

CMOS LSIレイアウト

横から見ていたものを上から見る

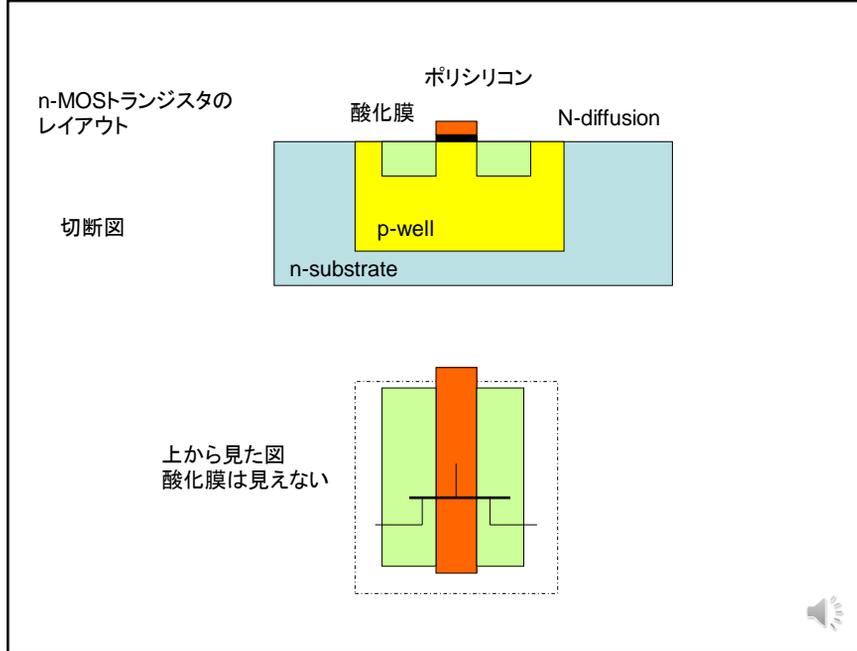
CMOSの構造を今までは断面図として理解していた

今回は上から見た図を理解し、実際にどのように半導体上に作られるかを理解する

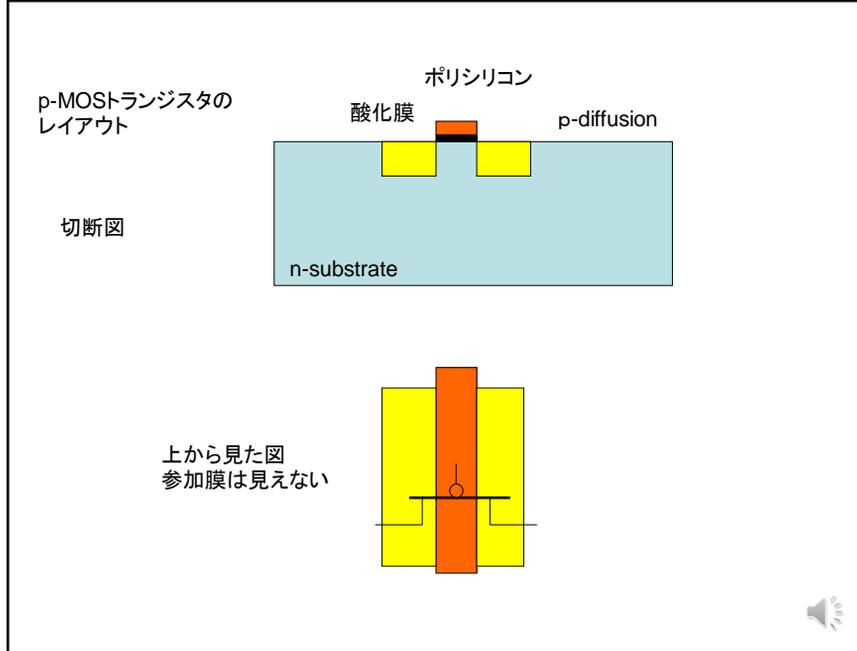
LSI設計の常識を学ぶ たくさん用語がでてくるけどびびっちゃダメ、本格的な紹介は別の授業でやるので概念を掴んで欲しい



今までは、CMOSの構造を断面図として理解していました。断面図はトランジスタの性質を説明する場合などに使われますが、実際の設計では上から見た図が重要です。正しくは、3次的に積み重なった構造のそれぞれが上から見たらどうなっているかが重要です。ICを設計する場合、それぞれの層が上から見たときどのような形状をしているかを示す設計図が必要になります。これをレイアウトと呼びます。それぞれの層のことをレイヤと呼びます。各レイヤのレイアウトができれば、この形(マスクパターン)にしたがって、不純物を拡散させたり、ビーム線を打ち込んだり、エッチングをしたりして半導体を作っていきます。レイアウト設計はIC設計の最終段階に相当し、4年のVLSI設計論で詳しく学びますが、今回はこの基礎として重要です。

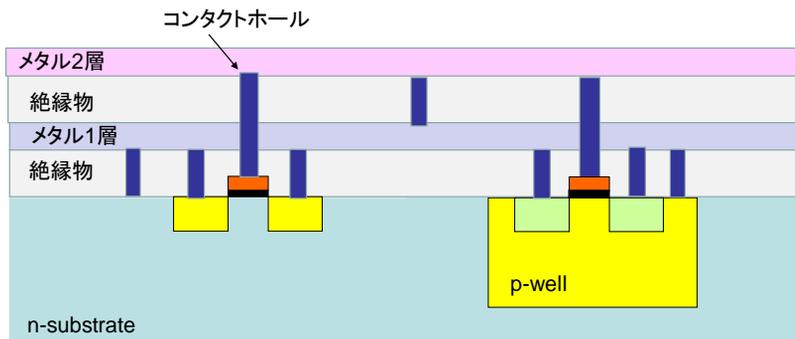


まずnMOSTランジスタを上から見てみましょう。CMOSではpMOSとnMOSが混在します。ここではpMOSのサブストレートであるn型のサブストレートを基本とします。nMOSを作るためにはこの内部に入れ子の形でp型のサブストレートを作る必要があります。そこで、一定の大きさのp型の領域を作ってやります。これをp-wellと呼びます。(wellは井戸です。)このp-wellの中にn型の拡散層(diffusion)を二つ作って、これをソース、ドレインとします。この拡散層の間のチャンネルの上に酸化膜を形成し、さらにその上に導体であるシリコン化合物(ポリシリコン)の層を作ってゲートとします。これを上から見た図を下方に示します。酸化膜はポリシリコンに遮られて見えませんし、さらにその下のチャンネルも見えません。拡散層をポリシリコンのゲートをぶち抜いているように見えます。これがトランジスタに相当します。ちなみにソース、ドレインは交換可能です。トランジスタのチャンネル幅は、微細加工技術が許す限り短く取るのが普通です。その方が、動作が高速になり、ON抵抗も小さくできるからです。このため、強力なトランジスタを作るためには、この図の縦方向を延ばして、ソース、ドレイン、チャンネルの長さを長くしてやります。ここではp-wellは点線で示します。

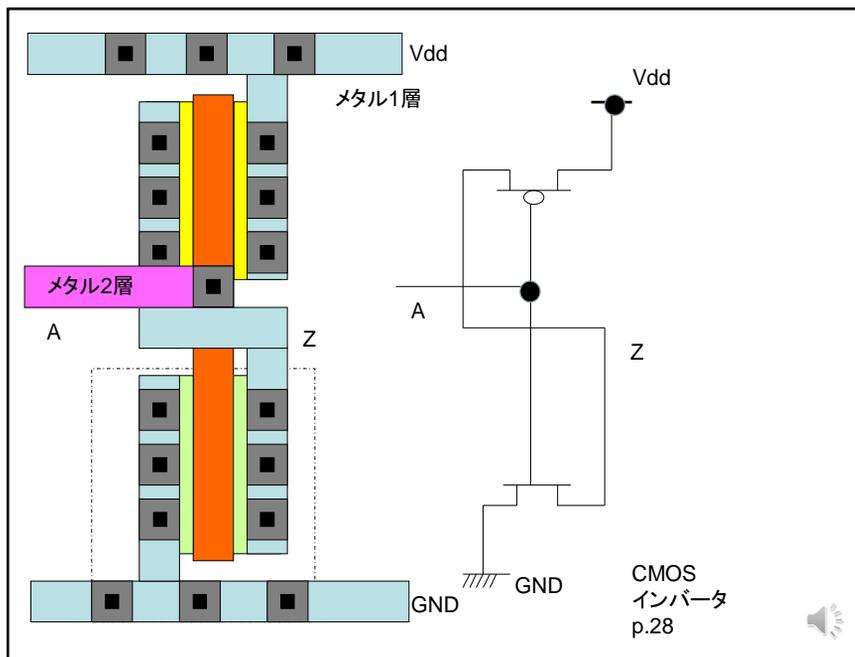


次にpMOSを示しましょう。今度はp型の拡散層をポリシリコンのゲートが貫いている形になります。もちろん、このポリシリコンのゲートの下には酸化膜があって、さらにその下では拡散層が切れているのです。上からの図では見えないですが、断面図と照らし合わせて理解してください。

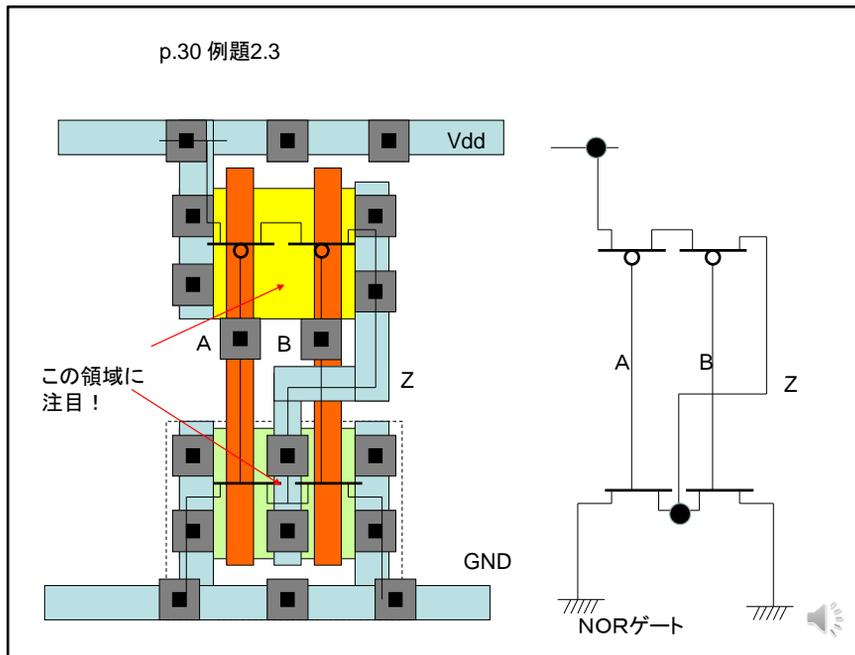
配線用メタル層とコンタクトホール(ビアホール)



トランジスタ同士を配線するためには、ICの上に配線用の金属層(メタル層)を用意します。この図に示すように絶縁物を介してメタル1層、メタル2層と重ねていきます。複雑で大規模な回路を搭載するためにはメタル層はたくさんあった方が有利です。ここでは2層しか書いていませんが、実際は12層くらいまで持っているものもあります。メタルの材料はアルミニウムが多く使われますが、抵抗を減らすために銅を使う場合もあります。この層と層の間を接続するために、絶縁物に穴を掘って導体を注入して、層間を繋いでやります。これをコンタクトホールまたはビア(Via)ホールと呼びます。

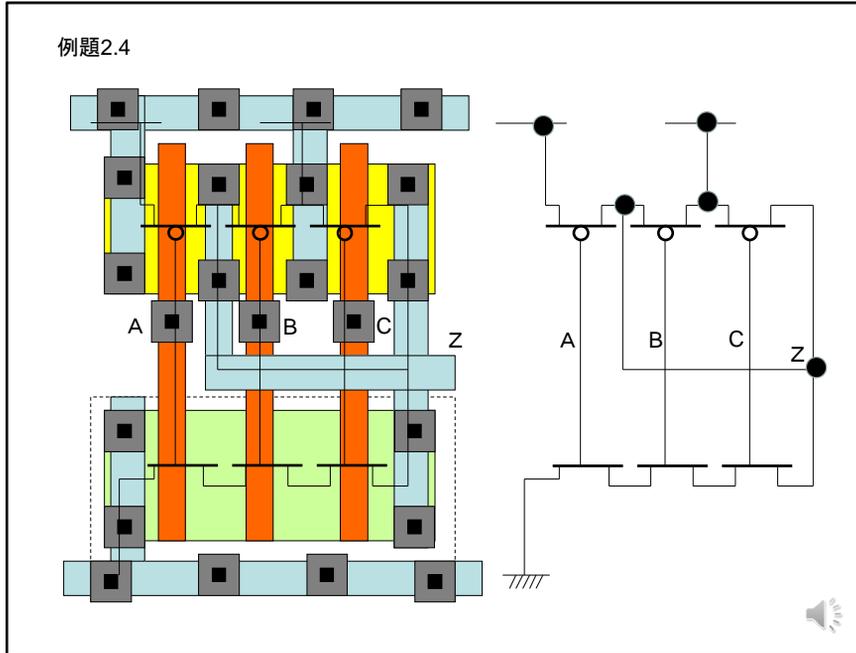


ではCMOSのインバータのレイアウトが具体的にどうなるかを見てみましょう。この図ではコンタクトホールを■で表します。まず注目したいのは、ポリシリコンのゲートがpMOS,nMOS双方のトランジスタを貫通している点です。CMOSではゲートを共有するペアのトランジスタを作りますので、このやり方は多くの場合うまく行きます。ポリシリコンは導体なのでメタル層を使わなくても配線の代わりに使えます。ここではメタル1層を水色、2層をピンク色で示します。最上部の横線はVDDを供給する線で最下部はGNDです。まずこれを片方の拡散層に繋がります。次にもう片方の拡散層を繋いで出力とします。この配線はメタル1層を使って行います。それぞれの端子と結合するためにコンタクトホールを使います。抵抗を減らすため、今回は3個のコンタクトホールを並列に使っています。下のGND線とのコンタクトホールでpWellをGNDにつなぎ、上のコンタクトホールでVddをサブストレートに繋いでいます。ゲートに入力を与える線はメタル1層と重ならないようにメタル2層を使います。層が違うメタル層は立体交差することができます。(逆に繋ごうと思ったらコンタクトホールが必要です。)



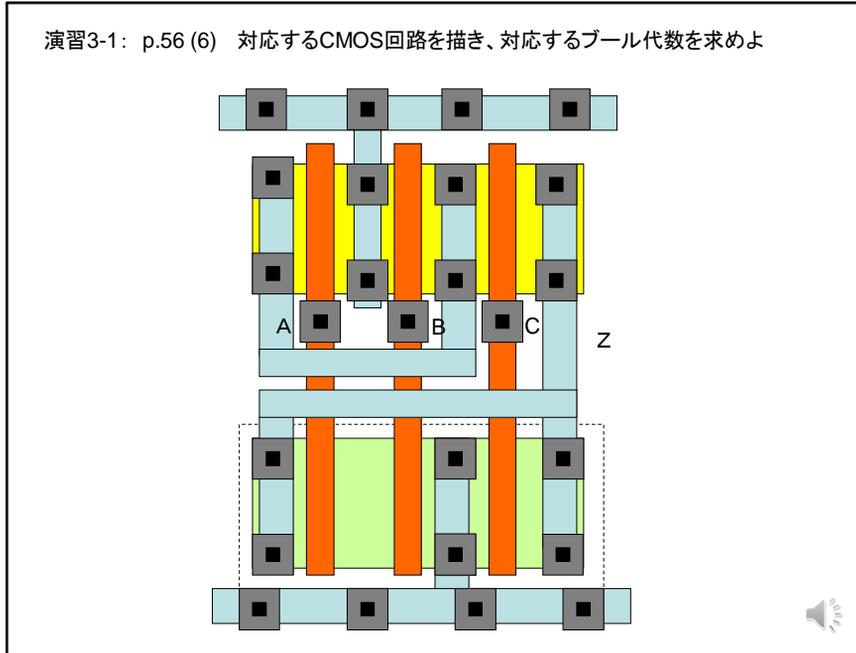
次はNORゲートのレイアウトを説明します。ここで注目したいのは、一つの拡散層が左右のトランジスタの両方の端子を兼ねている点です。p型拡散層の口形とn型拡散層の口形に2本のポリシリコンの棒が貫通しているように見えますが、この棒と棒の間の拡散層は左右二つのトランジスタで共通のソース(ドレイン)になっています。このため、わざわざ二つのトランジスタを接続する必要がなくなります。NORゲートではnMOSは並列、pMOSは直列に接続されるので、nMOS側は両方をGNDに落として中央から出力を取り出します。pMOSの方は共通領域をトランジスタの直列接続に用いており、ここで使われていない端子の片方をVdd、片方を出力に繋ぎます。ここではA,B,Zを引っ張り出すメタル2層は省略しています。

例題2.4

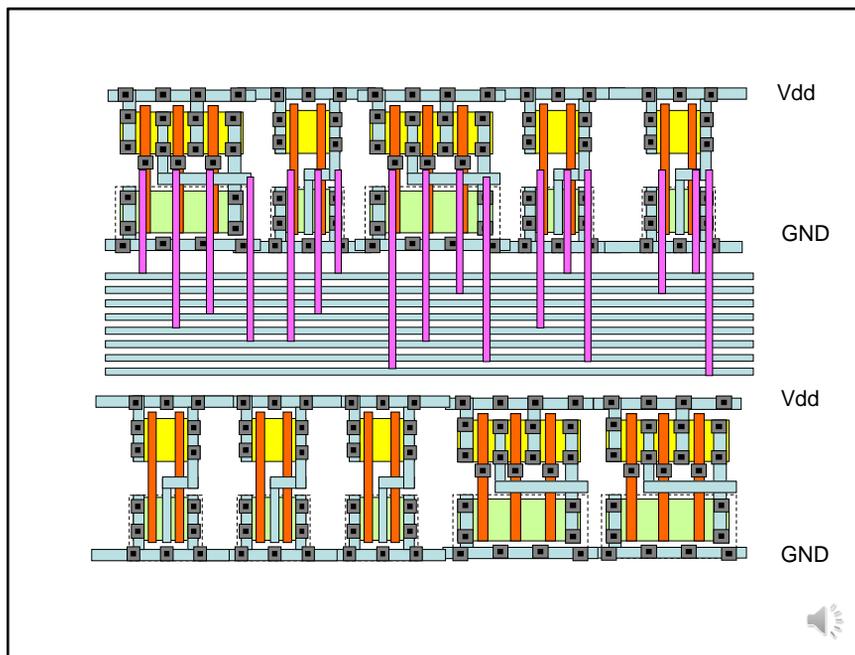


次は3入力NANDの例を示します。3入力なので、ポリシリコンのゲートの棒は3本になります。この場合nMOSが3つ直列に接続され、pMOSは並列になります。トランジスタの数が増えると拡散領域が大きくなります。レイアウトを見ると、物理的にはMOS-FETのドレイン、ソースの区別はないことがわかります。

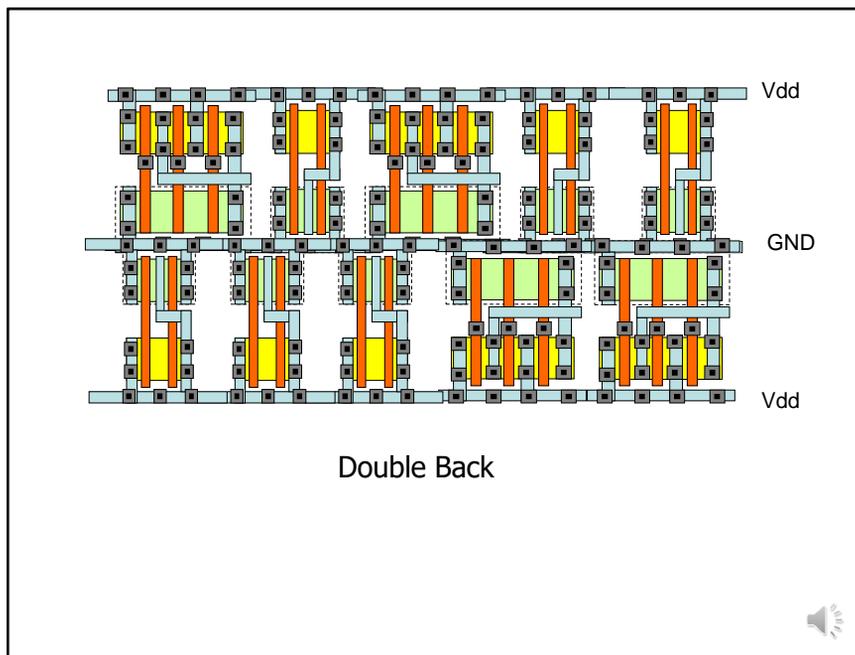
演習3-1: p.56 (6) 対応するCMOS回路を描き、対応するブール代数を求めよ



さて、今までの例を参考にこの配線に相当するCMOSの回路図とブール式を求めてみましょう。どのトランジスタが直列で、どれが並列接続かを見極めてください。



では、次にこのようにして作ったゲート同士をどのように接続するかを説明しましょう。それぞれのゲートは縦幅を揃えて作ります。このようにすれば、VddとGNDを共通にして横に並べて配置することができます。並べて行を作っておいてそれぞれの入出力から線を引っ張り出して配線します。この図はこのための古典的な方法です。ごちゃごちゃするので描いてないですが、下の行からも同様に線を引っ張り出します。このように縦幅を揃えて並べて行を作ることができるゲートのレイアウトを一式用意して、設計用のCAD(Computer Aided Design)を使って自動的に配置、配線を行う方法が現在の設計法の主流です。このようなゲートの一式をスタンダードセルと呼びます。



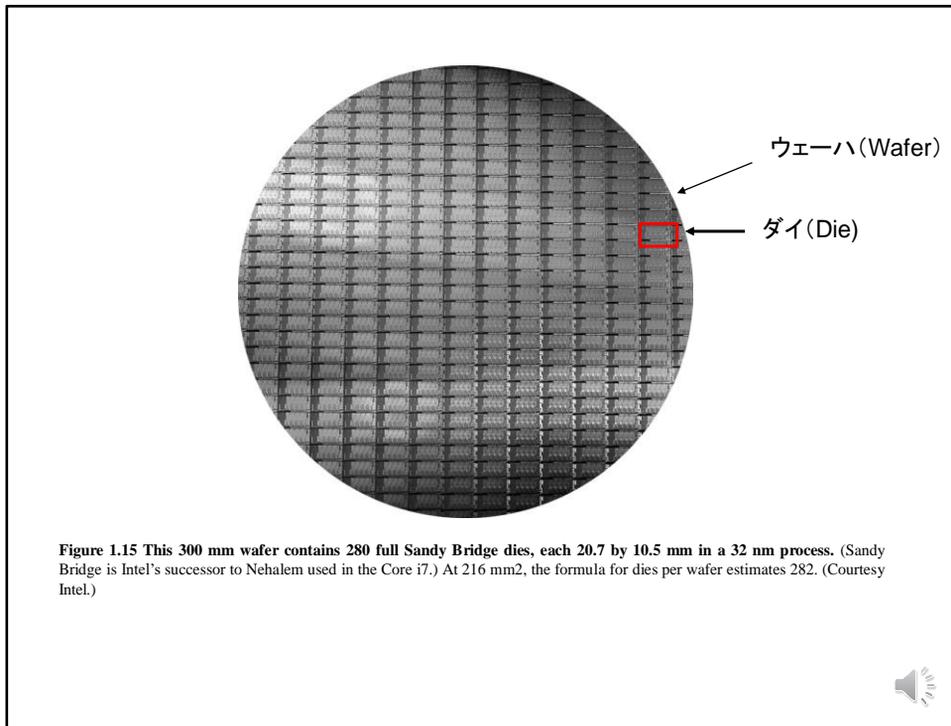
行を並べる際に、交互に上下をひっくり返せば、隣り合った行のGNDとVDDを共有することができます。このような配置法をDouble Backと呼びます。スタンダードセルでチップを作る場合は、まずセルの配置を決めます。これはCADが自動的に決めてくれます。

デジタルICのレイアウト

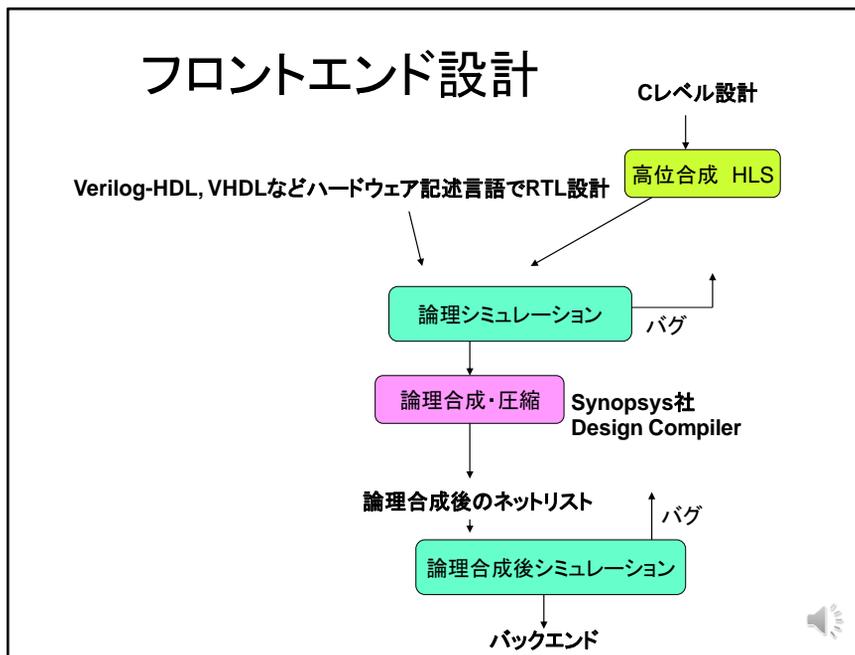
- フルカスタム
 - 全ゲートのサイズ、配置を自由に設定
 - 自動レイアウトツールが使えない
 - 高速動作が必要な回路などの特殊な設計法
- セルベースド
 - 高さの揃ったセルで様々な論理ゲートを一式用意する
 - 自動配置配線が可能
 - 現在のLSIの設計の主流
- ゲートアレイ
 - 拡散層、ポリシリコンのゲートは既に用意されている
 - 配線層のみレイアウト
 - 効率が悪いいため、最近はあまり使われない



今まで説明したスタンダードセルを用いた設計法をセルベースド設計と呼びます。CAD (Computer Aided Design)による自動配置配線を使うことができるから現在一般的に使われる方法です。レイアウトを作るには他にも方法があります。一つはフルカスタムと呼び、人手でトランジスタの形状を設定し、配線を行う方法です。高速動作を要求される回路、アナログ動作を行う回路に使います。一方、ゲートアレイは拡散層、ポリシリコンのゲートの構造を既に用意しておき、配線レイヤとコンタクトホールのみを設計する方法で、最も簡単にレイアウトができますが、効率が悪いいため、最近はあまり用いられません。後にこの授業で紹介し、現在最も良く用いられているFPGA(Field Programmable Gate Array)のゲートアレイはこの方法に由来しています。



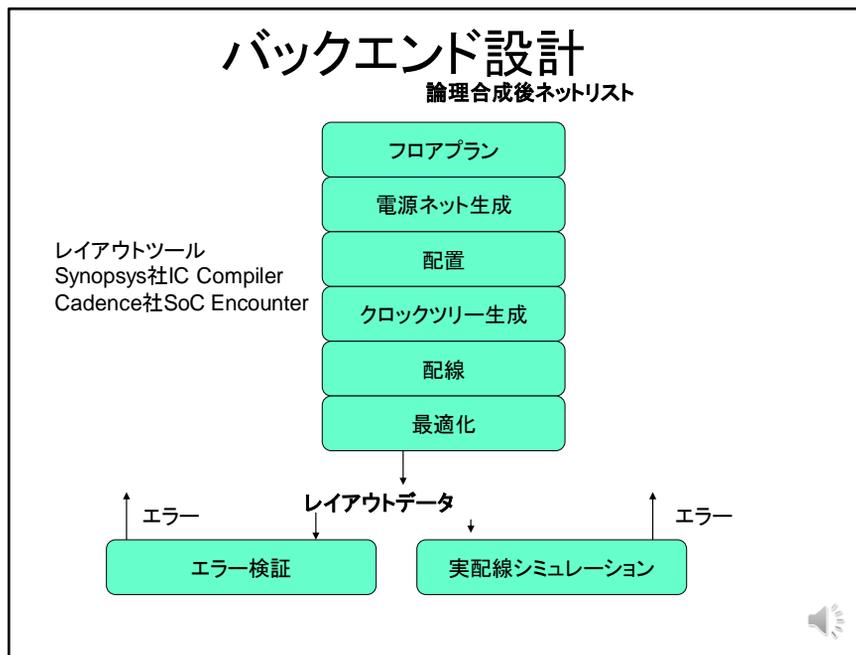
この図はIntelのCore i7(Sandy Bridge)のウェーハ写真です。直径30センチの円盤上に長方形のダイが並んでいます。これを切り離して、パッケージに組み込んで半導体チップができます。周辺部の模様が欠けているダイはもちろん使えません。ウェーハは半導体の製造工程上、どうしても30センチ程度の円盤になるので、ダイの面積が増えると、搭載できる個数が減ってしまうことがわかります。



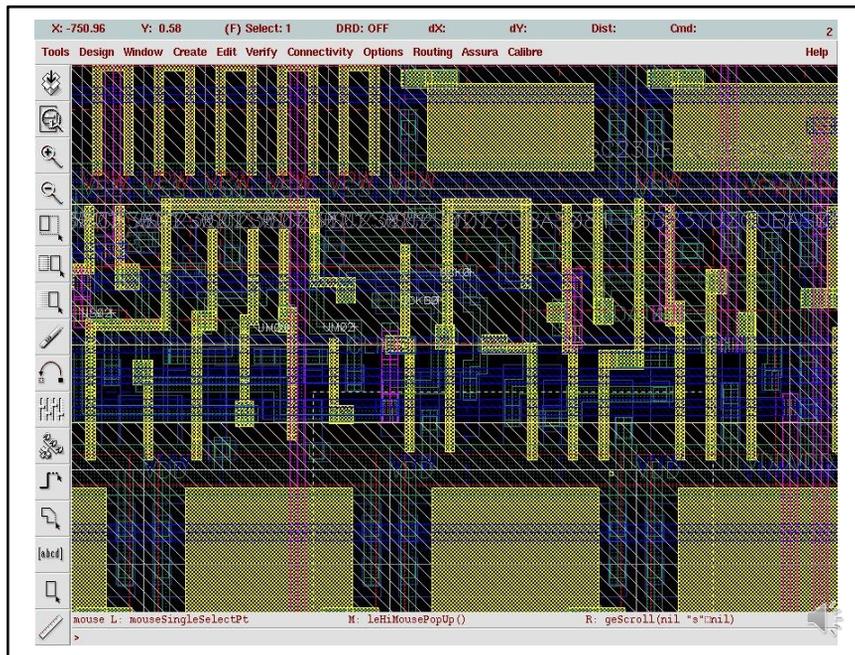
では、スタンダードセル方式の設計法を簡単に紹介します。この図の部分は秋学期の計算機構成同演習の時間に紹介しますので、その時に少し思い出していただけると嬉しいです。現在、デジタル回路を設計する場合は、通常、ハードウェア記述言語HDLと呼ばれる方法を使って記述します。これはちょうどコンピュータのプログラムを書くような感じでレジスタ間のデータの移動とその間に行われる仕事を書いていくのです。最近ではC言語で記述したハードウェアの動作を、HDLに変換するツールも使われており、この技術を高位合成 (High Level Synthesis: HLS) と呼びます。HDLの記述は、論理シミュレーション、すなわち設計とおりに回路が動いているかどうか確認する模擬動作を行って動作を確認します。これでうまく動くことが確認できたら、自動論理合成、圧縮を行います。この作業で、言語での記述は、ゲート同士の接続図に変換されます。これをネットリストと呼びます。変換後のネットリストで再び論理シミュレーションを行って、設計者の意図通りに動くかどうか確認します。この段階を設計のフロントエンドと呼びます。HLS、論理シミュレーション、論理合成、圧縮はすべてCAD (Computer Aided Design) を使います。設計と論理シミュレーションの検証で頭を使えば、論理合成、圧縮等は自動的にやってくれます (実はそんなに甘くはないのですが)。

バックエンド設計

論理合成後ネットリスト

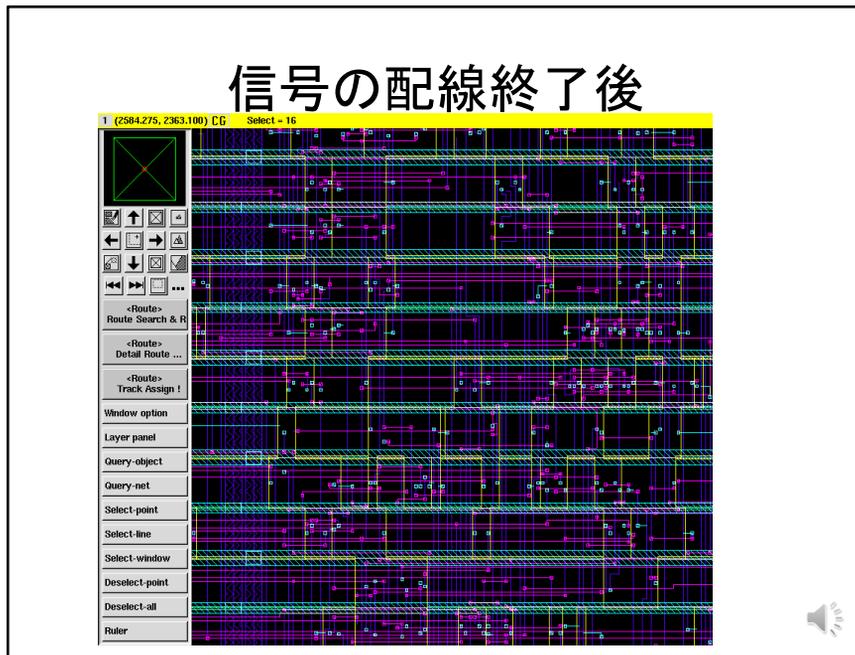


論理合成後のネットリストから、まず、ざっくりとした配置を決めてやります。これをフロアプランと呼びます。次に電源配線用の段取りをします。それから配置を行い、配置場所に応じてクロックが同じ時刻に届くように分配します。これをクロックツリーと呼びます。さらに配線して最適化を行います。最後に実際の配置配線に基づくシミュレーションと、きちんとルールとおりにレイアウトができているかどうかを確かめるためのエラー検証を行います。これらの作業も全てCADツールが自動的に行ってくれます。論理合成後のネットリストからレイアウトデータを作る作業をバックエンド設計と呼びます。これは4年生のVLSI設計論で詳しく紹介します。



実際のスタンダードセルを使った配置後のレイアウトはこんな感じになります。これはポリシリコンの層のみを示した例です。ゲートが並んでいる様子がわかります。

信号の配線終了後



次にCADを使ってやはり自動的に配線を行います。信号の配線を行った後の様子を
示します。線がコンタクトホールを介して配線されている様子が分かります。



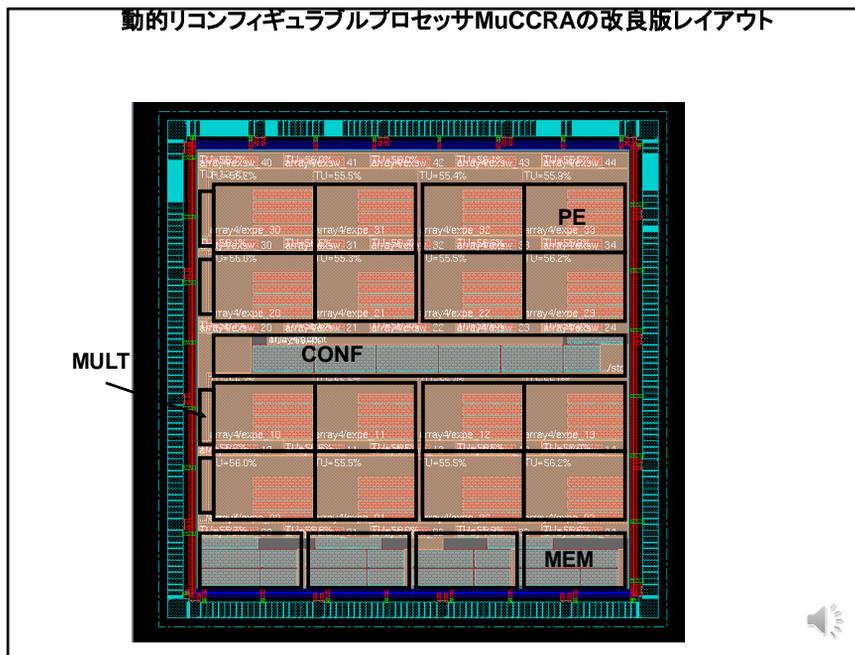
電源等を全て配線した後の図です。配線層が多いので大変ごちゃごちゃします。このような配置、配線の設計工程はCADが自動的にやってくれます。

IPベース設計

- IP(Intellectual Property)=既に設計、動作検証済みの設計資産を指す
 - ハードIP(ハードマクロ):レイアウトが既に設計済みのモジュール
 - ソフトIP(ソフトマクロ):ハードウェア記述言語、ネットリストの形で供給される設計資産、レイアウトはそれぞれのチップで行う
 - メモリ、入出力モジュール(I/Oモジュール)、クロック制御モジュール(PLL(Phase Locked Loop:位相同期回路))など
 - 大きいものにはコンピュータのCPU(中央処理装置)などもIPになっている 

ICの中でも搭載ゲート数の大きい大規模集積回路(Very Large Integrated Circuit: VLSI)では、全てスタンダードセルのゲートで作るわけではないです。メモリ回路(これは後の方で紹介します)、入出力モジュール、クロック制御モジュール(PLL: Phase Locked Loop:位相同期回路)などは、既に出来上がった設計資産をそのまま使います。このような設計資産をIP(Intellectual Property)と呼びます。メモリは大量のデータを記憶するため、PLLはクロックの位相調整、周波数調整を行うため、入出力モジュールは外部の仕様に合わせるために、それぞれ特殊な回路が必要になります。これを設計者が一から設計するのは大変なので、既に出来上がった設計資産を利用します。(IPは多くの場合有料です。)IPには、既にレイアウトが出来上がっているハードIPと、ハードウェア記述言語(計算機構成で習います)やゲート間の接続を示すネットリストの形で供給されるソフトIPがあります。ハードIPはレイアウトが最適化されていますが、チップが違えば違ったものを使わなければならないです。ソフトIPは、様々なチップで利用可能ですが、CADによる自動配置配線が必要です。大きいIPとしてはコンピュータのCPU(中央処理装置)などがあります。IPベース設計は設計資産の再利用が可能で、設計工数が節約できる点で大規模なIC設計は欠くことができない方式となっています。

動的リコンフィギュラブルプロセッサMuCCRAの改良版レイアウト



このレイアウトはメモリ用のIPを使った例です。中央部、下の方に灰色に見える□形がメモリのIPです。

システムLSI System-on-a Chip (SoC)

- LSI上にCPU、メモリ、PLL、入出力モジュール、専用目的のデジタル回路を搭載してシステムをまるごと搭載
- 携帯電話、スマートフォン、デジタルビデオ、高画像TV、自動車両制御、情報家電など
- ASIC(Application Specific IC): 応用に特化したICの一種
- かつては日本の半導体産業を支える製品として期待されていた



IPベース設計を大規模に用い、一つのチップ上に、CPU、メモリ、PLL、入出力モジュール、専用目的のデジタル回路を搭載することで、システムをまるごと乗っけることができます。このようなチップをシステムLSI、あるいはSoC(System on a Chip)と呼びます。ここで挙げた様々な製品で用いられます。目的に特化したICであることからASICとも呼ばれます。かつて、日本の半導体産業は様々なシステムLSIを自社生産していました。しかし、最近では生産拠点は海外に移り、設計のみを行う場合が多くなりました。日本の半導体業界の状況については最後のFPGAの時間でまとめて紹介する機会があると思います。

半導体のスケーリング則

Dennard Scaling

- 2005年くらいまで(90nm,65nmくらいまで)
 - プロセスサイズ(technology size):プロセス技術が許す最小加工幅が $1/k$ に
 - 集積度は k の2乗
 - スピードは k 倍
 - 電圧は $1/k$ →
 - 電力が k の2乗分の1(静電容量が減るので密度当たり一定)
 - 3年でプロセスサイズが70%に減っていく
- 1.5年で搭載ゲート数が1.5倍: Mooreの法則
- 1.2→1.0→0.8→0.65→0.35→0.25→0.18→0.13
→0.09(90nm)

では最後に半導体を理解する上で重要なスケーリング則を紹介しましょう。MOS FETは、チャンネルの幅が小さければ小さいほど性能が良くなり、サイズも小さくなることをお話しました。サイズを小さくするためには半導体を作る技術(プロセス技術)が許す最小加工幅を小さくする必要があります。この最小加工幅のことをプロセスサイズ(Technology Size, Technology Nodeと呼ぶ人も居る)と呼びます。今、プロセスサイズが $1/k$ になれば、動作速度はチャンネルが短くなるため k 倍になり、集積度は縦横で効いて来るので k の2乗で大きくなります。さらに電圧を $1/k$ にできる(あるいはしなければならぬ)ため、電力が k の2乗分の1になります。この良いことづくめの効果を半導体のスケーリング則(IBMのDennardさんが提唱したのでDennard Scalingと呼ぶこともある)と呼びます。2005年くらいまで、半導体のプロセス技術は、この効果に押される形でどんどん進歩し、3年でプロセスサイズは70%になりました。1990年頃には $0.8\mu\text{m}$ であったプロセスサイズはどんどん小さくなり、2005年には $0.09\mu\text{m}$ になりました。ここで単位が切り替えられて90nmと呼ばれるようになりました。

ムーアの法則



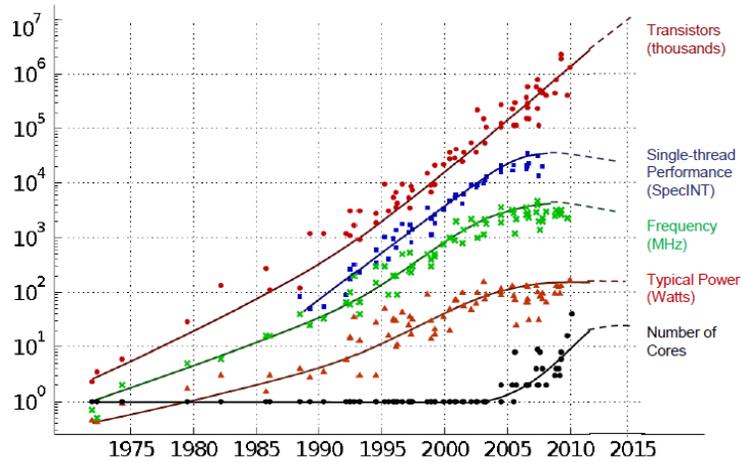
ゴードン・ムーア Intelの創業者の一人

一つのICに搭載可能なトランジスタの数は18か月で倍になる

一時期はプロセッサの性能にも使われたが、そちらは既に成り立たなくなっている。

<https://history.computer.org/pioneers/moore.html> 

35 YEARS OF MICROPROCESSOR TREND DATA



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten 
Dotted line extrapolations by C. Moore

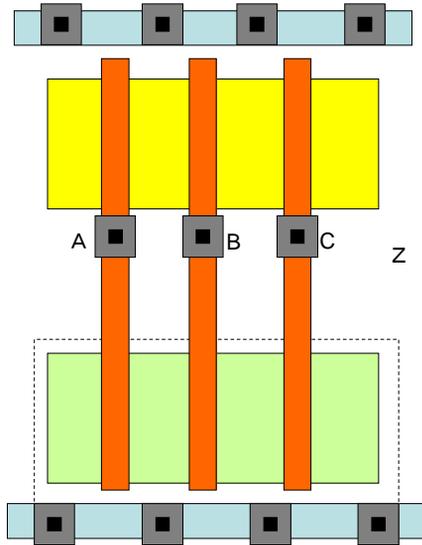
スケーリング則の崩壊

- 90nm,65nm前後からの傾向
- 配線遅延の増大:スピードが向上しなくなる
→ プロセッサのクロック向上も限界に、、
- 電圧の限界:0.8V以下には下げるのが困難
- 漏れ電流の増大:電力が下がらなくなる
- 3年で70%のペースが維持できなくなる。5、6年掛かる
- しかし、集積度は相変わらず増大
- 現在14nmが最先端、10nmが登場、7nmが準備中
- 最先端プロセスは非常に高価になる
- NRE(Non-Recurrent Engineering)コスト:つまり最初の1個を作るまでのコストが増大



しかし、65nm以降はではこのスケーリング則がうまく働くなってきています。今でもプロセスサイズが小さくなればトランジスタのスピードは上がるのですが、配線抵抗と浮遊容量が大きくなるので、配線遅延が大きくなり、新しいプロセスではうまく配線しないと思ったようにはスピードが上がらなくなりました。また、電源電圧も0.8Vより小さくするのは難しくなり、(これより小さくすると動作速度が急激に落ちてしまいます)さらに電源は、サイズを小さくしたことで漏れ電流が大きくなって、これも思うようには小さくならなくなりました。さらに、微細加工技術も限界達しつつあり、さらにプロセスを進めることが難しくなっています。今でも新しいプロセスの開発は進めてられており、現在は14nmが最先端で10nmが登場し、7nmが準備中です。しかし、このような最先端プロセスは最初の1個を作るまでのコスト(これをNREコストと呼びます)が、極めて高価になり、マイクロプロセッサCPUやFPGA(後にやります)など、付加価値の高い製品にだけ使われるようになりました。現在、システムLSIを最先端プロセスで作るためには、多数の出荷数が望めなければできません。この点をなんとかするためにはチップに柔軟性を持たせる技術、小さなチップ同士を組み合わせる技術が重要になります。これは我々の研究課題で、後に説明する機会もあると思います。

演習3-2: $(A+B) \cdot C$ のレイアウトを描け



それでは、今回の問題はこの論理式を実現するレイアウトを描いてください。

今日のポイント

トランジスタは上から見ると、n型(p型)の拡散層の四角形にポリシリコンの棒が刺さっている形に見えるが、下に酸化膜やチャネルがあるので注意

ゲートのレイアウト

上半分にpMOS、下半分にnMOSを配置して上下に電源とグランドを引く
ポリシリコンの棒は伸ばしてnMOS,pMOSの共用ゲートとする
隣り合ったトランジスタの拡散層はくっつけて両方で共用する
立体方向の配線にはコンタクトホールを打つ

スタンダードセルを使ったセルベース設計法は
基本的にはCAD任せ
IPベース設計は最近のVLSI設計のトレンド
スケーリング則とその崩壊は概念を理解して
話に付いていけるようにしましょう



今日のポイントをインフォ丸が示します。今回、若干インフォ丸も、もて余し気味だ！