

9ビットRDACの自動合成

東京工業大学大学院 理工学研究科

電子物理工学専攻 松澤・岡田研究室

○盛 健次、菅原 光俊、宮原 正也、松澤 昭

2013/8/28

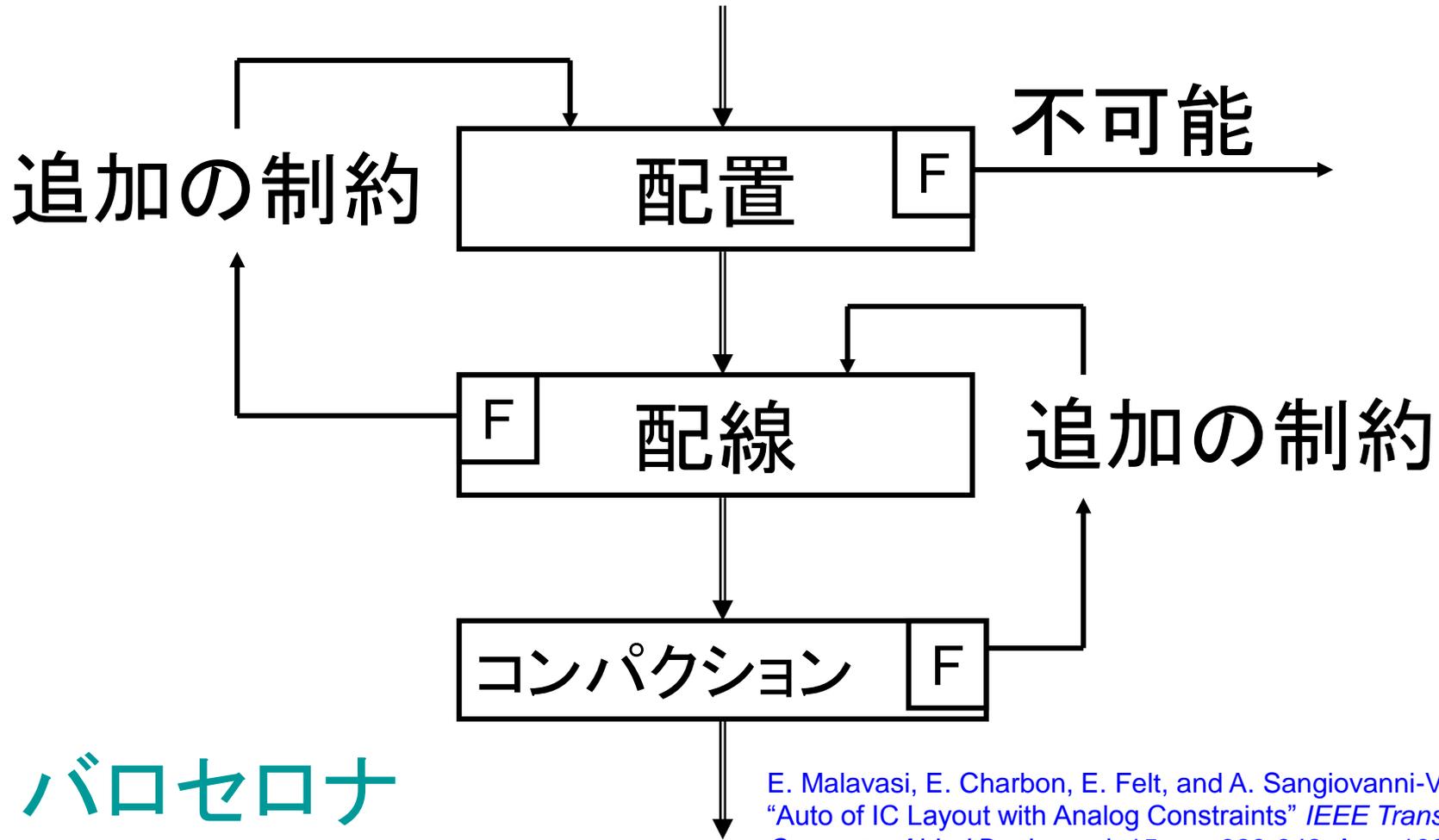
1. 開発目的
2. 従来の自動合成
3. 我々の自動合成
 3. 1 アナログ回路自動生成プログラムのフロー
 3. 2 開発方法
 3. 3 開発言語の紹介
 3. 4 下地の回路図とレイアウト図
 3. 5 回路図とレイアウト図
 3. 6 我々の開発の格子
 3. 7 プログラムによる階層化設計
4. シミュレーション回路とシミュレーション結果
5. レイアウト結果
6. 今後
質疑応答

1. 開発目的

- (1) 開発期間を短縮する。(プログラムで1分以内)
- (2) チップ面積を最小にする。
- (3) 仕様変更(6ビット～16ビットまで対応)、プロセス変更(90nm、65nm、45nm、28nm)を考慮して、自動設計を行う。
- (4) 少ない回路部品で、同じ設計手法を用いることにより、様々なアナログ回路(RDAC、CDAC等)に適用できるようにする。
- (5) レイアウトブロックの重ね合わせによる配線の為、配線遅延が少なくなり、回路動作は高速化が狙える。

2. 従来の自動合成

ハイレベルの制約



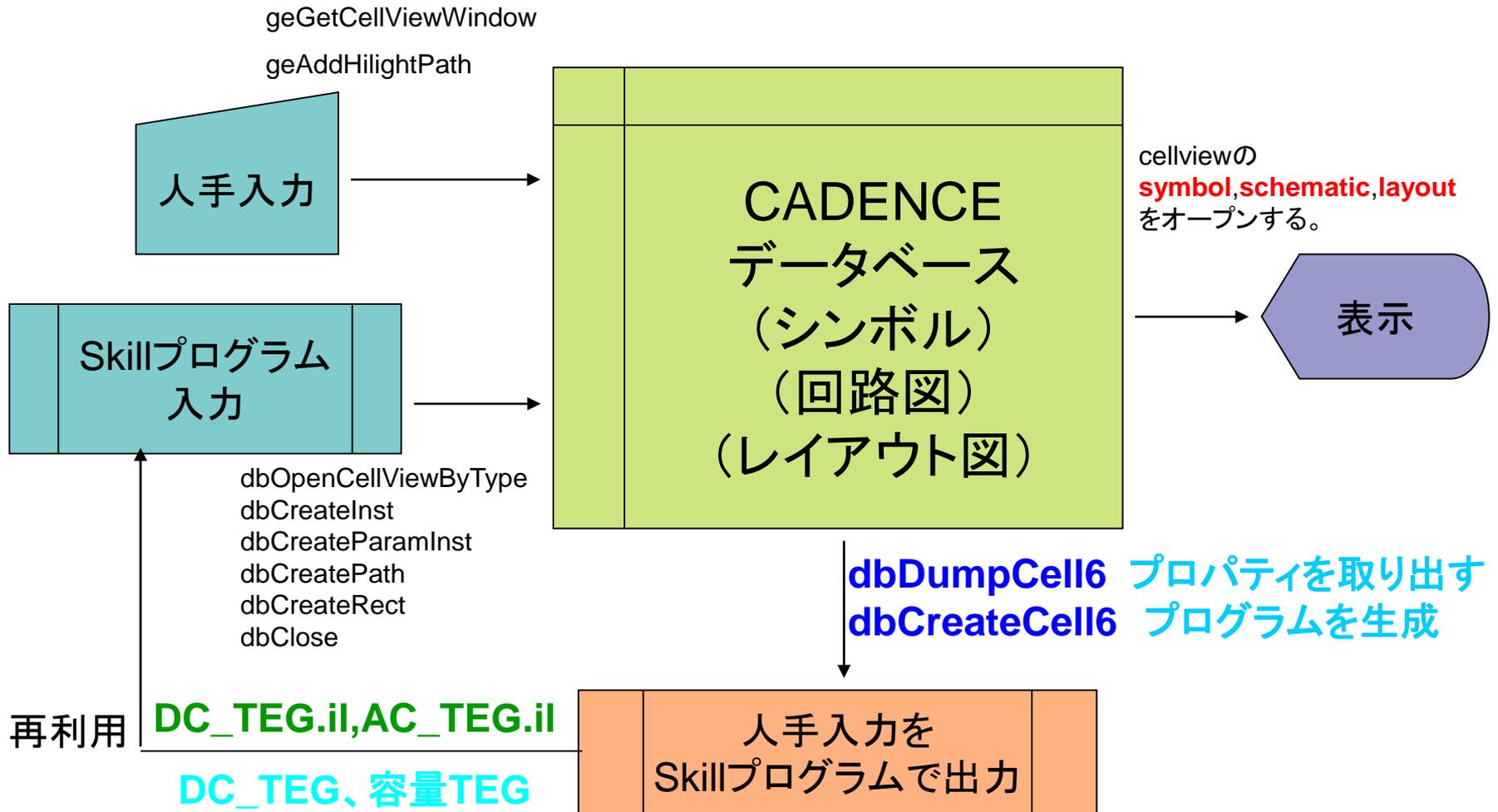
バロセロナ

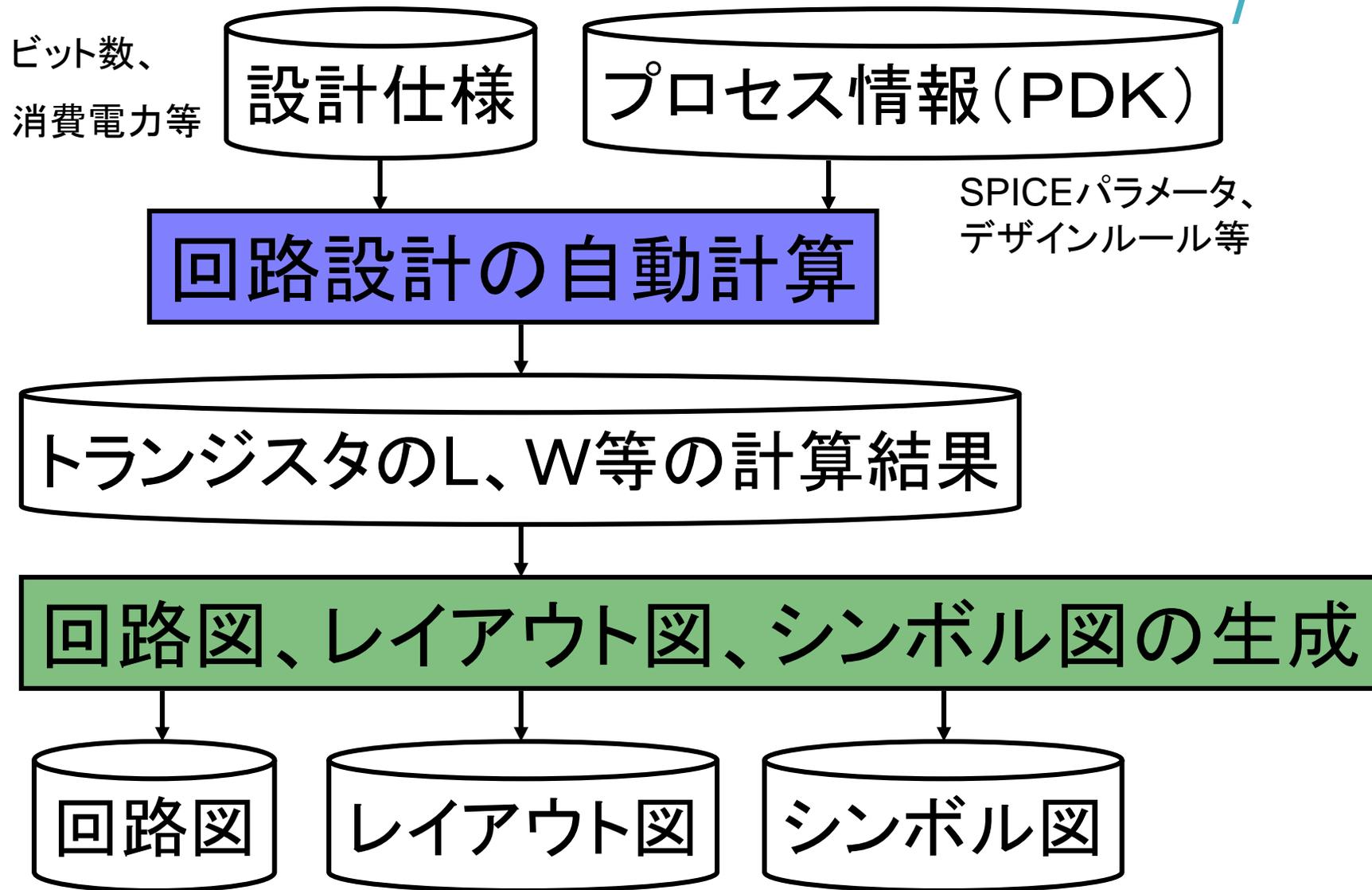
E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli, "Auto of IC Layout with Analog Constraints" *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 923-942, Aug. 1996.

3. 我々の自動合成

- 3. 0 開発フロー
- 3. 1 アナログ回路自動生成プログラムのフロー
- 3. 2 開発方法
- 3. 3 開発言語の紹介
- 3. 4 下地の回路図とレイアウト図
- 3. 5 回路図とレイアウト図
- 3. 6 我々の開発の格子
- 3. 7 プログラムによる階層化設計

3.0 開発フロー





- (1) SKILL言語を用いる。
- (2) DRCを満足する下地トランジスタ配置と、DRCを満足するように、トランジスタ間の配線を行う。(考え方が、ゲートアレーに近い)
- (3) Finger数4のトランジスタサイズを用いて、縦方向に、LOGIC回路、DFF回路、スイッチ回路を一直線に配置配線する。横は、Sourceを重ね、縦は、Viaを重ねることにより、アレー状に配置配線を行う。
- (4) LVSが掛けられるように、下の階層のpin情報をTOP階層まで持ち上げる。

3. 3 開発言語の紹介 (1)

2つの言語のライブラリの比較

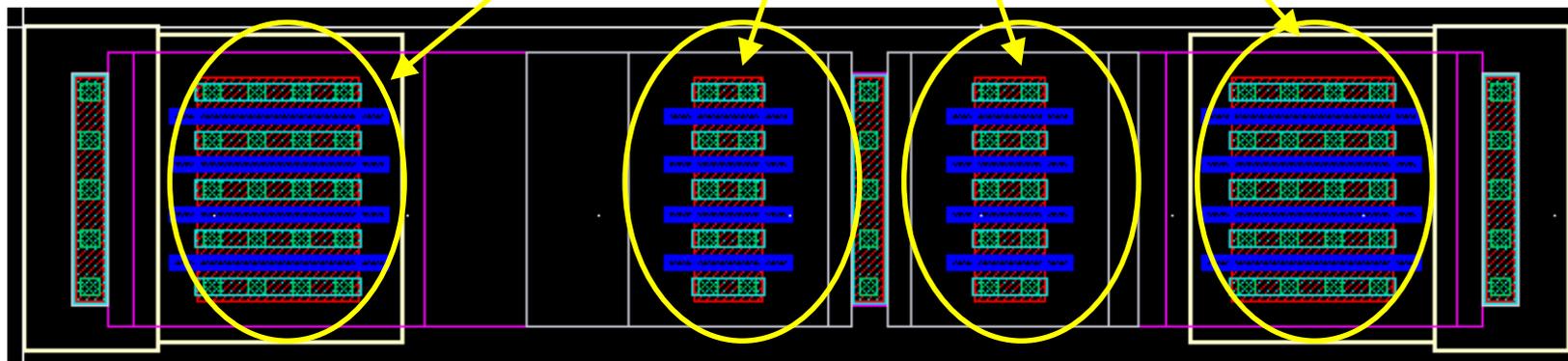
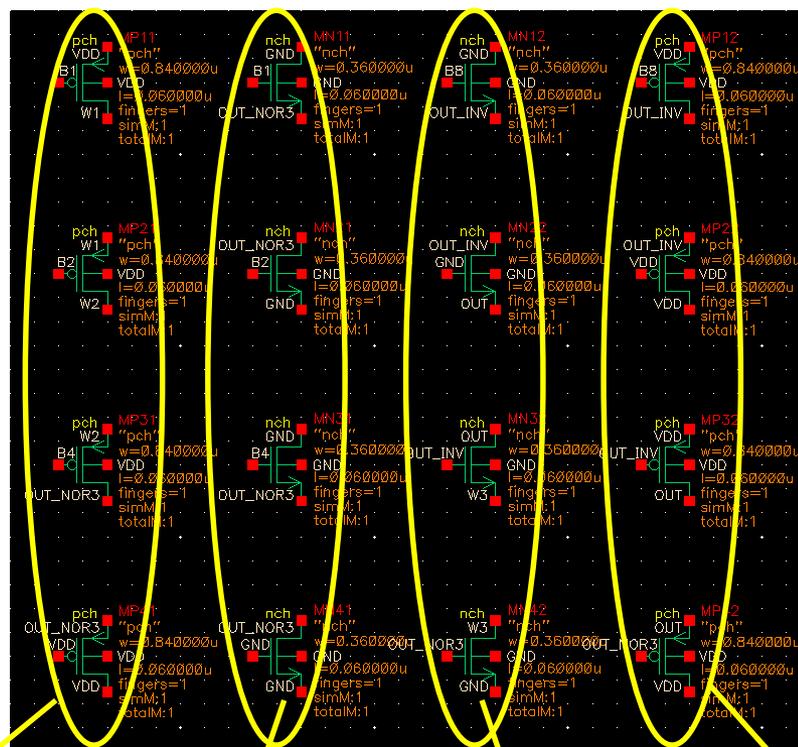
rect 層名 層番号 ((座標x1 座標y1) (座標x2 座標y2))
label 層名 層番号 ラベル名 (座標x 座標y) stick 1.0 ラベル位置 回転
path 層名 層番号 パス幅 座標数 ((座標x1座標y1) (座標x2 座標y2))
cell ライブラリ名 セル名 属性 ベース名 回転 (座標x 座標y)
pcell ライブラリ名 セル名 属性 ベース名 回転 (座標x 座標y) row column
pcell ライブラリ名 セル名 属性 ベース名 回転 (座標x 座標y) パラメータ

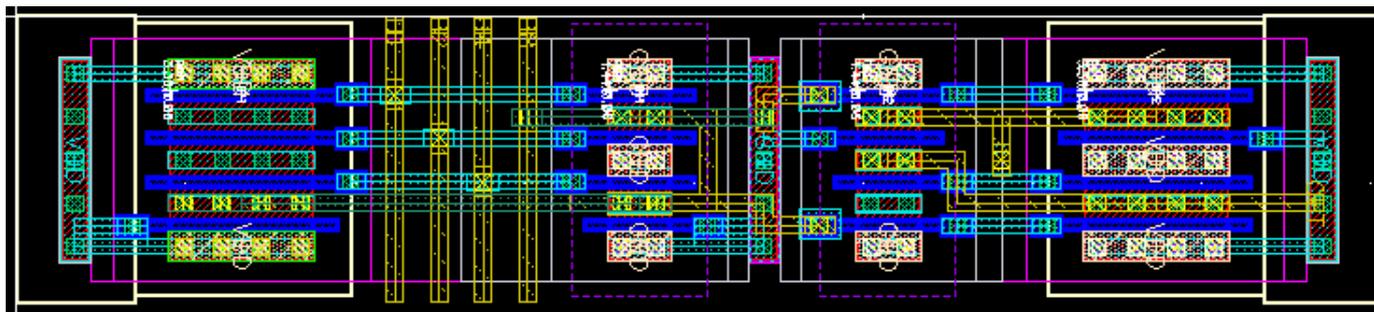
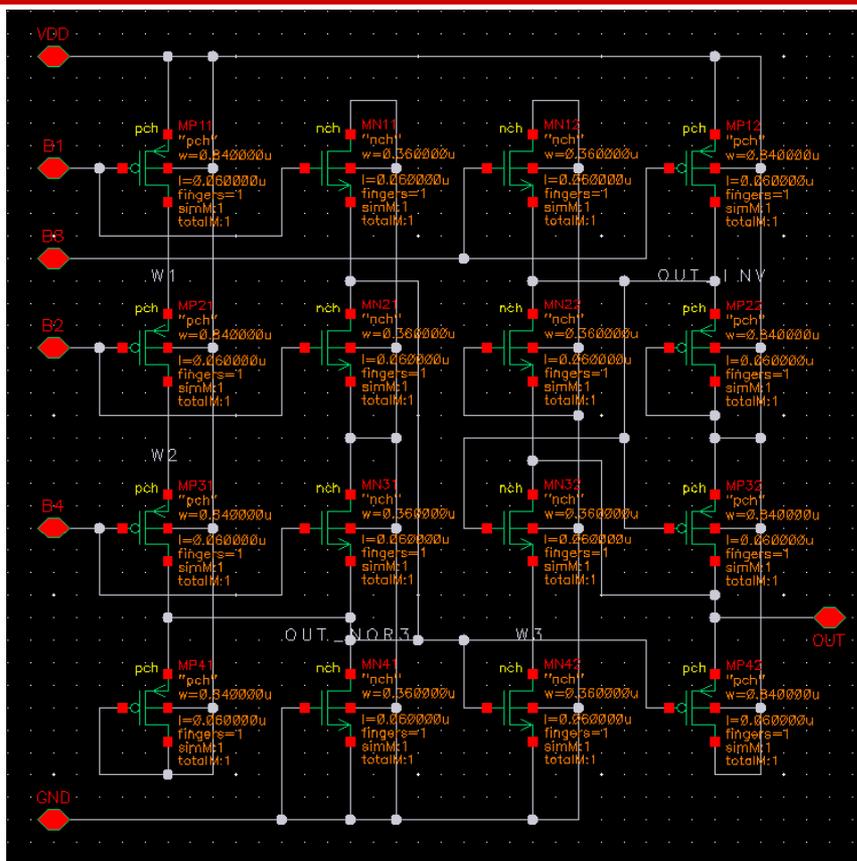
	SKILL	AXEL
RECT	dbCreateRect	fdd.AddRectangle
PATH	dbCreatePath	fdd.AddLine
CELL	dbCreateInst	fdd.AddCell
PCELL	dbCreateParamInst	fdd.AddParaO
LABEL	dbCreateLabel	fdd.Add

2つの言語の文法の比較

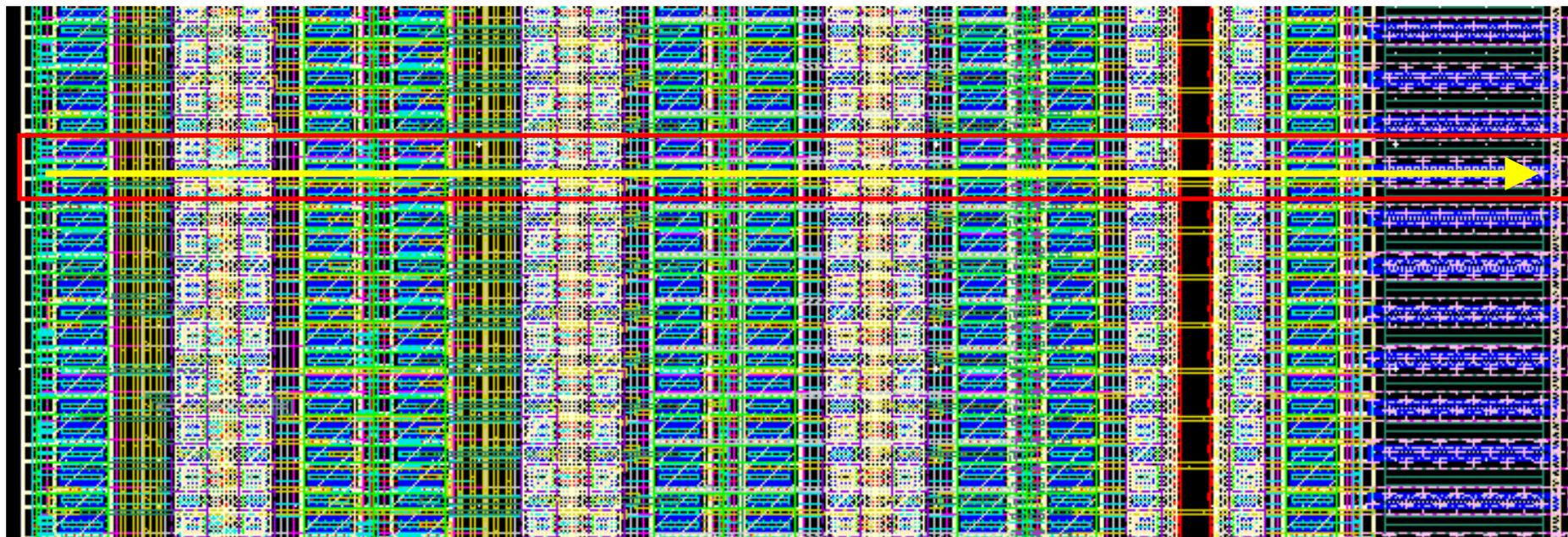
	SKILL言語	AXEL言語
関数	procedure(関数名(引数) prog((局所変数) 処理)	void 関数名(引数) {局所変数; 処理;}
ループ	for(i 1 2 処理)	for (i=1; i<=2; i++) {処理;}
分岐	if(判断 then 処理1 else 処理2)	if (判断) {処理1;} else {処理2;}
リスト	list(変数1 変数2) 変数1:変数2	pnts[0].x=変数1; pnts[0].y=変数2;
部品の 呼び出し	cv = dbOpenCellViewByType (libname cellname viewname "" "r") dbClose(cv)	関数名("libname/cellname/ viewname");

3. 4 下地の回路図とレイアウト図 (LOGICpn01)





LOGIC回路、DFF回路、スイッチ回路を 一直線に配置配線



```
load "LOGIC_common.il"
```

```
procedure(LOGICpn01(libname cellname Lmin Wn Wp)  
  prog())
```

```
  LOGICpn01_sch(libname cellname "schematic" Lmin Wn Wp)
```

```
  LOGICpn01_lay(libname cellname "layout" Lmin Wn Wp)
```

```
  LOGICpn01_sym(libname cellname "symbol")
```

```
)  
)
```

“LOGIC_common.il“の中に、下地の
回路図LOGIC_sch_base、レイアウト図LOGIC_lay_base
のプログラムが入っている。

```
procedure(LOGICpn01_sch(libname cellname viewname Lmin Wn Wp)
  prog((cv)
    cv = dbOpenCellViewByType(libname cellname viewname "" "w")

    LOGIC_sch_base(cv Lmin Wn Wp)
    LOGICpn01_sch_wire(cv)

    dbSave(cv) dbClose(cv)
  )
)
```

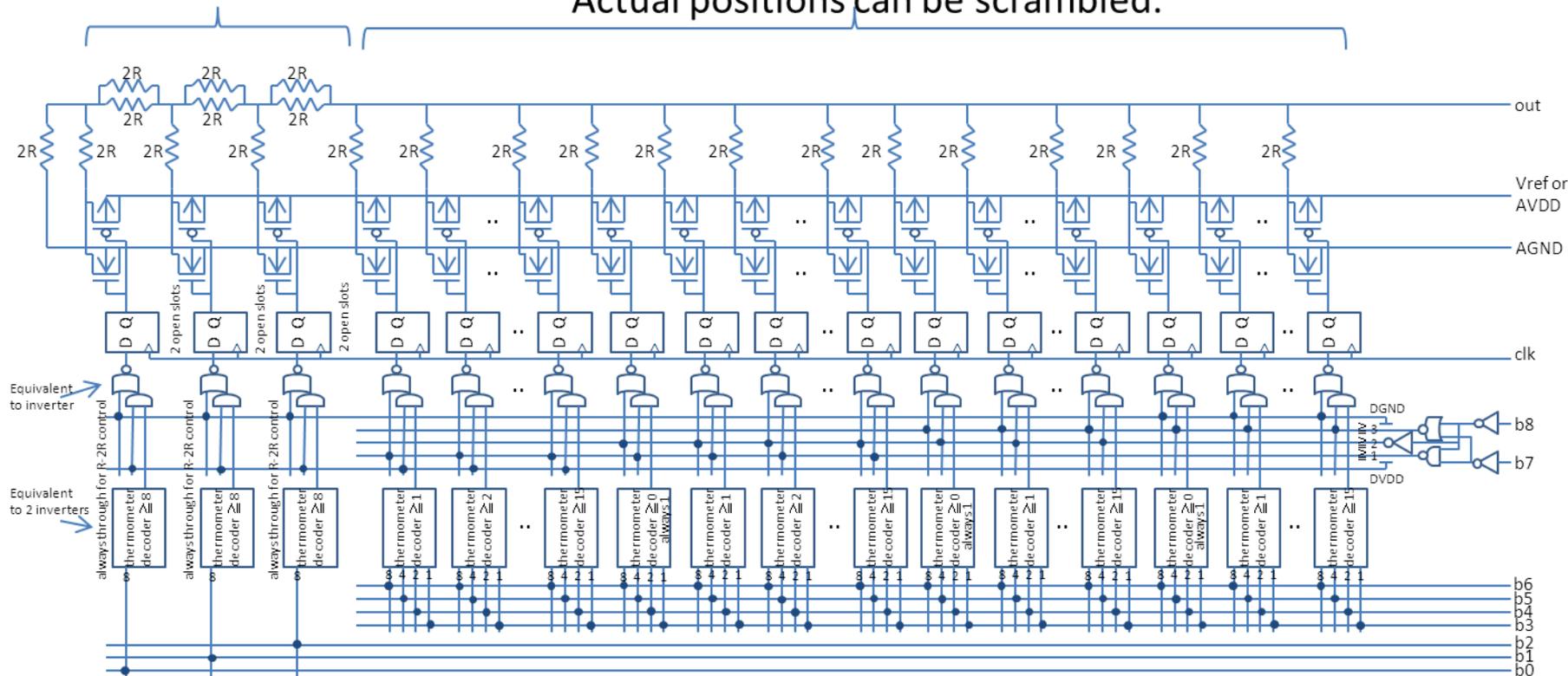
```
procedure(LOGICpn01_lay(libname cellname viewname Lmin Wn Wp)
  prog((cv width inout width_inout)
    cv = dbOpenCellViewByType(libname cellname viewname "" "w")

    width= LOGIC_lay_base(cv Lmin Wn Wp)
    inout= LOGICpn01_lay_wire(cv Lmin Wn Wp)
    width_inout=append(width inout)

    dbSave(cv) dbClose(cv)
    return(width_inout)
  )
)
```

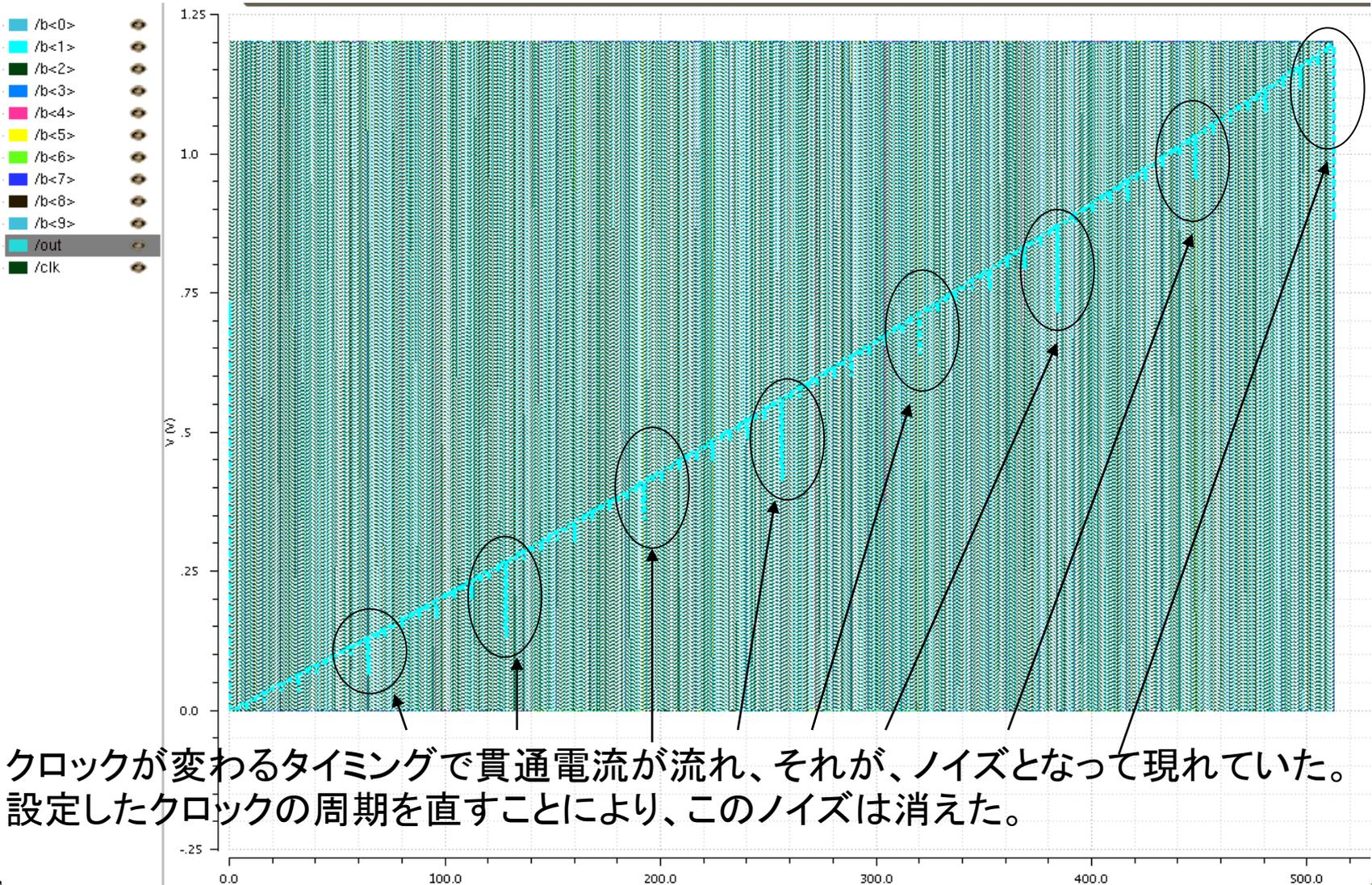
Total 3 pieces for R-2R

Total 63 pieces for thermometer code.
Actual positions can be scrambled.

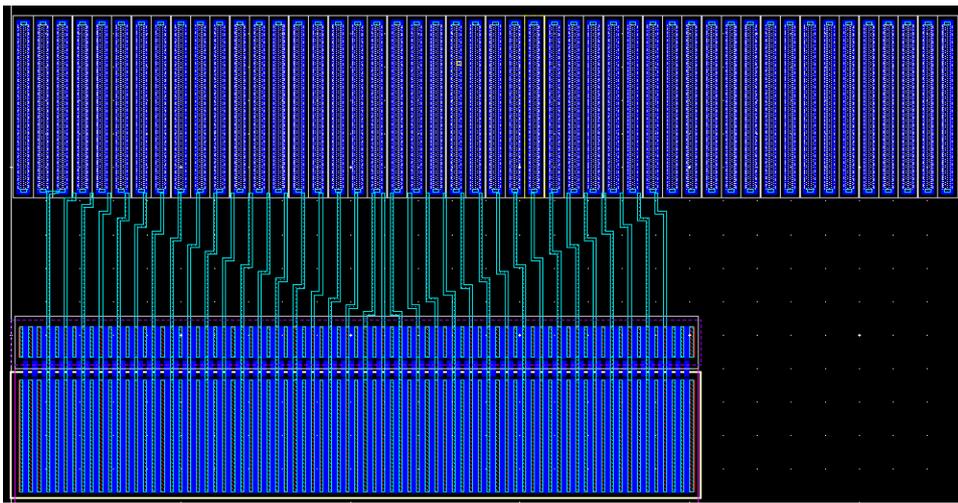


9ビットRDACの回路図

4. 2 シミュレーション結果 (1)

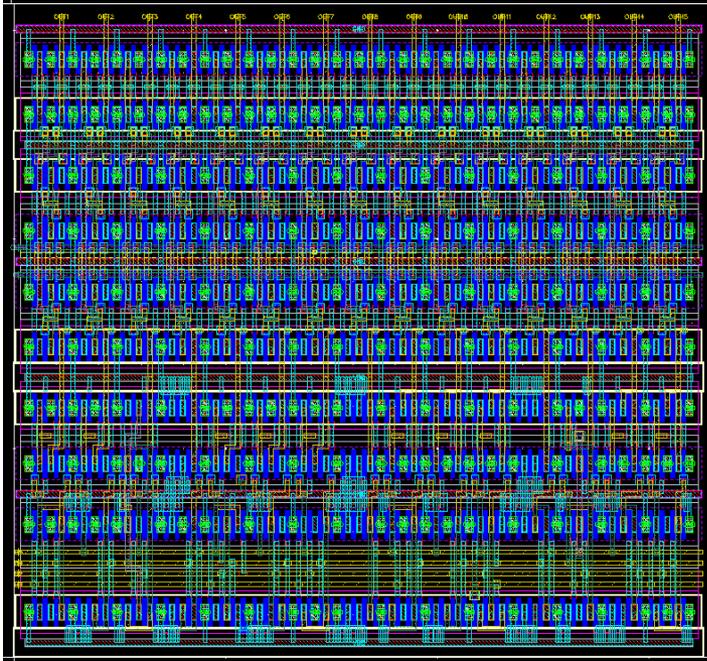


5. レイアウト結果 (1)



サーモ回路
(6ビット=2⁶=8×8=64)

スイッチ回路
(1~15)



LOGIC回路
(1~15)

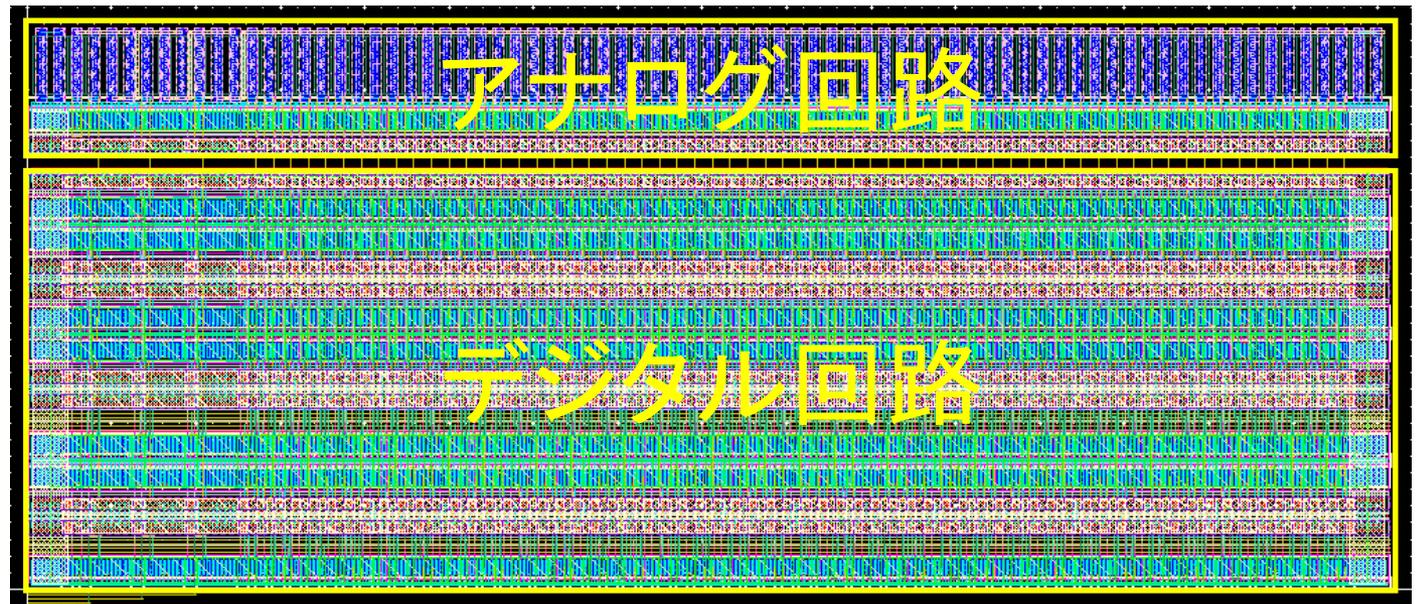
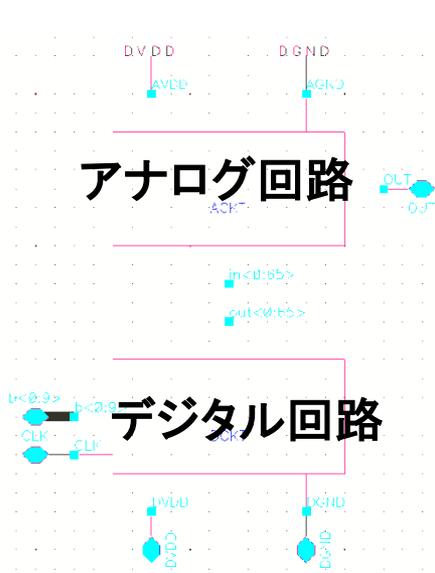
2013年5月13日時点

5. レイアウト結果 (2)

デジタル回路とアナログ回路の分離

デモ

デジタル回路でLVS
アナログ回路でLVS



12ビットRDAC、CDACの作成。

- (1) DFFのクロック信号をクロックツリーに変更する。
- (2) LOGICの入力信号をデータツリーに変更する。
- (3) 入力信号、クロック信号のツリー化に伴い、pnで終わる場合と、pnpで終わる場合がある為、下地レイアウトをpnタイプとnpタイプの2種類を用意する。
- (4) 引数指定で、下地レイアウトを4finger、8finger、12fingerに変えられるようにする。

9ビットRDACの自動生成に関して
お世話になった

東芝 吉富様、諏佐様

JEDAT 小野様

に感謝します。

ご清聴ありがとうございました。

質疑応答