

設計固有セルライブラリの自動生成手法

吉田 浩章^{†,††} 藤田 昌宏^{†,††}

[†] 東京大学 大規模集積システム設計教育研究センター (VDEC)

〒 113-8656 東京都文京区本郷 7-3-1

^{††} 科学技術振興機構 戦略的創造研究推進事業 CREST

E-mail: [†]hiroaki@cad.t.u-tokyo.ac.jp, ^{††}fujita@ee.t.u-tokyo.ac.jp

あらまし 近年の CAD 技術の進歩の一方で, ASIC とカスタム LSI の間には依然として大きな性能格差が存在する. セルライブラリは ASIC の性能を決める重要な要因の一つである. 本稿では高性能 ASIC 設計を対象として, 特定の設計と設計制約に最適化された論理関数や駆動能力を持つセルライブラリを自動的に生成する手法を提案する. ベンチマーク回路を用いた事例研究では, 設計固有セルライブラリを用いることで面積-遅延トレードオフ曲線が左下に移ることを確認し, 提案手法が本質的に回路を改善することが可能であることを示した.

キーワード スタンダードセル, セルライブラリ, 高性能設計, 論理合成, トランジスタサイジング

Automatic Generation of Design-Specific Cell Libraries

Hiroaki YOSHIDA^{†,††} and Masahiro FUJITA^{†,††}

[†] VLSI Design and Education Center(VDEC), University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

^{††} CREST, Japan Science and Technology Agency

E-mail: [†]hiroaki@cad.t.u-tokyo.ac.jp, ^{††}fujita@ee.t.u-tokyo.ac.jp

Abstract In spite of the significant advances of computer-aided design tools for LSIs over the decades, there has been a large performance gap between ASICs and custom LSIs. One of the major limiting factors is a use of generic-purpose cell libraries. In this paper, we propose an automated methodology for generating a cell library specific to a given circuit and design constraints. A case study using a benchmark circuit demonstrates that using the design-specific cell libraries, the area-delay tradeoff curve is shifted to the left-bottom from that using a typical cell library. This result shows that the proposed methodology can achieve an intrinsic improvement.

Keywords Standard cell libraries, high-performance design, logic synthesis, transistor sizing

1 はじめに

トランジスタレベル最適化は現在まで様々な手法が提案されており, ゲートレベル最適化に比べ非常に高い性能を得ることが可能であることが知られている. 代表的なものとしては, トランジスタサイジング [1] やマクロセルに基づいた手法 [2] などが挙げられる. また, 最近になってスタンダードセルに基づいたフローを対象にしたトランジスタレベル最適化手法が提案されている [3]. これらの技術は, スタンダードセルライブラリに必要な応じて新しいセルを追加することで, トランジスタレベ

ル最適化と同等の性能向上を実現することを可能としている. しかしながら, この手法には (1) 初期的なライブラリとして完全なライブラリを必要としている, (2) 最終的な最適化品質は初期ライブラリに大きく依存する, といったいくつかの問題がある.

セルライブラリのトランジスタレベル設計は大きく分けて論理種類決定と駆動能力決定の2つに分けることができる. 論理種類決定は論理関数とトランジスタ回路構造を決定する問題であり, 駆動能力決定は各論理種類に対して, いくつの駆動能力を提供するかを決定して各トランジスタのサイズを決定する. 理想的な回路では各論

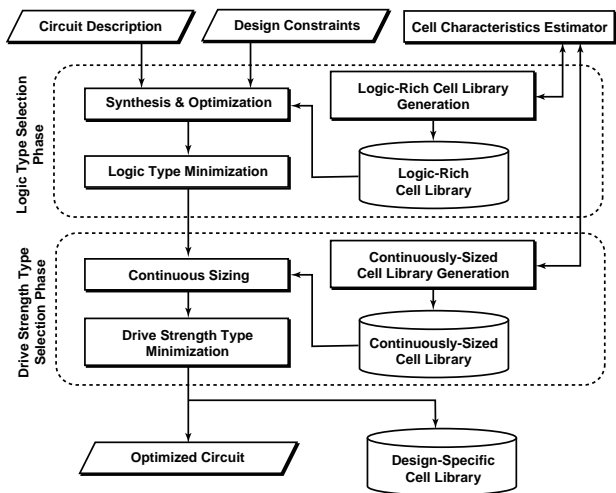


図1 設計固有セルライブラリ自動生成の全体の流れ

理ゲートをトランジスタレベルにおいて構造やサイズの最適化を行うことができる。しかしながら、ライブラリ内のセル数には制約があり、十分な数の論理種類や駆動能力を提供することは高性能カスタム設計の場合を除き現実的に不可能である。このような背景から、各設計に対して最適な論理種類および駆動能力を決定することが望まれている。論理種類決定 [4], [5] や駆動能力決定 [6], [7] に関してはこれまで研究されているが、これらの既存研究は汎用セルライブラリの設計指針に関するもので、設計固有セルライブラリを自動生成するものではない。

本稿では、与えられた設計記述と設計制約に対して最適なセルライブラリを自動生成する手法を提案する。全体の流れを図1に示す。流れは大きく論理種類決定と駆動能力決定の二つの段階に分かれている。二つの段階の基本的な考え方は共通しており、(1) まずセル数に制約がないものとして制約なし最適化を行い、(2) 設計制約に満たす範囲でセル数の最小化を行うという2段階からなる。この手法では、各段階において多数のセルの性能評価を行いながら最適化を行う。[8] を用いることで効率的に性能評価を行うことが可能である。各段階については第2節と第3節で詳細に説明する。全体としての有効性を示すため第4節ではベンチマーク回路を用いた事例研究を示す。

2 論理種類決定手法

2.1 論理種類豊富なセルライブラリの構築手法

本稿では、セルライブラリの各セルは一段のスタティック CMOS ゲートからなると想定している。多段のスタティック CMOS ゲートは一段のゲートの組み合わせで実現するものとする。スタティック CMOS ゲートはドレインとソースによって互いに接続されたトランジスタの集合である。また、互いに相補的な構造を持つ P 型トランジスタ部分と N 型トランジスタ部分の2つの部分からなる。これらのトランジスタを接点として見なすと、入

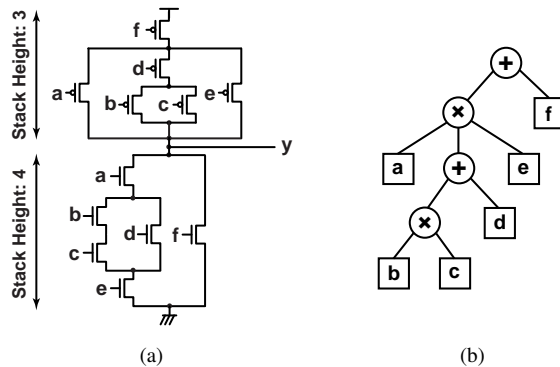


図2 (a) スタティック CMOS ゲートの例 (入力数 I : 6, P 型スタック高さ S_P : 3, N 型スタック高さ S_N : 4) と (b) 対応する AND/OR 木

れ子構造になった直並列 (または並列) 接点回路網であり、ある論理関数を実現している。スタティック CMOS ゲートは常に負ユネイト関数 (単調減少関数) を実現している。例えば、図2 (a) に示すスタティック CMOS ゲートは $y = ((ab + c) * d) + ef$ の論理関数を実現している。スタティック CMOS 回路はスタティック CMOS ゲートからなる回路と定義される。

2.1.1 トランジスタ回路構造の列挙

スタティック CMOS ゲートは入力数 I , P 型スタック高さ S_P , N 型スタック高さ S_N といった構造的性質によって分類できる。ここでスタック高さは以下のように定義される。

定義1 スタティック CMOS ゲートの $P(N)$ 型トランジスタ部分のスタック高さは直列に接続されたトランジスタ数の最大値である。

論理種類豊富なセルライブラリのトランジスタ回路構造の集合は以下の構造制約によって指定する。

- 最大入力数: I_{max}
- P 型スタック高さの最大値: S_{Pmax}
- N 型スタック高さの最大値: S_{Nmax}

これらの構造制約の下ですべての可能なトランジスタ回路構造を列挙する。本節ではスタティック CMOS ゲートの表現方法として AND/OR 木を導入し、次に上記の構造制約の下ですべての可能な AND/OR 木構造を列挙する方法について説明する。

AND/OR 木は以下のように定義される。

定義2 AND/OR 木は根付き順序木であり、葉ノードは入力に対応し、内部ノードが論理積または論理和に対応する。

AND/OR 木は木構造を直並列ネットワークに変換することで一意にスタティック CMOS ゲートを表現しており、またその逆も成り立つ。P 型部分に関しては、論理和ノードは部分ネットワークの並列接続に対応し、論理

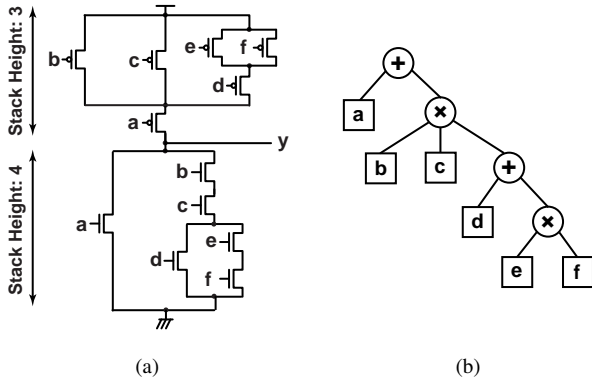


図3 (a) スタティック CMOS ゲートと (b) 対応する AND/OR 木. 図2と同じ論理関数を実現しているが子ノードの順序が異なる.

積ノードは直列接続に対応する. N 型部分についても同様に, 論理積ノードは部分ネットワークの並列接続に対応し, 論理和ノードは直列接続に対応する. AND/OR 木は木構造を深さ優先順にたどることで因数分解型の論理式に変換できる. 例えば, 図2に示す AND/OR 木は $a \cdot (b \cdot c + d) \cdot e + f$ を表現している. この論理式の否定がスタティック CMOS ゲートが実現する関数である. ゲートの性能はトランジスタが出力に対してどのように接続されているか, つまりどのように直列接続されているかに依存している. よって AND/OR 木では各ノードの子ノードの順序を区別する. 図3に示すスタティック CMOS ゲートおよび対応する AND/OR 木は図2と同じ論理関数を実現しているが, 子ノードの順番が異なる. ここでセルの入力の順序は任意のため葉ノードの順序は区別しないことに注意されたい. 次に AND/OR 木のノードの論理和または論理積の幅を以下のように定義する.

定義3 AND/OR 木のノード n の論理積の幅は

$$W_{AND}(n) = \begin{cases} 1 & n \text{ が葉ノードの場合} \\ \sum_{c \in C(n)} W_{AND}(c) & n \text{ が論理積ノードの場合} \\ \max\{W_{AND}(c) \mid c \in C(n)\} & n \text{ が論理和ノードの場合} \end{cases}$$

と定義される. ここで $C(n)$ は n の子ノードの集合である. 同様に AND/OR 木のノード n の論理和の幅は

$$W_{OR}(n) = \begin{cases} 1 & n \text{ が葉ノードの場合} \\ \sum_{c \in C(n)} W_{OR}(c) & n \text{ が論理和ノードの場合} \\ \max\{W_{OR}(c) \mid c \in C(n)\} & n \text{ が論理積ノードの場合} \end{cases}$$

と定義される.

定義4 AND/OR 木 t の論理和または論理積の幅 $W(t)$ はそれぞれ根ノードの論理和または論理積の幅と定義する.

AND/OR 木 t の論理和または論理積の幅 $W(t)$ は N 型または P 型部分のスタック高さに対応する. つまり, $S_P \equiv W_{OR}(t)$ and $S_N \equiv W_{AND}(t)$ である.

構造制約である I_{max} , S_{Pmax} , S_{Nmax} に対して, これらを満たすすべての可能な AND/OR 木構造を列挙する.

AND/OR 木においては, I_{max} は葉ノードの最大数に対応し, S_{Pmax} は論理和幅の最大値, S_{Nmax} は論理積幅の最大値に対応する. まず初期的な集合として各入力に対応する一つの葉ノードからなる AND/OR 木の集合を作成する. 次に, この集合の部分集合の論理積または論理和を計算し, これが構造制約を満たせば集合に追加する. これを新たな木が追加されなくなるまで繰り返すことで木構造を列挙する. 最後に列挙した木集合のそれぞれの木をスタティック CMOS ゲートに変換する.

2.1.2 駆動能力決定

セル内のトランジスタサイズが回路全体の性能に対して大きな影響を持つことはよく知られている. 提案手法ではセルの総数が制限されているため, 駆動能力の種類を多数準備することはできない. 論理種類豊富なセルライブラリでは, 各セルの P 型または N 型部分のトランジスタはすべて同じサイズを持つものとする. つまり P 型と N 型のサイズの比であるベータ比や直列接続のトランジスタの各サイズを変化させるテーパリングの最適化は行わない. ここで決定する駆動能力はあくまで論理種類決定手法でのみ使用されるものであり, 最終的な設計固有セルライブラリの駆動能力は第3節で説明する手法で決定する. 前節で決定したセルのトランジスタ回路構造に対して, 以下の制約に基づいてトランジスタのサイズを決定する.

- 各論理種類の駆動能力の集合: $D = \{d_1, \dots, d_n\}$
- 最小インバータの P 型/N 型のサイズ: w_{PINV}, w_{NINV}
- ベータ比係数: k_β

ここでベータ比は以下のようにして計算される.

$$\beta = (w_{PINV}/w_{NINV}) + k_\beta(S_P - S_N)$$

P 型および N 型の基本サイズ w_P, w_N は以下のように計算される.

$$\begin{aligned} w_P &= w_N \cdot \beta, w_N = w_{NINV} & \beta \geq w_{PINV}/w_{NINV} \text{ の場合} \\ w_P &= w_{PINV}, w_N = w_P/\beta & \beta < w_{PINV}/w_{NINV} \text{ の場合} \end{aligned}$$

各駆動能力 $d_i \in D$ に対して, P 型および N 型のトランジスタサイズ w_P, w_N は以下のように計算される.

$$\begin{aligned} w_{Pi} &= d_i \cdot w_P \\ w_{Ni} &= d_i \cdot w_N. \end{aligned}$$

ここで駆動能力の集合は論理種類によって異なることに注意されたい.

2.2 論理種類削減手法

ここでの目的は設計制約下を満たす対象回路を実装するために必要な論理種類の最小集合を求めることである. 提案手法は山登り法 [9] に基づいている. 前節までに説明した手法で生成された論理種類豊富なライブラリを用いて, まず制約なしで最適化を行う. 多くの場合, 最適化された回路は制約を十分に満たしているが, 非常に多

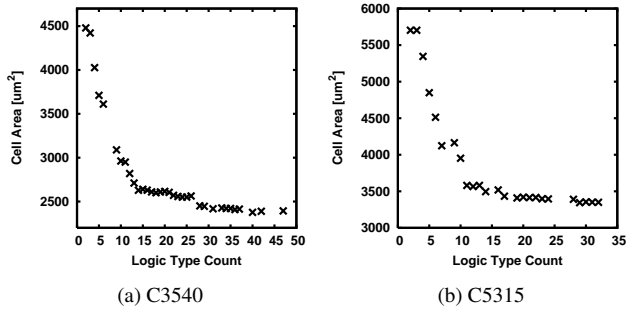


図4 面積最小制約下での論理種類総数と面積のトレードオフ曲線

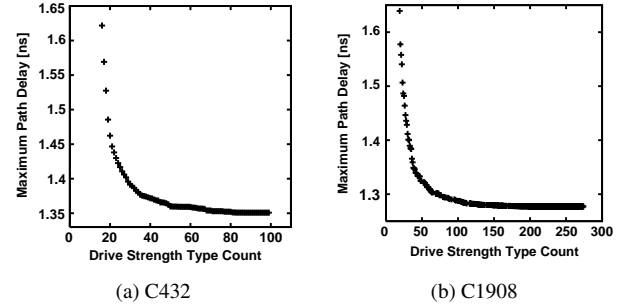


図5 遅延最小制約下での駆動能力総数と遅延のトレードオフ曲線

種類のセルが使用されている。重要度の低い種類のセルを順番にライブラリから除去して再合成を行い、これを制約を満たす範囲で繰り返すことで論理種類の最小化を行う。このヒューリスティック手法の品質はどの論理種類を除去するかへの決定に大きく依存する。提案手法では最も大きなスラックを持つ論理種類を優先的に除去する。ここで論理種類のスラックを以下のように定義する。

定義 5 回路内のある端子 t のスラックは t における要求時間と到着時間の差として定義される。あるゲート g のスラックは g 上の端子のスラックの最小値である。ある論理種類 l のスラックは l の種類を持つゲートのスラックの最小値である。ある論理種類 l の合計スラックは l の種類を持つゲートのスラックの合計値である。

2つ以上の論理種類が同じ最大スラックを持つ場合には、合計スラックが最大のものを決定する。このようにしてその論理種類が回路中から除去された場合にも、回路性能への影響を最小にすることが可能である。

2.3 論理種類決定手法の評価結果

論理種類決定手法の有効性を評価するため ISCAS 85 ベンチマーク回路 [10] と商用 90nm プロセスを用いて実際に論理種類決定を行った。まず、論理種類豊富なセルライブラリの最大入力数を 6、P 型および N 型スタック高さを 4 とし回路構造の列挙を行った。また各論理種類の駆動能力を 1x, 2x, 4x, 6x, 8x, 12x, 16x の 7 種類、ベータ比を 0.1 としセルライブラリを生成した。生成された論理種類は 461 種類で、セルの総数は 1844 であった。セルの性能解析には高精度見積もり手法 [8] と HSPICE を用いた。ベンチマーク回路の論理合成には Cadence PKS (Physically-Knowledgeable Synthesis) [11] を用い、次に示す 3 つの異なる設計制約の下で論理種類決定を行った。最初の設計制約は面積最小合成した場合の面積 A_{area} より 1% の面積増加を許容する、つまり最大面積制約を $(A_{area} * 1.01)$ とした。次の制約では遅延最小合成した場合の面積を A_{delay} 、最大パス遅延と D_{delay} とし、面積制約を $(A_{delay} * 1.1)$ 、最大パス遅延制約を $(D_{delay} * 1.1)$ とした。最後の制約は遅延最小合成した場合よりも 1% の遅延増加を許容する、つまり最大パス遅延制約を $(D_{delay} * 1.01)$ と

した。各設計制約の場合の論理種類決定の結果を表 1 に示す。全体の傾向として、面積最小制約の場合により多くの種類のセルが必要であることがわかる。また図 4(a) と (b) に論理種類の総数と面積のトレードオフ曲線を示す。これらの結果から複雑なセルは面積削減に有効であると確認した。

3 駆動能力決定手法

駆動能力決定手法は我々の既存研究 [12] に基づいているため、ここでは簡潔に説明を行う。前節の論理種類決定手法によって論理種類の集合とその集合によって実装された回路が与えられているものとする。提案手法は (1) まず各トランジスタが任意の連続サイズを持つものとして制約なしで最適化を行う、(2) 設計制約を満たす範囲で駆動能力の種類の最小化を行う、という 2 段階によって各論理種類の駆動能力の最小集合を求める。図 5(a) と (b) に遅延最小制約下での駆動能力総数と遅延のトレードオフ曲線を示す。論理種類決定の場合とは対照的に、この結果は駆動能力は遅延削減に有効であるということを示している。

4 事例研究: ISCAS 85 ベンチマーク回路 C432 + C499

ここでは ISCAS 85 ベンチマーク回路の C432 と C499 を結合した回路を例題に、実際に提案手法を適用することで有効性を示す。C432 は 27 チャンネル割り込み制御回路、また C499 は 32 ビット SEC 回路であり [13]、これら 2 つの回路は独立して配置されている。第 2.3 節と同様に論理合成には Cadence PKS を用いる。実験は以下のようにして行う。まず、比較用として商用 90nm プロセスを用いて典型的なセルライブラリを構築する。このライブラリを用いて例題回路を様々な設計制約下で合成し、面積-遅延トレードオフ曲線を求める。次に提案手法を適用して設計固有ライブラリを用いて例題回路を最適化する。公平な比較のため、設計固有ライブラリのセルの総数は多くとも典型ライブラリのセル総数の 50 とする。最後に設計固有ライブラリを用いた場合の面積-遅延トレードオフ曲線を求め、これを典型的ライブラリのトレードオフ曲線と比較することで提案手法の有効性を示す。

表 1 ISCAS 85 ベンチマーク回路に対する論理種類決定結果 (商用 90nm プロセス)

回路	面積最小から 1%増加許容			遅延最小から面積 10%, 遅延 10%増加許容			遅延最小から 1%増加許容		
	面積 [μm^2]	遅延 [ns]	論理種類 総数	面積 [μm^2]	遅延 [ns]	論理種類 総数	面積 [μm^2]	遅延 [ns]	論理種類 総数
C432	346.5	3.39	17	824.7	2.08	8	2333.2	1.23	15
C499	1055.7	1.98	9	1512.9	1.51	4	4180.7	1.06	10
C880	806.9	2.74	17	1114.3	1.85	13	2215.3	1.06	10
C1355	1057.9	1.99	9	1598.2	1.53	6	4325.0	1.10	3
C1908	1065.2	3.13	15	1438.7	2.29	13	4374.7	1.41	7
C2670	1472.1	2.72	19	1883.7	1.80	8	4539.1	0.95	6
C3540	2376.8	4.03	40	3298.6	3.05	10	8694.1	1.71	22
C5315	3344.5	3.33	29	4313.3	2.38	11	9708.2	1.36	14
C6288	6090.7	11.16	11	14398.6	8.06	8	36195.9	4.82	9
C7552	4589.8	5.08	31	5562.6	2.72	10	17515.6	1.29	7

表 2 参考用の典型的セルライブラリの論理種類と駆動能力の概要
(論理種類: 15, セルの総数: 50)

Logic Type	Function	Drive Strengths
INV	$\overline{A+B}$	1x, 2x, 4x, 8x, 16x
NAND2	$\overline{A \cdot B}$	1x, 2x, 4x, 8x
NAND3	$\overline{A \cdot B \cdot C}$	1x, 2x, 4x, 8x
NAND4	$\overline{A \cdot B \cdot C \cdot D}$	1x, 2x, 4x
NOR2	$\overline{A+B}$	1x, 2x, 4x, 8x
NOR3	$\overline{A+B+C}$	1x, 2x, 4x
NOR4	$\overline{A+B+C+D}$	1x, 2x, 4x
AOI21	$\overline{A \cdot B + C}$	1x, 2x, 4x
AOI211	$\overline{A \cdot B + C + D}$	1x, 2x, 4x
AOI22	$\overline{A \cdot B + C \cdot D}$	1x, 2x, 4x
AOI221	$\overline{A \cdot B + C \cdot D + E}$	1x, 2x, 4x
OAI21	$\overline{(A+B) \cdot C}$	1x, 2x, 4x
OAI211	$\overline{(A+B) \cdot C \cdot D}$	1x, 2x, 4x
OAI22	$\overline{(A+B) \cdot (C+D)}$	1x, 2x, 4x
OAI221	$\overline{(A+B) \cdot (C+D) \cdot E}$	1x, 2x, 4x

4.1 典型的セルライブラリ

商用 90nm プロセスを用いて表 2 に示す典型的な論理種類と駆動能力を持つセルライブラリを構築した。表では、各行が論理種類に対応しており、2 列目が論理関数、3 行目が駆動能力を表している。論理種類と駆動能力の集合は一般的なセルライブラリに準拠したものとなっている。トランジスタのサイズは第 2.1.2 節で説明した手法を用いている。論理種類は 15 種類あり、セルの総数は 50 である。セル性能解析は第 2.3 節と同様の方法で行っている。このライブラリを用いて例題回路を様々な設計制約下で合成し、面積-遅延トレードオフ曲線を得た。

4.2 設計固有セルライブラリ

まず、提案手法を用いて面積最小制約下で設計固有セルライブラリの生成を行った。面積最小回路のセルライブラリの論理種類の概要を表 3 に示す。論理種類の総数は 26 であった。面積最小回路ではすべてのトランジスタが最小となっており各論理種類に対して単一の駆動能力

表 3 面積最小回路の論理種類の概要

論理関数	使用回数	論理関数	使用回数
$\overline{A+B}$	70	$\overline{A \cdot B + C \cdot D + E \cdot F}$	2
$\overline{A \cdot B + C}$	53	$\overline{A+B+C+D}$	2
$\overline{A \cdot (B+C)}$	50	$\overline{A \cdot B \cdot (C+D) \cdot (E+F)}$	1
\overline{A}	36	$\overline{A \cdot B \cdot (C+D \cdot E) + F}$	1
$\overline{(A+B+C) \cdot D}$	32	$\overline{A \cdot B \cdot C \cdot D}$	1
$\overline{A \cdot B}$	19	$\overline{A \cdot B + C + D + E \cdot F}$	1
$\overline{A \cdot B \cdot C}$	17	$\overline{A \cdot (B+C) + D + E}$	1
$\overline{(A+B \cdot C) \cdot D}$	8	$\overline{A \cdot B + (C+D) \cdot (E+F)}$	1
$\overline{A \cdot B + C + D}$	6	$\overline{A \cdot (B+C+D \cdot E) + F}$	1
$\overline{A \cdot B + C + D \cdot E}$	5	$\overline{(A+B) \cdot (C+D)}$	1
$\overline{A \cdot B \cdot (C+D)}$	4	$\overline{(A+B \cdot C \cdot D) \cdot E}$	1
$\overline{A \cdot B + C \cdot D}$	3	$\overline{(A+B+C \cdot D) \cdot E \cdot F}$	1
$\overline{(A+B) \cdot C}$	3	$\overline{A+B+C}$	1

表 4 遅延最小回路の論理種類の概要

論理関数	使用回数
\overline{A}	263
$\overline{A \cdot B}$	180
$\overline{A+B}$	70
$\overline{(A+B) \cdot C}$	63
$\overline{A \cdot (B+C)}$	47
$\overline{A+B \cdot C}$	46

で十分であるため、セルの総数も 26 である。典型的なライブラリに含まれない 6 入力の複雑なセルも生成されていることがわかる。

次に、遅延最小制約下での設計固有セルライブラリを生成した。論理種類豊富なセルライブラリを用いて遅延最小回路の論理種類の概要を表 4 に示す。面積最小回路の場合とは対照的に、論理種類は単純なもののみから構成されている。また種類が少ないにも関わらず、OAI21 の論理関数を実現するセルとして $\overline{(A+B) \cdot C}$ と $\overline{A \cdot (B+C)}$ の異なる 2 つの構造が生成されている。これら 2 つのセル

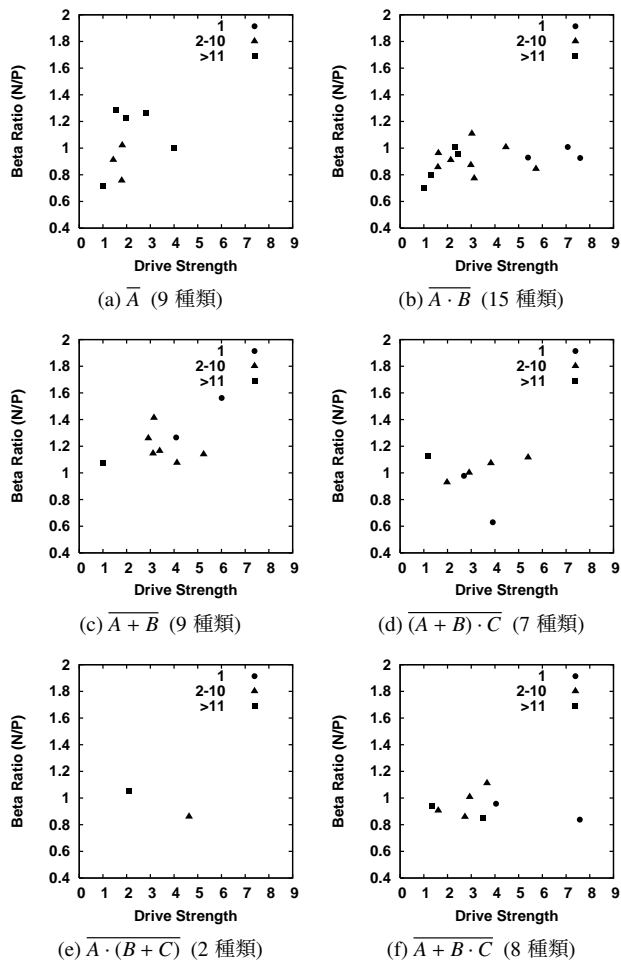


図6 遅延最小回路の駆動能力の分布 (●は使用回数が1回, ▲は使用回数が2~10回, ■は使用回数が11回以上)

異なる入出力間の遅延をもっているため、ゲート入力
の遅延の重要度に応じて使い分けられている。次に連続
トランジスタサイジングによって最適サイズを求めた後、
セルの総数が50となるように駆動能力の削減を行った。
図6(a)-(f)に全6論理種類の駆動能力の分布を示す。ほと
んどのトランジスタのサイズが典型的ライブラリのサイ
ズの範囲内に入っていることが確認できる。これは設計
固有セルライブラリが大きいトランジスタを用いて性能
改善を実現しているのではなく、最適なサイズやベータ
比を用いることによって性能改善を実現していることを
示している。

最後に、面積最小回路と遅延最小回路の間となる4
つの制約を満たす設計固有セルライブラリを生成した。
図7に典型的ライブラリと設計固有ライブラリを用いた
場合の面積と遅延のトレードオフ曲線を示す。設計固有
ライブラリを用いた場合のトレードオフ曲線は典型的ラ
イブラリの場合のトレードオフ曲線よりも左下となっ
ている。面積最小回路同士の比較では面積は27.3%改
善しており、遅延最小回路同士の比較では遅延は22.4%
改善した。この結果より提案手法によって回路を本質
的に改善することが可能であることを確認した。

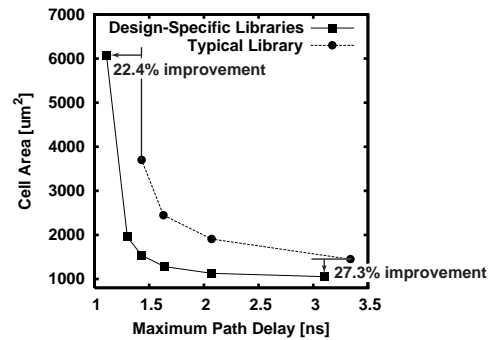


図7 典型的ライブラリと設計固有ライブラリを用いた場合の面積と遅延のトレードオフ曲線

5 結 論

本稿では設計固有セルライブラリを自動生成する手法
を提案した。ベンチマーク回路を用いた事例研究では、
設計固有セルライブラリを用いることで面積-遅延ト
レードオフ曲線が左下に移ることを確認し、提案手法が本
質的に回路を改善することが可能であることを示した。

文 献

- [1] J. P. Fishburn and A. E. Dunlop, "Tilos: A posynomial programming approach to transistor sizing," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1985, pp. 326-328.
- [2] J. L. Burns and J. A. Feldman, "C5M - a control-logic layout synthesis system for high-performance microprocessors," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 1, pp. 14-23, Jan. 1998.
- [3] D. Bhattacharya and V. Boppana, "Design optimization with automated flex-cell creation," *Closing the Gap Between ASIC & Custom*, pp. 14-23, 2002.
- [4] K. Keutzer, K. Kolwicz, and M. Lega, "Impact of library size on the quality of automated synthesis," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1987, pp. 120-123.
- [5] B. Guan and C. Sechen, "Large standard cell libraries and their impact on layout area and circuit performance," in *Proc. IEEE Int. Conf. on Computer Design*, Oct. 1996, pp. 378-383.
- [6] F. Beeffink *et al.*, "Gate size selection for standard cell libraries," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1998, pp. 545-550.
- [7] D. S. Kung and R. Puri, "Optimal P/N width ratio selection for standard cell libraries," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1999, pp. 178-184.
- [8] H. Yoshida, K. De, and V. Boppana, "Accurate pre-layout estimation of standard cell characteristics," in *Proc. ACM/IEEE Design Automation Conf.*, Jun. 2004, pp. 208-211.
- [9] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag, 2002.
- [10] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits," in *Proc. IEEE Int. Symp. Circuits and Systems*, Jun. 1985, pp. 695-698.
- [11] *PKS User Guide*. Cadence Design Systems, 2004.
- [12] H. Yoshida and M. Fujita, "Performance-constrained different cell count minimization for continuously-sized circuits," in *Proc. IEEE Design, Automation and Test in Europe*, Apr. 2008, pp. 208-211.
- [13] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *IEEE Design and Test*, vol. 16, no. 3, pp. 72-80, Jul. 1999.