愛知工業大学研究報告
第33号B　平成10年

(p,q) 論理を応用した10値全加算器の一構成法

Realization of 10 Valued Full Adder utilizing the (p,q)-Logic

長谷川 忠光 †,　　羽賀 隆洋 ‡

Tadamitsu HASEGAWA,　Takahiro HAGA

Abstract　In this paper, it is proposed that the p-valued full adder is realized by the (p,q)-adic·min·max scheme which is rather low cost. It is also shown, However, that the circuit cost varies largely with the numbering of the vertices of full adder whitch is p-valued logical function.

1　はじめに

多値論理の内でもしきい論理が注目されている．しきい論理は，しきい値関数の族が従来の論理回路構成の基本素子であるAND，OR，NOTを特別な場合として含んでいる，論理万能系をなしている，など，強力な論理機能を特徴としている．その特徴をいかし，論理回路実現，神経回路網，パーセプトロン等の自己組織系の素子として，パターン認識における線形判定関数，画像などの多レベル入力の処理，等に，しきい論理は研究，応用されている．しかし，しきい素子の多機能を追求した場合，一般に出力においてとり得る論理値の種類より1だけ少ない個数のしきい値を用意する必要がある．そのしきい値の個数の増大とともに，対応する素子の実現が困難となる．また，費用が増大してしまう問題点が生じてしまう．

そこで，現在の技術でも有望とされる (p,q) 論理を用いて，任意の p 値論理関数を設計する事を考える．

(p,q) 論理において，中心に位置付けられているのが $(p,q)-adic$ 素子であり，その性質は良く知られている．

$(p,q)-adic$ 素子を用いた任意の p 値論理関数の一設計方法として，$(p,q)-adic\cdot min\cdot max$ スキームがある．(p,q) 論理は頂点番号を定める必要があり，その定め方によって費用の違いが生じてしまう事がわかった．10値全加算器を $(p,q)-adic\cdot min\cdot max$ スキームを用いて設計し，その頂点番号の定め方により費用の比較した．

†愛知工業大学大学院 学生 (豊田市)
‡愛知工業大学 情報通信工学科 (豊田市)

2　(p,q) 論理とは

いま入力に対応する p 個の論理値の集合を

$$L=\{0,1,\ldots,p-1\} \tag{1}$$

と定義し，出力値に対応する q 個の論理値の集合を

$$J=\{i_1,i_2,\ldots,i_q\}\subseteq L \tag{2}$$

とする．ここで $2\leq q\leq p$，$3\leq p$，$0\leq i_1<i_2<\ldots<i_q\leq p-1$ である．

しきい関数は，多値の場合へ拡張しやすいと考えられるが，出力においてとり得る論理値の種類より1だけ少ない個数のしきい値を用意する必要があり，その値の増大とともに対応する素子の実現は困難となる．このような性質を考慮して (p,q) 論理が考えられている[1]．したがって，興味のあるのは q が小さい場合であるが，論理としては一般に $2\leq q\leq p$ として考察する．

(p,q) 論理関数 f とは，

$$f:L^n\xrightarrow{into}J \quad for\ some\ J \tag{3}$$

n入力1出力の写像を意味している．

このように，(p,q) 論理関数といえども，特別な p 値論理関数にすぎないので，p 値論理の性質の多くがそのまま (p,q) 論理に引き継がれる．

(p,q) しきい関数とは，式4を満たす重みベクトル $\boldsymbol{a}=(a_1,a_2,\cdots,a_n;t_1,t_2,\cdots,t_{q-1})$ が存在するような関数である。

$$\begin{cases} f(\boldsymbol{x}) = i_q & iff \quad w(\boldsymbol{x}) > t_{q-1} \\ f(\boldsymbol{x}) = i_{q-1} & iff \quad t_{q-1} > w(\boldsymbol{x}) > t_{q-2} \\ \qquad\qquad \vdots & \\ f(\boldsymbol{x}) = i_2 & iff \quad t_2 > w(\boldsymbol{x}) > t_1 \\ f(\boldsymbol{x}) = i_1 & iff \quad t_1 > w(\boldsymbol{x}) \end{cases} \tag{4}$$
$$(t_1 < \ldots < t_{q-1})$$

ここに $w(\boldsymbol{x}) = a_1 \cdot x_1 + \ldots + a_n \cdot x_n$ とする．
ここでいくつかの定義を述べる．

定義 1 p 値NOT を以下で定める。

$$\bar{x} = p - x - 1 \tag{5}$$
$$(x \in L)$$

定義 2 頂点番号 ρ を

$$\rho = (x_n, \cdots, x_1)_p \tag{6}$$

で定める．このとき，

$$f(\rho_1) \geq f(\rho_2) \tag{7}$$
$$for \ \ any \ \ pair \ \ \rho_1 > \rho_2$$

である (p,q) 論理関数 f を 代表 $(p,q)-adic$ 関数と呼ぶ．そして，代表 $(p,q)-adic$ 関数から，変数の置換，変数の否定の操作により得られる関数 (同族関数) を単に，$(p,q)-adic$ 関数という．

$(p,q)-adic$ 関数はしきい関数である．

定理 1 (p,q) 論理関数 $f(x_1, x_2, \cdots, x_n)$ は，2 変数 $(p,q)-adic$ 関数 ψ を用いて，

$$f = \psi^{(n-1)}((\cdots\psi^{(2)}(\psi^{(1)}(x_1, x_2), x_3), \cdots), x_n) \tag{8}$$

と、あらわされるとき、及び、そのときに限り、$(p,q)-adic$ 関数である。また、2 変数代表 $(p,q)-adic$ 関数 ψ によって、上記のように表されるとき、及びその時に限り f は代表 $(p,q)-adic$ 関数である。

この定理は，n 入力 $(p,q)-adic$ 素子は，2 変数 $(p,q)-adic$ 素子のカスケード接続によって実現可能であることを意味している．

3 $(p,q)-adic \cdot min \cdot max$ スキームとは

p 値論理回路の一構成法として羽賀 [2] の提案した $(p,q)-adic \cdot min \cdot max$ スキームがある．$(p,q)-adic \cdot min \cdot max$ スキームにおいて，min 素子，max 素子は，p 値論理素子の内でも低費用である事に注意する．

実現すべき任意の関数 p 値論理関数 $f(x_1, x_2, \cdots, x_n)$ が与えられたとする．式 6の定義により p 値論理関数 f に，頂点番号 ρ を定める．$(p,q)-adic$ 素子は式 7により広域単調増加であるため，2 つの $(p,q)-adic$ 素子の出力を min 素子に入力し，p 値論理関数 f の一部を実現する事ができる．これを複数用意し，各 min 素子の出力を max 素子に入力する事により，p 値論理関数 f 全体を実現する．これらを考慮すると，全体の構成は図 1 の様になる．ここで L_{ij} は，$(p,q)-adic$ 素子である．

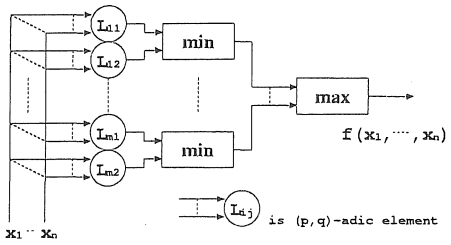

図 1: $(p,q)-adic \cdot min \cdot max$ スキームの構成

実現すべき任意の関数 p 値論理関数 $f(x_1, x_2, \cdots, x_n)$ に頂点番号を定め，$(p,q)-adic$ 素子の出力を割り当て実現する．その割り当て方により，全体の構成は異なってくる．割り当て方の例を図 2に示す．方略 1($(1)q = 2$, $(2)q = 3$ の①) では，必ず任意の関数 p 値論理関数が実現できるが，方略 2($(2)q = 3$ の②) の場合の様に設計する事によりより少ない構成での実現の可能性がある．

3.1 方略 1 を用いた場合

実現すべき任意の関数 p 値論理関数に頂点番号 $\rho = (x_n, x_{n-1}, \cdots, x_1)_p$ を定める．ある狭義単調増加域 ρ_1 から ρ_n のみを考える．すなわち頂点番号 ρ_1 から ρ_n まで

$$f(\rho_1) < \cdots < f(\rho_n) > f(\rho_{n+1}) \tag{9}$$
$$但し \rho_1 < \rho_2 < \cdots < \rho_n$$

の時，頂点番号 ρ_1 から ρ_n までを実現するための，min 素子の個数 m は，

$$m = \left\lceil \frac{n}{q-1} \right\rceil \tag{10}$$

となる．ここで，$[x]$ は，シーリング $min(n)$ $n \geq x$ (n は整数) である．

また，頂点番号 ρ_1 から ρ_n までを実現するため，m 個の $(p,q)-adic$ 素子，m 個の $(p,2)-adic$ 素子，m 個の min 素子，1 個の max 素子を用いて実現が可能である．実現すべき関数の単調増加域，及び，単調減少域の個数を考慮すれば，容易に全体の構成，費用が求まるのがこの設計方法の利点である．

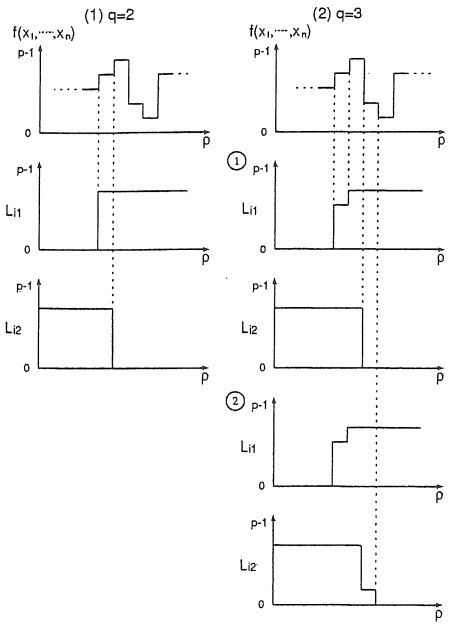

図 2: L_{ij} の選び方

3.2 方略 2 を用いた場合

同様にして実現すべき任意の関数 p 値論理関数に頂点番号 $\rho=(x_n,x_{n-1},\cdots,x_1)p$ を定める．頂点番号 ρ_1 から ρ_n までの単調増加のち単調減少 (もしくは，単調減少のち単調増加) 域を考える．

$$f(\rho_1)<\cdots<f(\rho_{k-1})<f(\rho_k)>f(\rho_{k+1})>\cdots>f(\rho_{n-1})>f(\rho_n) \quad (11)$$
但し$\rho_1<\rho_2<\cdots<\rho_k<\cdots<\rho_n$

頂点番号 ρ_1 から ρ_n までを実現するための，min 素子の個数 m は，

$$m=\left\lceil\frac{n}{q-1}\right\rceil \quad (12)$$

となる．

また，頂点番号 ρ_1 から ρ_n までを実現するため，m 個の min 素子と $2m$ 個の $(p,q)-adic$ 素子をもちいて表す事が出来る．$(p,q)-adic$ 素子の個数に関しては，一意的に定める事は出来ず，それぞれの場合において検討が必要である．

この方略 2 においては，図 2，(2)$q=3$ においては，②など，方略 1 に比べ，素子数が少なくなる場合がある．とくに単調増加のち単調減少が多い程，素子数が少なくなる場合がある．

4 10 値全加算器について

$(p,q)-adic\cdot min\cdot max$ スキームを用いて具体的に 10 値全加算器について設計する

4.1 10 値全加算器について

n 桁で表現される 2 数 $X=(x_{n-1}\cdots x_0)$，$Y=(y_{n-1}\cdots y_0)$ に対して，加算は 2 進と同様にして，以下の様に記述され，加算結果は $S=(s_{n-1}\cdots s_0)$ として得られるものとする [3].

$$x_i+y_i+c_{i-1}=c_ip+s_i \quad (13)$$

ここで，$i=0,\cdots,n-1$，$c_{-1}=0$ とし，また，$x_i,y_i,s_i\in\{0,1,\cdots,p-1\}$ である．

これを実現する一つのモジュールを全加算器と呼び入力を a，b，キャリ入力を c とし，$p=4$ の例を表 1 に示す．

サム出力					
		b			
c	a	0	1	2	3
0	0	0	1	2	3
	1	1	2	3	0
	2	2	3	0	1
	3	3	0	1	2
1	0	1	2	3	0
	1	2	3	0	1
	2	3	0	1	2
	3	0	1	2	3

キャリ出力					
		b			
c	a	0	1	2	3
0	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	1
	3	0	1	1	1
1	0	0	0	0	1
	1	0	0	1	1
	2	0	1	1	1
	3	1	1	1	1

表 1: 全加算器の例 ($p=4$，2 入力：a，b，キャリ入力：c)

4.2 費用の仮定について

$(p,q)-adic\cdot min\cdot max$ スキームを用いて最小費用の $(p,q)-adic\cdot min\cdot max$ 回路を与える最適な $q*$ を考える．その構成例を図 3に示す．定理 1 を考慮して，以下の仮定をする．

1. 2 入力 $(p,q)-adic$ 素子の費用は，$a(q)$.
2. 2 入力 min 素子の費用は，b.
3. m 入力 max 素子の費用は，$c\cdot(m-1)$.
4. 回路の総費用に関して，素子以外の費用は考慮しない．

ここで，n 入力 $(p,q)-adic$ 素子の費用は，式 8 定理 1 より，$(n-1)\cdot a(q)$ と，表す事ができ，$a(q)$ を

$$a(q)=a\cdot q^k(k\geq 1) \quad (14)$$

の場合を考える．この仮定は，指数ではなく，多項式で表されていることを考慮してあり，一般的に妥当である．

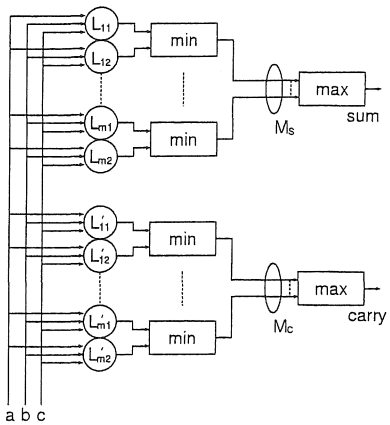

図 3: $(p,q)-adic\cdot min\cdot max$ スキームを用いた全加算器の構成

4.3 頂点番号の定め方に関して

$(p,q)-adic\cdot min\cdot max$ スキームを用いて任意の関数を実現する場合，頂点番号の定め方が問題になってくる．$(p,q)-adic\cdot min\cdot max$ スキームの方略 1 を用いて全加算器を設計し，頂点番号の定め方による費用の違いを示す．考えられる頂点番号の定め方は表 2の 2 種類を考える．$p=4$ の例を表 3に示す．

4.4 頂点番号の定め方 順番 1 の場合

頂点番号を表 2の “頂点番号の順番 1” の様に定める．サム出力を実現する max 素子への入力数を M_{S1}，キャリ出力を実現する max 素子への入力数を M_{C1} とすると，

$$M_{S1} = 4\left\lceil\frac{(p-1)}{(q-1)}\right\rceil+2(p-1)\left\lceil\frac{(p-2)}{(q-1)}\right\rceil \quad (15)$$

$$M_{C1} = 2(p-1) \quad (16)$$

となる．ここで，総費用 $C_1(p,q)$ を考えると，

$$\begin{aligned}C_1(p,q) = {} & c\cdot(M_{S1}-1)+b\cdot M_{S1}+\\ & 2\cdot a(q)\cdot M_{S1}+2\cdot a(2)\cdot M_{S1}+\\ & c\cdot(M_{C1}-1)+b\cdot M_{C1}+\\ & 2\cdot 2\cdot a(2)\cdot M_{C1}\end{aligned} \quad (17)$$

となる．

c	a	b: 0	⋯	$p-1$
	0	0	⋯	$p-1$
0	⋮	⋮	⋱	⋮
	$p-1$	$p\times(p-1)$	⋯	$p\times p-1$
	0	$p\times p$	⋯	$p\times(p+1)-1$
1	⋮	⋮	⋱	⋮
	$p-1$	$p\times(2p-1)$	⋯	$p\times 2p-1$

a. 頂点番号の定め方 (順番 1)

c	a	b: 0	⋯	$p-1$
	0	0	⋯	$p\times 2-2$
0	⋮	⋮	⋱	⋮
	$p-1$	$p\times(2p-2)$	⋯	$p\times 2p-2$
	0	1	⋯	$p\times 2-1$
1	⋮	⋮	⋱	⋮
	$p-1$	$p\times(2p-2)+1$	⋯	$p\times 2p-1$

b. 頂点番号の定め方 (順番 2)

表 2: 頂点番号の定め方 (2 入力：a，b，キャリ入力：c)

4.5 頂点番号の定め方 順番 2 の場合

同様にして頂点番号を “頂点番号の順番 2” の様に定める．サム出力を実現する max 素子への入力数を M_{S2}，キャリ出力を実現する max 素子への入力数を M_{C2} とすると，

$$M_{S2} = (p+1)\cdot\left\lceil\frac{(p-1)}{(q-1)}\right\rceil \quad (18)$$

$$M_{C2} = (p-1) \quad (19)$$

となる．ここで，総費用 $C_2(p,q)$ を考えると，

$$\begin{aligned}C_2(p,q) = {} & c\cdot(M_{S2}-1)+b\cdot M_{S2}+\\ & 2\cdot a(q)\cdot M_{S2}+2\cdot a(2)\cdot M_{S2}+\\ & c\cdot(M_{C2}-1)+b\cdot M_{C2}+\\ & 2\cdot 2\cdot a(2)\cdot M_{C2}\end{aligned} \quad (20)$$

となる．

4.6 計算結果

任意の条件のもとに実際にその費用を計算する．環和と同様に，max 素子，min 素子の費用は同じと仮定し，$a(2)$ との比を変化させ，$k=2,3$ の頂点番号の

c	a	b=0	b=1	b=2	b=3
0	0	0	1	2	3
	1	4	5	6	7
	2	8	9	10	11
	3	12	13	14	15
1	0	16	17	18	19
	1	20	21	22	23
	2	24	25	26	27
	3	28	29	30	31

a. 順番 1 の例

c	a	b=0	b=1	b=2	b=3
0	0	0	2	4	6
	1	8	10	12	14
	2	16	18	20	22
	3	24	26	28	30
1	0	1	3	5	7
	1	9	11	13	15
	2	17	19	21	23
	3	25	27	29	31

b. 順番 2 の例

表 3: 頂点番号の定め方の例 ($p = 4$, 2 入力 : a, b, キャリ入力 : c)

順番 1(Numbering1), 頂点番号の順番 2(Numbering2) の最小費用を図 4に示す. また, 参考までに, 最小費用を実現する $q*$ を図 5に示す.

4.7 検討

順番 1, 順番 2 の最小となる費用を比較すると, 図 4より, 順番 2 の方が全体の構成費用は低く実現できる事がわかる. これは, 順番 2 の方が, 単調増加の個数が少なく表現できる為である. その結果, 1 つの $(p,q)-adic$ 素子で任意の関数の多くの範囲を実現でき, 全体の費用は低く実現できる. max 素子 min 素子と $(p,q)-adic$ 素子の費用を比較し, $(p,q)-adic$ 素子がより高価な場合, その差は更に顕著に現れてくる.

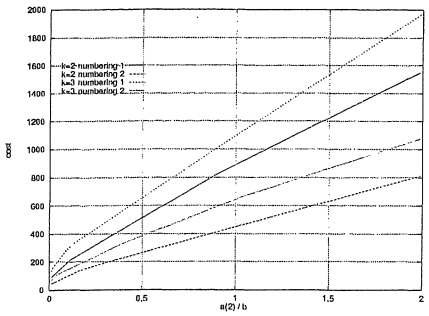

図 4: 全加算器の費用

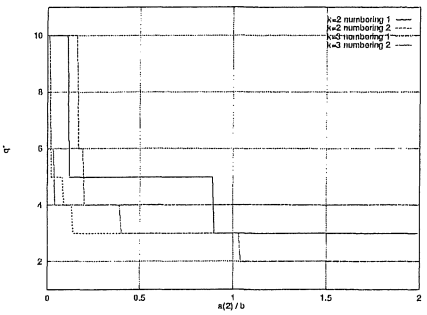

図 5: 全加算器の最適な費用の q*

5 まとめ

$(p,q)-adic \cdot min \cdot max$ スキームを用いて, 10 値全加算器を設計した. その結果, 頂点番号の定め方により全体の費用は異なる事がわかった. そのため頂点番号に関して十分検討し設計する必要がある.

参考文献

[1] 羽賀隆洋 : " しきい値論理関数に関する研究", 名古屋大学 学位論文, 1974.

[2] 羽賀隆洋 : " 多値 (p,q) 論理の提案と主要結果", 電子情報通信学会, 1992.

[3] 樋口龍雄, 亀山充隆 : " 多値情報処理", 昭晃堂, 1989.

(受理 平成10年 3 月20日)