

はじめて Chipathon に参加するために
勉強したほうがよいこと

土谷 亮 (滋賀県立大学)

自己紹介



土谷 亮 / Akira Tsuchiya

2005年 京都大学 博士 (情報学)

～2017年 京都大学 情報学研究科 助教

～現在 滋賀県立大学 工学部 准教授

卒論から現在に至るまでCMOS集積回路設計
最近は量子コンピュータ用のICをやったり
怪しいセンサーを作ったりしている

2020年ごろ？ アナジックス森山さん/MinimalFab を知って
オープンソースシリコンに目をつける

2022年 学生を煽って OpenMPW-6 でテーブアウトさせる

2022年 Boris Murmann と池田先生に煽られて Chipathon に参加

2023年 いろいろ煽ったり煽られたりして Team Japan を結成

今日の内容

タイトルに「勉強したほうがよいこと」と入っていますが...

どちらかというと「知っておいて欲しいこと」がメイン

- そもそも Chipathon って何
- 何がやりたいのか
- どういうスタンスで望めばいいのか

技術的な話はあまりできません

まずはじめに

「勉強」しないとダメ？

教科書を読んで、問題を解いて、
というタイプのお勉強は別に...

とりあえず見よう見真似でやってみて動けばOK
動かなかったら聞けばいい

「これから回路屋としてやっていくぞ！」という人は
体系的に勉強した方がいいですが

細けえこたあいいんだよ

弟子「普通 修行っていや長い時間かけて基本をくり返したり
技を伝授したりするもんじゃねエのか？」

弟子「いいのかよ？ シロウトのオレがこんな思いつきの
即席戦法 身につけちまって」

(中略)

師匠「それとも お前何十年も修行して達人にでもなるのを
待ってから戦場に出るつもりか？ 気の長げエ話だな」

— 三浦建太郎「ベルセルク」24巻より

目的に応じて

学問・業務としてやるなら

再現性, 汎用性, 条件の明確化, etc.etc. が必要

→ 理論に立脚した説明が必要

電気回路, 電子回路, 半導体工学, 電磁気学...

趣味でやるなら

動いたしおもしろいからヨシ！

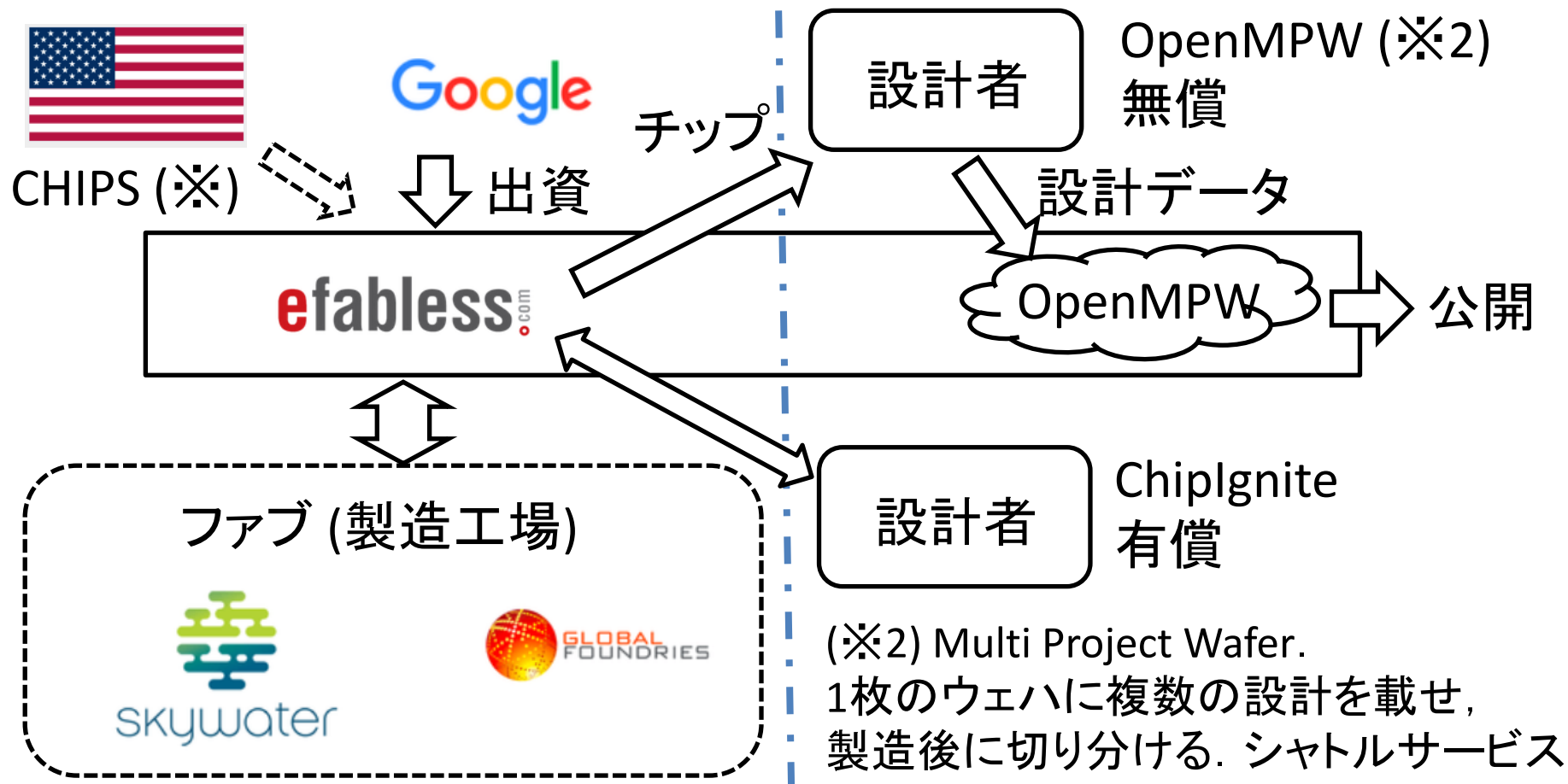
これまでは「趣味でやる」ことが不可能だった

(どうがんばっても個人でできるのは FPGA まで)

「趣味で集積回路をやる」のは完全に新しい世界

Chipathon とは何ぞや

その前に OpenMPW の仕組みについて



(※2) Multi Project Wafer.
1枚のウェハに複数の設計を載せ、
製造後に切り分ける. シャトルサービス

Chipathonの運営母体

Chipathon の正式名称(?)は

IEEE SSCS “PICO” Open-Source Chipathon



IEEE

Institute of Electrical and Electronics Engineers の略だが
IEEE (アイ・トリプル・イー) で固有名詞
超巨大専門職団体. 学術だけでなく標準化などもやる



SSCS

Solid-State Circuits Society

IEEE内で専門分野を細分化したソサイエティの一つ.
集積回路の回路設計を専門に扱う

SSCS PICO Chipathon

SSCS の中の委員会

Technical Committee on Open-Source Ecosystem (TC-OSE)
が運営する

SSCS PICO Program (Platform for IC Design Outreach)
の活動の一つが “Chipathon”

(従来の) 集積回路設計の専門家集団 (SSCS) が
オープンソースで広く多くの人々がIC設計できるような環境 (PICO) を
作ろうとしていて, そのための実験的な活動が **Chipathon**

Chipathon = Chip + Marathon

ちなみに

Boris Murmann : TC-OSE の Chair. (Stanford U. → U. Hawaii, 教授)

Sadayuki Yoshitomi (JR6PLB) : TC-OSE member

というわけで

Chipathon の目的は

「Outreach のための実験」

= 専門外の人でも作れるようにする

「知識がないから恥ずかしい.
こんなこと言ったらバカにされるのでは...」
などと考えるはいけない

むしろ

「ここが分からん. 調べてもでてこない」

を言ってもらわないと困る

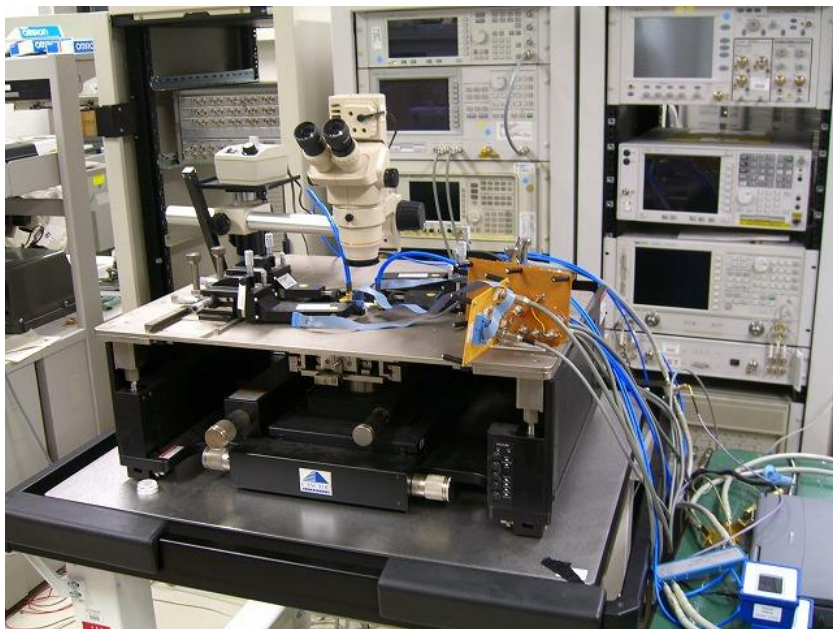
Q&A?

- Chipathon と efabless の関係は？
→ 関係ない. ただのお客さん
- お金はどこから出てるの？
→ SSCSの予算から

Chipathon 2023 がやりたいこと

“lab bench on a chip”

測定器をチップ上に載せてしまおう



別の言い方をすると “AnalogDiscovery2 on a chip”

やりたいことイメージ



(Photo from TinyTapeout.com)

OpenMPW の UserArea (10 mm²)
を一人で埋めつくせる人が
どのぐらいいるか？
→ いるとしたらそいつはガチ勢だ

Outreach のためには
「ちょっと作ってみた」回路を
たくさん載せて測りたい

デジタル回路では TinyTapeout が
すでに実現, サービスとして展開

このアナログ版をやりたい

AnalogDiscovery2

Digilent社の「万能測定器」

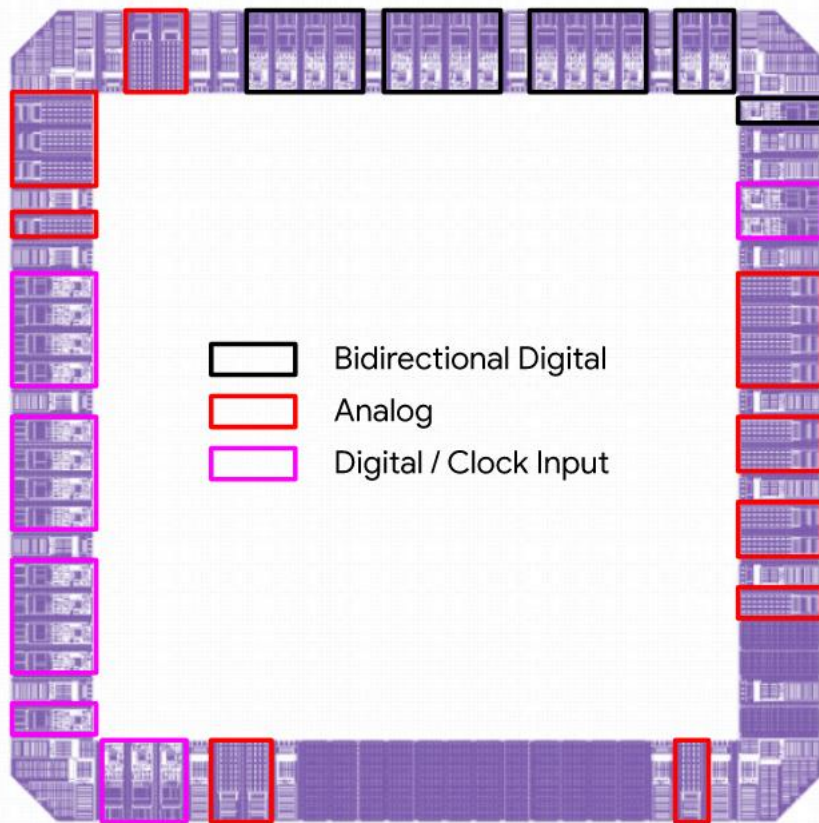
- ・パソコンにつないで使う
= ディスプレイやボタンがない小型
- ・2chオシロスコープ (14bit, 100 MSa/s)
- ・2ch 任意波形発生器 (14bit, 100 MSa/s)
- ・16chパターンジェネレータ (100 MSa/s)
- ・16ch 仮想デジタルIO
- ・16chロジックアナライザ
- ・2入力/出力デジタルトリガ
- ・2出力プログラマブル電源 (5 V, 2.1 W)
- ・電圧計 (AC/DC)
- ・ネットワークアナライザ (10 MHz)
- ・スペクトラムアナライザ
- ・バスアナライザ (SPI, I2C, UART, パラレル)



4万円ぐらい
たぶんその筋の人は
一人1台(以上)もってる
最近3が出た

じゃあ AD2 買ってくれば？

集積回路は測定のコストがとてつもなく高い



チップ - 外部をつなぐI/O は
(中の回路に比べると) 巨大

信号の入出力に使えるのは
わずか51本

15 Bidirectional Digital I/O

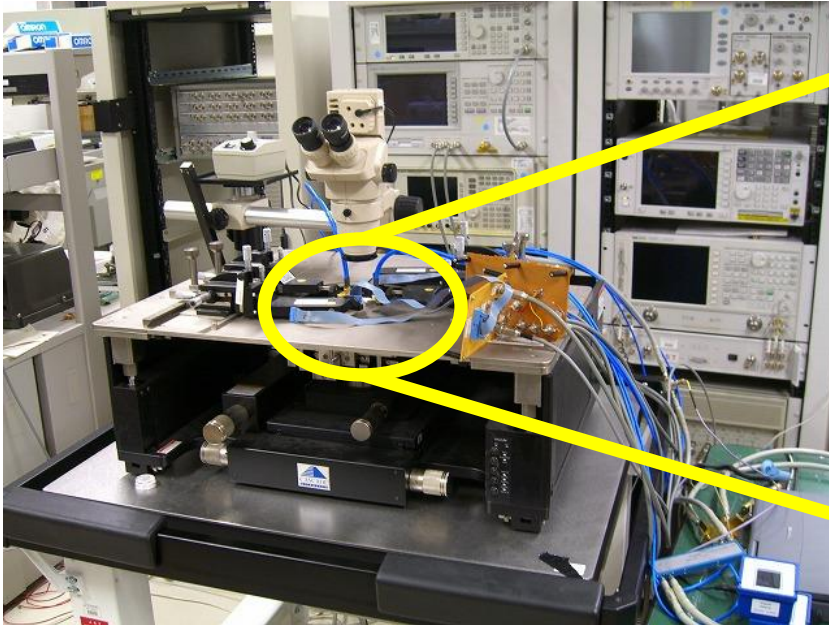
18 Analog I/O

18 Digital Input

ちょっとオシロのプローブ
当ててみよう, のようにはいかない

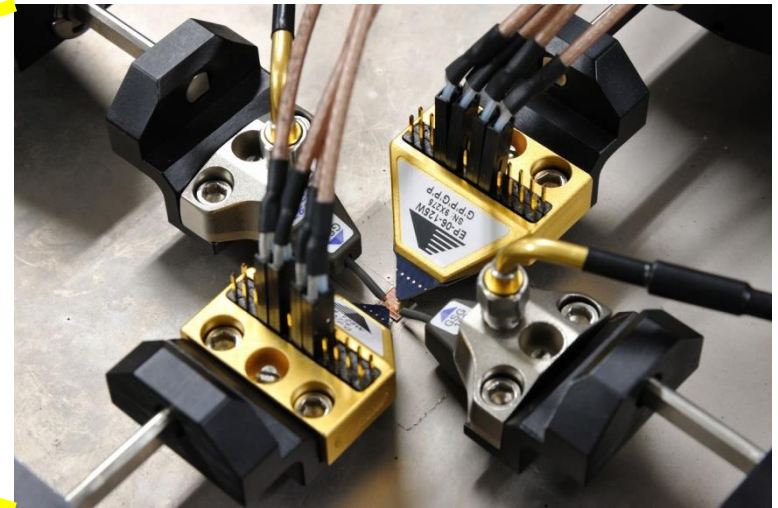
ちょっとプローブ当ててみよう

ということができないわけではない (現実的とは言っていない)



プローバ

μm 単位的位置合わせをする



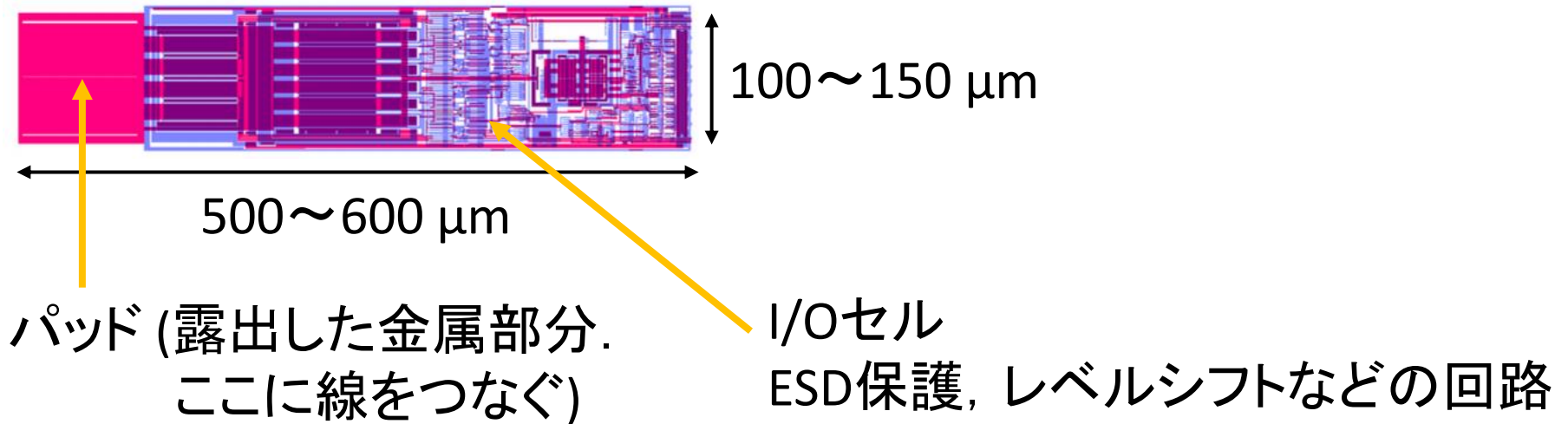
プローブヘッド

100 μm ぐらいの針

1個数十万円, 消耗品

※ どこにでも当てられるわけではない. 専用のパッドが必要

なぜこんなに大きいのか



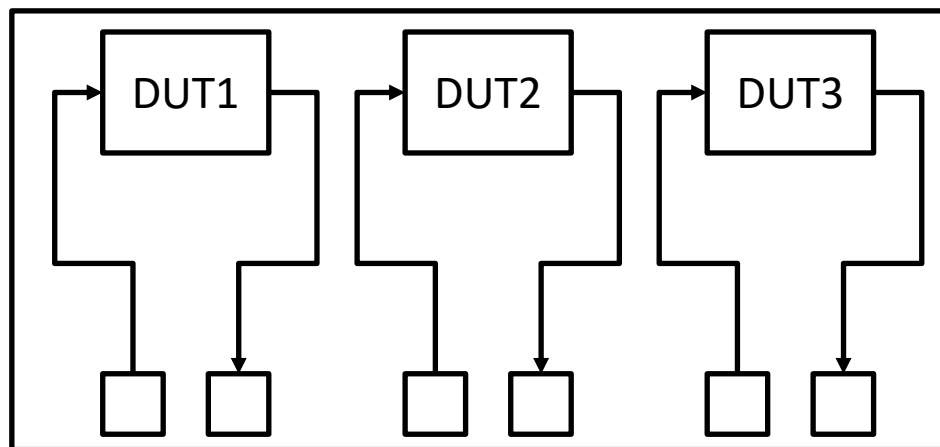
- ・ 静電気による破壊を防ぐ ESD保護ダイオード
- ・ チップ内の電源電圧とチップ外の電源電圧の差を吸収するレベルシフト回路
- ・ フローティングにならないためのプルアップ/プルダウン抵抗

大きさが何桁も違うデバイスをつなぐのは大変

ということとは

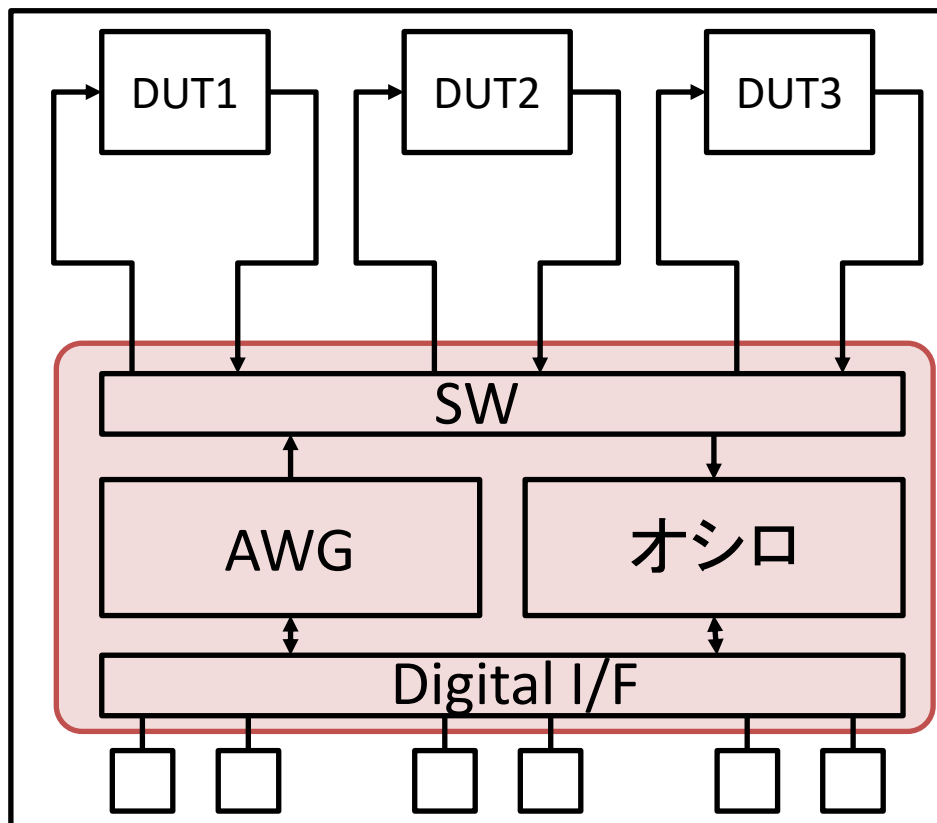
Analog I/O が18本しかないということとは

1つの回路に入力/出力で1本ずつ使うとしても
9個しか回路を載せられない



実際には1つの回路にアナログI/O 2本は少なすぎる
制御電圧も欲しいし出力も複数箇所見たい

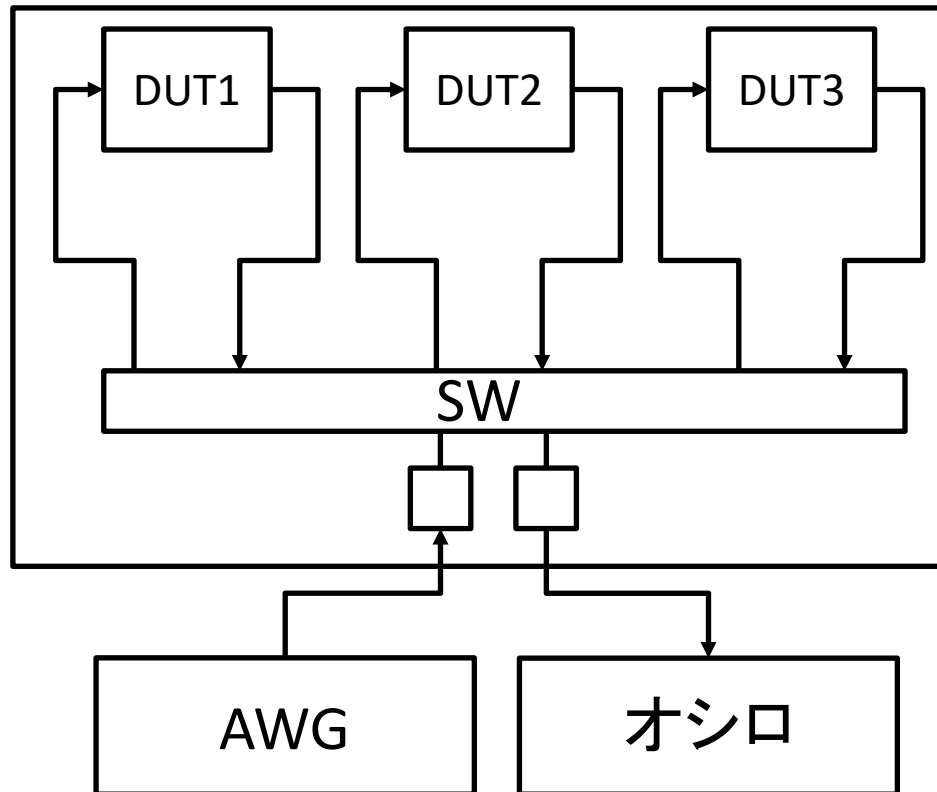
ということで



チップ内に測定器とどのDUTをつなぐかのスイッチを積めば
もっとたくさんの DUT を載せられる

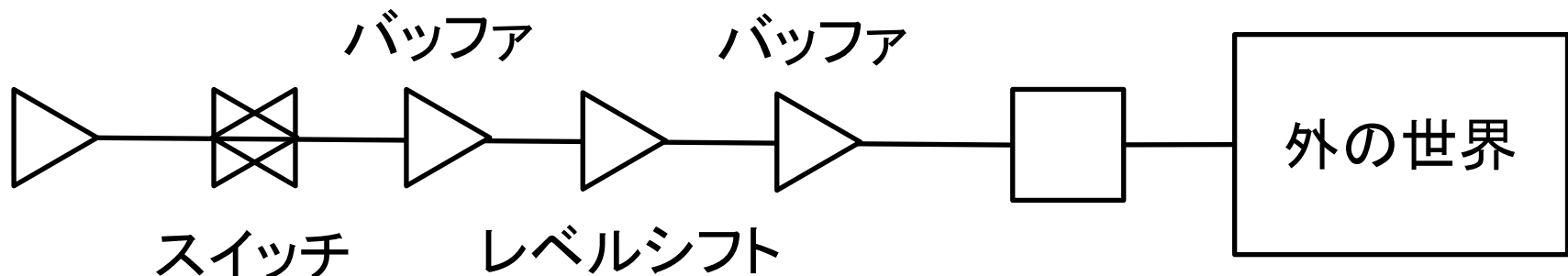
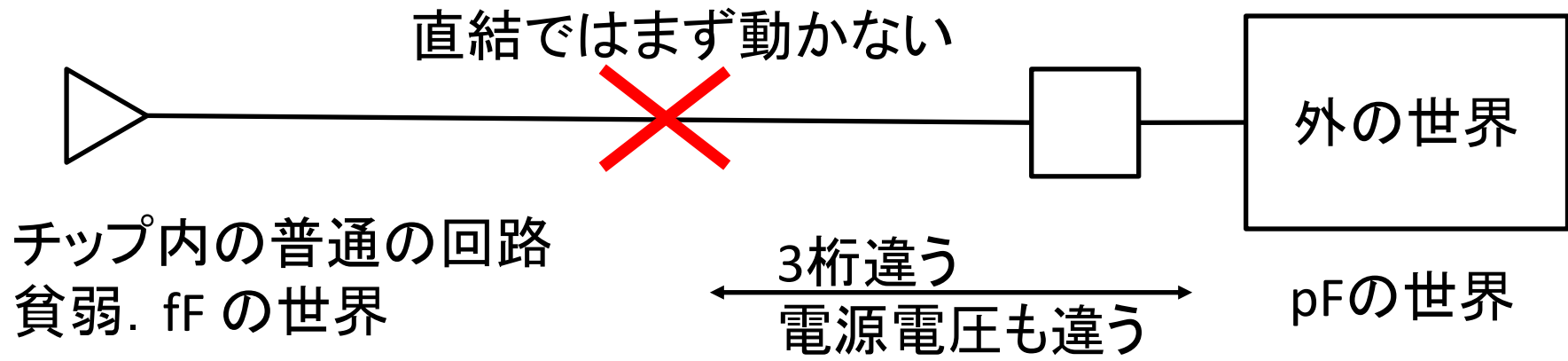
測定器は要るの？

チップにはスイッチだけ載せておけばいいのでは？



大きさが何桁も違うデバイスをつなぐのは大変
なのであまり現実的ではない

アナログ信号とI/O



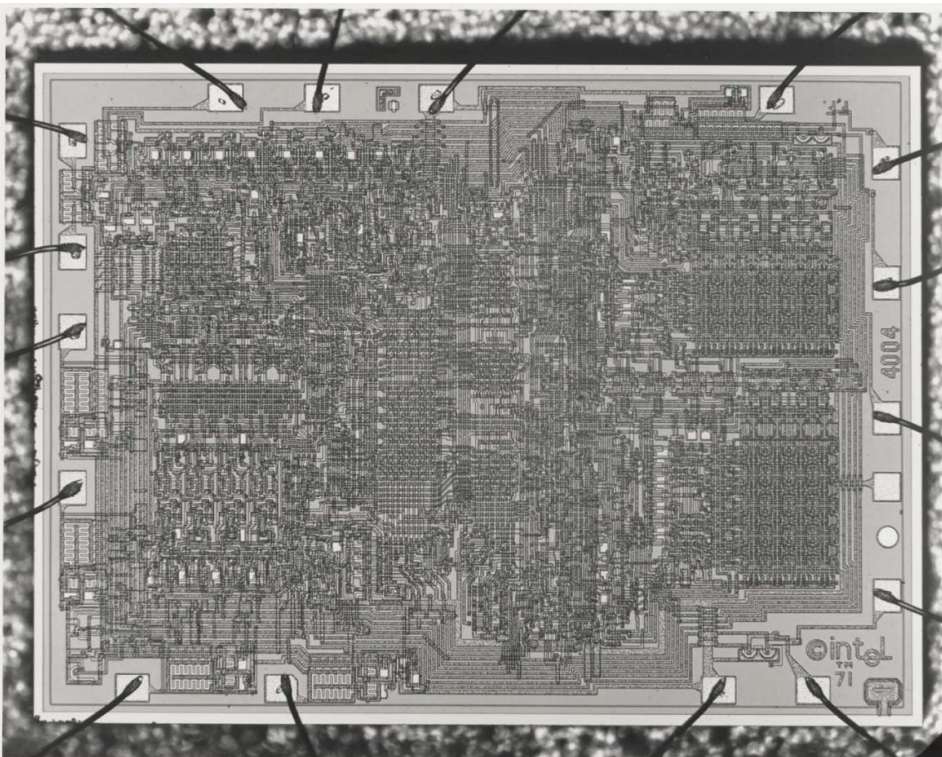
アナログ信号を扱うには「波形を歪ませることなく」
これだけのことができる回路が必要 → かなり困難

Q&A?

- Chipathon は何がやりたいの？
→ on-chip AnalogDiscovery2 であり
Analog^版TinyTapeout

ICを設計しよう

ICを設計する = 最終的にはレイアウト (マスクデータ) を作成すること
ファブに提出するデータには回路図も HDL も含まれていない
レイアウトデータ (GDSII) のみ



Intel 4004 プロセッサの
チップ写真

これが設計されたとき
SPICE はまだ存在していない

(Photo from Intel)

レイアウトを描く

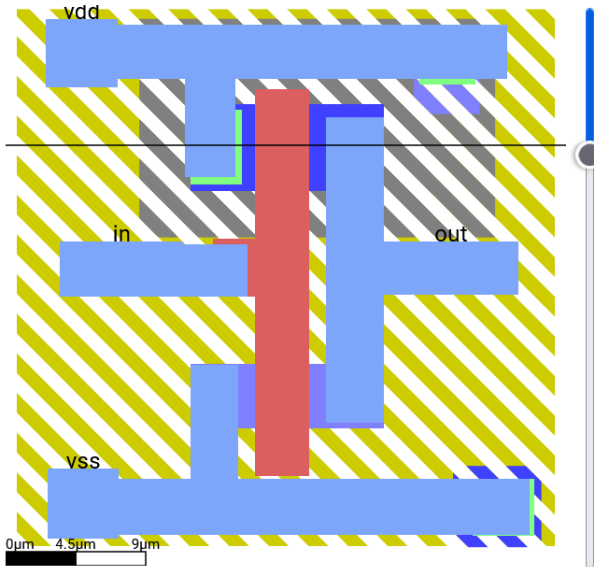
レイアウトを描く, おちうのがどういうことかは
Siliwiz を触ってみるのがおすすめ

レイアウト設計超入門: Siliwiz で遊ぶ

https://note.com/akira_tsuchiya/n/n753b78aed0fc

レイアウト設計と集積回路製造プロセス

https://note.com/akira_tsuchiya/n/n416b7f74b701



規則に従って絵を描いていく

DRC

検証工程 (Verification) の主要な2つのうちの1つ

DRC (Design Rule Check)

描いた絵が製造工場のルールを満たしているかのチェック
これを満たさないと製造してもらえない / 動作が保証されない
プロセスが進むごとにルールが鬼畜になっていく
基本的に小まめにDRCをかけておけばいいのだが...

配線密度ルール (Density rule), アンテナルール (Antenna rule)
のように設計の後半～最終段階で炸裂する地雷が
仕込まれているので油断してはならない

密度ルールは鬼畜

LVS

検証工程 (Verification) の主要な2つのうちの1つ

LVS (Layout Versus Schematic)

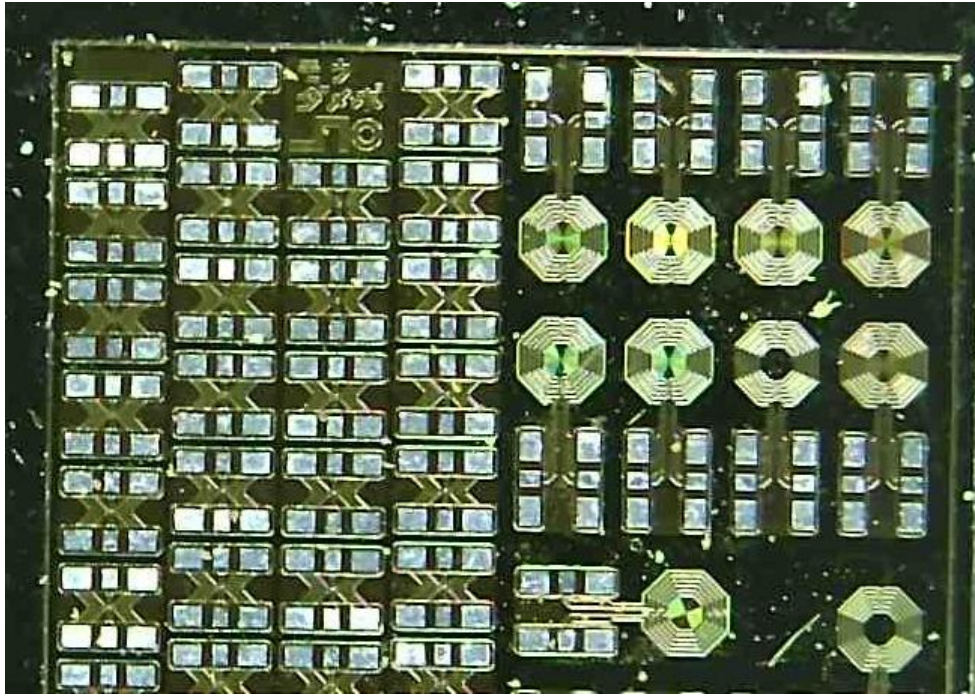
描いた絵が想定している回路図 (スキマティック, schematic) と
等価かどうかのチェック

これを満たさなくても作ってはもらえるが, 100% 動かない

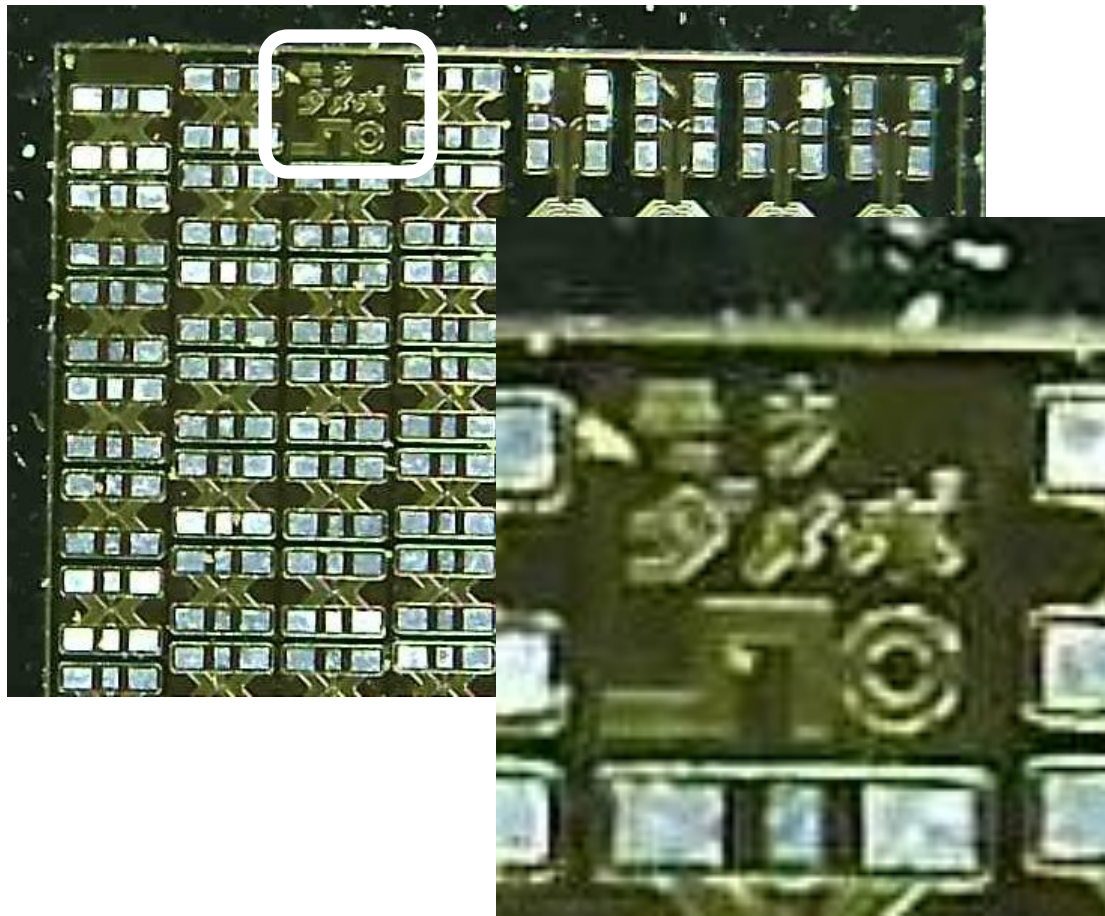
商用ツールを使っても「どこが間違っているか」の検出は
困難なことが多く, 地獄. 冗談抜きに心が折れる

DRCはまだ「どこが間違っているか」は表示されるのでマシ
電源-GNDのショートなどはひたすらに絶望

これは某大学で設計されたチップ



おわかりいただけただろうか



モウ
ダメポ
—|—|○

というわけで

人間の作業には必ずミスがあるので自動化したい

デジタル回路は

高位合成 and/or 論理合成 → 自動配置配線 で

自動的にレイアウトまでできあがるフローが確立されている

例えば OpenMPW の設計フローでは

make user_project_wrapper のコマンド1発で

Verilog からテープアウト可能なGDSII ができあがる

アナログ回路は.....？

Team Japan 目標

Tunable and Process Portable

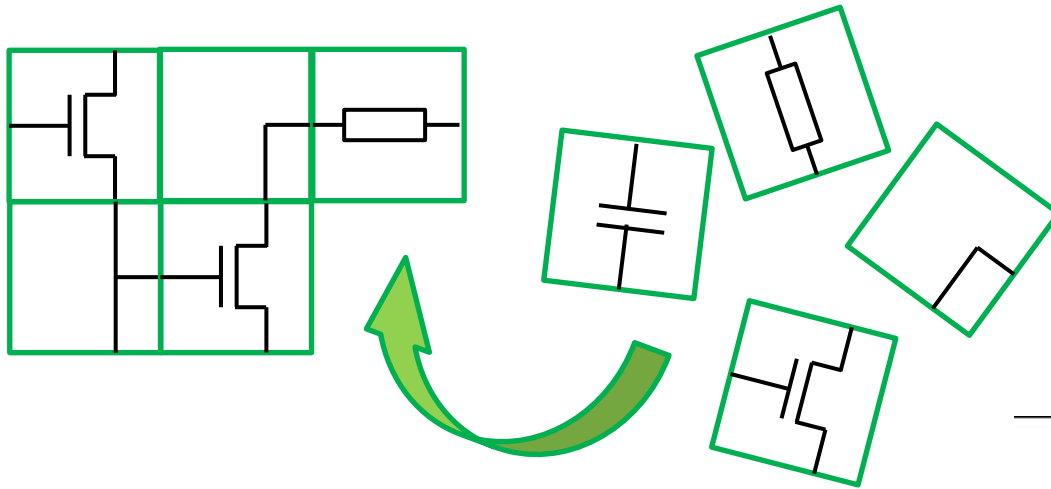
PLL and ADC with an Automatic Layout Generation System

Tunable : 性能をちょっと変えたいときに
いちいち作り直すの面倒くさい.

Process Portable : SkyWater 130nm で1回作ってるのに
GF180nm でまた同じ回路を作らないといけないのは
面倒臭い

すごい性能は出せなくていいから
手抜きができるシステムを作りたい

提案: ブロックベースレイアウト



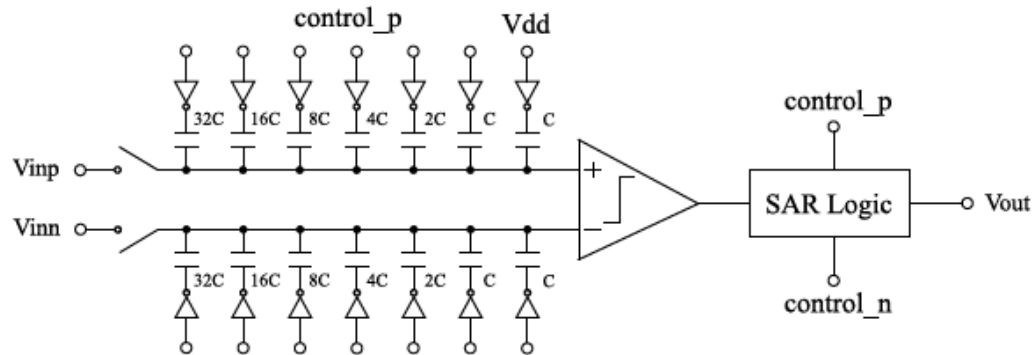
配線も含めてすべての素子を
形状が規格化された
「ブロック」にして並べる



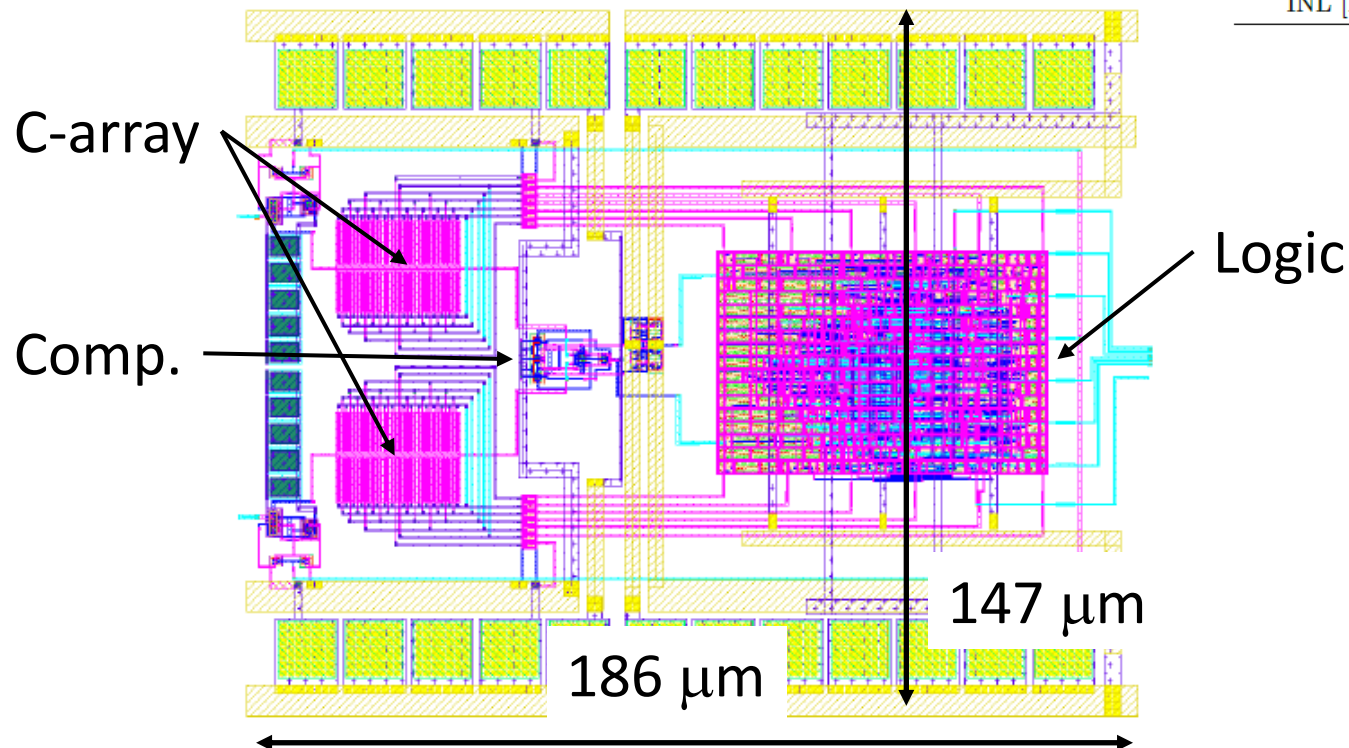
阿南 椋久, 山下 太一, 土谷 亮, 井上 敏之, 岸根 桂路,
「オープンソースのアナログ回路自動設計に向けた
ブロック方式設計のSkywater 130nmプロセスによる実証」,
DAシンポジウム 2022, pp.70-76, 2022.

(Photo: EX-150 from 大人の科学.net)

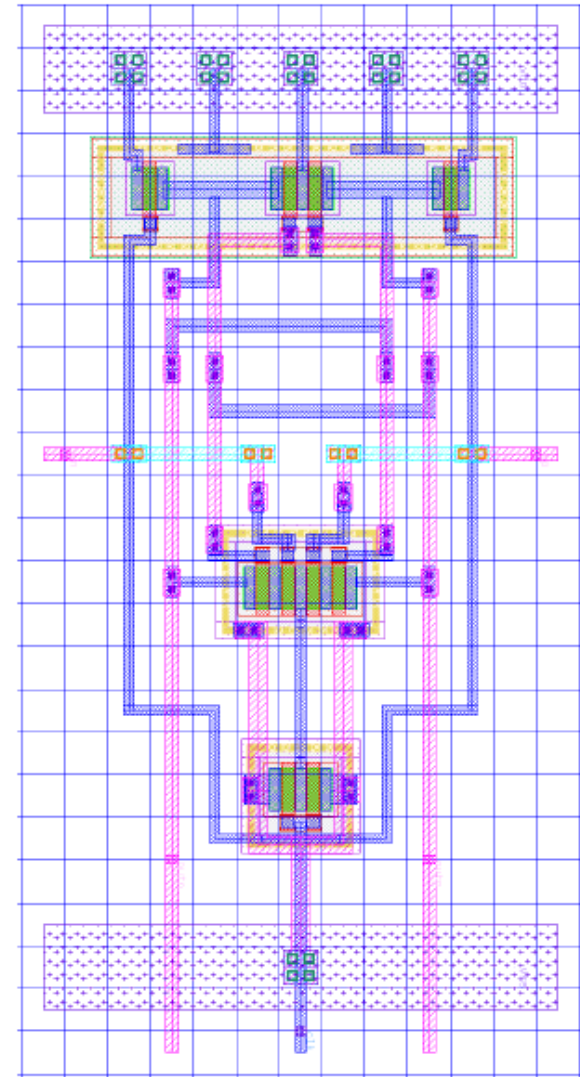
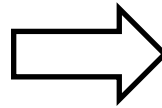
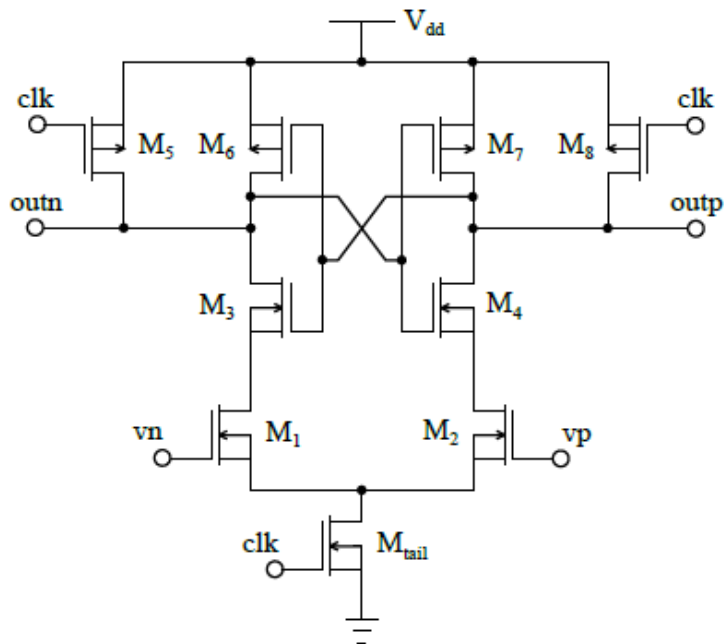
SAR-ADC @SkyWater 130nm



性能	カスタム設計	ブロック方式設計
Resolution [bits]	6	6
Fsample [MHz]	5	5
SNDR [dB]	28.80	27.97
ENOB [dB]	4.49	4.35
SNR [dB]	39.00	38.56
THD [dB]	29.24	28.37
SFDR [dBc]	31.72	30.39
DNL [LSB]	-0.05/0.19	-0.28/0.21
INL [LSB]	-0.19/0.09	-0.16/0.19



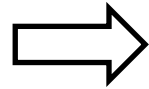
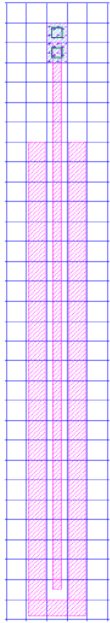
Dynamic Comparator



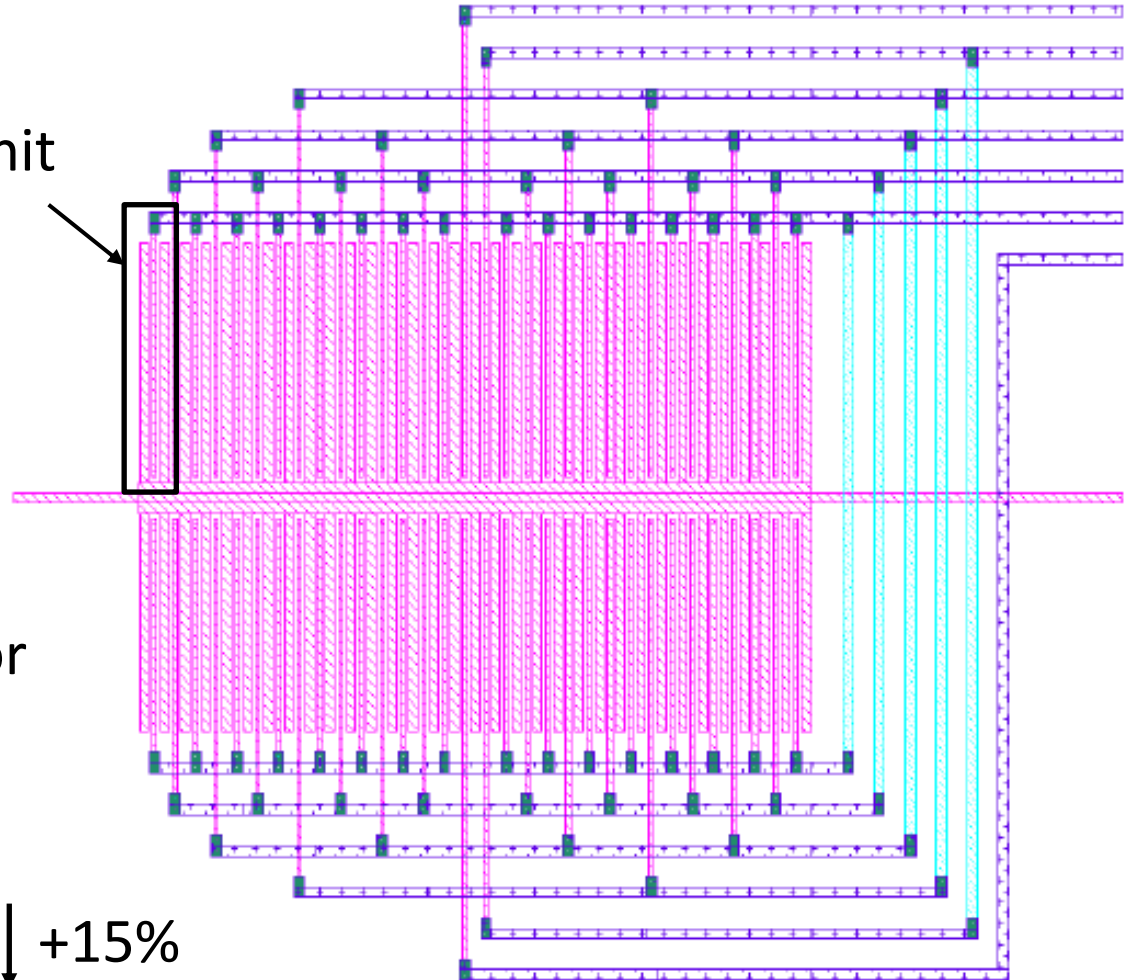
Area:

Full-custom: $206 \mu\text{m}^2$
Block-base: $225 \mu\text{m}^2$ $\downarrow +10\%$

Capacitor Array



Unit



Block-based unit capacitor

Area:

Full-custom: $440 \mu\text{m}^2$
Block-base: $506 \mu\text{m}^2$ $\downarrow +15\%$

Chipathon に向けて

ゴリゴリと回路を設計するだけでなく、
手抜きの仕組みを考えたい

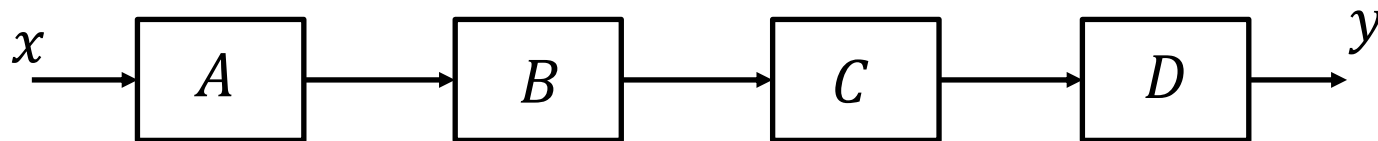
「アナログIC設計」という共通項の下で

- 回路設計, レイアウト設計
 - 増幅器, DAC, 発振器, フィルタ, デジタル回路など回路だけでも多数
- 自動化の仕組み作り
 - アルゴリズム
 - 部品作り
- ツールのテスト, サポート
 - 設計環境もまだ成熟していない

など, いろいろなスキルを集めたい

アナログ回路のとりえ方

アナログ回路が難しい理由



伝達関数のかけ算になる $y = A \times B \times C \times D \times x$

どこか一箇所が狂うと全部狂う

各ブロックはそれほど複雑ではない

各ブロックをきっちり作ればそこまで難しくない... とは言え
前後のブロックが動作に影響を与えるのでやっぱり難しい

SAR-ADC はバイナリサーチだよ, という Jupyter Notebook

https://github.com/atuchiya/openmpw-transistor-level-examples/blob/main/jupyter-notebook/ADC/SAR_ADC_Introduction.ipynb

初学者の方へ

- どんどん質問しましょう
 - ◆ 集積回路設計は「閉じた世界」(だった)
 - ◆ Technical Term が多数, 辞書にも教科書にも載ってないか探しにくい
 - 微妙に定義が統一されていないものも多い
 - 文脈によって意味が変わるものも多い
 - ex) 出力抵抗は高いのと低いのどちらがいいでしょうか？
 - ◆ 一般の電子回路とは違う常識も多数
 - いろんなパラメータが桁で違うので
- ISHI会で用語集を整備中
 - ◆ https://github.com/jun1okamura/glossary_semiconductor_design_terms_for_beginners/

まとめ

- SSCS は Outreach のための仕組みを作りたい
 - ◆ 専門家がガチってもしようがない. 初学者歓迎, といつか初学者がいてくれないと困る
- Team Japan は手抜のための仕組みを作りたい
 - ◆ 貢献の形は回路設計だけではない
- まず触ってみて感想でもなんでも上げよう
 - ◆ "Given enough eyeballs, all bugs are shallow"
— Eric Steven Raymond