

オープンソースEDAによるLSI開発（LSI事業の民主化を目指して）

1. ロジックリサーチの紹介
 2. オープンソースEDAを使ったきっかけ
EDA遍歴、EDAツールの稼働率とコスト分析
 3. ロジックリサーチでのオープンソース活用
オープンソースEDAと商用EDAの比較と使用実績
Klayoutとxschemの画面の紹介
ミニマルTEGの開発での実績(LayoutとLVS)
Qflowの概要紹介
ロジックIC開発（RISC-Vチップの例、OpenRAMの例）
アナログIC設計例（アンプ、コンパレータ、DAC）⇒コンパレータとDACは、作成中
 4. 今後の計画
- 参考情報
- ミニマルファブの紹介
 - ロジックリサーチの研究情報
 - ☆再配線によるLSI開発（研究成果）
 - ☆マスター回路を使ったLSI開発案



株式会社ロジック・リサーチ

Think Solution for the Customer's Success!

会社概要

代表取締役社長：土屋 忠明

資本金：4578万円

設立：1992年12月

事業内容：LSIの開発・製造・販売

本社：福岡市早良区百道浜3丁目8番33号
福岡システムLSI総合開発センター

電話番号：092-834-8441

FAX番号092-834-8442

E-mail：info@logic-research.co.jp

WebPage: <http://www.logic-research.co.jp>

[加盟団体]

九州半導体・エレクトロニクスイノベーション協議会(<http://www.siiq.jp>)

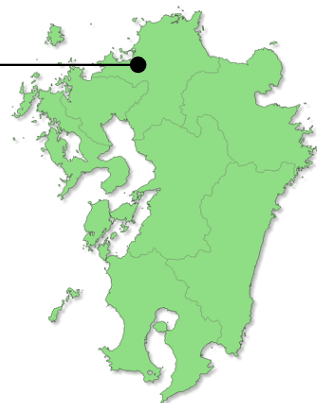
TOPPERS プロジェクト(<http://www.toppers.jp/index.html>)

ミニマルファブ推進機構(<http://www.minimalfab.com/>)

日本電子デバイス産業協会(<http://www.nedia.or.jp/>)



福岡システムLSI総合開発セ



沿革

1992/3 半導体メーカーからの受託設計(通信用GA)を開始

1993/4 受託設計を開始

1995/3 LSI受託設計へ移行、通信機器メーカーからLSI受託設計

1998/4 経済産業省の補助金により再構成可能LSIの研究開始

2000/5 半導体ファブレスメーカーのビジネスモデル開始

2001/3 自社ブランド製品量産開始(大手メーカーに採用)

2005/4 カスタムIC開発が20品種超

2006/12 ファブレスASIC事業に移行

2007/9 0.18umプロセス製品を初導入

2007/12 H8組込みASICの開発完了

2007/12 Flash組込みASICの開発着手

2010/6 ファブシステム研究会でミニマルファブの研究開発スタート

2011/1 SH4組込ASIC開発完了

2012/4 ミニマルファブ技術研究組合に参加

2012/8 ルネサスエレクトロニクスとCMOS GAのライセンス契約締結

2013/4 HD-PLCアライアンス加盟

2014/9 EOLマイコン(三菱電機製)の量産出荷スタート

2014/12 EOLマイコン(H8)の量産出荷スタート

2015/4 EOL Gate Array(NEC-9HD)の量産出荷スタート

2015/9 ISO9001/14001の取得

2016/12 EOLマイコンの量産出荷スタート

2017/10 EOL(マイコン、GA)開発スタート

2018/1 EOLマイコン(M32R)開発スタート

2018/10 ミニマルファブによるEOL対策研究開発

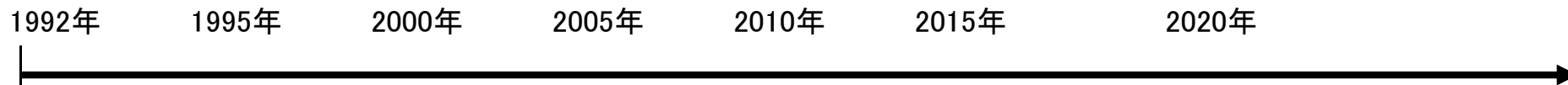
2019/11 EOLマイコン(H8, M32R)ES出荷



一般社団法人 日本電子デバイス産業協会



事業ロードマップ(創業から、現在、将来)



通信用GAの受託設計

- ☆E1/J1通信処理GA(回路図) ⇒FJオリジナルEDA
- ☆ATM通信処理GA(HDL設計) ⇒ModelSim
- ☆SDH通信処理GA(HDL) ⇒ModelSim
- ☆FEC(誤り訂正)ASIC ⇒ModelSim, Design Compiler
- ☆VDSLモデムSOC ⇒ModelSim, VCS, Design Compiler、SpyGlass
- ☆iBurst通信用SOC検証 ⇒ModelSim, Matlab、Scilab、SpyGlass, Design Compiler

画像処理ASICの受託設計

- ☆LCD/CCDインタフェース ⇒ModelSim, VCS, Design Compiler、RTQ
- ☆FOMA用ASSPの部分設計 ⇒ModelSim, VCS, Design Compiler、RTQ
- ☆MPEG4エンジン設計 ⇒ModelSim, VCS, Design Compiler、RTQ
- ☆画像補正回路設計 ⇒ModelSim, VCS, Design Compiler、RTQ
- ☆カーナビASICの設計&検証 ⇒VCS, NC-Verilog, Design Compiler、RTQ
- ☆JPEG2000コア開発 ⇒ModelSim, Design Compiler、Astro、Calibre

カスタムLSI

- ☆FAX用パネル制御ASIC(累計出荷3900万個)
- ☆FAX用メカ制御ASIC(累計出荷150万個)
- ☆GBルーター用FPGA置き換え(累計出荷4万個)
- ☆FAXモデム用ASIC(累計出荷300万個)
- ☆H8組込みASIC(量産準備)
- ☆ODD用DAC(累計出荷250万個)
- ☆インクジェットドライバIC(累計出荷30万個)

半導体EOLビジネス

- ☆H8, SH4混載ASICによるEOL対応
- ☆ルネサスエレクトロニクス CMOS GAのライセンス取得

ASIC.com ビジネス

- ☆ minimal Fab による 少量多品種・短TAT生産
- ☆ 究極の1個流し生産
- ☆ 納期: 1 week 価格: FPGA相当



カスタムLSI開発&販売事業

例)

- ・FAX用制御IC
- ・Panel制御、5V, 3k Gate

EOL製品の再開発

旧い製品(GA, 汎用品)で生産中止予定品を再開発します。

セカンドソース

コストダウンが難しいLSIを独自手法にて再開発します。

例)

- ・8ビット、16ビットマイコン
- ・0.35um CMOS GateArray

大手LSIメーカーが苦手な分野をお引き受けします。

・ハイブリッドICの集積

FPGAの置換

FPGA/PLDを置き換えてコストダウンします。

限量生産品

市場規模が小さな分野でも開発します。先端市場向けや研究開発に最適です。

例) 通信処理回路

- ・300k Gate, 125MHz

Ex. 産業用InkJet Driver

130出力、30V耐圧

Ex. 研究開発用 Display Driver

160出力、60V耐圧



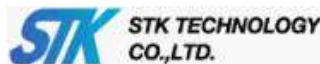
事業概要

➤ Wafer Fab



- ・IP調達（IPカスタマイズ）、MP(量産/試作)
- ・MPW(シャトル便による試作)

➤ Package & Test



MP(量産/試作)、少量試作・開発、カスタムリードフレーム開発
Multi-Chip-PKG

➤ Design Partner

株式会社シスウェーブ（アナログ、マイコン）
メイビスデザイン株式会社（メモリ、マイコン）
九州電子株式会社（アナログ）
アレックス電工株式会社（アナログ）

豊富な設計開発経験

- ・通信系LSI・FPGA→ASIC化
- ・画像処理LSI・CPU
- ・ドライバー系IC(LCD, InkJet, Motor)
- ・アナログフロントエンド etc. 70種以上

➤ Research & Development



大学(研究室)&公設研究機関

➤ Partner



OPTROMの調達
IP開発（IPのカスタマイズ）
Gate Array Solution
Design Partner



CPU IPの供給
EOL製品の移管（協議中）



少量半導体製品事業



高機能アナログIP開発

最適な解を組合せ、
お客様へ「価値」を創造

- ・最適設計（必要最小限の機能）
- ・最適なプロセス（製造し易さの追求）
- ・多品種・少量の用途に柔軟に対応
- ・継続的な生産・供給

お客様製品



ASIC事業

ロングテール ASICを実現！



研究開発(1995年～2015年)

中小企業の利点を生かし、大学、公設機関と
公的助成を活用して研究開発を実施

特定用途向けCPU(1500万円)

発光道路鎮向け低消費電力CPU
九大) 村上教授と共同研究

レチクルフリー露光技術(2000万円)

LCDパネルを使った再利用可能マスク
熊大と共同研究、
熊本地域結集型共同研究開発

RTOSのLSI化(1500万円)

リアルタイムOSの応答性を向上させる
九大) 村上教授と共同研究

低コストASIC製品(LRCT50039)

ドライバIC(LRCM50019)

汎用品のカスタム化(LRCM50002)

再構成可能LSI(1.5億円)

ビットスライスCPUをアレイ状に配置
九大と共同開発、特許取得
事業化資金不足で開発断念

CMOS無線LSI

半導体に整合回路実現

九大と共同研究、特許取得

H8組み込みASIC(LRCS18033)

SH4組み込みASIC(LRCS18050)

Digital制御アナログIC(LRCT25048)

薄型IC開発(1500万円)

バイパスコンデンサをPKGに内蔵したIC
福岡県工業技術センターと共同研究

組込RubyCPU(1500万円)

組込RubyとRTOSによる開発期間短縮マイコン
九州工業大学、名古屋大学と共同研究

電流センサー開発(2500万円)

スマートメーター向け電流センサーと制御ICの開発、
イーテック(株) & 九州工業大学と共同開発

MIMO装置向けLSI開発(2500万円)

スモールセル向けWifi基幹装置向けモジュール開発
PicoCera(株) & 九大と共同開発

ミニマルファブ用EDA(1500万円)

ハーフインチWaferを用いた
少量生産向けLSI
産業技術総合研究所と共同

1995年

2000年

2005年

2010年

2015年



研究開発(2016年～)

ミニマルファブ用EDA(1500万円)
ハーフィンチWaferを用いた
少量生産向けLSI
産業技術総合研究所と共同

新材料を使った半導体デバイス
ミニマルファブを活用し、Si以外の
材料を使ったデバイス開発

プロセッサ開発
新材料と非ノイマンアーキテクチャに
よるプロセッサの研究

ミニマルハイブリッドプロセス(75百万円)
配線工程のみミニマルファブ化する
産総研九州と共同研究

ミニマルEDAクラウド
Anagix殿と共同でオープンソースを
活用したクラウド型EDAツールを整備

ミニマル生産管理
ミニマル装置間での製造データ
流通システムの開発

少量生産半導体デバイス事業

20016年

2020年

2025年

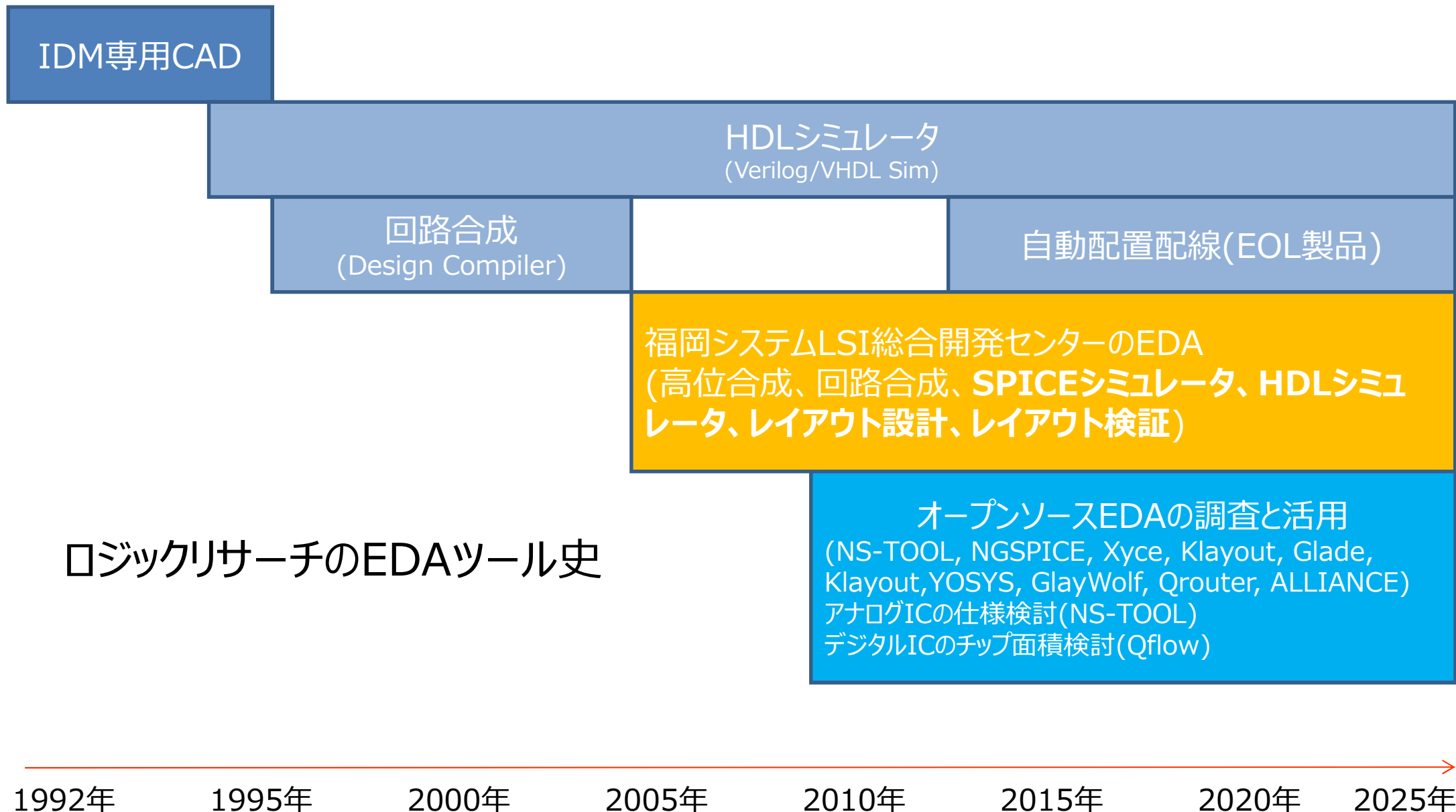
2030年

2035年

2040年



2. オープンソースEDAを使ったきっかけ（開発費用におけるEDAの割合）



2. オープンソースEDAを使ったきっかけ（開発費用におけるEDAの割合）

EDA価格(Time Base Lisence)

HDLシミュレータ(100万円/年間)

回路合成(1000万円/年間)

自動配置配線(1000万円/年間)

SPICEシミュレータ(100万円/年間)

レイアウト設計(500万円/年間)

レイアウト検証(500万円/年間)

合計費用=3700万円



1日の費用=3700÷225日⇒16万円

EDA稼働(5品種開発)

HDLシミュレータ(100日・人/年間)

回路合成(25日・人/年間)

自動配置配線(50日・人/年間)

SPICEシミュレータ(100日・人/年間)

レイアウト設計(100日・人/年間)

レイアウト検証(50日・人/年間)



1品種のEDA費用=3700÷5⇒740万円

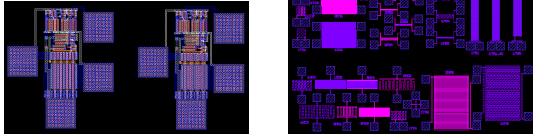
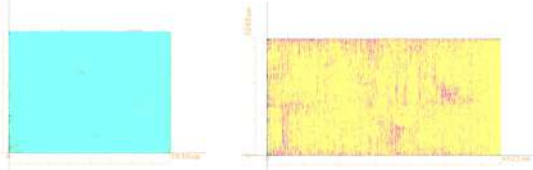
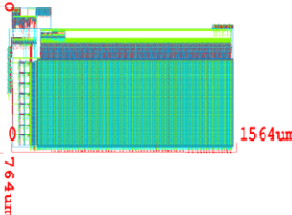
大企業では、設計者が多く稼働が高い
小規模企業では、EDAツールの稼働は、低い



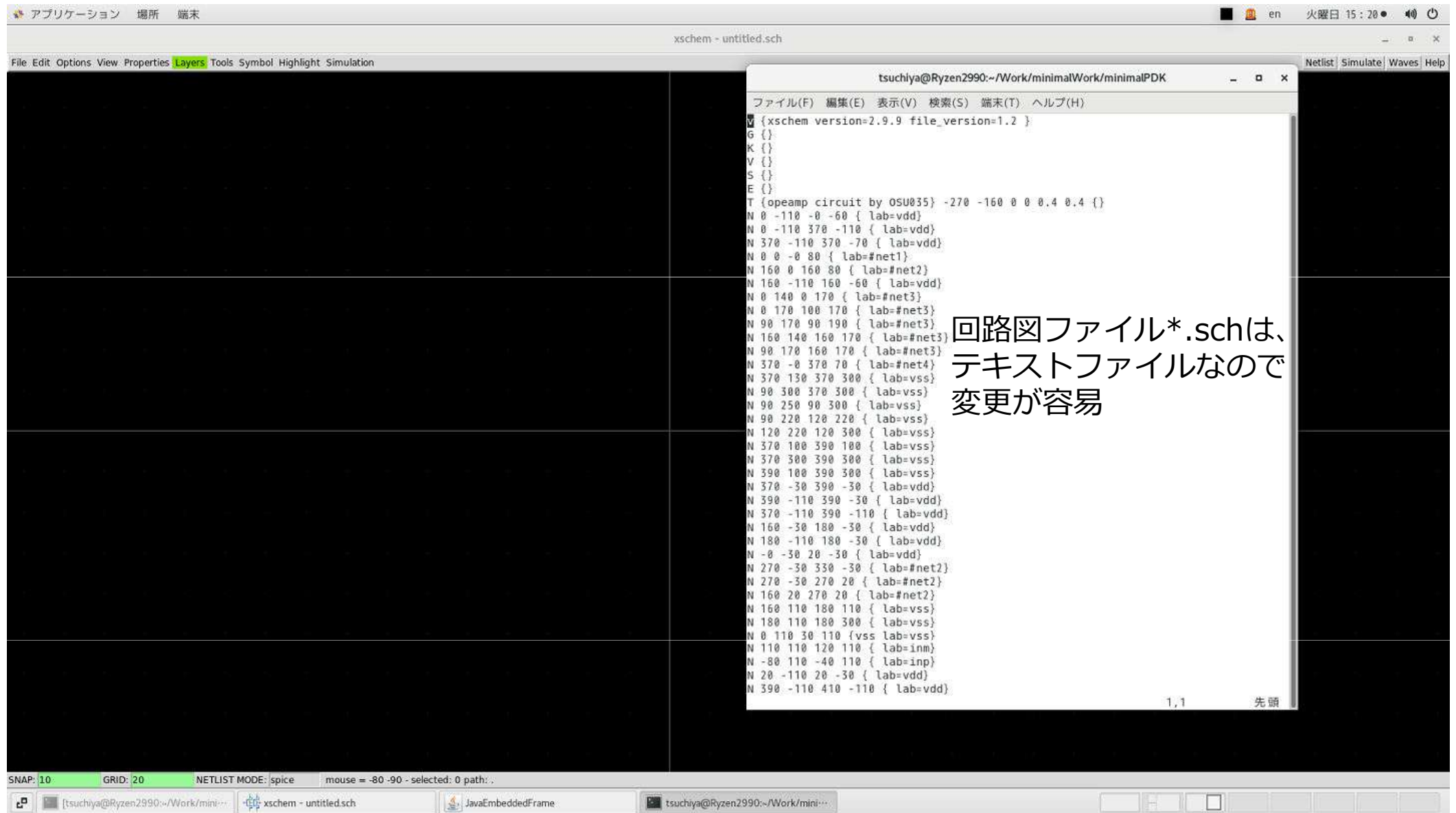
EDA費用削減が必要

3. ロジックリサーチでのオープンソース活用事例

オープンソースEDA一覧

開発ターゲット	開発工程	商用EDA	OSS-EDA	活用事例の一部
アナログIC開発	回路図エントリ	Virtuso/ISMO	Glade(回路図、レイアウト、レイアウト検証)、Klayout(レイアウト)、NGSPICE(シミュレーション)、Xyce(シミュレータ)、Xschem(回路図エントリ)	ミニマルファブ用TEG設計
	回路検証	HSPICE, Specture		IO Cell
	レイアウト設計	Virtuso/ISMO, Laker		配線確認TEG
	レイアウト検証	Calibre		
デジタル回路設計	RTL設計	EDITOR	NotePad++, Atom	RISC-Vチップの合成、配置、配線、STA
	回路検証	HDLシミュレータ	Icarus	0.5um StdCell
	回路合成	Design Compiler, Genus	Qflow(Verilog、自動配置配線、STA)	0.35um Std Cell
	レイアウト設計	Encounter, IC Compiler	Alliance(VHDL)	
	レイアウト検証	Calibre	Glade、Klayout	
SOC開発	高位合成	Catapult	調査中	OpenRAMによる SRAM実験
	CELL開発	特になし	Openlane, OpenROAD OpenRAM	
	IP開発			
	システム設計	Co-SIM環境	調査中	
	システム検証			
	レイアウト検証	Calibre	Glade、Klayout	

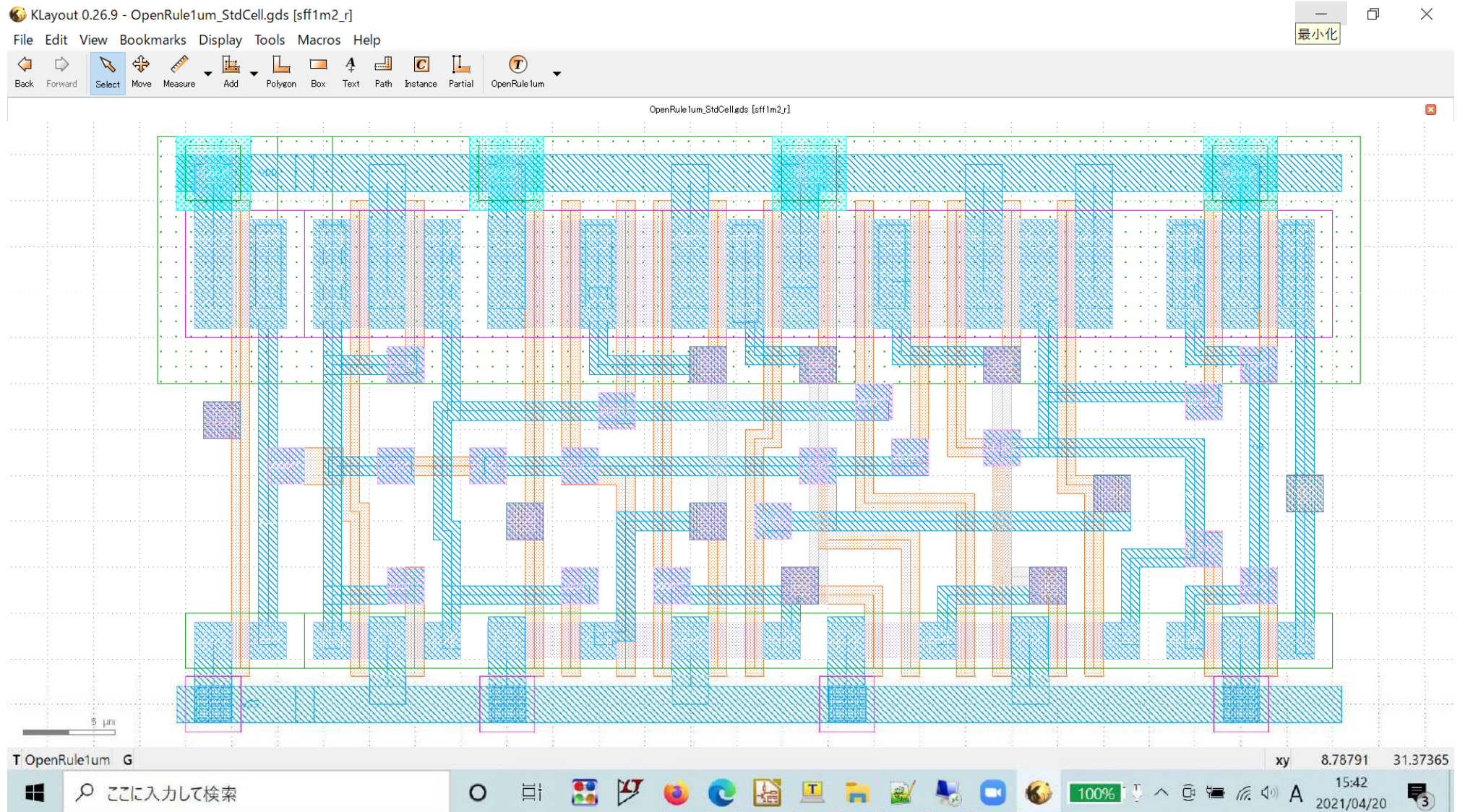
Xschemの画面と特徴



<https://xschem.sourceforge.io/stefan/index.html>

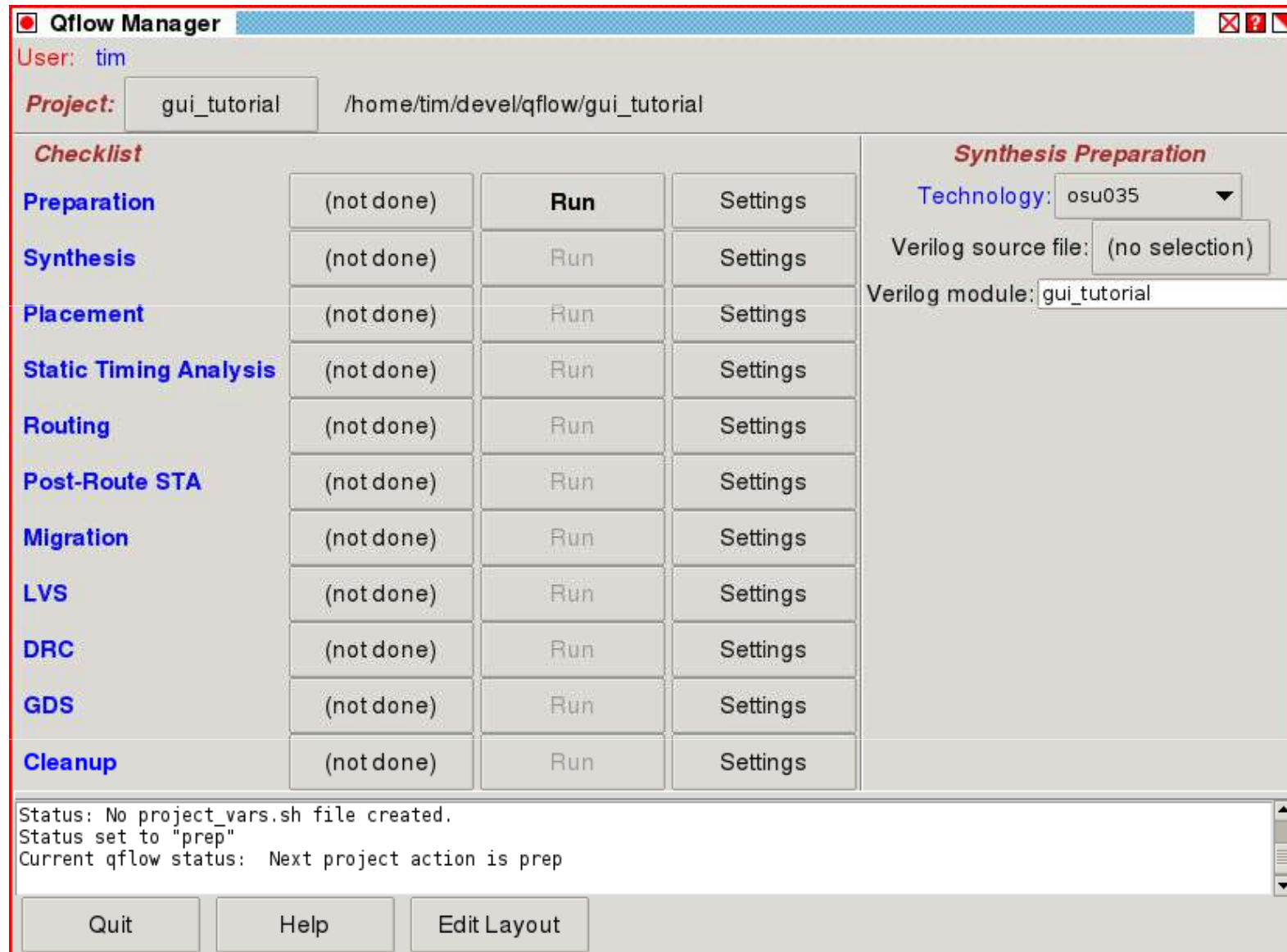


Klayoutの画面(Windows版)



<https://www.klayout.de/>





<http://opencircuitdesign.com/qflow/>

OpenRAM



OpenRAM

About OpenRAM

The OpenRAM project aims to provide a free, open-source memory compiler development framework for Random-Access Memories (RAMs). It is a joint development project between University of California Santa Cruz and Oklahoma State University to enable memory and computer system research by creating an open-source compiler infrastructure.

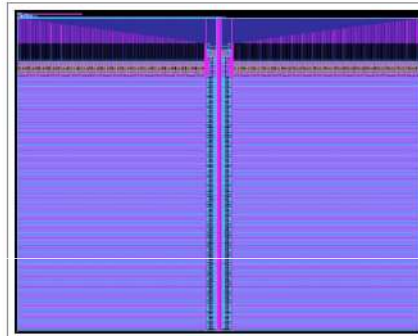
Get the Source Code

Please see our official distribution at:

<https://vlsida.github.io/OpenRAM/>

or clone a copy with:

git clone <https://github.com/VLSIDA/OpenRAM.git>



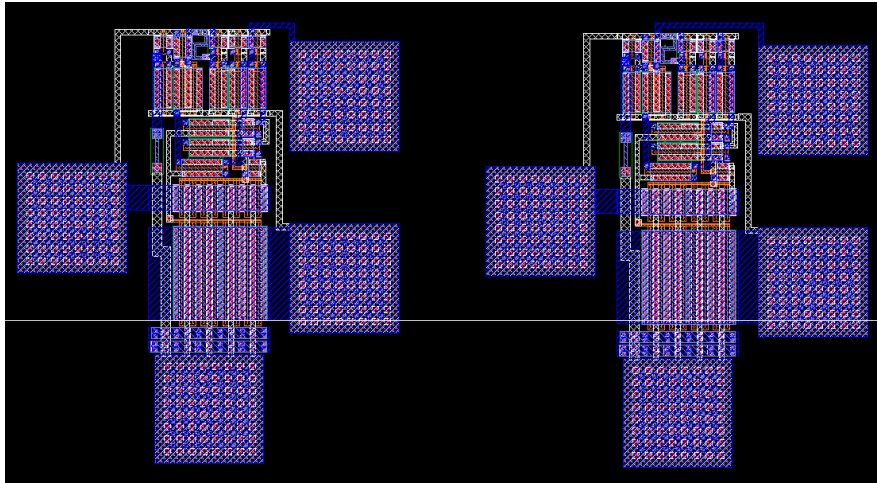
<https://openram.soe.ucsc.edu/>

```
192.168.100.244 - tsuchiya@raizangawa: ~/work/DesignWork/minimal/Ope VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[tsuchiya@raizangawa scn4m]$ ll 2kx8b/
合計 43200
drwxrwxr-x. 2 tsuchiya tsuchiya 4096 2月 17 12:55 spicesim
-rw-rw-r--. 1 tsuchiya tsuchiya 18781696 5月 22 2020 sram_8_2048_scn4m_subm.gds
-rw-rw-r--. 1 tsuchiya tsuchiya 90082 5月 22 2020 sram_8_2048_scn4m_subm.html
-rw-rw-r--. 1 tsuchiya tsuchiya 24272825 5月 22 2020 sram_8_2048_scn4m_subm.lef
-rw-rw-r--. 1 tsuchiya tsuchiya 8151 5月 22 2020 sram_8_2048_scn4m_subm.log
-rw-rw-r--. 1 tsuchiya tsuchiya 333 5月 22 2020 sram_8_2048_scn4m_subm.py
-rw-rw-r--. 1 tsuchiya tsuchiya 1021046 5月 22 2020 sram_8_2048_scn4m_subm.sp
-rw-rw-r--. 1 tsuchiya tsuchiya 1549 5月 27 2020 sram_8_2048_scn4m_subm.v
-rw-rw-r--. 1 tsuchiya tsuchiya 9322 5月 22 2020 sram_8_2048_scn4m_subm_FF_5p0V_25C.lib
-rw-rw-r--. 1 tsuchiya tsuchiya 9320 5月 22 2020 sram_8_2048_scn4m_subm_SS_5p0V_25C.lib
-rw-rw-r--. 1 tsuchiya tsuchiya 9310 5月 22 2020 sram_8_2048_scn4m_subm_TT_5p0V_25C.lib
drwxrwxr-x. 7 tsuchiya tsuchiya 4096 6月 3 2020 synthesis
drwxrwxr-x. 6 tsuchiya tsuchiya 4096 6月 10 2020 verilogsim
[tsuchiya@raizangawa scn4m]$
```



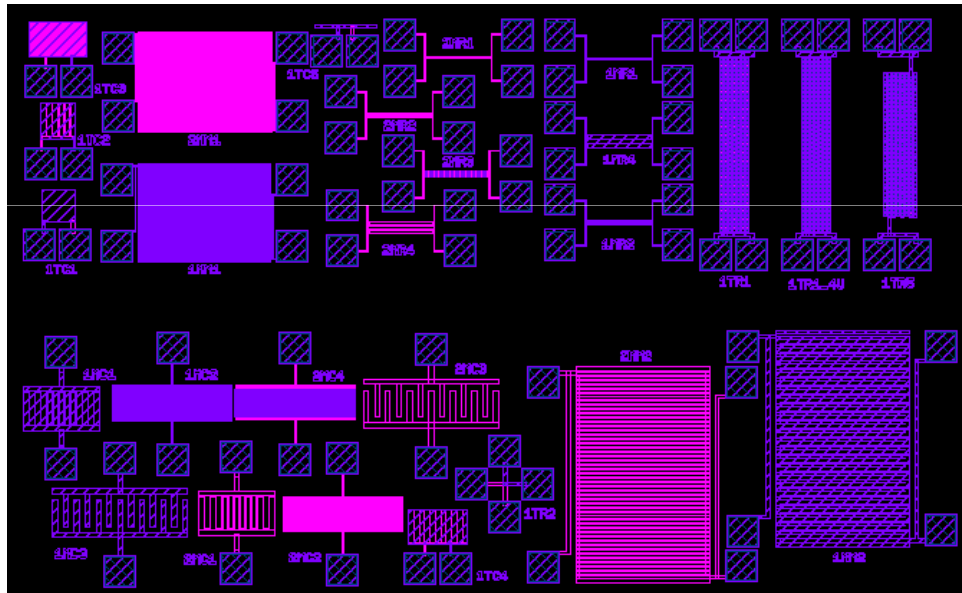
ミニマルファブでの試作回路設計での活用

【デジタルIOTEGの設計】



横河SS₂SOIにて試作しSPICEモデルパラメータを作成中

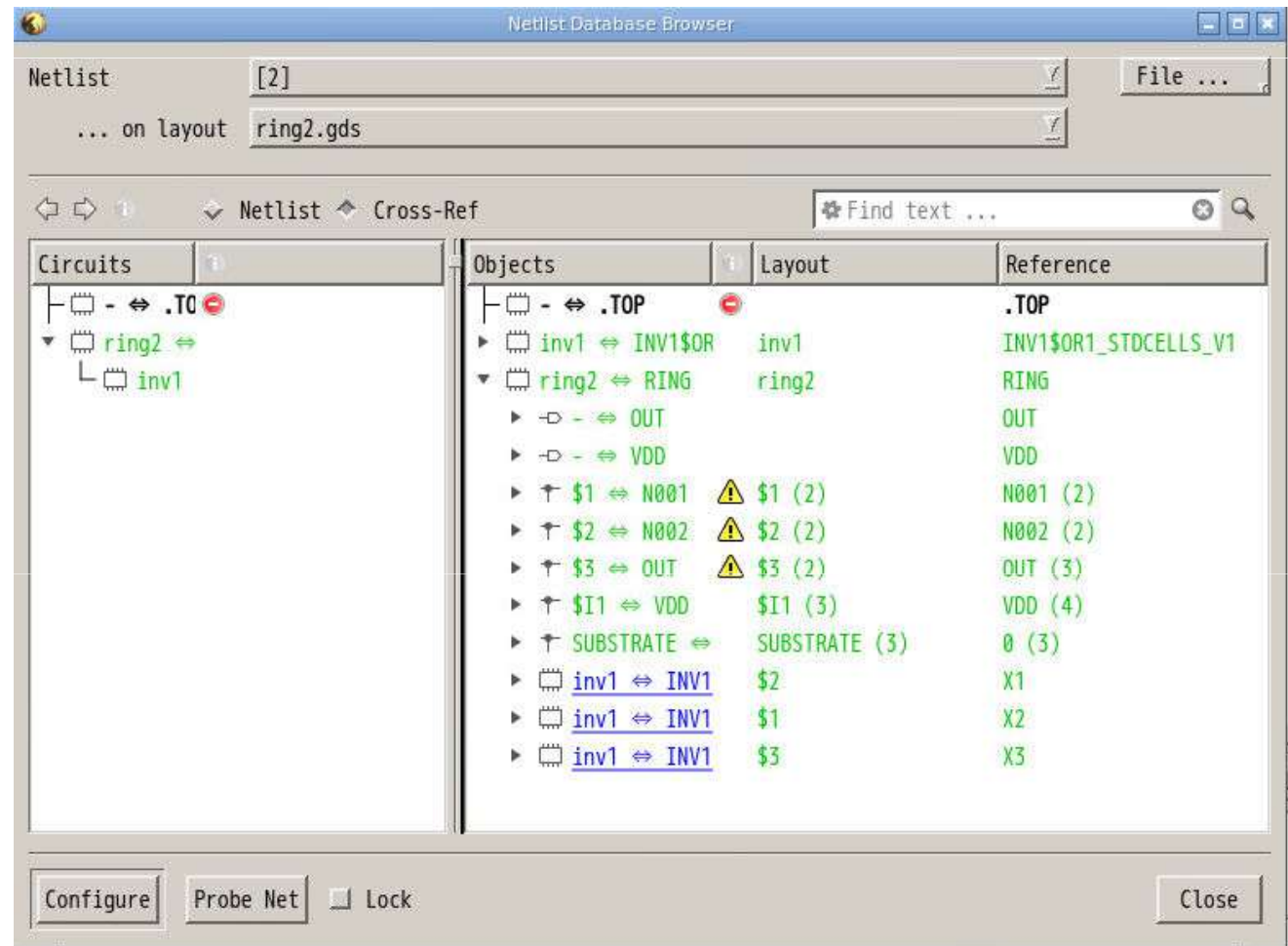
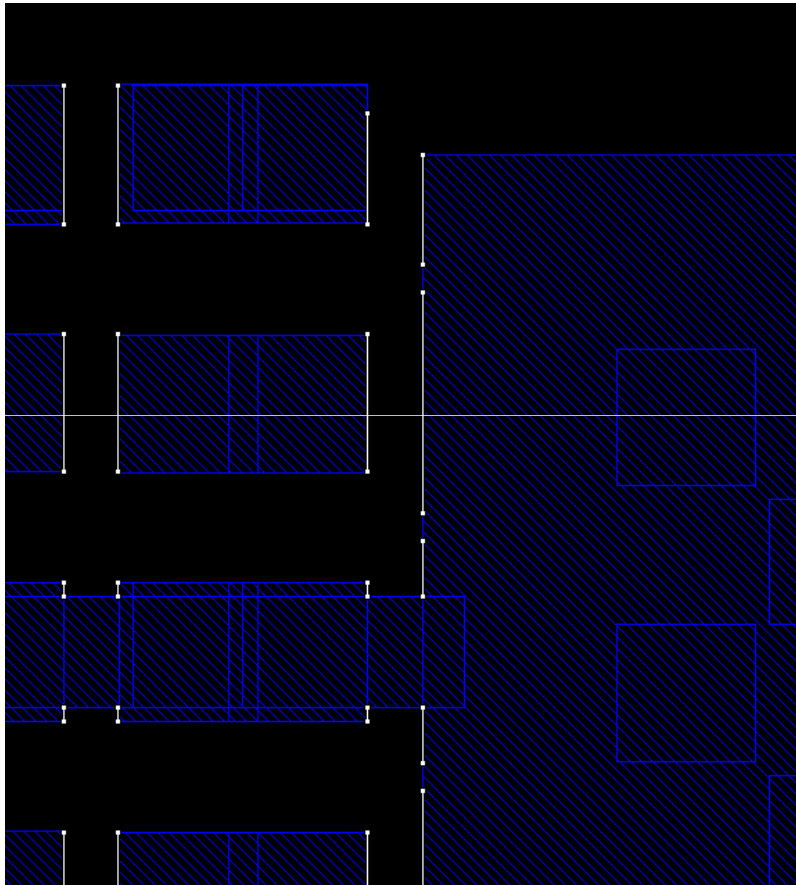
【メタル、VIA、層間絶縁膜評価TEGの設計】



M1,M2,VIA (PAD)のレイヤーのみで構成
各TEGの抵抗、容量をはかることにより、それぞれのプロセスの出来を
はかることが可能



ミニマルファブ向けPDK開発では、Klayoutでレイアウト設計、LVS, DRC実施



Digital回路の合成とレイアウト実験 (CUIによる)

```
[tsuchiya@raizangawa ~]$ ll work/DesignWork/minimal/qflow/coretexM0/
```

合計 40

drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	9月 25	2020	layout	レイアウト結果
drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	7月 16	2018	layout05u	
drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	9月 25	2020	log	ログ
drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	7月 16	2018	log05u	
-rw-rw-r--.	1	tsuchiya	tsuchiya	977	7月 16	2018	project_vars.sh	
-rwxrw-r--.	1	tsuchiya	tsuchiya	1062	1月 10	2019	qflow_exec.sh	実行スクリプト
-rw-rw-r--.	1	tsuchiya	tsuchiya	698	7月 16	2018	qflow_vars.sh	設定ファイル
drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	9月 24	2020	source	ソースファイル
drwxrwxr-x.	3	tsuchiya	tsuchiya	4096	9月 24	2020	synthesis	合成結果
drwxrwxr-x.	2	tsuchiya	tsuchiya	4096	7月 16	2018	synthesis05u	

```
[tsuchiya@raizangawa ~]$
```

```
#!/bin/tcsh -f
```

```
#-----
```

```
# qflow variables for project /mnt/work/DesignWork/minimal/qflow/coretexM0
```

```
#-----
```

```
set projectpath=/mnt/work/DesignWork/minimal/qflow/coretexM0
```

```
set techdir=/usr/local/share/qflow/tech/osu035
```

```
set sourcedir=/mnt/work/DesignWork/minimal/qflow/coretexM0/source
```

```
set synthdir=/mnt/work/DesignWork/minimal/qflow/coretexM0/synthesis
```

```
set layoutdir=/mnt/work/DesignWork/minimal/qflow/coretexM0/layout
```

```
set techname=osu035
```

```
set scriptdir=/usr/local/share/qflow/scripts
```

```
set bindir=/usr/local/share/qflow/bin
```

```
set logdir=/mnt/work/DesignWork/minimal/qflow/coretexM0/log
```

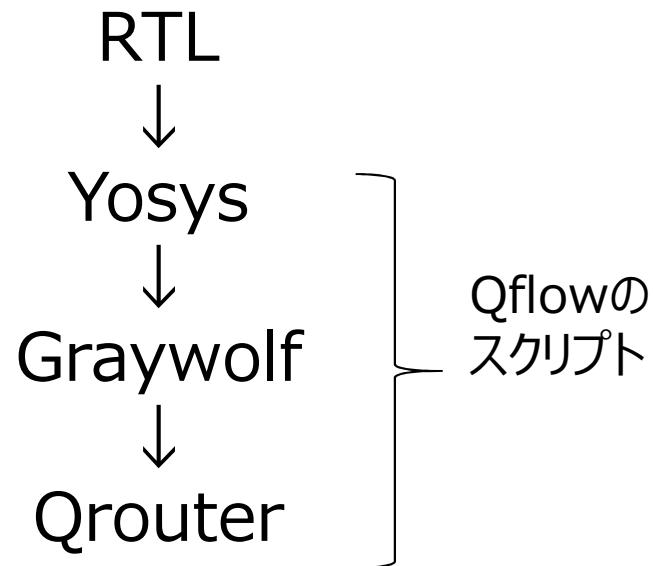
```
#-----
```

設定ファイル



Digital回路の合成とレイアウト実験 (RISC-Vコア)

【Tool Flow】



実行時間 ; ~2時間

【RTLとライブラリ】

SiFiveの評価用RTL

OSU050, OSU035, OSU018

【回路情報】

RISC-V ISA - RV32IMAC

Machine and User Mode with 4 Region Physical Memory Protection

3-stage pipeline with Simultaneous Instruction and Data Access

2 Banks of Tightly Integrated Memory

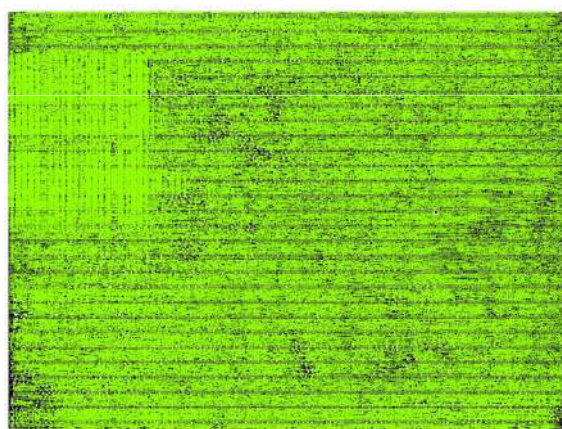
System, Peripheral, and Front Ports

CLIC interrupt controller with 127 interrupts

Advanced debug with 4 hardware breakpoints/watchpoints

2.27/1.46 DMIPS/MHz (Best Effort/Legal)

3.1 CoreMark/MHz

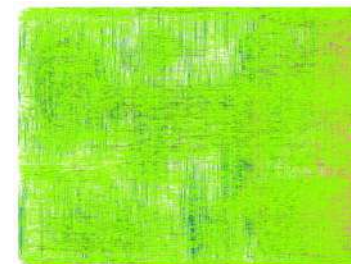


OSU050um

縦横のアスペクトが設定可能

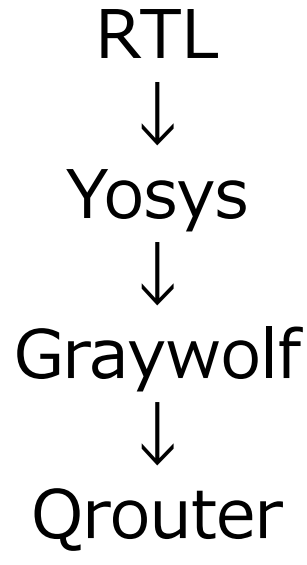


OSU035um



OSU018

【Tool Flow】



Qflowの
スクリプト

実行時間 ; ~ 6 時間

【RTLとライブラリ】
SiFiveの評価用RTL
OSU035, OSU018

E31 Key Features】

Fully compliant with the RISC-V ISA specification

RV32IMAC Support

RV32I – 32-bit RISC-V with 32 integer registers

Integer Multiplication and Division (M) support

Atomic Mode (A) support for high-performance, portable software

Compressed Mode (C) support for better code density

Machine and User Mode Support

In-order, 5-6 stage variable pipeline

Advanced Memory Subsystem

16KB, 2-way Instruction Cache

Instruction Tightly Integrated Memory (ITIM) option

Up to 64KB Data Tightly Integrated Memory (DTIM) support

Efficient and Flexible Interrupts

Local interrupts w/ vectored addresses — up to 16

Platform Level Interrupt Controller (PLIC) — 128 interrupts w/ 7 priority levels

RISC-V Core Local Interruptor (CLINT) — 1 timer, 1 SW

8-Region Physical Memory Protection (PMP)

High performance AMBA Interfaces

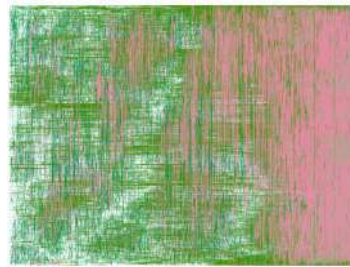
2.58/1.61 DMIPS/MHz (Best Effort/Legal)

3.01 CoreMark/MHz

Detailed Power, Performance, and Area (PPA) Information



OSU035um



OSU018

SiFive殿の評価用RTLを問題無く読
めて合成、レイアウトまで出来ました。
Formalityは、行ってませんので合成結果の
正確性は、確認できてません。

OpenRAMによるRAM生成

OpenRAMを使い以下の設定でネットリスト、.lib, LEF, GDSを生成

```
word_size = 4
num_words = 64
tech_name = "scn4m_subm"
nominal_corners_only = False
process_corners = ["TT"]
supply_voltages = [ 1.0 ]
temperatures = [ 25 ]
output_path = "64x4b"
output_name = "sram_{0}_{1}_{2}".format(word_size, num_words, tech_name)
```

次ページにOpenRAMが生成した仕様書を示す。



OpenRAMによるRAM生成

OpenRAMの生成データ



sram_4_64_scn4m_subm.html

Compiled at: 2020-09-01

DRC errors: skipped

LVS errors: skipped

Git commit id: 1dbf2a4ad6dea40bab54a49ff68d12b3934782d8

Ports and Configuration

Type	Value
WORD_SIZE	4
NUM_WORDS	64
NUM_BANKS	1
NUM_RW_PORTS	1
NUM_R_PORTS	0
NUM_W_PORTS	0
Area (μm^2)	12051

Operating Conditions

Parameter	Value
Power supply (VDD) range	1
Operating Temperature	2
Operating Frequency (F)	

Timing Data

Using analytical model: results

Parameter	Value
din0[3:0] setup rising	
din0[3:0] setup falling	
din0[3:0] hold rising	
din0[3:0] hold falling	
dout0[3:0] cell rise	
dout0[3:0] cell fall	
dout0[3:0] rise transition	
dout0[3:0] fall transition	
csb0 setup rising	
csb0 setup falling	
csb0 hold rising	
csb0 hold falling	
addr0[5:0] setup rising	
addr0[5:0] setup falling	
addr0[5:0] hold rising	
addr0[5:0] hold falling	
web0 setup rising	
web0 setup falling	
web0 hold rising	
web0 hold falling	

Power Data

Pins	Mode	Power	Units
!csb0 & !clk0 & !web0	Read Rising	10.064	mW
!csb0 & !clk0 & !web0	Read Falling	10.064	mW
!csb0 & !clk0 & web0	Write Rising	10.064	mW
!csb0 & !clk0 & web0	Write Falling	10.064	mW
csb0	leakage	0.00047	mW

Characterization Corners

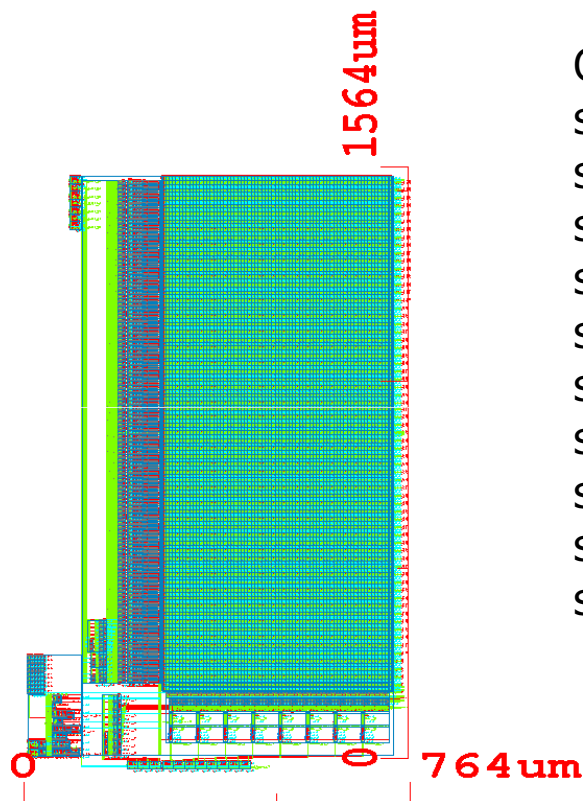
Transistor Type	Power Supply	Temperature	Corner Name
TT	5.0	25	_TT_5p0V_25C.lib
FF	5.0	25	_FF_5p0V_25C.lib
TT	1.0	25	_TT_1p0V_25C.lib
SS	5.0	25	_SS_5p0V_25C.lib

Deliverables

Type	Description	Link
.gds	GDSII layout views	sram_4_64_scn4m_subm.gds
.html	This datasheet	sram_4_64_scn4m_subm.html
.lef	LEF files	sram_4_64_scn4m_subm.lef
.lib	Synthesis models	sram_4_64_scn4m_subm_TT_5p0V_25C.lib
.lib	Synthesis models	sram_4_64_scn4m_subm_FF_5p0V_25C.lib
.lib	Synthesis models	sram_4_64_scn4m_subm_TT_1p0V_25C.lib
.lib	Synthesis models	sram_4_64_scn4m_subm_SS_5p0V_25C.lib
.log	OpenRAM compile log	sram_4_64_scn4m_subm.log
.py	OpenRAM configuration file	sram_4_64_scn4m_subm.py
.sp	SPICE netlists	sram_4_64_scn4m_subm.sp
.v	Verilog simulation models	sram_4_64_scn4m_subm.v



OpenRAMによるRAM生成



ファンダリーのRAM
コンパイラと同等

OpenRAMで生成されるファイル

sram_8_2048_scn4m_subm.gds ; レイアウトファイル

sram_8_2048_scn4m_subm.html : データシート

sram_8_2048_scn4m_subm.lef : 配線用LEFファイル

sram_8_2048_scn4m_subm.log : 生成ログファイル

sram_8_2048_scn4m_subm.py : 設定ファイル

sram_8_2048_scn4m_subm.sp : SPICEネット

sram_8_2048_scn4m_subm.v : Verilog シミュレーションモデル

sram_8_2048_scn4m_subm_FF_5p0V_25C.lib : 合成用ライブラリ

sram_8_2048_scn4m_subm_SS_5p0V_25C.lib : 合成用ライブラリ

sram_8_2048_scn4m_subm_TT_5p0V_25C.lib : 合成用ライブラリ

【設定ファイル】

word_size = 8

num_words = 2048

tech_name = "scn4m_subm"

nominal_corners_only = False

process_corners = ["TT"]

supply_voltages = [5.0]

temperatures = [25]

output_path = "2kx8b"

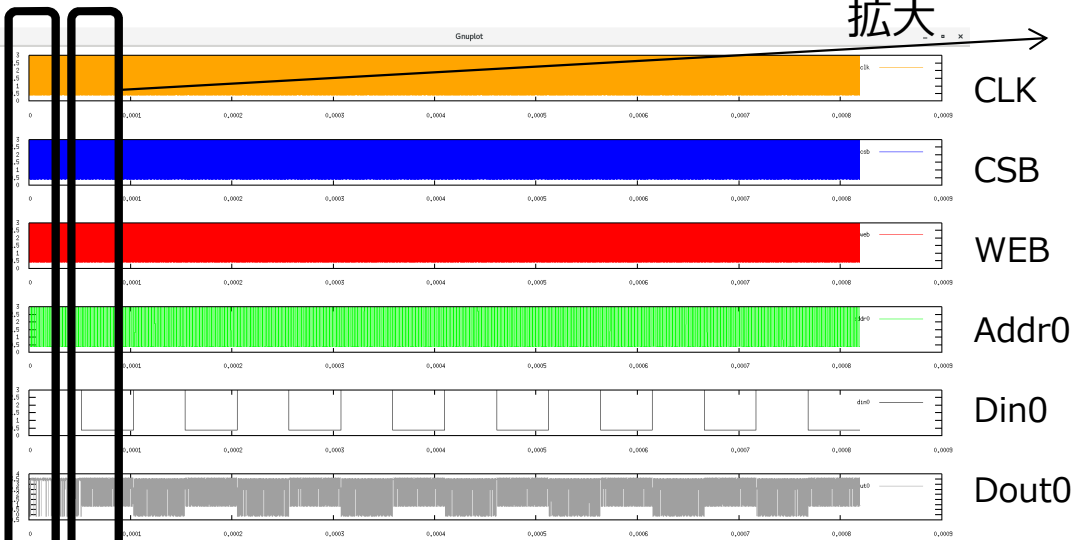
output_name = "sram_{0}_{1}_{2}".format(word_size,
num_words,
tech_name)



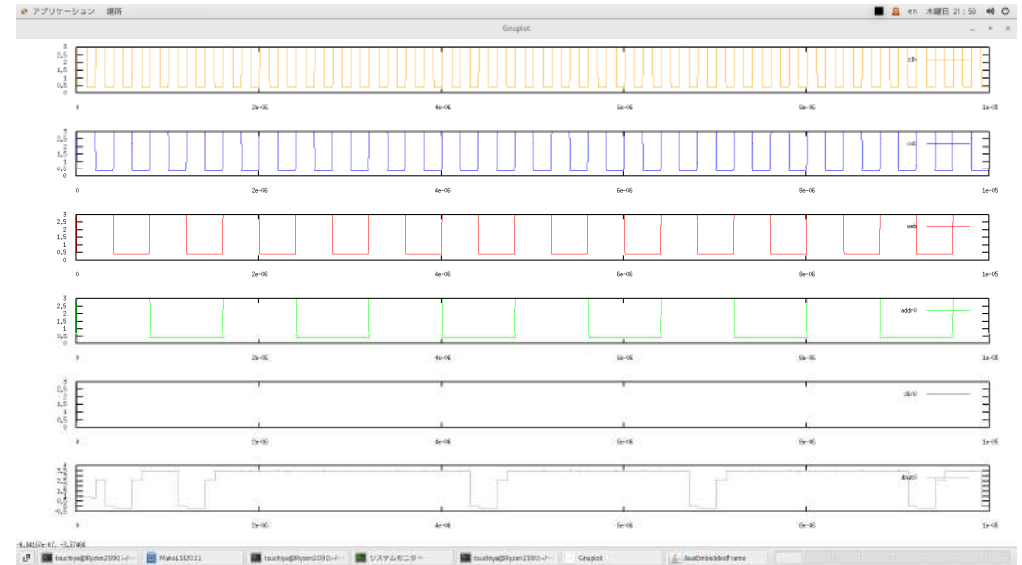
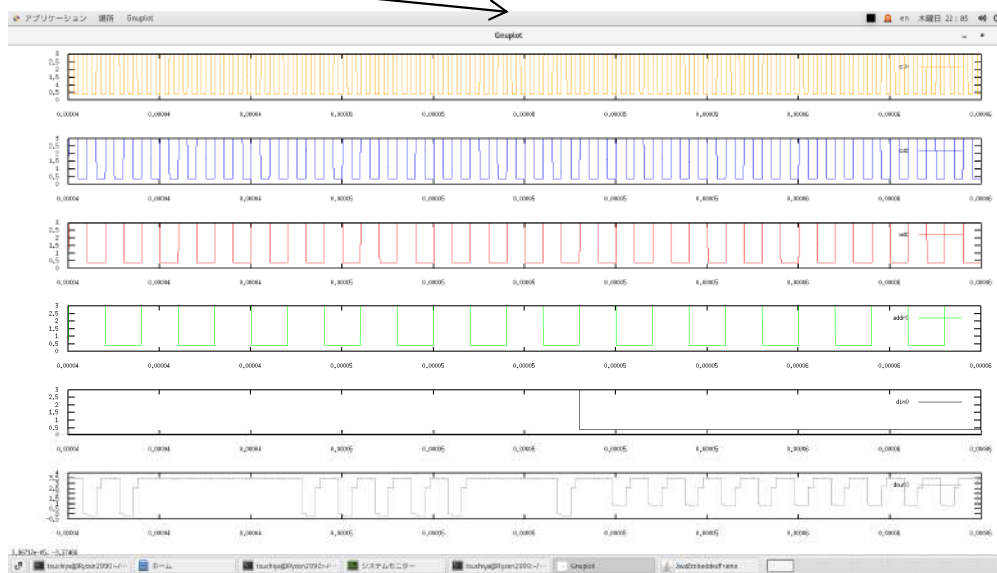
シミュレーション

OpenRAMが生成したネットリストを使いXyceでシミュレーション

拡大



拡大



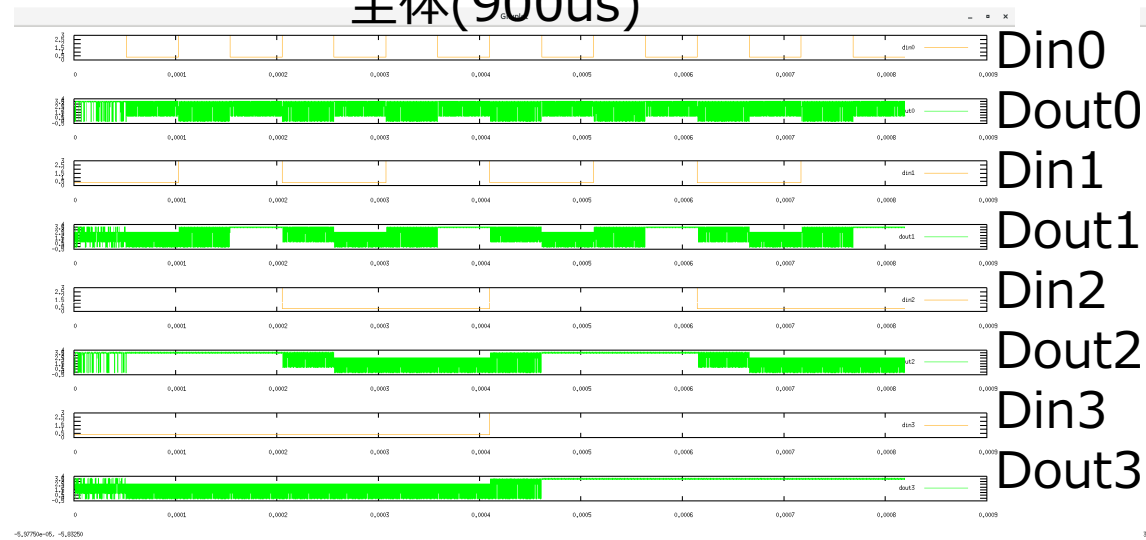
ファンダリーのトランジスタに変換しても動作しそう



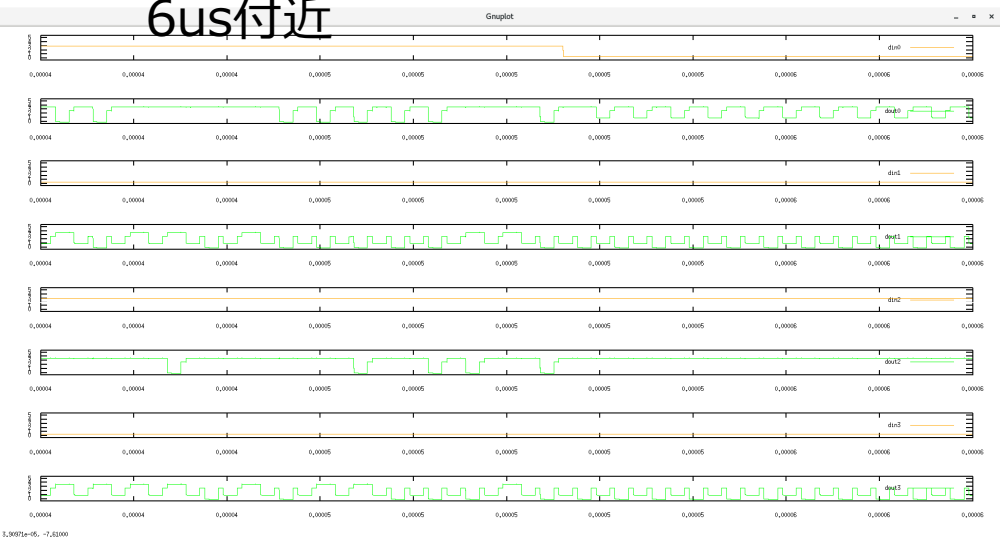
SRAMのシミュレーション

OpenRAMが生成したネットリストを使いファンドリーのトランジスタモデルに変換してXyceでシミュレーション

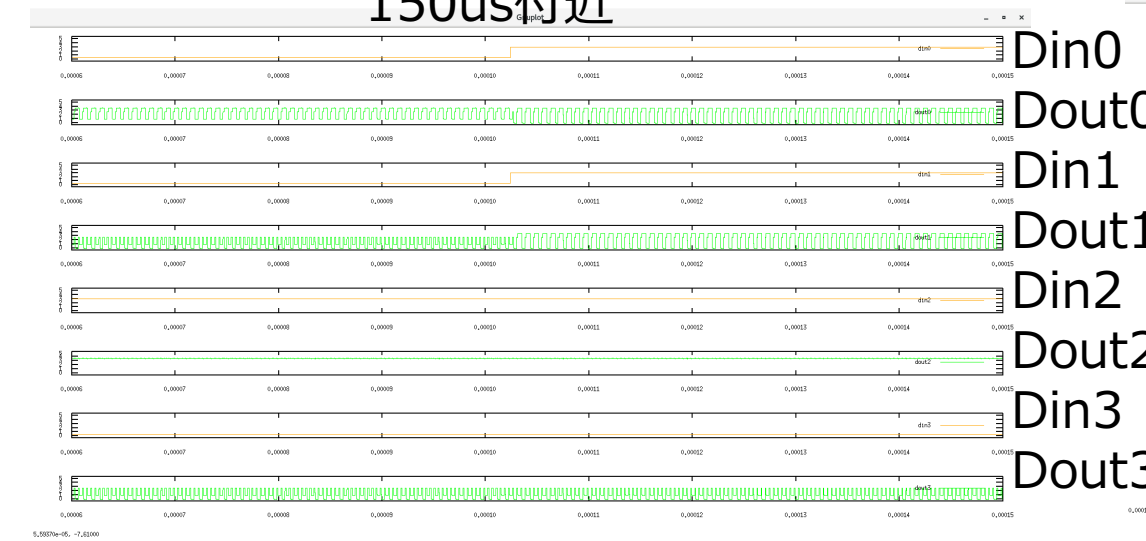
全体(900us)



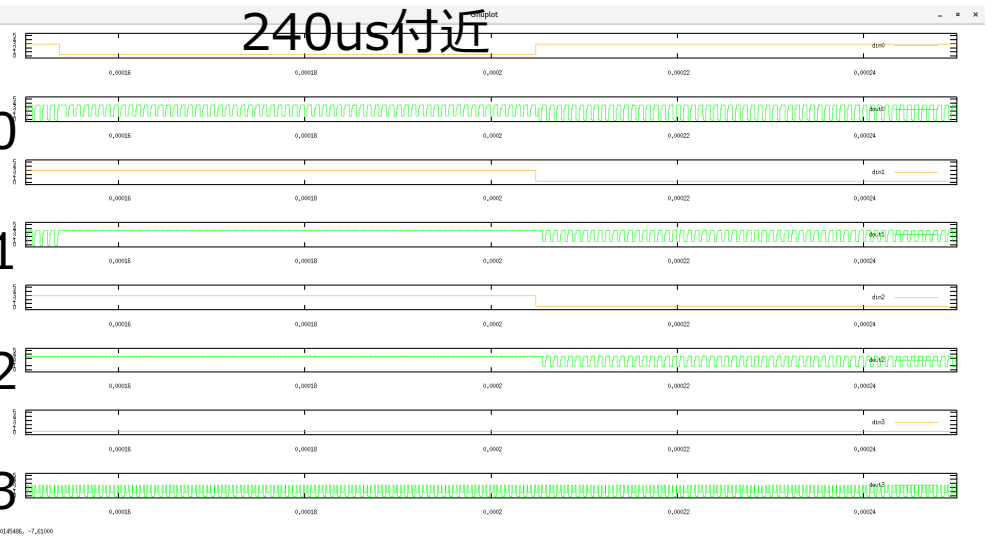
6us付近



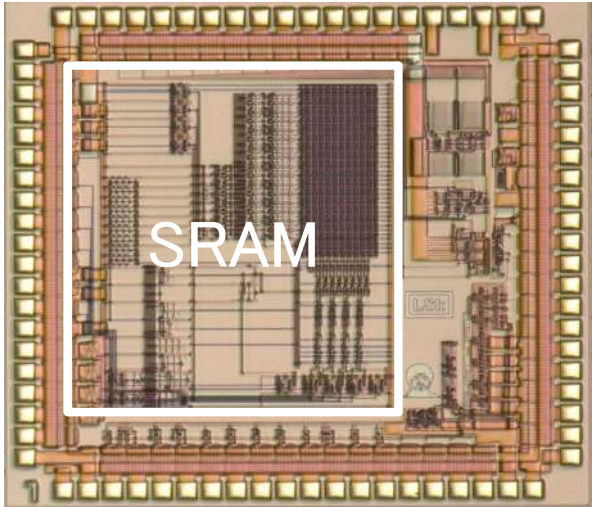
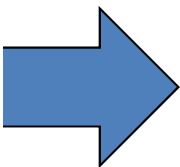
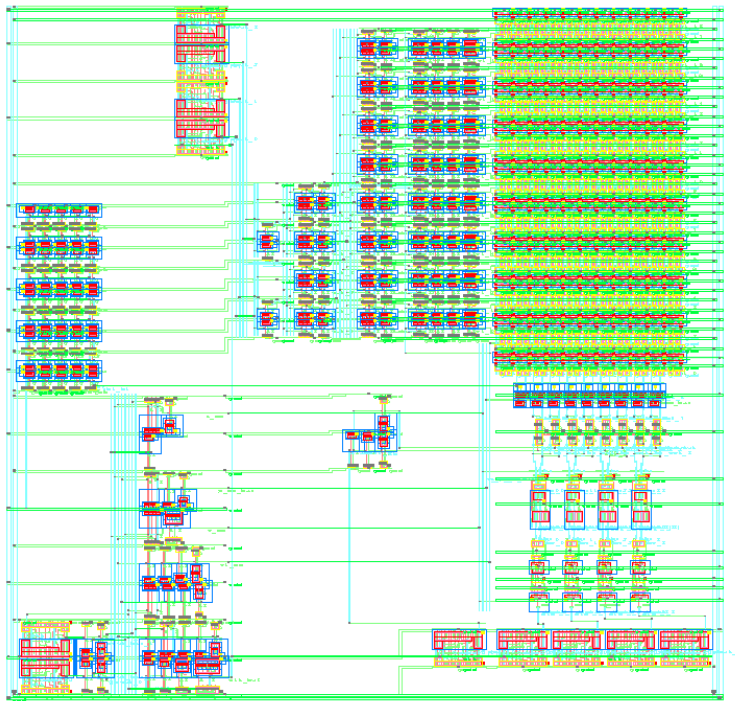
150us付近



240us付近



SRAMレイアウト図



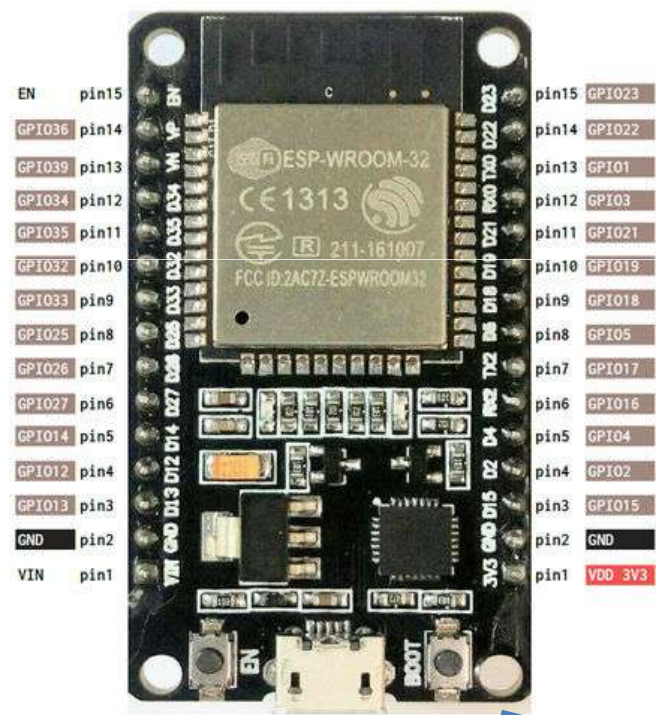
チップ写真

OpenRAMを使いOSU035で合成
レイアウトを1.5倍に拡大(秋田先生)
1.5倍のレイアウトをフェニテックルールでDRC
MakeLSIのプロジェクトで試作
動作検証中

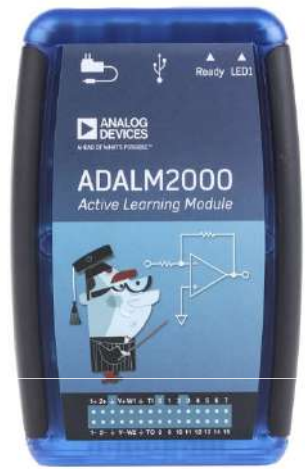


SRAMの測定環境

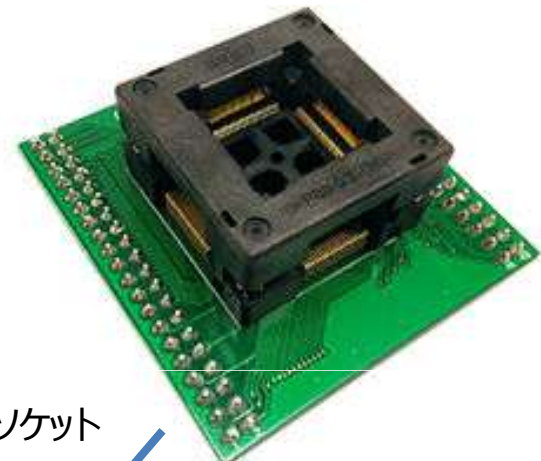
DOIT ESP32 DEVKIT V1 PINOUT



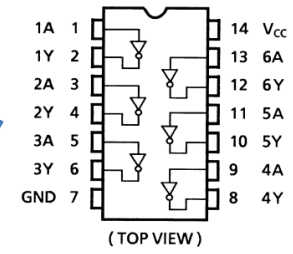
ESP32 DEVKIT:
パターン生成
GPIOを使って14信号 (CS, WE,
CLK, DIN[3:0],
ADDR[5:0]) 生成



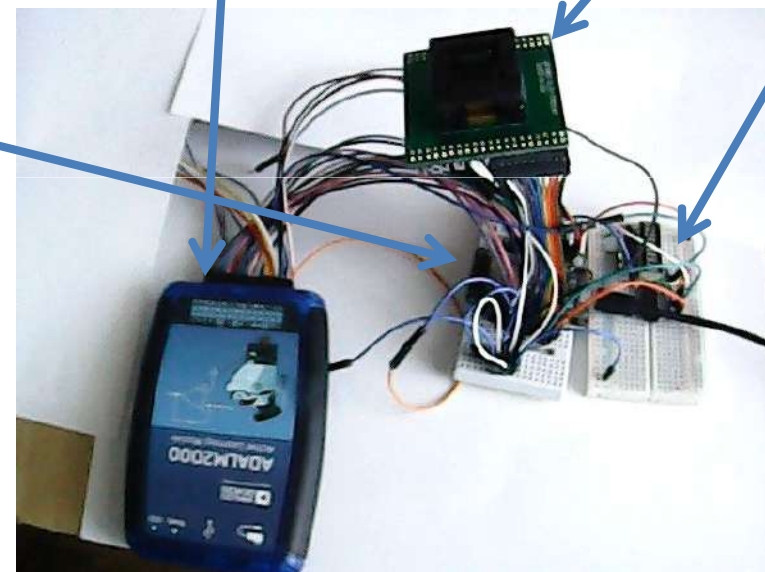
ロジックアナライザ
(16ch) + オシロス
コプ(2ch):



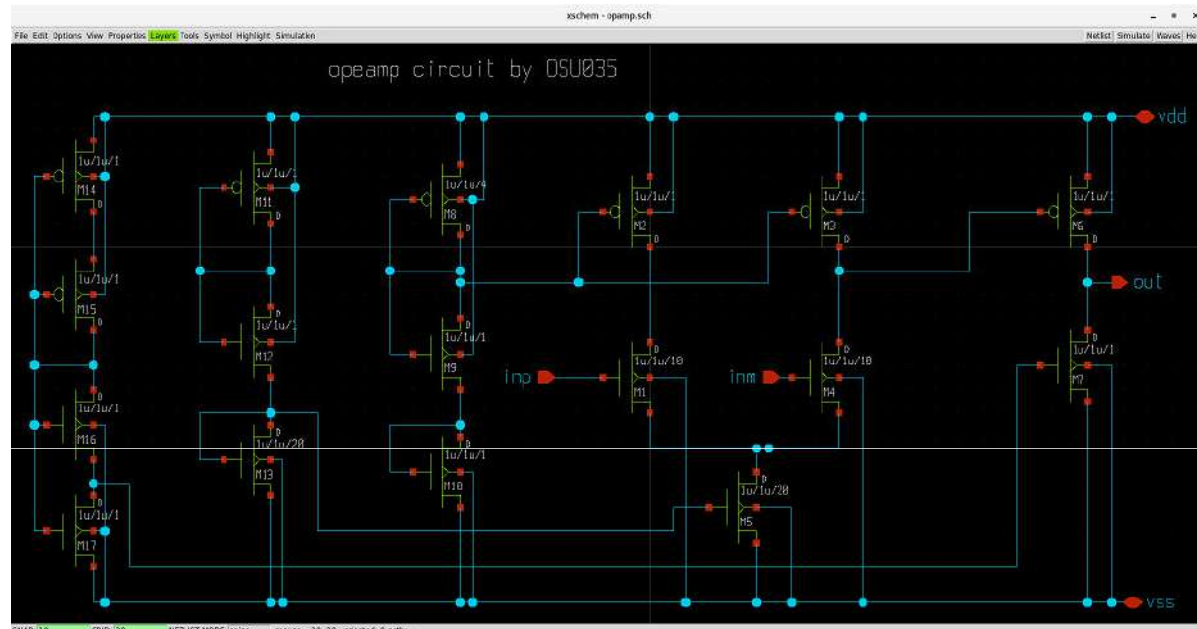
QFPソケット



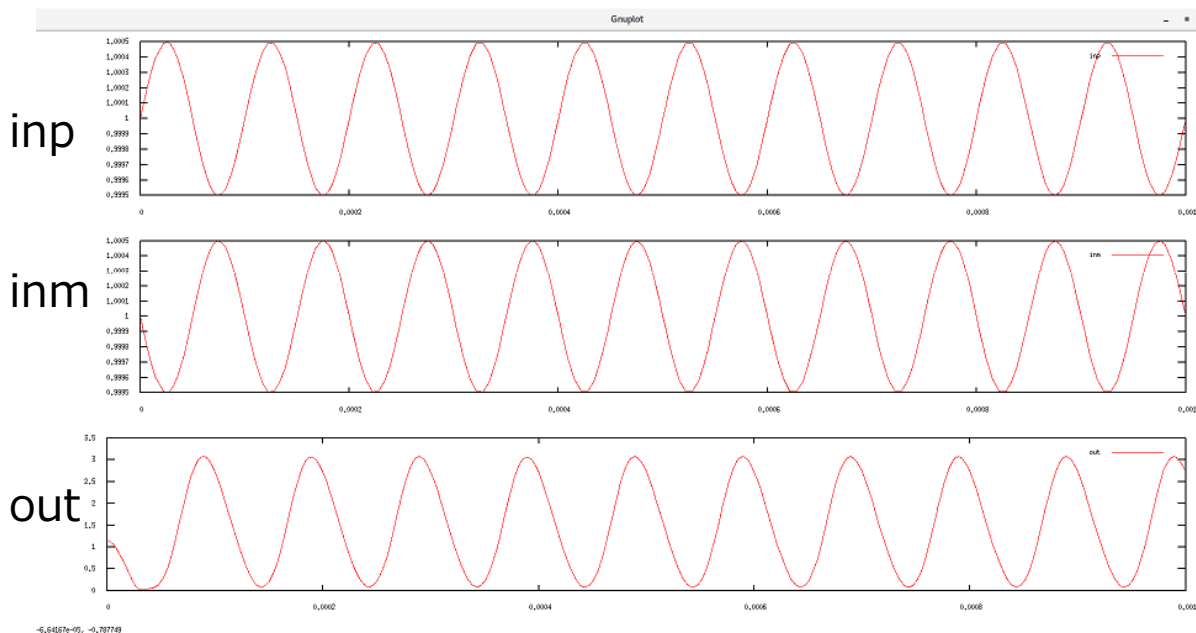
出力バッファ(74HC04)
フェニテックの出力バッファが
小さいのでオシロスコープの
プローブ容量(~10pF)を
駆動できない可能性有り



アナログIC設計実験 (オペアンプ)



Xschemでの回路図入力



Xyceでのシミュレーション

ネットリスト

```
M1 N002 N002 N007 vss n l=1um w=1um m=1
M2 vdd N002 N002 vdd p l=1um w=1um m=4
M3 vdd N002 N005 vdd p l=1um w=1um m=1
M4 vdd N002 N003 vdd p l=1um w=1um m=1
M5 N005 inp N008 vss n l=1um w=1um m=10
M6 N003 inm N008 vss n l=1um w=1um m=10
M7 N008 N006 vss vss n l=1um w=1um m=20
M8 N006 N006 vss vss n l=1um w=1um m=20
M9 N004 N006 N006 vdd p l=1um w=1um m=1
M10 vdd N003 out vdd p l=1um w=1um m=100
M11 N009 obn vss vss n l=1um w=1um m=10
M12 vdd N004 N004 vdd p l=1um w=1um m=1
M13 N007 N007 vss vss n l=1um w=1um m=1
M14 obn N001 vss vss n l=1um w=1um m=1
M16 N001 N001 N001 vdd p l=1um w=1um m=1
M17 vdd N001 N001 vdd p l=1um w=1um m=1
M15 N001 N001 obn vss n l=1um w=1um m=2
M18 out obn N009 vss n l=1um w=1um m=10
```

実験的に作った回路 (～ 3 時間)

使用したMOSFETモデルは、OSU035

入力pp値は、0.0005V(500uV)

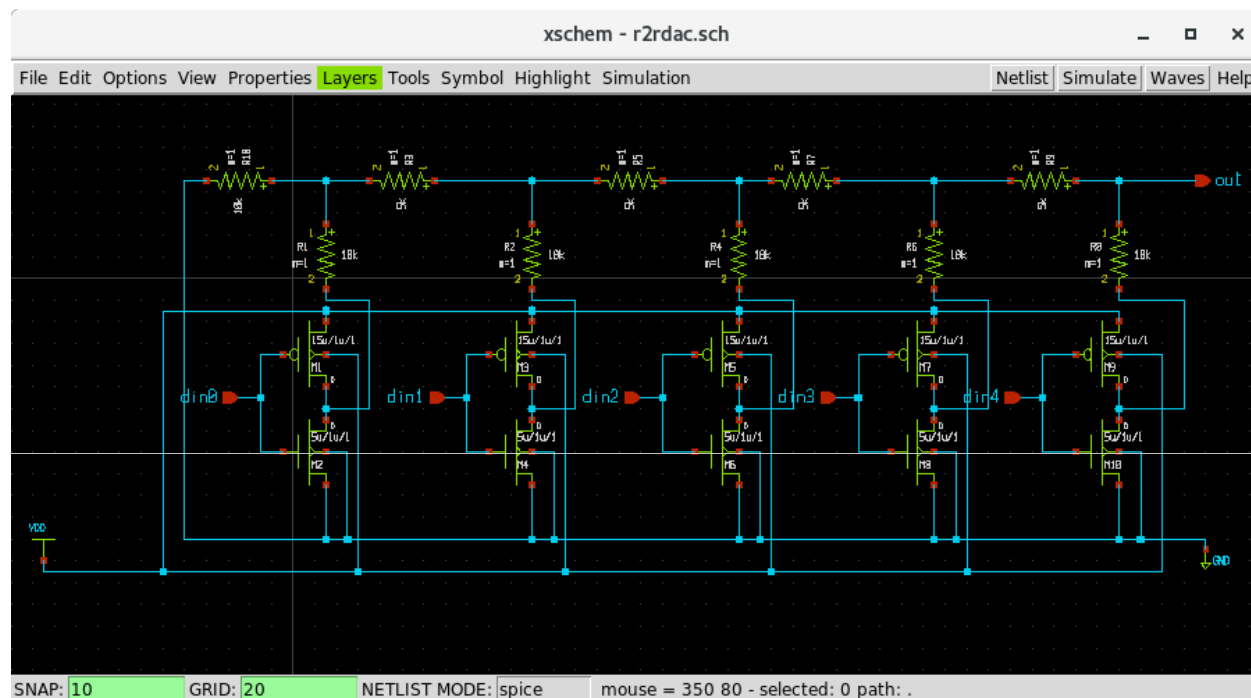
出力pp値は、2.84634V

Gainは、～30dB

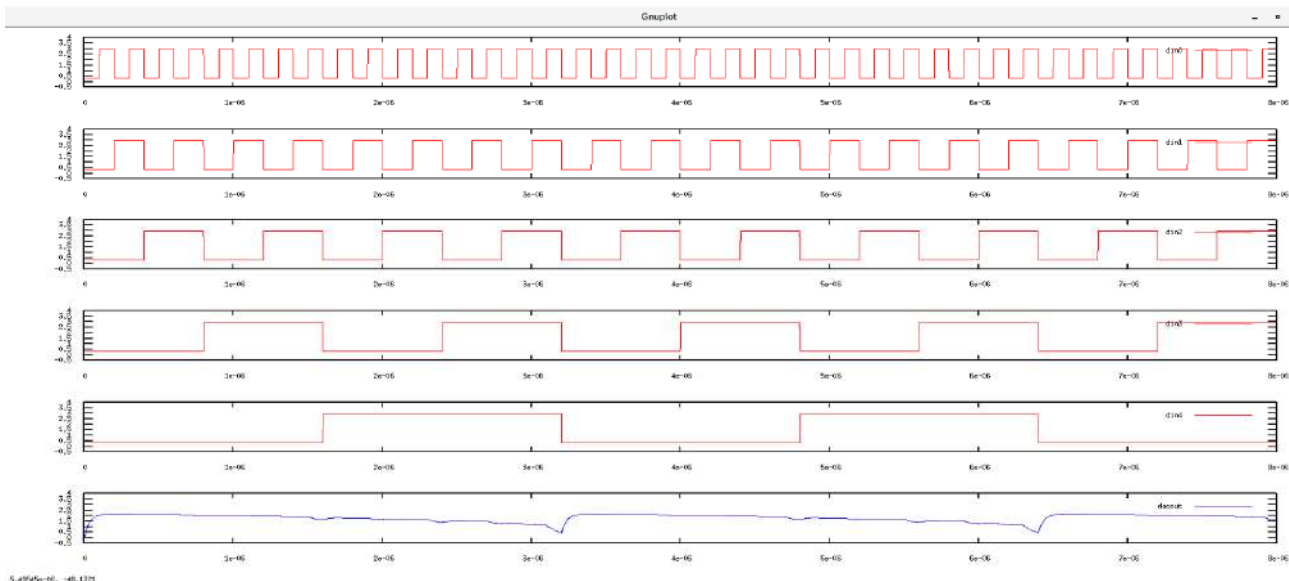
ファブのSPICEモデルで微調整が必要



アナログIC設計実験 (R2R DAC)



Xschemでの回路図入力



Xyceでのシミュレーション

ネットリスト

*R2R DAC.

subckt r2rdac din0 din1 din2 din3 din4 out VDD VSS

.ipin din0.ipin din1*.ipin din2*.ipin din3*.ipin

din4*.opin out

R1 net1 net5 10k m=1

R2 net2 net6 10k m=1

R3 net2 net1 5k m=1

R4 net3 net7 10k m=1

R5 net3 net2 5k m=1

R6 net4 net8 10k m=1

R7 net4 net3 5k m=1

R8 out net9 10k m=1

R9 out net4 5k m=1

M1 net5 din0 VDD VDD p w=15u l=1u m=1

M2 net5 din0 GND GND n w=5u l=1u m=1

R10 net1 GND 10k m=1

M3 net6 din1 VDD VDD p w=15u l=1u m=1

M4 net6 din1 GND GND n w=5u l=1u m=1

M5 net7 din2 VDD VDD p w=15u l=1u m=1

M6 net7 din2 GND GND n w=5u l=1u m=1

M7 net8 din3 VDD VDD p w=15u l=1u m=1

M8 net8 din3 GND GND n w=5u l=1u m=1

M9 net9 din4 VDD VDD p w=15u l=1u m=1

M10 net9 din4 GND GND n w=5u l=1u m=1

.ends

動作している。微調整は、必要



EDA費用は、削減可能

EDA価格(Time Base Lisence)
HDLシミュレータ(100万円/年間)
回路合成(1000万円/年間)
自動配置配線(1000万円/年間)
SPICEシミュレータ(100万円/年間)
レイアウト設計(500万円/年間)
レイアウト検証(500万円/年間)
合計費用=3700万円



1日の費用=3700÷225日⇒16万円



EDA価格(Time Base Lisence)
HDLシミュレータ(10万円/1月)
回路合成(100万円/1月)
自動配置配線(100万円/1月)
SPICEシミュレータ(10万円/1月)
レイアウト設計(50万円/1月)
レイアウト検証(50万円/1月)
合計費用=370万円

商用EDA稼働(5品種開発)
HDLシミュレータ(100日・人/年間)
回路合成(25日・人/年間)
自動配置配線(50日・人/年間)
SPICEシミュレータ(100日・人/年間)
レイアウト設計(100日・人/年間)
レイアウト検証(50日・人/年間)



1品種のEDA費用=3700÷5⇒740万円



商用EDA稼働(5品種開発)
HDLシミュレータ(10日・人/年間)
回路合成(2.5日・人/年間)
自動配置配線(5日・人/年間)
SPICEシミュレータ(10日・人/年間)
レイアウト設計(10日・人/年間)
レイアウト検証(5日・人/年間)
1品種のEDA費用=370÷5=74万円



今後の活動予定

オープンソースだけで簡単なLSIの試作（8ビットマイコン）に挑戦(2021/4Q)
ファンダリーのPDKが利用できるように変換手法の開発

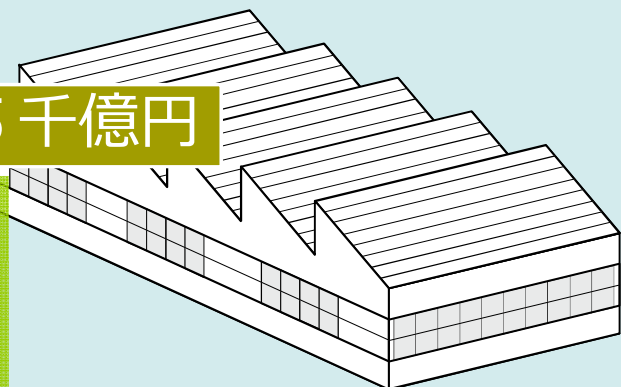
【オープンソースEDAの課題】

- ☆マニュアルの日本語化（ミニマルEDA向けで作成中）
- ☆高級言語の合成ソフトウェア（JAVA⇒VerilogやRuby⇒Verilog）の開発
- ☆クラウドでの利用環境
- ☆オープンで使用可能なPDKの準備(SkyWater Open PDK for the 130nm processのような)

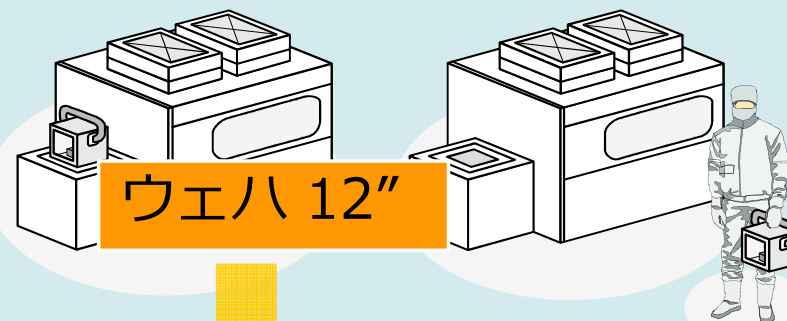


既存メガファブ

投資 5 千億円



200m

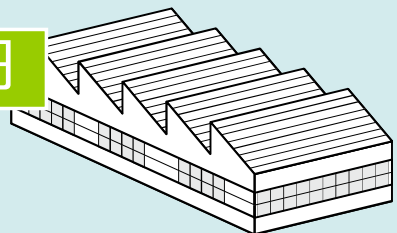


ウェハ 12"

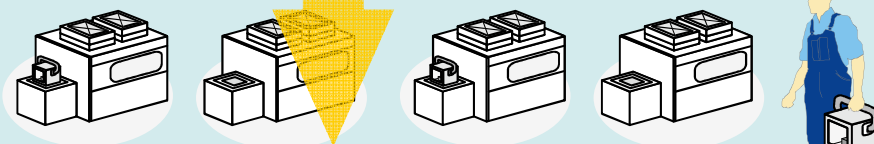
2m

ミニマルファブ

投資 5 億円



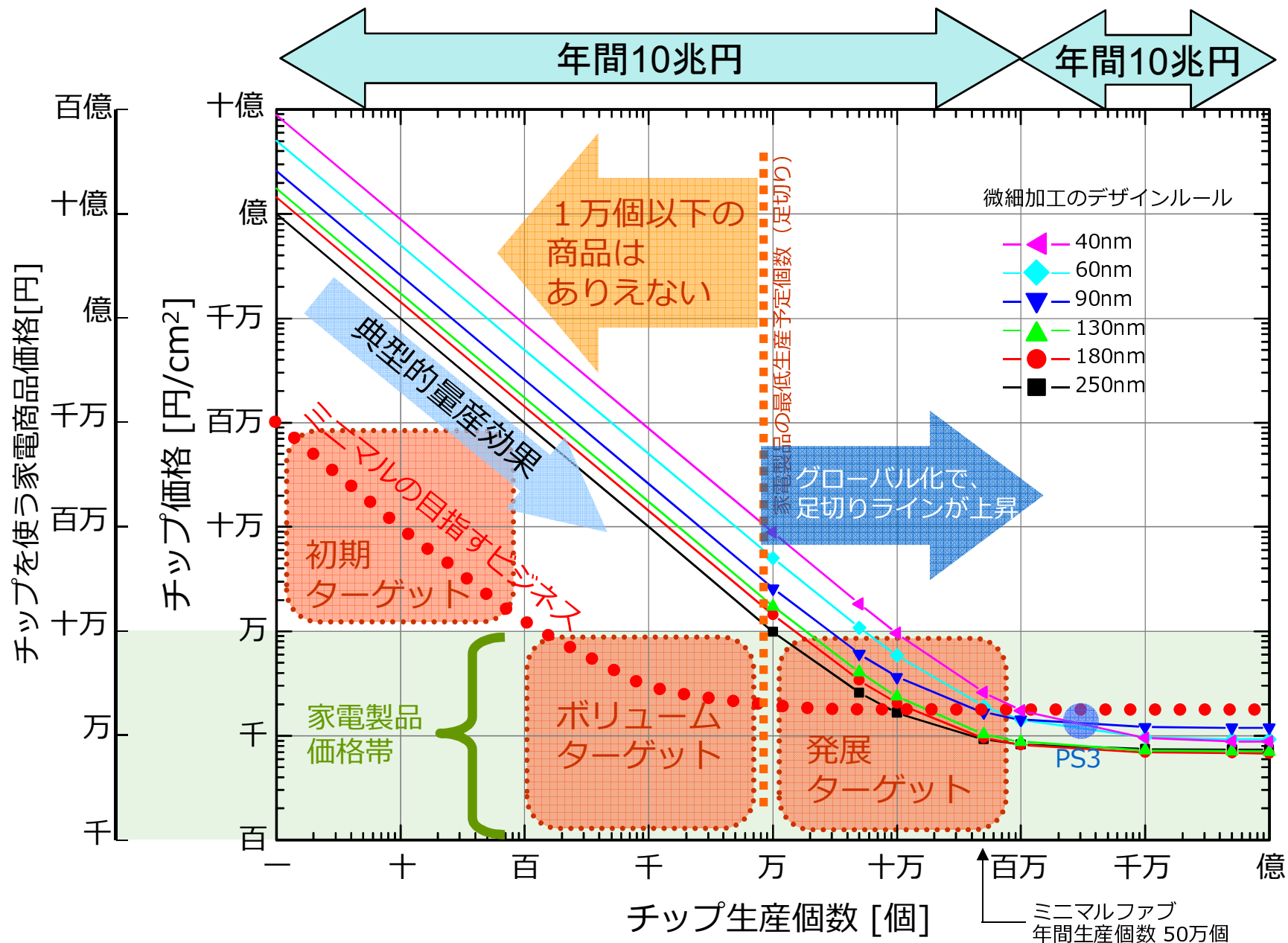
10m



0.3m

ウェハ 0.5"

クリーンルーム不要



ミニマルファブの紹介



半導体製造装置とWaferを小型化して
少量多品種LSIを短期間で製造する
夢のような技術

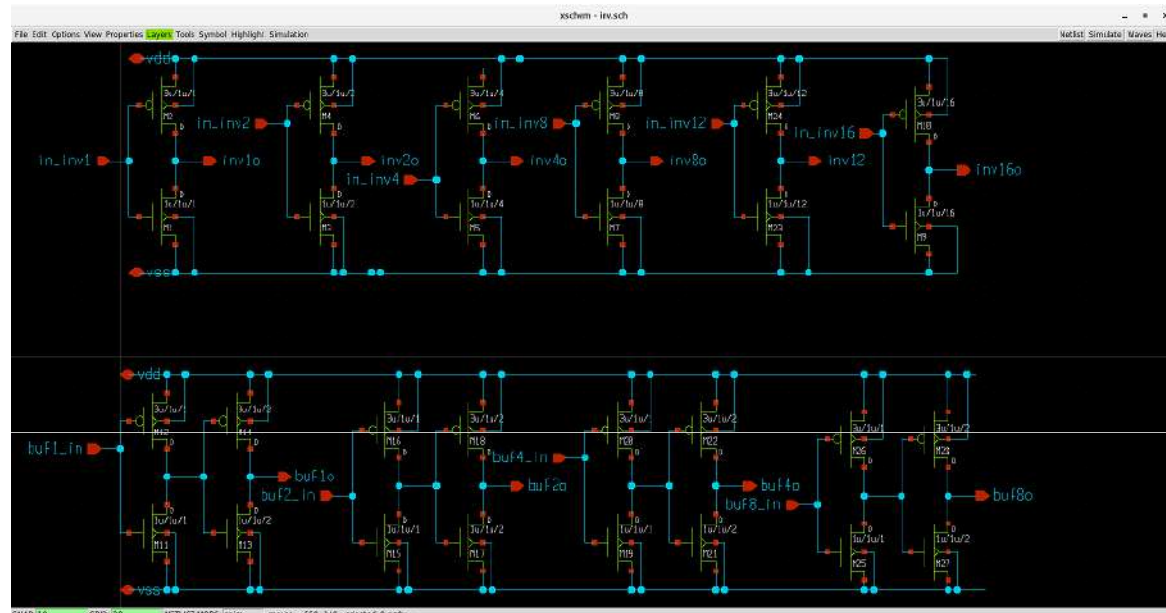
装置は、すべて
(45cmx33cmx144cm)サイズ

<https://www.minimalfab.com/>
<https://www.yokogawa.co.jp/industries/ssd/minimal-fab/>
<https://www.yokogawa.co.jp/industries/ssd/minimal-fab/>
<https://www.youtube.com/watch?v=6kWS9Wy6WdA>
<https://www.youtube.com/watch?v=6kWS9Wy6WdA>

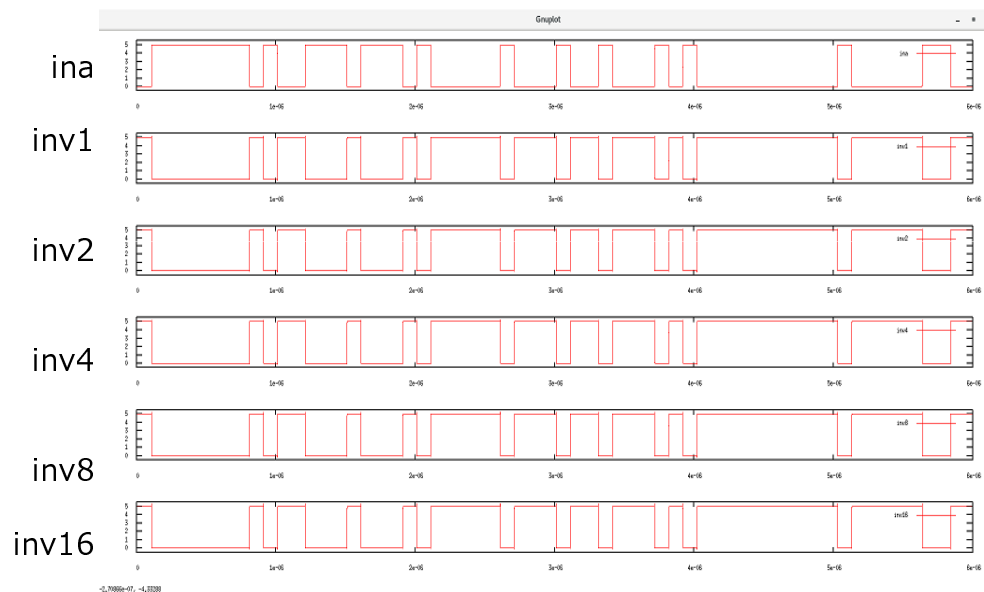
参照ください



ミニマルファブ向けStandard Cell開発では、XschemとXyceで回路設計とシミュレーション実施中



Xschemでの回路図入力

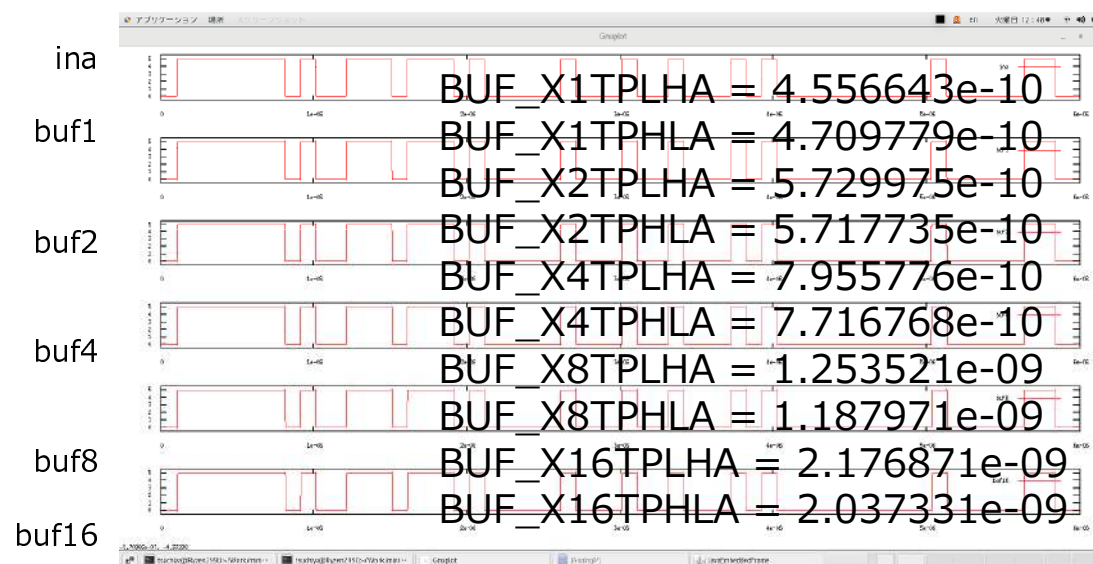


Xyceでのシミュレーション

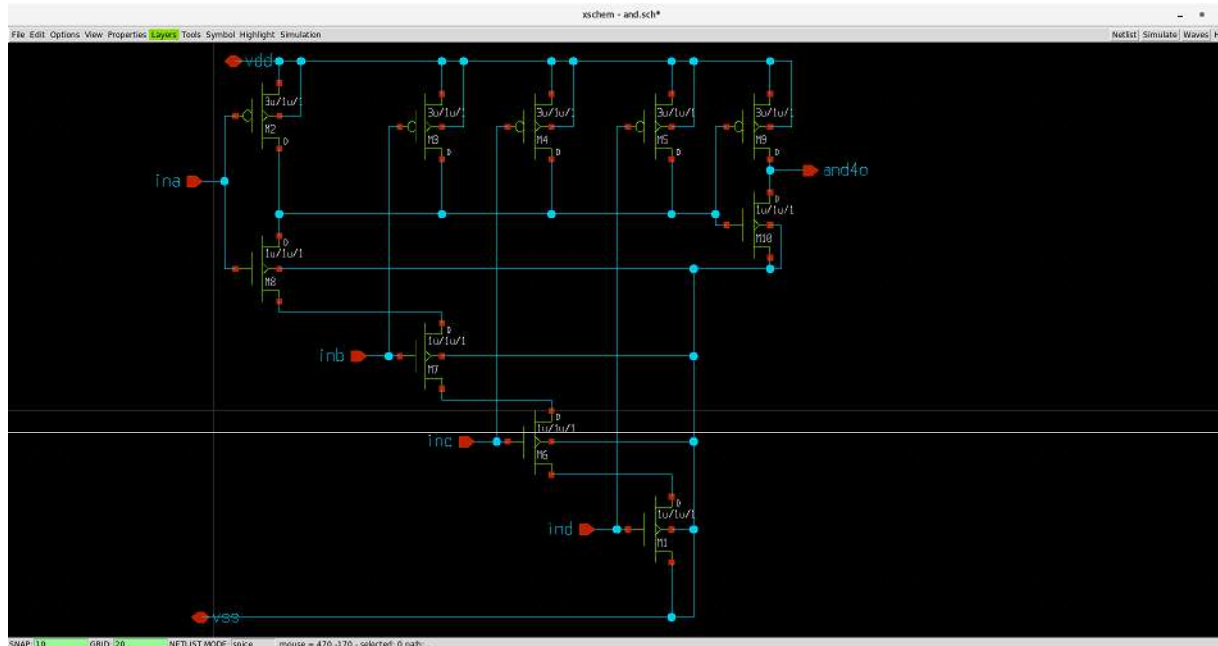
遅延時間検出

.measure tran ...

INV_X1TPLHA = 1.966268e-10
 INV_X1TPHLA = 2.275912e-10
 INV_X2TPLHA = 1.613415e-10
 INV_X2TPHLA = 1.683656e-10
 INV_X4TPLHA = 1.455578e-10
 INV_X4TPHLA = 1.392438e-10
 INV_X8TPLHA = 1.449618e-10
 INV_X8TPHLA = 1.273222e-10
 INV_X16TPLHA = 1.590202e-10
 INV_X16TPHLA = 1.288945e-10



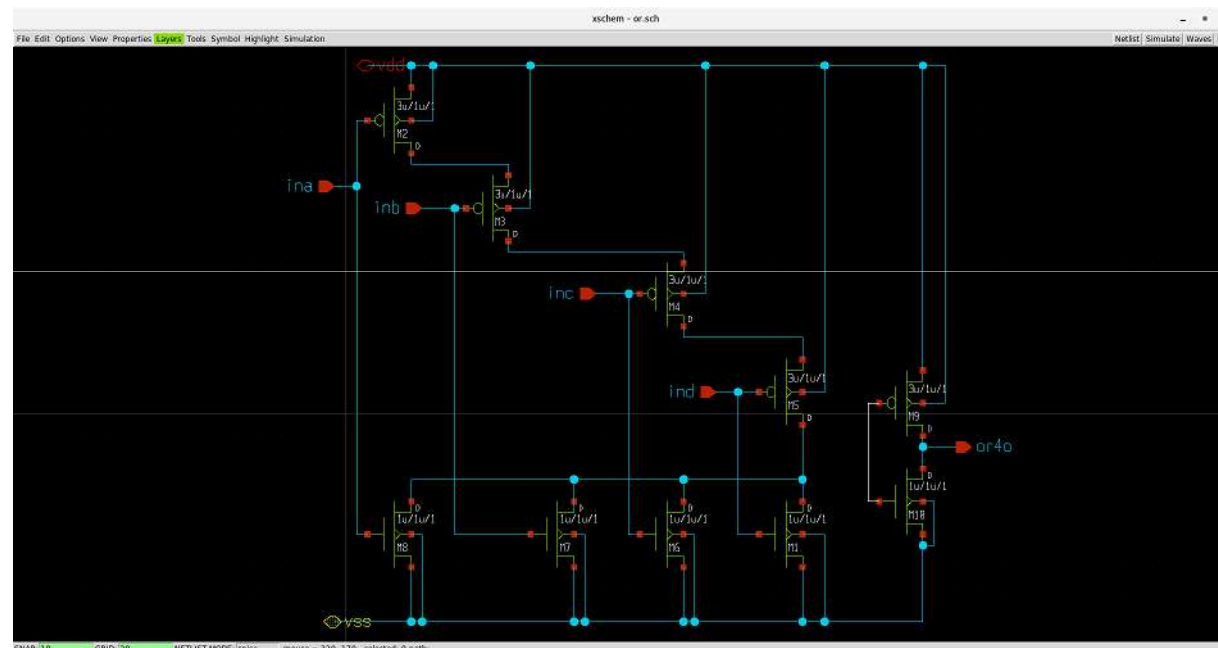
ミニマルファブ向けStandard Cell開発では、XschemとXyceで回路設計とシミュレーション実施中



遅延時間検出

.measure tran ...

$AND4_X1TPLHA = 9.144783e-10$
 $AND4_X1TPHLA = 5.088498e-10$
 $AND4_X1TPLHB = 9.174660e-10$
 $AND4_X1TPHLB = 5.360135e-10$
 $AND4_X1TPLHC = 9.128776e-10$
 $AND4_X1TPHLC = 5.545558e-10$
 $AND4_X1TPLHD = 9.110165e-10$
 $AND4_X1TPHLD = 5.653679e-10$



$OR4TPLHA = 6.439002e-10$
 $OR4TPHLA = 1.233979e-09$
 $OR4TPLHB = 6.500387e-10$
 $OR4TPHLB = 1.260135e-09$
 $OR4TPLHC = 6.006907e-10$
 $OR4TPHLC = 1.063810e-09$
 $OR4TPLHD = 4.978724e-10$
 $OR4TPHLD = 8.921570e-10$

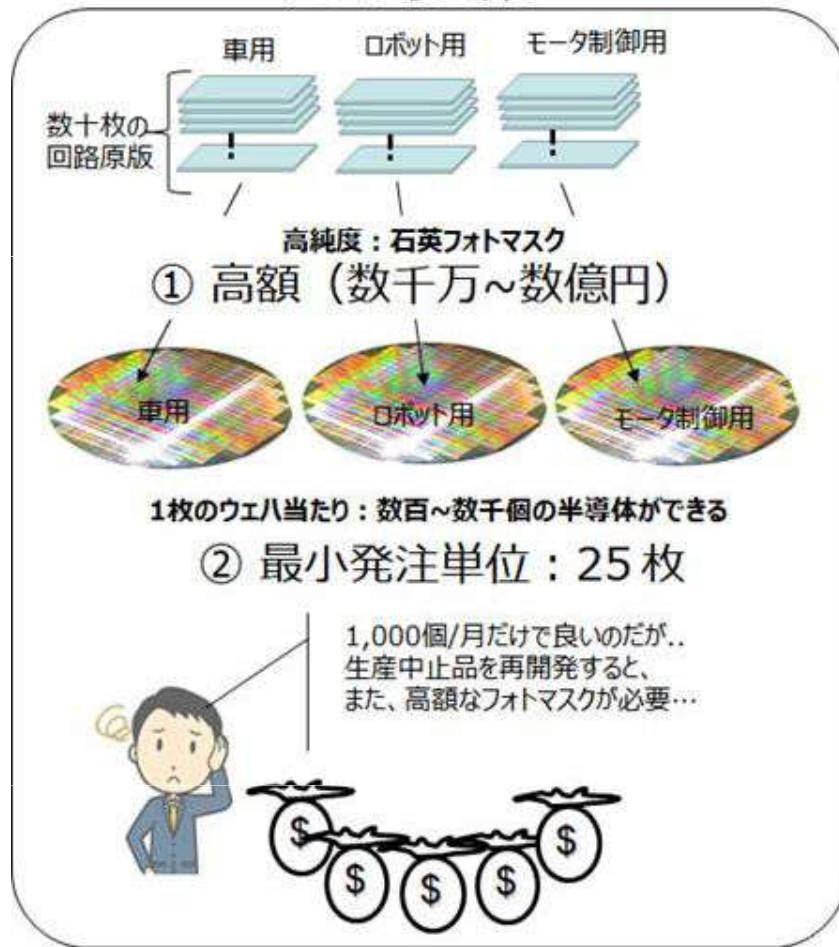




37

開発費用を削減するアイデア(再配線によるLSI製造技術の研究開発)

従来技術

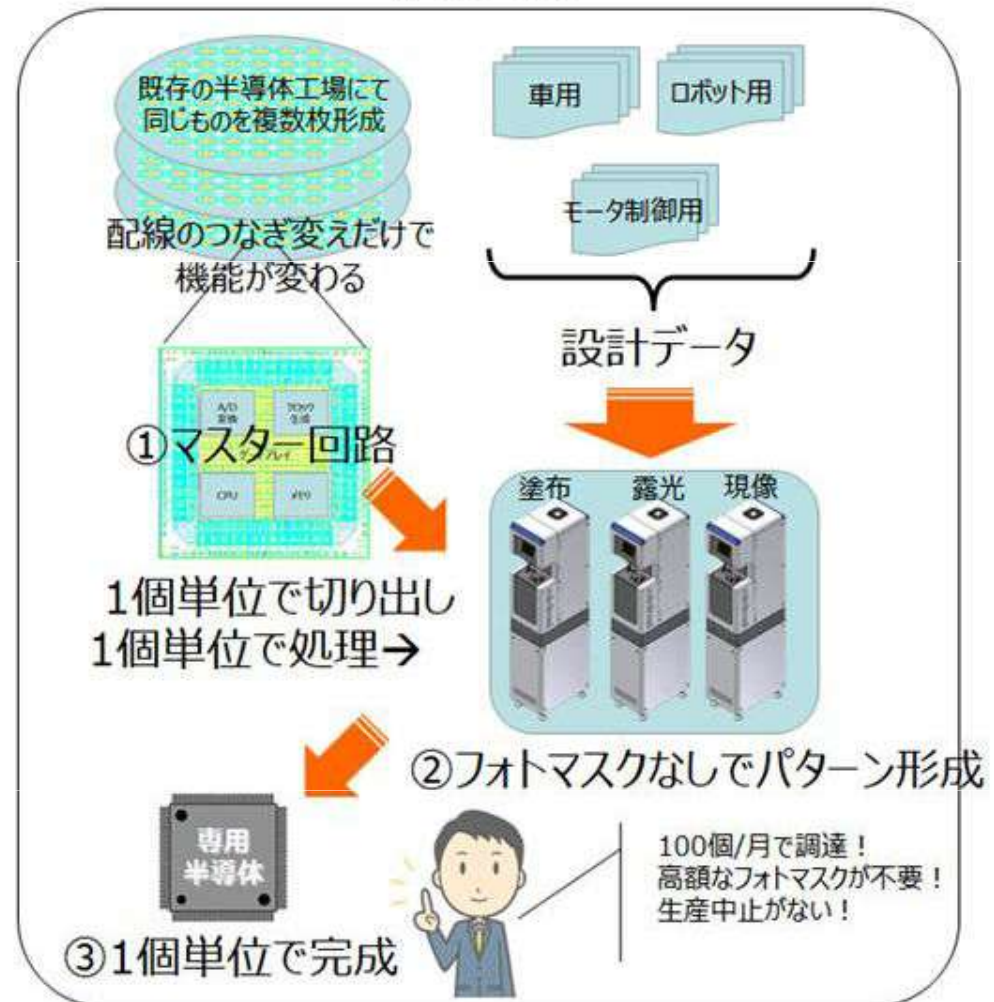


弊社での実績（H8マイコン）

開発費用：4000万円

開発期間：2年間

新技術



予想される費用＆期間

開発費用：2000万円

開発期間：8ヶ月

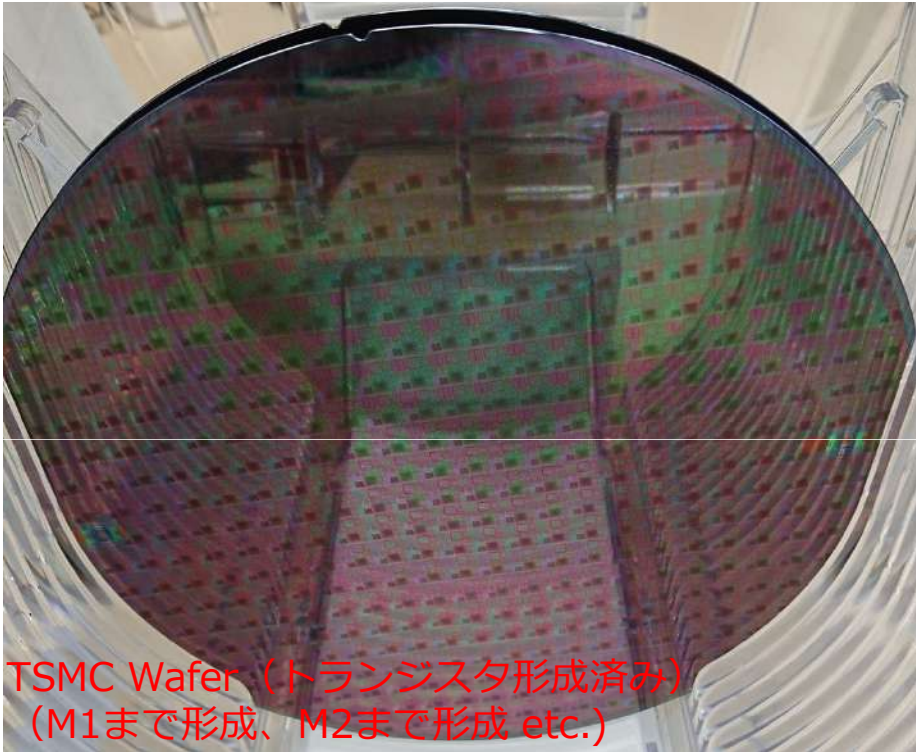


研究開発の成果概要

オープンソースEDAを活用



Process: TSMC 0.35um CMOS Mixed-Signal 3.3V/5.0V
※ 試作したデバイスは、全て5V系のデバイスを実装
※ 実質、0.5umプロセス相当と同程度
※ 赤枠部 = 配線出来栄え評価用TEG



TSMC Wafer (トランジスタ形成済み)
(M1まで形成、M2まで形成 etc.)

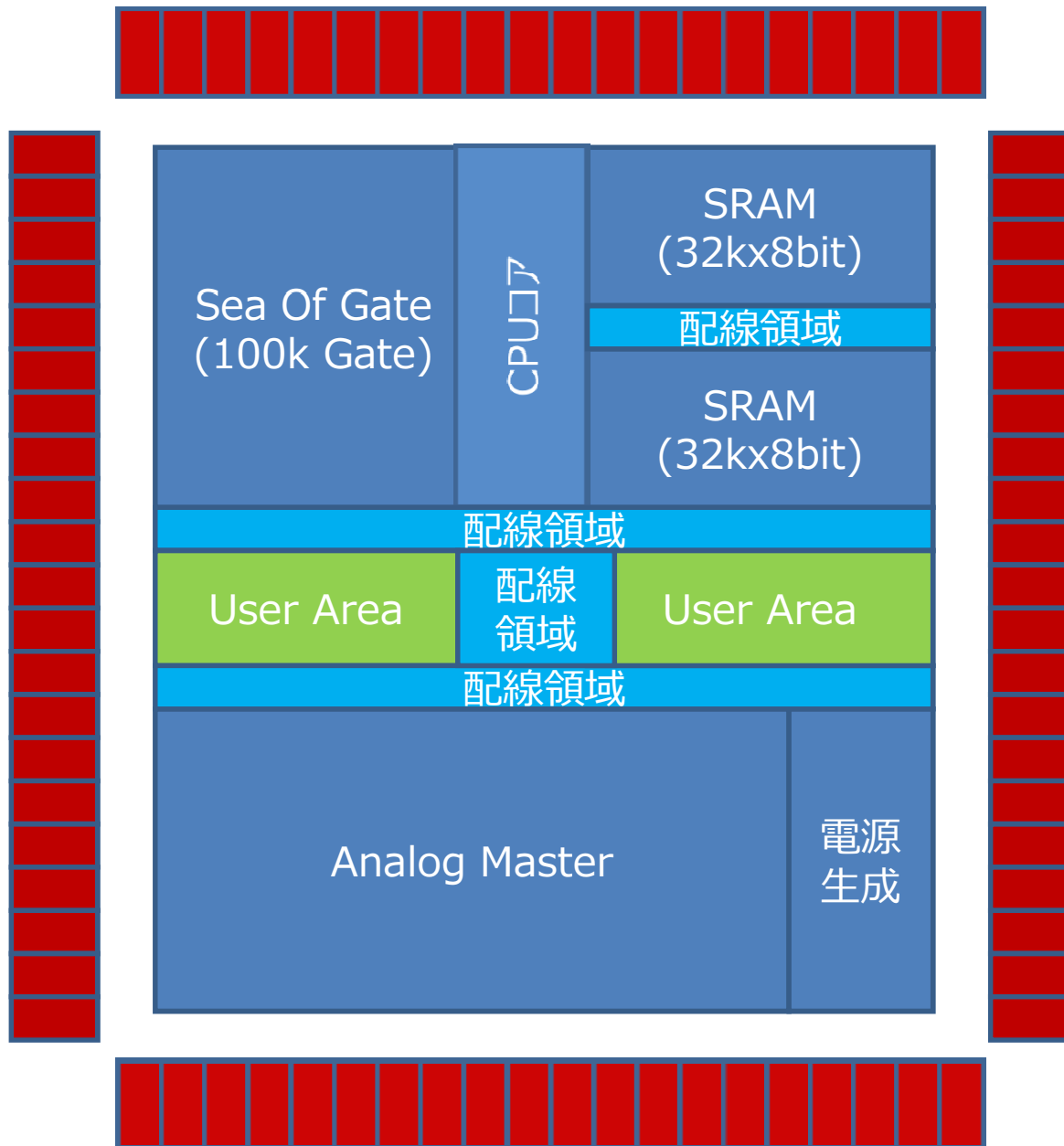
番号	サイズ	内容	機能切り替え Layer
①	3.0mm×3.0mm	Rail-to-Rail-OPAMP	M3のみ(ゲイン調整)
②	3.0mm×3.0mm	8bit マイコン	M2,M3で切り替え
③	3.0mm×3.0mm	ロジック・アレイ	M2,M3で切り替え
④	2.8mm×3.0mm	10bit DAC(R2-R) with 6ch	M3のみ(Ch選択)
⑤	3.0mm×3.0mm	アナログマスター (OPAMP/COMP/LPF)	M3のみ (ゲイン、LPF周波数)

※ M1: Metal 1層目、M2: Metal 2層目、M3: Metal 3層目

CPUとアナログIP、ロジック回路を配線工程のみで接続しICを試作する



シャトルマスターチップ案（検討中）



【仕様概要】

CPU:未定

SRAM:2kx8x2block(プログラムとデータ)

Sea Of Gate:100k Gate

Analog :Amp, Comparator, ADC, DAC

電源生成 : 3.3V, 1.8V, 1.2Vに対応

IO数:92本(programmable IO)

電源端子 : 固定

チップサイズ : 4mm x 4mm固定

【IPについて】

CPUコア : 未定

SRAM:SRAMコンパイラーで生成可能

Sea of Gate:ロジック・リサーチで準備中

Analog:ロジック・リサーチで準備中

電源生成 : ロジック・リサーチで準備中

このプロジェクトに興味がある人は、
tsuchiya@logic-research.co.jpに
連絡ください。

