数 値 解 析 39－4
（１９９１．１２．６）

機能／回路混在系シミュレーション手法

平井千秋　渡辺俊典　林晋一
日立製作所　システム開発研究所

LSIの大規模化に伴い，従来の回路シミュレータでは，解析困難な回路が出現している．この問題に対し我々は先に，従来回路シミュレータと連続形シミュレータを融合させることにより回路シミュレーションを高速化する方法を提案した．今回，この提案にもとずいた機能／回路混在系記述言語および言語処理系を開発した．この言語処理系は，ユーザが解析対象を宣言的に記述したデータから計算実行順序を生成するものである．この処理系により機能／回路混在系シミュレータは，実用的DA(Design Automation) システムとして利用可能になる．本報告では，機能／回路混在系シミュレーションの演算方式と，データから計算実行順序をスケジューリングする方法を明らかにする．

Mixed function and circuit system simulator

Chiaki Hirai　Toshinori Watanabe　Shin-ichi Hayashi
Systems Development Laboratory
Hitachi, Ltd.
1099 OHZENJI ASAO-KU　KAWASAKI-SHI, 215 JAPAN

Large scale integrated circuits in recent years are exceeding the capability of conventional circuit simulators especially in terms of long simulation time needed for a simulation. To accelerate circuit simulation, we proposed a new method based on the combination of the circuit simulation method and the continuous system simulation method. In this paper we describe the simulation method and data description language for the simulator. We show an algorithm to produce the calculation procedure from mixed function and circuit definition. With a complier developed on the algorithm, the mixed level simulator becomes a useful design automation tool .

1．まえがき

LSIの大規模化に伴い，従来の回路シミュレータでは，解析困難な回路が出現している．これは，非線形素子と後退形の積分公式に起因する収束演算の際に大規模行列計算を行うためである

この問題を解決する方法として，従来プラント等制御系の動特性解析に利用されてきた連続系シミュレータ[1][2]を用いてシミュレーションを行う方法がある[3][4]．これは，数値積分として前進形の積分公式を採用して収束計算を避けることにより，シミュレーションの高速化を図る方法と位置付けられる．しかし，回路混在の対象を直接入力記述できないため，人手による回路の関数ブロックへの抽象化を要するという難点が残された．

これらの問題に対し我々は，従来回路シミュレータと連続形シミュレータを融合させることにより回路シミュレーションを高速化する方法を提案した[5][6]．今回，この提案に基ずく機能／回路混在系記述言語および言語処理系を開発した．この言語処理系は，ユーザが解析対象を宣言的に記述したデータから計算実行順序を生成するものである．この処理系により機能／回路混在系シミュレータは，実用的DA(Design Automation) システムとして利用可能になる．

本報告では，機能／回路混在系シミュレーションの演算方式と，データから計算実行順序をスケジューリングする方法を明かにする．

以下，従来のシミュレーション方式（第２章），機能／回路混在系シミュレーション方式（第３章），計算スケジューリング（第４章）についての述べる．

2．従来のシミュレーション方式

2．1　回路シミュレータ

回路シミュレータが対象とする系は，素子間電圧と素子を流れる電流との関係式にキルヒホッフ則を連立させて得られる連立微分方程式である[7]．

容量の素子間電圧（v）と素子を流れる電流(i)との関係式は，容量をCとして，次式で与えられる．

$$i = C\frac{dv}{dt}.$$

線形抵抗の特性を表す式は，抵抗値をRとして，

$$i = \frac{1}{R}v$$

となる．

また，ダイオードの素子間電圧（v）と素子を流れる電流（i）との関係式は，一般に次のようにモデル化される．

$$i = Is(\exp(\lambda v)-1).$$

ここに，Is, λは，製造プロセスによって定まる定数である．この式の非線形特性によりシミュレーション対象は，連立非線形微分方程式となる．

一般に回路シミュレータでは，数値積分を行う方法として後退形積分公式を採用している．これは，後退形積分公式が，安定性の点において前進形積分公式に優っているためである．

後退形積分法は、微分方程式，

$$\frac{dy}{dt} = f(t,y),$$

を次式で差分化することによって得られる[8]．

$$y_{n+1}=y_n+\Delta t(1 - \frac{1}{2}\nabla - \frac{1}{12}\nabla^2 + \cdots + a_N\nabla^N)f_{n+1},$$

ただし，

$$t_{n+1} = t_n+\Delta t,$$
$$y_n = y(t_n),$$
$$f_n = f(t_n,y_n),$$
$$\nabla fn = f(t_n,y_n) - f(t_{n-1},y_{n-1}).$$

簡単のためN=0の後退オイラー法を採用するが，より高次の積分公式でも同様の議論が成立する．

このとき回路は次の連立方程式によって記述される．

容量特性式　$i_{n+1} = C\frac{v_{n+1} - v_n}{\Delta t}$,

線形抵抗特性式　$i_{n+1} = \frac{1}{R} v_{n+1}$,

非線形素子特性式　$i_{n+1} = Is(\exp(\lambda v_{n+1})-1)$,

キルヒホッフ則　.

この連立方程式は，時刻 t n+1のすべての素子についての端子間電圧を時刻 t nにおける端子間電圧の値より計算する式を与えるが，非線形素子特性式のために非線形連立方程式となる．

多くの回路シミュレータにおいては，非線形連立方程式を解くために，ニュートンラフソン法による反復公式を用いている．この連立方程式は，現在のLSI規模からして，10^3を超える次数になっている．

また，ニュートンラフソン法は，解の大域的収束性が保証されていないため，解析対象によっては解が得られないという問題もある．この方法への対処として求解法をホモトピー法に換える方法[9]も研究されているが，設計現場で広く利用されるには至っていない．

2．2　連続系シミュレータ

従来回路シミュレータの難点への対処として，従来，プラント等の動特性設計に用いられてきた連続系シミュレータを用いる方法が提案されている．

この方法は，解析対象となる連立微分方程式を前進形積分公式で離散化することにより，連立式に計算順序を持たせる点で回路シミュレータと異なっている．式を計算順序に従って逐次計算することにより行列計算を避け，シミュレーションを高速化できる．

前進形積分法は，微分方程式を次式で差分化することによって得られる．

$$y_{n+1}=y_n+\Delta t(1+\frac{1}{2}\nabla+\frac{5}{12}\nabla^2+\cdots+b_N\nabla^N)f_n.$$

簡単のため以下N=0の前進オイラー法を採用するが，より高次の積分法でも同様の議論がなりたつ．

具体的例により，前進オイラー法による回路の解析法を示す．図１に解析対象を示す．

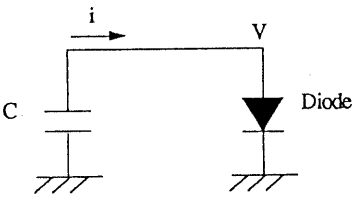

図１　解析例１
Fig.1 Example 1

この解析対象を，前進オイラー法によって離散化すると，

$$i_n = Is(\exp(\lambda v_n)-1),$$
$$i_n = C\frac{v_{n+1} - v_n}{\Delta t} .$$

初期値v_0が与えられると，逐次的にi_n，v_nを求めることができる．

連続系シミュレーションでは一般に解析対象をブロック線図によって表す．以下連続系シミュレーションによる解析対象には，ブロック線図を用いる．図１の解析対象をブロック線図化すると図２となる．

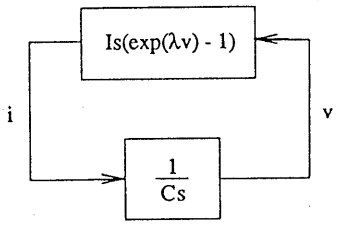

図２　ブロック線図１
Fig.2 Block Diagram 1

連続系シミュレータは，回路解析に必要な演算機構を欠いているため，回路を連立微分方程式で記述し，ブロック線図化する作業を人手で行う必要があり，使用上の難点となっていた．

3．機能／回路混在系シミュレーション方式

我々のシミュレーション方式は，回路シミュレータと連続系シミュレータの融合方式である．

回路設計の初期（上流設計）段階では，設計者の関心は，電子素子の振舞いにあるのではなく，回路のシステム構成をいかに構築するかにある．この段階での設計者の考察対象は，ブロック線図によって表すことができる．このとき各ブロックは，LSIの一部分が担う機能を表している．機能設計が終了すると設計者は，機能関数を順次回路に詳細化してゆく（中流設計）．この時点で，設計者の考察対象は，機能と回路が混在した系となる．中流設計が終了すると回路シミュレータでのシミュレーションが可能となる．

従来の回路シミュレータでは，全ての機能ブロックが詳細化されるまでシミュレーションを行うことができない．これは，従来回路シミュレータの機能的問題点といえると共に，無用に詳細化した回路を解析することにより解析時間に長時間を要しているともいえる．

そこで，機能と回路が混在した系をシミュレーションすることが出来れば，設計者は，中流設計レベルでシミュレーションが行えると共に，回路の一部を機能関数化して行列を小型化してシミュレーション速度を向上させることが期待できる．

基本的な考え方は，従来回路シミュレータをサブルーチンとして連続系シミュレータから呼び出すことによって実現される．以下に詳細を述べる．

入力端子と出力端子を持つ回路に入力物理量を与えたときの出力物理量は，回路シミュレータを用いて計算することができる．一方，連続系シミュレータの扱う系は，関数ブロックからなる系であり，扱える関数ブロックとしての要請は，入力に対して物理量を計算できる関数であることでしかない．従って，回路シミュレータを入力物理量から出力物理量を計算するサブルーチンと捉えることにより，電子素子からなるブロック（回路ブロック）を連続系シミュレーションの一ブロックとして置くことが可能である．

ここで，回路シミュレータで用いられている後退形積分公式に着目する．

$$i_{n+1} = C\frac{v_{n+1} - v_n}{\Delta t}$$

この式は，容量端子間の電流と電圧の同時刻での量を関係付けている．微積分作用素を含まない式も同時刻での電流と電圧の関係式を与えている．従って，入力物理量を与えて一時間刻み分の計算を行って得られた出力物理量は，入力物理量と同時刻での値となる．

このことより，回路シミュレータは，連続系シミュレータにおいて，時間遅れなしの関数ブロックを計算するためのサブルーチンと捉えることができる．

4．計算スケジューリング

4．1　解析例題

具体例を用いて，機能／回路混在系の入力データ記述言語と入力データからの計算順序スケジューリング方法を示す．

解析例を図3に示す．本回路は，電話器において，相手方が受話器を置いたことを検出するための回路である．検出方法として電話局から発信されるBusy Tone（周波数400Hz付近）をPLL(Phase Locked Loop)方式[10]で検出する．例えば，周波数410Hz，振幅200mVのbtinが回路に入力されると，実回路の場合，vout1の電位が上がる．vout1がしきい値（Vth）をこえると，btoutのレベルが下がる．btoutのレベルが下がった状態が，Busy Tone検出の状態である．図中mult1, mult2は，乗算器回路である．mult1の回路図を図4に示す．

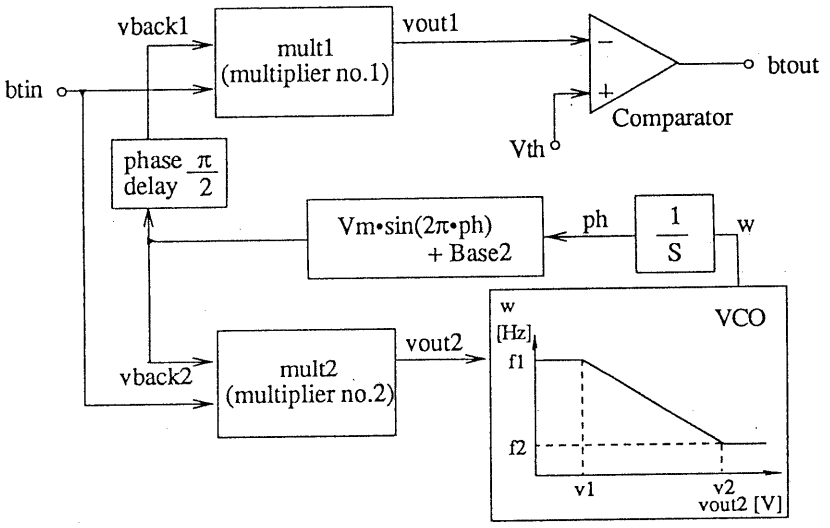

図3　LSIモデル

Fig.3 Model of the LSI

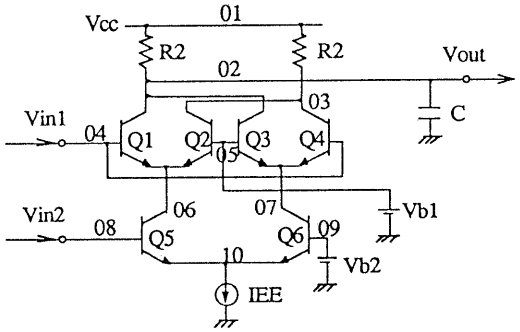

図4　乗算器1

FIg.4 multiplier no.1

図3，図4に示すように，本システムの解析モデルは，機能ブロックと回路ブロックの混在系となっている．

本回路のシミュレーションの目的は，入力周波数に対して検出可能な入力信号の大きさを解析し，検出可能信号領域を明かにすることである．

4．2　記述言語形式

本回路を記述した例を図5，図6に示す．図5は全体系の記述であり，図6は回路ブロックの記述である．図5に示すように機能ブロックの記述は，入力変数物理量と出力物理量を結び付ける代入式で表現する．回路記述は，図6に示すように従来回路シミュレータと互換性を持つネットリスト表現を採用している．

ネットリストの物理的意味は，例えば，

Vcc　01　00　5.0

と表現した場合，節点01と00の間に5.0[V]の電圧源があることを示す．R,C,Q,I　は，それぞれ抵抗，容量，トランジスタ，電流源を示す．

4．3　計算スケジューリング

記述されたデータを読み込んでシミュレーションを行うまでの手順を図7に示す．

```
busy_tone( )
  { function vco=(v1,f1),(v2,f2);
    constant {FIN=410.0,AIN=200.0e-3}
    btin     = AIN*pulse(60.0e-3,125.0e-3,
     250.0e-3)*sine(0.0,2.0*PAI*FIN,0.0)+Vb2;     (1)
    vout1    = mult1(vback1,btin);                (2)
    vout2    = mult2(vback2,btin);                (3)
    w        = afgen(vco,vout2);                  (4)
    ph       = intgrl(0.0,w);                     (5)
    vback1   = Vm*sin(2.0*PAI*ph+dph)+Vb1;        (6)
    vback2   = Vm*sin(2.0*PAI*ph)+Vb1;            (7)
    if(vout1>=Vth) btout= - 1.0;
    else           btout=  0.0;                   (8)
  /********Conditions***********/
  timer { delt=0.1e-3,fintm=250.0e-3 }
  graph { vout1,btin,btout,ymin=4.0,ymax=5.0}
}
```

図5　全体系記述例

Fig.5 Definition of the system

```
Vout=mult1(Vout,Vin1,Vin2)
  {Vout  02  00, Vin1  04  00, Vin2  08  00}
{
    Vcc   01   00   5.0
    R2    01   02   3.9E3
    R2    01   03   3.9E3
    C     02   00   4.7E-6
    Q1    02   04   06
    Q2    03   05   06
    Q3    02   05   07
    Q4    03   04   07
    Q5    06   08   10
    Q6    07   09   10
    IEE   10   00   298E-6
    Vb1   05   00   3.0
    Vb2   09   00   2.0
}
```

図6　乗算器1記述例

Fig.6 Definition of multiplier no.1

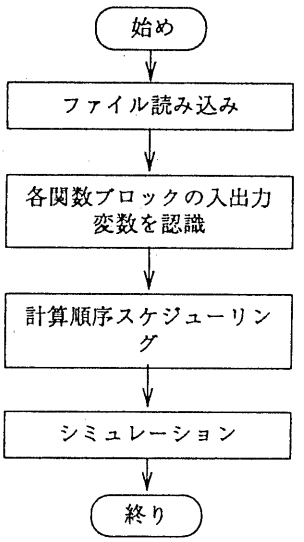

図7　シミュレーション実行手順

Fig.7 Simulation procedure

入力データを言語処理系が解析し，各関数ブロックの入出力変数を認識する．この認識結果に基づき，関数ブロックの計算実行順序を決め（計算順序スケジューリング），シミュレーションを行う．

以下に計算実行順の決定方法を述べる．

計算実行順序は，次に定義する関数ブロックの順序クラスに基づいて実行順序を決める．順序クラスを定義する準備として，入出力変数，関数O，Oの順序べきを定義する．

definition　入出力変数

*出力変数*とは，関数ブロックを記述する式の左辺に現れる変数である．

*入力変数*とは，関数ブロックの右辺中に現れる変数のうち，出力変数と同時刻の物理量を表す変数である．

definition　出力関数

$O(S) =$

$\{e \mid \exists f (\in F)\ \exists v (\in S)(\ e \in Out(f) \wedge v \in In(f)).\ \} \cup S$

ただし，

関数ブロックfの入力変数の集合をIn(f)，出力変数をOut(f)と表す．

definition　Oの順序数べき

$O\uparrow 0 = \{\}$

$O\uparrow \alpha = O(O\uparrow(\alpha - 1))$

関数ブロックの順序クラスを定義する．

definition　関数ブロックの順序クラス

$$C(\alpha) = \{ f(\in F) \mid \min_{\beta} (In(f) \subseteq O\uparrow \beta) = \alpha \}$$

関数ブロックの計算実行順序は，関数ブロックの順序クラスの小さい順に行う．同一順序クラス内の関数ブロックの実行順序は任意である．ただし，記述によっては，順序クラスが定義できない場合がある．これは，ユーザが定義したプログラムに代数的ループが生じている場合で，演算結果の正しさは保証されない．ここに，代数的ループとは，$\alpha \leqq \beta$で，順序クラスαの関数ブロック$f\alpha$と，順序クラスβの関数ブロック$f\beta$において，

$$\exists x\,(x \in Out(f\beta) \wedge x \in In(f\alpha)).$$

が成立することを言う．

先の解析例を用いて順序付けを説明する．言語処理系は，ファイルを読み込み，プログラムbusy_toneを解析し，各関数ブロックの入出力変数を認識する．図8に認識結果を示す．次にOの順序べきを生成しながら関数ブロックの順序クラスを決定する．図9に結果を示す．アルゴリズム的な実現は容易であるので説明は略す．

シミュレーションは，関数ブロックの順序クラスの小さい順に行う．従って本例では，例えば(1),(5),(6),(7),(2),(3),(4),(8)の順に計算を行う．この計算により，時間刻み一ステップ分のシミュレーションを行うことができる．これを所定時間まで繰り返すことにより，回路の過渡応答を解析できる．

回路ブロック(2),(3)の計算は，3章に述べたように従来回路シミュレータを用いて実行することができる．

ここで注意すべきは，関数ブロック(5)の第2引数は入力変数ではないことである．関数ブロック(5)は，積分を表すブロックであるが，数値積分法に前進法を採用しているため，積分

$$ph = \int w\,dt$$

は，次式で離散化される（オイラー法を採用）．

$$ph_{n+1} = ph_n + w_n \Delta t$$

この式に見るように入力物理量wは，出力物理量phと同時刻ではなく，定義によりw は入力変数ではない．

従って，本例が見かけ上ループを構成しているにもかかわらず，代数的ループをなしていない．言い替えれば，ブロック線図上のループ中には，積分作用素が含まれることが必要である．

式	入力変数	出力変数
(1)		btin
(2)	vback1, btin	vout1
(3)	vback2, btin	vout2
(4)	vout2	w
(5)		ph
(6)	ph	vback1
(7)	ph	vback2
(8)	vout1	btout

図8　入出力変数

Fig.8 input/output variables

α	O↑ α	C(α)
0	{ }	{ (1), (5) }
1	{ btin, ph }	{ (6), (7) }
2	{ btin, ph, vback1, vback2 }	{ (2), (3) }
3	{ btin, ph, vback1, vback2, vout1, vout2 }	{ (4), (8) }

図９　順序クラス
Fig.9 Class of Order

４．４　解析例

入力周波数410Hzの場合の過渡解析の例を図１０に示す．計算時間は回路シミュレーションに比べ大幅に高速化された．

５．まとめ

機能／回路混在系シミュレータの演算方式と機能／回路混在系記述言語処理系について述べた．今回開発した処理系により，機能／回路混在系シミュレータは，実用的DA(Design Automation) システムとして利用可能となる．

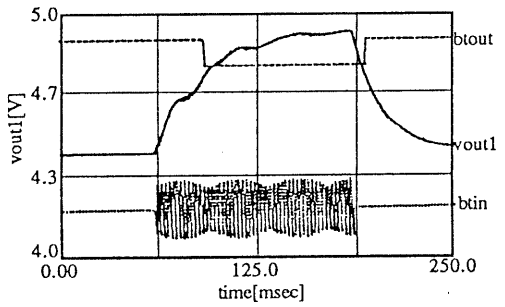

図１０　解析結果
Fig.10 Result of simulation

６　参考文献

１）藤井信生：アナログ集積回路の計算機援用解析・設計における問題点，電子情報通信学会論文誌A，Vol.J73-A No.8 pp.1313-1320(1990).

２）三巻達夫：ダイナミックシステムのディジタルシミュレーション，計測と制御，7,4,pp.256-268(1968).

３）Granino A.Korn and John V. Wait : Digital Continuous-system simulation, Prentice-Hall, Inc.(1978).

４）林晋一ほか：連続系シミュレーション手法によるVTR用カラー信号処理ＬＳＩの機能シミュレーション，電気情報通信学会論文誌A Vol.J72-A No.11 pp.1829-1843(1989).

５）K.Maio, et al : A Highly Efficient Design System for Mixed Analog/Digital LSIs, Proc. ESSCIRC'89, pp121-124(1989).

６）平井ほか：後退形／前進形両積分法を併用した機能／回路混在系の新シミュレーション手法，情報処理学会論文誌, Vol.32, No.8, pp. 1014-1021(1991年8月).

７）平井ほか：機能／回路レベル混在システムのシミュレーション方式，情報処理学会第４２回全国大会講演、講演論文集（６）pp.6-174 - 6-175.

８）Donald A. Calahan : コンピュータによる電子回路設計，日刊工業新聞社(1974).

９）赤坂隆：数値解析，コロナ社(1967).

10）K.Yamamura, et al : A Globally and Quadratically Convergent Algorithm for Solving Nonlinear Resistive Networks, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.9, no.5(1990).

11）畑，古川：PLL-ICの使い方，産報出版(1976).