quad (6)$$

但し, r_1, r_2 は f の 2 つの論理値, $r_1, r_2 \in L$, $-$, \times , $+$ は通常の代数のシンボル。

他の重要なユーナリ関数に, 階段状リテラル (staircase literal) とサイクリング関数がある。階段状リテラルは, Allen と Givone によって $X(a+b)$ として定義されているが⁽⁹⁾, これは $(b-a)$ 値 $(b-a)$ -しきい値関数に対応する。又 r 値変数に対するサイクリング関数は $(r-1)$ 値 $(r-1)$ -しきい値関数に対応する。

5. ユーナリ関数の回路実現

5.1 電圧動作形回路を用いた 2 値ユーナリ関数の実現

2. で述べたように、電圧動作形回路では加算の機能をもたせることが困難であるため、しきい素子を用いるこの方法では一般的にユーナリ関数を実現することはできない。しかし、ここで例外として2値の多しきい値関数すなわち2値のユーナリ関数を実現する方法について示す。

いま、2値の多しきい値関数 $f_k^b(e)$ は次のように多しきい値関数 f_1, f_2, \dots, f_k によって分解される⁽¹⁰⁾。

$$f_k^b(e) = \begin{cases} f_1 \cdot f_2' \vee f_3 \cdot f_4' \vee \dots \vee f_{k-1} \cdot f_k' & (k: \text{偶数}) \\ f_1 \cdot f_2' \vee f_3 \cdot f_4' \vee \dots \vee f_{k-2} \cdot f_{k-1}' \vee f_k & (k: \text{奇数}) \end{cases} \quad (17)$$

但し、 f_k^b はしきい値 t_1, t_2, \dots, t_k を持ち、 f_i のしきい値は t_i ($t_i < t_{i+1}$) とする。又、 f' は f の相補関数、 $e < t_1$ に対し、 $f_k^b = 0$ とする。

このことは、2値の多しきい値関数は、しきい値判定器と、OR や AND や否定といったブール・ゲートとを用いて構成できることを意味する。

ここでは、Russell によって発表された多レベルNMOS回路を取り上げ⁽¹¹⁾、上述の方法によって2値ユーナリ関数を実現する回路を示す。図5はしきい値判定回路であり、しきい値が0.5, 1.5, 2.5の3つの場合について示されている。

しきい値を3.5以上に設定することは理論上可能であるが、実用上は無理なようである。図6はOR, AND, 否定の各ブ

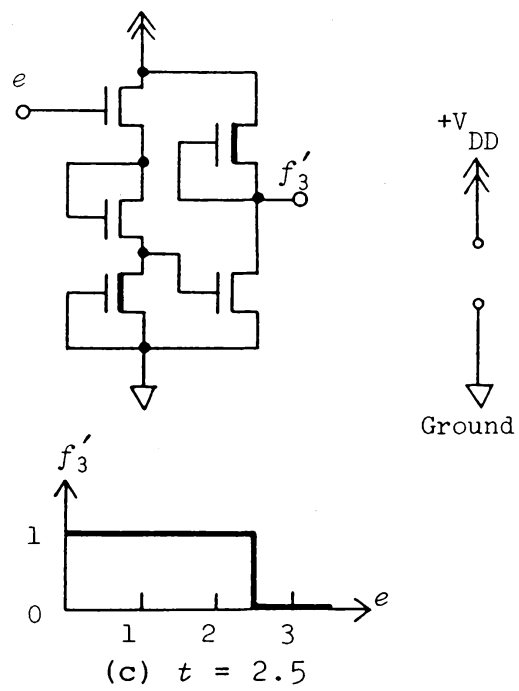
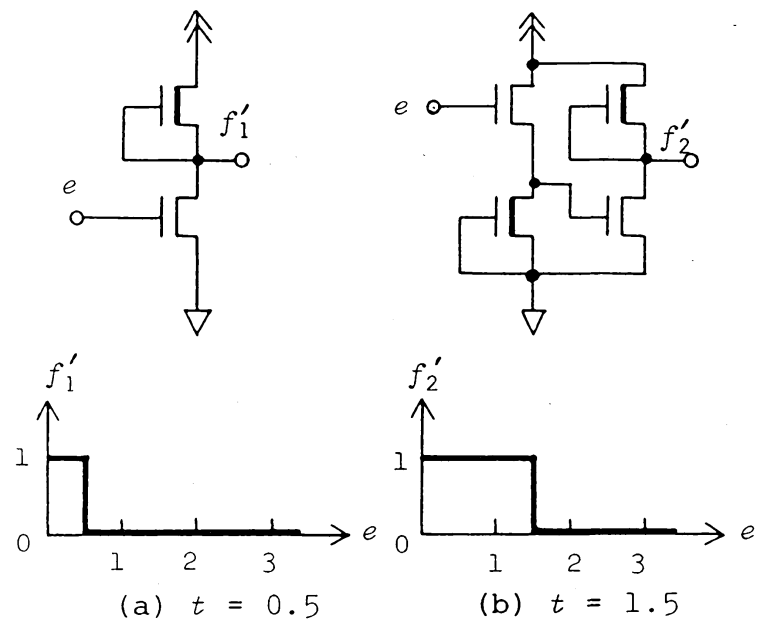


図5. NMOS多値論理回路のしきい値判定回路とその特性

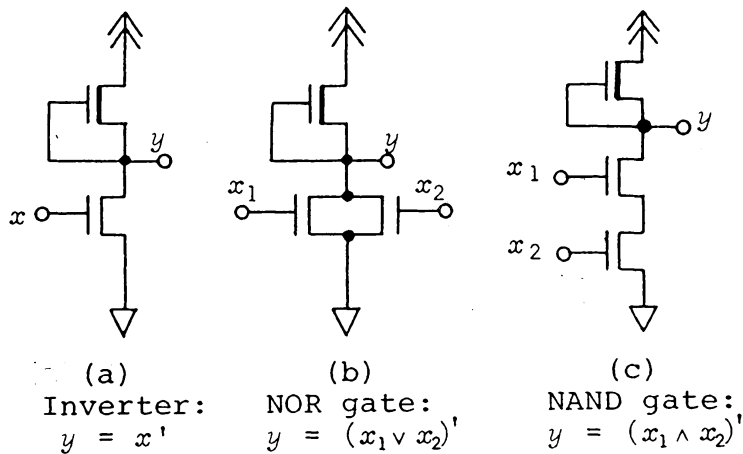
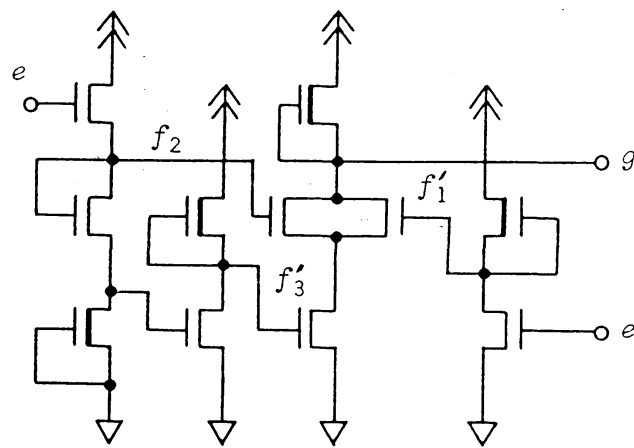


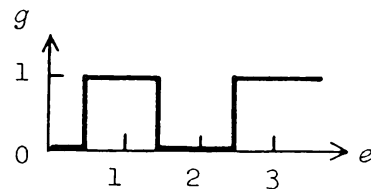
図6. NMOS多値論理回路のブール・ゲート

-ルゲートである。この多レベルNMOS回路を用いた3値変数
のリテラル回路、すなわち2値より多い値回路はMcCloskey
によって提案されている⁽⁶⁾。

式(7)からわかるように r 値変数に対する2値ユータリ関
数を実現するためには、 $(r-1.5)$ のしきい値をもつしきい値
判定器が必要となる。前述のように、このNMOS回路では、
実用上、しきい値が2.5に制限されているため、4値変数ま
での任意の2値ユータリ関数を実現できることになる。図7
は2値より多い値回路を示しており、 $x = (0, 1, 2, 3)$ に対
し、 $X = (0, 1, 0, 1)$ となるユータリ関数を実現される。
図中、 f_1 、 f_3 は、図6(a)、(c)に同じ、 f_2 は図6(b)の
Inverterを取り除いたものである。これらの出力は中央の
NOR-NANDゲートに導かれ、 $y = f_1 \cdot f_2' \vee f_3$ によって図の



(a) 回路図



(b) 入出力特性

図 7.3 3しきい値関数を実現するNMOS多値論理回路

3しきい値関数を実現する。この方法を2しきい値関数に適用すると、McCluskeyの提案したliteral回路と一致する。

5.2 電流動作形回路

電流動作形回路では、2.2で述べたように加算の機能はWIRED-SUMで実行されるので、しきい素子としてしきい値判定器と信号複製回路が必要となる。これらの回路を用いれば、多値多しきい値関数、引いてはユーナリ関数を実現できる。DruzetaらはECL回路を用いた多値多しきい値関数の実現を示した⁽¹²⁾。又著者はI²L回路に基づく多値多しきい値

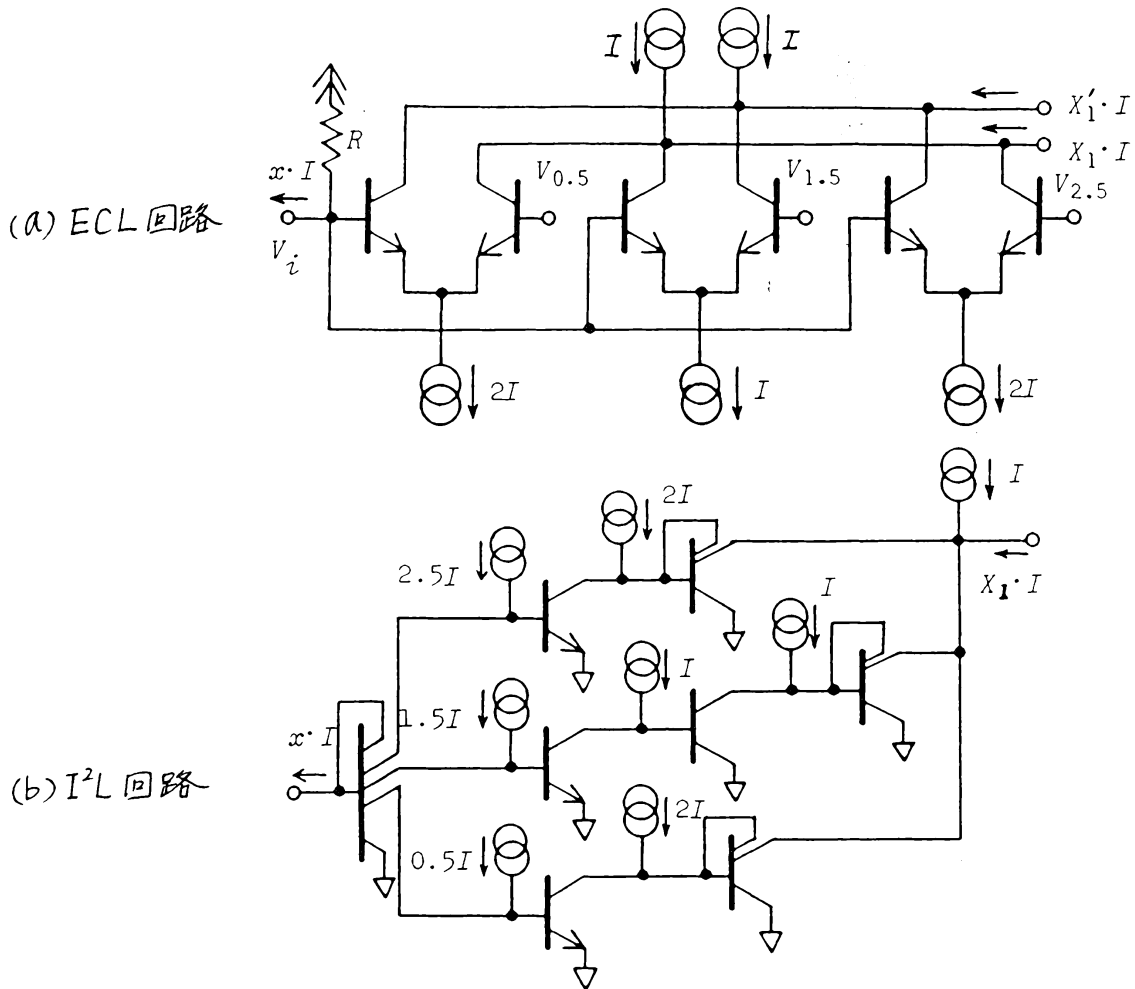


図8. 図4のユーナリ関数を実現するECL及び I^2L 回路

回路網を提示した⁽⁷⁾. 各回路でのしきい値判定器及び複製回路は、これらの文献に示されている。

例として、図4で取り上げたユーナリ関数を実現するECL及び I^2L 回路を図8に示す。これらの回路動作も上記の文献で明らかにされている。

5.3 電荷動作形回路(CCD回路)

KerkhoffらはCCD回路を用いた新しい多値論理回路の南

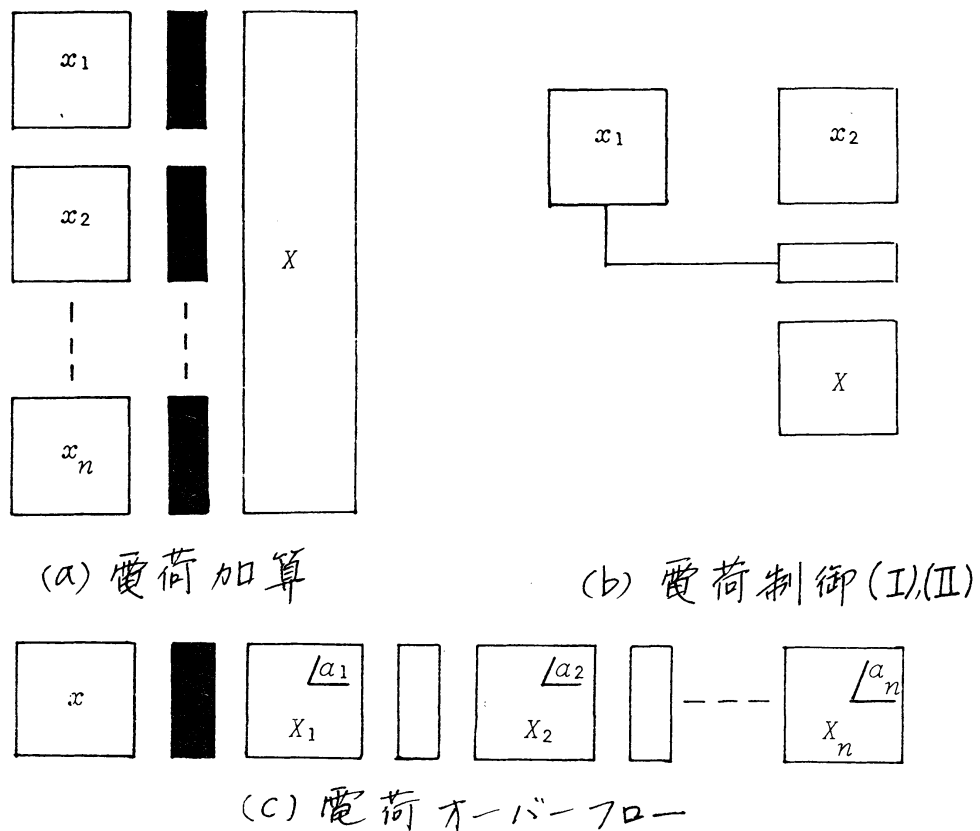


図9. CCD多値論理回路の4種の論理機能*

表に取り組んでいる⁽¹³⁾. このCCD多値論理回路では、図9*
 に示す4種の機能(電荷加算, 2種の電荷制御及び電荷オ
 ーバーフロー)を実行する3つの異なる構造が示されて
 いる. これらの機能は次のように表現できる.

1) 電荷加算:

$$X = x_1 + x_2 + \cdots + x_n \quad (8)$$

2) 電荷制御I:

* CCDの回路モデル及びその表記法については文献(13)に従う.
 用語等についても文献(13)を参照のこと.

$$X = \begin{cases} x_2 & (x_1 < t \text{ のとき}) \\ 0 & (\text{その他のとき}) \end{cases} \quad (9)$$

3) 電荷制御Ⅱ:

$$X = x_1 \quad (x_2 = \text{const.}) \quad (10)$$

4) 電荷オーバーフロー:

$$X_i = \begin{cases} 0 & (x < t_{i-1}) \\ x - t_{i-1} & (t_{i-1} \leq x < t_i) \\ a_i & (t_i \leq x) \end{cases} \quad (11)$$

但し, x 又は x_i は入力変数, X は出力関数, t 又は t_i はしきい値, a_i は各ストレージ・ウェル (Storage Well) の最大電荷量, $a_i = t_i - t_{i-1}$.

電荷加算は文字通り, 加算の機能, 電荷制御Ⅱは信号複製の機能として, $x_1 = x_2$ とするとき電荷制御Ⅰはしきい値判定の機能をもつため, CCDはしきい値素子として使用することができる。しかし, この回路は電荷制御Ⅰ ($x_1 \neq x_2$) と電荷オーバーフローのために, 他の回路素子には見られなかった新しい機能を有している。式(9)及び式(11)の入出力特性を図示すると, 図10となる。図10(a)は一見単なるしきい値判定の図に見えるが, x_2 が x_1 の関数又は x_2 , x_1 が共に入力変数 x の関数と成る場合, X には新たな関数が発生する。又図10(b)は, しきい値判定と信号複製が混合された機能といえる。

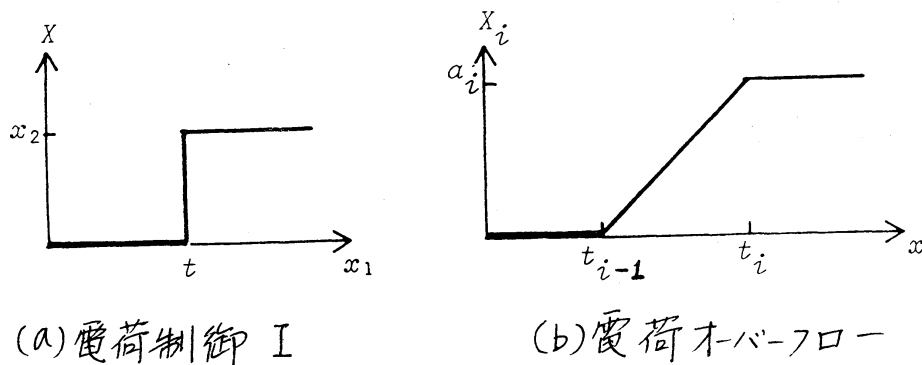


図 10. 電荷制御 I と電荷オーバーフローの入出力特性

CCD回路を用いたユーナリ関数の実現を4値変数の場合について考える。まず、電荷オーバーフローを用いて入力変数 x を変換する。このとき、入力変数の値は高々3であるから、式(11)において、 a_i は次の3つの組合せをとる。

$$(a) a_1 = 1, a_2 = 1, a_3 = 1$$

$$(b) a_1 = 1, a_2 = 2$$

$$(c) a_1 = 2, a_2 = 1$$

これらの a_i の組合せを式(11)に代入すると、

$$(a) X_1 = (0, 1, 1, 1), X_2 = (0, 0, 1, 1), X_3 = (0, 0, 0, 1)$$

$$(b) X_1 = (0, 1, 1, 1), X_2 = (0, 0, 1, 2)$$

$$(c) X_1 = (0, 1, 2, 2), X_2 = (0, 0, 0, 1)$$

が得られる。但し、出力関数 X_i は、入力変数 $x = (0, 1, 2, 3)$ に対する関数値を要素とするベクトルで表示している。

次に、これらの出力を電荷制御 I の2つの入力に加えると

各々の組合せに対し, 表2に示す出力が得られる. 但し簡単のため, 表中ベクトルの()は省いている.

表2において, $(0, 0, 0, 1)$, $(0, 0, 1, 0)$, $(0, 1, 0, 0)$ 及び $(1, 0, 0, 0)$ の出力が得られていることがわかる. このことは電荷制御IIの信号複製と電荷加算を用いて任意の関数が実現できることを意味する. 無論, 他の出力をうまく利用し

表2. 電荷制御Iによって得られる
4値変数ユナリ関数

(a) $a_1 = 1, a_2 = 1, a_3 = 1$				
$x_2 \backslash x_1$	0 1 1 1	0 0 1 1	0 0 0 1	1 1 1 1
0 1 1 1	---	---	---	1 0 0 0
0 0 1 1	0 1 0 0	---	---	1 1 0 0
0 0 0 1	0 1 1 0	0 0 1 0	---	1 1 1 0
(b) $a_1 = 1, a_2 = 2$				
$x_2 \backslash x_1$	0 1 1 1	0 0 1 2	1 1 1 1	
0 1 1 1	---	---	1 0 0 0	
0 0 1 2	0 1 0 0	---	1 1 0 0	
(c) $a_1 = 2, a_2 = 1$				
$x_2 \backslash x_1$	0 1 2 2	0 0 0 1	1 1 1 1	
0 1 2 2	---	---	1 0 0 0	
0 0 0 1	0 1 2 0	---	1 1 1 0	

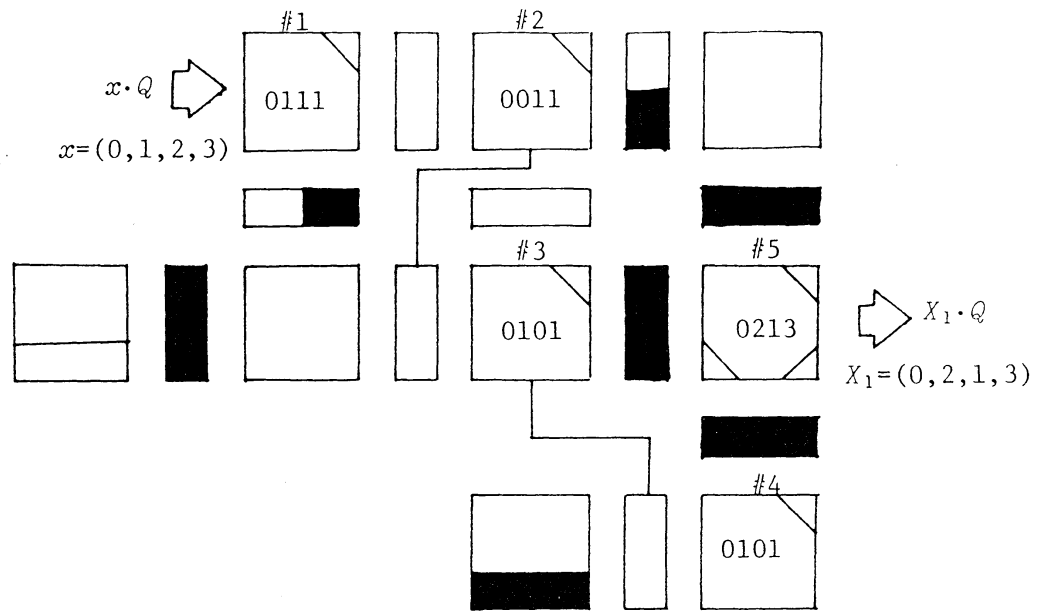


図 11. 図 4 のユニナリ関数を実現する CCD 回路

てより簡単な構成で、目的のユニナリ関数を実現できる。その 1 例を次に示す。

(例) 図 4 に示したユニナリ関数 $X_1(x)$ を CCD 回路によって実現する。

図 11 に示す CCD 回路を考へる。入力変換 (a) によって、#1 ~ #3 のウェルには $(0, 1, 1, 1)$, $(0, 0, 1, 1)$, $(0, 0, 0, 1)$ の出力が得られる。 $(0, 1, 1, 1)$ と $(0, 0, 1, 1)$ を電荷制御 I を適用し (表 2 参照), その出力 $(0, 1, 0, 0)$ を #3 のウェルに加算する。従って #2 のウェルに $X_{11} = (0, 0, 1, 1)$ が、#3 のウェルに $X_{12} = (0, 1, 0, 1)$ が得られる。更に電荷制御 II を用

いて、#4のウェルに#3と同じ出力 $X_{13} = (0, 1, 0, 1)$ を得る。これらを#5のウェルに加算すると、

$$X_1 = X_{11} + X_{12} + X_{13}$$

より、 $X_1 = (0, 2, 1, 3)$ が得られる。

6. むすび

本論文では、多値多しきい値関数とユーナリ関数との類似性に着目し、多値多しきい値関数の実現方法に基づいて任意のユーナリ関数を実現する方法について述べた。従って、新しい回路又は回路網の提案とけなっていないが、これまでに発表された多値論理ICは勿論のこと、これから開発されるであろう回路に対しても、この方法を容易に適用することができ、ユーナリ関数を実現する回路を生み出すことが可能となる。

しきい素子を実現する立場に立つと、回路が電圧動作形か電流動作形（電荷動作形を含め）かによって、実現方法が異なってくる。ここでは、電圧動作形回路に加算の機能を持たせることが困難なことから、2値のユーナリ関数の実現について示したが、何らかの方法で、この点を解決できれば、任意のユーナリ関数を実現することは、容易である。

これまでに提案されたユーナリ回路、リテラル回路をモナ

10化すれば、ほとんどがこの論文に述べた構成論に含まれる。今後、これとは異なった構成論の可能性について研究を続ける予定である。

本研究は、著者が文部省在外研究員として、米国スタンフォード大に留学中に行ったものである。有益な助言を戴いた同大 McCluskey 教授を始め、関係者各位に深く感謝の意を表します。

文 献

1. D. C. Rine(ed.), "Computer science and multi-valued logic theory and applications", (North-Holland, NY, 1977).
2. H. Dunderdale, "Current-mode circuits for the unary functions of a ternary variable", Electronics Letters, 6, 1 (Jan. 1970).
3. A. Lloris, A. Prieto and J. Velasco, "C.M.O.S. circuit for implementation of unary operators in ternary logic", Electronics Letters, 16, 5 (Feb. 1980).
4. J. H. Pugsley and C. B. Silio, Jr., "Some I^2L circuits for multiple-valued circuits", Proc. 8th Int. Symp. on Multiple-Valued Logic (May 1978).
5. J. G. Tront and D. D. Givone, "Multiple-valued logic gates using MESFET's", Proc. 9th Int. Symp. on Multiple-Valued Logic (May 1979).

6. E. J. McCluskey, "Logic design of MOS ternary logic", Proc. 10th Int. Symp. on Multiple-Valued Logic (May 1980).
7. O. Ishizuka, "Synthesis of multivalued multithreshold networks applying I^2L circuits", Proc. 9th Int. Symp. on Multiple-Valued Logic (May 1979).
8. 石塚, "多値多しきい値回路網の構成", 信学論(D), J60-D, 6 (昭52-06).
9. C. M. Allen and D. D. Givone, "Minimization technique for multiple-valued logic systems", IEEE Trans. C., C-17, 2 (Feb. 1968).
10. D. R. Haring, "Multi-threshold threshold elements", IEEE Trans. C., EC-15, 1 (Feb. 1966).
11. L. K. Russel, "Multilevel NMOS circuits", Compton Spring '81 (Feb. 1981).
12. A. Druzeta, Z. G. Vranesic, and A. S. Sedra, "Application of multi-threshold elements in the realization of many-valued logic networks", IEEE Trans. C., C-23, 11 (Nov. 1974).
13. H. G. Kerkhoff and M. L. Tervoert, "The implementation of multiple-valued functions using charge-coupled devices", Proc. 10th Int. Symp. on Multiple-Valued Logic (May 1980).