東海大学紀要情報通信学部
vol.6,No1,2013,pp.23-30

論文

オープンソース EDA ツールセットとディープサブミクロンプロセス対応 λ ルールベースセルライブラリとを用いた Rohm 0.18μm プロセス向けレイアウトの設計手法

細川 達也[*1]，　清水 尚彦[*2]

The Layout Design Method for Rohm 0.18μm Process Using Open Source EDA Tool-set and λrule Based Cell Library Corresponding to Deep Sub-micron Process

by

Tatsuya HOSOKAWA[*1] and Naohiko SHIMIZU[*2]

（Received on March 29, 2013 & Accepted on July 19, 2013）

Abstract

Start-up company and rising country are difficult to design application specific integrated circuit(ASIC) since it is very expensive both The forefront chip that can be implemented very large circuit by improving LSI fabrication technique and the software to design the circuit(EDA). Therefore we resolve the matter by developing the layout designing method for the chip that fabrication cost is inexpensive and be in demand using open source EDA software corresponding to standard cell system that it is main way to plan the ASIC. Moreover we adopt λ rule based cell library that can used free since the library that need for the system and be provided by fabrication company(Cell library) is only use in the pay EDA. We demonstrate our research result to compare and verify the chip-layout that can fabricate designed by our method and it developed by the pay EDA.

Keyword: Open Source EDA, Deep Sub-micron Process, λ Rule, Standard Cell Library, Rohm 0.18μm
キーワード: オープンソース EDA, ディープミクロンプロセス, λ ルール, スタンダードセルライブラリ, Rohm 0.18μm

1. はじめに

近年、集積回路 (LSI) 製造技術の向上による微細化が進み性能が向上するとともに製造費 (NRE) も高額になっている。また、特定目的集積回路 (ASIC) の設計には設計ソフトウェア (EDA[*1]) が必須であるが、ライセンス料が非常に高額である。加えて EDA ベンダーは寡占状態にあるためその他の選択肢がない。最先端のチップを利用したディジタル ASIC の開発には膨大なコストがかかるので、ベンチャー企業や新興国には困難である。

Fig.1 に示すように、NRE が非常に高価な最小設計線幅 0.08μm 未満の LSI が世界シェアの半分以上を占める一方、それ以下のものが未だ 4 割のシェアを持っていることがわかる [1)]。また、0.2μm 以上のものは四半期ごとにシェアを落としているものの、ディープサブミクロンプロセスは継続して 2 割程度のシェアを確保している。また、NRE を下げるため、1 枚のウェハを複数のプロジェクトでシェアするシャトルサービスも使われるが、NRE を軽減しても EDA のコストが大きな負担となる。この問題を解決するため、最小設計線幅 0.2μm から 0.08μm までの LSI(ディープサブミクロンプロセス) を対象としてオープンソース EDA とラムダルールベースセルライブラリとを用いるスタンダードセル方式の設計手法を提案する。チップごとの開発費を比較したグラフと研究成果によるコスト削減効果予測を Fig.2 に示す。

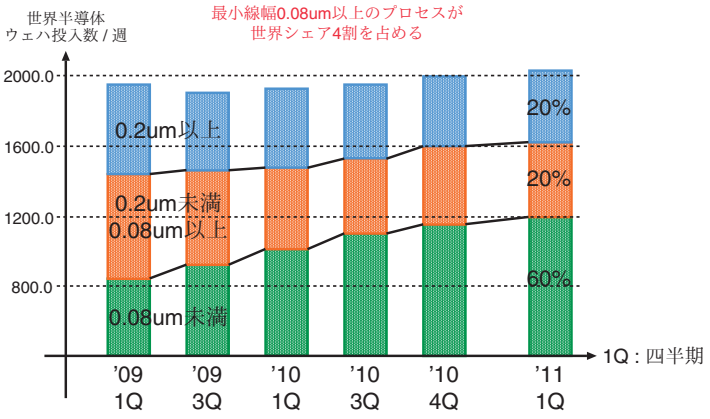


Fig.1　The wafer sales numbers each a quarter

スタンダードセルライブラリ方式の概要を Fig.3 に示す。この方式は、設計した論理回路をレイアウト済みの基本ゲート[*2](セル) と置き換え、セルどうしを論理回路どおりに配線することでチップレイアウトを設計する手法である。この手法はフルカスタム手法[*3]やゲートアレイ手法[*4]と比較して、フルカスタムと同等の集積度を実現でき、かつ EDA による自動化が可能であるという利点がある。しかし、セルの設計はプロセス[*5]ごとに行わなければならず、プロセス間の互換性はない。また、基本論理とな

[*1] 工学研究科 情報通信制御システム工学専攻
[*2] 情報通信学部 組込みソフトウェア工学科

[*1] Electronic Design Automation
[*2] 論理回路の and 素子や or 素子などを総称して基本ゲートという
[*3] 論理回路設計からチップレイアウト設計までを全て人手で行う手法
[*4] 予めチップ領域に格子状に敷き詰めた基本ゲートをつなぎ合わせる手法
[*5] チップの種類を指す。チップは設計会社ごと、最小設計線幅ごとに区別さ

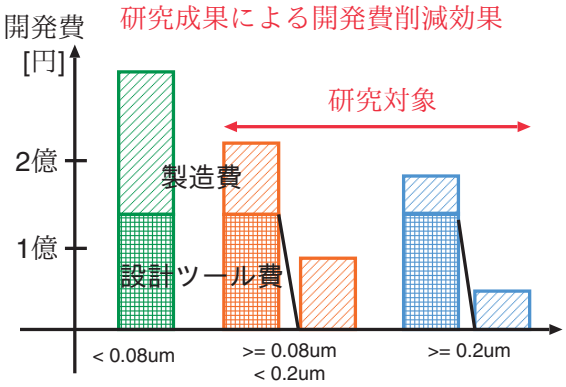


Fig.2　The development cost each process and cost-cut effectivity by our reserch

る回路は非常に多く数百種類に及ぶため、セルライブラリの設計には非常に大きなコストが発生する。

この設計手法に対して、1981 年に Mead と Conway がλルールを提唱した [2]。これはトランジスタのゲート長をλないし 2λとすることでレイアウト中の全ての寸法をλで表現する手法である。このルールによってセルライブラリにプロセス間の互換性が生まれ、生産性の向上が目された。しかし実際には約 0.5μm 以下のプロセスでスケーラビリティを実現できる EDA が存在しないため、λルールベースセルライブラリを用いた設計は 1μm 前後、もしくはそれ以上のプロセスでのみ可能である。

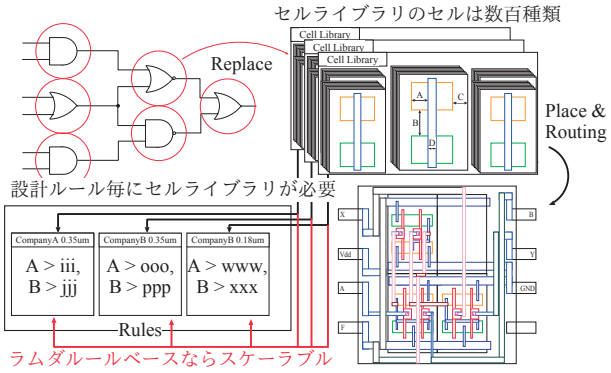


Fig.3　Outline of standard cell library method and *lambda* rule

先行研究 [3] では 1μm プロセス用のスタンダードセルのλを変更し、調整することで 1.2μm、0.35μm 及び 0.18μm プロセスを用いるチップレイアウトを設計できる手法を開発した。実際に試作したチップの写真を Fig.4 に示す。しかし、この手法で設計したチップは商用の EDA ソフトウェアと専用のセルライブラリを用いて設計した場合と比較して面積が数倍から数十倍になるため早急な改善が必要であった。この手法で用いているセルライブラリはλをゲート幅より大幅に大きくしなければデザインルールを満足できなかったからである。そこで我々は 0.13μm プロセス向けに開発されたλルールベースセルライブラリ (vsclib[4]) を採用し、スタンダードセル方式に対応しているオープンソース EDA(Alliance[5]) の修正を行うことによって新しい設計手法を開発した。また、この手法を用いて 0.18μm プロセス向けのチップレイアウトの設計を行った。

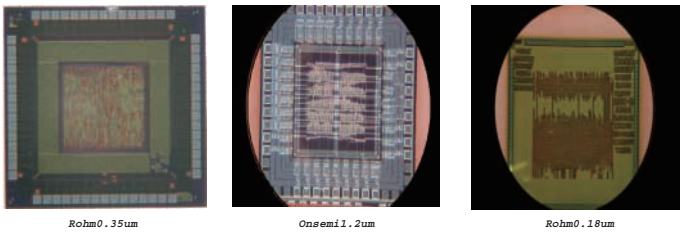
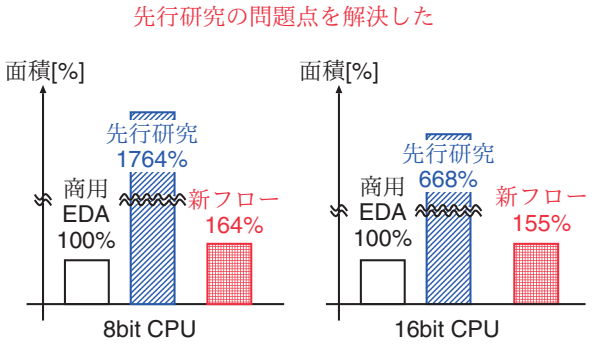


Fig.4　The chip pictures our designed

新規設計手法の開発にあたり、Alliance の論理合成*6 ソフトウェアと配置配線*7 ソフトウェアとを改造し、λルールとプロセスのデザインルールの差とを設定ファイルで調整する機能 (補正機能) を用いてデザインルールの調整を行った。また、補正機能では修正できないエラーがあったため、プログラムを新たに開発し解決した。これによってデザインルールチェック (DRC) をクリアできるレイアウトの設計に成功した。レイアウト面積を計測・比較した結果、先行研究にて試作したチップの面積を大幅に下回っており、先行研究での課題を解決することができた。双方の面積比を Fig.5 に示す。



	8bit CPU	16bit CPU
商用EDA	100%	100%
先行研究	1764.26%	668.54%
新フロー	164.08%	155.54%

Fig.5　Area comparison of chips designed each method(commercial, previous research, new work)

類似研究として Grad, stein らの研究 [6] が挙げられる。この研究は、学生へ ASIC 開発の実践教育を提供するために 0.5μm 向けのスケーラブルセルライブラリと、それを配置配線するためのスクリプトとを併せて開発したものである。しかし、この研究で

れる。e.g. Rohm 社の最小設計線幅 0.18μm チップ:Rohm 0.18μm プロセス

*6 設計した論理を目的に合わせて最適化する工程
*7 論理回路と置き換えたセルを、指定の領域に敷き詰め (配置)、セル間を配線する (配線) こと

は論理合成に商用ツールを使用している。また、対象のプロセスは AMI0.5μm であるため、我々の、安価なディープサブミクロンプロセスの設計手法を確立する目的を果たせない。小野寺らはプロセスごとにスタンダードセルライブラリを自動生成するシステムを開発し、教育に用いることができるようにすべての情報をライブラリとして提供する機能をもたせた [7]。しかし、このシステムのスケーラビリティは 0.5μm までであり、我々の目的と合致しない。この研究と同様の研究として吉田, 藤田の研究 [8] があるが、彼らは商用ツールを用いて、最適なセルライブラリを作成するための指針を示すのみであり、実際のセルライブラリ作成は行っていない。

2. 新セルライブラリ適用上の問題点

新セルライブラリをこれまでの設計手法に適用するためには、Table 1 に示す問題点を解決する必要があった。以下の項でそれぞれの問題の詳細と解決法とを述べる。

Table 1　The solution, cause, and problem on new research

問題点	原因	解決法
新ライブラリ利用不可	ハードコーディングされたデザインルール	ソフトウェア改変
	ライブラリに必要なレイヤーの不足	
デザインエラー発生	設定ファイルの不適合	設定ファイル再設計
	セグメント調整による副作用	セルレイアウト調整
	セル境界のルール違反	新プログラム設計

2.1　EDA ソフトウェアの改造

Alliance の配置配線ソフトウェアにはデザインルールがハードコーディングされている。そのため仕様の異なるセルライブラリを利用することができない。セル内部の仕様の違いを Fig.6 に示す。

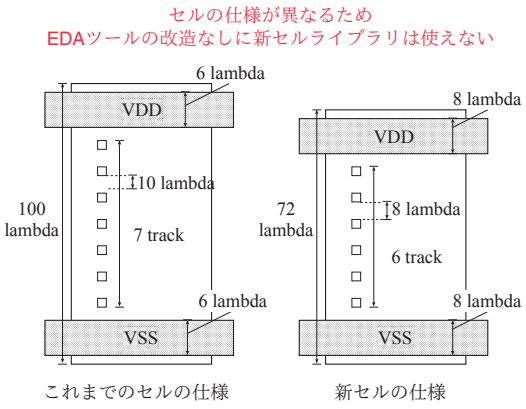


Fig.6　The cell specification difference

新セルライブラリではこれまで使用していなかったレイヤーが必要であったり、様々なレイヤーのサイズが異なったりしたため、使用するセルライブラリに応じて設定値を変更できるよう配置・配線ソフトウェアを改造した。新セルライブラリを対応させるために行った具体的な作業内容を Table 2 にまとめる。

Table 2　The remaking and additional point by software

	追加・変更点
レイヤー	メタル 8 層の追加
オプション	x 軸方向にハーフピッチで計算するオプションを追加
ルール	ピッチとトラックをプログラム外で調整可能とした

レイアウト調整フローは
ラムダ調整、セグメントサイズ調整、セルレイアウト調整
の順で行う

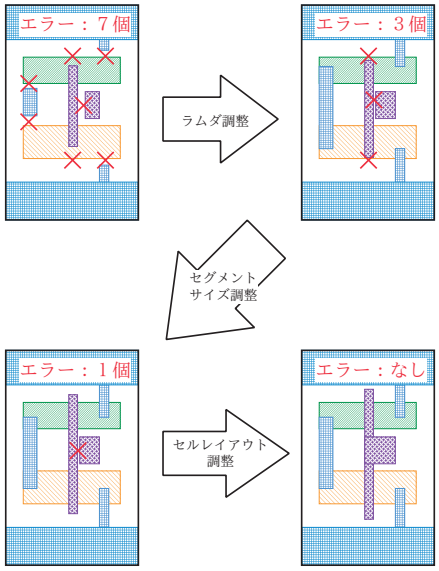


Fig.7　The layout designing method flow on previous research

2.2　設定ファイル書き換え

vsclib は 0.13μm プロセス向けに作られており、0.13μm 用 Alliance 設定ファイルが用意されている。我々はそれを元に Rohm が提供するゲート長が 0.18μm のチップ (Rohm 0.18μm プロセス) 向けの設定ファイルの作成を行った。設定ファイルの調整手法は石黒によって解説されており [9]、樋口はこれを参考に試作可能なレイアウトを設計できる手法を開発した (先行研究)[10]。開発手法を Fig.7 に示す。

本研究では GDS 上の 1 グリッドのサイズを 0.005μm に、λ を 0.085 に設定した。セグメントサイズを補正する最小単位は 1 グリッドのサイズで決定するため、設定できる最小値を与えた。vsclib はゲート幅を 2λ としているため、当初 λ には 0.09 を与えた。しかし、λ に 0.03 の倍数の値を設定するとセグメント幅が一部正しく計算されないという Alliance の問題を発見したため、λ の値を 0.085 とし、0.005 の誤差をセグメントの調整によって補完することとした。実際に λ を 0.09 にした場合に発生するポリシリコン幅のエラーを Fig.8 に示す。

補正機能に与える設定値は石黒の論文で示されている式によって計算可能であるが、Alliance の本機能は異なるセグメント間の関係を考慮、補正することができないため、設定値を算出した後は gds もしくは cif に変換し、DRC でレイアウトを確認することが不可欠である。代表的なレイアウトエラーを Fig.9 に示す。

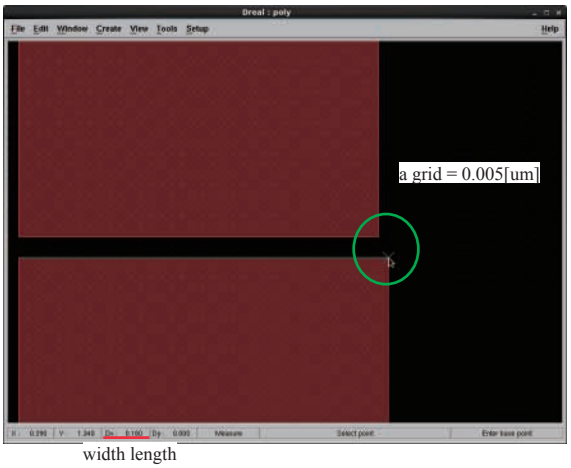


Fig.8　We resolve segment error as λ=0.085

この例は最小幅エラーが起きているセグメントの補正をした結果、他のセグメントとの間隔が狭まってしまい、新たなエラーを引き起こす状態を示している。補正機能は異なるセグメント間の距離を設定する機能を持たないため、このようなエラーは補正機能とその設定値だけでは修正できない。

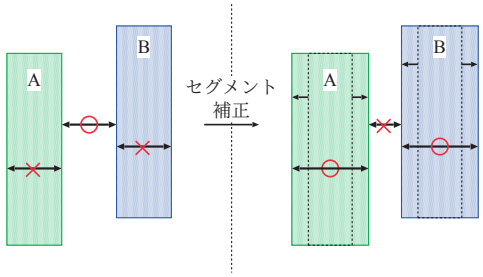


Fig.9　The new error caused by fixing layout error

コンタクトレイヤーに付属しているメタル 1 レイヤーを縮小し、同じサイズにした。セル内部の配線が混みいっているため、コンタクトレイヤーからオーバーラップしているメタル 1 レイヤーが最小間隔エラーを引き起こすからである。具体的な事例を Fig.10 に示す。

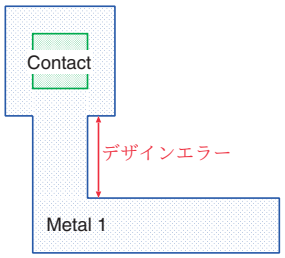


Fig.10　A minimum distance error caused by metal under contant

ゲートのポリシリコンを長めに、論理の入力部分のコンタクトに付属するポリシリコンを大きく宣言した。補正しない状態のポリシリコンは大半が短すぎる状態だったからである。この状態を Fig.11 に示す。図中の赤いセグメントはポリシリコンを示し、紫色の正方形は論理回路の入力にあたるコンタクトを示している。

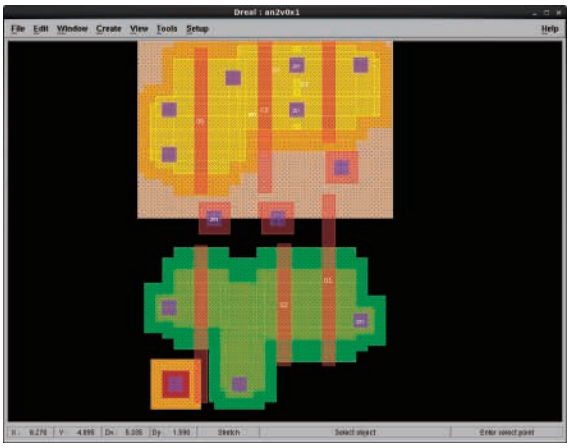


Fig.11　The layout in case of no complement poly layer

Alliance には同じ種類のセグメントの間隔が一定以下であった場合に自動的に間隔を補完する機能 (補完機能) がある。補完機能のインプラントレイヤーの設定値を最小間隔ルールより大きく設定した。トランジスタが何個も並ぶようなレイアウトの場合、インプラントが凹凸になってしまい、凹部で最小間隔エラーが発生したからである。この状況を Fig.12 に示す。

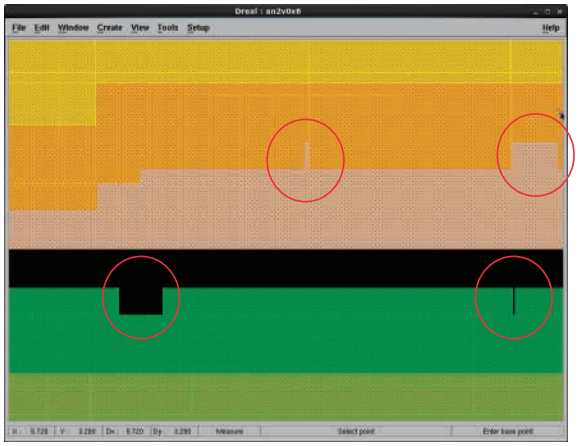


Fig.12　A layout error example in case of no using merge function

2.3　セルライブラリの修正

補正機能の設定値を調整してもデザインルール違反となるセルが数点あったため、セルを直接編集することによってエラーを解決した。この違反は設定ファイルによる調整では解決できなかった。編集したセルは、

- an4v0x1
- nd2v5x2
- nr3abv0x05

の 3 つである。それぞれのレイアウトを Fig.13、14、15 に示す。それぞれの図の上部が編集前、下部が編集後である。an4v0x1

セル (Fig.13) では完全に接続されていないコンタクトと配線とを (上)、配線を付け足すことで接続させた (下)。nd2v5x2 セル (Fig.14) では正しく接続されていない配線を (上)、配線を付け足すことで接続させた (下)。nr3abv0x05 セル (Fig.15) ではコンタクトからのオーバーラップが足りない配線 (上) をルールを満足するまで延長した (下)。

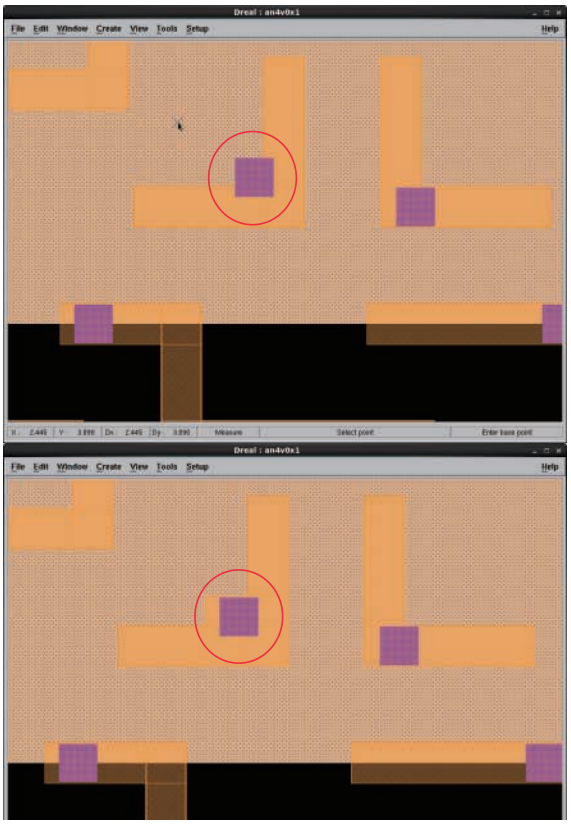


Fig.13　We fix contact layer shown figure below

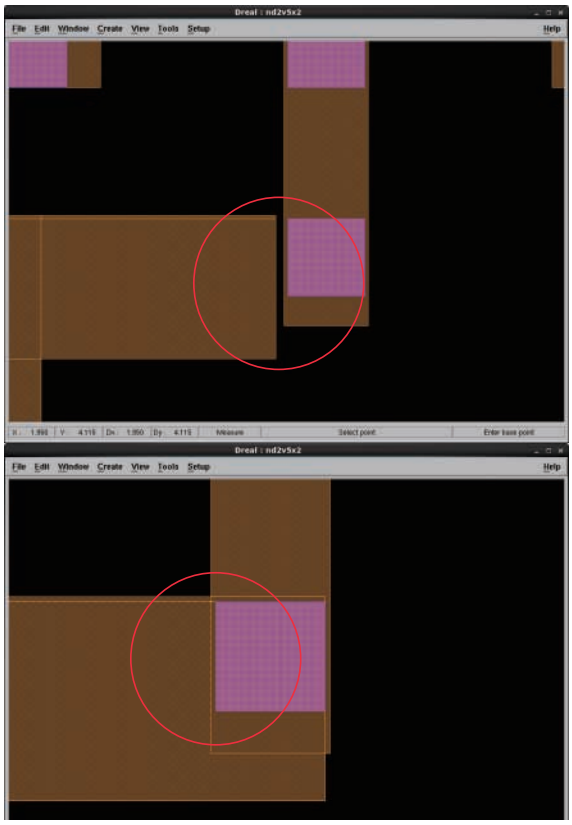


Fig.14　The example routing extention

2.4　レイアウト調整用新プログラムの設計開発

セル間のレイアウトエラーを新規にプログラムを開発して解決した。セル間でインプラントレイヤーの最小間隔エラーが発生している例を Fig.16 に示す。補完後のレイアウトを Fig.17 に示す。以下はプログラムが実行する補完工程である。

1. gds ファイルを S 式に変換
2. S 式で表現したレイアウト情報からセル位置定義とセル内部定義を取得
3. セル内部定義から指定レイヤーを取得
4. セル位置定義からコアを基準としたセグメントの座標を計算
5. セグメント間の距離を計算
6. 補完セグメントをトップセルに書込
7. S 式を gds に変換

1 と 7 は C で実装し、残りは Scheme で実装した。

gds と S 式との相互変換工程 (工程 1, 工程 7) では gds フォーマットと入れ子になった連想リストとを相互変換する。出力される連想リストの例を Fig.18 に示す。

工程 2 ではトップセルに記述されているセルの位置定義とセルの内部定義とを抽出し、別々に保存する。工程 3 では内部定義からセグメント記述を示すキーワードとレイヤー番号とを用いてフィルタリングする。工程 4 は位置定義とフィルタリングしたセグメントとを照会し、コアの原点を基準とした位置情報に統一する (Fig.19)。

工程 5 では近傍のセルとセグメントの距離を計算し、設定された値以下の距離であったら補完箇所として記録する。近傍のセルの検出は x 軸方向と y 軸方向とに分けて行う。x 軸方向では隣合うセルのセグメントとのみ比較する。隣接判定はセル位置情報によって行う。y 軸方向では上下の行について、隣接するセルの原点の x 座標より遠くまで延びているセルとを検出、比較する。x 軸と y 軸との比較を Fig.20、21 に図示する。

工程 6 は記録した補完箇所をもとにセグメントをレイアウト記述に追加する。補完セグメントはトップセルの定義箇所に挿入する。記述は Fig.18 に沿って行う。こうして作成した補完セグメントを含む S 式のレイアウト記述ファイルを GDS 形式に戻し、セル間のセグメント merge 処理が完了する。

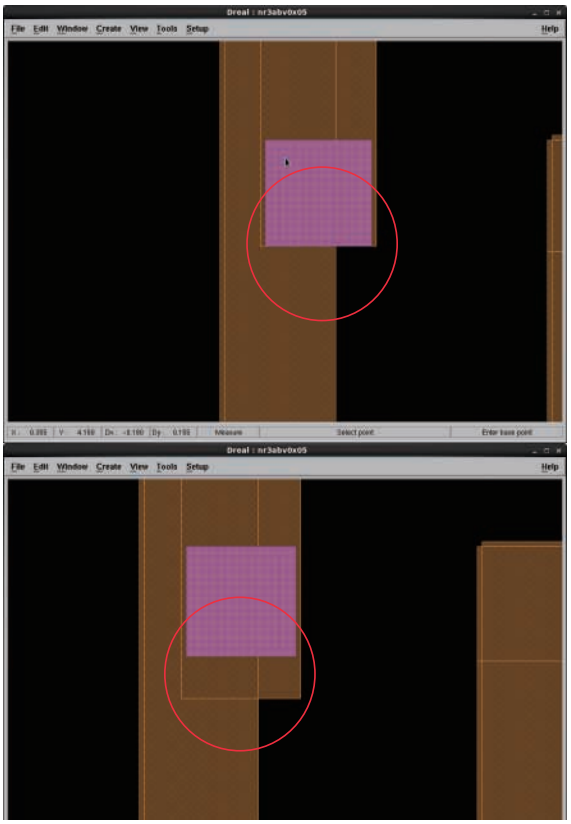


Fig.15　The overlap error example

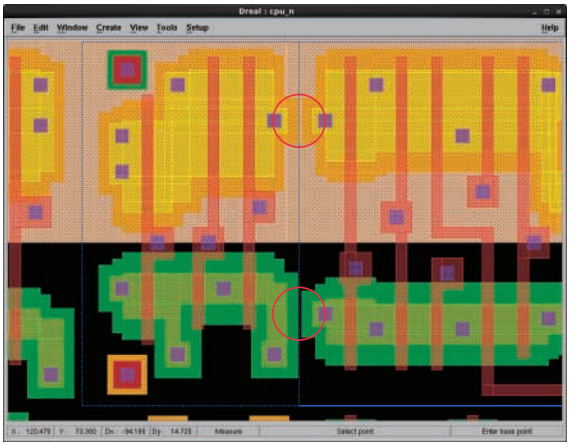


Fig.16　The example minimum distance fault between cells. Blue line is cell border. Red line is area there is fault.

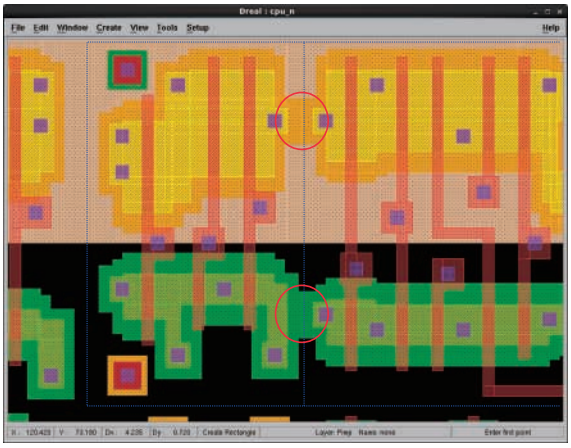


Fig.17　The layout after complement by the our program. complemented area is in the circle red line.

Fig.18　This is association list showing layout files

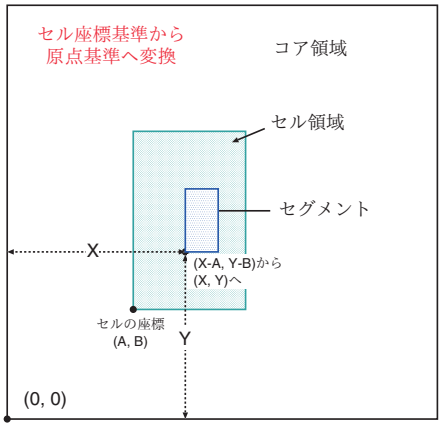


Fig.19　The convert from based on cell position to core position

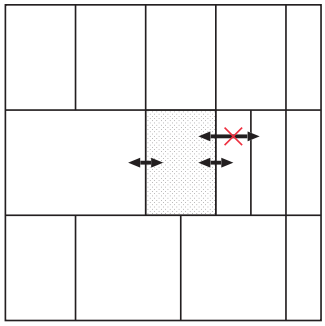


Fig.20　The distance comparison example between neighbor cell align x-axis

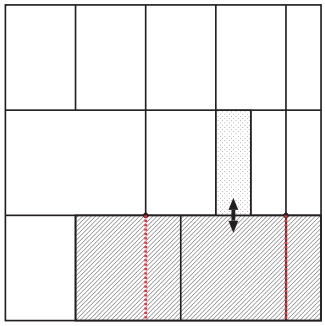


Fig.21　The distanc comparison example between neighbor cell align y-axis 上下の行の中で、x 軸について隣接するセルの原点の x 座標を含むセルのセグメントとのみ比較する

3. デザインルールチェック

これまで開発した手法とプログラム、改造した Alliance を用いてチップレイアウトを設計した。このレイアウトをメンター社 Calibre を用いて DRC を行った結果、デザインエラーがないことを確認した。

4. 考察

商用 EDA と専用セルライブラリとを用いる手法と、本研究の手法とで同じプロセス向けのレイアウトを設計すると、どのセルも新セルライブラリのほうが小さいにもかかわらず、本手法のレイアウト面積は商用 EDA で設計したものより約 50% から 60% ほど大きくなってしまう。レイアウトの面積を Table 3 に、セル面積の比較を Table 4 に示す。この問題を考察するため、双方のレイアウト全体で使用されているセルの数を比較した。その結果新手法のほうが必要な余白も使用しているセル数も多いことが判明した。余白なしで配置可能であり、かつ使用するセルがどちらも同数であると仮定して面積比較を行うと、新手法で設計したレイアウトは商用 EDA で設計したものと同程度の面積であった。それぞれのレイアウトの面積比と余白・セル数を考慮した推定値との比を Fig.22 に示す。なお、ターゲットとして Rohm 0.18μm プロセスのデザインルールを用いた。

それぞれのセルの面積では vsclib のほうが小さいにもかかわ

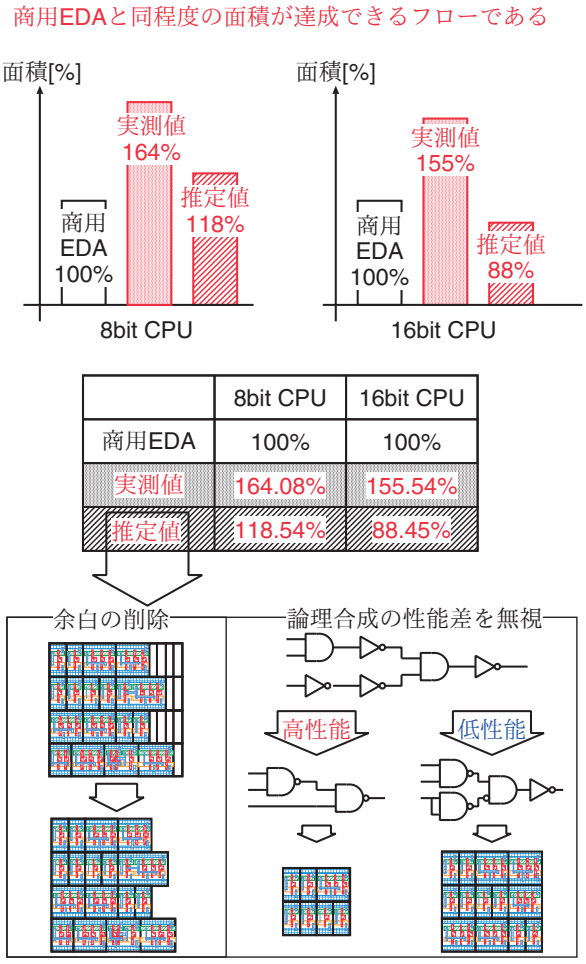


Fig.22　Layout area designed by new method and estimated value in case that cell number is same

Table 3　The chip area designed by new method in case area of chip developed by commercial library as 100%

	8bit CPU	16bit CPU
新ライブラリ	164.08%	155.54%

Table 4　The relative cell area in the vsclib compared to the commercial cell. The unit is percent.

	vsclib		vsclib
2 入力 and	83.25%	2 入力 or	83.25%
3 入力 and	96.83%	3 入力 or	98.53%
4 入力 and	98.98%	4 入力 or	98.98%
2 入力 nand	80.02%	2 入力 nor	80.02%
3 入力 nand	83.25%	3 入力 nor	83.25%
4 入力 nand	98.53%	4 入力 nor	98.53%
インバータ	75.49%	2 入力 xor	89.06%

らずレイアウトの面積が専用ライブラリに劣る理由は、論理合成の性能差によって使用するセルの数が異なるからである。使用するセルの数の比を Table 5 に示す。

Table 5 The cell numbers comparison between chip designed by commercial cell library and our method as commercial one is 100 %

	8bit CPU	16bit CPU
新手法	146.55%	115.35%

そこで、論理合成の性能が同等であり、配置のための余白が必要でない場合の面積比較を考える。予想面積比率は Table 6 のようになる。

Table 6 The chip area comparison as one designed commercial library is 100%. Result considering mergin and cell number is below line.

	8bit CPU	16bit CPU
実測値	164.08%	155.54%
予測値	118.54%	88.45%

この結果より、論理合成の性能を上げ、配置の余白を削減できれば、本手法は専用セルライブラリに劣らない面積のレイアウト設計ができると予想される。

5. まとめ

我々は安価にディジタル ASIC を設計できる手法を、オープンソース EDA ソフトウェアとλルールベースセルライブラリとを用いて開発した。開発した手法を用いて Rohm 0.18μm プロセス向けにレイアウトを設計し、デザインルールチェックを行ったところ、エラーの発生する箇所がないことを確認できた。また、商用 EDA と専用のセルライブラリとを用いて設計したものと比較したところ、約 1.5 倍の面積となった。今後の改良によってほぼ同等の面積を達成できる可能性があることがわかり、先行研究の課題であった面積削減を達成できたと言える。同時に、論理合成と配置工程とに課題が残ることも判明した。そのため、配置するセルの合計面積だけでコアを作ることができる配置ソフトウェアと、セル数を削減できる論理合成ソフトウェアとの開発が今後の課題となる。

6. 謝辞

本研究は東京大学大規模集積システム設計教育研究センターを通し、メンター株式会社の協力で行われたものである。

参考文献

1) Semiconductor capacity urilization reports. http://www.sia-online.org/industry_statistics/semiconductor_capacity_utilization_sicas_reports/.
2) Carver Mead and Lynn Conway. *Introduction to VLSI Systems*. Addison-Wesley Publishing Company, 1981.
3) 樋口拓哉, 山本小太郎, 大金淳一郎, 北嶌卓也, 末広尚義, 清水尚彦. 0.18μm プロセスルールにおけるオープンソース cad ツールを用いた lsi 設計環境. 第 35 回パルテノン研究会, Vol. 35, pp. 17–23, 2010.
4) Asic standard cell library design by graham petley. http://www.vlsitechnology.org/.
5) Alliance home page. http://www-asim.lip6.fr/recherche/alliance.
6) J. Grad and J.E. Stine. A standard cell library for student projects. In *Microelectronic Systems Education, 2003. Proceedings. 2003 IEEE International Conference on*, pp. 98–99, 2003.
7) HIDETOSHI ONODERA, AKIO HIRATA, TERUO KITAMURA, KAZUTOSHI KOBAYASHI, and KEIKICHI TAMARU. P2lib:process portable library and its generation system (special issue on design technologies and design automation of electronic systems). *IPSJ Journal*, Vol. 40, No. 4, pp. 1660–1669, apr 1999.
8) Hiroaki YOSHIDA and Masahiro FUJITA. Automatic generation of design-specific cell libraries. *IEICE technical report. Dependable computing*, Vol. 109, No. 316, pp. 179–184, nov 2009.
9) 石黒健史. Alliance の scalable cmos library によるチップ試作の研究. 東海大学工学部卒業研究論文, 2003.
10) 樋口拓哉, 細川達也, 今井紘士, 清水尚彦. λ ルールスケーラブルセルライブラリの 0.18μm プロセスへの実装試行と teg チップ開発. 東海大学紀要情報通信学部, Vol. 4, No. 1, pp. 19–25, 2011.