

# パストランジスタ論理による加算器の設計手法と設計事例

李 副 烈† 瀧 和 男††  
宇田 研一郎† 北 村 清 志†

パストランジスタ論理に基づく加算器の設計手法を提案する。提案手法は、演算回路の BDD 表現が規則性を有することを利用して手作業で BDD の最適化を行い、さらに NMOS と CMOS-TG を使い分けることで回路の遅延と面積を削減する。BDD に基づいて必要なセルを作成し、スタンダードセル方式により配置配線する。これらのセルは基本セルを組み合わせた複合セルとして実現し、設計の手間を省いている。提案手法を用いた設計事例として 32 ビット・キャリ・セレクト加算器を 3 つ設計した。NMOS, CMOS-TG とその 2 つを使い分けた HYBRID である。HYBRID 加算器は ED 積、面積ともに最も小さい結果となった。また、CMOS の 32 ビット・キャリ・ルックアヘッド加算器と比較した結果、面積で 77%、ED 積で 69% という良好な結果を示した。

## A Pass-transistor Logic Design Method for Adders and Application Examples

BU-YEOL LEE,<sup>†</sup> KAZUO TAKI,<sup>††</sup> KEN'ICHIRO UDA<sup>†</sup>  
and KIYOSHI KITAMURA<sup>†</sup>

This paper describes a pass-transistor logic (PTL) design method for adders. First, we design PTL circuits with Binary Decision Diagrams (BDDs). Then they reduce delay and area to make proper use of NMOS-PTL and CMOS-TG. At last, we design physical layouts with PTL cell library. We can design compact PTL cells easily using compound cells which consist of some basic cells. We designed 3-type 32-bit carry select adders (NMOS-PTL, CMOS-TG, HYBRID). A HYBRID adder, with the proper use of NMOS-PTL and CMOS-TG, shows best performance in ED product and area. Furthermore, a HYBRID adder shows that our method reduces ED product by 31% and area by 23% in comparison with a CMOS carry look-ahead adder.

### 1. ま え が き

近年、LSI の低消費電力化が必要不可欠となっており、それを実現する回路方式の研究もさかんに行われている<sup>5)</sup>。中でもパストランジスタ論理は従来の CMOS 回路よりも低消費電力、高速、小チップ面積を実現できる可能性を持つ回路方式として注目を集めている<sup>1)~4),6),7),10),12)~17),20),21),23),24)</sup>。

パストランジスタ論理の設計方法として、レイアウトまで考慮したものに LEAP<sup>19),20)</sup> がある。LEAP では BDD (Binary Decision Diagram, 二分決定グラフ) を用いて合成された回路をあらかじめ用意された少数のパストランジスタ論理用スタンダードセルに自

動でマッピングすることによりレイアウトを行う。一方、本論文では加算器を手設計でチューニングし設計する手法を提案する。提案手法では、加算器の規則性を利用して、BDD を手設計でチューニングし、任意の形の BDD を必要に応じて複合セルとして設計しレイアウトのチューニングを行う。複合セルはあらかじめ設計した基本セルを組み合わせることによって簡単に設計でき、基本セルを手設計でチューニングすることで小面積設計を実現する。設計した複合セルは既存の配置・配線ツールを用いてレイアウトする。以下、2 章で提案手法の概略について述べ、3 章で BDD を用いた回路の設計手法を示す。また、4 章で用途よっての NMOS パストランジスタと CMOS トランスミッション・ゲート (以降、CMOS-TG) の使い分けについて述べる。また、5 章では BDD を用いて設計されたパストランジスタ回路のレイアウト手法を提案する。さらに、6 章ではシングル・ルール構成とダブル・ルール構成について考察を行う。最後に 7 章で、提案手法

† 神戸大学自然科学研究科  
Graduate School of Science and Technology, Kobe University

†† 神戸大学工学部  
Faculty of Engineering, Kobe University

を用いて 32 ビット加算器を設計し評価を行う。

## 2. パストランジスタ論理による回路の設計手法

パストランジスタ論理を構成する手法の 1 つに BDD<sup>22)</sup> を使用した構成方法がある<sup>13), 23)</sup>。これは論理関数を BDD で表現し、トランジスタ回路にマッピングするというものである。加算器をはじめとする演算要素の論理関数は各ビットごとにみると同一の演算を行っているため、その論理関数を BDD を用いて表現すると、規則性の高い繰り返し構造を持つグラフとなるのが一般的である。そこで、手設計で加算器のチューニングを行う設計手法を提案する。規則性の高い BDD を手作業でチューニングし、BDD のノード単位で手設計したセルを用いてレイアウトを行う。提案手法は大きく 3 つの設計フェーズに分けられる。

**BDD を用いた回路設計：** 演算系の回路を BDD ノードの共有化と同形部分グラフの置換という手法<sup>17)</sup> を用いて設計する。設計された回路は BDD ノードと入出力バッファ（インバータ）、中間バッファ（インバータ）のネットリストとして表現する。

**NMOS と CMOS-TG の使い分け：** クリティカルパスがパス入力信号の伝搬かセレクト入力信号の伝搬かによって、NMOS パストランジスタと CMOS-TG を使い分け、遅延時間を短縮する<sup>7)</sup>。

**BDD ベースのレイアウト設計：** BDD のネットリストからセル数が少なくなるように共通の形を 1 つのセルとして設計する。セルの設計には手作業でチューニングを行った基本セルの組合せで複合セルを作ることで小面積に設計する。

## 3. BDD を用いた加算器の設計

### 3.1 BDD による回路構成法

BDD を用いたパストランジスタ回路の設計では、回路規模および動作速度が BDD の構造に大きく依存する。図 1 に示すように BDD のノードとパストランジスタのセレクトは 1 対 1 の対応をしているため、ノード数と直列段数が少ない BDD を作ることが良い回路を作ることにつながる。我々は以下の方法を組み合わせることによって、BDD のノード数や直列段数を削減し、消費電力と遅延を小さくする<sup>17)</sup>。

**変数順序の決定とノードの共有化：** 小さな BDD を得るために複数の論理関数どうして BDD のノードを共有させる（共有 BDD を構成する）。BDD の大きさは変数順序により大きく変化するので、最良の変数順序を求める必要がある。一般的に

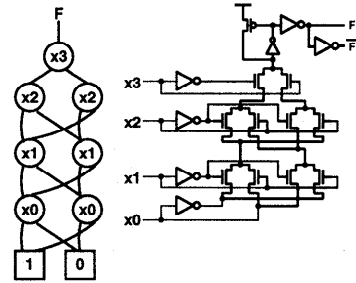


図 1 BDD とパストランジスタ論理  
Fig. 1 BDD and its pass-transistor logic.

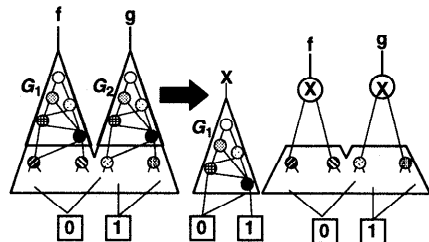


図 2 部分グラフの置換  
Fig. 2 Partial diagram replacement.

ランダム・ロジックは動的変数順序付けに基づくヒューリスティクスを用いて近似最小化を行うが<sup>3), 4)</sup>、演算系回路では規則性の強い繰り返し構造を持つグラフとなるため、手設計可能である。

**同形部分グラフの置換：** BDD ではノード変数とノード間の接続が同じである同形部分グラフが存在しても、それらの 0 枝、1 枝の指すノードが異なれば共有することはできない。しかし 2 入力 1 出力の部分グラフが複数個存在する場合、それらの部分グラフを新たな変数のノードに置換し、新たな変数の値は置換された部分グラフを用いて生成する。図 2 に例を示す。部分グラフ  $G_1$  と  $G_2$  をノード変数  $X$  に置き換えることで、部分グラフを 1 つにし BDD のサイズを小さくできる。また、部分グラフがノード 1 つに置き換わるため BDD の段数も小さくすることが可能である。部分グラフの出力が左の入力とつながる場合は  $X$  は 0、右の入力とつながる場合は  $X$  は 1 となるので、置換後の BDD は同論理を表す。多段化された BDD を用いたパストランジスタ論理の設計手法は MPL<sup>15)</sup> にて提案されたものである。

**否定枝の利用：** BDD の否定枝表現を利用して木が十分小さくなる場合、否定枝をインバータにマッピングする。

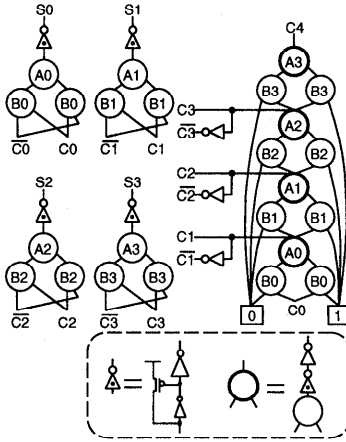


図3 4ビット・リプル・キャリ加算器のBDD表現  
Fig.3 BDD description of 4-bit ripple carry adder.

3.2 リプル・キャリ加算器の設計

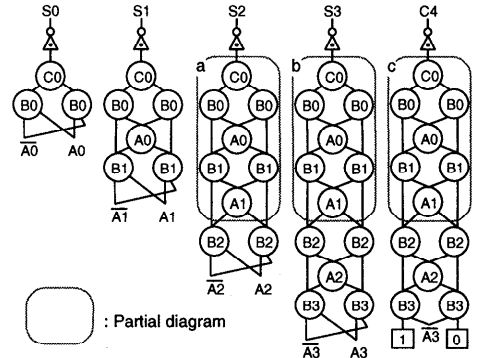
この節ではBDDノードの共有化を用いた回路の設計法をリプル・キャリ加算器を用いて説明する。

演算系回路の各出力の論理関数をBDDで表現し、それらのBDDを共有化させる。共有化を行うために、各論理関数は共通の入力変数をBDDのリーフ側に置く変数順序のBDDで構成しなければならない。演算系回路の場合、Nビット目の出力信号は最下位ビットからNビットまでの入力信号によって演算されるので、下位ビットの信号ほど共通の入力信号になる。したがって、各出力の論理関数を共有化するためには、BDDの変数順序を最上位ビットの入力変数から最下位ビットの入力変数まで順に並べればよい。

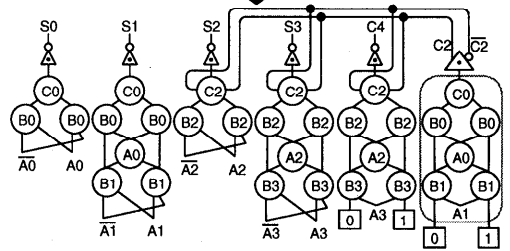
実際に、4ビット・リプル・キャリ加算器の設計例を示す。図3は、4ビット加算器の最もコンパクトなBDD表現である。既約共有BDDにおける共有が最大になるようBDDの変数展開順序を選び、かつ否定枝表現も用いてBDDの最小化を行ったものである。具体的にS3の論理関数の変数順序は、{A3, B3}, {A2, B2}, {A1, B1}, {A0, B0, C0}である。ただし、{}の変数順序は任意である。否定枝の部分インバータで置き換え、回路図に近い表現をとった<sup>6)</sup>。C4を生成するBDDの途中の値がC1, C2, C3となっており、キャリが順次伝搬してC4を出力する。すなわちリプル・キャリ加算器である。ここでC1~C3は各々S1~S3の生成に使われている。図3に示した回路はトランジスタ数最小であるが、BDDの高さ(段数)が大きく遅延時間は長い。

3.3 キャリ・セレクト加算器の設計

BDDを構築するときに変数順序を図3の場合とまったく逆にすると、図4(1)のBDDが得られる。この



(1) Before "partial Diagram Replacement"



(2) After replacement (Carry select adder)

図4 4ビット・キャリ・セレクト加算器のBDD表現  
Fig.4 BDD description of 4-bit carry select adder.

場合は部分グラフの共有はまったく行われていない。ところがこのBDDには、破線で囲んだ部分に同形部分グラフが存在し、3.1節で述べた同形部分グラフの置換の手法を用いると、図4(2)の多段化されたBDDが得られる。

すなわち、抜き出された同形部分グラフの出力を新しい中間変数C2とし(同図右端)、もとのグラフの破線内を1個のノードC2で置き換えている。これにより、S2, S3, C4を生成するBDDは、C2を使うことでノード数の減少とともにBDDの高さも低くなっている。リプル・キャリ加算器と比べるとBDDの高さ(段数)が低くなった分だけ遅延短縮の効果がある。

中間変数C2はちょうど2ビット目のキャリに相当している。たとえば、S3を生成するBDDでは、ノードC2の左枝と右枝の入力は、それぞれC2がゼロの場合のS3、C2が1の場合のS3である。すなわちC2が決まる前に、C2がゼロの場合と1の場合のS3をそれぞれ生成しておき、それらをあとから決まるC2で選択すると考えることができる。これらのことから図4(2)の回路は、キャリ・セレクト加算器である。

### 4. NMOS パストランジスタと CMOS-TG の使い分け

後述の 5.1 節のパストランジスタ論理用セルを用いて、図 5 に示す回路を作成し、NMOS、CMOS-TG それぞれの回路のパス入力とセレクト入力に対する遅延時間を計測した。0.35  $\mu\text{m}$  プロセス、電源電圧 3.3 V で負荷容量 (図 5(a) の C)、トランジスタの直列個数を変化させながら計測を行った。

**負荷容量と遅延時間：** 図 6(a) にトランジスタの直列個数が 3 の場合の負荷容量に対する特性を示す。パス入力については NMOS の方が CMOS-TG より高速であり、負荷容量が増加しても遅延時間の差は変化しない。したがって、負荷容量が小さい場合は NMOS を用いた方が有利である。一方、セレクト入力については CMOS-TG の方がつねに高速であった。特に負荷容量が増加するにつれ、その差が大きくなる。NMOS の場合は図 5(c1) に示すように左側のパストランジスタがオンになる入力パターンでは、入力インバータを通して入力されるが、CMOS-TG の場合は図 5(c2) に示すようにどちらの CMOS-TG がオンになってもつねにセレクト入力が NMOS か PMOS のどちらかに直接入力される。したがって、CMOS-TG の方がより高速になる。

**トランジスタの直列個数と遅延時間：** 図 6(b) に負荷容量を 40 fF とした場合のトランジスタ直列個数に対する特性を示す。パス入力については個数が増えるにつれ、NMOS の方が高速となる。一方、セレクト入力に関しては、直列個数とは無関係に、NMOS の方が入力インバータの通る分遅くなった。

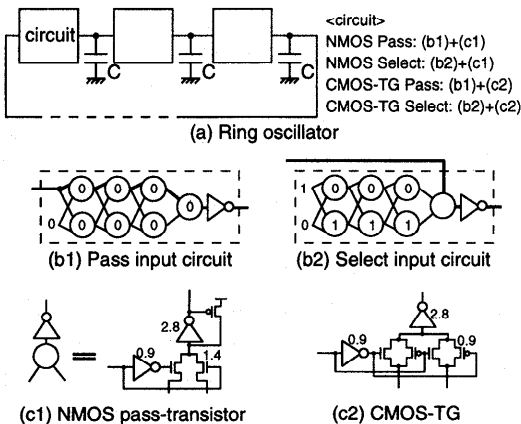


図 5 動作特性測定用のテスト回路

Fig. 5 Test circuits for measurement characteristic of pass-transistor logic.

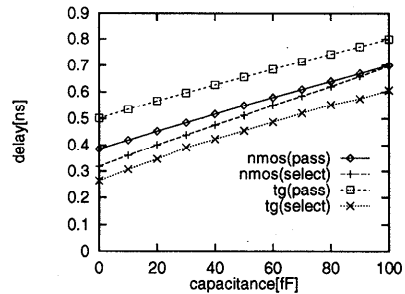
係に、NMOS の方が入力インバータの通る分遅くなった。

以上のことから、CMOS-TG はクリティカルパスがセレクト入力による場合や、負荷容量が大きい場合に適する。一方、NMOS はクリティカルパスがパス入力による場合や直列個数が多い場合に適する。したがって、パストランジスタ論理の実現方法は目的によって使い分ける必要がある。

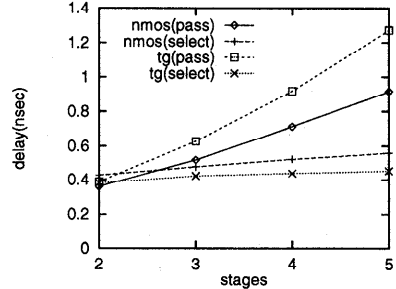
### 5. パストランジスタ回路のレイアウト手法

#### 5.1 パストランジスタ論理用セル

BDD ベースのパストランジスタ回路をレイアウト



(a) load capacitance (#stages=3)



(b) transistor stages(C=40(fF))

図 6 NMOS および CMOS-TG の遅延特性  
 Fig. 6 Delay of NMOS and CMOS-TG.

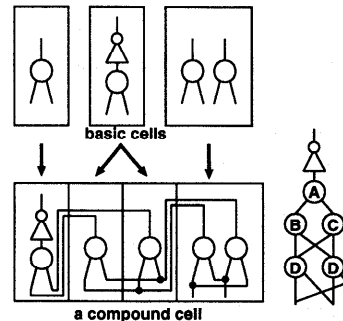


図 7 複合セルの設計

Fig. 7 Compound cell design.

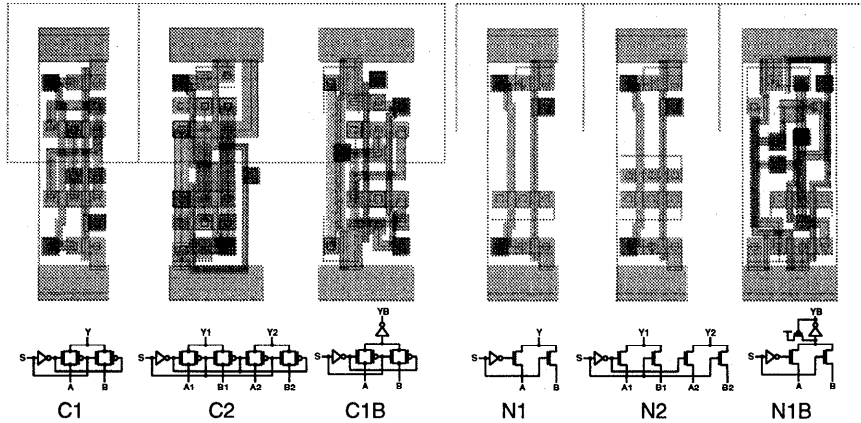


図 8 基本セル☆

Fig. 8 Basic cells.

する手法として、基本セルと複合セルを用いたセル設計手法を提案する。提案するセル設計手法は数個の基本セルを組み合わせで設計した複合セルを用いて回路を構成するものである。基本セルの単位は BDD のノード 1 個または 2 個、および出力インバータ付きノードとする。複合セルはこれらの数個の基本セルを並べることによって簡単に設計できるため、基本セルを手設計でチューニングし小面積にするだけで複合セルを容易に小面積化することができる。

図 7 に 3 種類の構造を持った基本セルから、複合セルを設計する簡単な例を示す。合成したい回路を BDD で表現し、構造やノードの段数を考慮したひとかたまりの BDD を複合セルとして設計する。加算器用、乗算器用、論理合成用といった用途別の複合セルライブラリを簡単に用意することが可能である。また、ライブラリごとに複合セルを必要に応じて追加していくことも容易である<sup>18)</sup>。

提案設計手法以外にも、パストランジスタ論理用セルの自動合成手法としては ALPS (An Automatic Layouter for Pass-Transistor Cell Synthesis)<sup>14)</sup> が提案されている。ALPS は小さいパストランジスタと入力インバータの縦積みを許容しながら複合セル単位で自動合成を行う。

## 5.2 基本セルの設計

パストランジスタ論理の場合、出力インバータやバッファによって駆動力を確保しておく、パストランジスタのトランジスタサイズを小さくできる。このためパストランジスタ論理用セルを構成する際は、サイズの小さいトランジスタを用いることにより、セル

内部のトランジスタを縦積みに配置することができる。これは面積および消費電力の削減ができる点で有利である。

図 8 に設計した基本セルを示す。左の 3 つは CMOS-TG の基本セルで、BDD のノード 1 個分 (C1)、2 個分 (C2)、出力インバータ付きノード (C1B) を構成するセルである。右の 3 つは NMOS パストランジスタ論理の基本セルである (N1, N2, N1B)。セレクト入力の反転論理を生成するインバータをセル内に含めることで、セル間の配線を減らした。このインバータは BDD のノードを構成するトランジスタのみ駆動できればよいので、サイズの小さいトランジスタを使用することができる。

## 6. シングル・レール構成とダブル・レール構成について

たとえば、図 4(2) に示した BDD に対して図 7 に示した手法により複合セルを構成し、論理変数の正論理と負論理を必要とする場合 (図 4(2) では C2 と C2 の否定) には、図 1 のようにインバータを用いて一方から他方を生成するならば、シングルレール構成の回路が得られる。BDD から最も単純な変換で得られる回路である。

一方、C2 を生成するための BDD と C2 の否定を生成するための BDD を別々に用意し、各々に対応する複合セルを設けるならば、その部分はダブルレール構成の回路となる。ダブルレール構成を用いると、倍のトランジスタ数を必要とするかわりにインバータ 1 段分遅延が短縮できる。さらに、C2 と C2 の否定のようにそれらが次段のセレクト入力につながる場合には、CMOS-TG の基本セルを用いる場合に限り出力

☆ 図 8 のレイアウトに関し特許出願されている。

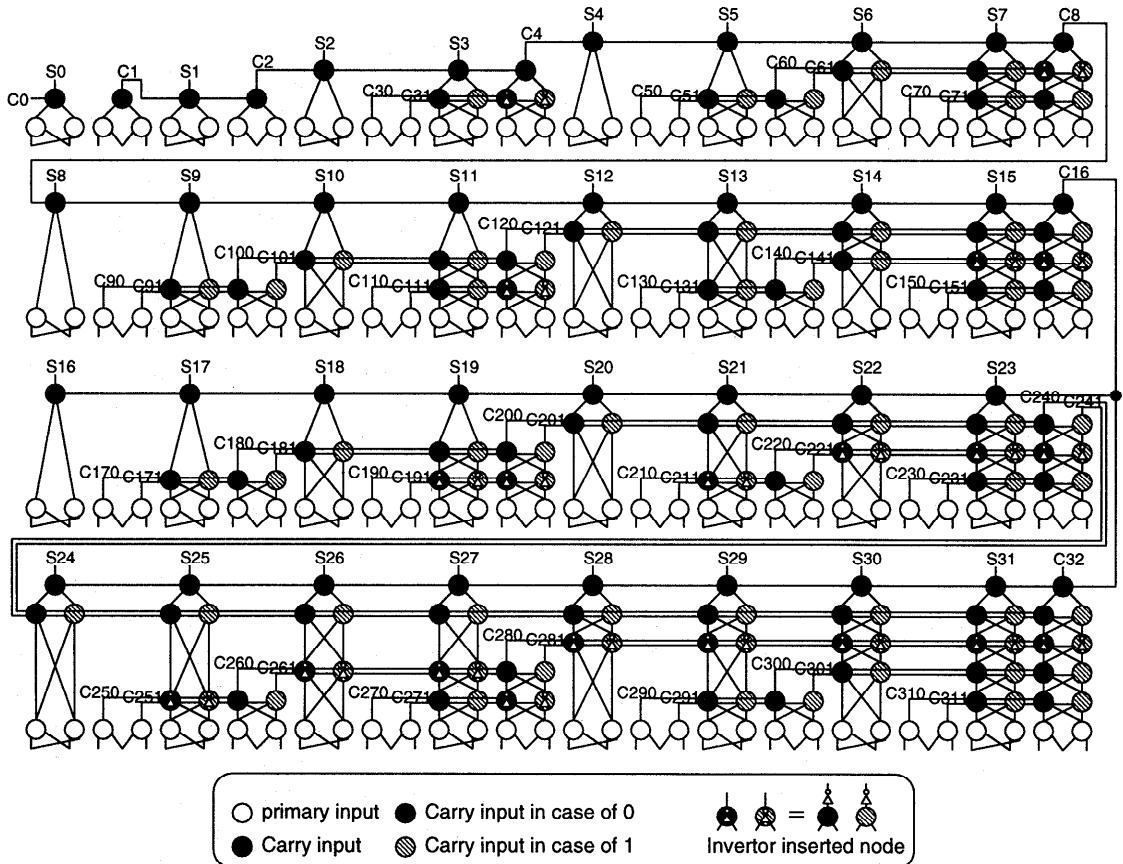


図9 32ビット・キャリ・セレクト加算器  
Fig. 9 32-bit carry select adder.

インバータ（BDDの出力部のインバータ）を省略することもでき、さらなる遅延短縮が可能となる（ただし遅延を短縮したければCMOS-TG基本セルには大きいトランジスタを使う必要が生じる）。

これらにより、セレクト入力クリティカルパスとなる回路に関しては、上記ダブルレール構成を活用することで遅延短縮が可能である。ただし、トランジスタ数が増え、大きなトランジスタを使う必要もあることから消費電力は増大する。この方法で高速化した例は文献(8), 9)に見られる。以下の例では、高速化よりも消費電力（ED積）の削減に重点を置いたため、ダブルレール構成はとっていない。

### 7. 32ビット・キャリ・セレクト加算器の設計と評価

#### 7.1 32ビット・キャリ・セレクト加算器の設計

3.3節の方法を応用し32ビットのキャリ・セレクト加算器を設計した結果を図9に示す。より正確にいうと、C2の生成まではキャリ・セレクト加算器になっ

ているが、それより上位ではキャリの伝搬経路はC4, C8, C16, C32と2のべき乗になっており、Conditional Sum加算器の構成である。したがって、伝搬遅延は桁数nに対してlog nのオーダーであり、キャリ・ルックアヘッド加算器と同じになっている。

BDDの構成上注意すべき点<sup>24)</sup>を述べる。C30, C31は各々、C2がゼロの場合のC3とC2が1の場合のC3である。これらの信号はS3, C4の生成に使われる。C4生成のBDDについて説明する。C30およびC31各々のノードの左枝および右枝の入力は、各々C3がゼロおよび1の場合のC4の値となっている。それをC30およびC31で選択し、その結果をさらにC2で選択してC4を得ている。この構成は、最も遅れて決まるC2による選択を出力に最も近いところでを行い、遅延を短縮しようとするものである。

C8の生成では、上記と同様の構成が二重になって現れる。C70とC71はそれぞれC6がゼロおよび1のときのC7である。C60とC61はそれぞれC4がゼロと1のときのC6である。C8の生成では、C7が

表 1 32ビット加算器のシミュレーション結果  
Table 1 Simulation results of 32-bit adders.

	CMOS-CLA-1	CMOS-CLA-2	NMOS-CSA	CMOSTG-CSA	HYBRID-CSA
トランジスタ数	2036	2670 (1.31)	1710 (0.84)	2052 (1.01)	1736 (0.85)
面積 [ $\mu\text{m}^2$ ]	30041	40260 (1.34)	24345 (0.81)	27622 (0.92)	23260 (0.77)
遅延時間 [ns]	2.31	1.88 (0.81)	2.93 (1.27)	2.17 (0.94)	2.30 (1.00)
エネルギー [pJ]	90.7	106.1 (1.17)	73.5 (0.81)	67.8 (0.75)	62.9 (0.69)
ED 積 (比)	(1.0)	(0.95)	(1.03)	(0.70)	(0.69)

ゼロおよび1の場合のC8の値をあらかじめ作っておき、それをC70およびC71で選択し、さらにその結果をC60とC61で選択し、最後にその結果をC4で選択している。C8の生成についても、最も遅れてくるC4による選択を最も出力に近いところで行い、遅延を短縮している。

また、キャリ伝搬が行われるノードの直前のパスにインバータを挿入し、インピーダンスを下げた。このことによって、遅れて切り替わるセレクト入力(キャリ入力)に対する遅延を短縮した<sup>17)</sup>。図9では、入出力インバータは省略し、中間インバータのみを表記した。

## 7.2 比較評価

キャリ選択加算器を用いてNMOSパストランジスタ論理とCMOS-TGパストランジスタ論理の比較を行う。

キャリ選択加算器をNMOSパストランジスタ、CMOS-TG、両方を合わせたもので設計し(設計した加算器を以降、NMOS-CSA、CMOSTG-CSA、HYBRID-CSAとよぶ)、パストランジスタ用スタンダードセルを用いてレイアウトを行った。各加算器は図9の加算器のキャリ生成用BDD(32個)およびサム生成用BDD(32個)を各々複合セルとして設計した。各加算器の基本セル、複合セル数は以下である。ただし、基本セル1個からなる複合セルは基本セルとして数えた。

**NMOS-CSA** 基本セル6個と複合セル15個。

**CMOSTG-CSA** 基本セル7個と複合セル14個。

**HYBRID-CSA** NMOS-CSAとCMOSTG-CSAの複合セルを用いた。その内訳はNMOSの基本セル5個と複合セル6個、CMOS-TGの基本セル6個と複合セル8個。

比較評価用のCMOS回路として、自動合成ツールを用いて2種のキャリ・ルックahead加算器を設計した。提案手法で設計したHYBRID-CSAと同速度に合成したCMOS-CLA-1とできるだけ高速に合成したCMOS-CLA-2である。

各加算器のレイアウトから配線容量を抽出し、電源

電圧を3.3VとしてHSPICEによるシミュレーションを行った。シミュレーション結果を表1に示す。

まず、NMOS-CSA、CMOSTG-CSA、HYBRID-CSAの比較を行う。遅延時間ではCMOSTG-CSAが最も良い性能を示した\*。しかしながら、面積、消費電力、ED積でHYBRID-CSAが最も良い性能を示し、NMOSとCMOS-TGの使い分けによって性能が向上することを示せた。

また、CMOS-CLA-1とCMOS-CLA-2ではED積で5%の差しかなく、面積でCMOS-CLA1が約30%小さかったため、CMOS-CLA-1を基準にCMOS回路とHYBRID-CSAの比較を行う。CMOS-CLA-1に比べ、提案手法で設計したHYBRID-CSAは同遅延時間でありながら、面積で77%、消費電力で69%、ED積で69%となり、提案手法の有効性が示せた。

一方、遅延時間のみを比較すると、パストランジスタ加算器で最も高速であったCMOSTG-CSAは、CMOS-CLA-2より約15%増加している。CMOSTG-CSAをED積の削減のためシングル・レール構成で設計したことが原因である。エネルギーではCMOSTG-CSAはCMOS-CLA-2より約36%削減されているため、エネルギーが同程度になるようにCMOSTG-CSAの電源電圧を上げれば、CMOS-CLA-2より遅延時間が短くなると見込まれる。

## 8. むすび

パストランジスタ論理に基づく加算器の設計手法を提案した。提案手法は、回路をBDDを用いて設計し、NMOSとCMOS-TGを使い分けることで回路の遅延と面積を削減する。さらに基本セルを組み合わせて複合セルを用いてレイアウトを行う。

提案手法を用いた設計事例として32ビット・キャリ・セレクト加算器を3つ設計した。NMOS、CMOS-TGとその2つを使い分けたHYBRIDである。HYBRID

\* NMOS-CSAがトランジスタ数、面積ともに小さいにもかかわらず、CMOSTG-CSAよりもエネルギーが大きくなったのは、NMOS-CSAのキャリ生成部の出力インバータのサイズを大きくしたので貫通電流が増加したためであると考えられる。

加算器が ED 積、面積ともに最も小さい結果となり、NMOS と CMOS-TG の使い分けによって性能が向上することを示せた。また、CMOS の 32 ビット・キャリルックアヘッド加算器と比較した結果、面積で 77%、ED 積で 69% という良好な結果を示した。

パストランジスタ論理を用いた他の設計事例として、文献 11) では 16 ビット冗長二進乗算器の設計し、評価している。

謝辞 本研究全般にご協力いただいたシャープ (株) の今井繁規氏、爲本博氏、小西健三氏、長澤和広氏に感謝いたします。

### 参 考 文 献

- 1) 李 副烈, 瀧 和男: パストランジスタ論理に基づく低消費電力回路方式, 電子情報通信学会技術研究報告, VLD95-115, Vol.95, No.421, pp.1-6 (1995).
- 2) 李 副烈, 瀧 和男, 田中秀樹: 低閾値 NMOS を用いるパストランジスタ論理回路方式, 電子情報通信学会技術研究報告, VLD96-73, Vol.96, No.425, pp.65-70 (1996).
- 3) Konishi, K., Kishimoto, S., Lee, B.-Y., Tanaka, H. and Taki, K.: A logic synthesis system for the pass-transistor logic SPL, *Proc. 6th SASIMI'96*, pp.32-39 (1996).
- 4) 小西健三, 岸本 悟, 李 副烈, 瀧 和男: パストランジスタ論理 SPL 用論理設計 CAD, DA シンポジウム'96 論文集, pp.13-18, 情報処理学会 (1996).
- 5) Kuroda, T. and Sakurai, T.: Overview of Low-Power ULSI Circuit Techniques, *IEICE Trans. Electronics*, Vol.E78-C, No.4, pp.334-344 (1995).
- 6) 李 副烈, 瀧 和男, 田中秀樹, 今井繁規: パストランジスタ論理 SPL とその設計事例, DA シンポジウム'96 論文集, pp.19-24, 情報処理学会 (1996).
- 7) 李 副烈, 宇田研一郎, 瀧 和男: NMOS および CMOS-TG パストランジスタ論理による加算器の設計と比較評価, DA シンポジウム'98 論文集, pp.65-70, 情報処理学会 (1998).
- 8) Makino, H., Nakase, Y., Suzuki, H., Morinaka, H., Shinohara, H. and Mashiko, K.: An 8.8-ns 54×54-Bit Multiplier with High Speed Redundant Binary Architecture, *IEEE J. Solid-State Circuits*, Vol.31, No.6, pp.773-783 (1996).
- 9) Ohkubo, N., Suzuki, M., Shinbo, T., Yamanaka, T., Shimizu, A., Sasaki, K. and Nakagome, Y.: A 4.4 ns CMOS 54×54-b Multiplier Using Pass-Transistor Multiplexer, *IEEE J. Solid-State Circuits*, Vol.30, No.3, pp.251-257 (1995).
- 10) Parameswar, A., Hara, H. and Sakurai, T.: A high-speed, low-power and swing restored pass-transistor logic based multiply and accumulate circuit for multimedia applications, *IEEE J. Solid-State Circuits*, Vol.31, No.6, pp.804-809 (1996).
- 11) 坂本 拓文, 宇田研一郎, 李 副烈, 越智裕之, 瀧和男, 津田孝夫: 低消費電力パストランジスタ論理 SPL を用いた 16 bit 冗長二進乗算器の設計と評価, DA シンポジウム'98 論文集, pp.179-184, 情報処理学会 (1998).
- 12) Sakurai, T. and Kuroda, T.: Low-power circuit design for multimedia CMOS VLSI's, *Proc. 6th SASIMI'96*, pp.3-10 (1996).
- 13) Sakurai, T., Lin, B. and Newton, A.R.: Multiple-Output Shared Transistor Logic (MOSTL) Family Synthesized using Binary Decision Diagram, Memo UCB/ERL M90/21, Dept. EECS, Univ. of Calif. Berkeley (1990).
- 14) Sasaki, Y., Rikino, K. and Yano, K.: ALPS: An Automatic Layouter for Pass-Transistor Cell Synthesis, *Proc. 3rd ASP-DAC'98* (1998).
- 15) 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. IEEE Symp. on Low Power Electron.*, pp.14-15 (1995).
- 16) Suzuki, M., Ohkubo, N., Shinbo, T., Yamanaka, T., Shimizu, A., Sasaki, K. and Nakagome, Y.: A 1.5-ns 32-b CMOS ALU in double pass-transistor logic, *IEEE J. Solid-State Circuits*, Vol.28, No.11, pp.1145-1151 (1993).
- 17) 瀧 和男, 李 副烈: パストランジスタ論理に基づく低消費電力回路方式と設計事例, 電子情報通信学会論文誌, Vol.J80-A, No.5, pp.753-764 (1997).
- 18) 田中秀樹, 李 副烈, 瀧 和男, 宇田研一郎, 北村清志, 村田 豊: パストランジスタ論理のためのセル開発とその応用, 情報処理学会 DA 技術研究報告, DA97-86, Vol.97, No.119, pp.25-32 (1997).
- 19) Yano, K., et al.: Lean Integration: Achieving a Quantum Leap in Performance and Cost of Logic LSIs, *Proc. Custom Integrated Circuits Conference*, pp.603-604 (1994).
- 20) Yano, K., Sasaki, Y., Rikino, K. and Seki, K.: Top down pass-transistor logic design, *IEEE J. Solid-State Circuits*, Vol.31, No.6, pp.792-803 (1996).
- 21) Yano, K., Yamanaka, T., Nishida, T., Saito, M., Shimohigashi, K. and Shimizu, A.: A 3.8-ns CMOS 16×16-b multiplier using complementary pass-transistor logic, *IEEE J. Solid-State*



- Circuits*, Vol.25, No.2, pp.388-395 (1990).
- 22) 石浦菜岐佐: BDDとは, 情報処理, Vol.34, No.5, pp.585-592 (1993).
- 23) 日経マイクロデバイス(編): 低電力 LSI の技術白書—1 ミリ・ワットへ挑戦, 日経 BP 社 (1994).
- 24) 瀧 和男: 低電力デジタル VLSI を支えるパストランジスタ技術, 低消費電力, 高速 LSI 技術, リアライズ社, pp.349-370 (1998).

(平成 10 年 9 月 24 日受付)

(平成 11 年 2 月 8 日採録)



李 副烈 (学生会員)

昭和 46 年生。平成 6 年神戸大学工学部システム工学科卒業。平成 8 年同大学院博士前期課程情報知能工学修了。現在, 同博士後期課程在学中。LSI の低消費電力化技術に興味を持つ。平成 10 年山下記念研究賞授賞。電子情報通信学会会員。



瀧 和男 (正会員)

昭和 27 年生。昭和 51 年神戸大学工学部電子工学科卒業。昭和 54 年同大学院修士課程システム工学修了。工学博士。同年(株)日立製作所入社。昭和 57 年(財)新世代コンピュータ技術開発機構に出向。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。平成 2 年同機構第 1 研究室室長。平成 4 年 9 月神戸大学工学部情報知能工学科助教授。平成 7 年 4 月同学科教授。LSI 設計技術と CAD, 並列処理とマシンアーキテクチャ, 脳型コンピュータ等に興味を持つ。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。



宇田研一郎

昭和 49 年生。平成 9 年神戸大学工学部情報知能工学科卒業。現在, 同大学院博士前期課程情報知能工学在学中。プロセッサの低消費電力化に興味を持つ。電子情報通信学会会員。



北村 清志

昭和 49 年生。平成 9 年神戸大学工学部情報知能工学科卒業。現在, 同大学院博士前期課程情報知能工学在学中。レイアウト手法に興味を持つ。