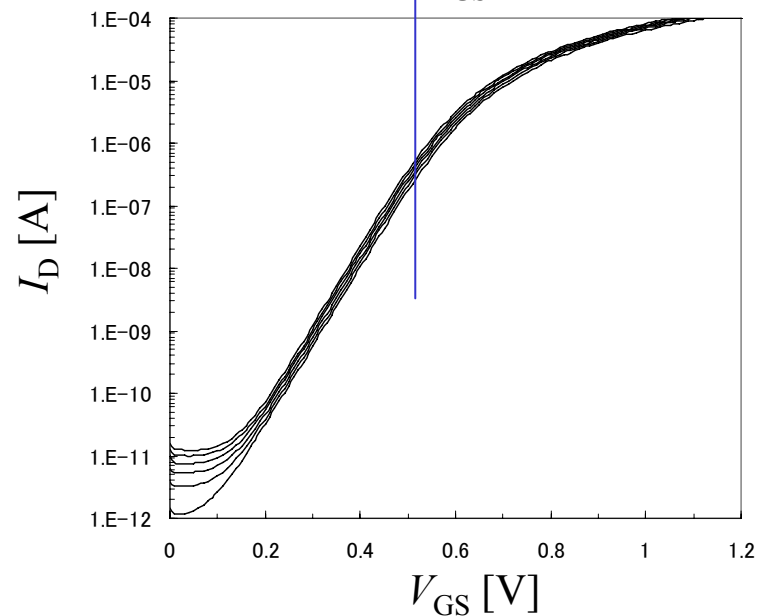
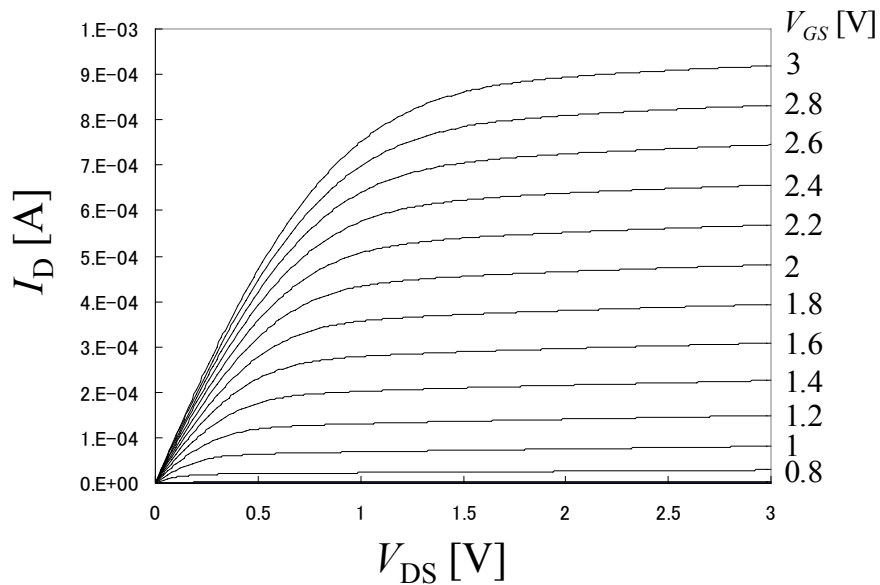
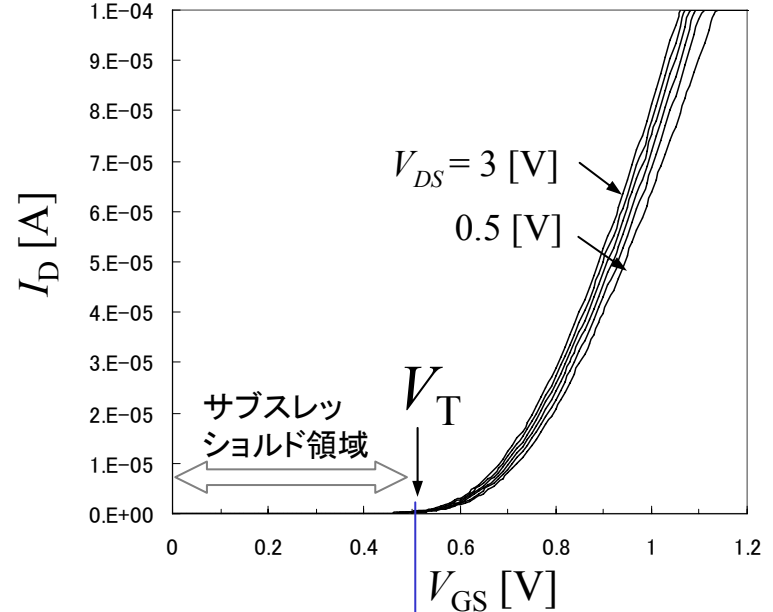
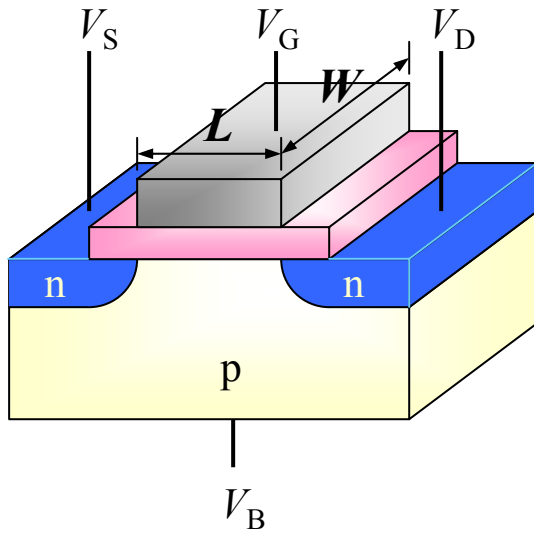


# 情報デバイス工学特論

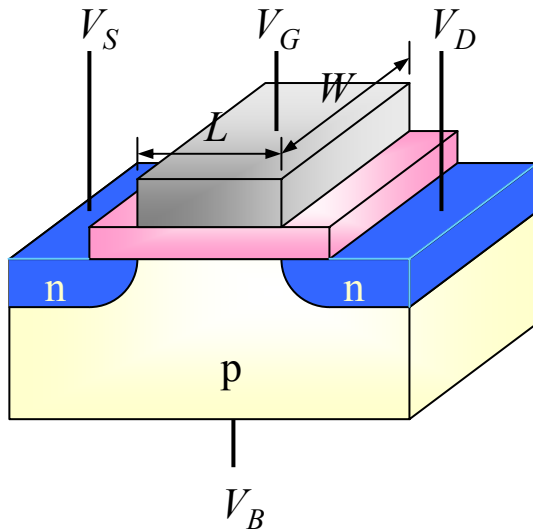
## 第4回

### CMOS基本特性とシミュレーション技術

# nMOSFET 直流特性



# nMOSFET 直流特性



線形領域  $V_{GS} - V_T > V_{DS}$

$$I_D = \beta \left[ (V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right]$$

飽和領域  $V_{DS} > V_{GS} - V_T > 0$

$$I_D = \frac{\beta}{2} (V_{GS} - V_T)^2 (1 + \lambda V_{DS})$$

理想トランジスタ・モデル

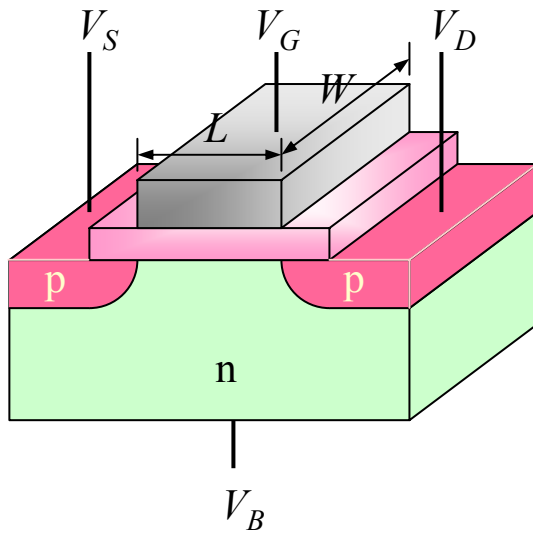
$$\beta = \frac{W \mu_n C_{ox}}{L}$$

$$V_T = V_{FB} + 2\phi_F + \frac{\sqrt{2qN_A \epsilon_S (2\phi_F - V_{BS})}}{C_{ox}}$$

$$V_{FB} = \Phi_M - \chi - \phi_F - \frac{E_C - E_{Fi}}{q}$$

二次の効果

# pMOSFET 直流特性



線形領域  $V_{GS} - V_T < V_{DS}$

$$I_D = -\beta \left[ (V_{GS} - V_T) V_{DS} - \frac{1}{2} V_{DS}^2 \right]$$

飽和領域  $V_{DS} < V_{GS} - V_T < 0$

$$I_D = -\frac{\beta}{2} (V_{GS} - V_T)^2 (1 - \lambda V_{DS})$$

理想トランジスタ・モデル

$$\beta = \frac{W \mu_p C_{ox}}{L}$$

電圧と電流の符号が変わる  
通常、閾値  $V_T$  は負

$$V_T = V_{FB} + 2\phi_F - \frac{\sqrt{2qN_D \epsilon_S (-2\phi_F + V_{BS})}}{C_{ox}}$$

$$V_{FB} = \Phi_M - \chi - \phi_F - \frac{E_C - E_{Fi}}{q}$$

# 実際のトランジスタ・パラメータはどの程度？

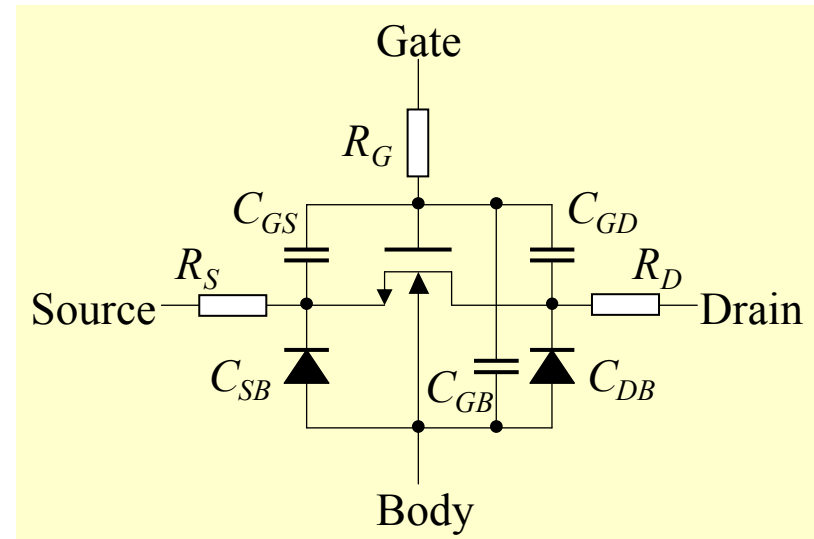
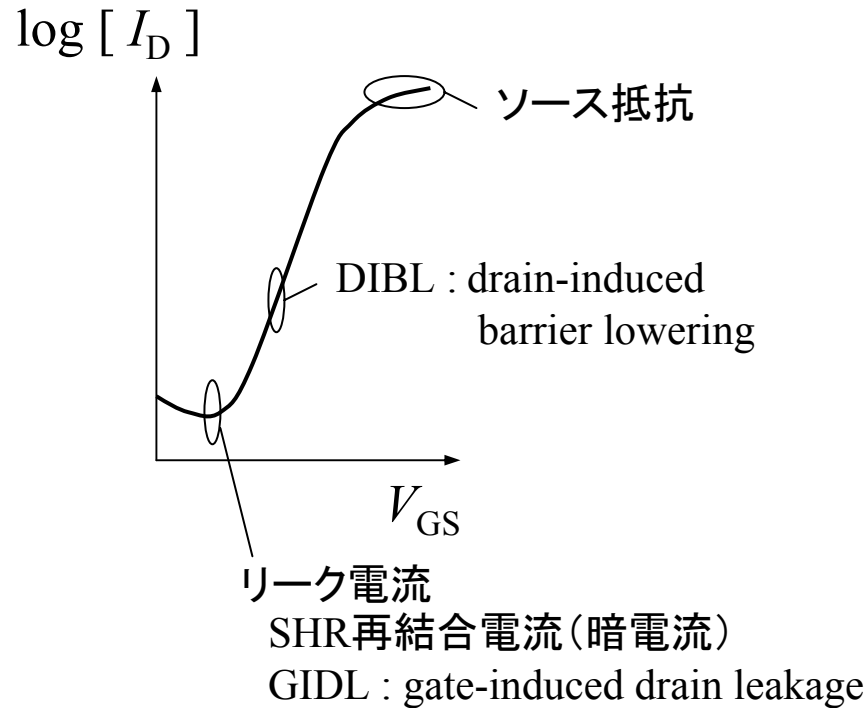
VDEC ローム 0.35  $\mu\text{m}$  プロセス

$$L = 0.4 \mu\text{m} \quad W = 2 \mu\text{m}$$

	nMOS $V_{DS} = 3\text{V} \quad V_{GS} = 0.7\text{V}$	pMOS $V_{DS} = -3\text{V} \quad V_{GS} = -0.7\text{V}$
$I_D$	20 $\mu\text{A}$	12 $\mu\text{A}$
$V_T$	0.5 V	-0.65 V
$V_{Dsat}$	0.15 V	-0.05V
$\beta$	$1.3 \times 10^{-3} \text{AV}^{-2}$	$4.3 \times 10^{-4} \text{AV}^{-2}$
$g_m$	$1.8 \times 10^{-4} \text{S}$	$2 \times 10^{-5} \text{S}$
$r_o$	$6 \times 10^5 \Omega$	$2.5 \times 10^6 \Omega$

理想トランジスタ・モデルとのずれは結構大きい

# トランジスタの特性は複雑



- ショートチャンネル効果
- 高電界効果
  - インパクト・イオン化
  - 移動度劣化
- ゲート薄膜化に伴う現象
  - トンネル電流
  - ゲート空乏化

# シミュレーション技術

- 計算機実験 ⇒ 内部電位等、測定困難な物理量を調べることにより現象を理解
- 新しい素子の検討 ⇒ 種々の効果を取り入れ最適化した特性の予測
- 考え間違いの防止 ⇒ シミュレーションにより思わぬ考え忘れを発見
- 試作に時間・金がかかる ⇒ シミュレーションにより条件を絞り込むことが必要

シミュレーションに頼りすぎるのも危険

シミュレーションの前に理想化された(単純化した)モデルで十分検討しておかないと時間の浪費に陥る

# LSI におけるシミュレーション





# UNIX の基本コマンド

## ほとんどのシミュレータはUNIX 上で動作

- ・互換性・信頼性・自由度が大きい(ソフトはソース・ベース)
- ・豊富なユーティリティ
- ・細かい設定が可能
- ・バッチによる自動処理
- ・プログラミングが容易 ( fortran, c, perl, ... )

ls	list files
cd	change directory
pwd	path name of the current working directory
cp	copy files
rm	remove files
mv	move files
mkdir	make directory
rmdir	remove directory
ln	link file
cat	concatenate files
more	
less	
echo	
man	manual document
su	super user
chmod	change permission mode
vi	visual text editor

# プロセス・シミュレーション

SUPREM      Stanford University Process Emulation Module

## 2次元シミュレーション

diffuse	・拡散 ・酸化
implant	・イオン・インプラ
deposit	・デポジッション
etch	・エッチング

# プロセス・シミュレーションの例

File: ldd.s4

```

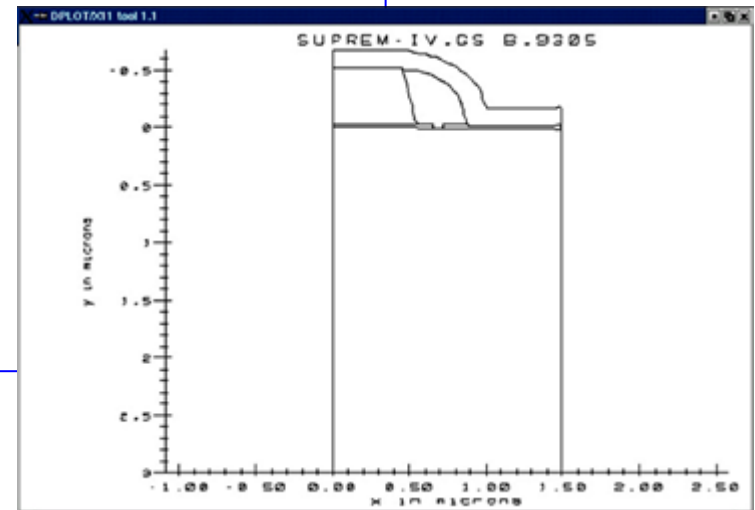
line x loc=0.0 tag=lft spacing=0.25
line x loc=0.45 spacing=0.03
line x loc=0.75 spacing=0.03
line x loc=1.4 spacing=0.25
line x loc=1.5 tag=rht spacing=0.25
line y loc=0.0 tag=top spacing=0.01
line y loc=0.1 spacing=0.01
line y loc=0.25 spacing=0.05
line y loc=3.0 tag=bot
region silicon xlo=lft xhi=rht ylo=top yhi=bot
bound exposed xlo=lft xhi=rht ylo=top yhi=top
bound backside xlo=lft xhi=rht ylo=bot yhi=bot
init boron conc=1.0e16
deposit oxide thick=0.025
implant boron dose=1.0e12 energy=15.0
deposit poly thick=0.500 div=10 phos conc=1.0e19
diff time=10 temp=1000
etch poly right p1.x=0.55 p1.y=-0.020 p2.x=0.45 p2.y=-0.55
diffuse time=30.0 temp=950
implant phos dose=1.0e13 energy=50.0
deposit oxide thick=0.400 spac=0.05
etch dry oxide thick=0.420
method vert fermi grid.ox=0.0
diffuse time=30 temp=950 dry
implant ars dose=5.0e15 energy=80.0
deposit oxide thick=0.15 space=0.03
diffuse time=20 temp=950
plot.2d bound

```

実行

\$ **suprem ldd.s4**

← ファイル名



# デバイス・シミュレーション

PISCES      Poison and Continuity Equation Solver

## Drift-Diffusion モデル

Poisson 方程式       $\vec{\nabla} \cdot (\epsilon \vec{\nabla} \psi) = -\rho$

キャリア連続の式       $\frac{\partial n}{\partial t} = \frac{1}{q} \vec{\nabla} \cdot \vec{J}_n + G_n - R_n$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \vec{\nabla} \cdot \vec{J}_p + G_p - R_p$$

$G$  : 電子・正孔対形成  
 $R$  : 電子・正孔再結合

輸送方程式       $\vec{J}_n = -q \mu_n n \vec{\nabla} \phi_n$

$$\vec{J}_p = -q \mu_p p \vec{\nabla} \phi_p$$

$\phi_n, \phi_p$  : Imref

$$n = n_i e^{\frac{q}{kT}(\psi - \phi_n)} \quad p = n_i e^{-\frac{q}{kT}(\psi - \phi_p)}$$

# デバイス・シミュレーションの例

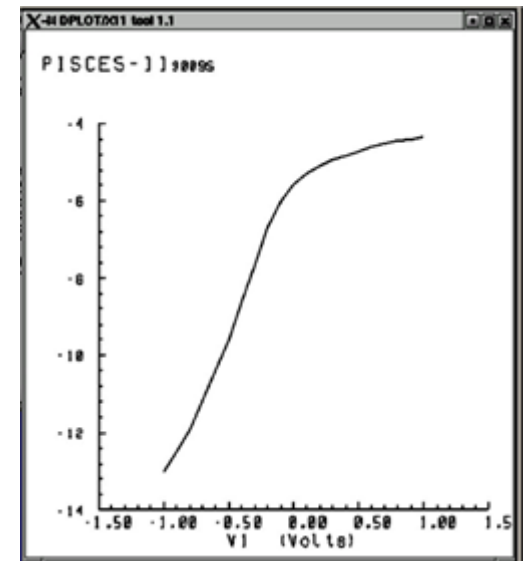
4-13

File: mosfet.p2

```
Title Short Channel (L=1.5 um) MOSFET
mesh      rectangular nx=38 ny=23
x.m       n=1      l=0      r=1
x.m       n=4      l=0.50   r=.7
x.m       n=8      l=0.70   r=.75
x.m       n=15     l=0.95   r=1.25
x.m       n=24     l=2.05   r=1.0
x.m       n=31     l=2.30   r=.8
x.m       n=35     l=2.50   r=1.33
x.m       n=38     l=3.00   r=1.40
y.m       n=1      l=-.035  r=1
y.m       n=4      l=0      r=1
y.m       n=9      l=0.10   r=1.25
y.m       n=16     l=0.50   r=1.15
y.m       n=23     l=2.0    r=1.25
region    num=1 ix.l=1  ix.h=38 iy.l=1  iy.h=4 oxide
region    num=2 ix.l=1  ix.h=38 iy.l=4  iy.h=23 silicon
elec      num=1   ix.l=4   ix.h=35 iy.l=1   iy.h=1
elec      num=2   ix.l=1   ix.h=38 iy.l=23  iy.h=23
elec      num=3   ix.l=1   ix.h=3  iy.l=4   iy.h=4
elec      num=4   ix.l=36  ix.h=38 iy.l=4   iy.h=4
dop       reg=2  unif  conc=1e15 p.type
dop       reg=2  n.type gaussian x.right=.5 ratio.lateral=.8
+        conc=2e19 junction=.5
dop       reg=2  n.type gaussian x.left=2.5 ratio.lateral=.8
+        conc=2e19 junction=.5
$
symb      gummel carr=1 electrons
method    iccg damped
mater      num=2 g.surf=0.75
contact   num=1 n.poly
models    conmob temp=300 fldmob print
solve     initial
solve     v4=0.5 vstep=0.5 nsteps=4 electrode=4
log       outf=log
solve     v4=3
symb      newton carriers=1 electrons
method    itlimit=50 autonr
solve     v1=-1.0 vstep=0.1 nsteps=20 electrode=1
$options  plotdev=stout
plot.ld   x.axis=v1 y.axis=i4 log
end
```

実行

\$ pisces2 mosfet.p2



# プロセス・シミュレーションと デバイス・シミュレーションとの連携

File: process.s4

```
#mesh
line x loc=0.0 tag=lft spacing=0.25
line x loc=0.45          spacing=0.03
line x loc=0.75          spacing=0.03
line x loc=1.4           spacing=0.25
line x loc=1.5 tag=rht spacing=0.25
#
line y loc=0.0 tag=top spacing=0.01
line y loc=0.1          spacing=0.01
line y loc=0.25         spacing=0.05
line y loc=3.0 tag=bot
#
```

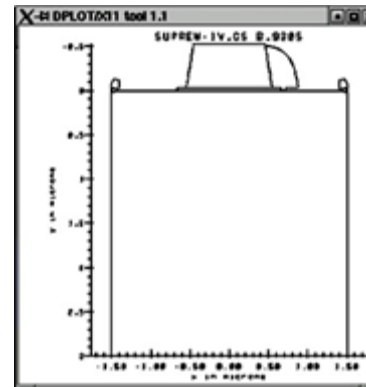
途中省略

```
#do the final anneal
diffuse time=20 temp=950
#put down the aluminum
deposit alum thick=0.1
#reflect the structure
struct mirror left
#etch off aluminum
etch alum start x=-1.4 y=0.1
etch continue  x= 1.4 y=0.1
etch continue  x= 1.4 y=-0.7
etch done      x=-1.4 y=-0.7
plot.2d boundary
#save it in Pisce format
struct pisc=ldd.mesh
```

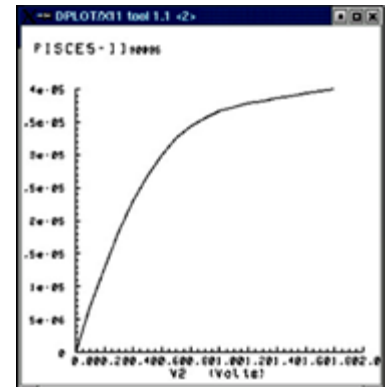
File: mosfet.p2

```
Title Short Channel (L=1.5 um) MOSFET
mesh infile=ldd.mesh geom
symb gummel carr=1 electrons
method iccg damped
mater num=4 g.surf=0.75
contact num=1 n.poly
models conmob temp=300 fldmob print
solve initial
solve v1=0.6 vstep=0.6 nsteps=2 electrode=1
log outf=log
solve v1=1.8
symb newton carriers=1 electrons
method itlimit=50 autonr
solve v2=0.1 vstep=0.1 nsteps=17 electrode=2
plot.1d x.axis=v2 y.axis=i2
end
```

\$ **suprem** process.s4



\$ **pisces2** mosfet.p2



# Drift-Diffusion モデルの課題

- impact ionization の収束を得るのが困難  
流体方程式にもってきたために関数の電界依存性が強い

$$G = \alpha_n J_n + \alpha_p J_p \quad \alpha_n = a_n e^{-b_n/E} \quad \alpha_p = a_p e^{-b_p/E}$$

- ホットエレクトロン注入等でモデル・パラメータ依存性が大きい  
実験を説明するというより実験に合うようにパラメータを選ぶ

- 量子効果をどう取り入れるか  
ゲート・トンネル電流  
チャンネルでの2次元閉じ込め } 最先端 MOSFET では重要

密度傾斜法(Density Gradient)でかなりの効果を入れることができる

$$\vec{J}_n = -q\mu_n n \vec{\nabla}(\phi_n - \Lambda) \quad \Lambda = -\frac{\gamma \hbar^2}{6m} \frac{\nabla^2 \sqrt{n}}{\sqrt{n}}$$

# ボルツマン方程式に基づくモンテカルロ・シミュレーション

Drift-Diffusion デバイス・シミュレーション

## 流体方程式

次の流体量を計算

電子濃度  $n$

電流密度  $\vec{j}$

電子温度  $T$

電子の局所平衡分布を仮定しており、ホットエレクトロンの関与する現象を記述するには不十分

ボルツマン方程式に基づくモンテカルロ・シミュレーション

## ボルツマン方程式

$$\frac{\partial}{\partial t} f(\vec{r}, \vec{p}, t) + \frac{\partial \varepsilon}{\partial \vec{p}} \cdot \frac{\partial}{\partial \vec{r}} f(\vec{r}, \vec{p}, t) + \vec{F} \cdot \frac{\partial}{\partial \vec{p}} f(\vec{r}, \vec{p}, t) = \left( \frac{\partial}{\partial t} f(\vec{r}, \vec{p}, t) \right)_{collision}$$

$\varepsilon(\vec{p})$  : 電子のエネルギー

電子分布関数  $f(\vec{r}, \vec{p}, t)$  を計算

$\vec{F}$  : 電子に働く力

$\vec{r}$  : 電子の位置

$\vec{p}$  : 電子の運動量

電子濃度等は分布関数のモーメントから求められる

$$n = \int f(\vec{r}, \vec{p}, t) d\vec{p}$$

$$\vec{j} = -q \int \frac{\partial \varepsilon}{\partial \vec{p}} \cdot f(\vec{r}, \vec{p}, t) d\vec{p}$$

$$T = n^{-1} \int \varepsilon(\vec{p}) \cdot f(\vec{r}, \vec{p}, t) d\vec{p}$$

異なるエネルギーの電子状態を独立して扱うことができ、ホットエレクトロン現象を正確に取り扱うことができる。



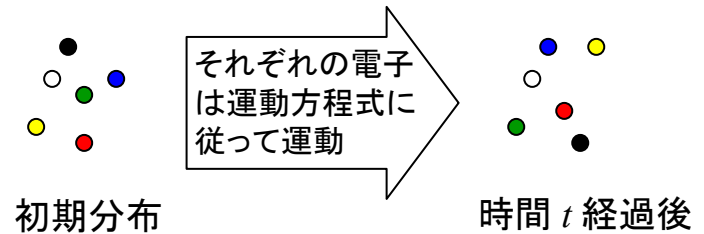
# モンテカルロ法

散乱を確率的に起こし、個々の電子の運動を追っていく

$$\frac{\partial}{\partial t} f(\vec{r}, \vec{p}, t) + \frac{\partial \varepsilon}{\partial \vec{p}} \cdot \frac{\partial}{\partial \vec{r}} f(\vec{r}, \vec{p}, t) + \vec{F} \cdot \frac{\partial}{\partial \vec{p}} f(\vec{r}, \vec{p}, t) = \left( \frac{\partial}{\partial t} f(\vec{r}, \vec{p}, t) \right)_{\text{collision}}$$

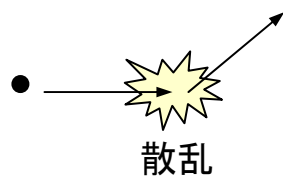
散乱が無ければ電子は運動方程式に従って動く

$$\frac{d\vec{r}}{dt} = \frac{\partial \varepsilon(\vec{p})}{\partial \vec{p}} \quad \frac{d\vec{p}}{dt} = \vec{F}(\vec{r}, t)$$



$$f(\vec{r}, \vec{p}, t) = \sum_i \delta(\vec{r} - \vec{r}_i(t)) \delta(\vec{p} - \vec{p}_i(t))$$

自由運動に散乱を確率的に起こす



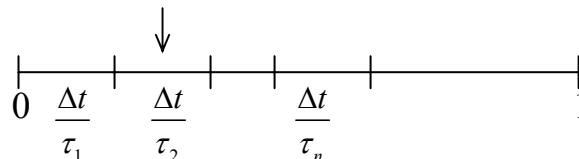
時間  $t$  と時間  $t + \Delta t$  の間で

$$\text{散乱を受けない確率 } P_0 = \exp\left(-\int_t^{t+\Delta t} \sum_i \frac{1}{\tau_i} dt\right) \cong 1 - \sum_i \frac{\Delta t}{\tau_i}$$

$$\text{散乱を受ける確率 } P_s = 1 - \exp\left(-\int_t^{t+\Delta t} \sum_i \frac{1}{\tau_i} dt\right) \cong \sum_i \frac{\Delta t}{\tau_i}$$

0~1の間の乱数  $r$  を発生

$r$  の値がこの範囲にあれば散乱2が起こるとする



# モンテカルロ法の例

4-18

```

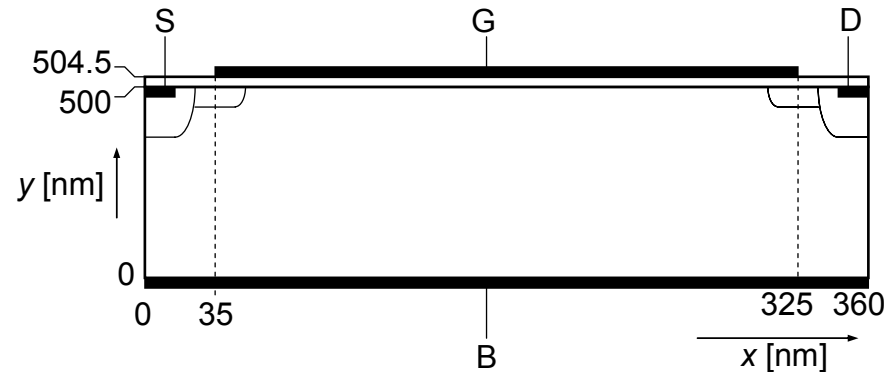
----- Mesh -----
&xmesh      nelmts=      99,
             xextnt= 3600.d-8,
             /
&yMesh      nintvl=      4,
             nelmts=      15,      10,      20,      4,
             dymin =   60.d-8,  12.d-8,   5.d-8,   8.d-8,
             yextnt= 4600.d-8, 300.d-8, 100.d-8, 45.d-8,
             grade =      -1,      -1,      0,      1, /
----- Device -----
&regn      mater='si', /
&regn      mater='sio', ylow=5000.d-8, /
             substrate doping; constant
&doping    dop=-6.5d15 /
             gaussian channel implant
&doping    shape='gauss', con=-7.0d17,
             xloc= 850.d-8, yloc=4300.d-8,
             xlen=20000.d-8, sigy=300.d-8, /
             gaussian source implant; shallow
&doping    shape='gauss', con=5.00d19,
             xloc= 0.d-8, yloc=5000.d-8,
             xlen=900.d-8, sigx= 50.d-8, sigy=200.d-8, /
             gaussian source implant; deep
&doping    shape='gauss', con=1.00d20,
             xloc= 0.d-8, yloc=4800.d-8,
             xlen=300.d-8, sigx=100.d-8, sigy=300.d-8, /
             gaussian drain implant; shallow
&doping    shape='gauss', con=5.00d19,
             xloc=3600.d-8, yloc=5000.d-8,
             xlen=900.d-8, sigx= 50.d-8, sigy=200.d-8, /
             gaussian drain implant; deep
&doping    shape='gauss', con=1.00d20,
             xloc=3600.d-8, yloc=4800.d-8,
             xlen=300.d-8, sigx=100.d-8, sigy=300.d-8, /
----- Run -----
&phys      rigor='mc', 'dde', lmp=t,
             impacti(1)=t, impacti(2)=f, /
&time      dtpois=5.0d-16, dtmc=5.0d-16, dtsc=5.0d-16, /
             stime=10.0d-12, /
&cntl      iofreq=200, logfile=f,
             levelp=1, levela=1, levelc=1,
             dbout='nmosfet', /
&ninitial  number =100000, /

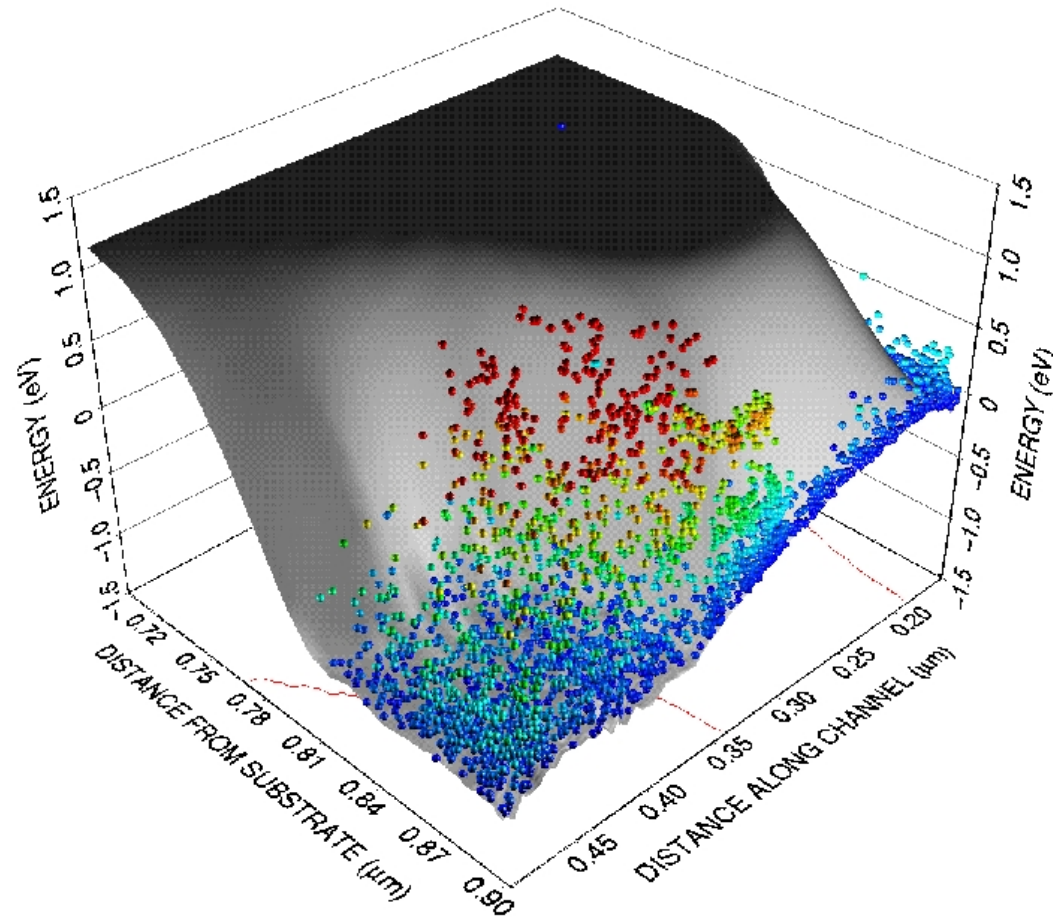
```

```

----- Boundary conditions -----
&defcon    number=1, xlow=0.d-8, xhgh=3600.d-8,
             yloc=0.d0, /
&defcon    number=2, xloc=0.d-8, xlen=435.d-8,
             sigx=73.d-8,
             yloc=5000.d-8, sigy=125.d-8, /
&defcon    number=2, xloc=0.d-8, xlen=0.d-8,
             sigx=218.d-8,
             yloc=4875.d-8, sigy=375.d-8, /
&defcon    number=3, xloc=3600.d-8, xlen=435.d-8,
             sigx=73.d-8,
             yloc=5000.d-8, sigy=125.d-8, /
&defcon    number=3, xloc=3600.d-8, xlen=0.d-8,
             sigx=218.d-8,
             yloc=4875.d-8, sigy=375.d-8, /
&defcon    number=4, xlow=350.d-8, xhgh=3250.d-8,
             yloc=5045.d-8, /
             back, source, drain, gate
&contcs    vappld= 0.00d0, 0.00d0, 3.00d0, 1.00d0, /
             phims='ec', /
&intfac    default=-1.0, reflect=0.5,
             xlow=0.d-8, xhgh=3600.d-8, yloc=5000.d-8, /
----- Statistical enhancement -----
&statbx    percent=f, /
&statbx    sfactor=10.d0,
             xlow=450.0d-08, xhgh=3150.0d-08,
             ylow=100.0d-08, yhgh=5000.0d-08, /

```





nMOSFETにおける電子のエネルギー分布  
IBM DAMOCLESによる  
(<http://www.research.ibm.com/DAMOCLES/>)

# モンテカルロ・シミュレーションの課題

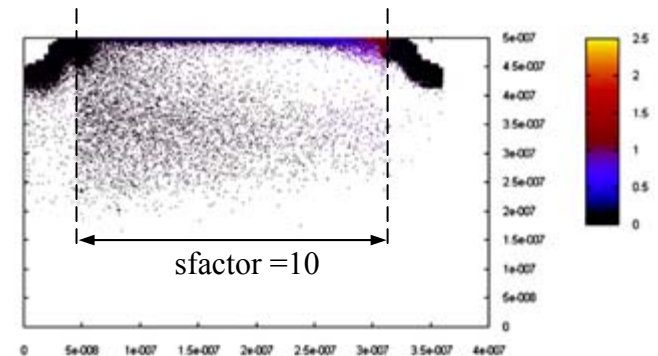
・散乱時間  $\sim 1\text{a sec}$  ( $10^{-15}$  秒)

⇒ 100,000 個の電子を用いて  $1\text{p sec}$  の過渡解析を行うのに 計算時間10 時間 (Pentium4 3GHz)

定常電流を求めるのは困難

・ソース・ドレイン・ゲートのように不純物濃度の多いところに多くの電子 チャンネルのような特性上重要な領域の電子数が少ない

⇒ statistical factor によりある程度  
解消



⇒ 計算機の更なる性能アップが必要

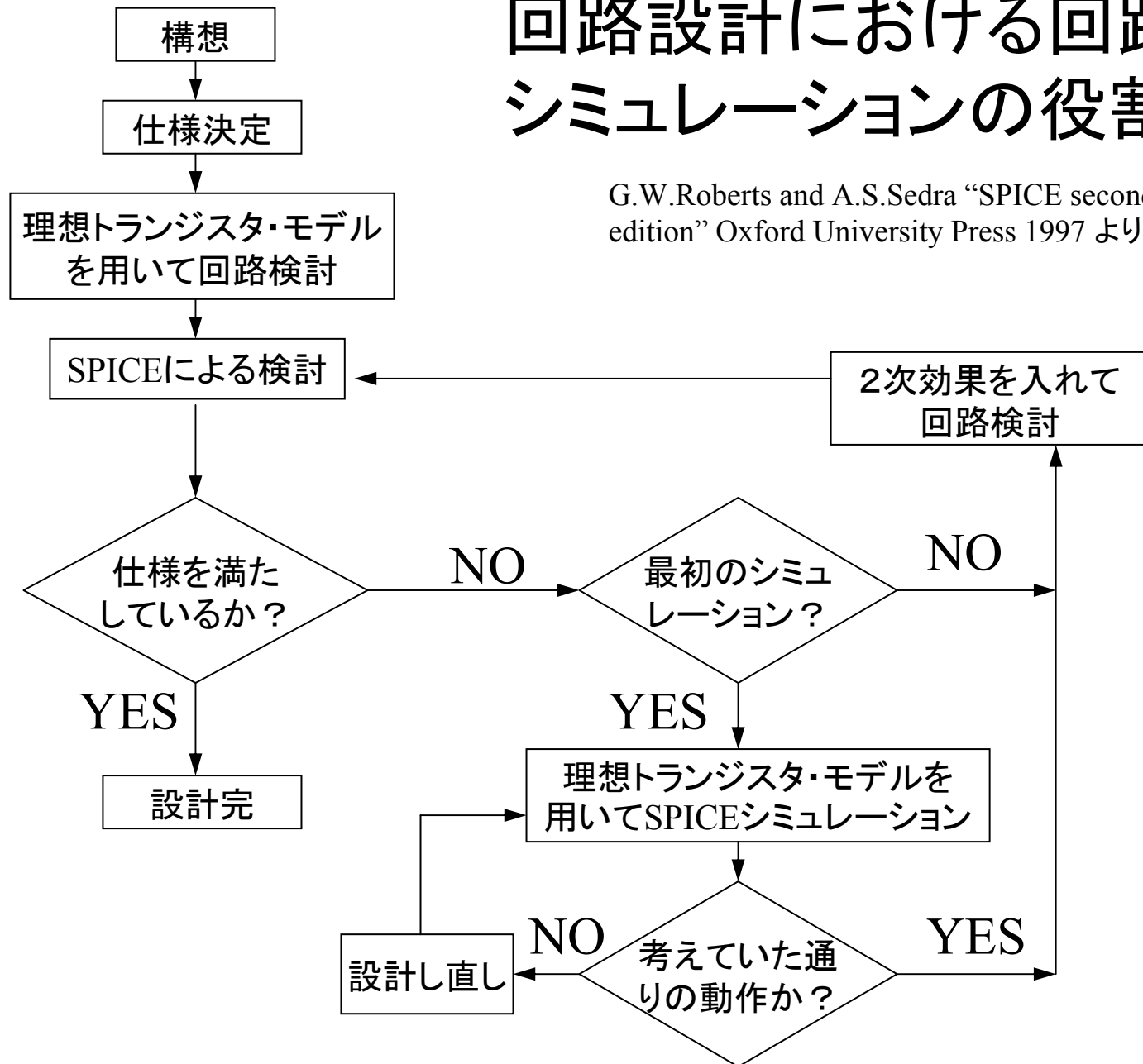
# SPICE

Simulation Program with Integrated-Circuit Emphasis

1970年代 University of California, Berkeley で開発  
回路シミュレータの標準

# 回路設計における回路シミュレーションの役割

G.W.Roberts and A.S.Sedra “SPICE second edition” Oxford University Press 1997 より

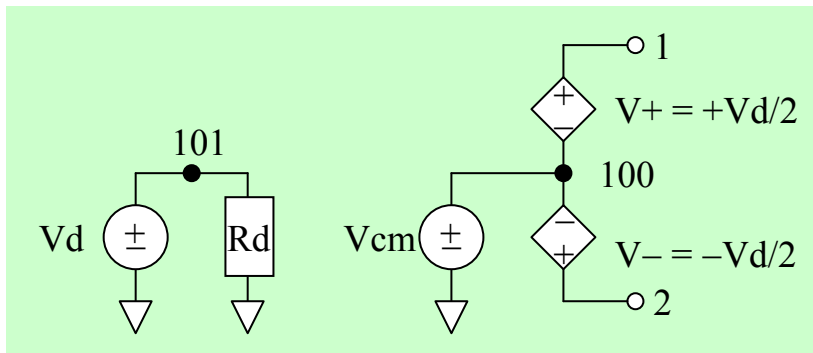
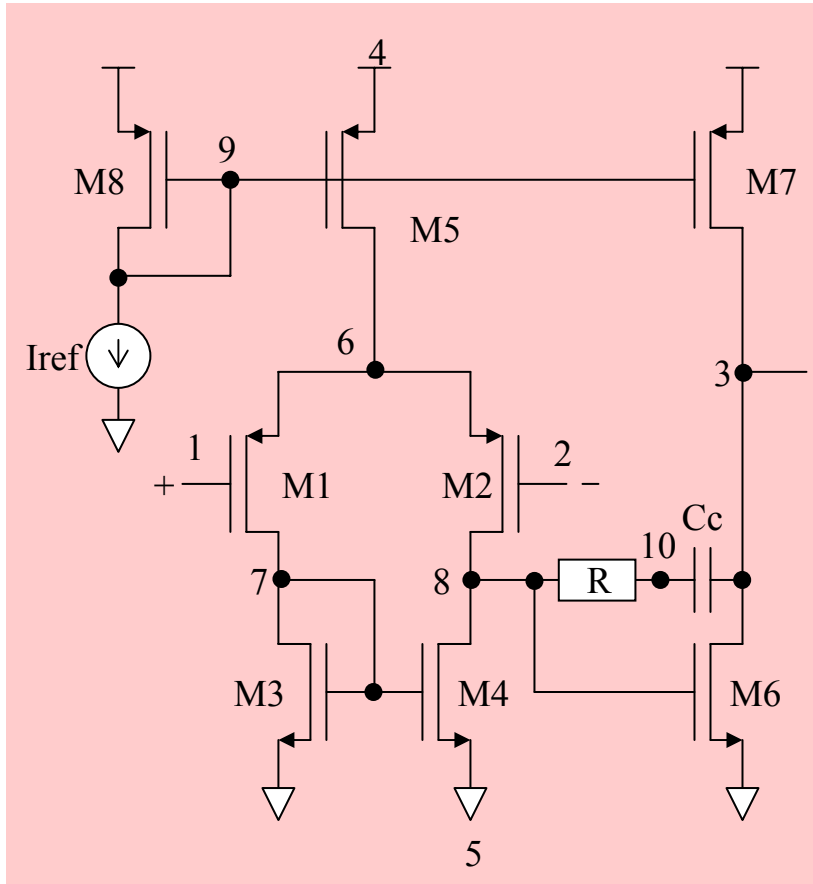


# SPICE 入力文

タイトル	←	最初の1行目はタイトル(回路名、設計者、日付等)を入れる 実際のシミュレーションでは無視される
回路記述		
電源・信号源		
素子記述		行の最初の文字が
モデル記述		* の時は注釈行となる
解析リクエスト		+ の時は継続行となる
出力リクエスト		
.END	←	最後の行は .END

大文字・小文字の区別は無い

# 例 CMOS opamp



```

A CMOS Operational Amplifier
** Circuit Description
* Power suppliers
Vdd 4 0 DC +5V
Vss 5 0 DC -5V
* Differential-mode signal level
Vd 101 0 DC 0V
Rd 101 0 1
EV+ 1 100 101 0 +0.5
EV- 2 100 101 0 -0.5
* common-mode signal level
Vcm 100 0 DC 0V
* front-end stage
M1 7 1 6 4 pmostr L=8u W=120u
M2 8 2 6 4 pmostr L=8u W=120u
M3 7 7 5 5 nmostr L=10u W=50u
M4 8 7 5 5 nmostr L=10u W=50u
M5 6 9 4 4 pmostr L=10u W=150u
* second gain stage
M6 3 8 5 5 nmostr L=10u W=100u
M7 3 9 4 4 pmostr L=10u W=100u
* current source biasing stage
M8 9 9 4 4 pmostr L=10u W=150u
Iref 9 5 DC 25uA
* compensation network
R 8 10 10k
Cc 10 3 10pF
* model statements
.MODEL nmostr nmos level=2 vto= 1 tox=8.5e-8
.MODEL pmostr pmos level=2 vto=-1 tox=8.5e-8
** Analysis Requests
.DC Vd -4mV +4mV 100uV
** Output Requests
.PLOT DC V(3)
.end
    
```



# SPICEの実行

4-25

```
$ spice3
Program: Spice, version: 3f5
Date built: Sun Dec 26 09:36:30 JST 2004

Type "help" for more information, "quit" to leave.

Spice 1 -> source cmosopamp.cir
Circuit: A CMOS Operational Amplifier

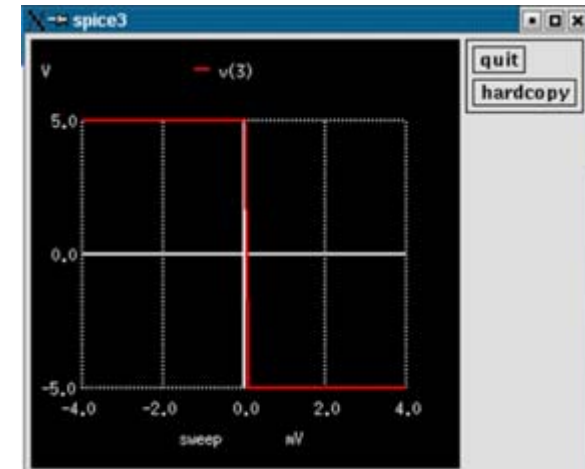
Spice 2 -> run
Spice 3 -> display
Here are the vectors currently active:

Title: A CMOS Operational Amplifier
Name: dcl (DC transfer characteristic)
Date: Tue Mar 29 09:43:01 2005

V(1)          : voltage, real, 81 long
V(10)         : voltage, real, 81 long
V(100)        : voltage, real, 81 long
V(101)        : voltage, real, 81 long
V(2)          : voltage, real, 81 long
V(3)          : voltage, real, 81 long
V(4)          : voltage, real, 81 long
V(5)          : voltage, real, 81 long
V(6)          : voltage, real, 81 long
V(7)          : voltage, real, 81 long
V(8)          : voltage, real, 81 long
V(9)          : voltage, real, 81 long
ev+#branch    : current, real, 81 long
ev-#branch    : current, real, 81 long
sweep         : voltage, real, 81 long [default scale]
vcm#branch    : current, real, 81 long
-- hit return for more, ? for help --

vd#branch     : current, real, 81 long
vdd#branch    : current, real, 81 long
vss#branch    : current, real, 81 long

Spice 4 -> plot v(3)
Spice 5 -> write cmosopamp.raw
Spice 6 -> quit
Spice-3f5 done
$
```



# 回路エレメント

## 頭文字でエレメントの種類を表す

C	キャパシタ	Cname n+ n- value
D	ダイオード	Dname A C model_name
E	電圧制御電圧源	Ename n+ n- nc+ nc- value
F	電流制御電流源	Fname n+ n- vname value
G	電圧制御電流源	Gname n+ n- nc+ nc- value
H	電流制御電圧源	Hname n+ n- vname value
I	電流源	Iname n+ n- signal
J	JFET	Jname D G S model_name
K	結合インダクタ	Kname Lname Lname value
L	インダクタ	Lname n+ n- value
M	MOSFET	Mname D G S B model_name
O	損失伝送路	Oname n1+ n1- n2+ n2- model_name
Q	バイポーラトランジスタ	Qname C B E model_name
R	抵抗	Rname n+ n- value
T	無損失伝送路	Tname n1+ n1- n2+ n2- Z0=value
V	電圧源	Vname n+ n- signal
X	サブ回路	Xname n1 n2 .. sub_name

## 例

```

Cc 8 10 10pF
DCLMP 3 7 DMOD
EV+ 1 100 101 0 +0.5
F1 13 5 VSENS 5
G1 2 0 5 0 0.1MMHO
HX 5 17 V2 0.5K
Iref 9 5 DC 25uA
J1 7 2 3 JM1
K43 LAA LBB 0.99
LAA 42 69 1uH
M1 7 1 6 4 pmostr L=8u W=120u
O23 1 0 2 0 LOSSYMOD
Q23 10 24 13 QMOD
Rd 101 0 1

Vdd 4 0 DC +5V
X1 2 4 17 3 1 MULTI

```

n : ノード (0は常にグランド =0V)

A : anode      C : cathode

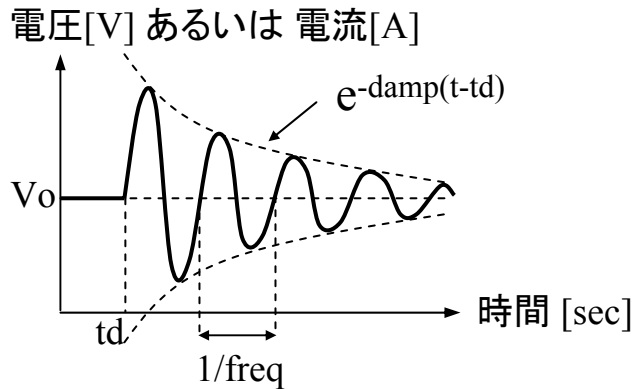
D : drain      G : gate      S : source      B : body (sub)

C : collector      B : base      E : emitter

DC value

AC magnitude phase\_degrees

SIN( Vo Va freq td damp )

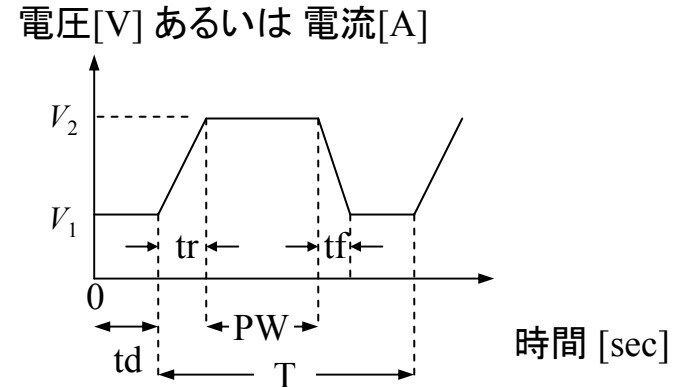


$$V = V_o + V_a \cdot e^{-damp(t-td)} \sin[2\pi freq(t-td)]$$

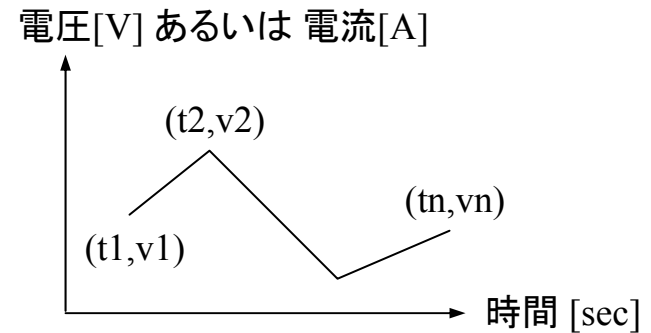
SFFM( Vo Va freq mdi fs )

$$V = V_o + V_a \cdot \sin[2\pi freq \cdot t + mdi \cdot \sin(2\pi fs \cdot t)]$$

PULSE( V1 V2 td tr tf PW T )



PWL( t1 v1 t2 v2 ... tn vn )



例) vdd dd 0 dc 5  
vclk clk 0 dc 0 pulse(0 5 40n 2n 2n 40n 80n)

# 解析リクエスト

## 制御変数

```
.OPTIONS
```

例) `.OPTIONS ABSTOL=1e-15` (電流精度を1aAに変更。デフォルトは1pA)  
`.OPTIONS NOMOD` (モデル・パラメータの出力をしない)

## 初期値

```
.IC V(node1)=value V(node2)=value ....
```

## 直流掃引

```
.DC source_name start_value stop_value step_value
```

2重にバイアスを掃引することができる

例) `.DC VDS 0 10 .5 VGS 0 5 1`  
VGS=0, 1, 2, 3, 4, 5Vに対しそれぞれVDSを0Vから10Vまで0.5V  
ステップで掃引

## 交流周波数応答

```
.AC DEC points_per_decade freq_start freq_stop  
.AC OCT points_per_octave freq_start freq_stop  
.AC LIN total_points freq_start freq_stop
```

## 過渡解析

```
.TR time_step time_stop
```

# 出力リクエスト

```
.PRINT DC output_variables  
.PRINT AC output_variables  
.PRINT TRAN output_variables  
  
.PLOT DC output_variables  
.PLOT AC output_variables  
.PLOT TRAN output_variables
```

output\_variables

ノード電圧	V(node)
2ノード間電圧	V(node1, node2)
電圧源を流れる電流	I(Vname)

# ブロック化

- ・1つの回路の中で同じ部分回路を2回以上使う `.SUBCKT`
- ・同じ部分回路を2つ以上の回路で使う `.include`

```
.include CMOSopamp1
```

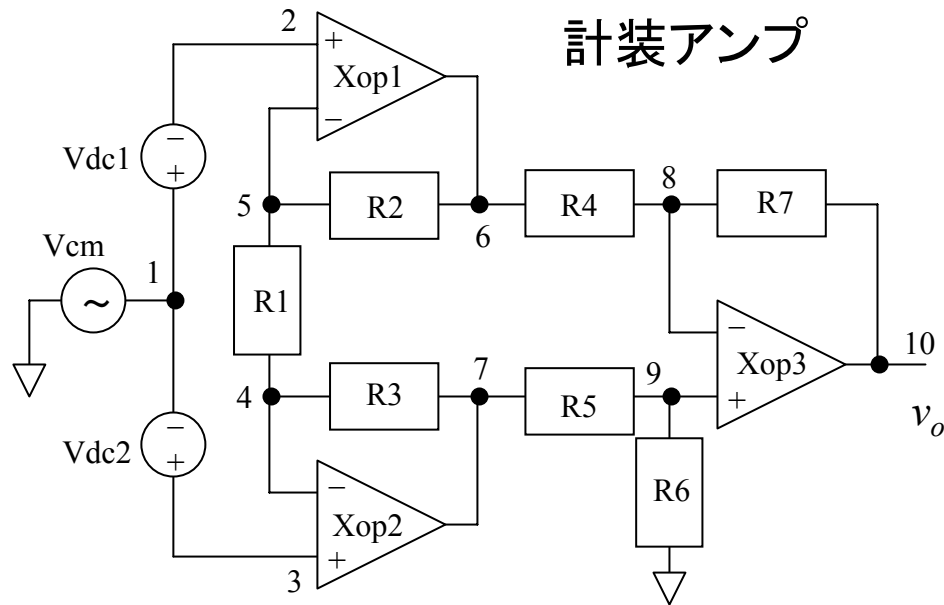
```
Xop1 2 5 6 cmos_opamp  
Xop2 3 4 7 cmos_opamp  
Xop3 9 8 10 cmos_opamp
```

File : CMOSopamp1

```
.SUBCKT cmos_opamp 1 2 3  
*connection +ve----| | |  
*           -ve-----| |  
*           output-----|  
*A CMOS Operational Amplifier
```

```
.ENDS
```

# ブロック化の例

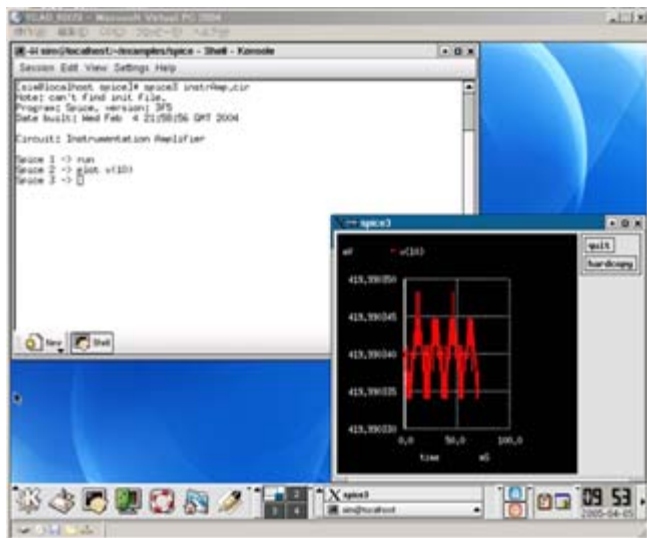


File: instrAmp.cir

```
Instrumentation Amplifier
.include opamp.mod
* signal sources
Vcm 1 0 0 SIN(0 25V 60Hz)
Vdc1 1 2 DC 10mV
Vdc2 3 1 DC 10mV
* instrumentation amplifier
Xop1 2 5 6 opamp
Xop2 3 4 7 opamp
Xop3 9 8 10 opamp
R1 5 4 10k
R2 5 6 100k
R3 4 7 100k
R4 6 8 10k
R5 7 9 10k
R6 9 0 10k
R7 8 10 10k
* analysis request
.tran 0.1ms 70ms 0 0.1ms
* output request
.print tran v(2) v(3) v(10)
.end
```

File: opamp.mod

```
.SUBCKT opamp 1 2 3
*connection +ve-| | |
*           -ve---| |
*           output--|
*ideal Operational Amplifier
Iopen1 1 0 0A
Iopen2 2 0 0A
Eopamp 3 0 1 2 1e6
.ENDS
```



# モデル

商用のSPICEではきめ細かくモデルをサポート

例: SILVACO SmartSpice の MOSFET モデル

LEVEL=1	MOSFET Model
LEVEL=2	MOSFET Model
LEVEL=3	MOSFET Model
LEVEL=4	MOSFET Model (BSIM1)
LEVEL=7	MOSFET Model (BSIM3 Version 2.0)
LEVEL=81	Berkeley BSIM3 MOSFET model
LEVEL=8	SmartSpice MOSFET Model BSIM3
LEVEL=14	Berkeley MOSFET Model BSIM4
LEVEL=9	Philips MOSFET Model
LEVEL=30	Philips MOSFET Model
LEVEL=31	Philips MOSFET Model
LEVEL=18	Deep Sub-Micron MOSFET Model
LEVEL=44	EKV MOSFET Model
LEVEL=88	High Voltage MOSFET Model
LEVEL=20	Honeywell SOI Model (HSOI)
LEVEL=21	Florida SOI Model
LEVEL=22	Florida SOI Model
LEVEL=23	MOSFET SOI Model (BSIM3SOI)
LEVEL=24	STAG(Southampton Thermal AnaloGue) SOI Model
LEVEL=25	MOSFET SOI Model (BSIM3SOIv1)
LEVEL=26	MOSFET BSIM3SOIFD version 2 Model
LEVEL=27	MOSFET BSIM3SOIDD version 2
LEVEL=29	MOSFET BSIM3SOIPD version 2
LEVEL=32	CEA/LETI SOI model (LETISOI)
LEVEL=15	Leroux's Amorphous-Si TFT Model
LEVEL=16	Berkeley Polysilicon TFT Model
LEVEL=35	RPI Amorphous-Si TFT Model
LEVEL=36	RPI polysilicon TFT Model



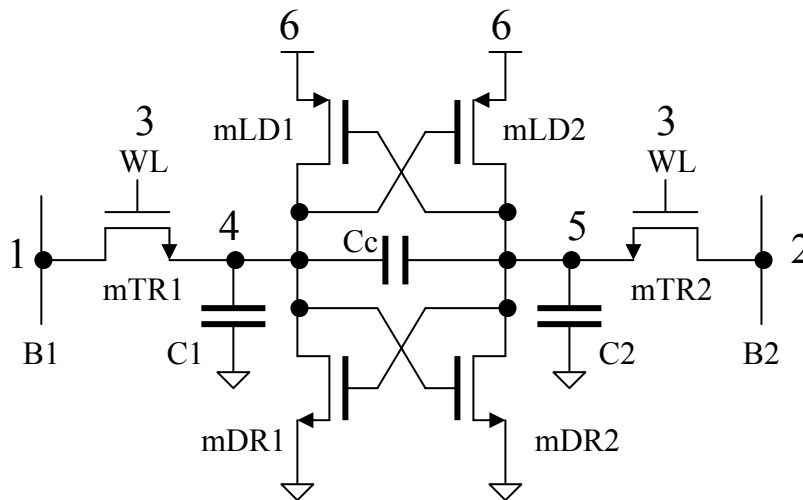
# モデルが無い場合 - Mixed Mode

新しい素子を発明 →

モデルが無いので回路シミュレーションができない?!

回路シミュレーションとデバイス・シミュレーションを結合

例) Double-gate FD SOI-MOSFET を用いたSRAMセル



File: sram.cir

```

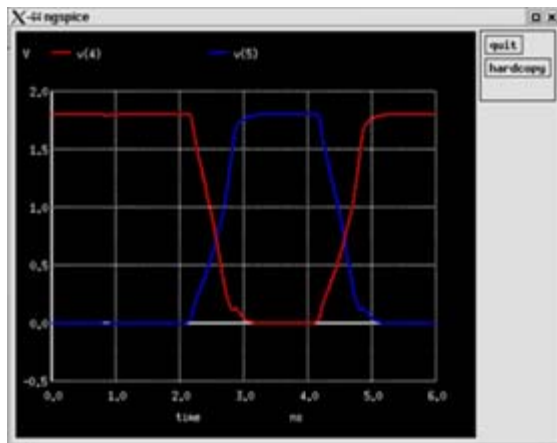
Double Gate SOI 6T-SRAM
mLD1 6 5 4 5 acmp w=0.3u
mLD2 6 4 5 4 acmp w=0.3u
mTR1 1 3 4 3 acmn w=0.2u
mTR2 2 3 5 3 acmn w=0.2u
mDR1 4 5 0 5 acmn w=0.13u
mDR2 5 4 0 4 acmn w=0.13u
Cc 4 5 5fF
C1 4 0 1fF
C2 5 0 1fF
Vdd 6 0 1.8
VB1 1 0 1.8 PWL(0n 1.8 .9n 1.8 1n 0 4n 0 4.1n 1.8
+ 4.9n 1.8 5n 0)
VB2 2 0 0.0 PWL(0n 0 2n 0 2.1n 1.8 2.9n 1.8 3n 0)
VWL 3 0 1.8 PWL(0n 1.8 .8n 1.8 .9n 0 2.1n 0
+ 2.2n 1.8 2.8n 1.8 2.9n 0 4.1n 0
+ 4.2n 1.8 4.8n 1.8 4.9n 0)
.include dgmosN.m
.include dgmosP.m
.options acct bypass=1 method=gear
.control
tran 1n 6n
plot v(4) v(5)
.endc
.end
    
```

File: dgmosN.m

```

.model acmn numos
+
+output all.debug mesh doping
+x.m n=1 l=0
+x.m n=2 l=0.1
+x.m n=12 l=0.5
+x.m n=13 l=0.6
+y.m n=1 l=-0.02
+y.m n=2 l=0.0
+y.m n=12 l=0.1
+y.m n=13 l=0.12
+
+region num=1 material=1 ix.l=1 ix.h=13 iy.l=1 iy.h=2
+region num=2 material=2 ix.l=1 ix.h=13 iy.l=2 iy.h=12
+region num=3 material=3 ix.l=1 ix.h=13 iy.l=12 iy.h=13
+ material num=1 oxide
+ material num=2 silicon
+ material num=3 oxide
+
+elec num=1 ix.l=1 ix.h=1 iy.l=2 iy.h=12
+elec num=2 ix.l=1 ix.h=13 iy.l=1 iy.h=1
+elec num=3 ix.l=13 ix.h=13 iy.l=2 iy.h=12
+elec num=4 ix.l=1 ix.h=13 iy.l=13 iy.h=13
+
+dop uniform p.type conc=1e+16 x.l=0.0 x.h=0.6 y.l=0.0 y.h=0.1
+dop uniform n.type conc=2.e20 x.l=0.0 x.h=0.1 y.l=0.0 y.h=0.1
+dop uniform n.type conc=2.e20 x.l=0.5 x.h=0.6 y.l=0.0 y.h=0.1
+
+contact num=2 workfunction=4.9
+contact num=4 workfunction=4.9
+ models concmob fieldmob tempmob bgn srh conctau
+ method ac=direct
    
```

\$ cider sram.cir



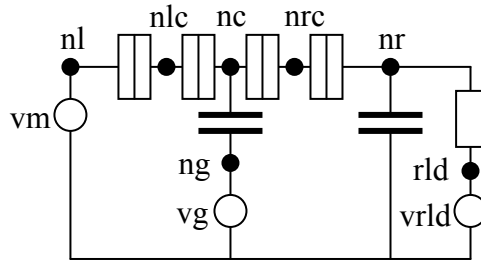
## 更に、新しいデバイス・シミュレーションで動作解析したい場合

例) Single-electron デバイスを確率過程モデルでシミュレーションしたい

⇒ SPICE にユーザー定義素子を追加することにより可能

CAMSET (Circuit Analysis program including a Model of Single Electron Tunneling)

M.Kirihara, K.Nakazato, and M.Wagner, *Jpn.J.Appl.Phys.*, vol.38, p.2028, 1999



File: turnstile.cir

```
asymmetric turnstile + load capacitor
rload nr rld 1e+25
cload nr 0 1.6e-17
Yturnstile.cam 3 0 nl nr ng turnstile
vrlld rld 0 0
vm nl 0 dc 0 pwl(3e-7 0 4e-7 0.1)
vg ng 0 dc 0 pulse(0 0.5 1e-6
+ 0.499999e-6 0.499999e-6 0.00001e-6 1e-6)
.tran 1e-10 10e-6 0 2e-9
*.option temp=-273.05 rawstore=100
.option temp=-273.05
.save v(nr) v(ng)
.model turnstile y()
.end
```

File: turnstile.cam

```
connect nl,nr,ng
{Temp = 0.1}
Vdd = 0.1
Cg = 2.67e-20 / Vdd
Cl = Cg * 3
Cr = Cg * 5
Rtr= 1e6
Rtl= 1e6
{-----}
put t0 nl, nlc, Cl, Rtl
put t1 nlc, nc, Cl, Rtl
put t2 nc, nrc, Cr, Rtr
put t3 nrc, nr, Cr, Rtr
put cg nc, ng, Cg
{-----}
do montecarlo
```

\$ cspice3 turnstile.cir

