

# オープンソースEDAを活用した LSI開発環境の開拓

日置雅和、堀洋平、片下敏宏、更田裕司、秋田一平

先端半導体研究センター  
集積回路設計研究チーム

## 自己紹介

- 2003年3月 東北大学大学院工学研究科電子工学専攻修了 舩岡富士雄先生（フラッシュメモリの発明者@東芝）  
SGT（GAA）型フラッシュメモリセルの設計と形状効果の研究、博士（工学）
- 2003年4月～ 産総研入所 電力再構成可能FPGAであるFlex Power FPGAのアーキテクチャ設計、  
回路設計、チップ評価
- 2012年11月～2013年11月 経産省情報通信機器課（現情報産業課） 出向
- 2013年～2018年 SOTB（Silicon On Thin Buried oxide）版Flex Power FPGA設計・評価、  
超伝導量子ビット設計環境整備・簡易PDK作成・設計
- 2018年9月～ 産総研・東大 AIチップデザインオープンイノベーションラボラトリ チーム長
- 2020年4月～ デバイス技術研究部門先端集積回路研究グループ グループ長
- 2023年10月～ 先端半導体研究センター（SFRC）集積回路設計研究チーム チーム長

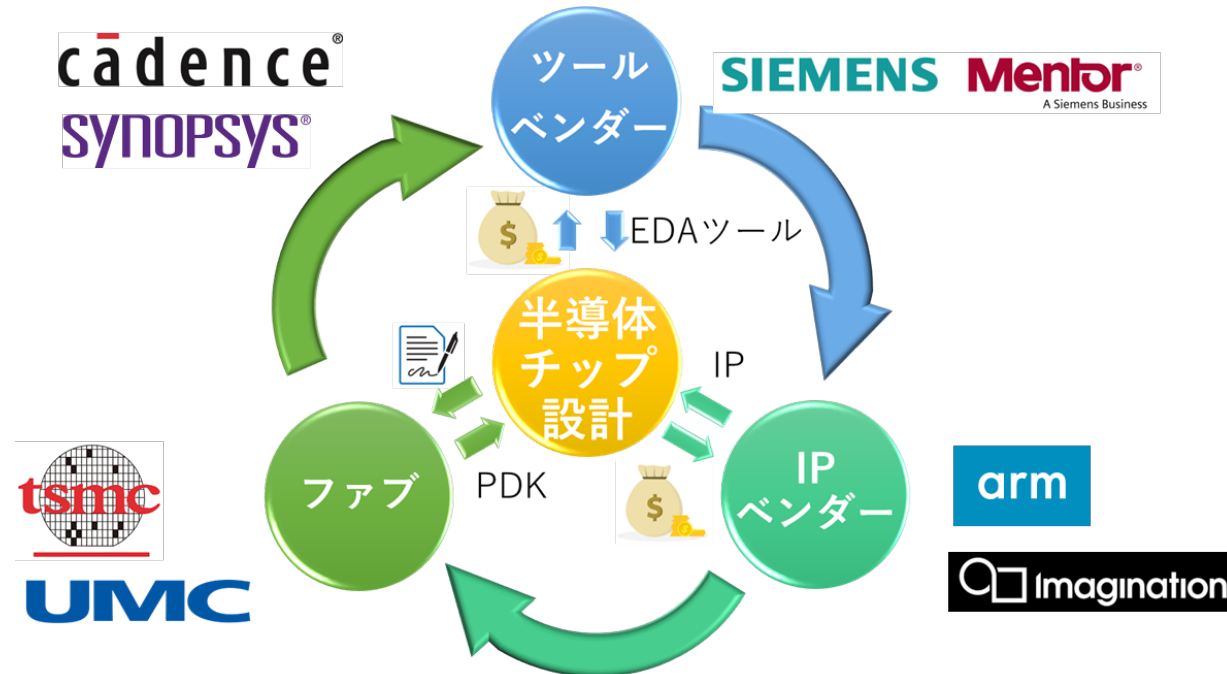
# LSI開発のエコシステム

✓現在のLSI開発：既存エコシステムの束縛 ⇒ **新規参入者にとって高い参入障壁**

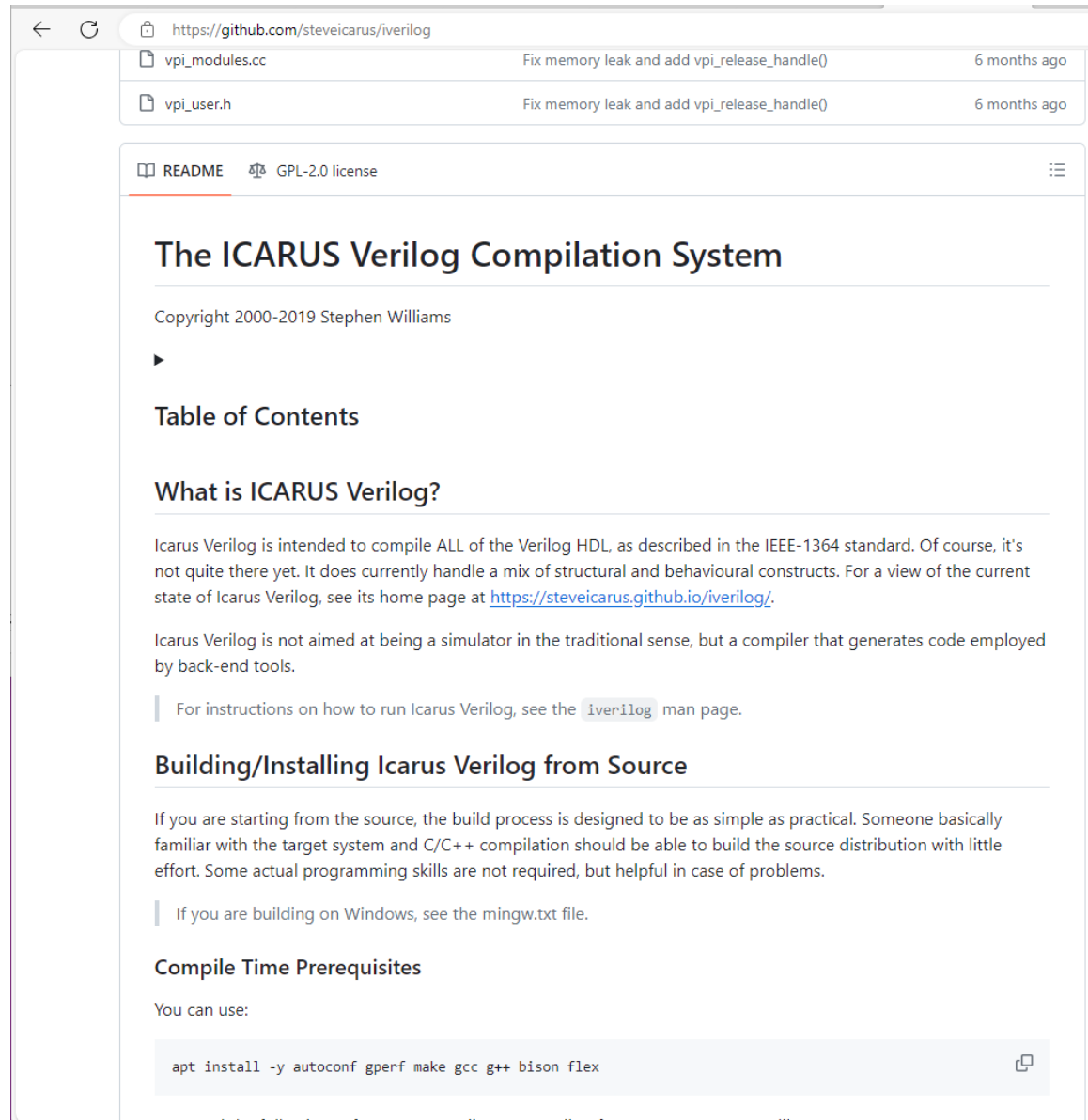
- 高価な**EDAツール**（設計ソフトウェア）
- 高価な**IP**（設計資産：インターフェース、メモリ等）
- 入手に手間のかかる**PDK**（Process Design Kit：デザイン情報）

⇒ どれから梯子をかける？

フリーのEDAツールは昔から脈々と開発されていた



# ICARUS Verilogシミュレータ (1998年～)



The screenshot shows the GitHub repository page for 'steveicarus/iverilog'. The browser address bar shows 'https://github.com/steveicarus/iverilog'. Below the repository name, there are two commit entries for 'vpi\_modules.cc' and 'vpi\_user.h', both dated '6 months ago' with the message 'Fix memory leak and add vpi\_release\_handle()'. The main content is the README file, which includes a 'Table of Contents' and sections for 'What is ICARUS Verilog?' and 'Building/Installing Icarus Verilog from Source'. The 'What is ICARUS Verilog?' section states that Icarus Verilog is intended to compile ALL of the Verilog HDL, as described in the IEEE-1364 standard. The 'Building/Installing Icarus Verilog from Source' section provides instructions for building from source, mentioning that the build process is designed to be as simple as practical.

https://github.com/steveicarus/iverilog

vpi\_modules.cc Fix memory leak and add vpi\_release\_handle() 6 months ago

vpi\_user.h Fix memory leak and add vpi\_release\_handle() 6 months ago

README GPL-2.0 license

## The ICARUS Verilog Compilation System

Copyright 2000-2019 Stephen Williams

### Table of Contents

### What is ICARUS Verilog?

Icarus Verilog is intended to compile ALL of the Verilog HDL, as described in the IEEE-1364 standard. Of course, it's not quite there yet. It does currently handle a mix of structural and behavioural constructs. For a view of the current state of Icarus Verilog, see its home page at <https://steveicarus.github.io/iverilog/>.

Icarus Verilog is not aimed at being a simulator in the traditional sense, but a compiler that generates code employed by back-end tools.

For instructions on how to run Icarus Verilog, see the `iverilog` man page.

### Building/Installing Icarus Verilog from Source

If you are starting from the source, the build process is designed to be as simple as practical. Someone basically familiar with the target system and C/C++ compilation should be able to build the source distribution with little effort. Some actual programming skills are not required, but helpful in case of problems.

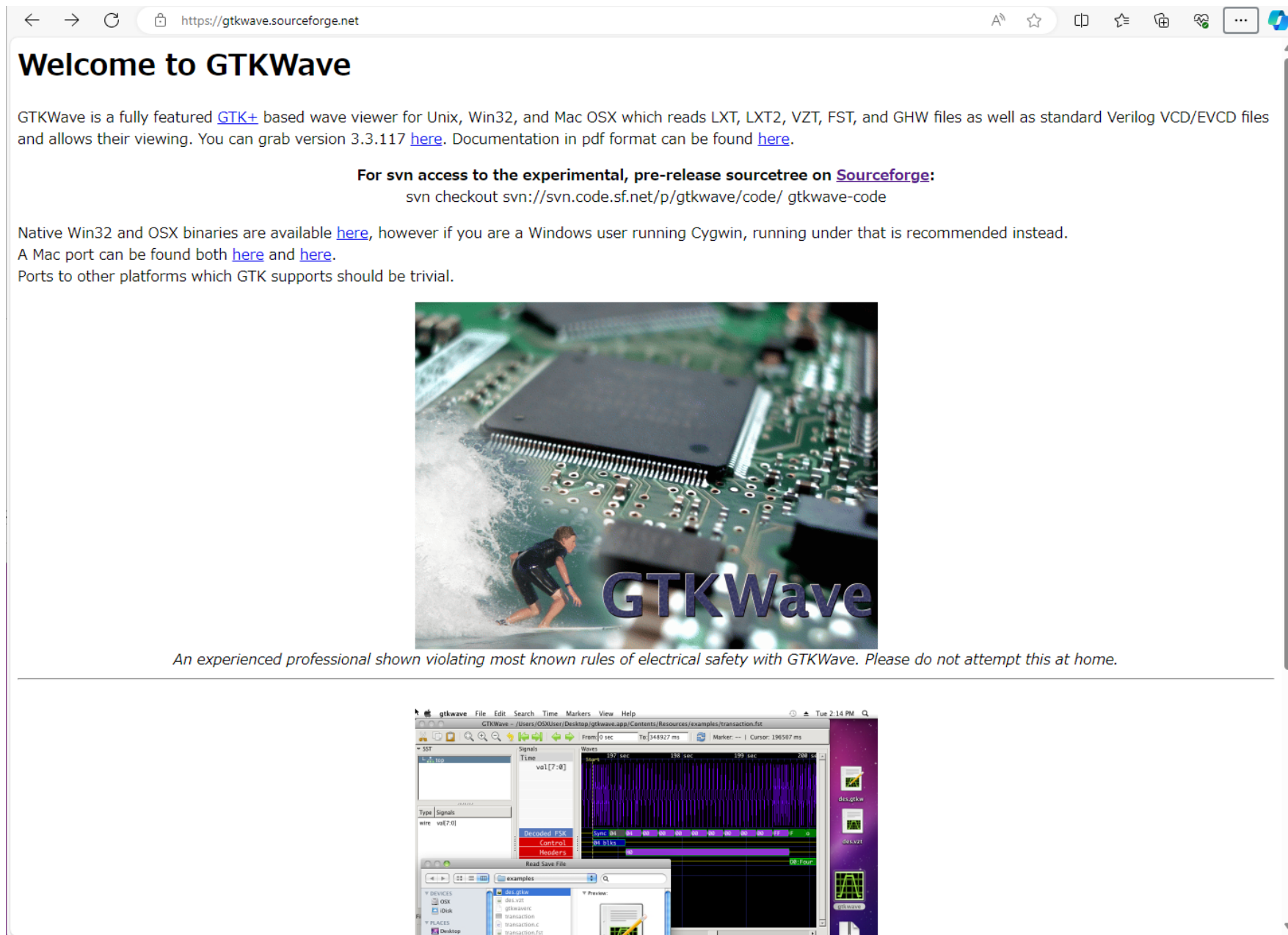
If you are building on Windows, see the mingw.txt file.

### Compile Time Prerequisites

You can use:

```
apt install -y autoconf gperf make gcc g++ bison flex
```

# GTKWave 波形表示 (10年くらい前～)




← → ↻ <https://gtkwave.sourceforge.net> 🔍 ☆ 📄 📌 📁 🌐 ⋮

## Welcome to GTKWave

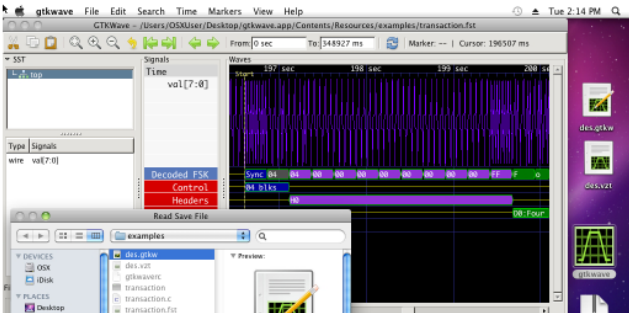
GTKWave is a fully featured [GTK+](#) based wave viewer for Unix, Win32, and Mac OSX which reads LXT, LXT2, VZT, FST, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing. You can grab version 3.3.117 [here](#). Documentation in pdf format can be found [here](#).

**For svn access to the experimental, pre-release sourcetree on Sourceforge:**  
svn checkout svn://svn.code.sf.net/p/gtkwave/code/ gtkwave-code

Native Win32 and OSX binaries are available [here](#), however if you are a Windows user running Cygwin, running under that is recommended instead. A Mac port can be found both [here](#) and [here](#). Ports to other platforms which GTK supports should be trivial.



*An experienced professional shown violating most known rules of electrical safety with GTKWave. Please do not attempt this at home.*

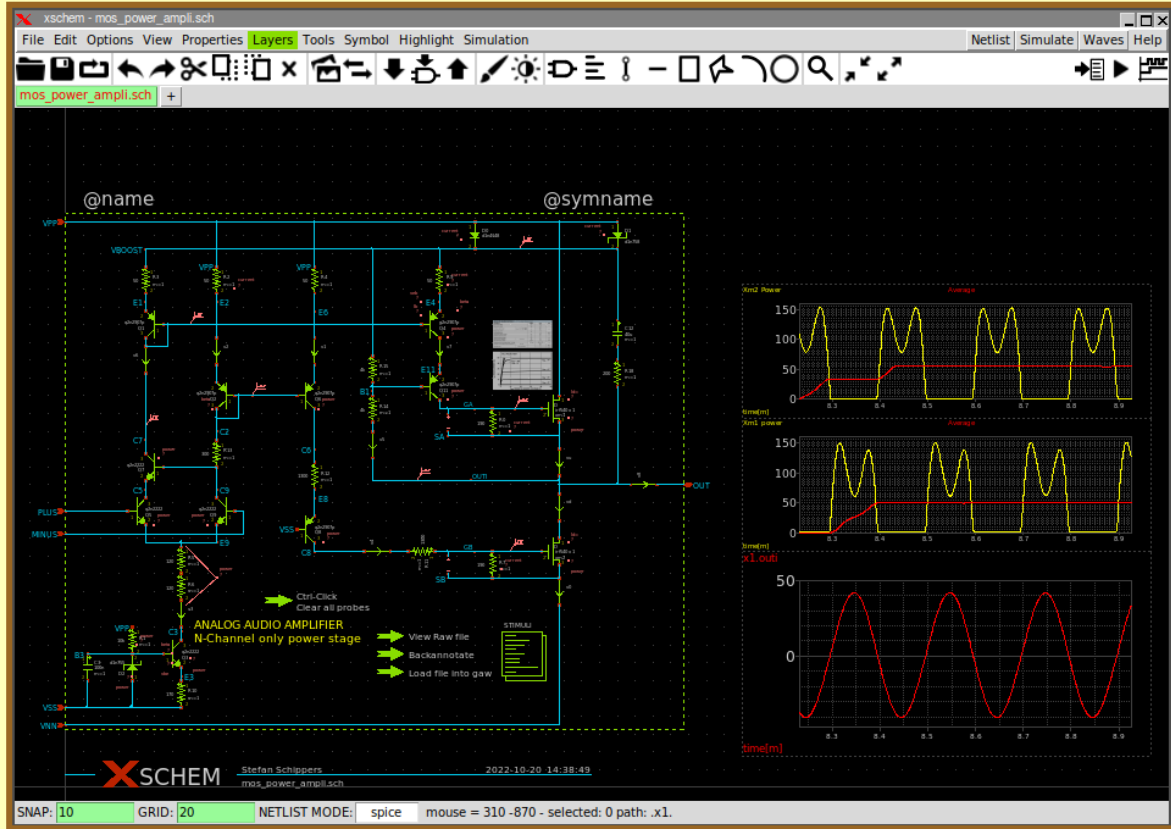


# Xschem 回路図エディタ + ネットリスティング (1998~)

HOME XSCHM MANUAL GOOGLE  
- Solaris sparc  
- Windows (with the cygwin layer and cygwin/Xorg X11 server, plus the tcl/tk toolkit and the -dev libraries)

### Screenshots

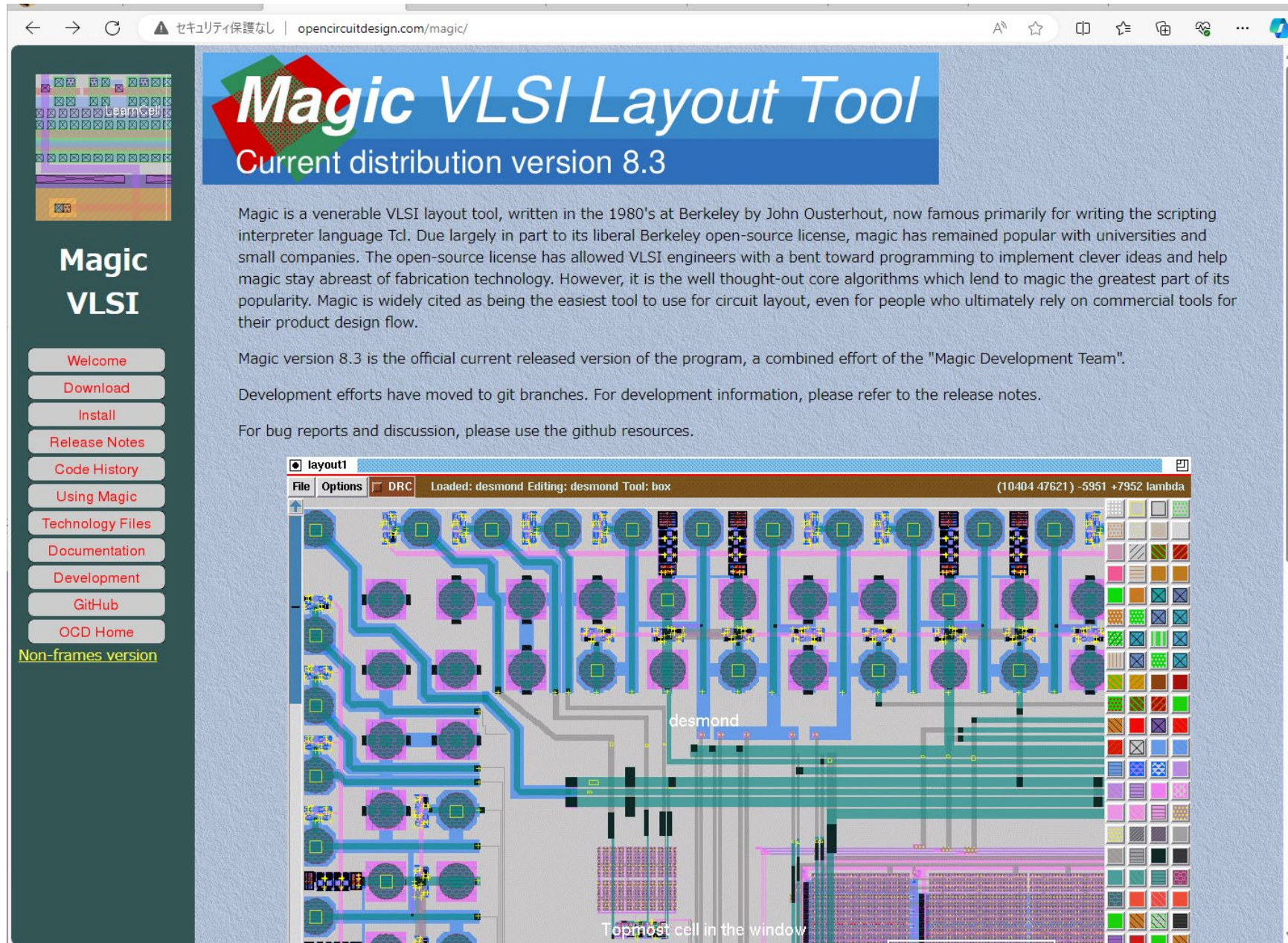
- analog circuit example



The screenshot displays the Xschem interface for a project named 'mos\_power\_ampli.sch'. The main window shows a detailed circuit diagram of an 'ANALOG AUDIO AMPLIFIER N-Channel only power stage'. The circuit includes a VBOOST section, a main amplifier stage with transistors (Q1, Q2, Q3, Q4), and an output stage (Q5, Q6). It features various components like resistors (R1-R10), capacitors (C1-C5), and a transformer (T1). The circuit is powered by VDD, VSS, and VDD5. The output is labeled 'OUT'. The interface includes a menu bar (File, Edit, Options, View, Properties, Layers, Tools, Symbol, Highlight, Simulation), a toolbar, and a status bar at the bottom showing 'SNAP: 10', 'GRID: 20', 'NETLIST MODE: spice', and 'mouse = 310 - 870 - selected: 0 path: .x1'. To the right of the circuit diagram, three simulation waveforms are shown, plotting signal amplitude against time (m) from 0.3 to 0.9. The top waveform shows a complex signal, the middle one shows a similar signal with a different amplitude, and the bottom one shows a clean sine wave. The status bar at the bottom of the screenshot indicates the user is 'Stefan Schippers' and the file is 'mos\_power\_ampli.sch'.

- digital system for VHDL simulation

# Magic レイアウトエディタ (1980's~)



The image shows a screenshot of the Magic VLSI Layout Tool website and a screenshot of the tool's interface. The website header features the title "Magic VLSI Layout Tool" and "Current distribution version 8.3". Below this, there is a paragraph of text describing the tool's history and popularity. A sidebar on the left contains navigation links such as "Welcome", "Download", "Install", "Release Notes", "Code History", "Using Magic", "Technology Files", "Documentation", "Development", "GitHub", "OCD Home", and "Non-frames version". The interface screenshot shows a complex circuit layout with various components and connections, including a menu bar with "File", "Options", and "DRC", and a toolbar with various icons. The text "desmond" is visible in the layout, and "Topmost cell in the window" is written at the bottom of the interface screenshot.

**Magic VLSI Layout Tool**  
Current distribution version 8.3

Magic is a venerable VLSI layout tool, written in the 1980's at Berkeley by John Ousterhout, now famous primarily for writing the scripting interpreter language Tcl. Due largely in part to its liberal Berkeley open-source license, magic has remained popular with universities and small companies. The open-source license has allowed VLSI engineers with a bent toward programming to implement clever ideas and help magic stay abreast of fabrication technology. However, it is the well thought-out core algorithms which lend to magic the greatest part of its popularity. Magic is widely cited as being the easiest tool to use for circuit layout, even for people who ultimately rely on commercial tools for their product design flow.

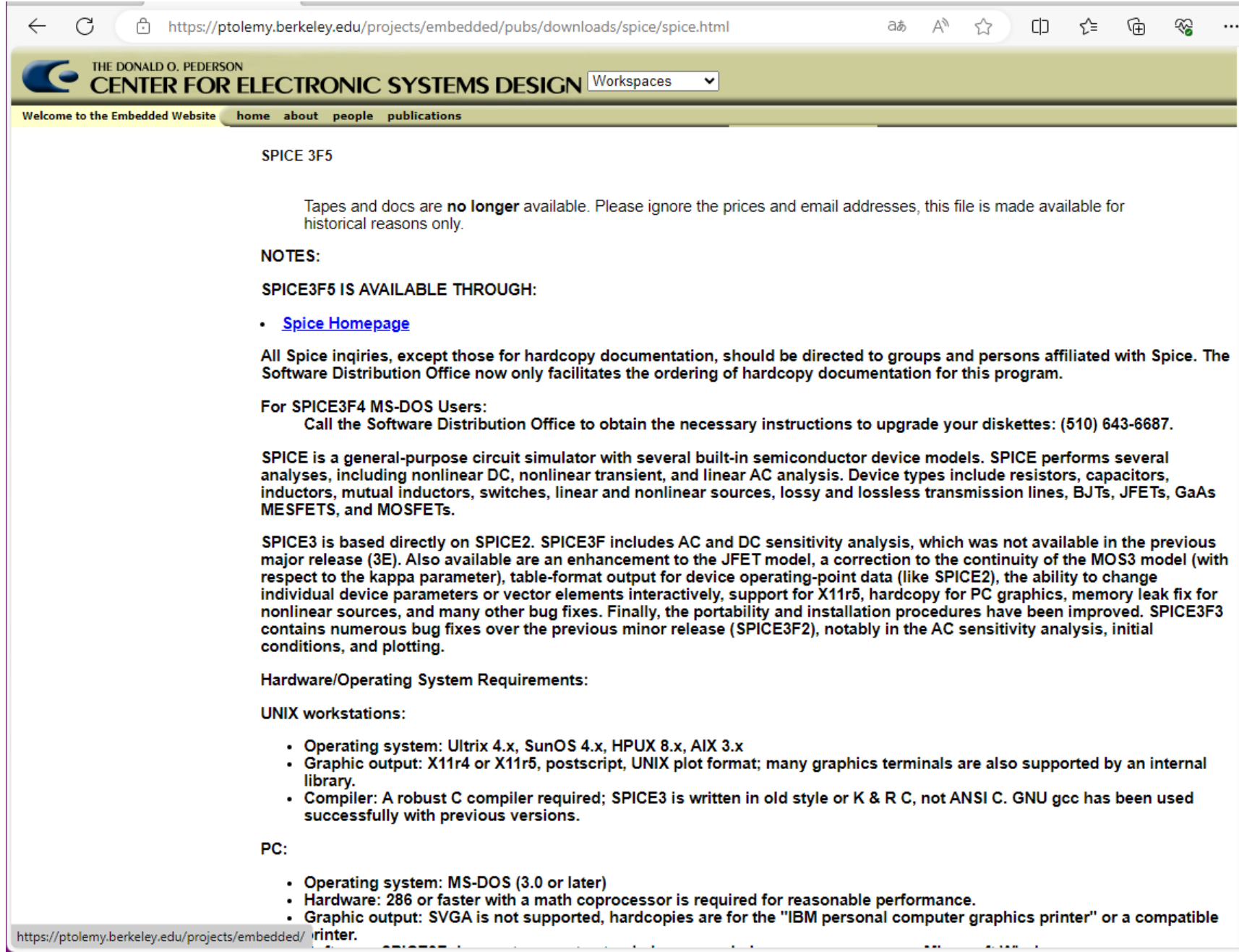
Magic version 8.3 is the official current released version of the program, a combined effort of the "Magic Development Team".

Development efforts have moved to git branches. For development information, please refer to the release notes.

For bug reports and discussion, please use the github resources.

layout1  
File Options DRC Loaded: desmond Editing: desmond Tool: box (10404 47621) -5951 +7952 lambda  
desmond  
Topmost cell in the window

# SPICE3 アナログ回路シミュレータ (1996~)



The screenshot shows a web browser window displaying the SPICE3 website. The browser's address bar shows the URL: <https://ptolemy.berkeley.edu/projects/embedded/pubs/downloads/spice/spice.html>. The website header includes the logo for "THE DONALD O. PEDERSON CENTER FOR ELECTRONIC SYSTEMS DESIGN" and a "Workspaces" dropdown menu. Below the header, there is a navigation bar with links for "home", "about", "people", and "publications". The main content area is titled "SPICE 3F5" and contains the following text:

Tapes and docs are **no longer** available. Please ignore the prices and email addresses, this file is made available for historical reasons only.

**NOTES:**

**SPICE3F5 IS AVAILABLE THROUGH:**

- [Spice Homepage](#)

All Spice inquiries, except those for hardcopy documentation, should be directed to groups and persons affiliated with Spice. The Software Distribution Office now only facilitates the ordering of hardcopy documentation for this program.

For SPICE3F4 MS-DOS Users:  
Call the Software Distribution Office to obtain the necessary instructions to upgrade your diskettes: (510) 643-6687.

SPICE is a general-purpose circuit simulator with several built-in semiconductor device models. SPICE performs several analyses, including nonlinear DC, nonlinear transient, and linear AC analysis. Device types include resistors, capacitors, inductors, mutual inductors, switches, linear and nonlinear sources, lossy and lossless transmission lines, BJTs, JFETs, GaAs MESFETS, and MOSFETs.

SPICE3 is based directly on SPICE2. SPICE3F includes AC and DC sensitivity analysis, which was not available in the previous major release (3E). Also available are an enhancement to the JFET model, a correction to the continuity of the MOS3 model (with respect to the kappa parameter), table-format output for device operating-point data (like SPICE2), the ability to change individual device parameters or vector elements interactively, support for X11r5, hardcopy for PC graphics, memory leak fix for nonlinear sources, and many other bug fixes. Finally, the portability and installation procedures have been improved. SPICE3F3 contains numerous bug fixes over the previous minor release (SPICE3F2), notably in the AC sensitivity analysis, initial conditions, and plotting.

**Hardware/Operating System Requirements:**

**UNIX workstations:**

- Operating system: Ultrix 4.x, SunOS 4.x, HPUX 8.x, AIX 3.x
- Graphic output: X11r4 or X11r5, postscript, UNIX plot format; many graphics terminals are also supported by an internal library.
- Compiler: A robust C compiler required; SPICE3 is written in old style or K & R C, not ANSI C. GNU gcc has been used successfully with previous versions.

**PC:**

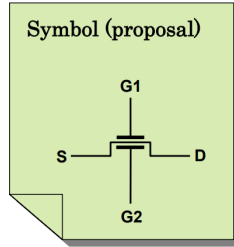
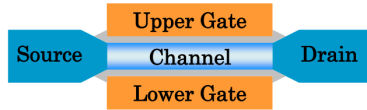
- Operating system: MS-DOS (3.0 or later)
- Hardware: 286 or faster with a math coprocessor is required for reasonable performance.
- Graphic output: SVGA is not supported, hardcopies are for the "IBM personal computer graphics printer" or a compatible printer.

# 過去の経験 1

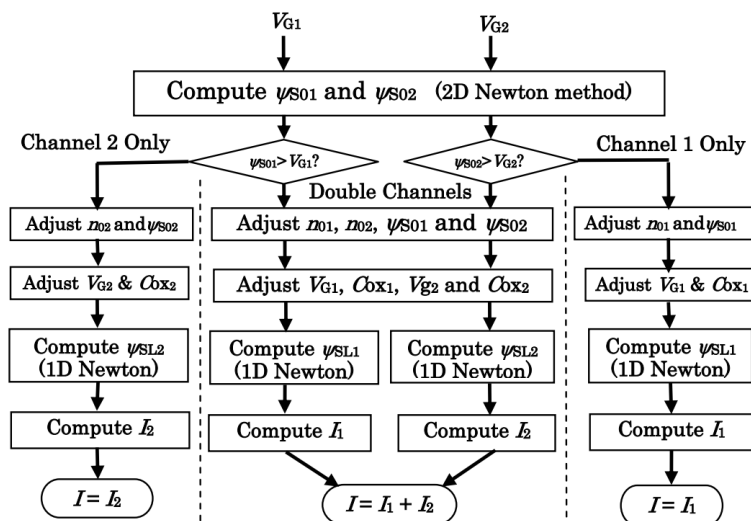
## XMOS(double gate MOS)モデルをspice3f5に組み込む@2004

### DG MOSFET

Two gates sandwich the Si channel.  
The additional gate can be used as  
- the gate electrically tied with the other, or  
- another signal input, or  
- an alternative for the body bias electrode.



In all usage, **minimum short-channel effect** is distinct.  
When used as electrically tied gates (as FinFETs),  
**high current drivability** and **ideal S factor** are prominent.  
When used as an alternative for the body bias electrode,  
high frequency body bias, aggressive forward body bias  
and other sophisticated body bias techniques can be easily achieved.  
Predominant disadvantage is **difficulty of manufacturing**,  
now overcome in research level, but  
long way to go in LSI or production level.



```

[Macintosh:] hkoike% spice3f5f1:xmos/obj/bin/spice3 xmosinv.cir
Program: Spice, version: 3F5
Date built: Wed Mar 3 19:11:30 JST 2004

Type "help" for more information, "quit" to leave.

Circuit: Variable Threshold Voltage X MOS Inverter

Spice 1 -> run
          1 is selected for VERSION. (default)
          1 is selected for VERSION. (default)
Spice 2 -> plot v(out)
Spice 3 -> plot -vdd#branch
Spice 4 -> |

[Macintosh:] hkoike% more xmosinv.cir
Variable Threshold Voltage X MOS Inverter
* commands: run -> plot v(out) -> plot vdd#branch

.DC VIN 0.0 1.0 0.01 VG2 -0.5 0.5 0.1

VIN In 0 DC 0.0
VDD Vdd 0 DC 1.0
VG2 PMOSvg2 Vdd DC 0.5
EG2 NMOSvg2 0 PMOSvg2 VDD -1

MOSP Out In Vdd PMOSvg2 PP L=1u W=1u TEMP=300
MOSN Out In 0 NMOSvg2 NN L=1u W=1u TEMP=300

.MODEL PP PMOS(LEVEL=77 TS=5.e-9 TOX1=2.e-9 TOX2=2.e-9
+ VFB1=0.622801 VFB2=0.622801 U0=0.135 RS=170.0 RD=170.0)
.MODEL NN NMOS(LEVEL=77 TS=5.e-9 TOX1=2.e-9 TOX2=2.e-9
+ VFB1=-0.622801 VFB2=-0.622801 U0=0.135 RS=170.0 RD=170.0)

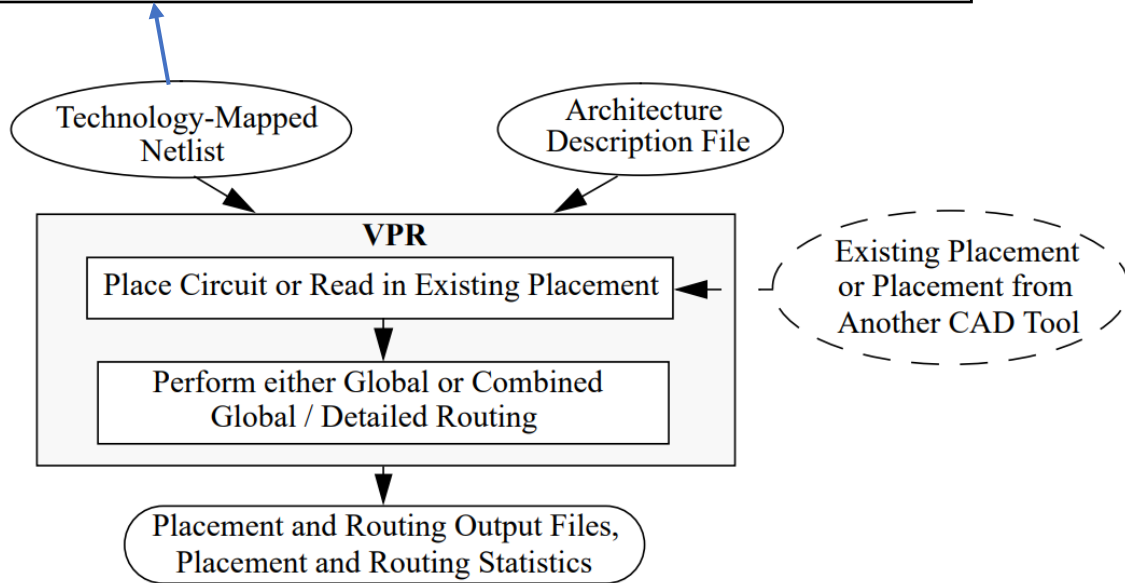
.END
[Macintosh:] hkoike% |
  
```

Ref. 中川他@AIST、nanotech2004

## 過去の経験 2

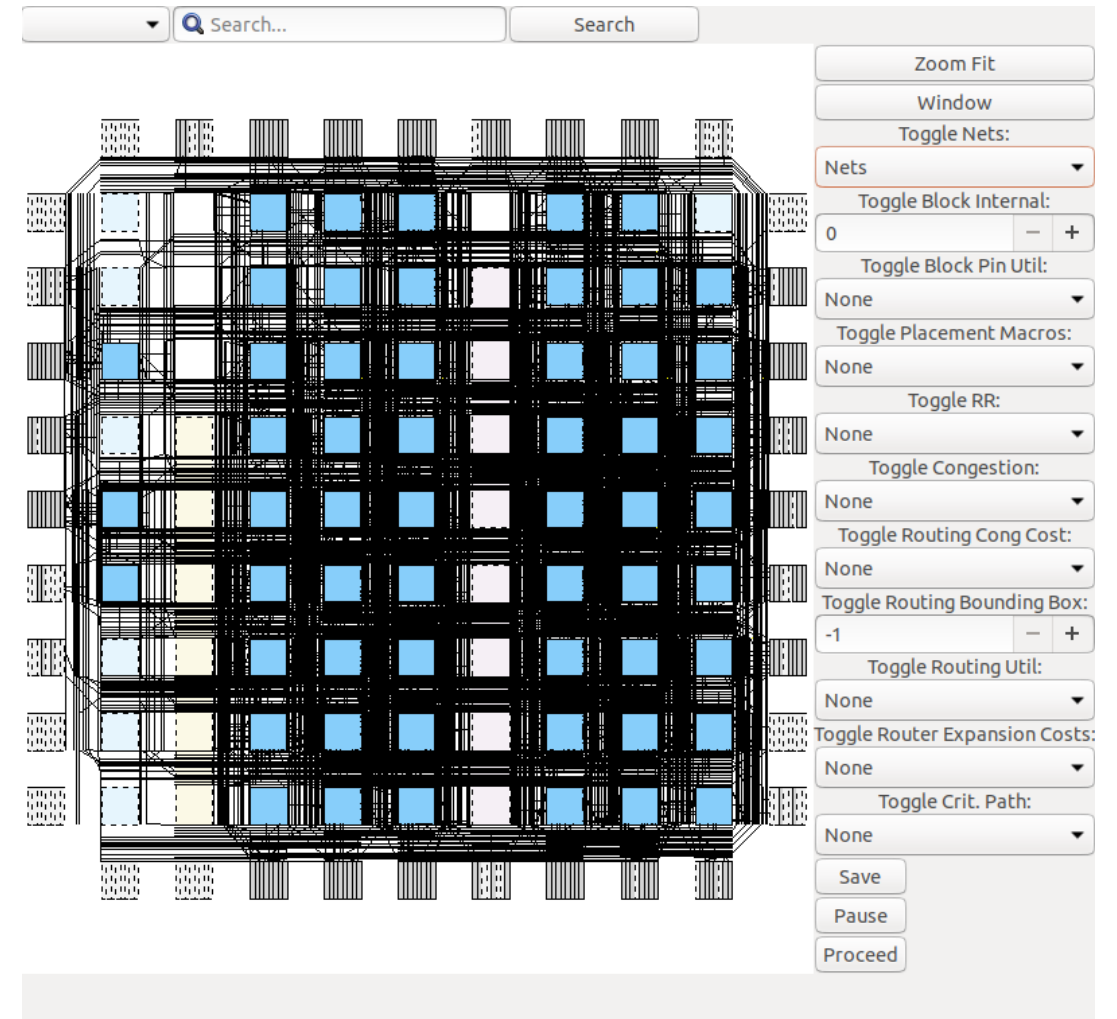
# VPR: Versatile Place and Route @ Toronto University (1997~2011?)

### BLIF: Berkeley Logic Interchange Format



ビットストリームは吐かない

オープンソースEDAツールでアカデミアにおけるFPGA研究を加速

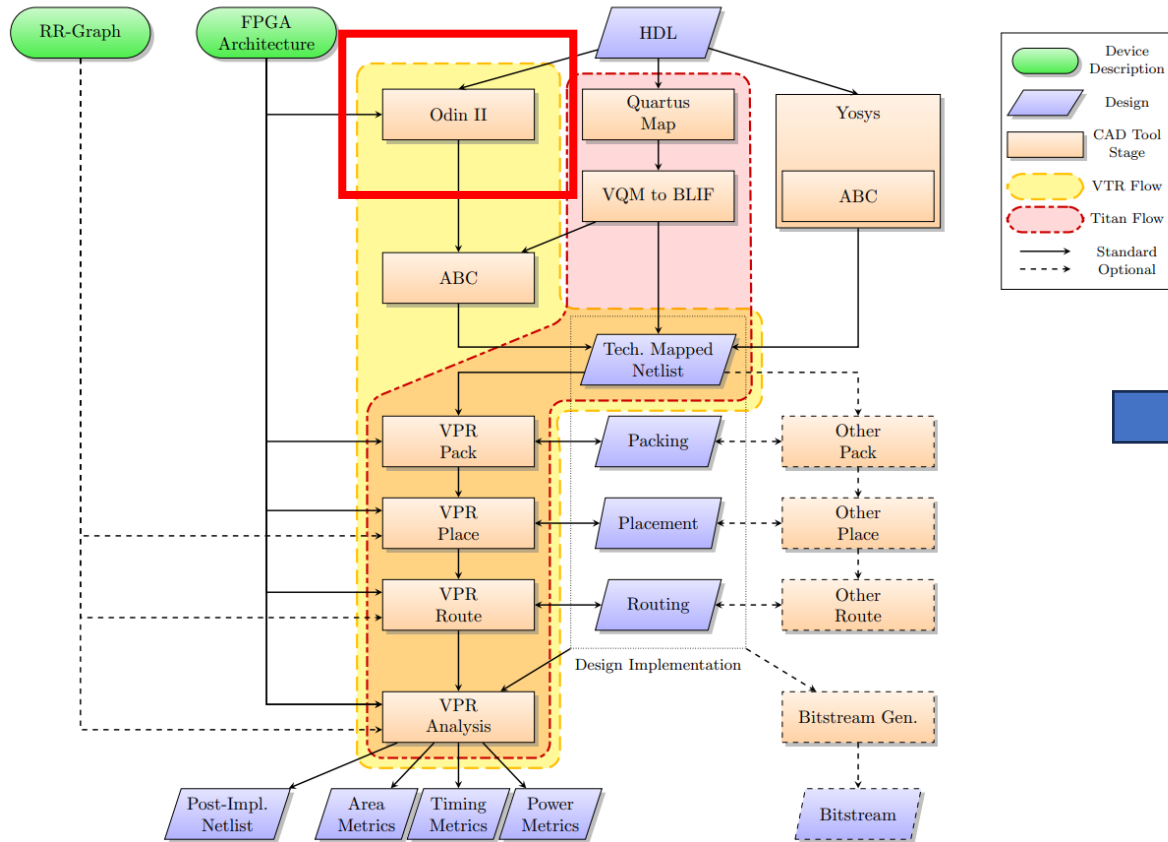


# FPGA用CAD VPRの進歩

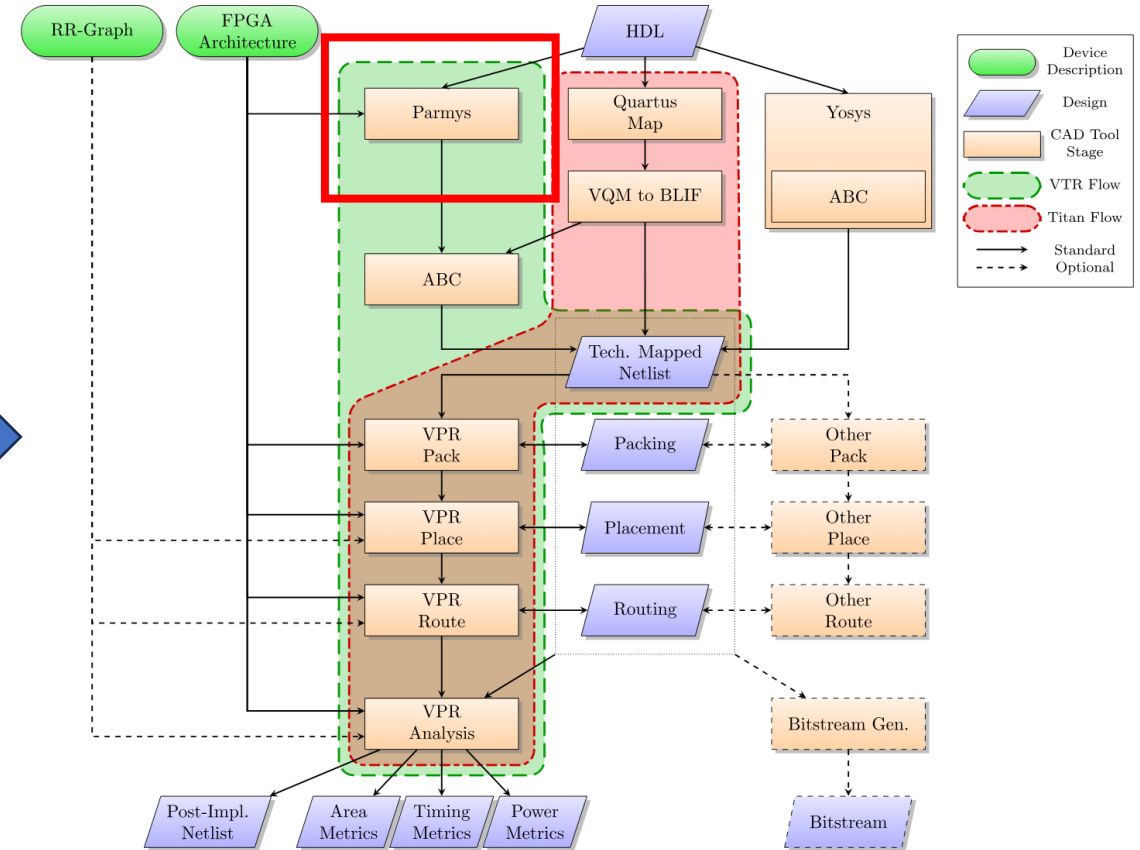
## VTR: Verilog to Routing @ Toronto University (2012~)

Input: BLIF -> Verilog

Older version



Latest version (VTR8)



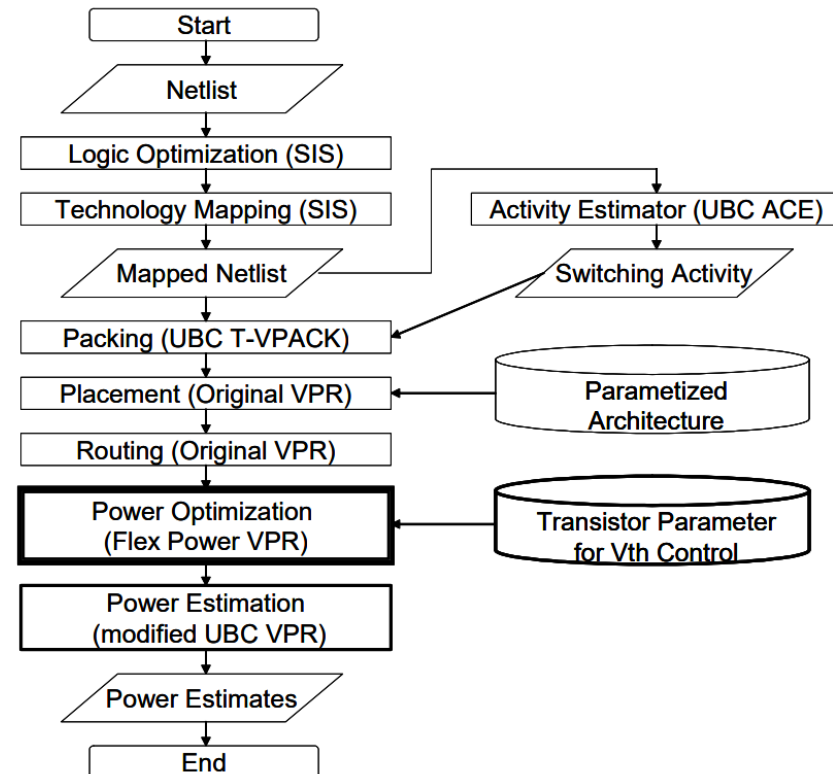
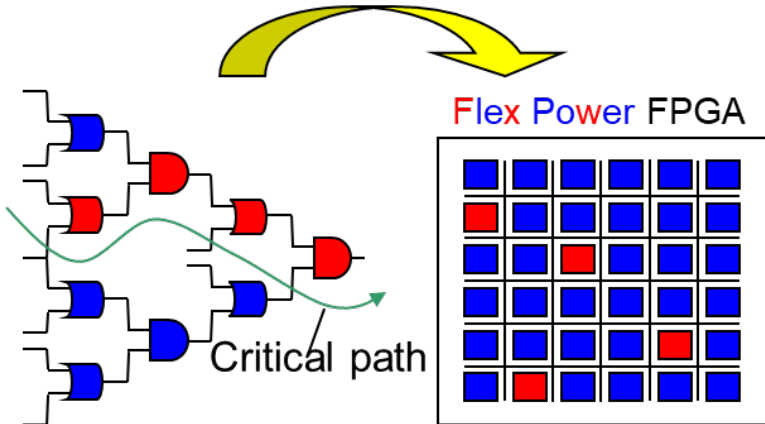
# 過去の経験 3

## Flex Power FPGA用のCADフローを開発する (Ref. 河並他@AIST、IEICE2004)

- 電力を再構成可能なFPGA：Flex Power FPGAの研究開発
- Toronto大のVPRにVT Mapperを追加
- タイミングスラックをパワーに変換

Circuit Reconfigurability + *Power Reconfigurability*

Circuit Mapping + *Power Mapping*

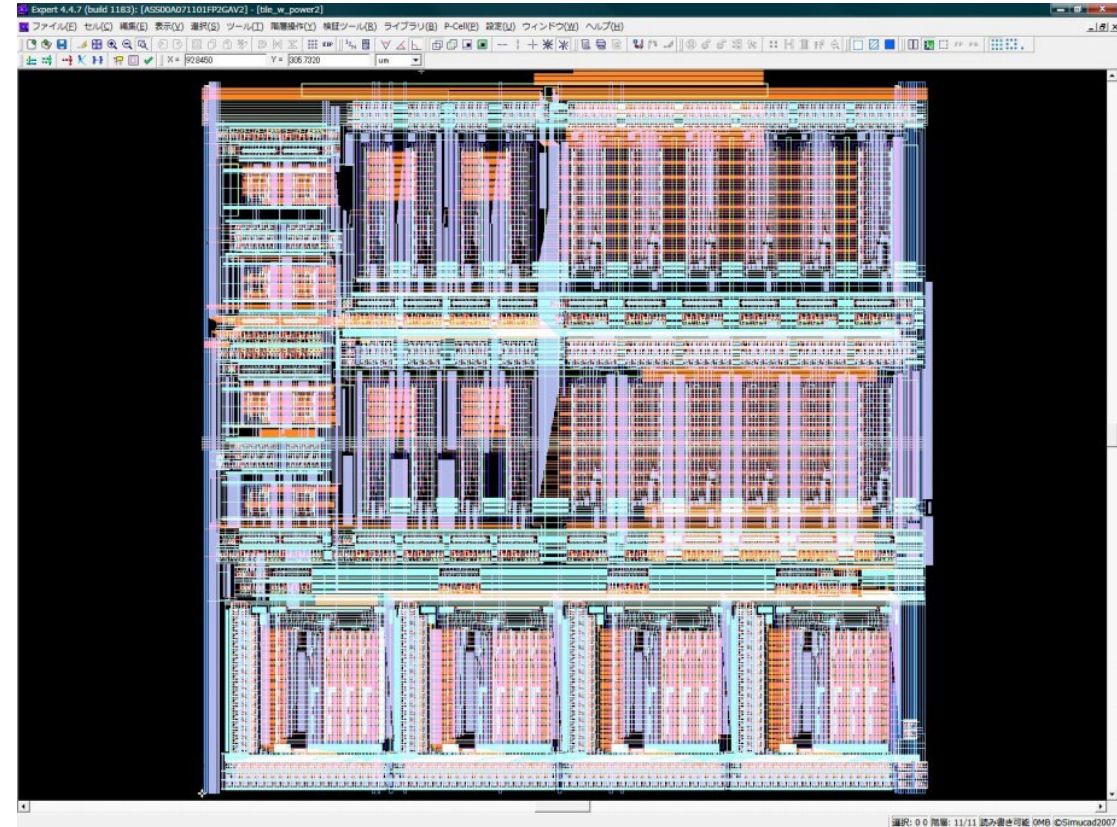
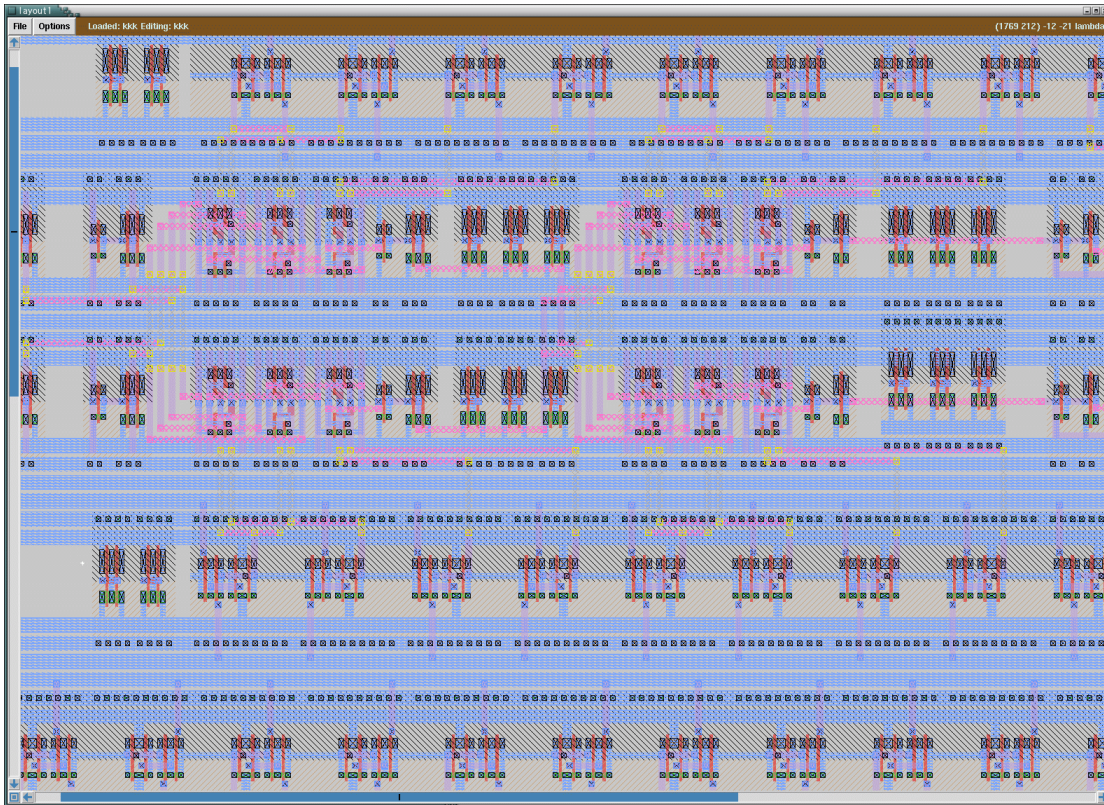


# 過去の経験 4

## オープンソースレイアウトエディタMagicを使って Flex Power FPGAレイアウトを予備評価した@2007

Magicを使った面積評価

商用レイアウトツールで設計



商用ツールへ移行  
理由：ツール向けPDKがあった（TSMC90nm）

# (オープンソースではないですが) Flex Power FPGA CADの刷新

## Flex Power FPGA CAD Flow

•Original Flex Power FPGA CAD tools are newly developed.

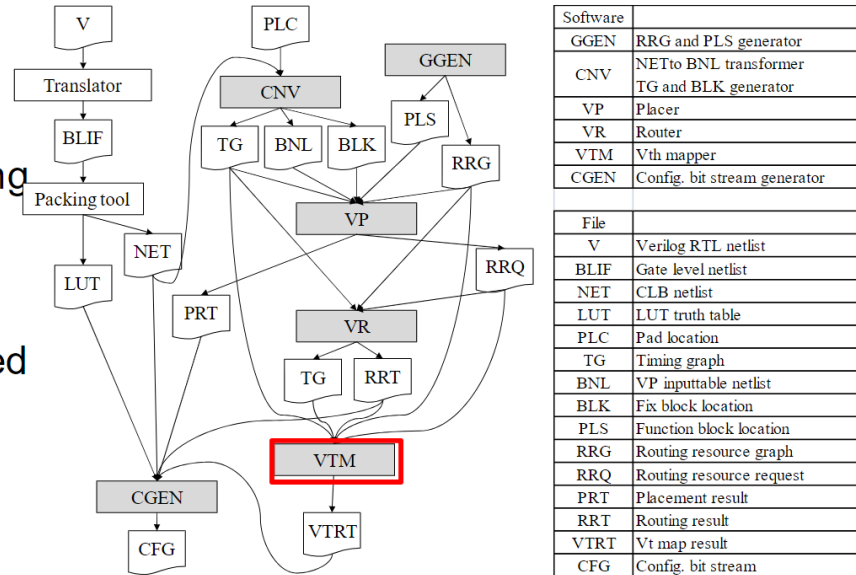
•VTM automatically maps Vt according to the slack analysis

•Successfully Formal-Verified using modest-size application such as microprocessor core

•Interfaces to the standard commercial CAD tools are included

•Why not VPR (4.3)?

- Unidirectional Wire
- 3D Extensibility
- e.t.c.



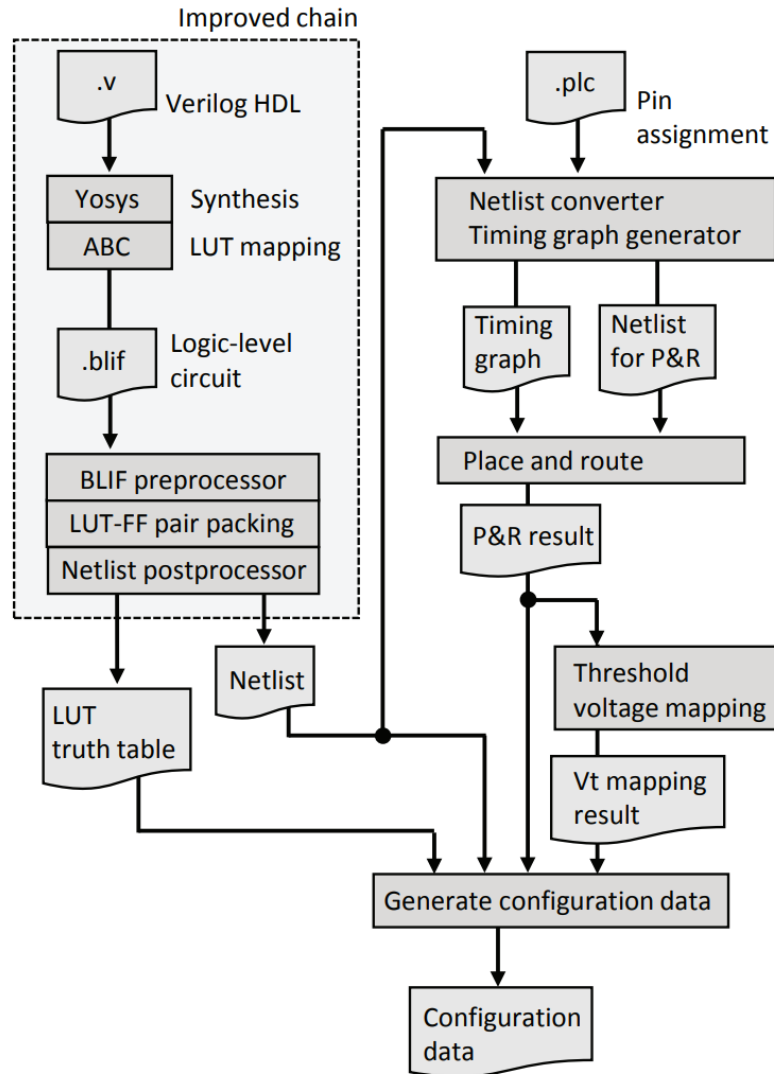
• Flex Power FPGAアーキテクチャの探索を可能とするCADを新たに開発

• Verilog to BLIFはAlteraのQuartusを利用

• ビットストリームまで出力可能

• 実チップが完全動作

# (オープンソースではないですが) Flex Power FPGA CADの改良 (Ref. 片下他、IEICE論文誌、2017)

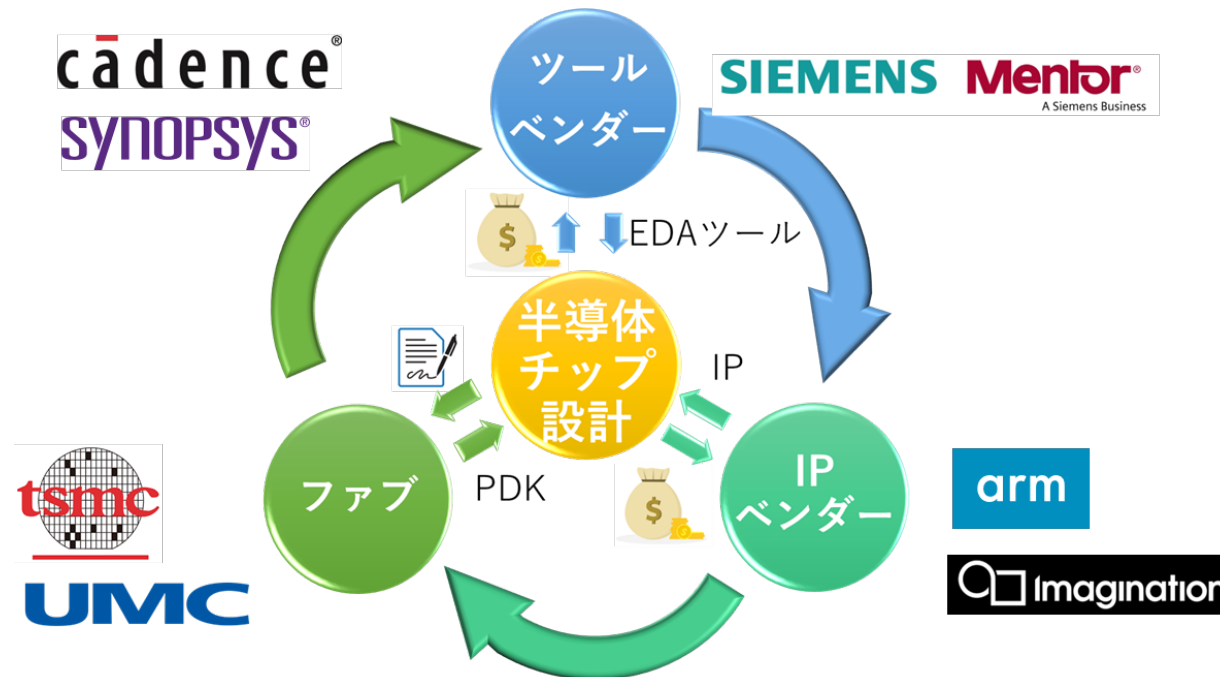


- Verilog to BLIFにオープンソースツールを利用
- 複数のファイルの入力に対応
- モジュール記述ファイルの生成とLUTマッピング情報を生成するためのポストプロセッサ

## (再掲) LSI開発のエコシステム

- ✓ 現在のLSI開発：既存エコシステムの束縛 ⇒ **新規参入者にとって高い参入障壁**
  - 高価な**EDAツール**（設計ソフトウェア）
  - 高価な**IP**（設計資産：インターフェース、メモリ等）
  - 入手に手間のかかる**PDK**（Process Design Kit：デザイン情報）

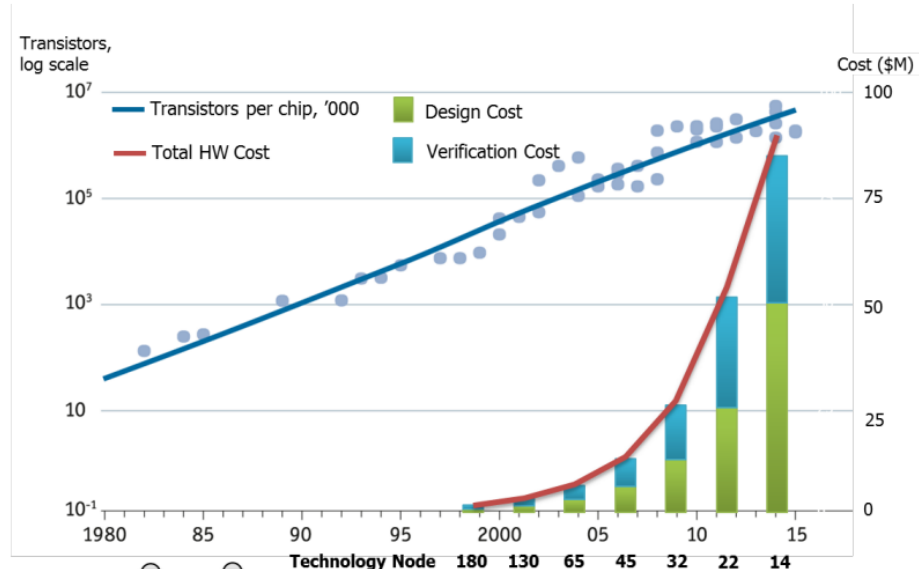
⇒ **ハードウェアイノベーションのバリアを下げるための試み@USA**



# ERI: Electronics Resurgence Initiative, 2017 IDEA & POSH Program



## Lowering barriers to hardware innovation



MOSIS

**NVIDIA** **Qualcomm**  
**Broadcom** **Xilinx**

Fabless companies

**New procedures for physical design and verification will lower the design barrier, enabling rapid specialization**

**Intelligent Design of Electronic Assets (IDEA)**

- No human in the loop" 24-hour layout generation for mixed signal integrated circuits, systems-in-package, and printed circuit boards. Machine generated layout of electrical circuits and systems

**Posh Open Source Hardware (POSH)**

- An open source System on Chip (SoC) design and verification eco-system that enables cost effective design of ultra-complex SoCs.

The 1980's DARPA MOSIS effort removed fab cost and fab access barriers and launched the fabless industry. The ERI Design effort will address today's design complexity and cost barriers, creating the environment needed for the next wave of US semiconductor innovation.

# IDEA : OpenROADプロジェクト (2018~2023)

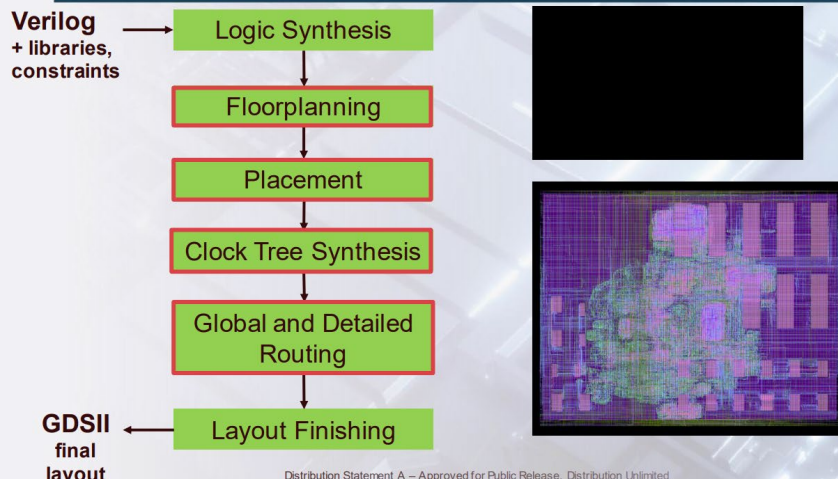
## OPENROAD: NO HUMANS, 24 HOURS

- FOCUS: ULTIMATE ease of use and runtime
- Directly attack the crises of design and innovation
  - **Schedule** barrier: RTL-to-GDS in 24 hours
  - **Expertise** barrier: No-human-in-the-loop, tapeout GDS
  - **Cost** barrier: Open source (and runs in 24 hours)
- Unleash system innovation and design innovation
- Enable tool customization to system, application needs

Distribution Statement A - Approved for Public Release, Distribution Unlimited

6

## OPENROAD V1.0 IN ACTION

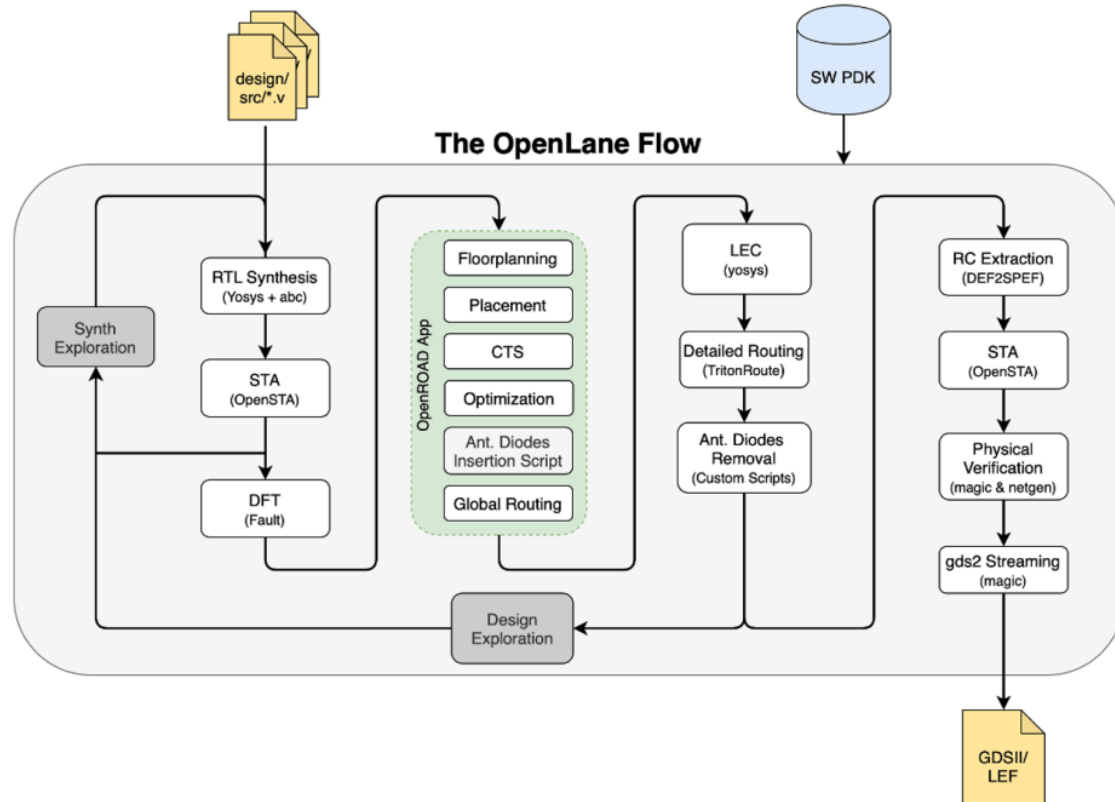


Distribution Statement A - Approved for Public Release, Distribution Unlimited

9

- PL: Prof. ANDREW B. KAHNG@UCSD
- DARPA ERI(Electronic Resurgence Initiative) Summit 2018
  - 24-hours, Non-Human-In-Loop layout design for SOC, Package and PCB with no Power-Performance-Area (PPA) loss, Tapeout-capable tools in source code form -> [Open Source](#)
- Funded by DARPA 2019-2023, Now supported by Precision Innovations
- Used in research and commercial apps.
  - OpenROAD-flow-scripts(OpenROAD)
  - OpenLane(Efabless)
  - Silicon Compiler (ZeroASIC)
  - Hammer (UCB)
  - OpenFASoC (IEDA-FASoC)
- PDK Support
  - Open Source: GF180nm, SKY130nm, FreePDK45nm, ASAP7nm
  - Proprietary: GF55nm, GF12nm, Intel22nm, Intel16nm, TSMC65nm

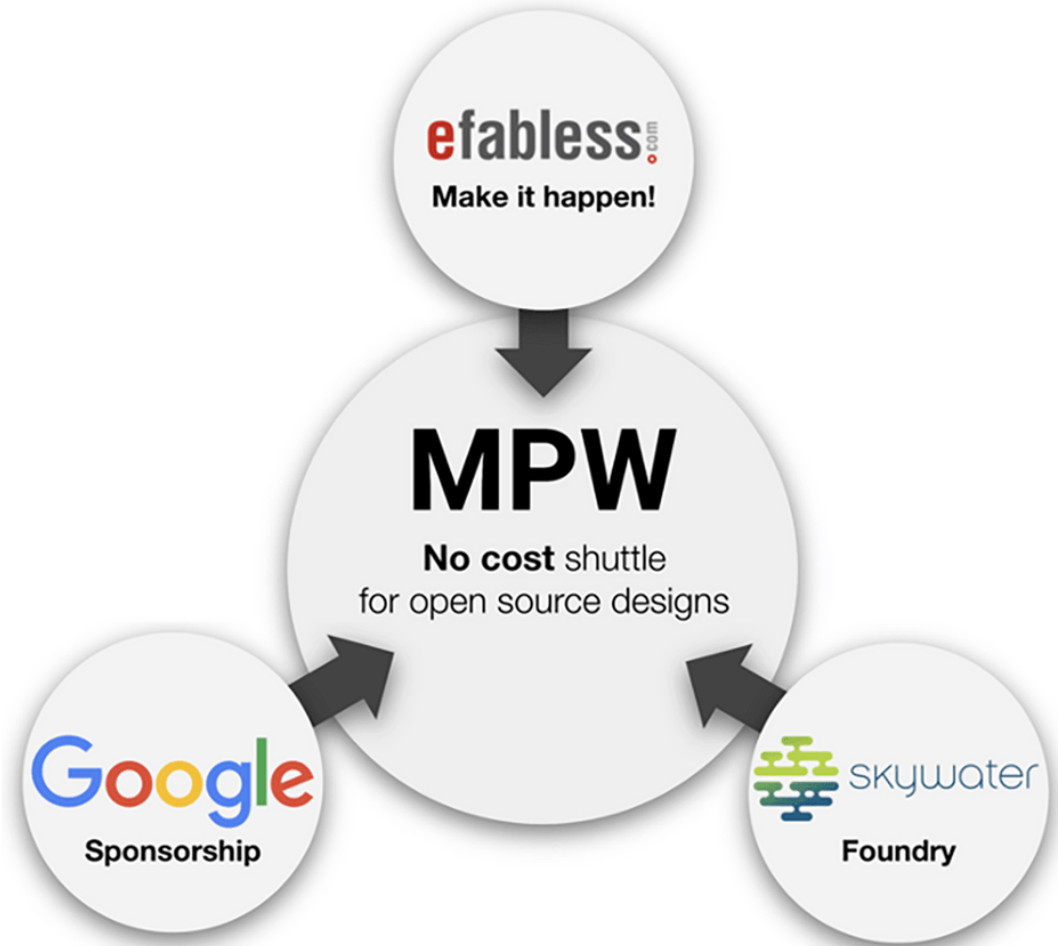
# OpenLaneプロジェクト (2020~)



- A 130nm OpenROAD-based Tapeout-Proven Flow (ICCAD 2020)@efabless
- OpenLane = OpenROAD + Yosys, Magic, Netgen, CVC, Klayout and custom scripts
- PDK Support
  - Open Source: SKY130nm, GF180nm

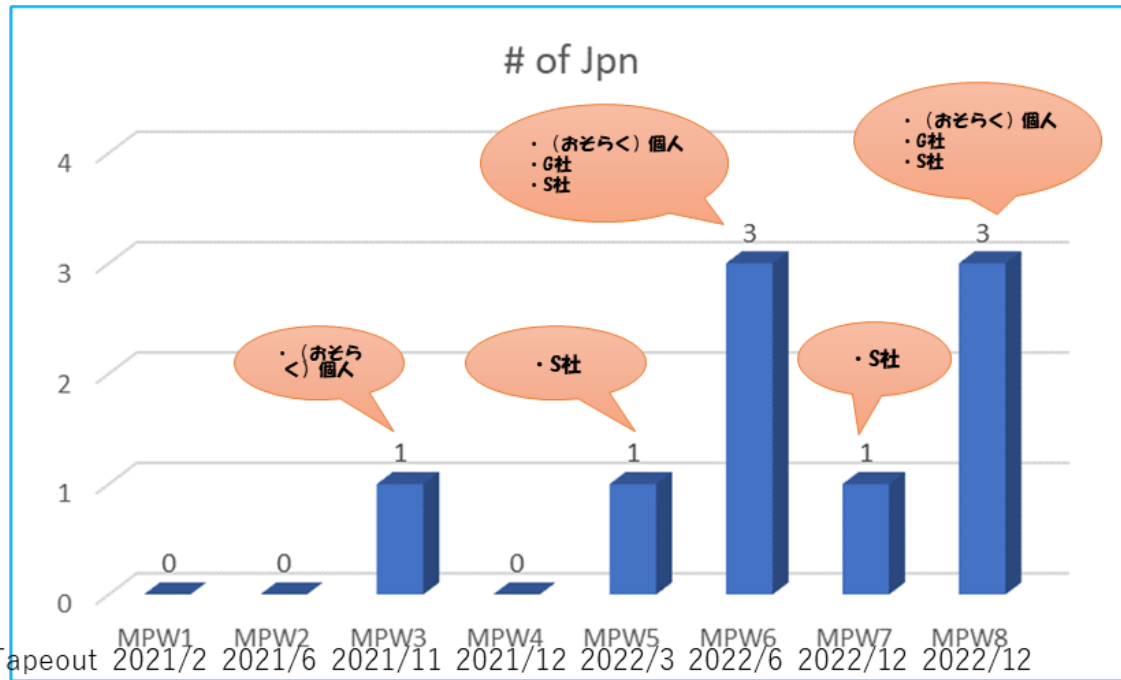
# Open MPW Shuttle Program (2020~)

- Google-efabless-SkyWater Open MPW(:Multi Project Wafer) Shuttle Program from 2020
- 無償のシャトルプログラム
- 有償シャトルプログラムあり (Chiplgnite@efabless)
- 設計IPは公開
- efabless: Open Source EDA 環境(OpenLane)
- SkyWater: 130nmプロセスファウンドリサービス
- Google: スポンサー、オープンソースPDK管理、高位合成ツールチェーン (XLS) 開発など
- Globalfoundriesが2022年に参画



# 国内の利用状況

- ✓ 2020年よりGoogle社, skywater社, efabless社がオープンソースEDA環境・PDK/IPを開発し、無償シャトルサービス（情報公開が原則）を開始。全世界で**600件以上**の利用実績。
- ✓ 2022年よりGlobalFoundriesが参画
- ✓ 日本国内からの利用例は少なかった（**10件未満**）
- ✓ **設計に係る情報や知見が足りない？**

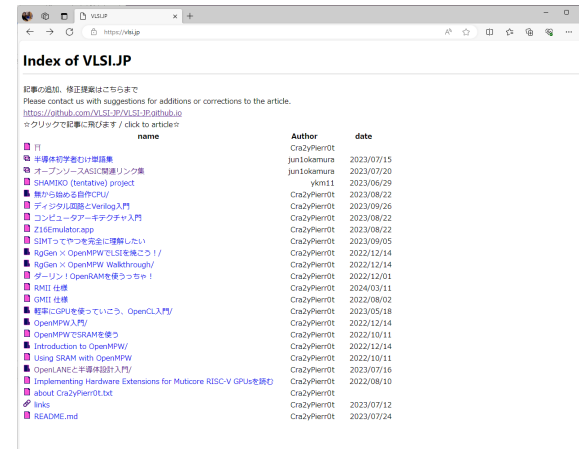
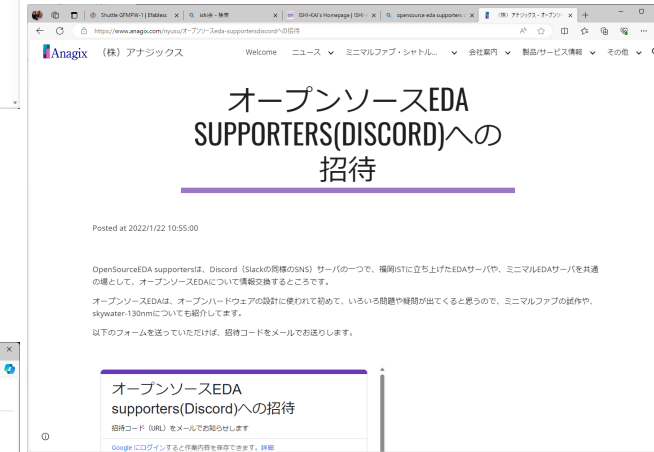
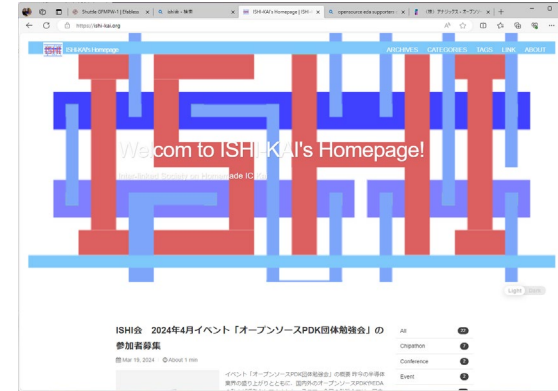


直近のGFMPW-1の日本エントリー：10件⇒TOフェーズ：6件！！

Project Name	Type	Lead	Country	Phase 1	Phase 2	Phase 3	Count
prova	N/A	Gian Domenico Licciardo	Italy	No	N/A	N/A	0
rvcore_chip2	Digital	Kenji Kise	Japan	No	Pass	Pass	1
test	Digital	Miho Yamada	Japan	No	Fail	N/A	2
omotya	Digital	Riku Anan	Japan	Yes	Pass	Pass	3
xls-group-tapeout	Digital	Johan Euphrosine	Japan	No	Fail	N/A	1
micro_irritating_maze	Digital	Noritsuna Imamura	Japan	Yes	Pass	Pass	29
ISHI-KAI_Multiple_Projects_OpenGFMPW-1	Digital	KAI ISHI	Japan	No	Pass	Pass	9
my_gf180	Digital	Riku Anan	Japan	Yes	Pass	Pass	3
JSpi	Digital	Herbert T	Japan	No	Pass	N/A	5
ishikai-gds-test-homelith	Digital	Toru Homemoto	Japan	No	Pass	N/A	4
gitifu_check	Digital	Gitifu	Japan	No	Pass	Pass	1
Oahelia eFPGA rerun	Digital	Faor Lukvanchenko	Kazakhstan	Yes	Fail	Pass	2

# 国内コミュニティ

- ISHI会(website,discord)
- Open Source EDA supporters(discord)
- VLSI.jp(website)
- 個人ブログの記事
- AIST Solutions AI・半導体プロデュース事業部 (slack)



# AIST所内プロジェクトの立ち上げ（2023/6～）

目的：国内のオープンソースEDA/PDKの利用普及に向けた開拓を産総研 がも 担う

## 研究項目1. オープンソースEDAの利用手法の確立とチップ作成

- オープンソースEDAツールであるOpenLaneの利用環境構築と利用技術の確立
- efabless社の商用シャトル等を用いたLSIチップ試作・機能検証・評価

## 研究項目2. チップセキュリティ技術を題材としたオリジナルチップ作成の試行

- 開発実績のある暗号・符号回路IPと新規開発の簡易RISC-Vコア・PUF回路を組み合わせたチップの作製・評価
- オープンソースEDA を用いたオリジナルセキュリティSoCの開発

## 研究項目3. オープンEDA/PDKを活用したアナログ回路自動設計ツールの開発

- 知識のない人でも簡単にアナログ回路を作成できる自動設計ツール開発

- ✓ 所内LSI研究者の英知を結集
- ✓ 産総研回路IP※をフル活用



### ※代表的な産総研の回路IP

- PUF回路（NEDOプロ@2017-2021）
- 暗号回路（科研費等）
- モーションセンサー（Sensors 2021, Q1）
- 量子ビット読み出し回路（VLSI2022、プレス発表）
- 磁気センサー回路（ISSCC2022、プレス発表）

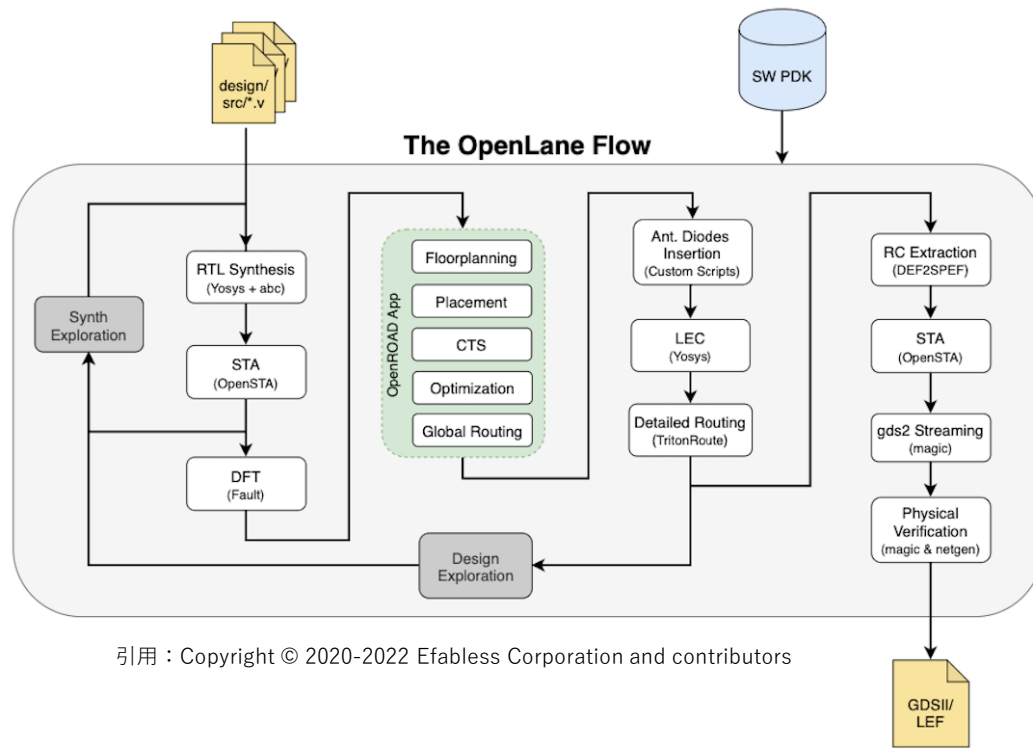
# 研究項目①：オープンソースEDAの利用手法の確立とチップ作成

## • 目的

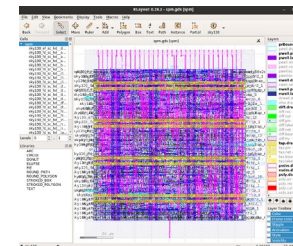
オープンソースEDAの自動化ツール**OpenLane**の利用環境を構築し、RTL設計からGDSIIデータ出力までの**利用技術**を確立する。商用シャトル等を用いて**実際にLSIチップ**を作製し、その機能検証・評価を行う。

## • 2023年度実施内容と目標

**OpenLane環境構築と利用ノウハウの蓄積**、およびGDS-II等の**チップデータの作成**を実施する。（例：シリパラ変換回路など）  
ツールに付属のチップ制御回路 **Caravel** と自作HDLコードを結合し、**シミュレーションによる動作確認**を実施する。



引用：Copyright © 2020-2022 Efabless Corporation and contributors



HDLから生成したチップ作製データの例



オープンソースEDAの利用技術の蓄積

他のHDLからチップ作製データの生成方法は？

チップ作製前の検証方法は？

配置指定や制約設定のほか他のPDK使用など可能か？

生成したデータにより実際にチップを作製できるか？

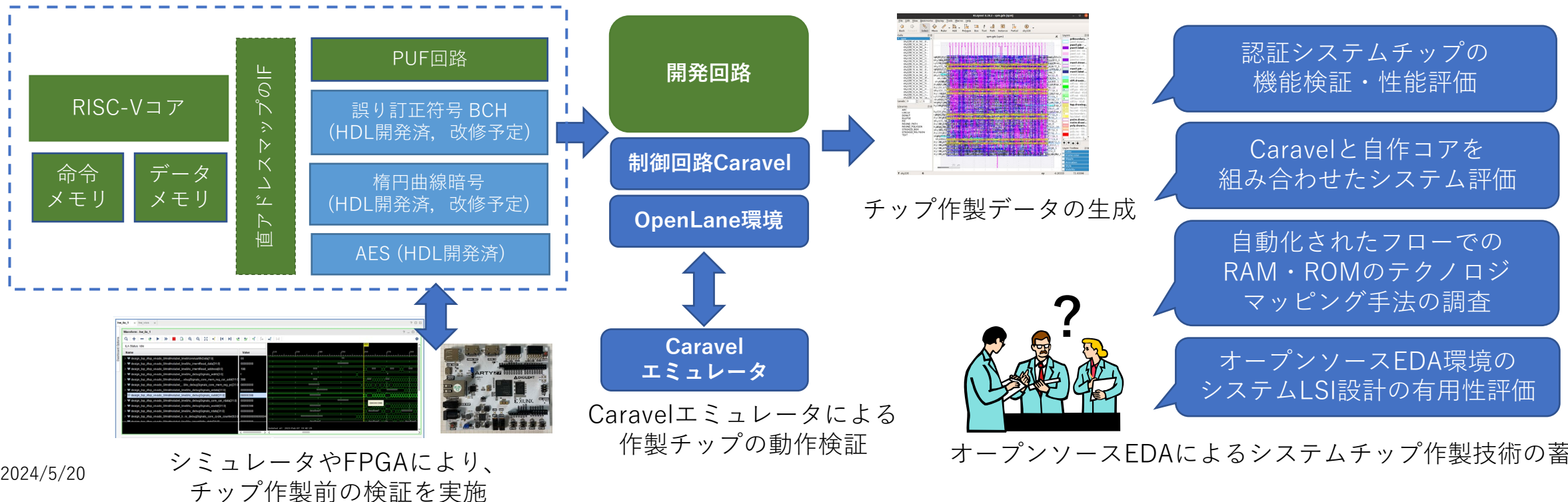
# 研究項目②：チップセキュリティ技術を題材としたオリジナルチップ作製の試行

## 目的

開発実績のある**暗号・符号回路IP**と新規開発の簡易**RISC-Vコア・PUF回路**を組み合わせたチップの作製と評価を行う。  
真贋判定や認証等のためのオリジナルの**セキュリティSoC**をオープンソースEDAを用いて開発する。

## 2023年度実施内容と目標

SoC制御とセキュリティ機能の実現のため、**新規RISC-VプロセッサとPUF回路の設計・開発**を実施する。



# 研究項目③：オープンEDA/PDKを活用したアナログ回路自動設計ツールの開発

## 目的

オープンソースEDAは、アナログ回路設計環境の整備が不十分。また、アナログ回路設計は、集積回路・デバイスに関する知識や、レイアウトに関するノウハウなどが必要で、デジタル回路設計に比べて参入障壁が高い。

オープンPDKやAI/機械学習を活用して、知識のない人でも簡単にアナログ回路を作成できる自動設計ツールを開発する。最終ゴールはアナログ回路の仕様を与えることで、それに相当したGDSファイルを生成するツールの開発を目指す。

## 2023年度実施内容・目標

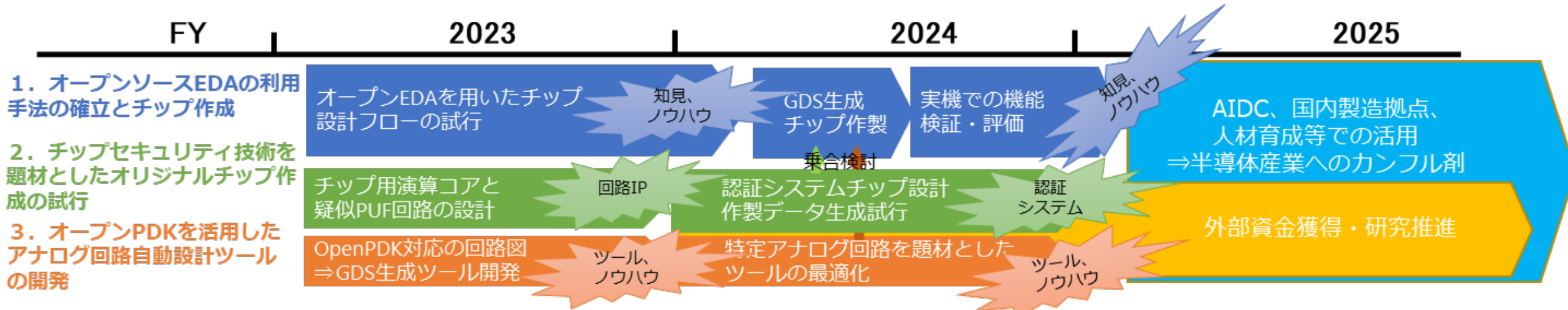
既存のネットリスト→GDS生成ツールを改良して、オープンPDKに対応させる。

- ✓ ツールの実装詳細調査
- ✓ ツールを修正するか、PDKを変換するか検討

生成されるGDSが、デザインルールを満たし製造可能であることを確認する。



# 本研究のロードマップ



## ✓ 期待されるアウトプット

- ツール活用ノウハウや知見の蓄積
- セキュア回路・システムのIP
- アナログ自動設計ツール



- 国内半導体の活性化
- 人材育成

## ✓ 本研究成果の活用

- AIDC等での情報発信
- 国内製造拠点の活性化
- LSI人材育成等

- SCR用PDK開発
- 設計フロー構築

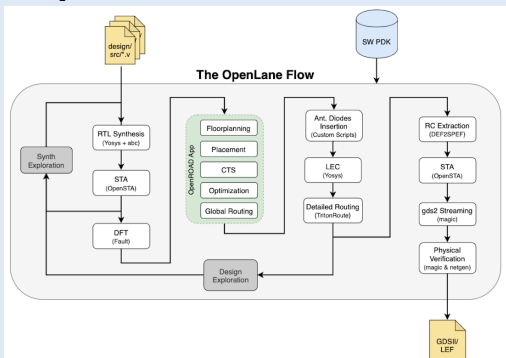


# 2023年度の進捗：利用環境構築と利用方法の知見の蓄積

## Summary

- オープンソースEDAであるOpenLane Flowの環境とシャトルテープアウト用のCaravel統合環境を試行。両環境を構築とフローの実行を完了
- 上記で得られた知見を逐一ドキュメント化

## OpenLane flow

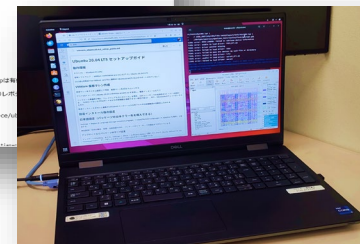


### 概要

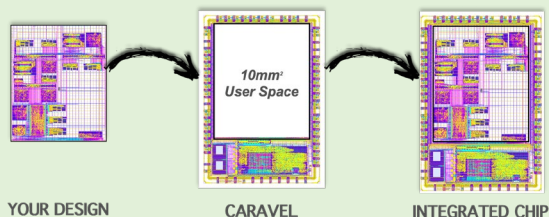
- 複数のプログラムでできたツールチェーン
- 回路記述 (RTL) からレイアウト (GDS) までを一気通貫で生成するフロー

### 本家マニュアルに未記載の事項

- フローのアップデート方法
  - 階層設計パラメータの設定
  - 書式チェックの制限
- など



## Caravel統合環境



### 概要

- SkywaterとGlobalfoundriesのシャトルにテープアウトできるプログラムとチップフレームマクロのセット

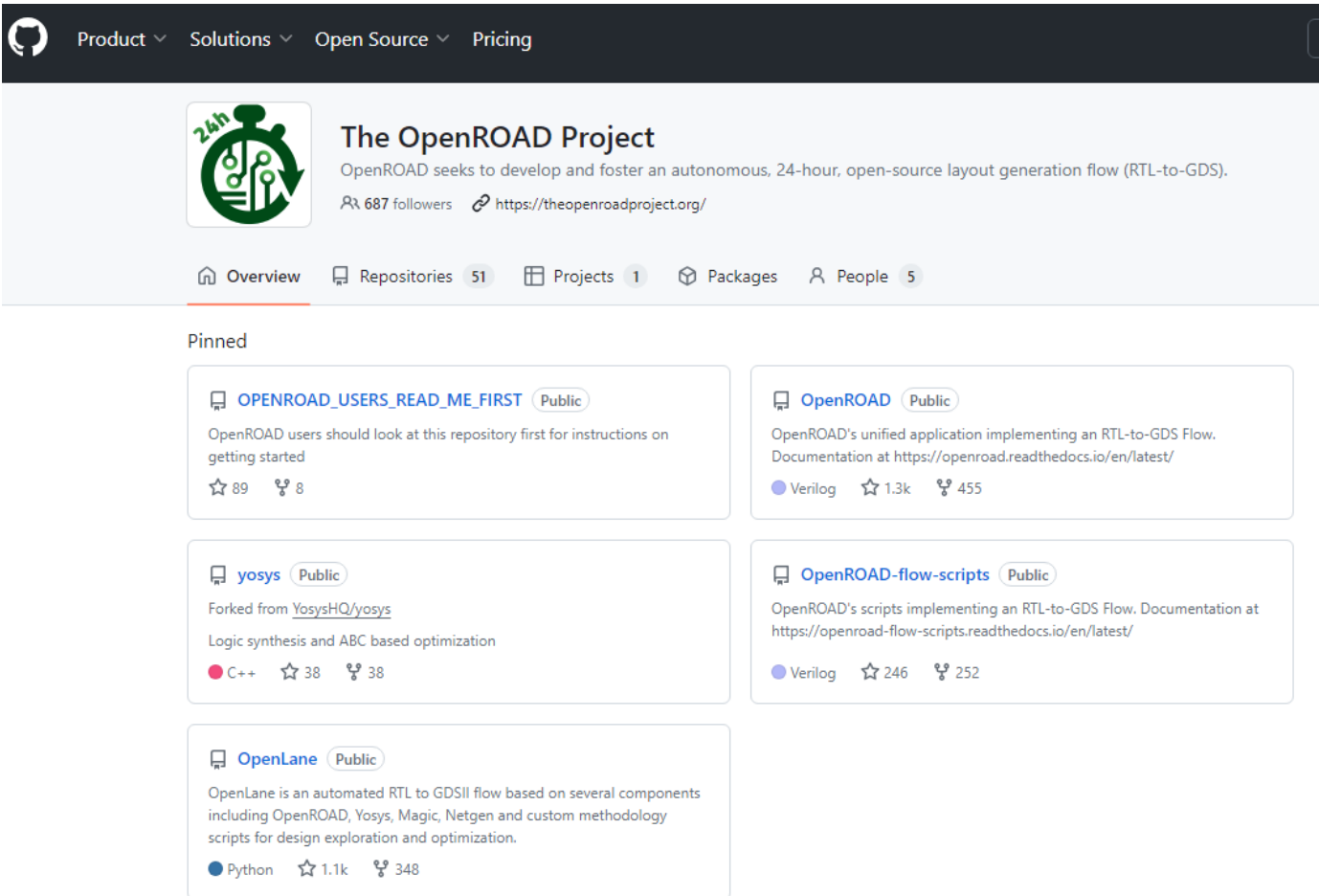
### 本家マニュアルに未記載の事項

- READMEファイルの変更
  - チップIO初期値の再設定
  - 周辺回路IPやメモリIPの使い方
- など

## 成果物

- OpenLane環境とCaravel環境の動作確認済み仮想イメージ
- 利用ドキュメント
  - Linux仮想環境構築
  - OpenLane導入と利用
  - Caravel導入と利用
  - Caravelと自作回路の結合とシミュレーション実行

# OpenLaneの導入



The screenshot shows the GitHub repository page for 'The OpenROAD Project'. The repository is public and has 687 followers. The description states: 'OpenROAD seeks to develop and foster an autonomous, 24-hour, open-source layout generation flow (RTL-to-GDS)'. The repository is categorized under 'Open Source' and 'Pricing'. The 'Pinned' section lists several repositories:

- OPENROAD\_USERS\_READ\_ME\_FIRST** (Public): OpenROAD users should look at this repository first for instructions on getting started. 89 stars, 8 forks.
- OpenROAD** (Public): OpenROAD's unified application implementing an RTL-to-GDS Flow. Documentation at <https://openroad.readthedocs.io/en/latest/>. 1.3k stars, 455 forks.
- yosys** (Public): Forked from YosysHQ/yosys. Logic synthesis and ABC based optimization. C++ language. 38 stars, 38 forks.
- OpenROAD-flow-scripts** (Public): OpenROAD's scripts implementing an RTL-to-GDS Flow. Documentation at <https://openroad-flow-scripts.readthedocs.io/en/latest/>. 246 stars, 252 forks.
- OpenLane** (Public): OpenLane is an automated RTL to GDSII flow based on several components including OpenROAD, Yosys, Magic, Netgen and custom methodology scripts for design exploration and optimization. Python language. 1.1k stars, 348 forks.

- OS

- Ubuntu 20.04+
- macOS 11+
- Windows10, 11+
- Other Linux

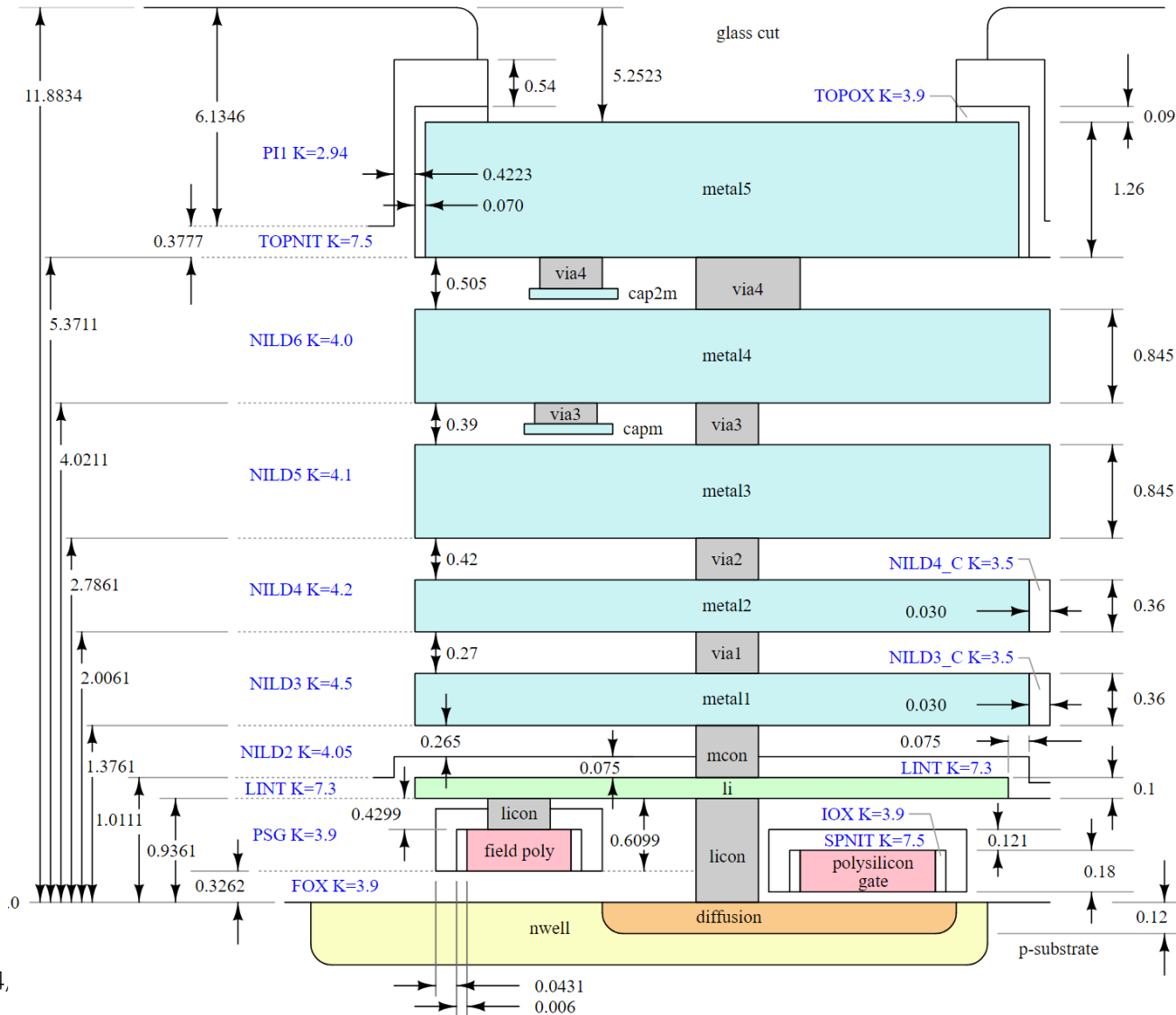
- Download and Installation

- Get python3.6
- Get docker
- git clone openlane
- cd Openlane
- Make (PDKも入る)
- make test (インストールテスト)
- make mount (OpenLane環境の立ち上げ)

- バーチャルマシン上でも動く (ex. Ubuntu on Windows) : 可搬性

# SkyWater 130nmプロセス

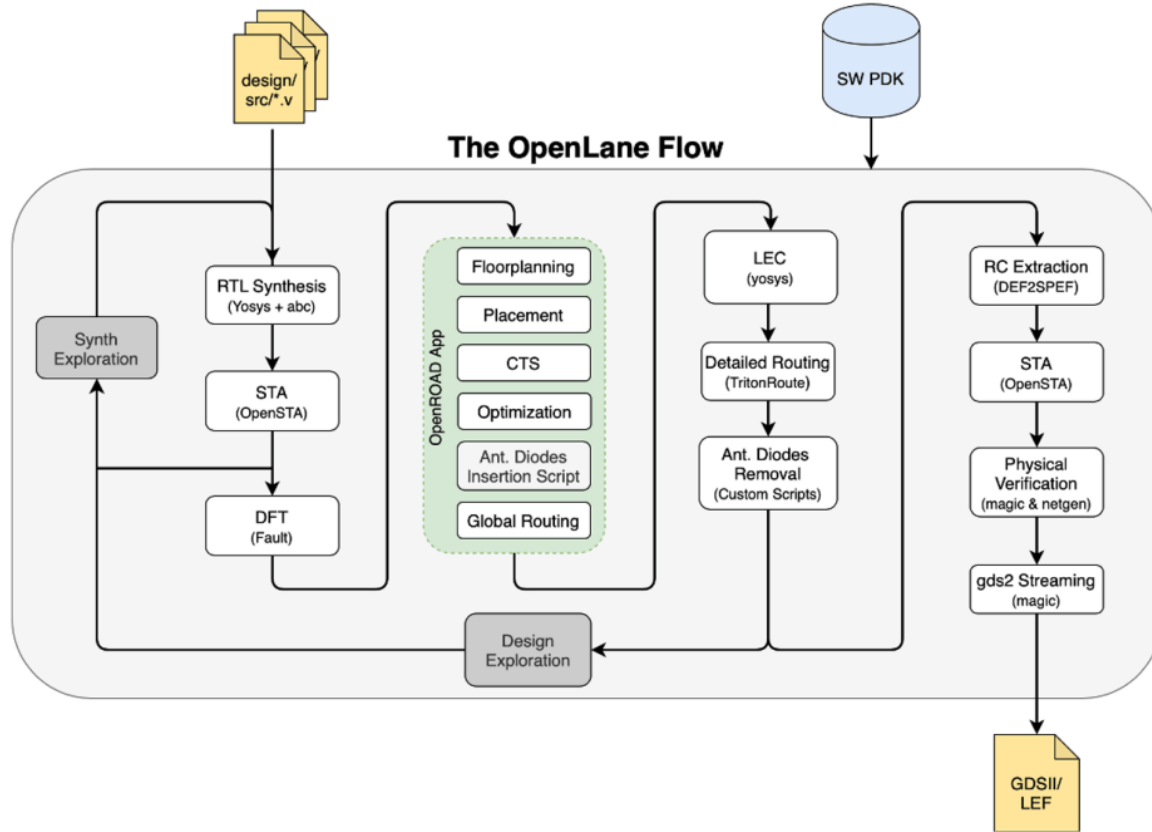
(Diagram not to scale!)



- 5 layer Al
- 1 local interconnect
- Poly gate
- Internal: 1.8V, I/O: 5V
- Inductor capable
- Poly resistor
- MIM Cap.

[GitHub - google/skywater-pdk: Open source process design kit for usage with SkyWater Technology Foundry's 130nm node.](https://github.com/google/skywater-pdk)

# OpenLane Design Stage



LEC: Logic Equivalence Check  
SPEF: Standard Parasitic Exchange Format

## 1. (Lint)

- Verilator

## 2. Synthesis

- Yosys/abc
- OpenSTA

## 3. Floorplanning

- Init\_fp
- ioplacer
- Pdngen
- tapcell

## 4. Placement

- RePLace
- Resizer
- OpenDP

## 5. CTS

- TritonCTS

## 6. Routing

- FastRoute
- TritonRoute
- OpenRCX

## 7. Tapeout

- Magic
- KLayout

## 8. Signoff

- Magic
- Klayout
- Netgen
- CVC

# 実際に実行すると48ステップある (Lint含む)

```
masa@osed: ~/OpenLane
OpenLane Container (9dbd8b5):/openlane/designs/regfile_2r1w/runs/full_guide/logs$ ls *
cts:
18-cts.errors 18-cts.log 18-cts.warnings 19-cts_sta.errors 19-cts_sta.log 19-cts_sta.warnings 20-resizer.errors 20-resizer.log 20-resizer.warnings

floorplan:
3-initial_fp.errors 3-initial_fp.log 3-initial_fp.warnings 4-io.errors 4-io.log 4-io.warnings 8-tap.errors 8-tap.log 8-tap.warnings 9-pdn.errors 9-pdn.log 9-pdn.warnings

placement:
11-gpl_sta.errors 12-global.warnings 14-gpl_sta.log 16-detailed.errors 17-dpl_sta.warnings 6-gpl_sta.log 9-global_skip_io.errors
11-gpl_sta.log 12-io.errors 14-gpl_sta.warnings 16-detailed.log 4-global.errors 6-gpl_sta.warnings 9-global_skip_io.log
11-gpl_sta.warnings 12-io.log 15-resizer.errors 16-detailed.warnings 4-global.log 7-basic_mp.errors 9-global_skip_io.warnings
12-global.errors 12-io.warnings 15-resizer.log 17-dpl_sta.errors 4-global.warnings 7-basic_mp.log
12-global.log 14-gpl_sta.errors 15-resizer.warnings 17-dpl_sta.log 6-gpl_sta.errors 7-basic_mp.warnings

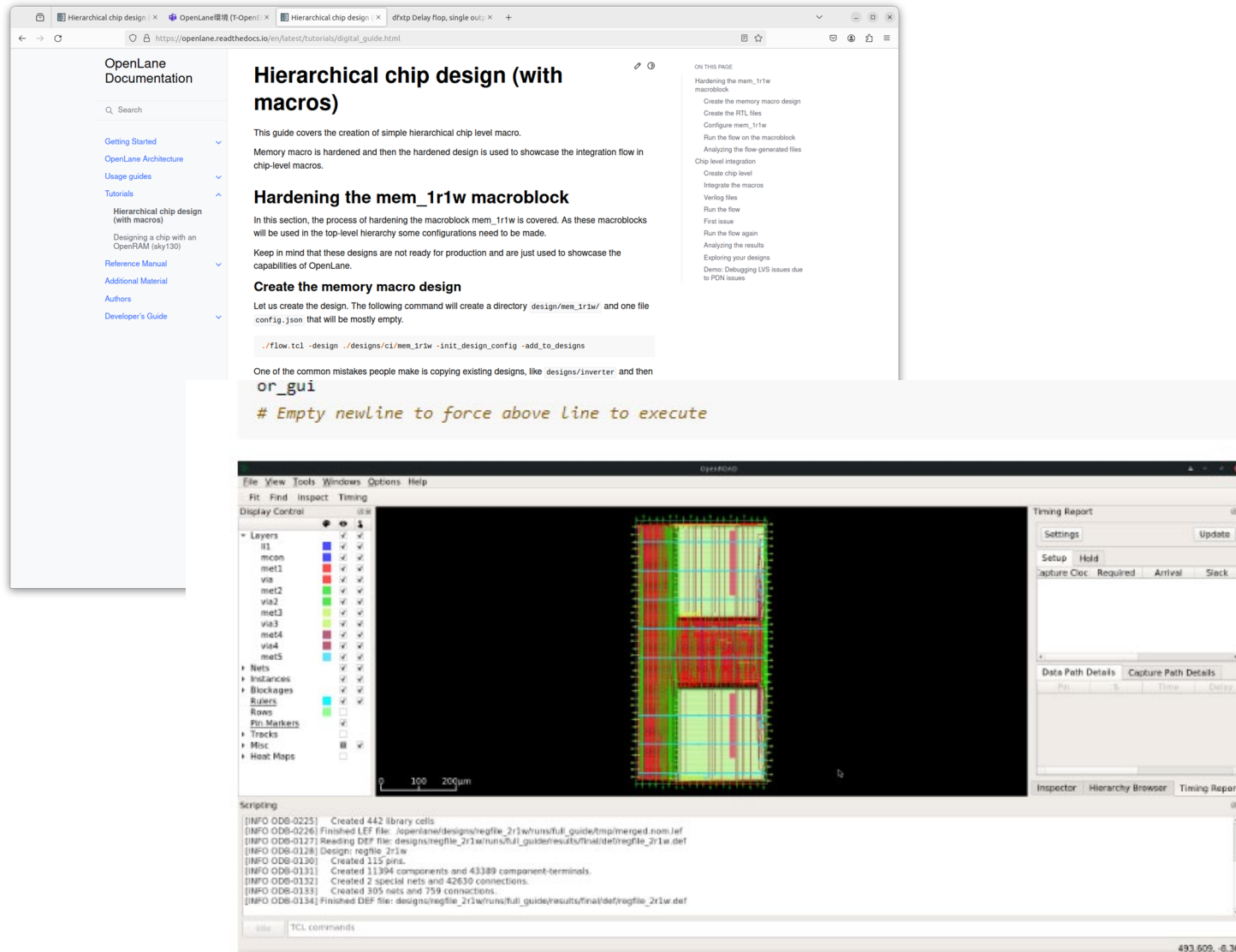
routing:
21-resizer_design.errors 23-resizer_timing.errors 25-antenna_diodes_1.log 25-global.warnings 27-grt_sta.warnings 29-detailed.warnings
21-resizer_design.log 23-resizer_timing.log 25-antenna_route_1.errors 25-global_write_netlist.errors 28-fill.errors 30-wire_lengths.log
21-resizer_design.warnings 23-resizer_timing.warnings 25-antenna_route_1.log 25-global_write_netlist.log 28-fill.log
22-rsz_design_sta.errors 24-rsz_timing_sta.errors 25-antenna_route_1.warnings 25-global_write_netlist.warnings 28-fill.warnings
22-rsz_design_sta.log 24-rsz_timing_sta.log 25-global.errors 27-grt_sta.errors 29-detailed.errors
22-rsz_design_sta.warnings 24-rsz_timing_sta.warnings 25-global.log 27-grt_sta.log 29-detailed.log

signoff:
31-parasitics_extraction.min.errors 34-rcx_mcsta.max.errors 37-irdrop.errors 38-lef.errors 41-spice.log 45-drc.errors
31-parasitics_extraction.min.log 34-rcx_mcsta.max.log 37-irdrop.log 38-lef.log 41-spice.warnings 45-drc.log
31-parasitics_extraction.min.warnings 34-rcx_mcsta.max.warnings 37-irdrop.warnings 38-lef.warnings 42-write_powered_def.log 45-drc.warnings
32-rcx_mcsta.min.errors 35-parasitics_extraction.nom.errors 38-gdsii.errors 38-maglef.errors 42-write_powered_verilog.errors 46-drc-klayout.log
32-rcx_mcsta.min.log 35-parasitics_extraction.nom.log 38-gdsii.log 38-maglef.log 42-write_powered_verilog.log 47-arc.errors
32-rcx_mcsta.min.warnings 35-parasitics_extraction.nom.warnings 38-gdsii.warnings 38-maglef.warnings 42-write_powered_verilog.warnings 47-arc.log
33-parasitics_extraction.max.errors 36-rcx_mcsta.nom.errors 38-gds_ptrs.errors 39-gdsii-klayout.log 44-lvs.lef.log 47-arc.warnings
33-parasitics_extraction.max.log 36-rcx_mcsta.nom.log 38-gds_ptrs.log 40-xor.log 44-regfile_2r1w.lef.lvs.json
33-parasitics_extraction.max.warnings 36-rcx_mcsta.nom.warnings 38-gds_ptrs.warnings 41-spice.errors 44-regfile_2r1w.lef.lvs.log

synthesis:
1-synthesis.errors 1-synthesis.log 1-synthesis.warnings 2-sta.errors 2-sta.log 2-sta.warnings linter.log
OpenLane Container (9dbd8b5):/openlane/designs/regfile_2r1w/runs/full_guide/logs$
```

実際に実行すると48ステップある

# チュートリアルの試行：Hierarchical Chip Design



手順

1. 1Read 1Write Memoryマクロブロックを作成
2. 上記マクロブロックを2個並べて2Read 1Write Regfile作成がゴール

どちらもVerilogコード、設定ファイル記述例あり。実行コマンドもあり。

- ⇒ 1. は特に問題なく実行が完了する。  
2. でエラーが起こる

# Lintチェック（Verilog文法チェック）で止まっている

マクロブロックのパラメータ継承あたりにエラーあり

```
masa@oseda: ~/OpenLane
[INFO]: Using configuration in 'designs/regfile_2r1w/config.json'...
[INFO]: PDK Root: /home/masa/.volare
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/regfile_2r1w/runs/full_guide
[INFO]: Removing existing /openlane/designs/regfile_2r1w/runs/full_guide...
[INFO]: Saving runtime environment...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[INFO]: Running linter (Verilator) (log: designs/regfile_2r1w/runs/full_guide/logs/synthesis/linter.log)...
[ERROR]: 5 errors found by linter
[ERROR]: Step 0 (verilator_lint_check) failed with error:
-code 1 -level 0 -errorcode NONE -errorinfo {
  while executing
"throw_error"
  (procedure "run_verilator" line 86)
  invoked from within
"run_verilator"
  (procedure "run_verilator_step" line 3)
  invoked from within
"run_verilator_step") -errorline 1
[INFO]: Saving current set of views in 'designs/regfile_2r1w/runs/full_guide/results/final'...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/regfile_2r1w/runs/full_guide/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/regfile_2r1w/runs/full_guide/reports/metrics.csv'.
[INFO]: Saving runtime environment...
[ERROR]: Flow failed.
[INFO]: The failure may have been because of the following warnings:
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
OpenLane Container (9dbd8b5):/openlane$
```

```
masa@oseda: ~/OpenLane
-code 1 -level 0 -errorcode NONE -errorinfo {
  while executing
"throw_error"
  (procedure "run_verilator" line 86)
  invoked from within
"run_verilator"
  (procedure "run_verilator_step" line 3)
  invoked from within
"run_verilator_step") -errorline 1
[INFO]: Saving current set of views in 'designs/regfile_2r1w/runs/full_guide/results/final'...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/regfile_2r1w/runs/full_guide/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/regfile_2r1w/runs/full_guide/reports/metrics.csv'.
[INFO]: Saving runtime environment...
[ERROR]: Flow failed.
[INFO]: The failure may have been because of the following warnings:
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
OpenLane Container (9dbd8b5):/openlane$ more designs/regfile_2r1w/runs/full_guide/logs/synthesis/linter.log
%Error-PINNOTFOUND: /openlane/designs/regfile_2r1w/src/regfile_2r1w.v:39:13: Parameter not found: 'DEPTH_LOG2'
39 | mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane0(
   |           ^~~~~~
   |           ... For error description see https://verilator.org/warn/PINNOTFOUND?v=5.018
%Error-PINNOTFOUND: /openlane/designs/regfile_2r1w/src/regfile_2r1w.v:39:38: Parameter not found: 'WIDTH'
39 | mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane0(
   |                                           ^~~~~
%Error-PINNOTFOUND: /openlane/designs/regfile_2r1w/src/regfile_2r1w.v:53:13: Parameter not found: 'DEPTH_LOG2'
53 | mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane1(
   |           ^~~~~~
%Error-PINNOTFOUND: /openlane/designs/regfile_2r1w/src/regfile_2r1w.v:53:38: Parameter not found: 'WIDTH'
53 | mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane1(
   |                                           ^~~~~
%Error: Exiting due to 4 error(s)
... See the manual at https://verilator.org/verilator_doc.html for more assistance.
OpenLane Container (9dbd8b5):/openlane$
```

## モジュール呼び出しからパラメータを削除

```

mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane0(
  .clk(clk),

  .read_addr(rs1_addr),
  .read(rs1_read),
  .read_data(rs1_rdata),
  .write(write),
  .write_addr(write_addr),
  .write_data(write_data)
);

```

```

mem_1r1w #(.DEPTH_LOG2(DEPTH_LOG2), .WIDTH(WIDTH)) lane1(
  .clk(clk),

  .read_addr(rs2_addr),
  .read(rs2_read),
  .read_data(rs2_rdata),

  .write(write),
  .write_addr(write_addr),
  .write_data(write_data)
);

```

endmodule

U:--- regfile\_2r1w.v All L45 (Verilog)  
 <C-M-print> is undefined



```

mem_1r1w lane0(
  .clk(clk),

  .read_addr(rs1_addr),
  .read(rs1_read),
  .read_data(rs1_rdata),

  .write(write),
  .write_addr(write_addr),
  .write_data(write_data)
);

```

```

mem_1r1w lane1(
  .clk(clk),

  .read_addr(rs2_addr),
  .read(rs2_read),
  .read_data(rs2_rdata),

  .write(write),
  .write_addr(write_addr),
  .write_data(write_data)
);

```

endmodule

U:--- regfile\_2r1w.v All L54 (Verilog)  
 Wrote /home/masa/OpenLane/designs/regfile\_2r1w/src/regfile\_2r1w.v

# 今度はSTAで止まっている

“ブラックボックスファイルがRTLでなくゲートレベルネットリストであることを確認すべし”

```

masa@oseda: ~/OpenLane
[INFO]: 0 warnings found by linter
[STEP 1]
[INFO]: Running Synthesis (log: designs/regfile_2r1w/runs/full_guide/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: designs/regfile_2r1w/runs/full_guide/logs/synthesis/2-sta.log)...
[ERROR]: during executing sta script /openlane/scripts/openroad/sta/multi_corner.tcl
[ERROR]: Log: designs/regfile_2r1w/runs/full_guide/logs/synthesis/2-sta.log
[ERROR]: Last 10 lines:
Using 1e+00 for voltage...
Using 1e-03 for current...
Using 1e-09 for power...
Using 1e-06 for distance...
Reading netlist '/openlane/designs/regfile_2r1w/runs/full_guide/results/synthesis/regfile_2r1w.v'...
Error while reading /openlane/designs/regfile_2r1w/bb/mem_1r1w.bb.v:
Make sure that this a gate-level netlist not an RTL file
You can add the following comment '/// sta-blackbox' in the file to skip it and blackbox the modules inside if needed.
Error: /openlane/designs/regfile_2r1w/bb/mem_1r1w.bb.v line 5, syntax error, unexpected '=', expecting '('
child process exited abnormally

[ERROR]: Creating issue reproducible...
[INFO]: Saving runtime environment...
OpenLane TCL Issue Packager

EFABLESS CORPORATION AND ALL AUTHORS OF THE OPENLANE PROJECT SHALL NOT BE HELD
LIABLE FOR ANY LEAKS THAT MAY OCCUR TO ANY PROPRIETARY DATA AS A RESULT OF USING
THIS SCRIPT. THIS SCRIPT IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND.

BY USING THIS SCRIPT, YOU ACKNOWLEDGE THAT YOU FULLY UNDERSTAND THIS DISCLAIMER
AND ALL IT ENTAILS.

Parsing config file(s)...
Setting up /openlane/designs/regfile_2r1w/runs/full_guide/issue_reproducible...
Done.
[INFO]: Reproducible packaged at 'designs/regfile_2r1w/runs/full_guide/issue_reproducible'.
OpenLane Container (9dbd8b5):/openLane$

```

```

masa@oseda: ~/OpenLane
cts/      floorplan/ placement/ routing/  signoff/  synthesis/
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/
cts/      floorplan/ placement/ routing/  signoff/  synthesis/
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/
cts/      floorplan/ placement/ routing/  signoff/  synthesis/
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/synthesis/
1-synthesis.errors  1-synthesis.warnings  2-sta.log          linter.log
1-synthesis.log     2-sta.errors          2-sta.warnings
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/synthesis/
1-synthesis.errors  1-synthesis.warnings  2-sta.log          linter.log
1-synthesis.log     2-sta.errors          2-sta.warnings
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/synthesis/
1-synthesis.errors  1-synthesis.warnings  2-sta.log          linter.log
1-synthesis.log     2-sta.errors          2-sta.warnings
OpenLane Container (9dbd8b5):/openLane$ more designs/regfile_2r1w/runs/full_guide/logs/synthesis/2-sta.log
OpenSTA 2.4.0 75f2f325b7 Copyright (c) 2023, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show_copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.
define_corners Typical
read_liberty -corner Typical /home/masa/.volare/sky130A/libs.ref/sky130_fd_sc_hd/lib/sky130_fd_sc_hd__tt_025C_1v80.lib
Using 1e-12 for capacitance...
Using 1e+03 for resistance...
Using 1e-09 for time...
Using 1e+00 for voltage...
Using 1e-03 for current...
Using 1e-09 for power...
Using 1e-06 for distance...
Reading netlist '/openlane/designs/regfile_2r1w/runs/full_guide/results/synthesis/regfile_2r1w.v'...
Error while reading /openlane/designs/regfile_2r1w/bb/mem_1r1w.bb.v:
Make sure that this a gate-level netlist not an RTL file
You can add the following comment '/// sta-blackbox' in the file to skip it and blackbox the modules inside if needed.
Error: /openlane/designs/regfile_2r1w/bb/mem_1r1w.bb.v line 5, syntax error, unexpected '=', expecting '('
OpenLane Container (9dbd8b5):/openLane$

```

# チュートリアルでのブラックボックス作成箇所 RTLの頭を切り抜いただけ、、、

Create the verilog blackbox:

```
(*blackbox*)  
  
module mem_1r1w (clk, read_addr, read, read_data, write_addr, write, write_data);  
    parameter DEPTH_LOG2 = 4;  
    localparam ELEMENTS = 2**DEPTH_LOG2;  
    parameter WIDTH = 32;  
  
    input wire clk;  
  
    input wire [DEPTH_LOG2-1:0] read_addr;  
    input wire read;  
    output reg [WIDTH-1:0] read_data;  
  
    input wire [DEPTH_LOG2-1:0] write_addr;  
    input wire write;  
    input wire [WIDTH-1:0] write_data;  
  
endmodule
```

## 設定ファイル：Config.jsonを修正

1r1wメモリマクロブロック作成時にゲートレベルネットリストが既に作成されているので  
ブラックボックスファイルとしてパスを通す

```

emacs@oseda
File Edit Options Buffers Tools JavaScript Help
Save Undo
{
  "DESIGN_NAME": "regfile_2r1w",
  "VERILOG_FILES": "dir::src/*.v",
  "CLOCK_PORT": "clk",
  "CLOCK_PERIOD": 10.0,
  "FP_PDN_MULTILAYER": true,
  [
  "FP_ASPECT_RATIO": 2,

  "EXTRA_LEFS": "/openlane/designs/ci/mem_1r1w/runs/full_guide/results/final/lef/mem_1r1w.lef",
  "EXTRA_GDS_FILES": "/openlane/designs/ci/mem_1r1w/runs/full_guide/results/final/gds/mem_1r1w.gds",
  "VERILOG_FILES_BLACKBOX": "dir::bb/*.v"
}
  
```



```

emacs@oseda
File Edit Options Buffers Tools JavaScript Help
Save Undo
{
  "DESIGN_NAME": "regfile_2r1w",
  "VERILOG_FILES": "dir::src/*.v",
  "CLOCK_PORT": "clk",
  "CLOCK_PERIOD": 10.0,
  "FP_PDN_MULTILAYER": true,

  "FP_ASPECT_RATIO": 2,

  "EXTRA_LEFS": "/openlane/designs/ci/mem_1r1w/runs/full_guide/results/final/lef/mem_1r1w.lef",
  "EXTRA_GDS_FILES": "/openlane/designs/ci/mem_1r1w/runs/full_guide/results/final/gds/mem_1r1w.gds",
  "VERILOG_FILES_BLACKBOX": "/openlane/designs/ci/mem_1r1w/runs/full_guide/results/final/verilog/gl/mem_1r1w.v"
}
  
```

# Flow Complete!

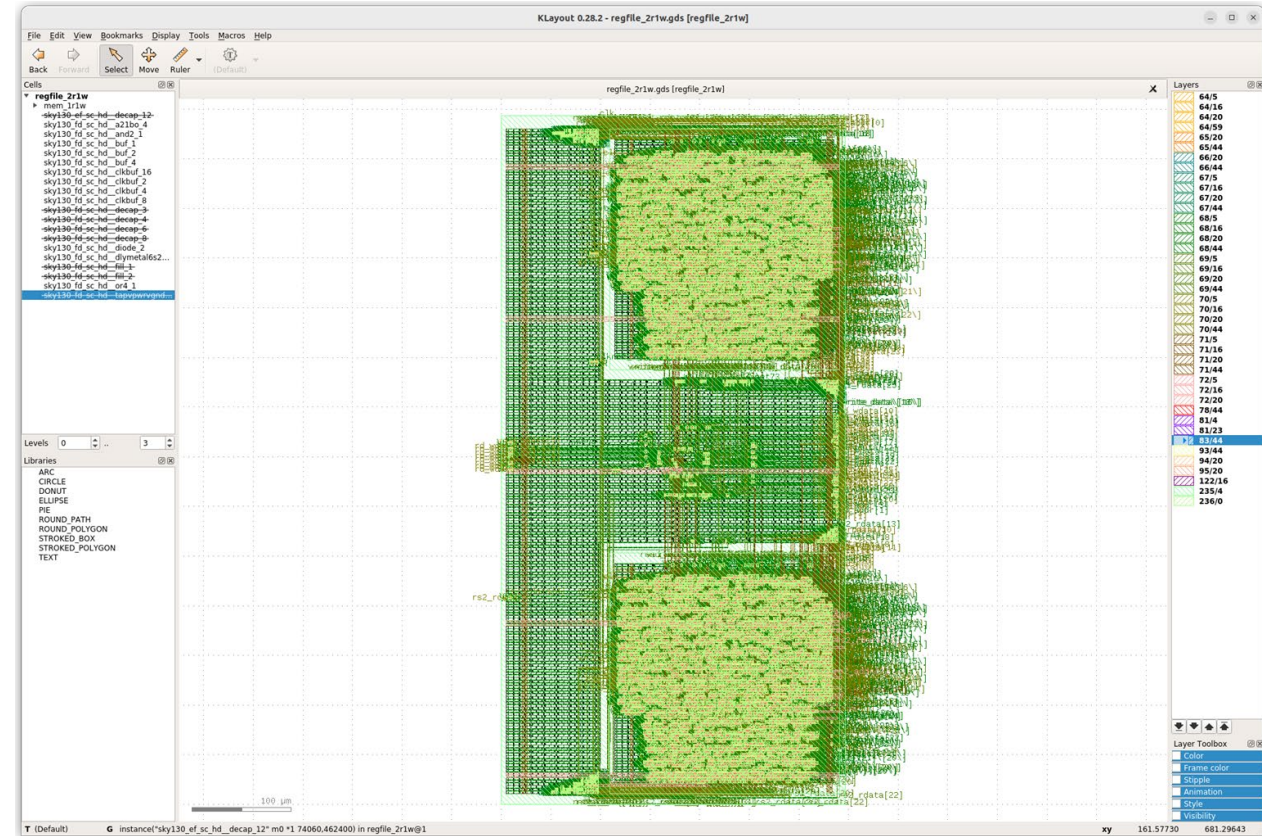
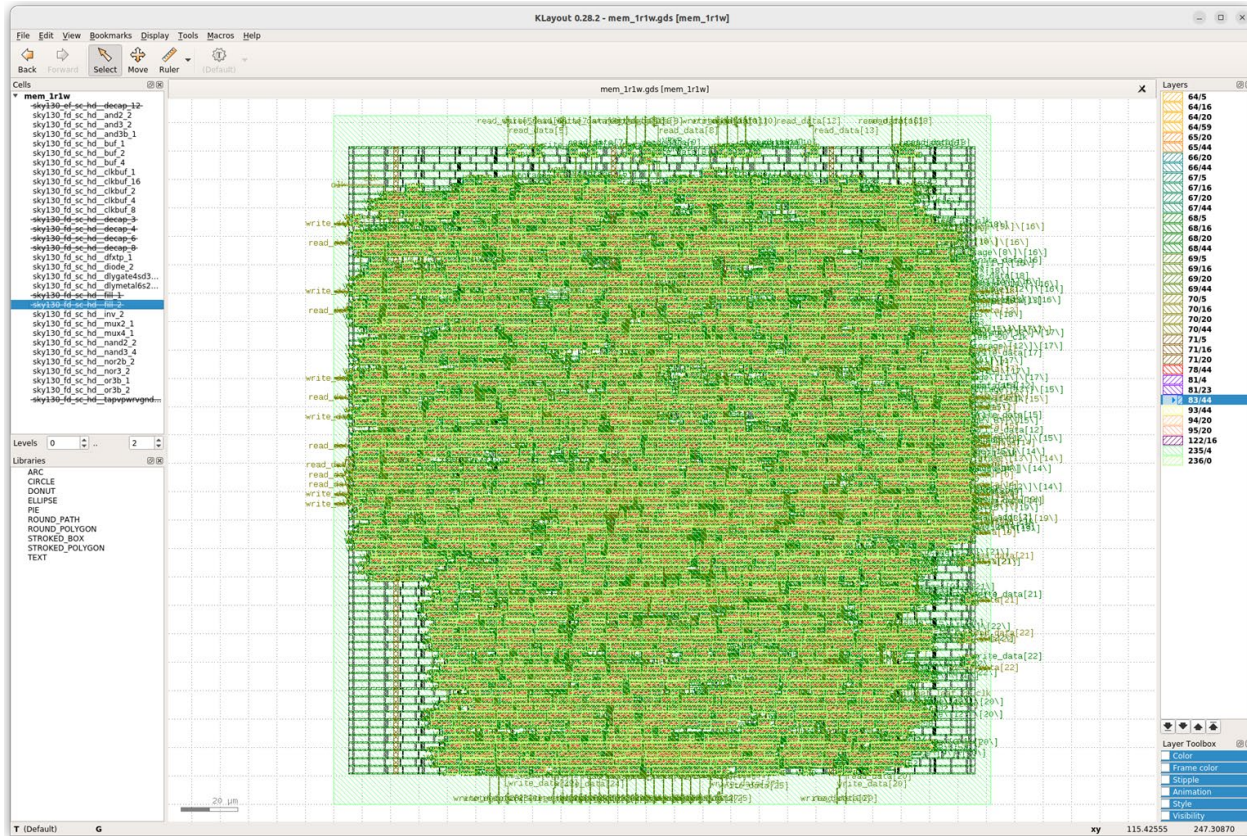
```
masa@oseda: ~/OpenLane
[STEP 42]
[INFO]: Writing Powered Verilog (logs: designs/regfile_2r1w/runs/full_guide/logs/signoff/42-write_powered_def.log, designs/regfile_2r1w/runs/full_guide/logs/signoff/42-write_powered_verilog.log)...
[STEP 43]
[INFO]: Writing Verilog (log: designs/regfile_2r1w/runs/full_guide/logs/signoff/42-write_powered_verilog.log)...
[STEP 44]
[INFO]: Running LVS (log: designs/regfile_2r1w/runs/full_guide/logs/signoff/44-lvs.lef.log)...
[STEP 45]
[INFO]: Running Magic DRC (log: designs/regfile_2r1w/runs/full_guide/logs/signoff/45-drc.log)...
[INFO]: Converting Magic DRC database to various tool-readable formats...
[INFO]: No Magic DRC violations after GDS streaming out.
[STEP 46]
[INFO]: Running KLayout DRC (log: designs/regfile_2r1w/runs/full_guide/logs/signoff/46-drc-klayout.log)...
[INFO]: No KLayout DRC violations after GDS streaming out.
[STEP 47]
[INFO]: Running OpenROAD Antenna Rule Checker (log: designs/regfile_2r1w/runs/full_guide/logs/signoff/47-arc.log)...
[INFO]: Saving current set of views in 'designs/regfile_2r1w/runs/full_guide/results/final'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at 'designs/regfile_2r1w/runs/full_guide/reports/manufacturability.rpt'.
[INFO]: Created metrics report at 'designs/regfile_2r1w/runs/full_guide/reports/metrics.csv'.
[WARNING]: There are max fanout violations in the design at the Typical corner. Please refer to 'designs/regfile_2r1w/runs/full_guide/reports/signoff/36-sta-rcx_nom/multi_corner_sta.checks.rpt'.
[INFO]: There are no hold violations in the design at the Typical corner.
[INFO]: There are no setup violations in the design at the Typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: PNR_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: SIGNOFF_SDC_FILE is not set. It is recommended to write a custom SDC file for the design. Defaulting to BASE_SDC_FILE
[WARNING]: 4 warnings found by linter
[WARNING]: VSRC_LOC_FILES was not given a value, which may make the results of IR drop analysis inaccurate. If you are not integrating a top-level chip for manufacture, you may ignore this warning, otherwise, see the documentation for VSRC_LOC_FILES.
[WARNING]: There are max fanout violations in the design at the Typical corner. Please refer to 'designs/regfile_2r1w/runs/full_guide/reports/signoff/36-sta-rcx_nom/multi_corner_sta.checks.rpt'.

OpenLane Container (9dbd8b5):/openlane$
```

# GDSファイルの作成結果 (Klayoutで表示)

Mem 1r1w

regfile 2r1w



- ツールは安定版に移行した
- ドキュメントの更新が追い付いていないのかも？

# 2023年度の進捗：HWセキュリティ向け回路IPの開発

## Summary

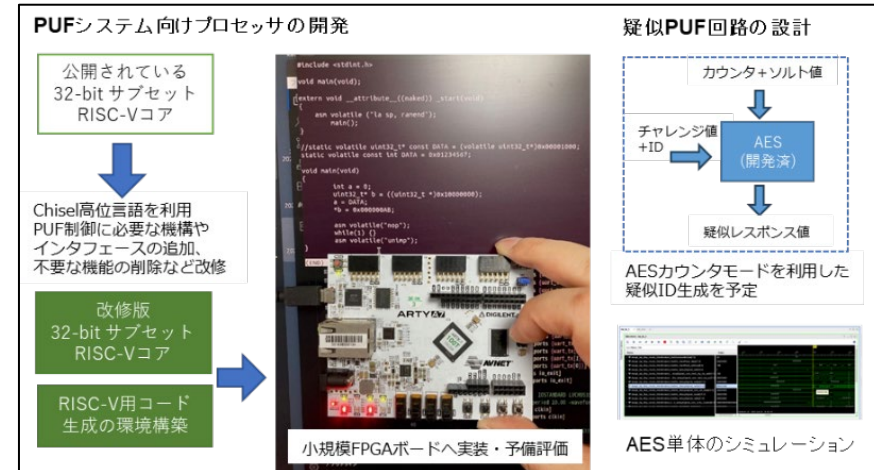
- HWセキュリティシステム用のRISC-V、疑似PUF回路、誤り訂正符号、楕円曲線暗号の回路IP開発を目標
- RISC-Vと疑似PUF回路は回路記述ファイル（RTL）・機能検証・レイアウト生成を完了。
- 誤り訂正符号と楕円曲線暗号はC言語で作成完了。  
⇒C言語から変換したRTLでOpenLane flowを実行してもレイアウト生成が完了していない。原因究明中。

## モチベーション

- オープンソースEDAの利用によって誰でも半導体を作れるようになる  
⇒偽造半導体の製造も容易になるかもしれない？
- 産総研がいち早くオープンソース半導体開発環境下のHWセキュリティを対策。  
そのための回路IPを整備

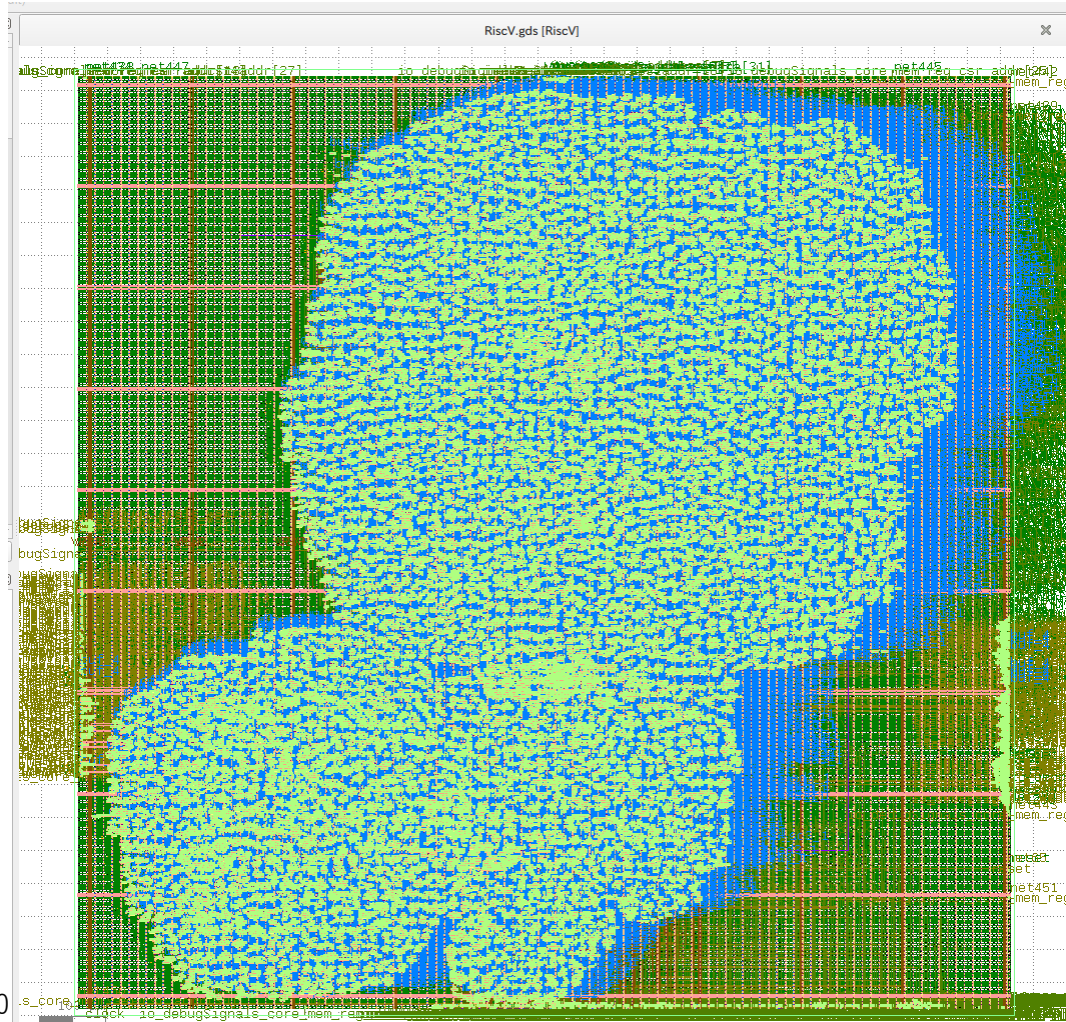
## 回路IPの開発状況

回路IP	回路記述 (RTL)	回路の機能検証	レイアウト生成
RISC-V	○	○ (FPGAを利用)	○
疑似PUF回路	○	○ (シミュレーション)	○
誤り訂正符号	△ (C言語)	×	×
楕円曲線暗号	△ (C言語)	×	×

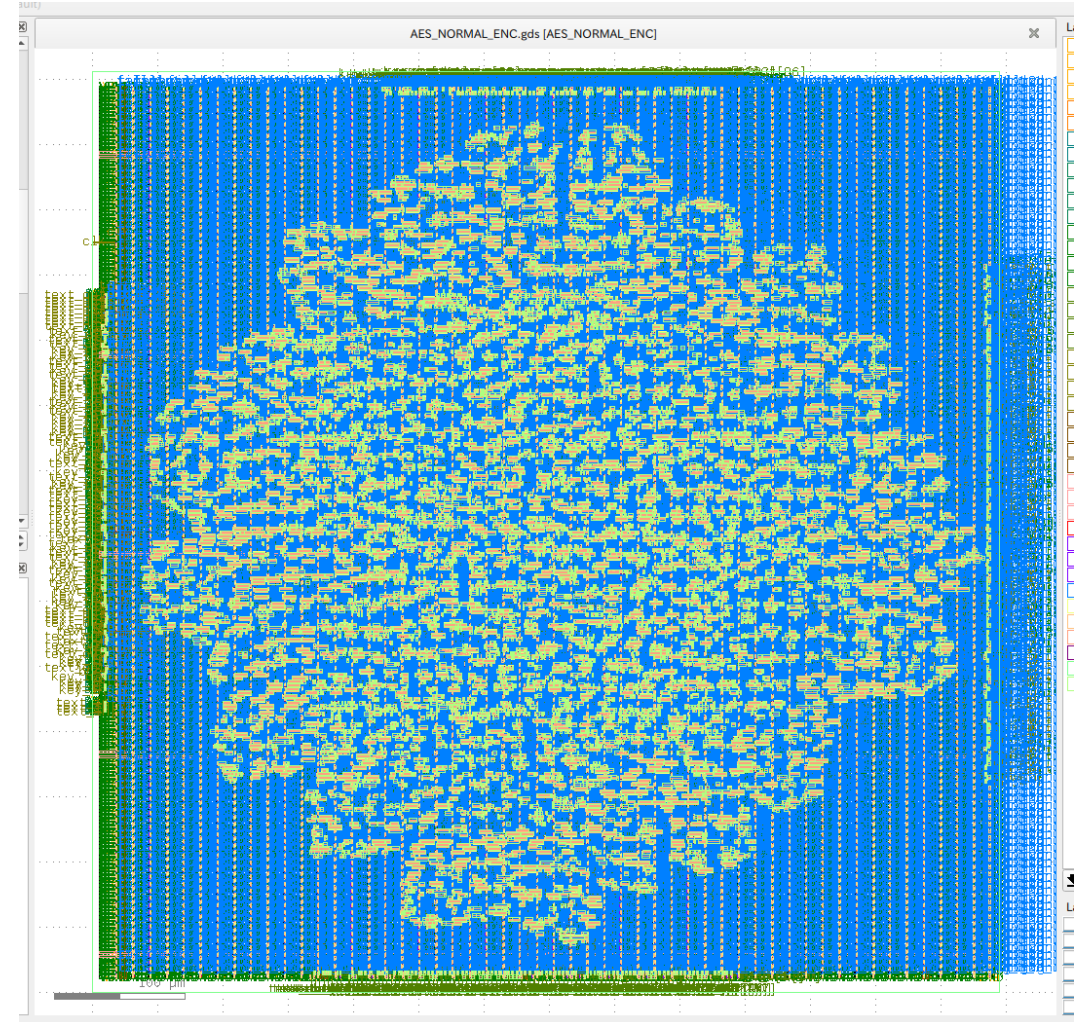


# GDS生成結果

RISC-V



疑似PUF



# 2023年度の進捗：アナログ設計環境の拡張

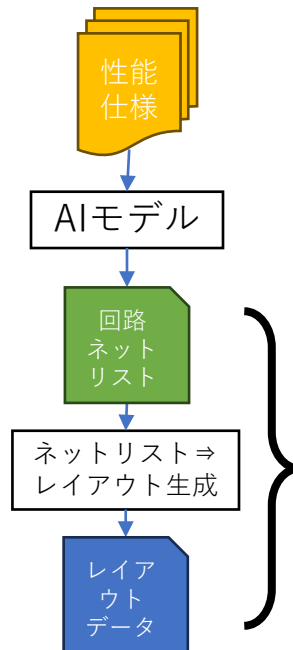
## Summary

- オープンソースの回路ネットリスト⇒レイアウト生成ツールを用いてオペアンプを例題に回路ネットリストからskywater130nmのレイアウト生成環境を整備
- 加えてオープンソースEDAで完結する形でデザインルール検証環境（DRC）、ネットリストとレイアウトの整合性検証環境（LVS）も整備

## モチベーション

- アナログ回路設計は回路・デバイス・レイアウト等の様々な知識・ノウハウが必要なためデジタルと比較して敷居が高い
- オープンPDKやAI/機械学習を活用して、知識のない人でも簡単にアナログ回路を作成できるようにオープンソースEDAのアナログ設計環境を拡張する

## 拡張アナログ設計フロー



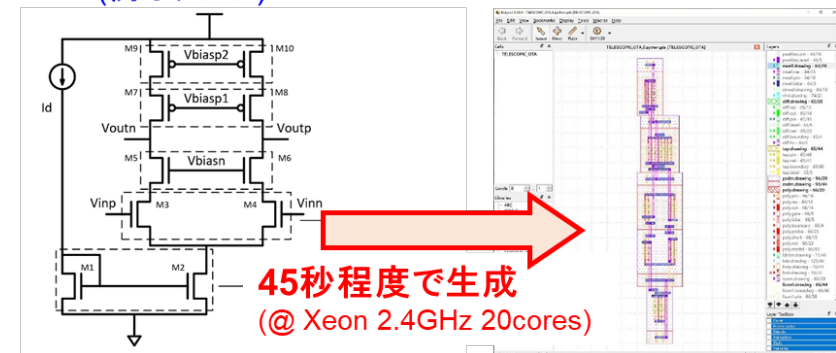
## 今年度実施内容

- ミネソタ大が開発した回路ネットリスト⇒レイアウト生成オープンソースツール“ALIGN”でskywater130nmオープンソースPDKを利用できるように設定修正
- オペアンプを例題にALIGNで回路ネットリストからレイアウト生成を完了
- 加えてオープンソースEDAで完結する形でデザインルール検証環境（DRC）、ネットリストとレイアウトの整合性検証環境（LVS）も整備

## オペアンプを用いた変換例

回路図(ネットリスト)を入力  
(例はアンプ)

GDSを出力



# ALIGN: Analog Layout, Intelligently Generated from Netlists



The screenshot shows the ALIGN website interface. The main content area is titled "ALIGN: flow" and lists the following steps:

- Circuit annotation** creates a multilevel hierarchical representation of the input netlist. This representation is used to implement the circuit layout in using a hierarchical manner.
- Design rule abstraction** creates a compact JSON-format representation of the design rules in a PDK. This repository provides a mock PDK based on a FinFET technology (where the parameters are based on published data). These design rules are used to guide the layout and ensure DRC-correctness.
- Primitive cell generation** works with primitives, i.e., blocks the lowest level of design hierarchy, and generates their layouts. Primitives typically contain a small number of transistor structures (each of which may be implemented using multiple fins and/or fingers). A parameterized instance of a primitive is automatically translated to a GDSII layout in this step.
- Placement and routing** performs block assembly of the hierarchical blocks in the netlist and routes connections between these blocks, while obeying a set of analog layout constraints. At the end of this step, the translation of the input SPICE netlist to a GDSII layout is complete.

Below the text is a diagram titled "ALIGN Layout Generator" showing the workflow:

- Input: Unannotated netlist** (Proprietary) feeds into **Netlist auto-annotation**.
- Netlist auto-annotation** feeds into **Electrical constraint generation**.
- Electrical constraint generation** feeds into **Primitive layout generation**.
- Primitive layout generation** feeds into **Block assembly (placement, floorplanning, routing)**.
- Block assembly (placement, floorplanning, routing)** feeds into **Output: GDSII** (Proprietary).

The diagram also shows **Machine learning models** (Proprietary) influencing the **Netlist auto-annotation** and **Block assembly** steps. The **CORE LAYOUT GENERATION ENGINE** is shown as a central component receiving input from **Design rules** (Proprietary) and **Machine learning models**.

**Inputs**

- Design netlist (SPICE format): Input from designers
  - Example of the analog circuit.
- Library (SPICE format): Commonly used basic building blocks for analog design
  - Basic built-in **template library** is provided to identify hierarchies in the design.
  - More library elements can be added in the **user\_template\_library**.
  - Each of the library elements can be associated with a set of constraints.
- PDK rules (JSON format): Abstracted **design rules**
  - Mock FinFET 14nm PDK **rules file** is provided, which is used by the primitive cell generator and the place and route engine.
  - New PDK can be represented using a JSON-format design rule abstraction, similar to mock-PDK design rules file provided.
  - Primitive cells (NMOS/PMOS/Resistor/Capacitor) must be redefined for any new PDK.

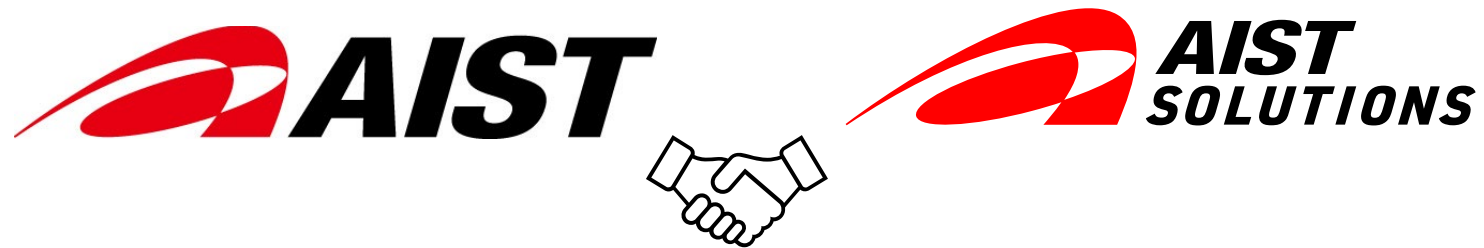
# AIST SOLUTIONSとのコラボレーション

本研究で得られた知見や情報を産総研単独で広く公開するのは簡単ではない

- ヒューマンリソース
- コネクション
- メソッド etc

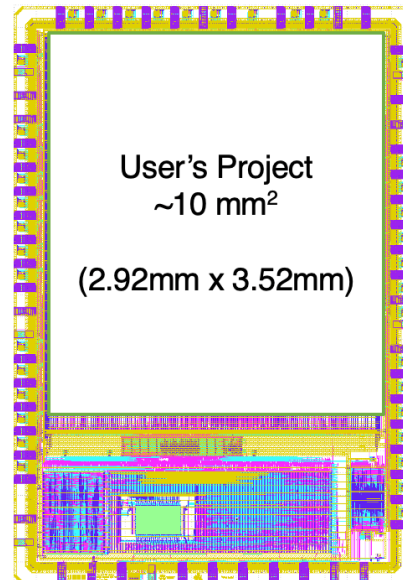
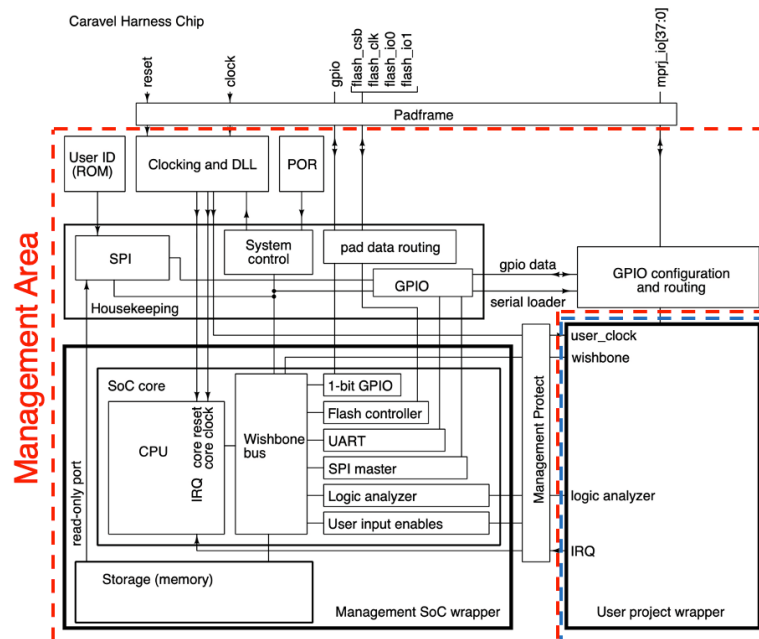
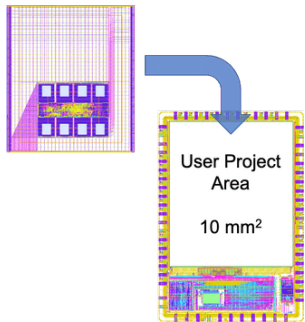
⇒AIST SOLUTIONSで加速

- Open Source Silicon活用検証委員会を立ち上げた
- 委員会活動の一環で本研究で得た知見を公開予定
- efablessのChiplgniteシャトルでチップ試作



# Chiplgniteシャトルサービス

- Chiplgniteシャトル：efablessによる有償シャトル（\$9,750）
- Caravelハーネス、オープンPDK、Openlaneからなる統合環境
- ユーザ回路のVerilog to GDS、ハーネスへの配置、テープアウトチェックまでほぼ自動
- シミュレーション環境もある
- QFNチップ100個、評価ボード付



- 38 programmable IO's
- 10 mm<sup>2</sup> of User Project Area
- Diagnostic port including IO configuration and Flash pass-thru access for programming
- VexRiscv CPU with debug
- 3 kbytes of RAM
- Flash controller supporting execute-in-place
- SPI, UART and GPIO
- Counter/Timers
- 128 signal logic analyzer for project

# Chiplgniteシャトルでの試作方針

- efablessから提供されているexampleを活用し、IPを組み込む
- Exampleは必要最小限の変更に留める
  - IPポート名、IPポート数
  - ピン配置
  - もともと存在するモジュールに手を付けない
- IPは公開ベースのものを使用（人為的エラー回避、設計時間の短縮）

# Our design: AES SBOX-like function

```
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
//===== AES_SBOX_ENC
module AES_SBOX_ENC
(input wire [7:0] x,
 output wire [7:0] y);

//-----
assign y = s(x[7:0]);

function [7:0] s (input [7:0] x);
  case (x)
    8'h00: s=8'h63; 8'h01: s=8'h7c; 8'h02: s=8'h77; 8'h03: s=8'h7b;
    8'h04: s=8'hf2; 8'h05: s=8'h6b; 8'h06: s=8'h6f; 8'h07: s=8'hc5;
    8'h08: s=8'h30; 8'h09: s=8'h01; 8'h0A: s=8'h67; 8'h0B: s=8'h2b;
    8'h0C: s=8'hfe; 8'h0D: s=8'hd7; 8'h0E: s=8'hab; 8'h0F: s=8'h76;

    8'h10: s=8'hca; 8'h11: s=8'h82; 8'h12: s=8'hc9; 8'h13: s=8'h7d;
    8'h14: s=8'hfa; 8'h15: s=8'h59; 8'h16: s=8'h47; 8'h17: s=8'hf0;
    8'h18: s=8'had; 8'h19: s=8'hd4; 8'h1A: s=8'ha2; 8'h1B: s=8'haf;
    8'h1C: s=8'h9c; 8'h1D: s=8'ha4; 8'h1E: s=8'h72; 8'h1F: s=8'hc0;

    8'h20: s=8'hb7; 8'h21: s=8'hfd; 8'h22: s=8'h93; 8'h23: s=8'h26;
    8'h24: s=8'h36; 8'h25: s=8'h3f; 8'h26: s=8'hf7; 8'h27: s=8'hcc;
    8'h28: s=8'h34; 8'h29: s=8'ha5; 8'h2A: s=8'he5; 8'h2B: s=8'hf1;
    8'h2C: s=8'h71; 8'h2D: s=8'hd8; 8'h2E: s=8'h31; 8'h2F: s=8'h15;

    8'h30: s=8'h04; 8'h31: s=8'hc7; 8'h32: s=8'h23; 8'h33: s=8'hc3;
    8'h34: s=8'h18; 8'h35: s=8'h96; 8'h36: s=8'h05; 8'h37: s=8'h9a;
    8'h38: s=8'h07; 8'h39: s=8'h12; 8'h3A: s=8'h80; 8'h3B: s=8'he2;
    8'h3C: s=8'heb; 8'h3D: s=8'h27; 8'h3E: s=8'hb2; 8'h3F: s=8'h75;

    8'h40: s=8'h09; 8'h41: s=8'h83; 8'h42: s=8'h2c; 8'h43: s=8'h1a;
    8'h44: s=8'h1b; 8'h45: s=8'h6e; 8'h46: s=8'h5a; 8'h47: s=8'ha0;
    8'h48: s=8'h52; 8'h49: s=8'h3b; 8'h4A: s=8'hd6; 8'h4B: s=8'hb3;
    8'h4C: s=8'h29; 8'h4D: s=8'he3; 8'h4E: s=8'h2f; 8'h4F: s=8'h84;

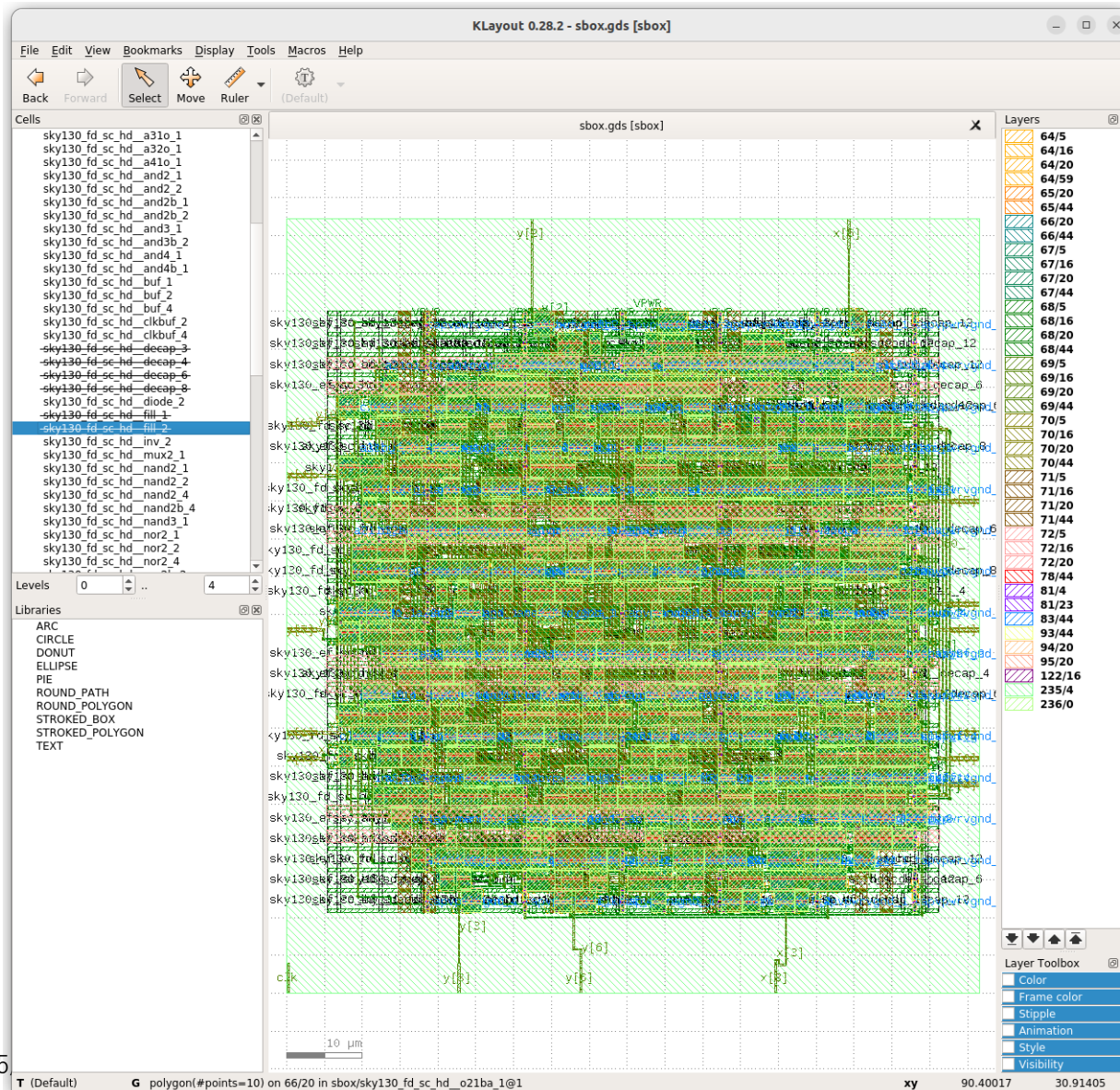
    8'h50: s=8'h53; 8'h51: s=8'hd1; 8'h52: s=8'h00; 8'h53: s=8'hed;
    8'h54: s=8'h20; 8'h55: s=8'hfc; 8'h56: s=8'hb1; 8'h57: s=8'h5b;
    8'h58: s=8'h6a; 8'h59: s=8'hcb; 8'h5A: s=8'hbe; 8'h5B: s=8'h39;
    8'h5C: s=8'h4a; 8'h5D: s=8'h4c; 8'h5E: s=8'h58; 8'h5F: s=8'hcf;

    8'h60: s=8'hd0; 8'h61: s=8'hef; 8'h62: s=8'haa; 8'h63: s=8'hfb;
    8'h64: s=8'h43; 8'h65: s=8'h4d; 8'h66: s=8'h33; 8'h67: s=8'h85;
    8'h68: s=8'h45; 8'h69: s=8'hf9; 8'h6A: s=8'h02; 8'h6B: s=8'h7f;
    8'h6C: s=8'h50; 8'h6D: s=8'h3c; 8'h6E: s=8'h9f; 8'h6F: s=8'ha8;

    8'h70: s=8'h51; 8'h71: s=8'ha3; 8'h72: s=8'h40; 8'h73: s=8'h8f;
    8'h74: s=8'h92; 8'h75: s=8'h9d; 8'h76: s=8'h38; 8'h77: s=8'hf5;
```

- 8bit inputs and outputs
- Defines 8bit functions
- Output a unique value corresponding to an input

# SBOXのGDS生成結果（Klayoutで表示）



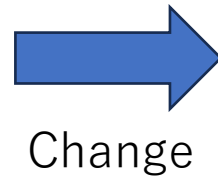
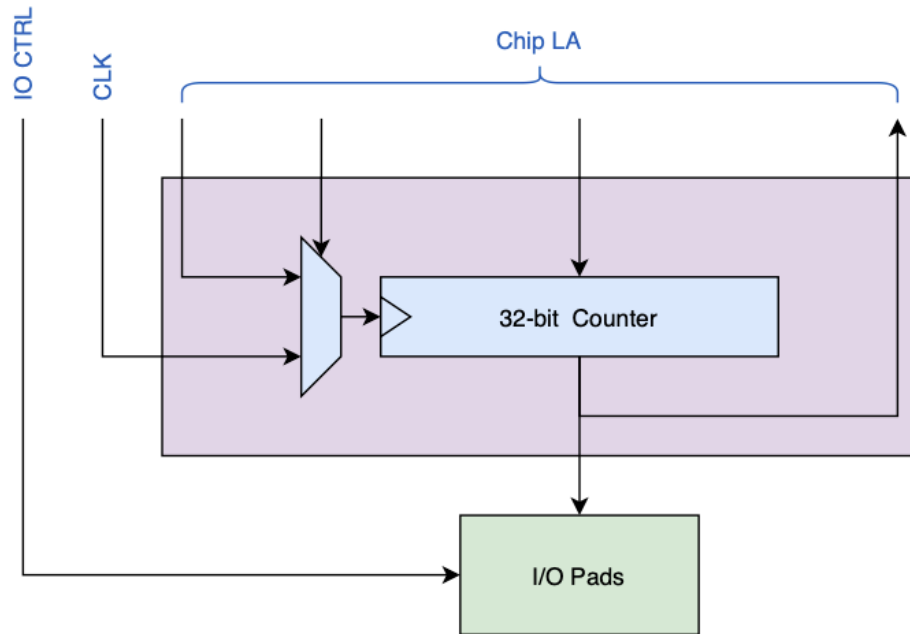
コマンド

`./flow.tcl -design sbox`

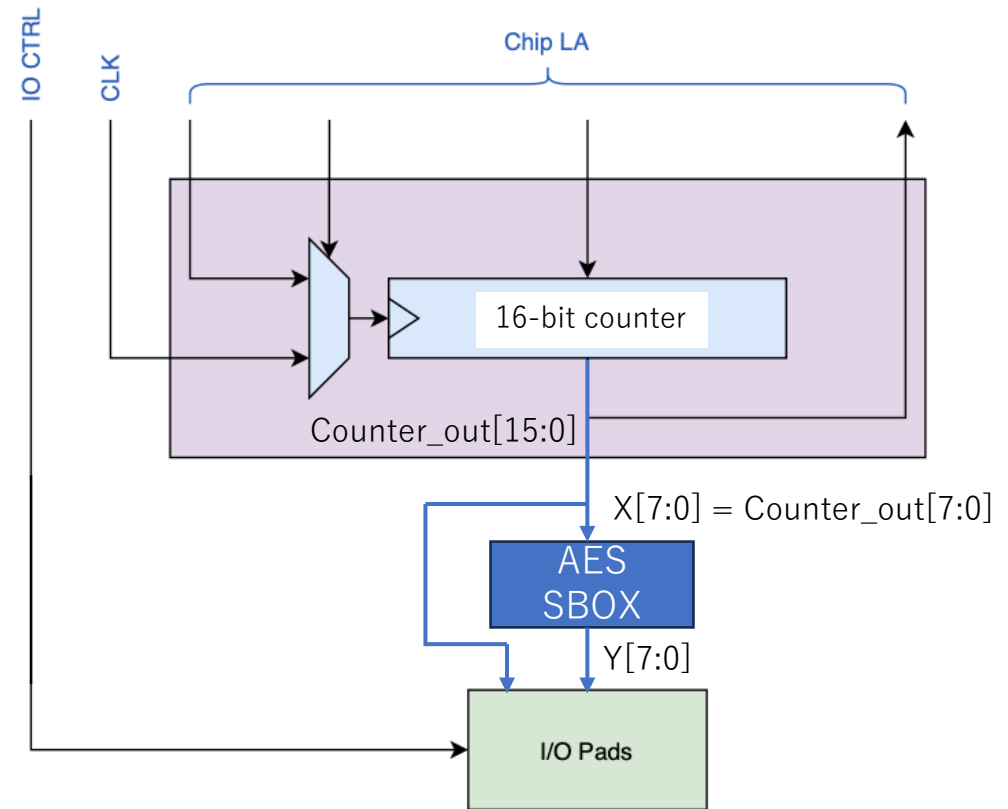
VerilogからGDSまで一気に生成

# Embedding AES SBOX into the counter example

Original counter circuit example  
in a caravel user project



Our design



# Change points in User\_proj\_example.v

```

user_proj_example - メモ帳
ファイル(F) 編集(E) 形式(O) 表示(V) ヘルプ(H)
assign rst = (~la_oenb[65]) ? la_data_in[65]: wb_rst_i;

counter #(
    .BITS(BITS)
) counter(
    .clk(clk),
    .reset(rst),
    .ready(wbs_ack_o),
    .valid(valid),
    .rdata(rdata),
    .wdata(wbs_dat_i[BITS-1:0]),
    .wstrb(wstrb),
    .la_write(la_write),
    .la_input(la_data_in[65:64-BITS]),
//
    .count(count)
    .x(x), .y(y)
);

endmodule

module counter #(
    parameter BITS = 16
) (
    input clk,
    input reset,
    input valid,
    input [3:0] wstrb,
    input [BITS-1:0] wdata,
    input [BITS-1:0] la_write,
    input [BITS-1:0] la_input,
    output reg ready,
    output reg [BITS-1:0] rdata,
//output reg [BITS-1:0] count
    output wire [7:0] x, y
);
    reg [BITS-1:0] count;

    always @(posedge clk) begin
        if (reset) begin
            count <= 1'b0;
            ready <= 1'b0;
        end else begin
            ready <= 1'b0;
            if (~la_write) begin
                count <= count + 1'b1;
            end
            if (valid && !ready) begin
                ready <= 1'b1;
                //rdata <= count;
                rdata <= {x, y};
                if (wstrb[0]) count[7:0] <= wdata[7:0];
                if (wstrb[1]) count[15:8] <= wdata[15:8];
            end else if (!la_write) begin
                count <= la_write & la_input;
            end
        end
    end
end

assign x = count[7:0];
AES_SBOX_ENC sbox (.x(x), .y(y));

endmodule
default_nettype wire

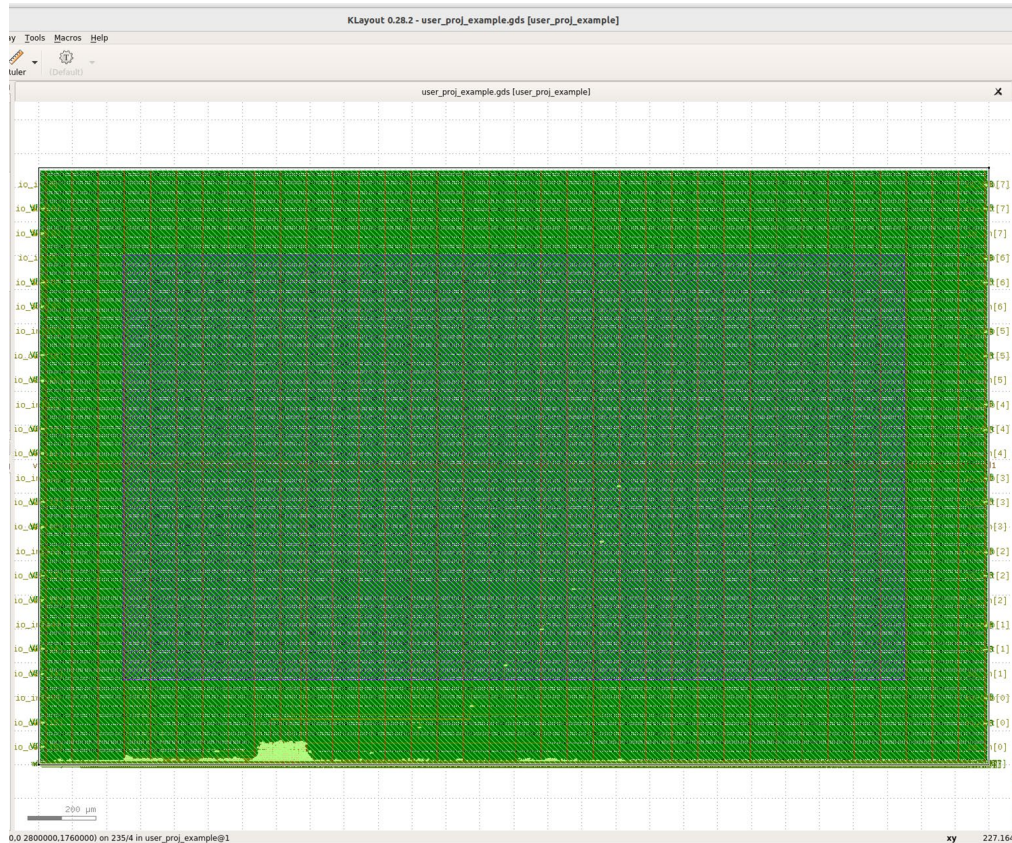
```

- In counter subcircuit:
  - Comment out “counter” port
  - Add “x/y” ports
- In counter module:
  - Comment out “count” output
  - Add wire type “x/y” output
  - Add register type “count”
  - Assign “x” = “count”
  - Insert sbox subcircuit

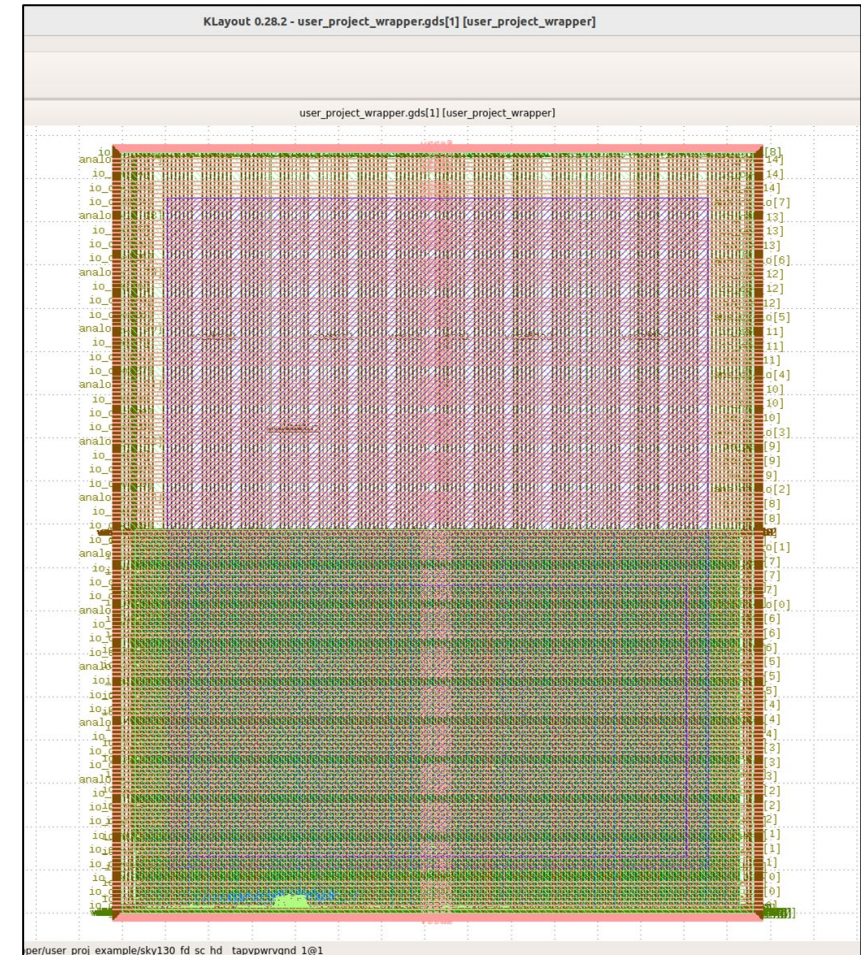


# User領域のGDSファイル生成

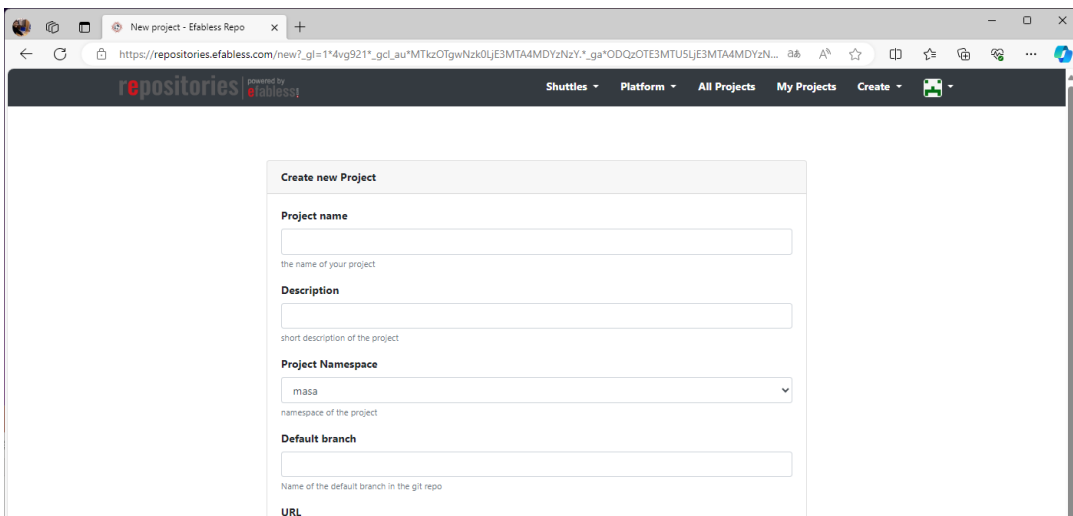
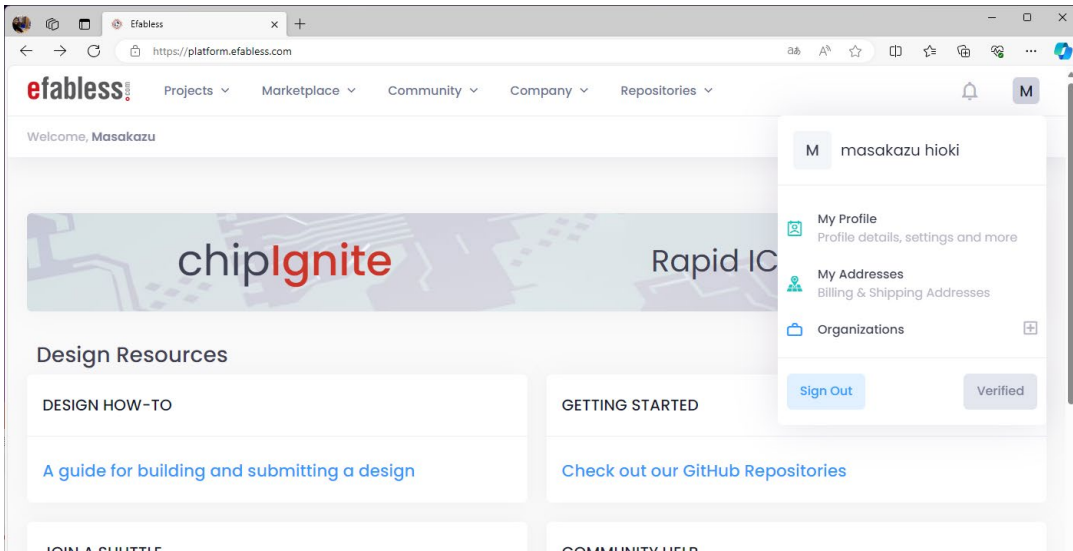
Example counter circuit + our sbx



User area gds including our IP

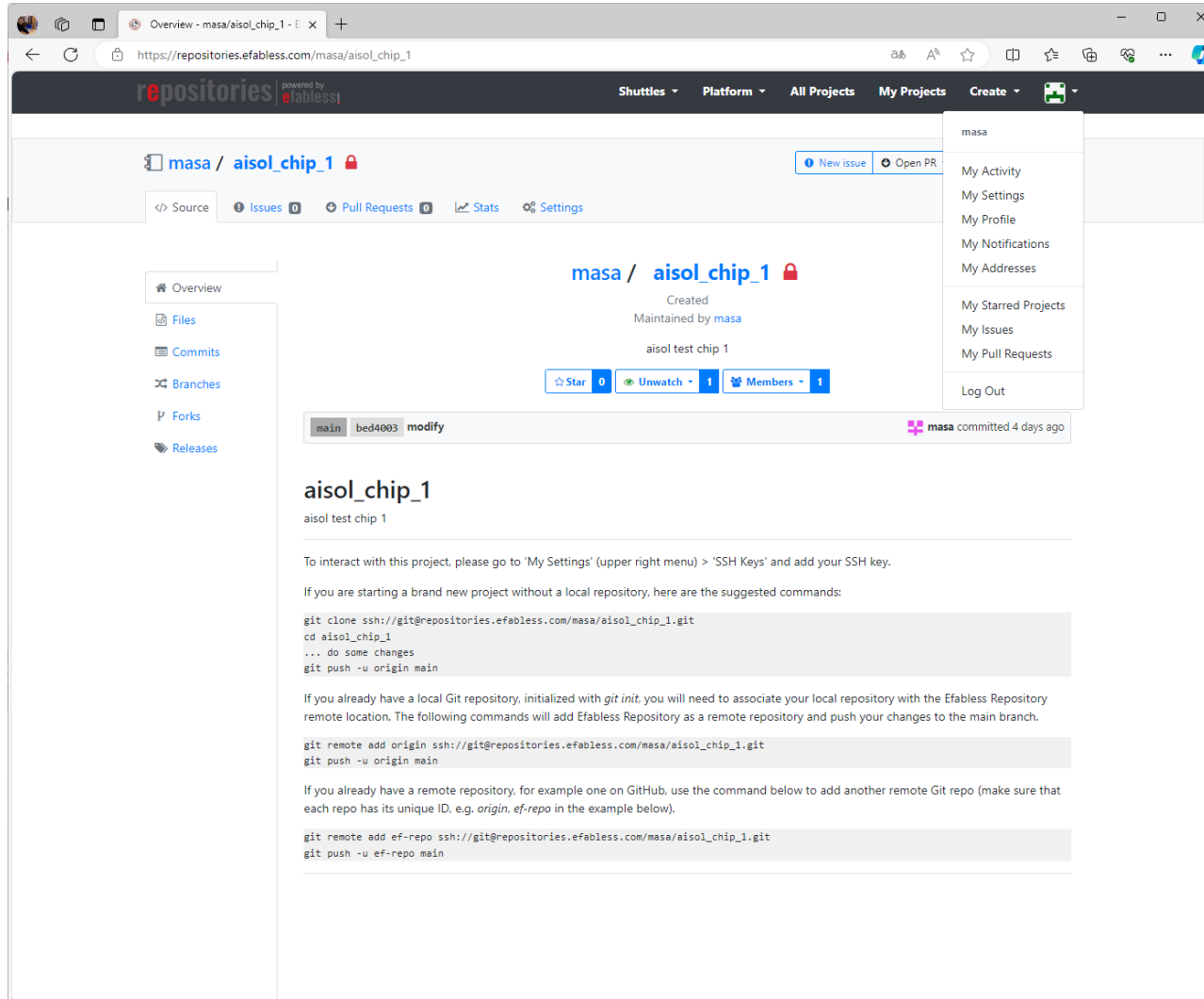


# Chiplgniteシャトルにおけるデータサブミットへの道



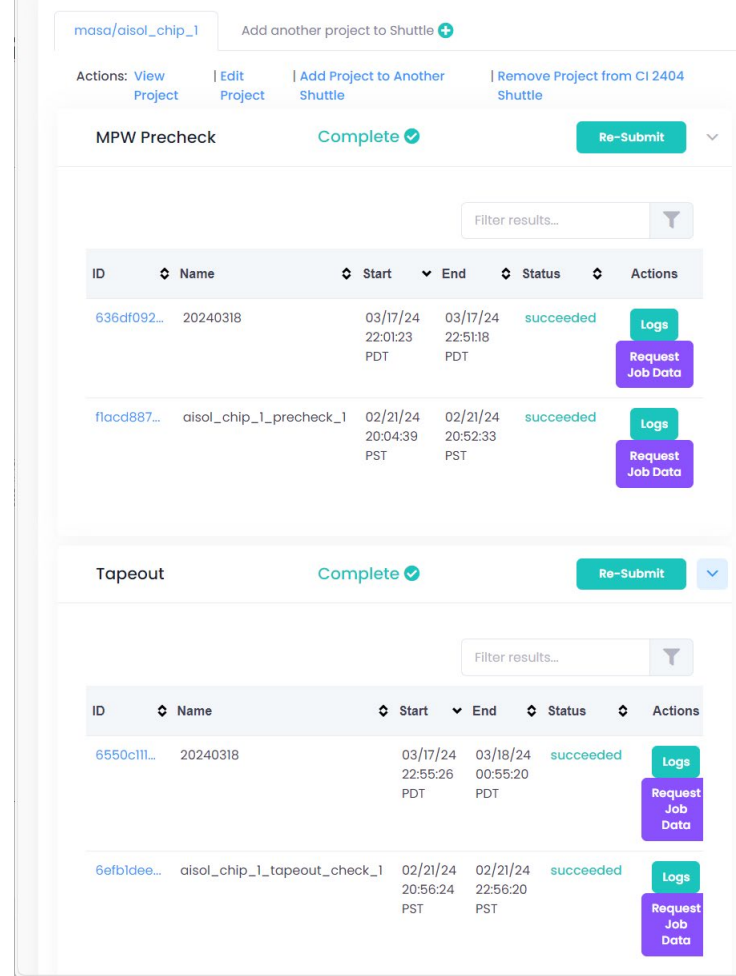
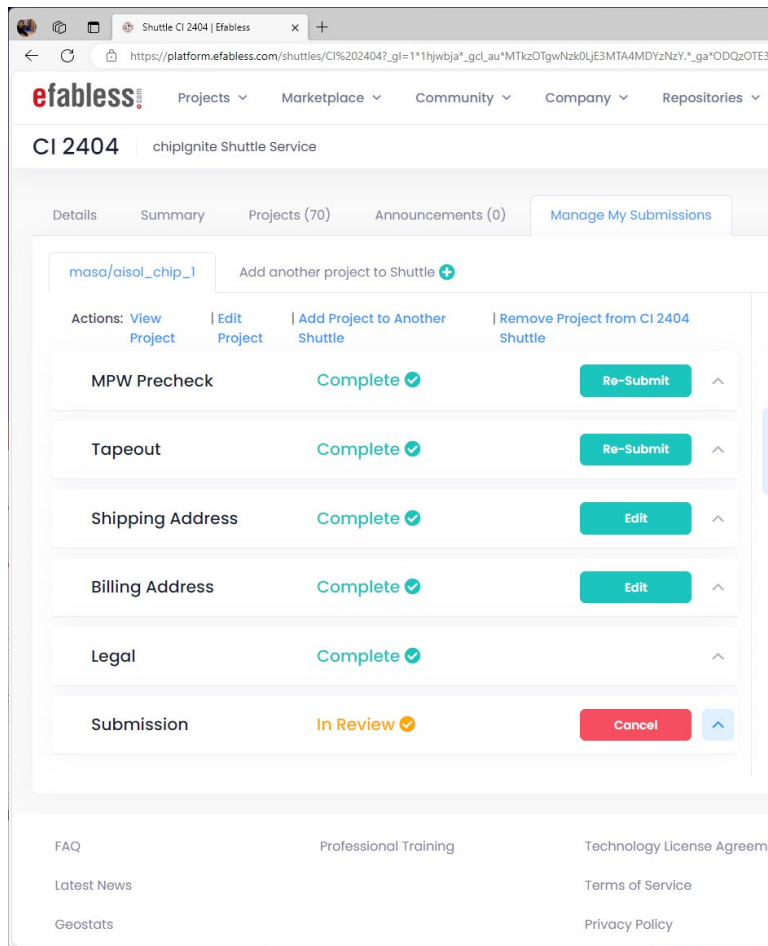
- アカウント作成 今回はaist-solutionsメールアドレスでLinkedInアカウントを作成し、efablessのサイトでアカウント作成
- ブラウザ右上のアイコンをクリックし、profileとbilling/shippingアドレスを作成
- ブラウザ右上のアイコンをクリックし、verificationを実行（数日で完了）
- レポジトリからCreate repositoryを選択しレポジトリ作成
- 今回は、project name, description, privateをチェック、create readmeをチェック、=>create
- gitの画面に移り、SSH key作成と登録、レポジトリ作成
- 画面上のシャトルをクリック、シャトルCI2404を選択

# レポジトリ作成



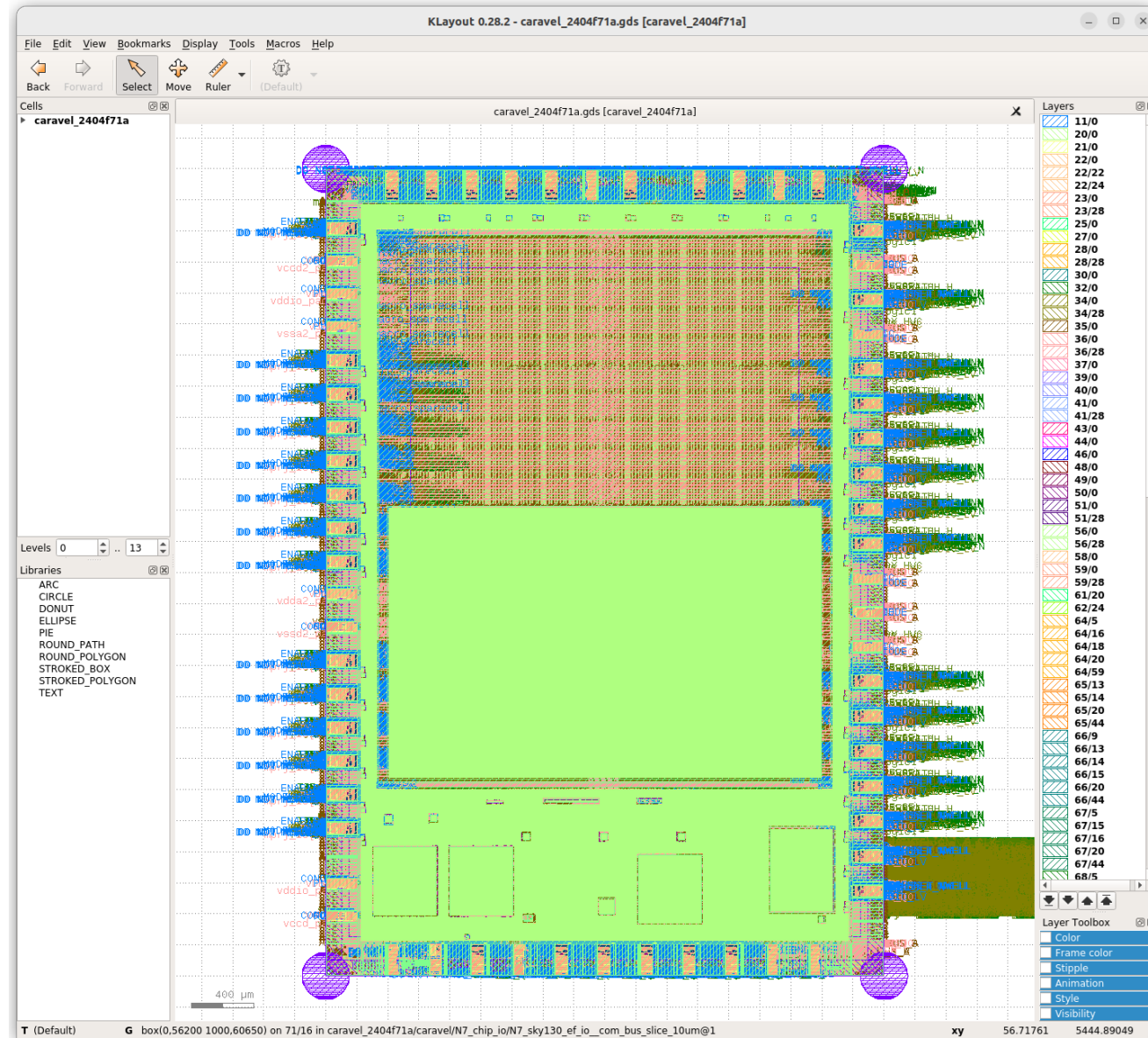
- 右上のアイコンをクリック、Mysettingを選択
- 左のSSH Keysをクリック ssh keysのページに移動
- 手順に従ってローカルマシンでssh keyを作成
- ページ右上のadd ssh keyをクリック、作成されたid\_rsa.pubの中身をボックスにコピーしadd。
- 一番上の手順でレポジトリをクローン
- ローカルレポジトリにファイル追加
- git add, git commit, git push

# データサブミッションプロセス

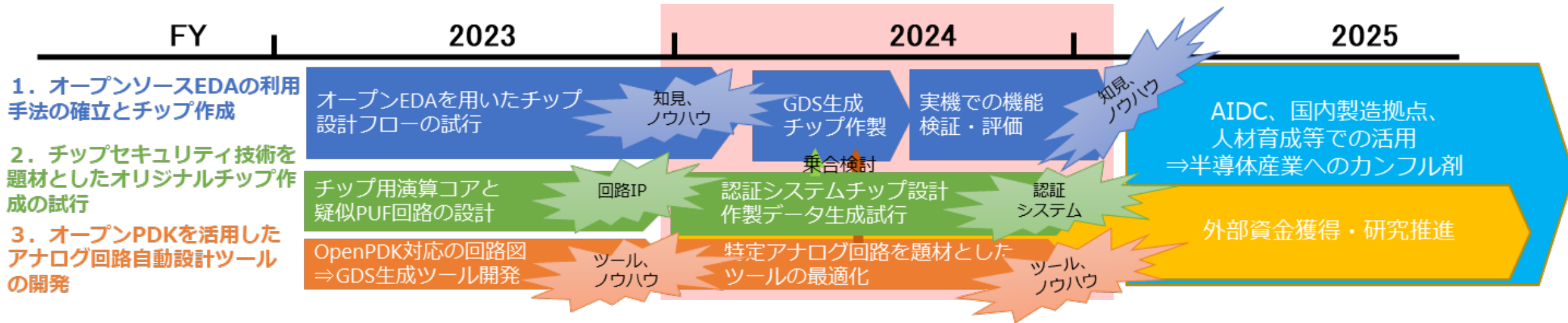


- Manage My Submissionsをクリック、シャトルにレポジトリを追加する。add this project to CI2404 shuttle
- 事前にローカルでプリチェック済みデータをプッシュし、再度プリチェックする。
- MPW precheckをサブミット、ジョブ名を入力・ジョブ実行
- テープアウトはMPW precheckのコンプリートデータを選択してジョブをサブミット
- シッピングアドレス、ビルディングアドレスを編集。最初に入力したデフォルトをラジオボタンで選択可能。
- Legal: terms and conditions利用規約と export compliance輸出管理に同意
- サブミッションで終了
- と思いきや、efablessによるデザインレビューが行われた

# ファイナルGDS



# 今後の展開



- ① Caravel統合環境を用いたサンプル回路のテープアウトと実チップでの性能評価
- ② HWセキュリティチップの個別回路IPの設計完了とそれらの統合、レイアウトデータ生成
- ③ オペアンプ回路を題材として性能仕様から回路ネットリストを生成するツールの開発
- ④ 得られた知見の公開
- ⑤ Chiplgniteシャトル試作チップの評価

# まとめ

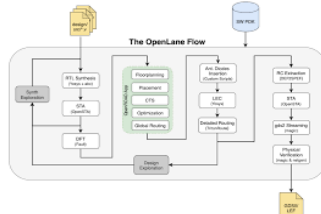
## 背景と問題点

- “誰でも手が届く半導体”に向けたオープンソースEDAの世界的な潮流
- 国内の利用事例が少なく情報が不足気味だったが、増え始めてきた

## 本研究の目的と想定するアウトカム

- オープンソースEDAの利用方法等の知見を蓄積・公開し普及促進を図る
- 期待する効果：国内半導体の活性化や人材育成、SCR用PDK開発や設計フロー構築

半導体化のハードルを  
下げる取り組み@dapa  
Open Source EDA flow



Open Shuttle Program



国内OpenSourceHWの取り組みの  
さらなる活性化に助力

オープンソースEDA  
ツール群/PDKを寄っ  
てたかって繰り返し  
試行する！！

期待されるアウトプット

研究項目1: ツール活用/ノウハウ  
や知見の蓄積

研究項目2: セキュア回路やセキュア  
システムのIP群の獲得

研究項目3: 容易にアナログチップを作  
れる自動設計ツールの創出

期待する効果



- 国内半導体の活性化
- 人材育成

- SCR用PDK開発
- 設計フロー構築



# 以降、予備スライド

## で、結局オープンソースEDAはどうか？

- とても気軽/手軽にVerilogからGDSを生成できる
- オリジナルチップに手が届きそう
- テクノロジは130nmと180nmしかない
  - 以前、90nmのアナウンスはあったが音沙汰ない
  - 先端プロセスにも対応できるみたいなことは書いてあったが、そのためにどれくらいの作業量が必要か未知数
- ツールはマルチスレッドに対応していないので、スピードアップの余地あり
- 階層設計の深さは2まで。
- ハードマクロの余白が意外と大きい印象（検証の余地）

## で、結局オープンソースEDAはどうか？

- ピン数の多いデザインではピン数律速で物理面積が大きくなる（検証の余地）
- 実用性のあるデザインでPPAを評価したい
- Caravelハーネスに内蔵のロジアナ、wishbornバス、RISC-Vの情報が少ない（実チップで確認）
- 有償シャトルプログラムChiplgnite
  - ユーザーサポートの反応が極めて鈍い
  - デザインレビューは（個人的に）大変だった