6X-2

ベクトル計算機を用いた論理関数の素項の生成

加賀谷 達次　高木 直史　矢島 脩三
(京都大学工学部)

1．はじめに

近年、並列パイプライン処理方式のベクトル計算機が開発され、科学技術計算のみならず、論理シミュレーション等にも利用され始めている[1]。しかし、その最大性能を引き出すには種々の工夫が必要である。本稿では論理設計における基本的な問題である論理関数の素項の生成に対する、ベクトル計算機の利用について考える。

2．和積形の展開に基づく素項の生成法

論理関数Ｆの和積形表現を積和形表現に展開すれば、Ｆの関数の全ての素項が生成されることが明らかにされている。しかし、このとき非素項も生成されることが多い。Slagleの方法[2] や節展開法[3] では、木状に和積形の展開を行ないながら、展開する和項の選び方やリテラルの順序付けなどに工夫することによって、非素項の生成を少なくしている。

3．ベクトル計算機向き和積形の展開法

ベクトル計算機の性能を十分に引き出すには、ベクトル化率を高く、ベクトル長を長くする必要がある。しかし、従来の和積形の展開に基づく方法をそのままベクトル計算機で実行しても、ベクトル長が短いなどの理由からベクトル計算機の性能を十分には引き出せない。本稿のアルゴリズムでは、和積形の木状の展開の一段を変数ごとに一度に行うことにより、ベクトル長を長くし、ベクトル計算機の性能を十分に引き出せるようにしている。図１に節展開法と本稿のアルゴリズムのベクトル化できる部分の比較を示す。アルゴリズムを以下に示す。木の各節点には和項の集合と、木の根からその節点に至るまでの道に対応したリテラルの論理積をとった積項を対応させる。このアルゴリズムに基づく和積形の展開の例を図２に示す。例えば、図２の節点２のABを積項、$(\bar{C}+\bar{D})(\bar{C}+D)$を和項の集合と考える。

(1) 与えられた論理関数Ｆの和積形表現の和項の集合と積項として１を木の根に対応させ、非終端節点とする。非終端節点がなくなるまで(2),(3),(4)を繰り返す。

(2) 木の同じ深さにある全ての非終端節点N_i(i=1,…,m)から、それぞれ１つの和項を選び、それをS_iとする。全ての変数について順序をつけ、その順序に従って(3)を行なう。

(3) S_iがリテラルL_j（X_jまたは$\bar{X}_j$）を含めば、N_iからリテラルL_jに対応した枝出しを行ない、新しい節点N_{ij}を次のようにして作る。

(3-1) N_iの積項P_iとリテラルL_jの論理積P_{ij}をN_{ij}に対応した積項とする。

(3-2) N_iの和項の集合の中からリテラルL_jを含まない和項をN_{ij}の和項の集合に入れる。このとき、その和項にリテラル$\bar{L}_j$が含まれれば、それを除いたものを入れる。

(3-3) 元のN_iの和項の集合の中の和項からリテラルL_jを除く。

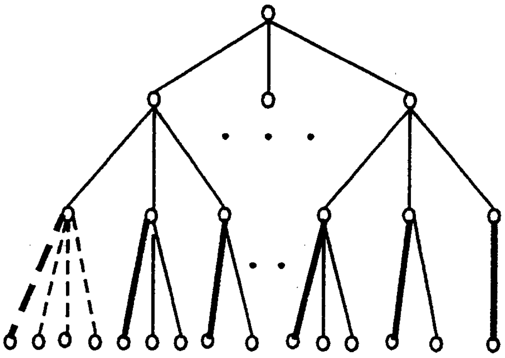
(4) N_{ij}の和項の集合が空ならば、N_{ij}を有効節点とし、リテラルが１つもない和項が存在すれば、N_{ij}を無効節点とする。有効節点でも無効節点でもない節点を非終端節点とする。

(5) 全ての有効節点の積項の中にＦの全ての素項が含まれているので、それらの包含関係を調べることにより、素項のみを抽出する。 ■

例えば、非終端節点(図２の節点1)の展開を説明する。節点1をBで展開する時は節点1の和項の集合{$(B+D)(B+\bar{C})(\bar{B}+\bar{C}+\bar{D})(\bar{C}+D)$}からBを含まない和項{$(\bar{B}+\bar{C}+\bar{D})(\bar{C}+D)$}を選び、それから$\bar{B}$を除いた和項{$(\bar{C}+\bar{D})(\bar{C}+D)$}を新しい節点2の和項の集合に入れる。次にDで展開する時は、元の和項の集合からBが除かれており、Dを含まない和項{$(\bar{C})(\bar{B}+\bar{C}+\bar{D})$}から$\bar{D}$を除いた和項{$(\bar{C})(\bar{B}+\bar{C})$}が新しい節点3の和項の集合となる。

各ステップのベクトル長を考えると、(3-1)は木

Generation of prime implicants of a logic function using a vector processor
Tatsuji KAGATANI, Naofumi TAKAGI, Shuzo YAJIMA
KYOTO University



従来の方法でベクトル化できる部分　　　　　… 点線の部分
本稿のアルゴリズムでベクトル化できる部分　… 太線の部分

図１　和積形の展開におけるベクトル化できる部分の比較

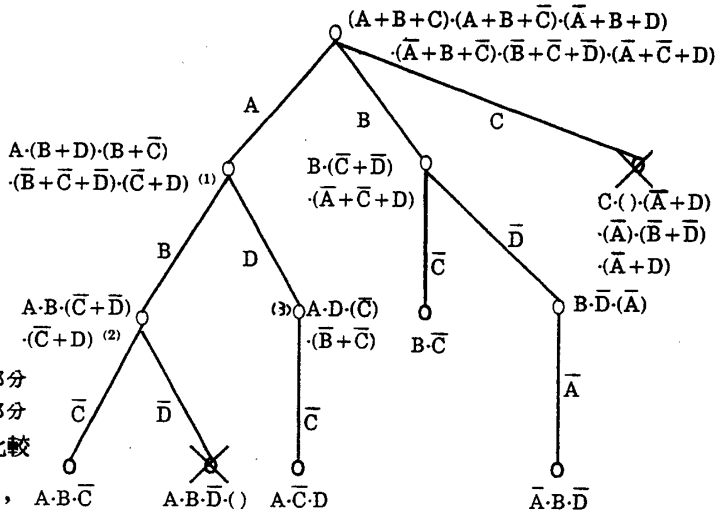


図２　和積形の展開の例

の同じ深さにある全ての非終端節点の数が、(3-2)，(3-3)は一度にそれらの非終端節点に含まれる和項の数の和がベクトル長になると考えられる。また、(4)は展開された節点の数やそれらに含まれる和項の総数がベクトル長となると考えられる。

４．評価および考察

３章のアルゴリズムに基づき、さらにベクトル化されるように工夫して作成したプログラムと節展開法のプログラムを京大大型計算機センタのVP-200とM-382で実行した。実験した論理関数は、onsetを乱数を用いて発生させた12変数関数200個とその否定の関数の計400個で、そのonsetを入力として与える。いくつかの関数と全関数の平均の計算時間を表１に、また、生成される非素項の数や、１段に含まれる節点や和項の数の最大値を表２に示す。

表１より、本稿のアルゴリズムをベクトル計算機で実行した場合、スカラ計算機で実行した場合に比べて、約4.4倍～17倍、平均で約9倍高速に処理できた。また、節展開法をスカラ計算機で実行した場合と比べると、平均で約37倍高速になった。節展開法では、ベクトル計算機で実行しても約1.4倍しか高速にならない。このことからも、本稿のアルゴリズムはベクトル計算機を利用すると高速に処理できることが示された。

表２より、非素項は本稿のアルゴリズムが節展開法の約1.5～2.5倍となっている。また、１段に含まれる節点や和項の数も本稿のアルゴリズムの方がそれぞれ、約1.5～3.5倍、約1.2～2.5倍であり、記憶量等の点では節展開法の方が優れているといえる。

５．おわりに

ベクトル計算機向きの論理関数の素項生成法を提案した。ベクトル計算機も工夫すれば、論理設計の種々の問題や組み合わせ問題に対しても有効であると考えられる。

表１　計算時間の比較　(ms)

()内は onsetの数	素項	本稿のｱﾙｺﾞﾘｽﾞﾑ		節展開法	
		M-382	VP-200	M-382	VP-200
f１(485)	7781	38168	2210	53737	23163
f２(2763)	1500	4198	958	39847	32605
f３(1317)	4966	13585	1282	53322	35084
f４(2072)	2788	6092	1028	46823	36105
400個の平均	3184	10688	1155	43700	31900

表２　記憶量等の比較

	非素項の数		１段の節点の数		１段の和項の数	
	本稿	節展開	本稿	節展開	本稿	節展開
f１	13431	5891	17612	5123	34753	17463
f２	792	626	2314	1378	94516	79521
f３	5394	3033	9753	3804	64452	42585
f４	2121	1544	4744	2415	79816	64334
平均	3766	2122	6249	2592	75727	60324

謝辞:日頃から御討論頂く矢島研究室の諸氏に感謝致します。

参考文献

[1] 石浦他:ベクトル計算機による高速論理シミュレーション,情処論文誌,Vol.27,No.5,pp.510-517

[2] Slagle,J.R.etc.:A new algorithm for generating prime implicants,IEEE Trans.Comp.,Vol.C-19,No.4,pp304-310.

[3] 上林他:節展開法を用いた論理関数の主項の生成,信学論(D),Vol.J62-D,No.2,pp89-96.