東海大学大学 2010 年度 卒業論文

Alliance ツールセットを用いた schelable なチップ設計 及び検証

指導 清水 尚彦 教授

東海大学 情報理工学部
ソフトウェア開発工学科

7ADF1221　細川　達也

平成 23 年 1 月 31 日

目 次

図 目 次

表 目 次

1　概要

チップ設計をする際の手順として、レイアウト検証は欠かせない。この工程は商用ツールを用いて行われるが、ツールのライセンス料が高額であったり決められたスケールでのレイアウトしかチェックできなかったりという問題点が存在する。そこでこの工程に必要なツールを AllianceVHDL ツールセット（Alliance）内の DRUC というツールを用いることでこれらの問題点を解消することを試みた。本論文では現在までの研究成果を記すものである。また、この１年の活動経過も同時に記す。

2　はじめに

今日の LSI 設計では配置配線チェックに Caliber などの商用ツールを用いることが主である。しかし、この類のツールは膨大な額のライセンス料を必要とするため、金銭的に一部の大企業しか採用することができない。このため、中小規模の企業では例え有用なアイデアがあったとしても簡単に採用することがいかず、LSI 設計は非常に敷居の高いものとなっている。

そこで、フリーで公開されている AllianceCAD ツールセット（Alliance）を用いてこの問題を解決しようと試みた。このツールセットはハードウェア記述言語 (HDL) からチップレイアウトを作成することができる機能を備えており、チップ作成に必要な工程を全てまかなうことができる。しかし、Alliance はドキュメントに不備が多く、操作に慣れていなければ単純に HDL からチップレイアウトを作成するだけでも多大な手間を必要とする。これらの問題に関しては石黒健史著作の論文にて大半が解析され、ドキュメントとして完成させているが、ネットリスト検証とレイアウト検証の部分のみ詳細な解析がなされていない。[1] そのため、Rohm 0.18μm チップを作成する過程においてチップ作成以外の全ての工程を Alliance を用いて行い、レイアウト検証の解析を行った。実際に、試作したチップ写真を図.1 に示す。

レイアウト検証を行うためには rds ファイルにファウンドリ指定のルールを記述する必要がある。しかし、rds ファイルのフォーマットは非常に独特のものであり、マニュアルにも詳細な記述はない。そのため、Druc のソースコードを解析してルールファイルの記述をどのように読み取っているかを明らかにした。この点の詳細を述べる。

４章では全体のまとめ及び今後の課題について述べる。Alliance 全体でみた配置検証の位置づけや他のツールとの関連もこの章で述べる。

3　Druc を用いたレイアウトチェック

Druc では rds ファイルに記述されたデザインルールを読み込み、仮想レイアウトを検証する。記述するルールは作成するチップのスケールごとに変える必要がある。

rds ファイルに記述するルールは大まかに分けて 3 つの種類がある。レイヤー 1 つだけのルールと、同じレイヤーどうしのルール、異なるレイヤーどうしのルールである。それぞれの場合における記述書式、チェックできるルールを示す。

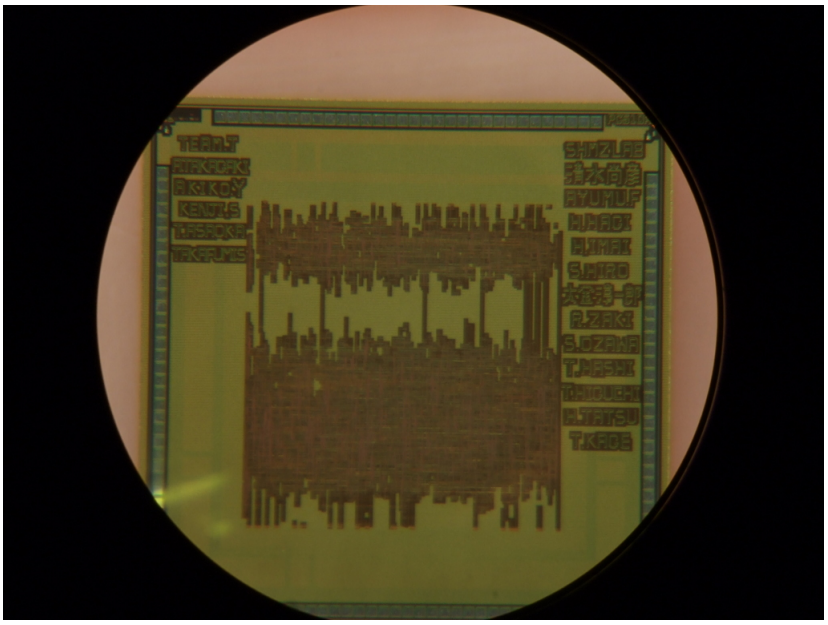

図 1: Photo of ROHM 0.18 μm chip

3.1 ルール記述書式

まず、ルールを記述するレイアウトを具体的に示す。なお、図中の数値の単位は全てμmである。図.2 はレイヤー１つだけを対象としたルールである。

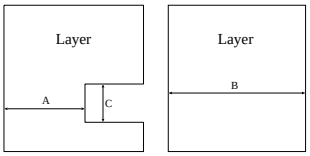

図 2: A layer rule

このように、あるレイヤー１つでは主に３つのルールを見ることができる。これを rds ファイルのフォーマットに沿って記述すると以下のようになる。

```
caracterise LAYER_NAME (
  regle 1 : largeur >= A ;
  regle 2 : longueur_inter min B ;
  regle 3 : notch >= C ;
);
```

次に、図.3 で同じレイヤーどうしの関係を対象としたルールを示す。同じレイヤーどうしでは見るルールは主に１つだけである。これを rds ファイルのフォーマットに沿って記述すると以下のようになる。

```
relation LAYER_NAME1 , LAYER_NAME2 (
  regle 4 : distance axiale >= D ;
);
```

最後に、図.4 に異なるレイヤーどうしの関係を対象としたルールを示す。

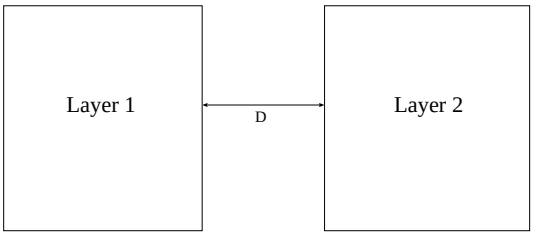

図 3: Two layers rule

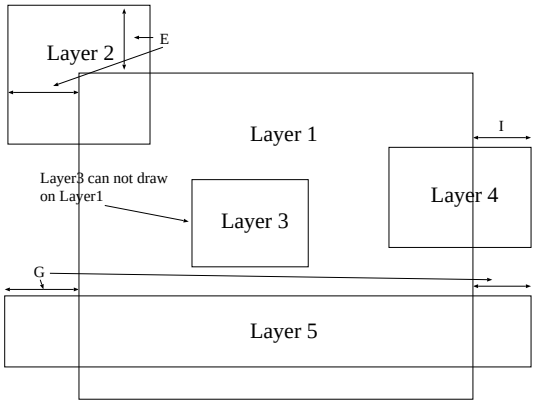

図 4: Different two layers rule

異なるレイヤーの関係は実際のレイアウトにも頻繁にあらわれるため、もっとも記述すべき項目が多い。また、単純に数値のチェックだけでなく、レイヤーが置かれる場所の正当性も問われる。これを rds ファイルのフォーマットに沿って記述すると以下のようになる。

```
relation LAYER_NAME1 , LAYER_NAME2 (
  regle 5 : distance axiale >= D ;
  regle 6 : enveloppe longueur_inter < E ;
  regle 7 : marge longueur_inter < F ;
  regle 8 : croix longueur_inter < G ;
  regle 9 : intersection longueur_inter < H ;
  regle 10 : extension longueur_inter < I ;
  regle 11 : inclusion longueur_inter < J ;
);
```

これらを総合して他の細々としたフォーマットと合わせて記述すると以下のようになる。

```
DRC_RULES
layer RDS_NWELL X ;
layer RDS_NTIE  Y ;

regles
caracterise LAYER_NAME (
  regle 1 : largeur >= A ;
  regle 2 : longueur_inter min B ;
  regle 3 : notch >= C ;
);
```

```
relation LAYER_NAME1 , LAYER_NAME2 (
  regle 4 : distance axiale >= D ;
);

relation LAYER_NAME1 , LAYER_NAME2 (
  regle 5 : distance axiale >= D ;
  regle 6 : enveloppe longueur_inter < E ;
  regle 7 : marge longueur_inter < F ;
  regle 8 : croix longueur_inter < G ;
  regle 9 : intersection longueur_inter < H ;
  regle 10 : extension longueur_inter < I ;
  regle 11 : inclusion longueur_inter < J ;
);
fin regles


DRC_COMMENT
1 (LAYER_NAME) minimum length A
2 (LAYER_NAME) minimum width B
3 (LAYER_NAME) Manhatan distance min C
4 (LAYER_NAME1,LAYER_NAME2) Manhatan distance min D
5 (LAYER_NAME1,LAYER_NAME2) Manhatan distance min D
6 (LAYER_NAME1,LAYER_NAME2) must never been in contact
7 (LAYER_NAME1,LAYER_NAME2) must never been in contact
8 (LAYER_NAME1,LAYER_NAME2) must never been in contact
9 (LAYER_NAME1,LAYER_NAME2) must never been in contact
10 (LAYER_NAME1,LAYER_NAME2) must never been in contact
11 (LAYER_NAME1,LAYER_NAME2) must never been in contact
END_DRC_COMMENT
END_DRC_RULES
```

このように、Druc が読み込む箇所は DRC_RULES と END_DRC_RULES で囲まれた箇所である。この中の式を変えることによってレイアウト検証のルールを変えることができる。この中の記述は大きく分けて 3 つにすることができる。layer で始まるレイヤー毎の最小幅宣言部、regle と fin regle とで囲まれるルール記述部、DRC_COMMENT と END_DRC_COMMENT とで囲まれるメッセージ記述部である。これらについて詳細な解説を行う。

3.2 最小幅宣言部

ここではレイヤー毎の最小幅を宣言する。書式を以下に示す。

```
layer LAYER_NAME MIN_SIZE ;
```

最初は layer で始め、LAYER_NAME にレイヤー名を、MIN_SIZE にそのレイヤーの最小幅を記述し、セミコロンで終わる。これらは全て半角スペースで区切る。レイヤー名は rds ファイルの最初の方にある所定の場所に宣言されている必要があるが、そこの書式は割愛する。ここでレイアウトに使われている全てのレイヤーを宣言し、最小幅を記述しておく必要がある。

3.3 ルール記述部

ルール記述部ではレイヤーそのもののルールを定義するために caracterise が、レイヤーどうしの関係性についてのルールを定義するために relation が用意されている。

caracterise でレイヤー単独のルールを定義する時の書式を次に示す。

```
caracterise LAYER_NAME (
  regle NUM   : CHARA_RULE OPE DEF_SIZE ;
  regle NUM+1 : CHARA_RULE OPE DEF_SIZE ;
  regle NUM+2 : CHARA_RULE OPE DEF_SIZE ;
);
```

LAYER_NAME には先に定義したレイヤー名を、NUM には通し番号を振る。ここでの番号は他のルールと被ってはいけない。CHARA_RULE には定義したいルールを記述する。ここで使用できるルールはいくつかあり、目的に応じて使い分ける必要がある。ルールの種類については後述する。OPE には最大、最小を指定する不等号及び min、max を記述し、DEF_SIZE にサイズを定義する。サイズを記述する際は必ず小数で記述する必要がある。

次に、relation で複数のレイヤーの関係をチェックする場合の書式を次に示す。

```
relation LAYER_NAME1 , LAYER_NAME2 (
  regle NUM : REL_RULE1 REL_RULE2 OPE DEF_SIZE ;
);
```

基本的な書式は caracterise で解説したとおりであるが、relation の場合は 2 つのレイヤーの関係を見ているので多少異なる。LAYER_NAME1 と LAYER_NAME2 には関係性をチェックしたいレイヤーを記述する。この時、同じレイヤーを指定することで同じ種類のレイヤーの関係性をチェックすることができる。また、relation では characterise で用いたルールとはまた別にルールを 2 つ指定する事ができる。このルールの組み合わせによって様々なルールをチェックする事が可能となる。また、OPE で指定できる不等号と min、max では異なる意味をもつ。この点を表.1〜4 にまとめた。

表中で unknown と示されているところは未だ検証中のものである。

このように、ルール書式はフランス語で設定されている。また、OPE で使われている min、max は多角形のレイヤーでのみ用いることができる記述であり、不等号は長方形でのみ用いることができる。

description	meaning
surface	A surface area of layer
longueur	A longer side of layer
largeur	A shorter side of layer
notch	minimum distance in notch

表 1: List of operator to CHARA_RULE

description	meaning
distance	distance
intersection	over two sides on layer from under layer
extension	over one side on layer from under layer
inclusion	on layer can exist on under layer
enveloppe	unknown
marge	unknown
croix	cross layer

表 2: List of operator to REL_RULE1

description	meaning	usage
min	minimum	polygon
max	maximum	polygon
infeq	less-than	unknown
supeq	greater-than	unknown
$<$	less	rectangle
$>$	greater	rectangle
$=$	equal	rectangle
diff	not-equal	unknown

表 3: List of operator to OPE

description	meaning
surface_inter	unknown
longueur_inter	minimum distance internal layer
largeur_inter	unknown
penetre_inter	unknown
paralel_inter	unknown
perpend_inter	unknown
longueur_min	unknown
longueur_max	unknown
largeur_min	unknown
largeur_max	unknown
frontale	unknown
laterale_min	unknown
laterale_max	unknown
sup	unknown
inf	unknown
axiale	unknown
geodesique	unknown

表 4: List of operator to REL_RULE2

rule no.	description	meaning
regle1	largeur	Define minimum distance in rectangle
regle2	longueur_inter	Define minimum distance in polygon
regle3	notch	Define minimum distance in notch
regle4	distance axiale	Define minimum distance between same layers
regle6	enveloppe longueur_inter	Unknown
regle7	marge longueur_inter	Unknown
regle8	croix longueur_inter	When cross different layers, verify edge distance between layers
regle9	intersection longueur_inter	When over two sides on layer from under layer, verify edge distance between layers
regle10	extension longueur_inter	When over one side on layer from under layer, verify edge distance between layers
regle11	inclusion longueur_inter	Verify to on layer can exist on under layer

表 5: Example rule description meaning

3.4 メッセージ記述部

DRC_COMMENT と END_DRC_COMMENT で挟まれた記述はルール違反があった場合出力されるエラーメッセージを表示するときに使われる。そのため、それぞれの番号はルールの regle number と対応していなければならない。メッセージ表示のための書式を以下に示す。

```
DRC_COMMENT
NUM (LAYER_NAME) ERROR_MESSAGE
NUM (LAYER_NAME,LAYER_NAME) ERROR_MESSAGE
NUM (LAYER_NAME,LAYER_NAME) ERROR_MESSAGE
END_DRC_COMMENT
```

NUM には通し番号を振る。この番号は対応したルールと同じ番号でなければならない。LAYER_NAME には定義されたレイヤー名を、ERROR_MESSAGE には自由にメッセージを書くことができる。定義しているエラーがすぐわかるようなメッセージを書いておくことが好ましい。

ここではエラーの内容がわかりやすいようにこのようなフォーマットになっているが、ルールに対応した番号だけ正しく振ればあとはフリーフォーマットである。そのため独自のメッセージを考案して設定する事も可能である。

4 サンプルコードで見るルール

これまでルール記述のフォーマットについて述べたが、ルールを組み合わせたりレイヤーの形によって演算子が異なったりとやや分かり辛い箇所が多々ある。そのため、先にあげたサンプルコードを実際に通したとき、どの部分がどこに対応しているのかを前の図を使って図.5〜7 にまとめた。また、図に示したルールの詳細な解説を表 5 にまとめた。

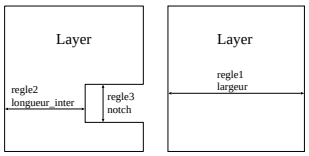

図 5: A layer rule

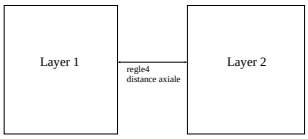

図 6: Same layers rule

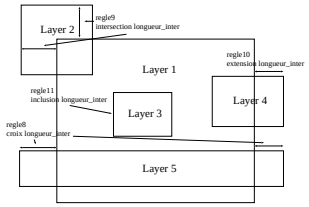

図 7: Different layers rule

表は図中のルールを解説したものである。総じて longueur_inter は最小幅を見るルールであり、その前の単語によって場所を指定している。また、サイズ指定が 0 以下になっているものは最小幅ではなく重なっているかどうかをチェックする式である。

5　現時点でのルール実装状況

このように、どのようなルール記述法があるかは解析済みであるが、それが何を意味するのかは不明な点が多い。しかし、解析できたルールから実際のレイアウトに適用できるように実装している。今回は ROHM 社の 0.18 μm デザインルール [**?**] に適用されるルールに則ってルールファイル作成を行っている。実際に使用できるレイヤーとレイヤー毎のルール実装状況を表.6 に示す。

Layer name	All rule	Implemented rules
NWELL	13	6
PDIFF, NDIFF	22	6
POLY	26	0
NIMP	17	0
PIMP	17	0
CONTACT	9	0
PROTECT	10	0
METAL1	8	2
METAL(2 to 4)	8	0
METAL5	7	0
VIA(1 to 3)	6	0
VIA4	5	0
ALL LAYERS	148	14

表 6: Number of all rules and implemented rules

6　まとめ

このように、Druc でレイアウト検証を行うためにソースコード解析を行う必要があったため、Druc 中で定義されているルール内容の解析は一時中断している。しかし、ソースコードからのルール内容解析に著しい進捗があり、記述されたルールを読み取り、レイアウトをチェックするフローが全て解析できた。これによって、現在ではそれに基づいて実際にレイヤーを描き、テストする段階を残すのみとなっている。今後はそれに鋭意努力していく予定である。

7　研究室での過去の活動

2010 年 2 月 ~2010 年 6 月
　C 言語の仕様書を和訳する。また、SFL を用いて HDL の基礎を学ぶ。

2010 年 6 月 ~2010 年 9 月
　ET ロボットコンテストに参加。プロジェクトマネジメントについて実践的に学ぶ。

2010 年 9 月 ~ 現在
　Alliance 検証部分の解析に着手する。asimut、lvx、cougar、druc などを調査する。

8　謝辞

　これまでの研究に多大なる協力をしていただいた清水尚彦先生、友人の皆様に厚く御礼申し上げます。

参考文献

[1] 石黒健史著　Alliance の Scalable CMOS Library によるチップ試作の研究