

1群(信号・システム) - 7編(電子回路)

7章 電子回路の計算機シミュレーション

(執筆: 渡邊貴之)[2008年12月受領]

概要

電子回路を設計する場合、設計に誤りがなく期待通りの性能で動作するかどうかを確かめるには、どのような方法があるだろうか。電子計算機が普及する以前は、実際に回路をブレッドボード上に試作して、その回路特性をオシロスコープなどで測定することで、初めて回路の動作を検証することができた。しかし、このような検証の方法では、万が一設計に問題があり、正常に動作しなかった場合、試作に要した時間と費用が無駄となり多大な手戻りが発生する。そこで、現在では計算機上のソフトウェアを用いて回路の動作を模擬(シミュレート)することで、試作前に回路の動作検証を行うことが一般的である。この際に使うソフトウェアを回路シミュレータと呼ぶ。

回路シミュレータを用いて電子回路の計算機シミュレーションを行う場合、回路記述、回路解析、解析結果出力という手続きを踏む。回路記述では、解析対象の回路がどのような素子(受動素子、能動素子)で構成されており、素子同士がどのように結線されているのかを、ネットリストと呼ばれるテキストファイルに記述する。次に、検証したい回路特性に合わせて、直流解析、小信号交流解析、過渡解析などの回路解析を実行する。解析結果出力では、所望の回路特性をグラフや表の形式で出力することができる。

回路シミュレータでは、抵抗やコンデンサ、トランジスタなどの基本素子ごとに、素子の端子電圧と素子を流れる電流の関係を、代数方程式や微分方程式の形式で表したデバイスモデルを内蔵している。回路解析では、これらのデバイスモデルを組み合わせた連立方程式を構成し、素子の端子電圧や素子を流れる電流を未知変数として解くことで回路特性を求めている。従って、デバイスモデルの良し悪しが、回路シミュレータの解析精度を大きく左右する。また、演算増幅器などの機能モジュールは、多数の受動・能動素子から構成されるが、これらのモジュールを複数含む回路は大規模となり、解析に要する時間が長大となる。そこで、機能モジュールの入出力特性を近似的に再現するような、少数の素子を組み合わせで構成を簡略化した等価回路を用いて回路シミュレーションを行うことで、解析時間を短縮することができる。このような等価回路をマクロモデルと呼ぶ。

【本章の構成】

本章では、電子回路の計算機シミュレーションについて、以下の二つの節に分けて述べる。まず、回路シミュレータの基礎(7-1節)において、回路シミュレータを用いた解析フローや機能モジュールのモデル化について述べる。次に、回路シミュレータの計算アルゴリズム(7-2節)では、回路シミュレータの内部で用いられているシミュレーションアルゴリズムやデバイスモデルについて述べる。

1群-7編-7章

7-1 回路シミュレータの基礎

(執筆者：渡邊貴之)[2008年12月受領]

回路シミュレータは、1960年代の米国において、主に軍事用の電子回路の動作検証を行うために開発されたのがその始まりである。1970年代になると、半導体集積回路技術の進歩により回路規模が増大し、その動作検証は手計算では到底対応することが難しく、回路シミュレータの必要性が一層高まった。そのため、企業や大学が回路シミュレータを競って開発するようになった。

とりわけ、1972年に米国カリフォルニア大学バークレー校(U.C. Berkeley)で開発されたSPICE(Simulation Program with Integrated Circuit Emphasis)と名付けられた回路シミュレータは、それまでに開発された数多くの回路シミュレータの優れた計算アルゴリズムを取り入れるだけでなく、使い勝手のよいユーザインタフェースを備えており、ソースコードを無償で公開したことにより、世界的な標準シミュレータとして広く使われるようになった¹⁾。また、多くの企業がSPICEと互換性のある商用回路シミュレータを販売しており、これらをひと括りにSPICE系回路シミュレータと呼んでいる。

7-1-1 回路シミュレーションの解析フロー

回路シミュレーションの解析フローを図7-1に示す。実際の回路設計では、回路記述の前に回路の仕様決めや回路構成の検討を行い、解析結果出力の後に仕様を満たしているかを評価し、満たしていなければ回路構成の検討に戻る手続きを踏む。

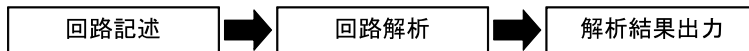


図7-1 回路シミュレーションの解析フロー

(1) 回路記述

解析対象回路内の素子の結線情報と物理パラメータをネットリスト(netlist)と呼ばれるテキストファイルに記述する。SPICE系回路シミュレータであれば、ネットリストの書式はほぼ共通しているため、一度文法を覚えてしまえば数多くの回路シミュレータを使いこなすことができる²⁾。簡単なネットリストの例を図7-2に示す。

図7-2(b)のR1, C1, V1で始まる各行が、素子の情報を記述した行である。例えば、抵抗素子の定義については次のような構文で記述する。

【抵抗素子の定義】 Rxxxxxxx 節点名1 節点名2 抵抗値(Ω)

ここで、行の最初に素子名を「Rxxxxxxx」という形式で記述している。先頭の「R」は、この素子が抵抗であることを表しており、そのほかにキャパシタならば「C」、電圧源ならば「V」などの決まりがある。表7-1に、代表的な基本素子の一覧を示す。次の「xxxxxxx」に

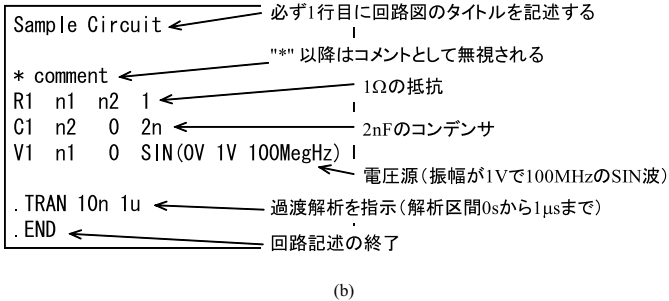
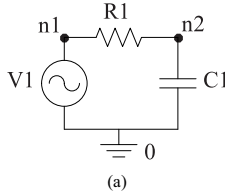


図 7・2 簡単なネットリストの例

は任意の英数文字列を指定できる。回路内に抵抗素子が複数ある場合には、それぞれの素子に異なる名前を付ける必要がある(R1, R2, ROUT, RIN, R3A2 など)。

次に、節点名を節点の数だけ空白で区切って記述する。節点(Node)とは、素子の端子を結ぶ接続ポイントのことである。節点名についても任意の英数文字列を使ってほかの節点名と重複しないように命名する。SPICE系回路シミュレータでは、全節点における電圧値と(一部の)素子を通る電流値を計算する。ここで、電圧とは2点間の電位の差で定義される相対的な量である。従って、節点のうち少なくとも一つを基準電位(接地、アース、GND)として指定する必要がある。各節点の電圧は、基準電位との電位差で定義される。ネットリストでは、素子の端子が基準電位に接続されている場合、節点名として「0」と記述する。最後に、素子のモデルパラメータ(抵抗の場合は抵抗値)を記述する。ただし、多数のモデルパラメータを指定する必要がある半導体素子や伝送線路では、素子定義の末尾にパラメータを直接記述することは非効率である。そのため、これらの素子では、.MODEL行によりモデル名を付けたパラメータセットを定義し、素子定義の末尾にはモデル名を記述する。

以上のように、テキストエディタを使って直接ネットリストを記述することも可能であるが、多くの回路シミュレータでは、GUI(Graphical User Interface)に基づくスクエマティック(回路図)エディタが用意されている。これらのエディタでは、ユーザが入力した回路図をネットリストに変換し、シミュレータに受け渡している。

表 7-1 代表的な基本素子

区分	記号	素子名	節点	パラメータ
受動素子	R	抵抗 (Resistor)	N+ N-	抵抗値 (Ω)
	C	キャパシタ (Capacitor)	N+ N-	容量値 (F)
	L	インダクタ (Inductor)	N+ N-	インダクタンス (H)
	K	相互インダクタ (Mutual Inductor)	L1 L2	結合係数
半導体素子	D	ダイオード (Diode)	N+ N-	モデル名など
	Q	バイポーラ接合型トランジスタ (BJT)	NC NB NE NS	モデル名など
	J	接合型電界効果トランジスタ (JFET)	ND NG NS	モデル名など
	M	MOS 型電界効果トランジスタ (MOSFET)	ND NG NS NB	モデル名など
伝送線路	T	無損失伝送線路 (Lossless Transmission Line)	N1 N2 N3 N4	特性インピーダンスなど
	U	有損失伝送線路 (Lossy Transmission Line)	N1 N2 N3 N4	モデル名
電源	V	独立電圧源 (Independent Voltage Source)	N+ N-	種別, パラメータ
	I	独立電流源 (Independent Current Source)	N+ N-	種別, パラメータ
	E	電圧制御電圧源 (Voltage Controlled V-Source)	N+ N- NC+ NC-	電圧ゲイン
	G	電圧制御電流源 (Voltage Controlled C-Source)	N+ N- NC+ NC-	トランスコンダクタンス
	H	電流制御電圧源 (Current Controlled V-Source)	N+ N- VNAME	トランスレジスタンス
	F	電流制御電流源 (Current Controlled C-Source)	N+ N- VNAME	電流ゲイン

(2) 回路解析

SPICE 系回路シミュレータでは、検証したい回路特性に応じて、様々な種類の解析機能が用意されている。解析機能の指定は、ネットリストの中で「.(ドット)」から始まる制御コマンドを記述し指定する。下記に、主な解析機能とその制御コマンドを紹介する。そのほかにも、極・ゼロ解析 (.PZ)、小信号歪み解析 (.DISTO)、感度解析 (.SENS) がある。

(a) 直流解析 (.DC, .OP)

ある節点に直流電源を接続したときの、各節点の電圧や素子に流れる電流を計算するのが直流解析である。従って、回路中のすべてのキャパシタは開放、インダクタは短絡して計算が行われる。単に動作点を求めたい場合は、.OP コマンドによる動作点解析を実行すればよい。一方、.DC コマンドを用いることにより、印加する直流電源の値をスイープ（ある一定

の増分で変化)して計算を繰り返し行うことができるため、一度の解析で回路の直流特性を求めることができる。

(b) 小信号交流解析 (.AC)

ある節点に交流電源を接続したときの、各節点の電圧や素子に流れる電流の周波数特性を計算する。この場合、回路中の非線形素子(ダイオード、トランジスタなど)については、動作点周りで線形化した小信号等価回路により計算が行われる。また、印加する交流電源の周波数をスイープさせて計算を繰り返し行うことができる。

(c) 過渡解析 (.TRAN)

ある節点に時間変化する電源を接続した時の、各節点の電圧や素子に流れる電流の過渡応答を、時刻を進めながら計算する。この場合、回路中の非線形素子(ダイオード、トランジスタなど)については、素子特性の非線形性を考慮した大信号等価回路により計算が行われる。

7-1-2 マクロモデル

大規模回路は様々な機能モジュールから構成される。代表的な機能モジュールとしては、数十のトランジスタから構成される演算増幅器がある。例として、図 7-3(a) に演算増幅器 $\mu A741$ の回路図を示す³⁾。一般に、R, L, C などの受動素子に比較して、トランジスタなどの半導体素子を含む回路の解析には多くの計算コストを必要とする。従って、演算増幅器を複数組み合わせた回路の解析には、さらに膨大な計算コストを要することになる。

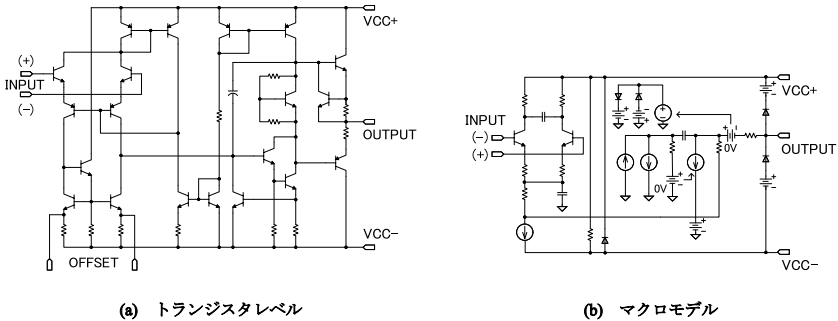


図 7-3 演算増幅器 $\mu A741$ の回路図

一方、このような機能モジュールを効率的に解析するために、モジュールの入出力のみに着目し、内部を少数の基本素子を組み合わせた等価な簡易モデルに置き換えたものがマクロモデルである。図 7-3(a) のマクロモデルとして提案されている等価回路を図 7-3(b) に示す。マクロモデルでは、電流制御電圧源などの従属電源を活用したモデル化が行われていることが分かる。SPICE 系回路シミュレータでは、電流制御電圧源・電流源を記述する場合、制御

電流として独立電圧源の枝電流を指定する必要がある．そのため，図 7-3(b) では枝電流を取り出す目的で，ダミーとして 0 V の独立電圧源を二つ挿入している．

SPICE 系回路シミュレータでは，マクロモデルを.SUBCKT 行を用いて部分回路として定義できる．また，定義したマクロモデルは，X 素子として回路内に配置することができる．

7-1-3 アナログ動作記述言語

マクロモデルを用いることで，複雑な機能モジュールを計算コストを抑えて解析することができる．また，機能モジュールを様々な抽象度で記述できるため，トップ・ダウン設計にも活用できる．例えば，アナログ回路を含むシステムの仕様策定段階では，より抽象度の高い簡易モデルを用いたシステム全体の動作検証を行う．次に，その仕様を実現するための機能モジュールの詳細設計へと進む．一方で，マクロモデル自体を記述するためには，例えば機能モジュールの入出力特性を表す関数を，従属電源などを組み合わせた等価回路として記述しなければならず，設計者にとって大きな負担となる．

デジタル回路設計の分野では，早くから VHDL や Verilog HDL といったハードウェア記述言語が機能モジュールの動作記述に活用されており，トップ・ダウン設計やコンカレント設計に活用されている．一方で，アナログ・デジタル混載回路技術の進展により，デジタル部だけでなくアナログ部にもトップ・ダウン設計による設計効率の向上が求められるようになった．しかし，先に述べたように，マクロモデルによる動作記述は設計者にとって容易でない．そのため，1990 年代後半以降，ハードウェア記述言語をアナログ記述にまで拡張した VHDL-AMS や Verilog-AMS が実用化された．

抵抗素子の Verilog-AMS による記述例を図 7-4 に示す⁴⁾．動作記述言語の構文は C 言語や Pascal 言語などのプログラミング言語と類似しており，回路特性を表す関数をそのまま数式として記述できるため，等価回路合成の手間が不要で修得が容易である．

```
module resistor(pos, neg);
input pos, neg;
electrical pos, neg;
parameter real res=50.0;
analog
V(pos, neg) <- res * I(pos, neg);
endmodule
```

図 7-4 抵抗素子の Verilog-AMS による記述例

また，アナログ動作記述言語を活用することで，回路シミュレータ内に素子（デバイスモデル）が準備されていないような，モータ，センサや MEMS などの特殊デバイスのモデル化も可能である³⁾．さらに，PSP や HiSIM などの回路シミュレータ用の次世代トランジスタモデルの研究開発においても，動作記述言語が活用されている⁵⁾．

参考文献

- 1) D. O. Pederson, "A Historical Review of Circuit Simulation," IEEE Trans. Circuits and Systems, vol.CAS-31, no.1, pp.103-111, Jan, 1984.

- 2) 黒瀬能幸, 岡田和之, “Spice を使った電子回路設計工学,” pp.151-171, 森北出版, 2002.
- 3) D. Jansen, “The Electronic Design Automation Handbook,” pp.238-325, Kluwer Academic Pub., 2003.
- 4) 浅井秀樹, 渡邊貴之, “電子回路シミュレーション技法,” pp.264-291, 科学技術出版, 2003.
- 5) 大黒達也, “HiSIM と PSP のベンチマーク結果,” 第 19 回 回路とシステム軽井沢ワークショップ論文集, pp.63-68, 2006.

1群-7編-7章

7-2 回路シミュレーションの計算アルゴリズム

(執筆者: 渡邊貴之)[2008年12月受領]

回路シミュレーションは、回路方程式の定式化と、回路方程式を数値計算により解く過程からなる。本節ではまず、これらの過程について説明する。その後、回路シミュレーションの計算精度を左右する重要な役割を担うデバイスモデルについて説明する。

7-2-1 回路方程式の定式化

ネットリストとして与えられた素子の接続情報やモデルパラメータをもとに、回路方程式を構成する。回路の定式化の手法としては、すべての節点電圧を未知変数として方程式を構成する節点解析 (NA) や、すべての節点電圧、枝電圧、枝電流を未知変数として方程式を立てるスパースタブロー解析 (STA) がある。ここで、素子を挟んだ節点間の電位差を枝電圧、素子に流れる電流のことを枝電流と呼ぶ。また、SPICE系回路シミュレータでは、すべての節点電圧と一部の必要となる枝電流を未知変数とする修正節点解析 (MNA) を採用している。

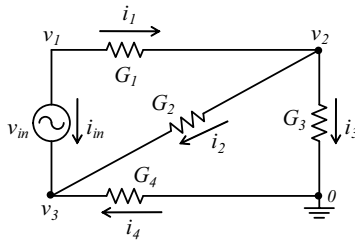


図 7-5 修正節点解析の回路例

一例として、図 7-5 の回路について、修正節点解析により回路方程式を構成する。キルヒホッフの電流則と、 $v_1 - v_3 = v_{in}$ の関係から、下記の修正節点方程式が得られる。

$$\begin{bmatrix} G_1 & -G_1 & 0 & 1 \\ -G_1 & G_1 + G_2 + G_3 & -G_2 & 0 \\ 0 & -G_2 & G_2 + G_4 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ i_{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ v_{in} \end{bmatrix} \quad (7-1)$$

上記の方程式を解くことにより、すべての節点電圧と独立電圧源に流れる電流を求めることができる。

一方、回路シミュレータでは以下に示すスタンプと呼ばれるルールを用いて、ネットリストの記述内容から修正節点方程式の係数行列を機械的に生成している¹⁾。表 7-2 に、コンダクタンス G を持つ抵抗素子のスタンプルールを示す。このルールでは、抵抗素子の両端の節点電圧が未知変数ベクトルの i 行と j 行に格納されている場合に、係数行列の i 行 i 列および j 行 j 列に G を加算し、 i 行 j 列および j 行 i 列に $-G$ を加算する操作を表している。実際に、

図 7.5 の抵抗素子 G_1 について、 v_1 を未知変数ベクトルの 1 行目、 v_2 を 2 行目としてスタンプルールを適用すると、式 (7.1) の係数行列と同様に G_1 がスタンプされることがわかる。

一方、独立電圧源については、表 7.3 のルールに従って値を設定すればよい。ただし、電圧源に流れる電流が未知変数ベクトルの k 行に格納されていると仮定している。

表 7.2 抵抗素子のスタンプルール

	i	j	RHS
i	G	$-G$	
j	$-G$	G	

表 7.3 独立電圧源のスタンプルール

	i	j	k	RHS
i			1	
j			-1	
k	1	-1		v

7-2-2 回路方程式の解法

(1) 大規模線形連立方程式の求解

スタンプ処理により構成された回路方程式は大規模線形連立方程式となるため、ガウスの消去法や LU 分解法を用いて解を求める。係数行列が密行列である場合、 N 変数からなる線形連立方程式の解法には $O(N^3)$ の計算量が必要である。一方、一般に回路方程式の係数行列は 90% 以上の要素がゼロとなる疎行列（スパース行列）である²⁾。従って、SPICE 系回路シミュレータでは、計算機のメモリ容量を節約し、計算時間を短縮化するために、非ゼロ要素のみを格納し、ゼロ要素の計算をスキップする効率的なスパース行列技法が採用されている。

また、LU 分解法を用いて係数行列を上三角行列 L と下三角行列 U に分解する際には、対角要素による除算が必要となる。従って、係数行列の行と列を並べ替えることにより、対角要素がゼロとなるのを防ぐ枢軸（ピボット）選択が行われる。一方、枢軸選択は LU 分解後の行列に多くの非ゼロ要素が生じる（フィル・インが発生する）ことを抑制する目的でも行われる。枢軸選択の順序を適切に決定する処理を、リオーダリング（re-ordering）と呼ぶ。

(2) 非線形素子の取り扱い

ダイオードやトランジスタなどの半導体素子の電圧・電流特性は、非線形関数として記述される。これらの素子を含む回路方程式は、未知変数ベクトル x に関する非線形代数方程式

$$f(x) = 0 \tag{7.2}$$

となる。SPICE 系回路シミュレータでは、これをニュートン・ラフソン（Newton-Raphson）法（以下、ニュートン法）によって解く。ニュートン法を式 (7.2) に適用すると、反復式

$$x^n = x^{n-1} - \left(\frac{\partial f}{\partial x}\right)^{-1} f(x^{n-1}) \quad (7.3)$$

が得られる。ここで n は反復回数であり、 $\partial f/\partial x$ は関数 f の変数 x に関する偏導関数からなるヤコビアン (Jacobian) 行列である。ヤコビアン行列は、 N 変数からなる回路方程式では $N \times N$ の正方行列となる。

ニュートン法では、ある初期値 x^0 を与え、関数の勾配に従って解を追跡し、反復計算により解を求める。実際には、ヤコビアン行列の逆行列を直接求めることは非効率であるため、以下に示すように計算を分割して行う。ヤコビアン行列を A と置き、式 (7.3) を

$$\begin{cases} AX = -f(x^{n-1}) \\ x^n = x^{n-1} + X \end{cases} \quad (7.4)$$

と変形すると、各反復は線形方程式の求解とベクトル和に帰着される。

ニュートン法では、解に近い初期値を用いた場合には二次収束性が得られるが、解から遠い初期値から開始した場合に収束性が悪化する。

(3) 過渡解析と数値積分法

キャパシタ、インダクタを含む回路は、時間領域において常微分方程式として記述される。また、半導体素子を含む回路であれば、非線形常微分方程式

$$Cx(t) + f(x(t)) = 0 \quad (7.5)$$

となる。ここで、 C はキャパシタンス及びインダクタンスからなる係数行列である。

過渡解析では、これらの方程式を数値積分法を用いて時間ステップ幅 Δt を刻みながら差分化する。例えば、後退オイラー法では、 $(x_t - x_{t-1})/\Delta t$ を時刻 t における x の微分値とみなし、

$$C \frac{x_t - x_{t-1}}{\Delta t} + f(x_t) = 0 \quad (7.6)$$

と差分化を行う。また、より高精度な方法として、SPICE 系回路シミュレータで用いられている台形公式では、 $(x_t - x_{t-1})/\Delta t$ を時刻 $t + 1/2$ における x の微分値とみなす。関数 $f(x_{t+1/2})$ の値を時刻 $t + 1$ と t での値の平均とみなし、

$$C \frac{x_t - x_{t-1}}{\Delta t} + \frac{f(x_t) + f(x_{t-1})}{2} = 0 \quad (7.7)$$

と差分化を行う。

式 (7.6)、式 (7.7) を改めて

$$F(x_t) = 0 \quad (7.8)$$

と置けば、未知変数ベクトル x_t に関する非線形代数方程式となる。従って、過渡解析は数値積分法の各時間ステップごとに、非線形代数方程式をニュートン法により解く問題に帰着されることが分かる。未知変数ベクトル x_t を求めるためのニュートン法の初期値としては、 x_{t-1} がよい初期値となり得る。従って、数値積分法の時間ステップ幅が小さい場合には、 x_{t-1}

はよりよい初期値となり少数の反復で解が求まるが、反対に時間ステップ幅が大きすぎる場合には収束性が悪化する。

7-2-3 デバイスモデル

回路シミュレータは、抵抗やキャパシタ、トランジスタなどの素子ごとに、素子の端子電圧と枝電流の関係を数式で表したデバイスモデルを内蔵している。例えば、抵抗値 R を持つ抵抗素子の場合、オームの法則から導かれる $i_{12} = (v_1 - v_2)/R$ をモデル式として利用している。

一方、ダイオードやトランジスタといった半導体素子では、モデル式はより複雑な非線形関数となる。本来であれば、半導体内における物理現象を材質や形状を含めて高精度に計算するためには、緻密なプロセスシミュレーションや、2次元・3次元のデバイスシミュレーションを行う必要がある。しかし、これらの方法では、トランジスタ数個の動作を計算するだけで膨大な計算コストを要するため、大規模回路の解析は困難である。そこで、回路シミュレータでは、半導体素子の動作をコンパクトな数式で近似したモデル(コンパクトモデル)を用いることで、計算コストの低減を図っている。

一方で、コンパクトモデルの良し悪しが回路シミュレータの計算精度や計算コストに大きな影響を持つ。従って、より高精度かつ計算コストの少ないモデルの研究開発が継続的に行われている。

(1) BJT モデル

SPICE 系回路シミュレータにおけるバイポーラトランジスタのコンパクトモデルとしては、古典的なエバース・モル(Ebers-Moll)モデルを改善した、より厳密なガンメル・プーン(Gummel-Poon)モデルが用いられている。また近年、より多くの物理特性を考慮した VBIC や MEXTRAM などのモデルが提案されている³⁾。

(2) MOSFET モデル

SPICE には複数の MOSFET デバイスモデルが内蔵されており、MODEL 行でモデルパラメータを定義する際に、LEVEL パラメータによってデバイスモデルの種類を切り替えることができる。例えば、ゲートソース間電圧 V_{GS} がしきい値電圧 V_T を超えると、ドレイン電流 I_D が流れるという MOSFET の基本原理を単純にモデル化したものが LEVEL = 1 の MOS1 (Shichman-Hodges) モデルである。MOS1 の直流電流特性(NMOS)は下記の区分連続関数で表される²⁾。

a) 遮断領域 ($V_{GS} < V_T$)

$$I_D = 0$$

b) 線形領域 ($V_T < V_{GS}$, $0 < V_{DS} < V_{GS} - V_T$)

$$I_D = K_p \frac{W}{L_{eff}} \left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2} \right] (1 + \lambda V_{DS})$$

c) 飽和領域 ($V_T < V_{GS}$, $V_{GS} - V_T < V_{DS}$)

$$I_D = K_p \frac{W}{2L_{eff}} (V_{GS} - V_T)^2 (1 + \lambda V_{DS})$$

ただし, $V_T = V_{T0} + \gamma[(\Phi - V_{BS})^{1/2} - \Phi^{1/2}]$, $L_{eff} = L - 2L_D$ である. また, K_p はトランスクンダクタンス, λ はチャネル長変調パラメータ, Φ は表面電位, V_{T0} は零バイアスしきい値電圧, L_D は横方向拡散長である. チャネル長変調に関する項である ($1 + \lambda V_{DS}$) の効果は, 物理的には飽和領域でのみ観測されるが, MOS1 では線形領域と飽和領域の境界で関数の微分値の連続性を保つために線形領域においても付加されている. 更に, 過渡解析では各端子間の寄生容量が考慮される.

SPICE には, 上述の MOS1 を始めとして, 短・狭チャネル効果, キャリア移動度の変化, サブスレッショルド電流などの物理特性のモデル化を試みた MOS2, モデル式に経験的パラメータを導入した MOS3 などが含まれている. しかし, これらのデバイスモデルはサブミクロン世代のシミュレーションには不正確であり, その後, BSIM1・BSIM2 が開発された. 更に, $0.25 \mu\text{m}$ 以下のディープサブミクロン世代を対象とした BSIM3・BSIM4 が開発された⁴⁾.

一方, BSIM モデルは, しきい値電圧によりモデル式が切り替わるため不連続性を有しており, アナログ・RF 回路への適用に限界が生じた⁵⁾. また, モデルパラメータ数が肥大化し, かつ, 異なるデバイスサイズ(チャネル長・幅)に対して同一のパラメータセットを用いることができず, サイズごとに異なるパラメータを抽出するピニング処理が必要であり, パラメータ抽出に時間を要するなど問題があった. そのため, 近年, チャネル内の表面ポテンシャルに基づく, より物理モデルに忠実な PSP や HiSIM が開発されている.

参考文献

- 1) 浅井秀樹, 渡邊貴之, “電子回路シミュレーション技法,” pp.264-291, 科学技術出版, 2003.
- 2) D. Jansen, “The Electronic Design Automation Handbook,” pp.238-325, Kluwer Academic Pub., 2003.
- 3) 新原盛太郎, “SPICE とデバイス・モデル,” pp.177-192, CQ 出版社, 2005.
- 4) Y. Cheng and C. Hu 著, 鳥谷部達 訳, “MOSFET のモデリングと BSIM3 ユーザーズガイド,” 丸善, 2002.
- 5) M. Miura-Mattausch, et.al, “Circuit Simulation Models for Coming MOSFET Generations,” IEICE Trans. Fundamentals, vol.E85-A, no.4, pp.740-748, April, 2002.