

CMOS/パス・トランジスタ論理の混在による 低消費電力回路の合成

高田 賢吾^{†,††} 神野 元彰[†] 黒木 修隆[†]
沼 昌宏[†] 瀧 和男[†] 山本 啓輔[†]

パス・トランジスタ論理は、CMOS 論理と比較して少ないトランジスタ数で論理回路を実現できる場合が多いが、いかなる回路に対しても有利とは限らない。(N)OR, (N)AND などの単純な関数については、CMOS 論理が有利となる場合が多い。そこで、論理関数の単純直交分解の結果を表す分解グラフをもとに、CMOS 論理をパス・トランジスタ論理と混在させることでトランジスタ数を削減し、低消費電力化を実現する手法を提案する。分解グラフをもとに、単純な関数については CMOS 論理を適用する一方で、パス・トランジスタ論理で構成する他の部分には BDD (Binary Decision Diagram) 分割を適用し、必要となる中間バッファ数を削減する。本手法による 40 例の MCNC ベンチマーク回路に対する実験の結果、面積最小化指向で合成した CMOS 回路や、SPL 回路に対して、それぞれ幾何平均で 27%、12%の低消費電力化を実現した。

Synthesis of Low Power Circuits Combining CMOS/Pass Transistor Logic

KENGO TAKATA,^{†,††} CHIKATERU JINNO,[†] NOBUTAKA KUROKI,[†]
MASAHIRO NUMA,[†] KAZUO TAKI[†] and KEISUKE YAMAMOTO[†]

Logic functions can be implemented with fewer transistors based on the pass transistor logic (PTL) than on the static CMOS logic in many cases, but not always. For simple functions such as (N)OR and (N)AND, static CMOS logic circuits can often be implemented with fewer transistors than PTL. In order to reduce transistor counts for lower power dissipation, we propose a method to synthesize logic circuits combining CMOS and PTL based on the decomposition graph obtained as the result of simple disjunctive decomposition for a given logic function. Based on the decomposition graph, we implement simple functions with CMOS logic. On the other hand, we implement the other functions with PTL based on sliced BDD (Binary Decision Diagram) to reduce buffer counts. On a set of 40 MCNC benchmarks, our method has synthesized circuits with lower power dissipation by 27% than area-oriented CMOS, and by 12% than SPL.

1. はじめに

近年の携帯機器の急速な普及や、LSI チップの発熱量の増加などから、低消費電力化を目的とするさまざまな研究が行われている^{1),2)}。そのなかで、おもにトランジスタ数を抑えて負荷容量を減らすことにより低消費電力化を実現する手法として、パス・トランジスタ論理^{3)~8)}が有効である。

パス・トランジスタ論理の中でも、SPL (Single-rail Pass Transistor Logic)⁷⁾は消費電力に重点をおいた論理構成方式であり、MCNC ベンチマーク回路のうち多くの回路例に対して、CMOS 論理回路よりも少ないトランジスタ数で実現され、低消費電力となる結果が得られている⁷⁾。しかし、すべての回路に対して CMOS 論理よりパス・トランジスタ論理が有利とは限らない。

表 1 に、CMOS, SPL それぞれの論理で構成した、NOR, XOR, MUX (マルチプレクサ) 回路のトランジスタ数の例を示す。なお、回路名の末尾は入力数を表し、NOR2 は 2 入力 NOR を意味する。また MUX には、末尾の入力数以外にセレクト入力が存在する。NOR 回路については、CMOS の方が SPL より少数

[†] 神戸大学
Kobe University

^{††} 日本学術振興会特別研究員
Research Fellow of the Japan Society for the Promotion of Science
現在、奈良先端科学技術大学院大学
Presently with Nara Institute of Science and Technology

表1 各種回路のトランジスタ数

Table 1 Transistor counts of various circuits.

回路	CMOS	SPL			
		入力インバータ	論理部分	出力インバータ	合計
NOR2	4	2	2	3	7
NOR3	6	4	4	3	11
XOR2	10	4	2	3	9
XOR3	20	6	6	3	15
MUX2	12	2	2	3	7
MUX3	20	4	4	3	11

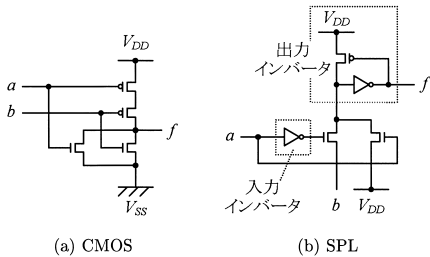


図1 NOR2回路例

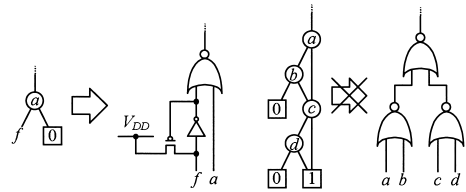
Fig. 1 NOR2 circuit examples.

のトランジスタで実現できる．NOR2回路をそれぞれの論理で構成した例を図1に示す．図1(b)のSPL回路に関して，入力インバータは入力変数の補を生成するために用いられる．出力インバータは，プルアップ・トランジスタによるレベル回復のために挿入される．NAND回路もNOR回路と同様に構成できる．またOR，AND回路は，それぞれNOR，NAND回路にインバータを付加することで構成できる．以上をまとめると，(N)OR，(N)AND回路については，CMOSの方がSPLより少数のトランジスタで実現できる．

その一方で表1より，XOR，MUX回路にはSPLがトランジスタ数の点で有利であることが分かる．

そこで，実現すべき関数の種類に応じてCMOS，パス・トランジスタ論理を使い分けて回路を構成することが考えられる．このようなCMOS/パス・トランジスタ論理混在手法として，パス・トランジスタ論理をベースとするPCCL(Pass-transistor/CMOS Collaborated Logic)⁹⁾や，CMOS論理をベースとする手法¹⁰⁾が提案されている．

PCCLでは，CMOS論理が(N)OR，(N)AND関数に適していることに着目している．まずパス・トランジスタ論理の合成に用いるBDDで，与えられた論理関数を表現し，そのBDDのノードのうち，図2(a)の例に示すような片方のエッジが定数ノードに接続されているノードのみをCMOS実現部の候補とする．このように，PCCLはBDD上の局所変換に依存してお



(a) 置換可能な例 (b) 置換不可能な例

図2 PCCLにおけるBDDノードからCMOSゲートへの置換

Fig. 2 Replacement of BDD nodes

with CMOS gates in PCCL.

り，実現すべき論理関数の性質を考慮するには限界がある．また，パス・トランジスタ論理回路に含まれる信号線を新たにCMOS回路に接続するには，図2(a)に示すように出力インバータが必要となるため，トランジスタ数が増加することが多い．さらに，置換後のCMOSゲートには図2(a)のaのようにBDDノードの入力変数が直接入力される．つまり，入力がすべて関数である(N)OR，(N)AND回路は生成されない．そのため，図2(b)におけるノードaのように，いずれのエッジも定数ノードに接続されていない場合，CMOS回路への置換は不可能となる．

その一方でCMOS論理をベースとする手法¹⁰⁾では，まず既存のCMOS用合成ツールを利用し，論理回路をCMOS論理で合成する．その後，パス・トランジスタ論理に置き換えることで面積が小さくなる部分を置換する．この手法では，最初にCMOS論理用に回路を単純化するため，パス・トランジスタ論理で構成すればトランジスタ数が削減できる部分が，分断される可能性がある．文献7)では，多くの回路例に対してパス・トランジスタ論理がトランジスタ数において有利となることが示されている．よって，最初にCMOS論理用に単純化することは，トランジスタ数削減の観点から不利と考える．

本稿では，トランジスタ数の削減による低消費電力化を目的とし，新たなCMOS/パス・トランジスタ論理混在手法を提案する．提案手法では，PCCLと同様にパス・トランジスタ論理をベースとして，CMOS論理では(N)OR，(N)AND関数のみを実現する．本手法の特徴は，CMOS実現部分の選定に分解グラフ^{11),12)}を利用することで，BDDに依存するPCCLでは選定できない部分をCMOS実現部として選定する点にある．さらに，パス・トランジスタ実現部分にBDD分割¹³⁾を適用することで，レベル回復用のバッファ数を削減する．このような特徴を持つCMOS/パス・トランジスタ論理混在手法によってトランジスタ数を削減し，低消費電力化を実現する．

分解グラフの作成
CMOS 実現部の選定と合成
初期入力変数順序の決定
BDD の構築と DVO
BDD 分割
パス・トランジスタ実現部の合成

図 3 提案手法の処理概要
Fig. 3 Overview.

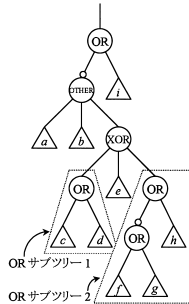


図 4 分解グラフの例
Fig. 4 Decomposition graph.

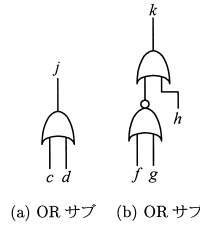


図 5 CMOS 実現部
Fig. 5 CMOS subcircuits.

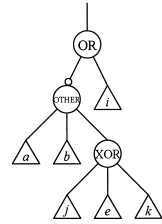


図 6 OR サブツリーを置換した分解グラフ
Fig. 6 Decomposition graph after replacement.

2. CMOS/パス・トランジスタ論理の混在による論理回路の合成手法

提案する CMOS/パス・トランジスタ論理の混在による論理回路の合成手法について、まず単一出力回路に限定して説明する。本手法では、2.1 節で述べるように、単純直交分解^{14)~16)}の分解構造を表現する分解グラフ^{11),12)}を利用している。分解グラフは各出力ごとに独立して得られるため、多出力回路に対しては 2.5 節で述べる工夫が必要である。

図 3 に本手法の処理概要を示す。

2.1 分解グラフの作成

ある論理関数 $F(X)$ が、互いに素な変数集合 X_1, X_2, \dots ($X = X_1 \cup X_2 \cup \dots$) に対する関数 $H_1(X_1), H_2(X_2), \dots$ を用いて

$$F(X) = G(H_1(X_1), H_2(X_2), \dots) \quad (1)$$

と表現できるとき、これを関数 $F(X)$ に対する単純直交分解^{14)~16)}という。

単純直交分解の分解構造を表現する分解グラフについて、BDD を用いて効率良く求める手法^{11),12)}が提案されている。図 4 に分解グラフの例を示す。提案手法では、この分解グラフを以下の 3 点で利用する。

- (1) CMOS 実現部分の選定
- (2) パス・トランジスタ回路の合成に用いる BDD の初期入力変数順序決定
- (3) BDD 分割のレベル区間選定¹³⁾

2.2 CMOS 実現部の選定と合成

1 章で述べた比較結果より、提案手法では (N)OR、(N)AND 回路を CMOS 回路として実現する。分解グラフでは、これらすべてが OR ノードで表される。

まず、以下の用語を定義する。

定義 1 OR サブツリー

分解グラフのある OR ノードにおいて、その子孫のノードがすべてリテラルノードか OR ノードであるサブツリーを OR サブツリーと呼ぶ。 □

図 4 に示した分解グラフは、2 つの OR サブツリーを含んでいる。OR サブツリーにおいて、そのサブグラフもまた OR サブツリーとなる場合、本稿では図 4 における OR サブツリー 2 のように、極大となる OR サブツリーを扱う。

分解グラフ中のすべての OR サブツリーを CMOS 実現部として選定する。OR サブツリーを扱うことで、PCCL では選定できなかった図 2 (b) のような (N)OR、(N)AND 関数も CMOS 実現部として選定可能である。なお OR サブツリーに含まれない OR ノードについては、CMOS で実現すると図 2 (a) に示したように出力インバータが必要となり、トランジスタ数が増加するケースが多いため、パス・トランジスタ論理で実現する。選定した部分を、図 5 に示すように合成する。各 OR サブツリーの出力を新たな入力変数 (図 5 では j, k) で置き換えることにより、図 4 の分解グラフは図 6 のように変換される。

なお、CMOS ゲートは最大 4 入力とし、OR ノードの入力がそれより多い場合は多段化する。

2.3 初期入力変数順序の決定、

BDD の構築と DVO

CMOS 実現部を入力変数に置換したあとの分解グラフに対応する部分を、パス・トランジスタ論理で実現する。本稿では、パス・トランジスタ論理として SPL を採用し、文献 13) の手法により合成した。すなわち、まず与えられた論理関数を表す BDD を構築した後、2.4 節で述べる方法によってパス・トランジスタ論理を合成する。

まず BDD を構築する前に、初期変数順序を決定する。BDD は、入力変数順序によってそのノード数が大きく異なるため、ノード数が少ない入力変数順序を求めることが重要となる。そこで提案手法では、分解グラフを用いた次の手順で初期変数順序を決定する。
手順 1 分解グラフを出力から深さ優先で探索する。このとき、出力から遠いノードを優先的に探索する。

出現したりテラルノードの逆順を変数順序とする。□

図 6 では, i, a, b, j, e, k の順となり, BDD のルートノードの変数が i となる。なお, 分解グラフにおいて, 同じ親を持つ子ノードどうしは交換可能であるため, a, b , もしくは j, e, k の順序は任意である。

求めた変数順序を初期入力変数順序とし, BDD を構築する。BDD 構築中には DVO (Dynamic Variable Ordering)⁷⁾ を行い, サイズを小さく抑えられる変数順序を探索する。なお, 3 章における実験の対象とした各回路に対して変数順序を求める実験を行った結果, DVO によって最終的に決定した変数順序において, 分解グラフの任意のサブツリーに含まれるすべての変数が連続することが分かった。これより, 手順 1 で求めた初期入力変数順序は, 最終決定に近い変数順序であると考えられる。

2.4 BDD 分割, パス・トランジスタ実現部の合成
構築された BDD に対して, 必要に応じて BDD 分割¹³⁾ を適用し, パス・トランジスタ論理を合成する。BDD 分割は,

- (1) トランジスタ数の削減
- (2) 中間バッファ数の削減

のいずれかが可能な場合に行う。特に (2) について, BDD 分割を適用するとトランジスタ数が増加する場合でも, 中間バッファ数を削減できる可能性があるため, 多少のトランジスタ数の増加は許容する¹³⁾。

分割するレベル区間 (あるレベル i と j の間の連続する区間 $[i, j)$ ($i < j$)¹³⁾) の選定にも, 分解グラフを利用する。前節で述べたように, 任意のサブツリーに含まれるすべての変数は BDD において連続する。このようなサブツリーに含まれる全変数からなるレベル区間, すなわちサブツリーに対応するレベル区間を分割する。たとえば図 6 では, XOR ノード以下のサブツリーに対応する j, e, k のレベル区間や, OTHER ノード以下のサブツリーに対応する a, b, j, e, k のレベル区間などが対象となる。他方, b, j, e, k のレベル区間はサブツリーと対応しないので分割対象としない。実験結果から, 分解グラフのサブツリーに対応するレベル区間を分割すると, サブツリーと対応しない区間を分割する場合と比べてトランジスタ数を削減できることを確認している。

2.5 多出力回路に対する合成手法

提案手法で利用している分解グラフは, 各出力ごとに独立して得られるため, 多出力回路に適用する場合は以下の 2 つの処理を追加する。

- (1) CMOS 回路実現部の共有
多出力回路において図 7 のように複数の出力で共有で

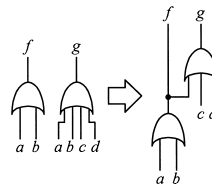


図 7 CMOS 実現部のゲート共有
Fig. 7 Gate sharing in CMOS subcircuits.

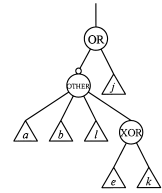


図 8 分解グラフの例 2
Fig. 8 Decomposition graph 2.

きる部分が存在すれば, それらを共有して合成する。図 7 では, $a + b$ の OR ゲートを共有することで, トランジスタ数を 16 個から 14 個に減少できる。

(2) 多出力回路に対する BDD 初期変数順序決定
まず分解グラフより, 各出力ごとに単一出力回路と同様に変数順序を得た後, それらを可能な限り維持できるようにマージして, 多出力回路に対する BDD の初期変数順序とする。たとえば図 6 と図 8 の 2 つの出力より得られた i, a, b, j, e, k と j, a, b, l, e, k をマージする場合, j に関しては両方の変数順序を維持することができないため, いずれかの変数順序を優先する。ここでは前者を優先するとする。また l については, a, b と j, e, k を維持できるように, その間に挿入される。これより変数順序は i, a, b, l, j, e, k となる。

3. 実験評価

MCNC LGSynth91 ベンチマーク回路の Combinational Multi-Level Examples¹⁸⁾ の中で単純直交分解可能な回路のうち, 実験時間の都合より 40 例を選択して提案手法を用いた合成実験を行った。比較対象として, CMOS のみで合成した回路と SPL のみで合成した回路を用い, トランジスタ数, 遅延時間, 消費電力について評価した。

プロセスルールは $0.35 \mu\text{m}$, 電源電圧は 3.3V とした。CMOS 回路は, スタンドセル・ライブラリを適用して市販の論理合成ツールで合成した。制約条件として, 最大面積 0 を与えた。また, 平坦化を行う場合と行わない場合の 2 種類の方法で合成し, トランジスタ数が少ない方を採用した。SPL 回路は, SPL 用論理合成 CAD⁷⁾ を用いて合成を行った。なお, SPL 回路の論理部分のトランジスタサイズに対する CMOS 回路の標準インバータの nMOS トランジスタサイズの比は 1.23 である。

3.1 トランジスタ数に関する結果

表 2 にトランジスタ数に関する結果を示す。「CMOS」は CMOS 回路, 「SPL」は SPL 回路のトランジスタ数を示す。また「提案/CMOS」, 「提案/SPL」は, そ

表2 トランジスタ数に関する結果

Table 2 Transistor counts.

回路名	入力	出力	CMOS			提案手法		SPL部全体
			SPL	提案	提案	CMOS	SPL	
alu2	10	6	1109	410	403	0.36	0.98	0.99
apex6	135	99	1958	1741	1541	0.79	0.89	0.81
b1	3	4	14	14	13	0.93	0.93	0.69
b9	41	21	335	395	329	0.98	0.83	0.82
c8	28	18	354	245	251	0.71	1.02	0.76
cc	21	20	166	171	148	0.89	0.87	0.69
cht	47	36	440	364	354	0.80	0.97	0.99
cm138a	6	8	76	75	76	1.00	1.01	0.11
cm151a	12	2	70	43	43	0.61	1.00	1.00
cm163a	16	5	118	100	102	0.86	1.02	0.69
cm42a	4	10	80	95	78	0.98	0.82	0.10
cm82a	5	3	66	50	49	0.74	0.98	1.00
cm85a	11	3	132	109	103	0.78	0.94	0.98
cmb	16	4	98	119	76	0.78	0.64	0.00
cordic	23	2	196	214	204	1.04	0.95	0.66
count	35	16	452	312	380	0.84	1.22	0.64
cu	14	11	159	134	153	0.96	1.14	0.46
decod	5	16	114	146	128	1.12	0.88	0.06
example2	85	66	969	937	908	0.94	0.97	0.80
f51m	8	8	315	159	161	0.51	1.01	0.96
fg1	28	3	162	203	220	1.36	1.08	0.82
i2	201	1	705	1589	574	0.81	0.36	0.11
i3	132	6	380	634	428	1.13	0.68	0.00
i4	192	6	628	874	688	1.10	0.79	0.00
lal	26	19	279	274	285	1.02	1.04	0.58
ldd	9	19	258	210	265	1.03	1.26	0.46
majority	5	1	34	25	25	0.74	1.00	1.00
parity	16	1	150	93	98	0.65	1.05	0.91
pcl	19	9	210	195	198	0.94	1.02	0.68
pcler8	27	17	277	292	322	1.16	1.10	0.50
pm1	16	13	132	148	132	1.00	0.89	0.30
t481	16	1	106	95	95	0.90	1.00	0.66
term1	34	10	377	349	334	0.89	0.96	0.77
ttt2	24	21	527	341	386	0.73	1.13	0.82
unreg	36	16	276	216	216	0.78	1.00	1.00
x1	51	35	921	1240	983	1.07	0.79	0.55
x2	10	7	137	101	109	0.80	1.08	0.54
x3	135	99	1957	1627	1547	0.79	0.95	0.82
x4	94	71	1106	1550	879	0.79	0.57	0.67
z4ml	7	4	100	74	74	0.74	1.00	0.97
幾何平均	-	-	234.8	216.6	200.7	0.85	0.93	0.56

それぞれ CMOS 回路と SPL 回路に対する提案手法により合成した回路の比を示す「SPL 部/全体」は、提案手法で合成した回路中の SPL 実現部の比を示す。

表 2 より、CMOS 回路に対して幾何平均で 15% の削減効果が得られた。提案手法のトランジスタ数が CMOS 回路より増加した回路は 40 例中 9 例だけであり、トランジスタ数削減に対する有効性が確認できた。SPL 回路に対しては 7% の削減効果が得られるとともに、40 例中 21 例について、より少ないトランジスタ数で回路を実現できた。

また比較のため、CMOS 回路に対して XOR, MUX

回路のみを SPL 回路に置換してトランジスタ数を求めたところ、CMOS のみの回路に対する削減率は 5% にとどまった。提案手法では、分解グラフを用いて実現すべき論理関数の性質に応じて CMOS 実現部を選定することで、10 ポイント高い削減効果が得られた。

3.2 遅延時間と消費電力に対する結果

HSPICE を用いた回路シミュレーションによって、遅延時間と消費電力を評価した。仮想配線容量として、CMOS 回路では各セルの出力に、SPL 回路では各 BDD ノードの出力に対応する端子に、それぞれ 10fF の容量を付加した。モデルには、MOS Level 28 を用いた。消費電力については、ランダムに生成した 100 個の入力パターンを各回路に与えて測定した。遅延時間については、消費電力測定時に与えた 100 個の入力パターンに対して遅延時間を測定するとともに、次のようにして求めたクリティカルパスの遅延時間を測定し、それらの最大値を採用した。CMOS 回路のクリティカルパスについては、市販の論理合成ツールによる遅延解析によって得られた最長 true パスとした。SPL、および提案手法で合成した回路に対しては、文献 7) と同様に、回路を RC 木で表現して遅延時間を見積もる手法によってクリティカルパスを求めた。

表 3 に遅延時間と消費電力に関する結果を示す。遅延時間について、CMOS 回路に対して幾何平均で 4% の増加となった。一方で、実験回路例の約半数においては CMOS 回路よりも遅延時間が短縮されている。SPL 回路に対しては、幾何平均で 34% 短縮された。40 例中 28 例において SPL 回路より遅延時間が短縮されている。特に、回路例 i2 に対して高い効果が得られた。i2 はその大部分が OR 関数で構成されており、SPL に対して不利な回路である。これらの OR 関数が提案手法によって CMOS に置換された結果、遅延時間が短縮されたと考える。

次に消費電力について、表 3 より、多くの回路でトランジスタ数と同様の比率で消費電力が削減されていることが分かる。CMOS 回路に対して幾何平均で 27% の低消費電力化を実現できた。提案手法の方が消費電力が増加した回路は、4 例のみであった。SPL 回路に対しては、幾何平均で 12% の削減結果となり、40 例中 24 例に対して低消費電力化が実現された。

ここで、CMOS 回路に対する遅延時間短縮効果が最高となった回路例 cc について考察する。図 9 に cc の CMOS 回路を示す。SPL 回路の参考として、図 10 に回路例 cc の BDD を示す。また、図 11 に回路例 cc の分解グラフを示す。

CMOS 回路のクリティカルパスはゲート 7 段を通

表3 遅延時間と消費電力に関する結果
Table 3 Delay and power dissipation.

回路名	遅延時間 [ns]					消費電力 [μ W]				
	CMOS	SPL	提案手法	提案/CMOS	提案/SPL	CMOS	SPL	提案手法	提案/CMOS	提案/SPL
alu2	3.23	3.22	3.49	1.08	1.09	401.9	136.6	146.6	0.36	1.07
apex6	1.76	2.31	2.99	1.70	1.29	569.0	542.5	456.0	0.80	0.84
b1	0.22	0.16	0.15	0.69	0.91	4.9	3.8	3.8	0.78	1.01
b9	1.10	2.37	1.83	1.67	0.77	90.8	99.0	90.3	0.99	0.91
c8	2.11	1.46	1.46	0.69	1.00	123.8	73.9	61.6	0.50	0.83
cc	1.07	0.62	0.50	0.47	0.82	56.8	50.1	40.9	0.72	0.81
cht	0.86	0.62	0.52	0.61	0.85	165.7	110.0	104.4	0.63	0.95
cm138a	0.69	0.67	0.43	0.63	0.64	13.8	8.4	11.8	0.85	1.39
cm151a	0.82	0.50	0.56	0.68	1.12	22.2	13.1	13.1	0.59	1.00
cm163a	1.04	0.88	0.89	0.86	1.01	34.0	27.7	24.9	0.73	0.90
cm42a	0.39	0.70	0.49	1.27	0.71	20.4	16.3	13.7	0.67	0.84
cm82a	0.72	0.66	0.54	0.75	0.82	27.2	15.5	15.8	0.58	1.02
cm85a	1.55	1.34	1.30	0.84	0.97	49.1	30.3	30.0	0.61	0.99
cmb	0.68	3.17	0.62	0.92	0.20	14.0	20.4	9.1	0.65	0.45
cordic	1.36	6.77	2.38	1.75	0.35	64.5	48.6	42.9	0.66	0.88
count	4.57	1.21	2.18	0.48	1.80	125.6	72.4	80.7	0.64	1.12
cu	1.05	1.92	1.29	1.23	0.67	44.0	25.2	32.3	0.73	1.28
decod	0.58	0.64	0.45	0.78	0.70	23.5	21.4	19.8	0.84	0.93
example2	2.08	3.46	4.14	1.99	1.20	231.5	198.4	187.9	0.81	0.95
f51m	2.72	1.15	1.05	0.39	0.92	141.5	55.3	58.9	0.42	1.07
frg1	1.22	7.95	4.60	3.77	0.58	53.2	59.3	61.4	1.15	1.03
i2	1.50	19.03	2.29	1.53	0.12	96.2	293.8	73.3	0.76	0.25
i3	0.95	4.25	0.73	0.77	0.17	104.3	142.7	106.1	1.02	0.74
i4	1.47	7.14	1.47	1.00	0.21	199.0	278.2	198.4	1.00	0.71
lal	2.50	4.30	1.20	0.48	0.28	77.9	59.3	54.9	0.70	0.93
ldd	1.93	1.42	0.95	0.49	0.67	72.6	60.4	73.1	1.01	1.21
majority	0.61	0.42	0.34	0.55	0.81	10.8	7.6	7.8	0.72	1.03
parity	1.51	3.22	1.23	0.82	0.38	72.4	36.3	37.4	0.52	1.03
pcle	0.82	0.73	1.18	1.43	1.61	50.5	48.6	46.4	0.92	0.95
pcler8	1.20	1.71	0.98	0.82	0.57	59.7	75.5	70.7	1.18	0.94
pm1	0.73	1.42	0.92	1.25	0.64	31.5	31.2	24.2	0.77	0.78
t481	0.78	3.55	1.25	1.61	0.35	31.8	29.7	27.3	0.86	0.92
term1	1.61	7.45	2.77	1.72	0.37	118.6	107.7	79.6	0.67	0.74
ttt2	1.58	4.55	1.92	1.21	0.42	176.5	108.7	110.5	0.63	1.02
unreg	1.00	3.25	3.25	3.26	1.00	91.6	65.7	65.7	0.72	1.00
x1	1.31	6.18	3.33	2.55	0.54	246.9	338.5	238.4	0.97	0.70
x2	0.72	0.83	1.09	1.52	1.32	32.6	24.0	26.7	0.82	1.11
x3	1.57	2.39	3.51	2.24	1.47	590.7	520.5	463.3	0.78	0.89
x4	1.78	3.20	1.78	1.00	0.56	345.6	562.1	252.6	0.73	0.45
z4ml	0.90	0.99	0.99	1.10	1.00	41.3	24.6	24.6	0.59	1.00
幾何平均	1.16	1.84	1.21	1.04	0.66	68.3	56.5	49.9	0.73	0.88

る．これに対して SPL 回路に対応する BDD の最大段数は 6 段であり，トランジスタの直列段数は最大 5 段となる．ゲートと BDD の段数を直接比較することはできないが，表 3 より遅延時間が削減されていることが分かる．提案手法ではこれよりさらに遅延時間を削減できている．図 11 より cc には複数の OR サブツリーが含まれていることが分かる．このうち SPL 回路のクリティカルパスに対する出力関数は，全体が OR サブツリーで表現されている．つまり，パス・トランジスタ回路に対して不利な (N)OR, (N)AND 関数が最長遅延を決定づけていたといえる．提案手法によ

て，SPL 回路でのクリティカルパスは 3 段の CMOS ゲートに置き換えられた．その結果，遅延時間が削減されたのみならず，トランジスタ数減少によって消費電力も削減された．

以上の結果より，分解グラフを用いて単純直交分解された関数に適した論理構成方式を採用する提案手法による，低消費電力化の効果が確認された．

4. ま と め

CMOS/パス・トランジスタ論理を混在させ，それぞれが得意とする回路を有効に割り当てることでトラ

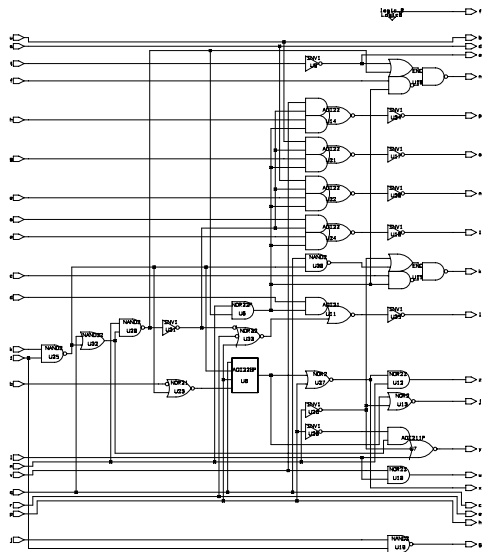


図9 回路例 cc の CMOS による実現
Fig. 9 CMOS circuit for "cc".

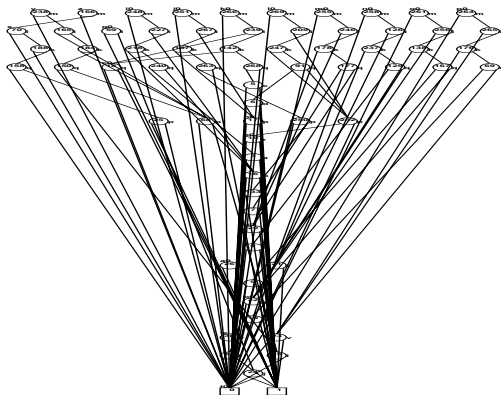


図10 回路例 cc の BDD による表現
Fig. 10 BDD representation for "cc".

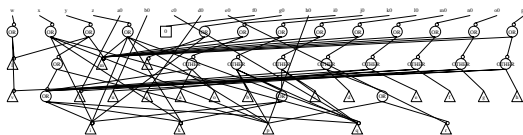


図11 回路例 cc の分解グラフ
Fig. 11 Decomposition graph for "cc".

ンジスタ数を削減し，低消費電力化を実現する手法を提案した．

本手法は，単純直交分解を表す分解グラフを利用して CMOS 実現部を抽出することで，実現すべき関数に適した論理構成方式を採用する点に特徴を持つ．さらに，パス・トランジスタ論理で実現する部分を合成する際にも分解グラフを利用することで，適切な初期

変数順序，ならびに BDD 分割結果を得た．その結果，面積最小化指向で合成した CMOS 回路や，SPL 回路に対して，それぞれ 27%，12%の低消費電力化を達成した．

今後は，多出力回路に対する CMOS 実現部の抽出法について，さらに改良を加えることを課題とする．

謝辞 単純直交分解手法についてご教示をいただいた(株)富士通研究所松永裕介氏に深謝します．

なお本研究の一部は，文部省科学研究費補助金(特別研究員奨励費)による．

参考文献

- 1) Devadas, S. and Malik, S.: A survey of optimization techniques targeting low power VLSI circuits, *32nd Design Automation Conf.*, pp.242-247 (1995).
- 2) 日経マイクロデバイス(編): 低電力 LSI の技術白書—1 ミリ・ワットへの挑戦, 日経 BP 社 (1994).
- 3) Sakurai, T., Lin, B. and Newton, A.R.: Multiple-output shared transistor logic (MOSTL) family synthesized using binary decision diagram, Dept. EECS, Univ. of Calif., Berkeley, ERL Memo M90/21 (1990).
- 4) Yano, K., Yamanaka, T., Nishida, T., Shimo-higashi, K. and Shimizu, A.: A 3.8-ns CMOS 16×16 -b multiplier using complementary pass-transistor logic, *IEEE Journal of Solid-State Circuits*, Vol.25, No.2, pp.388-395 (1990).
- 5) Sasaki, Y., Yano, K., Yamashita, S., Chikata, H., Rikino, K., Uchiyama, K. and Seki, K.: Multi-level pass-transistor logic for low-power ULSIs, *Proc. International Symposium on Low Power Electronics* (1995).
- 6) Yano, K., Sasaki, Y., Rikino, K. and Seki, K.: Top-down pass-transistor logic design, *IEEE Journal of Solid-State Circuits*, Vol.31, No.6 (1996).
- 7) Konishi, K., Kishimoto, S., Lee, B.-Y., Tanaka, H. and Taki, K.: A logic synthesis system for the pass-transistor logic SPL, *SASIMI '96*, pp.32-39 (1996).
- 8) Buch, P., Narayan, A., Newton, A.R. and Sangiovanni-Vincentelli, A.: On synthesizing pass transistor networks, *IWLS '97* (1997).
- 9) Yamashita, S., Yano, K., Sasaki, Y., Akita, Y., Chikata, H., Rikino, K. and Seki, K.: Pass-transistor/CMOS collaborated logic: The best of both worlds, *Symposium on VLSI Circuits*, pp.31-32 (1997).
- 10) 岡部直樹, 北村清志, 瀧 和男: パストラジスタ/CMOS 混在による低消費電力 LSI 設計, 第 2 回システム LSI 琵琶湖ワークショップ, pp.289-

291 (1998).

- 11) Bertacco, V. and Damiani, M.: The disjunctive decomposition of logic functions, *ICCAD '97*, pp.78–82 (1997).
- 12) Matsunaga, Y.: An exact and efficient algorithms for disjunctive decomposition, *SASIMI '98*, pp.44–50 (1998).
- 13) 高田賢吾, 井上真一, 沼 昌宏, 瀧 和男, 平野浩太郎: BDD 分割を用いたパス・トランジスタ論理の合成, *情報処理学会論文誌*, Vol.40, No.4, pp.1557–1564 (1999).
- 14) Ashnhurst, R.L.: The decomposition of switching functions, *ISTIS*, pp.74–116 (1957).
- 15) Roth, J.P. and Karp, R.M.: Minimization over Boolean graphs, *IBM Journal of Res. and Dev.*, pp.227–238 (1962).
- 16) Curtis, H.A.: *A new approach to the design of switching circuit*, Van Nostrand, Princeton, NJ (1962).
- 17) Rudell, R.: Dynamic variable ordering for ordered binary decision diagrams, *IWLS '93* (1993).
- 18) Yang, S.: *Logic synthesis and optimization benchmarks user guide version 3.0. MCNC* (1991).

(平成 12 年 9 月 18 日受付)

(平成 13 年 2 月 1 日採録)



高田 賢吾 (学生会員)

1997 年神戸大学工学部電気電子工学科卒業。1999 年同大学大学院修士課程修了。現在、同博士課程在学中。日本学術振興会特別研究員。低消費電力回路技術に興味を持つ。



神野 元彰

2000 年神戸大学工学部電気電子工学科卒業。在学中に低消費電力回路の合成に関する研究に従事。現在、奈良先端科学技術大学院大学修士課程在学中。



黒木 修隆

1990 年神戸大学工学部電子工学科卒業。1995 年同大学大学院自然科学研究科博士課程修了。同年同大学工学部電気電子工学科助手。工学博士。画像処理, LSI 設計に関する研究に従事。電子情報通信学会会員。



沼 昌宏 (正会員)

1983 年東京大学工学部精密機械工学科卒業。1985 年同大学大学院修士課程修了。同大学助手を経て 1989 年同大学講師。1990 年神戸大学大学院自然科学研究科講師。1995 年同大学工学部電気電子工学科助教授。工学博士。主に LSI CAD, FPGA 応用, 画像処理に関する研究に従事。IEEE, ACM, 電子情報通信学会各会員。



瀧 和男 (正会員)

1952 年生。1979 年神戸大学大学院修士課程システム工学専攻修了。同年(株)日立製作所入社。1982 年(財)新世代コンピュータ技術開発機構研究所に出向。逐次型および並列型推論マシンと並列応用プログラムの研究開発に従事。1990 年同機構第 1 研究室室長。1992 年神戸大学工学部情報知能工学科助教授。1995 年同学科教授。工学博士。1987 年元岡賞, 2000 年山下記念研究賞受賞。LSI 設計と CAD, 並列マシンのアーキテクチャ, 並列プログラミング等に興味を持つ。電子情報通信学会, IEEE, ソフトウェア科学会, ACM, 日本神経回路学会各会員。



山本 啓輔

1962 年神戸大学工学部電気工学科卒業。同年松下電器産業(株)入社。主としてテレビ受信機の開発, 研究に従事。2000 年 5 月より神戸大学工学部電気電子工学科教授。工学博士。放送と通信の融合, 画像処理, LSI CAD に関する研究に従事。映像情報メディア学会会員。