

回路シミュレーションにおける数値解析 ～その問題点～

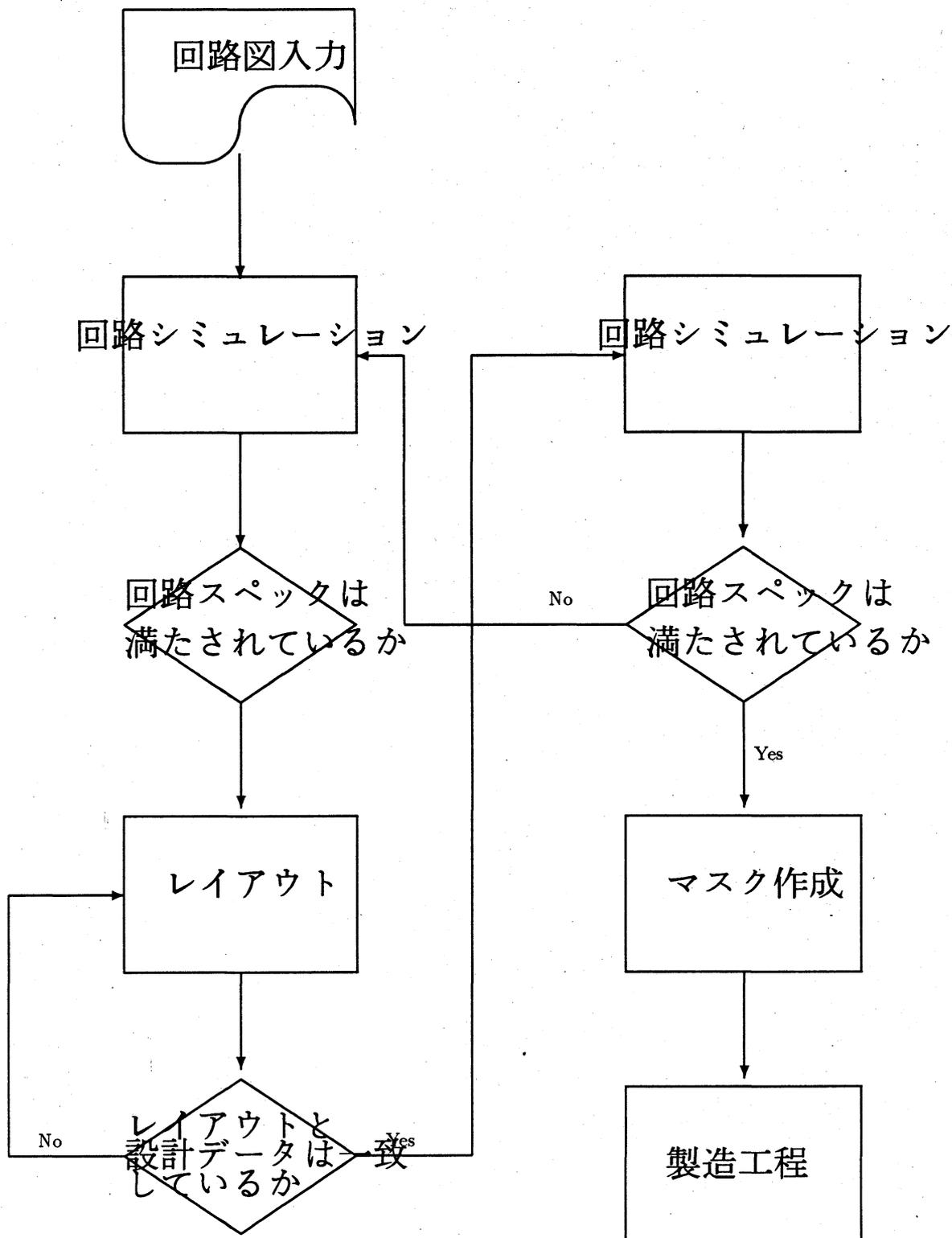
ソニー株式会社 村山 敏夫 (Toshio Murayama)

近年の L S I の大規模化、高集積化に伴い、L S I 設計をサポートする C A D (Computer Aided Design) の重要性がますます高まっている。中でも、L S I 回路の動作を製造前に設計段階でシミュレーションし設計の誤りを修正することが設計効率や製造コストを改善するために不可欠である。回路動作のシミュレーションは、各同電位節点を独立変数にとり、Kirchhoff's Current Law を用いて定式化することにより連立常微分方程式に帰着される。本稿では、回路設計におけるシミュレーションの持つ問題点について考察する。

1. L S I 設計のながれ

L S I は次ページの図に示されるように回路スペック (仕様) に沿って回路図を入力し、各同電位節点 (以下節点) を独立変数として Kirchhoff's Current Law に基づき以下のように定式化される。

LSI設計のながれ



$$F(x, x', t) = 0$$

x : 節点電位

t : 時間

さらに初期値が得られれば、常微分方程式の初期値問題と考えることができる。しかし、一般にLSIに用いられるキャパシタやインダクタは

$$q(v) = C(v)V(t)$$

$$\phi(i) = LI(t)$$

といった時変素子であるため、初期値を得るために $C = L = 0$ という条件化で回路を定常化する。これはキャパシタを解放しインダクタを短絡することに相当する。これにより、 x', t によらない関係が得られる。

$$G(x) = 0$$

この値を初期値として常微分方程式の初期値問題に帰することができる。この初期値は、回路ではバイアス値に相当し、フィルタやアンプの設計の際に重要な値である。

2. 収束性の問題

初期値を得るために一般には非線形連立方程式である $G(x) = 0$ を解く必要がある。しかし、よく用いられる Newton-Raphson 法では大域的な収束が保証されず、従って初期値が得られない

場合があり、微分方程式を数値的に解くことができないことがある。そこで、Homotopy 法などの連続法が提案されている。さて、バイポーラトランジスタを常微分方程式に定式化する際に良く用いられる Ebers-Moll モデルでは、Homotopy 法による大域的な収束が保証されているが、収束までに膨大な計算回数を要するため実用的でない。そこで、回路特性を利用した実用的な連続法がいくつか提案されている。これらは必ずしも大域的な収束を保証するものではないが、実際の回路方程式には多くの場合に置いて有効でかつ実用的な時間内に終了する。

2. 1 Gmin Stepping

各 Newton 反復の際に線形化された連立方程式を

$$Ax = b$$

とした場合、その方程式を

$$(A + nE)x = b$$

E : 単位行列

と変形する。 $n \rightarrow \infty$ により $x \rightarrow 0$ であるから、充分大きな n および初期値 $x = 0$ により n を連続的に ∞ から 0 に近付けることにより解が得られる可能性が大きい。この解曲線追跡は回路的には各節点とグラウンドノードとの間に小さな抵抗を接続し、その抵抗値を徐々に ∞ に近づけ、最終的に最初の

回路と等価な回路を得ることに相当する。

2. 2 Source Stepping

独立変数 x を、強制入力 (x_f) とそれ以外の変数 (x_n) とに分割し、強制入力 $x_f = 0$ から $x_f = x_f(t = 0)$ へと連続的に変形し、解曲線を追跡する。

$$G(x) = g(x_f, x_n) = 0$$

$$x_f = 0 \rightarrow x_f(t = 0)$$

回路方程式では、強制入力が $x_f = 0$ の場合、全節点が $x = 0$ となるため初期解 $x = 0$ が容易に得られる。実際の回路では、電源を 0 V から徐々に (回路を常に安定させながら) 電源電圧まで立ちあげていくことに相当する。

2. 3 Pseudo Transient

$G(x) = 0$ を解くのではなく、 $F(x, x', t) = 0$ から関数 F を連続的に変化させて G に漸近させる。

$$F(x, x', t) = f(x_f, x_n, x'_f, x'_n, t) = 0$$

$$x_f = 0 \rightarrow x_f(t = 0)$$

$$F \rightarrow G(C \rightarrow 0, L \rightarrow 0)$$

実際の回路では、電源を 0 V からリアルタイムで (回路が安定しているとは限らない) 電源電圧まで立ちあげていくことに相当する。

これらは経験上かなり有効な手段だが、収束が保証されている訳ではない。大域的な収束を保証しかつ実用的な時間内で計算が終了するアルゴリズムが望まれる。

3 初期値問題

初期値が得られれば、常微分方程式を解き、回路の時間領域応答を求めることが可能である。微分方程式の初期値問題

$$F(x, x', t) = 0$$

$$G(x) = 0$$

を解く際に重要なのは、精度を保ちながら実用的な時間内に計算できるアルゴリズムを用いることである。具体的には、アナログ回路で2000素子の回路の解析を行なう場合、通常は有効な精度を得ながら数十 μ 秒の解析を行なうのに10数時間のCPU時間を要する(通常のワークステーション使用時)ため、より速いアルゴリズムが期待される。

3.1 精度

精度を保ちながら計算時間を少なくするために、Dynamic Timestep Controlを行なうことが一般的であるが、よく用いられる Timestep Control には、

$$t_{k+1} = t_k \left| \frac{\Delta_{k+1}}{\Delta_k} \right|^{\frac{1}{n}}$$

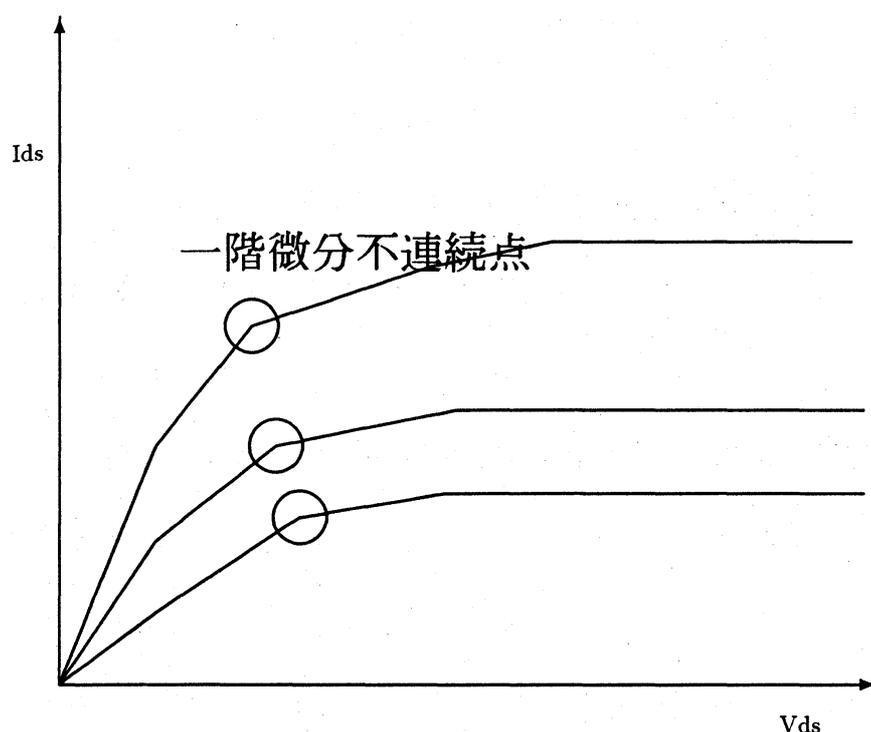
n : 積分次数

$\Delta_k : t_k$ における誤差の見つもり

があるが、 Δ_{k+1} の評価に要する時間が無視できず、高速のシミュレータでは誤差評価を行わず陰的積分の繰り返し回数を評価値として、繰り返しが少ない領域では Timestep を大きく、繰り返しが多い領域では Timestep を小さくするような評価基準が採用されているものもある。この方法では精度の方に難があるが、実用的な時間内で計算させるためには有効な手段と言える。

3. 2 安定性

回路方程式は一般には stiff (固い) 系であるため、A-安定性を得るために陰的数値積分法が採用される。しかし、半導体素子のモデリングではモデルの I/V 特性 (端子電圧に対する電流値特性) に、一階導関数不連続なものが存在する。(例えば、電界効果トランジスタの strong inversion region)



一階微分不連続なモデリング

そのため、陰的積分が収束せず、従って Timestep を小さくするプロセスが繰り返され、数値積分が進まなくなることがある。(Timestep Too Small Error) これは半導体素子の微細化、高性能化に伴い、素子のモデリングが複雑化するために起こるもので、モデリングの改良と共に、一階導関数不連続点での積分方法を考慮する必要がある。

3.3 解析時間

LSI回路を設計する段階でもっともコストがかかるのは、シミュレーションに要する計算機コストである。そこで解

析時間を短縮するために、高速な積分アルゴリズムの他にも、回路の特性を生かしたアルゴリズムが提案されている。

3. 3. 1 Waveform Relaxation

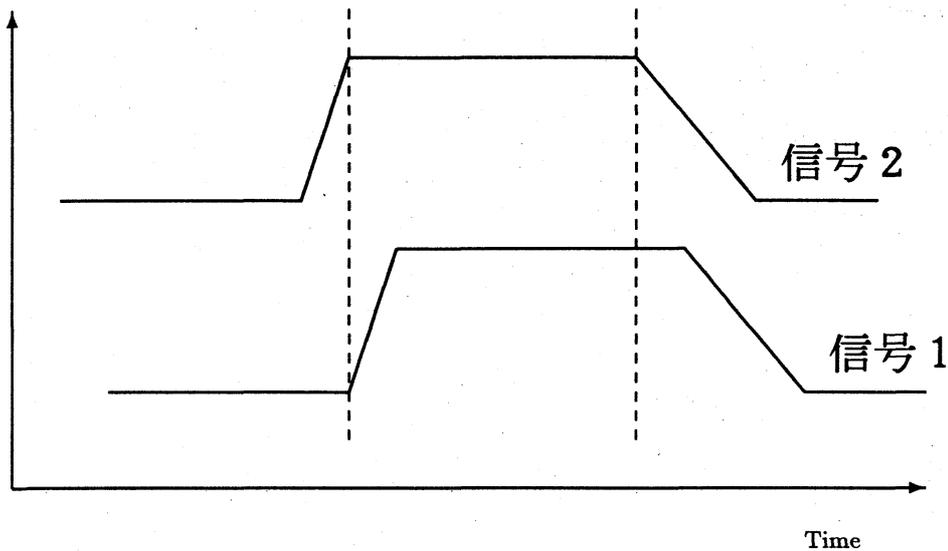
n 次元独立変数ベクトルの Latency を利用し、 $x_i' < \delta$ を満たす x_i とそれ以外の x_j を区別して Timestep をコントロールする。実回路では、動作している節点と動作していない節点とを別々に解析することであり、トップダウン設計における各機能ブロックごとに別々の Timestep で解析することに相当する。実際の回路では動作している節点はある一時刻においては数%程度であり、Latency を利用することにより大きな解析速度の向上が期待される。Latency が有効に活用されるためには、

- ・信号の流れが一方向であること（フィードバックがないこと）
- ・サブベクトル同士の結合が疎であること

であることが要求される。これはデジタル回路には有効であるが、アナログ回路では、回路状態がダイナミックに変化するため、サブベクトル化が困難な場合が多い。

また、数学的には、回路行列が対角優位である場合に収束性が保証されているが、実用上問題がある場合がある。以下に示すような信号 1、信号 2 の場合、時間領域における誤差は大きいですが、関数空間上では位相にわずかな誤差があるのみで、事実上収束していると判断して問題ない。しかし、Waveform

Relaxation では実用的で収束判定が難しく、上記のような状態を収束したと判定するために緩やかな収束条件を設定することにより誤った解に収束する例が報告されている。



時間領域誤差と位相誤差

3. 3. 2 Bypassing

陰的積分の繰り返し計算において、

$$f(x, x', t) = f_1(x_1, x'_1, t) + f_2(x_1, x'_1, t) + \cdots + f_n(x_n, x'_n, t)$$

と表される場合に

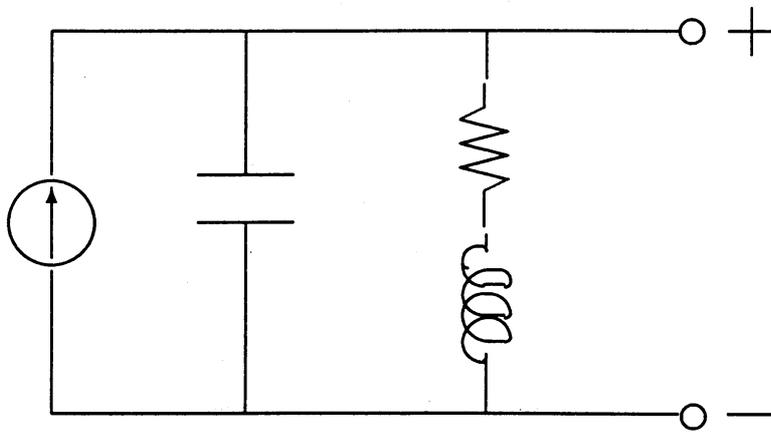
$$|x_i^{k+1} - x_i^k| < \delta$$

が見たされる場合、 f_i^{k+1} の評価をせずに f_i^k の値を利用して、補間等の手段により f_i^{k+1} を計算し時間を短縮する。実回路では、常微分方程式は各半導体素子のモデル関数の和に分解する

ことができ、従ってこの方法が有効なことが多い。

4 stiff (固い) 回路

LSI回路を常微分方程式に定式化した場合には通常 stiff な方程式になる。これは、LSIが扱う信号が時定数比が大きいものも多く（ビデオ信号など）、回路方程式の最大固有値と最小固有値が大きく異なるものが多いためである。



stiff な回路の例

$$\begin{bmatrix} C \frac{d}{dt} & 1 \\ -1 & R + L \frac{d}{dt} \end{bmatrix} \begin{bmatrix} v(t) \\ i(t) \end{bmatrix} = \begin{bmatrix} I(t) \\ 0 \end{bmatrix}$$

$$R = 1000\Omega$$

$$L = 1nH$$

$$C = 1pF$$

このような回路では安定性を得るために陰的積分を行なう必要があり、計算に時間がかかる場合が多い。また、時定数比が

大きいため、複合信号の周期が長くかつ局所的に短周期で動作しているため Timestep を小さくする必要があり、全体の計算（一周期）を行なうには膨大な計算量が必要となる。

5 今後の展望

実回路のシミュレーションにはこれら以外にも様々な問題があるが、今後の研究がどの方向に進められるかを、応用の観点から考えたい。

5.1 収束性問題

実用的な時間内で収束するアルゴリズムで、かつ収束を保証するものが望まれる。必ずしも絶対収束が求められる訳ではなく、実際上の回路について収束が保証されれば良い。さらに解析時間を短縮するためには、解曲線が追跡しやすいものである必要がある。この観点から、球面法などの報告がなされている。

5.2 積分方法、Timestep Control

実際の回路では回路状態はダイナミックに変化するため積分公式の次数や、陽／陰的積分など、状態に応じて積分方式を変え、短時間でもっとも精度が得られるような制御が考えられる。

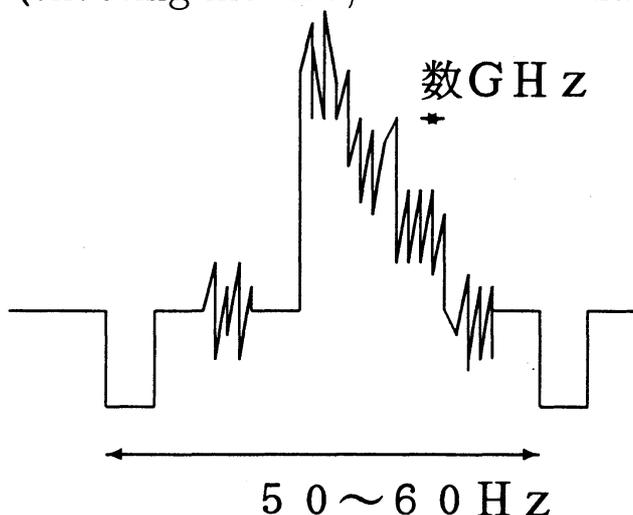
5.3 デバイスマデリング

サブミクロン領域での物理現象（量子力学的効果、チャネル

長の変動による効果)などが複雑になるにつれ、積分のために評価する関数が「高価」(expensive function)になる傾向がある。モデリングに際し、積分時の評価が「安価」であるような関数、パラメータ数であるようにモデリングを進める必要がある。

5.4 定常波解析

時定数比が大きい回路では、回路が定常状態に達するまで計算を行なうには膨大な計算量を必要とする。そこで、定常状態に達するまでの途中の(無駄な)計算を行わずに、定常状態を予測して時間応答を求めるアルゴリズムが望まれる。(shooting method, harmonic balance など)



時定数比が大きい信号例 (ビデオ信号)

5.5 並列処理

Nested Dissection, Multi-Frontal Method など、疎行列の性質を生かした並列行列処理の手法がよく知られているが、これ

らは節点をどのようにCPUに分割するかが計算速度を大きく左右する。回路特性を利用することにより、各CPU間の通信量を最小にし負荷バランス、並列効果を最大にするマッピングが期待できる。

以上のような研究領域での成果により、高速のシミュレーションシステムを構築することが期待される。