

SwCAD のヘルプ

新原 盛太郎

2012年8月

目次

		i
第 1 章	シンボルの概要	1
1.1	序	1
第 2 章	動作モード	7
2.1	概要	7
第 3 章	回路図入力	11
3.1	素子の編集	17
3.2	新しいシンボルの作成	19
3.3	階層	23
第 4 章	画像表示	27
第 5 章	LTspice	41
5.1	入門	41
5.2	ドット・コマンド	45
5.3	過渡解析オプション	64
5.4	回路素子	66
第 6 章	コントロール・パネル	103
6.1	コントロール・パネルの起動	103
6.2	圧縮	103
6.3	機能	104
第 7 章	FAQ	113
付録 A	シンボルの概要	123
A.1	電力製品	123
A.2	演算増幅器	123
A.3	その他	123
A.4	光学アイソレータ	123
A.5	特別な関数	123
A.6	フィルタ製品	123
A.7	デジタル	123

A.8	比較器	123
-----	---------------	-----

第 1 章

シンボルの概要

ここでは、LTspice が供給する回路図シンボルについて述べています。

1.1 序

1.1.1 前書き

他の SPICE が、必要でしょうか。

アナログ回路シミュレーションは、アナログ IC 設計にとって切っても切り離せないものです。SPICE シミュレータは、チップ上に形成する前に回路を試験する唯一の方法です。さらに SPICE シミュレーションは、他の手段では実行することの出来ない電流や電圧の測定を可能とします。これらのアナログ回路シミュレータの成功は、回路基板のシミュレーションにも適用されました。多くの場合ブレッドボードよりも、シミュレートすることが簡単です。そして機能や問題点のシミュレーションにおいて回路を解析する能力は、きちんと理解された強靱な回路の設計を高速に行えます。

多くの商用 SPICE シミュレータを用いるならば、どの様にシミュレータは記述されるべきでしょうか。あるアナログ関数は、商用の SPICE シミュレータでシミュレートすることが極端に困難です。スイッチ・モード電源は、ゆっくりと動作する全閉路応答と同時に高速にスイッチする方形波があります。このことは、スイッチング・レギュレータの全体応答を見るために 100,000,000 回シミュレーションしなければならないことを示しています。商用の SPICE シミュレータは、普通に用いられている手法で実行し、多くの時間を費やしています。スイッチ・モード電源のシミュレーション時間は、役に立つシミュレーションとして時間ではなく分の単位でなければなりません。

スイッチ・モード電源シミュレーションを高速に行えるアナログ・シミュレーション方法がありました。しかし簡略化の手間において、任意の制御ロジックを実行できませんし、スイッチング波形が複雑であるところを十分にシミュレートすることもできません。スイッチ・モード制御を実現する論理的な原理が搭載された新しい SPICE は、より良い結果を生み出します。これは、高速なシミュレーションや詳細な波形を与え、任意の回路変更に対し融通の利く機能を与えます。

SwitcherCAD III は、ボード・レベルのスイッチング・レギュレータ・システムをモデル化するために開発された新しい SPICE です。新しい SPICE に組み込まれたこととして、実際のボード・レベル部品をモデル化した素子があります。容量やインダクタは、直

列抵抗やその他のサブサーキットあるいは内部ノードを用いなくて、それらの寄生特性をモデル化することが出来ます。またシミュレーション回路素子は、サブサーキットや内部ノードを用いることなくゲート電荷の特性を正確に示すパワー MOSFET を導入しています。ノードの数を減らすために、シミュレータはスイッチング波形の精度あるいは詳細を曖昧にすることなく、実行するシミュレーション時間を減少します。もう一つの新しいシミュレーション・デバイスの長所は、基板レベル素子のようにモデルが全ての周波数において有限のインピーダンスを持つことによって、収束問題を回避しています。

現代のスイッチ・モード電源は、多くの動作モードを持つ制御された論理を含んでいます。例えば、デバイスは回路動作に応じてパルス・スイッチ変調からバースト・モードあるいは繰り返しスキップへと変化することが出来ます。オリジナルの新しいミックス・モード・コンパイラやシミュレータは、SwitcherCAD III に記述されています。このシミュレータは、計算速度を上昇させる現実のモデルとするため、上記の内容が含まれています。

SwitcherCAD III において約 700 の Linear Technology の技術があります。このプログラムは、Linear Technology ウェブサイトから自由にダウンロードすることが出来ます。そしてこのプログラムは、高機能な汎用のシミュレータです。このプログラムに含まれているのは、次のようなデモンストレーション用ファイルがあります。ステップ入力の負荷特性、サイクルごとのスタートアップおよび過渡応答特性などです。SPICE に含まれるのは、全ての機能を持つ新しい回路入力のための回路図入力です。

SwitcherCAD III は、設計エンジニアの三つのタイプによって用いられるように設計されています：設計者が行っていることを知る、設計者が知るべきことを考える、スイッチング・レギュレータについて何も知らないと感じているタイプです。経験豊かな設計者は、what if プログラムを必要としています。これは最適な設計を見つけるため、回路を素早く変更することです。初心者は、最も簡単な入力に基づいた、信頼おける設計による料理本的なアプローチを必要としています。中堅の技術者は、彼の自由意志を実行してくれるプログラムを求めています。しかしこの SwitcherCAD III は、致命的な流れから彼を救い出す優れたプログラムでしょう。

終わりに、我々は、非常に柔軟性を持った what if な電子設計ツールである SwitcherCAD III を作りました。我々は、基本的に電圧や電力の要求に基づいた完全な初期設計サイクルを持つプログラムを書きました。このことは、巧く機能している回路で始めることの出来る恐るべき設計者を実現します。回路を変形することに躊躇しない経験豊かな設計者とします。そして我々が期待することは、良くない設計を防ぐ十分なセーフガードを供給することです。

しかし SwitcherCAD III は、トータル・ソリューションを目指してはいないことに注意して下さい。設計手順を簡単にする道具にすぎません。設計手順には、ブレードボードやテストも含めなければいけません。シミュレーション結果で共通の認識を使って下さい。

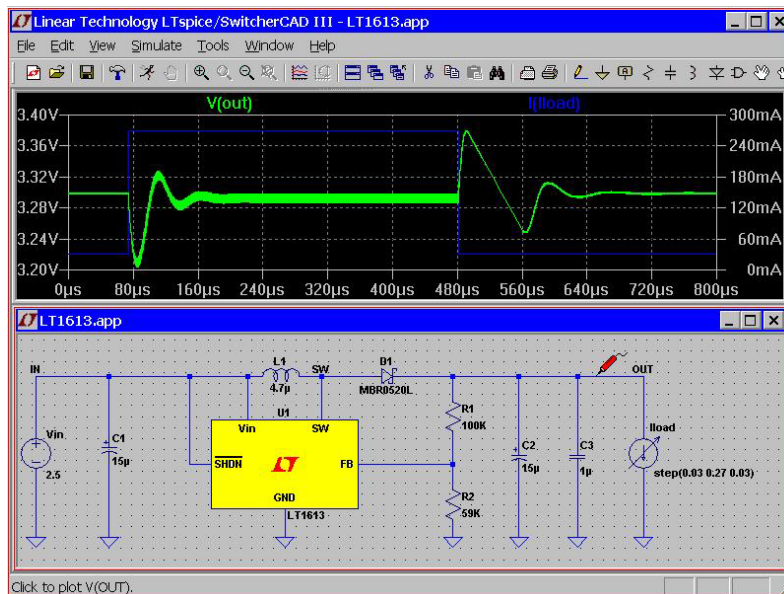
1.1.2 SwitcherCAD III 概観

SwitcherCAD III は、Linear Technology 社が開発した、第三世代スイッチング・レギュレータ設計プログラムです。このプログラムは、マクロモデル Switch Mode Power Supply(SMPS) コントローラおよびレギュレータの新しい SPICE 素子を含んだミックス・

モード・シミュレータを拡張した高機能な SPICE シミュレータから構成されています。このプログラムは、使用者が編集することの出来る SMPS 回路あるいは新しい回路設計を行うことが出来る一つに纏められた改装された回路図入力プログラムを含んでいます。含まれている波形ビューアーは、シミュレーションした波形を表示し、シミュレーション・データを解析することが出来ます。ほとんどの Linear Technology のパワー IC のデータ・ベースが含まれていますし、多くの受動素子が含まれています。素子データ・ベース、回路図編集、シミュレーション制御それと波形解析が、一つのプログラムに含まれています。

ミックス・モード・シミュレーション機能とこれまでの SPICE プログラムを越えた多くの機能により、シミュレーション速度は、精度を保ったまま飛躍的に改善されました。詳細なサイクルごとの SMPS シミュレーションは、数分の内に実行され解析することが出来ます。使用者は、素子、SPICE あるいは回路図入力プログラムを知らなくても数回マウスをクリックするだけで、パワーシステムの詳細な解析結果を得ることが出来ます。合成されたあるいは前もって記述されたデモ回路が、異なった電源要求に適合するように回路を再構築するためのスタートとして用いることが出来ます。新しい回路図を作成した後、シミュレーションを実行し報告書を作成することが出来ます。

プログラムに一体として組み込まれた階層化回路図入力や SPICE シミュレータは、汎用の用途として問題なく使うことが出来ます。SPICE シミュレータ・エンジンの改良された機能は、汎用のアナログ回路シミュレーションに有効ですし、全ての電子回路技術者に興味を抱かせるでしょう。すでに 500,000 コピー以上が配布されていますし、多くの使用者が次のように報告してくれています。LTspice/SwitcherCAD III は、主に使っているシミュレータであり回路図入力ツールだと。我々は皆さんがこのプログラムを使うのを喜び、役に立つことを期待しています。



1.1.3 必要なハードウェア

LTspice/SwitcherCAD III は、PC で動作する windows 95, 98, 2000, NT4.0, Me, あるいは XP 上で動きます。windows 3.1 あるいは DOS 上では、動きません。シミュレーションは、数分間に数メガバイトのデータを生成することが出来ますので、ハードディスクの空きスペース (>200MB)、RAM (>128MB) が必要です。基本的にプログラムは、windows 95 あるいはそれ以上を用いた任意の PC 上で動作します。しかしシミュレーションは、もしハードディスクの十分な空き容量がないと完了しないかもしれません。

LTspice/SwitcherCAD III は、Linux 上でも動きます。このプログラムは、WINE version 20030219 を用い RedHat 8.0 上で試験されました。

1.1.4 ソフトウェアのインストール

SwitcherCAD III は、LTC ウェブサイト <http://www.linear.com/software> からダウンロードすることが出来ます。配布されているファイルへの直接のリンクは、<http://ltspice.linear.com/software/swcadiii.exe> です。このファイル swcadiii.exe は、自動解凍 gzip ファイルで、SwitcherCAD III をインストールします。

SwitcherCAD III は、しばしばアップデートされます。SwitcherCAD III が最初にインストールされた後、組み込まれたメニュー・コマンドを使うことが出来ます。ウェブに接続しますと、現在のバージョンをインストールしようとしています。アップデート・プロセスは、Linear のウェブサイトから最初にマスター・インデックスをダウンロードします。それは配布されたファイルの大きさとチェックサムです。もしファイルに抜けがあると、つまり大きさが違っていたりインデックス・ファイルのチェックサムと異なっていると、ファイルが自動的にアップデートされます。素子のデータベースは、インストールされている素子であるならばアップデート過程で実施されます。これらの素子は、アップデートで失われることはありません。

1.1.5 ライセンス

SwitcherCAD III--License Agreement/Disclaimer

Copyright © 2001, Linear Technology Corporation

All rights reserved.

SwitcherCAD III は、Linear Technology Corporation のスイッチ・モード電力源合成およびアナログ回路シミュレーション・ソフトウェアです。

このソフトウェアは、著作権があります。皆さんは、LTC 製品を自分のためにのみ、人に渡さず、別の製品に適用したりしない限り自由に使うことが出来ますし、汎用の回路シミュレータとして無料で実行することが出来ます。しかしソフトウェアを変更したり、何かへ変換したり、逆コンパイル、逆アセンブルしたりすることは出来ません。LTC あるいは使用者によって提供されるシミュレータの中で用いられている第三者のモデルの精度に

ついて、何の責任も持ちません。

SwitcherCAD III の機能を補償するためあらゆる努力をしていますが、エラーが含まれていないことを補償するものではありません。このプログラムのアップグレード、変更、修正は、LTC の判断で厳格に行われます。LTC 製品を選択し評価する目的のために SwitcherCAD III をインストールし動作させるときに問題が生じる場合、次の所に電話をし、技術サポートを受けることができます。

Applications Department at (408) 432-1900, between 8:00 am
and 5:00 pm Pacific time, Monday through Friday

しかし LTC 製品の評価ではなく、一般的な回路シミュレーションについての技術的サポートは受け付けられません。非常に多くの PC 互換性コンピュータや OS バージョン、その他の用法のため、それら全てのシステムについて SwitcherCAD III の完璧な動作を保証するものではありません。SwitcherCAD III を使うことが出来ないようなら、LTC は必要性によっては LTC スイッチング・レギュレータ IC の設計サポートを行います。

ソフトウェアおよび関連資料は、現状あるがままとして供給され、何の保証もありません。Linear Technology Corporation は、全ての保証を行いませんし、公表しません。しかし特定の目的に対して商業上の限られた保証は行っています。詳細にはありませんが LTC は、このソフトウェアの使用によってあるいは実行不可能な直接あるいは関係したダメージに対して責任を持っています。その様なダメージの関係したことに対して修正します。このソフトウェアの再配布は、全ての資料例えばファイル、シンボルあるいは変更したり付け加えられていないを資料添付して、完全な形である限り許可します。

このプログラムは、特に製品の宣伝やデモンストレーションあるいは販売で半導体製造メーカーによって用いることは出来ません。これらの応用に対して SwitcherCAD III を用いるためには、特別な許可を Linear Technology から受けなければなりません。

第 2 章

動作モード

2.1 概要

SwitcherCAD III には、シミュレーションを動かす二つのモードがあります。

1. 一般の目的を持つ回路図入力プログラムとして、統合されたシミュレータを用いること。メニュー・コマンド File=ꞆNew, それと File=ꞆOpen (ファイル・タイプ .asc)
2. 手入力のネットリストとか異なった回路図入力ツールで作成した別の種類のネットリストを用いること。メニュー・コマンド File=ꞆOpen (ファイル・タイプ .cir)

LTspice/SwitcherCAD III は、汎用の目的を持った回路図入力プログラムとして、統合された SPICE シミュレータをを供給するものです。理想的には自分で回路図を描き（あるいは既に用意された例題回路で初め）そしてシミュレーションでその動作を観察することです。望んだ回路動作がシミュレーションで達成されるまで繰り返し回路設計を行います。LTspice/SwitcherCAD III の初期のバージョンは、ユーザーが供給する特性から SMPS 設計を導くような合成器を含んでいました。しかしその機能のモードは止めになりました。

回路図は最終的にシミュレータが理解できるテキスト形式の SPICE ネットリストへ変換されます。一方ネットリストは、LTspice/SwitcherCAD III で描かれた回路図から通常抽出されます。インポートされたネットリストは、回路図を作ることなく直接走らせることができます。このことはいくつかの方法があります。(i) リニア・テクノロジーのフィルタ合成プログラム FilterCAD は、LTspice がフィルタの時間軸や周波数軸応答のネットリストを合成することができます。(ii) 他の SPICE プログラム対してよりも LTspice の方が簡単です。(iii) SPICE 回路シミュレータに対して経験豊かな人は、テキスト形式のネットリストで直接にシミュレーションすることに慣れているでしょう。と言うのも昔のシステムでは回路図入力が SPICE シミュレータに含まれていなかったからです。

2.1.1 回路例

LTspice/SwitcherCAD III には、いくつかの例題回路が入っています。ディレクトリ c:

Program files

LTC

SwCADIII

examples

Educational にあります。これらの例題回路は、異なった解析タイプや方法あるいはプログラム形式の非商用回路例が入っています。C:

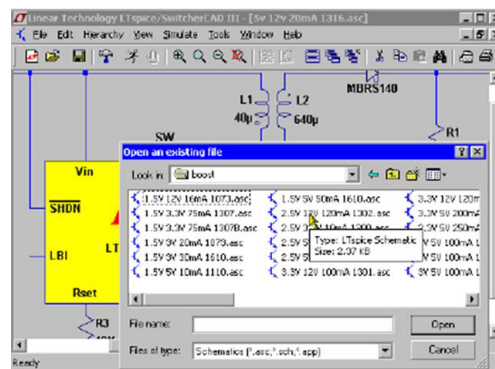
ProgramFiles

LTC

SwCADIII

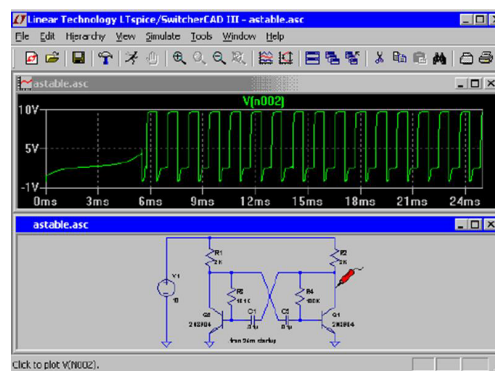
examples

jigs には、LTspice/SwitcherCAD III のマクロモデルである全ての Linear Technology デバイスの例題シミュレーションがあります。これらの jig 回路は、マクロモデルのテスト jig であって必ずしも推奨する設計ではありません。最も重要なことは、Linear Technology オフィスが貴方の必要な設計サポートを供給することが出来るということです。



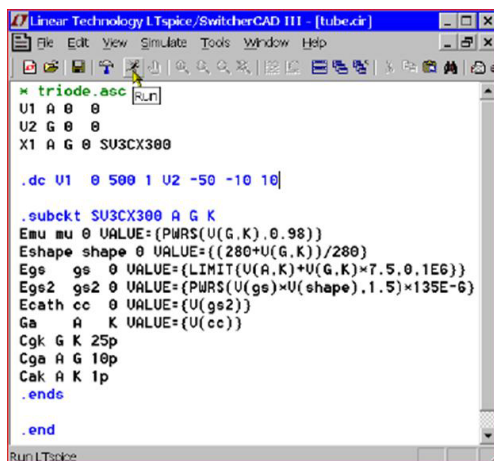
2.1.2 汎用回路図入力 SPICE

LTspice/SwitcherCAD III は、無料の汎用の回路図入力、SPICE プログラムです。このプログラムは、SMPS 設計のみならず多くのアナログ技術の面に対して役に立ちます。例題回路は、ディレクトリ C:/program Files/LTC/SwCADIII/examples/Educational に様々な LTspice の可能性を示してあります。



2.1.3 外部作成ネットリスト

手入力あるいは他の回路図入力プログラムによって作られるネットリストを開くことが出来ます。これらのファイルの拡張子は、通常.cir となっていますが.net や.ap も認識します。ネットリストファイルに用いられている ASCII エディターは、無制限のファイルサイズをサポートし undo/redo にも制限がありません。メニュー・コマンド Tools->Color Preferences は、ASCII エディターで用いられる色を調整するために用いられます。



```

* triode.asc
U1 A 0 0
U2 G 0 0
X1 A G 0 SU3CX300

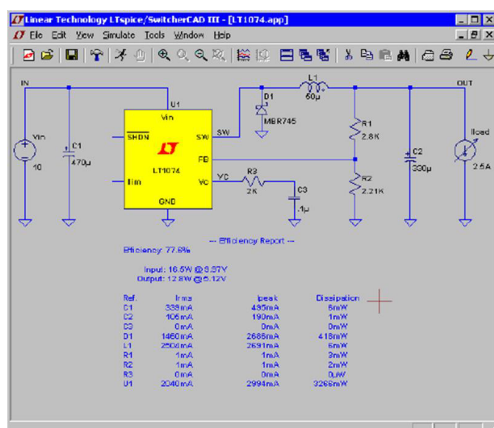
.dc U1 0 500 1 U2 -50 -10 10

.subckt SU3CX300 A G K
Emu mu 0 UVALUE={PWRS(U(G,K),0.96)}
Eshape shape 0 UVALUE={{(280+U(G,K))/280}
Egs gs 0 UVALUE={LIMIT(U(A,K)+U(G,K)*7.5,0.1E6)}
Egs2 gs2 0 UVALUE={PWRS(U(gs)*U(shape),1.5)*135E-6}
Ecath cc 0 UVALUE={U(gs2)}
Ga A K UVALUE={U(cc)}
Cgk G K 25p
Cga A G 10p
Cak A K 1p
.ends

.end
  
```

2.1.4 有能な報告書

キーワード”定常”を含む時間領域.tran 解析から DC-DC コンバータの有能な報告書を得ることが出来ます。定常状態シミュレーションの後、有能な報告書がコメント・テキストのブロックとして回路図上に示すことが出来ます。



DC-DC コンバータの能力は、次の方法で導かれます。入力と出力を区別するため一つの電圧源と一つの電流源がなくてはなりません。電圧源は入力に、そして電流源は出力と仮定されます。回路はシミュレータによって定常状態が検知されるまで走ります。このことはどの様にして定常状態を検知するかという情報を記述する SMPS マクロモデルが必

要です。通常このことは、エラー・アンブ電流が一つのクロック・サイクルの平均値を取るとき数サイクルの間に小さい値に減少するように検知されます。そこでクロックの端で各リアクタンスに蓄えられるエネルギーが書き留められます。そしてシミュレーションは次の 10 サイクルを走ります。しかし全てのデバイスの電力消費を保存していきます。最後のサイクルにおけるクロックの端で、全てのリアクタンスに蓄えられているエネルギーが書き留められシミュレーションは止まります。効率は、リアクタンスに蓄えられたエネルギーの変化を調整した後、供給される電力対入力電圧から供給される電力による比として与えられます。各デバイスの消費電力もまた記録されるので、全てのエネルギーの和がいかにゼロに近いかを見ることが出来ます。

エディット・コマンド・エディター上で”Stop simulating if steady state is detected”をチェックして描いた SMPS 回路の効率を常に計算することが出来ます。シミュレーション後にメニュー・コマンド View⇒Efficiency Report を用いて下さい。

定常状態の自動検知は必ず働いているわけではありません。時々定常状態検知基準は、非常に厳密でしかも時々寛大であります。そこでオプション・パラメータ SSTOL を調整するか単に効率の制限を設定して下さい。

2.1.5 コマンドライン・スイッチ

フラッグ	記述
-ascii	ASCII .raw ファイルを用いる。プログラム機能を低下させる。
-b	バッチモードで走らせる。例えば”scad3.exe -b deck.cir”は、deck.raw でデータを残す。
-big	最大ウィンドウで走らせる。
-	モデル・ライブラリを暗号化する。第三者などに対して。
encrypt	詳細な道具を見せびらかすことなくライブラリを用いるようにする。リニア・テクノロジのモデルでは使われていない。
-	バイナリ .raw ファイルのバッチ変換
FastAccess	アクセス・フォーマットを急がせる。
-max	-big と同じ。
-nowine	WINE(Linux) の使用を妨げる。
-	PCB へのパッチによる回路図変換。
PCBnetlist	ネットリストをフォーマットする。
-	LTspice をユーザーの希望値に保存する、MRU
y	%EINDIR—%/scad3.ini ファイル
-run	run ボタンを押さずにコマンド・ライン上に回路図を開きシミュレーションする。
-SOI	サブサーキット展開で MOSFET が七つのノードまで持つことを許す。
-	アンインストールのステップを実行する。
uninstall	過程
-web	メニュー・コマンドを実行することに等しい。
update	Tools⇒Sync リリース。
-wine	WINE(Linux) を実行する。

第 3 章

回路図入力

3.0.6 基本的な回路図編集

回路図入力プログラムは、新しい回路図を作成したり既存の例題回路を変更したりします。回路サイズと階層構造は、使っているコンピュータのリソースのみに依存します。

このソフトウェアは、約 800 のシンボルが含まれています。これらのシンボルは、回路設計のために次のようなものが含まれています。LTC パワー IC、オペアンプ、コンパレータ、そして汎用デバイス。このプログラムへインポートしたい自分の素子シンボルを描くこともできます。

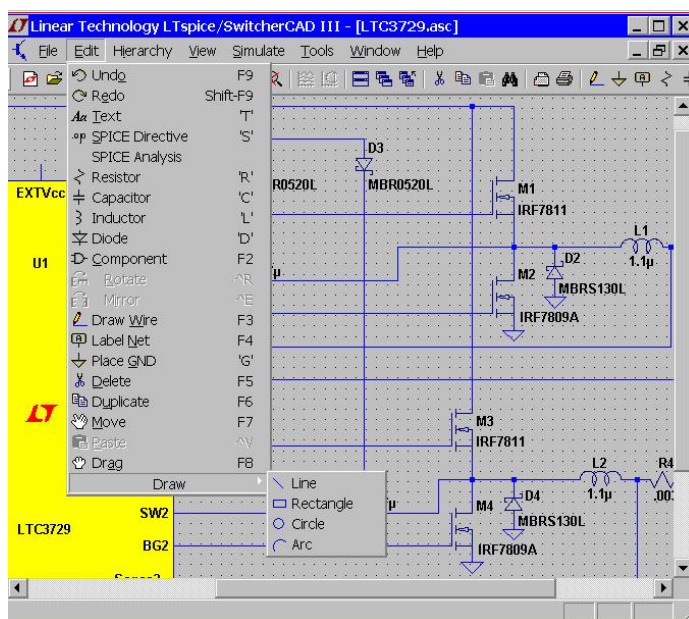


図 3.1 回路図編集

多くの回路図入力プログラムと違って、このプログラムは SPICE シミュレーションを走らせるために特別に書かれています。つまり対象をクリックするとワイヤ上の電圧や素子の電流を表示するようになっています。編集のために選択するのではありませんし、シミュレーションを行わないようにビヘービアを選択するのでもありません。そのため移動する、鏡像を求める、回転させる、素子を消すなどの動作を行うためには、最初に move、

drag、delete コマンドを選択する必要があります。その後素子の上でクリックし、選択することが出来ます。箱をドラッグすることによって、多数の素子を選択することが出来ます。右のマウスボタンをクリックするあるいは **Esc** ボタンが押されるまで、この状態が続きます。全ての回路図編集で **undo、redo** を行うことが出来ます。

Undo 最後のコマンドを **Undo** します。

Redo 最後の **Undo** コマンドを **Redo** します。

Text 回路図上にテキストを配置します。これは単に情報を持った図を反映しているだけです。

SPICE Directive 図の上でネットリストに含まれているテキストを配置します。このコマンドは、回路図入力と **SPICE** ネットリストとを一緒にします。シミュレーション・オプションを設定したり、モデルを含んだファイルを含めたり、新しいモデルを定義したりあるいは他の有効な **SPICE** コマンドを用いたりします。シンボルを持たないサブサーキット (**SPICE** 上では **X** で始まります) を、回路図上にモデルの特性を記述することによってまた定義を含ませることによって走らせることが出来ます。

SPICE Analysis シミュレーション・コマンドに入り編集すること。このコマンドでは、回路合成から直接回路を走らせることは出来ません。始め回路を拡張子 **.asc** で保存し、再び開いてシミュレーションコマンドを編集します。

Resistor 回路図に新しい抵抗を配置します。

Capacitor 回路図に新しい容量を配置します。

Inductor 回路図に新しいインダクタを配置します。

Diode 回路図に新しいダイオードを配置します。

Component 回路図に新しい素子を配置します。このコマンドは、シンボル・データベースを示し編集するためのダイアログボックスを表示します。このダイアログボックスは、抵抗、容量、インダクタそれとダイオードについてのコマンドの汎用的な書式となっています。

Rotate 選択された素子を回転します。素子が選択されていないときは、灰色となっていることに注意して下さい。

Mirror 選択された素子を鏡映します。素子が選択されていないときは、灰色となっていることに注意して下さい。

Draw Wire 左マウスボタンをクリックし、配線をスタートします。各マウスクリックは、新しいワイヤを定義します。既に存在しているワイヤと新しいワイヤとを接続するには、存在しているワイヤ上でクリックします。クリック一回は、現在のワイヤ描画をキャンセルします。もう一度クリックしますと、コマンドを中止します。抵抗の場合と同じように、素子を通過してワイヤを引くことが出来ます。抵抗がワイヤと直列になっているこのワイヤは、自動的にカットされます。

Label Net ノード名を特定します。その結果任意のノードは、ネットリストによって示されません。

Place GND **GROUND** シンボルを配置します。ノード **0** で、全回路共通となります。

Delete 素子の上でクリックするあるいは、素子の上に **Box** をドラッグして素子を消去します。

- Duplicate** 素子の上でクリックし、あるいは **box** をドラッグして素子を倍にします。LTdpice/SwitcherCAD III を用いて開いている回路図から別の回路図へ、コピーすることが出来ます。最初の回路図ウインドウの **Duplicate** コマンドで始めます。次に二番目のウインドウで **Ctrl-V** を押します。
- Move** 動かしたい素子周りのボックスをクリックしドラッグし、次に新しい場所へ移動します。
- Paste** 素子が **Duplicate** コマンドで選択されているとき新しい回路図ウインドウの中で、**paste** することが出来ます。
- Drag** drag したい素子の周りの **box** をクリックしドラッグします。次に新しい場所へ素子を移動することが出来ます。そのときワイヤは、新しい位置までラバーバンドします。
- Draw => Line** 回路図上でラインを引きます。この様なラインは、回路図上で電氣的な影響を与えません。しかしメモの付いた回路として有効です。
- Draw => Rectangle** 回路図に直角を描きます。直角は、回路図上で電氣的な影響を与えません。しかしメモの付いた回路として有効です。
- Draw => Circle** 回路図に円を描きます。円は、回路図上で電氣的な影響を与えません。しかしメモの付いた回路として有効です。
- Draw => Arc** 回路図に円弧を描きます。円弧は、回路図上で電氣的な影響を与えません。しかしメモの付いた回路として有効です。
- Note** 回路図へのグラフ的な注釈；ライン、直角、円および円弧；デフォルトとしてワイヤやピンの電氣的接点に用いられている同じグリッドに接触します。この接触を行わないためには、位置を保ったままコントロール・キーを押し続けて下さい。

3.0.7 ノード名のラベル

各ノードは、回路図の中で単独の名前でなければなりません。ネットリストが付けた名前と、別の名前を付けることが出来ます。ノード 0 は、回路のグローバルな名前です。ノードの名前 0 の代わりに、特別な描画シンボルとして書くことが出来ます。

ノード **COM** に対して定義された、描画シンボルもあります。これは **SPICE** のグローバルな共通シンボルではありませんし、グローバルなノードでもありません。グラウンドとは区別された描画シンボルとして、便利さにより用いられています。

最初の文字として **\$G_** を用いる、例えば **\$G_VDD** は、ノードがグローバルであり、階層回路に用いられます。

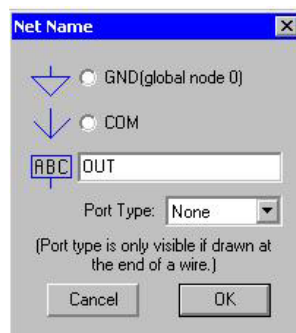


図 3.2 ノード名のラベル

入力、出力あるいは両方向の端子であることを示すことができます。これらの端子は、異なった形状で描かれますが、ネットリストに対しては意味がありません。端子のタイプを示すことは、回路図を見やすくするためです。グローバルノードも、異なった形状で描かれます。これらの名前は、box で囲まれた形で描かれます。



図 3.3 ノード

3.0.8 回路図の色

メニュー・コマンド Tools => Preferences colors は、回路図表示に用いられる色を設定します。例題回路図の素子をクリックし、赤、緑、青のスライダーを好みの色に合わせて下さい。^{*1}

^{*1} 電氣的に関係のないグラフ例えばラインや円などは、素子と同じ色で描かれます。

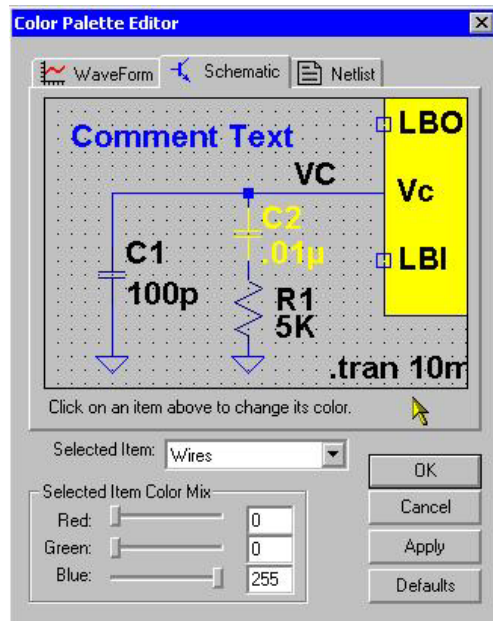


図 3.4 回路図の色

3.0.9 新しい素子の配置

よく用いられる素子；例えば抵抗、容量そしてインダクタ；これらは、ツールバー・ボタンを使って回路図上に配置するべく選ばれます。

ほとんどのシンボルに対して、メニュー・コマンド Edit => Component を用い、必要な素子を検索するためのダイアログ・ボックスをスタートします。

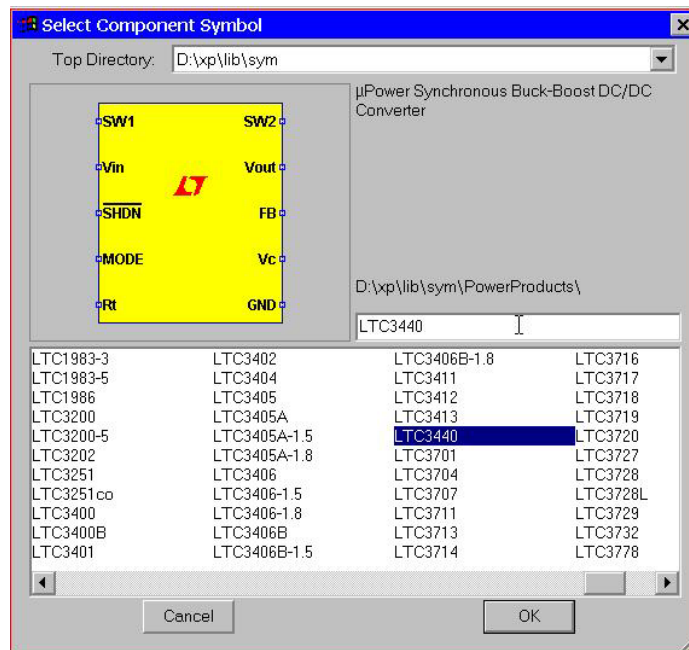


図 3.5 素子の配置

3.0.10 プログラム用キーボード・ショートカット

メニュー・コマンド Tools => Control Panel => Drafting Options => Hot Key は、ほとんどのコマンド用キーボード・ショートカットをプログラムすることが出来ます。コマンドの上で単にマウスをクリックし、キーを押すかコマンドにしたいコードのキー・コンビネーションを押します。ショートカットを削除するには、コマンド上でクリックし Delete キーを押します。

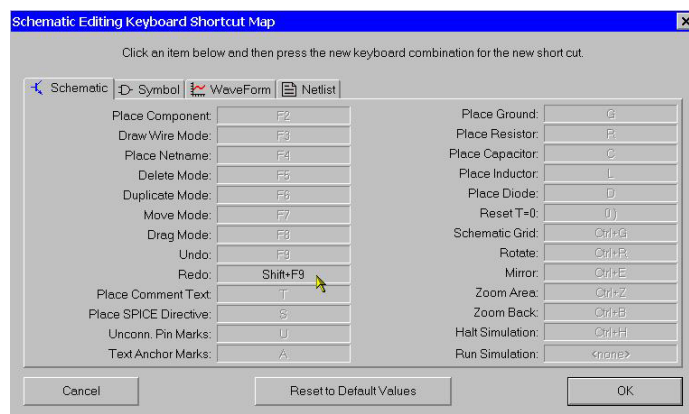


図 3.6 ショートカット・キーの変更

3.0.11 PCB ネットリストの抽出

回路図メニュー・コマンド Tools => Export Netlist は、PCB レイアウトの ASCII ネットリストを生成します。ピン・ネットリスト順と同じ順番となっているシンボルのセットを作っておかねばなりません。例えば LTspice 回路図ネットリストを ExpressPCB へインポートしたいのであれば、使いたい全てのシンボルと同じ順番を持つネットリストの LTspice あるいは ExpressPCB のシンボル・セットを作らなければなりません。その他ダイオードは、ネットリストを反映させることが出来ますし、トランジスタの配線の接続は、入れ替えることが出来ます。

次のフォーマットが使えます：Accel,Algorex, Allegro, Applicon Bravo, Applicon Leap, Cadnetix, Calay, Calay90, CBDS, Computervision, EE Designer, ExpressPCB, Integraph, Mentor, Multiwire, PADS, Scicards, Tango, Telesis, Vectron, and Wire List.

3.1 素子の編集

3.1.1 素子編集概要

素子は、素子タイプにより二つあるいは三つの異なった方法で編集することが出来ます。

1. ほとんどの表示される素子の特性は、マウスを用いて右クリックして編集することが出来ます。マウスのカーソルは、テキストのカレットに変化しテキストをしめすことが出来ます。
2. 多くの素子タイプ、例えば抵抗、容量、インダクタ、ダイオード、バイポーラ・トランジスタ、MOSFET トランジスタ、JFET トランジスタ、独立電圧源、独立電流源それと階層回路ブロックは、特別な編集機能を持っています。これらの編集機能は、デバイスの適切なデータ・ベースを扱うことが出来ます。これらの機能を扱うには、素子本体の上で右のマウス・ボタンをクリックします。
3. 素子シンボルの上にマウスを置き、コントロール・キーを押し下げます、それから右のマウスボタンをクリックして下さい。ダイアログ・ボックスが現れ、全てのシンボル特性が表示されます。各々のフィールドの次で、フィールドが回路図上で表示されるようにしたいところのチェックを入れて下さい。

3.1.2 表示特性の編集

ほとんどの表示される素子特性フィールドは、マウスを用い右クリックすることによって編集することが出来ます。マウス・カーソルは、テキストを示すことによってテキスト・キャレットへ変わります。これは、素子の値を変更する種には便利な方法です。

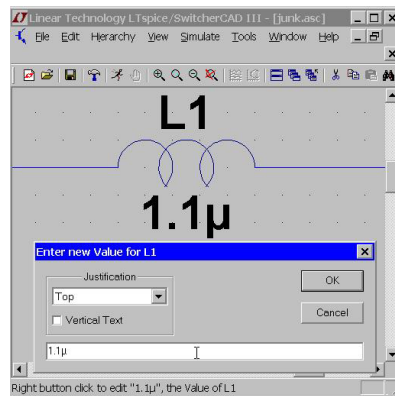


図 3.7 表示特性の編集

3.1.3 特別な素子編集

多くの素子タイプ、例えば抵抗、容量、インダクタ、ダイオード、バイポーラ・トランジスタ、MOSFET トランジスタ、JFET トランジスタ、独立電圧源、独立電流源それと階層回路ブロックは、特別な編集機能を持っています。これらの編集機能は、関係している素子の適切なデータ・ベースにアクセスすることが出来ます。これらの編集機能を用いるためには、素子の上で右のマウス・ボタンをクリックして下さい。

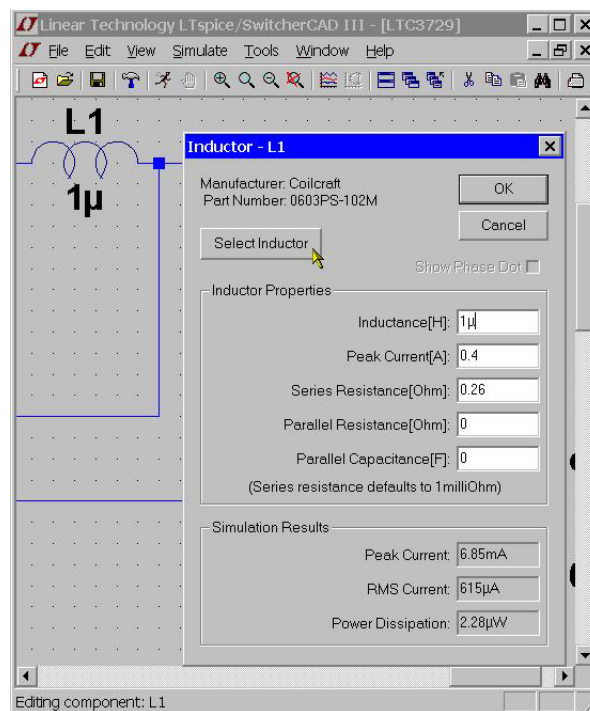


図 3.8 特別な素子編集

3.1.4 一般特性の編集

ときどき素子の内容と可視性を編集するために、全ての素子の特性について直接アクセスすることが必要になります。このことを実行するための編集機能は、素子シンボルの上にマウスを置き、コントロール・キーを押し下げ、右のマウス・ボタンをクリックすることによって実行できます。全ての有効な素子特性を表示するダイアログ・ボックスが現れます。回路図上に表示するかどうかの各フィールドに対するチェックボックスです。

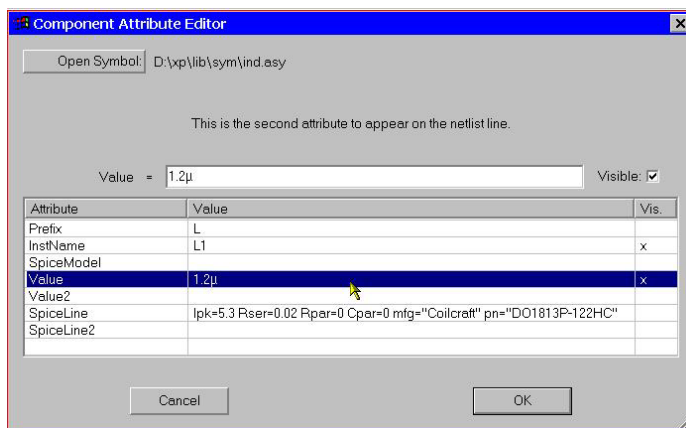


図 3.9 一般特性の編集

3.2 新しいシンボルの作成

3.2.1 シンボル作成の概要

シンボルは、基本デバイス例えば抵抗あるいは容量；別なファイルにあるサブサーキット用ライブラリ；あるいは回路図の別のページを表現することができます。ここではどのようにして自分の新しいシンボルを定義するか、とすることについて記述しています。新しいシンボルを始めるに当たって、メニュー・コマンド **File => New Symbol** を用います。^{*2}

3.2.2 本体を描く

ライン、直角、円、円弧として、シンボルの本体を描くことができます。描かれたものは、回路図上において電気的な特性を持ってはいません。**Draw => Text** コマンドを用いてシンボル上に、これも電氣的に影響を及ぼさないテキストを描くこともできます。この図形のアンカー・ポイントは、小さな赤い円で示されています。このアンカー・ポイントは、素子をドラッグするときに使われます。メニュー・コマンド **View => Mark Object Anchors** を用いて、この赤い円を表示させたりさせなかったりすることができます。

^{*2} シンボルを作成している間は、画面がアップデートされるのが遅くなります。使っておられるビデオ・カードに問題があるようでしたら、画面を再描画する速度を上昇させるためにシンボル編集画面の面積を小さくするあるいは、スクリーンの解像度を下げて下さい。それによってマウスの動きが改善されます。

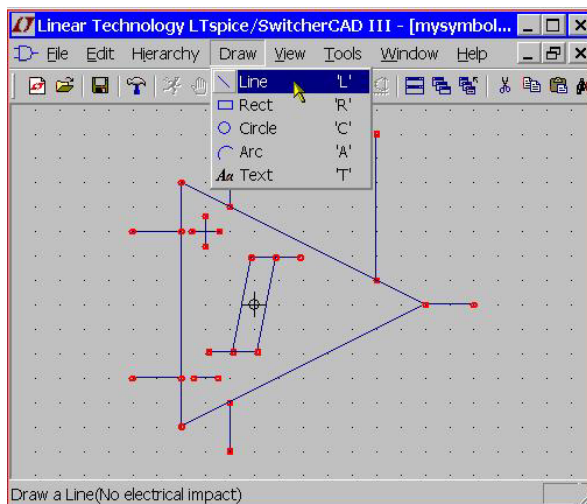


図 3.10 本体を描く

3.2.3 ピンを付け加える

ピンは、シンボルに電気的な節点を付け加えます。メニュー・コマンド Edit => Add Pin/Port を用いて新しいピンを付け加えます。

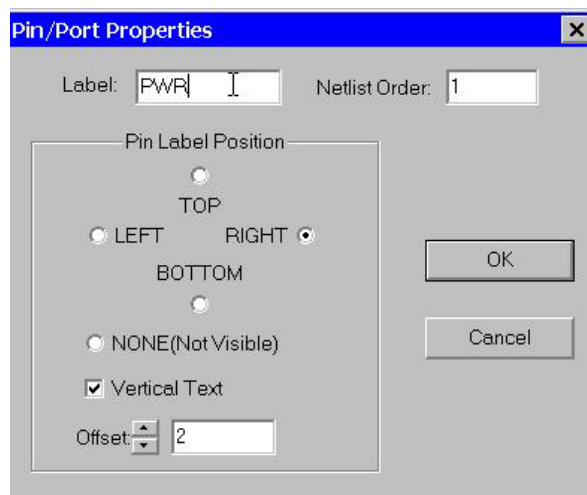


図 3.11 ピンを付け加える

Pin Label Position は、どの様にピンラベルを表示するかを決めます。TOP,BOTTOM,LEFT および RIGHT は、テキストの並びを示しています。例えばピン・ラベルが TOP となっていると、ピン（ラベルの位置を決めるアンカー・ポイント）は、ラベルを上にしします。シンボルが SPICE のライブラリにある基本素子あるいはサブサーキットを表しているとき、ピン・ラベルは回路上で電気的な特性を持っていません。しかしシンボルが回路図階層構造の下のレベルの回路を表しているときには、ピンの名前は回路図の下のレベルの回

路図における名前として重要となります。

Netlist Order は、このピンが SPICE のネットリストになる順番を決めます。

3.2.4 特性を付け加える

メニュー・コマンド Edit => Attributes => Edit Attributes を用いてシンボルのデフォルト特性を定義することが出来ます。最も重要な特性は、Prefix と呼ばれます。これは、シンボルの基本タイプを決定しています。シンボルが SPICE の基本的なところを表現しているならば、シンボルは適切な prefix、抵抗なら R、容量なら C、MOSFET なら M などを持つべきです。LTspice の完全な SPICE の基本素子については、reference を参照して下さい。ライブラリで定義されているサブサーキットを表現するシンボルを用いたのであれば、X という prefix を用いるべきです。

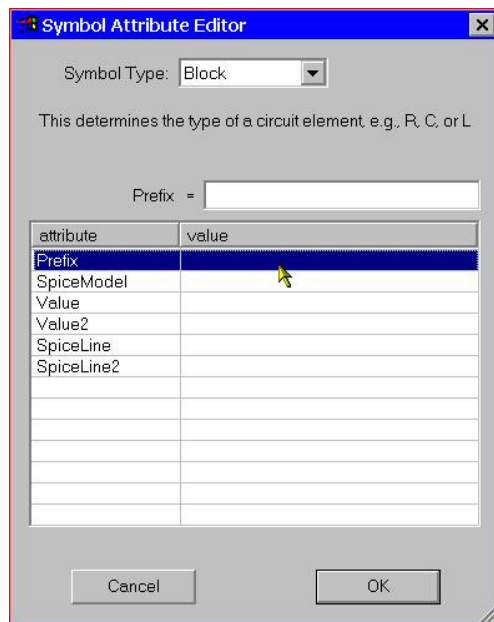


図 3.12 特性を付け加える

シンボルの特性は、回路図の成分としてシンボル特性において上書きされます。例えば prefix として M の MOSFET シンボルがあるとしますと、instance-by-instance 基準において prefix X を上書きすることが出来ます。この結果トランジスタは、サブサーキットとしてモデル化されます。

特性の特別な結合があります。これは、シンボルとして用いる各回路図において自動的に含まれる必要なライブラリとなります。

Prefix: X
 SpiceModel: <name of file including the spicemodel>
 Value: <What ever you want visible on the schematic>
 Value2: <The value as you want in the netlist>

Value2 は、spicemodel に含まれるファイルで定義されたサブサーキット名と一致するように作られます。そして付加的なパラメータをサブサーキットへ転送することが出来ます。シンボルがこの方法で定義されますと、回路図素子としてのシンボルの特性は、異なった特性を持つため編集することは出来ません。

階層的な回路図の別なページを表示するためのシンボルが必要な場合、全ての特性は、左ブランクとすべきであってシンボル・タイプは **Cell** から **Block** へ変えるべきです。何の特性値も設定する必要がありません。

特別なシンボル特性 **ModelFile** があります。これはライブラリとして、ネットリストに含まれるファイル名として用いられます。symbol/subcircuit pair /lib/sym/Opamps/1pole.asy およびこの特性のユーティリティの例を見るため/lib/sub/1pole.sub を見て下さい。prefix 特性が **X** であり、モデル・ファイルに定義されたサブサーキットとしてシンボル特性 **SpiceModel** が存在する場合、シンボルの特性が回路図上で編集されるととき全てのサブサーキットのリストは、有効となります。

3.2.5 特性の可視化

メニュー・コマンド **Edit => Attributes => Attribute Window** を用いて、特性の可視化を編集することが出来ます。このダイアログで特性を選択した後、シンボルに関係した希望の位置に置くことが出来ます。

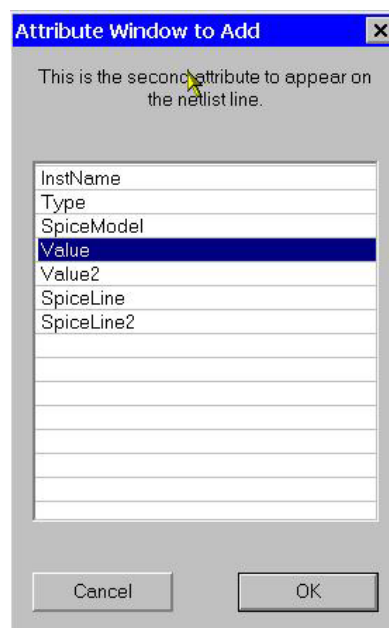


図 3.13 シンボル特性選択

特性を示すテキスト上を左クリックすることにより、既に可視化されたテキストや特性の内容を編集することが出来ます。

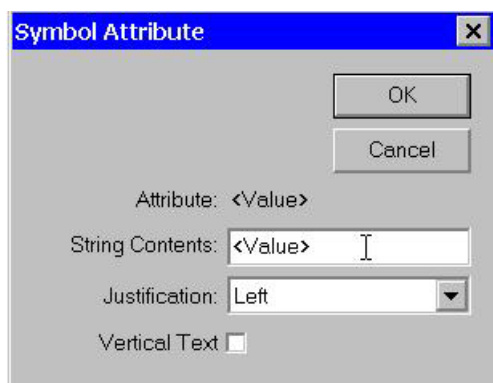


図 3.14 シンボル特性編集

3.3 階層

3.3.1 階層概論

階層的に回路図を描くことは、非常に大きなメリットがあります。非常に大きな回路は、紙の上いっばいに回路図を描くよりも、小さな明確に分かりやすい回路にして描くことができます。繰り返しの多い回路は、特殊な方法で簡単に扱えます。回路ブロックは、別のプロジェクトで再利用するため、ライブラリ化することができます。

3.3.2 階層化の規則

上の階層にあるブロックとして別の回路図として参照する方法は、ブロック図として同じ名前を持ったシンボルを作成することです。そのシンボルをより上のレベルの回路図上に置きます。例えば `topXYZ.asc` と名付けられたトップレベルの回路図 `topXYZ` の回路図に置きたい `preamp.asc` と名付けたもう一つの回路があり、それから `preamp.asy` と呼ぶシンボルを作ります。`topXYZ` の回路図上にそのシンボルを置きます。回路図間の電気的な接続は、下のレベルの回路図でのノード名に対応した下のレベルでのブロックシンボル上のピンに上のレベルの回路図のワイヤを接続することによって実現します。

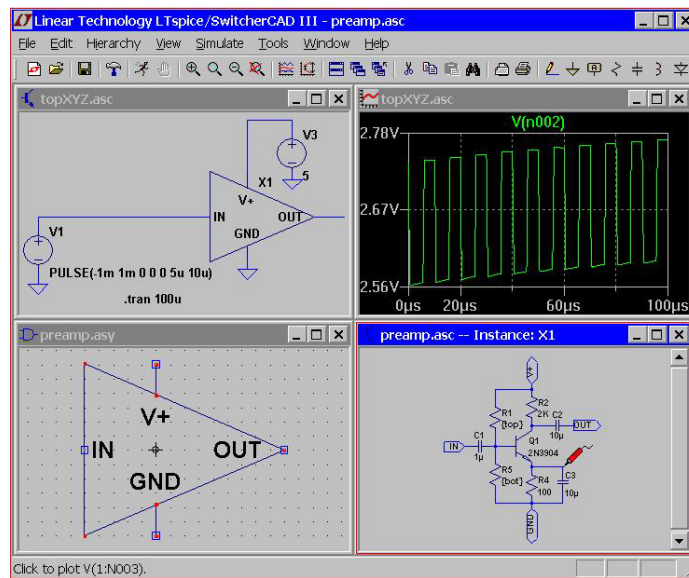


図 3.15 階層化の規則

LTspice は、シンボルやブロック図をトップレベル回路図のディレクトリにおいて回路が完全であることを検知します。

低いレベルを表現するために作ったシンボルは、特性を定義すべきではありません。

3.3.3 階層を見いだす

File => Open コマンドで開かれたファイルは、トップレベル回路と見なされます。そのブロックに SPICE ディレクトリを付け加え、それと任意の参照したい低い階層の回路図を使ってシミュレーションすることが出来ます。

より高い階層レベルの回路として回路ブロックを開くためには、最初に高い階層の回路を開き、マウスを呼び出すブロック回路の本体の上に移動します。シンボルの上で右マウスボタンをクリックすると、回路図を開くための特別なダイアログボックスが開きます。この方法で回路図を開き、ブロックの電圧と電流を見ることが出来ます。コントロール・パネルの Save Default Pane 上にあるオプション Save Subcircuit Node Voltages および Save Subcircuit Device Currents をチェックすることに注意して下さい。またトップレベル回路図上のノードを点滅させると、低い階層レベルにおいても点滅します。

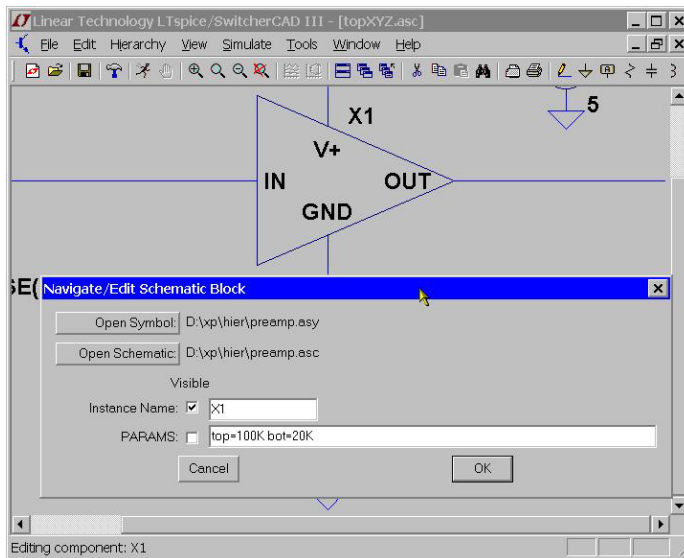


図 3.16 階層を見いだす

ダイアログは、preamp.asc 回路のブロックにパラメータを引き渡すことに注意して下さい。

第 4 章

画像表示

4.0.4 概要

SwitcherCAD III は、シミュレーション結果を表示するための完全な機能を集めた波形表示となっています。

4.0.5 波形選択

波形を選択する、三つの基本方法があります。

1. 回路図から直接選択する。
2. メニュー・コマンド Plot Settings => Visible Traces
3. メニュー・コマンド Plot Settings => Add Trace

どのような選択方法にも関わらず、選択した異なった描画を再描画するための undo や redo を行えます。

回路図から直接選択

最も簡単な方法は、単に回路図を選択することです。見たいと思うワイヤ電圧の上でワイヤを単にクリックします。任意の素子（抵抗、容量あるいはインダクタ）に流れている電流が見たいときには、その素子の上でクリックします。この動作は、階層構造のどのレベルでも有効です。多数のピンがある素子で、ある接続点に流れ込む電流を見たいときには、その素子のピンの上でクリックします。同じ電圧あるいは電流を二度クリックすると、表示されているその他の表示は消えてしまい、ダブルクリックした表示だけが描かれます。delete コマンドを選び描画されたラベル上でクリックすることにより、個別に消去することが出来ます。次の図は、どの様にピン電流を指し示すかということを示しています。電流を示すときに、マウス・カーソルは電流メータのプロープのようなアイコンに変化します。

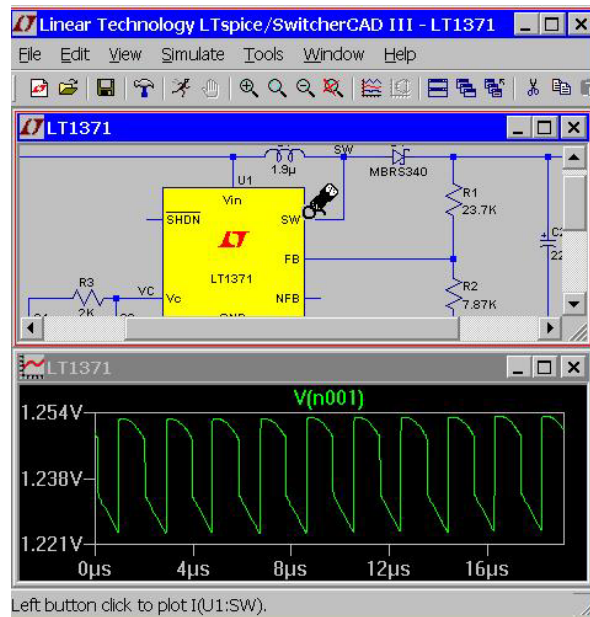


図 4.1 電流プローブの表示

ピン電流を示すとき、電流の正の方向は、通常ピンに流れ込む方向です。

マウスを用いて、電圧差を示すこともできます。一つのノードをクリックし、次にもう一方のノードへドラッグします。最初のノードで赤い電圧プローブ、二番目では黒いプローブへ変化します。こうすることにより、電圧差を見ることが出来ます。

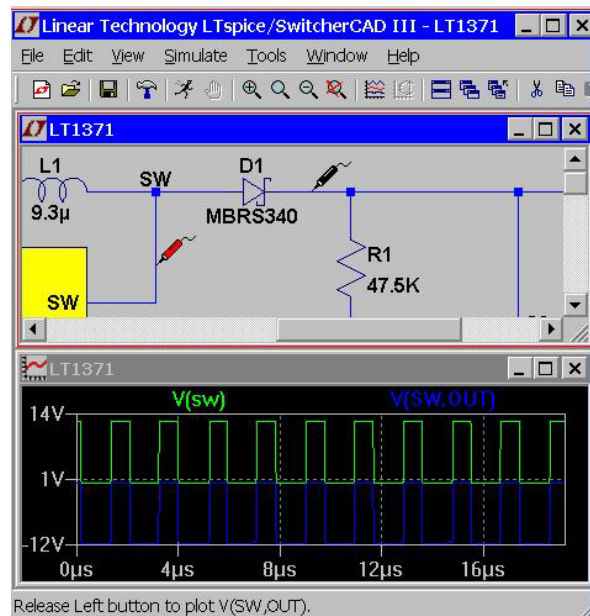


図 4.2 電圧差の測定

もう一つ別の測定方法は、素子の瞬時電力測定です。これを行うには、素子記号の本体

をクリックします。瞬時電力が電圧、電流表現として表示されます。この表示は、ワットです。消費電力を測定しているとき、マウスのカーソルは、温度計のアイコンに変わります。ラベルをクリックすると、平均電力を見ることが出来ます。

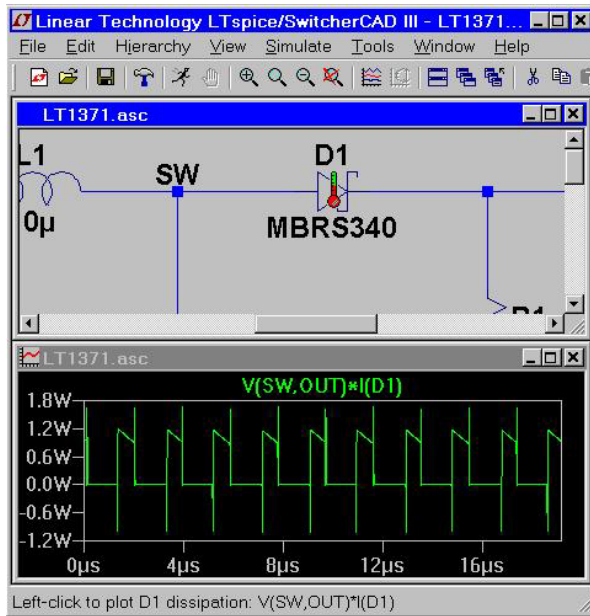


図 4.3 電力の測定

メニューコマンド Plot Setting => Visible Traces

メニューコマンド Plot Setting => Visible Traces は、シミュレーション・データを最初に表示するときに見るダイアログです。描画を始める最初の項目を選択します。描画できる全ての軌跡を示しています。

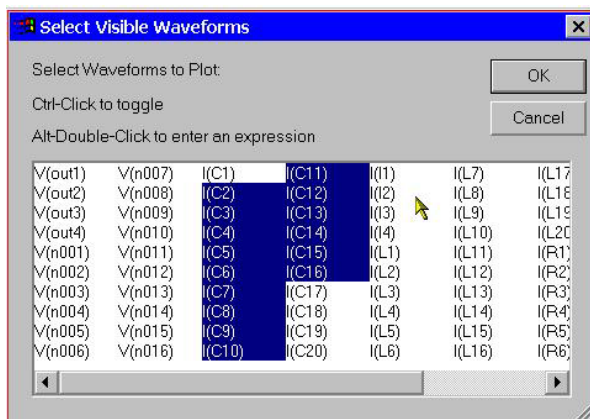


図 4.4 描画選択

メニュー・コマンド View => Add Trace

メニュー・コマンド View => Add Trace は、Plot Settings => Visible Trace コマンドに似ています。しかしそれによって描かれた描画を消去することは出来ません。二つの便利な方法があります。一つは、ダイアログのすぐ上にあるボックスを編集することです。それによって特性を入れることが出来ます。そのパターンに適合した描画の唯一の名前がダイアログの中に示されます。これは、部分的に記憶している名前前の描画を見つけだすのに便利です。また描画データの表現を完成するのにも役立ちます。名前をタイプ入力する代わりに、ダイアログ上の名前をクリックして動かすことが出来ます。

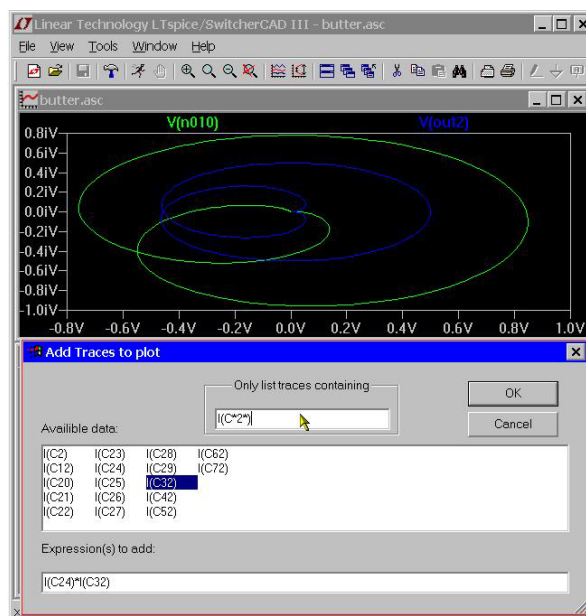


図 4.5 描画の追加

4.0.6 拡大

LTspice/SwitcherCAD III は、新しいデータが表示されるときはいつでも自動的に拡大されます。ある領域を拡大するとき、拡大して見たい領域の周りをボックスでクリックするだけです。

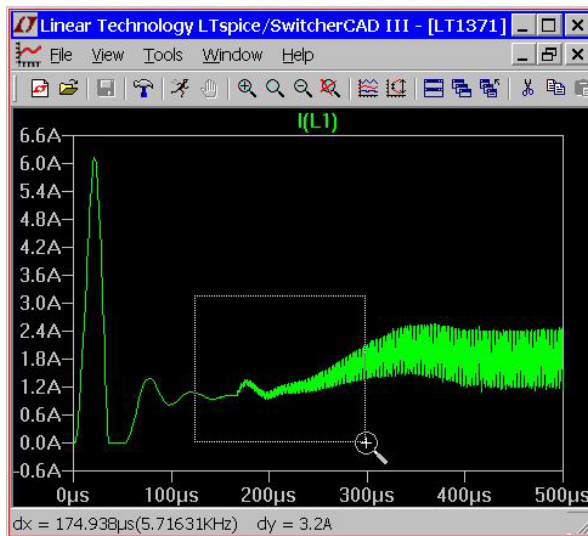


図 4.6 拡大

拡大、パンニング、それと自動ズームに戻すためのツール・ボタンやコマンドがあります。undo、redo は、違った拡大を行った後に用います。

4.0.7 波形の計算

波形のデータを演算するための、三つの数学演算方法があります。

1. 結果表現を描く
2. 描画の平均値や RMS 値を計算
3. 描画のフーリエ変換を表示

結果表現を描く

View => Visible コマンドや View => Add Trace コマンドは、データ表現を入力するために使います。有効なシミュレーション・データを描画するもう一つの方法は、描画ラベルへマウスを移動し、右クリックすることです。表示されるダイアログボックスは、描画色を設定できますしカーソルを波形に移動することが出来ます。LTspice は、表示の大きさの解析を行い、適切な単位で表示される縦軸に対して描画します。同じ単位の表示窓にある全ての波形は、同じ軸上に描かれます。

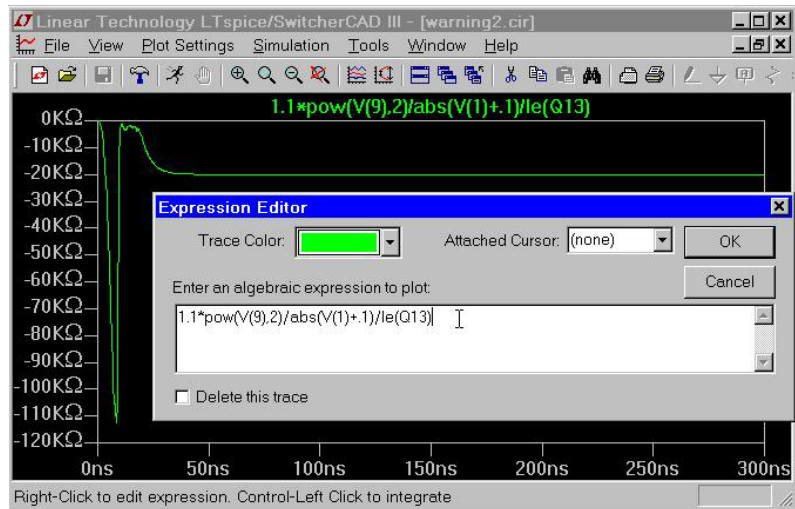


図 4.7 結果表現を描く

電圧の差；例えば $V(a) - V(b)$ ；は、 $V(a, b)$ と同じです。実データについて、次の関数が使えます。

名前	関数
$abs(x)$	絶対値
$acos(x)$	逆余弦
$acosh(x)$	逆ハイパブリック余弦
$asin(x)$	逆正接
$asinh(x)$	逆ハイパブリック正弦
$atan(x)$	逆正接
$atan2(y, x)$	y/x の逆正接
$atanh(x)$	逆ハイパブリック正接
$buf(x)$	$x > .5$ ならば 1 その他は 0
$ceil(x)$	x に等しいか大きい整数
$cos(x)$	余弦
$cosh(x)$	ハイパブリック余弦
$exp(x)$	指数
$floor(x)$	x に等しいか小さい整数
$if(x, y, z)$	$x > .5$ のとき y その他は z
$int(x)$	x を整数へ変換
$inv(x)$	$x > .5$ のとき 0 その他は 1
$limit(x, y, z)$	$nim(\max(x, y), z)$ と同じ
$ln(x)$	自然対数
$log(x)$	常用対数
$log10(x)$	底 10 の常用対数
$min(x, y)$	いずれか小さい方
$max(x, y)$	いずれか大きい方
$rand(x)$	0 と 1 の間の乱数
$random(x)$	0 と 1 の間の乱数、滑らかに変化
$re(x)$	実数部分

表 4.1: 予約関数

$sgn(x)$	符号; ジグン
$sin(x)$	正弦
$sinh(x)$	ハイパブリック正弦
$sqrt(x)$	平方根
$table(x, a, b, \dots)$	テーブルとして x に値を付ける
$tan(x)$	正接
$tanh(x)$	ハイパブリック正接
$u(x)$	単位ステップ関数
$uramp(x)$	もし $x > 0$ ならば、 x 、その他は 0
$white(x)$	$random()$ より滑らかな -0.5 と 0.5 の間の乱数

表 4.1: 予約関数

複素数のデータに関しましては、 $atan2()$ 、 $sgn()$ 、 $u()$ 、 $buf()$ 、 $inv()$ 、 $uramp()$ 、 $int()$ 、 $floor()$ 、 $ceil()$ 、 $rand()$ 、 $min()$ 、 $limit()$ 、 $if(\dots)$ 、 $table(\dots)$ は使えません。関数 $Re(x)$ および $Im(x)$ は、複素数として使えます。 $Re(x)$ は、複素数の実数部で虚数部はゼロを示し、 $Im(x)$ は、複素数の虚数部で実数部はゼロとして表示します。関数 $Ph(x)$ および $Mag(x)$ も複素数に対して有効で、位相および絶対値として複素数として示します。関数 $conj(x)$ も複素数を扱うことが出来、 x の複素共役を示します。

次の演算は、適用順位の逆優先順位でグループ化されていますが、実数データに対して使えます。

記号	演算
&	ブール代数の AND
	ブール代数の OR
^	ブール代数の XOR
<	左が右に較べ小さい場合は
>	左が右に較べ大きい場合は真
<=	左が右に較べ大きいか等しい場合は真
>=	左が右に較べ小さいか等しい場合は真
+	加算
-	減算
*	掛け算
/	割り算
**	左の数の右の数乗
!	逆
@	ステップ選択

表 4.2 演算子

真の数值は 1、偽の数值は 0 です。値が 0.5 より大きいと、ブール変換を行うと 1 となり、以下では 0 となります。

ステップ選択演算子 @ は、複数のシミュレーションが $.step$ $.temp$ あるいは $.dc$ 解析において有効であるときに役立ちます。特別なシミュレーションからのデータを選択すること

が出来ます。例えば $V(1)@3$ は、どの様なステップ描画しようとも三番目のシミュレーションからデータを描きます。

複素数データに対して、唯一 $+$, $-$, $*$, $/$, $**$, が使えます。複素数データと見なし、ブール代数の演算子 XOR ; \wedge は、指数演算子 $**$ と見なされます。

次の定数は、国際的な定義でもあります。

定数記号	値
e	2.7182818284590452354
pi	3.14159265358979323846
k	$1.3806503e-23$
q	$1.602176462e-19$

表 4.3 定数

キーワード $time$ は、過渡解析波形データを表示するときに用いられます。同様に AC 解析では、 $freq$ および $omega$ です。w は $omega$ の代わりに用いられます。

描画の平均値や RMS 値を計算

波形表示は、表示領域上の平均値や RMS 値を計算することが出来ます。最初に必要な領域の波形を拡大します。次にグラフの上にカーソルを移動し、コントロールキーを押しながら左のマウスボタンをクリックします。

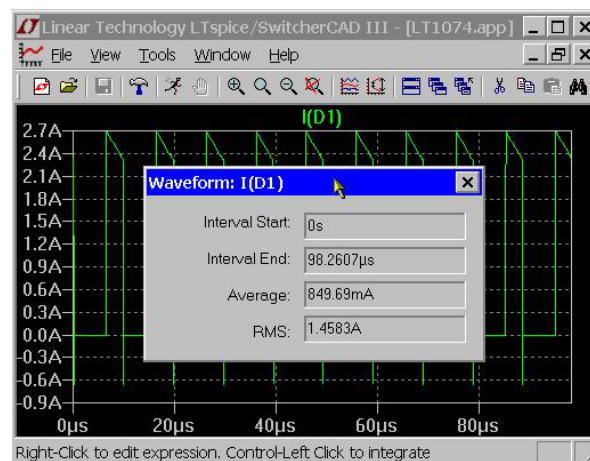


図 4.8 波形の平均値と RMS 値

描画のフーリエ変換を表示

メニュー・コマンド $View \Rightarrow FFT$ によって、様々なデータの高速フーリエ変換を求めることが出来ます。

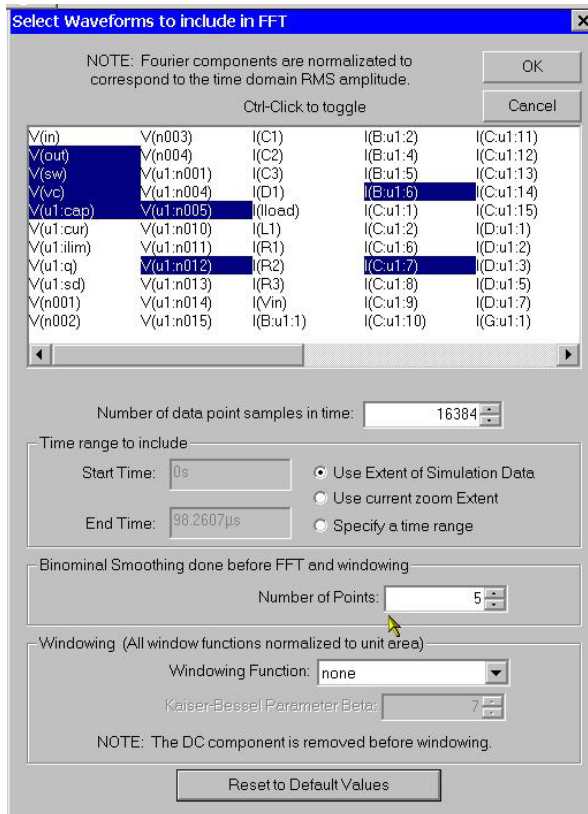


図 4.9 高速フーリエ変換

4.0.8 ユーザ定義関数

メニュー・コマンド Plot Settings => Edit Plot Defs File は、波形表示で用いる自分自身の関数やパラメータを記入することが出来ます。これらの関数は、SwCAD III が実行できる scad3.exe として、同じディレクトリにファイル名 plot.defs に保存されます。

シンタックスは、パラメータ化された回路に用いる .param および .func ステートメントと同じです。例えば次の行

```
.func Pythag(x, y) {sqrt(x*x+y*y)}
```

は、二つの数値の和の平方根として、関数 Pythag() を定義します。

同様に次の行

```
.param twopi = 2*pi
```

は、twopi を 6.28318530717959 と定義します。波形表示の中で、既に定義済である定数 pi を用いていることに注意して下さい。

4.0.9 軸の制御

データが表示されている領域を越えてマウスカーソルを移動したとき、カーソルは定規に変化します。これは、その軸の特性で表示しようとしています。左クリックするとダイヤログが開き、その軸の範囲や描画の特性を入力できるようになります。例えば実数デー

タに対して、画面の下へマウスを動かし左クリックすると、水平方向の描画範囲をダイアログへ入力することが出来ます。パラメトリック・プロットも行えます。

複素数のデータに対して位相、群遅延を描画するように選択できます。右側の縦軸に関しては、何もしません。複素数による表現を、マウスを複素数データの左縦軸へ移動することにより、ボード線図からナイキスト線図あるいは直角座標へと変換することが出来ます。

4.0.10 描画面面

複数描画面面は、一つのウインドウ上に表示されます。描画はきちんと分離され、自動的に座標が表示されます。描画は、ラベルをドラッグすることによりウインドウ間で移動できます。描画のコピーは、コントロールキーを押し下げマウス・ボタンを離すことにより実行されます。

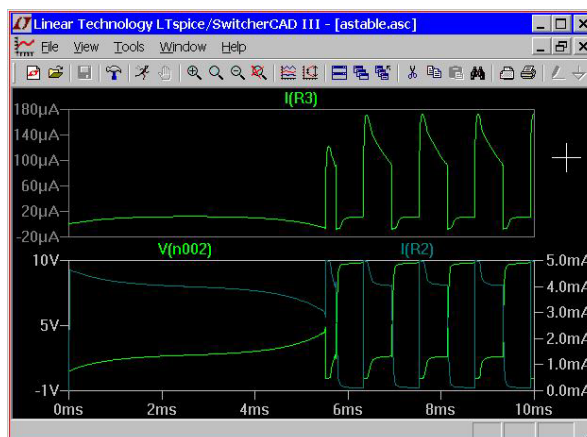


図 4.10 描画について

4.0.11 色の制御

メニュー・コマンド Tools => Color Preferences は、描画色を設定することが出来ます。サンプル描画の対象物をクリックし、赤、緑、青のスライダーを用いて希望の色を調整します。

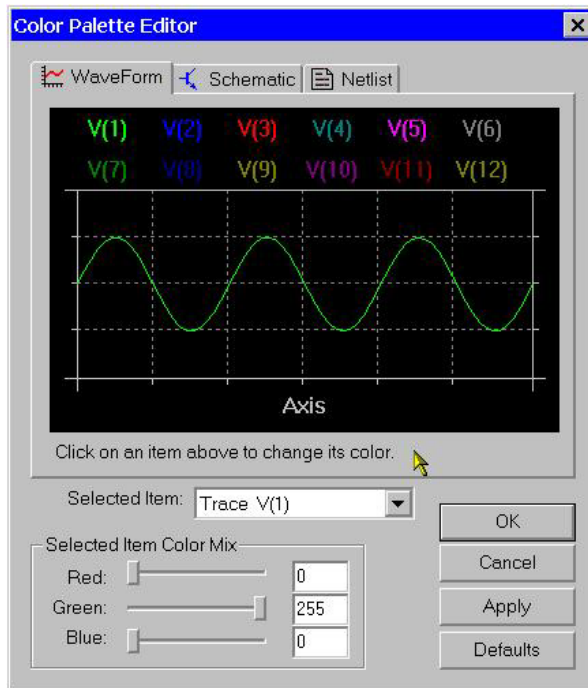


図 4.11 色の制御

4.0.12 カーソルについて

カーソルは、二つまで使えます。描画ラベル上で、左のマウスボタンをクリックし両方のカーソルを扱えるようになります。描画ラベル上で右クリックし 1st & 2nd を選択し、一つのカーソルに対し二つのカーソルを扱えるようにします。1st あるいは 2nd あるいは二つのカーソルを描画上で右クリックしカーソルのドロップダウン・ボックスを用いることによって使えるようになります。カーソルは、マウスでドラッグするかあるいはカーソルキーで動かすことができます。



図 4.12 二つのカーソル

使用するカーソルがアクティブであるとき、読み出し可能な表示が、カーソルの場所と違いが分かるように表示されます。

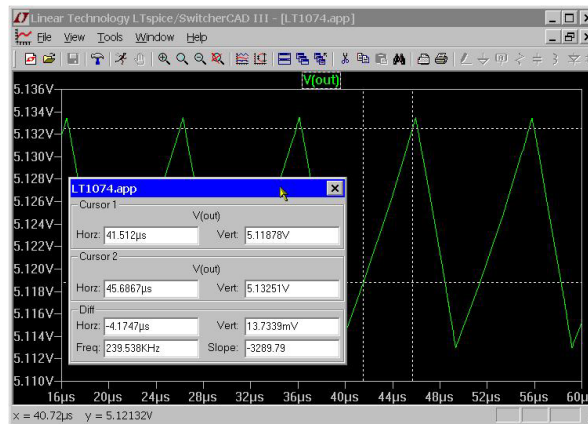


図 4.13 カーソルの表示

読み出しできるカーソルとは独立して、マウスのカーソルもあることに注意して下さい。波形ウインドウ上でマウスを動かしますと、マウスの位置がステータス・バー上に表示されます。拡大するようにマウスをドラッグしますと、ボックスの大きさがステータス・バー上に表示されます。これはマウスカーソルで、簡単に差を見る場合に便利です。もし水平軸が時間である場合には、この時間差は周波数へも変換します。

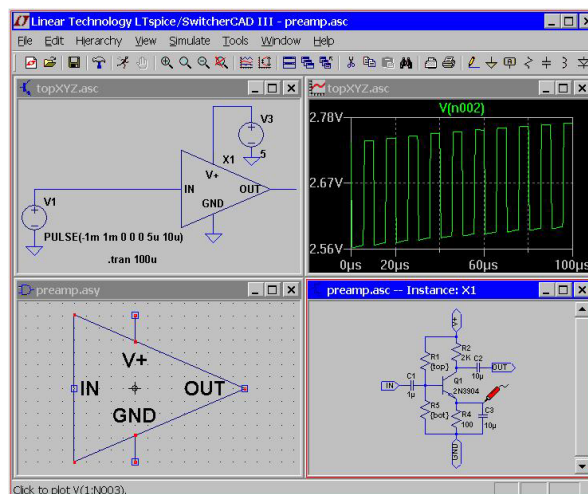


図 4.14 カーソルの扱い

左のマウスボタンを離す前に、Esc キーを押すか右のマウスボタンをクリックすることによって、拡大することなく差を測定することが出来ます。

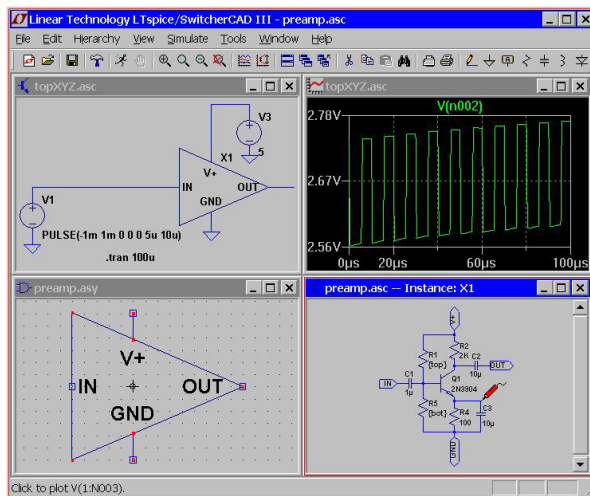


図 4.15 カーソルによるステップ情報

カーソルは、.step/.dc/.temp シミュレーションのどの計算に属しているかを見つけるために使うことができます。up/down キーボードによるカーソル・キーを用いてデータベースからデータベースへとカーソルを移動させることができます。そしてカーソル上で右クリックしステップ情報を見ることが出来ます。

4.0.13 プロット条件の保存

メニュー・コマンド Plot Settings => Save Plot Settings/Open Plot Settings ファイルは、ディスクへのプロット条件の書き込み読み出しを可能にします。プロット設定ファイルは、ASCII ファイルであり .plt の拡張子を持っています。デフォルトのファイル名は、データファイルの .raw 拡張子から .plt 拡張子を付けて求められます。もしデータファイルを最初に開いたときこのファイルが存在しますと、プロット設定ファイルは最初のプロット条件を読み込みます。

各解析タイプ ; .tran , .ac , .noise など ; は、プロット設定ファイルに独自の入力を持っています。一つのタイプから別のタイプを設定することは出来ません。しかし同じシミュレーション・タイプの別のシミュレーションからプロット設定ファイルを使うことは出来ます。

4.0.14 高速アクセスファイル・フォーマット

シミュレーションの間 LTspice は、圧縮されたバイナリ・ファイルを用います。これはファイルの残りを変更することなく、その他のシミュレーションデータを付加することが出来ます。しかし一度シミュレーションが完了しますと、このファイル・フォーマットは、ファイルに新しい一つの描画を付け加える速度を緩やかに実行させることが出来ます。

この時間を減らすために、ファイルを高速にアクセスするフォーマット、いつでも変更することが出来ます。このフォーマットは、シミュレーションが完了した後にのみ行うことが出来ます。そのあとに新しいデータは、何も付け加えられません。しかし一度この

フォーマットへ変換させられますと、新しい描画に対する負荷時間は、ファイルに保存される描画データの数に比例して極端に減少します。例えば 2000 描画データを持つ 5 [GB] のファイルがありますと、新しい描画を描くのに 4 分掛かります。しかし高速ファイルフォーマットへ変換しますと、4 分の負荷時間が 1 秒間へ減少します。このことは、巨大なシミュレーション・データであっても大きな回路の間でやりとりが出来るようになることを意味しています。高速アクセス・フォーマット・ファイルから描画を行う正確な時間は、ハードディスクの速度よりも物理メモリの大きさに依存します。

波形表示ウィンドウを高速アクセス・フォーマットへ変換するには、波形表示ウィンドウをアクティブとし、メニュー・コマンド `Files => Convert` を `Fast Access` を実効します。変換過程は、空きディスク容量が変換しようとしているファイルの大きさより大きい必要があります。しかし返還後のファイルは、元のファイルよりわずか 11 [bytes] 大きいだけです。

変換過程は、永い時間と搭載しているメモリの四分の一を使います。実際高速アクセス・フォーマットへの変換は多くの時間が必要で、最初のシミュレーションに対して必要です。変換に必要な正確な時間は、ハードディスク・フラグメントの状態と搭載している物理メモリに依存します。変換の間マウスやキーボードの応答が遅くなります。次のシンタックスを用いれば、バッチでファイル変換をすることが出来ます。

```
scad3.exe -FastAccess <file>
```

ここで <file> は、高速アクセス・フォーマットへ変換したい raw ファイルの名前です。

このフォーマットは、実数データに対してのみサポートされています。ac 解析から出てくる複素数データに対しては、サポートされていません。

第 5 章

LTspice

5.0.15 LTspice 概要

LTspice は、SwitcherCAD III の回路シミュレータ・エンジンです。LTspice は、回路図入力シミュレーション・プログラムです。LTspice シミュレータは、基本的な部分として BSIM3v3.3.4 その他の新しい MOSFET デバイスを含んだ Berkeley spice 3F4/5 に基づいています。このシミュレータは、シミュレータ機能改善、バグのフィックスそれとシミュレータの拡張が行われ、その結果業界標準の半導体およびビヘービア・モデルとして動作します。デジタル・シミュレータ機能としては、機能拡張してアナログ spice シミュレータに組み込まれ LTspice をスイッチング・レギュレータやスイッチド・キャパシタ・フィルタの様な多くの種類の回路に使える優れたボードレベルアナログおよびミックスと・モードシミュレータとなっています。

多くの Linear Technology 製品が、カスタム・マクロモデルである正確で現実的な特性を持った専門のビルディング・ブロックとしてモデル化されています。これによって電力システム基板をプロトタイプとして直ちに作成することが出来ます。

LTspice は、汎用の spice シミュレータとしても用いることが出来ます。新しい回路は、組み込まれた回路図入力で描くことが出来ます。シミュレーション・コマンドやパラメータは、確立された spice シンタックスを用いて回路図上にテキストとして置かれます。回路のノード電圧や素子電流は、シミュレーションの間あるいは後で、回路図のノード点をマウスでクリックすることによって描くことが出来ます。

LTspice は、Linear Technology Corporation の登録商標です。

5.1 入門

5.1.1 回路記述

回路は、テキストのネットリストによって定義されます。ネットリストは、回路素子、ノード、モデル定義および spice コマンドのリストから出来ています。

ネットリストは、普通グラフを用いて入力されます。新しく回路入力を始めるには、File => Open メニューから始めます。ウインドウ・ファイル・ブラウザが現れます。既にある回路を選びそれを新しい名前でも保存するか、新しい空の回路を作って新しい名前として入力します。LTspice は非常に多くのファイルや文章を扱います。拡張子.asc を持つファイルを作りたいとします。回路図入力コマンドは、Edit メニューにあります。このコ

マンドに対するキーボード・ショートカットは、Schematic Editor Overview のところにあります。

回路図をシミュレーションするとき、ネットリスト情報は、回路図と同じ名前を持つファイルへ回路図情報から抽出されます。ただし拡張子は、.net となります。LTspice は、このネットリストで読み込みます。

手書きまたは別の回路図入力によって作られたテキスト形式のネットリストを開き、シミュレーションを行い編集することが出来ます。拡張子.net、.cir あるいは.sp のファイルが、ネットリストとして LTspice によって認識されます。

このヘルプの章では、ネットリスト上のシンタックスが文章化されています。しかし時々回路図レベルのデバイスも述べています。

5.1.2 一般構造および用語

解析される回路は、ネットリストと呼ばれるテキスト・ファイルによって記述されています。ネットリストの最初の行は、無視されます。つまりコメントとなります。最後の行は普通単に.END 行で終わり、これも省略されます。.END から後の行は、無視されます。

コメントと.END の間にある行の順番は、特に関係がありません。各行は、コメント、回路素子宣言あるいは、シミュレーション方法から構成されています。例を見てみましょう。

- 最初の行は、無視されます。
- 回路が 1 [MHz] 方形波で駆動される RC 回路を示しています。

```
R1 n1 n2 1k ; ノード n1 とノード n2 の間の 1 [kΩ] 抵抗
C1 n2 0 100p ; ノード n2 とグラウンドの間に 100 [pF] の容量
V1 n1 0 PULSE(0 1 0 0 0 .5μ 1μ) ; 1 [MHz] の方形波
.tran 3μ ; 3 [μs] の過渡解析を実効
.end
```

最初の二つの行は、コメントです。* で始まる行は、コメントで、無視されます。R1 で始まる行は、ノード n1 と n2 に接続された 1 [kΩ] の抵抗です。セミコロン ; は、行の途中でコメントが始まることを意味しています。C1 で始まる行は、ノード n2 とグラウンドの間に 100 [pF] の容量が置かれることを意味しています。ノード 0 は、グローバル回路の共通グラウンドです。

次に示していますのは、LTspice の用語概説です。

1. 空白や空欄タブは、無視されます。
2. 行の最初に現れる文字は、回路素子のタイプを示しています。

- * コメント
- A 特別な関数のデバイス
- B 任意のビヘービア電源
- C 容量
- D ダイオード

- E 電圧制御電圧源
- F 電流制御電流源
- G 電圧制御電流源
- H 電流制御電圧源
- I 独立電流源
- J JFET トランジスタ
- K 相互インダクタンス
- L インダクタ
- M MOSFET トランジスタ
- O 損失のある伝送線路
- Q バイポーラ・トランジスタ
- R 抵抗
- S 電圧制御スイッチ
- T 無損失伝送線路
- U 均一の RC ライン
- V 独立電圧源
- W 電流制御スイッチ
- X サブサーキット
- Z MESFET トランジスタ

. シミュレーション方法を示します。例えば：`.option reltol=1e-4`

+ 継続行であることを示します。シミュレーションの中では、+記号は取り外されその前の継続行として扱われます。

数値は、科学表現；例えば $1e12$ ；技術表現を用いることができます。1000.0 や $1e3$ は、 $1 [k]$ と書くこともできます。次の表は、よく理解されているサフィックスです。

サフィックス	数値
<i>T</i>	$1e12$
<i>G</i>	$1e9$
<i>Meg</i>	$1e6$
<i>K</i>	$1e3$
<i>Mil</i>	$25.4e-6$
<i>M</i>	$1e-3$
<i>u(or μ)</i>	$1e-6$
<i>n</i>	$1e-9$
<i>p</i>	$1e-12$
<i>f</i>	$1e-15$

表 5.1 サフィックスと数値

数値や技術表現の後に続く認識できない文字は、無視されます。つまり 10、10V、10Volts および 10Hz は、全て同じ数値です。また M、MA、MSec、MMhos は全て同じスケール $1e-3$ を示しています。共通のエラーとしてよく行われることは、1 MegaOhm として *1M* を用いることです。しかし *1M* は、1 milliOhm とみなされます。不幸なこと

ではありますが、標準の spice では、大切なことです。

LTspice は、6K34 と書かれた数値を 6.34K と解釈します。この表現は、上記の表現において用いることが可能です。これは、Tools => Control Panel => spice で、Accept 3K4 as 3.4k のチェックをはずすことによって解除することが出来ます。

ノードの文字は、任意の文字列を使うことが出来ます。グローバル回路の共通ノード（グラウンド）は、0 ですが特別な GND 名称も使えます。ノードは、文字列ですので、0 と 00 は別のノードです。

マニュアルの次の章から、鍵括弧がデータ・フィールドに現れます。これには、特別な情報が記載されています；例えば <srcname> は、ある特別な電源の名前です。四角括弧は、囲まれたデータ・フィールドがオプションであることを示しています。

5.1.3 回路素子のクイック・レファレンス

素子	シンタックス
特別な関数	Axx n1 n2 n3 n4 n5 n6 n7 n8 <model> [extra parameters]
任意のビヘービア電源	Bxx n+ n- <V=... or I=...>
容量	Cxx n+ n- <capacitance> [ic=<val.>] [Rser=<val.>] + [Lser=<val.>] [Rpar=<val.>] [Cpar=<val.>] [m=<val.>]
ダイオード	Dxx A K <model> [area]
電圧制御電圧源	Exx n+ n- nc+ nc- <gain>
電流制御電流源	Fxx n+ n- <Vnam> <gain>
電圧制御電流源	Gxx n+ n- nc+ nc- <transcond.>
電流制御電圧源	Hxx n+ n- <Vnam> <transres.>
独立電流源	Ixx n+ n- <current>
JFET trans istory	Jxx D G S <model> [area] [off] [IC=<Vds,Vgs>] [area] [off]
相互インダクタンス	Kxx L1 L2 <coeff.>
インダクタンス	Lxx n+ n- <inductance> [ic=<val.>] [Rser=<val.>] + [Rpar=<val.>] [Cpar=<val.>] [m=<val.>]
MOSFET Transistor	Mxx D G S B ,model> [L=<len>] [W=<width>] [AD=<area>] + [AS=<area>] [PD=<perim>] [PS=<perim>] [NRD=<value>] + [NRS=<value>] [off] [IC=<Vds, Vgs, Vbs>] [temp=<T>]
損失のある伝送線路	Oxx L+ L- R+ R- ,model>
バイポーラ・トランジスタ	Qxx C B E [S] <model> [area] [off] [IC=Vbe,Vce] [tem=<T>]
抵抗	Rxx n1 n2 <value>
電圧制御スイッチ	Sxx n1 n2 nc+ nc- <model> [on,off]
損失のない伝送線路	Txx L+ L- R+ R- ZO=<value> TD=<value>
均一 RC 線路	Uxx n1 n2 ncommon <model>
独立電圧源	Vxx n+ n- <voltage>
電流制御スイッチ	Wxx n1 n2 <Vnam> <model> [on,off]
サブサーキット	Xxx n1 n2 n3 ... <subckt name>
MESFET transistor	Zxx D G S model [area] [off] [IC=<Vda,Vgs>]

表 5.2: クイック・レファレンス

5.2 ドット・コマンド

シミュレーションを走らせるには、回路が定義されないといけないばかりか実行する解析のタイプも指定しなければいけません。解析には、六つの異なった解析のタイプがあります：線形小信号交流解析、直流スイープ、雑音、直流動作点解析、小信号直流伝達関数および過渡解析。これらの六つの解析の一つを、明確な形で指定しなければいけません。

一方回路トポロジーは、常に回路図として描かれます。コマンドは、テキストとして回路図上に置かれます。この様な全てのコマンドは、ピリオッドで始まります。そのためドット・コマンドと呼ばれます。

5.2.1 .AC

直流動作点の周りで、小信号交流線形解析を実行。

LTspice の小信号（線形）交流解析は、周波数の関数として交流ノード電圧を計算します。最初に回路の直流動作点電圧が計算されます。次に回路の非線形デバイスの線形化された小信号モデルが、動作点に対して求められます。最後に駆動する信号として、独立電圧源や独立電流源を用いて、周波数のある範囲についての線形回路が計算されます。

この解析モードは、フィルタ、ネットワーク、安定性解析や雑音解析に対して有効です。

Syntax : `.ac <oct, dec, lin> <Nsteps> <StartFreq> <EndFreq>`

周波数は、StartFreq と EndFreq の間をスイープします。ステップ数は、次に示します表に従いキーワード oct, dec, lin, および Nsteps で定義されます。

キーワード	Nstep
oct	オクターブ（二倍）当たりのステップ数
dec	ディケード（10倍）当たりのステップ数
lin	StartFreq と EndFrq の間の等間隔ステップ数

表 5.3 .AC の Nsteps

5.2.2 .BACKANNO

サブサーキットのピン名称を端子電流へ接続する。

Syntax : `.backanno`

このコマンドは、回路図から自動的に SwitcherCAD III によって実行されます。LTspice は、.raw ファイルの情報を取り込み、ピン名称によって端子電流に反映させます。シンボルのピンをマウスによってクリックすることにより、ピンに流れ込む電流を表示させることができます。

5.2.3 .DC

直流電源スイープ解析。

これは電源の直流値をスイープし、直流解析を実行します。増幅器の直流伝達関数を計算したり、トランジスタ・モデルの特性を確認するための曲線を描かせたりするのに便利です。

Syntax: .dc <srcnam> <Vstart> <Vstop> <Vincr>
 [<srcnam2> <Vstart2> <Vstop2> <Vincr2>]

<Vstart> は、<Vstart> から <Vstop> まで、<Vincr> ステップでスイープする独立の電圧源あるいは電流源です。

* Example .dc sweep

```
*
M1 2 1 0 0 nbsim
Vgs 1 0 3.5
Vds 2 0 3.5
.dc Vds 3.5 0 -0.05 Vgs 0 3.5 0.5
.model nbsim NMOS Level=8
.save I(Vds)
.end
```

5.2.4 .END

ネットリストの終わり。

このコマンドは、ネットリストが終わることを示しています。このコマンドの後のラインは、全て無視されます。回路図のテキストとして、このコマンドを置いてはなりません。ネットリスト抽出が、終わりと判断してしまうためです。

5.2.5 .ENDS

サブサーキット定義の終了。

このコマンドは、サブサーキット定義の終了を示しています。詳しくは、.SUBCKT を見て下さい。

5.2.6 .FOUR

.TRAN 解析の後フーリエ成分を計算します。

Syntax: .four <frequency> [Nharmonics] [Nperiods] <data trace1>
 [<data trace2> . . .]

Example: .four 1kHz V(out)

このコマンドは、過渡解析の後で実行されます。一般に市販されている SPICE シミュレータと同じように与えられます。このコマンドの出力は、.log ファイルに記述されます。出力結果を見るために、メニュー・アイテム View => Spice Error Log を用いて下さい。ほとんどの目的に対しては、waveform viewer にあります FFT が役に立ちます。

もし整数値の Nharmonics が書かれていると、解析はその数だけ実行されます。特に指定しなければ、高調波のデフォルト数は 9 です。

フーリエ解析は、整数 Nperiod が Nharmonics の後にない場合、最終時間 Tend から Tend 前の一周周期分解析されます。Nperiod が -1 となっていると、フーリエ解析は全てのデータ範囲で解析されます。

5.2.7 .FUNC

ユーザー定義関数。

Syntax: `.func <name> ([args]) {<expression>}`

Example: `.func Pythag(x,y) {sqrt(x*x+y*y)}`

コマンド.func は、ユーザーによってパラメータ化される回路とビヘービア電源のユーザー定義関数を作成することが出来ます。このコマンドは、明確なパラメータ化されたサブサーキットの関数を使うのに役立ちます。その結果抽象的な名前の回路をライブラリとして登録することが出来ます。

.func 記述は、サブサーキット定義の中に記述することが出来ます。このコマンドは、そのサブサーキットおよびサブサーキットによって呼び出されるサブサーキットに対して関数という制限を設けることが出来ます。

次の例は、.func および.param 記述です。

```
* Example deck using a .func statement
.func myfunc(x,y) {sqrt(x*x+y*y)}
.param u=100 v=600
V1 a 0 pulse(0 1 0 1n 1n .5^ef^bd^b5 1^ef^bd^b5)
R1 a b {myfunc(u,v/3)}
C1 b 0 100p
.tran 3^ef^bd^b5
.end
```

全てのパラメータ記述内容は、シミュレーションが始まる前に実行されます。

5.2.8 .GLOBAL

グローバル・ノードの宣言。

Syntax: `.global <node1> [node2] [node3] [. . .]`

Example: `.global VDD VCC`

.global コマンドは、サブサーキットで記述されているあるノードがローカルなサブサーキット・ノートと言うことばかりでなく絶対的なグローバルなノードを宣言するために用いられます。

グローバルな共通ノードは、0 でありグローバル宣言は必要ありません。また \$G_ の名前のノードは、.global statement で記述しなくてもグローバルです。

5.2.9 .IC

初期条件の設定。

.ic コマンドは、過渡解析の特別な初期条件を与えます。ノード電圧やインダクタ電流の初期化が出来ます。直流解は、制限事項として実行されます。インダクタは通常、spice プログラムの直流解析で短絡回路として扱われます。もし初期条件が与えられますと LTspice において無限のインピーダンスを持つ電流源として扱われます。

Syntax : .ic [V(<n1>)=<voltage>] [I(<inductor>)=<current>]

Example: .ic V(in)=2 V(out)=5 V(vc)=1.8 I(L1)=300m

5.2.10 .INCLUDE

別のファイルを挿入する。

Syntax : .include <filename>

このコマンドは、そのファイルが include コマンドの代わりにネットリストに記入しているかのように名前づけられたファイルを含ませます。このコマンドは、モデルやサブサーキットのライブラリを含ませるために役立ちます。

絶対パスをファイル名の中に含ませることが出来ます。その他 LTspice は、最初にディレクトリ <SwCADIII>/lib/sub を見に行き、次に呼ばれたネットリストを含んだディレクトリを見に行きます。ここで <SwCADIII> は、実行可能な scad3.exe を含んだディレクトリです。普通は、C:/Program Files/LTC/SwCADIII にインストールされています。

ファイル名の拡張子は、特に指定していません。ファイルが myfile.lib としていれば、inc myfile ではなく .inc myfile.lib を用いる必要があります。

ファイル名として、次の形式の url を用いることも可能です。

```
.inc http://www.company.com/models/library.lib
```

ファイル library.lib は、回路ディレクトリへ http 転送され保存されます。引き続いたシミュレーションに対して、シミュレーションするたびのダウンロードを避けるには、inc 記述を次のように編集することが出来ます。

```
.inc library.lib
```

もし url が存在しなければ、ほとんどの web サーバーは、エラーを返してきません。しかしブラウザ上にエラーを示す html ウェブ・ページを返してきます。LTspice は、必ずこれらのメッセージをエラーだと判断してくれるわけではありません。その結果シミュ

レーションは、有効な spice シンタックスとしてシミュレーションに含まれる html 言語エラーを含んだままシミュレーションを実行しようとして、奇妙なエラー・メッセージを表示します。

もし転送される url が.pdf ファイルであるならば、シミュレーションは、ダウンロードした後に停止してしまいます。例えば次の命令文は、このマニュアルを.pdf ファイルとしてダウンロードすることになります。

```
* Dummy simulation to download the help file.
* The simulation will abort with an error, but
* you'll be left with the file scad3.pdf in the
* same directory containing the netlist.
.inc http://ltspice.linear.com/software/scad3.pdf
.end
```

5.2.11 .LIB

ライブラリを含ませる。

Syntax : `.lib <filename>`

このコマンドは、ファイルが.lib コマンドの代わりにネットリストへ記載するように名前つきファイルのモデルやサブサーキット定義を含めます。グローバルな回路素子は、無視されます。

絶対パス名が、ファイル名に入れられます。その他 LTspice は、最初にディレクトリ <SwCAD III>/lib/cmp を見ます。それから <SwCAD III>/lib/sub、次に呼ばれたネットリストを含むディレクトリを見ます。<SwCAD III> は、実行可能な scad3.exe を含んでいます。一般には、C:/Program Files/LTC/SwCAD III としてインストールされています。

ファイル名の指定はありません。ファイルを myfile.lib とするなら、lib myfile とすべきでなく.lib myfile.lib としなければなりません。ファイル名として、次の形式の url を用いることが出来ます。

```
.lib http://www.company.com/models/library.mod
```

ファイル library.mod は、回路ディレクトリに転送され、ライブラリとして含まれます。続くシミュレーションに対して、シミュレーションを走らせるたびにファイルをダウンロードすることがないように、.lib 記述を次のように編集することが出来ます。

```
.lib library.mod
```

もし url が存在しなければ、ほとんどの web サーバーは、エラーを返してきません。しかしブラウザ上にエラーを示す html ウェブ・ページを返してきます。LTspice は、必ずこれらのメッセージをエラーだと判断してくれるわけではありません。その結果シミュレーションは、有効な spice シンタックスとしてシミュレーションに含まれる html 言語エラーを含んだままシミュレーションを実行しようとして、奇妙なエラー・メッセージを表示します。

もし転送られる url が.pdf ファイルであるならば、シミュレーションは、ダウンロードした後に停止してしまいます。例えば次の命令文は、このマニュアルを.pdf ファイルとしてダウンロードすることになります。

```
* Dummy simulation to download the help file.
* The simulation will abort with an error, but
* you'll be left with the file scad3.pdf in the
* same directory containing the netlist.
.inc http://ltspice.linear.com/software/scad3.pdf
.end
```

5.2.12 .LOADBIAS

以前に行った直流解析を呼び出す。

Syntax: `.loadbias <filename>`

`.loadbias` コマンドは、`.savebias` コマンドと対をなすコマンドです。最初にシミュレーションを走らせ、`.savebias` コマンドを実行します。次に、差う`えびあ s コマンドから`.loadbias` コマンドに変更します。

5.2.13 .MEASURE

ユーザー定義の電気的特性を展開。

二つの異なったタイプの`.MEASURE` 記述があります。軸に沿った点に関するもの（独立変数、すなわち`.tran` 解析の時間軸）それと軸の上の範囲に関するものです。軸上の点を示す最初のバージョンは、特別な点あるいはある条件の点でのデータ値あるいは表現を印刷するために用いられます。次のシンタックスが用いられます。

```
Syntax:      .MEAS [SURE] [AC|DC|OP|TRAN|TF|NOISE] <name>
             + [<FIND|DERIV|PARAM> <expr>]
             + [WHEN <expr> | AT=<expr>]]
             + [TD=<val1>] [<RISE|FALL|CROSS>]=[<count1>|LAST]]
```

`.MEAS` 記述が適用される解析タイプを、オプション的に述べる事が出来ます。ある解析タイプのみ`.MEAS` 記述を用いる事が出来ます。その名前は、結果に対して他の`.MEAS` 記述で用いる事が出来るパラメータ名を与えることを求められます。下記は、軸に沿った一つの点に対する`.MEAS` 記述の例です。

```
.MEAS TRAN res1 FIND V(out) AT=5m
```

Print the value of V(out) at t=5ms labeled as res1.

```
.MEAS TRAN res2 FIND V(out)*I(Vout) WHEN V(x)=3*V(y)
```


Print the value of the expression $V(\text{out}) * I(\text{Vout})$ the first time the condition $V(x) = 3 * V(y)$ is met. This will be labeled res2.

```
.MEAS TRAN res3 FIND V(out) WHEN V(x)=3*V(y) cross=3
```

Print the value of $V(\text{out})$ the third time the condition $V(x) = 3 * V(y)$ is met. This will be labeled res3.

```
.MEAS TRAN res4 FIND V(out) WHEN V(x)=3*V(y) rise=last
```

Print the value of $V(\text{out})$ the last time the condition $V(x) = 3 * V(y)$ is met when approached as $V(x)$ increasing wrt $3 * V(y)$. This will be labeled res4.

```
.MEAS TRAN res5 FIND V(out) WHEN V(x)=3*V(y) cross=3 TD=1m
```

Print the value of $V(\text{out})$ the third time the condition

$V(x) = 3 * V(y)$ is met, but don't start counting until the time as elapsed to 1ms. This will be labeled res5.

```
.MEAS TRAN res6 PARAM 3*res1/res2
```

Print the value of $3 * \text{res1} / \text{res2}$. This form is useful for printing expressions of other .meas statement results. It's not intended that expressions based on direct simulation data, such as $V(3)$, are present in the expression to be evaluated, but if they are, the data is taken from the last simulated point. The result will be labeled res6.

上の例で、軸に沿った点に関して、要求される結果は、軸上のデータ（従属変数）に基づいています。何も軸上の情報が要求されなければ、.MEAS 記述は、測定条件が発生する軸上の点を印刷します。

```
.MEAS TRAN res6 WHEN V(x)=3*V(y)
```

Print the first time the condition $V(x) = 3 * V(y)$ is met. This will be labeled res6.

.MEAS 記述の他のタイプは、軸上の範囲に関することです。次のシンタックスが用いられます。

```
Syntax: .MEAS [AC|DC|OP|TRAN|TF|NOISE] <name>
        +      [<AVG|MAX|MIN|PP|RMS|INTEG> <expr>]
```

```

+ [TRIG <lhs1> [[VAL]=]<rhs1>] [TD=<val1>]
+           [<RISE|FALL|CROSS>=<count1>]
+ [TRIG <lhs2> [[VAL]=]<rhs2>] [TD=<val2>]
+           [<RISE|FALL|CROSS>=<count2>]

```

軸の範囲は、TRIG と TARG によって決められます。TRIG 点は、省略されるとシミュレーションの開始をデフォルトとします。同様に TARG 点は、シミュレーション・データの終わりのデフォルトです。TRIG、TARG および WHEN が省略されますと、.MEAS 記述は全てのデータ点に渡って働きます。ある範囲に渡って実行される測定の種類は、

キーワード	ある範囲での演算
AVG	
MAX	<expr> の範囲を計算
MIN	<expr> の最大値を見つける
PP	<expr> の最小値を見つける
RMS	<expr> の peak-to-peak を見つける
INTEG	<expr> を積分

測定が実施されなければ、.MEAS 記述の結果は、TRIG 点と TARG 点間の軸上距離となります。次は、間隔を表す .MEAS 記述の例です。

```

.MEAS TRAN res7 AVG V(NS01)
+ TRIG V(NS05) VAL=1.5 TD=1.1u FALL=1
+ TARG V(NS03) VAL=1.5 TD=1.1u FALL=1

```

Print the value of average value of V(NS01) from the 1st fall of V(NS05) to 1.5V after 1.1us and the 1st fall of V(NS03) to 1.5V after 1.1us. This will be labeled res7.

.AC 解析に対して、複素データの条件付き表現は、表現を変換することにより実数条件をその大きさへ変換されます。この例として

```
.MEAS AC rel8 when V(out)=1/sqrt(2)
```

rel8 の結果は、V(out) の大きさが 0.7071067811865475 に等しくなる周波数です。

また .MEAS 記述の結果は、もう一方の .MEAS 記述でも用いることができます。この例では、3 [dB] 帯域幅が計算されます。

```
.MEAS AC tmp max mag(V(out)); find the peak response
```

```
;
```

```
and call it "tmp"
```

```
.MEAS AC BW trig mag(V(out))=tmp/sqrt(2) rise=1
```

```
+ targ mag(V(out))=tmp/sqrt(2) fall=last
```

Print the difference in frequency between the two points 3dB

down from peak response. NOTE: The data from a .AC analysis is complex and so are the .measurement statements results. However, the equality refers only to the real part of the complex number, that is, "mag(V(out))=tmp/sqrt(2)" is equivalent to $\text{Re}(\text{mag}(V(\text{out})))=\text{Re}(\text{tmp}/\text{sqrt}(2))$.

AVG、RMS、INTEG 演算は、.NOISE 解析とは違います。と言うのも雑音は、周波数全体に渡って二乗平均だからです。このことから AVG と RMS は、平均雑音を与えますし、INTEG は積分した雑音を与えます。もし.noise 解析に

```
.MEAS NOISE out_totn INTEG V(onoise)
.MEAS NOISE in_totn INTEG V(inoise)
```

SPICE 演算付け加えると、全入力および出力の rms 雑音が.log ファイルに出力されます。

.MEAS 記述は、シミュレーションが完了した後ポスト・プロセッサで実行されます。.MEAS 記述を記入しデータセットを実行することが出来ます。このため波形表示ウィンドウをアクティブとし、メニュー・コマンド File => Execute .MEAS Script を実行します。.MEAS 記述のもう一方の手順がシミュレーション完了後ポスト・プロセッサにおいて実行されます。.MEAS 記述の精度は、圧縮後の波形データの精度によって決められます。.MEAS 記述出力の精度について圧縮設定を調整することが出来ます。

品質を重要視する when <cond1> = <cond2> のような条件を調べる時、用いてはなりません。これは、浮動小数点を用いた等式は、保存された数字に用いられている有限の精度に基づいているからです。

5.2.14 .MODEL

SPICE モデルの定義

ダイオード、トランジスタ、スイッチ、損失のある伝送線路あるいは均一の RC 線路のモデルを定義します。

ある回路素子例えばトランジスタは、多くのパラメータを持っています。トランジスタの多くの特性を決める全てのパラメータを定義する代わりに、トランジスタはモデル名でグループ化され、共通のパラメータを持っています。同じモデルのトランジスタは、異なったサイズを持つこともあります。そして電氣的な振る舞いは、特性のサイズを変更できます。

Syntax: `.model <modname> <type>[(<parameter list>)]`

モデルの名前は、一つでなければいけません。回路素子の二つの異なったタイプ、例えばダイオードとトランジスタは、同じ名前を用いることは出来ません。パラメータのリストは、モデルのタイプに依存します。下記表はモデルのタイプのリストです。

タイプ	対応する回路素子
SW	電圧制御スイッチ
CSW	電流制御スイッチ
URC	均一分布 RC モデル
LTRA	損失のある伝送線路モデル
D	ダイオード・モデル
NPN	NPN BJT モデル
PNP	PNP BJT モデル
NJF	N チャネル JFET モデル
PJF	P チャネル JFET モデル
NMOS	N チャネル MOSFET モデル
PMOS	P チャネル MOSFET モデル
NMF	N チャネル MESFET モデル
PMF	P チャネル MESFET モデル
VDMOS	縦型二重拡散電力 MOSFET モデル

どのパラメータが特性値であるか、モデルに共通であるかのリストについて、素子の記述を見て下さい。

5.2.15 .NET

.AC 解析で、ネットワーク・パラメータを計算

この記述は、小信号 (.AC) 解析で、二端子網の入出力アドミタンス・インピーダンス、Yパラメータ、Zパラメータ、HパラメータおよびSパラメータを計算するために用いられます。一端子網の入力アドミタンスやインピーダンスも計算することが出来ます。この記述は、.AC 記述と一緒に用いなければいけません。.AC 解析は、ネットワーク解析の周波数範囲を決めます。

Syntax: `.net [V(out[,ref])|I(Rout)] <Vin|Iin>`
`+ [Rin=<val>] [Rout=<val>]`

ネットワーク入力、独立電圧源 $\langle Vin \rangle$ あるいは独立電流源 $\langle Iin \rangle$ によって決まります。オプションの出力端子は、ノード $\langle V(out) \rangle$ あるいは抵抗 $\langle I(out) \rangle$ によって決まります。端子は、抵抗 Rin および $Rout$ で終端されます。特に定めなければ、端子のデフォルト・インピーダンスは、 $Rser$ によって定められる電圧源あるいは抵抗のある出力端子の場合を除いて $1 [\Omega]$ となります。これら二つの場合終端のデフォルト・インピーダンスデバイスのインピーダンスです。.NET 記述で決められた端子の値は、.NET によって計算されたデバイス・インピーダンスを上書きします。しかし通常の.AC ノード電圧や電流には影響を与えません。すなわち.NET 記述は、.通常の電圧や電流のネットワーク上で端子インピーダンスに影響せず、AC 解析の部分として計算されます。

C:/Program Files/LTC/SwCADIII/examples/Educations/S-param のファイルを見て下さい。 $Rser$ を望ましい電源インピーダンスとして、電圧源 $V4$ を用いること、.NET 記述として単に `.net I(Rout) V4` と出力端子を設定することを進めています。 Rin あるいは $Rout$ が.NET 記述になれば、入出力デバイスの値が用いられます。これらの設定は、.AC 解析のノード電圧や電流を.NET 記述と同じ方法でネットワークを終端します。

5.2.16 .NODESET

最初の直流解析に対するヒントを与えます。

`.nodeset` コマンドは、直流動作点解析のためのヒントを与えます。もし回路が例えばフリップフロップのように複数の動作点状態を持つならば、直流動作点状態を求める解析は決して収束しません。`.nodeset` コマンドは、一つあるいはもう一つのの状態へと回路を導くために用いることが出来ます。基本的には、`.nodeset` コマンドで与えられた電圧に解が到達した後、この制限は繰り返し計算を停止します。

Syntax: `.NODESET V(node1)=<voltage> [V(node2)=<voltage> [...]]`

5.2.17 .NOISE

雑音解析を実行

この解析は、周波数領域で行われます。ジョンソン、ショットおよびフリッカ雑音を計算します。出力データは、単位帯域幅当たりの二乗信号電圧の平方根です。

Syntax: `.noise V(<out>[,<ref>]) <src> <oct, dec, lin>
+ <Nsteps> <StartFreq> <EndFreq>`

`V(<out>[,<ref>])` は、全出力雑音が計算されるノードです。これは二つのノード間電圧を表現する $V(n1, n2)$ として計算されます。`<src>` は、入力雑音が参照する独立電源の名前です。`<src>` は、雑音のない入力信号です。パラメータ `<oct, dec, lin>`、`<Nsteps>`、`<StartFreq>`、および `<EndFreq>` は、必要な周波数範囲を決め、`.AC` コマンドを用いて解析されます。

出力データ軌跡 `V(onoise)` は、上記シンタックスで出力として決められたノードに関する雑音スペクトル電圧密度です。もし入力信号が電圧源として与えられていますと、データ軌跡 `V(inoise)` は、入力に対する雑音電圧密度です。入力が電流源として与えられていますと、データ軌跡 `inoise` は、入力電流源に関係しています。各素子の電圧への寄与が描かれます。これらの寄与は、出力に関係しています。これらをデータ軌跡 `gain` で割ることにより、入力へ換算することが出来ます。

波形表示は、`<Ctrl-Key>` と対応するデータ軌跡ラベル上を左マウスボタンをクリックすることにより帯域幅当たりの雑音を積分することが出来ます。

5.2.18 .OP

直流動作点を見いだす。

容量は開放状態、インダクタンスは短絡状態で直流解析を実行します。通常直流解析は、回路の動作点を見つけるため解析の一部として実行されます。もし動作点だけを求めたいのであれば、`.OP` を用いて下さい。`.OP` シミュレーションの後、ノードあるいはブランチ上でクリックすると、`.OP` 解がステータス・バー上に表示されます。

5.2.19 .OPTIONS

シミュレーション・オプションの設定

キーワード	データ	デフォルト	説明
abstol	num.	1 pA	絶対電流誤差余裕
baudrate	num.	(none)	アイ・ダイアグラムで用いる。 波形描画パッケージに、どの様にしてビットを重ねるために時間軸をそろえるかを告げる
chgtol	num.	10 fC	絶対電荷余裕
defad	num.	0.	デフォルト MOS ドレイン拡散面積
defas	num.	0.	デフォルト MOS ソース拡散面積
defl	num.	100 μm	デフォルト MOS チャンネル長
defw	num.	100 μm	デフォルト MOS チャンネル幅
delay	num.	(none)	アイ・ダイアグラムで用いられる。 ダイアグラムでビット変化をさせる。
flagloads	flag	false	負荷として外部電流源を示す。
frain	num.	(none)	周波数解析の入力端子
fraout	num.	(none)	周波数解析の出力端子
frastop	num.	(none)	周波数解析の高周波端
frastart	num.	(none)	周波数解析の低周波端
fraamp	num.	(none)	周波数解析の周波数関数振幅
fravref	num.	(none)	周波数解析の付加的な直流オフセット
Gmin	num.	1e-12	収束性改善のために全ての PN 接合 に加えられるコンダクタンス
gminsteps	num.	25	初期直流解の gminstepping を使う ため non-zero を設定
itl1	num.	100.	直流繰り返し回数・リミット
itl2	num.	50.	直流伝達曲線繰り返し回数・リミット
itl4	num.	10.	過渡解析時間点繰り返し回数・リミット
itl6	num.	25.	電源ステップで用いられるステップ数 (直流動作点を求めにくいときに用いられる)
srcsteps	num.	25.	itl6 の変数名
maxclocks	num.	infin.	保存されるクロックサイクルの最大数
maxsteps	num.	infin.	過渡解析の最大ステップ・サイズ
measdgt	num.	6	.measure 記述出力で用いられる重要な図の数
method	string	trap	数値積分法、台形あるいは道具一式
minclocks	num.	10	保存されるクロック・サイクル数
nomarch	flag	false	マーキング・ウェブフォームを描画しない
noopiter	flag	false	直接 gmin ステップへ行く
numdgt	num.	6	歴史的に numdgt は、出力データに用いられる重要な図の数を設定するために用いられました。LTspice では、numdgt が 6 以上に設定されると従属変数データで倍精度が用いられます。
oversample	num.	(none)	周波数応答解析用
pivrel	num.	1e-3	もっとも大きな入力と需要可能な値との相対比
pivtol	num.	1e-13	ピボットとして受け入れられる行列入力の最小値

reltol	num.	0.001	相対誤差余裕
startclocks	num.	5	定常状態を見いだす待機クロック・サイクル数
sstol	num.	0.001	定常状態検知のための rel. 誤差
temp	num.	27 degC	特に指定しないときの素子デフォルト温度
tnom	num.	27 degC	デバイス・パラメータが温度を特別に指定しないときモデルに対して測定されるデフォルト温度
topologycheck	num.	1	フローティング・ノードおよび電圧源のループのチェックをスキップするためにゼロを設定します。
trtol	num.	1.0	過渡解析誤差余裕の設定。 このパラメータは、実際の過剰見積もりされる打ち切り要素を見積もる。
trytocompact	flag	not set	シミュレータは、フラグが立つとき LTRA 伝送線路の入力電圧、電流を圧縮しようとします。
vntol	num.	1 μV	絶対電圧誤差余裕を設定
plotreltol	num.	.0025	波形圧縮の相対誤差余裕を設定
plotvntol	num.	10 μV	波形圧縮の絶対誤差余裕を設定
plotsbstol	num.	1 nA	波形圧縮の絶対電流誤差余裕を設定
plotwinsize	num.	300	一つのウィンドウで圧縮するデータ点の数。 圧縮を設定

5.2.20 .PARAM

ユーザー定義パラメータ

.param は、ユーザー定義変数の生成が出来ます。これを用いることによって、分かりやすくする名前を持たせたり、サブサーキットをパラメータ化することが出来ます。その結果抽象化された回路をライブラリとして保存することが出来ます。

.param 記述は、サブサーキット定義に含ませることが出来ます。サブサーキットおよびそのサブサーキットによって呼び出されるサブサーキットのパラメータに制限を加えます。

パラメータの記述および展開したものを呼び出すには、丸括弧の表現を閉じます。閉じられた表現は、フローティング点の値で置き換えられます。

次に.param 記述およびサブサーキット記述上に直接パラメータを転送する例を示します。

```
*
* This is the circuit definition
.param x=y y=z z=1k*tan(pi/4+.1)
X1 a b 0 divider top=x bot=z
V1 a 0 pulse(0 1 0 .5efbdb5 .5efbdb5 0 1efbdb5)
```

```
* this is the definition of the subcircuit
```

```
.subckt divider n1 n2 n3
r1 n1 n2 {top}
r2 n2 n3 {bot}
.ends
*

.tran 3efbdb5
.end
```

パラメータ代入の図は、シンボリック宣言言語です。そのパラメータは、展開された値としてサブサーキットへ送られるのではなく、それ自身の表現や関連性によって送られます。丸括弧に出会ったとき、閉じられた表現は、全ての関係に基づいて展開されます。そしてフローティング点の値に集約されます。

次の関数や演算が使えます。

関数名	記述
abs(x)	x の絶対値
asoc(x)	arccos の実部 例 asoc(-5) は、3.14159 であり 3.14159+2.29243i ではない。
arccos(x)	acos() と同じ
acosh(x)	双曲線関数 cos の実部 例 acosh(.5) は、ゼロであり 1.0472i ではない。
asin(x)	sin の逆関数 例 asin(-5) は、-1.57080 であり -1.57080+2.29243i ではない。
arcsin(x)	asin() と同じ
asinh(x)	双曲線 sin 逆関数
atan(x)	tan() の逆関数
atan2(y, x)	y/x の四分の一
atanh(x)	双曲線関数の逆関数
buf(x)	x _{0.5} のとき 1、その他はゼロ
cbrt(x)	x の自乗平均
ceil(x)	x と等しいか大きな整数
cos(x)	余弦関数
cosh(x)	cos の双曲線関数
exp(x)	指数関数
fabs(x)	abs(x) と同じ
flat(x)	均一分布をしたランダムな数
floor(x)	x と同じか小さい整数
gauss(x)	ガウス関数をしたランダムな数
hypot(x,y)	sqrt(x**2+y**2)
if(x,y,z)	x _{0.5} のとき y その他は z
int(x)	x を整数に変換
inv(x)	x _{0.5} のときゼロその他は 1
limit(x,y,z)	x,y,z の中間値
ln(x)	自然対数

log(x)	常用対数
log 10(x)	底 10 の常用対数
max(x,y)	x,y の大きい方
mc	均一分布をした $x*(1+y)$ と $x*(1-y)$ のランダムな数
min(x,y)	x,y の小さい方
pow(x,y)	$x**y$ の実部 例 pow(-.5,1.5) はゼロであり 0.35355i ではない。
pwr(x,y)	$abs(x)**y$
pwrsgn(x,y)	$sgn(x)*abs(x)**y$
rand(x)	x の整数値に依存したランダムな数
random(x)	rand(x) と似ているが値がなめらかに変化する。
round(x)	x に一番近い整数
sgn(x)	x の符号
sin(x)	正弦
sinh(x)	sin の双曲線関数
sqrt(x)	x の平方根の実部 例 sqrt(-1) はゼロであり 0.707107i ではない。
table(x,a,b,c,d,...)	点のペアの組として与えられたテーブルに x の値を加える。
tan(x)	正接
tanh(x)	正接関数の双曲線関数
u(x)	単位ステップ $x \geq 0$ のとき 1 その他はゼロ
uramp(x)	$x \geq 0$ のとき x その他はゼロ

次の演算は、展開の逆順にグループ化されています。

演算	記述
&	ブール代数へ変換する。AND
—	ブール代数へ変換する。OR
^	ブール代数へ変換する。XOR
⋈	左側が右側より大きいとき真、その他が偽
⋋	左側が右側より小さいとき真、その他が偽
⋈=	左側が右側より大きいか等しいとき真、その他が偽
⋋=	左側が右側より小さいか等しいとき真、その他が偽
+	浮動小数点の和
-	浮動小数点の差
*	浮動小数点の掛け算
/	浮動小数点のわり算
**	左の数が底で右の数が指数実数のみを返す。例 -2**1.5 はゼロであり 2.82843i ではない。

5.2.21 .SAVE

保存データの質の制限

あるシミュレータ、特に時間領域シミュレータは、非常に多くのデータを発生します。出力される量は、.save コマンドを用いることによって制限することができます。その結果必要なノード電圧やデバイス電流のみを保存することができます。

Syntax: .save V(out) [V(in) [I(L1) [I(S2)]]] [dialogbox]

コマンド .save I(Q2) は、バイポーラ・トランジスタ Q2 のベース、コレクタ、エミッタ電流を保存します。単体端子電流を保存するには、Ic(Q2) です。

ワイルドカード * and ? を、特定のパターン・マッチしたデータの軌跡を指定するために用いることができます。例えば .save V(*) Id(*) は、各々全ての電圧および全てのドレイン電流を保存します。

もしキーワード dialogbox が指定されると、全ての可能なデフォルトの電圧、電流についてのアイヤログ・ボックスが表示され、保存可能なリストから選択することができます。ネットリストが回路図から生成されると、ノードおよびデバイスが指定でき、回路図上でクリックしダイヤログボックス上に指定されたとして点滅します。

5.2.22 .SAVEBIAS

動作点をディスクに保存

Syntax: .savebias <filename> [internal]

+ [temp=<value>] [time=<value> [repeat]] [step=<value>]
+ [DC1=<value>] [DC2=<value>] [DC3=<value>]

このコマンドは、テキスト・ファイルをディスクへ書き込みます。ディスクは、次のシミュレーションで loadbias コマンドで再読み込みします。もし計算することが困難である直流動作点を持つ回路があると、その解をディスクに保存することが出来、シミュレーションの残りを進める前に直流解を見いだす時間を節約することができます。

キーワード internal は、あるデバイスの内部ノード電圧も保存されることを指定するために付け加えられます。その結果完全な直流解が保存されます。

.tran 解析からある直流解析点を保存したいのであれば、時間を指定することができます。決めてある時間点が書き込まれた後、最初にある時間の点を計算します。この時間によって規定された全ての期間、変更された繰り返して直流解析を実行します。このファイルは、最後に解析された直流点のみを含んでいます。DC1、DC2、DC3 は、.dc スイープ解析から一つの動作点を抽出して与えられます。

.savebias コマンドは、.nodeset コマンドの形式でテキスト・ファイルを書き出します。.nodeset 記述は、唯一の解に対する推奨記述です。すなわち解析器は、.nodeset 記述で与えられているノード電圧の解を繰り返し求めることから始めます。しかし解が有効な範囲にはいるまで繰り返し続けます。もし直流動作点から .tran 解析を再スタートしたけ

れば、この直流状態から解析器を強制的に始めるため、`.nodeset` から `.ic` へ変更してファイルを編集することが出来ます。全ての回路に含まれるリアクタンスの現在までの状態は、`.savebias` ファイルに保存されませんので、この変更のまま進行します。

5.2.23 .STEP

パラメータ・スイープ

このコマンドは、温度、モデルパラメータ、グローバル・パラメータあるいは独立電源をステップ上に変化させて解析を実行します。ステップは、線形、対数あるいは特別に選んだ値のリストとして選ぶことが出来ます。

Example: `.step oct v1 1 20 5`

この例では、独立電圧源を対数的に 1 から 20 まで、オクターブ当たり 5 ポイントで求めます。

Example: `.step I1 10u 100u 10u`

この例では、独立電流源 I1 を対数的に $10\ \mu$ から $100\ \mu$ まで、 $10\ \mu$ 間隔で求めます。

Example: `.step param RLOAD LIST 5 10 15`

この例では、グローバル・パラメータ Rload を 5、10 および 15 の値で三回シミュレーションを実行します。

Example: `.step NPN 2N2222(VAF) 50 100 25`

この例では、NPN モデル・パラメータ VAF をステップ 25 で 50 から 100 までシミュレーションします。

Example: `.step temp -55 125 10`

この例では、 $10\ deg$ ステップで $-55\ deg$ から $125\ deg$ までシミュレーションします。ステップの掃引は、三階層までネストすることが出来ます。

5.2.24 .SUBCKT

サブサーキットの定義

回路を定義する補助として、繰り返し回路は、サブサーキット定義に含ませることが出来、同じ回路の多重回路として用いることが出来ます。シミュレーションを走らせる前に、回路はサブサーキットで定義された回路素子を持った回路を当てはめ全体の回路を作成します。サブサーキットの大きさや複雑さに対する制限は、ありません。

サブサーキットの終わりは、`.ends` コマンドでなければなりません。

次は、サブサーキットを用いた例です。

```
*
* This is the circuit definition
X1 a b 0 divider
```

```
V1 a 0 pulse(0 1 0 .5efbdb5 .5efbdb5 0 1efbdb5)

* this is the definition of the subcircuit
.subckt divider n1 n2 n3
r1 n1 n2 1k
r2 n2 n3 1k
.ends

.tran 3efbdb5
.end
```

Which runs after expanding to

```
* Expand X1 into two resistor network
r:1:1 a b 1k
r:1:2 b 0 1k
*
v1 a 0 pulse(0 1 0 .5efbdb5 .5efbdb5 0 1efbdb5)
.tran 3efbdb5
.end
```

サブサーキット上の単一名称とサブサーキットで定義されている名前が、サブサーキットを回路に適用するとき回路素子として作成されます。

5.2.25 .TEMP

温度掃引

これは温度に対するステップ・コマンドの古典的形式です。示されている各温度に対して、シミュレーションを実行します。

```
syntax .TEMP <T1> <T2> ...
```

これは、次と同じです。

```
.STEP TEMP LIST <T1> <T2> ...
```

5.2.26 .TF

直流小信号伝達関数を見いだす。

これは、独立電源の小変化に対するノード電圧あるいは枝電流の直流小信号伝達関数を見いだします。

```
Syntax: .TF V(<node>[, <ref>]) <source>
        .TF I(<voltage source>) <source>..
```

Examples:

```
.TF V(out) Vin
.TF V(5,3) Vin
.TF I(Vload) Vin..
```

5.2.27 .TRAN

非線形過渡解析を実行。

過渡解析を実行します。これは、回路のもっとも直接的なシミュレーションです。回路に電力が供給されているとき、何が起こるかと言うことを計算します。テスト信号が、独立電源としてしばしば適用されます。

Syntax: `.TRAN <Tstep> <Tstop> [Tstart [dTmax]] [modifiers]`
`.TRAN <Tstop> [modifiers]`

当初の形は、通常の `.tran spice` コマンドです。Tstep は、波形の描画間隔ですが、初期ステップ・サイズの推定値としても用いられます。LTspice は、波形圧縮を用います。そこでこの値は、小さい値で、省略されたりあるいはゼロに設定されます。Tstop は、シミュレーション期間です。過渡解析は、常に時間ゼロから解析されます。しかし Tstart が指定されていますと、波形データは、ゼロ時間から Tstart まで保存されません。このコマンドは、最初のシミュレーションを無視することにより波形のファイル・サイズを変更する一つの方法です。最後のパラメータ dTmax は、回路方程式を積分する間に行われる最大時間ステップです。もし Tstart や dTmax が指定されなければ、Tstep を指定しなければなりません。

いくつかの変形を、`.tran` 行に置くことができます。

5.2.28 .WAVE

`.Wav` ファイルへ選択するノードを書き込む。

LTspice は、`.wav` オーディオ・ファイルを書くことができます。このファイルは、聞くことができますし、他のシミュレーションの入力として用いることもできます。

Syntax: `.wave <filename.wav> <Nbits> <SampleRate>`
`V(out) [V(out2) ...]`

example: `.wave C:\output.wav 16 44.1K V(left) V(right)`

<ファイル名.wav> は、作成したい wav ファイルの完全な絶対パスかシミュレーション回路図あるいはネットリストを含む相対パスです。ダブル・クォーテーションは、空白を含むパスを指定することに用いることができます。<Nbits> は、サンプリング・ビット数です。有効範囲は、1 から 32 ビットです。<SampleRate> は、一シミュレーション時間あたりに記載されるサンプル数です。有効範囲は、一秒あたり 1 から 4294967285 サンプルです。シンタックスの残りの行は、保存したいノードです。各ノードは、`.wav` ファイルにおいて独立したチャンネルです。チャンネル数は、一つのチャンネルから 65535 チャンネル

まで扱えます。ノード電圧と同様にデバイス電流例えば $I_b(Q1)$ を記入することが出来ます。`.wav` アナログからデジタルへの変換は、 $-1 V (or A)$ から $1 V (or A)$ までのフル・スケールです。

パソコンのサウンド・システム上で演奏できない `.wav` ファイルを書くことが出来ます。と言うのもチャンネル数、サンプル数あるいはビット数は、パソコンのコーデックの制限に依存するからです。しかしこの `.wav` ファイルは、他のシミュレーションの入力として LTspice で用いることが出来ます。`.wav` ファイルを LTspice で演奏するための情報については、章 LTspice=>Circuit Elements=>V. Voltage Source I. Current Source を見て下さい。パソコンのサウンド・カードで `.wav` ファイルを演奏したいのであれば、多くの汎用の `.wav` ファイル・フォーマットは、1 あるいは 2 チャンネル、8 あるいは 16bits/channel および 11025、22050 あるいは 44100 Hz サンプル・レートです。

5.3 過渡解析オプション

5.3.1 .TRAN 変形

UIC 直流動作解析をとばし、ユーザーが定めた初期条件を用いる。

steady 定常状態に達したとき、シミュレーションを停止する。

nodiscard 定常状態に達する前に過渡解析シミュレーション部分を消去しない。

startup 独立電圧源や独立電流源を切断して初期動作点を解析する。次に過渡解析をスタートし、シミュレーションの最初の 20 [μs] でこれらの電源を接続する。

atep 回路のステップ応答を計算する。

5.3.2 UIC

初期条件で用いる。通常直流動作点解析は、過渡解析が始まる前に実行されます。このコマンドは、この初期値を示します。ある回路素子の初期条件は、特性の初期値という形で定められます。Uic は、spice の特に推奨される特性ということではありません。直流動作点解析をスキップすることは、実際の初期条件からはずれてしまいます。例えば容量に並列接続された電圧源を考えてみます。ノード電圧は、特に与えられなければゼロです。そして最初の電圧ステップで、無限大の電流が容量に電荷を蓄積させるため流れます。シミュレータは、電流を極点にするような非常に短い時間ステップを見いだすことが出来ません。そして時間ステップが極端に短く収束しない*1 というメッセージを出力します。

5.3.3 startup

このコマンドは、spice のオリジナル・コマンド `uic` に似ています。独立電源は、シミュレーションの最初の 20 [μs] 間徐々に立ち上がるべきであることを意味しています。しかし直流動作点解析は、`.ic` コマンド上で規定された制限を使って実行されます。

*1 time step too small convergence fail.

5.3.4 steady

定常状態に達したとき、シミュレーションを停止します。これは有効な計算結果報告として要求されます。定常状態の検知は、SMPS マクロ・モデルに書き込まれます。一般には、これらのことは、誤差アンプのクロック当たりの平均出力電流がゼロを見つけるように記述されます。このアルゴリズムは、誤差アンプの出力が境地の範囲となるようにします。ゼロ電流と考えられるようなピーク電流の場合は、sstol オプションで考慮されます。

自動定常状態検知は、非常に微妙な状態であるいはそれほど微妙でないような状態で失敗することがあります。次の方法でインターラクティブに定常状態を定めることが出来ます。シミュレーションが始まるとすぐに、メニュー・コマンド **Simulate => Efficiency Calculation => Mark start** を実行します。このコマンドを最初に行ったとき、LTspice へ手動で制限を加えたいと伝えます。回路が定常状態に達しそうになるとき、このコマンドを再び実行します。このコマンドは、過去の履歴を消去し、与えられた条件で再度シミュレーションを始めます。しばらくの後、10 クロック少しして **Simulation => Efficiency Calculation => Mark End** を実行します。**Simulation => Efficiency Calculation => Mark Start** を実行するたびに、制限された計算を再実行し、波形の履歴を消去します。これはファイル・データを巨大にせず描画表示を加速するための良い方法です。これは保存したくない蓄積されるデータを消去したいとき、繰り返し **Simulation => Efficiency Calculation => Mark Start** を実行することが望まれます。

定常状態を見いだすとき、過渡解析時間を減らすためノード電圧やインダクタ電流の初期値を与えるために .ic コマンドが使われます。

5.3.5 nodiscard

定常状態に達する前に、過渡解析部分を消去しない。

5.3.6 step

回路のステップ応答を計算。この関数は、ステップ電流のリストを持つ負荷として用いられる電流源と一緒に動作します。手順は、次の通りです。

1. 定常状態を計算し、もし nodiscard が設定されていなければ履歴を消去します。
2. 20 [A/ μ s] の割合で次の電流をステップ状に印可します。
3. 定常状態を計算します。
4. 次のステップへ変更します。もし値がなければ終了します。

回路の複雑さにより、自動的に STEP 変化を検知することは出来ません。過渡シミュレーションを走らせ必要なステップ負荷応答の出発点終了点を観測するには、.TRAN コマンドを用いるのがもっとも適切です。出力負荷電流を変化させ希望の時間で異なったレベルへと変更するためには、PWL コマンドを用いて下さい。例えば

```
PWL(0 0.5 1m 0.5 1.01m 0.1 3m 0.1 3.01m 0.5)
```

負荷電流は、時間ゼロにおいて 0.5 A からスタートします。 1 ms では、 0.5 A に留まります。時間 1.01 ms で 0.1 A に切り替わります。 3 ms まで 0.1 A に留まります。 3.01 ms で 0.5 A へと変化し、 0.5 A に留まります。PWL は、(時間、値) の制限はありません。

5.4 回路素子

5.4.1 A 特別な関数

Symbol names: INV, BUF, AND, OR, XOR, SCHMITT, SCHMTBUF,
SCHMTINV, DFLOP, VARISTOR, and MODULATE

Syntax: Annn n001 n002 n003 n004 n005 n006 n007 n008
<model> [instance parameters]

Linear Technology Corporation 所有の特別な関数であるミックス・モード・シミュレーション・デバイスがあります。これらのモデルやビヘービア・モデルのほとんどが、LTspice で度々新しいモデルとして変更されているため、文章化されていません。しかしこれらのモデルは、一般的に注目されているため、いくつかは文章化しています。

INV、BUF、AND、OR および XOR は、一般化された理想的なビヘービア・モデルです。全てのゲートは、8個の端子でネットリストが作られています。これらのゲートは、外部電力が必要になりません。電流は出力、6および7 pin に出力されたり入力されたりし、共通端子8 pin へ戻ってきます。1から5 pin は、入力です。使われていない入出力は、8 pin です。デジタル・デバイス・コンパイラは、その端子が使われておらずシミュレーションから取り除くことを知っています。このことは、ANDゲートの入力が接地あるいはゼロ電圧となっているとき違った動作をするのではないかと勘違いされます。もし接地がゲートの共通端子となっていると、接地入力は論理偽として動作しませんが、単にシミュレーションの素子とは見なされないだけです。これらのゲートがこの様に構成されているのは、一つの素子が2, 3, 4, 5端子を真、逆、コンプリメンタリ出力を持ち、シミュレーション速度に対して使われていないゲートが悪さをしないようにするためです。つまり AND デバイスは、ANDゲートの1 2の異なったタイプとして動作します。ゲートは、 0.5 V の論理スレッシュホールドを持ち、伝送遅延無し、 1 ohm 出力インピーダンス、 $0\text{ V}/1\text{ V}$ ロジックがデフォルトとなっています。出力特性は、次のように設定されています。

パラメータ	デフォルト	意味
Vhigh	1 V	論理高レベル
Vlow	0 V	論理低レベル
Trise	0	上昇時間
Tfall	Trise	下降時間
Tau	0	出力 RC 時定数
Cout	0	出力容量
Rout	1	出力インピーダンス
Rhigh	Rout	論理高レベルでのインピーダンス
Rlow	Rout	論理低レベルでのインピーダンス

全てのパラメータが、同時に同じ特性で定義できるわけではありません。例えば出力特性は、上昇時間あるいは RC 時定数であり同時ではありません。

伝送遅延は、デフォルトでゼロであり、特性パラメータは、Td です。入力保持時間は伝送遅延と等しくなります。

入力論理スレッシュホールドのデフォルトは、0.5 ($V_{high} + V_{low}$) ですが、特性パラメータ Ref に設定することが出来ます。保持時間は、伝送遅延時間に等しくなります。

シュミット・トリガ素子は、ゲートと似たような出力特性を持っています。これらの変換点は、特性パラメータ Vt と Vh によって決められます。低い方の転換点は、Vt-Vh で、高い方の転換点は、Vt+Vh です。

ゲートとシュミット・トリガ素子は、時間ステップ情報をデフォルトではシミュレーション・エンジンへ伝えません。すなわち状態を変化し状態変化のいずれかのサイドで時間ステップが閉じていても検知しません。特性パラメータ tripdt は、シミュレータが状態を変化させる最大の時間ステップを規定するように設定することが出来ます。

VRISTOR は、電圧可変バリスタです。ブレイクダウン電圧は、端子 1 と 2 の間の電圧によって設定されます。ブレイクダウンでのインピーダンスは、特性パラメータ rclamp によって設定されます。例として与えた図./examples/Educational/varistor.asc を見て下さい。

MODULATE 素子は、電圧制御発振器です。例図./examples/Educational/PLL.asc を見て下さい。瞬時発振周波数は、FM 入力に電圧を掛けて設定します。電圧から周波数への変換は、線形で二つの特性パラメータ mark と space によって設定されます。Mark は、FM 入力が 1 V のときの周波数、space は、入力が 0 V のときの周波数です。振幅は、AM 入力のときの電圧です。入力が決まっていなかったり (MODULATE common に接続されていると) デフォルトは、1 V です。

LTspice の回路図入力を特別な方法でこれらの素子のシンボルを結びつけます。全ての接続されていない端子は、自動的に端子 8 に接続されます。また端子 8 が接続されていない場合、ノード 0 に接続されます。

5.4.2 任意ビヘービア電圧・電流源

Symbol names: BV, BI

```
Syntax: Bnnn n001 n002 V=<expression> [ic=<value>]
        + [tripdv=<value>] [tripdt=<value>]
        + [laplace=<expression> [window=<time>]]
        + [nfft=<number>] [mtol=<number>]]

        Bnnn n001 n002 I=<expression> [ic=<value>]
        + [tripdv=<value>] [tripdt=<value>] [Rpar=<value>]

        + [laplace=<expression> [window=<time>]]
        + [nfft=<number>] [mtol=<number>] ]
```

最初のシンタックスは、ビヘービア電圧源、後のシンタックスはビヘービア電流源を定義しています。電流源に対して、並列抵抗が Rpar 特性パラメータとして与えられます。

tripdv および tripdt は、syep rejection を制御します。電源電圧が、tripdt 秒で tripdv 電圧より大きく変化しますと、シミュレーション時間ステップが拒否されます。

Expression を次のように含ませることが出来ます。

- ノード電圧、例えば V(n001)
- ノード電圧差、例えば V(n001, n002)
- 回路素子電流；例えば I(S1)、スイッチ S1 に流れる電流あるいは Ib(Q1)。しかし回路素子電流は、準静的に変化しています。すなわち参照デバイスに流れる電流とビヘービア電源出力の間に瞬間的な帰還はありません。
- キーワード time は、シミュレーションでの現時点を意味しています。
- キーワード pi は、3.14159265358979323846 です。
- 関数

名前	機能
abs(x)	絶対値
acos(x)	x のアーク・コサイン
acosh(x)	アーク・ハイパブリック・コサイン
asin(x)	アーク・コサイン
asinh(x)	アーク・ハイパブリック・サイン
atan(x)	アーク・タンゼント
atan2(y, x)	y/x の四現象アーク・タンゼント
atanh(x)	アーク・ハイパブリック・タンゼント
cos(x)	コサイン
cosh(x)	ハイパブリック・コサイン
exp(x)	指数関数
ln(x)	自然対数
log(x)	常用対数
log10(x)	底 10 常用対数
sgn(x)	ジグン
sin(x)	サイン
sinh(x)	ハイパブリック・サイン
sqrt(x)	平方根
tan(x)	タンゼント
tanh(x)	ハイパブリック・タンゼント
u(x)	単位ステップ、もし $x > 0$ なら 1, その他は 0
buf(x)	もし $x > 0.5$ なら 1, その他は 0
inv(x)	もし $x > 0.5$ なら 0, その他は 1
uramp(x)	もし $x > 0.5$ なら x, その他は 0
!(x)	inv(x) の変形シンタックス
int(x)	x を整数へ変換
floor(x)	x と同じか小さい整数
ceil(x)	x と同じか大きい場合整数
rand(x)	x の整数値に依存する 0 と 1 の間の乱数
random(x)	rand() に似ているが値が滑らかに変化する
min(x, y)	x あるいは y の内小さい方
max(x, y)	x あるいは y の内大きい方
limit(x,y,z)	min(max(x,y),z) と同じ
if(x,y,z)	もし $x > 0.5$ ならば、y または z
table(x,a,b,c,d,...)	点の組み合わせとして与えられているテーブルに基づいた x の値

- 演算、展開の逆順でグループ分けしてあります。

	説明
&	表現をブール数へ変換、AND
—	表現をブール数へ変換、OR
^	表現をブール数へ変換、XOR
<	左が右より小さければ真、その他は偽
>	左が右より大きければ真、その他は偽
<=	左が右より小さいか等しければ真、その他は偽
>=	左が右より大きい等しければ真、その他は偽
+	浮動点加算
-	浮動点減算
*	浮動点掛け算
/	浮動点割り算
**	左の数値に対する右の数値乗
!	次の表示をブール数へ変換し逆数を取る

真の数値は 1、偽の数値は 0 です。ブール数への変換は、もし値が 0.5 より大きければ 1、その他では 0 へ変換します。

オプションのラプラス変換が定義されていれば、ビヘービアの電流あるいは電圧信号の結果に対してこの変換が適用されます。ラプラス変換は、 s の関数でなければいけません。周波数 f での周波数応答は、 s を代入し $\sqrt{-1} \cdot 2 \cdot \pi \cdot f$ となります。時間領域の振る舞いは、インパルス応答から求められ、周波数応答のフーリエ変換から求められます。LTspice は、適切な周波数範囲と精度を推測しなければなりません。応答は、高周波で省略され、エラーが報告されます。LTspice は、最初にこのことを推測するようにし、`retol` を減らして精度をチェックし、明示的にウィンドウの `nfft` を設定するようにすべきです。ウィンドウ表示値の逆数が周波数解像度です。この解像度の `nfft` 倍の値がもっとも高い周波数となります。ブール代数 XOR 演算子 `^` は、ラプラス変換で用いられるとき、指数関数 `**` を意味すると解釈されます。

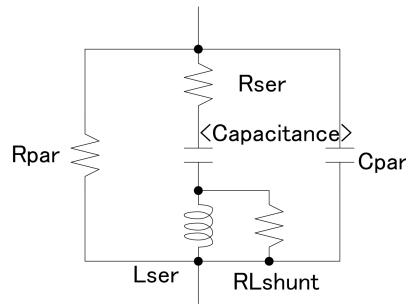
5.4.3 容量

Symbol names: CAP, POLCAP

```
Syntax: Cnnn n1 n2 <capacitance> [ic=<value>]
        + [Rser=<value>] [Lser=<value>] [Rpar=<value>]

        + [Cpar=<value>] [m=<value>]
        + [RLshunt=<value>]
```

容量は、直列抵抗、直列インダクタンス、並列抵抗と直並列容量によって表現されます。等価回路を、次に示しておきます。



名前	パラメータ
Rser	等価直列抵抗
Lser	等価直列インダクタンス
Rpar	等価並列抵抗
Cpar	等価並列容量
RLshunt	Lser に並列な分流抵抗
m	並列ユニットの数
ic	uic が、tran カードでフラグが立っているときだけに使われる初期電圧

表 5.7 Capacitor Instance Parameter

改めて描くより、容量に寄生の Rpar、Rser、RLshunt、Cpar、Lser を含ませる方が計算上好ましくなります。LTspice は、内部ノード無しで物理的容量のこのモデルシミュレーションする適切な回路シミュレーション技術を使っています。この方法は、シミュレーション行列をより小さくし、計算速度が速く、短時間の時間ステップで異常値をより少なくします。

容量は、これらの寄生素子を含んでいますので、 piezo 結晶の基本マクロモデルを使うのに便利です。

また汎用的な非線形容量もあります。容量を細工する代わりに、電荷の表現を書きます。LTspice は、この表現をコンパイルし、全ての変数に渡って記号的に微分し、容量に関する偏微分を見いだします。

Syntax: Cnnn n1 n2 Q=<expression> [ic=<value>] [m=<value>]

素子間の電圧を示す特別な変数 x が存在します。そこで 100 pF の容量は、次のように記載されます。

```
Cnnn n1 n2 Q=100p*x
```

ゼロ電圧で 100 pF から 300 pF へと急激に変化する容量は、次のように書くことができます。

```
Cnnn n1 n2 Q=x*if(x<0,100p,300p)
```

この素子は、仮想的な電荷モデルの振る舞いを表現するのに便利です。例えばトランジスタ。

5.4.4 ダイオード

Symbol Names: DIODE, ZENER, SCHOTTKY, VARACTOR.

Syntax: Dnnn anode cathode <model> [area]
+ [off] [m=<val>]

例

```
D1 SW OUT MyIdealDiode
.model MyIdealDiode D(Ron=.1 Roff=1Meg Vfwd=.4)
```

```
D2 SW OUT dio2
.model dio2 D(Is=1e-10)
```

ダイオードは、その特性を決めるための.model カードが必要です。二つのダイオード・タイプがあります。一つは、conduction region-wise linear model です。これは、線形ダイオードの軽量化された表現のモデルです。伝導について、三つの領域を持っています：on、off と逆電圧でのブレークダウンです。

下記はこのタイプのダイオードのモデル・パラメータです。

名称	パラメータ	単位	デフォルト
Ron	順方向の抵抗	Ohm	1.
Roff	off 状態の抵抗	Ohm	1/gmin
Vfwd	伝導状態に入るときの順方向電圧	V	0.
Vrev	逆電圧のブレークダウン電圧	V	infin.
Rrev	ブレークダウン抵抗	Ohm	Ron

この理想的なモデルは、Ron、Roff、Vfwd、Vrev あるいは Rrev のいくつかがモデルの中で変更されるときに用いられます。

他の有効なモデルは、標準的な Berkeley SPICE semiconductor diode です。しかし詳細なブレークダウン特性や再結合電流を扱うことが出来ます。面積ファクターは、このモデルの並列される数を与えます。下記は、このダイオードのモデル・パラメータです。

名称	パラメータ	単位	デフォルト	例
Is	飽和電流	A	$1e-14$	$1e-7$
Rs	オーミックな抵抗	Ohms	0	10
N	エミッション係数	-	1	1.
Tt	トランジット・タイム	sec	0	0.1 ns
Cjo	ゼロ・バイアス接合容量	F	0	2 p
Vj	接合ポテンシャル	V	1	0.6
M	傾斜係数	-	0.5	0.5
Eg	活性エネルギー	eV	1.11	1.11 Si 0.69 Sbd 0.67 Ge
XTI	飽和電流温度指数	-	3.0	3.0 jn 2.0 Sbd
Kf	フリッカ雑音係数	-	0	
Af	フリッカ雑音指数	-	1	
Fc	順方向電圧空乏層容量式の係数	-	0.5	
BV	逆方向ブレイクダウン電圧	V	infin.	40.
Ibv	ブレイクダウン電圧の電流	A	$1e-10$	
Tnom	パラメータ測定温度	°C	27	50
Isr	再結合電流パラメータ	A	0	
Nr	Isr エミッション係数	-	2	
Ikf	高注入ニー電流	A	infin.	
Tikf	線形 Ikf 温度係数	/°C	0	
Trs1	線形 Rs 温度係数	/°C	0	
Trs2	二乗 Rs 温度係数	/°C/°C	0	

表 5.8 ダイオード・パラメータ

モデルの電圧、電流、電力消費率を決めることが可能です。これらのモデル・パラメータは、電氣的振る舞いに影響しません。ダイオードが能力を超えて用いられているかどうかを調べることが、LTspice では可能です。次のパラメータがモデルに対して適用できます。

名称	パラメータ	単位	デフォルト
Vpk	ピーク電圧率	V	(none)
Ipk	ピーク電流率	A	(none)
Iave	平均電流率	A	(none)
Irms	RMS 電流率	A	(none)
diss	最大電力消費率	W	(none)

5.4.5 電圧制御電圧源

Symbol Names: E, E2

三つのタイプの電圧制御電圧源があります。

Syntax: Exxx n+ n- nc+ nc- <gain>

この回路素子は、端子 nc+ と nc- の間の入力電圧に依存した端子 n+ と n- の出力電圧を発生します。この素子は、一定の利得によって決められ、線形に入力電圧に依存します。

Syntax: Exxx n+ n- nc+ nc- table=(*<value pair>*, *<value pair>*, ...)

伝達関数を決めるために、ルックアップ・テーブルが用いられます。テーブルは、二つの数値のペアとなっています。ペアの二番目の値は、制御電圧がペアの最初の値に等しいときの出力電圧です。出力は、制御電圧が指定した点の間にあるとき、線形に自動付与されます。もし制御電圧がルックアップ・テーブルの範囲を超えるときには、出力電圧はルックアップ・テーブルの最後の点の電圧と同じ値となります。

Syntax: Exxx n+ n- nc+ nc- Laplace=*<func(s)>*
+ [window=*<time>*] [nfft=*<number>*] [mtol=*<number>*]

この回路素子の伝達関数は、ラプラス変換によって決められます。ラプラス変換は、 s の関数でなければなりません。周波数 f での応答は、 $\text{aqrt}(-1)*2*\text{pi}*f$ の値の s を代入することによって見いだされます。時間領域の振る舞いは、周波数領域のフーリエ変換から求められるインパルス応答から求められます。LTspice は、適切な周波数領域と精度とを推測する必要があります。応答は、高周波のところで止まるかあるいはエラーを報告する必要があります。LTspice が、最初にこの推測を行うことが出来るようにし、*reitol* を減少させるか、自動的に *nfft* や *window* を調べられるようにすることが望まれます。*window* の値の逆数は、周波数解像度です。この解像度の *nfft* 倍の値は、考慮している最高の周波数です。ブール代数式[^]は、ラプラス表現で用いられるとき、自乗**と見なされます。

Syntax: Exxx n+ n- value={*<expression>*}

これは、ビヘービア電圧源 B の変形シンタックスです。

Syntax: Exxx n+ n- POLY(*<N>*) *<(node1+,node1-) (node2+,node2-)>*
+ ... *<(nodeN+,nodeN-)> <c0 c1 c2 c3 c4 ...>*

これは、多項式を含んだビヘービア・モデルの古い形式です。Linear Technology ビヘービア・モデルを含んだ回路をシミュレーションするとき役に立ちます。

E 電源を用いるよりも E 電源を近似するために、抵抗が並列に挿入された G 電源を用いる方が望ましいでしょう。抵抗が並列に挿入された電圧制御電流源は、電圧制御電圧源よりも計算が速く、収束の問題があまり生じません。また有限の出力インピーダンスは、実際の回路を巧く表現しています。

5.4.6 電流制御電流源

Symbol Name: F

Syntax: Fxxx n+ n- *<Vnam>* *<gain>*

この回路素子は、端子 n+ と n- の間に電流を供給します。電流の値は、*<Vnam>* で決められた電圧源を流れる電流の *gain* 倍の値に等しくなります。

Syntax: Fxxx n+ n- value={<expression>}

これは、ビヘービア電圧源 B の変形シンタックスです。

Syntax: Fxxx n+ n- POLY(<N>) <V1 V2 ... VN> <c0 c1 c2 c3 c4 ...>

これは、多項式で記述されるビヘービア・モデルの古い形式です。Linear Technology ビヘービア・モデルを計算するときに役立ちます。

5.4.7 電圧制御電流源

Symbol Names: G, G2

三つのタイプの電圧制御電流源があります。

Syntax: Gxxx n+ n- nc+ nc- <gain>

この回路素子は、ノード nc+ と nc- の入力電圧に依存した出力電流をノード n+ と n- に発生します。この出力電流は、一定の利得によって生じます。

Syntax: Gxxx n+ n- nc+ nc- table=(<value pair>, <value pair>, ...)

ここで伝達関数を決めるために、ルックアップ・テーブルが用いられます。ペアになっている二番目の値は、制御電圧がペアの最初の値と等しくなったときの出力電流です。出力は、制御電圧が規定された点の間にあるとき線形に内挿されます。もし制御電圧がルックアップ・テーブルの範囲を超えたとき、出力電流はルックアップ・テーブルの最後の値の一定電流として外挿されます。

Syntax: Gxxx n+ n- nc+ nc- Laplace=<func(s)>
+ [window=<time>] [nfft=<number>] [mtol=<number>]

この素子の伝達関数は、ラプラス変換によって決められます。ラプラス変換は、s の関数でなければなりません。周波数 f での応答は、 $\sqrt{-1} \cdot 2 \cdot \pi \cdot f$ の s を代入することによって求められます。時間領域の振る舞いは、インパルス応答から求められます。それは、周波数領域応答のフーリエ変換から求められます。LTspice は、適切な周波数範囲と精度を推測する必要があります。応答は周波数の高いところで止まり、エラーを報告する必要があります。LTspice が最初にこの推測が出来るようにし、reitol を減少し精度をチェックするかもしくは nfft や window を設定しておくことを薦めます。window 値の逆数は、周波数精度です。精度の nfft 倍は、考慮されているもっとも高い周波数です。ブール代数⁶は、ラプラス表現で用いられている**を意味しています。

Syntax: Gxxx n+ n- value={<expression>}

これは、ビヘービア・モデル B の変形シンタックスです。

Syntax: Gxxx n+ n- POLY(<N>) <(node1+,node1-) (node2+,node2-)>
+ ... (nodeN+,nodeN-)> <c0 c1 c2 c3 c4 ...>

これは、多項式を持ったビヘービア・モデルの古典的表現です。Linear Technology ビヘービア・モデルが存在する回路を計算するときに有効です。

5.4.8 電流制御電圧源

Symbol Name: H

Syntax: Hxxx n+ n- <Vnam> <transresistance>

この素子は、ノード n+ と n-間の電圧に適用されます。適用される電圧は、電圧源<Vnam>を流れる電流の利得倍となります。

Syntax: Hxxx n+ n- value={<expression>}

これは、ビヘービア電圧源 B の変形シンタックスです。

Syntax: Hxxx n+ n- POLY(<N>) <V1 V2 ... V3> <c0 c1 c2 c3 c4 ...>

これは、多項式であるビヘービア・モデルの古典的形式です。Linear Technology ビヘービア・モデルがある回路を計算するときに役立ちます。

5.4.9 電流源

Symbol Name: CURRENT

Syntax: Ixxx n+ n- <current> [AC=<amplitude>] [load]

この素子は、ノード n+ と n-に一定の電流を供給します。電源が負荷として用いられている場合、その電源は電力を消費します。すなわちもしノード n+ と n-の間の電圧がゼロあるいは負の値ですと流れ出す電流はゼロです。この目的は、出力電圧がゼロであるなら電流が流れ出さない電力源の電流負荷となります。

交流解析に対して、AC の値は解析周波数点での電源の振幅として用いられます。

Syntax: Ixxx n+ n- PULSE(Ioff Ion Tdelay Trise Tfall Ton Tperiod Ncycles)

時間依存パルス電流源

名称	パラメータ	単位
Ioff	初期値	Amps
Ion	パルスの値	Amps
Tdelay	遅延	seconds
Tr	立ち上がり時間	seconds
Tf	たち下がり時間	seconds
Ton	パルス幅	seconds
Tperiod	周期	seconds
Ncycles	サイクルの数	cycles

表 5.9 時間依存パルス電流源

Syntax: Ixxx n+ n- SINE(Ioffset Iamp Freq Td Theta Phi Ncycles)

時間依存性正弦波電流源

名称	パラメータ	単位
Ioffset	直流オフセット	Amps
Iamp	振幅	Amps
Freq	周波数	Hz
Td	遅延	seconds
Theta	ダンピング・ファクター	1/seconds
Phi	正弦波の位相	degrees
Ncycles	サイクル数の数	cycles

表 5.10 時間依存性正弦波電流源

Td より遅い時間あるいは Ncycles より後の時間に対して出力電流は、Ioffset+Iamp*(pi*phi/180) で与えられます。その他の場合には Ioffset+Iamp*exp(-(time-Td)*Theta)*sin(2*pi*freq*(time-Td)+pi*phi/180) で与えられます。

damping Factor あるいは Theta は、減衰時定数の逆数です。

Syntax: Ixxx n+ n- Ixxx n+ n- EXP(I1 I2 Td1 Tau1 Td2 Tau2)

時間依存性指数関数型電流源

名称	パラメータ	単位
I1	初期値	Amps
I2	パルスの値	Amps
Td1	立ち上がり遅延時間	seconds
Tau1	立ち上がり時定数	seconds
Td2	立ち下がり遅延時間	seconds
Tau2	立ち下がり時定数	seconds

表 5.11 時間依存性指数関数型電流源

Td1 より小さな時間に対して、出力電流は I1 です。Td1 と Td2 の間の時間に対して電流は、次のように与えられます。

$$I1+(I2-I1)*(1-\exp(-(time-Td1)/Tau1)) \\ +(I1-I2)*(1-\exp(-(time-Td2)/Tau2))$$

Syntax: Ixxx n+ n- SFFM(Ioff Iamp Fcar MDI Fsig)

時間依存性単一周波数 FM 電流源

名称	パラメータ	単位
Ioff	直流オフセット	Amps
Iamp	振幅	Amps
Fcar	搬送波周波数	Hz
MDI	変調指数	(none)
Fsig	信号周波数	Hz

表 5.12 時間依存性単一周波数 FM 電流源

電流は次のように与えられます。

$$I_{off} + I_{amp} \sin((2 \cdot \pi \cdot F_{car} \cdot \text{time}) + MDI \cdot \sin(2 \cdot \pi \cdot F_{sig} \cdot \text{time}))$$

Syntax: Ixxx n+ n- tbl=(`<voltage, current>`,
`<voltage, current>`, ...)

電流は、ルックアップ・テーブルでの出力ノード電圧の関数として決められます。これは負荷の特性をモデル化するのに役立ちます。

Syntax: Ixxx n+ n- `<value>` step(`<value1>`, [`<value2>`],
`<value3>`, ...) [load]

これは電流源の特別な形式です。電流は、過渡解析ステップ負荷応答で用いられる電流のリストとして決められます。このモードでは、シミュレーションは、定常状態がリスト `<value1>` の最初の電流に達するまで計算し続けます。電流は、リスト `<value2>` の次の値で変化します。シミュレーションは、定常状態がその電流値になるまで進行します。電流は、次のステップへ移動します。それからリストが完了するまで繰り返します。もし `.tran` コマンドが `step` を規定していなければ、最初の `<value>` が用いられます。

Syntax: Ixxx n+ n- R=`<value>`

これは電流源ではなく、抵抗です。負荷が電流源として用いられるとき、抵抗負荷をモデル化するために用いられます。

Syntax: Ixxx n+ n- PWL(`t1 i1 t2 i2 t3 i3...`)

任意のピースワイズ・リニア電流源

`t1` 時間の前で、電流は `i1` です。時間 `t1` と `t2` では、電流は `i1` と `i2` の間を直線変化します。任意の時間では、その点の電流を与えます。最後の時間の後は、電流は最後に指定した時間のままです。

Syntax: Ixxx n+ n- wavefile=`<filename>` [chan=`<nnn>`]

LTspice への入力として用いられる `.wave` ファイルが使えます。`<filename>` は、`.wave` ファイルへのフルパス、絶対パスであるかシミュレーションしている回路図もしくはネットリストを含むディレクトリからの相対パスです。ダブル・クォテーションは、パスを含むスペースを規定するために用いることができます。`.wave` ファイルは、65536 チャンネル、

0 から 65535 まで含ませることが出来ます。Chan は、どのチャネルを用いるかを規定することが出来ます。デフォルトは、最初のチャネル、0 が用いられます。.wave ファイルは、-1 A から 1 A までのフルスケールを持つとして変換されます。

この電源のみ、.tran 解析で意味があります。

5.4.10 JFET トランジスタ

Symbol Names: NJF, PJF

Syntax: Jxxx D G S <model> [area] [off] [IC=Vds, Vgs] [temp=T]

例

```
J1 0 in out MyJFETmodel
.model MyJFETmodel NJF(Lambda=.001)
```

```
J2 0 in out MyPJFETmodel
.model MyPJFETmodel PJF(Lambda=.001)
```

JFET は、その特性を表現するために.model カードが必要です。モデル・カードのキーワード NJF と PJF は、トランジスタの極性を示しています。area factor は、モデルがいくつ並列に接続されているかと言うことを決めています。

JFET モデルは、ゲート再結合電流、インパクト・イオン化を含むように拡張された Shichman and Hodges の FET モデルから導かれています。直流特性は、パラメータ VTO および BETA によって定義されています。それらは、ドレイン電流のゲート電圧による変動；LAMBDA を決め、出力コンダクタンスを決めます。そして Is、二つのゲート接合の飽和電流を決めます。電荷の蓄積は、二つのゲート接合の非線形空間電荷総容量によってモデル化されています。これは、接合電圧の-1/2 乗で変化し、パラメータ Cgs、Cgd、PB によって定義されます。最適化パラメータ B が付け加えられています。次の文献を見て下さい。A. E. Parker and DDDDD. J. Skellern, AnImproved FET Model for Computer Simulators, IEEE Trans CAD vol. 9, no. 5, pp. 551-553, May 1990

名称	パラメータ	単位	デフォルト	例
Vto	スレッシュホールド電圧	V	-2.0	-2.0
Beta	トランスコンダクタンス・パラメータ	A/V/V	1e-4	1e-3
Lambda	チャンネル長変調パラメータ	1/V	0	1e-4
Rd	ドレイン抵抗	Ohm	0	100
Rs	ソース抵抗	Ohm	0	100
Cgs	ゼロバイアス G-S 接合容量	F	0	5pF
Cgd	ゼロバイアス G-D 接合容量	F	0	1pF
PB	ゲート接合ポテンシャル	V	1	0.6
Is	ゲート接合飽和電流	A	1e-14	1e-14
B	ドーピング・テイル・パラメータ	-	1	1.1
KF	フリッカ雑音係数	-	0	
AF	フリッカ雑音指数	-	1	

FC	順方向空間電荷容量係数	-	0.5	
Tnom	パラメータ測定温度	°C	27	50
betatce	トランスコンダクタンス・パラメータ指数温度係数	0	%/°C	
vtotc	スレッシュホールド電圧温度係数	V/°C	0	
N	ゲート接合エミッション係数	-	1	
Isr	ゲート接合再結合電流	A	0	
Nr	Isr のエミッション係数	-	2	
alpha	イオン化係数	1/V	0	
Vk	イオン化ニー電流	V	3	
xTi	飽和電流温度係数	-		

表 5.13: JFET パラメータ

5.4.11 相互インダクタンス

Symbol Names: None

これは、回路図上のテキストとして配置されます。

Syntax: Kxxx L1 L2 [L3 ...] <coefficient>

L1 と L2 は、インダクタの名前です。相互結合係数は、 -1 と 1 の間の値でなくてはなりません。

K1 L1 L2 L3 L4 1.

は、次の 6 個の宣言と同じことです。

K1 L1 L2 1.

K2 L2 L3 1.

K3 L3 L4 1.

K4 L1 L3 1.

K5 L2 L4 1.

K6 L1 L4 1.

値 1 の相互結合係数から始める方がよいようです。これはダンピングが行われていなくシミュレーションを遅くする漏れインダクタンスを少なくします。しかしもし uic コマンドか .tran カードが実行されていると、 -1 あるいは 1 の相互インダクタンス値は、シミュレーションを難しくします。

5.4.12 インダクタ

Symbol Names: IND, IND2

Syntax: Lxxx n+ n- <inductance> [ic=<value>]
+ [Rser=<value>] [Rpar=<value>]

+ [Cpar=<value>] [m=<value>]

等価直列抵抗、直列インダクタンス、並列抵抗それと並列容量とを規定することが出来ます。等価回路は、次のようになります。

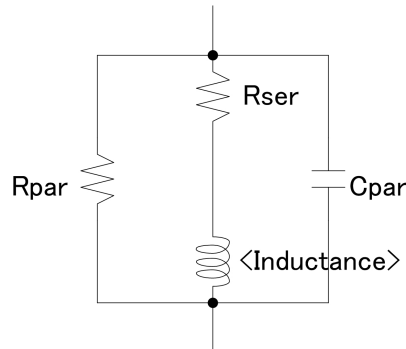


図 5.1 インダクタンス等価回路

名称	パラメータ
Rser	等価直列抵抗
Rpar	等価並列抵抗
Cpar	等価並列容量
m	並列になっている数
ic	初期電流 (uic が.tran カードで使われているときのみ)
tc1	線形インダクタンス温度係数
tc2	温度係数第2項
temp	素子温度

表 5.14 インダクタンス・パラメータ

別の形で描くよりもインダクタンスの中に寄生素子 Rpar や Rser、Cpar を含ませるほうが役に立ちます。LTspice は、任意の内部ノード無しで物理的なインダクタをシミュレーションするために、専用のシミュレーション・テクノロジーを使っています。こうすることによってシミュレーション行列を小さくし、計算を早くできます。そして全ての時間ステップに渡って正常に動作します。

デフォルトでは、LTspice は SMPS 過渡解析に対してインダクタに損失を持たせます。SMPS に対してこれらの損失は、一般に付属しているわけではなく場合によっては取り除くことが出来ます。Tools=>Control Panel=>Hacks! ページで Supply a min. inductor damping if no Rpar is given チェックをはずします。この設定は、プログラムが起動している間記憶しています。またインダクタに対して 1 ミリオームのデフォルト直列抵抗もあります。これは相互インダクタンス記述では述べられていません。この Rser は、SwitcherCAD III が回路の線形行列の大きさを減少させるためテブナン回路の代わりにノートン等価回路としてインダクタに含まれるようにします。もし LTspice にこの最小値の抵抗を含ませたくないならば、そのインダクタに対して明示的に Rser=0 と設定しなければいけません。このことは、LTspice に過渡解析の間インダクタ二の複雑なテブナン等価回路を用いることを要求します。

LTspice で有効な非線形インダクタの二つの型があります。一つは、フラックスの表現を持ったビヘービア・インダクタです。インダクタの電流は、キーワード x によって与えられます。次にしますのは、ネットリストの例です。

```
*
L1 N001 0 Flux=1m*tanh(5*x)
I1 0 N001 PWL(0 0 1 1)
.tran 1
.end
```

上の例で I1 は、単一の dI/dT を与えます。その結果インダクタは、ノード N001 の電圧として読み飛ばされます。

LTspice には、もう一つの非線形インダクタがあります。これは、John Chan et al. in IEEE Transactions On Computer-Aided Design, Vol. 10, No. 4, April 1991 によって最初に提案されたヒステレシス・コア・モデルです。このモデルは、僅か三つのパラメータでヒステレシスを定義しています。

名称	パラメータ	単位
Hc	強制力	
Br	残存フラックス密度	
Bs	飽和フラックス密度	

表 5.15 ヒステレシスがあるインダクタ

上側および下側のヒステレシス・ループは、次のように与えられます。

$$B_{up}(H) = B_s \frac{H + H_c}{|H + H_c| + H_c(B_s/R_r - 1)} + \mu_0 H$$

$$B_{dn}(H) = B_s \frac{H - H_c}{|H - H_c| + H_c(B_s/R_r - 1)} + \mu_0 H$$

これらの関数は、次の図のように描かれます。 H_c および B_r は、主ヒステレシス・ループと H および B 軸との交点です。 B_s は、漸近線と B 軸との交点です。 $B_{sat}(H) = B_s + \mu_0 H$ は、 H が無限になるにつれ近づいていきます。

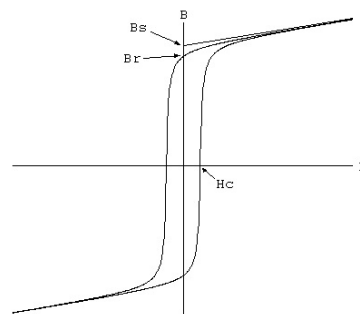


図 5.2 ヒステレシス・ループ

初期磁化曲線は、次のように与えられます。

$$B_{mag}(H) = .5[B_{up}(H) + B_{dn}(H)]$$

マイナー・ループは、引用する参照点当たり上の式の様々な変更によって得られます。コアの絶対あるいは相対パーミアビリティは、H の関数および H の値の履歴です。典型的な電力用フェライト (Hc=16 A-turns/m, Bs=.44T, Br=.10T) の非対称マイナー・ループの例が、下の図に示してあります。

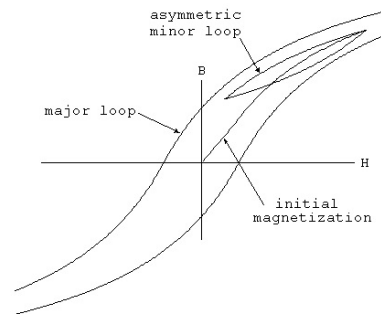


図 5.3 非対称マイナー・ループ例

Hc、Br と Bs のコア特性パラメータに加えて、コアの機械的寸法が必要です。

名称	パラメータ	単位
Lm	磁気長 (excl. gap)	meter
Lg	ギャップ長	meter
A	断面積	meter**2
N	巻き数	-

表 5.16 機械寸法

磁場の有限ギャップを規定したい場合、H は巻き線の電流には比例しません。LTspice は、均一な断面積および薄くて均一に広がっているギャップを仮定してコアおよびギャップの磁場を計算します。

下記の内容は、ギャップのあるインダクタ L1 に対するインダクタンスと電流の例を示しています。電流源 I1 は、単一の di/dt を供給していますので、インダクタンスを読みとることが出来ます。コアは、初期磁場曲線に従います。そしてパーミアビリティは、最初に電流が上昇するにつれ増加し、飽和に達したところで止まります。ギャップは、インダクタンスをコアのパーミアビリティ対して敏感にしなくします。その増加を観察するためには、V(n001) 上で拡大する必要があります。コアの中で H が Hc に等しいとき最大となります。

*

```
L1 N001 0 Hc=16. Bs=.44 Br=.10 A=0.0000251
+ Lm=0.0198 Lg=0.0006858 N=1000
I1 0 N001 PWL(0 0 1 1)
```

```
.tran .5
.options maxstep=10u
.end
```

5.4.13 MOSFET

Symbol Names: NMOS, NMOS3, PMOS, PMOS3

LTspice では、二つの基本的に異なった MOSFET があります。モノリシック MOSFET と新しい垂直二重拡散パワー MOSFET です。

モノリシック MOSFET

```
Syntax: Mxxx Nd Ng Ns Nb <model> [m=<value>] [L=<len>]
        + [W=<width>] [AD=<area>] [AS=<area>]
        + [PD=<perim>] [PS=<perim>] [NRD=<value>]
        + [NRS=<value>] [off] [IC=<Vds, Vgs, Vbs>]
        + [temp=<T>]
```

例

```
M1 Nd Ng Ns 0 MyMOSFET
.model MyMOSFET NMOS(KP=.001)

M1 Nd Ng Ns Nb MypMOSFET
.model MypMOSFET PMOS(KP=.001)
```

垂直二重拡散パワー MOSFET

```
Syntax: Mxxx Nd Ng Ns <model> [L=<len>] [W=<width>]
        + [M=<area>] [m=<value>] [off]
        + [IC=<Vds, Vgs, Vbs>] [temp=<T>]
```

例

```
M1 Nd Ng Ns Si4410DY
.model Si4410DY VDMOS(Rd=3m Rs=3m Vto=2.6 Kp=60
+ Cgdmax=1.9n Cgadmin=50p Cgs=3.1n Cjo=1n
+ Is=5.5p Rb=5.7m)
```

MOSFET モデルカードは、どのタイプのモデルを使うかということを示さねばなりません。モデル・カード・キーワード NMOS と PMOS は、モノリシック N あるいは P チャネル MOSFET を示しています。モデル・カード・キーワード VDMOS は、垂直二重拡散パワー MOSFET を示しています。

モノリシック MOSFET は、四端子デバイスです。Nd、Ng、Ns それと Nb は、ドレイン、ゲート、ソースそれとバルクを示しています。L および W は、メータ単位のチャンネル

長とチャンネル幅を示しています。AD と AS は、平方メートルで示したドレインおよびソース拡散の面積です。サフィックス u は、 μm を示し、p は、 μm の平方を示しています。L、W、AD および AS が示されていない場合、デフォルトの値が用いられます。PD および PS は、メートルで表したドレインおよびソースの周囲長です。NRD および NRS は、ドレインあるいはソース拡散の正方形の数を示しています。これらの値は、.MODEL 制御において決められシート抵抗 RSH に掛けられます。PDPS のデフォルトはゼロ、NRD、NRS のデフォルトは 1 です。OFF は、直流解析のデバイス初期条件を与えます。過渡解析が定常状態の直流値よりも別の点で始めたいとき、IC=VDS、VGS、VBS を用いて与えられる初期条件は、.TRAN 制御上において UID オプションで用いるためです。オプション TEMP 値は、素子が動作するときの温度で、.OPTION 制御の規定されている温度を上書きします。温度の規定は、レベル 1, 2, 3 および 6 MOSFET でのみ有効で、レベル 4, 5, あるいは 8 BSIM 素子では使えません。

LTspice は、七つの異なったタイプのモノリシック MOSFET と一つの垂直二重拡散パワー MOSFET が含まれています。

七つのモノリシック MOSFET 素子モデルがあります。モデル・パラメータ LEVEL は、使われるモデルを規定します。デフォルト LEVEL は、1 です。

LEVEL	モデル
1	Shichman-Hodges
2	MOS2*2
3	MOS3 半経験モデル (LEVEL2 の文献参照)
4	BSIM*3
5	BSIM2*4
6	MOS6*5
8	BSIM3v3.2.4 from University of California, Berkeley as of December 2001
9	BSIMSOI3.2 (Silicon on insulator) *6
12	EKV 2.6 based on code from Ecole Polytechnique Federale de Lausanne*7
14	BSIM4.4.0 from University of California, Berkeley as of March 4, 2004

表 5.17 LEVEL of MOSFET

Level 1 から Level 3 MOSFET の直流特性は、デバイス・パラメータ VTO、KP、LAMBDA および GAMMA によって定義されています。これらのパラメータは、もしプロセスパラメータ NSUB、TOX... が与えられているならば計算されますが、ユーザー定義の値も上書きすることが可能です。VTO は、エンハンス・モードでは正 (負)、デプレッション・モード N チャンネル (P チャンネル) では負 (正) です。電荷蓄積は、三つの一定の値を持つ容量 CGSO、CGDO および CGBO によってモデル化されています。これらは、オーバーラップされた容量を表しています。またこれらの非線形薄酸化膜容量は、ゲート、ソース、ドレインおよびバルク領域の間に分布しています。二つのサブストレート接合の非線形空間電荷層容量は、底辺と周囲とに分割されます。これらの容量は、接合電圧の MJ、MJSW によって変化します。そしてパラメータ CBD、CBS、CJ、CJSW、MJ、MJSW および PB によって決められます。電荷蓄積効果は、Meyer によって提案されたピーシワイズ・リニア電圧依存性容量によってモデル化されています。薄膜酸化電荷蓄積効果は、Level=1 モデルでは少し違った形で与えられています。これらの電圧依存容量は、TOX が規定され

るときのみ含まれます。

接合を記述しているパラメータには、あるオーバーラップがあります。飽和電流は、 I_s [Amp] あるいは J_s [Amp/m/m] を通じて規定されます。一方最初のパラメータは、絶対値であり、二番目の値はドレインもしくはソース接合の逆電流を与えるため各々 A_d 、 A_s を掛けることによって得られます。同じ考えがゼロ・バイアス接合容量 CBD と CBS [Farad] それと CJ [Farad/m/m] にも適用されています。寄生ドレインおよびソース直列抵抗は、RD、RS [Ohms] あるいは RSH [Ohms/square] として表現されます。後者は、NRD および NRS の平方を掛けます。

MOSFET Level 1, 2, 3 および 6 のパラメータ

名称	パラメータ	単位	デフォルト	例
VTO	ゼロ・バイアス・スレッシュホールド電圧	V	0.	1.
KP	トランスコンダクタンス	sqrt(V)	2e-5	3e-5
GAMMA	バルク・スレッシュホールド・パラメータ	V	0.	0.37
PHI	表面ポテンシャル	V	0.6	0.65
LAMBDA	チャネル長変調 (レベル 1, 2 のみ)	1/V	0.	0.02
Rd	ドレイン抵抗	Ohms	0.	1.
Rs	ソース抵抗	Ohms	0.	1.
Cbd	ゼロ・バイアス B-D 接合容量	F	0.	20f
Cbs	ゼロ・バイアス B-S 接合容量	F	0.	20f
Is	バルク接合飽和電流	A	1e-14	1e-15
N	バルク・ダイオード・エミッション係数	-	1.	
Pb	バルク接合ポテンシャル	V	0.8	0.87
Cgso	単位チャネル幅当たりのゲート・ソース容量	F/m	0.	4e-11
Cgdo	単位チャネル幅当たりのゲート・ドレイン容量	F/m	0.	4e-11
Cgbo	単位チャネル幅当たりのゲート・バルク容量	F/m	0.	2e-10
Rsh	ドレイン・ソース拡散シート抵抗	Ohms	0.	10
Cj	単位接合面積当たりのゼロ・バイアス バルク容量	F/m/m	0.	2e-4
Mj	バルク接合傾斜係数	-	0.5	0.5
Cjsw	接合周囲の単位長当たりのゼロ・ バイアス壁面容量	F/m	0.	1e-9
Mjsw	壁面接合傾斜係数 (Level 1)	-	0.50	
	壁面接合傾斜係数 (Level 2, 3)	-	0.33	
Js	単位接合面積当たりのバルク接合飽和電流	A/m	0.	1e-8
Tox	酸化膜厚	m	1e-7	1e-7
Nsub	サブストレート・ドーピング	1/cm/cm/cm	0.	4e15
Nss	表面状態密度	1/cm/cm	0.	1e10
Nfs	高速表面状態密度	1/cm/cm	0.	1e10
TPG	ゲート材料の型 +1 opp. to substrate -1 サブストレートと同じ 0 Al ゲート	-	1.	
XJ	金属接合深さ	m	0.	1 μ
LD	カテラル拡散	m	0.	0.8 μ
UO	表面移動度	cm*cm/V/s	600	700
Ucrit	移動度減衰の境界場 (MOS2 のみ)	V/cm	1E4	1e4
Uexp	移動度減衰指数の境界場 (MOS2 のみ)	-	0.	0.1
UTRA	逆方向場係数 (移動度) (MOS2 では削除)	-	0.	0.3
Vmax	キャリアの最大ドリフト速度	m/s	0.	5e4
Neff	全チャネル電荷 (固定および可動) 係数 (MOS2 のみ)	-	1.	5.0
KF	フリッカ雑音係数	-	0.	1e-26
AF	フリッカ雑音指数	-.	1.	1.2

FC	順方向バイアス空間電荷層の係数	-	0.5	
DELTA	スレッシュホールド電圧上の幅の影響 (MOS2 および MOS3)	-	0.	1.
THETA	移動度変調 (MOS3 のみ)	1/V	0.	0.1
ETA	静的帰還 (MOS3 のみ)	-	0.	1.
KAPPA	飽和場要因 (MOS3 のみ)	-	0.2	0.5
Tnom	パラメータ測定温度	°C	27	50

表 5.18: MOSFET Level 1, 2, 3 および 6 のパラメータ

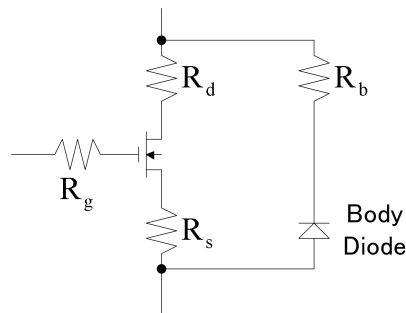


図 5.4 VDMOS

ボード・レベル・スイッチ・モード電力源で一般的に用いられている個別の垂直二重拡散 MOSFET トランジスタ (VDMOS) は、上記のモノリシック MOSFET モデルとは質的に異なった振る舞いをします。特に

1. VDMOS トランジスタに含まれるダイオードは、モノリシック MOSFET のサブストレート・ダイオードとは違う外部端子に接続されています。
2. 非線形ゲート・ドレイン容量 (C_{gd}) は、モノリシック MOSFET モデルの簡単な傾斜型容量でモデル化することは出来ません。

VDMOS トランジスタにおいて、 C_{gd} は、ゼロ電圧ゲート・ドレインのところ急激に変化します。 V_{gd} が負のとき、 C_{gd} は、一つの電極のゲートともう一つの電極としての基板上のドレインとの容量に物理的に基づいています。この容量は、伝導性がない基板の厚みにはあまり依存していません。しかし V_{gd} が正のとき、基板は伝導性となり、 C_{gd} はゲート酸化膜圧の容量に物理的に依存します。

伝統的に巧く作られたサブサーキットが、パワー MOSFET を複製するために用いられてきました。新しい本来の spice 素子が、計算速度を向上するため、取束性の信頼度を向上させるため、モデルの記述を簡単にするために記述されています。直流モデルは、デフォルトで長さと同幅を 1 にする場合を除いてレベル 1 モノリシック MOSFET と同じです。その結果トランスコンダクタンスは、スケーリングしなくても直接規定することが出来ます。交流モデルは、次のようになっています。ゲート・ソース容量は、一定と考えられます。このことはゲート・ソース電圧が負の方向に駆動されていると、パワー MOSFET の良い近似となっていることが経験から分かります。ゲート・ドレイン容量は、次の経験的に観察される形に従います。

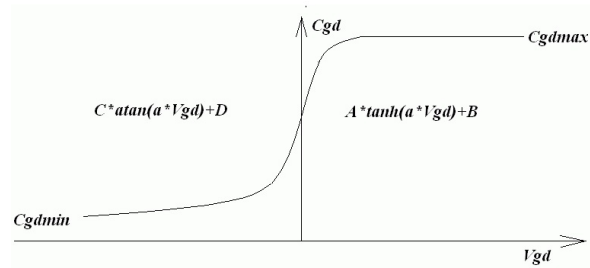


図 5.5 ドレイン・ゲート容量

正の V_{gd} に対して C_{gd} は、 V_{gd} のハイパブリック・タンゼントで変化します。負の V_{gd} に対して C_{gd} は、 V_{gd} のアーク・タンゼントで変化します。モデル・パラメータ a 、 C_{gdmin} 、 C_{gdmax} は、ゲート・ドレイン容量のパラメータです。ソース・ドレイン容量は、ソース・ドレイン端子に接続された基板ダイオードの傾斜容量によって決められ、ソース・ドレイン抵抗の外側にあります。

名称	パラメータ	単位	デフォルト	例
l	長さ	m	1.	2.
w	幅	m	1.	1.
R_g	ゲート抵抗	Ohms	0.	
R_{ds}	ドレイン-ソース・シャント抵抗	Ohms	0.	
V_{TO}	ゼロバイアスレシヨルド電圧	V	0.	1.
K_P	トランスコンダクタンス	A/V	1/	3.
Φ	表面ポテンシャル	V	0.6	0.65
$LAMBDA$	チャンネル長変調	1/V	0.	0.02
C_{bd}	ゼロバイアス B-D 接合容量	F	0.	20f
C_{bs}	ゼロ・バイアス B-S 接合容量	F	0.	20f
R_d	ドレイン抵抗	Ohms	0.	
R_s	ソース抵抗	Ohms	0.	
C_{gs}	ゲート・ソース・オーバーラップ容量	F	0.	4e-11
C_{gdmin}	最少非線形 G-D 容量	F	0.	4e-11
C_{gdmax}	最大非線形 G-D 容量	F	0.	4e-11
a	非線形 C_{gd} 容量パラメータ	1	1.	.5
I_s	基板ダイオード飽和電流	A	1e-14	1e-15
R_b	危難ダイオード抵抗	Ohms	0.	
n	基板ダイオード・エミッション係数	-	1/	
C_{jo}	基板ダイオード接合容量	F	0.	4e-11
V_j	基板ダイオード接合ポテンシャル	V	0.75	
m	基板ダイオード傾斜係数	-	0.5	0.5
F_c	基板ダイオード順方向バイアス接合適合パラメータ	-	0.5	
t_t	基板ダイオード遷移時間	sec	0.	
E_g	I_s の温度効果に対する基板ダイオード活性エネルギー	eV	1.11	
X_{ti}	基板ダイオード飽和電流温度指数	-	3	
$nchan[*]$	n チャンネル VDMOS	-	true	

pchan[*]	P チャンネル VDMOS	-	false	
Tnom	パラメータ測定温度	°C	27	
Kf	フリッカ雑音計数	-	0	
Af	フリッカ雑音指数	-	0	

表 5.19: VDMOS パラメータ

VDMOS という名称は、N チャンネルおよび P チャンネル素子に対して用いられます。デフォルトの極性は、N チャンネルです。P チャンネルとするためには、キーワード pchan にモデルのフラッグを立てなければ行けません。例えば.model xyz VDMOS(Kp = 3 pchan) は、p チャンネル・トランジスタを示します。

5.4.14 損失のある伝送線路

Symbol Names: RES, RES2

Syntax: Oxxx L+ L- R+ R- <model>

例

```
O1 in 0 out 0 MyLossyTline
.model MyLossyTline LTRA(len=1 R=10 L=1u C=10n)
```

これは単一線路の損失がある伝送線路です。N1 および N2 は、端子対 1 のノードです。N3 と N4 は、端子対 2 のノードです。モデル・カードは、この回路素子の電気的特性を定義します。

名称	パラメータ	単位/型	デフォルト
R	単位長当たりの抵抗	Ohm/unit	0.
L	単位長当たりのインダクタンス	Henrys/unit	0.
G	単位長当たりのコンダクタンス	mhos/unit	0.
C	単位長当たりの容量	Farads/unit	0.
LEN	ライン長		none
REL	ブレーキング・ポイントの制御	-	1
ABS	ブレーキング・ポイントの制御		1
NOSTEPLIMIT	線路遅延より少ない時間ステップを禁止	flag	not set
NOCONTROL	複雑な時間ステップを禁止	flag	not set
LININTERP	線形近似を行う	flag	not set
MIXEDINTERP	履歴を圧縮するための特別な reltol	flag	not set
COMPACTABS	履歴を圧縮するための特別な rabstol	flag	RELTOL
TRUNCNR	時間ステップ制御に Newton-Raohson 法を用いる		ABSTOL
TRUNCNONTCUT	インパルス応答のエラーを少なくするため時間ステップを制限しない	flag	not set

表 5.20 損失のある伝送線路のパラメータ

5.4.15 バイポーラ・トランジスタ

Symbol Names: NPN, PNP, NPN2, PNP2

Syntax: Qxxx Collector Base Emitter [Substrate Node] model
+ [area] [off] [IC=<Vbe, Vce>] [temp=<T>]

例

```
Q1 C B E MyNPNmodel
.model MyNPNmodel NPN(Bf=75)
```

バイポーラ・トランジスタは、その特性を規定するためにモデル・カードが必要です。モデル・カードのキー NPN、PNP は、トランジスタの極性を示しています。area ファクターは、指定されたトランジスタが何個並列に接続されているかを示しています。

バイポーラ接合トランジスタ・モデルはガンメルとプーンの積分型電荷制御モデルを採用しています。この変形されたガンメル・プーン・モデルは、オリジナルを拡張し高電圧でのいくつかの効果、準飽和、サブストレート伝導を含んでいます。このモデルは、あるパラメータが規定されていないとき自動的に簡略化されエバース・モル・モデルとなります。直流モデルは、次のパラメータによって定義されます。Is、Bf、Nf、Ise、Ikf、Neなどは、順方向電流利得特性を決めます。Is、Br、Nr、Isc、Ikrなどは、逆方向電流利得特性を決めます。VafやVarは、順方向および逆方向の出力コンダクタンスを決めます。三つの抵抗Rb、RcおよびReが含まれます。ここでRbは、高電流依存性があります。ベース電荷蓄積は、順方向逆遷移時間Tf、方向遷移時間Trによってモデル化されています。順方向遷移時間Tfは、必要ならばバイアス依存性を持たせることが出来ます。非線形空間電荷層容量は、B-E接合ではCje、Vje、Mje、B-C接合ではCjc、Vjc、Mjc、C-S接合ではCjs、Vjs、Mjs、によって決められます。飽和電流Isの温度依存性は、エネルギーギャップEgと飽和電流温度指数XTIによって決められます。加えてベース電流温度依存性は、新しいモデルでベータ温度指数XTBによってモデル化されています。規定される値は、温度TNOMで測定されると規定しています。これは、オプションで決められるか.model上で規定されています。

変形ガンメル・プーンモデルで用いられているBJTパラメータは、次のリストで示してあります。

名称	パラメータ	単位	デフォルト
Is	飽和電流	A	1e-16
Bf	理想最大順方向ベータ	-	100
Nf	順方向エミッション係数	-	1.0
Vaf	順方向アーリー電圧	V	infinite
Ikf	順方向ニー電流	A	infinite
NE	B-E リーク・エミッション係数	A	1.5
Br	理想最大逆ベータ	-	1
Nr	逆電流エミッション係数	-	1
Var	逆方向アーリー電圧	-	infinite

Ikr	逆方向ベータが落ち始める電流値	V	infinite
Isc	B-C リーク飽和電流	A	0
Nc	B-C リーク・エミッション係数	A	2
Rb	ゼロバイアスペース抵抗	-	0
Irb	逆方向ニー電流	Ohm	infinite
Rbm	高電流での最少ベース抵抗	A	RB
Re	エミッタ容量	Ohm	0
Rc	コレクタ抵抗	Ohm	0
Cje	B-E ゼロバイア空間電荷層容量	Ohm	0
Vje	B-E ビルトイン・ポテンシャル	F	0.75
Mje	B-E 接合ポテンシャル要素	V	0.33
Tf	理想順方向遷移時間	sec	0
Xtf	TF のバイアス依存性係数	-	0
Vtf	TF の VBC 依存性	V	infinite
Itf	TF の高電流パラメータ	A	0
PTF	freq=1.0/(TF*2PI) Hz での過剰位相	deg	0
Cjc	B-C ゼロバイア空間電荷層容量	F	0
Vjc	B-C ビルトイン・ポテンシャル	V	0.75
Mjc	B-C 接合指数要素	-	0.33
Xcjc	内部ベースノード接続 B-C 空間容量要素	-	1
Tr	理想逆遷移時間	sec	0
Cjs	ゼロ・バイアス C-S 容量	F	0
Vjs	サブストレート接合ビルトイン・ポテンシャル	V	0.75
Mjs	サブストレート接合指数要素	-	0
Xtb	順逆ベータ温度指数	-	0
Eg	Is の影響によるエネルギーギャップ	eV	1.11
Xti	Is の影響による温度指数	-	3
Kf	フリッカ雑音係数	-	0
Af	フリッカ雑音指数	-	1
Fc	順バイアス空間容量式の係数	-	0.5
Tnom	パラメータ測定温度	°C	27
cn	ホール移動度の準飽和温度係数		2.42 NPN 2.2 PNP
d	散乱飽和速度の準飽和温度係数		.87 NPN .52 PNP
gamma	エピタキシャル領域ドーピング要素		1e-11
qco	エピタキシャル領域電荷要素	Cou1	0
quasimod	温度依存性の準飽和フラッグ	-	0
rco	エピタキシャル領域抵抗	Ohm	0
vg	0K での準飽和推定バンドギャップ電圧	V	1.206
vo	キャリア移動度ニー電圧	V	10.
Tre1	Re 第 1 項温度係数	/°C	0
Tre2	Re 第 2 項温度係数	/°C/°C	0
Trb1	Rb 第 1 項温度係数	/°C	0
Trb2	Rb 第 2 項温度係数	/°C/°C	0
Trc1	Rc 第 1 項温度係数	/°C	0
Trc2	Rc 第 2 項温度係数	/°C/°C	0
Trm1	Rmb 第 1 項温度係数	/°C	0
Trm2	Rmb 第 2 項温度係数	/°C/°C	0

Iss	サブストレート接合飽和電流	A	0
Ns	サブストレート接合エミッション係数	-	1

表 5.21: 変形ガンメル・プーン BJT パラメータ

モデル・パラメータ level は、LTspice のその他の BJT と区別するために用いられています。Dr. -Ing. Dietmar Warning of DAnalyse GmbH, Berlin, Germany の寛大な貢献により、LTspice は VBIC を含んでいます。このデバイスを用いるには、Level=9 と設定します。Level=4 は、Level=9 と同じです。次の資料は、Dr. Warning から提供されています。

VBIC - Vertical Bipolar Inter Company model

VびCモデルは、現代の半導体技術における集積バイポーラ・トランジスタに注目した標準のガンメル・プーン (SGP) モデルを拡張したモデルです。Kull や Nagel によって研究された準飽和モデルを含み、スイッチング・トランジスタの特別な出力特性をモデル化することができます。シリコン、SiGe そして III-V HBT 素子の SGP モデルに広く用いられています。

VBIC の優位性が、標準のガンメル・プーン・モデルと比較されています。

- 集積回路の寄生素子であるサブストレート・トランジスタが組み込まれています。
- 弱い雪崩現象とベース・エミッタ・ブレイクダウン・モデル
- アーリー効果の改善
- I_c と I_b の物理的な分離
- 改良された空間電荷容量モデル
- 自己加熱モデル (このバージョンでは含まれていない)

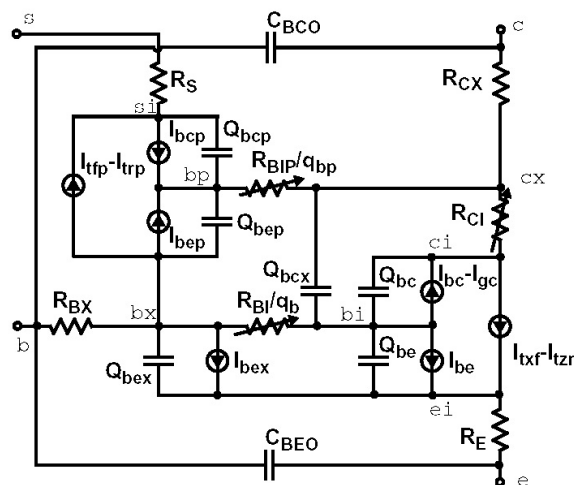


図 5.6 モデルの構造

パラメータ

VBIC は、SGP モデルに基づいていますので、SGP パラメータから始めていくつかの変換を行うことができます。次のパラメータは、VBIC version 1.2 です。これらは、過剰位相回路および自己加熱効果を含んでいませんが、LTspice に組み込まれています。SGP から BVIC へ切り替えるには、Level を 9 に設定しなければ行けません。

名称	パラメータ	単位	デフォルト
tnom	パラメータ測定温度	deg C	27.0
rcx	外部コレクタ抵抗	Ω	0.1
rci	コレクタ抵抗	Ω	0.1
vo	エビ・ドリフト飽和電圧	V	8
gamm	エビ・ドーピング・パラメータ		0.0
hrcf	高電流 RC 要素		8
rbx	外部ベース抵抗	Ω	0.1
rbi	ベース抵抗	Ω	0.1
re	エミッタ抵抗	Ω	0.1
rs	サブストレート抵抗	Ω	0.1
rbp	寄生ベース抵抗	Ω	0.1
is	転送飽和電流	A	1e-16
nf	順方向エミッション係数		1.0
nr	逆方向エミッション係数		1.0
fc	順方向バイアス空間電荷容量制限		0.9
cbeo	外部 B-E オーバーラップ容量	F	0.0
cje	ゼロバイアス B-E 空間電荷容量	F	0.0
pe	B-E ビルトインポテンシャル	V	0.75
me	B-E 接合傾斜係数		0.33
aje	B-E 容量滑らかさ要素		-0.6
cbco	外部 B-C オーバーラップ容量	F	0.0
cjc	ゼロバイアス B-C 空間電荷容量	F	0.0
qco	エビ電荷パラメータ	C	0.0
cjep	B-C 外部ゼロバイアス容量	F	0.0
pc	B-C ビルトインポテンシャル	V	0.75
mc	B-C 接合傾斜係数		0.33
ajc	B-C 容量滑らかさ要素		-0.5
cjcp	ゼロバイアス S-C 容量	F	0.0
ps	S-C 接合ビルトインポテンシャル	V	0.75
ms	S-C 接合傾斜係数		0.33
ajs	S-C 容量滑らかさ要素		-0.5
ibei	理想 B-E 飽和電流	A	1e-18
wbe	Vbei から IBEI、Vbex から I-WBE の部分		1.0
nei	理想 B-E エミッション係数		1.0
iben	非理想 B-C 飽和電流	A	0.0
nen	非理想 B-C エミッション係数		2.0
ibci	理想 B-C 飽和電流	A	1e-16
nci	理想 B-C エミッション係数		1.0
ibcn	非理想 B-C 飽和電流	A	0.0
ncn	非理想 B-C エミッション係数 2		1.0

avc1	B-C 弱アバランシェ・パラメータ 1	1/V	0.0
avc2	B-C 弱アバランシェ・パラメータ 2	1/V	0.0
isp	寄生転送飽和電流	A	0.0
wsp	ICCP の部分		1.0
nfp	寄生順方向エミッション係数		1.0
ibeip	理想寄生 B-E 飽和電流	A	0.0
ibenp	非理想寄生 B-E 飽和電流	A	0.0
ibcip	理想寄生 B-C 飽和電流	A	0.0
ncip	非理想寄生 B-C エミッション係数		1.0
ibcnp	非理想寄生 B-C 飽和電流	A	0.0
ncnp	非理想寄生 B-C エミッション係数		2.0
vef	順方向アーリー電圧		8
ver	逆方向アーリー電圧		8
ikf	順方向ニー電流	A	8
ikr	逆方向ニー電流	A	8
ikp	寄生ニー電流	A	0
tf	理想順方向転移時間	sec	0.0
qtf	ベース幅変調に関する TF 変化		0.0
xtf	TF のバイアス依存係数		0.0
vtf	TF の VBC 電圧依存	V	8
itf	TF の高電流依存	A	8
tr	理想逆転移時間	sec	0.0
td	順方向過剰位相遅延時間	sec	0.0
kfn	B-E フリッカ雑音係数		0.0
afn	B-E フリッカ雑音指数		1.0
bfn	B-E フリッカ雑音 1/f 依存性		1.0
xre	RE の温度指数		0.0
xrbi	RBI の温度指数		0.0
xrci	RCI の温度指数		0.0
xrs	RS の温度指数		0.0
xvo	VO の温度指数		0.0
ea	IS の活性エネルギー	V	1.12
eaie	IBEI の活性エネルギー	V	1.12
eaic	IBCI/IBEIP の活性エネルギー	V	1.12
eais	IBCIP の活性エネルギー	V	1.12
eane	IBEN の活性エネルギー	V	1.12
eanc	IBCN/IBENP の活性エネルギー	V	1.12
eans	IBCNP の活性エネルギー	V	1.12
xis	IS の温度指数		3.0
xii	IBEI、IBCI、IBEIP、IBCIP の温度指数		3.0
xin	IBEN、IBCN、IBENP、IBCNP の温度指数		3.0
tnf	NF の温度指数		0.0
tavc	AVC2 の温度指数		0.0
rth	温度抵抗	K/W	0.0
cth	温度容量	Ws/K	0.0
vrt	内部 B-C 接合パンチスルー電圧	V	0.0
art	滑らかさパラメータ		0.1
ccso	固定 C-S 容量	F	0.0

qbm	SGP qb 式 of 選択		0.0
nkf	高電流ベータ。ロールオフ		0.5
xikf	IKF 温度指数		0.0
xrcx	RCX 温度指数		0.0
xrbx	RBX 温度指数		0.0
xrbp	RBP 温度指数		0.0
isrr	順逆方向分離 IS		1.0
xisr	ISR 温度指数		0.0
dear	ISRR デルタ活性エネルギー		0.0
eap	ISP の励起エネルギー		1.12
vbbe	B-E ブレークダウン電圧	V	0.0
nbbe	B-E ブレークダウン・エミッション係数		1.0
ibbe	B-E ブレークダウン電流		1e-06
tvbbe1	VBBE の線形温度係数		0.0
tvbbe2	VBBE の 2 次温度係数		0.0
tnbbe	NBBE の温度係数		0.0
ebbe	$\exp(-VBBE/(NBBE*Vtv))$		0.0
dtemp	ローカル温度差	dec	0.0
vers	リビジョン・バージョン		1.2
vref	レファレンス・バージョン		0.0

表 5.22: VBIC パラメータ

参照

C. C. McAndrew et al., "Vertical Bipolar Inter Company 1995: An Improved Vertical, IC Bipolar Transistor Model", Proceedings of the IEEE Bipolar Circuits and Technology Meeting, pp. 170 - 177, 1995

C. C. McAndrew et.al., VBIC95, "The Vertical Bipolar Inter-Company Model", IEEE Journal of Solid State Circuits, vol. 31, No. 10, October 1996

C. C. McAndrew, VBIC Model Definition, Release 1.2, 18. Sep. 1999

5.4.16 抵抗

Symbol Names: RES, RES2

Syntax: Rxxx n1 n2 <value> [tc=tc1, tc2, ...]

抵抗は、ノード n1、n2 に簡単な線形抵抗を供給します。温度依存性は、パラメータ tc で各々の抵抗に対して定義することが出来ます。抵抗 R は、次のように記述されます。

$$R = R_0 * (1. + dt * tc1 + dt**2 * tc2 + dt**3 * tc3 + \dots)$$

ここで R_0 は、通常の温度での抵抗です。 dt は、抵抗の温度と通常温度との差です。

5.4.17 電圧制御スイッチ

Symbol Names: SW

Syntax: Sxxx n1 n2 nc+ nc- <model> [on,off]

例

```
S1 out 0 in 0 MySwitch
```

```
.model MySwitch SW(Ron=.1 Roff=1Meg Vt=0 Vh=-.5 Lser=10n Vser=.6)
```

ノード $nc+$ とノード $nc-$ の間の電圧は、ノード $n1$ とノード $n2$ の間のスイッチインピーダンスを制御します。モデル・カードは、スイッチの振る舞いを定義するために必要です。回路ファイル./examples/Educational/Vswitch.asc を見て下さい。これは、spice コマンドとして回路図上に直接置くことの出来るモデル・カードの例を示しています。

名称	パラメータ	単位	デフォルト
Vt	スレッシュホールド電圧	Volts	0.
Vh	ヒステレシス電圧	Volts	0.
Ron	オン抵抗	Ohms	1.
Roff	オフ抵抗	Ohms	1/gmin
Lser	直列インダクタンス	Henry	0.
Vser	直列電圧	Volts	0.
ilimit	電流制限	Amps	infin

表 5.23 電圧制御スイッチ・モデル・パラメータ

スイッチは、電圧制御による三つのモードがあります。この電圧とは、ヒステレシス電圧 V_h の値に依存しています。 V_h がゼロの場合、入力電圧がスレッシュホールドの値より高いかどうかによって、いつでもオンまたはオフとなります。 V_h が正の場合、 $V_t - V_h$ および $V_t + V_h$ の点でシュミット・トリガによって制御されているかのようにスイッチはヒステレシスとなります。 V_h が負の場合には、スイッチは滑らかにオンとオフのインピーダンスの間を移動します。移動は、 $V_t - V_h$ と $V_t + V_h$ 制御電圧の間で起こります。滑らかな変化は、スイッチ伝導の対数に一致した低い次数に従います。

5.4.18 無損失伝送線路

Symbol Name: TLINE

Syntax: Txxx L+ L- R+ R- Zo=<value> Td=<value>

$L+$ および $L-$ は、一端子網のノードです。 $R+$ と $R-$ は、もう一方の端子網のノードです。 Z_o は、特性インピーダンスです。線路の長さは、伝送遅延 T_d によって与えられます。

この素子は、伝送モードのみをモデル化しています。全ての四つのノードが実際の回路で区別されているとすれば、二つのモードが励起されます。このような状況をシミュレートするには、二つの伝送線路が必要となります。

5.4.19 均一 RC 線路

Symbol Names: URC

Syntax: Uxxx N1 N2 Ncom <model> L=<len> [N=<lumps>]

N1 と N2 は、RC 線路が接続している二つの素子ノードです。一方 Mcom は、容量が接続されているノードです。MNAME は、モデル名で、LEN は、RC 線路をメータで表示する長さです。もし規定するつもりであるならば、Lumps は、RC 線路をモデル化するために用いる集中素子の数を示しています。用いられる適当な集中定数素子の数は、規定することが出来ます。

URC モデルは、L. Gertzberg in 1974 によって提案されたモデルから導かれています。このモデルは、内部によって生成されているノードによって集中定数 RC 素子の回路網となる URC 線路のサブサーキット・タイプの拡張によって実現されています。この RC 素子は、URC 線路の中心に向かって増加するように幾何学的に拡張され、比例定数 K を持っています。

URC 線路は、ISPERL パラメータが非ゼロの値に設定されていなければ、厳密に抵抗と容量とで構成されています。この場合容量は、ゼロ・バイアス接合容量の値を持った逆向きダイオードの容量で置き換えられます。それと単位伝送線路当たり ISPERL amps の飽和電流とオプションの単位長さ当たり RSPERL ohms に等しい直列抵抗で置き換えられます。

名称	パラメータ	単位	デフォルト
K	伝搬定数	none	2.
FMAX	取り扱う最大周波数	Hz	1 G
RPERL	単位長の抵抗	Ohm/m	1000
CPERL	単位長の容量	F/m	1e-15
ISPERL	単位長当たりの飽和電流	A/m	0
RSPERL	単位長当たりのダイオード抵抗	Ohm/m	0

表 5.24

5.4.20 電圧源

Symbol Names: VOLTAGE, BATTERY

Syntax: Vxxx n+ n- <voltage> [AC=<amplitude>]
+ [Rser=<value>] [Cpar=<value>]

この素子は、ノード n+ と n- に一定の電圧を供給します。交流解析に対して AC の値は、解析周波数での電源の大きさとして用いられます。直列抵抗と並列抵抗が、定義でき

ます。等価回路は、次の図として与えられます。

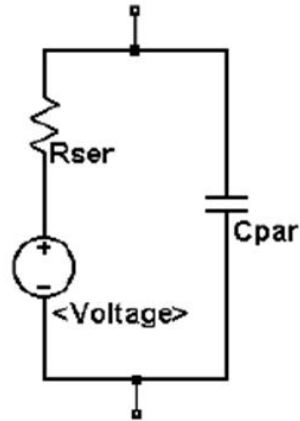


図 5.7 電圧源等価回路

電圧源は、SPICE で歴史的に電流メータとして、電流制御素子の電流センサとして使われていました。Rser が規定されていると、電圧源は F、H あるいは W 素子の感応素子として用いることは出来ません。しかし任意の回路素子の電流は、電圧源を含んで描画することが出来ます。

Syntax: Vxxx n+ n- PULSE(V1 V2 Tdelay Trise Tfall Ton Tperiod Ncycles)

名称	パラメータ	単位
Voff	初期値	Volts
Von	パルス値	Volts
Tdelay	遅延	seconds
Tr	立ち上がり時間	seconds
Tf	立ち下がり時間	seconds
Ton	オン時間	seconds
Tperiod	期間	seconds
Ncycles	サイクル波の数	cycles

表 5.25 時間依存性パルス電圧源

Syntax: Vxxx n+ n- SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

名称	パラメータ	単位
Voffset	直流オフセット	Volts
Vamp	振幅	Volts
Freq	周波数	Hz
Td	遅延	seconds
Theta	ダンピング・ファクタ	1/seconds
Phi	位相	degrees
Ncycles	サイクル波の数	cycles

表 5.26 時間依存性正弦波電圧源

Td より短い時間あるいは Ncycles が完了した後に対して、走ります。出力電圧は、次のように与えられます。

$$V_{\text{offset}} + V_{\text{amp}} \cdot \sin(\pi \cdot \Phi / 180)$$

その他電圧は、次のようになります。

$$V_{\text{offset}} + V_{\text{amp}} \cdot \exp(-(time - Td) \cdot \Theta) \cdot \sin(2 \cdot \pi \cdot \text{Freq} \cdot (time - Td) + \pi \cdot \Phi / 180)$$

ダンピング・ファクタ Θ は、遅延時間定数の逆数です。

Syntax: Vxxx n+ n- EXP(V1 V2 Td1 Tau1 Td2 Tau2)

名称	パラメータ	単位
V1	初期値	Volts
V2	パルス値	Volts
Td1	立ち上がり遅延時間	seconds
Tau1	立ち上がり時定数	seconds
Td2	立ち下がり遅延時間	seconds
Tau2	立ち下がり時定数	seconds

表 5.27 時間依存性指数電圧源

Td1 より少ない時間に対して、出力電圧は、V1 です。Td1 と Td2 の時間の中で、電圧は次のように与えられます。

$$V1 + (V2 - V1) \cdot (1 - \exp(-(time - Td1) / \text{Tau1}))$$

Td2 の後の時間では、電圧は次のようになります。

$$V1 + (V2 - V1) \cdot (1 - \exp(-(time - Td1) / \text{Tau1})) + (V1 - V2) \cdot (1 - \exp(-(time - Td2) / \text{Tau2}))$$

Syntax: Vxxx n+ n- SFFM(Voff Vamp Fcar MDI Fsig)

名称	パラメータ	単位
Voff	直流オフセット	Volts
Vamp	振幅	Volts
Fcar	搬送波周波数	Hz
MDI	変調度	none
Fsig	周波数	Hz

表 5.28 時間依存性単一周波数 FM 電圧源

電圧は、次のように与えられます。

$$V_{off} + V_{amp} \sin((2 \cdot \pi \cdot F_{car} \cdot \text{time}) + MDI \cdot \sin(2 \cdot \pi \cdot F_{sig} \cdot \text{time}))$$

Syntax: Vxxx n+ n- PWL(t1 v1 t2 v2 t3 v3...)

任意のピース・ワイズ線形電圧源

T1 より前の時間に対して、電圧は V1 です。時間 T1 と T2 の間では、電圧は V1 と V2 の間を線形に変化します。任意の数の時間と電圧をとることが出来ます。最後の指定時間の後、電圧は最後の時間のままです。

Syntax: Vxxx n+ n- wavefile=<filename> [chan=<nnn>]

この記述は、.wave ファイルに対して LTspice への入力として用いることが出来るようにします。.wave ファイルへの絶対パスあるいは相対パスは、シミュレーション回路図あるいはネットリストに含まれるディレクトリから計算されます。ダブル・クォーテーションは、空白を含むパスを指定するために用いられます。

.wave ファイルは、65536 チャンネルまで、0 から 65535 までの数値を含むことが出来ます。Chan は、どのチャンネルが用いられるか設定するために用いられます。デフォルトは、最初のチャンネル 0 です。.wave ファイルは、フルスケール範囲 -1 V から 1 V までになるように変換されます。

この電源は、過渡解析だけで意味があります。

5.4.21 電流制御スイッチ

Symbol Names: CSW

Syntax: Wxxx n1 n2 Vnam <model> [on,off]

例

```
W1 out 0 Vsense MySwitch
```

```
Vsense a b 0.
```

```
.model MySwitch CSW(Ron=.1 Roff=1Meg It=0 Ih=-.5)
```

名前が付けられた電圧源を流れる電流は、スイッチのインピーダンスを制御します。モデル・カードは、電流制御スイッチの振る舞いを定義するために必要です。

名称	パラメータ	単位	デフォルト
It	スレッシュヨルド電流	Volts	0.
Ih	ヒステレシス電流	Volts	0.
Ron	オン抵抗	Ohms	1.
Roff	オフ抵抗	Ohms	1/gmin

表 5.29 電流制御スイッチ・モデル・パラメータ

このスイッチは、電流制御、ヒステレシス電流の値依存性、Ih の三つの違うモードがあります。Ih がゼロの場合、スイッチは制御電流がスレッシュヨルドを越えるかどうかによってオンまたはオフになります。Ih が正の場合、スイッチは It-Ih または It+Ih の電流点を持つヒステレシスを示します。Ih が負の場合、スイッチはオンとオフのインピーダンスの間を滑らかに変化します。変化は、It-Ih と It+Ih の制御電流の値で起こります。滑らかな変化は、スイッチ・コンダクタンスの対数に適合した低次元の多項式となります。

5.4.22 サブサーキット

Symbol Names:

```

LT1001 LT1006 LT1028 LT1070 LT1071
LT1072 LT1074 LT1109-12 LT1109-5 LT1109
LT1109A-12 LT1109A-5 LT1109A LT1112 LT1124
LT1169 LT1170 LT1170HV LT1171 LT1171HV
LT1172 LT1172HV LT1211 LT1227 LT1300
LT1301 LT1302-5 LT1302 LT1303-5 LT1303
LT1304-3.3 LT1304-5 LT1304 LT1305 LT1307
LT1307b LT1309 LT1316 LT1317 LT1317b

LT1351 LT1371 LT1372 LT1375-5 LT1375
LT1376-5 LT1376 LT1377 LT1413 LT146
LT1490 LT1498 LT1533 LT1610 LT1611
LT1613 LT1614 LT1627 LT1676 LT1735
LTC1047 LTC1174-3.3 LTC1174-5 LTC1174 LTC1174HV-3.3
LTC1174HV-5 LTC1174HV LTC1625 LTC1628 LTC1736

```

(and others)

Syntax: `Xxxx n1 n2 n3... <subckt name> [<parameter>=<expression>]`

サブサーキットは、名前によって後から呼び出されるため、ライブラリに定義され保存された回路です。下記は、電圧分割回路として定義され呼び出される回路で、回路の中で呼び出します。

```

* calling a subcircuit
*
* This is the circuit

```

```
X1 in out 0 divider top=9K bot=1K
V1 in 0 pulse(0 1 0 .5m .5m 0 1m)
```

```
* This is the subcircuit
.subckt divider A B C
R1 A B {top}
R2 B C {bot}
.ends divider
.tran 3m
.end
```

5.4.23 MESFET トランジスタ

Symbol Names: MESFET

Syntax: Zxxx D G S model [area] [off] [IC=<Vds, Vgs>]

MESFET トランジスタは、その特性を規定するためのモデル・カードが必要です。モデル・カードのキーワード NMF と PMF は、トランジスタの極性を規定します。MESFET モデルは、H. Statz et al., GaAs FET Device and Circuit Simulation in SPICE, IEEE Transactions on Electron Devices, V34, Number 2, February, 1987 pp160-169 に記載されている GaAs FET モデルを用いています。

二つの抵抗 R_d と R_s が、含まれています。電荷蓄積は、ゲート・ドレインとゲート・ソースの関数として全ゲート電荷によってモデル化されています。それは、パラメータ CGS、CGD と PB によって定義されています。

名称	パラメータ	単位	デフォルト
VTO	ピンチオフ電圧	V	-2.
BETA	トランスコンダクタンス	A/V/V	1e-4
B	ドーピング拡張パラメータ	1/V	.3
ALPHA	飽和電圧パラメータ	1/V	2
LAMBDA	チャンネル長変調	1/V	0
RD	ドレイン抵抗	Ohms	0
RS	ソース抵抗	Ohms	0
CGS	ゼロバイアス G-S 接合容量	F	0
CGD	ゼロバイアス G-D 接合容量	F	0
PB	ゲート接合ポテンシャル	V	1
KF	フリッカ雑音係数	none	0
AF	フリッカ雑音指数	none	1
FC	順バイアス空間電荷係数	none	.5

表 5.30 MESFET パラメータ

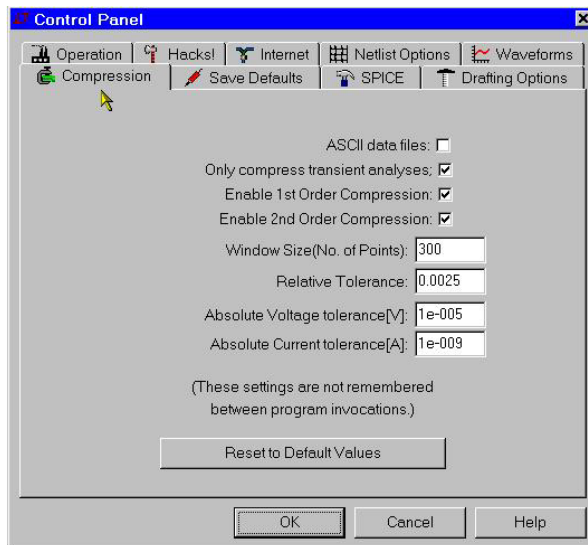
第 6 章

コントロール・パネル

6.1 コントロール・パネルの起動

コントロール・パネルを起動するには、メニュー・コマンド Tools=>Control Panel を使います。そこで LTspice/SwitcherCAD III の多くの点を変更できます。

6.2 圧縮



LTspice は、再生できるように生のデータファイルを圧縮します。圧縮されたファイルは、圧縮前のファイルに較べ 5 0 分の一にすることが出来ます。この圧縮は、削減される圧縮です。コントロールパネルのこれについての窓は、どの様に削減して圧縮するかを制御します。

Window Size(No. of Points): 二つの終端点へ圧縮する点の最大数

Relative Tolerance: 圧縮データと非圧縮データで許容される相対誤差

Absolute Voltage tolerance[V :] 圧縮アルゴリズムで許容される電圧誤差

Absolute Current tolerance[A :] 圧縮アルゴリズムで許される電流誤差

これらの圧縮に関する設定は、デフォルトの設定が用いられるため、プログラムを起動し直すと再現されません。これらは、修復目的のコントロール・ウィンドウ上で有効です。許容とウィンドウのサイズは、回路図上の spice コマンドとして置かれている.option 記述の中のオプションパラメータ plotreltol、plotvntol、plotabstol、plotwinsize で設定できます。

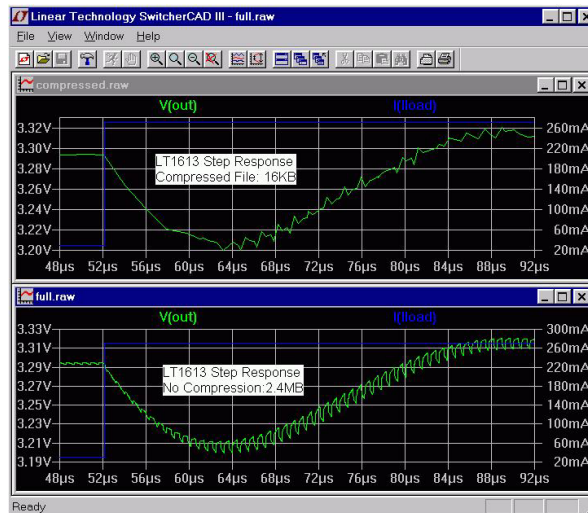


図 6.1 File Size Vs Fidelity study

6.3 機能

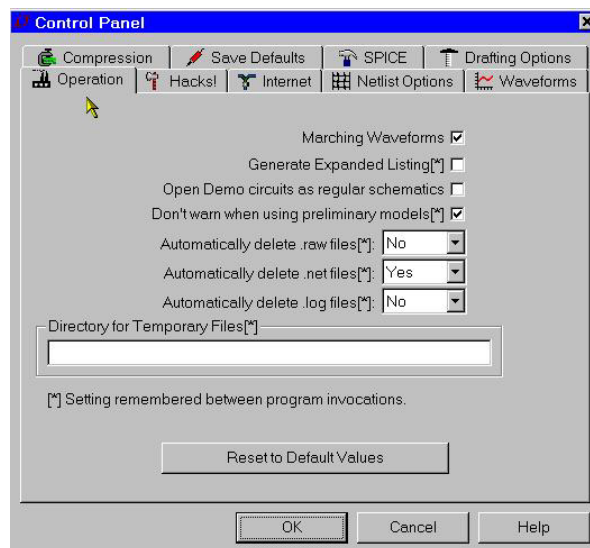


図 6.2 機能パネル

アスタリスクが付いている設定は、プログラムが起動している間記憶されています。

Marching Waveforms シミュレーションが進行しているとき、進行するたびに結果を表示することが出来ます。

Generate Expanded Listing サブサーキットを展開した後、フラットなネットリストを呼び出します。

Open Demo circuits as regular schematics /SwCADIII/lib/app/*.app のデモ回路を開くため [File][Open] を用います。全ての SPICE コマンドは、可視状態となります。回路図は、編集できるようになり、新しいファイルとして保存できます。ダブル・ドット.. は、デモ回路表示制御用途です。一つのドットは、編集のために必要です。

Don't warn when using preliminary models 以前の全てのモデルに対して警告メッセージを出しません。全ての SMPS モデルは、認められないとして予備のモデルと同じように警告されます。

Automatically delete .raw files シミュレーションを閉じた後に自動的に波形データ・ファイルを消去します。これは LTspice によって用いられたディスク・スペースを劇的に減少させます。しかしシミュレーションを再開するとき、もう一度走らせる必要があります。

Automatically delete .net files 回路図が閉じられるときに、自動的に回路図ネットリストを消去します。これらのファイルは、小さなテンポラリ・ファイルと見なされディレクトリが閉じられるときに消去されます。これらは、LTspice シミュレータに対し電氣的接続を定義します。さらにそれらの回路を用いたいと思っている人は、それらを消去したがりません。

Automatically delete .log files シミュレーションが閉じられるときに、自動的にシミュレーション・ログを消去します。これらのファイルは、様々なシミュレーションの計算時間などの統計データや step/.temp/.dc 解析で用いられたステップ・パラメータなどが含まれています。

Directory for Temporary Files 波形の一時的な蓄積データやアップデータファイルなどのディレクトリです。

6.3.1 デフォルトの保存

これらの設定は、どのノードがシミュレーションで保存されたかを明示的に示さないときに用いられます。便利な設定は、"Save Device Currents"、"Save Subcircuit Node Voltages" および "Save Subcircuit Device Currents" です。デバイス電圧や内部素子電圧は、内部プログラム生成において役立ちます。

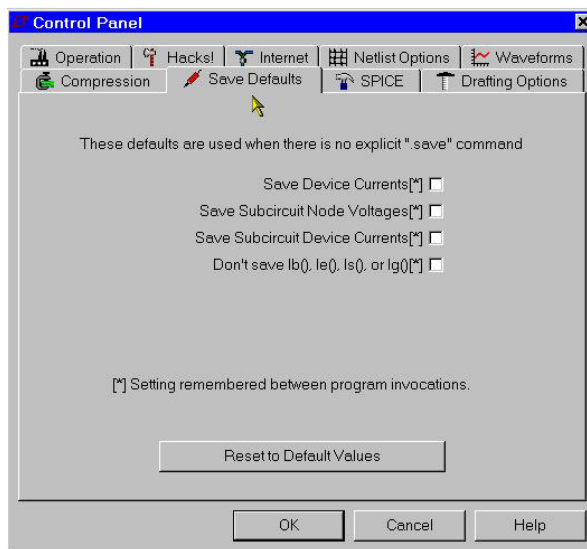


図 6.3 デフォルト設定

Save Device Currents これをチェックしますと、素子および端子の電流を描くことが出来ます。

Save Subcircuit Node Voltages 階層設計で電圧を描くときにはチェックする必要があります。

Save Subcircuit Device Currents これは出力.data ファイルの大きさを減少必要があるとき、トランジスタのコレクタ（ドレイン）電流のみを保存します。この機能は IC 設計において役に立ちます。しかしトランジスタの消費電力を計算するのに十分なデータがないときの手段として用います。

Don't save IB(), Ie(), Is(), Ig() これをチェックしますと、素子および端子電流を描くことが出来ます。消費電力を描くためにも用いられます。

6.3.2 SPICE

この窓は、LTspice の様々なデフォルトを定義できます。これのデフォルトは、シミュレーションのオプションを規定することによって、任意のシミュレーションにおいて上書きされます。通常これらのオプションをそのままの状態ではおいておくことが出来ます。もしウェブ上でしばしばプログラムをアップデートするなら、現状推薦の設定へリセットするため **Reset to Default Values** を押す必要があります。

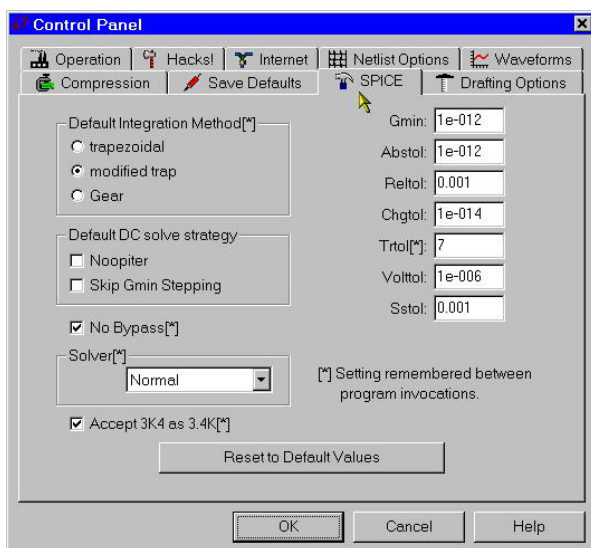


図 6.4 SPICE

変更したいと思う一つのデフォルトは、Trtol でしょう。ほとんどの市販 SPICE プログラムのデフォルトは、この値を 7 としています。LTspice でこのデフォルトは、1 です。その結果 MSPS マクロを用いたシミュレーションは、その波形でのシミュレーション結果を示すよりも少ない結果となります。シミュレーションに直接影響するよりも、Trtol は時間ステップにより多くの影響を与えます。トランジスタ・レベルでのシミュレーションに対して、1 より大きな値は、通常良い結果を全体の解に与えます。シミュレーション精度に影響を与えることなく Trtol を増加したいのであれば、速度が 2x となること分かるでしょう。Trol は、プログラムを起動する間記憶しています。しかし通常の SPICE 許容パラメータ gmin、abstol、reltol、chgtol、vntol のほとんどは、プログラム起動の間記憶していません。もしデフォルト値とは違うパラメータを用いたいのであれば、用いたい値を規定するため、option を書かねばなりませんし、回路図上にそれを置く必要があります。あるいはファイル.inc を設定する必要があります。

またどの解析器が用いられるかと言うことに興味があります。LTspice は、SPICE の二つの完成されたバージョンが含まれています。一つは、通常の解析器で、もう一つは変形された解析器です。変形された解析器は、丸め誤差を少なくする異なったスパース行列パッケージを用いています。特に変形された解析器は、通常の解析器より半分の速度でシミュレーションしますが、千倍の内部精度を持っています。これは、役に立つ手法となります。解析器が用いている特別な option はありません。選択はネットリストが、展開される前に行われなければなりません。というのも二つの解析器は、別の構文解析を行うからです。

LTspice が 4K99 を 4.99K として認識するように Accept 3K4 as 3.4K にチェックを入れて下さい。通常の SPICE は、このことを認識しません。しかし LTspice は、一般的な要求として認識が可能です。

6.3.3 ネットリスト・オプション

Convert μ to u 全ての μ を u へ変換します。餅ウィンドウがギリシャ語の μ (例えば中国語のウィンドウズ版では、デフォルトのフォントで表示できません) を表示することが出来ません。1e-6 倍として μ 文字を認識しない SPICE シミュレーターのネットリストを生成します。

Reverse comp. order 回路素子は、通常順番にネットリストに変換され回路図に付け加えられます。このボックスをチェックすると、この順番が逆になります。

Default Device ダイオードが LTSPICE 回路図に用いられているときはいつでもデフォルト・モデル記述 `.model D D` は、ネットリストに付け加えられ、デフォルト・モデルを用いることについてのメッセージを抑圧します。このオプションのチェックをはずすことによりバイポーラ、MOSFET および JFET トランジスタと同様にこの記述を含まないようにします。

Default Devices ダイオードが LTspice 回路図において用いられるときはいつでも、デフォルト・ライブラリ `standard.dio` は、`.lib` 記述によってシミュレーションに含まれます。このチェックをはずしますと、バイポーラ、MOSFET および JFET トランジスタのアナログ・ライブラリと同様にこのライブラリを含まなくなります。

Convergence Aids 内部プログラムに対してのみの用法

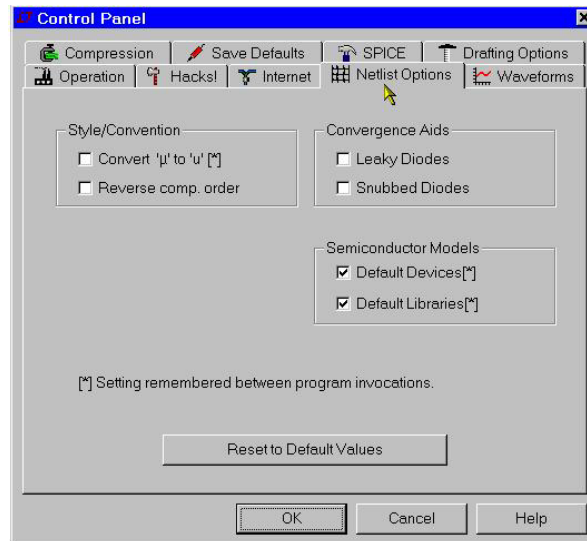


図 6.5 ネットリスト・オプション

6.3.4 ハック

この窓は、内部プログラム展開のために用いられます。しかし現在ではほとんど使われていません。

通常このままの状態ですべてのオプションをオフにすることが出来ます。ウェブ上でアップデートを度々行うようでしたら、現状の推奨設定をリセットするために `Reset to Default Values` を押すこ

とができます。

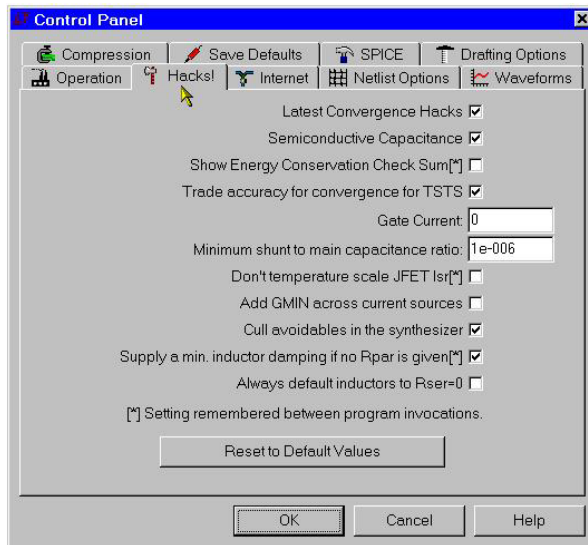


図 6.6 Hacks

6.3.5 描画オプション

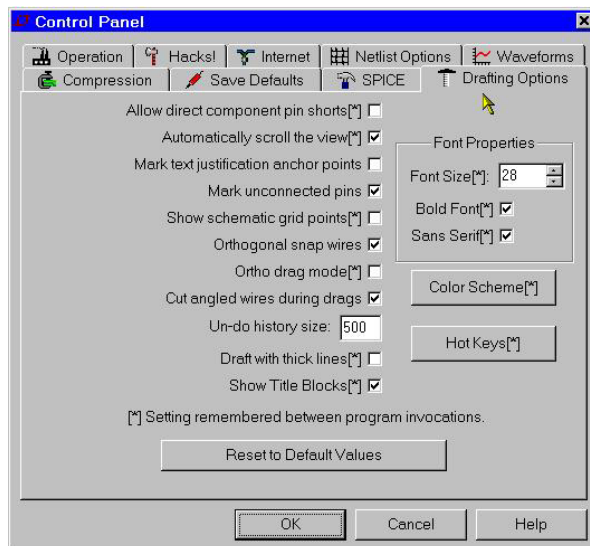


図 6.7 描画オプション

Allow direct component pin shorts 通常素子に直接ワイヤを描くことができます。ピンを短絡している部分のワイヤを消去します。これをチェックしますと短絡したワイヤは、自動的に消去されることはありません。

Automatically scroll the view このボックスをチェックしますと回路図スクロールを見えるようにし、回路図を編集する間、端を閉じます。

- Mark text Justification anchor points テキスト・ブロックの参照点を示すための小さい円を描きます。
- Mark unconnected pins 接続されていないことを示すため接続されていないピンに小さな四角を表示します。
- Show schematic grid points 可視グリッドを開始します。
- Orthogonal snap wires Force wires が、描いている間に垂直および水平に描かれます。チェックをしなければワイヤは任意の角度で描くことが出来任意のグリッドにスナップします。コントロール・キーを押さえますと、ワイヤを描いている間現在の設定を繰り返します。
- Cut angled wires during drags 度絡・コマンドの間直交していない線がワイヤの間にそってクリックすると二つの接続されたワイヤを切断します。
- Undo history size undo/redo バッファの大きさを設定します。
- Draft with thick lines 全てのライン幅を広げる。出版するイメージを作成するのに役立ちます。
- Show Title Block 内部の用法

6.3.6 インターネット・オプション

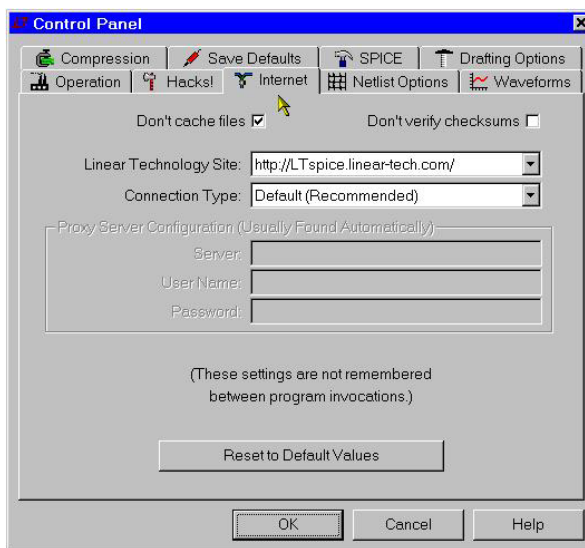


図 6.8 インターネット・オプション

コントロール・パネルのこの窓は、ウェブからのアップデートに用いられます。LTspice は、新しい機能とモデルのためにしばしばアップデートされます。現在のバージョンにアップデートするには、メニュー・コマンド Tools=> Sync Release を用いて下さい。もし数ヶ月の間アップデートしないと、LTspice はアップデートをチェックしないのですかと尋ねてきます。LTspice は、許可がなければウェブに決して接続したりはしません。LTspice は、スパイウェアを含んではいませんし、アップデート中にいかなるデータも転送されることはありません。

Don't cache files キャッシュしたりアップデートのためにマシン上のキャッシュ・ファイルを用いないようにします。

Don't verify checksums セキュリティのために LTspice は、所有のしかも秘密の 128bit チェックサム・アルゴリズムを用います。それによってアップデートのためウェブからのファイルを認証します。この認証は、アルゴリズムにエラーがある場合無効に出来ます。しかしこのエラーについての問題は報告されることはありません。よってこのセキュリティの設定をはずすことは推奨できません。

LTspice は、インターネット・アクセスに対して呼ばれる高機能演算システムを用います。まれな場合を除いてこれらの設定に何らかの調整を加える必要はありません。プロキシやパスワードを規定する必要があるとき、LTspice はインターネット・アクセスを扱うのではなく、コンピューターやオペレーティング・システムを扱います。このウインドウ上で設定することは、起動し直すと記憶されません。

第 7 章

FAQ

7.0.7 インストール問題

どの様にして SwitcherCAD III をインストールするか

1. <http://www.linear.com/software> へ行き、swcadiii.exe を自分のパソコンの一次ディレクトリへダウンロードします。
2. swcadiii.exe を走らせるとインストールできます。

ウインドウの中国語版を使っています。ギリシャ語の Mu が巧く表示できません。どうしたらよいでしょうか。

この問題については、LTspice/SwCAD III の現在のバージョンで完全に解決しています。メニュー Tools=>Control Panel=>Netlist Options の Convert μ to u をチェックすることが出来ます。このオプションはネットリストに適用されるばかりでなく、スクリーン上に表示するときにギリシャ語の Mu を u として描画します。

7.0.8 プログラム・アップデート

もっとも新しいバージョンをどうしたら手に入れることが出来るでしょうか。

一度インストールしてしまいますと、最新のバージョンを手に入れる二つの方法があります。Installation Problems に記述してある同じ方法で再インストールすることが出来ます。インストールする前に、古いバージョンを取り除く必要はありません。もしパソコンがインターネットに接続されていれば、Sync Release を用いて最新のバージョンを簡単にインストールすることが出来ます。

新しい機能が追加されたことは、どの様にすれば分かりますか

最新のバージョンにアップデートした後、ルート・ディレクトリの changelog.txt ファイル、通常 `c:\Program Files\ltc\swcadiii\Changelog.txt\verb`、は詳細なプログラム変更リストが記述されています。

Sync-Release の後古いバージョンへ戻すことが出来ますか

元に戻すことは、出来ません。全てのシンボル、モデルやプログラムは、新しいモノにアップデートされています。Sync-Release を始める前に、バックアップ・コピーを取って

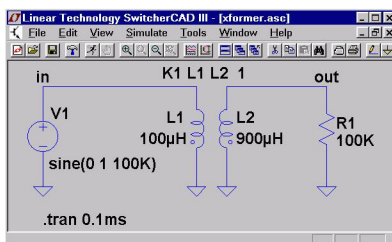
おく必要があります。部品データベース、標準は新しいモノで置き換えられます。もし新しいインダクタや容量を付け加えたとき、この素子が保存されプログラム・アップデートによって新しいモノに置き換えられます。ローカルで動作しているファイルは影響されません。

7.0.9 トランスフォーマ・モデル

どの様にしてトランスフォーマ・モデルを構築するか

回路図上で SPICE 素子として置かれている相互インダクタンス記述を持った結合インダクタのモデルを描くのがもっとも良い方法です。詳細は、mutual Inductance の所を見て下さい。相互インダクタンスのインダクタンスは、位相表示用のドットを付加して描かれます。

次の例は、1:3 巻き線比 (1:9 インダクタンス比) のトランスフォーマを正弦波入力、0.1 [ms] 時間シミュレーションを行った例です。K は、1 としトランスフォーマには漏れインダクタンスがないことを示しています。



7.0.10 他社モデル

ここでは、他社のモデルを LTspice/SwitcherCAD III に付け加えるための基礎について説明しています。

基本的に他社モデルには、二つのタイプがあります。MODEL 記述となっているか.SUBCKT 記述で定義されています。

.MODEL 記述で与えられるモデルは、ダイオードやトランジスタのような本来の SPICE 素子です。MODEL 記述は、規定された成分のパラメータを与えます。素子の振る舞いは、あらかじめ SPICE によって分かっています。素子の電気的特性を規定する必要があります。

一方.SUBCKT によって与えられているモデルは、SPICE 素子の回路の集まりによってモデル化された素子を定義しています。例えば、opamp の SPICE モデルは、サブサーキットとして与えられます。

LTspice にモデルを組み込む方法は、そのモデルが MODEL 記述あるいは.SUBCKT 記述であるかということに依存します。

.MODEL 記述を用いて定義される NPN トランジスタの例

1. シンボル NPN の特性を MODEL 記述に付け加える。
2. 値 NPN を編集し、目標の MODEL 記述に用いられている名前に近い BC547C と

する。

3. 次のいずれか

- (a) 回路図上で SPICE 指標として `.MODEL BC547C...` 記述を付け加えるか
- (b) `.MODEL BC547C...` (その他のモデルは、このファイルに多数含まれている) を含んでいる `bipol.lib` ファイルを持っているなら、回路図上の SPICE 指標に `.INCLUDE bipol.lib` を付け加えます。`bipol.lib` は、拡張子を持った完全なファイルでなくてはなりません。ウインドウズ・エクスプローラは、デフォルトでは拡張子を表示しません。もし `bipol.lib.txt`、これは notepad で演習・閲覧が出来ます、ウインドウ・エクスプローラが、そのファイルを `bipol.lib.sub` と表示している場合、SPICE 表示がこのファイルを `.inc bipol.sub.txt` とインクルードしますとこのファイルを用いるときファイルが見つからないと言うエラー・メッセージを受け取るようになります。あるいは
- (c) 代わりに `.MODEL BC547C...` 記述を次の既にあるファイルに付け加えることが出来ます。`C:\Program Files\LTC\SwCADIII\lib\cmp\standard.bjt`。もし付け加えたとすると、NPN トランジスタを編集のため選択するときに、自動的にモデルを見ることが出来るでしょう。もしこの `standard.bjt` ファイルを LTspice の外部で編集するならば、ファイルが変更されたと言うことを検知させるため LTspice を再起動する必要があります。

5 ピン opamp の例

この回路は、`.SUBCKT` 記述で定義されます。

1. 回路図にシンボル `opamp2` の図を付け加えます。
2. 値 `opamp2` を編集し `.SUBCKT` の名前と一致する `TL072` とします。
3. あるいは
 - (a) `.SUBCKT TL072ENDS` 定義を回路図上に複数行の SPICE 記述として張り付けます。
 - (b) サブサーキット `TL072` (`.SUBCKT TL072...` として始まる行を含んでいる) の定義を含んだ `TL.lib` と呼ばれるファイルがあるならば、SPICE ディレクトリ `.INCLUDE TL.lib` を回路図に加えます。

新しいシンボルを作成し、シミュレーションに必要なモデルを自動的に含ませるためプログラムすることが出来ます。ヘルプセクション Schematic Capture=>Creating New Symbols を見て下さい

3 ピン NPN トランジスタの例、`.SUBCKT` 記述で定義されている。

1. シンボル `NPN` を回路図に付け加えます。
2. 新しく置かれた `NPN` シンボルの上にカーソルを移動します。<Ctrl>RightMouseButton を押します。ダイアログボックスが現れます。プレフィックス `QN` をプレフィックス `X` へ変更します。これはバイポーラ・トランジスタをサブサーキットとしてネットリストへ認識させます。
3. 値 `NPN` を編集し、`BFG135` とし、`.SUBCKT` 行の名前と一致させます。
4. それから
 - (a) 回路図に `.SUBCKT BFG135` 行を付け加えるかあるいは

- (b) `.SUBCKT BFG135...` を含む `Phil.lib` ファイルがあるなら（このファイルに多数の素子が含まれていても構わない）コマンド行 `.INCLUDE Phil.lib` を付け加えなければいけません。

LTspice に `.SUBCKT` モデルを付け加える一つの点は、同じピンや端子を持つネットリストのサブサーキットやモデルを呼び出すために用いるシンボルがあることです。上記の例では、一般的なピン番号順となっている他社のモデルを仮定しています。

さらに詳しい情報は、`Schematic Capture` や LTspice ヘルプにあります。基本的な考えは、回路図描画プログラムは、LTspice が読み込むシミュレーション用のネットリストを作成します。他社モデルをインポートするいくつかの点は、`SPICE` ネットリスト・シンタックスを理解することによって解決することが出来、どの様に回路図入力プログラムがそのシンタックスを発生しているかを理解することが出来ます。このトピックスについてのチュートリアルが、独立のユーザー・グループ <http://groups.yahoo.com/group/LTspice> にあります。

7.0.11 インダクタ・モデル

どの様に結合インダクタを設計すればよいか

最初に少なくとも二つのインダクタを描きます。次の二つのインダクタの係数 K を定義します。mutual inductance を参照して下さい。

どの様にインダクタンス寄生抵抗を制御するか

デフォルトでは、LTspice は SMPS 過渡解析を補助するためインダクタに損失を挿入します。SMPS に対して、これらの損失は、一般にそれほど重要ではありません。もし必要であるならこの機能を止めることが出来ます。Tools=>Control Panel=>Hacks!のところで、Supply a min. inductor damping if no Rpar is given のチェックをはずして下さい。この設定は、プログラムが起動の間記憶されています。インダクタに対して1ミリオームのデフォルト直列抵抗もあります。これは相互インダクタンス記述の所では述べられていません。この R_{ser} は SwitcherCAD III に対して、回路の線形行列の大きさを減少させるためテブナンの等価回路の代わりにノートンの等価回路としてインダクタンスに組み込まれています。この最少抵抗を LTspice に導入したくなければこのインダクタの R_{ser} をゼロに設定しなければなりません。このことは過渡解析の間インダクタの煩わしいテブナン等価回路を用いるよう LTspice に要求します。

自分のインダクタ・モデルを付け加え、編集できますか。

`C:\Program Files\LTC\SwCADIII\lib\cmp\standard.ind` としてインストールされているファイルを開き、インダクタ・モデルを付け加え、編集します。

7.0.12 MOSFET モデル

SwitcherCAD III MOSFET と標準 SPICE MOSFET モデルの違いは何ですか

標準 SPICE MOSFET モデルに対し、SwitcherCAD III も他の SPICE プログラムに組み込まれていない独自の MOSFET モデルを含んでいます。垂直二重拡散 MOS トランジス

タの電荷動作を直接包含しています。このため他の SPICE プログラムでサブサーキットとして扱っているのとは異なり、VDMOS 素子でモデル化された電力素子として扱えます。詳細は、models 定義を見て下さい。

自分の MOSFET モデルを付け加えられますか

はい。C:\Program Files\LTC\SwCADIII\lib\cmp\standard.mos ファイルのモデルを付け加えることが出来ます。このファイルは、.model 記述で定義される素子に対してのみ使うことが出来ます。サブサーキットには使えません。もしサブサーキットを用いたのでしたら、次のステップに従って下さい。

1. シンボル X の素子特性について、プレフィックス特性を変更します。シンボルは変更しません、回路図上の成分としての特性を変更します。コントロール・キーを押して素子の上でマウスを右クリックすればこの特性が見えます。
2. 用いたいサブサーキットの名前を持つ素子の特性値を編集します。
3. .inc filename のように回路図上の SSPICE ディレクトリを付け加えます。このファイル名は、サブサーキットの定義が含まれているファイルの名前です。これは拡張子を含んだ完全な名称でなければなりません。この拡張子は、ウインドウ・エクスプローラのデフォルトでは表示されません。mylib.sub.txt という名称のファイルであれば、notepad で編集閲覧することが出来ます。ウインドウ・エクスプローラは、mylib.sub としてファイルが存在していることを示します。このファイルを含む SPICE ディレクトリは、.inc mylib.sub.txt です。もし.inc mylib.sub を使うとすると、ファイルが見つからないと言うエラー・メッセージを受け取るでしょう。

7.0.13 ライセンスと配布

ソフトウェアを再配布できますか

はい、Linear Technology のお客であってもなくても、このソフトウェアを配布することが出来ます。詳細は、license の所を見て下さい。Linear Technology のお客様でないときの技術サポートは、自由裁量となっています。

シェアウェア、フリーウェアそれともデモ

このプログラムは、シェアウェアやデモではありません。このプログラムは全ての機能を備えたフリーウェアです。このソフトウェアの目的は、我々の製品を使っているお客を助けることです。汎用の回路図入力とシミュレーションがある回路設計ソフトウェアとして用いることもできます。アナログ回路設計に親しんでもらうため、学生にこのプログラムを用いるよう推奨しています。Linear Technology に関係のない用法についてのサポートは出来ません。しかし全ての一般的なプログラムのバグをなくしたり、そのレポートは行っています。社内でテストもしておりこのプログラムが優れた取束性を持っていると信じています。知りうる限りのバグは含まれていません。

ヘルプのために Linear Technology とコンタクトできますか

全てのソフトウェア問題については e-mailscad3@linear.com ハードウェア問題について、例えば Linear Technology IC のアプリケーション・ノート、電話 Linear Technology application department (408)954-8400 通常営業時間

7.0.14 有効な計算

*.APP と *.ASC の差は何ですか

APP ファイルは、回路図ファイルで、計算効率を補助する制御記述が組み込まれその他の製品情報が含まれています。ASC ファイルは、いくつかの SPICE コマンドが無い汎用の回路図ファイルです。ASC ファイルは、より一般的で強力なファイル形式です。設計を保存するとき、ASC 拡張子で保存することを推奨します。

回路図に対して効率のレポートを作ることができますか

回路図上に .TRAN <Time> steady 記述を付け加える必要があります。プログラムは、自動的に LTC マクロモデルの内部状態を調べて、定常状態を検知します。LTC スイッチング・レギュレータ部分が、存在しなければ動きません。回路の中に一つの電圧源がなければなりません。入力についても同様です。回路の中に一つの電流源がなければなりません。これは負荷と同じです。シミュレーションが完了した後、回路図上のレポートを見るため View メニューの Efficiency Report を選ぶことができます。

7.0.15 カスタム・シンボル

自分のシンボルを作れますか

勿論作れます。

どの様にしてシンボルを作るのでしょうか

メニュー・コマンド File=>New Symbol から始めて下さい。

自分のスイッチング・レギュレータ・モデルを作れますか

簡単ではありません。LTspice/SwitcherCAD III と共に出荷されているスイッチング・レギュレータ・モデルは、LTC のスイッチング・レギュレータ製品の振る舞いを表現するため、新しい SPICE 素子を使っています。標準的な SPICE モデルを使うとなると、シミュレーション時間は大きくなるでしょう。ある人々は、このようなスイッチング・レギュレータ・モデルを標準的 SPICE 素子を用いています。LTspice は、これらのモデルを走らせることができますが、目標としているシミュレータとは違うパフォーマンスとなります。

7.0.16 メモリ問題

プログラムを走らせるには、どのくらいのメモリが必要ですか

ウィンドウズ 95, 98, NT4.0, Me, あるいは 2000 を走らせるのに十分なメモリが必要です。ウィンドウズ・システムを動かすことが出来れば SwitcherCAD III が基本的に動きます。このプログラムに必要なメモリを最小にするように多大の努力をしました。典型的なシミュレーションでは 8 ギガバイトの生データを生成します。このデータは、400 メガバイトでディスクに圧縮保存されます。一つの軌跡を見るには、65 メガバイト以下の RAM が必要です。勿論メモリが多いほどパフォーマンスは向上します。またウィンドウズ NT や 2000 の改善されたメモリ管理は、SwitcherCAD III にも有益です。その様なわけで、もしメモリが足りなくなるようであれば OS のアップグレードも考えるべきです。

シミュレーション中波形データはどこに保存されていますか

全ての波形データは、ハードディスクに保存されています。マーチング・ウエーブフォームをはずすと RAM メモリが少なくなくて済みます。ほとんどの解析方法で、ファイルの大きさについての制限はありません。非常に大きなギガバイト・ファイルとなる .raw ファイルを生成し見ることが出来ます。

十分なディスク容量が無く長時間シミュレーションをするとどうなりますか

波形データは、圧縮されます。しかし計算時間に比例し軌跡の数が保存されます。メモリ消費を少なくする簡単な方法は、シミュレーションが始まる前に保存する希望の軌跡を選んでおくことです。

全てを実行しました。メモリを越えて走らせています。どうすればよいでしょうか

過渡解析の間、過去の波形データを Okey を押すことによってその都度消すことが出来ます。これは現在の時間としてシミュレーション時間を $t=0$ とします。

7.0.17 モデル互換性

新しいモデルは、古いモデルと互換性がありますか

常にマクロモデルを改良しています。これらの変更は、通常より正確な素子振る舞いをするようになります。ユーザには気が付かないで変更がなされています。例えば動作点やピン電流が印可されますが、過渡解析波形の変化はそれほどありません。場合によっては、記号がある特定の環境に対して変更されます。記号の前のバージョンに対して描かれた回路図に影響するでしょう。用いた設計やシンボルのバックアップを保存しておくべきでしょう。

スイッチング・レギュラー・モデルは、PSPICE やその他のモデルと互換性がありますか

シミュレーションをスピードアップするためあるいは収束性を制御するためスイッチング・レギュレータ・モデルに独自のビルディング・ブロックを用いています。一方基本的に独自のビルディング・ブロックを標準的な SPICE 等価ブロックへ変換することが出来

ます。シミュレーションは数百分の一まで速度が落ちます。パフォーマンスに関して、全てのスイッチング・レギュレータは、独自のブロックで組み立てられています。そこに変換器はありません。LTspice は、Pspice 半導体やビヘービア・モデルモデルを走らせることが出来、高機能なシミュレータです。そこで Pspice シミュレーションを LTspice へ移動することが出来ます。

7.0.18 SPICE ネットリスト

SPICE ネットリストをどの様にして作成するか

ネットリストは、ASCII ファイルを作成することが出来る任意のテキスト・ファイル・エディターで作成することが出来ます。コマンド `View=>SPICE netlist` で SwitcherCAD III の任意の SPICE ネットリストを見ることが出来ます。この窓から全てのテキストを選択しネットリストをクリップボードへコピーすることが出来ます。そして `Ctrl-C` とタイプして、違うエディターへネットリストを持って行くことが出来ます。

どの様にネットリストを走らせるか

最初にテキストファイルを開き、走らせます。LTspice/SwitcherCAD III は、.cir の拡張子が有ればネットリストとしてファイルを認識します。

7.0.19 波形データの出力

他の応用のために波形データを出力できますか

波形表示窓をアクティブとし、`Ctrl-C` とタイプすることで描画をビットマップファイルとしてコピーできます。ビットマップを受け入れる word や Paint の様なソフトウェアにクリップボードから張り付けるため `Ctrl-V` とタイプします。これは回路図上のビットマップとしても動きます。これらのイメージはウィンドウズ・メタファイルとしても出力できます。(メニュー・コマンド `Tools=>Write` は、.wmf ファイルへ) これは様々な desktop publishing tools へインポートすることが出来る.wmf ファイルへベクター・グラフとして記述します。波形データのメタファイルを出力するとき、最初に `Tools=>Control Panel=>Waveform=>Font` へ行き、Arial を選んで下さい。デフォルトでは CRT に適切なフォントが選んでありますが、メタファイルでは正しくスケールされない固定フォントです。

ビットマップとして動くようですが、Excel の様なアプリケーションにデータそのものを持ってこれますか

エクスポート・ユーティリティがあります。(Waveform Menu: `File=>Export`) これはデータを ASCII ファイルで出力します。このほかに Helmut Sennewalt によって書かれた無料のユーティリティがあります。これは独立のユーザー・グループ <http://groups.yahoo.com/group/LTspice> から入手できます。このユーティリティは、様々な形で出力できます。その結果異なったシミュレーションからの結果を合成することが出来ます。

7.0.20 Linux で走らせる

このプログラムの Linux 版はありますか

別の特別なプログラムではなく、このプログラムは WINE 上で動きます。このプログラムは、WINE バージョン 20030219 を用いて Linux RedHat 8.0、WINE バージョン 20040716 を用いて Linux RedHat 9.0、20040716 を用いて Linux SuSE 9.1 でテストされています。

今まで WINE を使ったことがありません。どの様にインストールするのでしょうか

使っているシステムに対して WINE の現在のバージョンを見つけるため <http://www.winehq.com> をチェックして下さい。これが書かれているとき Red-Hat 8.0 に対して、<http://mecano.gme.usherb.ca/~vberon/wine> に指摘されています。適切な.rpm ファイルをパソコンにコピーして nautilus からそれを開いて下さい。<http://www.linear.com/software> から swcadiii.exe をダウンロードします。xterm で wine swcadiii.exe と実行し、LTspice をインストールします。gnome デスクトップ上に Linear Technology Logo が現れます。それをクリックしてスタートさせるかあるいは xterm から wine scad3.exe と入力してプログラムをスタートします。

回路図のフォントは、ウインドウズの場合と同じように WINE 上で滑らかに変更できないのはどうしてでしょうか

ウインドウ・システムから native Windows .dll を使ってみて下さい。WINE から LTspice 起動して wine -dll commctrl,comctl32=n scad3.exe

ウインドウズの場合と違って WINE/Linux では少し違っているようですが

LTspice は、WINE 上で動いているかどうかを検知します。もしそうならば、いくつかの WINE コマンドで動きます。コマンドライン switch -wine で LTspice を WINE の上で動かすように出来ます。WINE コマンドとは関係がないようにするには、switch -nowine とします。

7.0.21 紙のマニュアル

<http://LTspice.linear-tech.com/software/scad3.pdf> からこのヘルプファイルの pdf ファイルをダウンロードすることが出来ますまた必要なら印刷することも出来ます。

7.0.22 ユーザーグループについて

<http://groups.yahoo.com/group/LTspice> に独立のユーザーグループがあります。このグループは、他のチュートリアル、ライブラリ、例のファイルがあります。

付録 A

シンボルの概要

この付録は、LTspice によって供給されている回路シンボルについて述べています。

- A.1 電力製品
- A.2 演算増幅器
- A.3 その他
- A.4 光学アイソレータ
- A.5 特別な関数
- A.6 フィルタ製品
- A.7 デジタル
- A.8 比較器

