

# 回路設計基礎演習

コンピュータ設計の基礎知識

第3章 演習ノート

清水 尚彦

# (MOS)LSI開発の歴史

- 半導体プロセスの開発
  - 写真印刷技術を応用したマスクパターンと光によるフォトエッチング
- 1970年代
  - フォトマスクを手で作成
    - トランジスタ数に限界がある(数千トランジスタが限度)
    - 1971年 Intel i4004(世界初のマイクロプロセッサ) 2300Tr
- 1980年代
  - レイアウトを自動設計するCAD(Computer Aided Design)もしくはEDA(Electronic Design Automation)
  - 配置・配線の自動化が進む
    - 1983年 Intel iAPX386(32bit マイクロプロセッサ) 28万Tr
  - 回路設計はゲートもしくはトランジスタレベルで行う
- 1990年代
  - EDAツールが、ハードウェア記述言語(HDL)をサポートし論理合成により高位言語で設計
  - Register Transfer Level(RTL)設計が中心
- 2000年代以降
  - 動作合成が模索されている

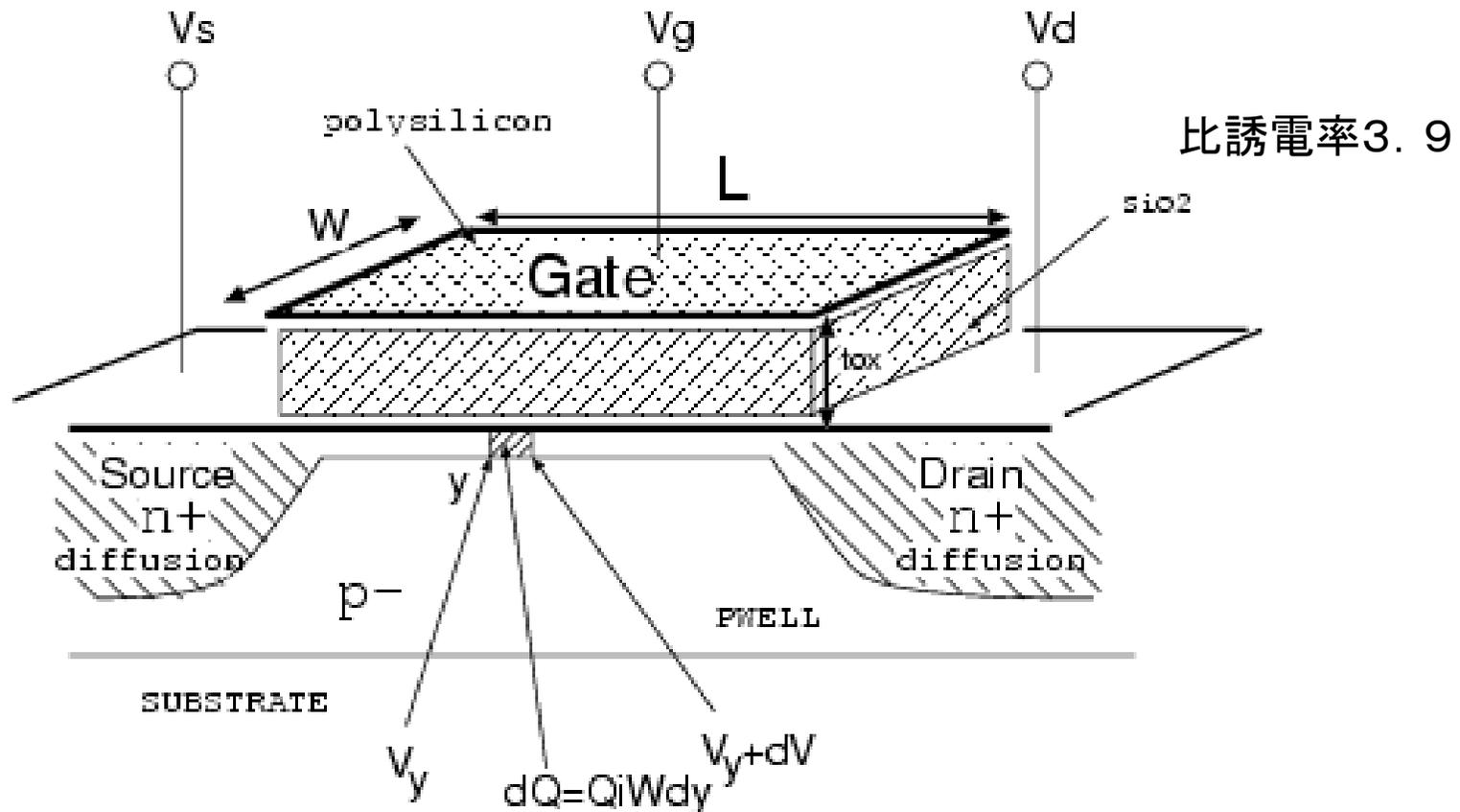
# 本演習で使うツール

- ツールセットをインストール済みのCDROMを配布(ブート可能なLinux CDROM)
- 回路シミュレーション
  - SPICE 3F5
- LSIレイアウト
  - MAGIC 7.1
  - テキストエディタ
    - vi
    - gedit, xedit, ...

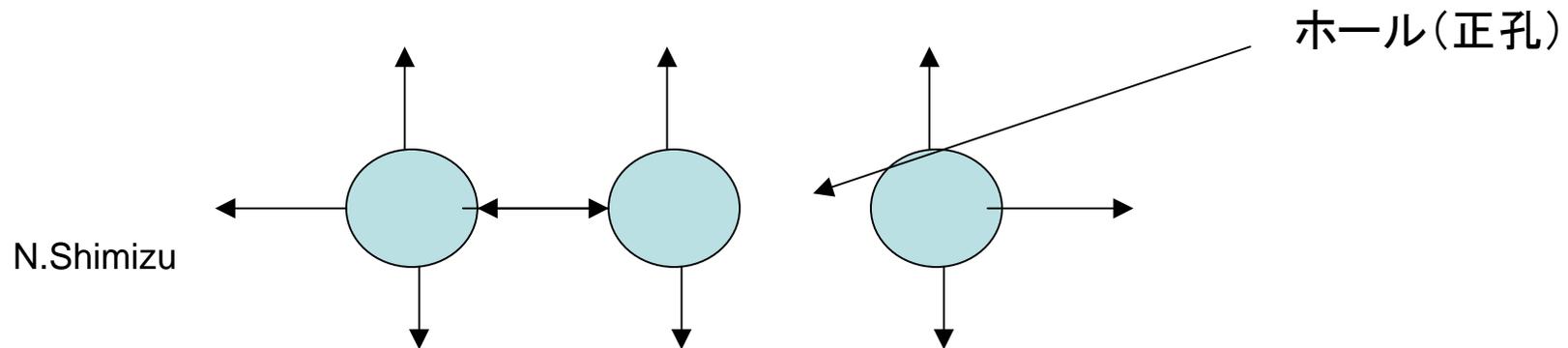
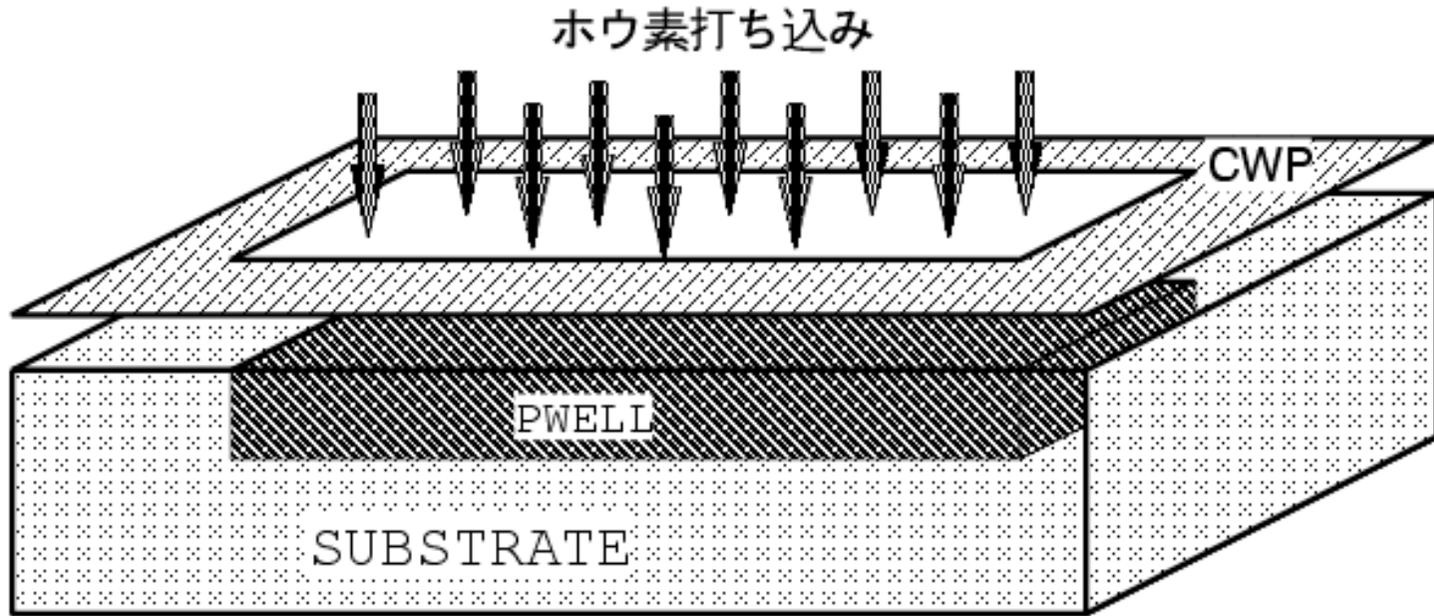
$$Q=CV$$

# テキスト3章 (P63)

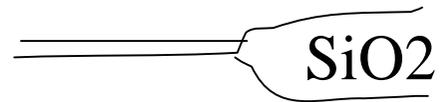
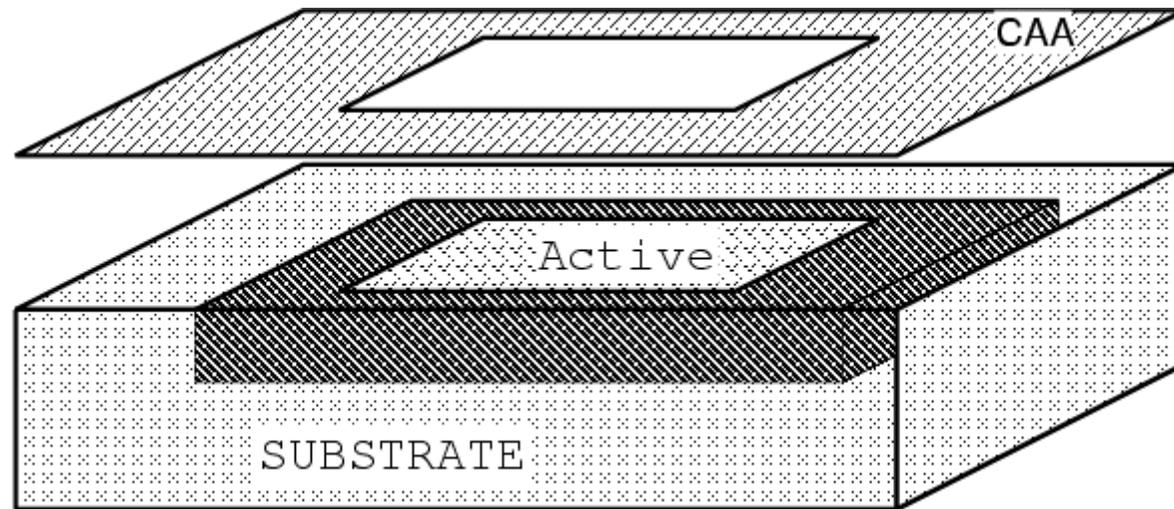
- MOSTランジスタの動作原理



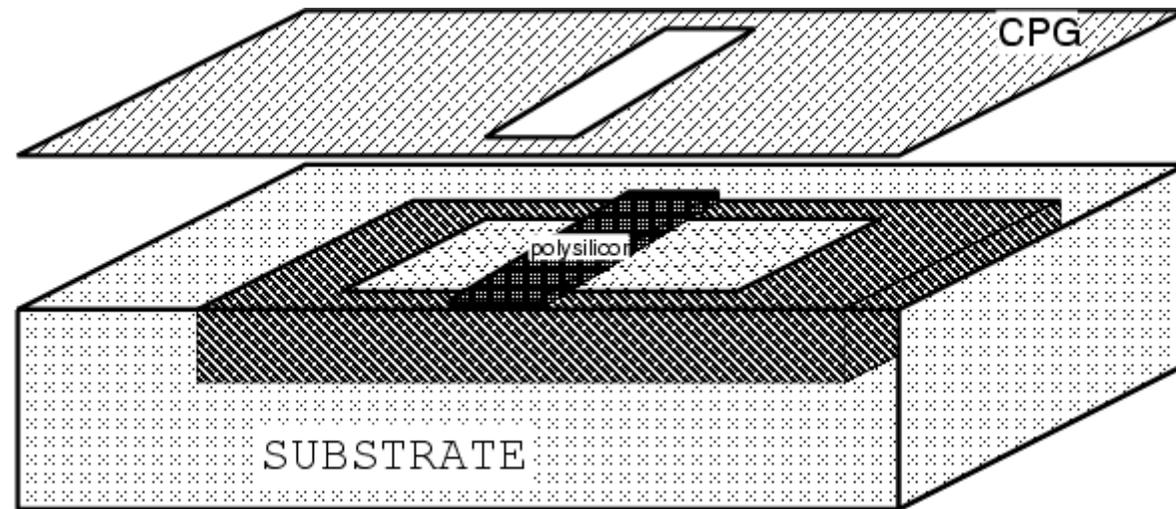
# PWELLの作成



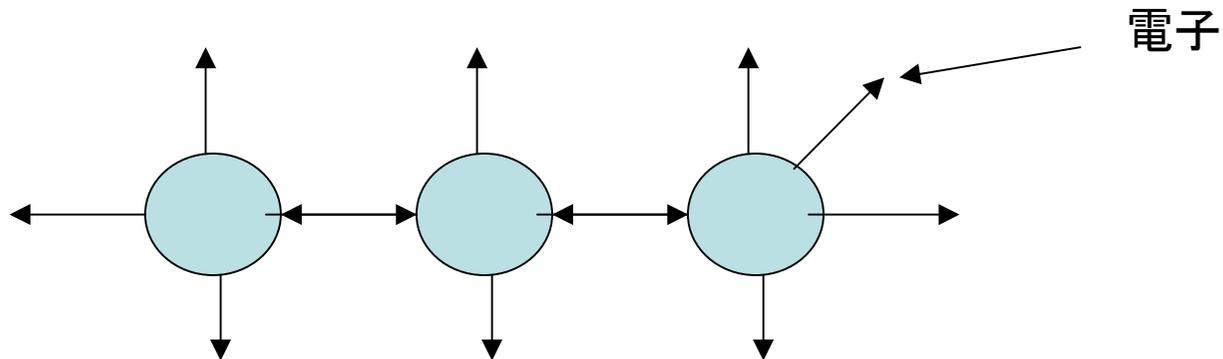
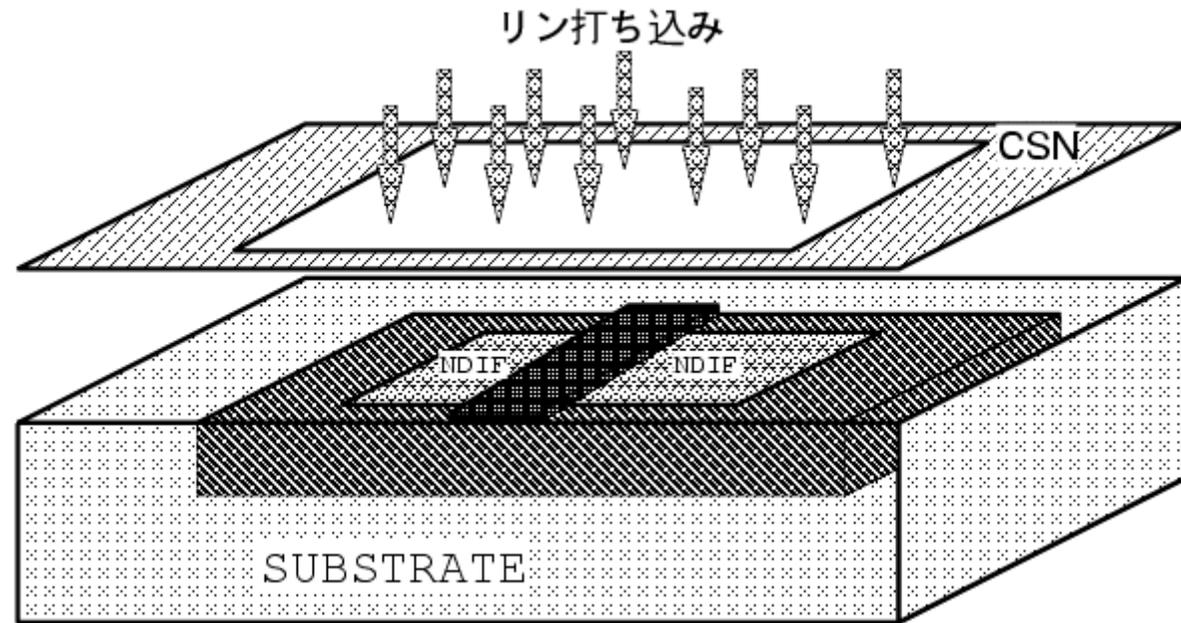
# アクティブ領域の生成



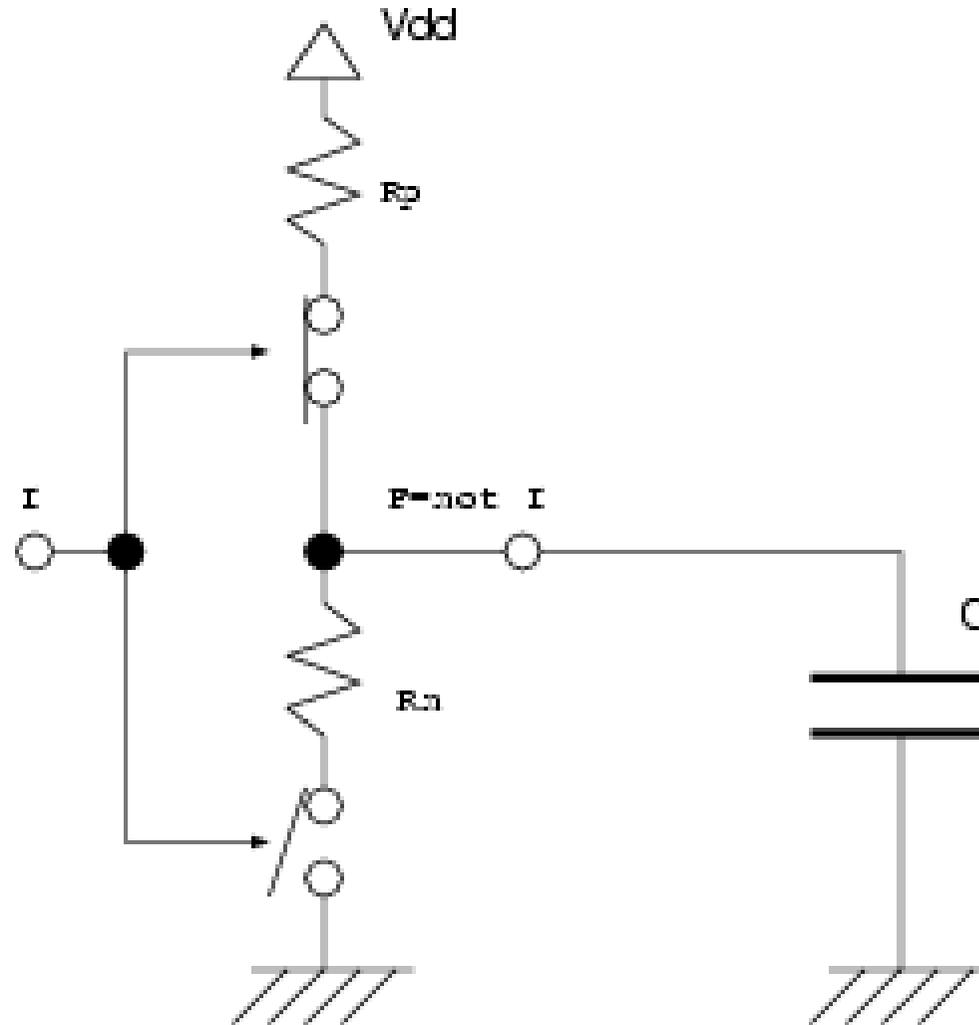
# ゲート電極のポリシリコンを乗せる



# ソース・ドレインのイオン打ち込み



# ゲート電圧によって開閉するスイッチ



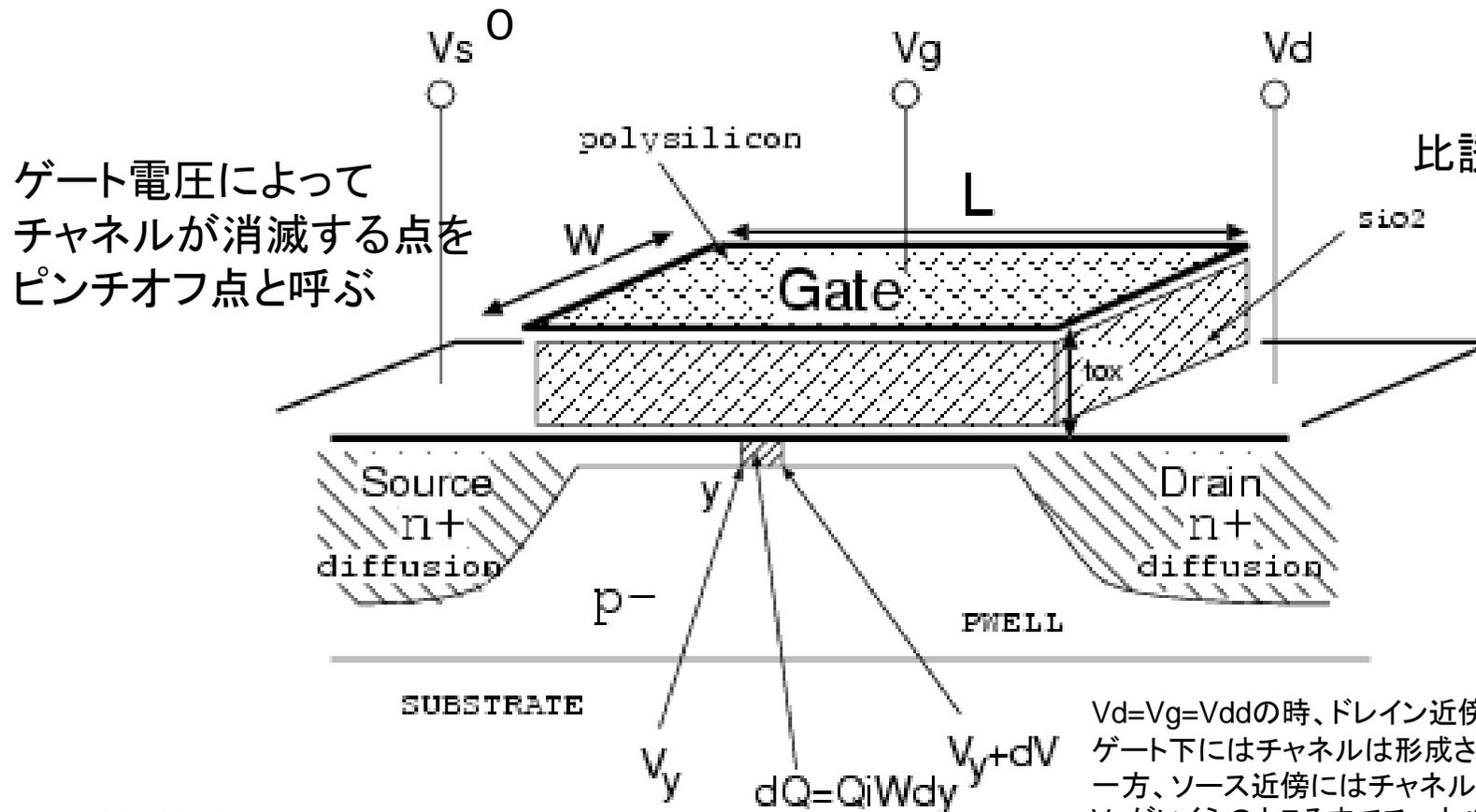
# MOSトランジスタのドレイン・ソース間 電流を導出せよ

- 3. 7式を導出せよ
  - 3. 6式を両辺積分して求める
  - ソース電圧は0V、ドレイン電圧は $V_{ds}$
  - ゲート電圧 $V_g$ とする
- 3. 11式の導出と式の説明をせよ
- テキストP86 演習3. 2を行え

$$Q=CV$$

# テキスト3章 (P63)

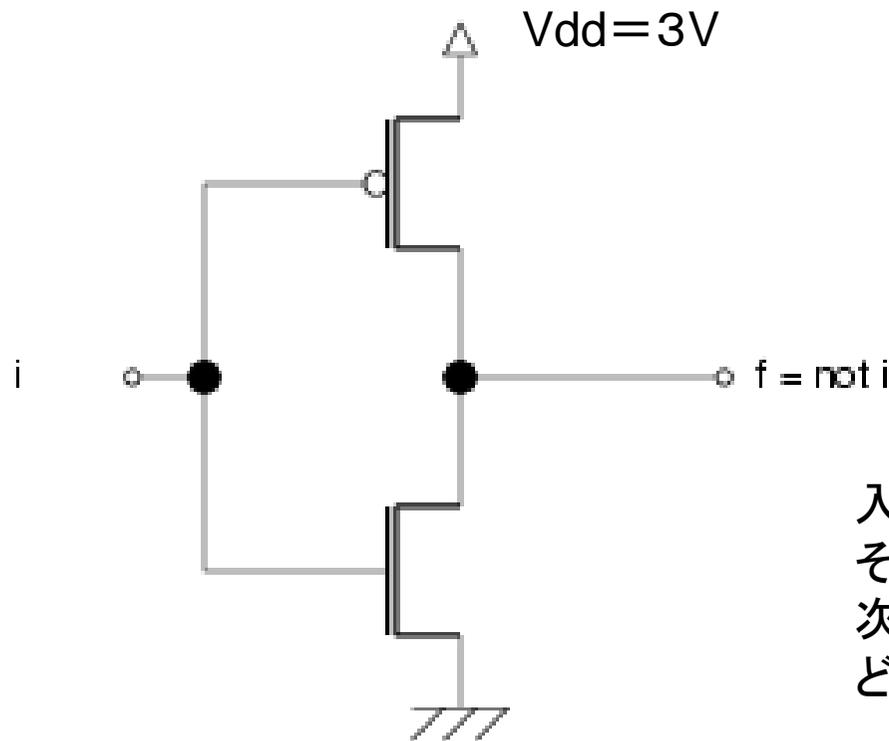
## • MOSTランジスタの動作原理



V<sub>d</sub>=V<sub>g</sub>=V<sub>dd</sub>の時、ドレイン近傍のゲート下にはチャンネルは形成されない。一方、ソース近傍にはチャンネルは形成されている。V<sub>y</sub>がいくらのところまでチャンネルが形成されているか？

# NMOS回路

- NMOS回路の入出力伝達特性を求めよ。



PMOSトランジスタの代わりに  
抵抗値 $5k\Omega$ の抵抗を入れて考えること。  
トランジスタの基本パラメータは  
P.86演習3. 2の $1\mu\text{m}$ NMOSを用いる。

入力 $i$ は $0\sim 3\text{V}$ とする。

入力を3つの領域に分けて  
それぞれの特徴式を求める  
次の入力状態でトランジスタのチャネルは  
どのようになっているかを考えること。

入力電圧が $0\text{V}$ 付近

入力電圧が $V_{th}$ となったとき

飽和・非飽和の切り替えポイント

# 演習解説

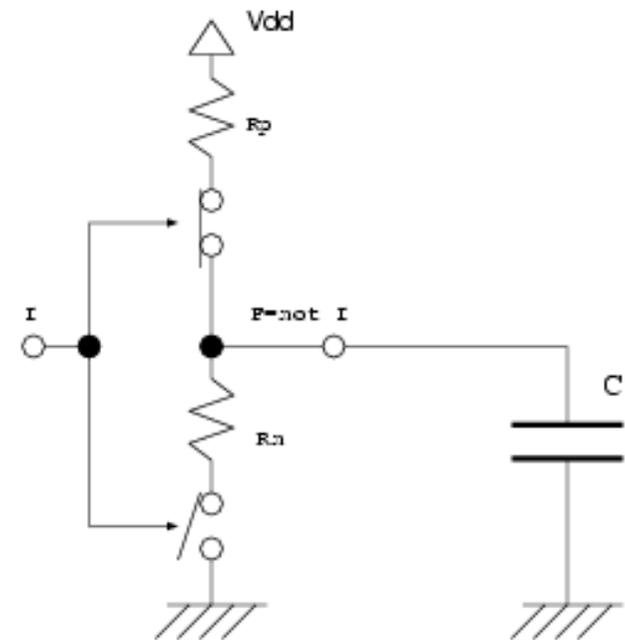
- トランジスタの動作領域を3つに分けて考える
  - 入力電圧が $0V \sim V_{th}$ まで(カットオフ)
  - 入力電圧が $V_{th}$ 以上、どこかまで(飽和)
    - 飽和条件は $V_g - V_d < V_{th}$ より、入力電圧が $V_{th}$ ちょうどの際には飽和となる。
    - 出力電圧 $V_f = V_{dd} - R \times I_{ds}$
  - どこかから入力電圧 $V_{dd}$ まで(非飽和)
    - 入力電圧が $V_{dd}$ となった時を考える。このとき、 $V_g = V_{dd}$ 、 $V_d < V_{dd}$
    - 飽和条件から非飽和への切り替えポイントは $V_g - V_d = V_{th}$
    - 出力電圧 $V_f = V_{dd} - R \times I_{ds}$

# MOS回路の抵抗近似と過渡特性

- $t=0$ で入力信号が $V_{dd}$ になるものとする。それ以前は $0V$ 、負荷容量 $C$ は $5pF$ とする。
  - NMOSTランジスタを式(3.11)で抵抗近似をしたときの過渡特性を求めよ。
  - トランジスタの $I_{ds}$ の式より、過渡特性を導出せよ

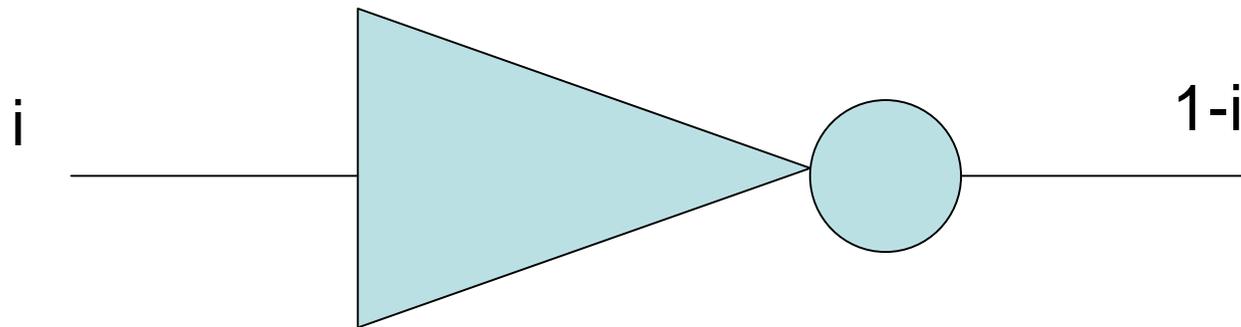
## 時間に対する出力電圧の式とグラフ

N. Srinivasan 図3.16に類似するグラフを求める



# inverter (反転) 回路

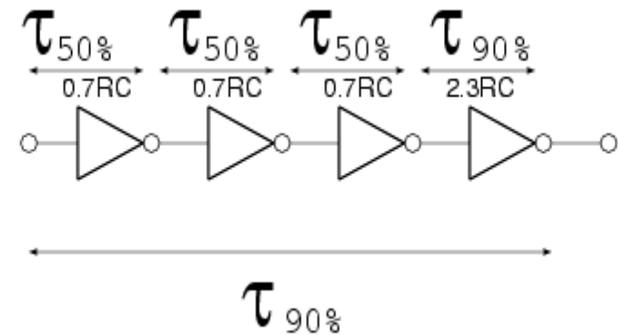
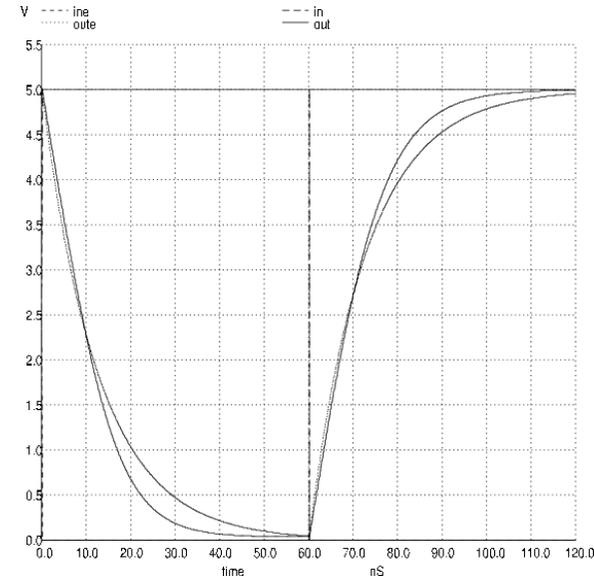
- 入力信号  $i$  に対し、 $(1-i)$  を出力する



回路の遅延時間は1段のゲートの遅延時間だけでは決まらない。  
複数段の遅延評価システムが必要とされる。

# 複数段のゲートからなるシステム

回路遅延時間は最終段だけ90%遅延を用い、その他は50%遅延を累積する。  
回路シミュレーション性能は大規模回路に追いつかない  
そこで、基本的な算術演算で回路遅延を計算する近似法がとられる



# 高速デジタル回路

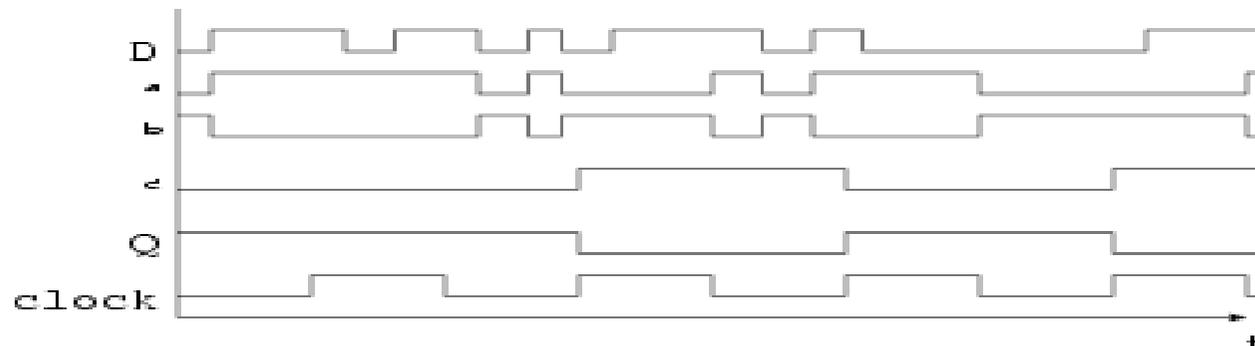
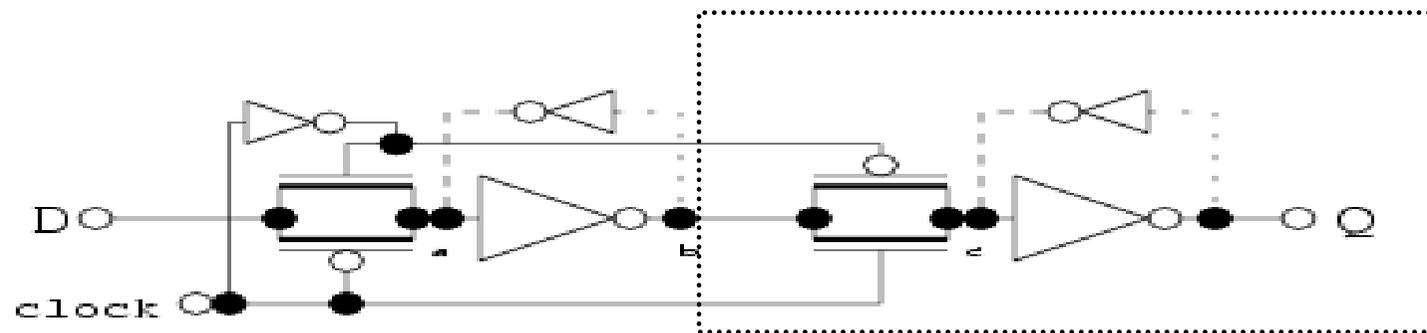
- 多くの高速システム設計において、回路設計者にはゲート段数の制約が課せられる。
  - ゲート段数とは、フリップフロップからフリップフロップの回路経路に配置する論理ゲートの個数
- 高速システムでは、制約ゲート段数は10～15段程度になっている(90年代まで)
- 設計制約が10段であったときに、テキスト図3. 16の論理ゲートにおいて、抵抗近似の誤差は何%か？
- テキスト表3. 3を導出せよ
- テキスト演習問題3. 3(P.86)

# 記憶素子を含む回路(順序回路)

- クロック同期デジタルシステムでは、順序回路を構成する記憶素子としてフリップフロップもしくはラッチを用いる。

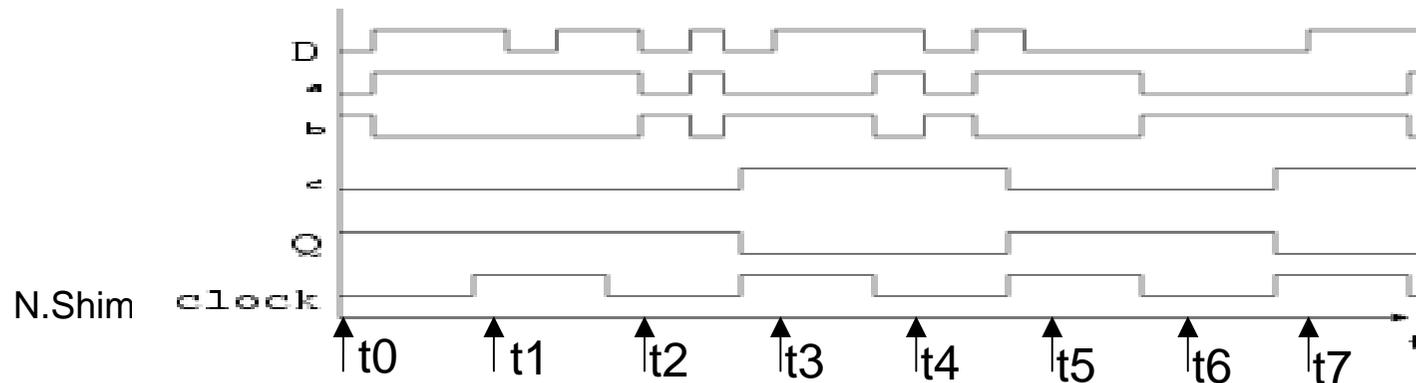
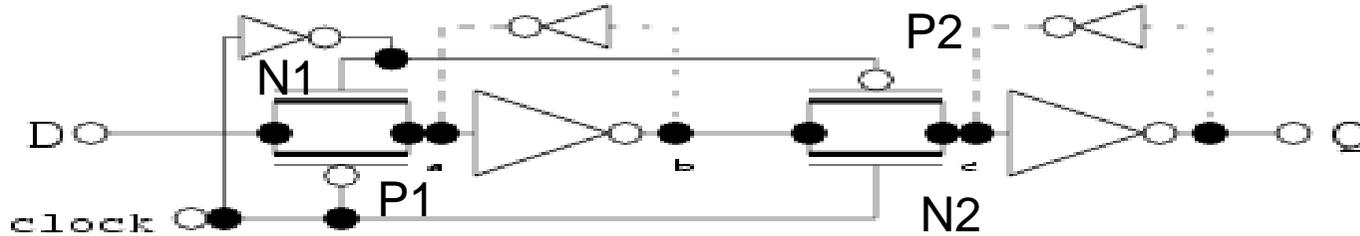
# ラッチ

- クロック信号が1のとき、入力データを出力する。クロック信号が0となると、その直前の入力データを保持する。(図3.19の後半分)



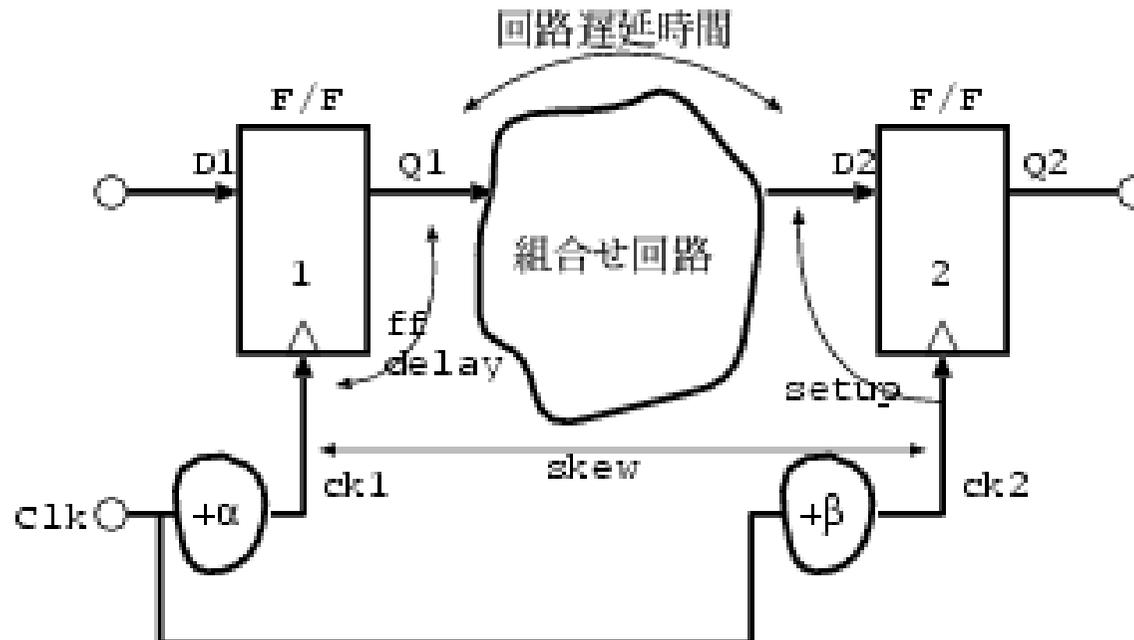
# フリップフロップ

- クロック信号が0から1に遷移すると、その直前の入力データを保持する。(図3. 19)
  - 図のタイムチャートのような出力となることを確認せよ(各トランジスタのON/OFFと各点の値)



# 順序回路のデータ伝送

- フリップフロップ間で伝送する時間がシステムの動作周波数を決定する。(3. 14式、3. 15式)
  - LSI中のクロック分配経路の分散によるクロック到着時間差(クロックスキュー)とFFのセットアップ時間、FF遅延時間、間の組合せ回路遅延を考慮する



# 回路の消費電力

- 理想的なCMOS回路では主要な電力消費は回路の負荷容量の充放電が原因となる。
  - クロック信号のように1周期中に2回状態遷移する信号の消費電力を式3.18に示す。
- 近年、高速化のため、トランジスタのスレッシユホールド電圧を引き下げる手法が多く、信号遷移の途中でPMOSとNMOSが同時にON状態となり、貫通電流が流れることによる電力消費、ならびに、トランジスタの縮小の結果、OFF時にも漏れ(リーク)電流が流れることによる電力消費が問題になりつつある。
- 演習： テキストP.86 演習問題3.4を行え。

# 熱設計(テキストP82)

- LSIのシリコン表面温度は信頼性確保のため100°Cを超えないように管理する。
- LSIの発熱はキャパシタの充放電によって発生する。
  - たとえば、演習3.4のプロセスで10000ゲートのLSIを作成したとする。消費電力はいくらか？
  - PentiumPRO トランジスタ数 550万トランジスタ(ゲート換算は4トランジスタが1ゲート)
- LSIは動いている部分だけ電力を消費する
  - 動く部分の割合を活性化率と呼ぶ
    - 一般的な論理回路では20%程度と考えられる
    - CPUの回路中一般論理回路は20%以下

# 回路シミュレーション演習

- CDRROMをPCに入れ、再起動
- BIOSメッセージが出たところでF12を押す
  - ブート選択メニューが出る
  - Windowsが起動したら再起動すること
- ブートデバイスにCDROMを指定
- ログインプロンプトが出たら、  
login: root  
password: toor  
としてログイン
- プロンプトでguiと入力し、Xウィンドウを立ち上げる
- Xの右下のアメリカ国旗のアイコンを左クリックし、日の丸に変更

# 回路シミュレータ

- 回路情報をコンピュータが認識できるテキスト情報(ネットリスト)として記述
  - 一部のシミュレータでは、回路図からネットリストを抽出する
- ネットリストに書いてある記述情報をコンピュータ内部のデータ構造に変換し、条件に従い回路動作をシミュレーションする。
- 過渡解析、DC特性解析、周波数特性解析
- 資料P1のネットリストを作成しよう
  - エディタは左下のメニューから選択もしくは端末を開きvi等を用いること
  - ファイル名を sample1.sp とする
  - 入力が終了したら、P7以降を参考にシミュレーションを実施
  - r1、c1の各値は演習3. 2のTrの近似抵抗値ならびに1mmの配線容量に修正してシミュレーションを実施すること
  - 結果はplotコマンドを用いて観測  
plot t2
  - グラフ上で拡大したい部分にマウスカーソルを置き、右クリックしたまま範囲を囲うと、当該領域が拡大されたグラフを表示できる。
  - 過渡解析の解析ステップを変更し、立ち上がり特性を観測しよう
- RC時定数と50%遅延時間、90%遅延時間、立上がり、立下り時間の関係をRとCの値を変更し確認せよ。RとCそれぞれ5種類ずつ程度

# トランジスタ回路のシミュレーション

- SPICEでは、素子名の命名法に規則がある
  - 抵抗はRで始まる
  - コンデンサはCで始まる
  - インダクタンスはLで始まる
  - 電圧源はVで始まる
  - MOSTランジスタはMで始まる
- 素子の宣言において、素子値などのパラメータは宣言の行に書く
  - たとえば、端子1と2に接続する1kΩの抵抗は
    - R1 1 2 1k
  - トランジスタなどのパラメータ数が多い素子は、別途モデル文でパラメータを宣言できる
    - パラメータ例はテキストP.73に示す
    - .model pfet pmos(level=3 kp=40u gamma=0.5 phi=0.36 ld=0.1u tox=20n nsub=4e+16 vmax=2e+5 cj=1.23e-4)
    - .model nfet nmos(level=3 kp=80u gamma=0.4 phi=0.37 ld=0.1u tox=20n nsub=2e+16 vmax=2e+5 cj=3.85e-4)
    - 配布資料のモデル文のパラメータは用いず、テキストのパラメータを使う
    - ただし、cjだけは配布資料の値を追加すること
- DC解析と過渡解析を行い、過渡解析はRC回路との差を調べること
- 過渡解析の結果の立上がり時間、立下り時間をグラフから読み取れ

# FDへ(から)のデータ移動

- データ書き込み
  - mcopy ファイル名 a:
- データ読み出し
  - mcopy a:ファイル名 .
- フロッピー中のファイル一覧
  - mdir a:

# CMOS論理回路の設計

- PMOSTランジスタとNMOSTランジスタを用いた、
  - 2入力NAND回路
  - 2入力NOR回路
  - $F = \text{not}(A \& B | C)$

を実現する回路図を作成し、SPICEシミュレーションで

- 動作を確認せよ。
- 1つの入力以外を1もしくは0に固定した時の立上がり・立下り時間を確認し、入力端子による違いがあれば、その理由を考察せよ
  - CMOS回路の回路構成については、テキストP.70,71を参照せよ

# CMOSレイアウト設計

- 設計ルールに従い、次のゲートのレイアウトを作成する  
MAGICでレイアウト設計せよ  
PMOSTランジスタの下にはNWELLを作成しておく。WELLは電源との接続(コンタクト)を必要とする。
  - インバータ
  - 2入力NAND
  - 2入力NOR
- CMOS設計ルール
  - 設計において、MOSISの設計ルール(SCNE)を適用する。
- [http://www.mosis.org/Technical/Layermaps/lm-scmos\\_scna.html](http://www.mosis.org/Technical/Layermaps/lm-scmos_scna.html)
- トランジスタのレイアウト例はテキストP65に示す。

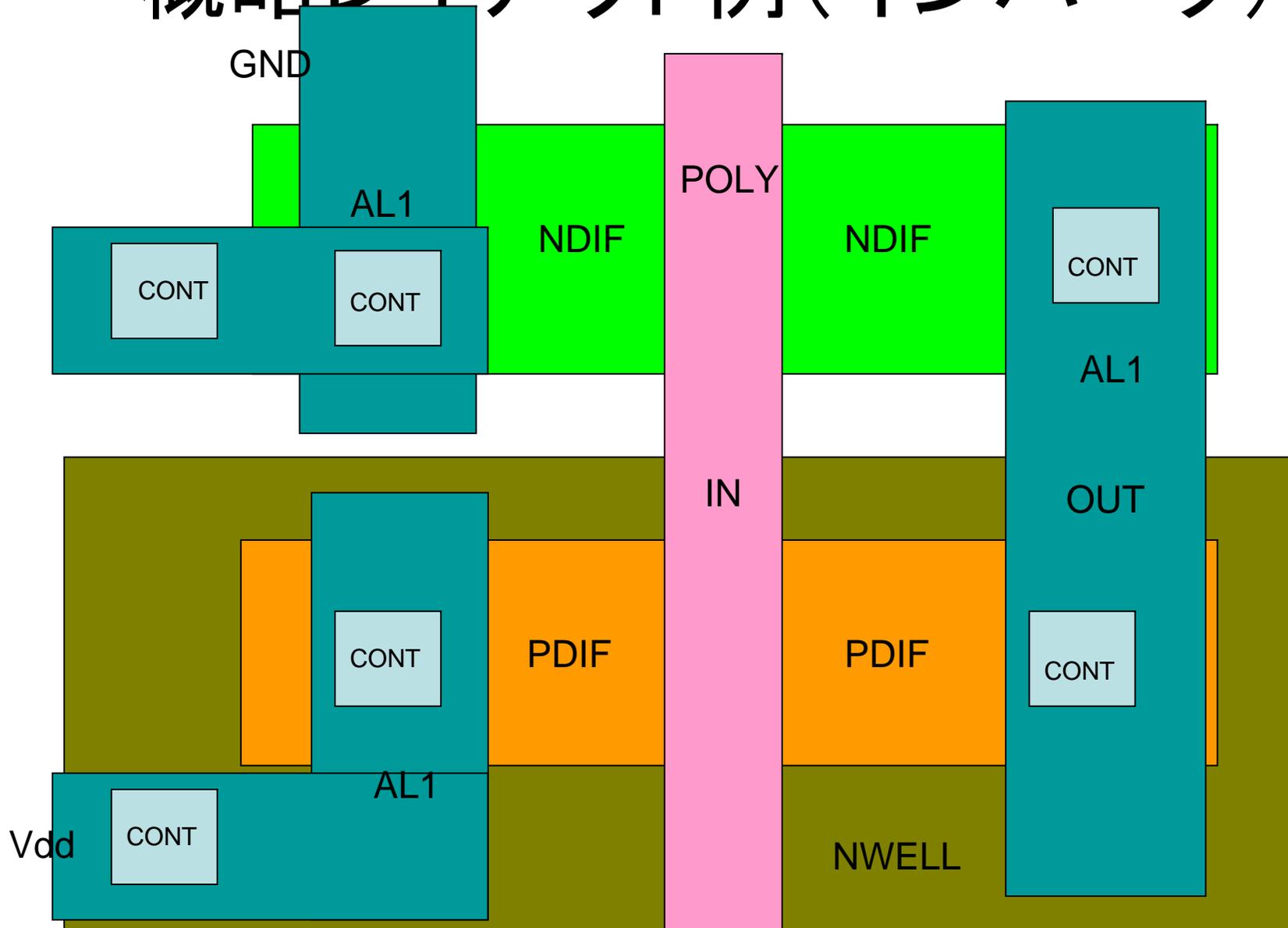
# 利用ルール

- NWELL
- ACTIVE
- POLY
- CONTACT
- METAL1
- VIA
- METAL2

# CMOSレイアウト

- PMOSTランジスタはNWELL上に作る
- NMOSTランジスタは基板上に作る
- アクティブ領域は下記領域として作成
  - N拡散領域(NMOSTランジスタ)
  - P拡散領域(PMOSTランジスタ)
- アクティブ領域上にポリシリコンを載せると、その部分がゲート領域になる
- アクティブ領域とアルミ1とは、コンタクトで接続
  - N拡散コンタクト
  - P拡散コンタクト
- 2層のアルミ同士はビア(VIA)で接続
- ポリシリコンとアルミ1はポリシリコンコンタクトで接続

# 概略レイアウト例 (インバータ)



# Magicを使いレイアウトをしよう

- 「Magicの使い方」と各自のグラフ用紙へのレイアウトを参考に、
    - インバータ
    - 2入力NOR
    - 2入力NANDを作成せよ
- 注: Magicは、コンタクトホール周辺の余裕部分を含んでコンタクトと定義している。
- 作成したレイアウトからSPICEネットリストを抽出し、SPICEで動作確認せよ

# Magicのレイアウトの印刷

- xwdとconvertコマンドを用いてPDF変換した後、Windowsマシンで行う  
Magicのレイアウト画面を他のWindowが隠さないように調整  
コマンドラインから  
xwd | convert - name.pdf  
と入力すると、カーソルが十字型になるので、レイアウト画面上で左クリック  
name.pdfをWindowsのマシンにコピーし、印刷  
name.pdfは任意の名前でOKなので、各自決めること

# 全加算器のレイアウト設計

- 作成したインバータ、NAND、NORを下位部品として呼び出し、全加算器のレイアウトを設計せよ
- さらに時間があれば、作成した全加算器を下位部品として、4ビット全加算器のレイアウトを設計せよ
  - 注：下位部品はコピーでなく、呼び出しとすること
  - ラベルにはグローバル符号！を使わない
  - レイアウトにNMOS用のPWELLがあるほうがベター

# 課題

- 全加算器の回路シミュレーションを行い、回路の立ち上がり時間、立ち下がり時間を入力端子ごとに調べよ
- 調べた立ち上がり時間・立下り時間のうち、最大のものを1/3にする手段を検討し、回路を修正し、シミュレーションで確認せよ
- 基本ゲートのレイアウト図(グラフ用紙・画面コピー)、全加算器のレイアウト修正前と修正後(画面コピー)を提出