

# フリーのEDAを用いた設計による バーサライタ制御チップ

東海大学 情報通信学部

組込みソフトウェア工学科 清水尚彦研究室

古川拓実, 河村まりや, 富山修平, 清水尚彦

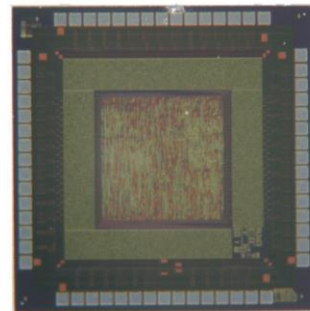
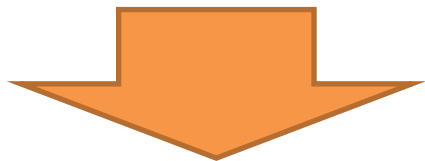
# 研究背景・目的

デジタルLSIチップ設計では  
EDAツールが  
必須になっている

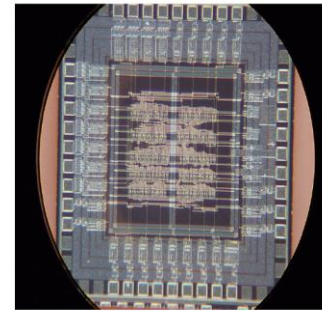


フリーのEDAツールAlliance\*を用いた  
スケーラブルなセルライブラリによる  
高位合成デジタルLSI設計フロー

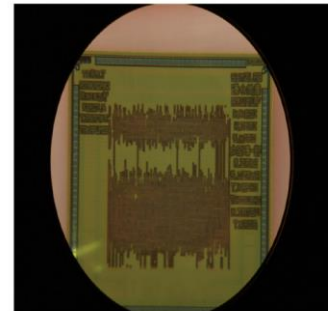
従来確立してきたプロセス



Rohm 0.35um



Onsemi 1.2 um



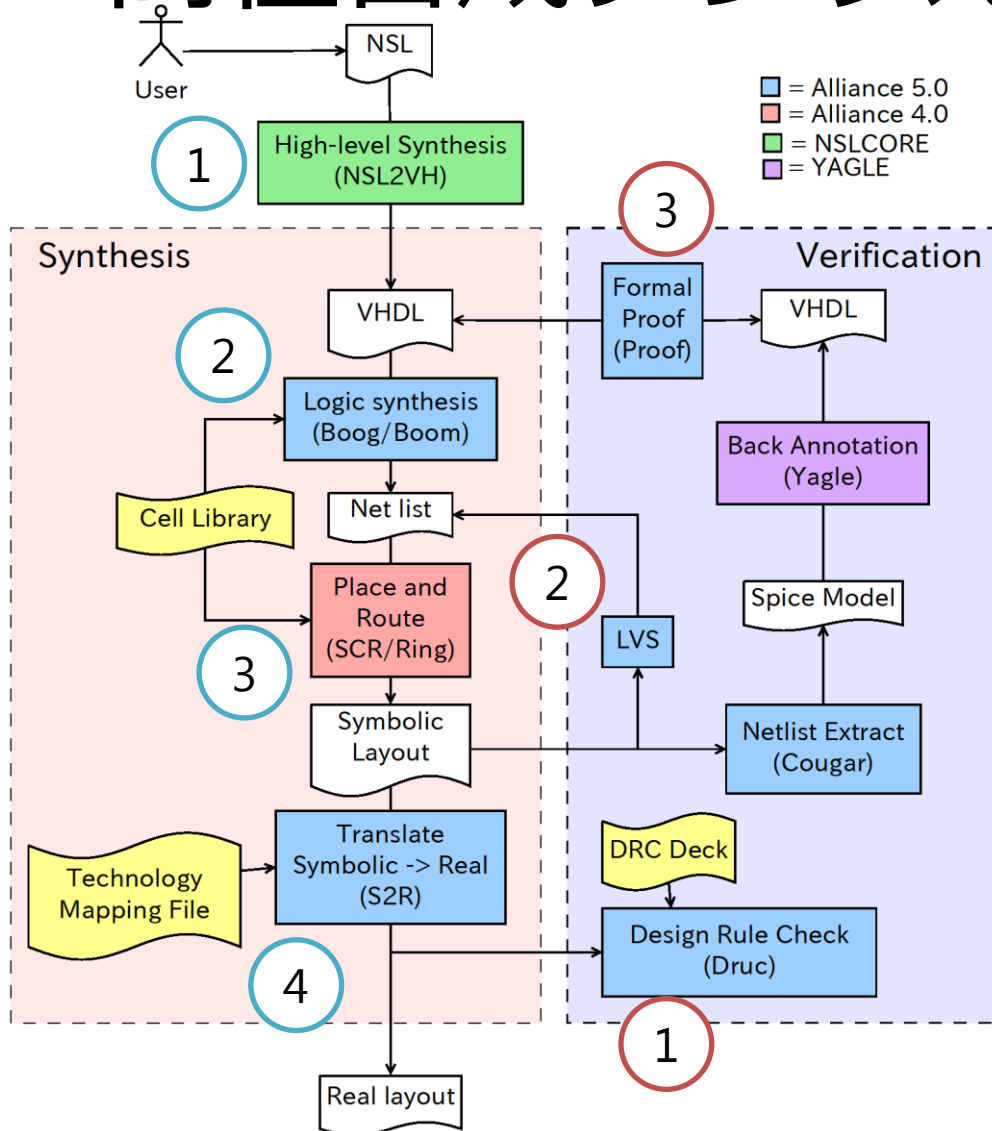
Rohm 0.18um

図1 過去に試作を行ったチップ[1]

新たに北九州学術推進機構  
2umプロセスルールに対応するLSI設計フローの確立

\*UPMC(フランス)が開発

# 高位合成デジタルLSI設計フロー



ユーザは高位合成言語NSLを用いて論理回路設計を行う

## レイアウト生成ツール

1. 高位合成
2. 論理合成
3. 配置配線
4. 仮想レイアウト変換

## 検証ツール

1. DRC
2. LVS
3. 形式検証

図2 デジタルLSI設計フロー[2]

## NSL記述

```

declare adder{
  input a,b; output x;}
module adder{
  x=a+b;
}

```

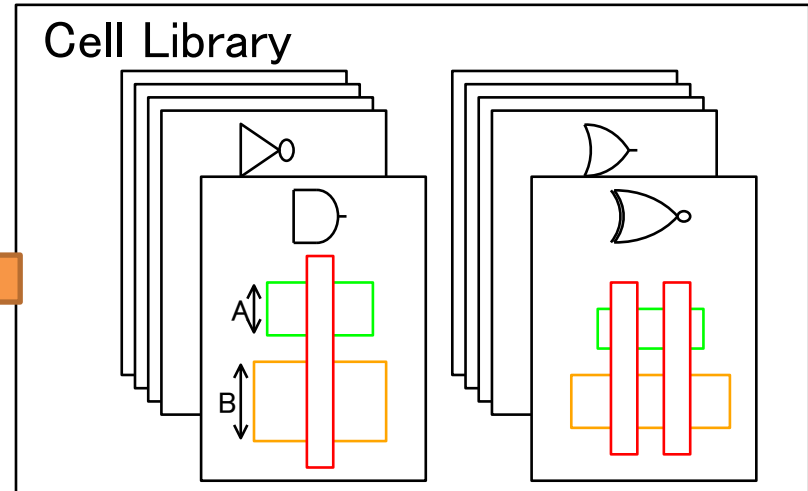
## 生成したVHDL

```

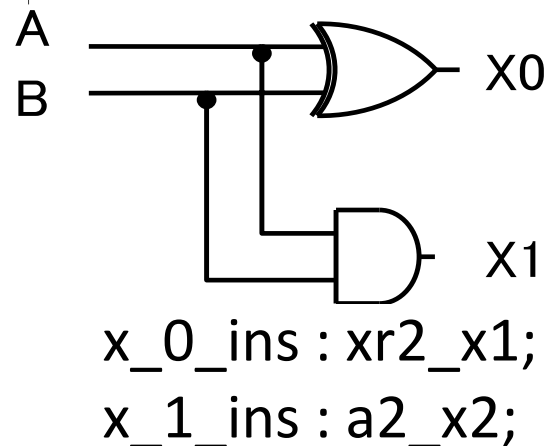
entity adder is port(
  x: out std_logic_vector
    (1 downto 0);
  b: in std_logic;
  a: in std_logic);
end adder;
architecture RTL of adder is
begin
  x <= internal_x;
  internal_x <= a+b ;
end RTL;

```

## 設計フロー1~2



## ネットリスト



ネットリスト

# 設計フロー3~4

配置配線  
(P&R)

Allianceでは仮想レイアウトで設計を行い  
プロセスごとの変換ルールを用いて  
実レイアウトへ変換する

仮想  
レイアウト

実レイアウト  
(GDS)

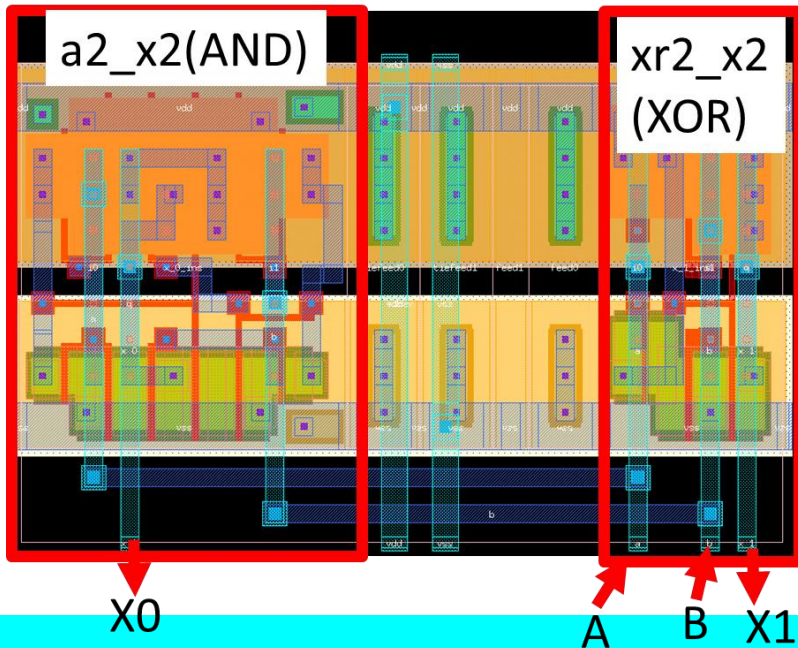
Technology Mapping File  
(変換ルール)

$\lambda=2.4\mu\text{m}$

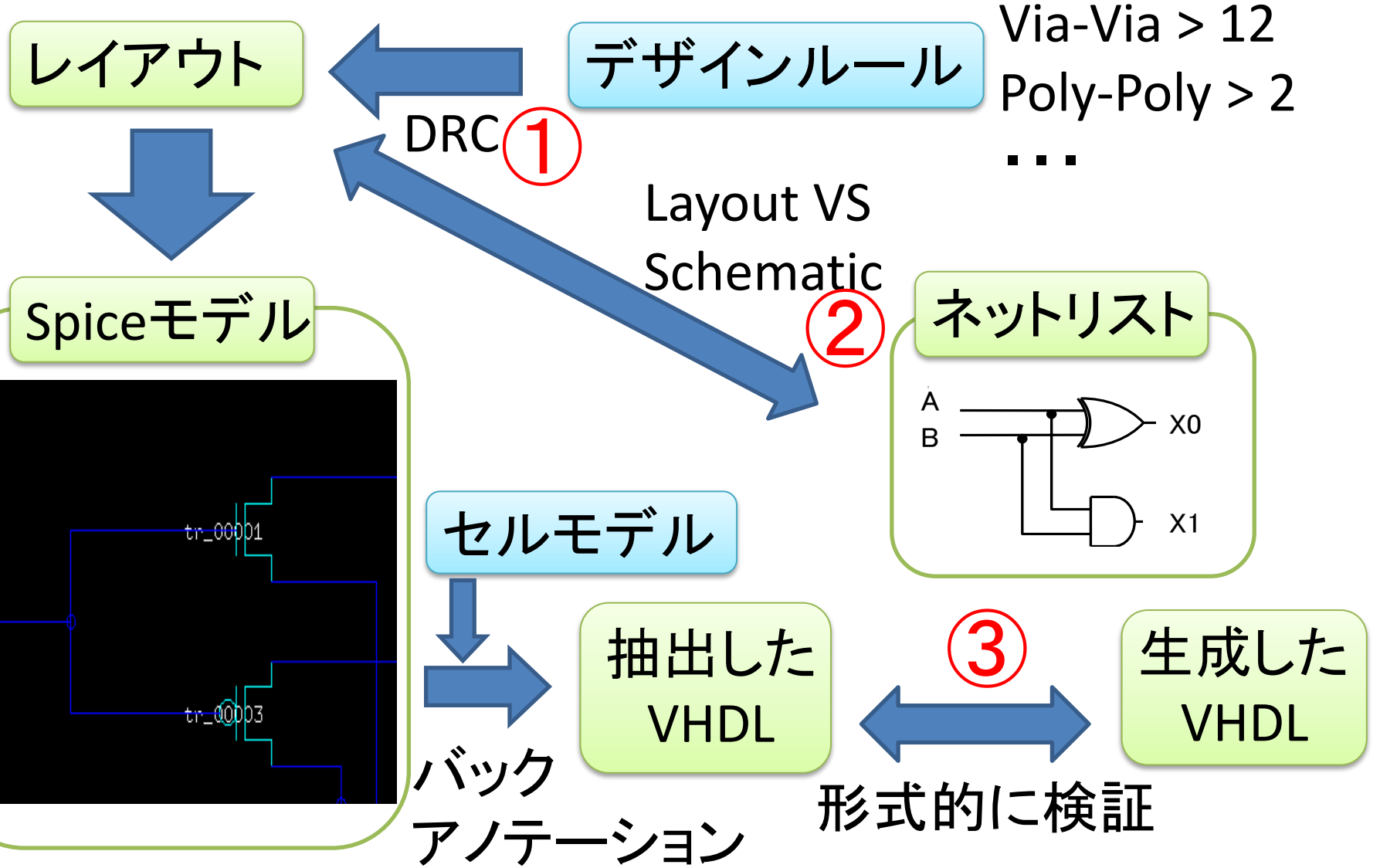
POLY VW 1.0 -0.4

METAL1 VW 3.0 1.2

...



# 検証フロー



# 実レイアウトへの変換ルール

## λ-based rule

- スケーラブルな設計で広く用いられている
  - グリッドサイズパラメータ $\lambda$ でサイズを変動させるため線形的な変換になってしまう
- >プロセスごとの細かいルールには対応できない

## Allianceが用いるシンボリックルール

- グリッドサイズパラメータは $\lambda$ を用いる
  - 実レイアウト変換時に長さ・幅・オフセットをum単位で適応できる
  - 仮想セグメント一つに対して複数の実レイヤを作れる
- >λ-based ruleより細かい対応が可能

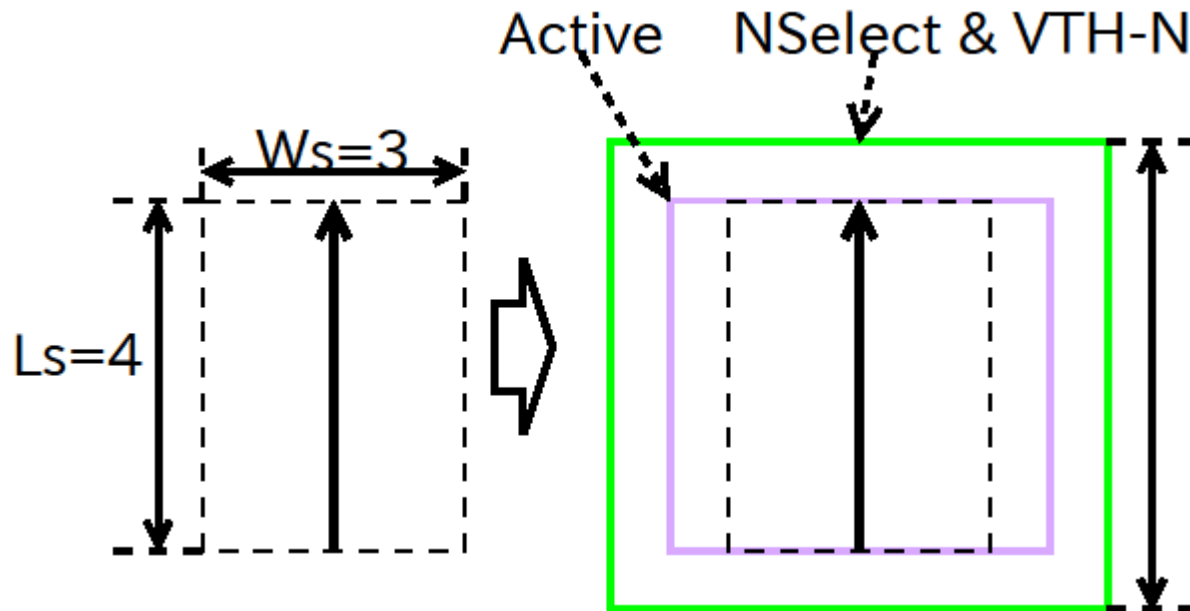
# 実レイアウトへの変換例

幅  $W_r = W_s * \lambda + \text{DWR}$       長さ  $L_r = L_s * \lambda + 2\text{DLR}$

Segment	Layer	DLR (um)	DWR (um)
NDIFF	NSELECT	4	2
	VTH-N	4	2
	ACTIVE	2	0

$\lambda = 3\mu\text{m}$

NDIFFセグメントは  
3つの実レイヤへ変換される



$Nselect.L_r =$   
 $3 * 4 + 2 * (4) = 20\mu\text{m}$

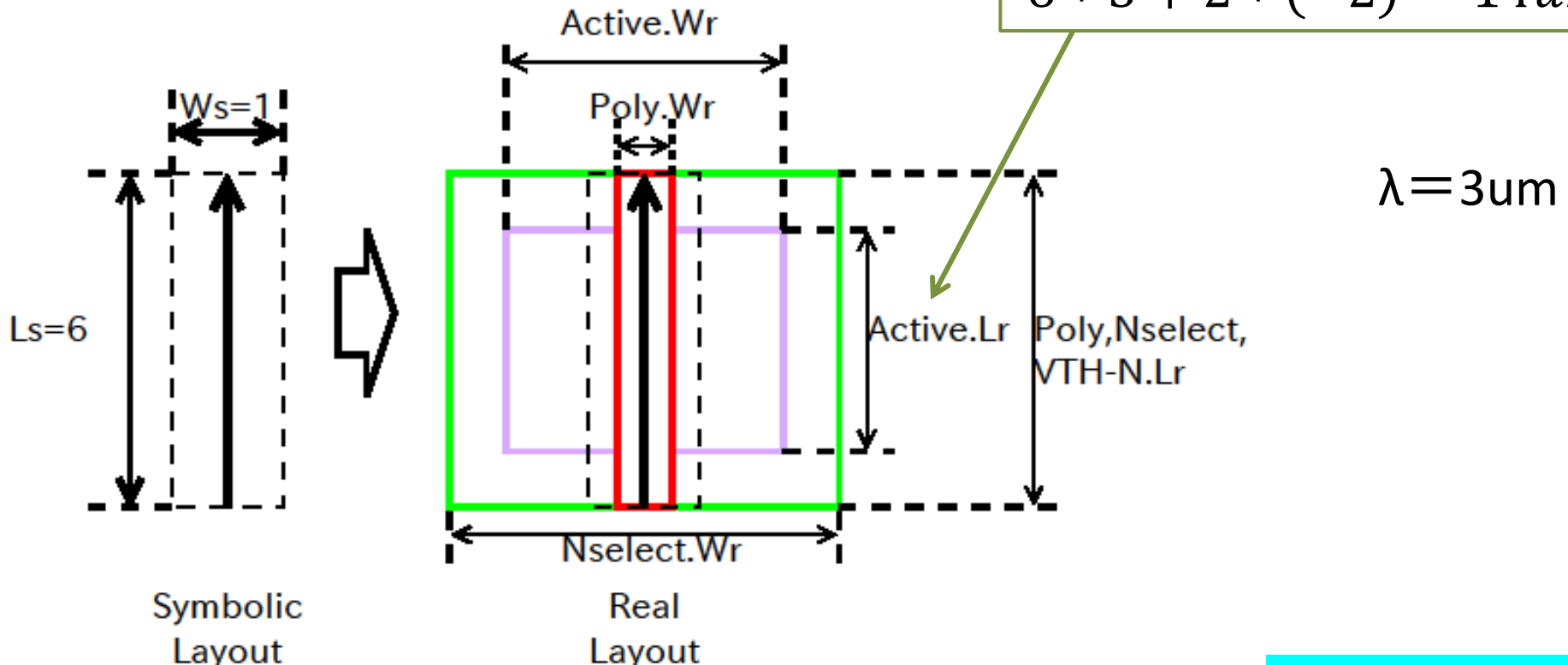


# トランジスタ変換例

Segment	Layer	DLR (um)	DWR (um)	Lr (um)	Wr (um)
NTRANS	POLY	0	-1.0	18	2
	ACTIVE	-2	14	14	17
	NSELECT	0	18	18	21
	VTH-N	0	18	18	21

$$\text{Poly.Wr} = 1 * 3 + (-1) = 2\mu\text{m}$$

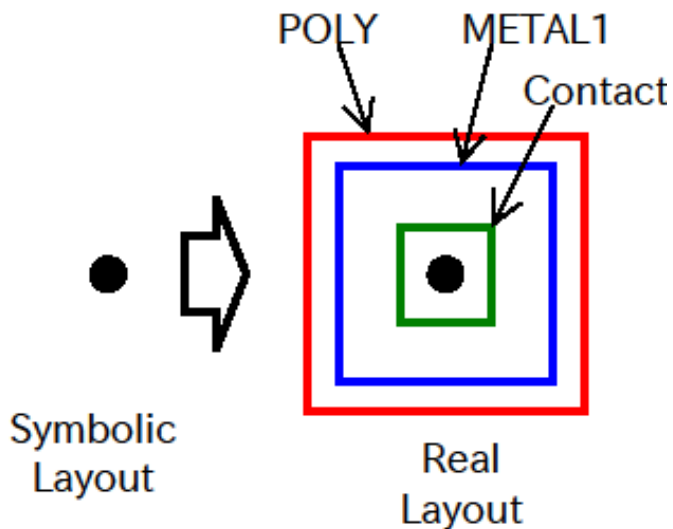
$$\text{Active.Lr} = 6 * 3 + 2 * (-2) = 14\mu\text{m}$$



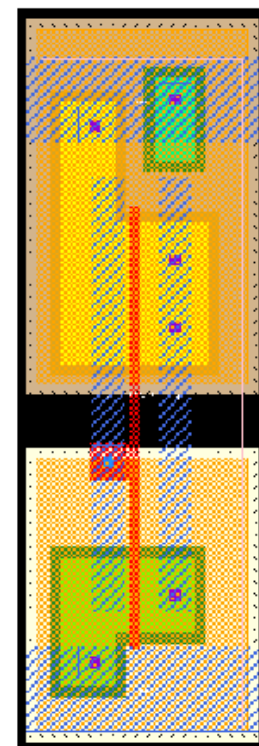
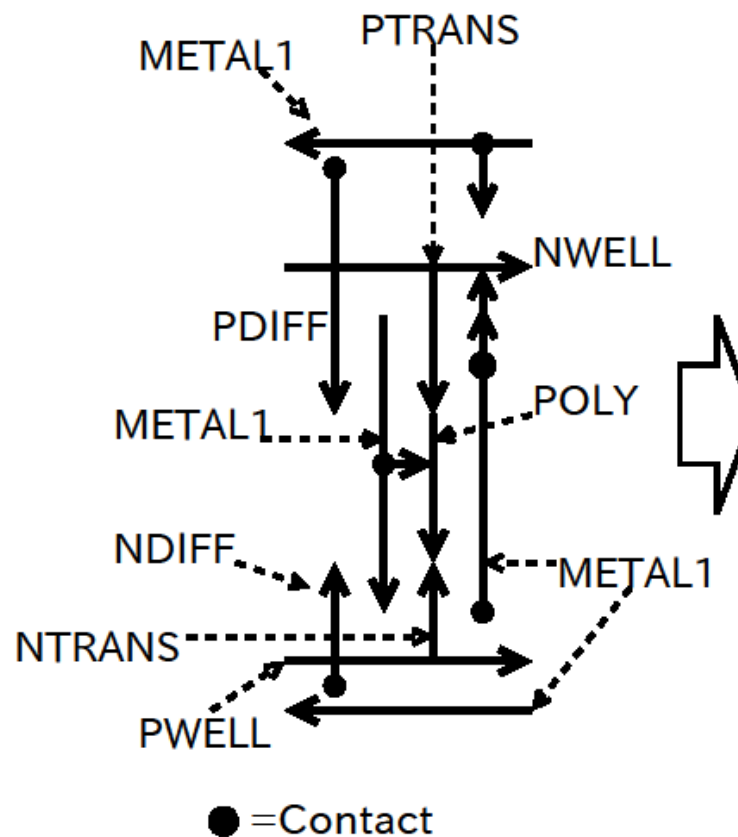
# VIA/コンタクト変換例

## セル変換例

Segment	Layer	Width (um)
POLY_CONTACT	METAL1	6.0
	CONTACT	2.0
	POLY	7.0



VIAやコンタクトは  
中心点で表す



# FAIS2umプロセスへの適用

## プロセスの制約

- ・Via-Via間隔が12um
- ・Metal幅と間隔が大きい

## 製造の制約

- ・100um四方のボンディングパッド
- ・チップ検証用回路

## Technology Mapping File

$\lambda=2.5\text{um}$ に設定することで配線間隔を調整

## ツールの制約

- ・配線ツールの配線間隔が $5\lambda$

## プロセスごとに作成が必要

- ・ Technology Mapping File (変換ルール)
- ・ DRC Deck
- ・ (Cell Library)

# FAIS2umプロセスへの適用

## 問題発生

Poly上にVIAを打てないため  
すでにPolyとViaが重なっている  
セルがエラーとなる



各種ロジックセルの修正

チップサイズに対して  
既存のI/Oパッドセルでは  
大きすぎて対応できない



I/Oパッドセルの新規開発

配置配線時にTieの無い  
Wellが発生する



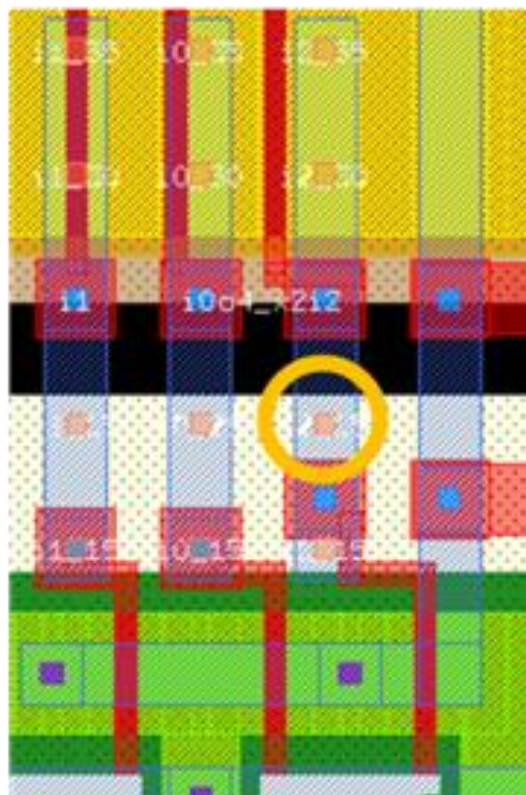
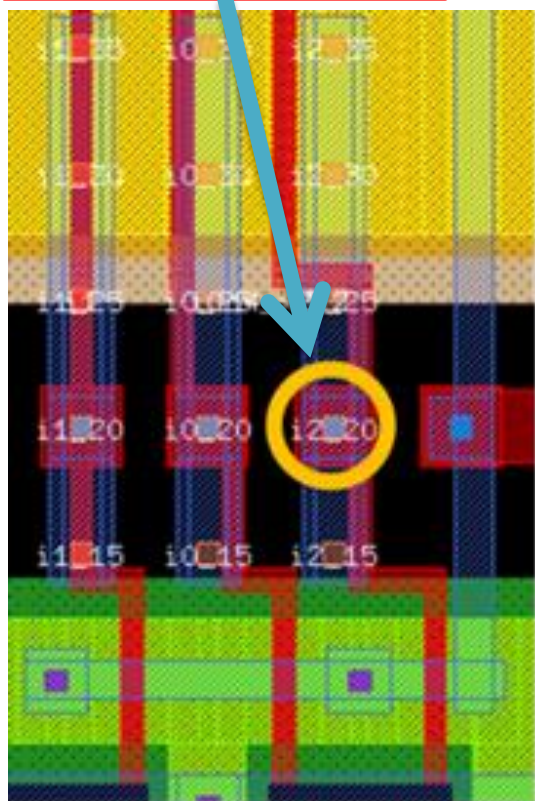
配置配線ツールに  
Tieを生成する機能を追加

# セルライブラリ開発

## ロジック・スタンダードセル

VIAとPOLYが重なっている

POLYの位置を修正




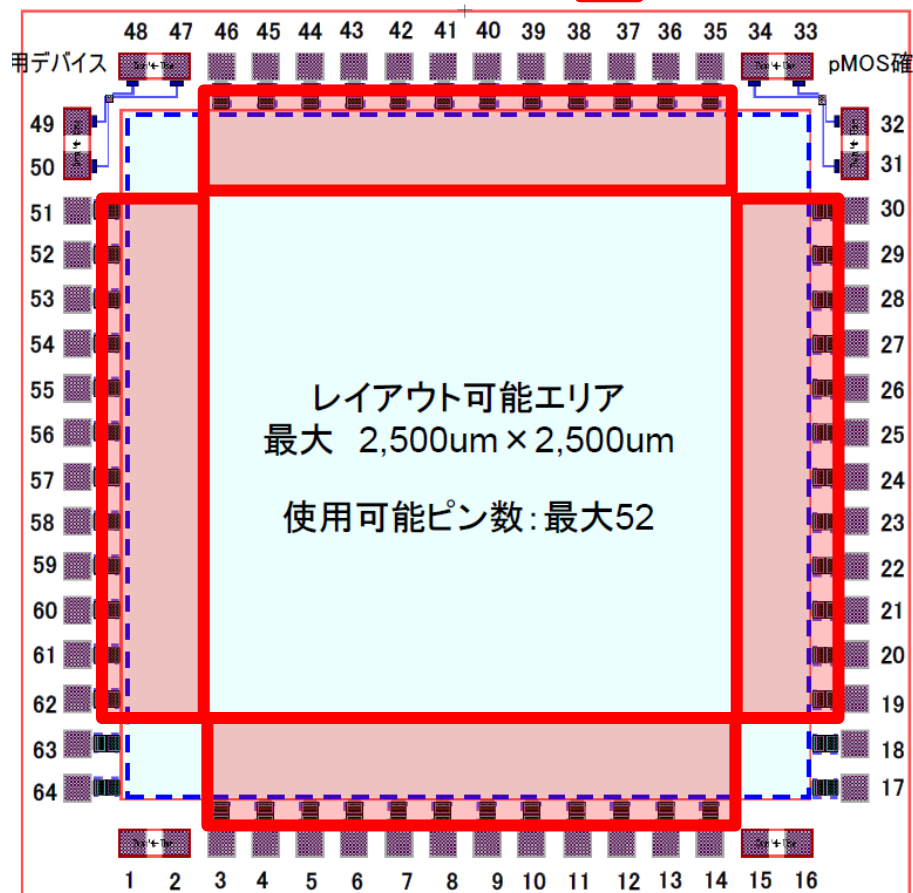
スタンダードセル

- ・2入力AND
  - ・3入力OR
- など計86種類用意

# セルライブラリ開発

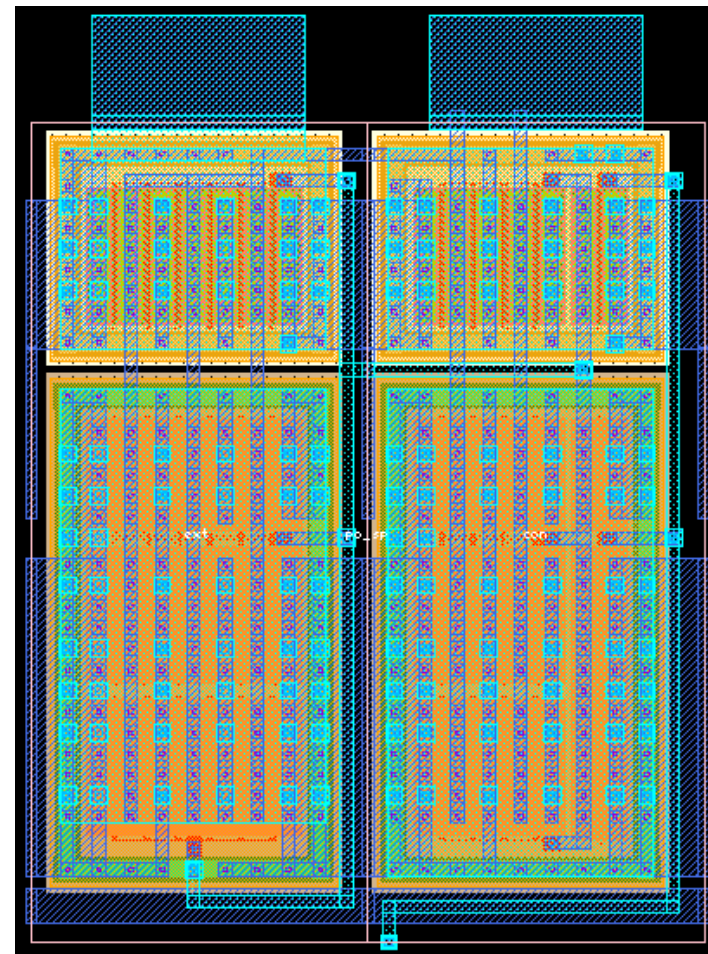
I/Oパッドセル

 I/Oパッド使用領域



FAIS製造プロセス仕様書より引用

OUTPUTセル



160um

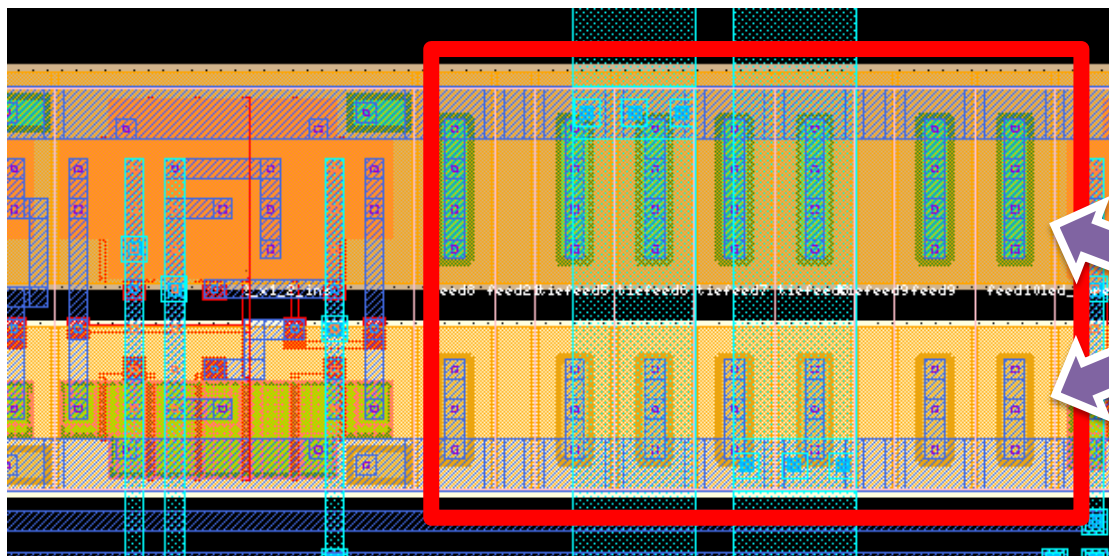
# ツール修正

ツールがオープンソースであることを活かし  
配置配線プログラムSCRの機能追加・改良

配置配線時にTieの無い  
Wellが発生する



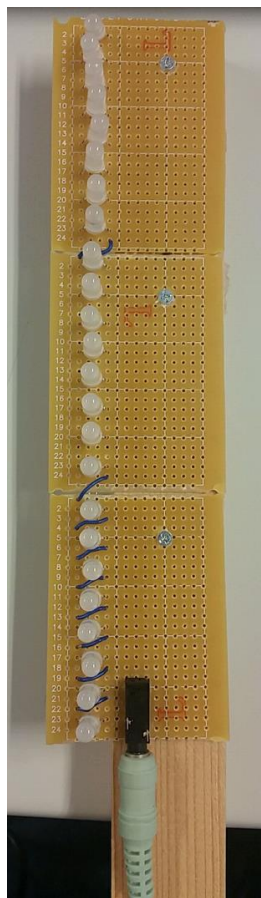
電源配線領域に  
Tieセルの挿入機能を追加



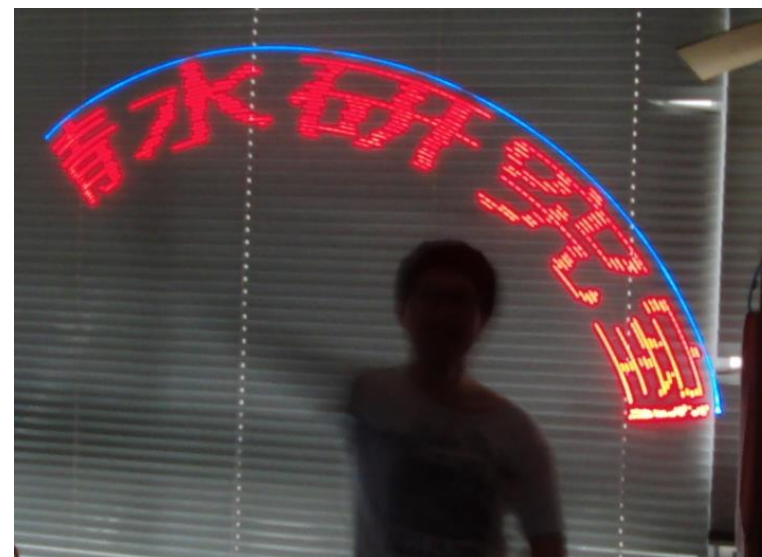
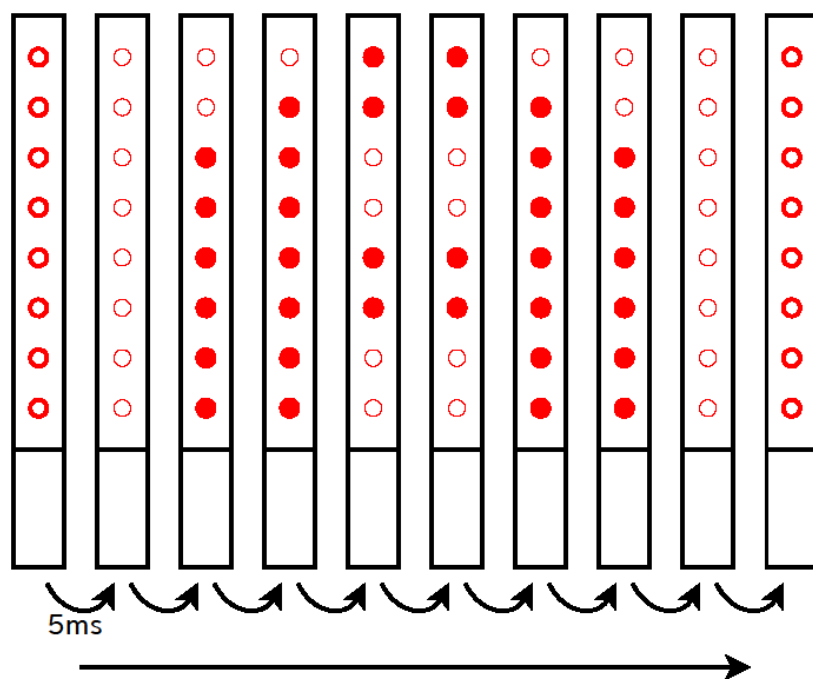
Well-Well間にある  
電源配線領域に  
Tieセルを挿入

# LEDバーサライタ

LEDが縦についた棒を左右に振ることで  
空中に図形を描くことができる



バーサライタ本体



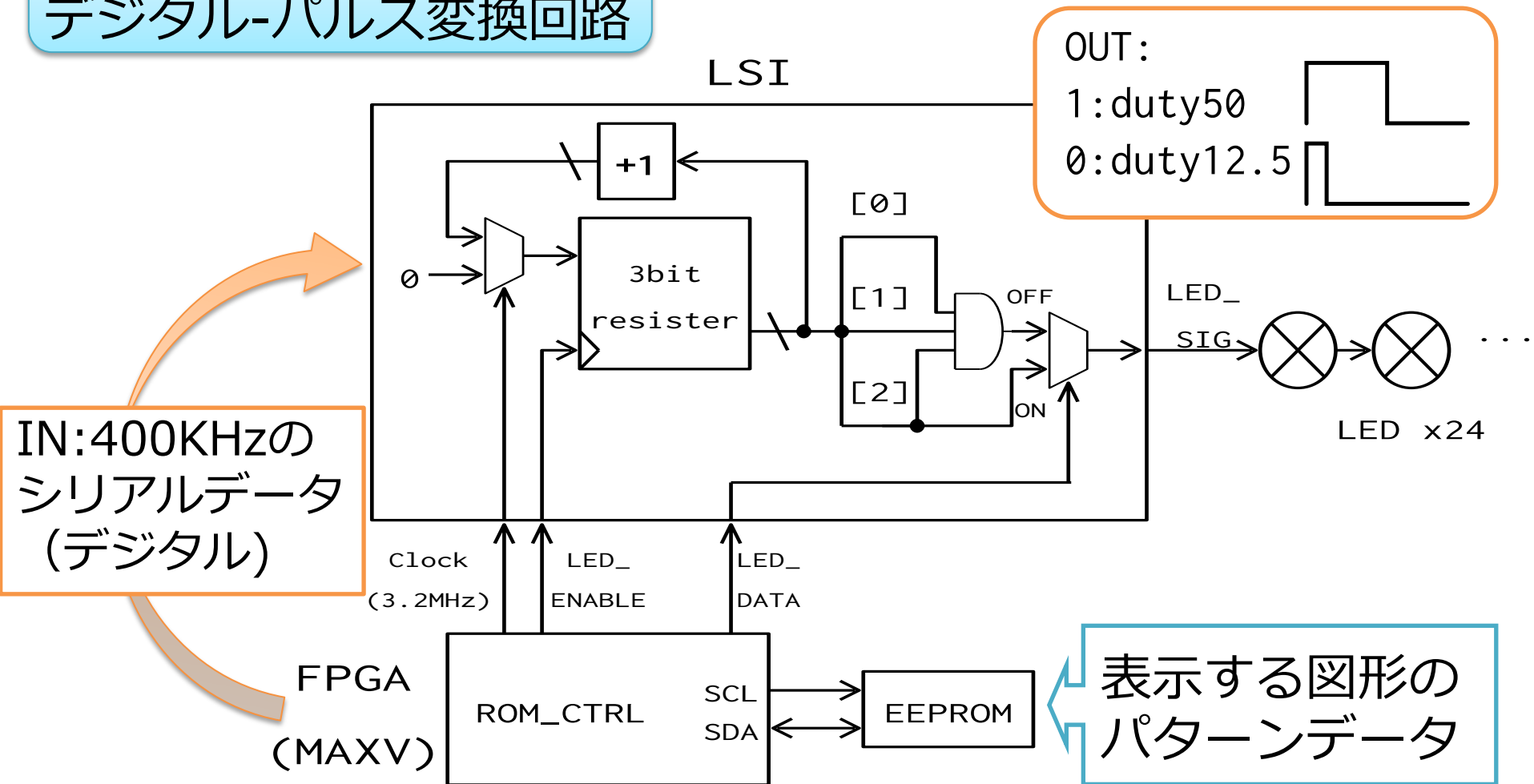
バーサライタ動作写真

高速にLEDパターンが変化することで  
空中に残像が残るので図形に見える

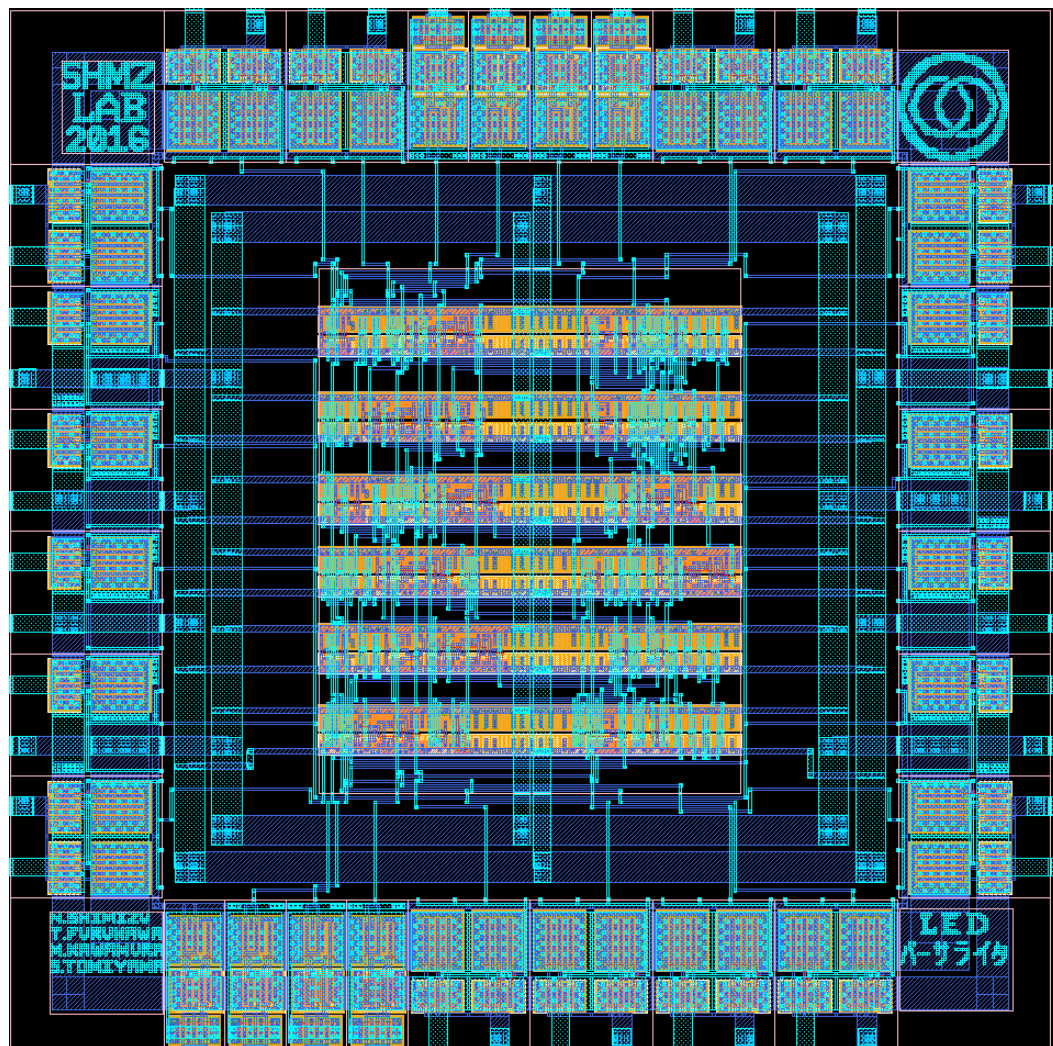


# 制御回路

マイコン内蔵LED用  
デジタルパルス変換回路



# レイアウト生成結果



## トランジスタ数

ターゲット回路286個  
+  
セル検証用回路172個

計458個

VDD:4本

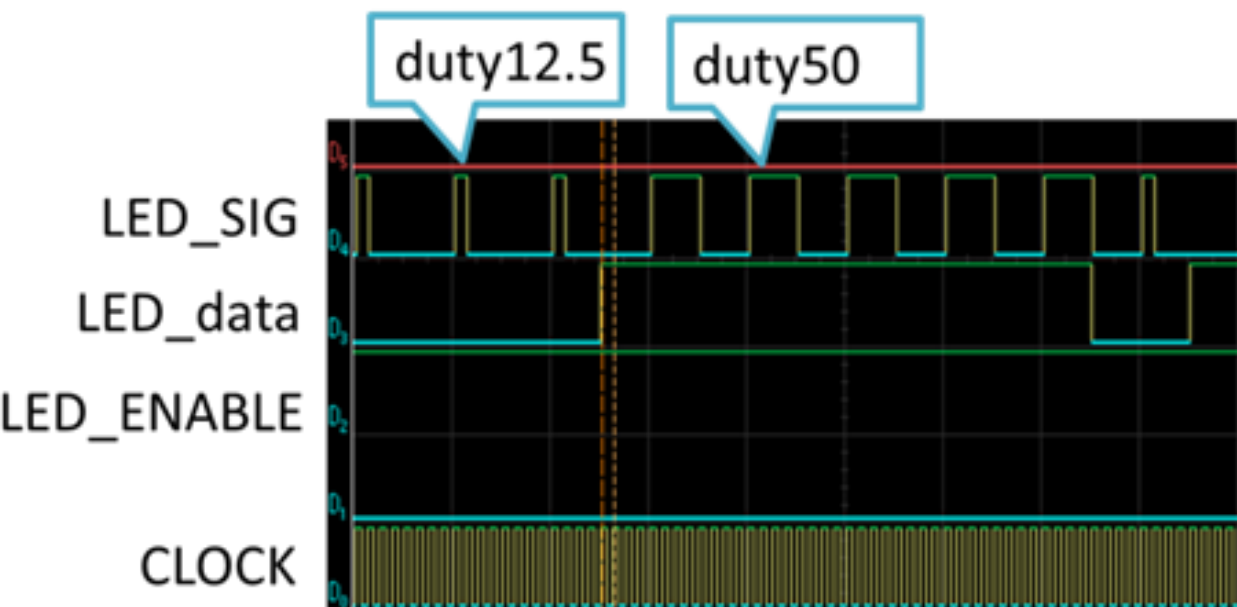
VSS:4本

INPUT:8本

OUTPUT:12本

計48本

# ターゲット回路評価



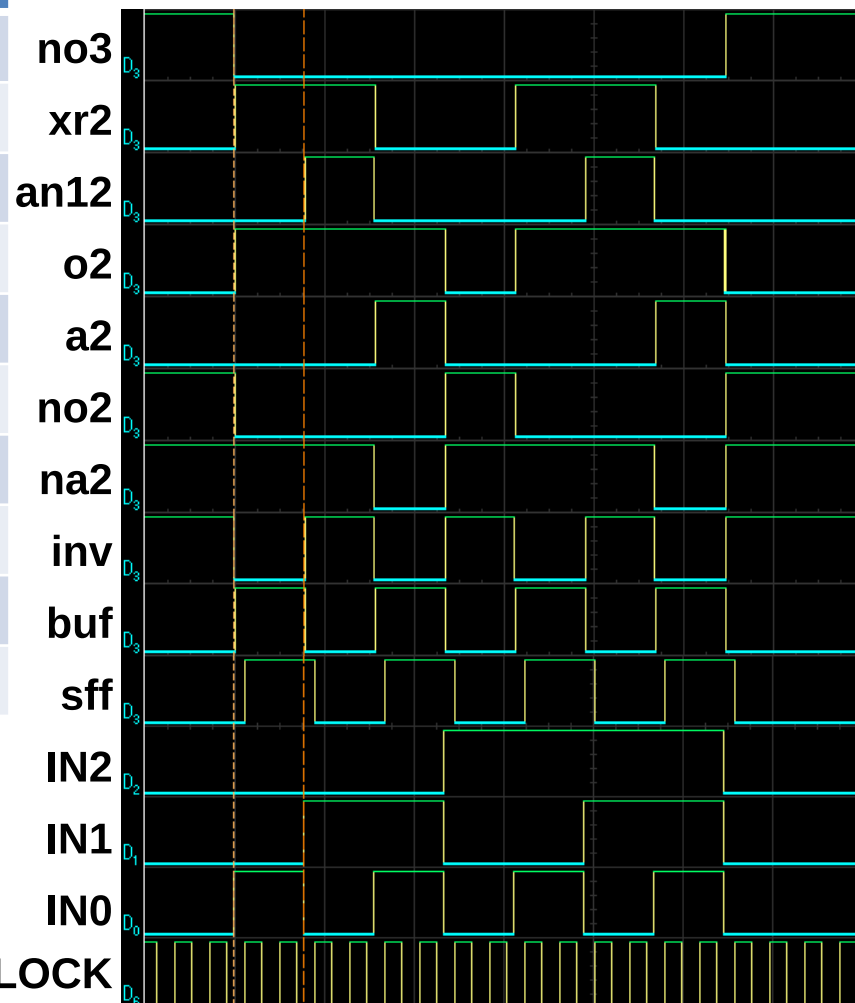
チップごとの回路動作

チップ番号	ターゲット回路	セル検証回路
No.1	×	○
No.2	○	○
No.3	×	×
No.4	○	○

正常な波形を確認できたが  
歩留まりが悪い・ノイズが多い等の問題があった

# ロジックセル評価

Cell name	Detail (Logic)	Input
no3	not (IN0 or IN1 or IN2)	3
xr2	IN0 xor IN2	2
an12	(not IN0) and I1	2
o2	IN0 or IN1	2
a2	IN0 and IN1	2
no2	not ( IN0 or IN1)	2
na2	not ( IN0 and IN1)	2
inv	not IN0	1
buf	IN0	1
sff	register; register = IN0	1



10個全てのセルで  
正常動作を確認

# まとめ

- 高位合成とフリーのEDAツールを用いたデジタルLSI開発フローを構築した
- マイコンLED制御回路をターゲットにレイアウトを作成しチップ試作  
->セル・ターゲット回路ともに動作波形を確認できた

FAIS2umプロセスルールを満足するデジタルLSI設計フローを確立できた

# 今後の展望

- マニュアル等の整備を行い一般ユーザに提供可能にする
- $\lambda=2.5\mu\text{m}$ の試作などさらなる検証を行いツールの信頼性を向上させる
- フェニテックやミニマルファブなど低価格のプロセスに対応させて本手法を提供する