東海大学紀要情報通信学部
vol.4,No1,2011,pp.19-25

論文

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

樋口 拓哉[*1]細川 達也[*1]今井 紘士[*2]清水 尚彦[*3]

The Implementation Trial and the TEG Chip Development of The Lambda Rule Scalable Cell Libraries to Rohm0.18μm Process

by

Takuya HIGUCHI[*1], Tatsuya HOSOKAWA[*1], Hiroshi IMAI[*2] and Naohiko SHIMIZU[*3]

(Received on April 26, 2011 & Accepted on September 9, 2011)

Abstract

We designed for a independent chip layout to the process scale to adopt the EDA tool having scalable cell libraries. Originally, this tool can not apply to deep sub-micron process. Therefore we improved this tool to design Rohm0.18μm process and perform to implementation trial. In this process, we designed for the TEG chip and verified it.

Keyword: A TEG Chip, Implementation Trial, Lambda Rule Cell Library

1. はじめに

LSI(Large Scale Integration) 開発の規模は増大し、短期間での開発が求められるようになっている。現在の設計手法での大規模な開発は記述量、可視性等の面から問題があり、効率のよい上位の設計手法、開発環境がもとめられている [1)]。

この問題に対し我々は高位技術を利用したハードウェア記述言語 NSL(Next Synthesis Language)[2)] と OSSEDA(Open Source Software Electronic Design Automation) ならびにスケーラブルセルライブラリとを用いて新しい LSI 設計手法を開発した。セルライブラリを用いる設計ではあらかじめ配置配線を行ったセルと呼ばれる論理ゲートを数十から数百種用意する必要がある。セルは各プロセスの設計規則に基づいて人間が注意深く設計しなければならない。また、セルは種類が多く、セルの性能が最終的に完成する LSI の性能に直結する。そしてセルごとにゲート遅延と浮遊容量、面積、入出力端子を定義する定義ファイルを作成する必要もある。さらにそれぞれのセルの検証を回路シミュレータで検証するため、スタンダードセル設計において、セルの設計には高度な知識と膨大な工数を要する。また、セルはプロセス独自のものであり、同じ論理でも搭載するプロセスが異なれば新たなセルライブラリを用意しなければならない。この問題に対して新たにセルライブラリを作り直すことなく、異なるプロセスに適応できるセルライブラリがスケーラブルセルライブラリである。スケーラブルセルライブラリとはトランジスタのゲート幅を λ もしくは 2λ とし、メタルやポリシリコンなどのセグメント間隔を 0.5λ もしくは λ の整数倍で表したセルライブラリを指す [3)]。この手法により Fig. 1 に示す Onsemi1.2μm プロセス、Fig. 2 に示す Rohm0.35μm プロセスのチップを実際に試作し、開発手法の確立を行ってきた。試作したプロセスの詳細を Table.1. にまとめた。Rohm、ONSEMI は LSI 生産受託会社の社名を表し、数字はその LSI に集積されているトランジスタのゲート幅を表している。今回我々は Rohm0.18μm プロセスを対象に LSI の試作を行った。このプロセスへの回路実装に必要なコストや工数の削減を目指した。

我々の設計手法は NSL で回路を設計し、これを OSSEDA ツールセットを用いてチップレイアウトに変換するものである。詳細な設計フローを 2 章に示す。

しかし、スケーラブルセルライブラリはディープサブミクロンプロセスに対応しきれない。今回採用した EDA ツールの開発元である LIP6 は、この EDA ツールを使った開発は 0.8μm プロセスまでしかできていない [4)]。この問題に対する解決法を 3 章に示す。

我々は配置配線検証を EDA ツールセット付属の検証ツールで行った。付属の検証ツールはそのままでは使用できない。これはツール使用法についてのドキュメントが整備されておらず、ルール記述法が全く不明であるためである。また、コマンドやオプションについても機能だけでなく用意されている数も不明であった。検証ツールの問題点と解決法を 4 章に示す。

実際に我々が提案する手法で TEG(Test Element Group) チップレイアウトを設計し、動作評価を行った。TEG チップには評価用の回路を実装し、様々なテストを行った。評価内容を 5 章に詳しく述べる。

また、我々の手法で設計したチップレイアウトと LSI 生産受託会社が提供するセルライブラリで設計したチップレイアウトとを比較、評価した。評価項目はチップ面積とゲート遅延時間である。この評価の詳細を 6 章に示す。

[*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
[*3] 専門職大学院 組込み技術研究科

Table.1　Processes summary

Process	Metal	Poly	Gate(μm)
Rohm 0.35μm	3	2	0.35
Rohm 0.18μm	5	1	0.18
ON SEMI 1.2μm	2	2	1.2

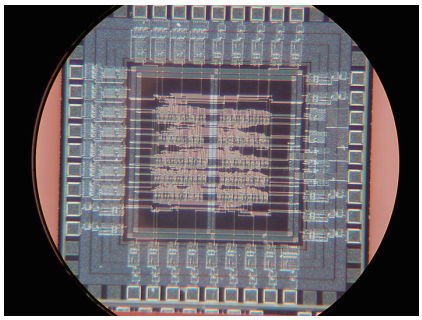


Fig.1　On semi1.2μm

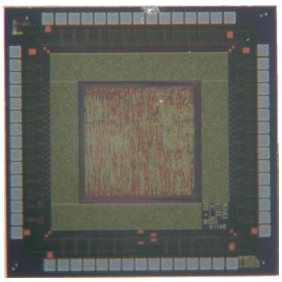


Fig.2　Rohm 0.35μm

し、ツールセットが定義した VHDL サブセットに変換する。これをモジュールごとに動作記述ファイルと構造記述ファイルとに分割する。動作記述ファイルは回路をデータフロー形式で記述したものであり、構造記述ファイルは各論理がどのように結ばれているかを示す回路の配線を記述したものである。この動作記述ファイルをスケーラブルセルライブラリを用いて論理合成する。論理合成した回路を元にチップ上に対応するセルを配置し、配置したセルどうしを配線することによってλを単位とした仮想レイアウトが生成される。そして仮想レイアウトのλに実際のゲート幅の数値を与えることによって実際の配置配線を示したチップレイアウトが完成する。

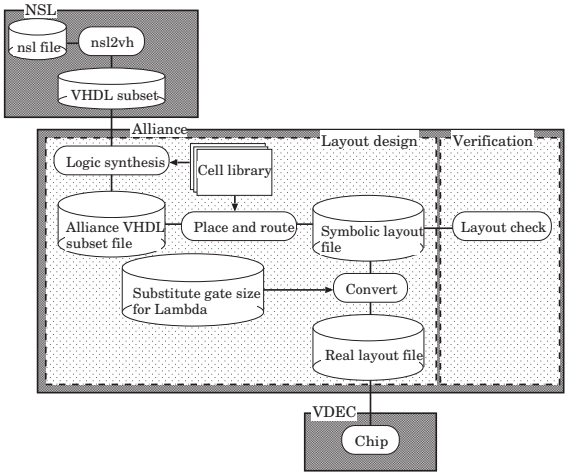


Fig.3　The flow chart of all process

2.　設計フロー

我々は、一つのレイアウトパターンから複数のプロセスに適合可能なスケーラブルセルライブラリを用いた設計手法を開発した。設計手法の全体像を Fig.3 に示すフローチャートを交えて述べる。

まず、設計手法に取り入れた回路記述言語 NSL と Alliance VLSI CAD System[5] との特徴を説明する。NSL は抽象度の高い記述が可能である。そのため開発者は下位記述に触れずに大規模かつ高機能な LSI 設計を行うことができる。また、NSL 合成ツールは Alliance が定義した VHDL(VHSIC(Very High Speed Integrated Circuits)Hardware Description Language) サブセットだけを用いた VHDL への変換をサポートしているため、NSL コードによる回路設計から複数のプロセスのレイアウト生成までをシームレスに行うことが可能である。Alliance はスケーラブルセルライブラリとセグメント調整機構を持つ。論理合成や自動配置配線にはスケーラブルセルライブラリを適用する。生成した、λを単位とする仮想レイアウトに対してそれぞれのセグメントを調整できる。また、独自の配置配線検証用ツールを備え、VHDL からチップレイアウト生成までを全て Alliance のみで行える。かつ、Alliance は教育用途に適している [6]。

次に、設計手法の全体像を述べる。まず、NSL で回路を設計

3.　ディープサブミクロンへの適用

我々の提案するチップレイアウト設計手法を確立するためにはスケーラブルセルライブラリをディープサブミクロンへ適用させる必要がある。我々はスケーラブルセルライブラリをディープサブミクロンに適用させるためにスケーラブルセルの変換ルールに基づいて λ の値を調整した。また、ツールセット中のセグメント幅調整機構を用いてセグメントのサイズを直接調整した。しかし調整しきれない箇所が存在したため、セルライブラリの内部を直接調整することで解決した。作業の概略を Fig.4 に示す。

3.1　スケーラブルセルの変換ルール

λベースセルの変換には、スケーラブルセルを配置配線したレイアウトと、各レイヤの変換に必要な値を使用する。Fig. 5 にトランジスタのレイアウト変換例を示す。左の表が変換に必要な値、右が出力されるレイアウトと寸法を表している。表内の $Segment$ は、スケーラブルセル内に宣言しているレイヤの種類を表している。Fig. 5 の例の TRANS はトランジスタを表している。Ls、Ws は、セグメントの長さをλ単位で示しており、この長さが配置するレイヤ寸法の基準となる。下の表には、各セグメントの配置するレイヤを示しており、対応する DLR と DWR

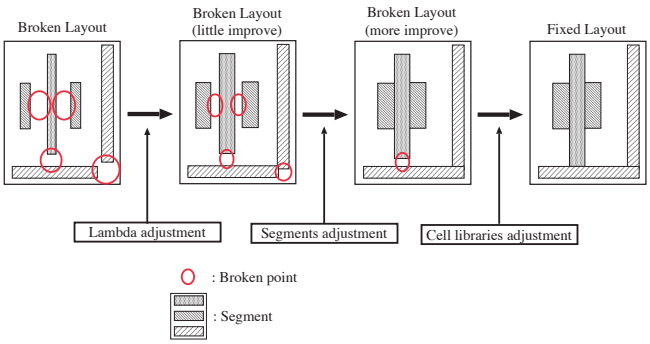


Fig.4　The flow chart of applying scalable cell libraries to deep submicron

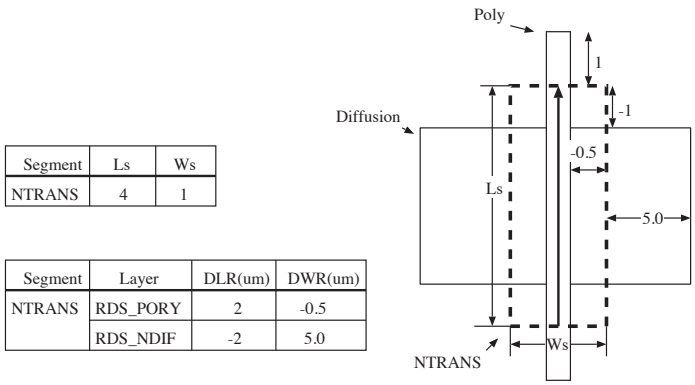
の値は、λ値で生成したレイヤの幅と長さのオフセットを表している。変換後のレイヤの長さは、以下の数式を基に決定される。

$$Lr = Ls\lambda + 2DLR \tag{1}$$

$$Wr = Ws\lambda + DWR \tag{2}$$

例では、トランジスタに配置する Polysilicon、Diffusion の宣言と、オフセットの値を示している。

すべてのセグメントに配置するレイヤを宣言し、スケーラブルセルから実際の大きさのレイアウトを Fig. 6 のように生成する。Fig. 6 は反転回路の変換例で、左はスケーラブルセルのセグメントの配置、右は実際の変換後のレイアウトとなる。



Segment	Ls	Ws
NTRANS	4	1

Segment	Layer	DLR(um)	DWR(um)
NTRANS	RDS_PORY	2	-0.5
	RDS_NDIF	-2	5.0

Fig.5　Ex-Symbolic to Real layout convert

3.2　SCMOS レイアウトを用いた Rohm0.18μm レイアウト設計

ディープサブミクロンプロセスは、サブミクロンプロセスに比べて設計規則が複雑になっている。Rohm0.18um の設計ルールは NDA により開示できないため、MOSIS[7] に記述されている SCMOS と DEEP 向けのデザインルールを Fig. 7 に示す。SCMOS 向けのレイアウトを DEEP に適合させるためには、ポリシリコン間隔を 3λ にする必要がある。そのため SCMOS 向けレイアウトは全てのサイズを 3/2 倍する事になる。全てのレイアウトサイズが 3/2 倍になったためチップサイズは 2.25 倍となり、トランジスタの Ids は 1/1.5 となる。すなわちチップサイズが大きくなり、ゲート速度が遅い LSI となる。

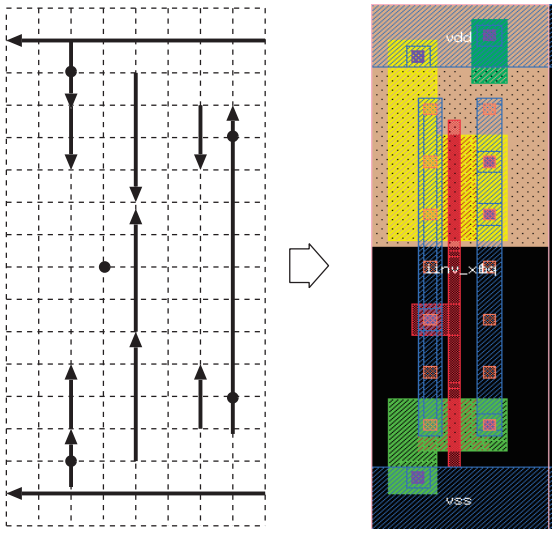


Fig.6　Ex-Symbolic to Real layout convert

この問題を解決する方法として、λの値を大きくとりセグメント幅調整機構を用いて最小ゲート幅を設計ルールに一致させる調整をした。調節例を Fig. 8、調節後のレイアウトを Fig. 9 に示す。各セグメント毎に幅、長さをλ単位で決定している。また、オフセット値を設定することで幅を調節することが可能である。Fig. 8 の赤い部分が調節箇所である。しかし、調整しきれない箇所が存在したため、セルライブラリの微調整を行った。Fig .10 の赤い部分に修正例を示す。

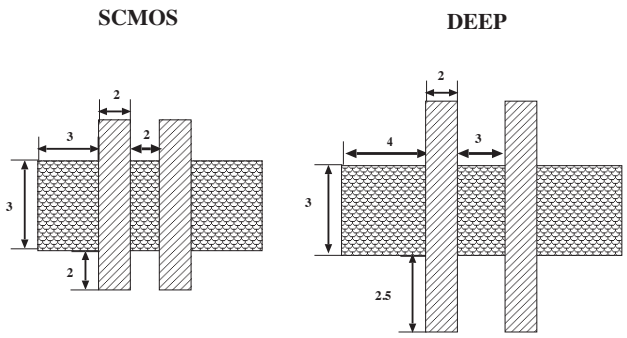
4.　配置配線検証

今回採用した EDA ツールは様々な機能を持つツールの集合体である。このツールセットの中には配置配線検証を行うためのツールも用意されている。しかし、このツールセットにはドキュメントが整備されておらず、検証に不可欠なレイアウトルールの記述法が示されていない。その上、ツール自体の使用方法も説明されていないため実際の使用に耐えないものとなっている。我々はこのツールを用いた検証方法をソースコードを解析することで解明し、配置配線検証を行うためのドキュメントの整備を行っている。実際の配置配線検証方法を以下に示す。

4.1　配置検証

配置検証では Fig.11 のようにセグメントの幅を指定したルールに基づいて検証する。しかし、レイアウトルール記述ができなかったため、このツールセットがオープンソースであることを生かし、ソースコードを解析することで Table.2 のようなルール記述法や種類を特定した。

その結果、このツールが用意しているルールは 56 種類あることが判明した。開発するプロセスが Rohm0.18μm プロセスの場合、これで全 253 種類のルールを表現しなければならない。しかし、レイヤーの総面積やチップ全体に占める特定のレイヤーの割合などをチェックするルールは実現できない。そのため、これらのルールを実現するためには外部ツールで補うか、配置検証



Description	Lambda		
	SCMOS	DEEP	SCMOS=>DEEP
Minimum width of poly	2	2	3
Minimum width of activ	3	3	4
Minimum gate extension of active	2	2.5	3
Minimum active extension of poly	3	4	4

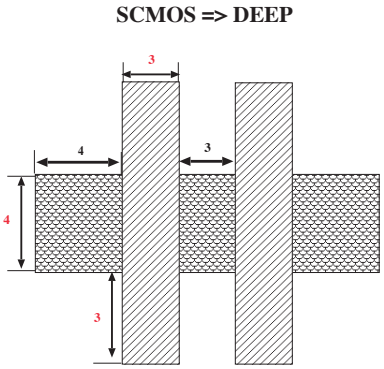


Fig.7　Mosis

ツールのソースコードを書き換える必要がある。また、このツールセットが生成するレイアウトファイルは Fig.12 に示すようにどのような形のレイヤーでも全て長方形のレイヤーの組み合わせで作られている。そのため複数のレイヤーが重なり合うもしくは隣り合うことでルールを満たしている場合、これを検出できない。しかし今回用いたツールセットはそれぞれの長方形のレイヤー一つ一つがルールを満たすように設計されるため、この問題は回避される。

Table.2　Example of converting process rule to Druc rule description

Process rules	Druc rule description
Min. width of A-layer is X micron	longueur_inter > X.
Min. space between A-layer and B-layer is Y micron	distance axiale min Y.

4.2　配線検証

配線検証はネットリスト抽出とネットリスト検証との二段階に分けて行われる。配線検証の全体フローを Fig.13 に示す。以下の節でその工程を示す。

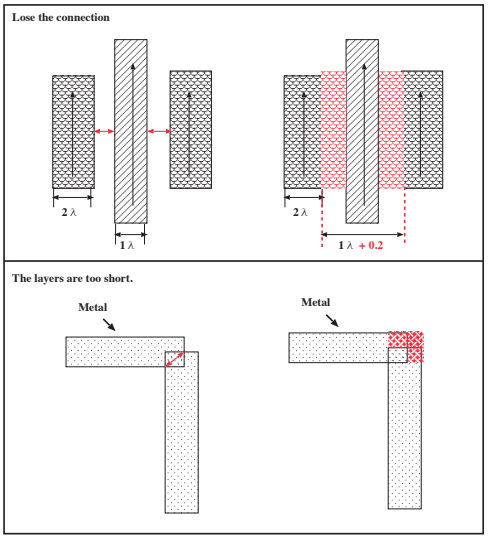


Fig.8　The adjusted examples of the segments

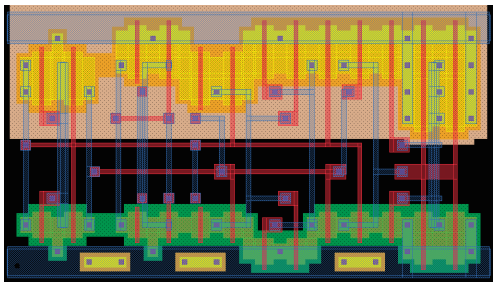


Fig.9　The adjusted examples of the layout

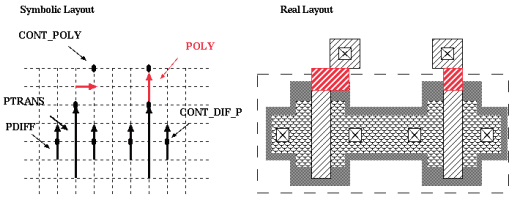


Fig.10　The modification examples of the cell libraries

4.2.1　ネットリスト抽出

ネットリストはツールセットの中のネットリスト抽出ツールによってチップレイアウトから抽出される。この時、レイアウト通りの配線情報が正しく抽出されているかを検証するためにレイアウトの一部分とそこに対応するネットリスト情報を比較した。Fig.14 は、作為的に断線させた箇所のレイアウトとそれから抽出された断線させた箇所のネットリスト情報と、正しく配線されている同一部分のそれらとを比較している。レイアウトが断線している場合、抽出されるネットリストでも断線している事が解る。

今回作成したレイアウトからネットリストを抽出すると配線が断線してしまう箇所が発生した。前述の検証によってネットリスト抽出ツールの不具合ではなくレイアウトの誤りだという

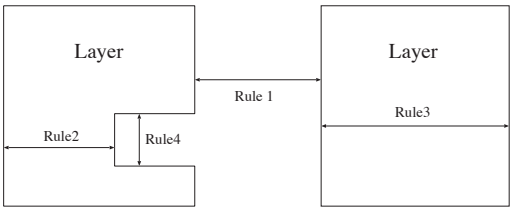


Fig.11　The process need to check rules of those places

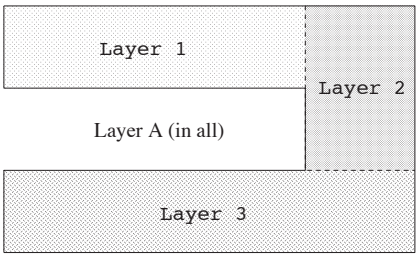


Fig.12　A layer A is composed 3 layers

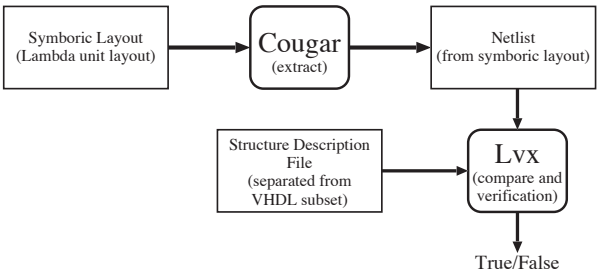


Fig.13　The flow in route verification

ことを断定した。この問題はツールセットが共有する設定ファイルを調整することによって解決することができた。

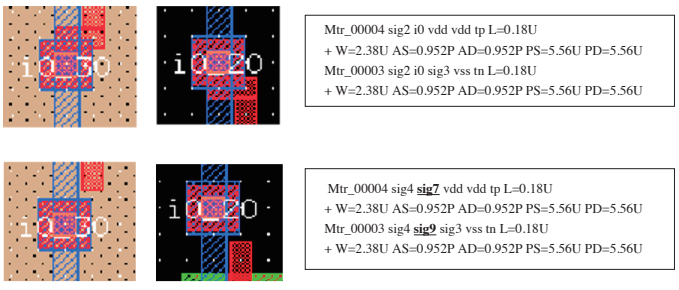


Fig.14　One of the netlist extracted by Cougar

4.2.2　ネットリスト検証

検証パターンはターミナル、インスタンス、コネクションの 3 つであり、どれか一つでも問題があればエラーとなる。Fig.15 で、エラーのパターンを、Fig.16 では、実際に動作させた場合のエラーメッセージを示している。しかし、比較を行えるのがゲートレベルのみで、トランジスタレベルでの検証を行なえない問題がある。我々は解決策として、Spice シミュレーションを用いて使用するセルライブラリをトランジスタレベルで検証し、同一のセルライブラリを使用したネットリストを、ゲートレベルでの検証を行なうことで整合性を保つ手法を考案した。

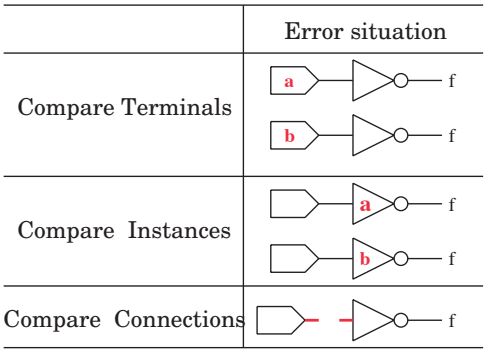


Fig.15　Circuit verification flow

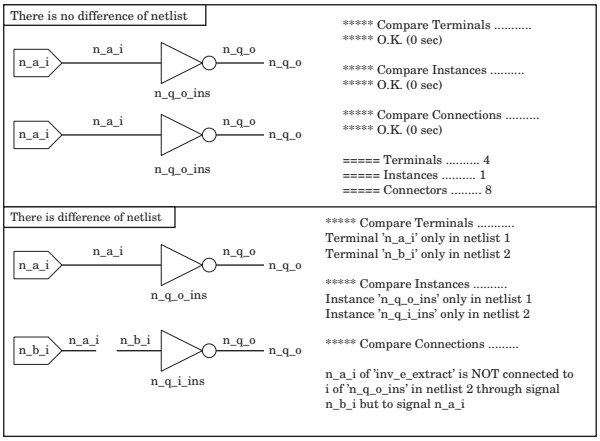


Fig.16　Difference of message between legal netlist and illegal one

5.　TEG 回路

我々はスケーラブルセルライブラリを用いて設計したレイアウトを VDEC を通して試作した。設計したレイアウトの動作評価を行うため、評価回路を実装した。実際に搭載した回路を Table. 3 に示す。実装した回路は、すべてのセルが正常に動作可能であるか確認するための評価回路と、チップ内のクロックスキュー計測用のラッチ回路、動作周波数の性能評価のための 50 段インバータ + 起動用 NAND のリングオシレータ、中規模順序回路 (メモリ) とを実装した。実装した回路を合わせて入出力を含めた順序回路のテスト回路とした。実際にテープアウトしたマスクパターンを Fig. 17 に示す。

6.　評価

我々の手法を用いて設計した基本回路の評価を行った。Table. 6. には、計測時の負荷容量、設計したレイアウトの出力遅延時間、Rohm 提供セルとの比率を示している。3 入力の NAND、NOR 回路では、Rohm 提供のセルに対して回路性能が大きく劣る。し

Table.3　on-board circuit

回路	テスト内容
全セルテスト	スケーラブルセルライブラリ内全セルの動作確認
リングオシレータ	動作周波数の性能評価
スキュー測定用ラッチ回路	チップ内部で発生するスキューの計測
メモリ	中規模順序回路の動作確認

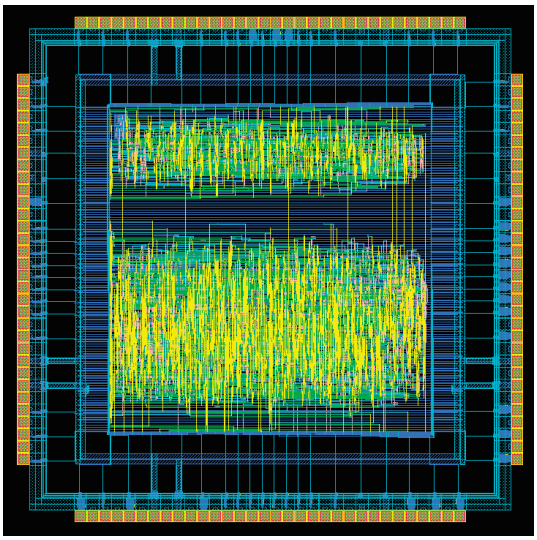


Fig.17　Chip layout of Rohm0.18μm

かし、それ以外の回路はほぼ同等のゲート遅延となった。回路性能が劣る回路に対しては今後詳細な原因追求を行う予定である。

logic	Capacitance(pF)	Delay time of SCMOS (pS)	ratio of SCMOS to Rohm cells
inverter	0.005	22	+10%
2-I/P NAND gate	0.005	57	+14%
3-I/P NAND gate	0.008	146	+83%
4-I/P NAND gate	0.009	185	+58%
2-I/P NOR gate	0.005	62	+4%
3-I/P NOR gate	0.009	204	+65%
4-I/P NOR gate	0.008	228	-37%
2x2-OR into 2-NAND gate	0.005	161	-13%
2x2-AND into 2-NOR gate	0.005	208	+37%

Table.4　delay time of standerd cell

	SCMOS	Rohm0.18μm
m8 (8bit CPU)	114105.6μm^2	6467.616μm^2
snx(16bit CPU)	187920μm^2	28108.9μm^2

Table.5　Comparing the chip core in area

7.　まとめ

我々はオープンソースで提供され、スケーラブルなセルライブラリを持つ EDA ツールセットを用いて 0.18μm プロセスの設計試行を行った。本来、このツールセットは 1.0μm から 0.5μm までのスケーラビリティしか保証していなかったため、0.18μm プロセスを設計する際に様々な不具合が発生した。だが、これらの

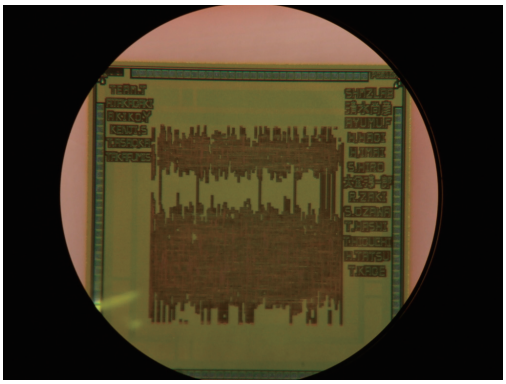


Fig.18　Chip figure of trial manufacture

不具合を解消し、ツールセット内の検証用ツールを用いた配置配線検証法を確立することによってこの EDA ツールセットのみで 0.18μm プロセスを試作する事が可能になった。Fig.18 に実際に作成したチップ写真を示す。

これによって、我々は 1.2μm、0.35μm、0.18μm の 3 つのプロセスがこの EDA ツールセットで試作可能であることを実証した。これは λ ルールベースのセルライブラリでディープサブミクロンプロセスの設計を行うことができるということを示している。

また、LSI 生産受託会社の提供するセルライブラリで設計したチップレイアウトと我々の方法で設計したチップレイアウトをチップ面積とゲート遅延との項目について比較、評価した。その結果、我々の方法で設計したチップレイアウトは LSI 生産受託会社の提供するセルライブラリで設計したチップレイアウトと遜色ない性能をだせることがわかった。

我々は過去にクロックツリーの自動生成手法を確立している[8)9)]。そのため、今後はクロックツリー生成ツールをデザインフローに組込み、実用的なディープサブミクロンのチップ製造を行う予定である。また、グラハム・ピートレイが 0.13μm 互換のセルライブラリを作成している。今後、グラハム・ピートレイのセルライブラリの導入を検討している。

8.　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、日本ケイデンス株式会社、メンター株式会社、シノプシス株式会社の協力で行われたものである。

参考文献

1) International Technology Road-map for Semiconductors 2009 Edition Design(JEITA 訳), p.8
2) http://www.overtone.co.jp/
3) C. Mead and L. Conway, Introduction to VLSI Systems, Addison-Wesley, 1980
4) Greiner. A, Lucas. L, Wajsburt. F and Winckel. L, ”Design of a High Complexity Superscalar Microprocessor with the

Portable IDPS ASIC Library”, pp.9-13, European Design and Test Conference, 1994
5) http://www-asim.lip6.fr/recherche/alliance/
6) Elias Kougianos, Saraju P. Mohanty, Priyadarsan Patra, ”Digital Nano-CMOS VLSI Design Courses in Electrical and Computer Engineering through Open-Source/Free Tools”, pp.265-270, 2010 International Symposium on Electronic System Design, 2010
7) http://www.mosis.com/Technical/Designrules/scmos/scmos-main.html
8) T. higuchi, J. Ogane, and N. Shimizu, ”Develop a design flow for deep sub-micron process(0.18um) with a scalable cell library”, pp.958-960, ITC-CSCC2010, 2010
9) 樋口拓哉, 大金淳一郎, 清水尚彦, ” オープンソース CAD システム Alliance へのクロックツリージェネレータの開発と ROHM0.18 μmテクノロジを用いたチップの試作”, pp.9-14, DA シンポジウム 2010 論文集, 2010