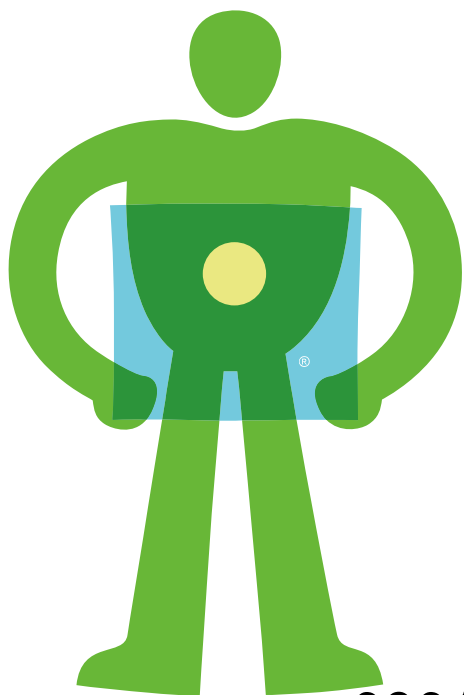


オープンハードウェア開発のための オープンソースEDAサーバ



森山 誠二郎 ((株)アナジックス)

2021年4月23日 @RISC-V Days Tokyo 2021 Spring

minimal[®]

発表の概要

オープンハードウェア開発のための オープンソースEDAサーバ

- (株)アナジックスは、少量生産の高付加価値LSI開発のサポートのためオープンソースEDA(OSEDA)を活用したローコストの設計環境を提供するとともに、お客様自身がLSI設計ができるように教育する
- 少量生産のLSI開発のためには、設計資産(IP)を共有し再利用する必要がある、そのためにはOSEDAの利用が必須である
- OSEDAの普及を助けるため、EDAサーバを立ち上げた
 - 面倒なインストール作業することなくOSEDAを体験できる
 - EDAサーバ設置の拡大を目指し、Chefを用いてインストールを極力自動化
 - オープンハードウェアの開発を支援する
 - EDAサーバの共同利用を契機に、OSEDAサポートコミュニティを構築

発表の概要(つづき)

■ OSEDAは、LSIハードウェアの開発に使われるものであり、オープンハードウェア開発が活発化しない限り、利用の拡大は望めない

■ オープンハードウェアは、米国のSkywater 130n PDKが脚光を浴びているが、誰でも参加できる我が国のMakeLSIプロジェクトではNDA不要のOpenRule1umを開発し2015年以來、年に1, 2回の試作をフリーまたはオープンソースのEDAツールを用いて行ってきた

■ EDAサーバには、OpenRule1umやミニマルファブを使ったカスタムLSI設計のできる環境(PDK)を整備した

■ ミニマルファブが得意な少量LSI開発において、アナログIPの共有と再利用を可能とするためにJupyter Notebookを用いた回路設計知識の共有に取り組む

Outline

■(株)アナジックスの取り組み

- ▶アナジックスの取り組む少量生産高付加価値LSI開発のサポート

■オープンハードウェア開発のプラットフォーム

■少量生産のLSI開発のためのOSEDA

■OSEDAの普及を助けるEDAサーバ

■アナログIPの共有と再利用

- ▶回路設計へのJupyter Notebook適用

■まとめ

(株)アナジックスの取り組み

■(株)アナジックスは、少量生産の高付加価値LSI開発のサポートのためオープンソースEDA(OSEDA)を活用したローコストの設計環境を提供するとともに、お客様自身がLSI設計ができるように教育する

■少量生産のLSI開発のためには、設計資産(IP)を共有し再利用する必要があり、そのためにはOSEDAの利用が必須である

■OSEDAの普及を助けるため、EDAサーバを立ち上げた

➤ミニマルEDAサーバ: ミニマルファブ用開発がメイン

➤オープンソースEDAサーバ

✓ #1: 福岡ISTに設置(メモリ312G)

✓ #2: 同上(メモリ520G)



最新ニュース (2021/3/6 23:31:00) :

[すべてのニュース](#)

3/6研究会発表スライドとWHITE PAPERを公開します

費用をかけずにアナログLSI開発を始めることのできるアナログ回路設計統合システムWebベースの回路設計環境ALB (Anagix Library Builder) およびALTA (Anagix Local Tools Assistant) を使えば、アナログ半導体集積回路の設計効率を向上すると同時に EDAコストを低減することができます。ALBはサーバとして回路データや設計文書をWeb (インターネットやイントラネット) 上で管理します。また、半導体回路設計で標



Outline

- (株)アナジックスの取り組み

- ➡ ■オープンハードウェア開発のプラットフォーム

- 少量生産のLSI開発のためのOSEDА

- OSEDАの普及を助けるEDAサーバ

- アナログIPの共有と再利用

 - 回路設計へのJupyter Notebook適用

- まとめ

オープンハードウェア開発のプラットフォーム

■ OSEDAは、LSIハードウェアの開発に使われるものであり、オープンハードウェア開発が活発化しない限り、隆盛は望めない

■ プラットフォーム

- Skywater 130n PDK
- MakeLSIプロジェクトのOpenRule1um
- ミニマルファブ
 - ✓ PMOSのみのプロセス
 - ✓ SOI CMOSプロセス

Skywater/Googleの130nm PDK

■ skywater(<https://www.skywatertechnology.com>)はCypressからスピンオフしたUSベースのファンダリ

■ NDA不要のPDK(Process Design Kit)

■ Googleがefablessに出資し、2020年11月に無料シャトル開始

➤ 簡単な回路で、必ずオープンにすること、githubを通じてGDS提出、オープンソースPDKのブラッシュアップに貢献しうること、などを条件に、40 slots(チップ)を募集

➤ 2021年は数回やりたい

Googleが半導体チップの設計に必要な「PDK」をオープンソース化するプロジェクトを支援



Fully open
ASIC!

Google + skywater
FOSS 130nm Production PDK
github.com/google/skywater-pdk

j.mp/du20-sky130

MPW-TWO is now open
April 19: Submission OPEN
June 18: Submission CLOSE
October: Wafer fabrication

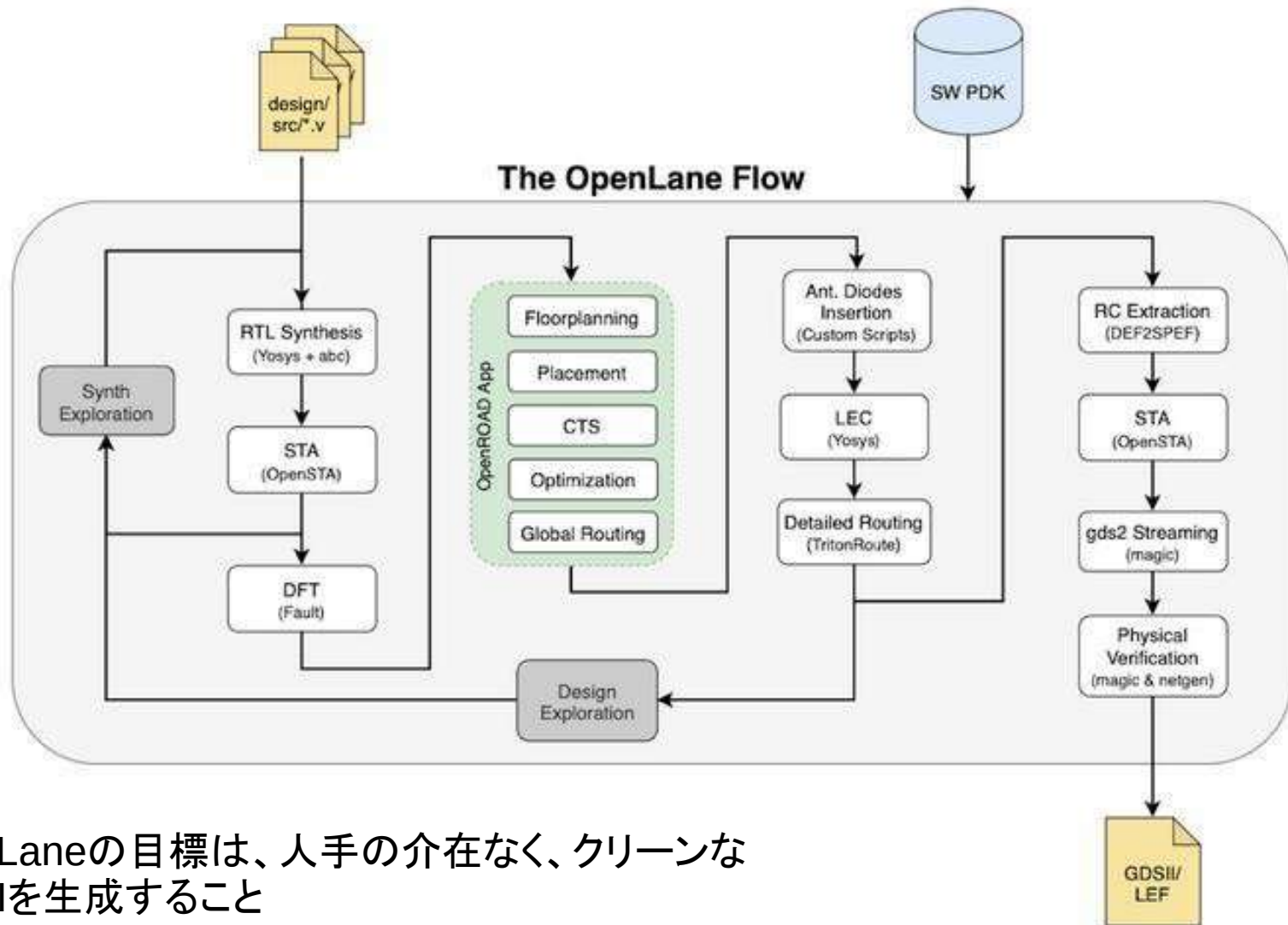
efabless.comのOpenLane

■ efabless.comは半導体チップを作りたい企業に、アナログ、ミックスドシグナルのIPを供給する企業（マーケットプレイスでもある） 収益モデル:

- 1.顧客からの半導体チップ設計リクエストを受け、IPを選択もしくはない場合はコミュニティなどに設計依頼をだし、そのIPを元に設計提案を行う
- 2.設計ソリューション(ツールやPDKなど)を提供して製造サービスを行う

■ OpenLaneは、efablessがGoogleとSkyWaterの130nmプロセス(NDA不要)を使うために整備したオープンソースEDA。OpenROAD, OpenRAMなど多数のオープンソースprjから16個のEDAツール採用。70個以上のスクリプトとユーティリティからなる。詳細は: OpenLaneとは

OpenLaneのフロー



OpenLaneの目標は、人手の介在なく、クリーンなGDSIIを生成すること

OpenLaneに採用されたOSS EDA

- Yosys+ABC: RTL synthesis
- OpenSTA: STA(Static Timing Analysis)
- Fault: DFT (Design for Test)
- OpenROAD: Placement and Routing
- Yosys: LEC(Logic Equivalence Check)
- DEF2SPEF: 寄生RC抽出
- Magic: Layout editor, DRCとSPICEネット抽出
- Netgen: LVS (Layout vs. Schematic)

NDA不要のOpenRule1um設計ルール

■レイアウトIPが流通できない原因の1つは、ファブと結ぶ設計ルールについての秘密保持契約(NDA)

■秋田先生のMakeLSI:プロジェクトでは、NDA不要のOpenRule1umルールを定義し、フェニテック0.6umプロセス等での試作に使用している

■OpenRule1umルールで作成したスタンダードセルも存在するが、Qflowで使うための環境整備ができていない

■Magic対策として、Qflow-nomagicフローを開発

➤ <https://scrapbox.io/makelsi/OpenRule1um>でQflowを使う
Qflow-NM_GUIについて

NDA不要のOpenRule1umとは

■ MakeLSI:プロジェクトで、秋田先生が中心になって開発しているNDA不要の設計ルール

	A	B	C	D	E	F	G	H
1		モデル_cmos 0.6μ Metal 3Layer			OpenRule1 um ver1	20 (18/0	2018/03/11 v 1.0 03/25 v1.1	yamada
2		input A	input B	Width	Space	Enc	Ext	備考
3	1	NWL	NWL	4.0	4.0			
4	2	NWL_dp	NWL_dp	4.0	5.0			異電位NWL
5								
6	11	Parea	Parea	0.5	0.5			
7	12	Narea	Narea	0.5	0.5			
8	13	Parea	Narea		0.5			
9	14	Parea	diff			0.5		
10	15	Narea	diff			0.5		
11								
12	21	diff	diff	1.0	1.5			
13	22	NWL	Pdiff			2.0		$Pdiff=diff \cap Parea$
14	23	NWL	Ndiff			3.0		$Ndiff=diff \cap Narea$
15	23a	NWL	Ndiff			1.0		対象は NWL 基板CNTに限る
16	24	Pdiff	DM_nscn	外形サイズ>	0.5			DM_nscn 提供セルでの使用のみ $DM_nscn=DM_dcn+(Ndiff)+(NWL)$
17	25	DM_nscn	DM_nscn	外形サイズ>	0			
18	26	Ndiff	DM_pscn	外形サイズ>	1.0			DM_pscn 提供セルでの使用のみ $DM_pscn=DM_dcn+(Pdiff)+ML1$
19	27	DM_pscn	DM_pscn	外形サイズ>	0			
20								
21	31	POL	POL	1.0	1.0			
22	32	Gate	Gate		1.0			Gate=Diff∩POL
23	33	POL	diff		0.5		1.0	注) width=0.6に変換する場合には Extは1.5μ必要
24								
25	41	DM_dcn	外形サイズ>ク×ク		0			DM_dcnは提供セルでの使用のみ $DM_dcn = CNT+diff+ML1$

OpenRule1umを使った設計例

The image displays a CAD tool interface showing a circuit schematic on the left and its corresponding physical layout on the right. The schematic is a complex multi-stage circuit with various components labeled, including resistors (R1-R22), capacitors (C1-C4), and transistors (M1-M20). A parameter `param rhpo = 4k` is defined at the bottom of the schematic. The physical layout shows the circuit components mapped to a multi-layer PCB stack-up. A scale bar indicates $100\ \mu\text{m}$.

The physical layout window includes a toolbar with icons for Box, Text, Path, Instance, Partial, and YSS_SOI. The title bar shows the file name `bgr8_moriyama.GDS [bgr_simple]`. The Layers panel on the right lists the following layers:

- NWL
- NWL_dp
- DIFF
- POL
- HPOL
- CNT
- ML1
- VIA1
- ML2
- VIA2
- ML3
- TEXT
- RES
- FRAME
- Parea
- Narea
- DM_dcn
- DM_pcn
- DM_nscn
- DM_pscn
- DM_via1
- CAP

The status bar at the bottom shows the page number `15 / Anagix`, the tool name `T OpenRule1um`, the grid `G`, the coordinates `xy`, the zoom level `305.25904`, and the page number `-3.34714`.

Qflow: デジタル設計フロー

■ Qflowは、Tim Edwards (<http://opencircuitdesign.com>) さんが管理するツール群の1つで、Magic, yosys, graywolf, Qrouter などのOSSで構成されるVerilog入力のデジタル設計フロー

■ Oklahoma State Universityの開発するSCMOSのスタンダードセルライブラリ (osu050, osu035, osu018) を使ってMOSISなどで試作できる

■ MakeLSI: プロジェクトでも、osuライブラリのラムダ値をOpenRule1umルールに合うように調整して試作した実績あり

➤ MakeLSI: の金沢大秋田先生が、3000ゲート程度のRisc-Vを使ったLチカ専用チップを設計

ミニマルファブとは

■少量製品のLSI化を担う新産業の中核となる新しい生産方式

- 0.5インチウェーハ
- 幅294 mm × 奥行き450 mm × 高さ1440 mmの製造装置
- マスクレス露光
- LSI一個からの製造が可能
- 最短3日間でLSI試作



■産総研が主導した国家プロジェクト

- 社団法人化し、多数の中小半導体製造装置メーカーが参加
- 商社も参加し、ロシアや東南アジアへの売込活発化
- ユーザ企業の参加も始まっている

ミニマルファブは

- 速い
- 安い
- 面白い

ミニマルファブは速い

- 超短TAT

- PMOSのみのプロセスの場合、実績で

3.5日

ミニマルファブは安い

- 少量多品種
- 1個から作ることができる
- 高価なマスクの要らない、マスクレス露光方式
- 製造設備は、おおざっぱにメガファブの

1/1000のコスト

ミニマルファブは面白い

- 装置を見ているだけでも楽しめる
- 自分で操作することができる
- クリーンルームに入る必要なし
- 出来あがる過程を楽しむことができる
(上級編)
- 製造レシピを変えることができる
- 特殊な材料を使うこともできる

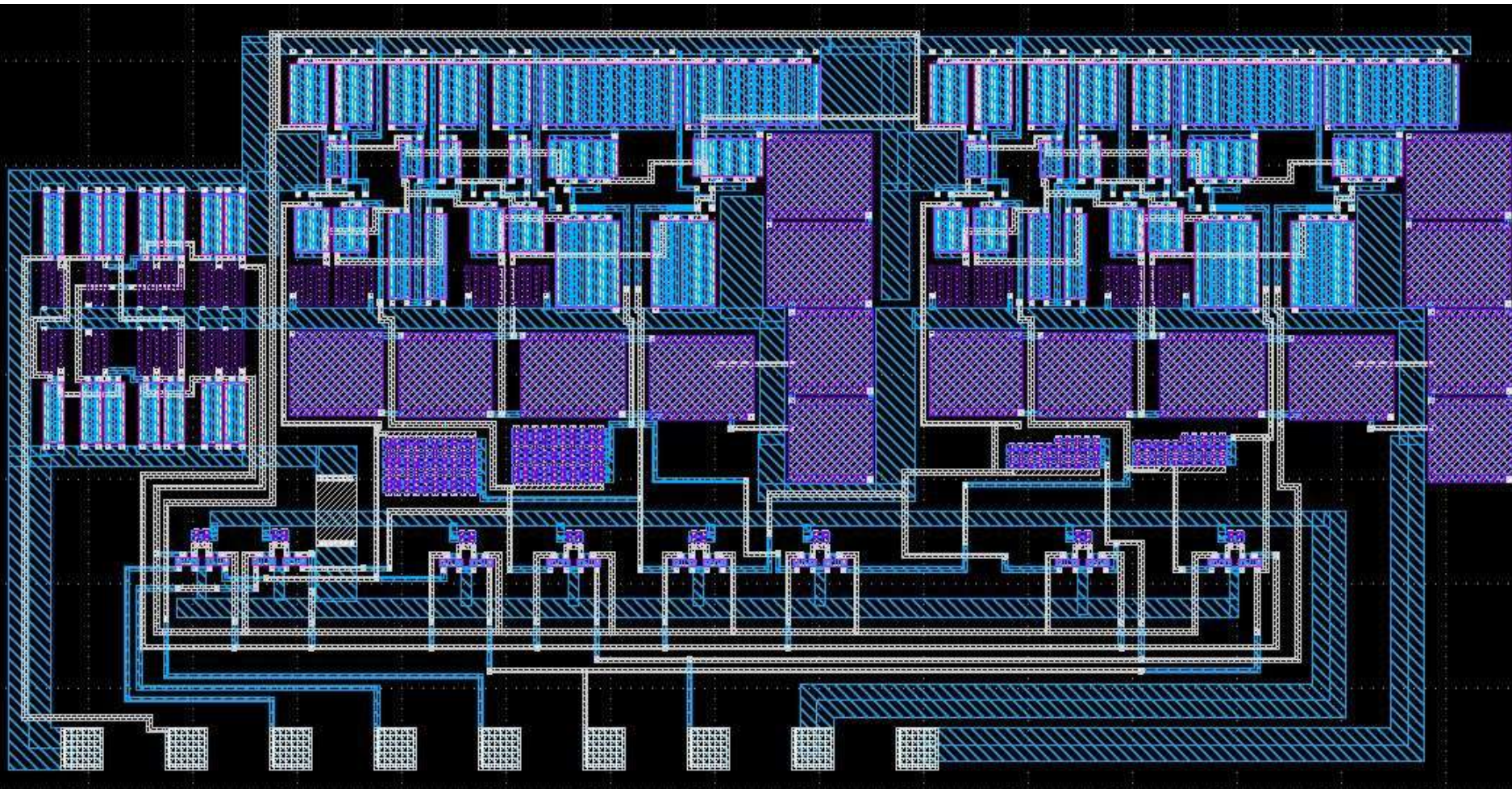
回路とレイアウト設計も面白い

- 自分のアイデアを試すことができる
- 計算機シミュレーションにより検証できる
- PDKがあるので、レイアウトをかくのは難しくない
- レイアウトルール違反は、DRCで検証できる
- 回路とレイアウトの一致は、LVSで確認できる
- 回路設計とレイアウト設計の良しあしを、

実際に作って評価できる

→ あなたも今日からLSI設計者の仲間

SCF回路のレイアウト



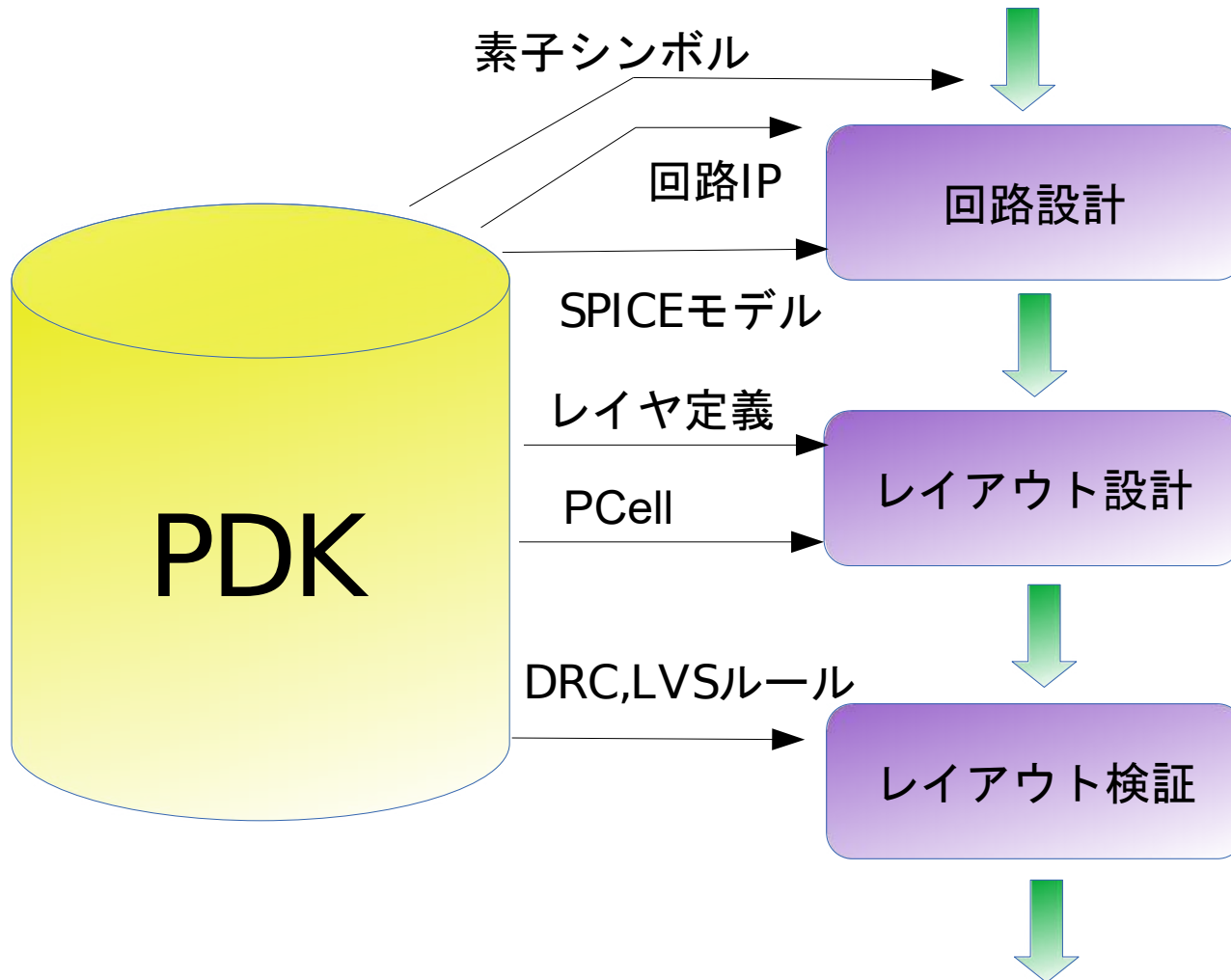
PDKとは

■ Process Design Kit

LSI設計に必要な情報・部品を集めたもの

- 素子シンボル
- SPICEモデル
- レイアウトのレイヤ定義
- PCell (Parameterized Cell)
- DRCルール
- LVS抽出ルール
- IPライブラリ

PDKと設計の流れ



PMOSプロセス用PDKユーザマニュアル

ファイル (F) 編集 (E) 表示 (V) ブックマーク (B) ツール (T) ウィンドウ (W) ヘルプ (H) Mineda2020_1 PDKユーザマニュアル - Dropbox Paper

Mineda2020_1 PDKユーザマ

paper.dropbox.com/doc/Mineda2020_1-PDK--A~0dC0CUG9rM74AY... Googleで検索

Mineda2020_1 PDKユーザマニュアル ★
フォルダに追加 他1人と共有しました

招待

Mineda2020_1 PDKユーザマニュアル

- 回路図の素子シンボル
- SPICEモデル
- レイアウトのレイヤ定義
- DRCルール
- LVS抽出ルール
- PCell (Parameterized Cell)
- IPライブラリ

EDAツール

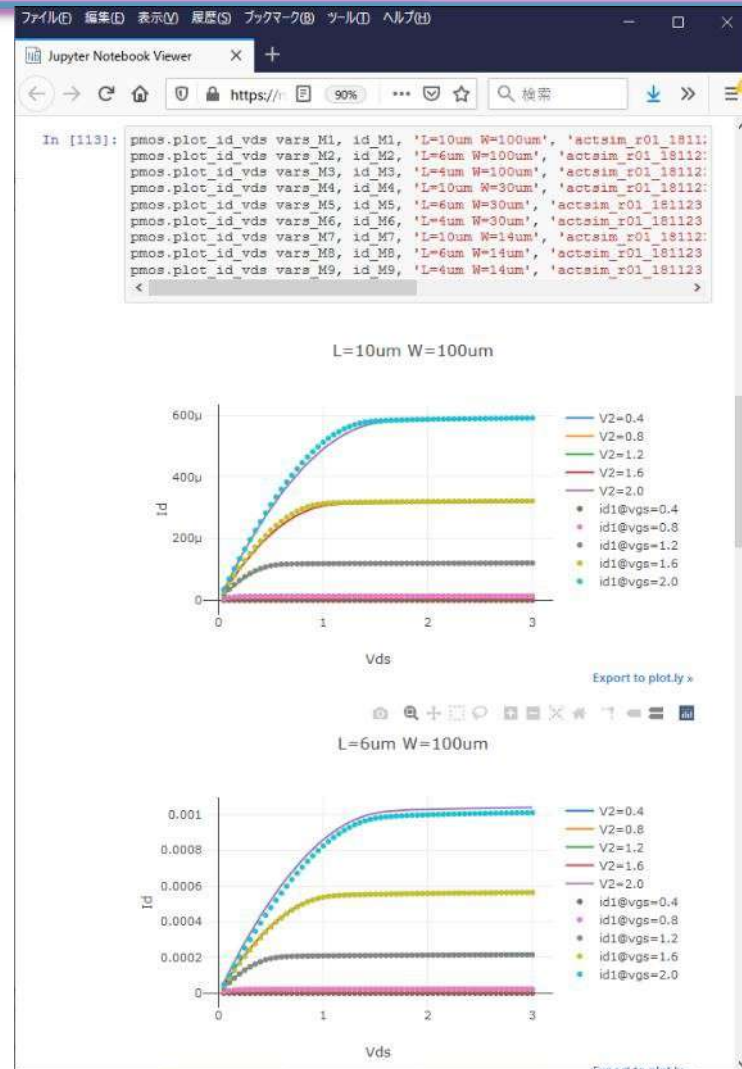
回路図の作成に使えるEDAツールは、LTspiceのほか、オープンソースのQucs-S、XschemおよびEESchemaです。レイアウト設計に使えるEDAツールは、KLayout (v0.26.4以降) です。PDKは、KLayout用パッケージとして提供されるので、回路図作成ツールの設定は指示にしてください。なお、デジタル設計用のスタンダードセルライブラリは存在せず、自動配置配線ツールも対応していません。

更新: 14 日前

SPICEモデル

■モデルファイルとともに実測データとの比較グラフを示す

```
mos_tt.lib - ノモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
*
*.LIB mos_tt
* DATE: Mar 4/10
* LOT: 1
* DIE: 1
* Temp= 27
* WAF: 1
* DEV: nW120L360
.MODEL nch nmos (
+VERSION = 3.3
+XJ = 1.5E-7
+K1 = 0.651491
+K3B = 0.1534115
+DVTOW = 0
+DVT0 = 0.7513662
+UO = 0.0475844
+UC = -2.72549E-11
+AGS = 0
+KETA = -5.06235E-3
+RDSW = 2.330784E3
+WR = 1
+DWG = -6.56397E-8
+DWB = -5.57927E-8
+CIT = 0
+CDSCB = 0
+DSUB = 0.0889854
+PDIBLC2 = 1E-5
TNOM = 27
NCH = 1.7E17
K2 = 0
WO = 1.187536E-6
DVT1W = 0
DVT1 = 0.0394835
UA = 9.703369E-10
YSAT = 1.26498E5
BO = 1.401621E-6
A1 = 0
PRWG = 6.334957E-4
WINT = 2.563443E-7
VOFF = -0.0819032
CDSC = 1.002311E-4
ETAO = 0
PCLM = 2.305038
PDIBLCB = 0
LEVEL = 8
TOX = 2.5E-8
VTH0 = 0.5170719
K3 = 29.04952
NLX = 5.847095E-8
DVT2W = -0.032
DVT2 = -0.0491886
UB = 1E-21
AO = 0.56427
B1 = 1E-7
A2 = 1
PRWB = -1E-3
LINT = 2.689635E-8
NFACTOR = 0.4459316
CDSCD = 0
ETAB = 0
PDIBLC1 = 0
DROUT = 0.2827366
)
```



プロセス情報とレイアウト定義

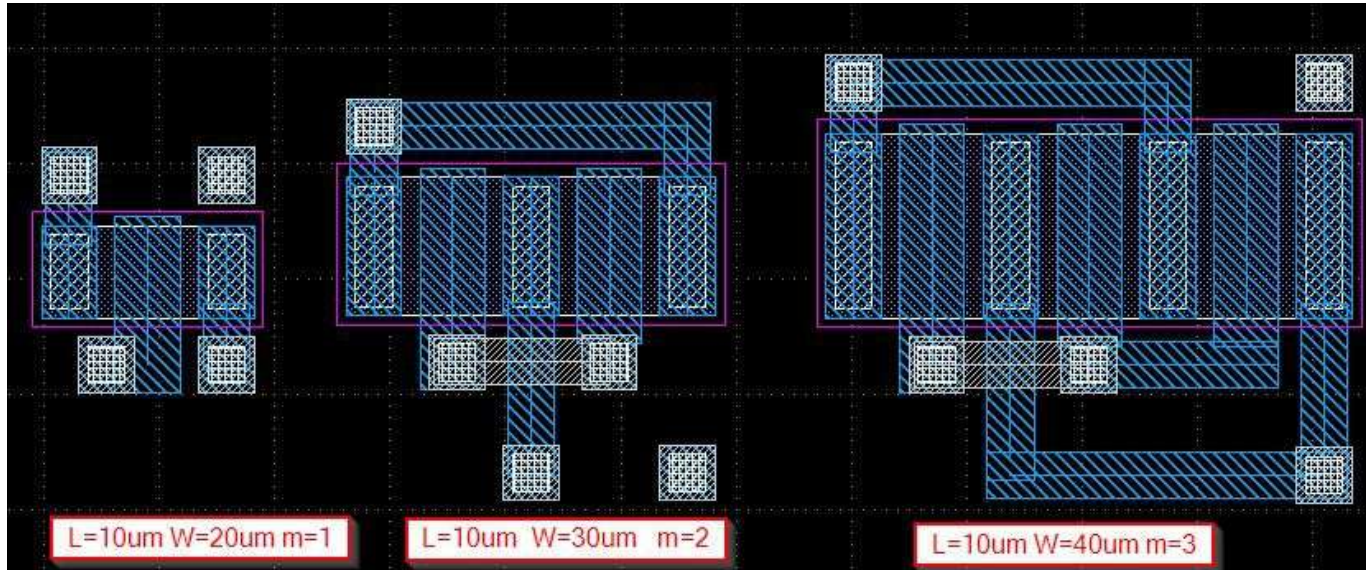


レイヤ名称	GDS番号	用途
DIFF	3	拡散レイヤ
CNT	7	コンタクトレイヤ
ML1	8	メタル1レイヤ
VIA1	9	ビアレイヤ
ML2	10	メタル2レイヤ
RES	15	抵抗認識レイヤ
CAP	16	キャパシタ認識レイヤ
Parea	18	P型アクティブレイヤ
DM_dcn	101	レイヤ自動生成のためのダミーレイヤ
DM_nscn	103	同上
DM_via1	105	同上

注意： Xが表示されているレイヤは、標準的なCMOSプロセスに対応したOpenRule1umで使用されているものです。Mineda2020_1では使用できません。

PCell

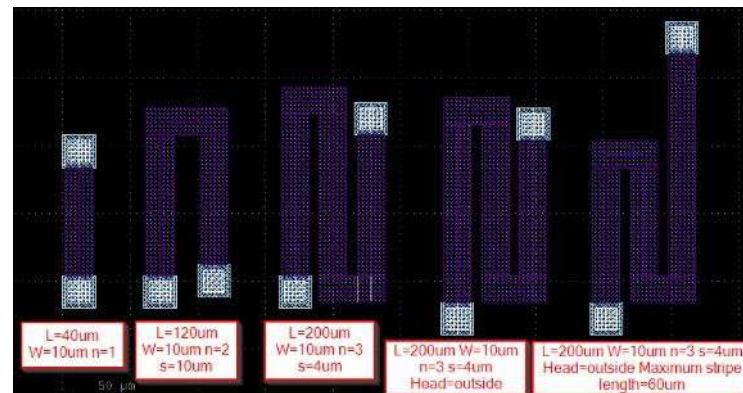
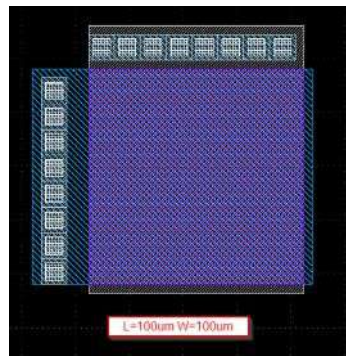
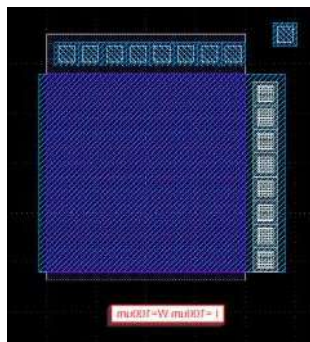
PMOSトランジスタ



MIM容量

拡散容量

拡散抵抗



DRCルールファイル

```
# Mineda2020_1 Rule for YSS process
## ver0.3: 2020/2/7: Seijiro Moriyama
seijiro.moriyama@anagix.com
```

```
report("Output database")
```

```
# Get raw layers
```

```
nwl = input(1, 0)
```

```
nwl_dp = input(2, 0)
```

```
diff = input(3, 0)
```

```
pol = input(5, 0)
```

```
hpol = input(6,0)
```

```
cnt = input(7, 0)
```

```
ml1 = input(8, 0)
```

```
via1 = input(9, 0)
```

```
...
```

```
...
```

```
puts 'Check GAP'
```

```
r_pdiff_s = pdiff.space(10.um);
```

```
r_pdiff_s.output("PDIFF space < 10")
```

```
r_ml1_s = ml1.space(4.um);
```

```
r_ml1_s.output("ML1 space < 4")
```

```
r_ml2_s = ml2.space(10.um);
```

```
r_ml2_s.output("ML2 space < 10")
```

```
puts 'Check Width'
```

```
ml1_w = ml1.width(10.0.um);
```

```
ml1_w.output("ML1 width < 10")
```

```
ml2_w = ml2.width(10.0.um);
```

```
ml2_w.output("ML2 width < 10")
```

```
via1_w = via1.width(8.um);
```

```
via1_w.output("Via1 width < 8")
```

```
cnt_w = cnt.width(8.um); cnt_w.output("cont  
width < 8")
```

検証ルールの整備

【DRCルール】

spaceチェック

```
r_pdiff_s = pdiff.space(10.um); r_pdiff_s.output("PDIFF space < 10")
r_ml1_s = ml1.space(4.um); r_ml1_s.output("ML1 space < 4")
r_ml2_s = ml2.space(10.um); r_ml2_s.output("ML2 space < 10")
```

widthチェック

```
ml1_w = ml1.width(10.0.um); ml1_w.output("ML1 width < 10")
ml2_w = ml2.width(10.0.um); ml2_w.output("ML2 width < 10")
via1_w = via1.width(8.um); via1_w.output("Via1 width < 8")
cnt_w = cnt.width(8.um); cnt_w.output("cnt width < 8")
```

enclosureチェック

```
diff_e = diff.enclosing(cnt, 2.um); diff_e.output("cnt enclosure in diff < 2.0")
```

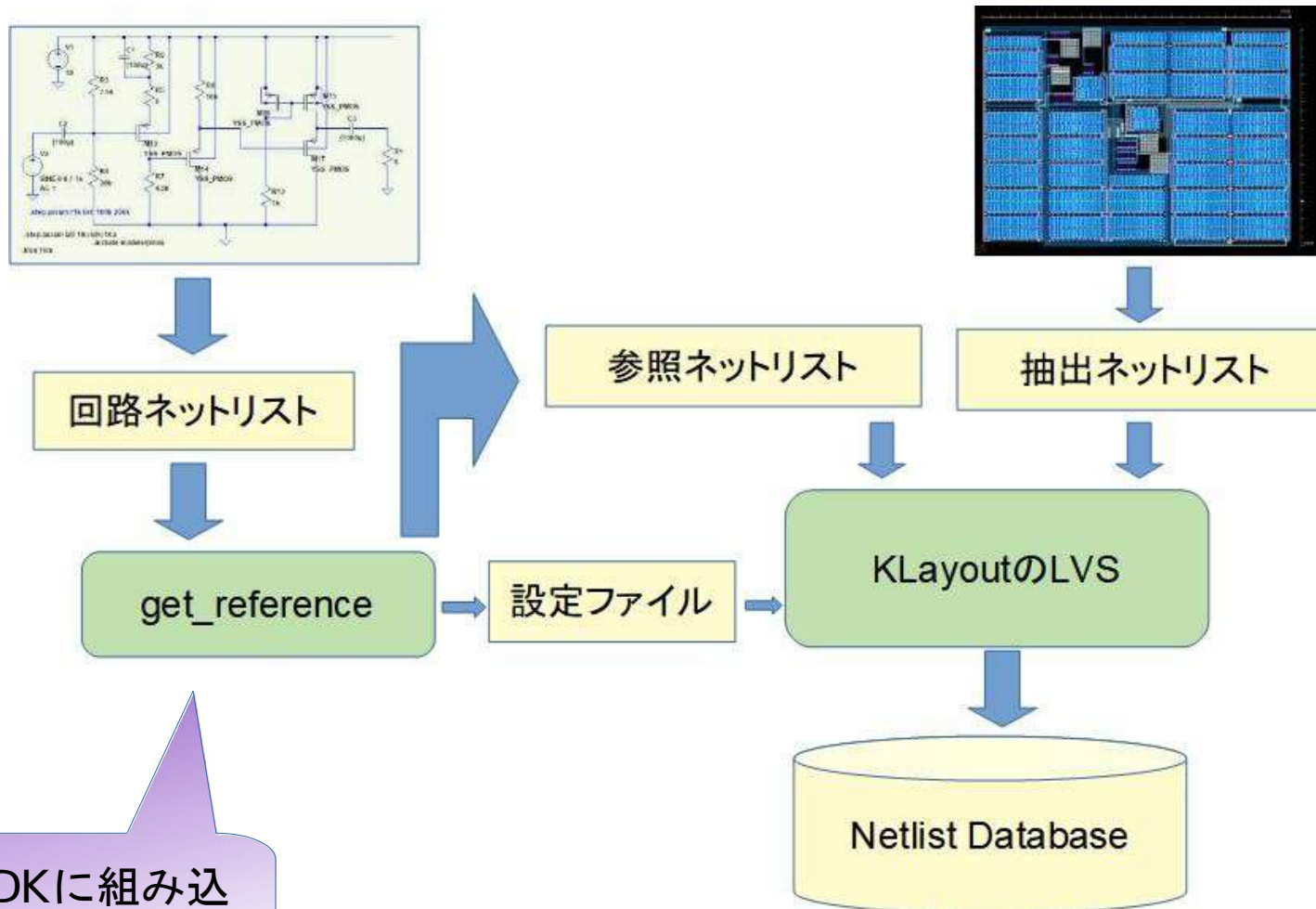
LVSルール

```
# Mineda2020_1 lvs rule v1.4  
Apr. 6, 2020 copy left by S.  
Moriyama (Anagix Corporation)
```

```
deep  
report_lvs  
# Drawing layers  
nwl = input(1, 0)  
nwl_dp = input(2, 0)  
diff = input(3, 0)  
pol = input(5, 0)  
hpol = input(6,0)  
cnt = input(7, 0)  
ml1 = input(8, 0)  
via1 = input(9, 0)  
...  
ml1_in_parea = ml1 & para  
ml1diff =  
ml1_in_parea.interacting(diff)
```

```
ml1cnt = ml1_in_parea.interacting(cnt)  
gate = ml1diff - ml1cnt  
new_diff = diff - (diff & gate.sized(-2.um))  
...  
sheet_rho = 700  
model_name = "RES"  
cap_layer = res.sized(4.0.um) & (diff - res)  
extract_devices(resistor(model_name,  
sheet_rho), {"R" => diff_in_res, "C" =>  
cap_layer, 'tA' =>cap_layer, 'tB'=>cap_layer })  
  
extract_devices(mos4("PMOS"), { "SD" => psd,  
"G" => lv_pgate, "W" => bulk, "tS" => psd, "tD"  
=> psd, "tG" =>ml1, "tW" => bulk })  
...  
connect(psd, cnt)  
connect(diff_not_res, cnt)  
connect(cap_layer, ml1)  
...
```


LVSの前処理

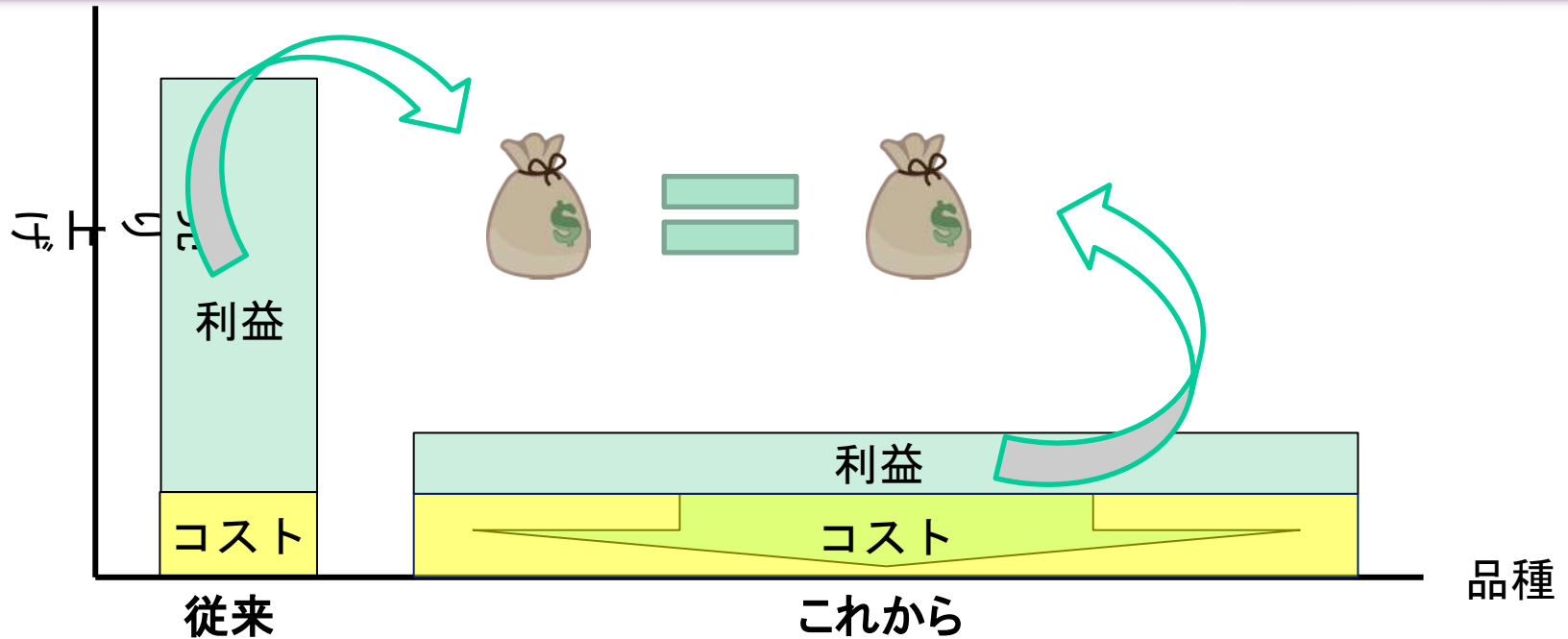


PDKに組み込まれている

Outline

- (株)アナジックスの取り組み
- オープンハードウェア開発のプラットフォーム
- ➡ ■少量生産のLSI開発のためのOSEDA
- OSEDAの普及を助けるEDAサーバ
- アナログIPの共有と再利用
 - 回路設計へのJupyter Notebook適用
- まとめ

今がターニングポイント



LSI生産が大量生産から少量・中量生産にシフトする

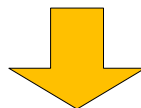
コストが下がれば、利益は出る

それを可能にするミニマルファブ

IoTデバイス開発のためのIP共有

高付加価値の
少量生産デバイス

IP共有を可能にする



Open Source EDA (OSEDA) を
使うことが重要

ミニマルEDA

なぜOSS/OSEDAが重要か

OSS = Open Source Software

OSEDA = Open Source EDA

~~ベンダーツール~~
~~無料ツール~~

オープンソース
EDA

OSSだけがIP共有を可能とする

例: レイアウトPCell (Parameterized Cell)

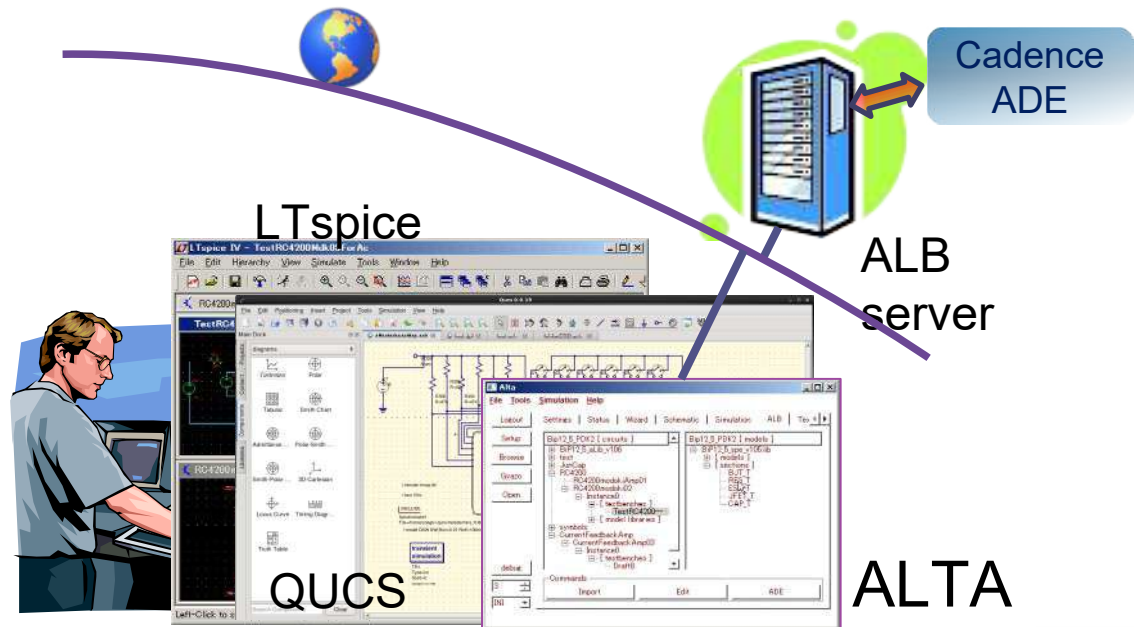
ミニマルEDA

ミニマルEDA



ミニマル
ファブ

■ 私共のALB/ALTA と様々なフリーまたはオープンソースツールから構成される



回路図エディタ比較

ミニマルEDAではALBを使って回路図の相互変換ができます

	LTspice	QUCS-S	Eeschema	Xschem
オープンソース	×	○	○	○
データ形式の情報	×	○	○	○
階層構造	○	○	×	○
使いやすさ	◎	○	△	△
日本語の情報	◎	×	○	×
JNBでの利用	○	○	○	○

回路設計環境

■推奨ツールは、オープンソースのeeschema

■無料ツールのLTspiceは、自己責任でご使用ください

▶ライセンス原文

This program is specifically not licensed for use by semiconductor manufacturers in the design, promotion, demonstration or sale of their products. Specific permission must be obtained from Linear Technology for the use of LTspice for these applications.

▶Google訳

このプログラムは、半導体メーカーが製品の設計、販売促進、デモンストレーション、または販売に使用することを特に許可されていません。これらのアプリケーションでLTspiceを使用するには、リニアテクノロジーから特定の許可を得る必要があります。

ミニマルEDAではEEschemaを推奨

■階層の扱いには制限があるが、ミニマルファブでは実用に耐える

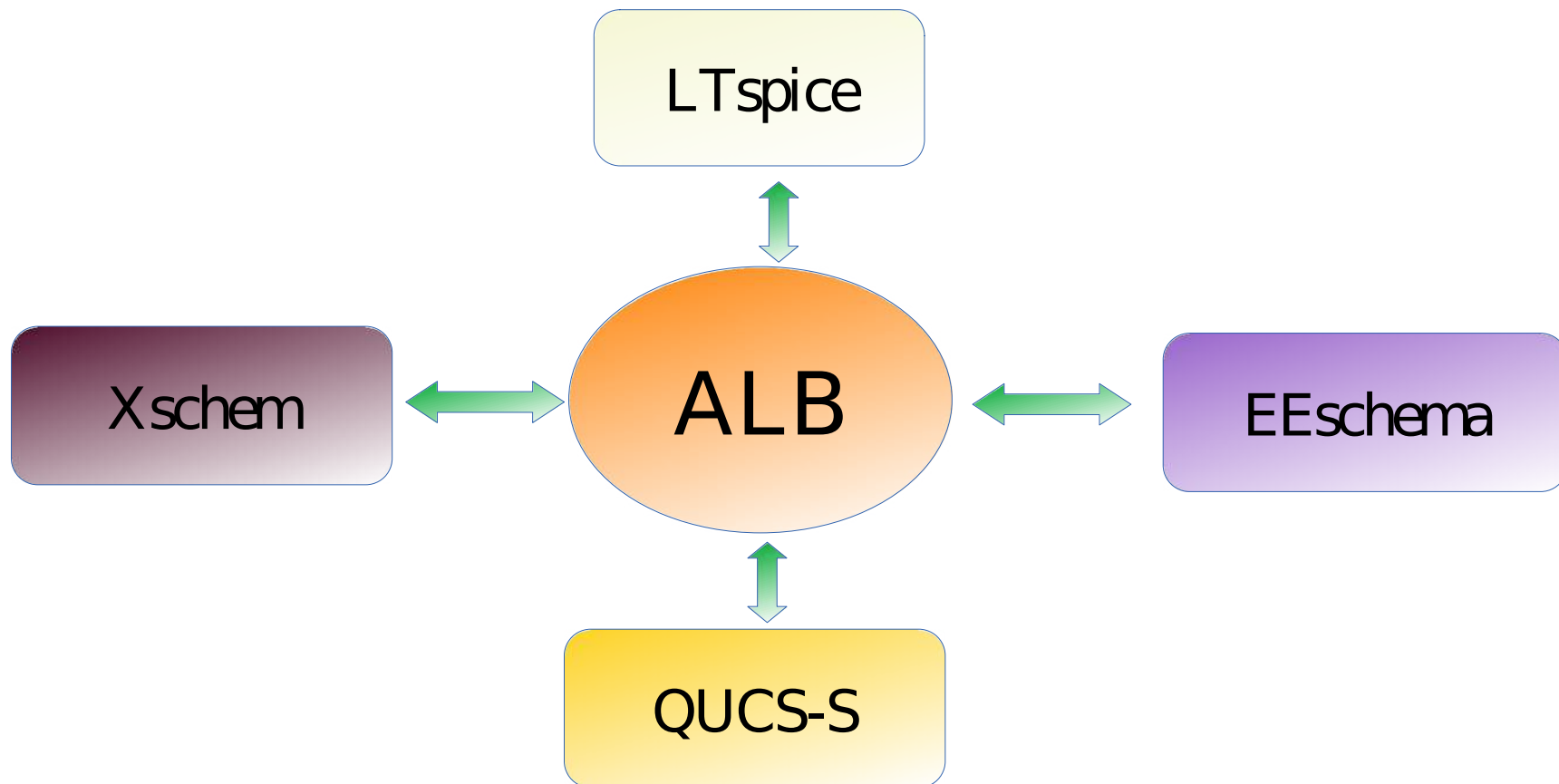
- 実現できる回路規模が大きくなり、
- 共有IPを積極的に利用する設計スタイルが主流

■利用者が多いためにコミュニティがしっかりしていて情報が多く、容易に廃れることはない

■Jupyter Notebookでの利用も可能
→ ほかに、Xschem、LTspice利用可



各ツール間の回路図データ変換



KLayoutとは

■ ドイツのMatthias Köfferleinさんが中心になって開発しているオープンソースのレイアウトCADソフト (<https://www.klayout.de>)

■ LSI設計に使えるのは当然ですが、Photonic IC分野での利用が目立つ

▶ ワシントン大(シアトル)のナノファブ

✓ https://ebeam.mff.uw.edu/ebeamweb/training/cad_tut_main/cad_tutorial_main.html

✓ MentorのほかKlayoutのサポートに力を入れているようで、チュートリアルも作成し、KlayoutのナズカシステムからPDKをインストールできる

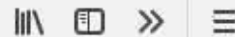
▶ 設計環境(ナズカ) **実績があり商用利用にも耐える**

✓ <https://nazca-design.org>

✓ klayoutにpythonの便利ツールを付け加えた **オープンソースのレイアウトCAD**

✓ ファブ依存のPDKは有料サポートという

ビジネスモデル



KLayout ユーザマニュアル

<https://www.klayout.de/doc-qt4/manual/index.html> を seijiro.moriyama@anagix.com が中心になって翻訳したユーザマニュアルです。Dropbox paperを使って編集をしています。編集に参加していただける方は、メールでご連絡ください。

このマニュアルは、以下のトピックで構成されています。

- +KLayoutの基本 (KLayout Basics)
- 編集機能 (Editing Functions)
 - +編集モード (Edit Mode)
 - +編集モードの基本原理 (Basic Principles Of Editor Mode)
 - +基本的な編集操作 (Basic Editing Operations)
 - +Basicライブラリについて
 - +パッケージについて
 - +PCellについて
 - +Rubyを使ったPcellコーディング
 - +高度な編集操作 (Advanced Editing Operations)
- +高度なトピック (Advanced Topics)
- デザインルールチェック (DRC) (Design Rule Check (DRC))



ミニマルEDAのアナログIP利用環境

■回路設計

- ▶ LTspice (無料・非公開) と EESchema、QUCS-S/XYCE と Xschem (無料・公開) を利用可能、EESchema を推奨
- ▶ Jupyter Notebook からの利用
 - ✓ LTspice、QUCS-S/XYCE、Xschem、EESchema

■レイアウト設計

- ▶ PCell は、オープンソースの KLayout を推奨
- ▶ ドキュメント作成は、Dropbox Paper が便利

■ミニマルEDAサーバで、スタンドアローンもしくは ALB/ALTA 環境から利用可能

Outline

- (株)アナジックスの取り組み
- オープンハードウェア開発のプラットフォーム
- 少量生産のLSI開発のためのOSEDA
- ➡ ■OSEDAの普及を助けるEDAサーバ
- アナログIPの共有と再利用
 - 回路設計へのJupyter Notebook適用
- まとめ

アカウント取得とログイン

■ ふくおかISTオープンソースEDAサーバスタートアップマニュアル参照

1. アカウントの取得 (Keycloakを利用)
2. EDAサーバにログイン
3. vncserverを使ってデスクトップを生成
4. vncviewerを使ってデスクトップにアクセス
(その際、セキュリティを確保するためsshを使ったポートフォワード(トンネリング)を行う)

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

- Groups
- Users**
- Sessions
- Events
- Import
- Export

Users

Search...		Q	View all users			Unlock users			Add user
ID	Username	Email	Last Name	First Name	Actions				
528bb67e-268a-461c-99...	Edit	Impersonate	Delete		
2f9bf2d7-45f0-4317-a96...	Edit	Impersonate	Delete		
38d05700-5a80-4c0e-ab...	Edit	Impersonate	Delete		
ea0e2f79-8404-49ac-bba...	Edit	Impersonate	Delete		
f42d2470-a708-4b5f-9ff...	Edit	Impersonate	Delete		
076d41e4-5034-4374-8a...	Edit	Impersonate	Delete		
b6b32515-3aed-461c-ad...	Edit	Impersonate	Delete		
b6f99b89-6032-449d-9d...	Edit	Impersonate	Delete		
cf2839c-9e83-44cc-92f...	Edit	Impersonate	Delete		
4be12683-16d2-4d2f-ac...	Edit	Impersonate	Delete		
dac7181a-1270-40b6-87...	Edit	Impersonate	Delete		
62afc50d-3fab-472d-add...	Edit	Impersonate	Delete		
0937dbbc-5de8-4a45-ab...	Edit	Impersonate	Delete		
ae7e9ca5-121d-401f-b4c...	Edit	Impersonate	Delete		
67e40ba0-aa42-4433-b2...	Edit	Impersonate	Delete		
0ee15c75-f6c6-4c71-92e...	Edit	Impersonate	Delete		
0694e981-d5b3-43de-9b...	Edit	Impersonate	Delete		
c3c5681c-7458-4568-84f...	Edit	Impersonate	Delete		

VNC on sshによるセキュアな通信

1. sshでログインしてvncserverをスタート
→ ポート番号が割り当てられる
2. sshでログインする際に、EDAサーバのポート番号にセキュアにアクセスするトンネルを掘る
3. トンネルを使って、vncviewerをEDAサーバのポートの接続する
4. 次ページのようなデスクトップを使える

Openboxとtint2による軽量デスクトップ

The screenshot displays a desktop environment with the following components:

- Terminal emulator:** Shows system statistics:

```
average: 0.41, 0.21, 0.10
topped, 8 zombie
wa, 0.0 hi, 0.0 si, 0.0 st
used, 480136.8 buff/cache
used. 506554.3 avail Mem
```

%CPU	%MEM	TIME+	COMMAND
6.6	0.1	0:21.38	Web Con+
2.0	0.0	2:54.05	Xtigerv+
2.0	0.1	354:57.38	firefox
- Web browser:** Displays a page titled "ふくおかIST オープンソース EDAサーバ スタートアップマニュアル" (Fukuoka IST Open Source EDA Server Start-up Manual) on a Dropbox link.
- Storage usage graph:** A line graph showing storage usage over time, with a tooltip indicating "day: 12 Storage usage: 340GB".
- System menu:** A dropdown menu is open, listing applications such as "Terminal emulator", "Web browser", "Applications", "File Manager", "Start Panel (tint2)", "tint2 Configuration", "Openbox Appearance", "EDA tools", "Wallpaper", "Screen capture (flameshot)", "ObConf", "Reconfigure", "Restart", and "Exit".

日本語入力、スクリーンショットなど

■日本語入力

➤ fcitx + mozcを使用

■スクリーンショット

➤ flameshotを使用

■これらは、EDAサーバ上で得られた情報(バグなど)をコミュニケーションするために不可欠である

インストール済みのもの

- Qflow --- デジタル設計フロー
- Kicad --- プリント基板設計ソフト
- Glade --- オープンソースでないレイアウトエディタ
- Eeschema --- Kicad付属の回路図エディタ
- ALTA --- IP設計環境、回路図変換
- KLayout --- レイアウトエディタ
- Jupyter Notebook --- 設計・ドキュメント環境

オープンソースEDA普及のための方策

- EDAサーバは、オープンソースに触れるきっかけを与えるもの
- オープンEDA利用者のコミュニティを作るための手段
- EDAサーバの利用率向上は目標ではない
- コミュニティの活性化が目標である
- ハードウェア設計のニーズがなければツールは不要
- 方策
 1. コミュニティSNSの立ち上げ
 2. オープンハードウェア設計の場をつくる
 3. オープンソースを使ったLSI設計教育

方策1:コミュニティSNSの立ち上げ

■Facebookの運用

- ▶ イベントなどの案内
- ▶ Discordへの誘導

■Discordサーバの活用

- ▶ テーマごとにチャンネルをわけると
 - ✓ ツールごと
 - ✓ はじめての利用者向けなど
- ▶ 技術情報の交換
- ▶ リリースのアナウンス
- ▶ バグ情報など

OpenSource EDA supporter (Discord)

The screenshot shows the Discord interface for a server named "Open Source EDA supporter". The server is set to "公開" (Public) and has a "非公開" (Private) button. The server icon is a blue circle with "OSEs" and a red "LSI:" tag. The server description reads: "さあ、冒険へ出よう 友達を追加しよう！" (Let's go on an adventure, let's add friends!).

The server has several text channels: # 一般 (General), # サーバーアップデート..., # xschem, # eeschema, # qucs-s, # openlane, # klayout, and # 質問があればこちらへ... (If you have questions, please go here...). There are also voice channels, including # 一般.

The main chat area shows a welcome message: "Open Source EDA supporter へようこそ" (Welcome to Open Source EDA supporter) and "ここが、このサーバーの始まりです。" (This is the beginning of this server.). Below this, a list of recent activity is shown:

- Junichi Akitaがパーティーに加わりました。 2020/09/10
- chikauがサーバーに飛び乗りました。 2020/09/10
- LR)uraがやってきました。 2020/09/10
- LR_Wakasugiが出たぞー！ 2020/09/10
- やあ、Yamada3君。ピザ持ってきたよね？ 2020/09/10
- Ryo Okawaがパーティーに加わりました。 2020/09/10

A date separator indicates "2020年10月1日" (October 1, 2020), followed by:

- shunsuke.mochizukiがサーバーに滑り込みました。 2020/10/01

Another date separator indicates "2020年11月23日" (November 23, 2020), followed by:

- Yasudaがサーバーに飛び乗りました。 2020/11/23

The right sidebar shows the online status of members: オンライン-5 (Online - 5) and オフライン-4 (Offline - 4). The online members listed are chikau, Junichi Akita, LR)ura, LR_Wakasugi, and seijiro.moriyama (with a crown icon). The offline members listed are Ryo Okawa, shunsuke.mochizuki, Yamada3, and Yasuda.

The bottom of the interface shows the user profile for seijiro.moriyama (#7033) and a text input field for sending a message to the # 一般 channel. There are also icons for sending gifts, GIFs, screenshots, and emojis.

方策2: オープンハードウェア設計の場をつくる

- Skywater 130nmファブに相当する国内ファブ育成
- NDA不要のPDKの開発
- OpenLaneに相当するデジタル設計フロー
- Xschem/EEschema+NGspice/XYCEのアナログ設計環境
- シャトルのサポート

方策3: オープンソースを使ったLSI設計教育案

■ OpenLane/Qflowを使ったデジタル設計

■ ミニマルEDAを使ったアナログ設計

- Jupyter Notebookを使った回路設計
- KLayout/PCellを使ったレイアウト設計

■ ミニマルファブを使ったリモート試作

■ LSIの基礎教育

- 製造プロセスの知識
- デバイスの構造
- 回路設計理論と実践

Outline

- (株)アナジックスの取り組み
- オープンハードウェア開発のプラットフォーム
- 少量生産のLSI開発のためのOSEDA
- OSEDAの普及を助けるEDAサーバ
- ➡ ■アナログIPの共有と再利用
 - 回路設計へのJupyter Notebook適用
- まとめ

IP共有・再利用の課題

- アナログIPライブラリは事実上存在しない
- IPを開発できる人がほとんどいない
- IP開発は非常にめんどろ
- EDAツールの選定が悩ましい

設計者がアナログIPライブラリ開発に集中できる環境を整備する必要がある

PMOS向けに開発中の基本ブロック

名称	試作	評価	JNB対応
マルチリングオシレータ	2020/2	X	予定
A級アンプ	2020/2	O	予定
ダイナミックシフトレジスタ	2020/12	未	予定
差動増幅器	2020/12	未	開発中
スイッチングコンパレータ	2020/12	未	予定
IOバッファ	2020/12	未	未定

PMOS差動増幅器の設計

1. 仕様検討
2. 10倍前後の差動利得を持つ差動段のラフ設計
 - ・ 差動1段のGm見積もり
 - ・ テール電流源の設計
3. 差動1段の動作確認
 - ・ 差動利得の同相電圧依存性
4. 2段構成の動作検証
 - ・ Irefのばらつきに対する同相出力変動
 - ・ 同相負帰還の検討
 - ・ Vth相対バラツキ・絶対バラツキの影響
5. 電流源の寄生容量による同相発振の回避

Jupyter Notebookを使ったIP設計例

Jupyter Notebook interface showing a notebook titled "essential_pmos_diff_amp". The notebook content includes a title and a list of bullet points:

【10倍前後の差動利得を持つPMOS差動段のラフ設計】

- gmがあまり変わらないので、消費電流は少ない方が喜ばれるから、常識的に0.1mAとしておこう。つまり、設計の出発値として差動のテール電流を0.2mAにする。
- $V_{in1}=V_{in2}=0V$ のとき、出力の同相電圧 V_{ocm} をどうするかだが、 $V_{th0}=-0.9V$ だったから、共通ソースの電圧は約0.9Vになる。従って、仮に $V_{ocm}=-3V$ に設定すれば $|V_{ds}|=|V_{ocm}-V_{th0}|=3.9V$ となる。このとき負荷抵抗 R_L には $|V_{SS}-V_{ocm}|=2V$ の電圧降下が生ずる。差動対1個あたり $I_d=0.1mA$ だから、 $R_L=2V/0.1mA=20k\Omega$ とすればつじつまが合う。さて、この時の差動利得は $|A_v|=|g_m| \cdot R_L=0.3mS \cdot 20k\Omega=6$ 倍と計算される。
- 再調整：利得が少ないので、負荷抵抗を2倍にして再計算すると、 $|A_v|=12$ 倍、 $V_{ocm}=-1V$ 、 $|V_{ds}|=4.9V$ となり、まずまずである。このパラメータでシミュレーションした結果は、差動利得が $|A_v|=9.7$ 倍でほぼ設計の通り。ひとまず、これで次へ進める。

差動対検討用の回路図

```
;.op
.dc V3 -5 5
.lib ../parameter2/mineda2020_1_pmos.txt
```

In [36]: dtckt.set R1: '20k', R2: '20k'

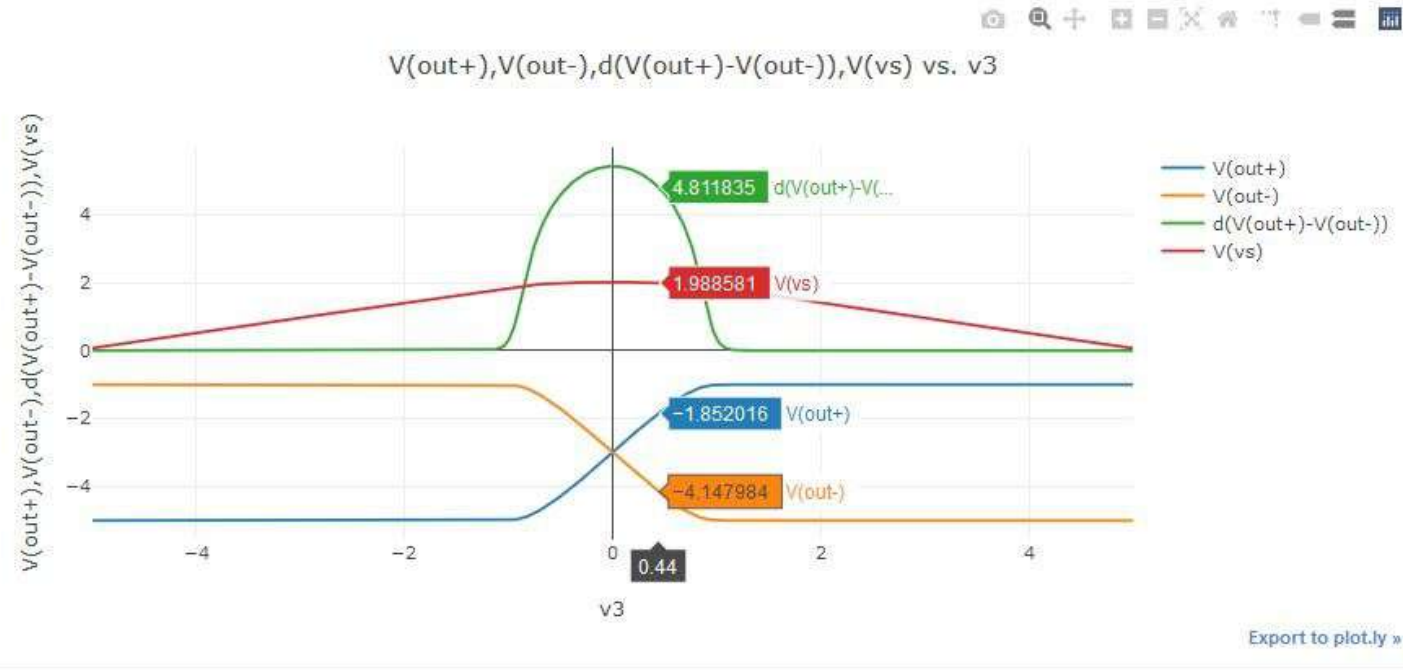
Out [36]: [true, true]

In [37]: dtckt.simulate

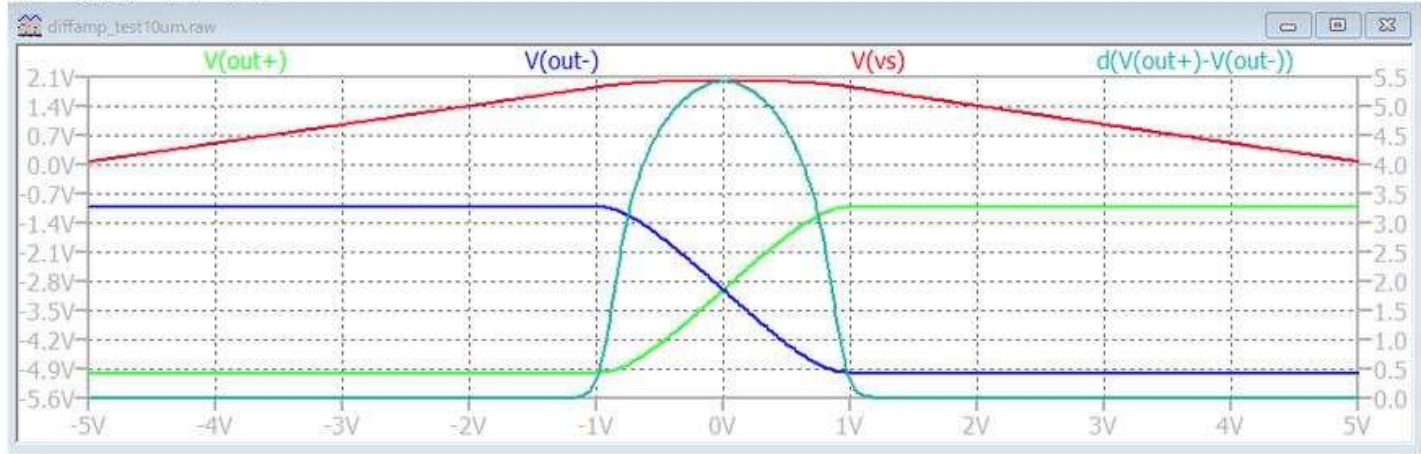
Jupyter Notebookから開いた回路図を張り付けた

```
(M2)", "Ig(M2)", "Ib(M2)", "Is(M2)", "I(I1)", "I(R4)", "I(R3)", "I(R2)", "I(R1)", "I(V2)", "I(V3)", "I(V1)"]
```

```
In [42]: dtckt-plot 'v3', 'V(out+)', 'V(out-)', 'd(V(out+)-V(out-))', 'V(vs)'
```



LTspiceのスクリーンショット

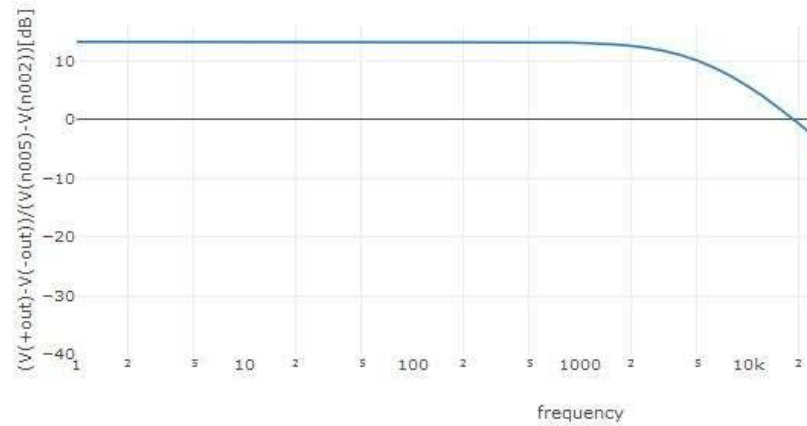


差動利得の同相電圧依存性

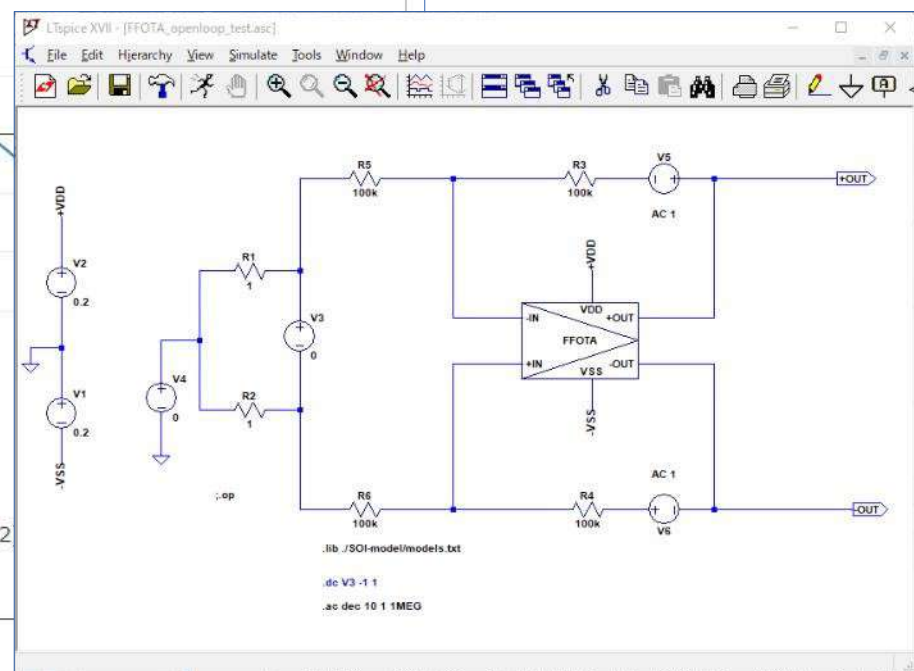
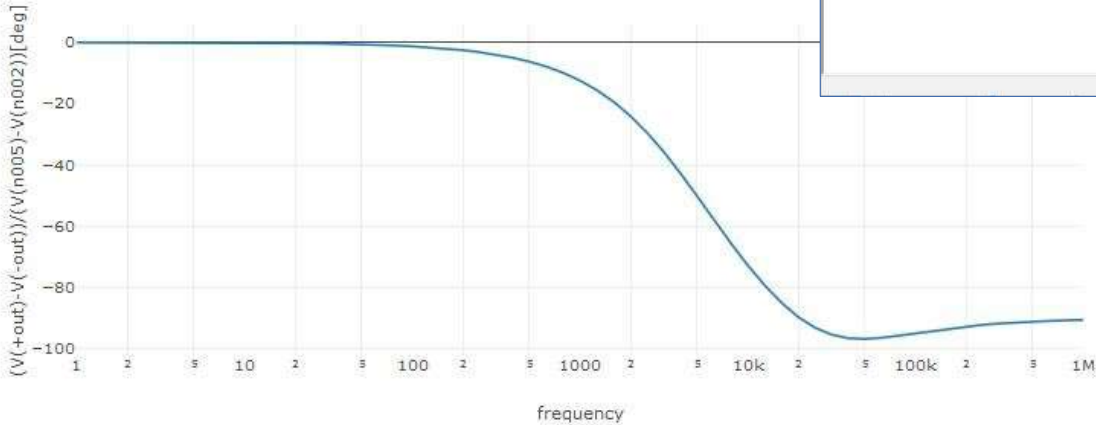
```
In [16]: execute_simulation()
C:\Program Files\LTspice\bin\ltspice.exe -netlist FFOTA_openloop_test.tsp
W:\U\Users\masaji\on\work\FFOTA
C:\Program Files\LTspice\bin\ltspice.exe -b -Run FFOTA_openloop_test.net
execute_sim_log() to show simulation log
```

```
In [17]: ffota_plot 'frequency', '(V(+out)-V(-out))/(V(n005)-V(n002))'
```

$$20\log_{10}((V(+out)-V(-out))/(V(n005)-V(n002)))$$



$$\text{phase of } (V(+out)-V(-out))/(V(n005)-V(n002))$$



Export to plotly »



```
In [18]: xdata = []
ydata = []
-l. step(1,0.2){|v4|
puts "v4=#{v4}"
xdata << v4
ffota.set V4: v4.round(4)

ffota.simulate

ffota.get_traces 'frequency', '(V(+out)-V(-out))/(V(n005)-V(n002))'
ydata << yval=ffota.y_db[0]
puts "=>#{yval}"
}
```

同相電圧をスイープするスクリプト

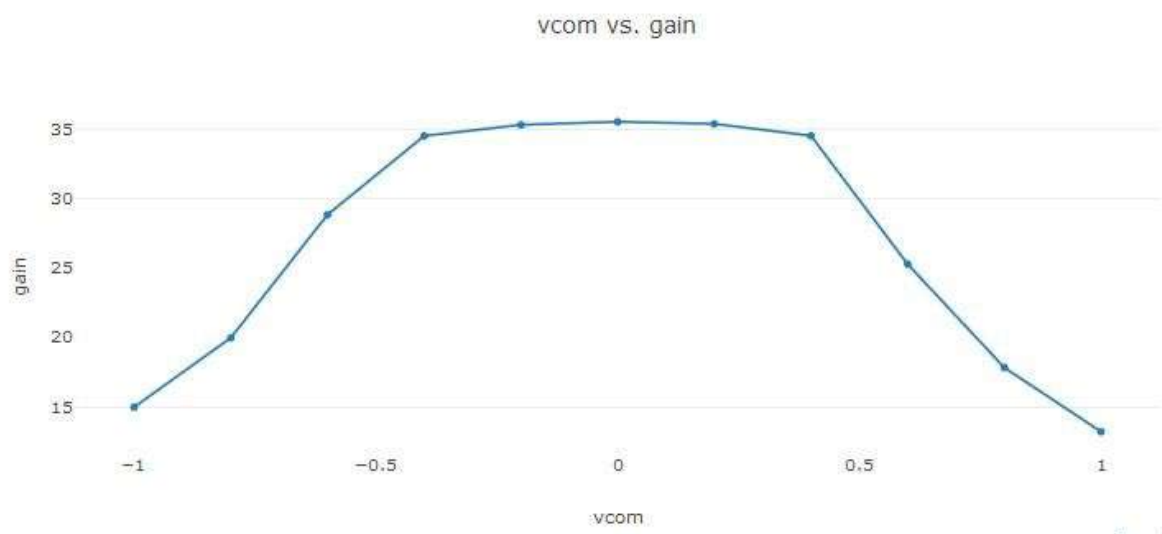
```
In [19]: xdata
```

```
Out[19]: [-1.0, -0.8, -0.6, -0.3999999999999999, -0.19999999999999996, 0.0, 0.20000000000000018, 0.40000000000000013, 0.6000000000000001, 0.8, 1.0]
```

```
In [20]: ydata
```

```
Out[20]: [14.986955042744416, 19.991967379709067, 28.878891637129847, 34.56920403957718, 35.37042170483485, 35.589963017108445, 35.440896651091606, 34.582210295161595, 25.311206633843955, 17.844054650521212, 13.213564365977495]
```

```
In [21]: ffota.plot(x: xdata, y: ydata, layout: {title: 'vcom vs. gain',
xaxis: {title: 'vcom'}, yaxis: {title: 'gain'}})
```



JupyterNotebookを使ったIP整備

- 回路設計データと設計ドキュメント(設計情報)をまとめて作る
- 設計情報を利用するのが容易
- 設計情報に新たに情報を追加するのが容易(自己増殖する)
- オープンソースEDA(EEschema、Xschemなど)を使う方向(無料のLTspiceも利用可能)
- GitHubで公開できる

JupyterNotebookを使うメリット

利用者

設計検討の意図を
理解しやすい
その場で実行
結果を確認できる
回路情報をそのまま
設計に利用できる

作成者

設計作業の記録になる
説明を簡単に書き加える
ことができる
設計資料としてそのまま
配布できる

自習教材 (Jupyter Notebook) 準備中

essential_pmos_diff_amp

localhost:8888/notebo 110%

jupyter essential_pmos_diff_amp (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Ruby 2.5.3

差動アンプ設計の真髄

```
In [1]: load File.join(ENV["HOME"], "/work/alb2/lib/altactl...)
```

```
In [34]: dtckt=LTspiceControl.new
```

```
In [43]: dtckt.open "diffamp_test10um.asc"
```

差動アンプの仕様検討 1

- ・利得 100前後: MEMSのアンプが20倍の利得だったこと, 100倍あれば閉ループ時の利得設定誤差が1/100になる. 良くはないがひどくもない. なお, 一般的なオペアンプならば少なくとも60dB以上が必要である.
- ・回路形式 差動入力・差動出力とし, 出力の同相電圧安定化のために同相帰還を内蔵させる. たぶん抵抗負荷の差動1段では利得が10倍前後であろうから, 回路構成としては「差動2段+ソースフォロワ出力段」となるだろう. さしたる根拠もないが, 負荷は電圧計等の計測器を想定して, 抵抗性で10kΩ以上で, 負荷容量は50pF以下としよう.

Coming Soon!

Outline

- (株)アナジックスの取り組み
- オープンハードウェア開発のプラットフォーム
- 少量生産のLSI開発のためのOSEDА
- OSEDАの普及を助けるEDAサーバ
- アナログIPの共有と再利用
 - 回路設計へのJupyter Notebook適用

➡ ■まとめ

まとめ

- (株)アナジックスは、ローコストの設計環境とお客様のためのLSI設計教育を提供
- 少量生産のLSI開発のためには、設計資産(IP)の共有・再利用とOSEDAの利用が必須
- OSEDAの普及を助けるため、EDAサーバを立ち上げオープンハードウェアの開発を支援し、コミュニティ構築を目指す
 - OpenRule1umやミニマルファブを使ったカスタムLSI設計のできる環境(PDK)を整備
- IPの共有・再利用容易化のため、設計者がアナログIP開発に集中できる環境を整備
 - Jupyter Notebookを用いた回路設計知識の共有に取り組む

今後の予定

■Skywater PDKを使った設計環境

- OpenLaneによるデジタル設計
- Xschem + NGspice (XYCE)によるアナログ・カスタム設計

■ミニマルEDAの環境整備

- Eeschema + NGspiceによるアナログ・カスタム設計

■Chefによるユーザ側サーバの立ち上げ支援

■ユーザサポート

■オンラインコミュニティの立ち上げとサポート

■ユーザ要求にもとづくツール増強

■ツールのアップデート