

集積回路設計 (INTEGRATED CIRCUIT DESIGN) 第10回

2020/06/05

レイアウト設計（5章）

9

- レイアウト設計の流れ
 - ▣ フロアプラン・配置配線
 - ▣ レイアウト検証
 - ▣ パターン設計規則
 - ▣ マスク・リソグラフィ
- 様々な設計方式
 - ▣ ASIC: ゲートアレー/スタンダードセル/フルカスタム/エンベデッドアレー
 - ▣ FPGA
- 配置配線
 - ▣ 配置ルール・アルゴリズム
 - ▣ チャネル配線ルール・アルゴリズム

レイアウト設計

10

回路・面積情報

組合せ回路・順序回路の設計（～4章）

1. フロアプラン

- 設計したモジュールの配置の概略を（データの流
れ，面積見積もりをもとに）決定
- 面積の見積もりはなるべく正確に（ただし難）
- 最終的な回路設計に大きく影響

2. 配置

- 配置の詳細化

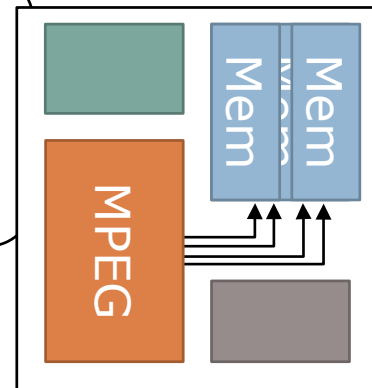
3. 配線

- どの空間＆何層使って配線するか決定
- 配線長＝信号遅延（距離の2乗で影響）
→バッファで増幅（バッファの配置も考慮）

4. レイアウト検証

- 所望の動作をするか確認（手設
計やツールのバグ）
- LVS (Layout Vs Schematics)
- DRC (Design Rule Check)

マスク情報



フロアプラン・配置

11



フロアプラン

12

□ モジュールレベル

□ チップ上のモジュールの配置を決定

- モジュールの形状を矩形に限定し，大きさの異なる矩形をなるべく隙間なく詰める

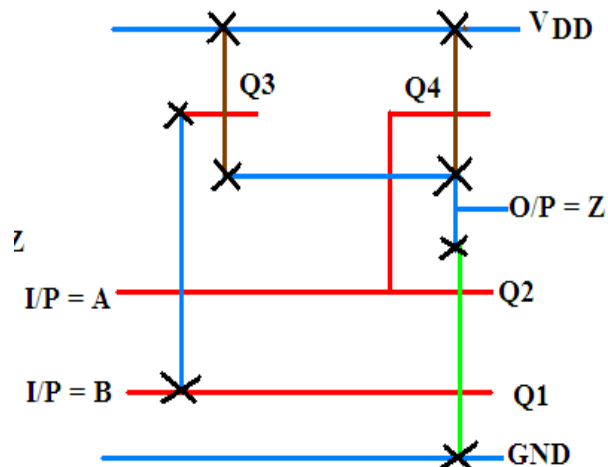
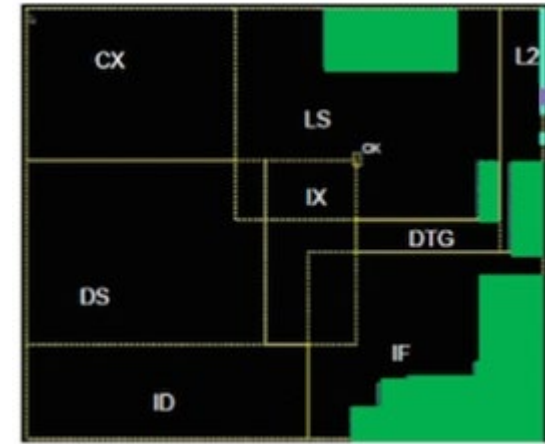
- **矩形パッキング問題**（NP困難）

□ モジュール間のデータの流を考慮

□ モジュール内

□ **スティックダイアグラム**

- トランジスタや配線を**線分**で表し，位置情報を付加



- モジュールを配置するための領域（矩形）をシリコンチップ上に割り当てるための表現と方法
- 矩形配置の表現
 - ▣ スライス構造
 - ▣ スライス木構造
 - ▣ Sequence-Pair
 - ▣ など

スライス構造・スライス木構造

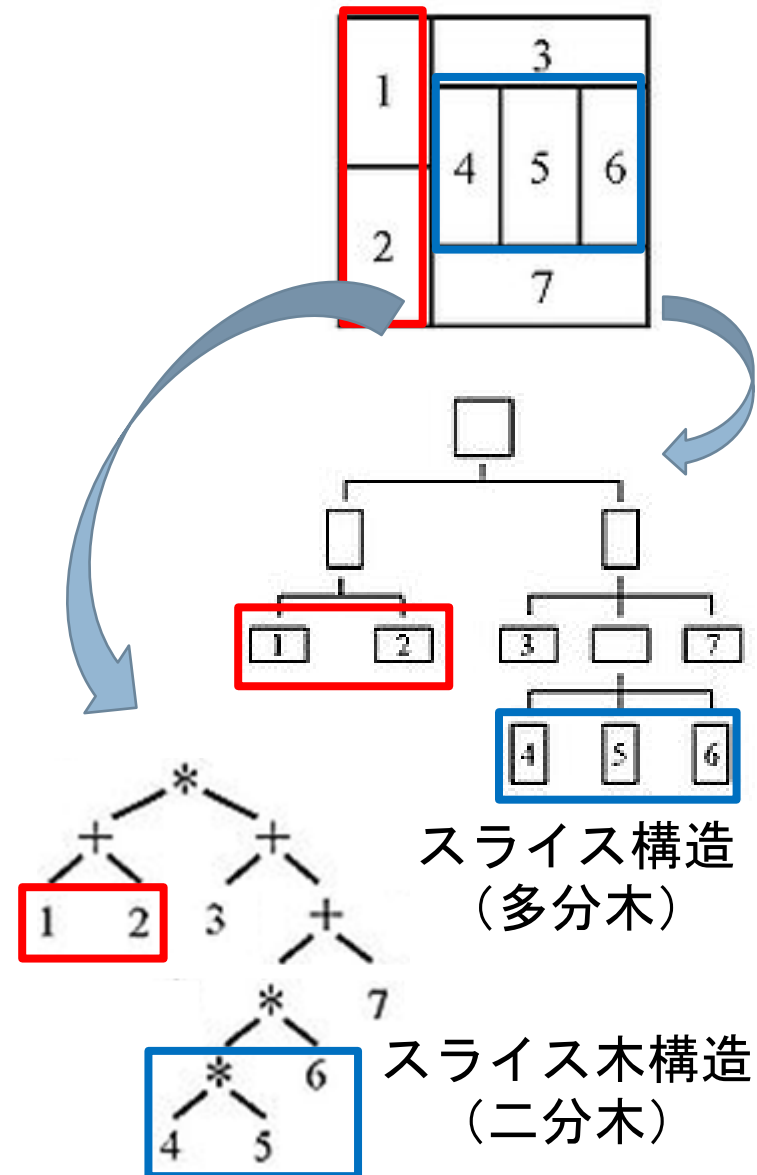
14

□ スライス構造

- 集積回路のシリコンチップ全面を**垂直**線分または**水平**線分にて**再帰的**に分割した構造
- 分割後の矩形領域を節点（ノード）とした**多分木**

□ スライス木構造

- スライス構造を**二分木**に改良
- + 記号：**水平**線分による分割
- * 記号：**垂直**線分による分割



Sequence-Pair

15

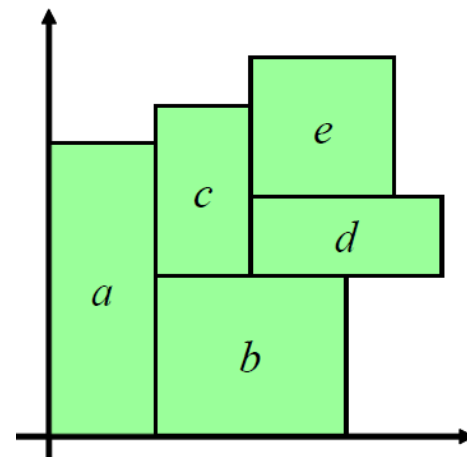
- $[(a,b,c),(b,a,c)]$: 矩形名の2つの順列で矩形の相対的な位置（上下左右）を表現

1. $[(\dots a \dots b \dots), (\dots a \dots b \dots)] \rightarrow b$ は a の右
2. $[(\dots b \dots a \dots), (\dots a \dots b \dots)] \rightarrow b$ は a の上
3. $[(\dots b \dots a \dots), (\dots b \dots a \dots)] \rightarrow b$ は a の左
4. $[(\dots a \dots b \dots), (\dots b \dots a \dots)] \rightarrow b$ は a の下

- Simulated Annealing (SA) と合わせて用いると矩形パッキング問題の解法として有効

- 例) $[(a,c,e,d,b),(a,b,c,d,e)]$

- b は a の右（ルール1）
- b は c の下（ルール4）
- Etc...



配置の表現（例題）

16

- 右図に示すモジュール配置をそれぞれ以下の方法で示せ

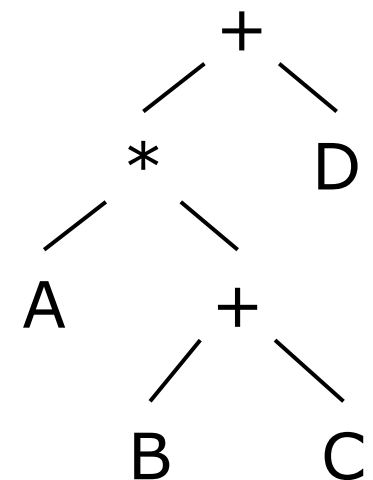
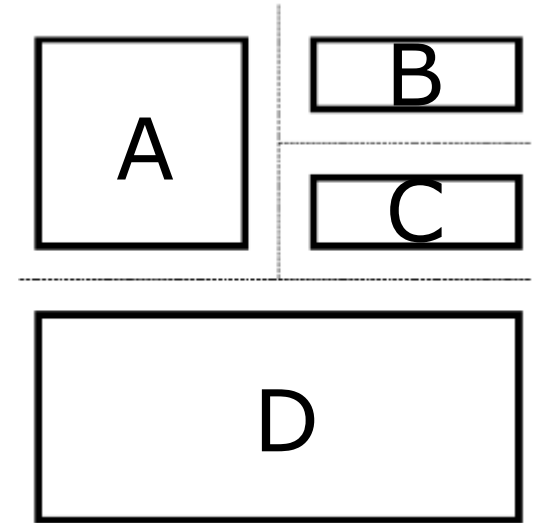
- スライス木構造

- + 記号：水平線分による分割
- * 記号：垂直線分による分割

- Sequence-Pair

- bはaの右： $[(...a...b...), (...a...b...)]$
- bはaの下： $[(...a...b...), (...b...a...)]$

→ $[(ABCD), (DACB)]$



配線

17

縦の配線（赤）と横の配線（青）を別層

□ 交差する配線は別層

- 別々にリソグラフィ

- 上下の配線層をビア（コンタクト）で接続

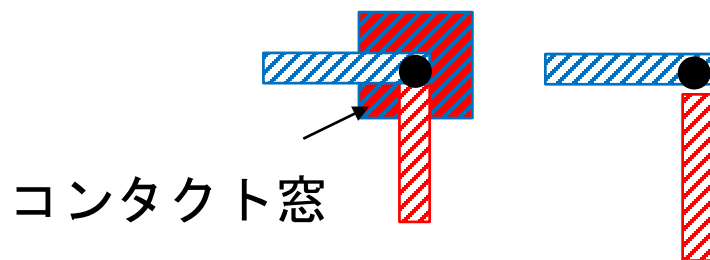
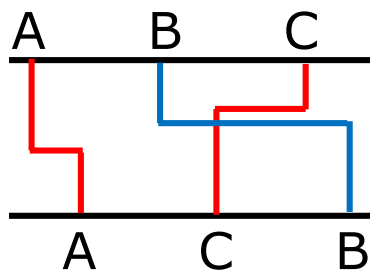
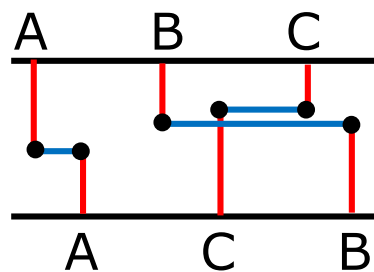
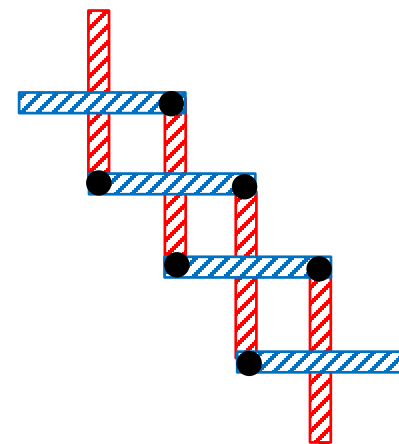
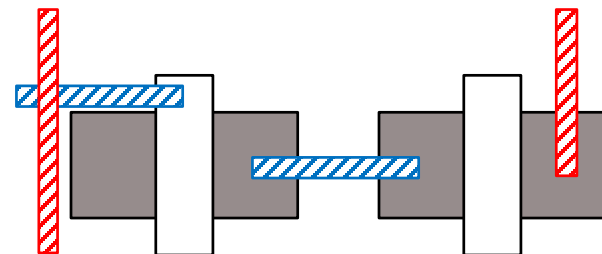
 - 上下でずれることも考慮 → コンタクト窓

- ビア数最小化 ⇔ リソグラフィ容易さ

 - ずれのリスク & 面積削減？

 - リソグラフィを考慮？

- 各層の組み合わせを考慮する必要がある



レイアウト検証

18

- DRC (Design Rule Checking)
 - ▣ 製造装置の制約から決まる幾何学的な設計ルールを満足しているかどうかを検証
- LVS (Layout Versus Schematic)
 - ▣ 論理・回路設計結果のデバイス（トランジスタなど）やデバイス間の接続が、レイアウト設計で正しく実現されているかを検証
- ERC (Electrical Rule Checking)
 - ▣ レイアウトのパターンの電氣的な設計ルールを検証

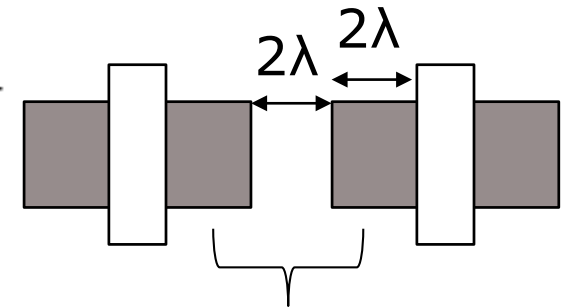
パターン設計規則

19

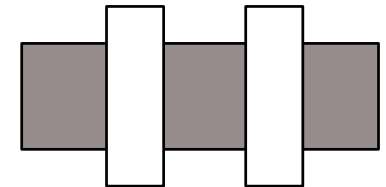
デザインルール (p167)

拡散層	最小幅	2λ
	最小間隔	2λ
	pn 拡散層間最小間隔	8λ
p ウェル	最小幅	4λ
	同電位 p ウェル間最小間隔	2λ
	異電位 p ウェル間最小間隔	6λ
	内部拡散層間最小間隔	3λ
	外部拡散層間最小間隔	5λ

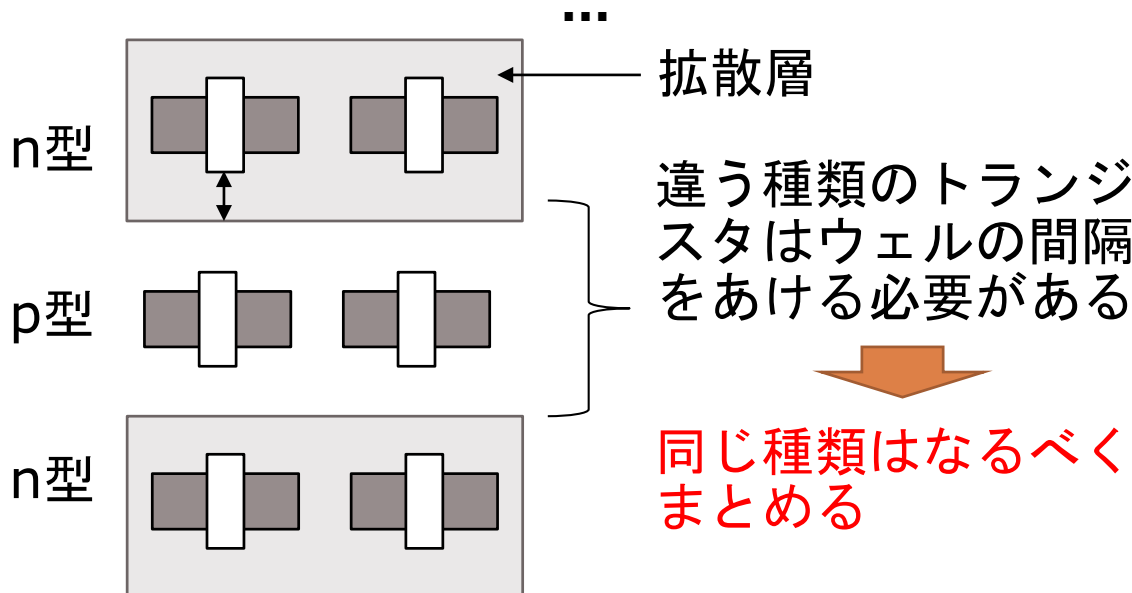
λ : 定数 (プロセスが変わってもルールをなるべく一般化できるように)



離して配置しないと導通

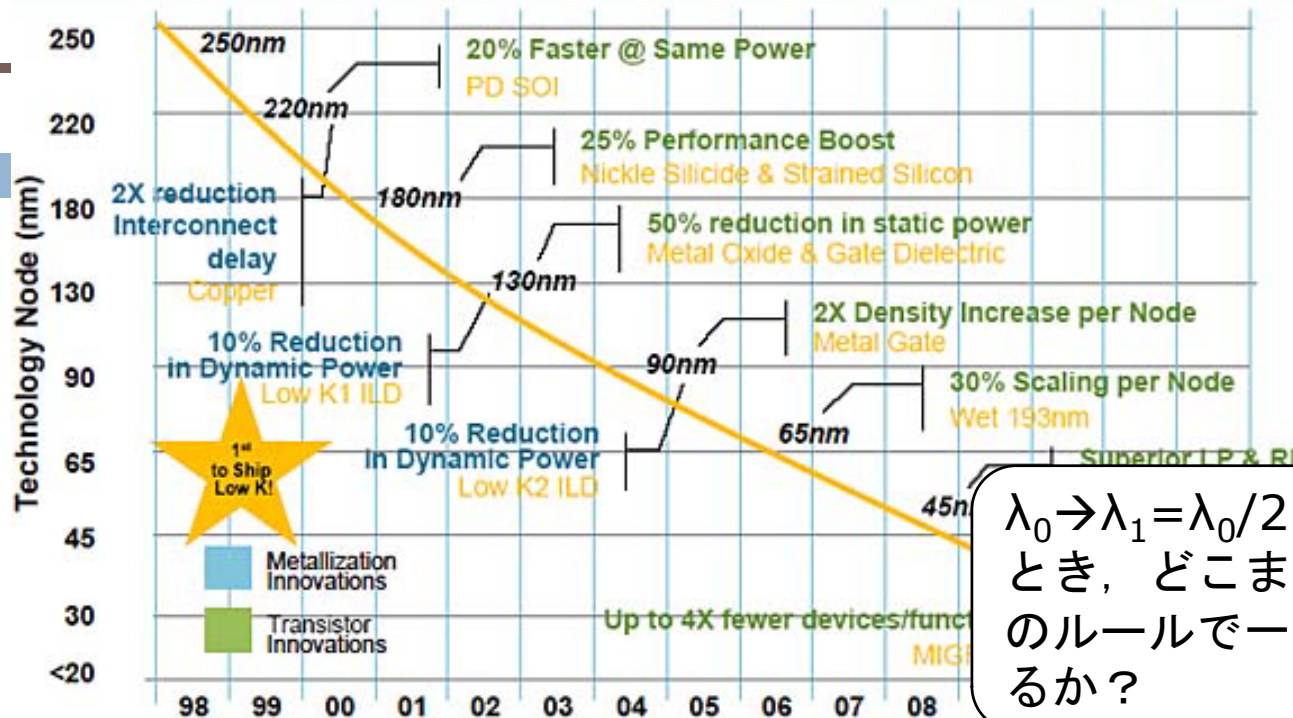


同電位なら繋げて
つくって面積削減

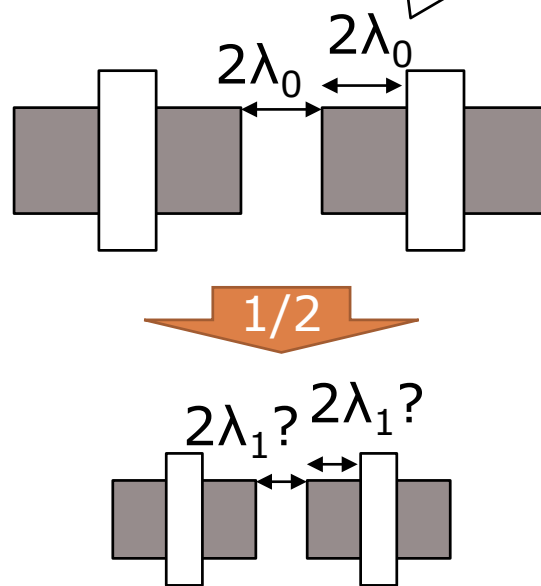
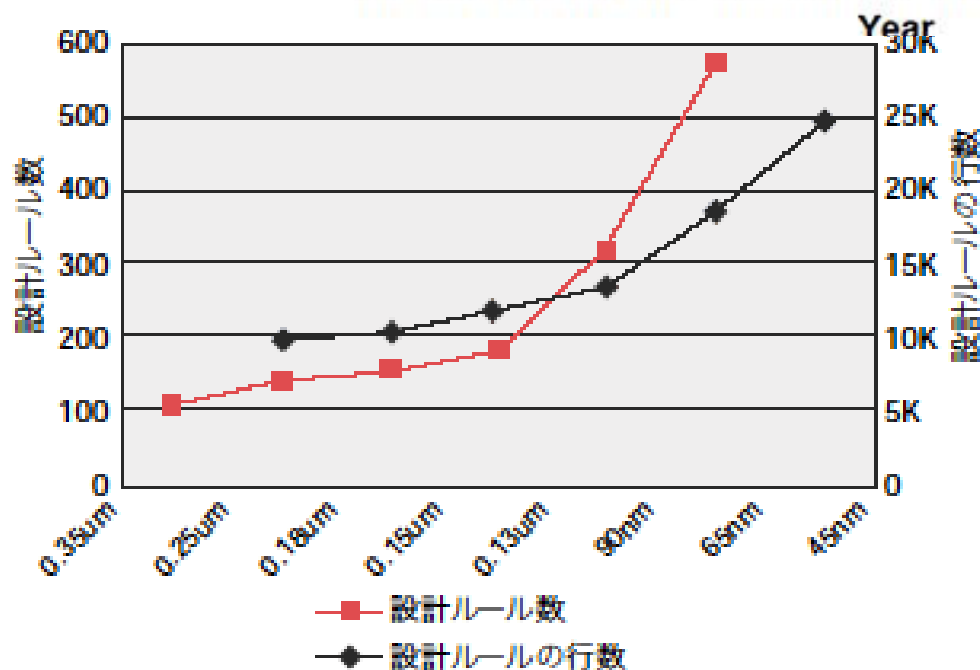


パター

20



$\lambda_0 \rightarrow \lambda_1 = \lambda_0 / 2$ となったとき、どこまでも 2λ のルールで一般化できるか？



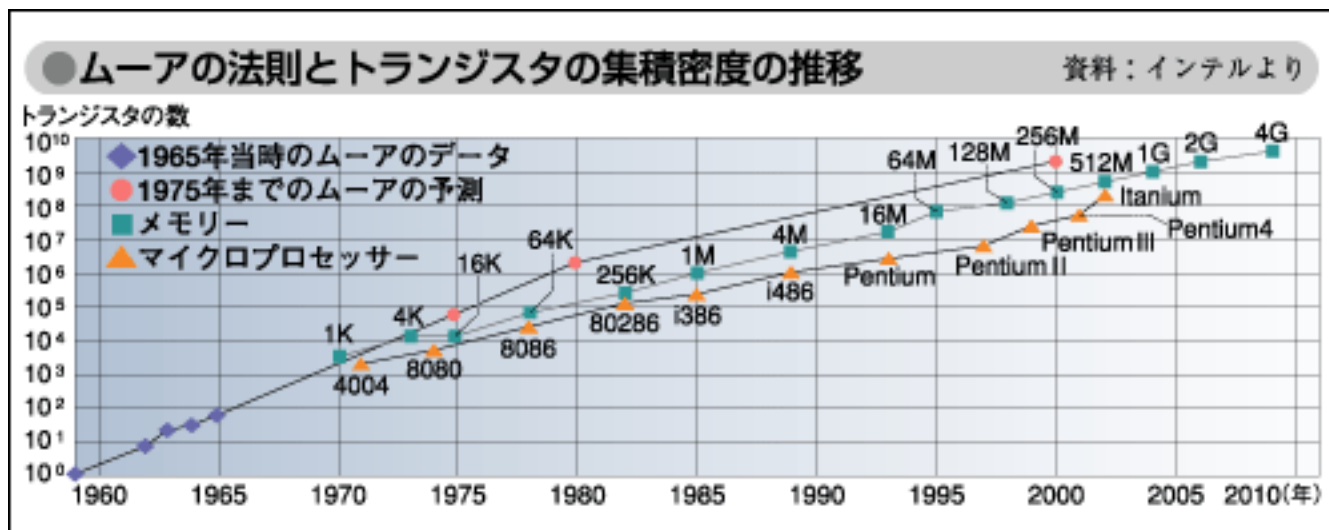
(豆知識) CMOS微細化

21

- Dennard Scaling/デナード則
 - ▣ Dennard=DRAM開発者（開発年：1968年）
 - ▣ MOSFETを微細化するときのメリットを工学的に導出した法則（???年）
 - ▣ ゲート寸法を k 倍にすると ($k < 1$)
 - 集積率は $1/k^2$ 倍
 - 遅延は k に削減（= $1/k$ 倍高速化）
 - 消費電力は k^2 に削減
 - ▣ $k=0.7$ になるように微細化が進んできた
- Moore's law/ムーアの法則
 - ▣ Moore=インテル創始者の1人
 - ▣ 半導体の集積率が18ヶ月で2倍になる（1965年）
 - デナード則をベースに

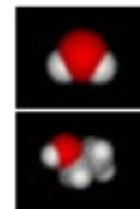
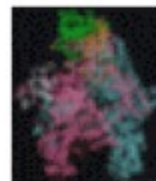
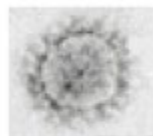
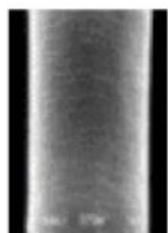
(豆知識) CMOS微細化

22



出典：<https://imidas.jp/genre/detail/K-107-0073.html>

Hair Red blood cell E. coli Virus Protein DNA Molecule Atom



0.1 μm
(100mm)

10 μm

1 μm

100 nm

10nm

1nm

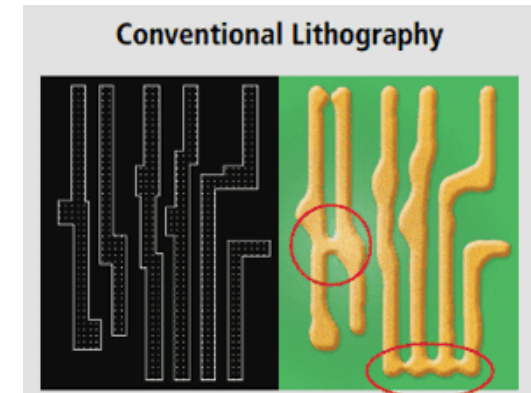
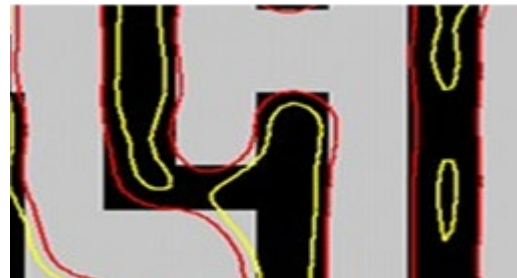
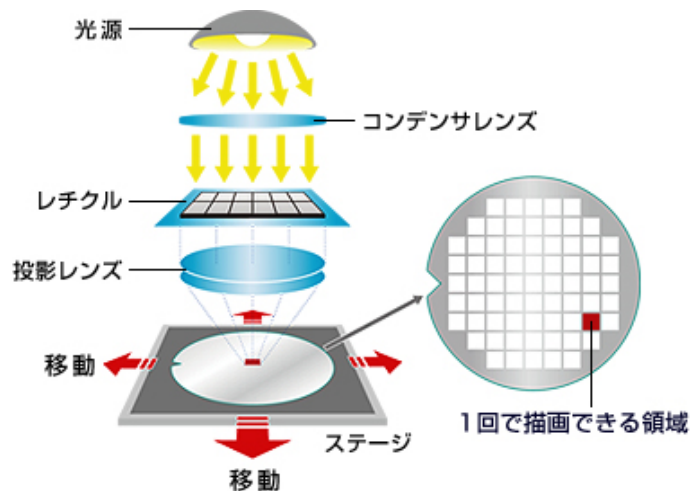
0.1nm
(1Å)



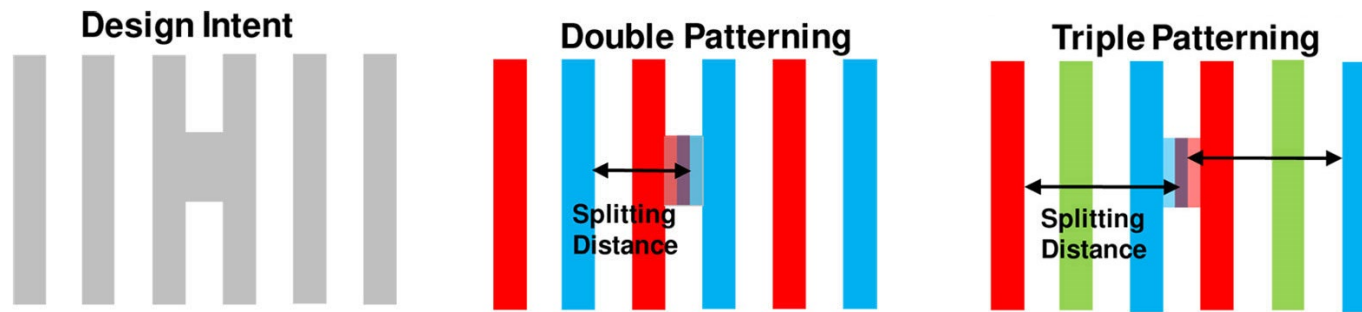
マスク

23

- 最先端デバイスではリソグラフィ難
 - ▣ 可視光 (380nm~) > プロセス (~45nm)
 - ▣ 極端紫外線リソグラフィ(EUV):13.5nm (世界に6台)



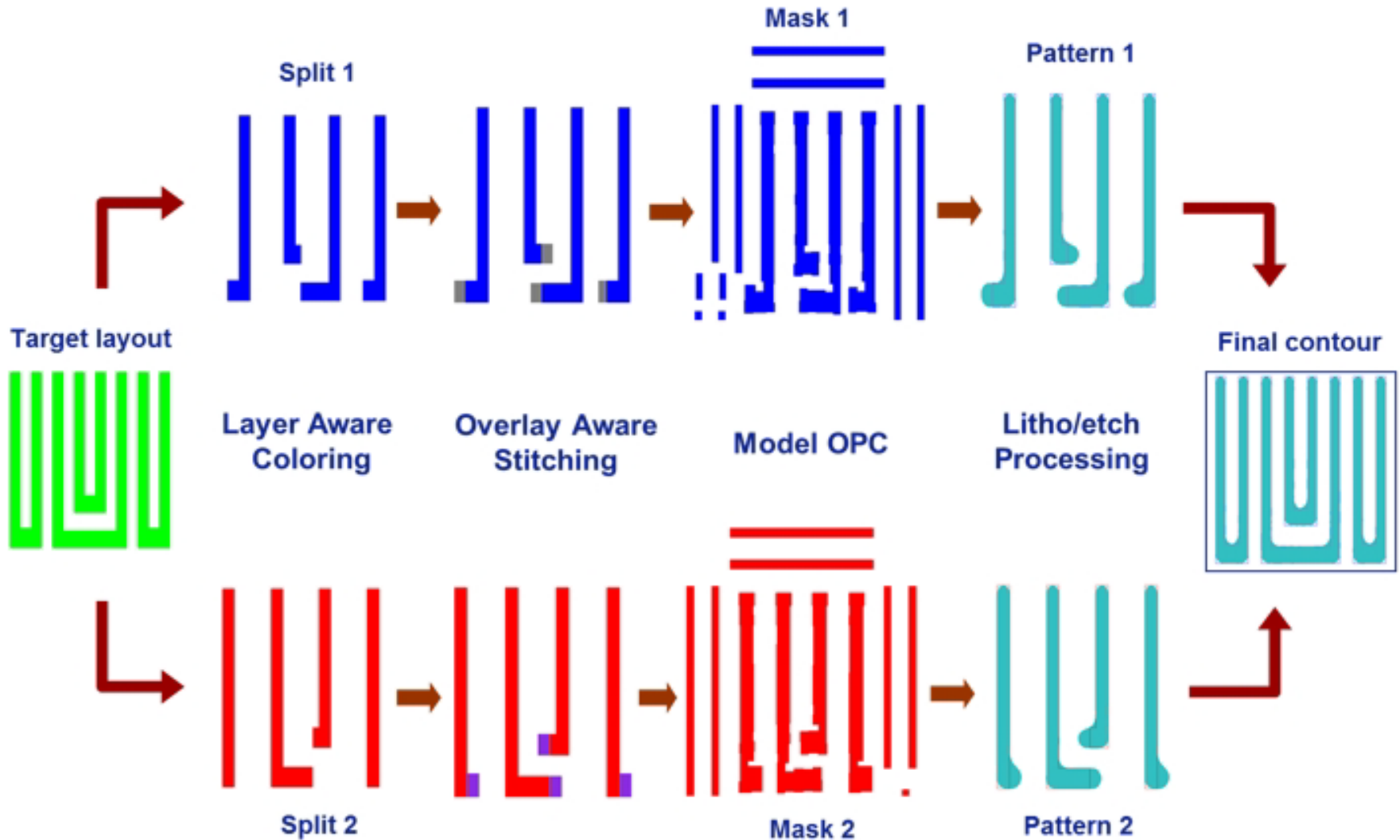
- ▣ 複数回に分けてマスクを生成



リソグラフィ

24

Layer aware coloring and overlay aware stitching



レイアウトスタイル（スタティックCMOS）

25

□ 信号配線

□ 水平方向

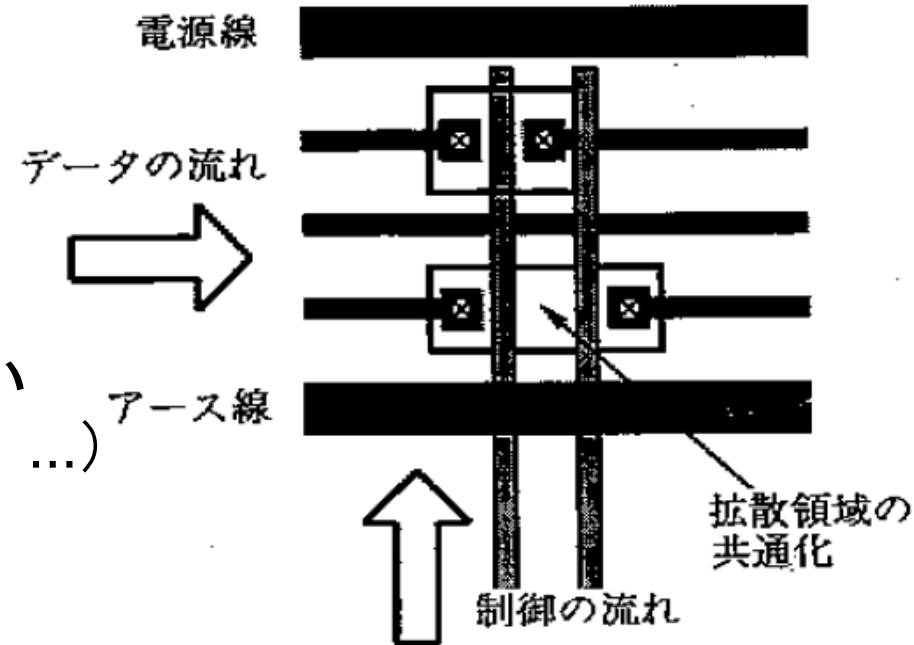
- データの流れ
- 材料：金属
- 長距離配線：抵抗値の低い
メタル層（メタル1， 2， ...）

□ 垂直方向

- 制御の流れ
- 材料：ポリシリコン
- 短距離配線：トランジスタのゲート～隣接する回路まで

□ 電源(V_{DD})・接地線(GND)の配線

- 各論理ゲートに共通に配線＝配線長：大
- 配線幅：広→抵抗：小



レイアウトスタイル（スタティックCMOS）

26

□ トランジスタ

■ 同じ入力を持つpMOSとnMOS:

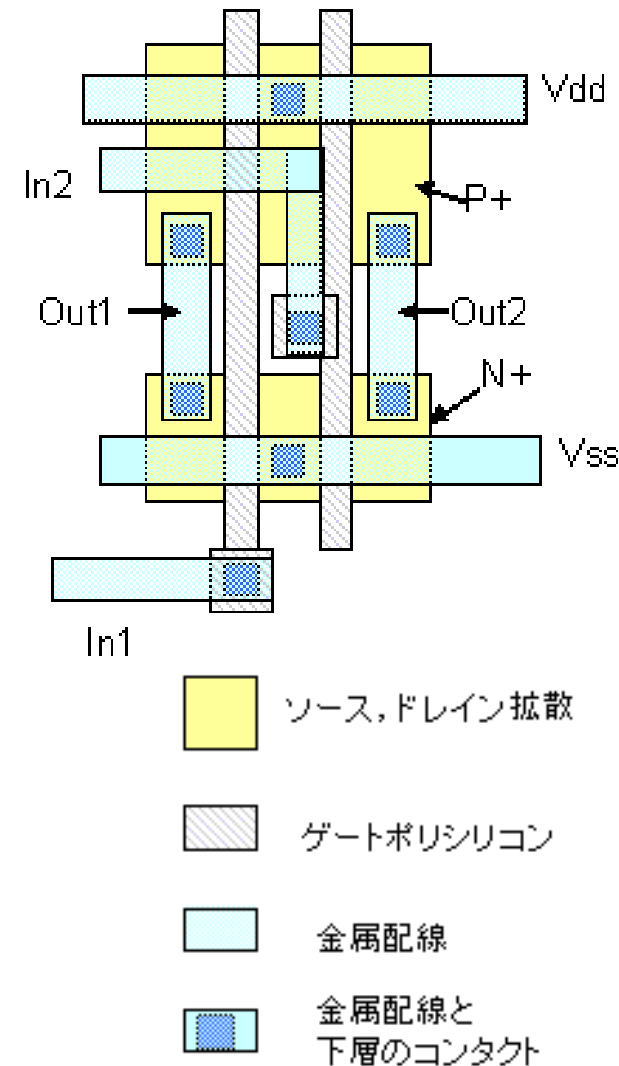
- 上下に配置
- 入力のゲート線を垂直に引く

■ 複数のnMOS (pMOS):

- 一直線上に配置
- それぞれのソース・ドレイン接続は拡散層を共有
 - 配線を省略→面積削減
 - 集めてグループ化

■ コンタクト窓

- 数は少なく
- コンタクト部分：接触抵抗
- 窓：大 → 抵抗：小

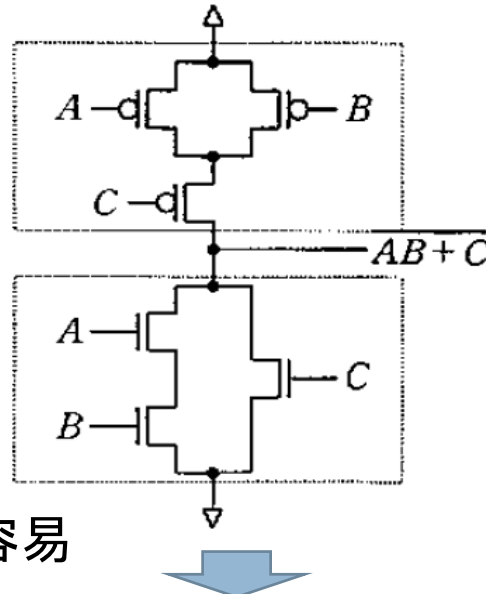


論理ゲートのレイアウトパターン例

27

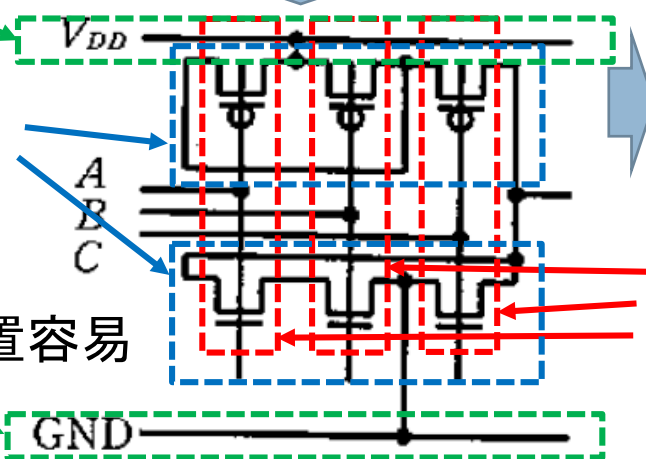
□ $F = \overline{AB} + C$ のCMOS相補型論理回路

■ 回路図

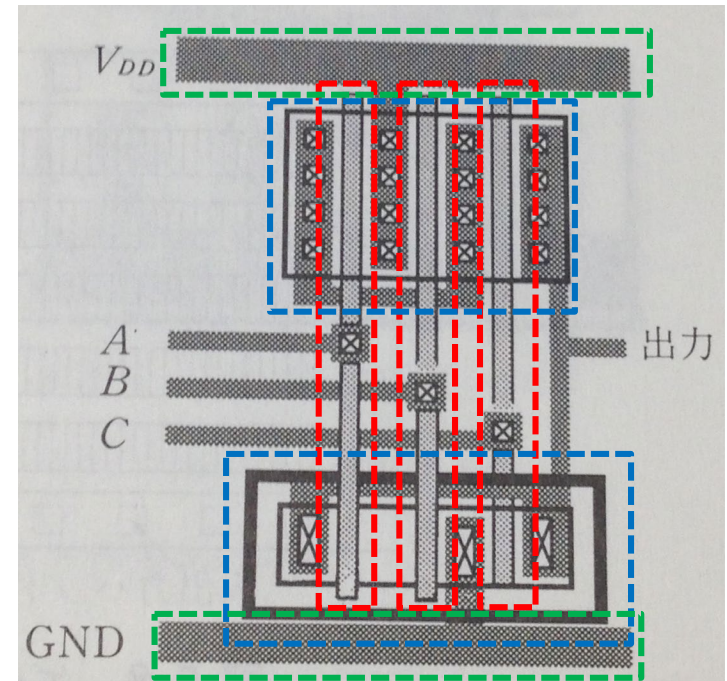


グループ化

接地線を水平に
→nMOSとの配置容易



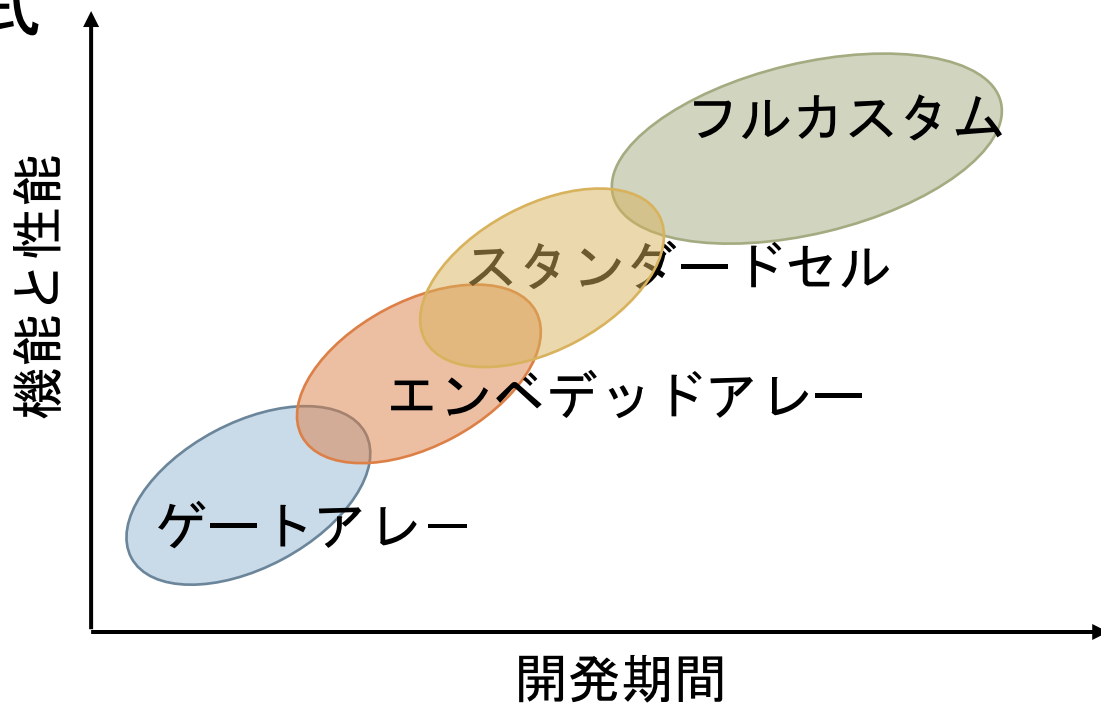
■ レイアウト図



設計方式

28

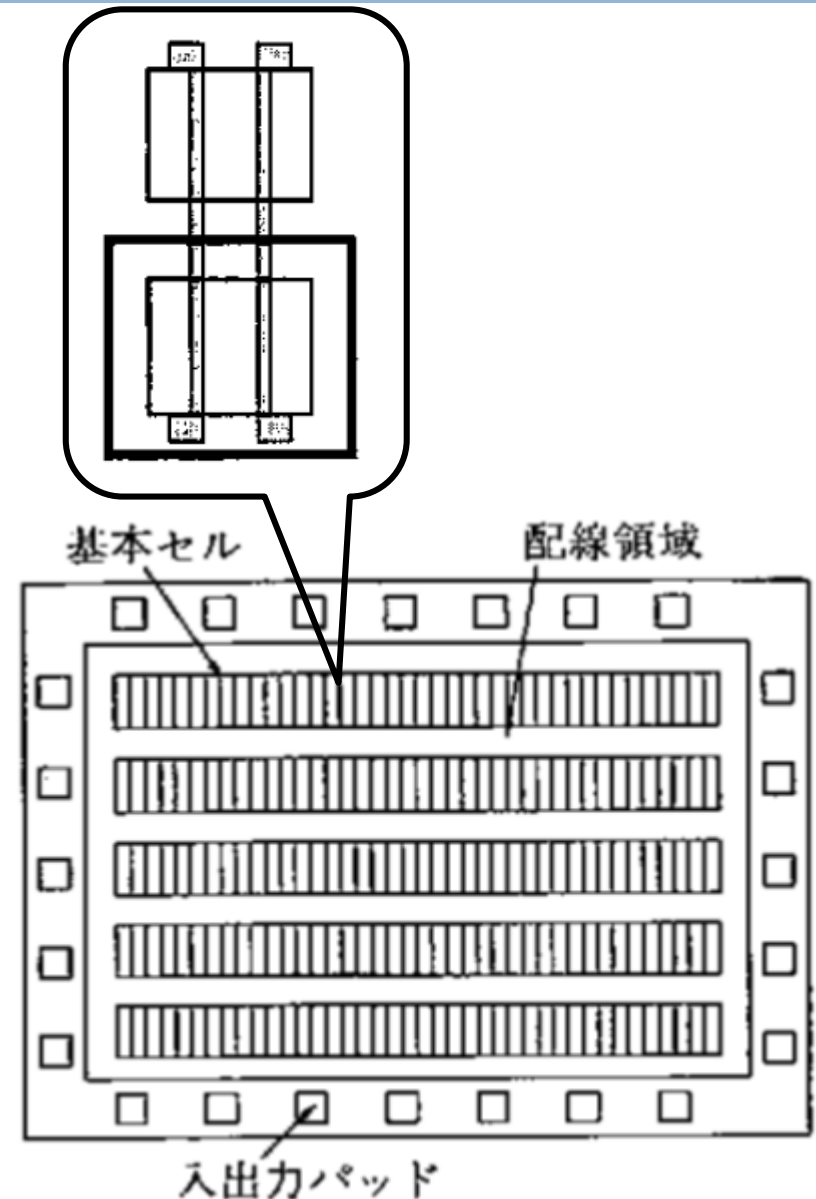
- 様々な設計方式
 - ▣ フルカスタム方式
 - ▣ スタンドードセル方式
 - ▣ エンベデッドアレー方式
 - ▣ ゲートアレー方式



ゲートアレー方式

29

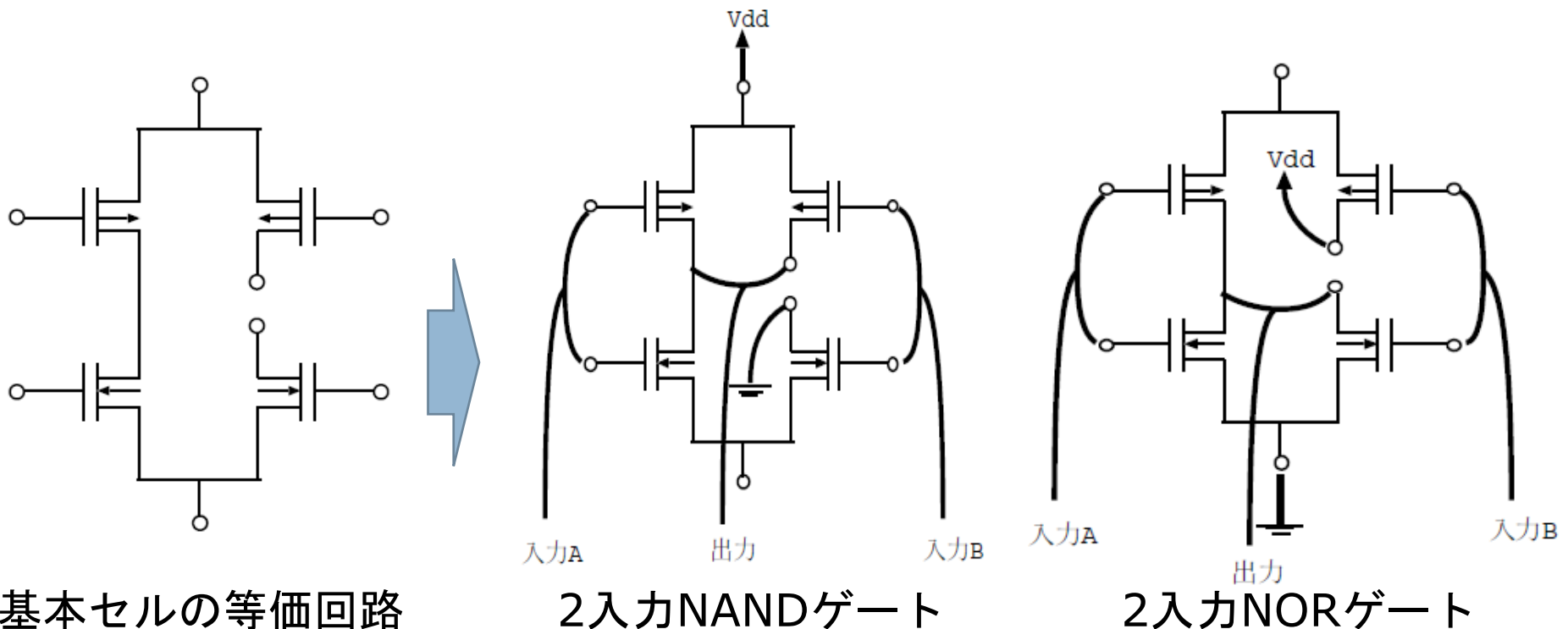
- 論理ゲートがチップ上に列に配置されたもの
 - 基本セル
 - 論理ゲート
 - 基本セルへの金属配線を変えて異なる論理ゲートを実現
 - 配線領域
 - 基本セル間の配線
 - 列の間を利用
 - 入出力パッド領域
 - 金属配線を変えることで入出力を変更可



ゲートアレー方式

30

- 基本ゲートは予め設計（＝セルライブラリ）
 - ▣ 基本論理ゲート：NAND, NOT, NOR, etc...
 - ▣ マクロセル：基本ゲートを組み合わせた大きな論理ブロック（FF, 基本的な複合ゲートなど）



ゲートアレー方式

31

□ 配線だけの設計で論理回路を実現可

- 配線はチップの最上部
- それ以外の層は予め製造

□ 設計方法

1. 論理割付設計（テクノロジマッピング）

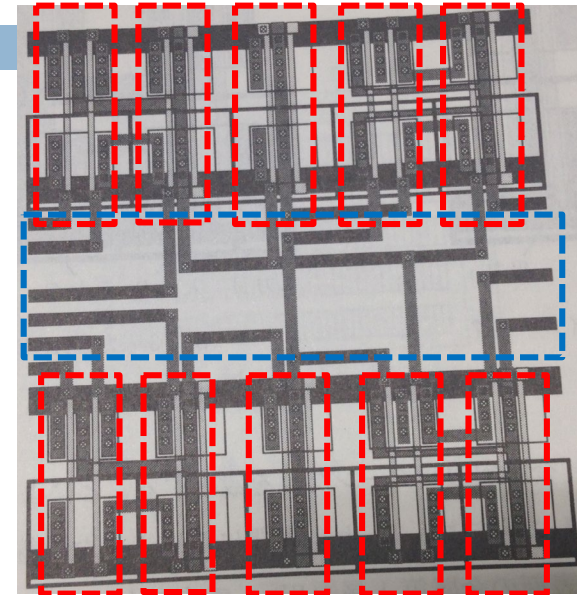
- 実現する論理回路をセルライブラリ中の**基本/マクロセルの組み合わせ**で表現
- プロセスによってライブラリは異なる

2. 配置設計

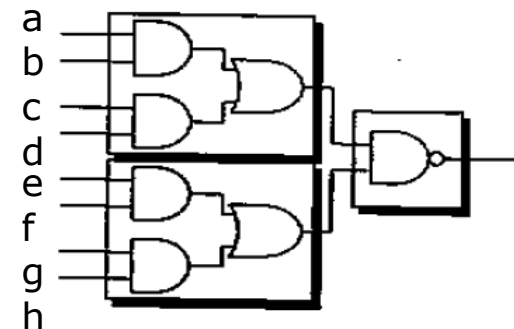
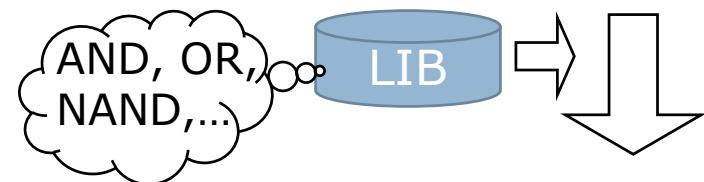
- セルにマップ
- 配線見積もりをもとに決定

3. 配線設計

- セル間の配線経路を決定
- **グローバル配線（3段以上離れたセル列間）**と
チャンネル配線（隣接するセル列間）を別々に決定



$$F = \overline{(ab + cd)(ef + gh)}$$

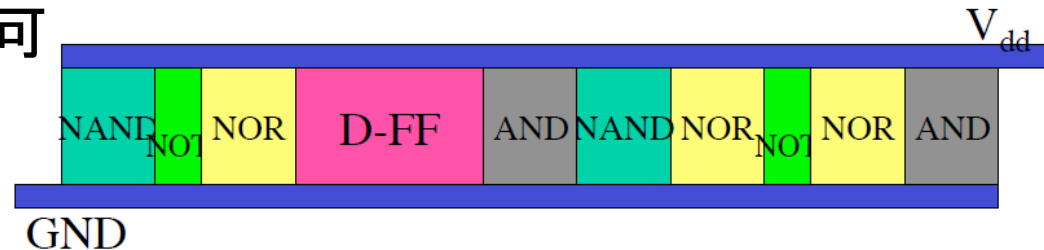


スタンダードセル・フルカスタム方式

32

□ スタンダードセル方式

- セルを列状に並べ，その列間に配線領域を設けて設計
- セルの高さ，電源線・接地線的位置は固定
- ゲートアレー方式より高い自由度
 - 全ての層のマスキパタンの設計・製造
 - 各セルのトランジスタサイズ：統一の必要無
 - 配線領域の幅：変更可



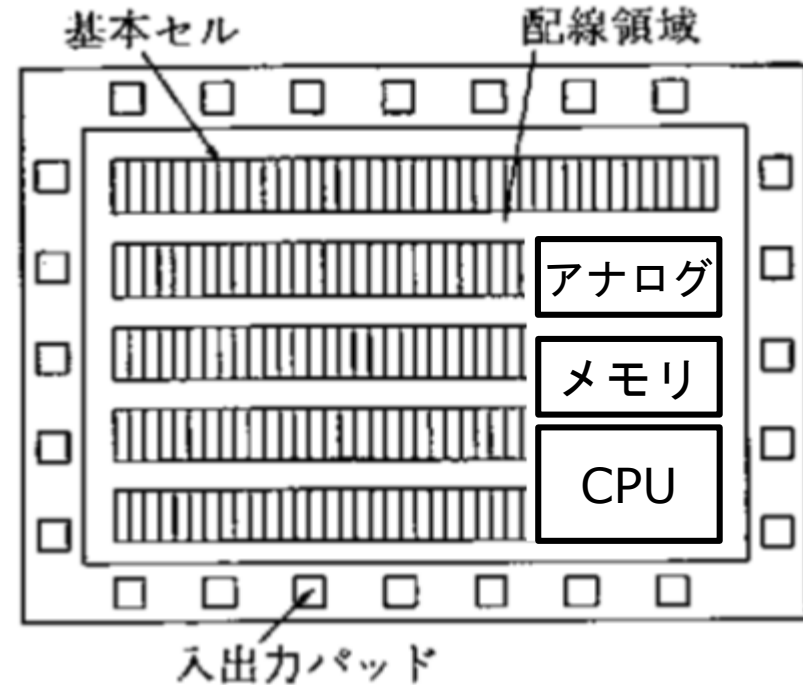
□ フルカスタム方式

- さらに自由度を高くした設計方式

エンベッドドアレー

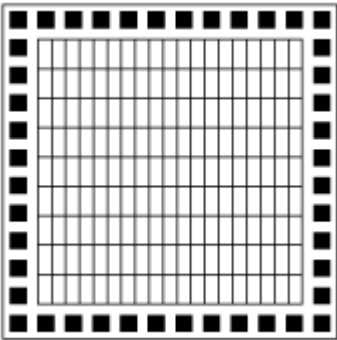
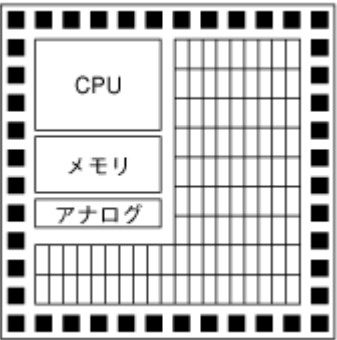
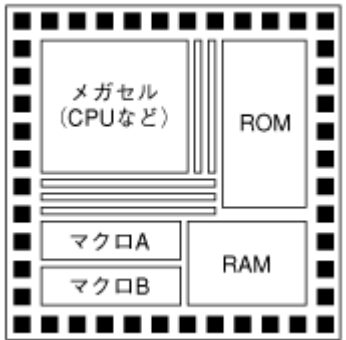
33

- ゲートアレーとスタンダードセル方式の中間的な方式
 - ハードマクロ（ハードIP）
が決定した時点でシリコンウェハを先行投入
 - ハードマクロ以外
 - ゲートアレー方式で設計
 - 配線（＝メタル層）のみ



設計方式

34

方式	ゲートアレー	エンベデッドアレー	セルベース	フルカスタム
回路イメージ				
製造パターン層	金属層	金属層	全ての層	全ての層
開発期間	小	小～中	大	より大
開発コスト	小	中	大	より大
搭載機能	中	中～大	中～大	中～大
製造コスト	小	中	大	より大
生産数量	中	中～大	大	より大
用途	産業	産業	民生	民生

- 論理ブロック(L)：基本論理ゲート
- スイッチ (S)：配線

どちらも製造済み
プログラマブル

