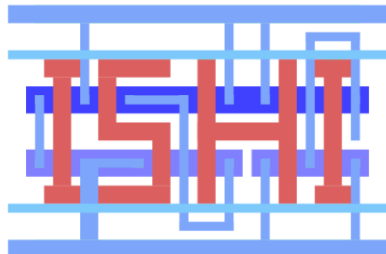


ISHI会 第1回ハンズオンセミナー Siliwiz で知識0から始めるIC設計

土谷 亮 / Akira Tsuchiya
a_tsuchiya@ieee.org



今日の予定

- 集積回路の設計と Siliwiz の使い方
- レイヤーの意味とデバイスの構造
- デザインルール
- シミュレーション
- 素子の特性を調べる
- 論理ゲートを作る

進め方

- ウェブブラウザ上で Siliwiz を使います
- スライドタイトルが「Work #: ...」になっているところは各自操作してみてください
 - ◆ 重いものから軽いものまで全部で18個あります

参考資料

公式のチュートリアル: How do semiconductors work?

<https://tinytapeout.com/siliwiz/>

紹介動画 by Matt Venn

<https://youtu.be/V9xCa4RNfCM>

土谷の書いた資料

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

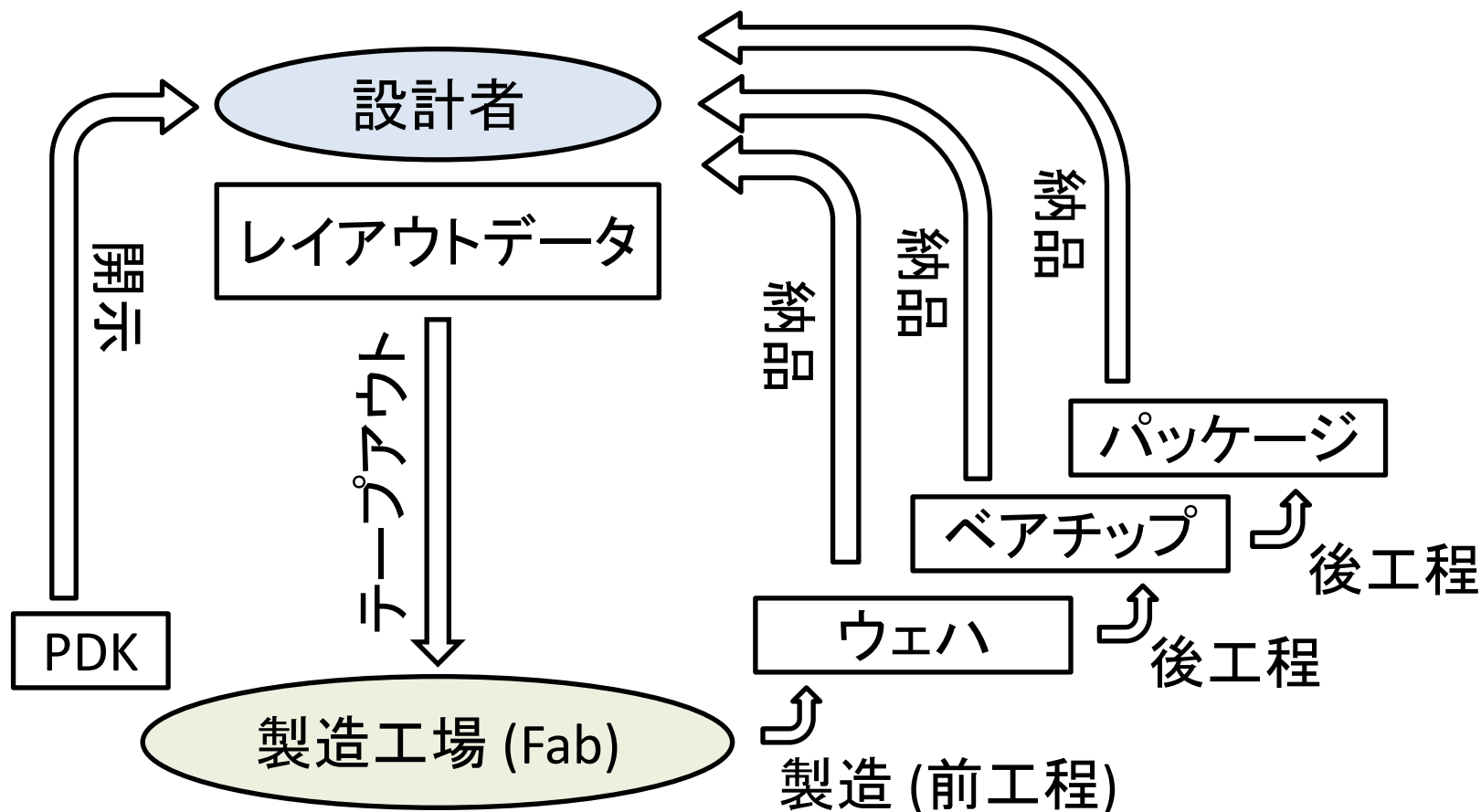
https://note.com/akira_tsuchiya/n/n416b7f74b701

レイアウト設計超入門: 番外編 Siliwizでお絵描き

https://note.com/akira_tsuchiya/n/n0c720b3498b4

集積回路の設計と Siliwiz の使い方

集積回路設計と製造



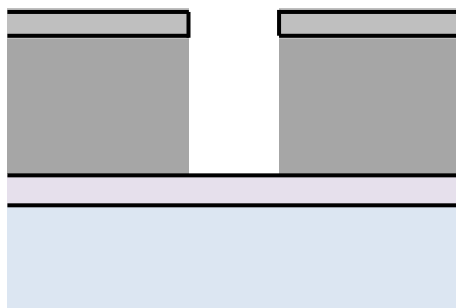
設計者と製造工場のやりとりは「レイアウトデータ」
集積回路設計は最終的にはレイアウトを作ること
昔は磁気テープに記録して送ったから「テープアウト」と言うらしい

レイアウトとは

フォトリソグラフィに使うマスクの形状を指定する 2次元の図形データ

ファイルフォーマットは GDSII がデファクトスタンダード
ストリーム (stream) ファイルとも呼ぶ

フォトリソグラフィ (Photolithography) : 超高度な影絵



フォトマスク

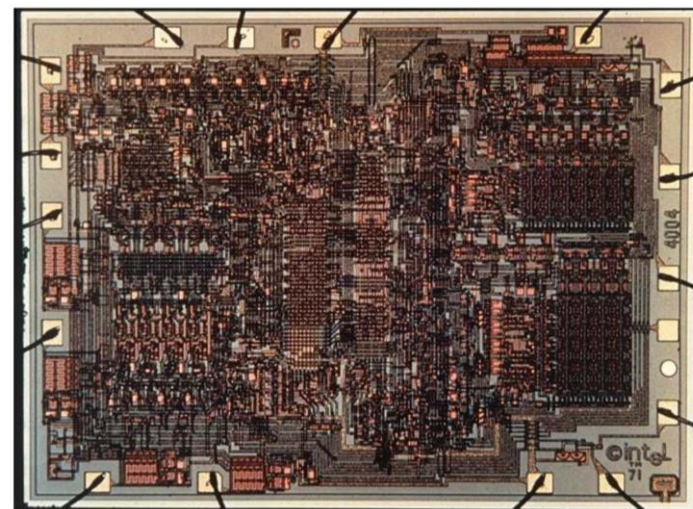
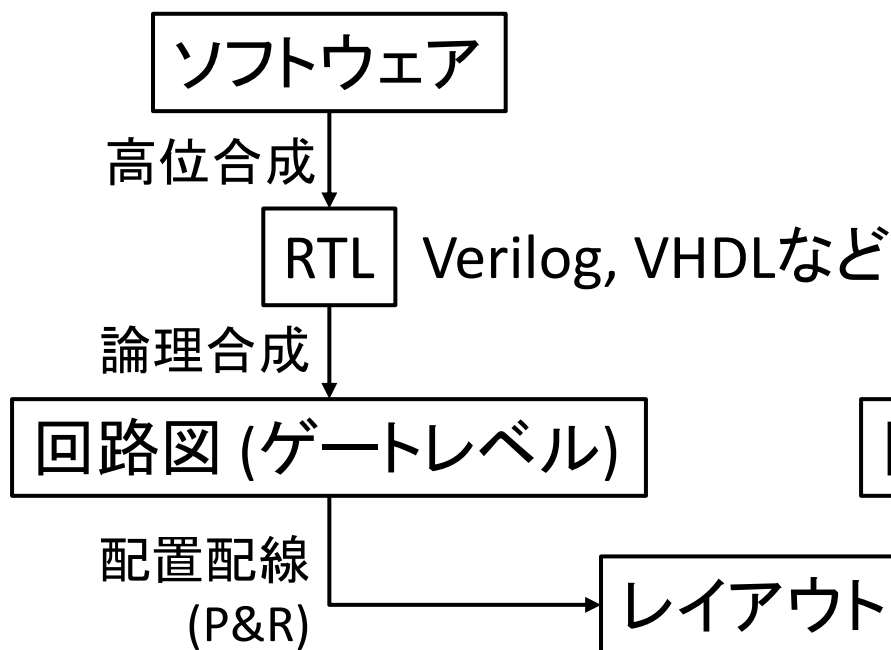
フォトレジスト(感光剤)
シリコンウェハ

設計フロー

昔はレイアウト手描きだった
「模造紙に色鉛筆で描いた」by 小野寺先生

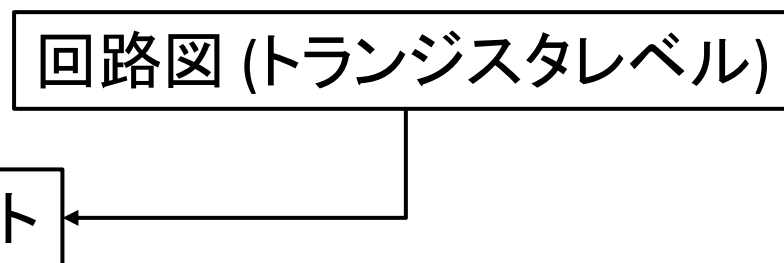
さすがに今は自動設計

デジタル回路

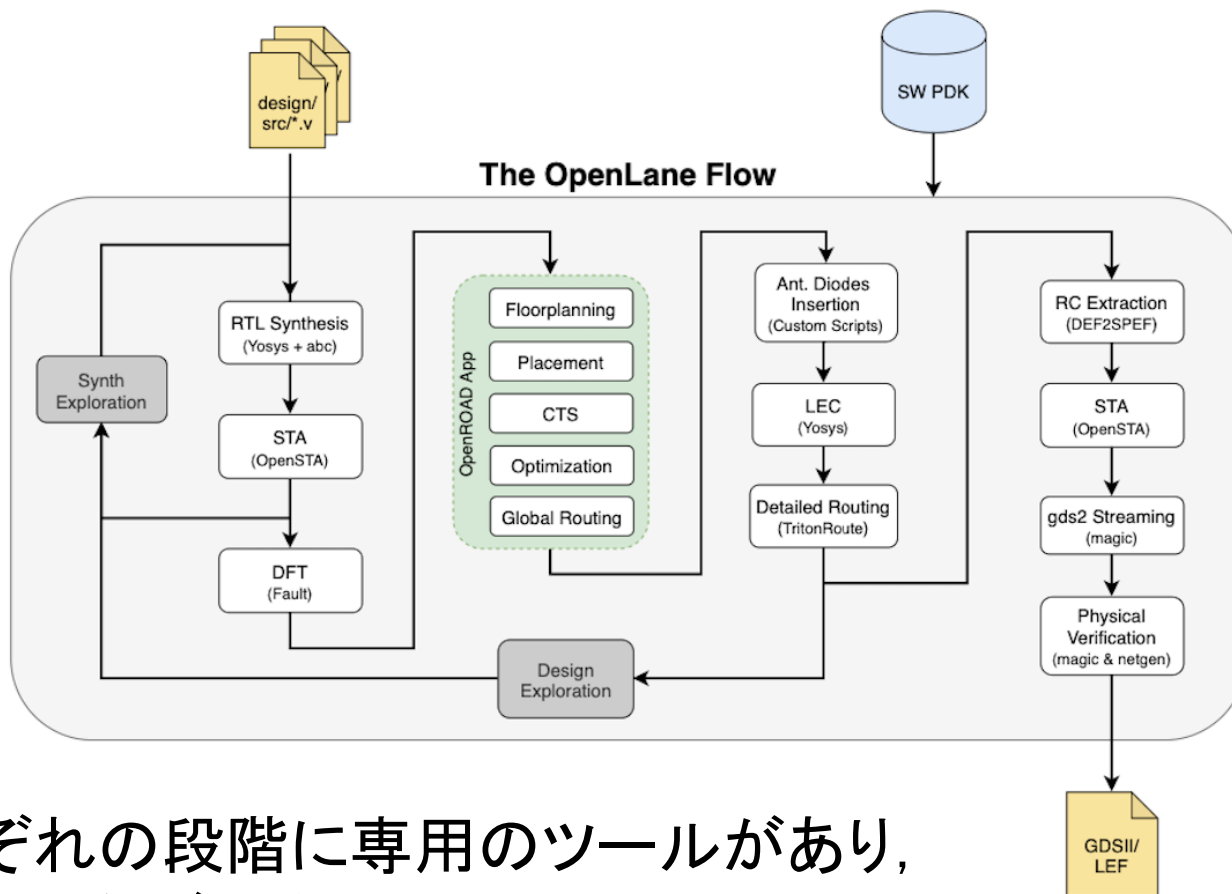


Intel 4004 (Photo from Intel)
2,300 Tr

アナログ回路



設計ツール



それぞれの段階に専用のツールがあり、
それをつなげて作る(ツールチェーン)

セッアップが大変めんどくさい

設計者はレイアウトを描くか？

アナログ設計者: 描く (フルカスタム設計)

Pcell など, ある程度の便利ツールは使う

デジタル設計者: 描かない

設計済みの部品 (スタンダードセル) を
並べて (配置 Placement) 配線をつなぐ (配線 Routing)
特性 (遅延, 電力) はすべて事前に抽出済みのため
人がいじると破綻する

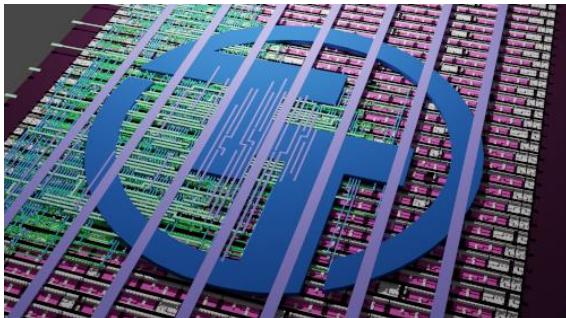
PDKによってはスタセルの中身が秘密で
見たくても見ることもできないことも

今日のハンズオンの目的は
設計ではなく動作原理の理解
もっとも原始的な設計をやってみる

Siliwiz とは？

ブラウザ上でレイアウトとシミュレーションができる

- ✓ レイアウトを描くだけで DRCもシミュレーションも即時反映
- ✓ 教育用ツールであって実際の設計用ではない
その分設計ルールが超シンプル



<https://tinytapeout.com/>



<https://app.siliwiz.com/>



<https://youtu.be/V9xCa4RNfCM>

とりあえず使ってみる

<https://app.siliwiz.com/>

The screenshot displays the SiliWiz web application interface. On the left, a 'Layers' panel lists various materials and their active/passive status. The central area shows a cross-section of a device with layers labeled 'vdd', 'in', 'out', and 'vss'. On the right, a 'SIMULATION' panel shows a graph of 'in' (blue) and 'out' (orange) signals over time. Below the graph, input parameters are set: Input voltage (Min: 0V, Max: 5V), Pulse delay (0μs), Rise time (50μs), and Time scale (60μs). A 'Show SPICE (advanced)' toggle is also present.

Layers

- active
 - ☒ p substrate
 - ☒ n well
 - ☒ n diffusion
 - ☒ p diffusion
 - ☒ p tap
 - ☒ n tap
- passive
 - ☒ polysilicon
 - ☒ polyres
 - ☒ metal1
 - ☒ mim capacitor
 - ☒ metal2
- via
 - ☒ metal1 via
 - ☒ metal2 via

CROSS SECTION & DRC **SIMULATION**

Plot signals: in out +

Input voltage:
Min: 0V
Max: 5V
Pulse delay: 0μs
Rise time: 50μs
Time scale: 60μs

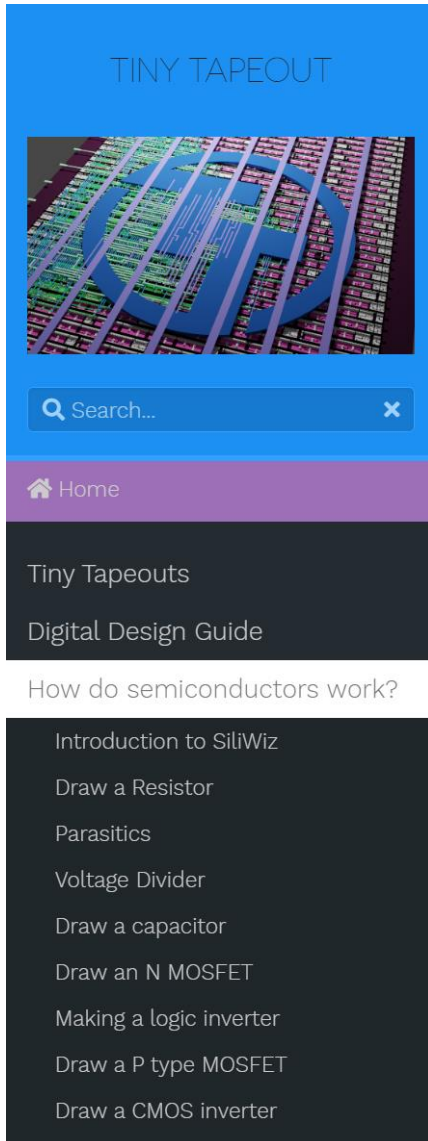
☐ Show SPICE (advanced)

Resources

- Get your digital designs manufactured in silicon for an affordable price at [Tiny Tapeout](#)
- Learn open source ASIC design with the [Zero to ASIC course](#)

SiliWiz revision [p98ec7f](#), built at 2023-04-12T18:32:37.314Z.

参考情報



<https://tinytapeout.com/>

→ How do semiconductor work?
に解説・サンプル

開発情報は github

<https://github.com/wokwi/siliwiz>

画面構成 (Simulation)

右ペインの
切り替えボタン

レイヤー操作

SILIWIZ

Layers

active

- ☒ p substrate
- ☒ n well
- ☒ n diffusion
- ☒ p diffusion
- ☒ p tap
- ☒ n tap

passive

- ☒ polysilicon
- ☒ polyres
- ☒ metal1
- ☒ mim capacitor
- ☒ metal2

via

- ☒ metal1 via
- ☒ metal2 via

Preset

LOAD SAVE CLEAR STL

vdd

in

out

vss

0μm 4μm 8μm

CROSS SECTION & DRC ✓

SIMULATION

Plot signals:

in out +

Input voltage:

Min: 0V

Max: 5V

Pulse delay: 0μs

Rise time: 50μs

Time scale: 60μs

Show SPICE (advanced)

Resources

- Get your digital designs manufactured in silicon for an affordable price at [Tiny Tapeout](#)
- Learn open source ASIC design with the [Zero to ASIC course](#)

SiliWiz revision [e98ec7f](#) built at 2023-04-12T18:32:37.314Z.

描画

シミュレーション

資料へのリンク

画面構成 (Cross section & DRC)

The screenshot displays the SiliWiz web interface for cross-section and DRC analysis. The interface is divided into several sections:

- Layers:** A list of layers categorized into active, passive, and via. Active layers include p substrate, n well, n diffusion, p diffusion, p tap, and n tap. Passive layers include polysilicon, polyres, metal1, mim capacitor, and metal2. Via layers include metal1 via and metal2 via.
- Top Bar:** Contains the SiliWiz logo, a Preset dropdown, and buttons for LOAD, SAVE, CLEAR, and STL.
- Main View:** A 2D cross-section diagram of a device structure. It shows a central red polysilicon gate, blue metal1 and metal2 layers, and various diffusion regions. Labels include vdd, in, out, and vss. A scale bar at the bottom indicates 0um, 4.5um, and 9um.
- Right Panel:** Contains two tabs: CROSS SECTION & DRC (selected) and SIMULATION. The CROSS SECTION & DRC tab shows a 3D cross-section view of the device and a DRC Errors section with a green checkmark indicating "DRC OK".
- Resources:** A section at the bottom providing links to "Tiny Tapeout" and "Zero to ASIC course".
- Footer:** A revision number "e98ec7f" and a build timestamp "built at 2023-04-12T18:32:37.314Z".

断面図・DRC結果

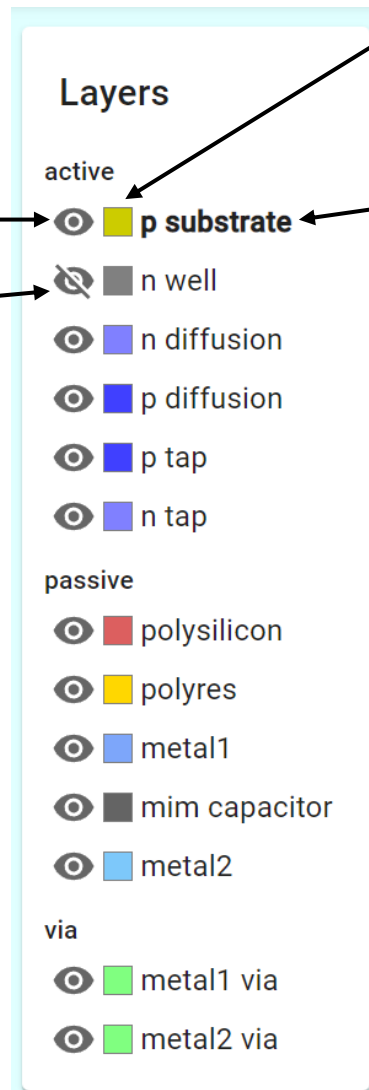
レイヤー操作

目のアイコン

→ 表示・非表示切替

表示

非表示



描画エリアでの色 (変更不可)

レイヤー名をクリックすると
選択状態

→ 描画エリアにそのレイヤーの
オブジェクトを描ける

描画エリアの操作

Undo/Redo Load/Save JSON file

画面クリア

STLファイル出力

断面表示位置

描画エリア

※ JSON:
JavaScript Object Notation
いろんなところで使われる
データ記述軽量フォーマット

※ STL:
Standard Triangulated Language
3次元形状データでよく使われる
ファイルフォーマット

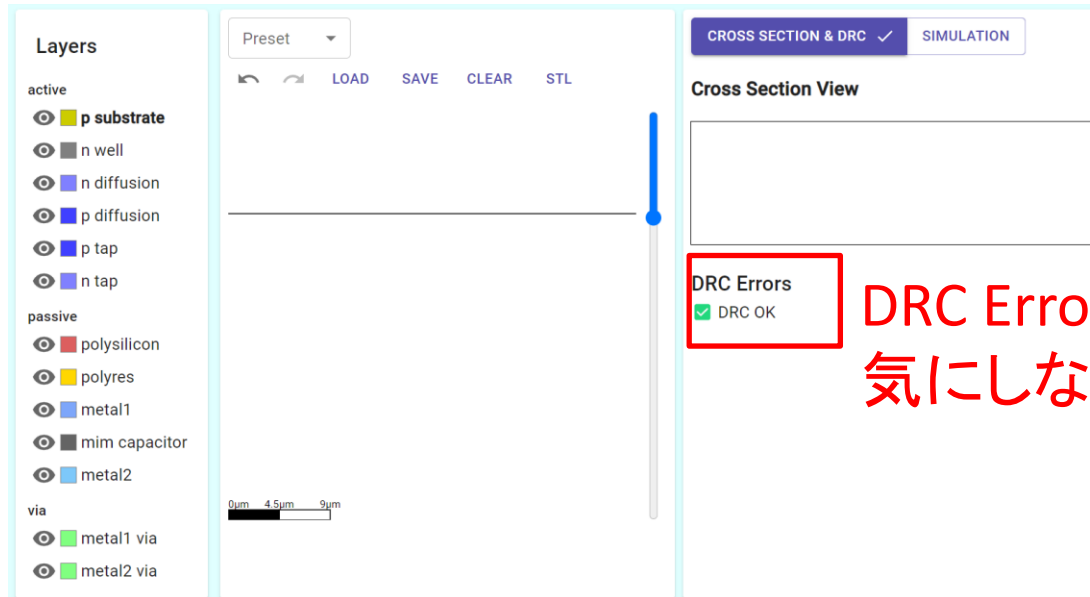
プリセット選択

The image shows a software interface for circuit simulation. On the left, a 'Preset' dropdown menu is open, showing a list of JSON files: (none), blank.json, inverter.json, m2cap.json, mimcap.json, nmos.json, nmosfet.json, pmos.json, pmosfet.json, resistor.json, and skullfet.json. The main area displays a schematic diagram of a circuit with components labeled vdd, in, out, and vss. A red vertical bar is positioned in the center of the diagram, and a blue vertical bar is on the right side. A scale bar at the bottom left indicates 0um, 4.5um, and 9um. Arrows point from the text labels to the corresponding UI elements: 'プリセット選択' points to the preset dropdown; 'Undo/Redo' points to the undo and redo buttons; 'Load/Save JSON file' points to the 'LOAD' and 'SAVE' buttons; '画面クリア' points to the 'CLEAR' button; 'STLファイル出力' points to the 'STL' button; '断面表示位置' points to the blue vertical bar; and '描画エリア' points to the schematic diagram.

Work 1: 何でもいいので描いてみる

準備: 右ペインは「CROSS SECTION & DRC」に
描画エリアを「CLEAR」でクリア

この状態に

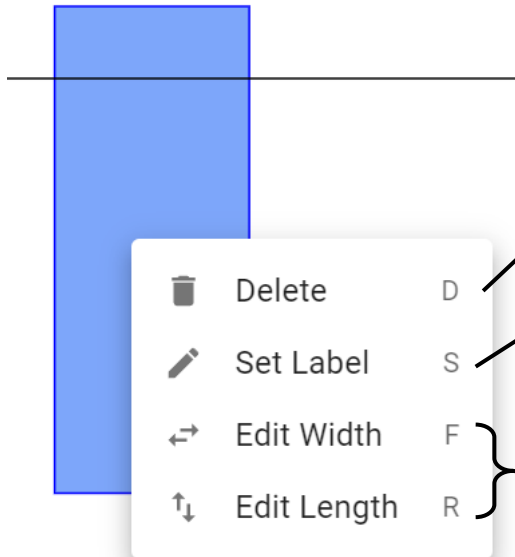


DRC Errors は今は
気にしなくてOK

- やってみよう:
- ・レイヤーを選んで描画エリアに矩形を描く
 - ・レイヤーを変えて描く
 - ・断面図が変わることを確認
 - ・レイヤーの表示・非表示を変えてみる

Work 2: 描いた図形の編集

描いた図形をクリックするとメニューが出る



図形を消す

ラベルをつける (metal1, metal2 のみ)
ラベルはシミュレーションに使う

幅・長さの直接指定

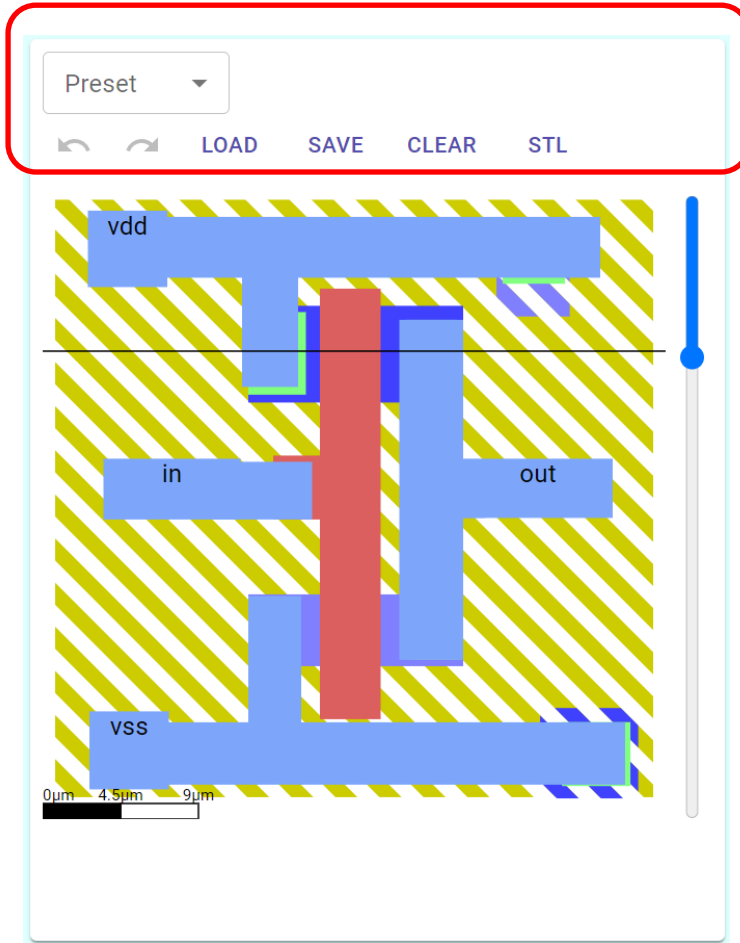
app.siliwiz.com の内容

Enter new width in μm

OK キャンセル

図形の移動・コピー・ペーストなどはできない...

Work 3: Save/Load



- 自分の描いたレイアウトを SAVE
- 一旦 CLEAR して, SAVE したものを LOAD してみる







JSONファイルはテキストデータなので
中身はテキストエディタで読めます

レイヤーの意味と デバイスの構造






レイヤーの意味

Layers



active

- ☒  p substrate
- ☒  n well
- ☒  n diffusion
- ☒  p diffusion
- ☒  p tap
- ☒  n tap

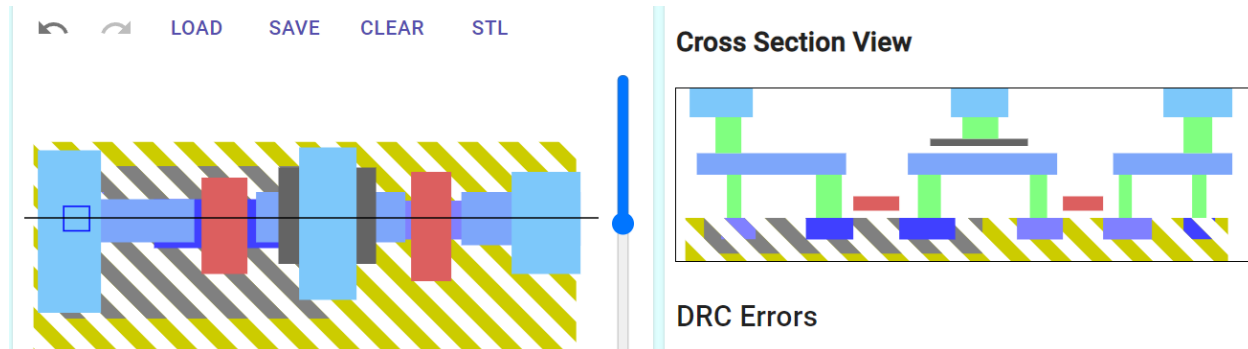
passive

- ☒  polysilicon
- ☒  polyres
- ☒  metal1
- ☒  mim capacitor
- ☒  metal2

via

- ☒  metal1 via
- ☒  metal2 via

たくさんある上に決まった組み合わせで
使わないといけないらしい...




細かいことは置いておいて
イメージで各レイヤーの役割を理解しよう


p と n


Layers

active

 p substrate

 n well

 n diffusion

 p diffusion

 p tap

 n tap

頭に p と n がついているものは半導体

p : p型半導体

正の電荷 (正孔, ホール) がたくさんある

n : n型半導体

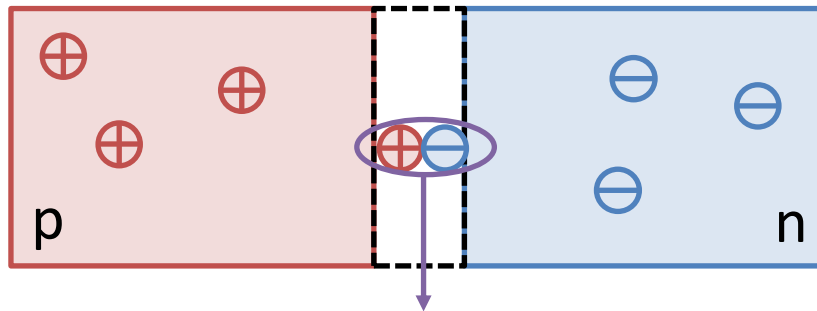
負の電荷 (電子) がたくさんある



電荷がたくさんあるのでそれなりに電気を通す

貼り合わせると

p型半導体とn型半導体を貼り合わせると
接合面に電気を通さない膜 (空乏層) ができる

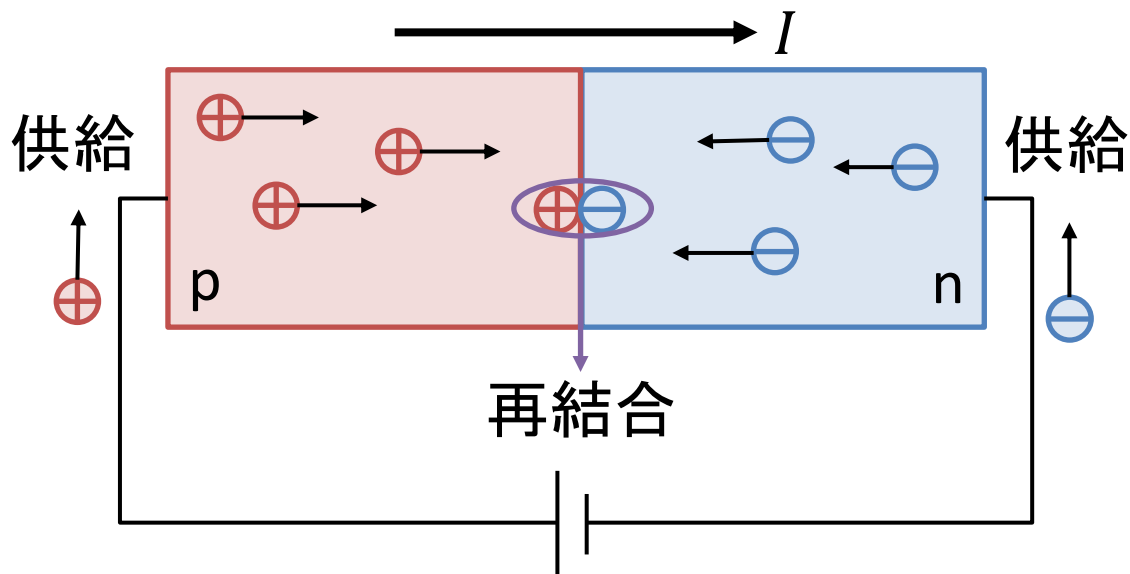


正孔と電子は引き合って再結合により消える

正孔も電子も消えて空っぽになった領域が
空乏層 (depletion layer)

順方向電圧

p と n に電圧を印加すると

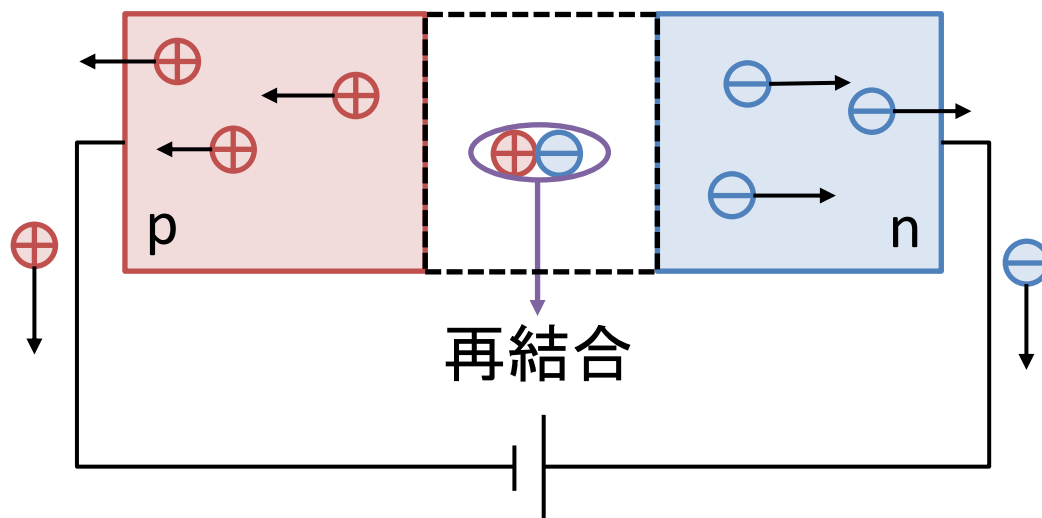


正の電荷 (正孔) は電位の高い方から低い方へ
負の電荷 (電子) は電位の低い方から高い方へ
再結合で消えても正孔も電子も電源から無限に供給

p型からn型に向かって電流が流れる

逆方向電圧

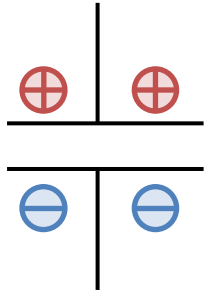
電圧を逆にすると空乏層が広がるだけで電流は流れない



pn接合:

p型からn型へは電流が流れるが, n型からp型には流れない
整流特性 → 半導体ダイオード

MOS



キャパシタ (コンデンサ)

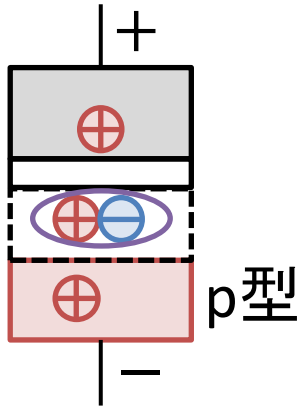
電圧をかけると極板の一方に正, もう一方に負の電荷が集まる $Q = CV$

極板の片方を半導体にする

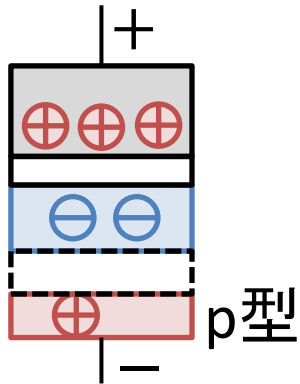


Metal-Oxide-Semiconductor = MOS

反転



金属が正，半導体が負の電圧を印加
→ 半導体側に負の電荷が集まる
p型がもともと持っている正孔と再結合，
空乏層ができる



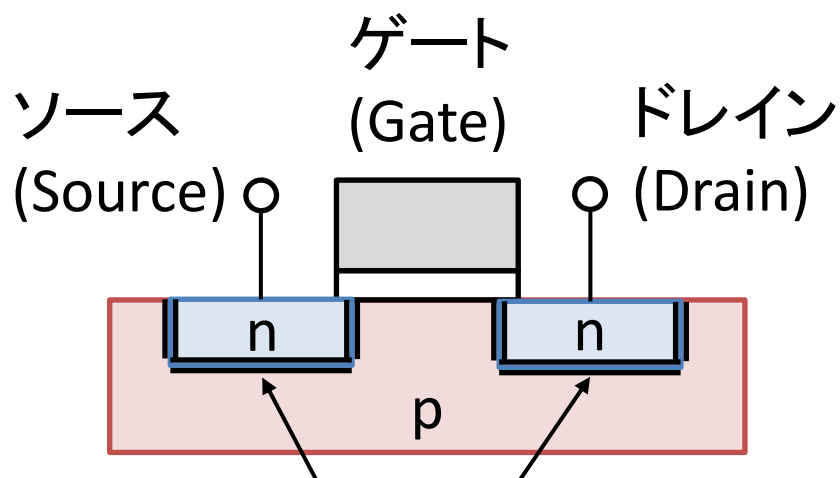
電圧を高くする
→ 半導体側にもっと負の電荷が集まる
p型がもともと持っている正孔を食い尽くして
電子が余る → n型半導体になる

電圧 (電界) によって半導体を制御する
電界効果トランジスタ (Field-Effect Transistor: FET)

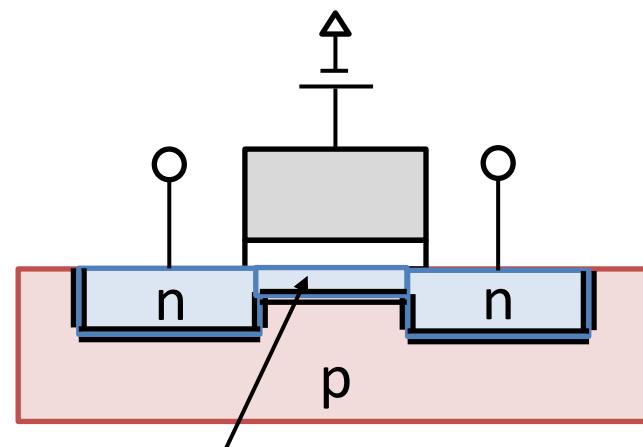
MOS + FET = MOSFET

MOSFET

反転したら電極間がつながるようにしよう



空乏層で囲まれるので
電流は流れない

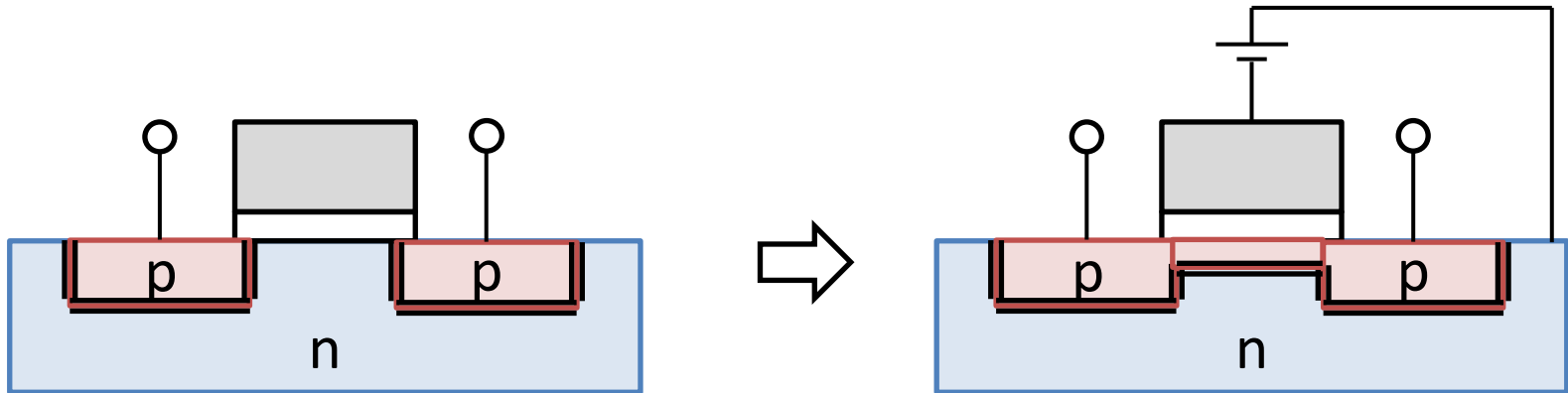


反転して n型になると
両側の電極間が全部n型 → 導通

反転して電極間をつなげる部分 = チャンネル (Channel)
電極間が n型でつながるMOSFET = nMOSFET

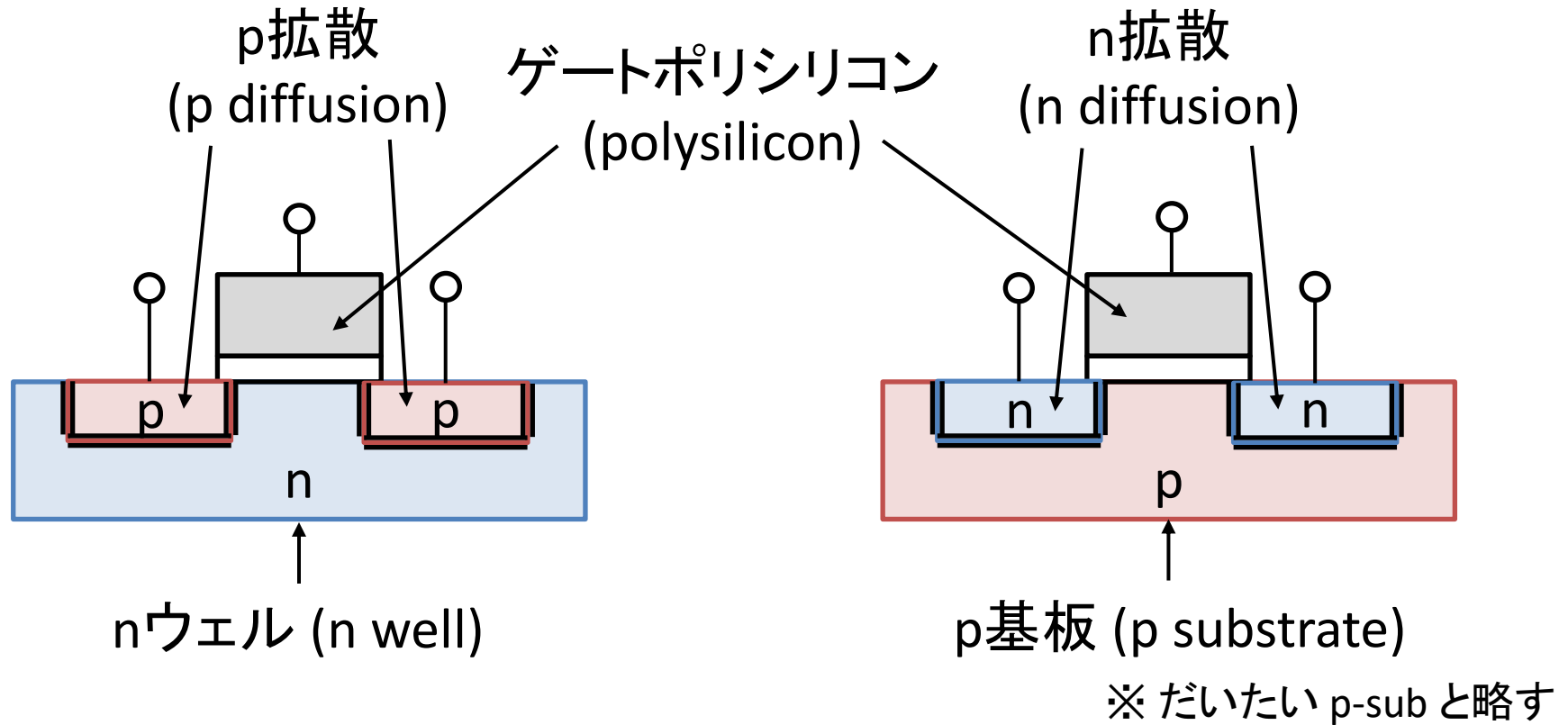
pMOS

p型とn型を入れ替えても動く



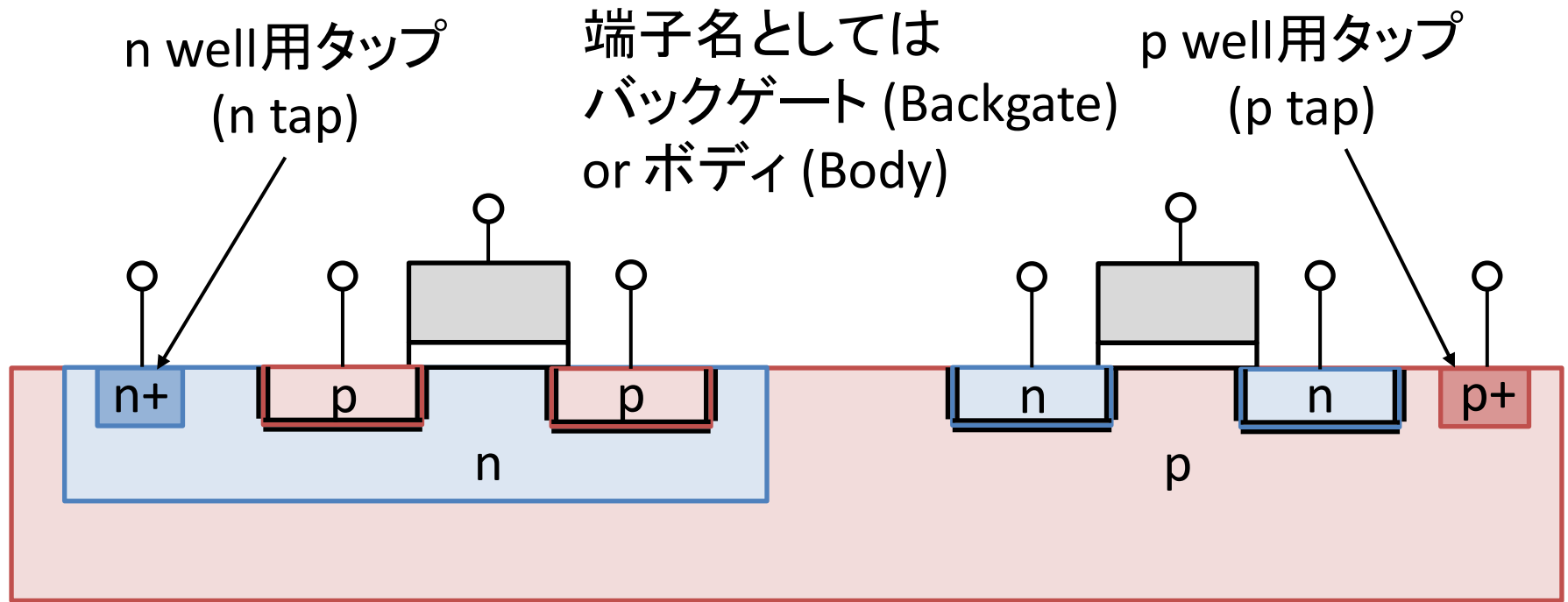
p-p-p でつながるので pMOSFET

MOSFETの構造



これだと n-well と p-sub の電位が決まらないので...

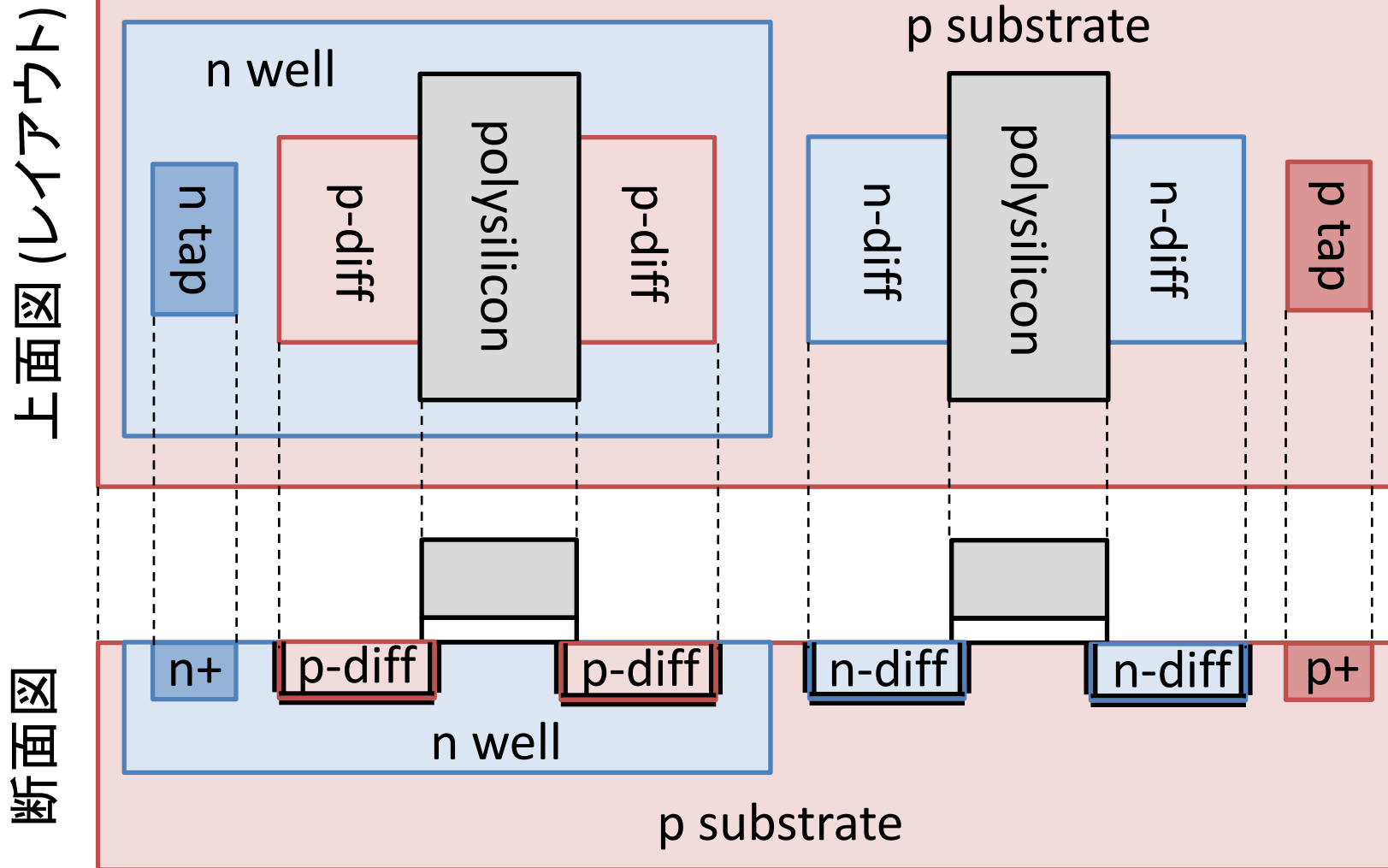
substrate と well にも端子を



p+, n+ は不純物密度が高い (正孔・電子が多い = 導電率高い) を示す

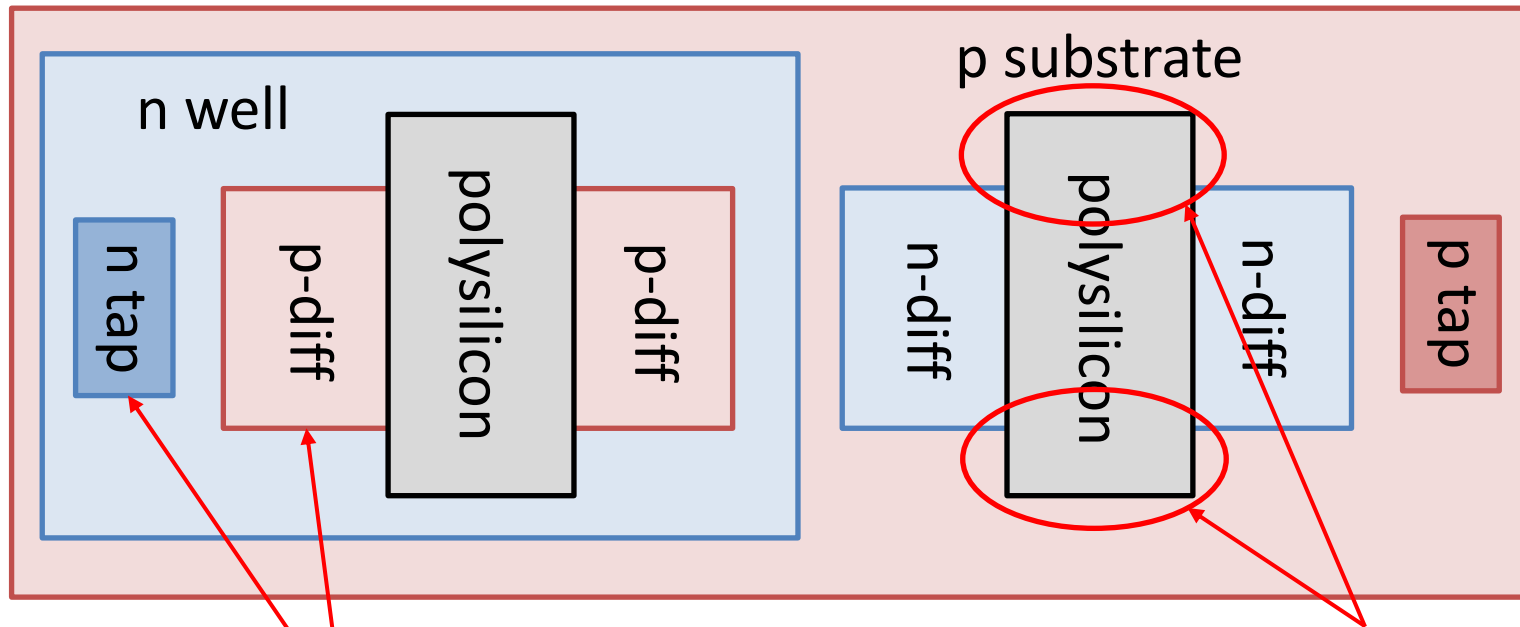
まずウェハ全体を p型にしてそこに n型の領域を作るので
p substrate と n well の呼び分けがある

MOSFETを作るレイヤー



ここまでが FEOL (Front-End-Of-Line)

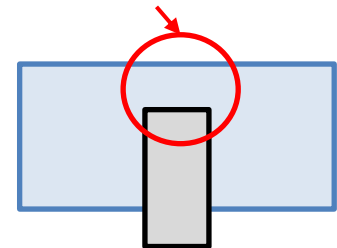
レイアウトのポイント



diff は sub/well と導通したら
困るので p/n 逆を配置
tap は sub/well と導通しないと
困るので p/n 同種を配置

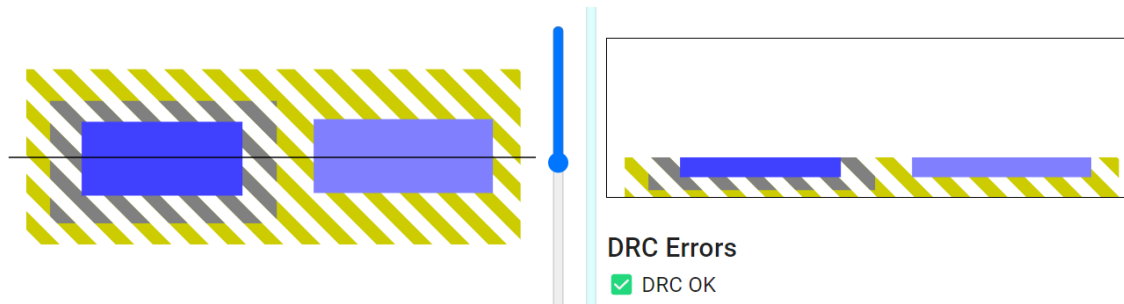
polysilicon は
diffusion を完全に横切る

ここがつながってしまう

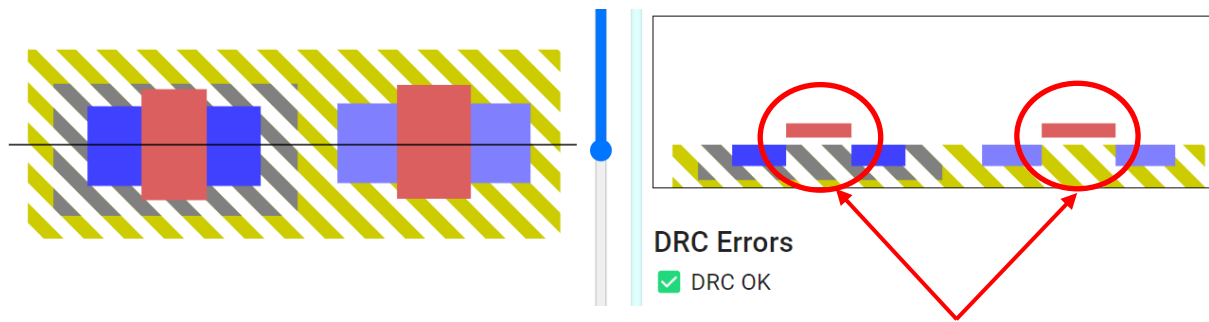


Work 4: MOSFETを描く

diffusion の上に polysilicon を重ねると, 自動的に diffusion が切れる
(自己整合プロセス, self-aligned process)



p substrate, n well, p diffusion, n diffusion を描いたところ
この上に polysilicon を重ねると...



polysilicon の下の diffusion が自動的に消える
(レイアウトのオブジェクトが変わるわけではない)

配線

ここから BEOL (Back-End-Of-Line)

配線は多層構造

下から1, 2, 3...

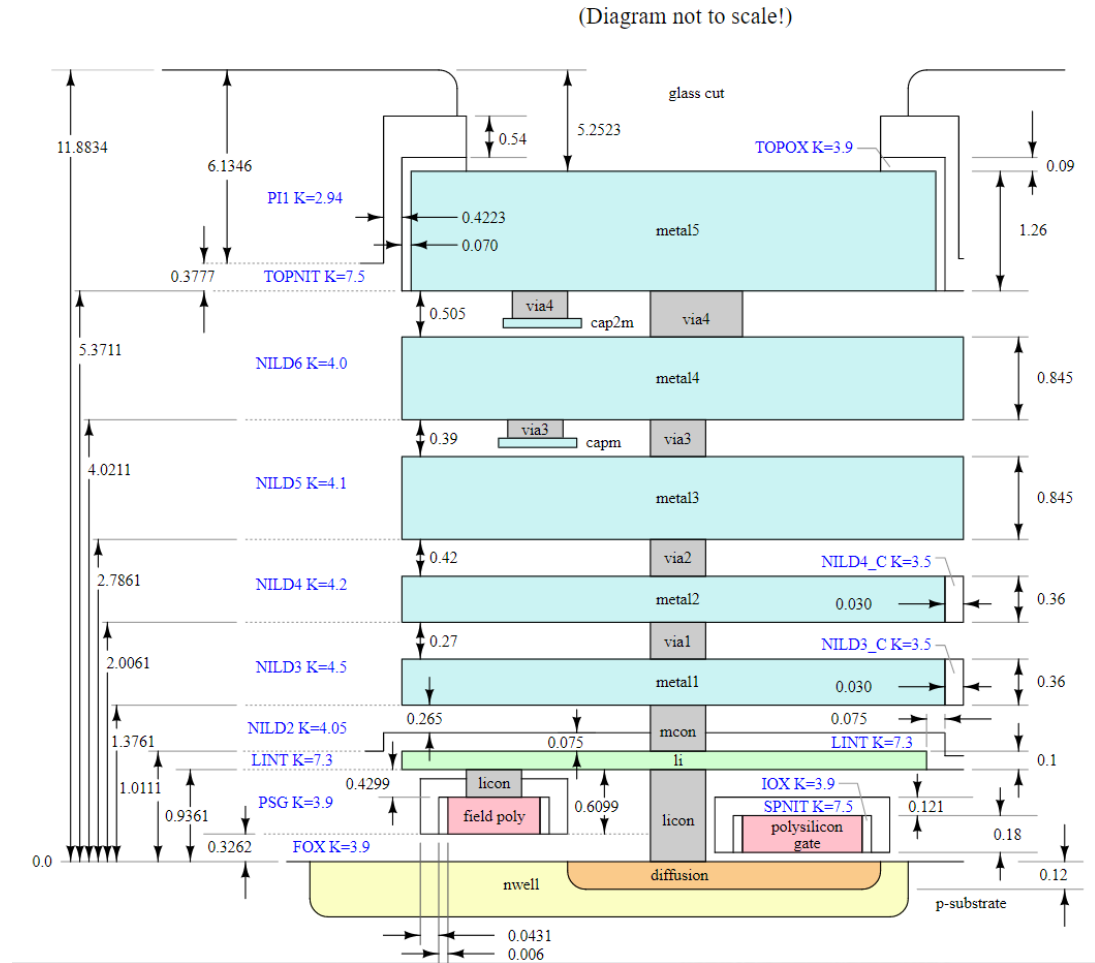
最上位を「トップメタル」
と呼ぶことも

右図では metal5 = top metal

階層同士をつなぐのが
ビア (via)

FEOL とつなぐものを特に
コンタクト (contact) と
呼ぶことが多い

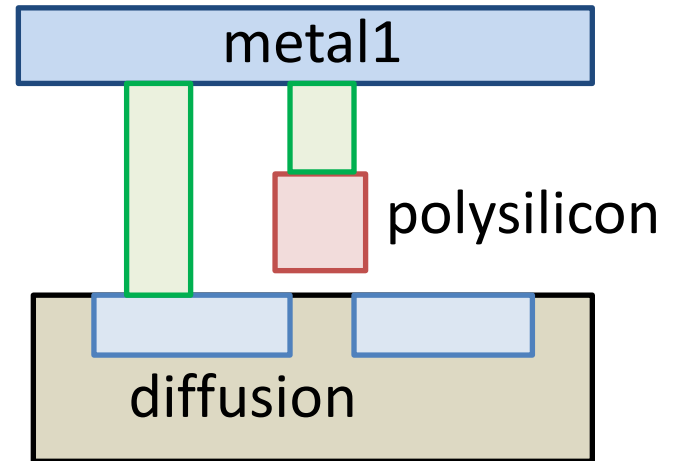
厚さは一切変更できない



配線について

コンタクト (Siliwiz では metal1 via) は
ポリシリコンにも基板 (diffusion, tap)
にもつながる

どうでもいいけどレイアウトで
メタルを描くのを「引く」
コンタクト・ビアは「打つ」と言います
(少なくとも講演者のまわりでは)



ビアがどの層とどの層をつないでいるのか勘違いしないように

“via1” というレイヤーは... ふつうは metal1 と metal2

“via12” と書いて metal1 – metal2 を明示することも

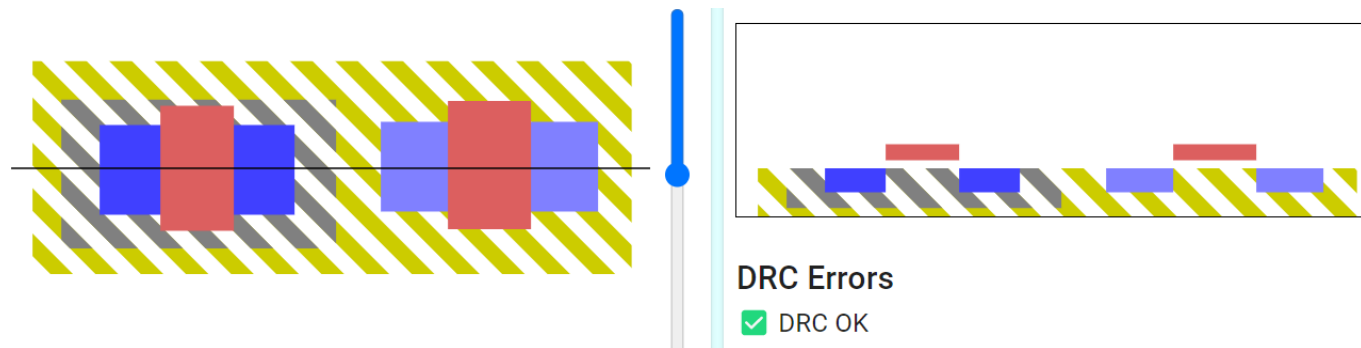
Siliwiz は “metal2 via” が metal1 – metal2 間の接続なので
間違えないように注意

“metal1 via” を “contact” にして “metal2 via” を “via1” にすればいいのに...

Work 5: コンタクトの接続先

下のようなレイアウトを作成し, いろいろな場所にコンタクト (metal1 via) を打って断面構造を確認しよう

※ DRC Errors は気にしない



polysilicon と diffusion の境界にコンタクトを打つと
そのコンタクトはどこにつながるか？

1. polysilicon
2. diffusion
3. 両方

受動素子用レイヤー: polyres

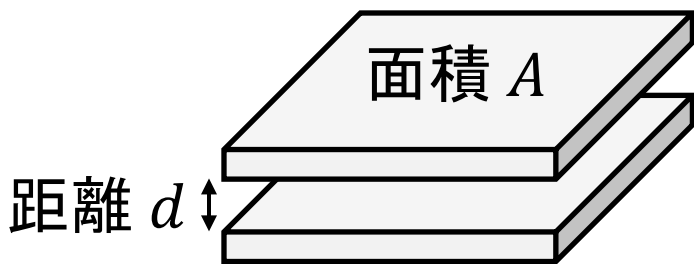
polyres : 抵抗率の高いポリシリコン (高抵抗ポリ)
ポリシリコンだがMOSFETのゲートとしては働かない



ゲートポリ (polysilicon) が抵抗率を下げるために行なう
合金化処理 (シリサイド化, silicide) をスキップする

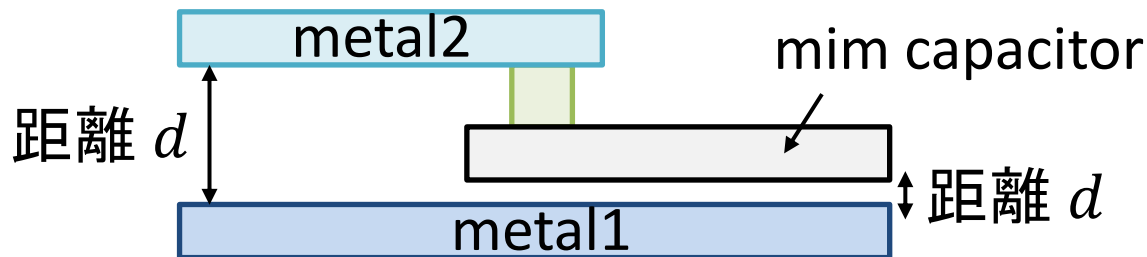
Siliwiz で抵抗を作れるのは polyres のみ.
他のレイヤーはどんなに細くしても抵抗はつかない

受動素子用レイヤー: mim cap



平行平板コンデンサ $C = \varepsilon \frac{A}{d}$

集積回路において面積 = コスト
小面積で容量を稼ぐには d を小さくする



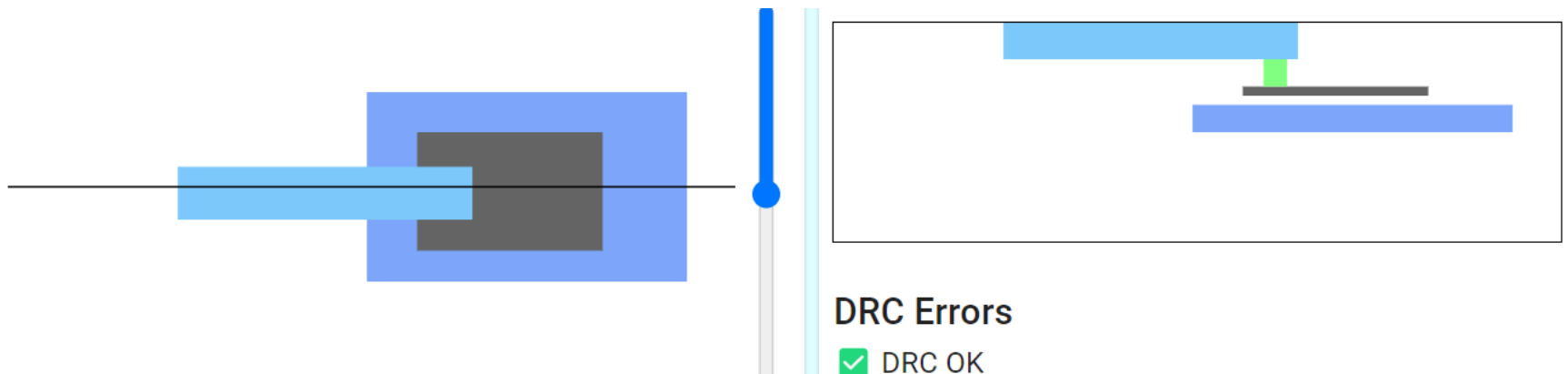
MIM:
Metal-Insulator-Metal
だいたい $1 \sim 2 \text{ fF}/\mu\text{m}^2$

mim capacitor を使わず metal1 と metal2 の間の容量を使うのを
MOM (Metal-Oxide-Metal) capacitor と呼ぶ
MIM cap はデザインルールがキツイのでMOMもよく使う

Siliwiz では MOM capacitor も構成可能

Work 6: MIM cap を描く

mim capacitor レイヤーを使って MIM cap を描く
mim capacitor は metal2 via で metal2 とつなぐ



参考: その他の受動素子

抵抗

配線抵抗: 抵抗率が低すぎるのであまり使わない

well抵抗: 寄生容量が巨大, 基板ノイズを拾いやすい

MOS抵抗: MOSFETの線形領域を使う. 可変抵抗が作れる

容量

MOS容量: ゲート容量はIC内で最も単位面積当たりの容量が大きい
電圧依存性がある. ノイズを拾いやすい

MOS Varactor: MOSで作った可変容量

インダクタンス

スパイラルインダクタ: 配線でコイルを作る. ものすごく巨大だが
IC内でLを作るには事実上これしかない

デザインルールチェック

デザインルールとは

製造工場が定めたレイアウトのルール

これを満たさないで作ってもらえない or 品質が保証されない

デザインルールをすべて満たしているかを確認するのが
DRC (Design Rule Check)

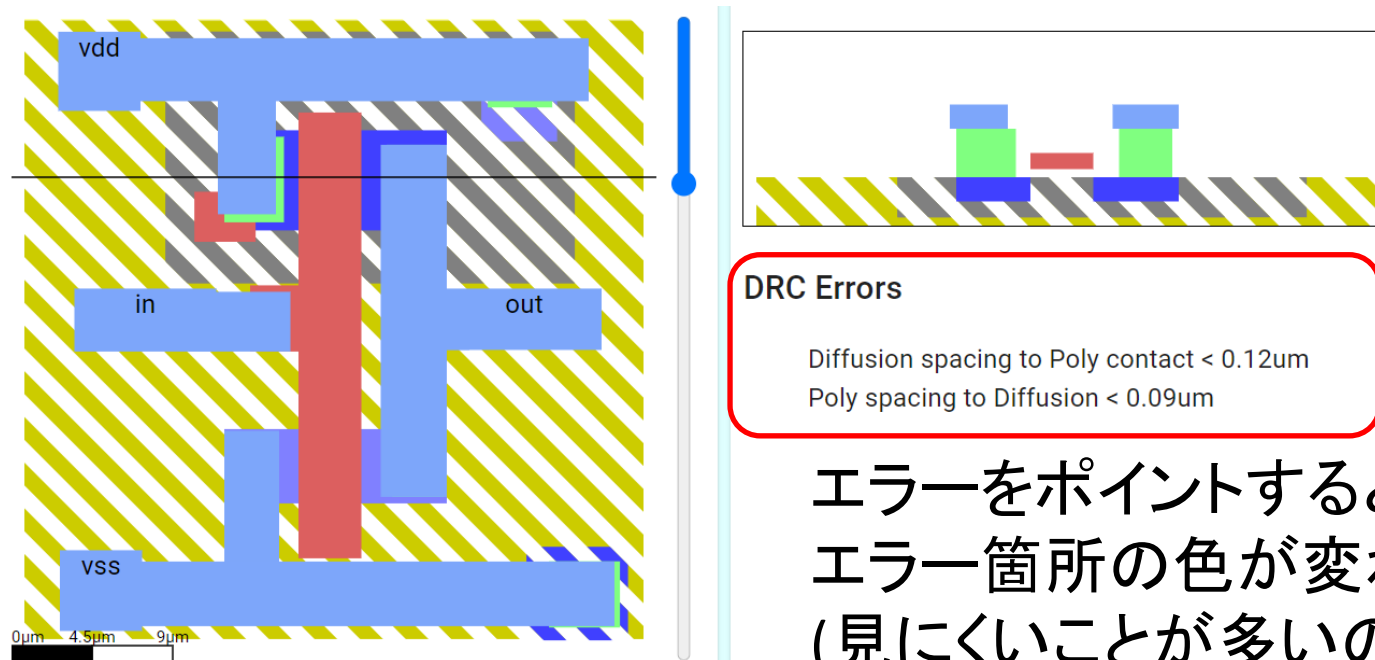
基本的には製造装置の加工限界に起因

ex.) 最小幅, 最小間隔, 最小面積, パターン密度 etc...
ばらつき防止の推奨ルール (DFM) などがある場合も

Siliwiz はデザインルールが非常に緩い

Work 7: DRC Errorチャレンジ

DRCエラーを何個 (何種類) 出すことができるかに挑戦



エラーをポイントすると
エラー箇所の色が変わる
(見にくいことが多いので
適宜レイヤーを非表示にする)

20種類以上あります

Siliwiz の DRC Errors はちょっと不安定なことがあるので注意

シミュレーション

Siliwiz でできるシミュレーション

準備: Preset から inverter.json をロード, 右ペインを Simulation に



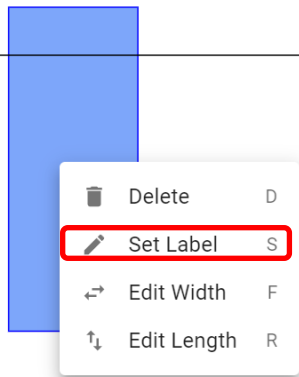
入力 (in) の電圧を
時間的に変化させて
反応を見る過渡解析
(Transient analysis)

- ・入力は1つだけ
- ・入力波形は
台形パルスのみ
という超割り切り仕様

※ 通常の設計では直流解析 (DC解析), 周波数応答 (AC解析)などを駆使します

ラベル

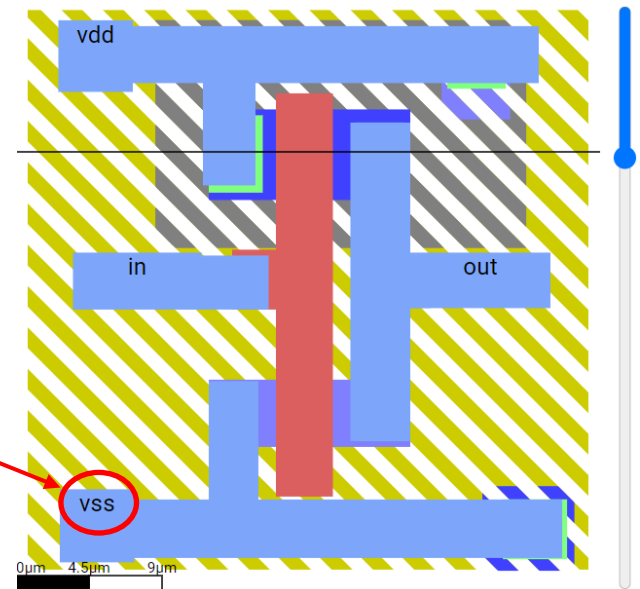
レイアウト内のオブジェクトに名前をつけて
シミュレーションで指定できるようにする



オブジェクトをクリック
→ Set Label

⇒ つけた名前が表示される

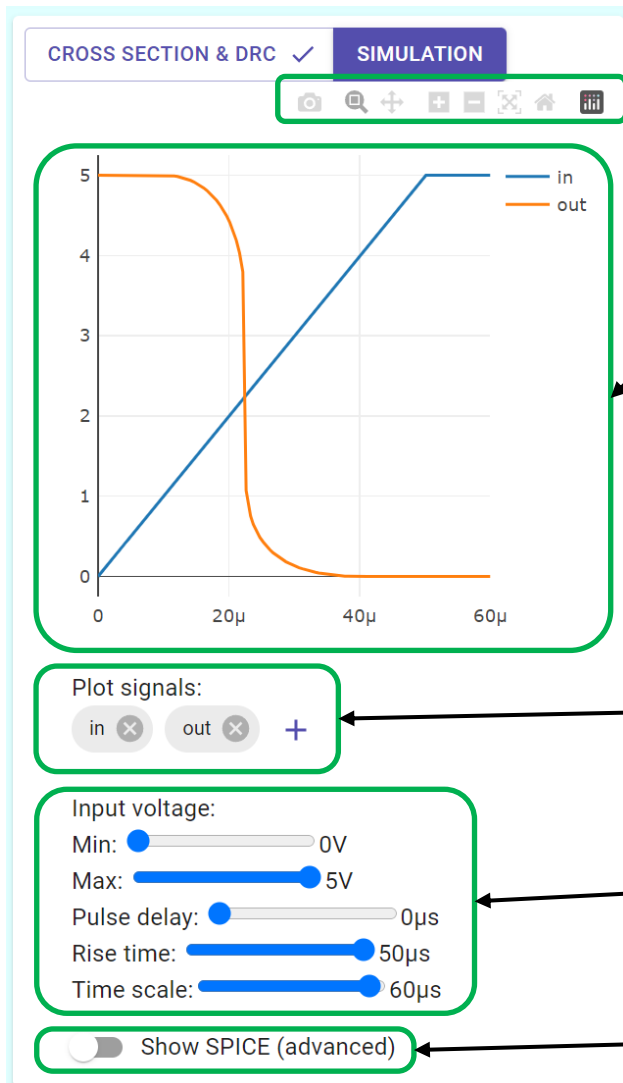
※ ラベルを設定できるのは
metal1, metal2 のみ



名前は好きな名前でもOK

ただし in : 入力, vdd : 電源 (5 V), vss : グラウンド (0V) は予約語

シミュレーション画面



表示系の操作ボタン

(グラフ領域にマウスポインタをもっていくと出てくる)

表示エリア

マウスポインタを線上におくと値が表示される

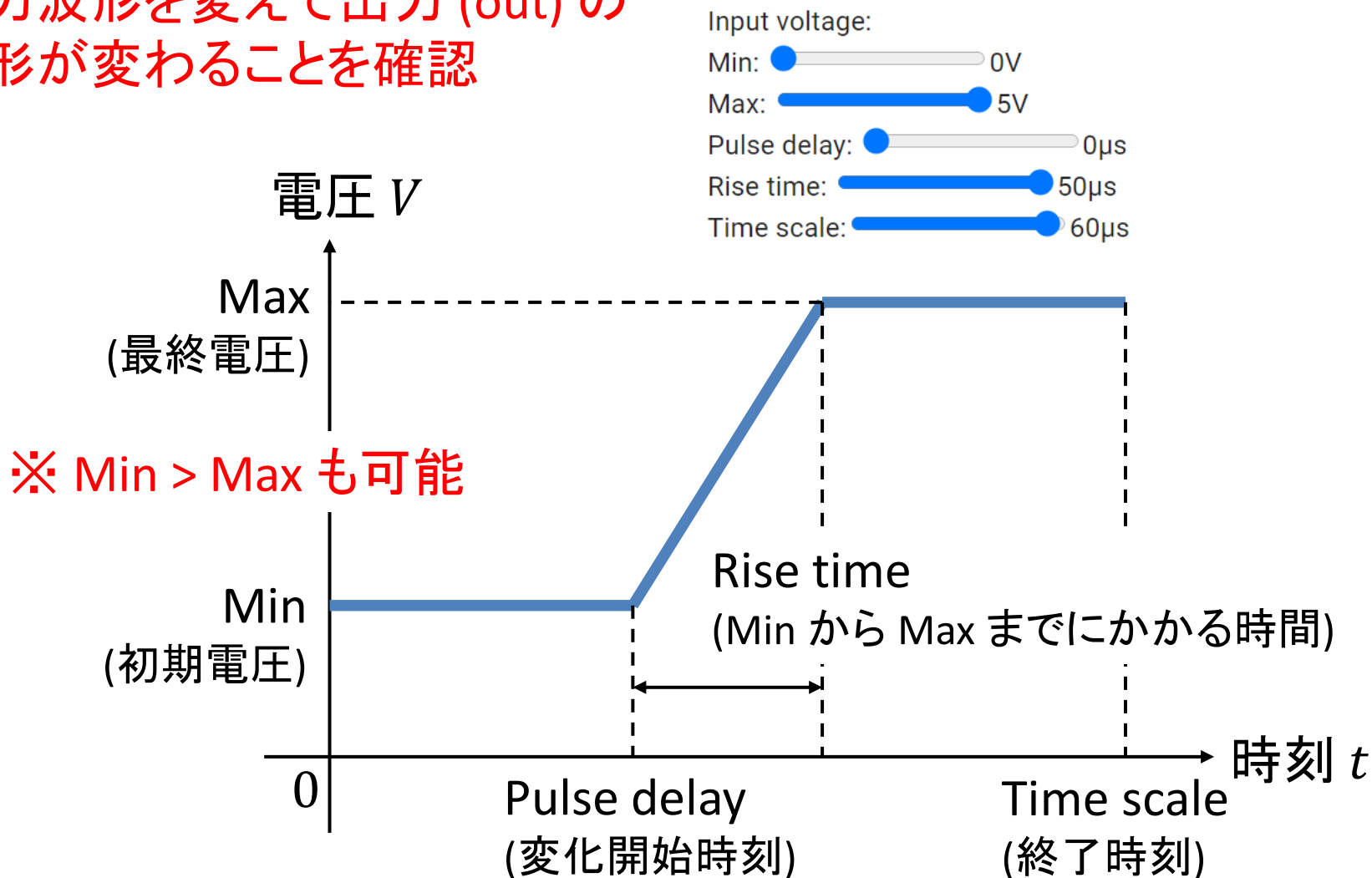
プロットする信号の選択

入力波形 (in) の設定

SPICEネットリストの表示

Work 8: 入力波形の設定

入力波形を変えて出力 (out) の
波形が変わることを確認



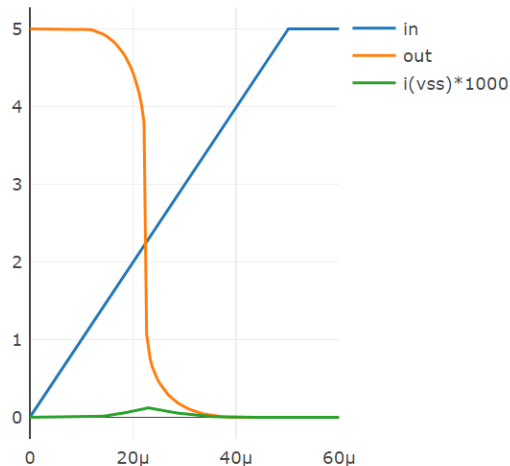
Work 9: 波形の追加と加工

1. out を消す (×ボタン)
2. レイアウトで out のラベルを Y に変更
3. Y の波形を追加 (+ ボタン)

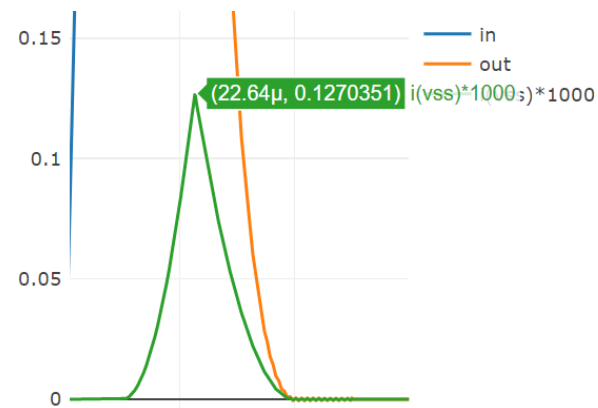
※ 存在しない
ラベルを指定すると
プロットが全部消える
ので注意

波形の追加で「i(ラベル)」とすると電流が得られる

4. 波形の追加で「i(vss)*1000」を追加
5. ズームで拡大表示



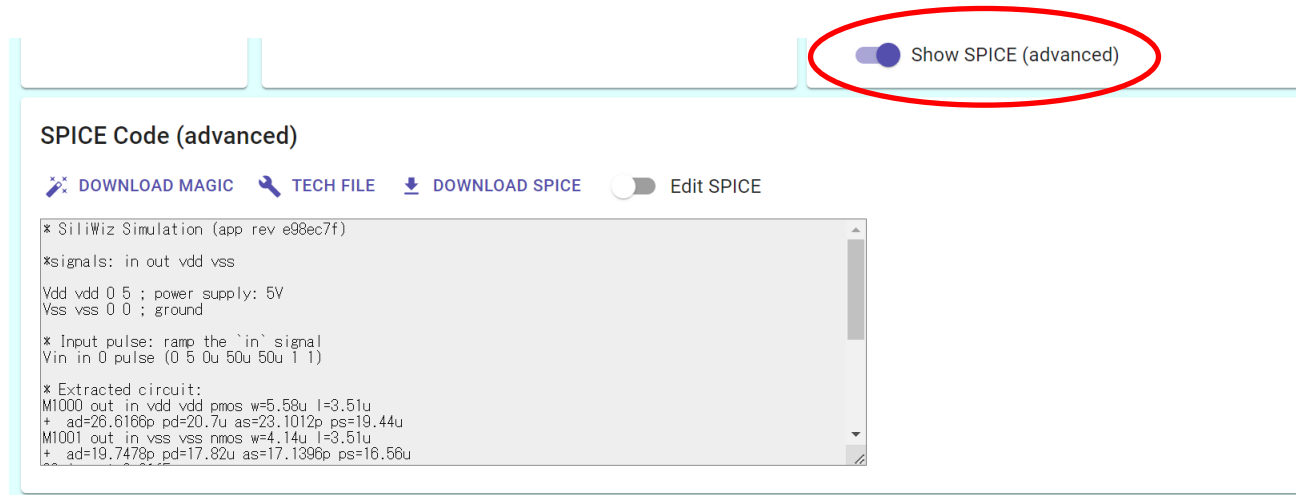
電流プロット追加 (Step 4)



電流プロット拡大 (Step 5)
最大 0.127 mA 流れている

SPICEネットリストの表示

SPICEネットリスト: 回路シミュレータ SPICE の入力ファイル



name node1 node2 ... value という形式で素子を記述

例) 「Vdd vdd 0 5」は

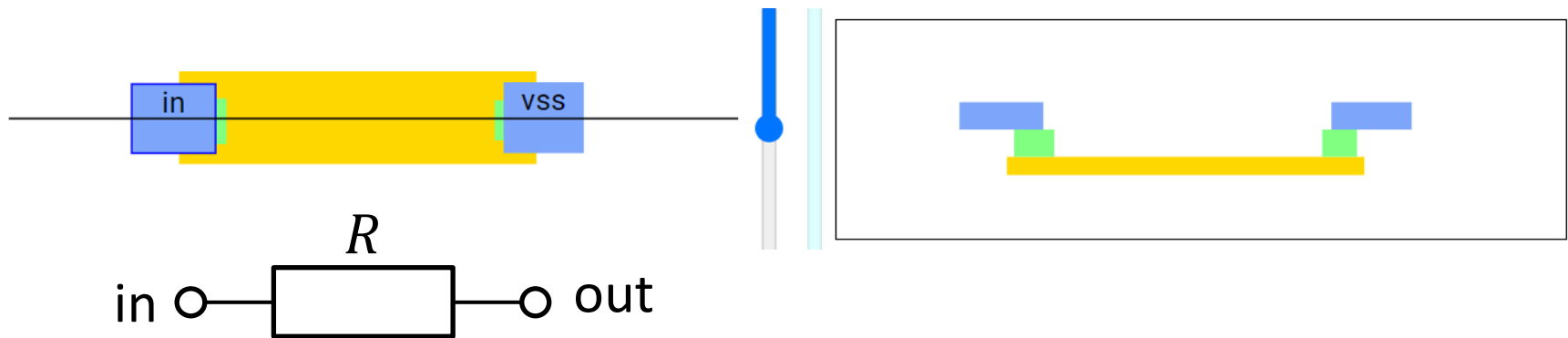
- ・ Vdd という名前 (先頭の V が電圧源を表す)
- ・ ノード vdd と ノード 0 の間に配置
- ・ 電圧は 5 V

抵抗値や容量値, MOSのサイズなどを読み取るのに使う

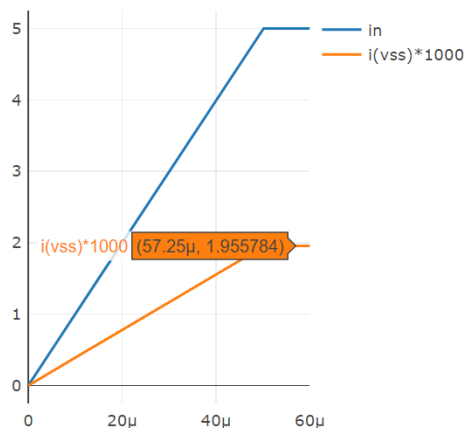
素子の特性を調べる

Work 10: polyres の抵抗値

polyres を適当に作って両側に端子 in, vss をつなぐ



流れる電流を「 $i(vss)*1000$ 」でプロット



$in = 5\text{ V}$ で
1.96 mA

$R = V/I$ より
2.55 k Ω

SPICE ネットリストで確認

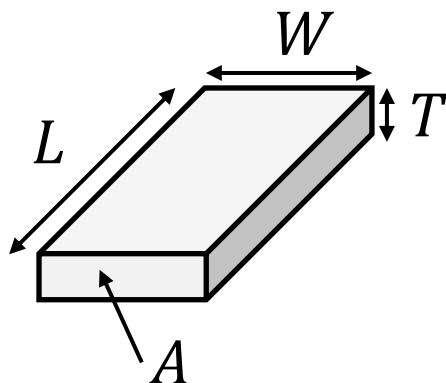
in と vss につながっている
 R を探すと...

R0 in vss 2556.520020

2.56 k Ω

※ 値は描いたレイアウトで変わります

シート抵抗



抵抗値 R は $R = \rho \frac{L}{A}$

抵抗率 $[\Omega \cdot \text{m}] \times$ 長さ $[\text{m}] /$ 断面積 $[\text{m}^2]$

形状が直方体の場合は $R = \rho \frac{L}{W \times T}$

集積回路内では厚さ T は変えられないので $R = \boxed{\frac{\rho}{T}} \frac{L}{W}$

これを覚えておけば L と W の比だけで抵抗が求まる

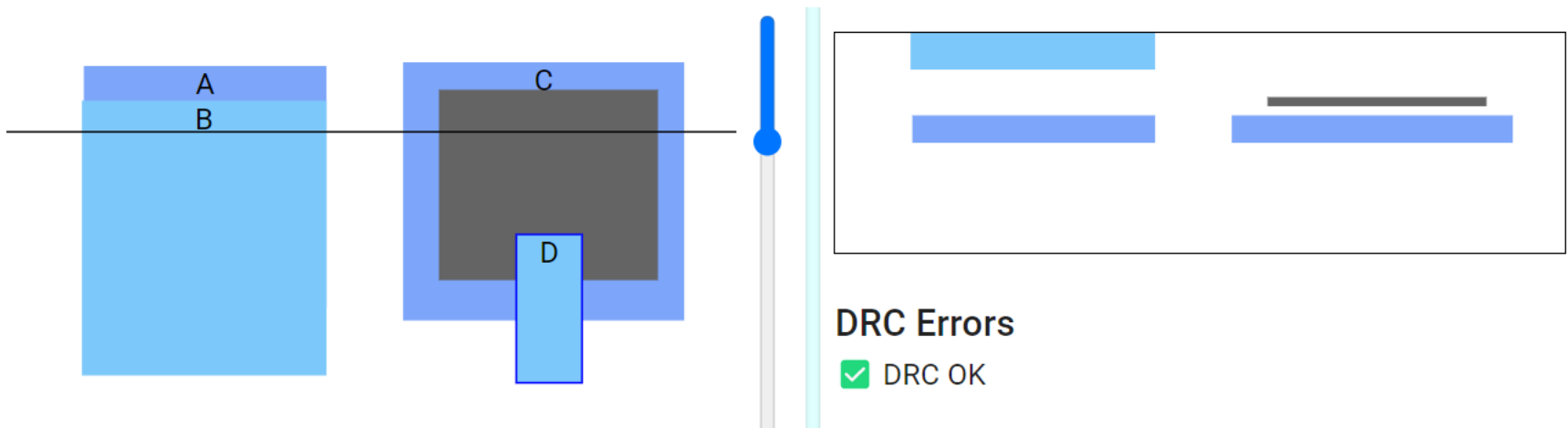
次元は Ω だが普通の抵抗値と混同しないように

単位として $[\Omega \square]$, $[\Omega \text{sq.}]$, $[\Omega / \square]$, $[\Omega / \text{sq.}]$ と書くことが多い

※ sq. = square

Work 11: 容量値

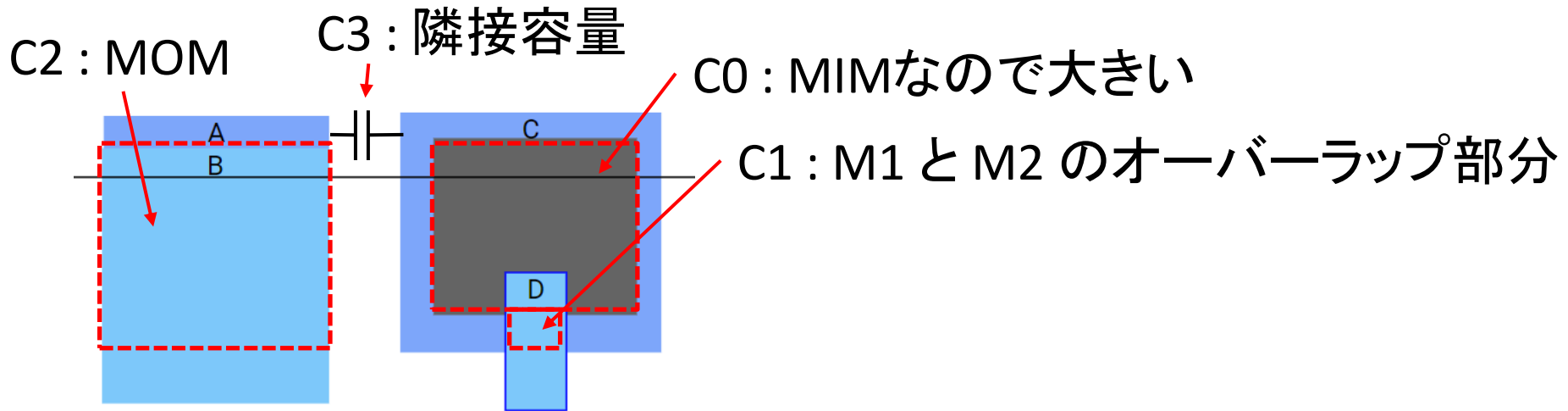
MOM と MIM を作ってみる
metal1 と metal2 にそれぞれラベルをつける



容量値はSPICEネットリストから読み取るのが簡単なので
SPICEネットリストから容量 (先頭が C) を探す

Work 11つづき

すべての配線間に容量は存在するのでたくさんでてくる



C0 D C 404.351990fF

C1 C D 0.01fF

C2 A B 0.04fF

C3 A C 0.07fF

C4 B SUB 0.22fF

C5 D SUB 0.05fF

C6 C SUB 0.16fF

C7 A SUB 0.14fF

一応 MOM は作れるが値が小さすぎる

Siliwiz では事実上 MIM一択

たぶん MOMの容量は正確ではない

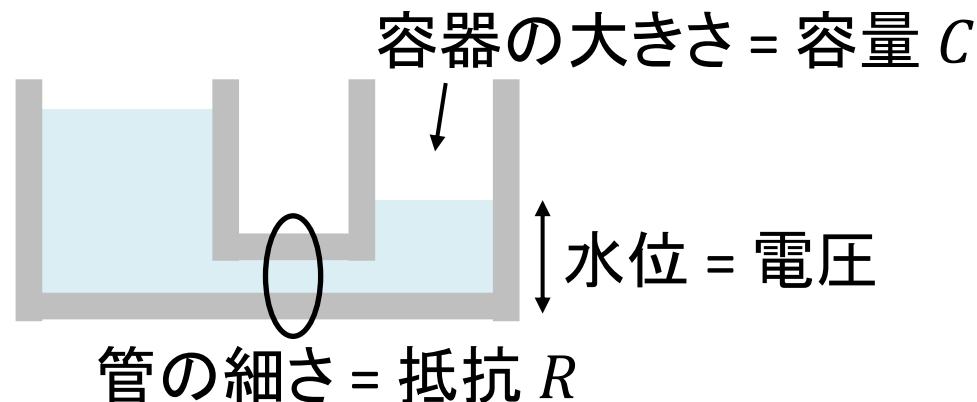
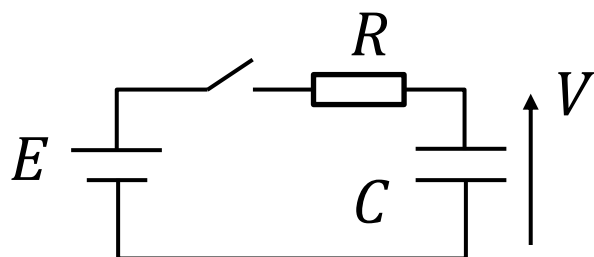
対基板容量

メタルの面積に比例？

BはAに遮蔽されてるから実際はもっと小さいような...

シミュレーションで容量を求める

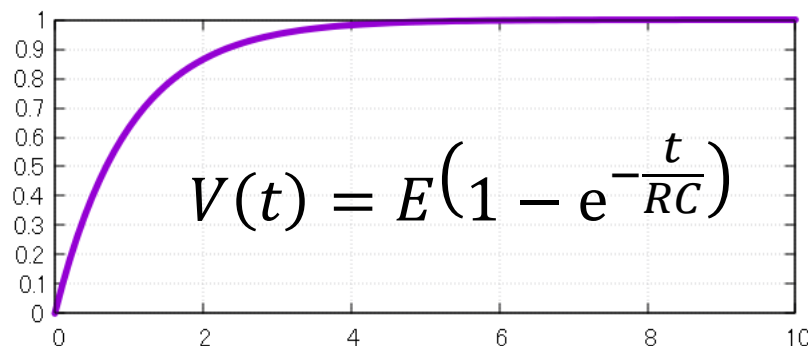
容量値は波形 (過渡応答) にどう影響するか？



水位 (電圧 V) の上がる速さは
管が細い (R が大きい) ほど, 容器が大きい (C が大きい) ほど遅い

速さは RC の積 (時定数)
で決まる

$0 \text{ s} \sim RC$ で約63%上昇
 $= 1 - 1/e$



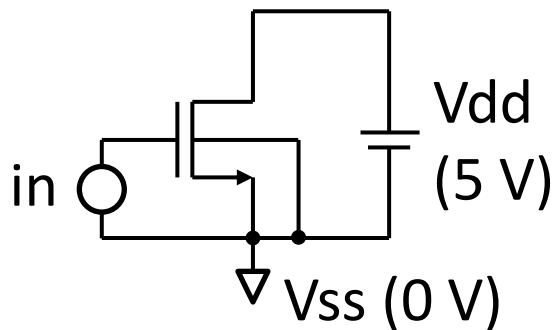
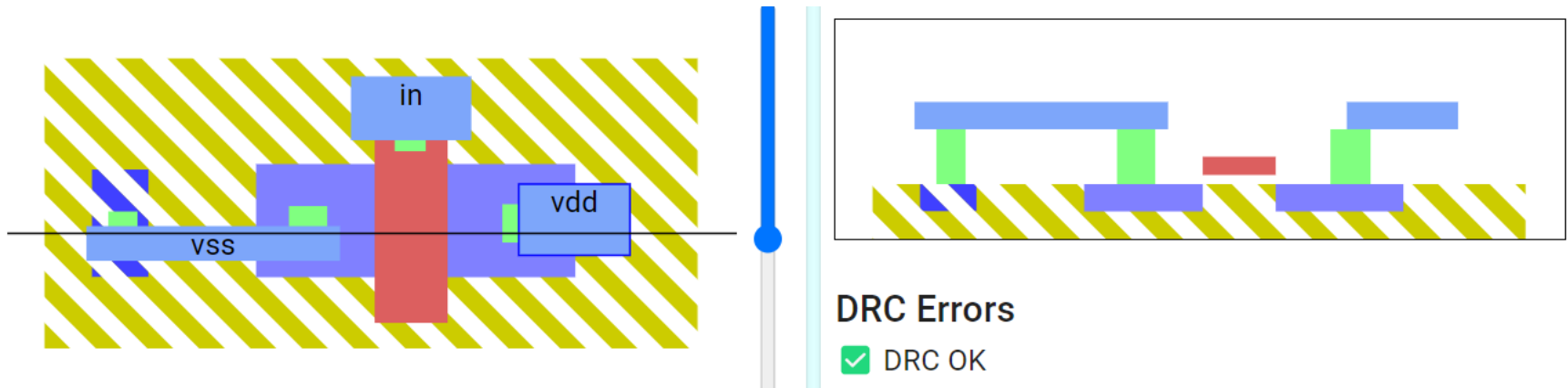
※ Siliwiz では見やすい RC の値を得るのはちょっと難しい

Work 12: MOSトランジスタ

適当に nMOS を描く

Diffusion の片方 (ソース) を Vss, もう片方 (ドレイン) を Vdd

ゲートポリシリコン (ゲート) を in, 基板タップ (バックゲート) を Vss

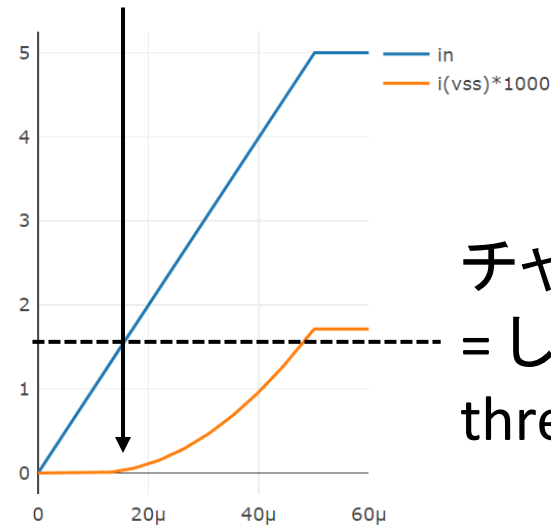
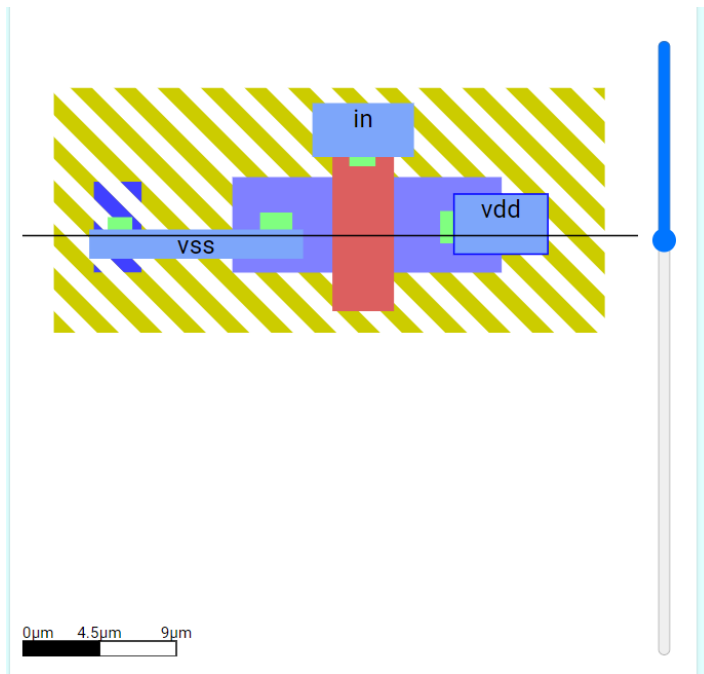


ゲート (in) の電圧が上がると
反転層 (チャネル) で
ドレイン (vdd) とソース (vss) が
つながって電流が流れるはず

Work 12 つづき

電流をプロットしてみると

電流が流れ始める
= この辺でチャネルができた



チャネルができる電圧
= しきい値電圧
threshold voltage, V_{th}

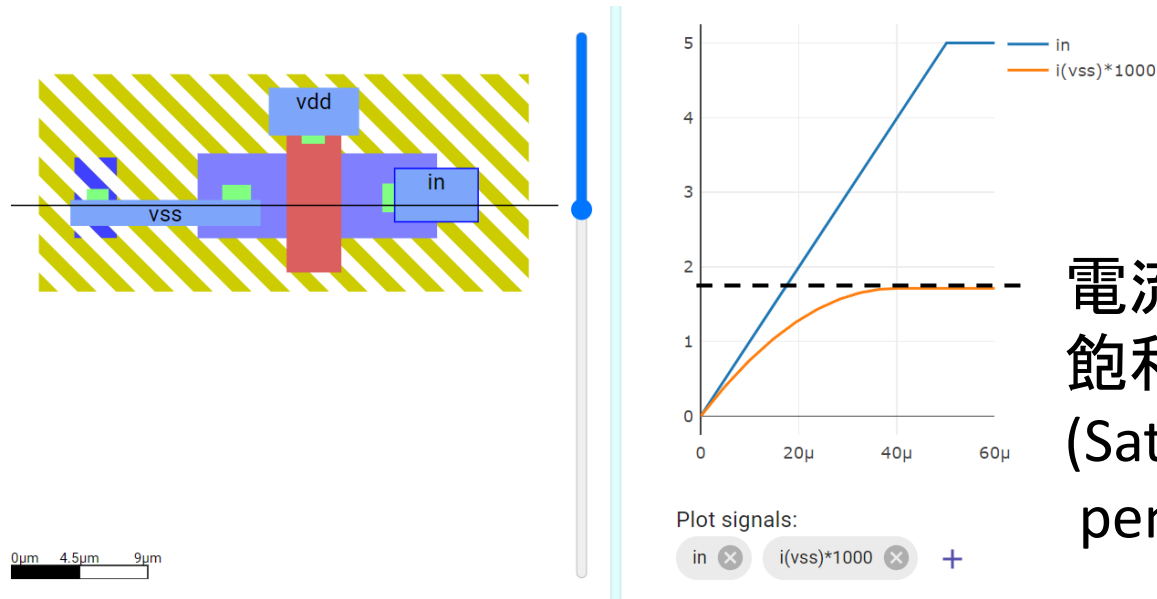
Plot signals:

in × i(vss)*1000 × +

MOSFET の I_d - V_{gs} 特性 (普通はDC解析で出します)
ドレイン電流 – ゲートソース間電圧

Work 13: 別の見方

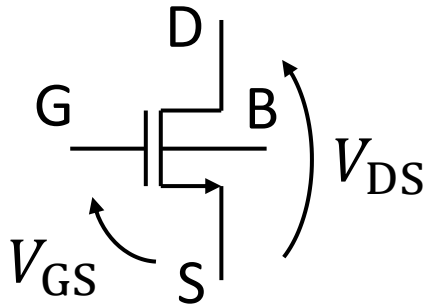
ゲートと vdd に, ドレインを in にしてみる



電流はある値で飽和
飽和領域
(Saturation region,
pentode region)

MOSFET の I_d - V_{ds} 特性 (普通はDC解析で出します)
ド레인電流 - ド레인ソース間電圧

MOSFETの特性



ゲートとソースの間の電圧 V_{GS} が
しきい値電圧 V_{th} を超えるとチャネルができて
ドレインとソースの間に電流が流れる

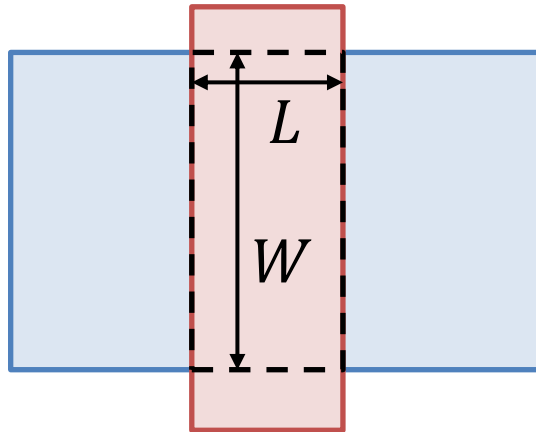
ドレインとソースの間の電圧 V_{DS} が
ある値を超えると電流が飽和して一定値になる
= MOSFET が電流源として振る舞う

$V_{GS} < V_{th}$: カットオフ, 非反転領域, サブスレシヨルド領域
ドレインに電流は流れない

$V_{GS} > V_{th}$ かつ $V_{DS} < V_{GS} - V_{th}$: 線形領域
MOSFETは抵抗として振る舞う (ほとんど使わない)

$V_{GS} > V_{th}$ かつ $V_{DS} > V_{GS} - V_{th}$: 飽和領域
MOSFET は電流源として振る舞う (ここを使う)

Work 14: 電流を増やすには



diffusion と polysilicon の
オーバーラップ部分
= チャンネルができる部分の形状が影響

チャンネル長 L , チャンネル幅 W を用いて飽和領域では

$$I_D = \frac{1}{2} \mu C_{\text{ox}} \frac{W}{L} (V_{\text{GS}} - V_{\text{th}})^2$$

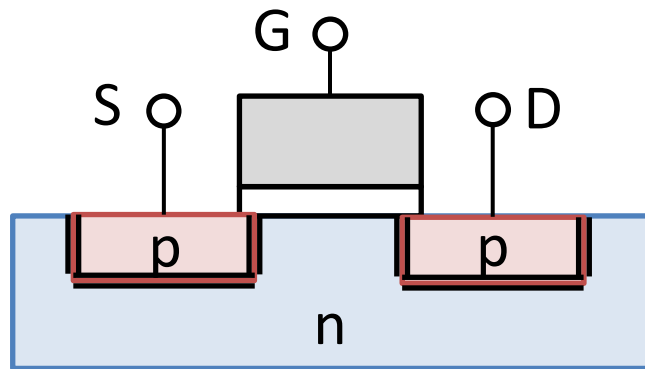
電流は W/L (アスペクト比) に比例する

チャンネル部分のサイズを変えてみて電流の増減を確認する
(W と L の取り違えに注意)

Work 15: pMOSFETの場合

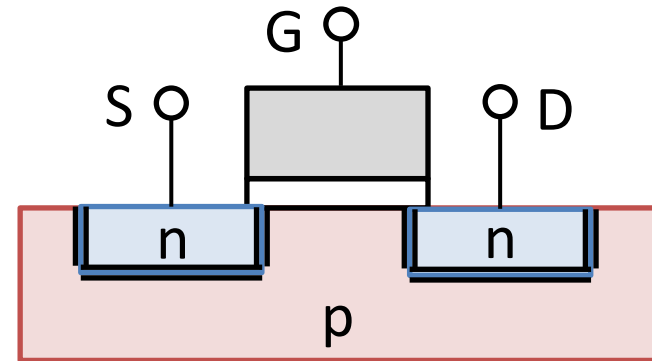
pMOSFET を描いて同様に特性を確認しよう

電圧と電流の向きに注意



pMOS

基板の電位を高く,
ゲートの電位を低くすると反転
ソースから正孔が供給される
= 電流は $S \rightarrow D$ の向き



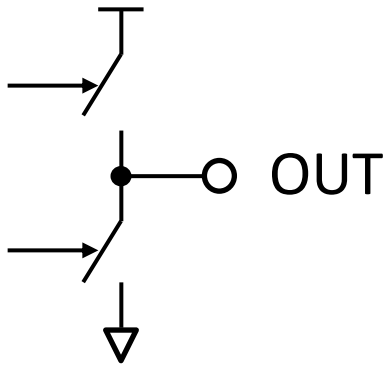
nMOS

基板の電位を低く,
ゲートの電位を高くすると反転
ソースから電子が供給される
= 電流は $D \rightarrow S$ の向き

論理ゲートを作る

論理ゲート

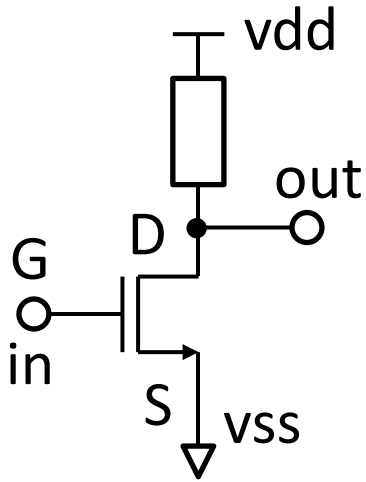
MOSFETをスイッチとして使う



出力端子を電源につなげれば 1 (HI)
グラウンドにつなげれば 0 (LOW)

MOSFET はチャネルを消せばスイッチOFF
チャネルを作ればスイッチON

Work 16: スイッチ1個で作る



入力 0 (LOW)

→ $V_{GS} < V_{th}$ なので nMOS は OFF

→ out は抵抗で vdd に釣り上げられて 1 (HI)

入力 1 (HI)

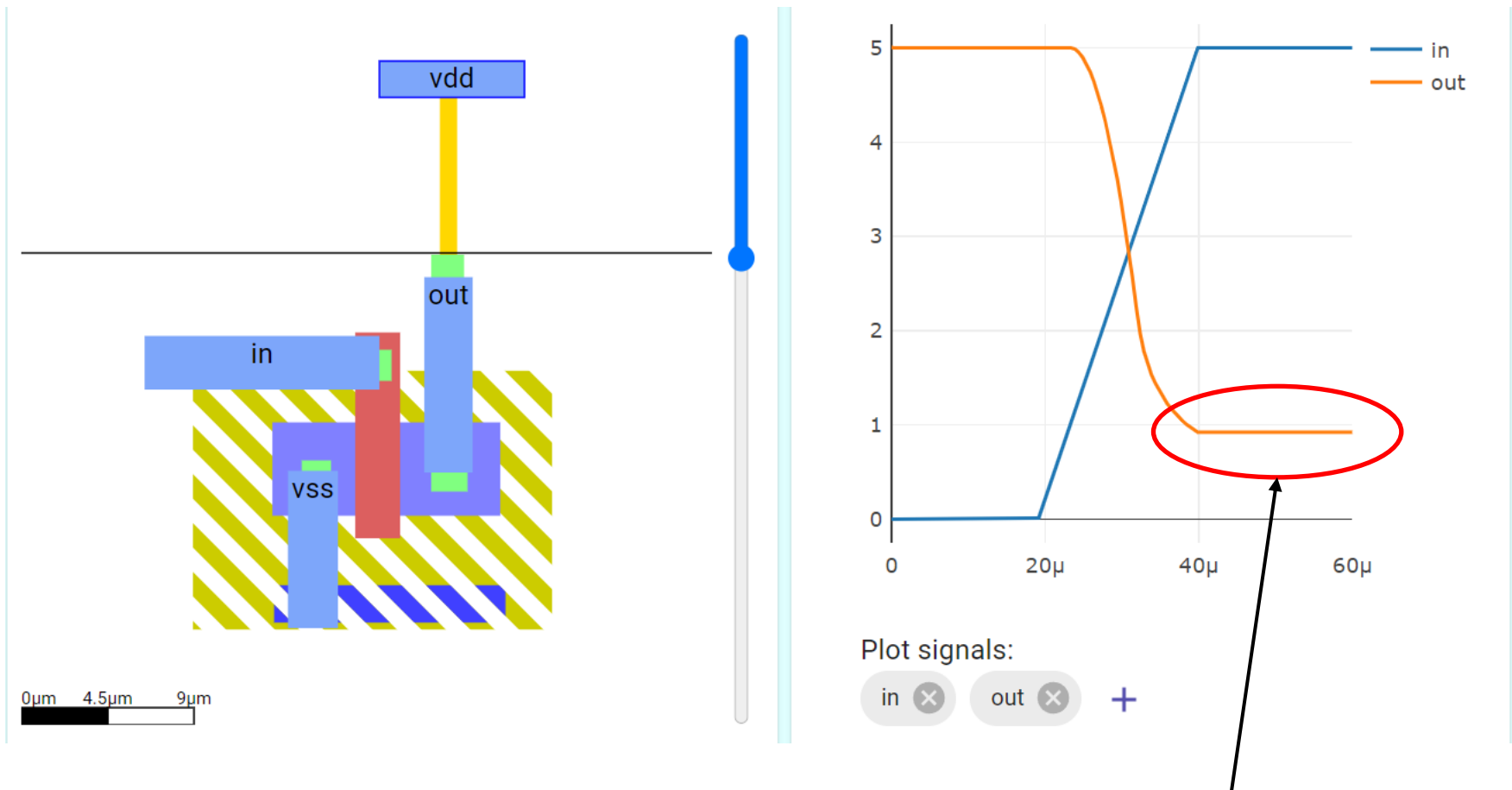
→ $V_{GS} > V_{th}$ なので nMOS は ON

→ out は nMOS が vss につなぐので 0 (LOW)

論理否定 (NOT, INV) になる

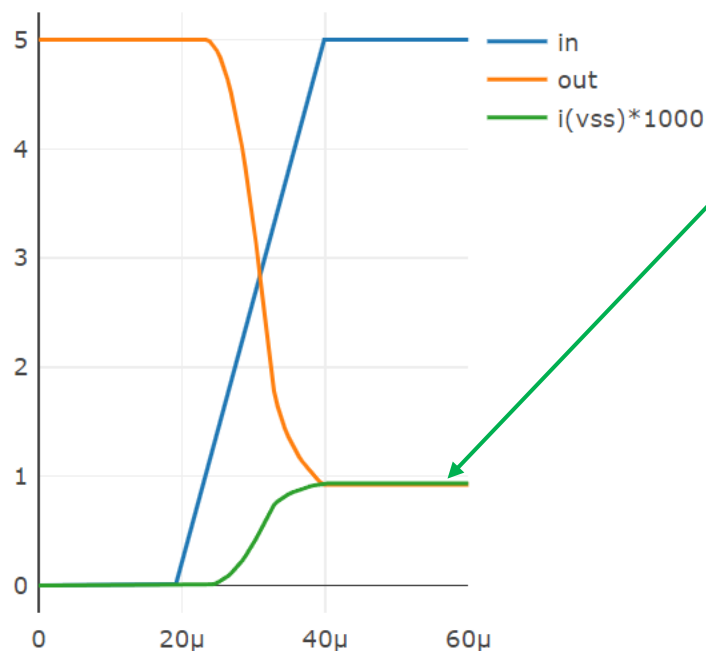
INV : inverter

Work 16: こんな感じ



ちゃんと LOW が 0 V になっていないが
NOTゲートになっている

Work 16: おまけ: 消費電力



$i(vss)*1000$ で
電流をプロットしてみると...

出力が1のときは電流0 (消費電力0)
出力が0のときは電流が流れる
(電力を消費する)

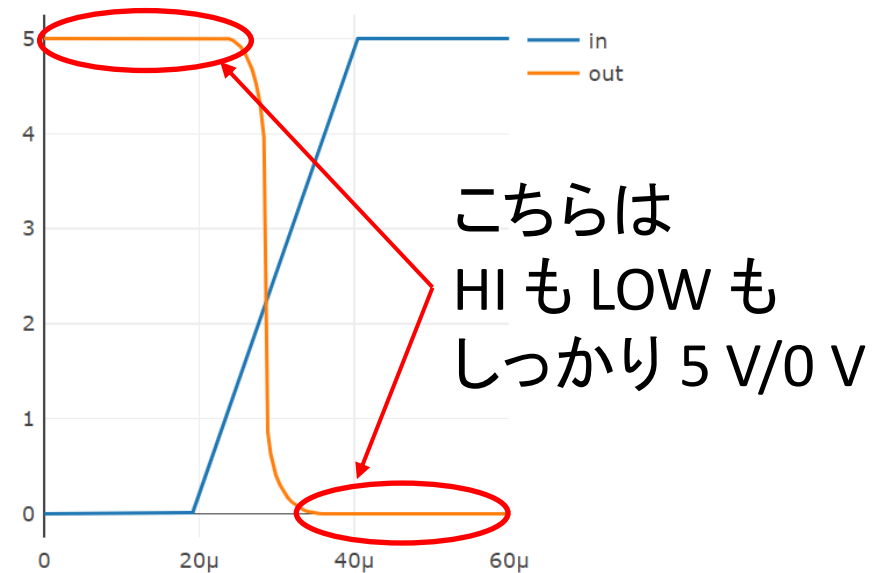
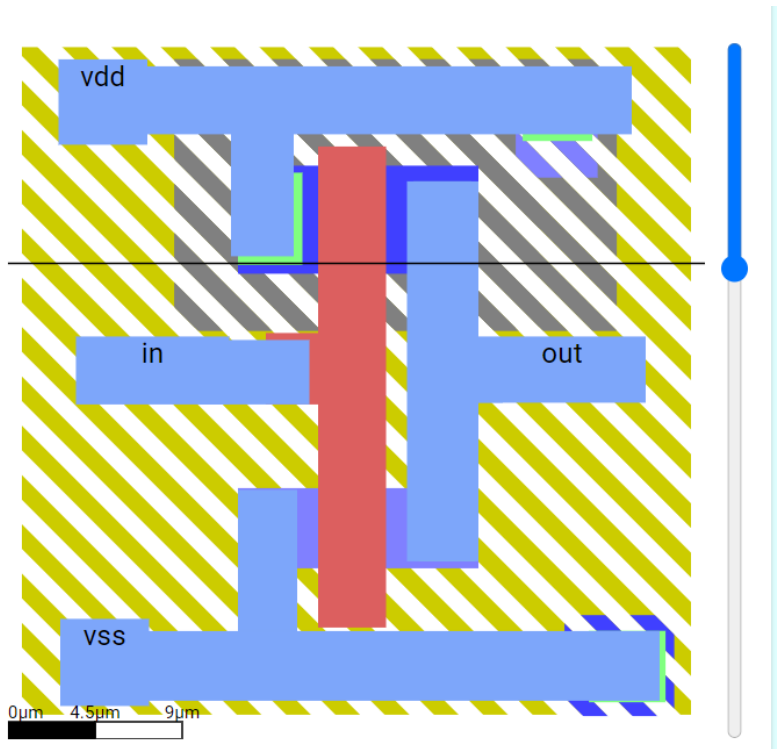
Plot signals:

in × out × $i(vss)*1000$ × +

Work 17: CMOS INV

pMOS と nMOSを使って INVゲートを作ろう

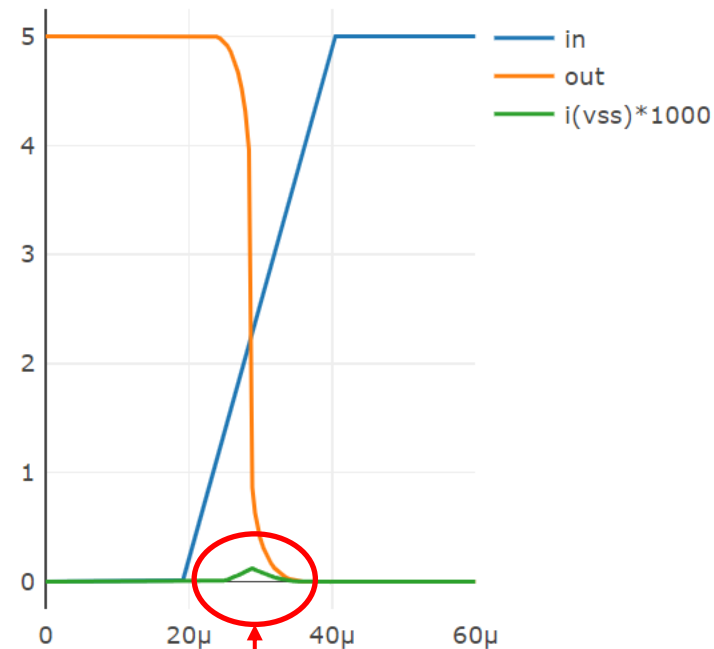
preset の inverter.json をロード



Plot signals:

in × out × +

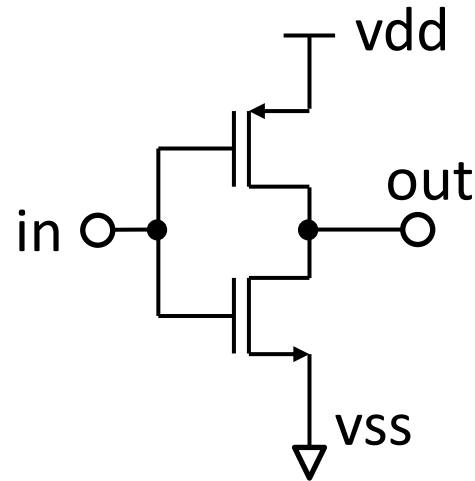
Work 17: つづき



Plot signals:

in ☐ out ☐ i(vss)*1000 ☐ +

電流も一瞬
ちょっと流れるだけ
= 低電力



in が 0 : pMOSが ON, nMOS が OFF
pMOS が引き上げて out は 1

in が 1 : pMOSが OFF, nMOS が ON
pMOS が引き上げて out は 0

必ずどちらかが OFF で電流を遮断
相補的 (Complementary) MOS = CMOS

Work 18: NAND と NOR を作ろう

NAND			NOR		
A	B	Y	A	B	Y
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

NOT-AND

入力が両方1の
ときだけ0
それ以外は1

NOT-OR

入力が両方0の
ときだけ1
それ以外は0

Siliwiz は入力を1つしか作れないので
もう一方の入力は vdd (1)につなぐか vss (0)につないで観察

まとめ

- Siliwiz で集積回路設計体験
 - ◆ 設計方法としては原始的
 - ◆ デバイス構造などが理解できる (はず)
- レイアウト, DRC, シミュレーション
 - ◆ CMOS論理ゲートの設計まで
 - ◆ これ以上の内容は Siliwiz のシミュレーションではちょっと厳しい...
- 是非 OpenMPW等のテープアウトに挑戦してください

各コミュニティ入口

ISHI会 (本イベント主催)

<https://ishikai.connpass.com/>

Open Source EDA Supporters

[Discord への招待ページ](#)

Make LSI

<https://scrapbox.io/makelsi/>

open-source-silicon.dev

<https://join.skywater.tools/>

#japan-region が日本人チャンネル