

# スケーラブルなセルライブラリを用いたLSI試作のデザインフロー設計試行(ON SEMI 1.2 $\mu$ m)

樋口 拓哉\*<sup>1</sup> 大金 淳一郎\*<sup>1</sup> 清水 尚彦\*<sup>2</sup>

## Building A Design Flow for ON SEMI 1.2 $\mu$ m Digital LSI with A Scalable Cell Library

by

Takuya HIGUCHI, Jun'ichiro OGANE and Naohiko SHIMIZU

(received on May 8, 2010 & accepted on September 21, 2010)

### Abstract

We have built design method for submicron LSI with three or more metal layers those utilize a scalable cell library within Alliance VLSI CAD System. We confirmed that there were no design rule errors for our Rohm 0.35 $\mu$ m and 0.18 $\mu$ m digital LSI TEGs with our method. In this paper, we try to build a new method for ON SEMI 1.2  $\mu$ m digital LSI which has only two metal layers with the same scalable cell library. The chip is on the way of fabrication.

**Keywords:** LSI design, Alliance VLSI CAD System, VDEC, Design flow, ON Semiconductor(1.2 $\mu$ m)

### 1. はじめに

LSI 設計は最終的にチップ製造のためのデザインルールに従うマスクパターンを作成する作業である。デザインルールは製造プロセス毎に異なるので、通常、プロセス毎にデジタル回路の基本回路パターンであるセルの一群(以下セルライブラリ)を作成する。論理回路の基本素子となるセルの設計によって、LSI の性能が大きく左右されるため、セルの設計は熟練した技術者によって細心の注意の元に行われる。さらに、各セルは、消費電力、遅延時間、負荷特性、回路シミュレーションモデルの検証を十分行い、EDA ツールへのライブラリとして登録可能な形にまとめる。プロセス毎に発生するこの大きな設計負担を低減するため、我々は異なるプロセスでもスケーラブルに対応可能なセルライブラリを有する EDA ツール Alliance によるチップ試作を行ってきた。この手法ではセルライブラリにおけるセグメントの長さ、幅をチューニングパラメータにより調整するルールファイルを作成し、スケーリングによりレイアウトの最適化を行う。しかし、Alliance の基本設計が行われた時期と、現在の LSI プロセスではデザインルール自体が大きく変わりつつあり、ツールにまかせたスケーリングではデザインルールエラーが多数発生する。そこで我々は、過去、Rohm 0.35 $\mu$ m, 0.18 $\mu$ m プロセスを用いて、実際に LSI 設計を行い、デザインルールエラーのないレイアウト作成のための手法を研究してきた。<sup>3)7)</sup> これらの研究成果により、サブミクロン、ディープサブミクロンプロセスにおける Alliance の問題点が明らかにされてきたが、個々のプロセスごとにライブラリに変更を加える必要が生じるとスケーラブルライブラリの価値が低減する。また、Alliance ではライブラリ以外に、プログラム中にデザインルールを仮定して配置配線処理を

行っているため、ライブラリ自体がスケーラブルになっても配置配線でエラーが発生する場合があった。我々は、従来、配置配線ツールが生成するエラーは、最小プロセスピッチを拡大することで対応していたが、若干の LSI レイアウトの拡大を招きあまり好ましい対策ではない。そこで、これらの問題点を整理して、今後の対策を検討する必要があった。今回、従来行ってきたディープサブミクロンプロセスである Rohm 0.35 $\mu$ m, 0.18  $\mu$ m に加え、サブミクロンプロセス、ON SEMI 1.2 $\mu$ m の試作を行い、デザインフローの確立をした。過去のディープサブミクロンプロセスと異なり、ON SEMI 1.2 $\mu$ m では配線に用いるアルミ層が 2 層となっているため、これまでとは異なる配線方法を取った。手法の変更に伴い発生した新たな問題点の分析を行う。Table 1 に各プロセスの概要を示す。また我々は、座学のみでなく、ものづくりを通して LSI 設計を学ぶことを目的とし、VDEC を利用したチップ試作を試みている。VDEC には設計を行うための各種 EDA ツール、デザインフローが用意されているが、技術者としての将来を見据え、LSI 設計フロー自体を見直し、作りあげる力を養うため、オープンソース EDA ツールを用いた新たな設計フローの確立をもう一つの目的とする。本稿では VDEC チップ試作を通して得られた知見を述べる。

### 2. デジタル LSI 設計

HDL からの一連のデジタル LSI デザインフローを Fig. 1 に示す。HDL の論理合成により得られたネットリストの持つセル情報、およびセル間の接続情報に基づき、セルの配置、配線を行う。

### 3. VDEC チップ試作概要

東京大学大規模集積システムセンター (VDEC) は、教育および学術研究目的に限って、商用ツールをアカデミックライセンスとして無償利用可能とするチップ試作環境を提供している。

\*<sup>1</sup> 工学研究科 情報通信制御システム工学専攻 1 年

\*<sup>2</sup> 情報通信学部組込みソフトウェア工学科 教授

我々は、VDEC が提供したプロセスのうち、Rohm 0.35 $\mu$ m、0.18 $\mu$ m に加え、新たに ON SEMI 1.2 $\mu$ m での試作を試みた。

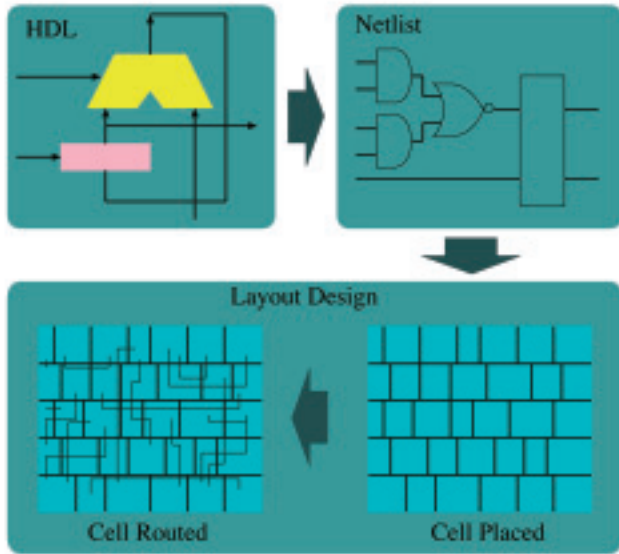


Fig. 1 Digital LSI design flow

VDEC はチップ製造において、大手 EDA ベンダーの Cadence Design Systems, もしくは Mentor Graphics 社のツールによってデザインルールチェックが通っていることをテラアウトの条件とする。

#### 4. OSS EDA Alliance 概要

##### 4.1 Alliance VLSI CAD System

LSI レイアウト設計にはフランスの Pierre et Marie Curie University にて開発されたオープンソース EDA ツール、Alliance VLSI CAD System(以下 Alliance) を使用した。Alliance はセルベース方式で LSI を設計するが、備えられたセルライブラリは欧州の CMP, Euro Practice での 1.0 $\mu$ m チップ試作を前提に作られており、本来は VDEC でのチップ試作には対応していないが、入ルールベースのセルライブラリを各種プロセスルールに適合させるという手段を持つ。そこで、デザインルールに適合させるためのレイアウトを決定するルールファイルを記述し、必要に応じてライブラリの修正を行った。

##### 4.2 Alliance の特徴

Alliance はスケーラブルなセルライブラリを持つ、仮想配置 (symbolic layout) で配置配線を行った後、s2r(symbolic layout to

Table 1 Processes summary

Process	Metal	Poly	Gate( $\mu$ m)
Rohm 0.35 $\mu$ m	3	2	0.35
Rohm 0.18 $\mu$ m	5	1	0.18
ON SEMI 1.2 $\mu$ m	2	2	1.2

real layout) で実配置に変換し、ファウンドリのデザインルールに適合したチップ製作可能なレイアウトデータとする。(Fig. 2 参照)

- 仮想配置… 軸の長さ、座標情報のみのセルライブラリを配置、配線した状態。
- 実配置… ルールファイル(テクノロジーファイル)に従い、ファウンドリに提出できるレイアウトデータを生成した状態。

変換は、ユーザが記述したルールファイルの記述に従い行われる。この手法により、プロセスルール毎に固有のスタンダードセルライブラリを必要としない設計が可能である。

ルールファイルは  $\lambda$ (ラムダ) を基準値とする。Alliance で意味する  $\lambda$  とは、通常のゲート幅ではなく、仮想配置で用いる座標 1 コマの値であり、ゲート幅は独立して宣言する。通常の  $\lambda$  ベース設計と異なり、Alliance では、 $\lambda$  ベースで構成されたセグメントの長さや幅をルールファイルによって修正し、個別プロセスへの最適化が可能である。しかし、標準のライブラリは 1.0 $\mu$ m のプロセスでの入ルールを前提とするので、VDEC の提供プロセスに適合させる上で後述の問題があった。

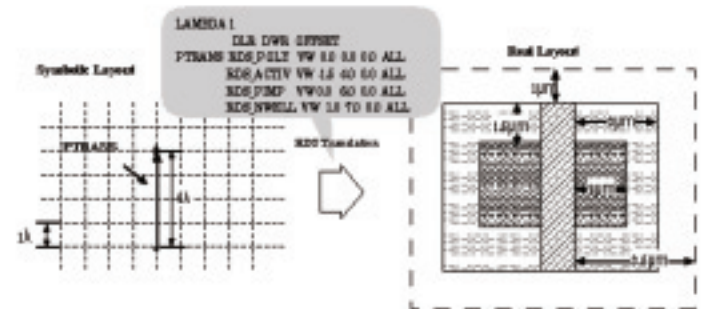


Fig. 2 Symbolic and real layouts

#### 5. 本手法による過去の試作

本研究室では、これまでにディープサブミクロンプロセス、0.35 $\mu$ m、0.18 $\mu$ m プロセスの LSI 設計を、Alliance によるスケーラブルなスタンダードセルライブラリを用いて行った。<sup>5)6)</sup> これらの試作では、各チップ毎に I/O パッドを VDEC の仕様に合わせて独自開発した。また、Alliance が前提とするラムダルールでは対応しきれないルール、RC 遅延によるクロックのスキュー問題等が生じる事が判明した。<sup>7)8)</sup> 試作を進める上でこれらに対処した。

##### 5.1 ディープサブミクロンプロセス設計時の問題点

###### 5.1.1 デザインルール

Alliance のセルライブラリは、スケーラブルなラムダルールによって仮想的に記述される。実際のデザインルールをラムダルールのみで適合させると不必要に大きなチップとなるため、ラムダルールのレイアウトの各サイズをパラメータ (RDS) により増減し、最適設計を行う仕組が導入されている。配置、配線プロ

グラムも同じラムダルールを採用し、RDS パラメータによりレイアウトを調整する。しかし、ディープサブミクロンプロセスでは、 $1.0\mu\text{m}$  向けに作成されたラムダルールでは対応しきれないルールがあったため、仮想配置自体の変更(セルライブラリ変更)が必要となる。一方、配置、配線プログラムが前提とするラムダルールは、パラメータ化されておらず、この調整不良を原因とするエラーが発生した。

### 5.1.2 クロックスキュー

Alliance をはじめとした、多くのオープンソース EDA ツールにはクロック信号が考慮されていないものが多い。ディープサブミクロンプロセスにおいて、Alliance のクロックのルーティング機能ではスキューが大きくなることが考えられる。そのため、高速なデジタルシステム設計を目的とした、クロック分配網の工夫によるスキュー低減を考察した。

## 5.2 対策

### 5.2.1 ライブラリの修正

セルライブラリは、ディープサブミクロンのルールに合わせて修正した。レイアウトエラーを解決し、最終的にデザインルールチェッカーを用いてデザインルールエラーが 0 個になる事を確認した。

### 5.2.2 低スキュークロック自動配線ツールの実装

Alliance のスキューを考慮しないルーティングへの対策として、メッシュ型クロックの自動生成ツールを Alliance ツールチェーンの一部として利用可能な形で実装を行った。メッシュ型クロックは、H-クロックツリーに比べ、Alliance への実装が容易であり、また、セルライブラリを解析した結果、それぞれのセルに対しメッシュを張る方法で、チップ面積、配線率等のコストをかけずに設計が可能であった。

## 6. 本論文の ON SEMI $1.2\mu\text{m}$ プロセスチップ試作

オンセミ  $1.2\mu\text{m}$  ではラムダルール変更は不要と考えられ、パラメータ調整のみの試作が可能となる事を期待して試作申し込みを行った。

### 6.1 ディープサブミクロンとの相違点

ディープサブミクロンプロセスは、配線は 3 層以上使用可能であった。しかし、オンセミ  $1.2\mu\text{m}$  プロセスは、アルミ 2 層である。ディープサブミクロンプロセス設計に使用したデザインフローにおける自動配線は、セルオーバーレーティングであり、2 層配線に対応しない。そのため、配線領域をセル領域外に確保する Alliance Ver.4 を使用することにした。

## 6.2 設計プロセス

オンセミ  $1.2\mu\text{m}$  プロセスをチップ試作する上で、ハードウェア記述から最終的に VDEC で試作可能なレイアウトデータを作成するまでのデザインフローを作成し、それを試行した。

実際に試行したデザインフローを Fig. 3 に示す。

Alliance は、多くの LSI 設計ツールに対する総称であり、今

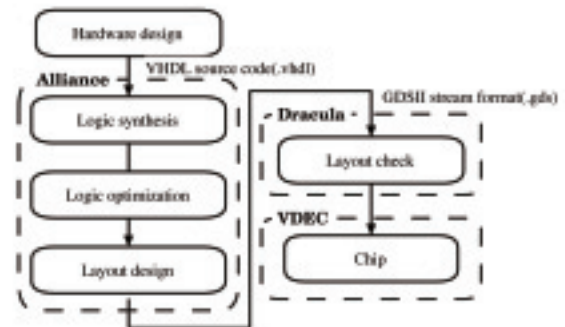


Fig. 3 Design flow

回は、Table 2 に示すツールを用いて VHDL から論理設計、論理合成、論理最適化、自動配置配線を行った。ただし、VHDL の記述レベルは低く設計が困難なため、設計言語は NSL<sup>9)</sup> とし、ns12vh で Alliance 用 VHDL を生成した、NSL から変換し



Fig. 4 Tool flow

た VHDL を、vasy を用いて VHDL ビヘイビア記述へ変換し、boom, boog, loon を用いて Alliance 付属のスタンダードセルライブラリ (sxlbr) に対し、論理合成、最適化を行う。その後、scr を用いて配置配線を行い、仮想配置のレイアウトを出力する。設計した回路の入出力パッドとの接続を記述した VHDL Structure ネットリストと、配置順序指定ファイル (.rin) を作成し、ring を用いてパッドを含めたレイアウト全体の仮想配置を生成する。

tool	function
vasy	VHDL Compilation
boom	Optimization of behavior
boog	Logic synthesis
loon	Optimization of netlist
scr	Placement and routing
ring	Placement and routing for pad ring
s2r	Symbolic to real layout translation
graal	Symbolic layout editor
dreal	Real layout editor
dracula	Design rule checker

Table 2 Tools

最後に、s2r を用いてテクノロジーファイル (.rds) をもとに、レイアウトデータ (.gds) を生成する。

設計されたレイアウトデータは、Cadence 社 Dracula を用いて試作可能か評価する。ツールフローは Fig. 4 の通りである。

### 6.3 日程, 作業の内訳

作成した設計プロセスにおいて、必要な作業を洗い出し、テーブルアウトまでの作業スケジュールを見積もった。その結果、大きく分けてファウンドリ指定の設計規則に適合したレイアウト設計と、組み立てベンダの指定するパッド配置に適合するための、I/O パッドの修正の二つの作業が必要であることが分かった。実際に行った 2010/3/4 からテーブルアウトまでのスケジュールを Fig. 5 に示す。

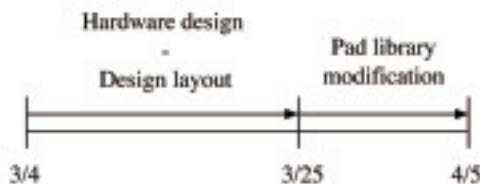


Fig. 5 Schedule

#### 6.3.1 レイアウト設計

今回採用した Alliance におけるレイアウトの作成は、スケーラブルなセルライブラリをテクノロジーファイルに沿って実際のレイアウトへと変換するものである。そのため、このテクノロジーファイルをオンセミ 1.2 $\mu$ m に対応させる事が目的となる。

テクノロジーファイル作成作業は、Fig. 6 の工程で行う。オンセミ 1.2 $\mu$ m 設計規則をもとに、テクノロジーファイルの各レイヤの形状を設定する。作成したレイアウトの確認は、dracula で行い、エラーが存在すればテクノロジーファイルを修正する。スタンダードセルの修正例を Fig. 8(sxlib) に示す。レイヤの凹みや、微妙な隙間でもレイアウトエラーを発生するため、Fig. 8(sxlib) のように、微妙な隙間を無くすように修正する。Fig. 6

の作業で、全てセルライブラリと配置配線後のレイアウトにおいて、デザインルールエラーを 0 個にする。

#### 6.3.2 I/O パッド

パッドはチップ組み立てベンダに依存するため、Alliance 付属のパッドライブラリ (padlib) を VDEC の要求に一致するように修正する。

パッド組み立ての要求は、パッド間隔、チップ面積、パッド配置位置がある。各パッドセルのレイアウトは、I/O バッファと電源リング構成用の配線で構成されており、ring 実行後は、Fig. 9 のように配置される。そのため、使用した各セルのバッファ、電源リングの幅を修正し、配置配線後に指定の位置にパッドを配置する。パッドセルの修正例を Fig. 8(padlib) に示す。

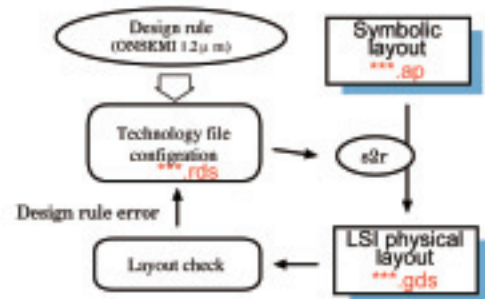


Fig. 6 Layout design

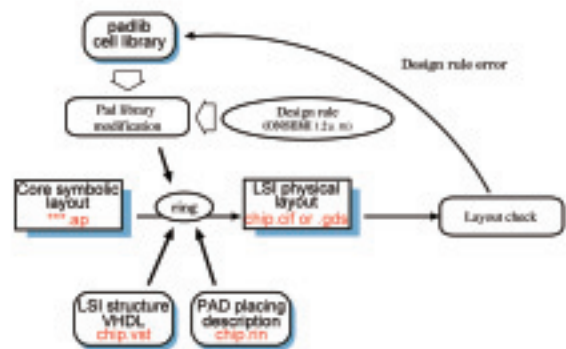


Fig. 7 Pad library modification

### 6.4 設計上の問題点

#### 6.4.1 ポリシリコン上の Via 配置違反

デザインルールにおいて、ポリシリコン上に Via が存在する場合、その箇所のポリシリコンは Via の幅よりも広くとる必要があった。使用したセルライブラリのレイアウトでは、ポリシリコン上に Via を配置する箇所が存在したため、Via の配置位置を修正し、デザインルールエラーを解決した。

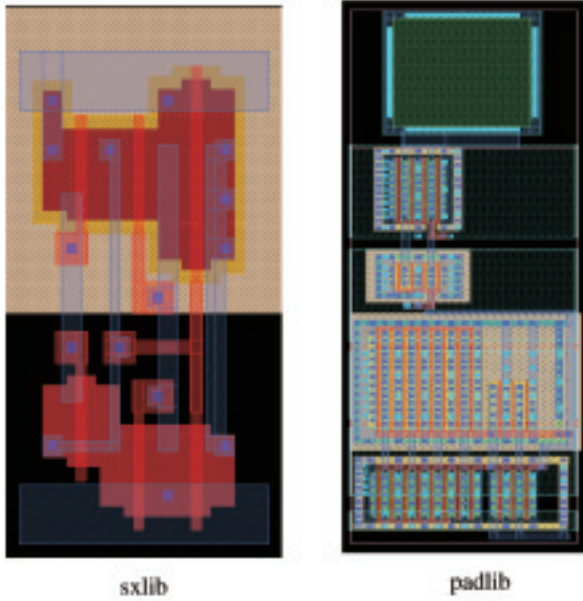


Fig. 8 Cells

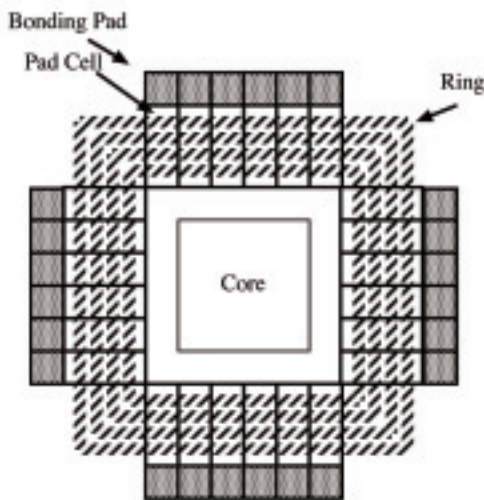


Fig. 9 Chip Layout

#### 6.4.2 チップ規模増大

オンセミ 1.2 $\mu\text{m}$  プロセスはアルミ 2 層であり、アルミ 3 層以上使用しオーバーテイングの Alliance Ver.5 ではなく、セル領域外に配線領域を確保する Alliance Ver.4 を使用した。Ver.4 の自動配線では、セル領域外に配線を確保するため、配線領域が増えてしまうとチップ規模が大きくなってしまふ。そのためチップ搭載可能な面積を越えてしまったため、縮小した回路を搭載した。

#### 7. 考察

今までの各製造プロセスにおいて、発生した問題の改良を考察する。過去のディープサブミクロンプロセスと、今回試作したオ

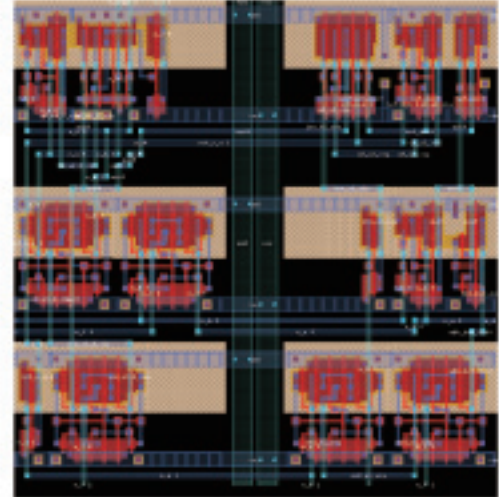


Fig. 10 Routing

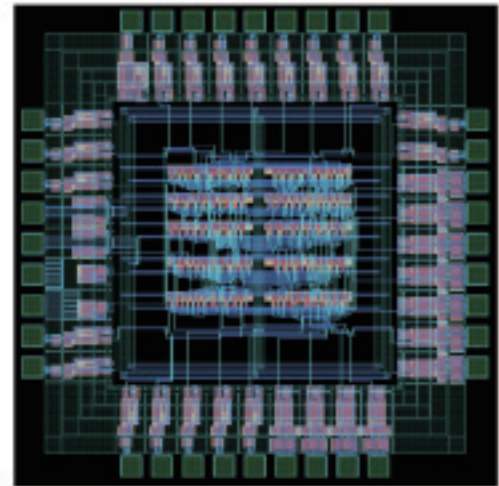


Fig. 11 Layout on Alliance deal

Process	error	provision
Rohm 0.35 $\mu\text{m}$	Unconnected between segment and contact	Segment addition
	Spaced error by different layer	Move the segment and contact
Rohm 0.18 $\mu\text{m}$	Unconnected between segment and contact	Segment addition
	clock skew	clock mesh
ON SEMI 1.2 $\mu\text{m}$	Set Via over Poly	Move the ref
	Scale expansion from low-metals routing	Cut down on circuit

Table 3 errors

ンセミ 1.2  $\mu\text{m}$  の試作の際に利明した、スケラブルルールの問題点を Table 3 に示す。問題点のほとんどは、セルライブラリのセグメント、コンタクトの配置位置によるものであり、ライブラリの修正により対処が可能であるため、各製造プロセスにおいて共通となるスタンダードセルライブラリの作成は可能であると考えられる。また、2 層の配線方法においてもセル配置位置の改善により、搭載規模の縮小が見込める。

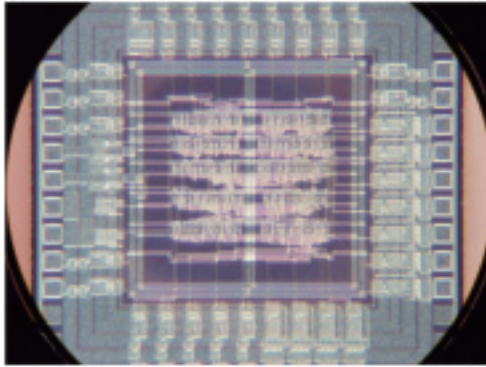


Fig. 12 Real layout photography

## 8. まとめと課題

今回作成したチップ写真を Fig. 12 に示す。本研究では、スケーラブルなセルライブラリを用いた EDA ツールを用い、オンセミ 1.2 $\mu$ m プロセスのチップ試作を行った。

結果、ディープサブミクロンプロセス、オンセミ 1.2 $\mu$ m の各プロセスにおいて、同一のスタンダードセルライブラリをベースとし、若干の修正により、デザインルールエラーのないレイアウトの設計が可能であることを実証した。

しかし、試作したチップは到着したばかりであるため、動作検証を行っていない。一連の試作によって、Alliance のスケーラブルルールの問題点が洗い出されてきたので、今後ディープサブミクロンを含む多くのルールに対してスケーラブルとなるセルライブラリ、ならびに配置配線ツールを開発。試作したチップの実機テストが今後の課題である。

## 9. 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の協力で行われたものである。

## 参考文献

- 1) Alliance Home Page, <http://www-asim.lip6.fr/recherche/alliance/>.
- 2) VDEC Home Page, <http://www.vdec.u-tokyo.ac.jp/>.
- 3) 清水尚彦, SFL による LSI 設計入門, IP ARCH. Inc, 2003.
- 4) 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.
- 5) 飯田佳洋, システム LSI 統合開発環境の構築と同環境を用いたコンピュータシステムアーキテクチャ設計手法に関する研究, 東海大学大学院修士論文, 2004.
- 6) 飯田佳洋, 清水尚彦, 商用ソフトを用いない SFL ベースの LSI 開発, 第 24 回バルテノン研究会, 2004.

- 7) Takuya Higuchi, Jun'ichiro Ogane, Naohiko Shimizu, "Building A Design Flow for Deep Sub-micron Process(0.18 $\mu$ m) with A Scalable Cell Library - Adjustment for Deep Sub-micron Process -", ITC-CSCC2010, 2010.
- 8) Jun'ichiro Ogane, Hisayoshi Suehiro, Takuya Higuchi, Naohiko Shimizu, "Building A Design Flow for Deep Sub-micron Process(0.18 $\mu$ m) with A Scalable Cell Library - A Solution Methodology of Clock Skew with Mesh-style Clock -", ITC-CSCC2010, 2010.
- 9) Naohiko Shimizu, Masami Ikura, Warangrat Wiriy, Sorawat Chivapreecha, "A New Logic Circuit Design Methodology with UML ", ITC-CSCC2009, 2009.