

BDD 形状を考慮したパス・トランジスタ論理セルによる 低消費電力回路レイアウト方式

福 岡 一 樹[†] 沼 昌 宏[†]
山 本 啓 輔[†] 金 銅 恒^{††}

低消費電力化に有効とされるパス・トランジスタ論理のレイアウトに関して、トランジスタが多列配置された専用セル生成に基づくライブラリ・フリーのレイアウト方式を採用することで、これまで課題となっていた配線の複雑度を軽減し、消費電力と遅延時間を削減する方式を提案する。従来よりも大きな部分回路をセルとして生成するとともに、実現すべき論理関数を表現する BDD の形状を考慮したトランジスタ配置を導入する点を特徴とする。ベンチマーク回路と乗算器の Wallace Tree 部に関して、従来方式ならびに CMOS 論理と比較実験を行った結果、エネルギー・遅延積 (ED 積) の点で優れた結果を得た。

A Layout Design Approach for Low Power Circuits Using On-the-fly Generation of Pass Transistor Logic Cells Based on BDD Structure

KAZUKI FUKUOKA,[†] MASAHIRO NUMA,[†] KEISUKE YAMAMOTO[†]
and HISASHI KONDOH^{††}

We present a layout design approach for pass-transistor logic circuits with lower power and delay by reducing wiring complexity based on a library-free approach and multiple row placement style for pass-transistors in the cells. This approach has two features: on-the-fly generation of dedicated cells of sizes larger than conventional predefined cells, and placement of transistors in the cells based on the structure of BDD for the given logic function. Experimental results for benchmark circuits and a Wallace tree for 16-bit multiplier have shown lower energy-delay products than conventional approaches including CMOS logic circuits.

1. はじめに

LSI 設計において、動作周波数の向上、回路の大規模化などによる消費電力の増加が大きな問題となっている。そのため、様々な消費電力削減のための手法が提案されている。なかでもパス・トランジスタ論理は、とくに XOR, MUX 関数などについて CMOS 論理よりも少ないトランジスタ数で回路を構成できる点で、低消費電力化、高速化などの性能向上効果が期待されている¹⁾。これはパス・トランジスタ論理が、一般的に XOR, MUX 関数の表現に有利な BDD^{2),3)}に基づいて合成される⁴⁾ことと関係している。しかし、すべての関数にパス・トランジスタ論理が適しているわけ

ではなく、AND/OR 関数に関しては CMOS 論理での実現が望ましい。そこで、実現すべき論理関数の性質を考慮して AND/OR 関数部分には CMOS 論理を適用し、それ以外の部分にはパス・トランジスタ論理を適用する CMOS/パス・トランジスタ混在論理^{5),6)}が提案されている。

その一方で、パス・トランジスタ論理に基づく回路のレイアウトに一般的なスタンダード・セル方式のレイアウト・ツールを適用するために、パス・トランジスタ論理 LEAP⁷⁾を対象とする ALPS (Automatic Layout-outer for Pass-transistor cell Synthesis)⁸⁾や、SPL⁹⁾を対象としたセル・ライブラリ¹⁰⁾などのように、パス・トランジスタ論理セル・ライブラリを利用する方式が提案されている。

BDD に基づくパス・トランジスタ論理合成手法⁴⁾の特徴は、BDD の各ノードに接続する 2 本のエッジをそれぞれトランジスタに置換することで、回路を生成

[†] 神戸大学工学部

Faculty of Engineering, Kobe University

^{††} 川崎マイクロエレクトロニクス株式会社

Kawasaki Microelectronics, Inc.

する点にある．そのため，パス・トランジスタのソースとドレインを接続する配線は BDD のノード間接続形態の影響を大きく受け，回路によって多様な形態をとる．しかし，BDD の多様な接続形態に合わせたパス・トランジスタ論理セルをすべて用意するのは困難であるため，従来は 1～数個程度のノードに対応する汎用的な小規模セルからなるセル・ライブラリを適用していた．その結果，セル割当ての際に BDD を数個のノードごとに細かく分割する必要性からセル間配線が増加し，消費電力，遅延が CMOS 論理よりかえって増える場合があった．我々も上記の方法に基づきパス・トランジスタ論理セル・ライブラリ⁶⁾を構築したが，セル間配線の混雑によって，遅延が増加する場合があった．

プロセスの微細化によって，素子遅延よりも配線遅延の割合が大きくなっている．そこで，配線遅延がクリティカルになる部分では，従来の汎用的な小規模セルからなるライブラリを利用するのではなく，設計する回路に応じて専用セルを生成することにより，遅延の増加を防ぐライブラリ・フリーのレイアウト方式が注目されている．パス・トランジスタ論理についても，ライブラリ・フリーのレイアウト方式¹¹⁾が提案されている．しかしこの手法では，1つの入力変数に対応する BDD ノードを通常は 2個に制限してセル生成を行うため，回路分割により BDD を多段化する必要がある．その結果，挿入すべきバッファ数が増加し，トランジスタ数の増加が懸念される．

本論文ではこれらの問題を解決するために，BDD の形状を考慮した新たなパス・トランジスタ論理回路のレイアウト方式を提案し，その適用例を示すとともに，ED 積，総配線長，面積の各点から提案方式の評価を行う．

2. BDD の形状を考慮したパス・トランジスタ論理セルの生成

本章では提案方式によるセル生成について述べる．本論文では規約な順序付き共有二分決定グラフ (SROBDD)¹²⁾を扱い，これを単に BDD と呼ぶ．その定義¹³⁾を次に示す．

定義 1 $B = \{0, 1\}$ に対して，BDD は次の 7 つ組 $B = (X, N_V, N_C, I, e^0, e^1, lev)$ で表される．

$X = \{0, 1, \dots, n-1\}$: 変数インデックスの集合

N_V : 変数ノードの集合

$N_C = \{c_0, c_1\}$: 定数ノードの集合

$I = \{i_1, \dots, i_m\} \subset (N_V \cup N_C)$: 初期ノードの集合

$e^0, e^1 : N_V \rightarrow (N_V \cup N_C)$: 変数ノードの 0 エッジ，

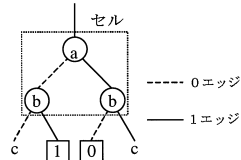


図 1 従来のパス・トランジスタ論理セルと入力信号例
Fig. 1 Conventional PTL cell.

1 エッジ

$lev : (N_V \cup N_C) \rightarrow (X \cup \{n\})$: ノードのレベルを示す．変数ノード $v \in N_V$ に対応する変数インデックスを $inx(v)$ で表すとき，以下が成立する：

$lev(v) = inx(v)$ iff $v \in N_V$, $lev(v) = n$ iff $v \in N_C$,
 $lev(u) < lev(e^0(u))$, $lev(u) < lev(e^1(u))$. □

従来のパス・トランジスタ論理を対象とするセル・ライブラリの多くは，BDD の 1 ノードに対応する 2 入力 1 出力セレクタ (2-1MUX) を 1～数個含むセルを用意している．図 1 に，このようなパス・トランジスタ論理セルで実現される部分的な BDD と，その信号入力の例を示す．従来のパス・トランジスタ論理を対象とするセル・ライブラリでは，用意すべきセルの数を抑えるため，図 1 のように 4 本の独立したパス入力を設けることが一般的である．しかし，図 1 の場合，パス入力に共通の信号 c が含まれるため，この部分をセル内で接続することでセル間配線が減らすことができる．さらに，トランジスタの拡散共有を行うことにより，セル内配線自体を減らすことも可能である．しかし，入力線の各種接続形態に応じたセルをあらかじめ用意することは，膨大な数のセルが必要となる点で現実的ではない．

そこで提案方式では，次の特徴によって問題点を解決する．

- i) ライブラリ・フリーのレイアウト方式
- ii) 多列配置方式の採用
- iii) BDD の描写形状の考慮による配線の混雑度軽減

i) について，提案方式では回路に応じて専用セルを生成するため，膨大な数のセルを用意することなく入力の接続形態に応じた様々なセルを生成可能となる．

ii) によって，図 2 に示すように BDD の描写形状を考慮したレイアウトを実現する．描写の決定した同一レベルの BDD ノードに対応するトランジスタを，水平方向に配置する．さらに，BDD の各レベルに対応するトランジスタ列を最大 n_{row} 段までセル内に縦積みする．パス・トランジスタ論理では，ドライブ能力はバッファによって決定されるので，論理を構成するパス・トランジスタについてはサイズを小さくする

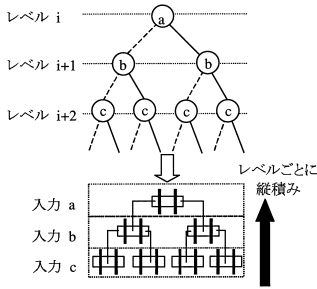


図2 提案方式の概念

Fig. 2 Concept of proposed approach.

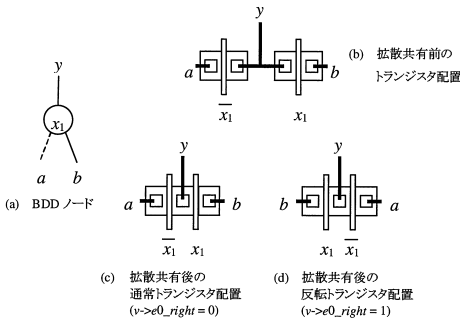


図3 同一ノードに属するトランジスタ間での拡散共有

Fig. 3 Sharing diffusion between transistors for BDD node.

ことが可能となり、縦積み構造に適している。もし各レベルに対応するトランジスタを垂直に配置すると、セルの高さによる制限によって効率良く配置できないが、水平に並べることにより、1レベルあたりのトランジスタ数に関する制約を大きく緩和して配置することが可能である。

iii) について、パス・トランジスタ論理におけるトランジスタ間の接続は、BDDにおけるノード間接続に対応する。そこで、BDDのノード間接続を考慮して、トランジスタ間配線が複雑にならない範囲でトランジスタ間の拡散共有がより多く行われるように描写を決定する。ただし、描写にあたり、同一レベルのノードはすべて同じ高さに描写する。同一レベルでのノード位置の決定法については3.1節で示す。

パス・トランジスタ論理において拡散共有が可能な箇所は、BDDにおけるエッジの接続関係より決定される。この結果に基づいてBDDの描写段階で拡散共有を決定する。本方式では、次の2種類について拡散共有を行う。

(1) 同一ノードでの拡散共有

図3(a)に示すようなBDDのノードは、2-1MUXに置換される。図3(b)に示すように、nMOSのドレインどうしが出力信号で結ばれることになり、拡散共

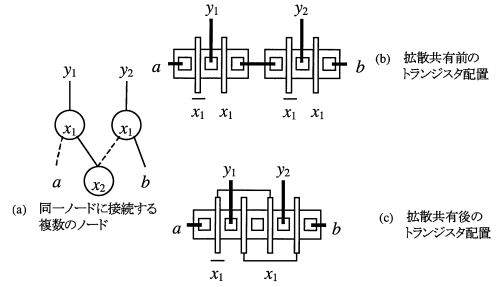


図4 異なるノード間での拡散共有

Fig. 4 Sharing diffusion between transistors for different BDD nodes.

有が可能となる。図3(c), (d)に示すように、この拡散共有を行った2つのトランジスタのレイアウトを配置における基本単位とする。図3(c)を通常トランジスタ配置、図3(d)を反転トランジスタ配置と呼び、通常は図3(c)を用いる。

(2) 異なるノード間での拡散共有

図4に示すように、異なるノード間であっても2本のエッジが同一ノードに接続する場合、この部分は共有可能である。このとき、以下の3.1節で述べるように、必要に応じて図3(d)の反転配置を適用して拡散共有を可能にする。

なお本方式では、CMOS/パス・トランジスタ混在論理に対応するために、パス・トランジスタ論理セルを既存のCMOS論理セルの高さに合わせる。そのため、セル内トランジスタ列の最大列数 n_{row} をCMOS論理セルの高さに応じて適宜設定する。具体的な設定例については、5章で述べる。

3. 提案方式によるレイアウト手順

提案方式を用いた設計フローを図5に示す。まず回路記述より、論理合成時にCMOS論理部分とパス・トランジスタ論理部分を選定する。CMOS論理と混在させずSPLのみの合成を行う場合は、インバータ、電圧レベル回復用バッファのみがCMOS論理部となる。パス・トランジスタ論理部については、BDDを構築するとともに、DVO^[14]、共通部分の共有により、BDDのノード数を小さく抑える。さらに提案方式に基づくセルへの割当てを行い、生成するセルを含むネットリストを生成する。そしてシンボリック・レイアウト生成処理を行い、セル・テンプレートを生成する。ライブラリ・フリー方式では、回路ごとに専用のセルを生成するために、セル生成時間の短縮やテクノロジー間での移植容易性が求められる。これらの要求を満たすた

選定の手法については文献5), 6)を参照。

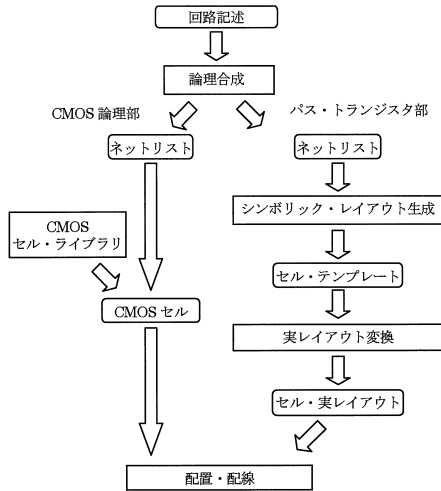


図 5 提案方式を用いた設計フロー
Fig. 5 Design flow with proposed approach.

BDD 形状の決定
中間インバータの挿入
セルへの割当て
トランジスタ配置
トランジスタ間の配線

図 6 シンボリック・レイアウト生成処理
Fig. 6 Symbolic layout process.

めに、シンボリック・レイアウトを適用する。シンボリック・レイアウトから、実レイアウトへの変換には VARDS¹⁵⁾を用いる。VARDSにおけるトランジスタ配置は二次元配置であったが、これを多列配置方式に対応させて適用する。パス・トランジスタ論理セルの実レイアウトが完了すると、CMOSセルとともに市販のレイアウトツールを用いて配置・配線を行う。

提案方式の実現に密接に関わる処理として、論理合成時の BDD 構築後からセル・テンプレート生成までの処理を図 6 に示す。以下、これらの処理について述べる。

3.1 BDD 形状の決定

まず、トランジスタの配置と完全に対応するような描写における BDD 形状を一意に決定する。本方式では、トランジスタ間配線が複雑にならない範囲でトランジスタ間の拡散共有がより多く行われるように、各レベルにおけるノードの相対位置を決定する。

BDD 形状の決定手法を図 7 に示す。set_BDD_structure 関数で示されるように 2 つの処理からなる。まず、set_node_order 関数により定数ノードに近い出力変数から順に深さ優先で探索を行う。0 エッジから探索を行い、訪れたノード v から順に、BDD の各レ

```

set_BDD_structure(B)
{
    foreach (v ∈ B -> I)
        set_node_order(v);
    set_edge_side();
}

set_node_order(v) /* 外部出力からの深さ優先探索により各レベルの BDD ノード順序を決定 */
{
    f0 = e0(v); /* 0 エッジの接続するノード */
    f1 = e1(v); /* 1 エッジの接続するノード */
    add_node_list((lev(v), v) /* ノードをレベルごとのリスト lev_list[lev(v)] の末尾に追加 */
    if (f0 を訪れていない) && f0 はバス変数、定数ノードでない)
        set_node_order(f0);
    if (f1 を訪れていない) && f1 はバス変数、定数ノードでない)
        set_node_order(f1);
}

set_edge_side() /* 拡散共有が多くなるように 0, 1 エッジの位置関係を決定 */
{
    for (level = n; -level >= 0;){ /* 各 BDD レベル level について */
        prev_right = NULL; /* 直前に着目したノードの右側エッジが指すノード */
        for (list = lev_list[level]; list != NULL; list = next) {
            v = list->v; /* 着目するノード */
            next = list->next; /* リスト次要素 */
            if (prev_right == e1(v) ||
                (next != NULL && (e0(v) == e0(next->v) || e0(v) == e1(next->v))))
                v->e0_right = 1;
            prev_right = v->e0_right ? e0(v) : e1(v);
        }
    }
}
    
```

図 7 描写における BDD 形状の決定
Fig. 7 Decision of BDD structure for drawing.

ベル単位で管理されたノード・リスト lev_list[lev(v)]の最後に加える。この操作により、同一レベルにおける隣接ノードとして最も近い先祖を持つノードをリストに加える。

さらに、拡散共有が多くなる場合は、図 3 (c) の通常トランジスタ配置を図 3 (d) の反転配置に変更する。図 7 における set_edge_side 関数がこの処理を行う。通常のトランジスタ配置では $v->e0_right = 0$ 、反転したトランジスタ配置では $v->e0_right = 1$ に設定する。

この方法により、描写時に BDD のエッジが他のエッジと交差することを少なくする。BDD レベルごとに拡散共有を最大とする配置を決定することも可能ではあるが、レベル間の関係を考慮しないと、回路によっては交差が多くなる場合がある。各レベルごとに得られたノード・リストに対応してトランジスタ列を配置することにより、列間配線が複雑になることを防ぐ効果がある。またこれらのノードでは、エッジが同一ノードに接続することが多い傾向にあるため、拡散共有を多くすることができる。ただし、探索時の出力変数順や 0 エッジと 1 エッジのいずれかを先に探索するかによって、描写形状が異なる場合がある。エッジ交差数の最小化と拡散共有の最大化については、さらなる改善の余地が残されている。

3.2 中間インバータの挿入

パス・トランジスタとして nMOS トランジスタのみを利用する SPL では、‘H’ 電圧レベルの回復のため、図 8 (a) に示す中間バッファを挿入する。本方式では

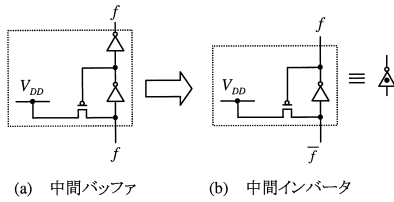


図 8 中間バッファから中間インバータへの変更
Fig. 8 Intermediate inverter.

中間バッファの代わりに、トランジスタ数が 2 個少ない中間インバータの利用を可能としている。そのために、図 8 (b) に示すように直前の論理 f を反転させた \bar{f} を生成する。BDD におけるあるノードでの論理反転は、 v を根とするグラフに存在する c_0 と c_1 をすべて入れ替えることを意味する。同型部分が共有された BDD においては、同じ先祖を持つ同レベルのノードにおける論理をすべて反転させる必要がある。

本方式では、バッファ挿入をセル境界に設定することで同レベルにおける論理反転を可能としているため、中間インバータを利用することができる。

3.3 セルへの割当て、トランジスタ配置

本方式は、CMOS/パス・トランジスタ混在論理のレイアウトに対応するため、既存の CMOS 論理セルに生成するパス・トランジスタ論理セルの高さを合わせる。本方式では多列配置方式の採用により、CMOS 論理セルの高さに柔軟に対応できる。

2 章 (1), (2) に従い、トランジスタを列ごとに水平方向に配置する。セルに含むべきトランジスタ列が n_{row} に満たない場合は、出力側に接続する出力インバータのプルアップ用 pMOS トランジスタを配置し、面積効率を向上させる。

入力インバータの扱いについては、小さなトランジスタ・サイズのインバータをセル内に付加する形と、必要最小限の入力インバータ・セルを付加する形のレイアウトが存在する。本方式では、セル内に多く含むパス・トランジスタの数に応じて柔軟にインバータ・サイズを変更できるように、後者を採用する。入力インバータを別のセルとして扱うため、プルアップ用 pMOS トランジスタを除くと、nMOS トランジスタのみがセル内に配置される。

3.4 トランジスタ間の配線

ゲートどうしの接続については、ポリシリコンでトランジスタの上下を櫛歯状に配線する。多列配置方式を採用しているため、トランジスタ列間の配線については、チャンネル配線を適用する。

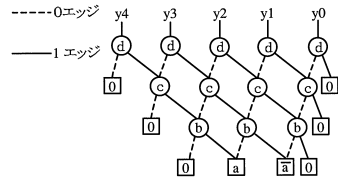


図 9 4 入力 tally 回路の BDD 表現
Fig. 9 4-input tally circuit.

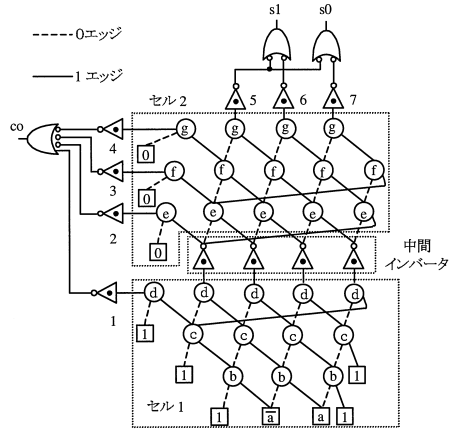


図 10 7-3BSA の BDD ノードを用いた表現とセル割当て
Fig. 10 Cell assignment for 7-3 BSA.

4. 本方式の適用例

提案方式を tally 回路¹⁶⁾に基づく 7 入力 3 出力ビット・スライスアダー (7-3BSA) の設計へ適用した例について述べる。

Tally 回路は、論理値 “1” をとる入力数を出力する。4 入力 tally 回路の BDD 表現を図 9 に示す。この回路を 2 つ組み合わせて 7-3BSA を設計した。その BDD ノードを用いた表現とセルへの割当てを図 10 に示す。BDD の形状決定後、中間インバータを挿入している。図中に BDD 形状決定の際の探索の際の出力変数順を示している。図において同一レベルでのノードの相対的な位置がトランジスタの配置と対応する。また、同一ノードでの 0, 1 エッジの位置関係が相対的なトランジスタ配置を表す。すなわち、0 エッジが左なら通常のトランジスタ配置、右なら反転したトランジスタ配置となる。この場合、隣接するすべてのノード間で拡散共有が可能である。よって、提案方式に用いた BDD 形状の決定法が単純ながら良好な結果を示していることが確認できる。

Tally 回路は BDD の接続関係が各ノードでほぼ同型になるという特徴を持ち、各レベルでのノード数のばらつきが少ない。このような BDD に対して提案方式を適用することにより、コンパクトなレイアウトを

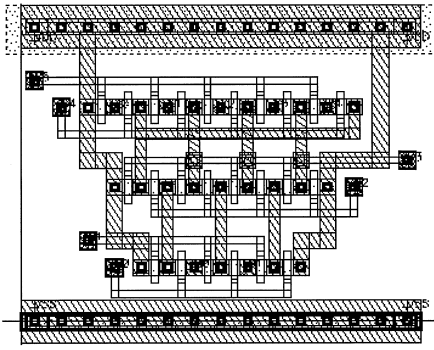


図 11 7-3BSA を構成するセル
Fig. 11 Cell layout for 7-3 BSA.

実現することができる。

図 10 におけるセル 1 のレイアウトを図 11 に示す。セル内の各トランジスタ列における拡散共有によって、コンパクトな配置が可能となった。このようなセルを乗算器の Wallace Tree 部に用いて低消費電力化を図る。

5. 実験評価

提案方式を C および C++ 言語 (gcc ver. 2.95.3) を用いて計算機 (PentiumIII 600 MHz Linux) 上に実装するとともに、パス・トランジスタ論理 SPL, ならびに CMOS/パス・トランジスタ混在論理のパス・トランジスタ論理部に適用し、0.35 μm プロセスでレイアウト設計を行った。最終的な配置配線結果より LPE を行い、HSPICE により 100 パターンの平均消費電力と HSPICE, PathMill により最大遅延時間を算出し、ED 積にて評価を行った。実験では、CMOS/パス・トランジスタ混在論理の CMOS 部分には、既存の CMOS セルを用いた。そしてトランジスタ列の最大段数 $n_{row} = 3$ と設定した。

仮想グリッド上において、 n_{row} とセルの高さ $cell_H$ が満たすべき関数は

$$cell_H > \sum_{i=1}^{n_{row}} row_i + \sum_{j=1}^{n_{row}-1} ch_j + reg_{Gnd} + reg_{vdd} \quad (1)$$

で表される。ここで、 $cell_H$ はセルの GND ラインから VDD ラインまでのグリッド数、 row_i は、各トランジスタ列が占めるグリッド数、 ch_j は、各配線領域の占めるグリッド数、 reg_{Gnd} , reg_{vdd} は、GND ライン、VDD ラインとトランジスタ間のグリッド数を示す。

適用した CMOS セルライブラリでは、 $cell_H = 11$

である。低消費電力化のためにパス・トランジスタのサイズを小さくするため $row_i = 1$ とする。各配線領域のグリッド数としては、配線の交差を考えたときの最小値 $ch_j = 2$ とした。そして、最小値として $reg_{Gnd} = reg_{vdd} = 1$ とすると、式 (1) より $n_{row} \leq 3$ の条件が導かれるため、許容される最大値 $n_{row} = 3$ に設定した。もし $n_{row} = 2$ とした場合、生成するセル全体の面積が増加することや、セル間配線の増加が懸念される。また、 $cell_H \geq 13$ の CMOS セルライブラリを適用する場合は、 $n_{row} \geq 4$ とすることが可能であるが、レベルごとのノード数が不均一になり、セル内に冗長な面積が増加する可能性を考慮する必要がある。

実験対象回路として MCNC LGSynth91 ベンチマーク回路の Combinational Multi-Level Examples¹⁷⁾の中で単純直交分解可能な回路のうち 10 例と、16 ビット乗算器の Wallace Tree 部を用いた。提案方式をパス・トランジスタ論理と CMOS/パス・トランジスタ混在論理に適用した結果について述べる。5.1 節では、従来の汎用セル・ライブラリ (従来方式) を用いた結果と比較することにより、提案レイアウト方式を評価した。5.2 節では CMOS 論理と比較することによって、回路構成方式を含めた評価を行った。5.3 節では、4 章で示したような実用的な回路を乗算器に適用した場合の評価を行った。

5.1 従来のパス・トランジスタ論理セル・ライブラリによるレイアウトとの比較

まず、従来方式との比較により提案方式を用いたことによるパス・トランジスタ論理および CMOS/パス・トランジスタ混在論理の性能向上の効果を示す。表 1 に拡散共有数、総配線長、面積における従来方式と提案方式の結果を示す。「SPL」、「混在論理」は、それぞれパス・トランジスタ論理 SPL, CMOS/パス・トランジスタ混在論理への適用結果を表す。拡散共有数はパス・トランジスタ部におけるものを示し、処理時間は BDD 構築からレイアウト生成までの時間である。拡散共有数に関しては、SPL で 26%、混在論理で 15% 増加させることができた。この結果は拡散共有の増加分に相当する配線を削減することができたことを示す。混在論理に関して 2 例の回路で拡散共有数が減少しているが、これは CMOS 部分の選定により BDD の構造が変化したためである。総配線長については、SPL で 20%、混在論理で 22% 削減できた。総配線長が増加している回路についてその要因として、面積の増加と、BDD エッジのあるレベルへの集中による配線の混雑が考えられる。面積に関しては、SPL で 8%、混

表 1 拡散共有数, 総配線長, 面積, 処理時間に関する結果

Table 1 Count of sharing diffusion, global wire length, area and CPU time.

回路名	拡散共有数 (従来比)		総配線長 [μm] (従来比)		面積 [μm^2] (従来比)		処理時間 [s]	
	SPL	混在論理	SPL	混在論理	SPL	混在論理	SPL	混在論理
b1	3 (1.50)	1 (1.00)	74.4 (0.63)	85.5 (0.96)	322 (0.93)	297 (0.92)	2.07	1.96
cc	51 (1.55)	28 (1.47)	2402.0 (0.79)	2856.4 (1.13)	4554 (0.90)	5198 (1.19)	16.44	6.93
cm151a	8 (1.00)	8 (1.00)	277.7 (0.59)	277.7 (0.52)	990 (0.83)	990 (0.83)	2.54	2.60
cm163a	24 (1.04)	26 (1.86)	1552.0 (0.96)	1517.6 (1.01)	2525 (0.87)	3366 (1.26)	5.77	3.14
cm85a	46 (1.18)	42 (1.27)	1499.7 (1.09)	1417.8 (0.96)	3104 (1.05)	3564 (1.23)	4.84	2.99
pcle	58 (1.53)	36 (1.24)	3244.9 (1.22)	2178.4 (0.64)	5816 (0.85)	4356 (0.86)	8.25	7.14
pm1	49 (1.14)	6 (0.75)	1861.1 (0.68)	1117.0 (0.62)	4703 (0.95)	2747 (0.97)	7.93	3.16
t481	35 (1.17)	11 (0.79)	1079.2 (0.60)	622.9 (0.60)	3366 (1.23)	2327 (0.92)	7.51	2.85
x2	38 (1.27)	18 (1.29)	1275.9 (0.83)	1470.8 (0.88)	2574 (0.85)	2673 (0.97)	5.69	3.40
z4ml	34 (1.42)	29 (1.21)	696.9 (0.80)	771.3 (0.78)	1559 (0.83)	1856 (0.78)	3.58	4.45
幾何平均	26.5 (1.26)	14.1 (1.15)	960.6 (0.80)	870.8 (0.78)	2303 (0.92)	2190 (0.98)	5.47	3.56

表 2 ED 積の従来法に対する結果

Table 2 Comparison of ED products with conventional approach.

回路名	SPL			混在論理		
	従来	提案	提案 従来	従来	提案	提案 従来
b1	1.03	0.56	0.55	1.06	0.75	0.71
cc	14.85	11.47	0.77	17.28	11.03	0.64
cm151a	3.94	3.99	1.01	4.09	3.99	0.98
cm163a	10.11	9.61	0.95	10.48	8.19	0.78
cm85a	26.50	8.51	0.32	23.75	15.61	0.66
pcle	24.12	22.15	0.92	28.02	10.54	0.38
pm1	15.82	11.77	0.74	6.66	5.58	0.84
t481	155.55	25.16	0.16	8.79	9.06	1.03
x2	7.86	14.77	1.88	12.66	6.74	0.53
z4ml	14.86	11.48	0.77	9.05	7.67	0.85
幾何平均	12.73	8.64	0.68	9.00	6.41	0.71

在論理で 2% の削減にとどまった。回路によってはセルに割り当てた BDD レベル間でノード数が大きく異なったため、冗長な面積を生じる場合がある。

さらに、単位面積あたりの平均配線長、すなわち配線の混雑度を考えると、回路 10 例の幾何平均で SPL で 13%、混在論理で 21% の削減効果が得られている。処理時間に関しては、すべての回路について実用的な時間で処理が行われている。シンボリックレイアウトを用いることにより、プロセスに依存せず提案方式の処理を短時間で行うことが可能となった。

表 2 に従来方式と、提案方式によるレイアウトでの ED 積に関する結果を示す。表 2 に示した ED 積に関する幾何平均に関して従来方式と比べると、SPL への適用例について 32%、混在論理への適用例については 29% 削減する結果が得られた。その要因として、BDD の描写形状を考慮してトランジスタを配置した専用セルの導入によってコンパクトなレイアウトが実現され、トランジスタの多列配置、拡散共有によって配線の多くをセル内に納めた結果、セル間配線が削減されたためと考えられる。SPL での合成における回路例 x2 の ED 積の増加は、バッファのプルアップ pMOS

表 3 ED 積, 総配線長, 面積の CMOS 論理に対する比較

Table 3 Comparison of ED products, wire length and area with CMOS.

回路名	ED 積比		総配線長比		面積比	
	SPL	混在論理	SPL	混在論理	SPL	混在論理
b1	0.96	1.29	0.88	1.01	1.08	1.00
cc	0.69	0.67	1.11	1.32	1.24	1.41
cm151a	0.70	0.70	0.61	0.61	0.80	0.80
cm163a	0.82	0.70	1.33	1.30	0.82	1.09
cm85a	0.40	0.74	1.40	1.32	0.93	1.07
pcle	0.38	0.18	1.27	0.85	1.01	0.76
pm1	1.44	0.68	1.19	0.71	1.46	0.85
t481	3.19	1.15	1.35	0.78	1.74	1.21
x2	2.03	0.93	0.85	0.98	0.79	0.82
z4ml	0.75	0.50	1.13	1.25	0.73	0.87
幾何平均	0.91	0.68	1.08	0.98	1.02	0.97

のサイズによる問題であり、トランジスタ長を 2 倍にすることで 7.91 とほぼ同等の ED 積に改善できることが確認できた。

5.2 CMOS 論理との比較

CMOS 論理との比較により、提案したレイアウト方式のみならず、回路構成方式を含めた評価を行う。表 3 に、CMOS 論理との ED 積、総配線長、面積に関する比較結果を示す。SPL への適用例については幾何平均で 9% の削減効果が、混在論理への適用例に関しては、幾何平均で 32% の削減効果があった。さらに、混在論理における pcle のように 82% と大きな削減効果が得られた回路が存在する。このことから提案方式が CMOS 論理との比較においても ED 積削減に効果的であることが確認できた。

総配線長に関しては、SPL では 8% 増加しているが、混在論理では、2% 削減することができ、提案方式によってほぼ同等の結果が得られた。

面積に関しては、SPL で 2% 増加、混在論理で 3% の削減となり、ほぼ CMOS と同等の結果ということが

表4 乗算器用セルの性能

Table 4 Power and delay of multiplier's cells.

セル名	消費電力 [μ W]			遅延時間 [ns]		
	CMOS	提案	提案 CMOS	CMOS	提案	提案 CMOS
Half Adder	20.45	15.74	0.77	0.37	0.23	0.62
Full adder	20.01	18.49	0.92	0.69	0.62	0.90
7-3 BSA	63.56	52.24	0.82	1.76	1.71	0.97

表5 16 bit 乗算器 (Wallace Tree 部) に対する結果

Table 5 Power and delay of Wallace Tree.

ブロック名	消費電力 [mW]			遅延時間 [ns]		
	CMOS	提案	提案 CMOS	CMOS	提案	提案 CMOS
Wallace	1.57	1.29	0.82	2.91	3.12	1.07

できる。

5.3 乗算器に関する結果

乗算器に対しての結果により実用的な回路に対しての提案方式の性能を検討する。まず、乗算器用に生成したセルの評価を表4に示す。Half Adder, Full Adderについてはライブラリ中のセルとの比較、そして7-3BSAはCMOS Full Adderセルを4つ用いて作成したものと比較である。提案セルはSPL、およびCMOS/パス・トランジスタ混在論理で設計した。セル単体としては、消費電力、遅延時間ともにCMOS論理セルと比較して削減が可能となった。

次にWallace Tree部に7-3BSAを適用した結果を表5に示す。遅延時間は7%増加するものの、消費電力は18%削減され、ED積では12%の削減効果が得られた。さらに遅延の短縮するためにセルのサイジングや配置方法について検討することが考えられる。

6. まとめ

パス・トランジスタ論理において、BDD形状を考慮することにより、消費電力、遅延時間を削減するレイアウト方式を提案した。ライブラリ・フリーのレイアウト方式とトランジスタの多列配置方式を採用し、配線を複雑にすることなく拡散共有を行うことで配線の削減を図った。

本方式により、従来のパス・トランジスタ論理セル・ライブラリを利用する場合とED積を比較した結果、パス・トランジスタ論理SPLへの適用では32%、CMOS/パス・トランジスタ混在論理への適用例では、29%の削減効果が得られた。さらにCMOS論理に対しては、SPLへの適用例でED積を9%、CMOS/パス・トランジスタ混在論理への適用例で32%削減した。また、乗算器のWallace Tree部については、ED積を12%削減する効果が得られた。

今後の課題として、BDD形状決定手法の改善やマ

ルチハイト・セルの利用による、さらなる配線の効率化があげられる。

謝辞 ライブラリ・フリーのレイアウト方式についてご教示をいただきました東京大学の藤田昌宏教授に深く感謝いたします。

VARDSをご提供くださった京都大学の小野寺秀俊教授、橋本昌宜助手に深く感謝いたします。

参考文献

- 1) 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).
- 2) Akers, S.B.: Binary decision diagrams, *IEEE Trans. Computers*, Vol.C-27, No.6, pp.509-516 (1978).
- 3) Brayant, R.E.: Graph-based algorithms for Boolean function manipulation, *IEEE Trans. Computers*, Vol.C-35, No.8, pp.677-691 (1986).
- 4) 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).
- 5) 高田賢吾, 神野元彰, 黒木修隆, 沼 昌宏, 瀧和男, 山本啓輔: 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, 情報処理学会論文誌, Vol.42, No.4, pp.967-974 (2001).
- 6) 高田賢吾, 福岡一樹, 藤田克也, 福富弘敦, 黒木修隆, 沼 昌宏, 瀧和男, 山本啓輔: CMOS/パス・トランジスタ混在論理の合成とレイアウト, 情報処理学会論文誌, Vol.43, No.5, pp.1357-1360 (2002).
- 7) Yano, K., Sasaki, Y. and Seki, K.: Top-down pass-transistor logic, *IEEE Journal of Solid State Circuits*, Vol.31, No.6, pp.792-803 (1996).
- 8) Sasaki, Y., Rikino, K. and Yano, K.: ALPS: An automatic layouter for pass-transistor cell synthesis, *ASPDAC '98*, pp.227-232 (1998).
- 9) 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).
- 10) 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, 情報処理学会研究報告, Vol.97, No.DA-86, pp.25-32 (1997).
- 11) Macchiarulo, L., Benini, L. and Macii, E.: On-the-Fly layout generation for PTL macro-cells, *DATE '01*, pp.546-551 (2001).
- 12) Minato, S., Ishiura, N. and Yajima, S.: Shared binary decision diagram with attributed edges for efficient Boolean function manipulation,

27th DAC, pp.52-57 (1990).

- 13) Ishiura, N.: Synthesis of multilevel logic circuits from binary decision diagrams, *IEICE Trans. Inf. Syst.*, Vol.E75-D, No.9, pp.1085-1092 (1993).
- 14) Rudell, R.: Dynamic variable ordering for ordered binary decision diagrams, *ICCAD '93*, pp.42-47 (1993).
- 15) Hashimoto, T., Onodera, T.: Layout generation of primitive cells with variable driving strength, *SASIMI 2000*, pp.122-129 (2000).
- 16) Mead, C. and Conway, L.: *Introduction to VLSI Systems*, Assison-Wesley, pp.78-79 (1980).
- 17) Yang, S.: *Logic synthesis and optimization benchmarks user guide version 3.0*, MCNC (1991).

(平成 14 年 10 月 23 日受付)

(平成 15 年 3 月 4 日採録)



福岡 一樹

1976 年生。1998 年神戸大学工学部電気電子工学科卒業。2002 年同大学大学院修士課程修了。同年同大学院博士課程進学。LSI 設計に関する研究に従事。



沼 昌宏 (正会員)

1960 年生。1983 年東京大学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。工学博士。1990 年 5 月より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。1996 年文部省在外研究員として米国カリフォルニア大学サンタバーバラ校に派遣。主に LSI, CAD, アクセラレータ, 画像処理に関する研究に従事。IEEE, ACM, 電子情報通信学会会員。



山本 啓輔

1939 年生。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 年神戸大学工学部電気電子工学科教授。2001 年同大学共同研究開発センター教授。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。



金銅 恒

1962 年生。1985 年京都大学工学部電気工学科卒業。1987 年同大学大学院修士課程修了。同年川崎製鉄(株)(現川崎マイクロエレクトロニクス(株))入社。1994 年客員研究員として米国カリフォルニア大学サンタバーバラ校に派遣。現商品開発部課長。主に LSI 設計, CAD 開発に従事。