

集積回路とEDA SPICEの使い方

土谷 亮

tsuchiya@vlsi.kuee.kyoto-u.ac.jp

詳細資料入手先: 小野寺研 集積回路工学特論のページ

<http://www-lab13.kuee.kyoto-u.ac.jp/~tsuchiya/spice-lecture.html>

今日の内容

- そもそもEDAとは
 - ◆ なぜ必要なのか
 - ◆ SPICEとは
- SPICEの使い方
 - ◆ 解析の種類
 - ◆ 回路・解析条件の与え方
 - ◆ 操作手順: LTspice, hspice

今日の内容

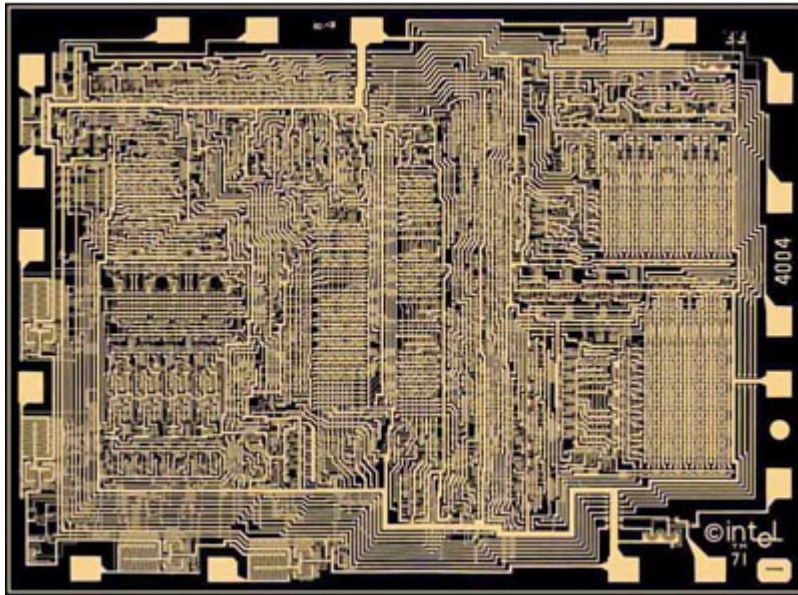
- そもそもEDAとは
 - ◆ なぜ必要なのか
 - ◆ SPICEとは
- SPICEの使い方
 - ◆ 解析の種類
 - ◆ 回路・解析条件の与え方
 - ◆ 操作手順: LTspice, hspice

集積回路とEDA

- Electric Design Automation (設計自動化技術)
 - ◆ 計算機による設計・製造支援ツール
 - CAD (Computer Aided Design)
 - CAE (Computer Aided Engineering)
 - CAM (Computer Aided Manufacturing)
 - 厳密な使い分けは存在しない
 - ◆ 多数の会議が開催される集積回路の一大研究分野
- 市場規模は年間50億ドル程度※

なぜEDAが必要なのか

EDAのなかった時代



Intel 4004 (1971年)

3mm x4mm, 10 μ mプロセス, 2300Tr

4bit, 動作周波数741kHz

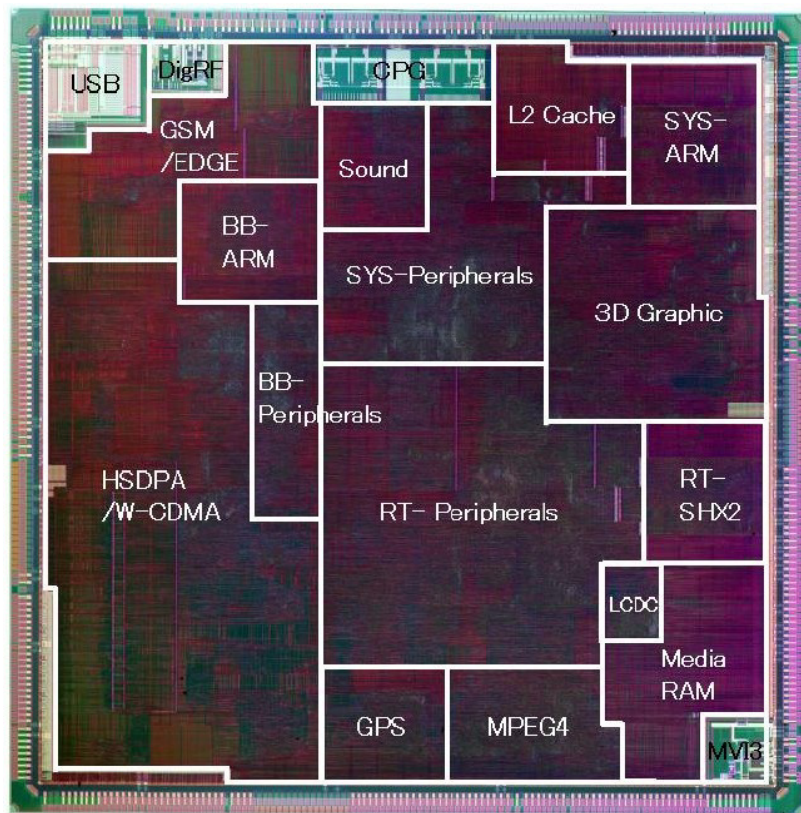
入出力16ピン

世界初の商用マイクロプロセッサ

- ほとんどのロジックは一人～数人の技術者が設計
 - ◆ ちなみに4004の設計者は嶋正利
- 紙と鉛筆の世界

このころはそれでよかったが...

現在の集積回路



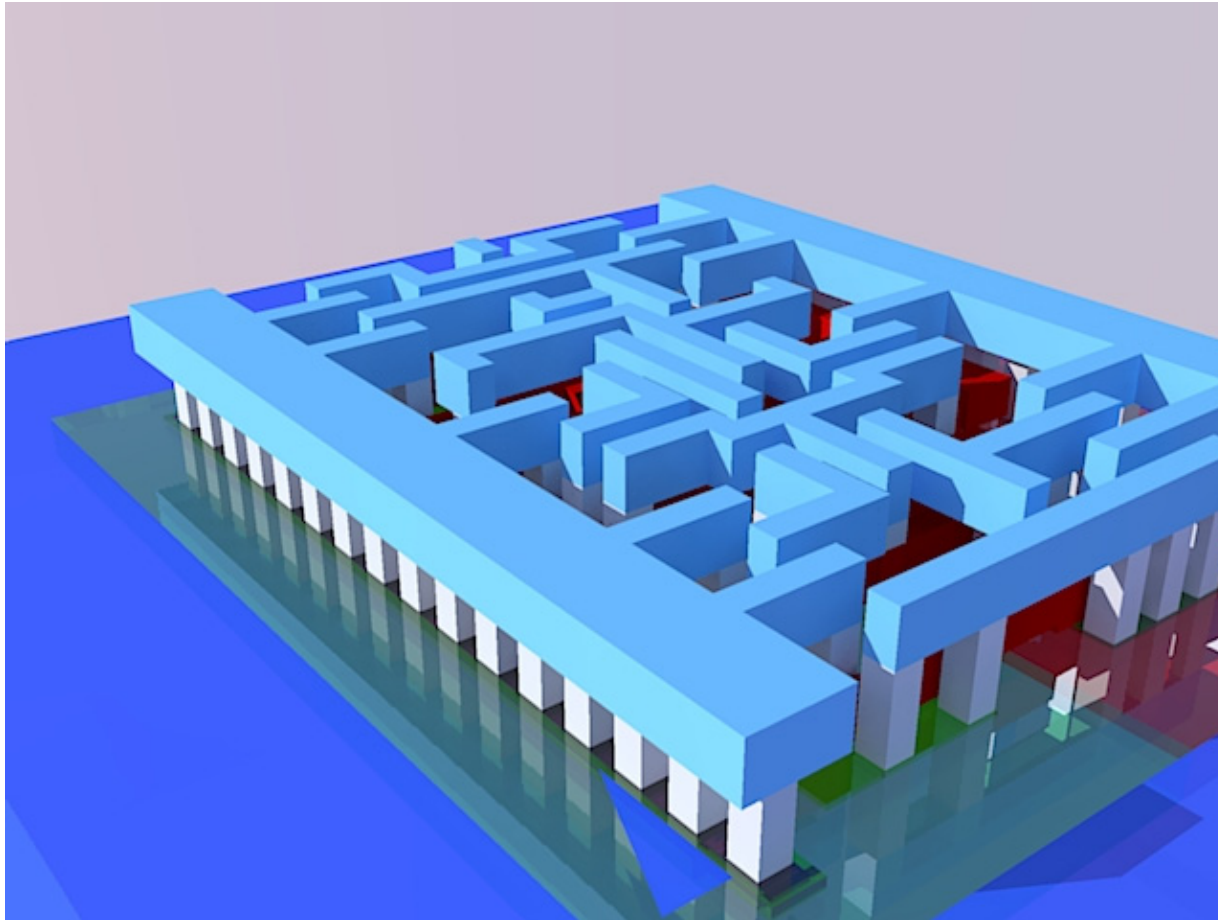
Renesas SH-Mobile G3 (2008)

9.3mm x 9.3mm, 65nmプロセス, 3億Tr
配線層8層, 電源ドメイン21個
入出力617ピン

Photo from ISSCC2008

- これを人手で?
 - ◆ 機能検証
 - ◆ 配置・配線
 - ◆ タイミング検証
 - ◆ マスクデータ作成
 - ◆ etc. etc...
- しかも開発期間は半年～1年

集積回路の複雑さ



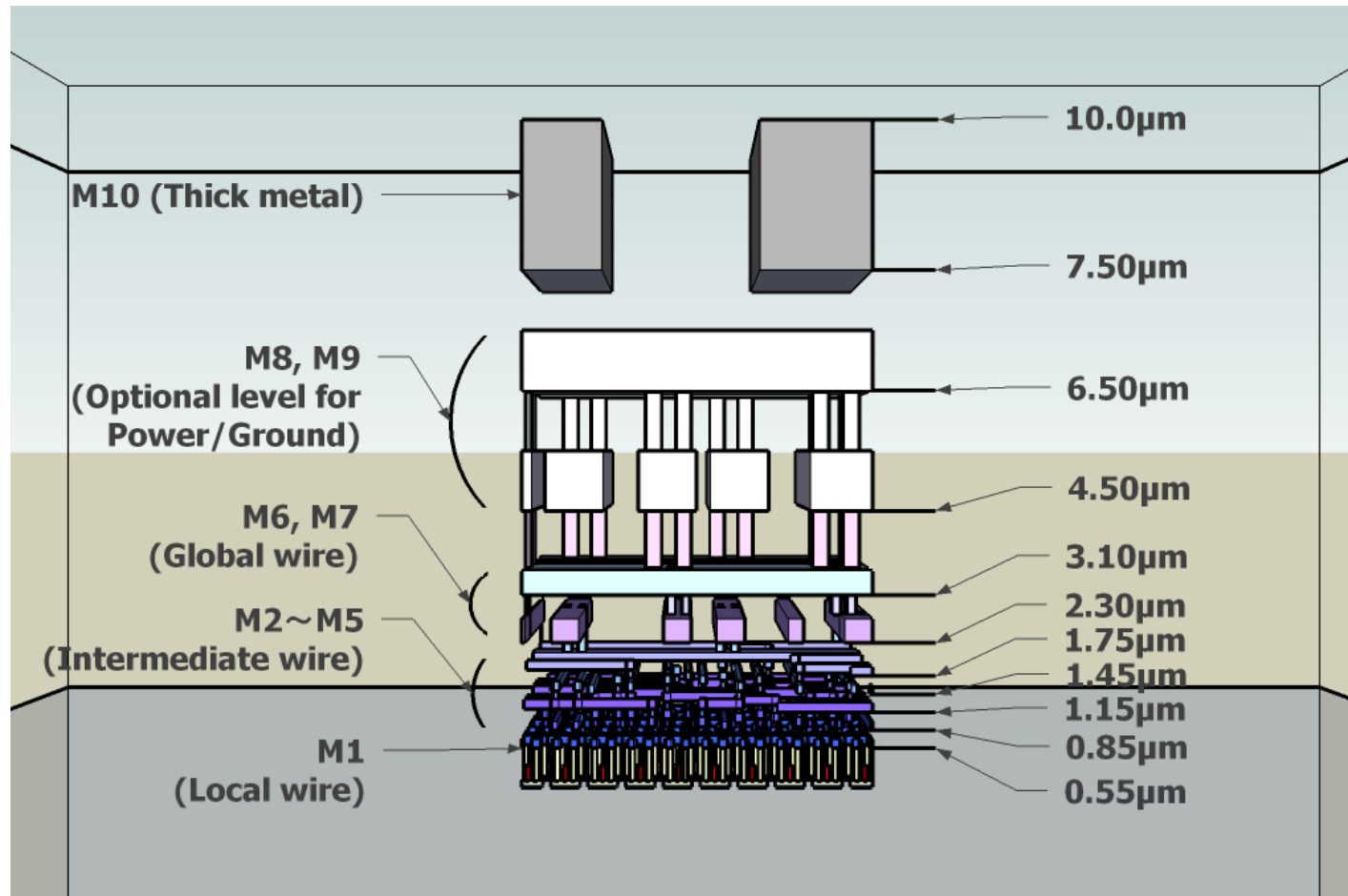
D-flipflopの立体イメージ
これで1ビット

レポートで作成した
NANDレイアウトは
4Tr, 配線層1



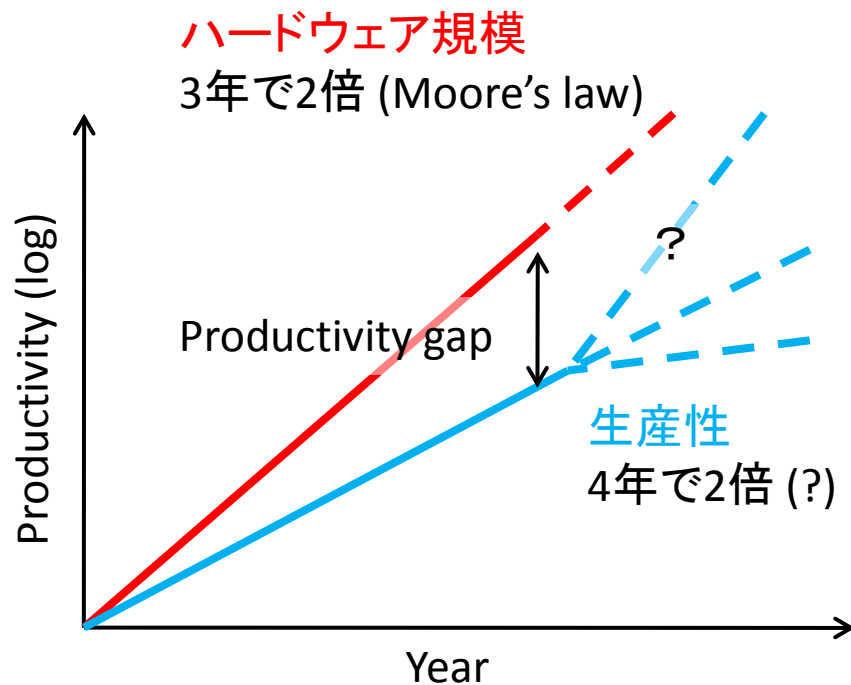
最先端プロセス
数億Tr
配線層10前後

集積回路の複雑さ



10層配線プロセスの立体イメージ
最小の配線幅は100nm以下

EDAの恩恵：生産性



Productivity gap:
ハードウェア規模の増大に
生産性が追いつかない

↳ 設計, 評価, 検証 etc.

「生産性を上げる」のが
大きな目的

試作して測って・・・を計算機上で

➡ 回路シミュレータ
(SPICEなど)

回路図からレイアウトを自動作成

➡ 自動配置配線

レジスタのレベルから回路図を生成

➡ ハードウェア記述言語
(Verilog, VHDL)

実測とEDA

実測

試作に時間がかかる



EDA

すぐに評価が可能

測定自体が難しい
擾乱なしの測定は不可能



任意の点を評価可能
擾乱なしに評価が可能

予期せぬ事態も捉える
ことができる



モデル化されていない
事象は評価できない

実測もシミュレーションも「何を評価しているのか」を
理解して使うことが重要

わけもわからずSPICEをまわし続ける人を貶める“SPICE monkey”という言葉も

EDAの出発点 SPICEとは

- Simulation Program with Integrated Circuit Emphasis
- UC Berkeley で1973年に開発
 - ◆ スパコンが100MFLOPSぐらいの時代
ちなみに Intel Core i7 は50GFLOPSぐらい
 - ◆ 入力はパンチカードだった
- 非線形素子を含む回路を解析できる
- “SPICE”は「トランジスタレベルの回路シミュレータ」の代名詞

SPICEの重要性

- 回路シミュレータのDefacto Standard
 - ◆ 無料で入手可能
- シンプルで強力な非線形問題ソルバー
 - ◆ 用途は集積回路だけではない
 - ◆ 回路にマッピングできれば電気回路以外にも
 - 熱抵抗と熱容量で熱の拡散を解く, など
 - 「問題をどうSPICEにマッピングするか」という研究も行われている

今日の内容

- そもそもEDAとは
 - ◆ なぜ必要なのか
 - ◆ SPICEとは
- SPICEの使い方
 - ◆ 解析の種類
 - ◆ 回路・解析条件の与え方
 - ◆ 操作手順: LTspice, hspice

SPICEでできる解析

- 直流解析 (DC analysis)
 - ◆ 時間的に変化しない電圧・電流の関係
 - ◆ トランジスタの動作点解析など
- 小信号解析 (AC analysis)
 - ◆ 特定バイアス下での周波数応答
- 過渡解析 (Transient analysis)
 - ◆ 時間的に変化する信号に対する応答

直流解析

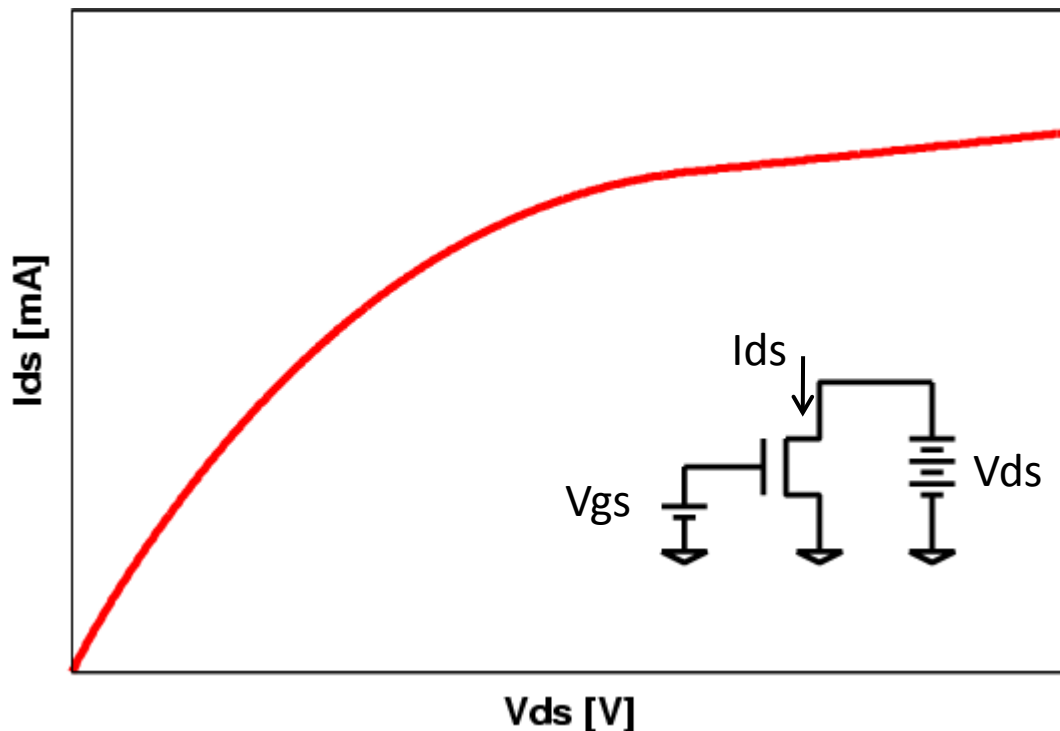
- 回路にある電圧・電流を与えた際に各部の電流・電圧がどうなるか?

例: トランジスタの 電流電圧特性

指定した電圧(V_{gs} , V_{ds})を与えたときに電流(I_{ds})はどのくらい流れるか? を解析

回路の入出力特性,
アナログ回路のバイアス確認
などに使う

対応する測定器:
デジタルマルチメータ



小信号解析

■ 回路の周波数応答を解析する

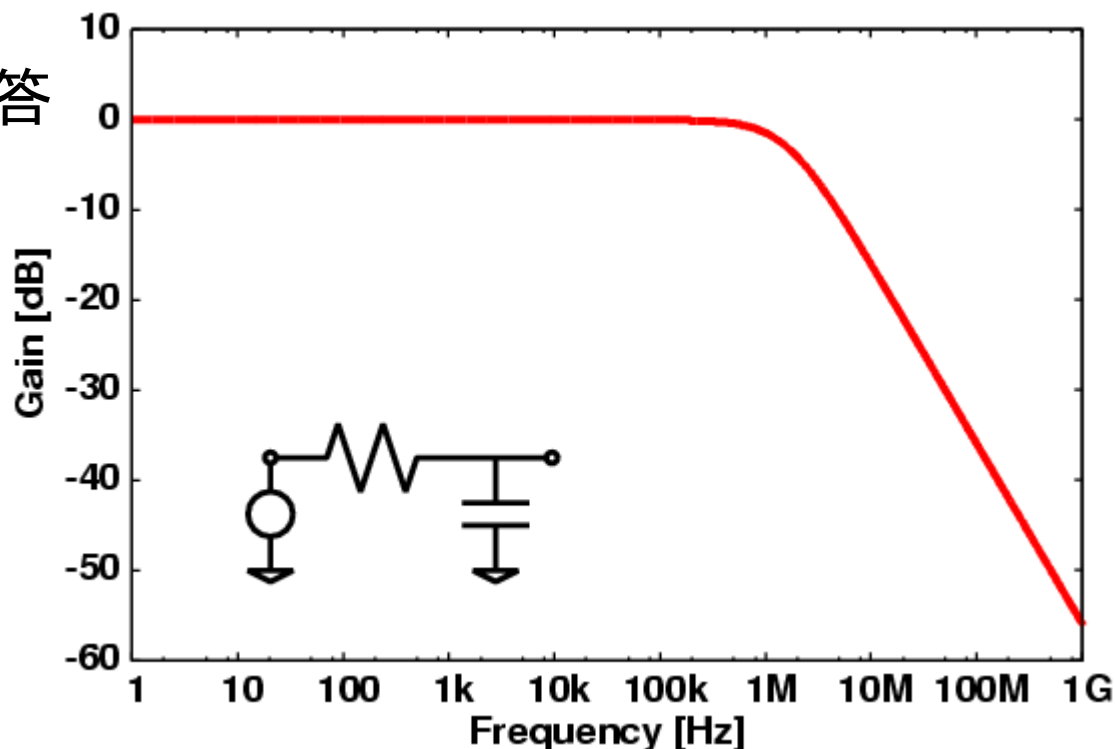
小信号：素子が線形素子とみなせる程度の大きさの信号

例：RCフィルタの周波数応答

入力信号の周波数を
上げていくとゲインはどうなるか？
を解析

フィルタ、アンプなどの
周波数特性の確認に使う

対応する測定器：
ネットワークアナライザ



過渡解析

■ 時間的に変化する信号に対する応答を解析

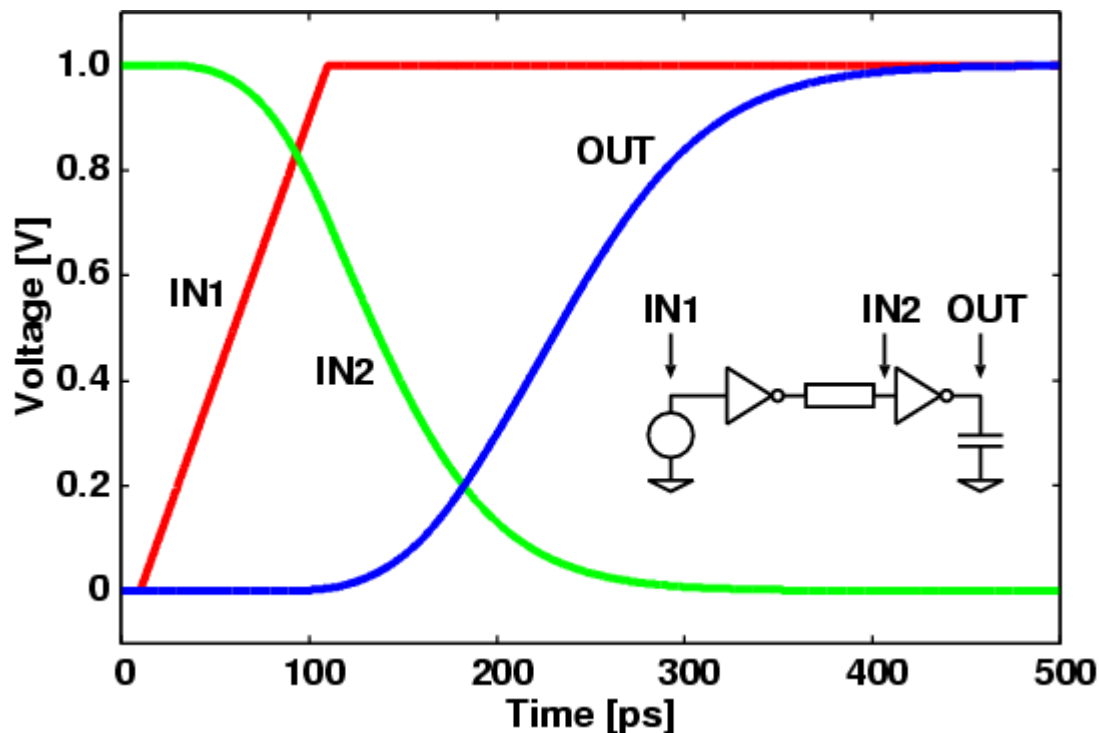
小信号解析に対して大信号解析と言うことも

例：インバータの入出力波形

IN1にパルスを入力したら
各部の波形はどうなるか？
を解析

遅延時間, 信号遷移時間,
遅延などの評価に使う

対応する測定器：
オシロスコープ



SPICEの入力ファイル (ネットリスト)

SPICEに与える入力ファイルを「ネットリスト」と呼ぶ

ネットリスト

素子の接続関係

電気回路は素子と
節点からなる
グラフとして記述できる

モデルパラメータ (モデルカード)

回路が非線形素子を
含む場合, その特性は
素子の種類に応じた
パラメータで表現する

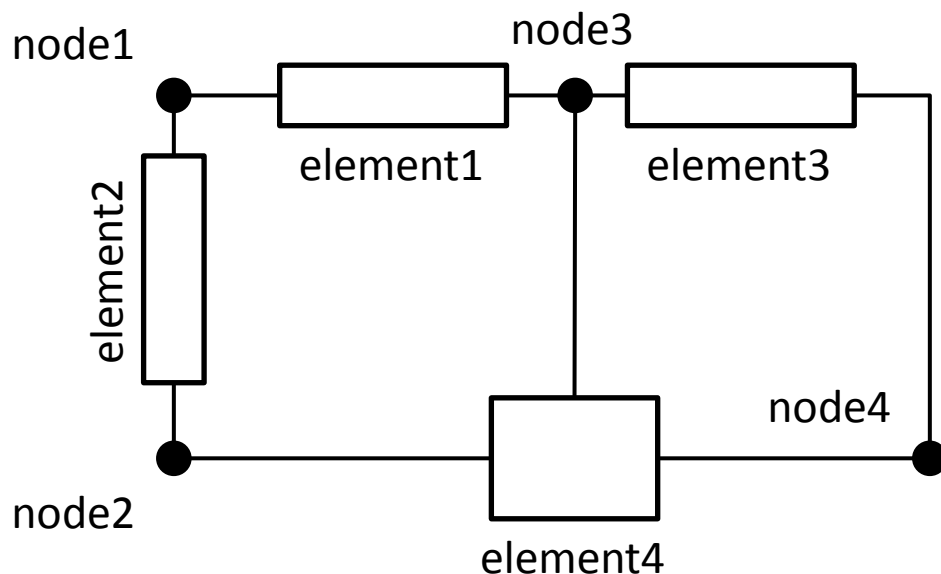
解析条件 オプションなど

電圧や周波数の範囲
何を出力するか
etc.

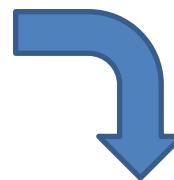
回路図や解析条件をグラフィカルに設定することもできるが,
今回はテキストベースの方法を説明

回路図の記述方法

電気回路は素子と節点で記述できる



これをテキストで書くと...



element1	node1	node3	
element2	node1	node2	
element3	node3	node4	
element4	node2	node3	node4

注1: 素子には方向があるものがある
(電圧源, 電流源など)

注2: 素子は3個以上の端子をもつことがある
(トランジスタはD, G, S, B の4端子素子)

ネットリストの書式：基本構造

記述の基本構造

Xname node1 node2 ... nodeN value PARAMETERS

素子の種類を示す
アルファベット1文字
例：

R → 抵抗
C → キャパシタ
V → 電圧源
M → トランジスタ

素子の名前
英数字からなる
文字列

素子の種類が
違えば重複可
例：

NG Rin 1 2 50
Rin 2 3 50

OK Rin 1 2 50
Cin 2 3 100p

接続されたノード
数は素子によって
決まっている
ノード名は
英数字文字列
ただしノード"0"は
グラウンド

素子の値
抵抗なら抵抗値
電圧源なら電圧
値がないものもある
(例：トランジスタ)

補助パラメータ
トランジスタの
サイズなど
必要に応じて指定

※記述は1行に書く。複数行にわたる場合は2行目以降の先頭に“+”をつける

ネットリストの書式: 2端子素子

抵抗	Rname node+ node- value
キャパシタ	Cname node+ node- value
電圧源	Vname node+ node- value
電流源	Iname node+ node- value



電圧源, 電流源には方向があることに注意

電圧源は node- が電圧の基準点

電流源は node- から node+ に向かって電流が流れる

抵抗, 容量には方向はない

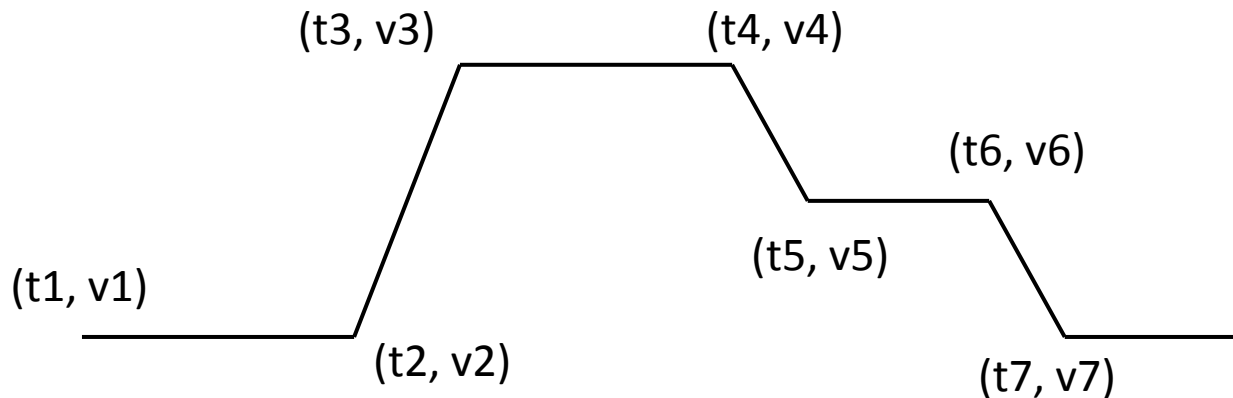
ネットリストの書式: 信号源

時間的に変化する電源は value をただの値ではなく関数にする

信号源 (PWL)

Vname node+ node- pwl t1 v1 t2 v2 ...

PWL (PieceWise Linear): 折れ線波形
(時刻, 電圧) の値を任意の個数列挙



ネットリストの書式:トランジスタ

MOSTランジスタは4端子素子であることに注意

MOSTランジスタ

```
Mname d g s b model L=l W=w  
+AD=ad PD=pd AS=as PS=ps
```

d : drain, g : gate, s : source, b : body

model : モデルの名前 (後述)

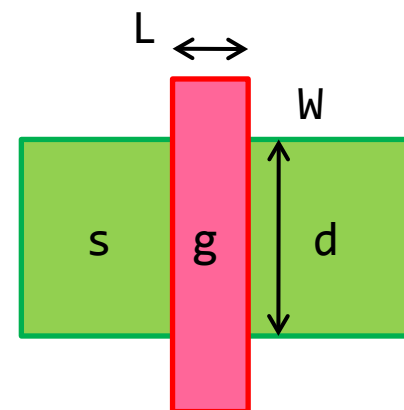
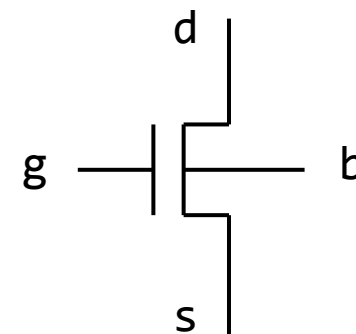
L : ゲート長

W : ゲート幅

AD : Area of drain ドレインの面積

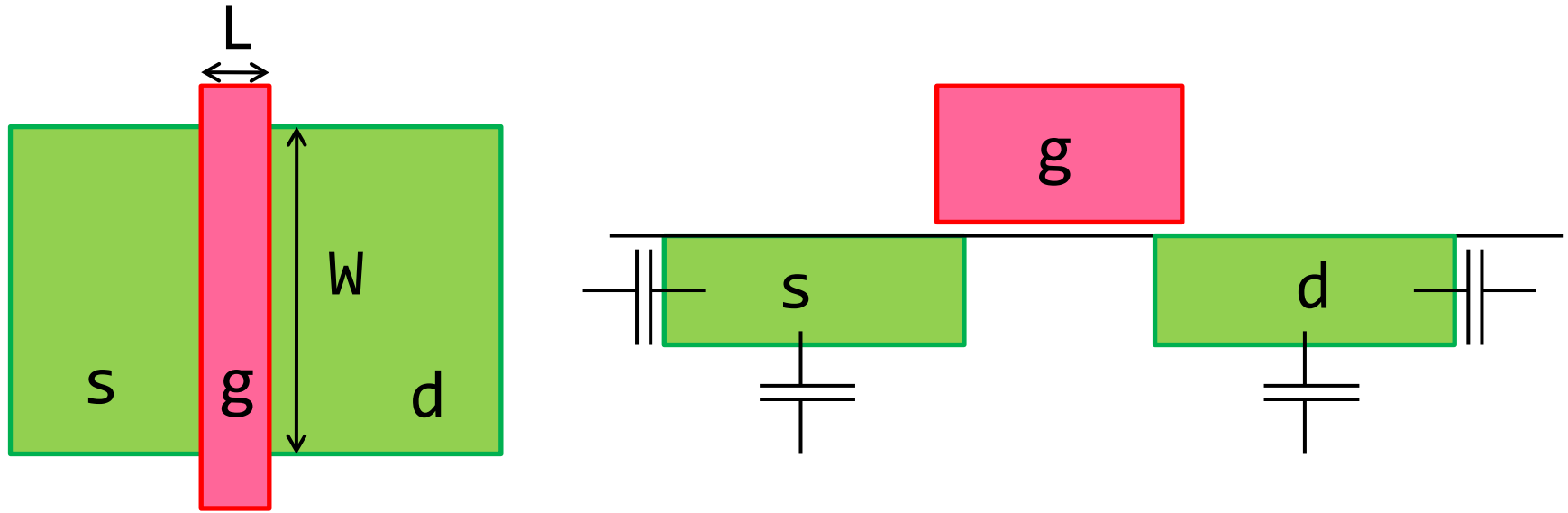
PD : Perimeter of drain ドレインの周囲長

AS, PS : ソースの面積, 周囲長



L, W, AD, PD, AS, PS は
レイアウトに依存する

AD, PD, AS, PSの意味



ドレイン/ソース領域のPN接合の容量を計算するのに使用

$$C_{drain} = C_j \times AD + C_{jsw} \times PD$$

$$C_{source} = C_j \times AS + C_{jsw} \times PS$$

C_j : 底面の容量

C_{jsw} : 側壁(Sidewall)の容量

正しく設定しないと遅延時間が狂う
桁の間違いに注意

ネットリストの書式：モデルパラメータ

モデルパラメータ (モデルカード) とは:

非線形素子の特性を記述する方程式のパラメータ

例えば
MOS飽和領域の電流

$$I_{ds} = \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{gs} - V_{th})^2 (1 + \lambda V_{ds})$$

μ , C_{ox} , λ はデバイス依存のパラメータ
→ モデルパラメータとして与える

モデルパラメータ (MOS)

.MODEL modelName type LEVEL=...

modelName: モデルの名前. MOSの素子記述内で指定する

type: MOSの場合は NMOS / PMOS

モデルパラメータは授業のWebページからコピーして使うこと

ネットリストの書式: 解析条件/出力

解析条件の設定 (過渡解析)

```
.tran step end
```

時刻0 から step 刻みで end まで解析

どういう値に設定すればよいかは回路によって変わる

自分の回路にとって適切な値がどの程度かよく考えること

解析結果の出力(過渡解析)

```
.print tran V(node1) V(node2) ...
```

指定したノードの各時刻における電圧が出力される

ネットリストの記述終了

```
.end
```

必ず書かなければならない. この行以降は無視される.

ネットリストの書式：数値

数値+接尾辞で記述が可能

1f = $1\text{e-}15$

1p = $1\text{e-}12$

1n = $1\text{e-}9$

1u = $1\text{e-}6$

1m = $1\text{e-}3$

1k = $1\text{e+}3$

1Meg = $1\text{e+}6$

マイクロ (u, $1\text{e-}6$)と
メガ(Meg, $1\text{e+}6$)に注意

1Meg のつもりで 1M と書くと
 $1\text{e+}6$ ではなく $1\text{e-}3$ になる

接尾辞以外の文字は無視される

Rload 1 2 10k	}	Ohmは無視されるのでどちらも同じ
Rload 1 2 10kOhm		

ネットリストの書式：雑多な注意

- 回路中にはかならずグラウンド (ノード “0”) がなければならない
- ネットリストの1行目はタイトル
- 行の先頭が “*” の行はコメント
- 大文字・小文字は区別されない
- “.end” を書き忘れないように注意

ネットリストの書式：サブサーキット

SPICEでは素子を組み合わせで独自の素子を定義することができる

サブサーキットの定義

```
.subckt name node1 node2 ...  
* circuit description  
.ends
```

サブサーキットを使う

```
Xname node1 node2 ... nodeN name
```

INV, NAND など同じ回路が何度も出てくるときに便利

サブサーキットの例

*inv

```
.tran 0.05n 4n  
.print tran v(3)
```

```
Vdd 1 0 5V  
Vin 2 0 pw1 0 5V ...
```

```
Mxp 3 2 1 1 CMOSP L=...  
Mxn 3 2 0 0 CMOSN L=...
```

```
Cout 3 0 0.2p
```

```
.MODEL CMOSP PMOS ...
```

...

```
.end
```

*inv-subckt

```
.tran 0.05n 4n  
.print tran v(3)
```

```
Vdd 1 0 5V  
Vin 2 0 pw1 0 5V ...
```

```
Xinv 2 3 1 0 INV
```

```
Cout 3 0 0.2p
```

```
.subckt inv in out vdd vss  
Mxp out in vdd vdd CMOSP L=...  
Mxn out in vss vss CMOSN L=...  
.ends
```

```
.MODEL CMOSP PMOS ...
```

...

ネットリストの書式:.measure

解析結果から指定した2点の距離を測定する

.measureによる測定

```
.measure tran name  
+ trig v(node1) val=v1 cross=m  
+ targ v(node2) val=v2 cross=n
```

「node1 の電圧がm回目に v1 になった時刻」(トリガ)から
「node2 の電圧がn回目に v2 になった時刻」(ターゲット)
までの時間を出力する

cross を rise にすると

「m回目に電圧が v1 を下から上に横切った時刻」

fall にすると

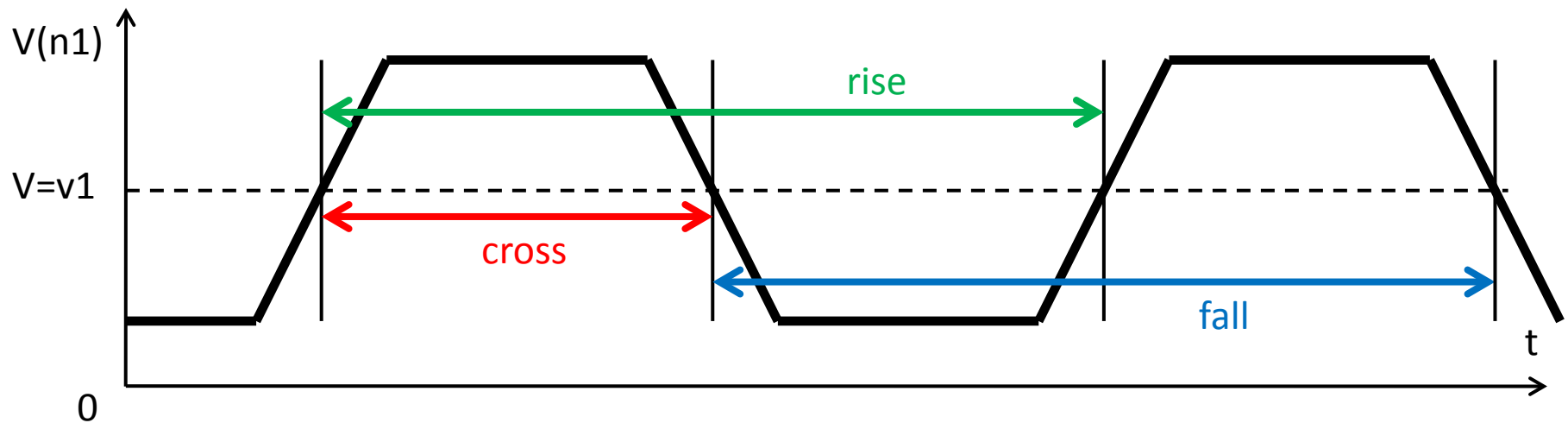
「m回目に電圧が v1 を上から下に横切った時刻」

.measure の例

```
.measure tran name
```

```
+trig v(n1) val=v1 {cross|rise|fall}=1
```

```
+targ v(n1) val=v1 {cross|rise|fall}=2
```



想定外のところでトリガがかかったり
ターゲットにひっかかったりするので
使う場合はきちんと波形を確認してから使うこと

LTspice の使い方

- LTspice : Linear Technology が配布している SPICE
 - ◆ <http://www.linear-tech.co.jp/designtools/software/>
 - ◆ spice3 に改良を加えたもの
 - ◆ Windowsで動く
- その他利用可能なSPICE
 - ◆ ngspice (Next Generation SPICE)
 - “ngspice for windows”でVectorからダウンロード可能
 - ◆ hspice (Synopsys社, 小野寺研WSで利用)
 - 商用のSPICE, 高性能

ネットリストの準備

- メモ帳で書く
 - ◆ まずサンプルで動作確認を

小野寺研 → 集積回路工学特論「SPICEの使い方」

<http://www-lab13.kuee.kyoto-u.ac.jp/~tsuchiya/spice-lecture.html>

Linux と Windows でファイルをやりとりした場合に
改行コードの問題でエラーになることがあるので注意

ファイルの読み込み・解析実行

- File → Open でファイル読み込み
 - ◆ 読み込んだファイルを編集することも可能
- Tools → Control Panel で出力形式を設定
 - ◆ “ASCII data files” にチェックを入れる
- Simulation → Run で実行
 - ◆ 波形表示ウィンドウが開く
 - ◆ 結果のファイルがいくつか生成される
 - filename.raw : 波形ファイル
 - filename.log : 実行ログ

結果の確認

- 波形を見る：波形ウィンドウ
 - ◆ 右クリック→ “Add trace” / “Delete traces” など
- 数値データの確認：filename.raw
 - ◆ メモ帳で開けない場合には “ASCII data files” にチェックが入っているか確認
- .measure の結果：filename.log

.raw ファイルの中身

```
Title: **INVERTER**
```

```
...
```

```
Variables:
```

0	time	time
1	V(2)	voltage
2	V(3)	voltage

データの
通し番号



```
Values:
```

0	0.000000000000000000e+000
---	---------------------------

	5.000000000000000000e+000
--	---------------------------

	2.071922426216685e-009
--	------------------------

1	3.906249931451836e-017
---	------------------------

	4.999996093750069e+000
--	------------------------

	-1.449166884356809e-007
--	-------------------------

```
...
```

.log ファイルの中身 (.measureの結果)

ネットリスト .measure tran period
 +trig v(3) val=2.5 cross=1
 +targ v(3) val=2.5 cross=2

filename.log

結果

Circuit: **INVERTER**

Model "cmosn05": Oxide thickness thinner than ...
...

→ period=1.83176e-009 FROM 3.04677e-010 TO 2.13644e-009

Date: Wed Jun 16 10:09:40 2010

Total elapsed time: 0.140 seconds.

...

hspice の使い方

- すべてCUI
 - ◆ collabo1.kuee.kyoto-u.ac.jp に ssh でログイン
 - ◆ ネットリストを scp で転送
 - ◆ hspice filename を実行
 - ◆ .print で指定した結果が標準出力に出力される
 - ◆ .measure の結果は filename.mt0 に格納

参考

小野寺研 集積回路工学特論のページ

<http://www-lab13.kuee.kyoto-u.ac.jp/~tsuchiya/spice-lecture.html>

Linear Technology (LTspice, マニュアルダウンロード)

<http://www.linear-tech.co.jp/designtools/software/>

UC Berkeley “The Spice Page”

<http://bwrc.eecs.berkeley.edu/Courses/lcBook/SPICE/>

NGSPICE

<http://ngspice.sourceforge.net/>