

はじめに

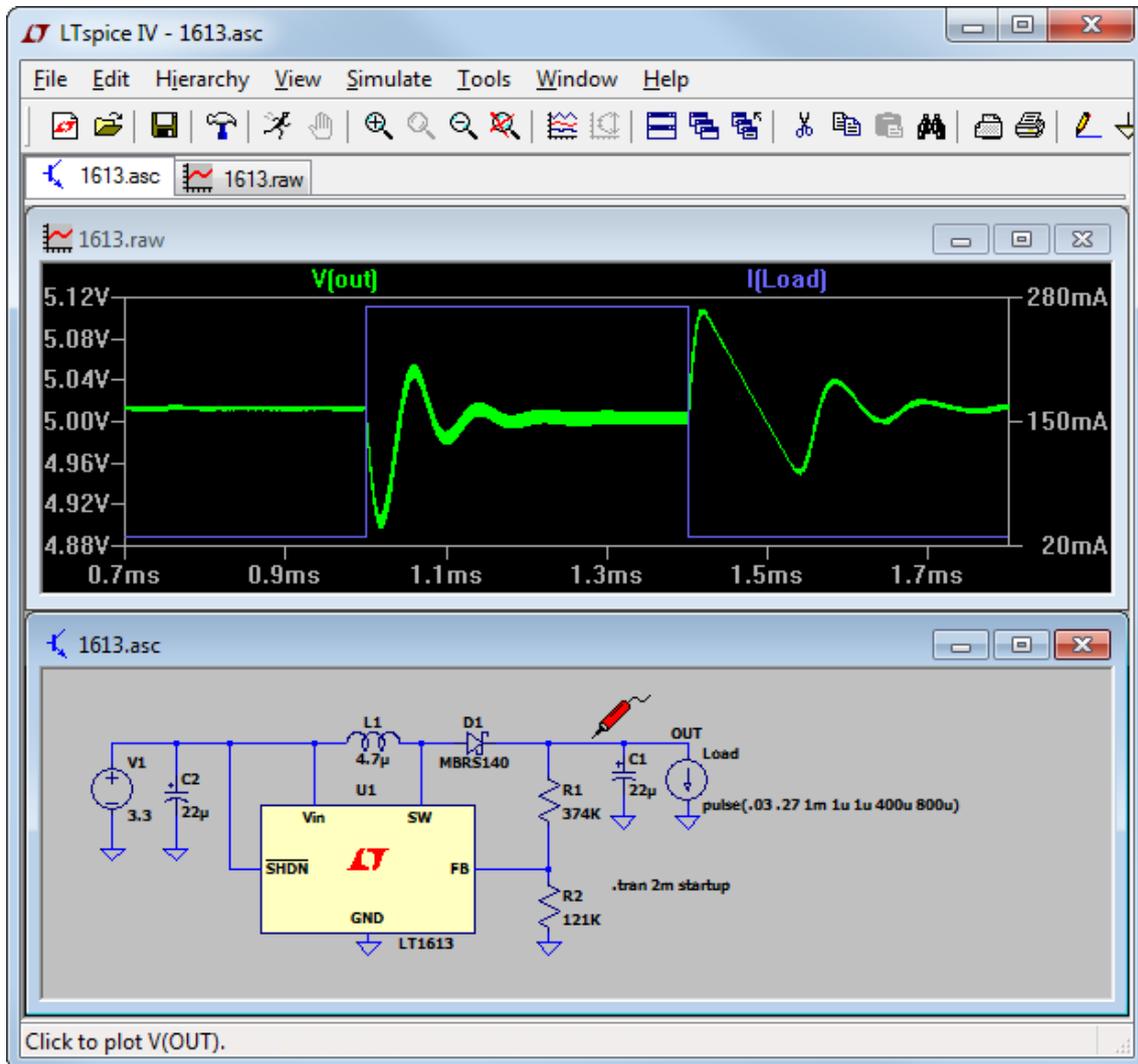
LTspice は、統合回路図取り込み機能と波形ビューワを備えたアナログ回路シミュレータです。LTspice は、リニアテクノロジー株式会社の半導体メーカーとしての競争面での優位性確保の一環として、社内での IC 設計に使用するために、ソフトウェア会社から販売されている類似のツールより性能面で上回ることを明確な目標として開発されました。これは既存の商用 SPICE 製品が過剰に出回っているにもかかわらず、合理的な戦略です。少なくとも私はそうと思いますが、数十年間物理シミュレータを開発してきた観点から利益をもとに話すと、最高のシミュレータを開発したのは、ソフトウェア会社ではなく、実際にシミュレータを必要とする会社であると理解しています。

LTspice に関しては、潜在的なお客様が LTC 製品をより高速なシミュレータでシミュレートできるように無償で配布されているのも特別なことです。さらに、無償で配布されたバージョンは、それが役立つことを期待して、機能を制限していません。

これは SPICE シミュレータに特有の状況であり、このために LTspice の人気は素晴らしさに引けを取りません。LTspice は圧倒的な支持を得て、現在まで業界で最も広く配布され使用された SPICE プログラムです。LTspice は、SPICE プログラムの事実上の業界標準になりました。

-Mike Engelhardt / 2012

まえがき



LTspice IV はリニアテクノロジーが開発した第 4 世代のスイッチング・レギュレータ設計プログラムです。このプログラムは、混合モード・シミュレーション機能によって拡張された高性能 SPICE シミュレータで構成されています。このシミュレータには、スイッチ・モード電源 (SMPS) コントローラおよびレギュレータをマクロモデル化するための新しい固有の SPICE デバイスが組み込まれています。このプログラムは、ユーザが SMPS 回路例を編集したり新しい回路を設計したりできる統合型階層回路図取り込みプログラムを内蔵しています。統合波形ビューワはシミュレートされた波形を表示し、シミュレーション・データの詳細な解析が可能です。リニアテクノロジーのパワー IC と多くの受動素子の大半について組み込みのデータベースがあります。デバイスのデータベース、回路図の編集、シミュレーションの制御、および波形解析が 1 つのプログラムに統合されています。

混合モード・シミュレーション機能や以前の SPICE プログラムに対するその他の多くの拡張により、シミュレーション速度を大幅に向上しつつ、シミュレーション精度を保持しています。サイクルごとの詳細な SMPS シミュレーションの実行と分析を数分以内に行うことができます。ユーザは、デバイス、SPICE または回路図取り込みプログラムについて何も知らなくても、マウスを数回クリックすれば電力系統の詳細なシミュレーションを行うこ

とができます。あらかじめ下書きされたデモ回路を出発点として使用して、さまざまな電源要件に適合するカスタム回路を作成できます。新しい回路図が作成されたら、系をシミュレートして、レポートを生成することができます。

このプログラムの統合型階層回路図取り込み機能と SPICE シミュレータは、一般的用途に利用可能です。LTspice は SMPS の設計と密接に関連しているにもかかわらず、SMPS 固有の SPICE ではなく、SMPS を対話的にシミュレートするのに十分高速な SPICE プログラムに過ぎません。SPICE シミュレーション・エンジンの性能向上は、一般的なアナログ回路をシミュレートする上で利点となり、すべての電子工学技術者が関心を示すはずです。部品の点数や内容に恣意的な制限はありません。インストール・ベースのライセンスはこれまでに 3,000,000 以上を記録しているので、LTspice は SPICE シミュレータの事実上の業界標準になっています。このプログラムがお役に立てば幸いです。

ハードウェア要件

LTspice IV は 32 ビットまたは 64 ビット・バージョンのすべての Windows 上で動作します。また、LTspice は、WINE を介して Linux 環境で動作し、Crossover、Parallels、または DARWINE を介して最新の Macintosh ハードウェア環境で動作します。Windows には、一部の LTspice ユーザに問題が発生する 2 つのバグが存在します。これらのバグは、両方とも Windows が真のファイル名およびファイルのパスを隠すことに関係があります。LTspice では、外部ファイル(つまり、サード・パーティ・ファイル)を使用するのが一般的なので、混乱が生じることがあります。

これらのバグを回避するため、次の手順に従ってください。

まず、エクスプローラで、「フォルダー オプション」の「表示」に移動し、「登録されている拡張子は表示しない」のチェックボックスをオフにします。次に、Windows Vista または Windows 7 では、「ユーザ アカウント制御」(UAC)をオフにします。UAC を使用すると、ディレクトリを表示するのに使用するアプリケーションによってファイルのパスが異なる場合があります。これにより、ファイルを名前ごとにコンピュータに格納する基本的な枠組みが崩れます。UAC を無効にすることを強く推奨します。無効にしない場合、明らかな理由なくファイルがなくなったように表示されることがあります。

ソフトウェアのインストール

LTspice IV は <http://www.linear-tech.co.jp> でダウンロードできます。配布ファイルへの直接のリンクは、<http://ltspice.linear.com/software/LTspiceIV.exe> です。ファイル LTspiceIV.exe は、解凍時に LTspice IV をインストールする自己抽出型 gzip ファイルです。

LTspice IV は頻繁に更新されます。Web にアクセスできる場合は、最初に LTspice IV をインストールした後、インストール環境を現行のバージョン・レベルにする組み込みの `update*` メニューコマンドを使用できます。更新処理では、配布時の各ファイルのサイズおよびチェックサムが入っているマスタ・インデックス・ファイルがリニアテクノロジーの Web サイトから最初にダウンロードされます。ファイルが欠落している場合、ファイルのサイズが異なる場合、またはローカルのチェックサムと索引ファイルのチェックサムの間に差がある場合、そのファイルは自動的に更新されます。部品データベースは更新処理時にマージされるので、デバイスをインストール環境に追加した場合は、自動更新ユーティリティを実行したときに追加分のデバイスが失われることはありません。

* Update

「Sync_Release」のコマンドを実行する前にすべてのウィンドウを閉じてください。インターネットへの接続が必要です。LTspice IV のプログラムは、マスタ・インデックス・ファイルが、LTC のサーバからダウンロードされます。マスタ・インデックス・ファイルは、サブディレクトリ内にあるすべてのファイルのチェックサムが含まれています。ローカルファイルのチェックサムと比較して最新のファイルが存在する場合、自動的にダウンロードし、LTspice IV のプログラムファイルと同じ名前の上書きします。マクロモデルのほとんどが 3 キロバイト未満なので、数秒で転送されます。SCAD3.EXE の更新ではダウンロードが完了後、最初に新しいファイルが Windows の一時ディレクトリにコピーされます。そのため、ユーザがファイル転送をキャンセルしても、古い SCAD3.EXE はそのまま保存されます。ファイル「The changelog.txt」は、プログラム改訂の変更をリストします。

LTspice -- 使用許諾契約/免責事項

Copyright © 2001-2012, Linear Technology Corporation. All rights reserved.

LTspice は、リニアテクノロジー株式会社のスイッチ・モード電源合成回路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で保護されています。お客様には、弊社 (LTC) 製品を評価し、また一般的な回路シミュレーションを行なうことに関して、非独占的で譲渡不能な、2 次ライセンスを与えない、ロイヤルティ・フリーの権利が与えられます。このソフトウェアはリニアテクノロジー株式会社が所有しています。お客様は、支給されたソフトウェアの実行可能ファイルまたは弊社製品のモデルの修正、改造、翻訳、リバース・エンジニアリング、逆コンパイル、逆アセンブルのいずれも行うことはできません。弊社は、弊社またはお客様によって提供されたシミュレータで使用されるサードパーティ・モデルの精度について責任を負いません。

弊社では、LTspice が記述どおり確実に動作するようにあらゆる取り組みを行っていますが、動作にエラーがないことを保証してはおりません。このプログラムのアップグレード、変更、または修復は、もっぱら弊社の判断により行われます。弊社製品を選択して評価する目的で LTspice のインストールまたは操作の問題が発生した場合は、太平洋標準時で月曜から金曜の午前 8:00 から午後 5:00 まで(408) 432-1900 で弊社の Applications Department にご連絡いただければ、技術的サポートを提供いたします。弊社製品の評価が目的でない一般的な回路シミュレーションについては、そのような技術サポートはいたしません。現在使用中の PC 互換コンピュータ・システム、オペレーティング・システムのバージョン、および周辺装置は多種多様であるため、弊社は、そのようなすべてのシステム上で LTspice を正常に使用できることは保証していません。LTspice を使用できない場合、弊社では、必要なあらゆる手段を講じることにより、弊社スイッチング・レギュレータ IC の設計サポートを提供します。

ソフトウェアおよび関連文書は「現状有姿」で提供されており、いかなる種類の保証もいたしません。また、リニアテクノロジー株式会社は、商品性および特定の目的に対する適合性の暗黙の保証を含むがこれらに限定されないその他すべての保証を、明示か黙示かにかかわらず、一切いたしません。この製品の使用または使用不能から生じる損害に対しては、そのような損害の可能性が弊社に前もって通知された場合でも、直接か間接かを問わず、弊社はいかなる責任も負わないものとします。このソフトウェアの再配布は許可されていますが、すべての文書、サンプル・ファイル、シンボル、モデルと共に、改変や追加のない状態でソフトウェア全体を配布する場に限りです。

このプログラムは、特に半導体メーカーが自社製品の販売促進、デモンストレーション、または販売で使用する場合には許可されません。これらのアプリケーションに対して LTspice を使用する場合は、リニアテクノロジーから個別の許可を得る必要があります。

動作モード

LTspice IV には、シミュレータを駆動する基本的な 2 つのモードがあります：

- 1) 統合シミュレータを備えた汎用の回路図取り込みプログラムとしてこのプログラムを使用する。メニューコマンドは「File」=>「New」、および「File」=>「Open」(ファイル・タイプは.asc)
- 2) 手作業で作成したネットリストか、別の回路図取り込みツールで生成した外部ネットリストをシミュレータに入力する。メニューコマンドは「File」=>「Open」(ファイル・タイプは.cir)

LTspice IV は、統合 SPICE シミュレータを備えた汎用回路図取り込みプログラムとして使用することを意図しています。発想は、回路を作成し(または既の下書きされている回路例から始めて)、その動作をシミュレータで観察するというものです。

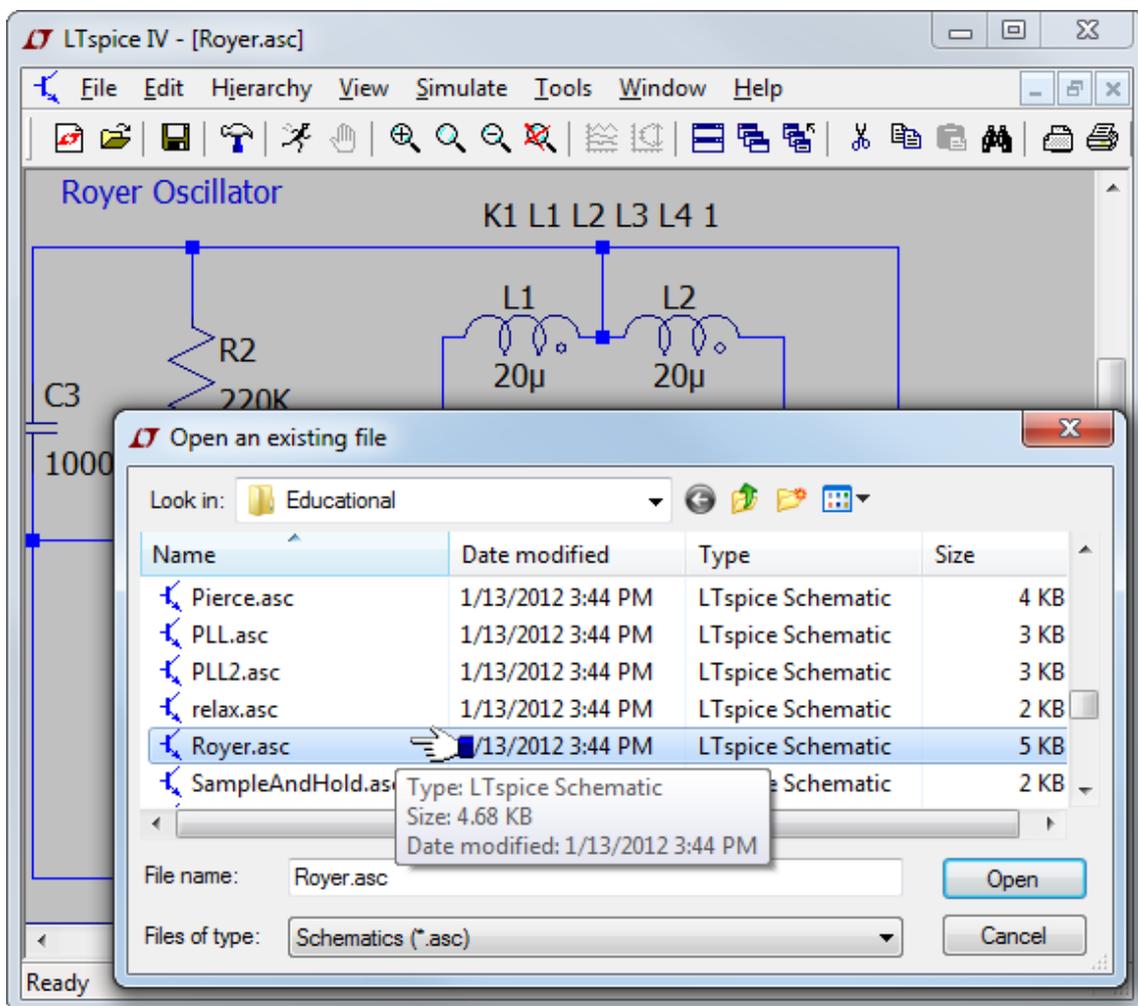
設計過程では、目的の回路動作がシミュレーションで実現するまで回路を繰り返し作成する必要があります。LTspice の旧バージョンには、ユーザ指定の仕様を基に SMPS 設計回路を推測しようとするシンセサイザが組み込まれていましたが、その動作モードは廃止されました。回路図は、最終的にはシミュレータに渡されるテキスト形式の SPICE ネットリストに変換されます。

ネットリストは、LTspice 内で下書きされた図式回路図から抽出されるのが普通ですが、インポートしたネットリストは回路図なしで直接実行できます。これにはいくつかの使用方法があります。

- i) リニアテクノロジーのフィルタ合成プログラムである FilterCAD は、LTspice のネットリストを合成して、フィルタの時間領域または周波数応答をシミュレートすることができます。
- ii) LTspice のベンチマークテストを簡素化して、他の SPICE と対比させます。
- iii) 従来のシステムでの SPICE シミュレータには回路図取り込み機能が組み込まれていなかったため、SPICE 回路シミュレータの使用経験が長い専門家は、テキスト形式のネットリストを直接処理することに精通しています。

回路例

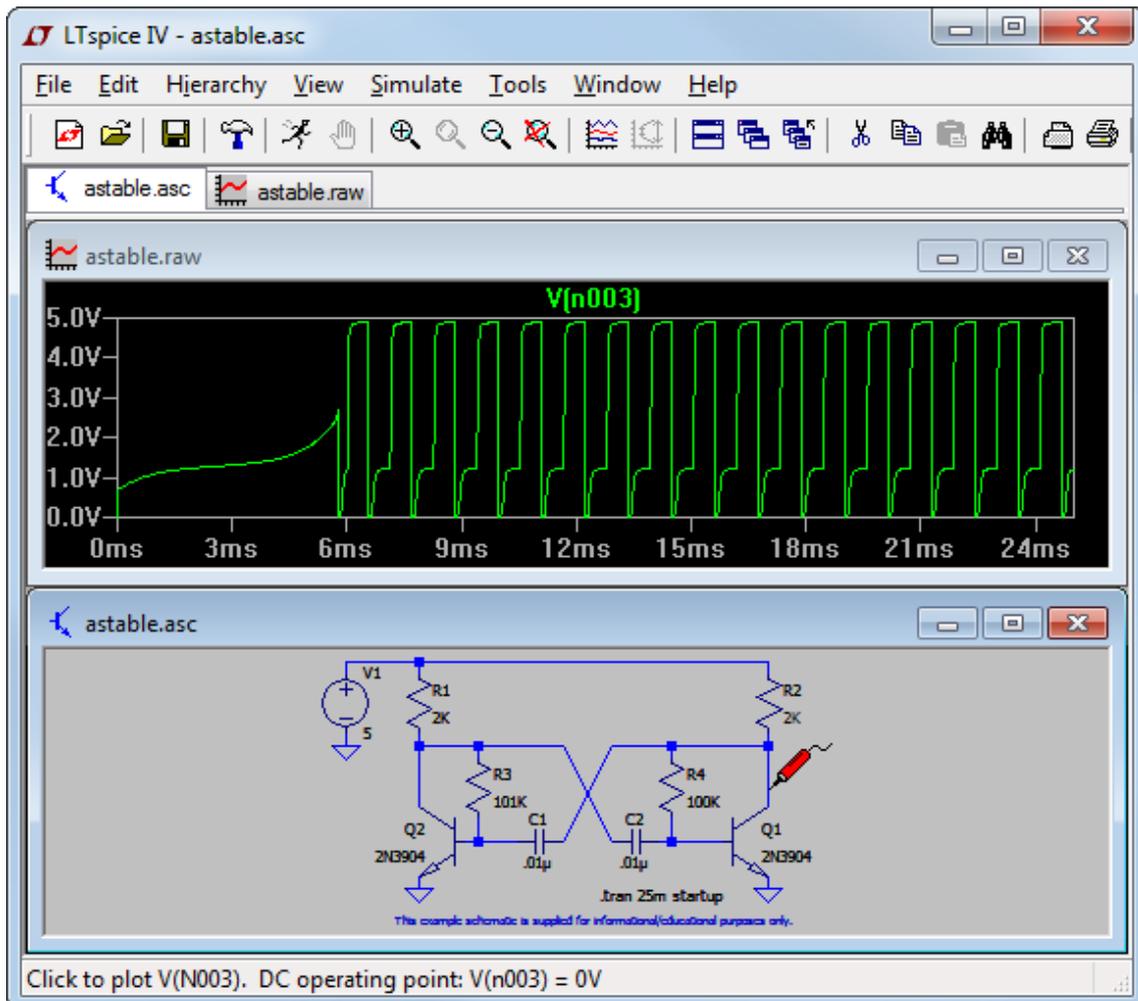
LTspice IV には、回路例のリソースがいくつかあります。標準的なインストール先ディレクトリは C:\Program Files\LTC\LTspiceIV\examples\Educational です。ここには、さまざまな種類の解析、方法、またはプログラム機能を示す非営利目的の SPICE シミュレーションの例が多数あります。ディレクトリ C:\Program Files\LTC\LTspiceIV\examples\jigs には、LTspice IV にマクロモデルが用意されているすべてのリニアテクノロジー・デバイスのシミュレーション例があります。これらの治具回路は、多くの場合、マクロモデルのテスト治具であるにすぎず、必ずしも推奨の基準設計回路ではないことに注意してください。最も重要なことは、リニアテクノロジーの最寄りの事業所は、お客様のアプリケーションの要求に合った具体的な設計サポートをほぼ確実に提供できるということです。



汎用回路図方式の SPICE

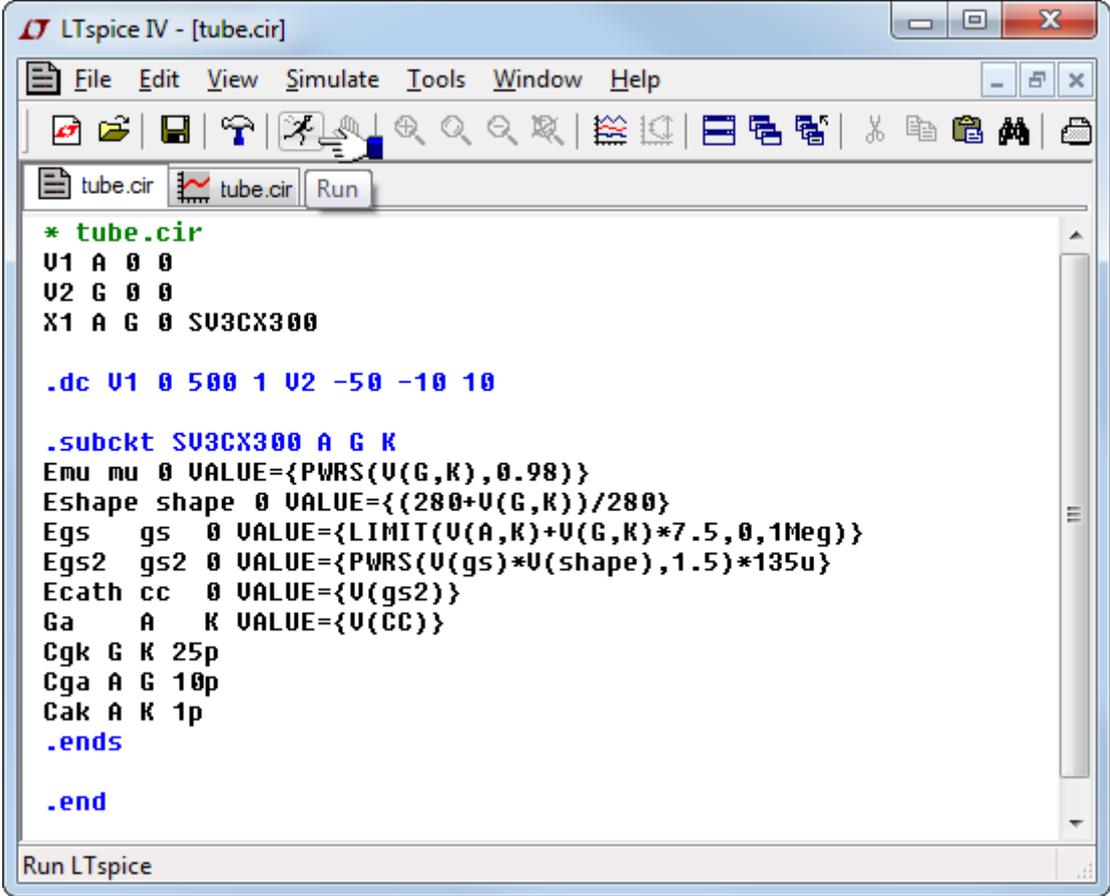
LTspice IV は汎用の回路図取り込み/SPICE プログラムとして自由に使用できます。これは SMPS の設計だけでなく、アナログ・エンジニアリングの多くの面で役立ちます。

標準ではディレクトリ C:\Program Files\LTC\LTspiceIV\examples\Educational\にインストールされるこの回路例は、さまざまな LTspice 機能を示します。



外部生成ネットリスト

手作業か他の回路図取り込みプログラムで生成したネットリストを開くことができます。これらのファイルのファイル名拡張子は、通常は「.cir」ですが、「.net」および「.sp」も解釈されます。ネットリスト・ファイルに使用されるASCIIエディタは、無制限のファイル・サイズおよび無制限の取り消し/やり直しをサポートします。メニューコマンド「Tools」=>「Color Preferences」を使用すると、ASCIIエディタで使用する色を調整できます。



The screenshot shows the LTspice IV interface with a netlist file open. The netlist content is as follows:

```

* tube.cir
U1 A 0 0
U2 G 0 0
X1 A G 0 SU3CX300

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

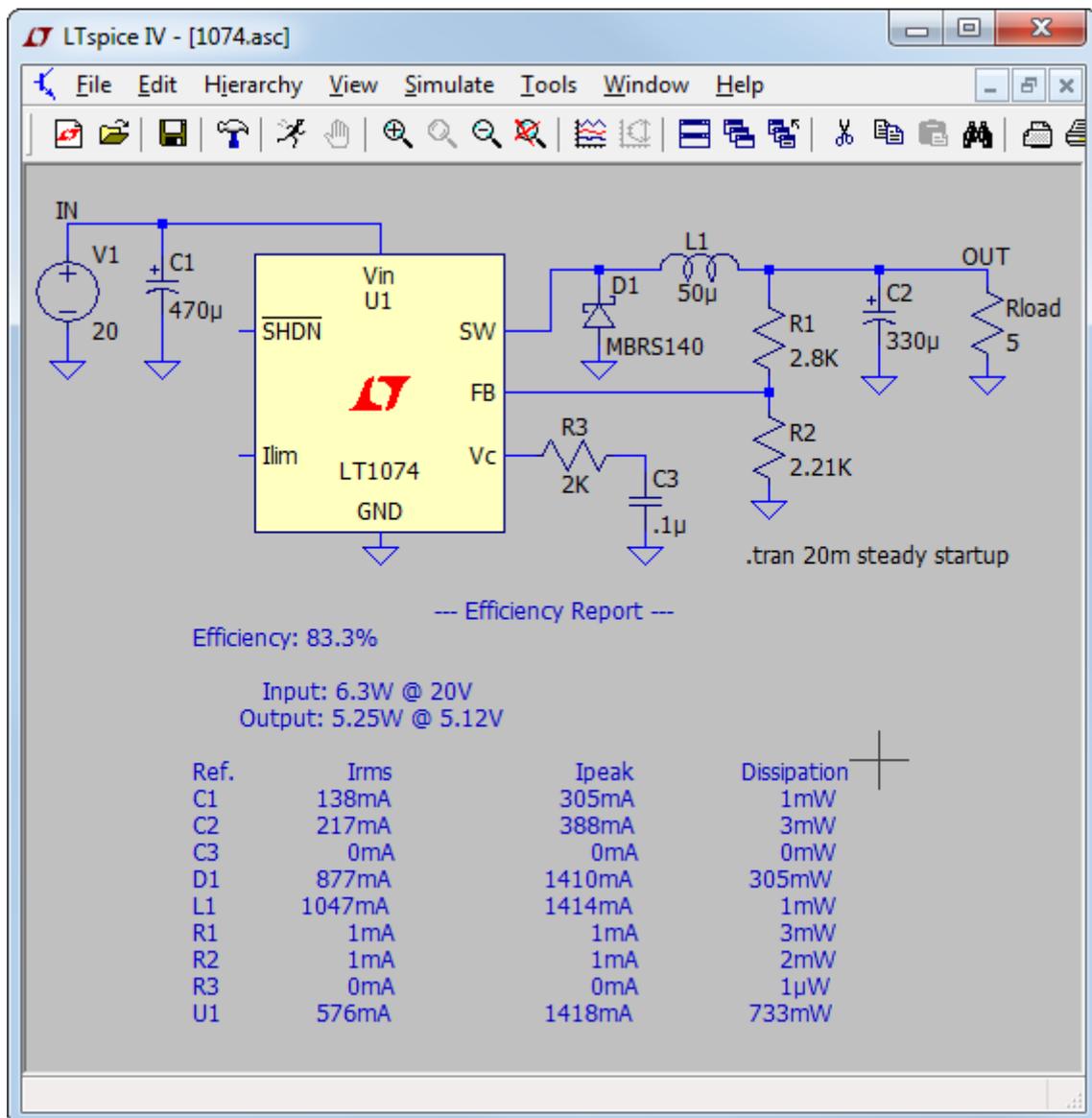
.subckt SU3CX300 A G K
Emu mu 0 VALUE={PWR(U(G,K),0.98)}
Eshape shape 0 VALUE={(280+U(G,K))/280}
Egs gs 0 VALUE={LIMIT(U(A,K)+U(G,K)*7.5,0,1Meg)}
Egs2 gs2 0 VALUE={PWR(U(gs)*U(shape),1.5)*135u}
Ecath cc 0 VALUE={U(gs2)}
Ga A K VALUE={U(CC)}
Cgk G K 25p
Cga A G 10p
Cak A K 1p
.ends

.end

```

効率レポート

キーワード「steady」が含まれる時間領域.tran 解析により、DC-DC コンバータから効率レポートを取得することが可能です。定常状態シミュレーション後、効率レポートは一連のコメント文字列として回路図上に表示できます。



DC-DC コンバータの効率は次の方法で導き出されます。入力と出力を識別するために、電圧源と電流源は必ず1つずつ存在する必要があります。電圧源は入力であるとみなされるのに対して、電流源は出力であるとみなされます。回路はシミュレータによって定常状態が検出されるまで実行されます。このためには、定常状態の検出方法に関する情報を使用して SMPS マクロモデルを作成する必要があります。通常、これは、クロック・サイクルの間平均化されたエラー・アンプ電流が数サイクルにわたって小さな値に減少するときを記録することにより検出されます。その後、クロック・エッジでは、各リアクタンスに蓄積されたエネルギーが記録され、シミュ

レーションはさらに 10 回のクロック・サイクルにわたって実行されますが、今度はすべてのデバイスでの電力損失が積算されます。

最後のサイクルのクロック・エッジでは、すべてのリアクタンスに蓄積されたエネルギーが再び記録され、シミュレーションは停止します。効率は、リアクタンスに蓄積されたエネルギーの変化について調整した後に、入力電圧によって供給された入力電力と負荷に供給された出力電力の比として報告されます。各デバイスの電力損失も記録されたので、エネルギー・チェックサムがどの程度 0 に近づいているか調べることができます。

「Edit Simulation Command」エディタで「Stop simulating if steady state is detected」チェックボックスをオンにすれば、通常はユーザが自分で下書きした SMPS 回路の効率を計算できます。シミュレーション後、メニューコマンド「View」=>「Efficiency Report」を使用します。

定常状態の自動検出は必ずしもうまく機能するとは限りません。定常状態の検出基準が厳しすぎることもあれば、緩すぎることもあります。その後、オプション・パラメータ sstol を調整するか、効率の組み込みについて制限値をそのまま対話式に設定します。

コマンド・ライン・スイッチ

次の表は、LTspice 実行可能ファイル (scad3.exe) によって解釈されるコマンド・ライン・スイッチを要約したものです。

フラグ	説明
-ascii	ASCII .raw ファイルを使用します。プログラムの性能が著しく低下します。
-b	バッチ・モードで動作します。たとえば、「scad3.exe -b deck.cir」と指定すると、ファイル deck.raw の中にデータが残ります。
-big	最大限の期間として起動します。
-encrypt	モデル・ライブラリを暗号化します。実装の詳細を明らかにせずに、ライブラリを使用できるようにしたいサードパーティ向けです。Linear Technology Corporation モデルでは使用しません。
-FastAccess	2 進.raw ファイルの Fast Access フォーマットへの一括変換です。
-ini <path>	%WINDIR%\scad3.ini 以外に使用する.ini ファイルを指定します。
-max	-big と同義
-netlist	回路図のネットリストへの一括変換です。
-nowine	WINE (Linux) 回避策を使用しないようにします。
-PCBnetlist	回路図の PCB フォーマットのネットリストへの一括変換です。
-registry	ユーザ設定 (MRU など) を %WINDIR%\scad3.ini ファイルではなくレジストリに格納するよう LTspice に指示します。
-Run	開いている回路図のシミュレーションを、「Run」ボタンを押さずに、コマンド・ライン上で開始します。
-SOI	サブ回路の展開であっても MOSFET に最大 7 ノードを許可します。
-uninstall	アンインストール処理の 1 ステップを実行します。
-wine	WINE (Linux) 回避策を強制的に使用します。

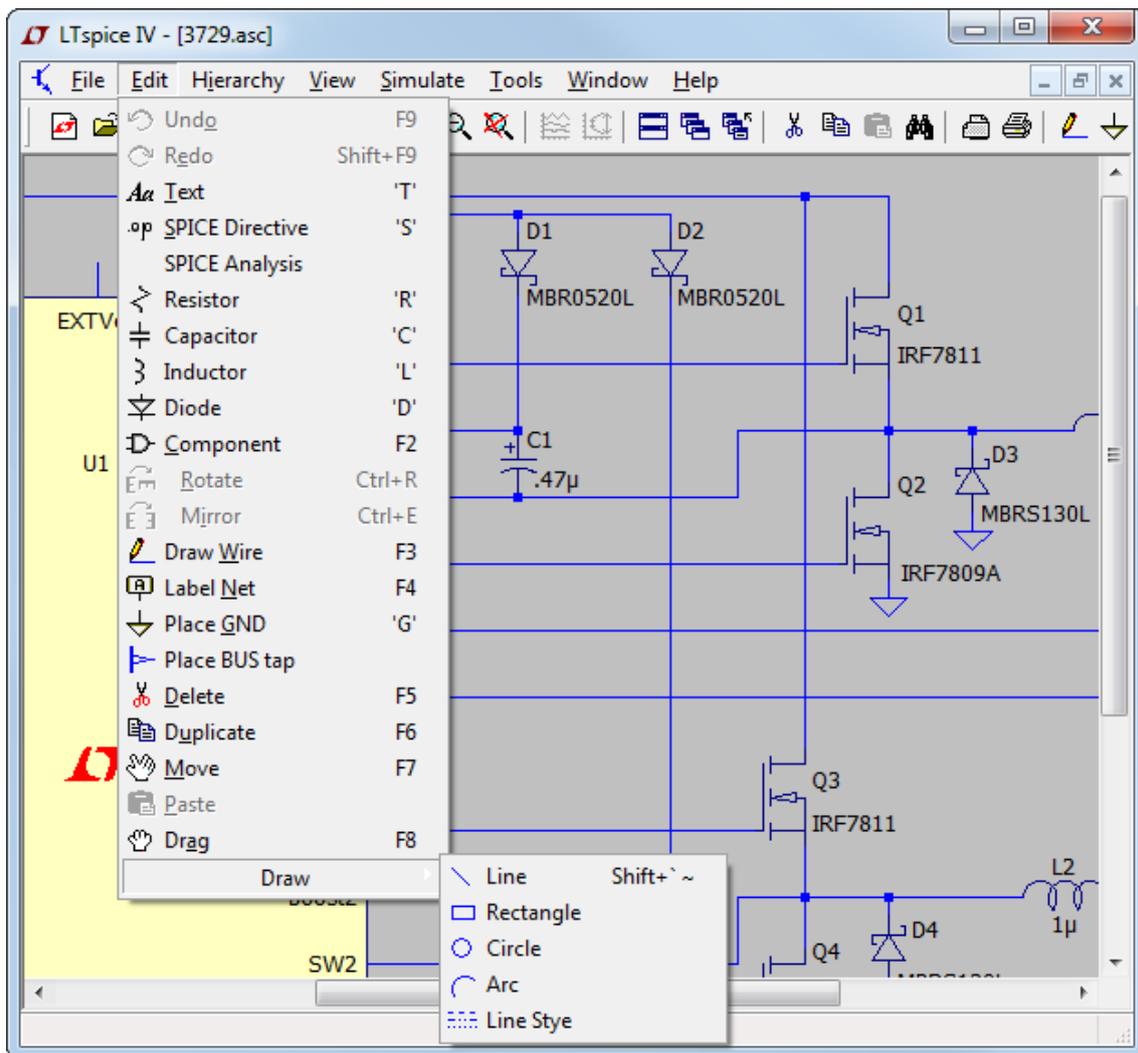
回路図の取り込み

LTspice IV は、汎用の回路図取り込みプログラムを内蔵しています。LTspice IV を使用すると、回路図の下書き、シンボルの作成、ネットリストの生成、シミュレーション・データの相互調査を実行できます。無制限の回路図サイズおよび階層がサポートされます。

基本的な回路図の編集

回路図取り込みプログラムは、新しい回路図を作成するか、用意された回路例を修正する目的で使用されます。回路サイズと階層の深さはコンピュータのリソースだけに制限されます。

このプログラムには、約 800 のシンボルが付属しています。これらのシンボルは、LTC のほとんどのパワーIC、オペアンプ、コンパレータ、および回路設計用の多くの汎用デバイスをカバーします。また、このプログラムにインポートするデバイスに対してオリジナルのシンボルを描くことができます。



多くの回路図取り込みプログラムと異なり、このプログラムは、明確に SPICE シミュレーションの実行を目的として作成されました。この意味は次のとおりです。オブジェクトをクリックした場合のデフォルトの動作は、該当する配線の電圧または部品に流れる電流をプロットすることであり、編集対象のオブジェクトを選択することや、それ以外の何らかの編集動作を選択して、実行直後のシミュレーションを無効にするすることではありません。従って、オブジェクトの移動、鏡映、回転、ドラッグ、または削除を行う場合は、最初に「Move」、「Drag」、または「Delete」コマンドを選択します。その後、オブジェクトをクリックすればそのオブジェクトを選択できます。オブジェクトを囲むボックスをドラッグすれば、複数のオブジェクトを選択できます。右マウス・ボタンをクリックする

か、Esc キーを押すまで、プログラムは移動、ドラッグ、または削除モードにとどまります。回路図の編集はすべて取消すかやり直すことができます。

Undo: 最後のコマンドを取り消します。

Redo: 最後の「Undo」コマンドをやり直します。

Text: 回路図にテキストを配置します。このコマンドが行うのは、回路図に情報を付記することだけです。このテキストが回路に電氣的な影響を及ぼすことはありません。

SPICE Directive: ネットリストに組み込まれるテキストを回路図上に配置します。このコマンドを実行すると、取り込み回路図と SPICE ネットリストが混合されます。このコマンドにより、シミュレーション・オプションの設定、モデルを含むファイルの組み込み、新しいモデルの定義、または他の有効な SPICE コマンドの使用が可能です。また、モデルのインスタンス(「X」で始まる SPICE コマンド)を回路図上に記述し、定義を含めることにより、このコマンドを使用して、シンボルのないサブ回路を実行できます。

SPICE Analysis: シミュレーション・コマンドを入力/編集します。

Resistor: 回路図に新しい抵抗を配置します。

Capacitor: 回路図に新しいコンデンサを配置します。

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: 回路図に新しい部品を配置します。このコマンドを実行すると、シンボル・データベースを参照してプレビューするダイアログが表示されます。これは、「Resistor」、「Capacitor」、「Inductor」、および「Diode」コマンドのより一般的な形式です。

Rotate: スプライト状態のオブジェクトを回転します。スプライト状態のオブジェクトがない場合、このコマンドは灰色表示されることに注意してください。

Mirror: スプライト状態のオブジェクトを鏡映表示します。スプライト状態のオブジェクトがない場合、このコマンドは灰色表示されることに注意してください。

Draw Wire: 左マウス・ボタンをクリックして、素線の描画を開始します。マウスをクリックするたびに、新しい素線部分が定義されます。既存の素線に新しい素線を連結するには、既存の素線部分をクリックします。現在の素線を取り消すには、1 回右クリックします。このコマンドを中止

するには、もう一度右クリックします。抵抗などの部品を貫いて素線を引くことができます。抵抗が素線と直列になるように、素線は自動的に切り取られます。

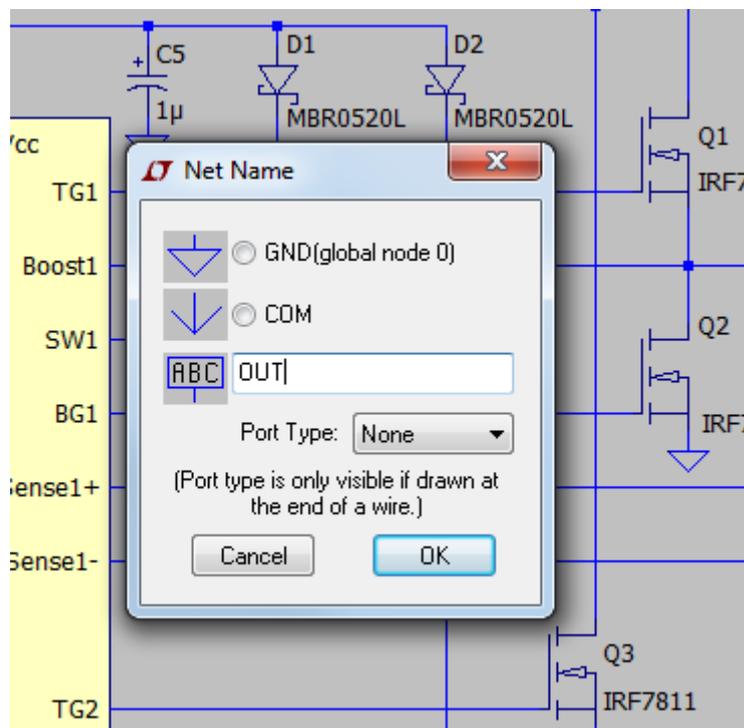
- Label Net: ノードの名前を指定して、このノードのネットリスタによって勝手な名前が生成されないようにします。
- Place GND: GROUND のシンボルを配置します。これはノード「0」で、グローバルな回路共通ノードです。
- Delete: オブジェクトをクリックするかオブジェクトを囲むボックスをドラッグすることにより、オブジェクトを削除します。
- Duplicate: オブジェクトをクリックするかオブジェクトを囲むボックスをドラッグすることにより、オブジェクトを複製します。LTspice IV の 1 回の起動中に 2 つの回路図を同時に開いている場合は、一方の回路図からもう一方の回路図へコピーすることができます。最初の回路図のウィンドウで「Duplicate」コマンドを起動します。次に、2 番目の回路図をアクティブ・ウィンドウにして、Ctrl-V を押します。
- Move: 移動するオブジェクトをクリックするか、オブジェクトを囲むボックスをドラッグします。これにより、該当のオブジェクトを新しい場所に移動できます。
- Paste: 「Duplicate」コマンドにより、オブジェクトが既に選択されていた場合、このコマンドは新しい回路図のウィンドウで有効になります。
- Drag: ドラッグするオブジェクトをクリックするか、オブジェクトを囲むボックスをドラッグします。これにより、該当のオブジェクトを新しい場所に移動できます。また、オブジェクトについての素線は、新しい場所への移動に合わせて伸縮します。
- Draw=>Line: 回路図上に線を引きます。こうした線は回路に対して電氣的な影響を及ぼしませんが、回路に注釈を付けるのに役立つことがあります。
- Draw=>Rectangle: 回路図上に四角形を描きます。この四角形は回路に対して電氣的な影響を及ぼしませんが、回路に注釈を付けるのに役立つことがあります。
- Draw=>Circle: 回路図上に円を描きます。この円は回路に対して電氣的な影響を及ぼしませんが、回路に注釈を付けるのに役立つことがあります。
- Draw=>Arc: 回路図上に円弧を描きます。この円弧は回路に対して電氣的な影響を及ぼしませんが、回路に注釈を付けるのに役立つことがあります。

注記:回路図に対する図式の注釈(線、四角形、円、および円弧)は、デフォルトでは、素線とピンの電氣的接
触に使用されるのと同じグリッドに収まります。このスナップ指定を無効にするには、注釈を配置するとき Ctrl
キーを押したままにしてください。

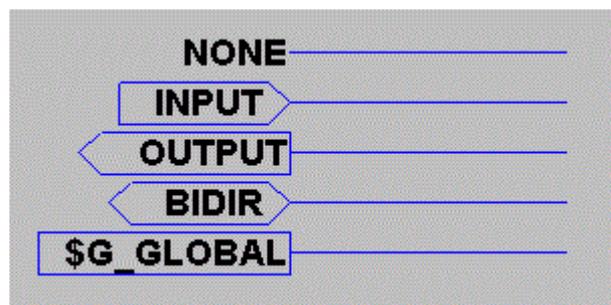
ノード名のラベル付け

回路内の各ノードには固有の名前が必要です。ノードの名前を指定して、ネットリスタによって勝手な名前が生成されないようにすることができます。ノード「0」は回路のグローバル・グランドであり、名前「0」ではなく特別なグラフィック・シンボルで描かれます。ノード「COM」に対して定義されたグラフィック・シンボルもありますが、このノードに特別な意味はありません。つまり、このノードは SPICE 全体の共通ノードではなく、グローバル・ノードでもありません。グランドとは異なるノードにグラフィック・シンボルを関連付けると、ちょうど便利なことがあります。

「\$G_VDD」など、文字「\$G_」で始まる名前をノードに付けると、その名前が回路階層内のどこにあっても、そのノードはグローバル・ノードになります。

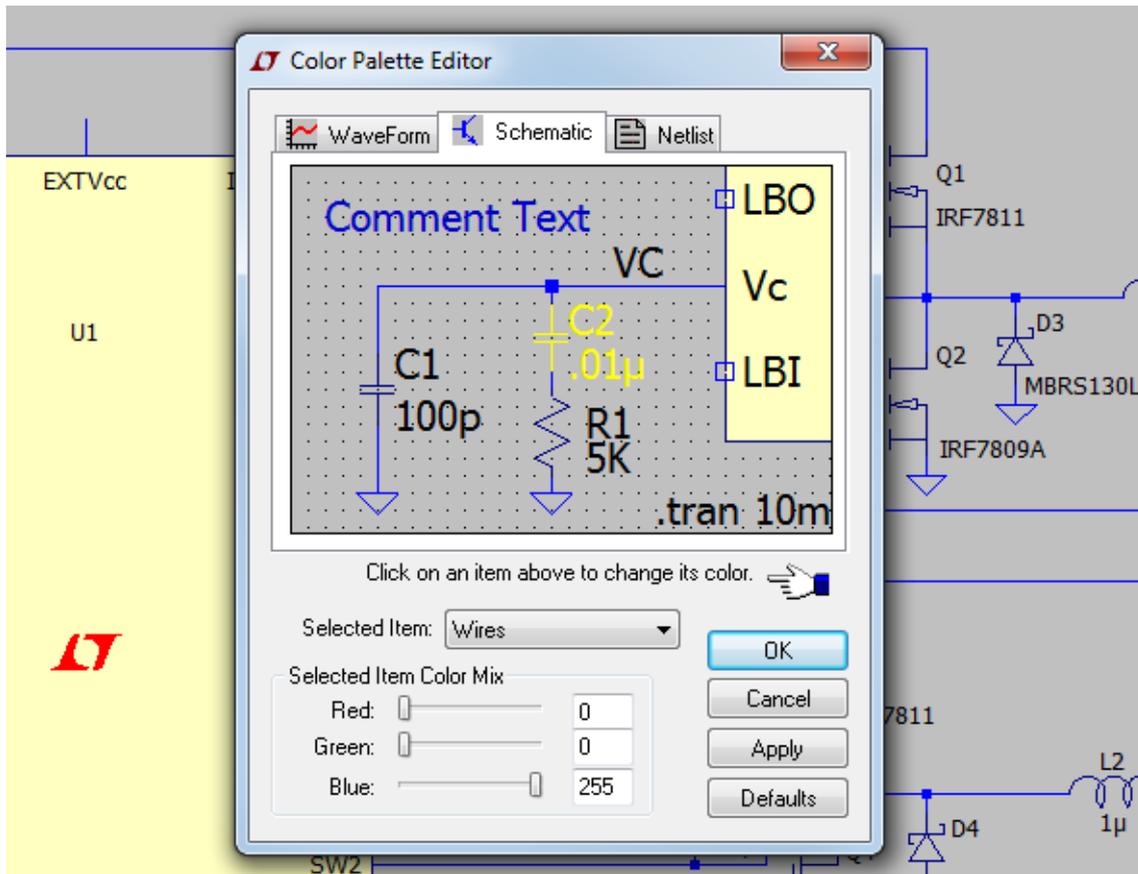


ノードが入力、出力、または双方向のうちどのタイプのポートであるかを示すことができます。これらのポート・タイプの描画方法は異なりますが、ネットリスタにとっては意味がありません。ポート・タイプを示すと、回路が分かりやすくなります。また、グローバル・ノードの描画方法も、名前のまわりにボックスが描かれるという点で異なります。



回路図の色

メニューコマンド「Tools」=>「Color Preferences」を使用すると、回路図を表示するとき使用する色を設定できます。サンプル回路図内のオブジェクトをクリックし、赤のスライダ、緑のスライダ、および青のスライダを使用して、好みに合わせて色を調整します。

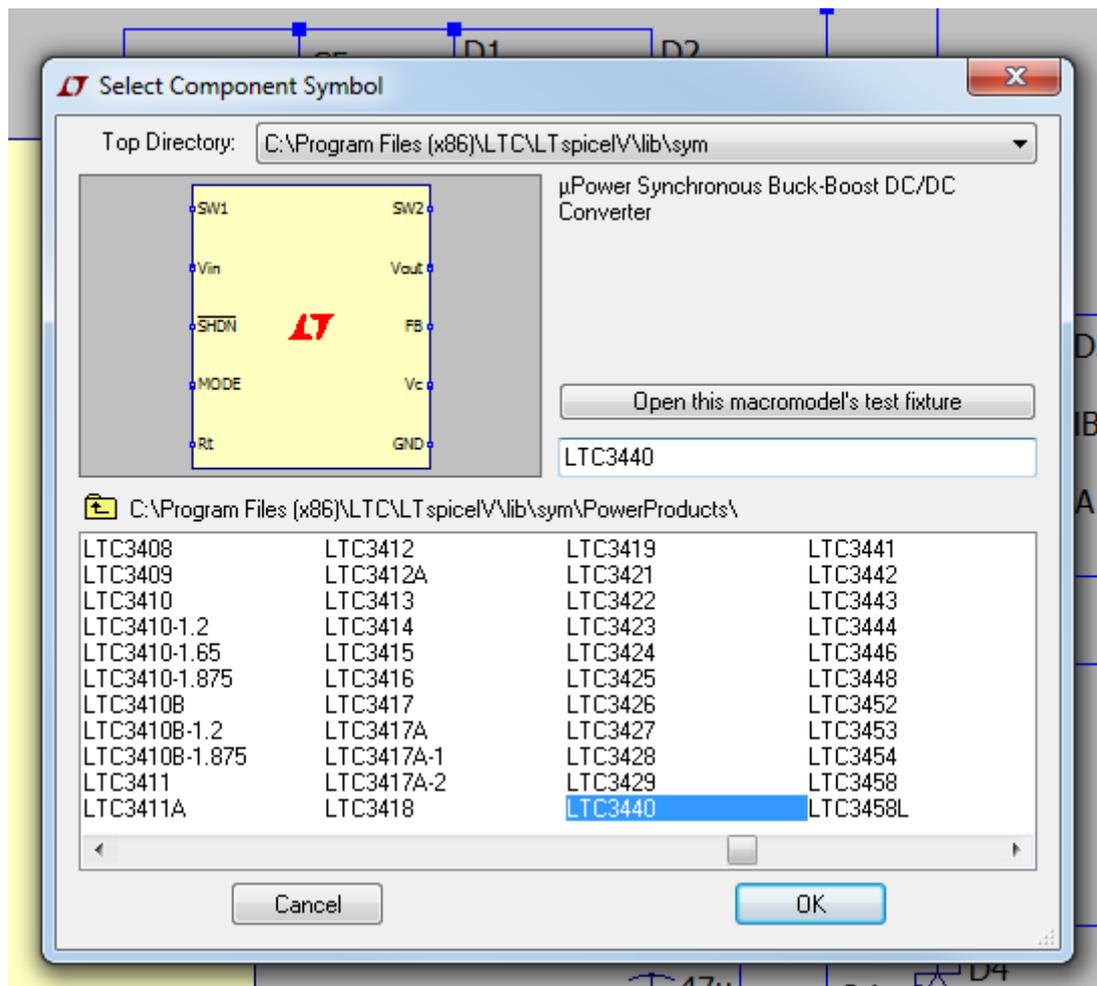


注記: 線や円など、回路図に作成した電気的には関係ない図式の注釈は、部品本体と同じ色になります。

新しい部品の配置

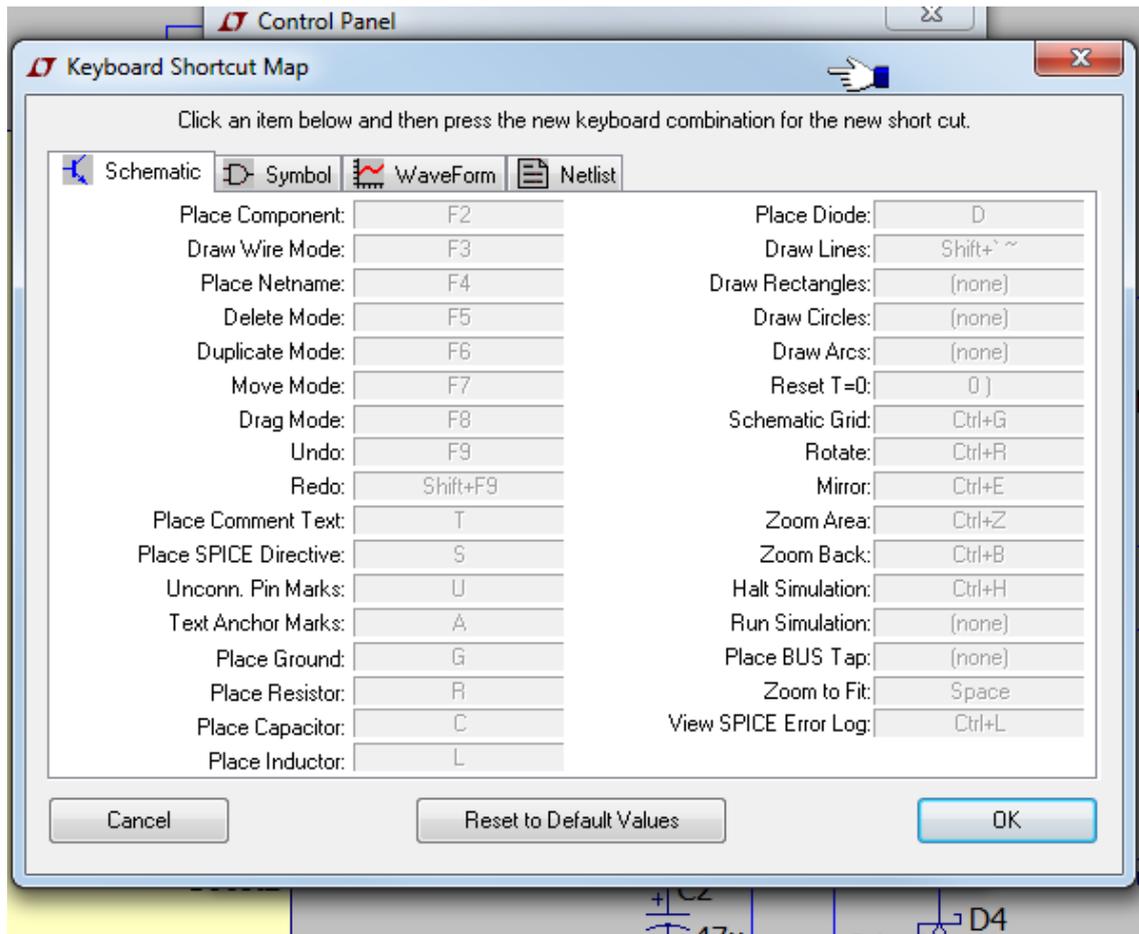
抵抗、コンデンサ、インダクタなど、よく使用される特定の部品をツールバー・ボタンを使用して選択し、回路図に配置できます。

ほとんどのシンボルについては、メニューコマンド「Edit」=>「Component」を使用してダイアログを起動し、目的のデバイスを参照します。



キーボード・ショートカットのプログラミング

メニューコマンド「Tools」=>「Control Panel」=>「Drafting Options」=>「Hot Keys」を使用すると、ほとんどのコマンドについてキーボード・ショートカットを設定できます。コマンドを単にマウス・クリックし、その後、そのコマンドに対してコード記述の対象にするキーまたはキーの組み合わせを押します。ショートカットを削除するには、該当のコマンドをクリックして、「Delete」キーを押します。



PCB ネットリストの抽出

回路図のメニューコマンド「Tools」=>「Export Netlist」を使用すると、PCB レイアウト用の ASCII ネットリストを生成できます。ピン・ネットリストの順序が同じ一連のシンボルを作成する必要があることに注意してください。たとえば、LTspice 回路図のネットリストを [xpressPCB](#) にインポートする場合は、LTspice と ExpressPCB の両方に対して、使用するすべてのシンボルのネットリストの順序が同じ一連のシンボルを作成する必要があります。そうしないと、ダイオードのネットリストが逆向きに作成される可能性や、トランジスタのリード接続が混乱する可能性があります。

以下のフォーマットを使用できます。

Accel、Algorex、Allegro、Applicon Bravo、Applicon Leap、Cadnetix、Calay、Calay90、CBDS、Computervision、EE Designer、ExpressPCB、Intergraph、Mentor、Multiwire、PADS、Scicards、Tango、Telesis、Vectron、および Wire List です。

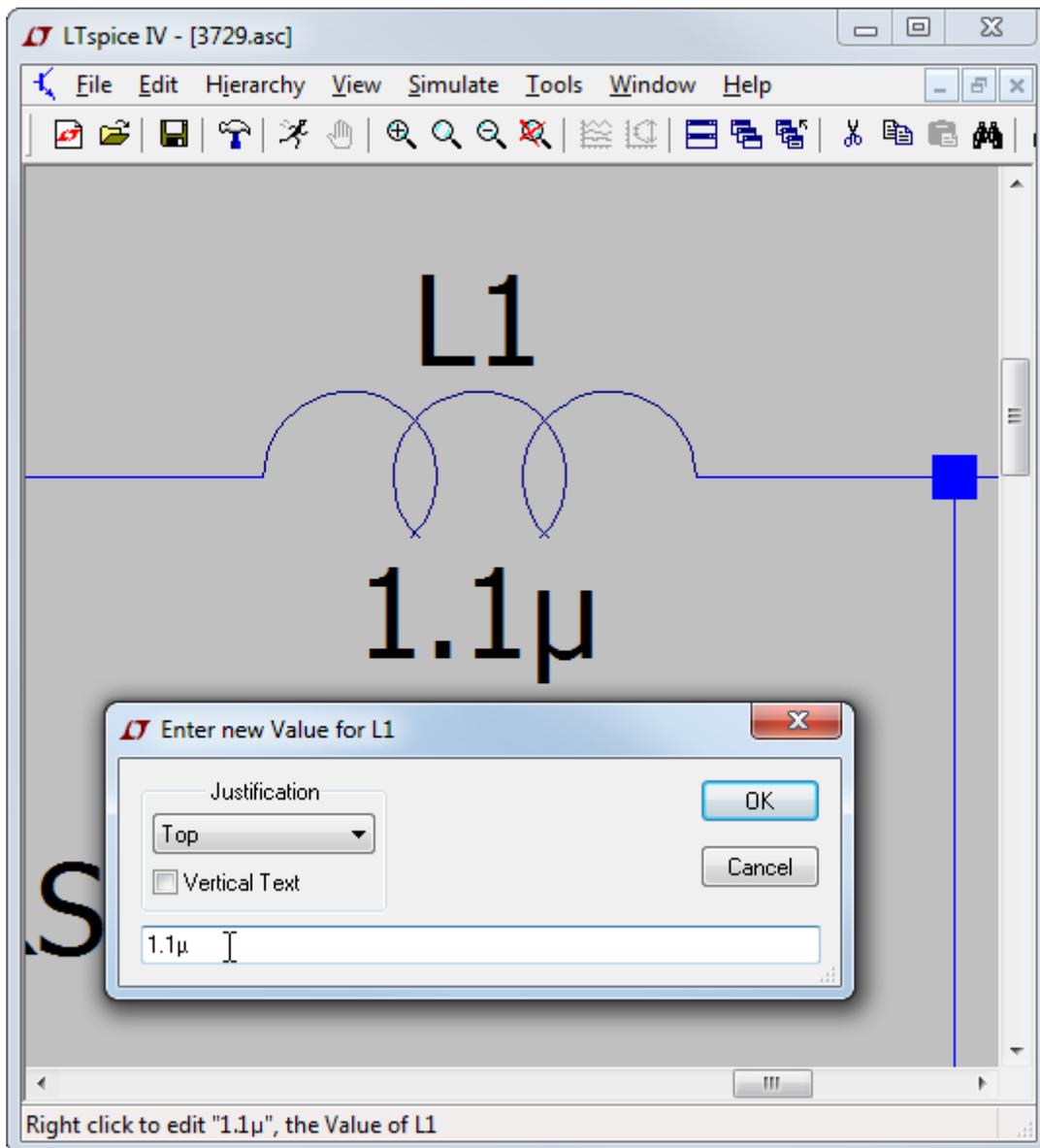
部品の編集

部品は部品のタイプによって、2 つまたは 3 つの異なる方法で編集できます。

- 1) 表示可能なほとんどの部品属性フィールドは、フィールドをマウスでポイントして右クリックすれば編集できます。マウス・カーソルは、テキストをポイントしているときはテキスト脱字記号に変わります。
- 2) 抵抗、コンデンサ、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、独立した電圧源、独立した電流源、階層回路ブロックなどの多くの部品タイプには、特別なエディタがあります。これらのエディタは、デバイスの適切なデータベースにアクセスできます。これらのエディタを使用するには、部品の本体をマウスで右クリックします。
- 3) シンボルの上にマウスを置き、Ctrl キーを押したまま右マウス・ボタンをクリックしてください。使用可能なすべてのシンボル属性を表示するダイアログボックスが表示されます。各フィールドの横には、回路図上でフィールドを表示するかどうかを指定するチェックボックスがあります。

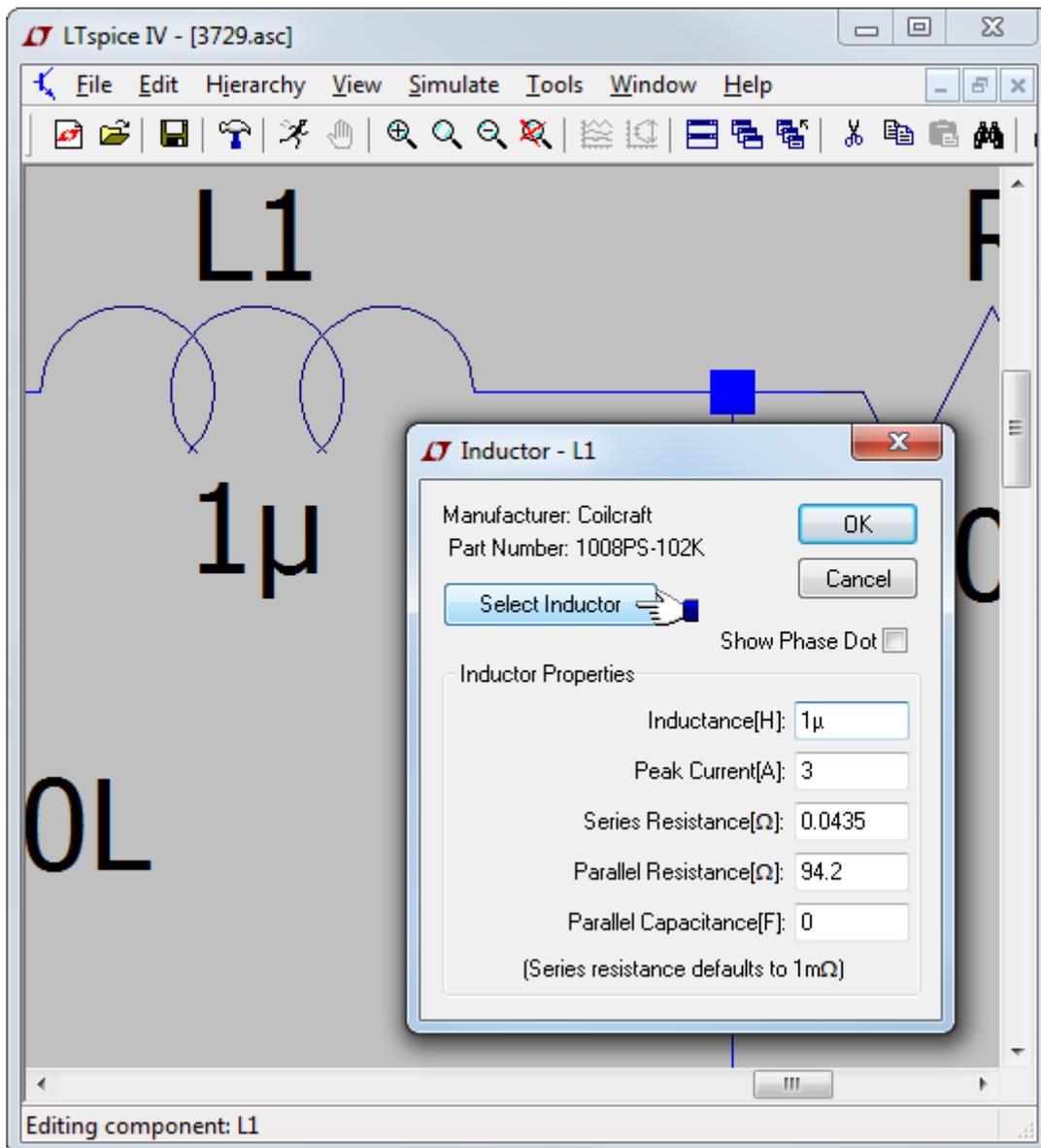
可視属性の編集

表示可能なほとんどの部品属性フィールドは、フィールドをマウスでポイントして右クリックすれば編集できます。マウス・カーソルは、テキストをポイントしているときはテキスト脱字記号に変わります。これは部品の値を変更するのに便利な方法です。



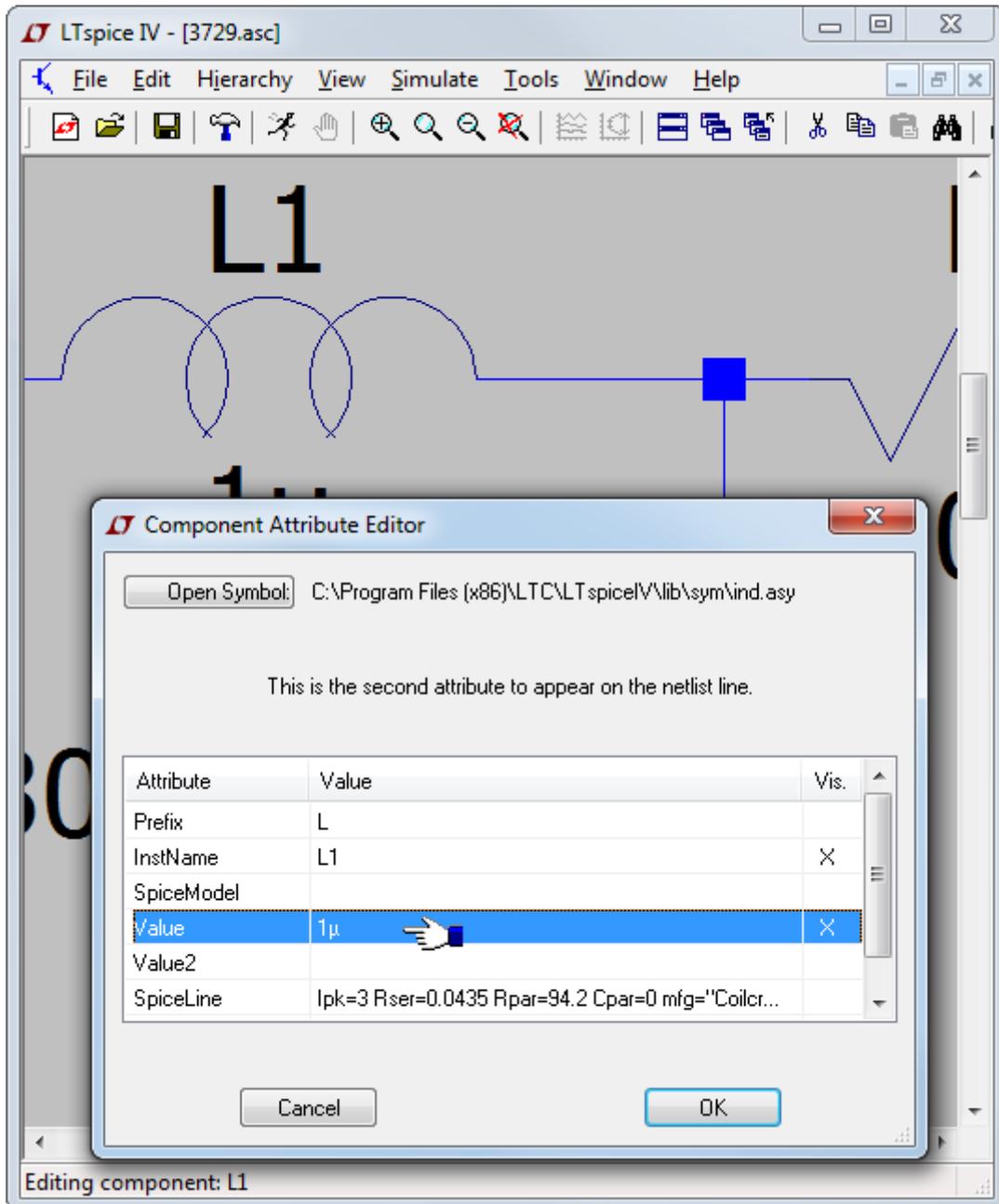
専用部品エディタ

抵抗、コンデンサ、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、独立した電圧源、独立した電流源、階層回路ブロックなどの多くの部品タイプには、特別なエディタがあります。これらのエディタは、関連部品の適切なデータベースにアクセスできます。これらのエディタを使用するには、部品の本体をマウスで右クリックします。



一般属性エディタ

使用可能なすべての部品の属性に直接アクセスして、その内容と可視性を編集することが望ましいことがあります。これを実行できるエディタを起動するには、シンボルの本体の上にマウスを置き、Ctrl キーを押したまま右マウス・ボタンをクリックします。使用可能なすべての記号属性を表示するダイアログボックスが表示されます。各フィールドの横には、回路図上でフィールドを表示するかどうかを指定するチェックボックスがあります。



属性 SpiceModel、Value、Value2、SpiceLine、および SpiceLine2 は、すべて部品全体の値の一部です。部品を SPICE に対してネットリスト化する方法の観点から、部品によって生成される 1 行の SPICE コードは以下のようになります。

```
<name> node1 node2 [...]<SpiceModel>
+ <Value> <Value2> <SpiceLine> <SpiceLine2>
```

接頭部属性文字は、参照指定子の最初の文字と異なる場合、参照指定子の前に付加されます。この場合、Prefix 文字と InstName は、「§」という文字で区切られます。たとえば、Prefix 属性が「M」で InstName 属性が「Q1」である場合、ネットリスト内の名前は M § Q1 になります。これにより、先頭文字が SPICE の場合とは異なる参照指定子を使用して、デバイスのタイプを識別できます。

上記の規則には例外が 3 つあります。回路素子に変換されないある特殊なシンボル(ジャンパ)がありますが、これはネットリスト・ジェネレータへの指令であり、電氣的に同一のノードに 2 つの異なる名前があるものです。もう 1 つの例外は、接頭部が「X」となるよう定義されたシンボルで、Value 属性と Value2 属性の両方が定義されています。こうした部品は、2 行の SPICE コードとして次のようにネットリスト化されます。

```
.lib <SpiceModel>
<name> node1 node2 [...]<Value2>
```

これにより、部品に呼び出されるサブ回路の定義を収録しているライブラリを自動的に組み込むシンボルを定義できます。ネットリスト・コンパイラは、重複した.lib 文を削除します。こうした部品は回路図上で編集できないことに注意してください。3 番目の例外は、接頭部が「X」となるよう定義され、ModelFile 属性が定義されたシンボルです。

こうした部品も、2 行の SPICE コードとして次のようにネットリスト化されます。

```
.lib <ModelFile>
<name> node1 node2 [...]<SpiceModel> <Value> <Value2> <SpiceLine> <SpiceLine2>
```

ライブラリ・ファイルを自動的に組み込むが、このシンボルのインスタンスを引き続き編集可能な状態にしておく場合は、この方法を使用してください。シンボル属性 SpiceModel が存在し、この属性が、<ModelFile>と指定されたファイルでサブ回路の名前になっている場合は、シンボルのインスタンスを回路図上で編集するときに、すべてのサブ回路名のドロップ・リストを使用できます。

新しいシンボルの作成

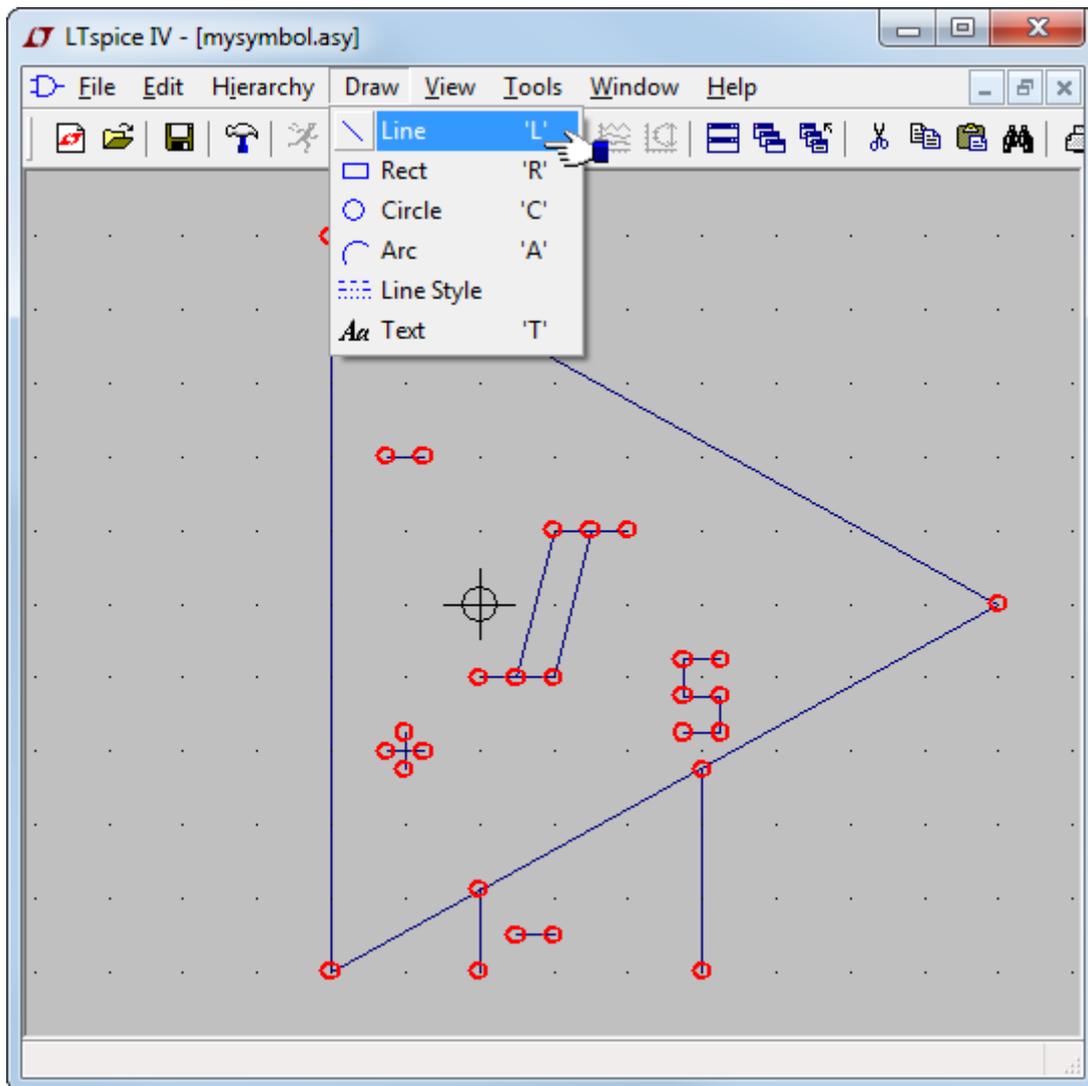
シンボルが表現できるのは、抵抗やコンデンサなどの基本デバイス、個別のファイル内でライブラリ化されたサブ回路、あるいは回路図の別のページです。このセクションでは、独自の新しいシンボルを定義する方法について説明します。新しいシンボルの作成から始めるには、メニューコマンド「File」=>「New Symbol」を使用します。

注記:シンボルの編集中は画面の更新が遅くなることがあります。これがビデオ・カードに関する問題である場合は、シンボル編集ウィンドウの面積を縮小して画面の再描画速度を高めるか、画面の色解像度を低くします。こうすると、マウスの動きに対する応答がより巧妙になります。

本体の描画

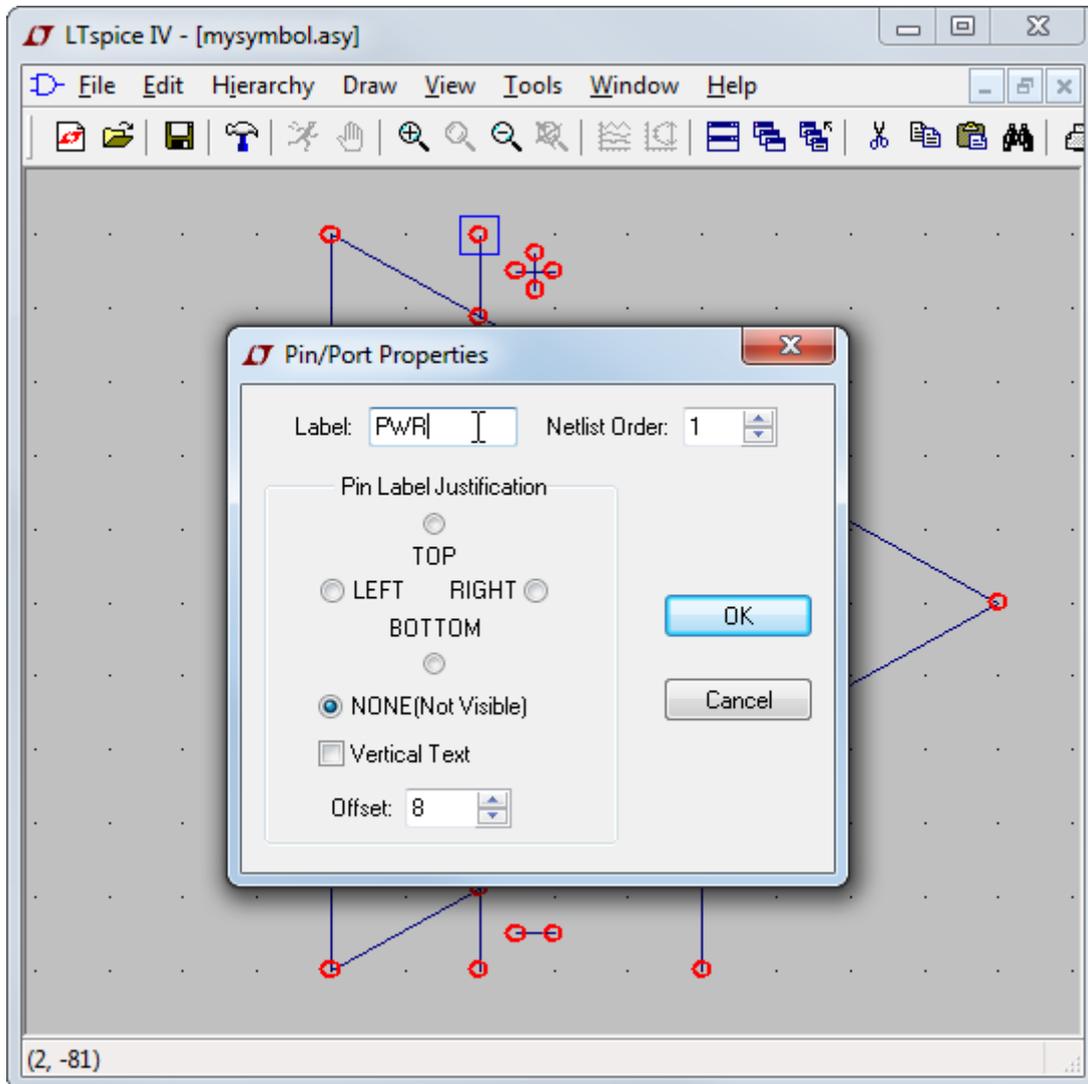
シンボルの本体を一連の線、四角形、円、および円弧として描きます。これらのオブジェクトが回路に電気的な影響を及ぼすことはありません。また、回路に影響を及ぼさない「Draw」=>「Text」コマンドを使用してシンボルにテキストを描くこともできます。これらのオブジェクトのアンカ・ポイントは小さな赤い円で描かれるので、アンカ・ポイントを囲んでドラッグする場合につかむ場所が分かります。

メニューコマンド「View」=>「Mark Object Anchors」を使用して、赤いマーカーのオフとオンを切り替えることができます。



ピンの追加

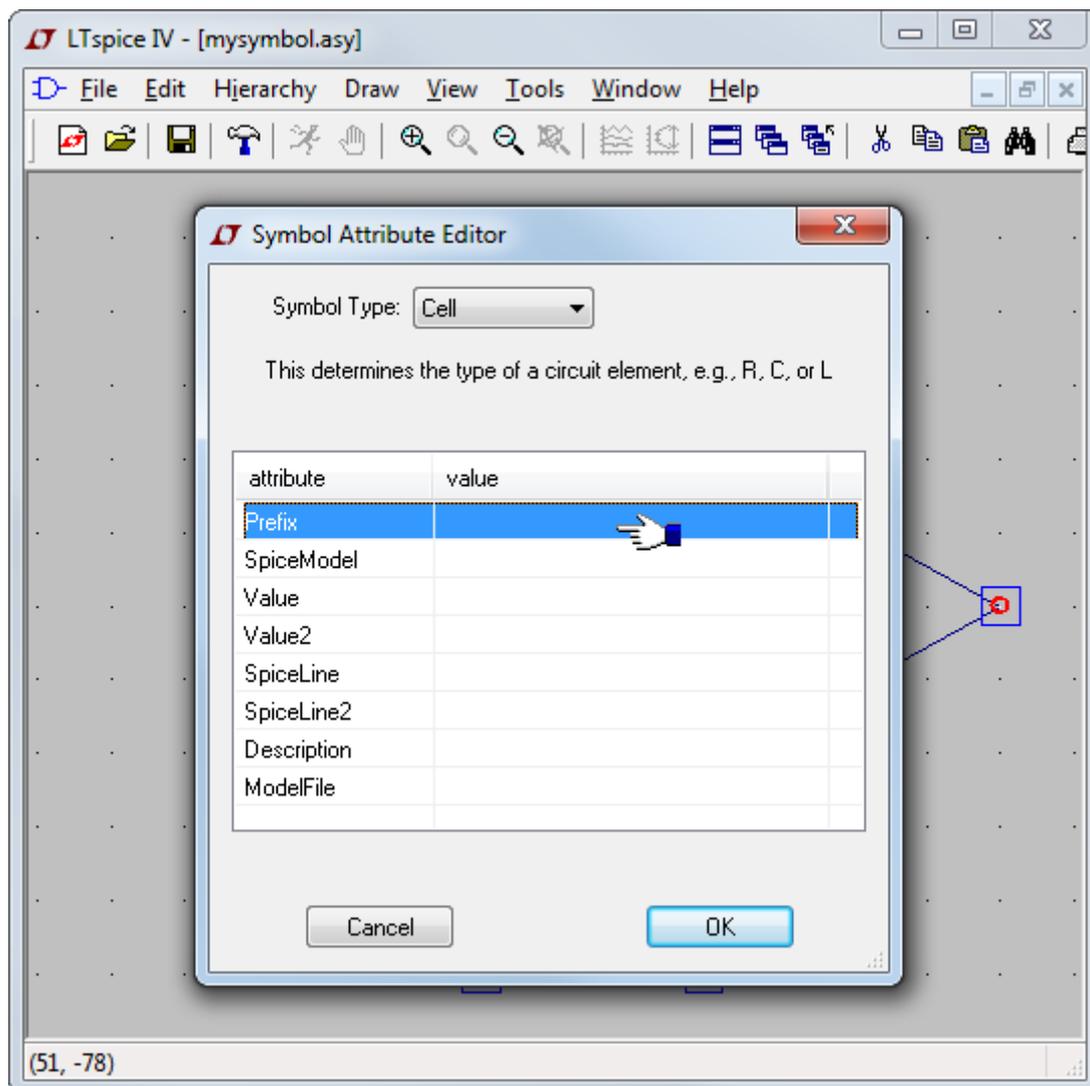
ピンはシンボルへの電氣的接続が可能です。メニューコマンド「Edit」=>「Add Pin/Port」を使用して、新しいピンを追加します。



「Pin Label Position」では、ピン・ラベルの提示法が決まります。「TOP」、「BOTTOM」、「LEFT」、および「RIGHT」は、テキストの位置揃えです。たとえば、ピン・ラベルが「TOP」で位置揃えされた場合、ピン(ラベルのテキスト位置揃えのアンカ・ポイント)はラベルの上になります。シンボルが SPICE の基本要素またはライブラリのサブ回路を表わす場合、ピン・ラベルが電氣的な影響を回路に直接及ぼすことはありません。ただし、シンボルが階層回路図の下位の回路図を表わす場合、ピン名は下位の回路図でのネットの名前として有効です。「Netlist Order」は、このピンが SPICE に対してネットリスト化される順序を決定します。

属性の追加

メニューコマンド「Edit」=>「Attributes」=>「Edit Attributes」を使用して、シンボルのデフォルトの属性を定義できます。最も重要な属性は「Prefix」と呼ばれます。この属性はシンボルの基本タイプを決定します。シンボルが SPICE の基本要素を表わすことを目的としている場合、シンボルには抵抗の R、コンデンサの C、MOSFET の M など、適切な接頭部があります。利用可能なすべての SPICE 基本要素については、LTspice の参考資料を参照してください。ライブラリ内で定義されたサブ回路をシンボルを使用して表わす場合は、接頭部を「X」にします。



回路図内の部品としてのシンボルのインスタンスでは、シンボルの属性を無効にすることができます。たとえば、MOSFET のシンボルがあり、その接頭部属性が「M」である場合、代わりにトランジスタをサブ回路としてモデル化できるようにするため、インスタンスごとに接頭部を「X」に変更することができます。シンボルを使用するすべての回路図に必要なライブラリを自動的に組み込む特別な属性の組み合わせがあります。

Prefix: X

SpiceModel: <SPICE モデルが含まれているファイルの名前>
Value: <回路図上で表示対象にするもの>
Value2: <ネットリスト内で必要とする値>

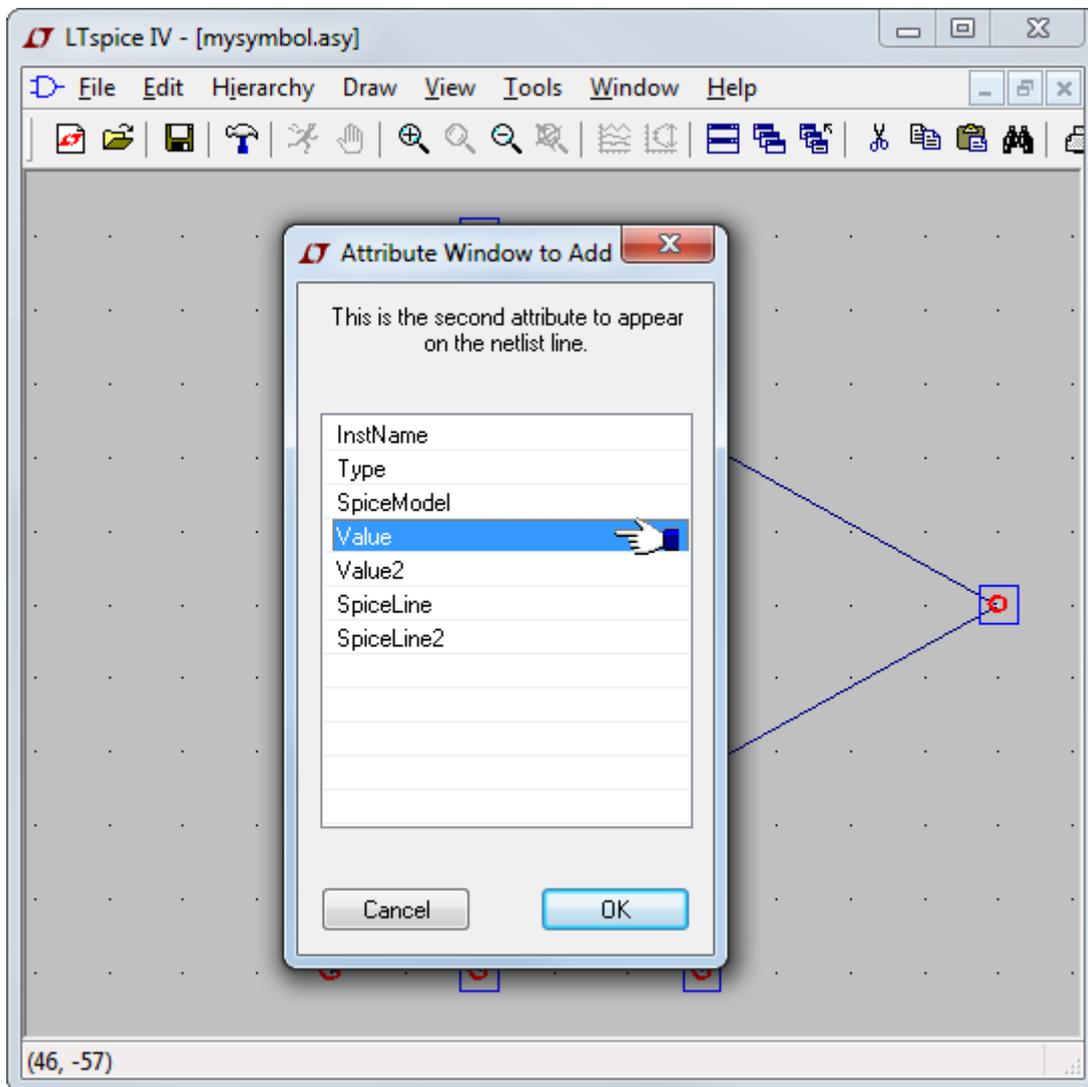
Value2 は SPICE モデルが含まれているファイルに定義されたサブ回路名と一致するよう指定します。また、サブ回路に追加のパラメータを渡すことができます。シンボルをこのように定義した場合は、回路図上の部品としてのシンボルのインスタンスを編集して別の属性にすることはできません。

シンボルが階層回路図の別のページを表すようにする場合は、すべての属性を空白のまま残し、シンボル・タイプを「Cell」から「Block」に変更する必要があります。属性値を設定する必要はありません。

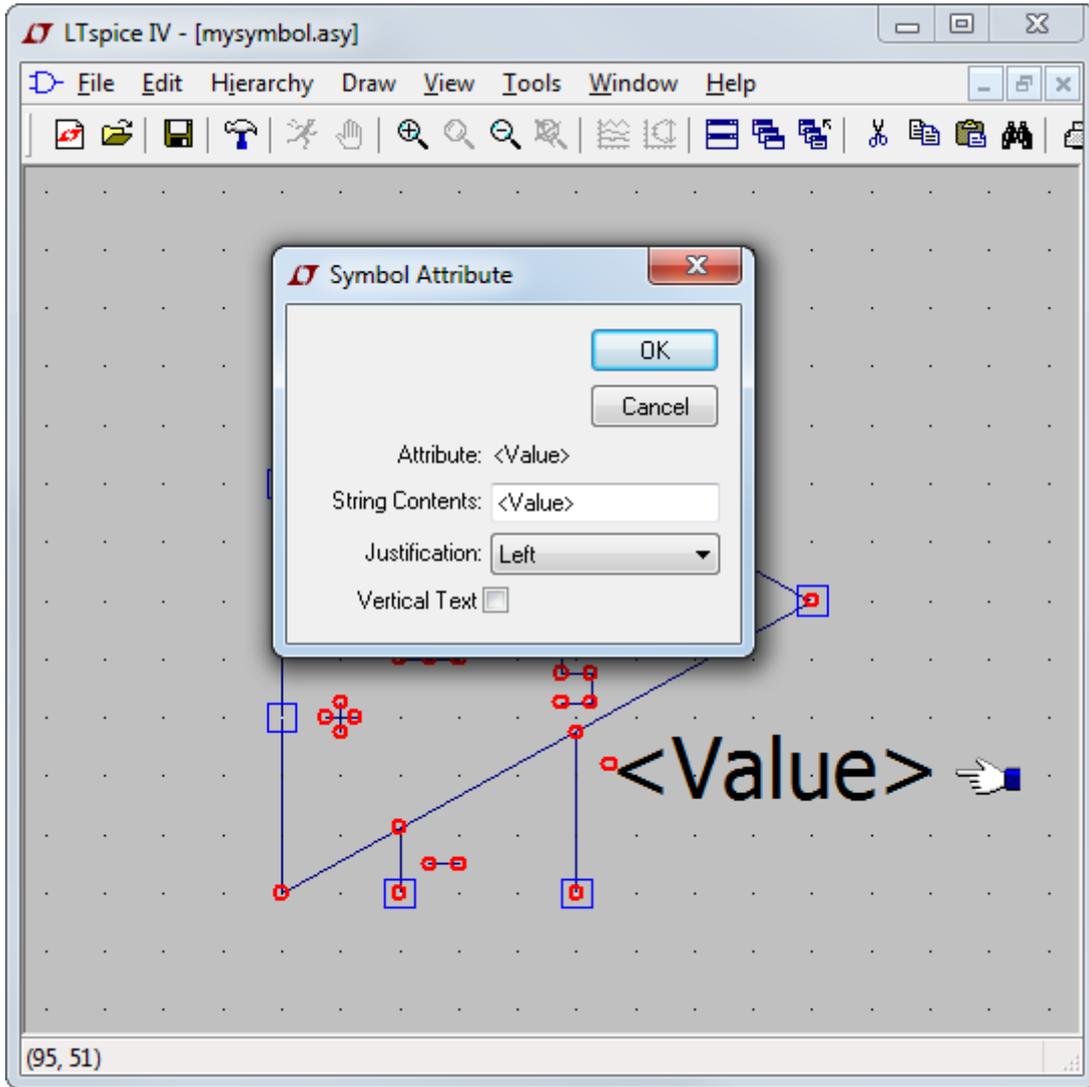
指定される場合があるシンボル属性 (ModelFile) があります。これは、ライブラリとしてネットリストに組み込まれるファイルの名前として使用されます。接頭部属性が「X」であり、モデル・ファイルで定義されたサブ回路であると定義されているシンボル属性 SpiceModel がある場合、シンボルのインスタンスを回路図上で編集するときに、すべてのサブ回路名のドロップ・リストを使用できます。

属性の可視性

メニューコマンド「Edit」=>「Attributes」=>「Attribute Window」を使用して、属性の可視性を編集できます。



このダイアログを使用して属性を選択したら、属性をシンボルに対して希望どおりに配置できます。属性のテキストをマウスで右クリックすることにより、既に表示可能にした属性のテキスト位置揃えおよび内容を修正できます。



シンボルの自動生成

シンボルは 2 つの状況で自動的に生成できます。

回路図を編集する場合は、メニュー項目「Hierarchy」=>「Open this Sheet's Symbol」を実行できます。シンボルが見つからない場合、LTspice はシンボルを自動的に生成するかどうか尋ねます。

その後、このシンボルを使用して、上位回路図でこの回路のシートを呼び出すことができます。

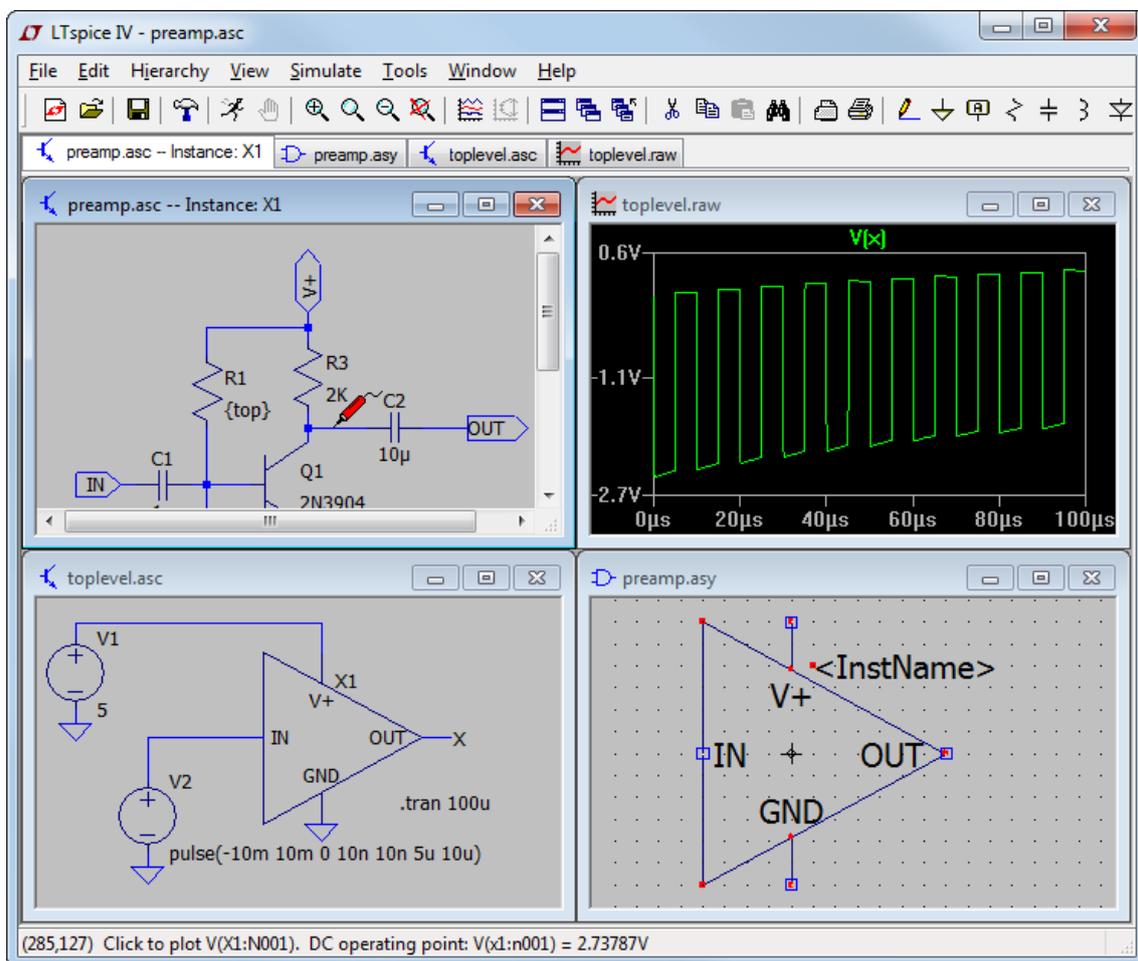
また、サブ回路の定義が入っている ASCII ネットリストを編集する場合には、サブ回路の名前が含まれている行にカーソルを置き、右クリックして、コンテキスト・メニュー項目「Create Symbol」を実行します。サブ回路として定義されたサードパーティ・モデルを追加する場合、ほとんどのユーザにとっては、これが検討に値する唯一の方法です。詳細が漏れなく処理されるからです。

階層

階層回路図の下書きには強力な利点があります。かなり大規模な回路の下書きを 1 枚の回路図に収めることができると同時に、小さな回路図の明瞭さを維持することができます。繰り返しの回路は、抽象的な方法で容易に処理されます。回路のブロックはライブラリ化して、後で別のプロジェクトで使用することができます。

階層の規則

別の回路図を上位回路図内のブロックとして参照する方法は、ブロック回路図と同じ名前でシンボルを作成し、その後そのシンボルを上位回路図に配置することです。たとえば、topXYZ.asc と呼ばれる最上位の回路図と、topXYZ の回路図に配置したい preamp.asc と呼ばれる別の回路図のファイルがある場合は、preamp.asy と呼ばれるシンボルを作成し、そのシンボルのインスタンスを topXYZ の回路図に配置します。回路図間の電気的な接続を確立するには、下位回路図内にあるノードの名前と一致する下位ブロックのシンボルのピンに上位回路図の素線を接続します。回路図ブロックとして使用したシンボルの名前と、それらのブロックに対応する回路図の名前は、ファイル名として使用できる有効な文字で構成する必要があります。回路図の名前には空白文字も使用できません。



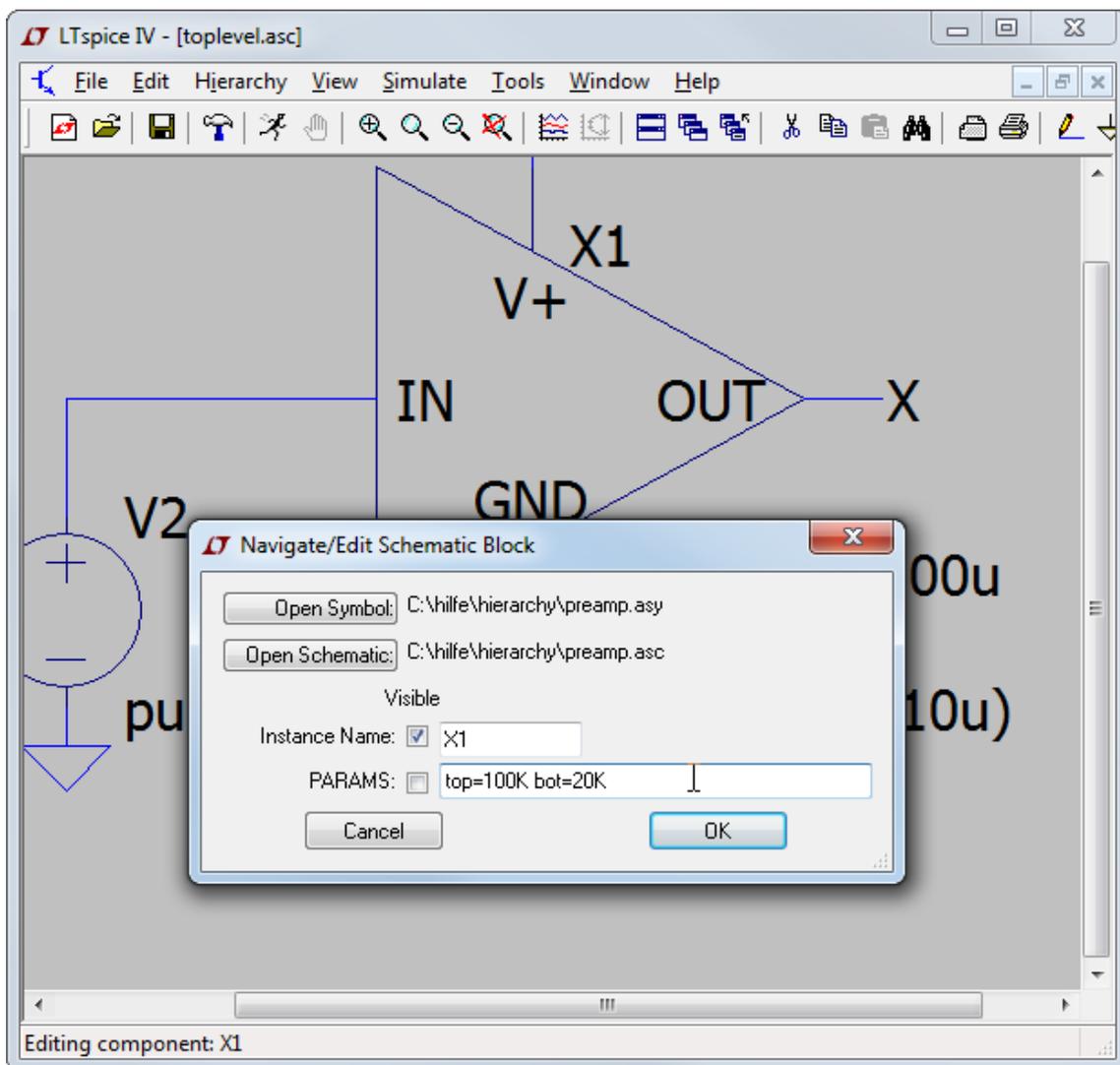
LTspice は、シンボルおよびブロックの最上位の回路図のディレクトリ内を調べて、最上位の回路図の回路を完成させます。

下位回路図のブロックを表わすために作成するシンボルには、属性を定義しないでください。

階層間の移動

「File」=>「Open」コマンドを使用して開いたすべてのファイルは最上位の回路図とみなされます。特定のブロックに SPICE 指令を追加し、そのブロックと参照先の下位回路図だけを使用してシミュレーションを実行できます。

ある回路図のブロックを上位回路図のブロックのインスタンスとして開くには、最初に上位回路図を開き、次にそのブロックを呼び出すシンボルのインスタンスの本体にマウスを移動します。そのシンボルのインスタンスの本体でマウスを右クリックすると、回路図を開くための特別なダイアログが表示されます。このように回路図を開くと、ブロック内のノードおよび電流を相互に調査することができます。「Control Panel」の「Save Defaults Pane」で、オプション「Save Subcircuit Node Voltages」および「Save Subcircuit Device Currents」チェックボックスをオンにしておくことに注意してください。最上位の回路図でノードを強調表示していた場合は、下位ブロックでもそのノードが強調表示されます。



このダイアログでは、preamp.asc 内の回路のこのインスタンスに渡すパラメータを入力することもできます。

波形ビューワ

LTspice IV は、シミュレーション・データのプロット方法を完全に制御できる波形ビューワを内蔵しています。

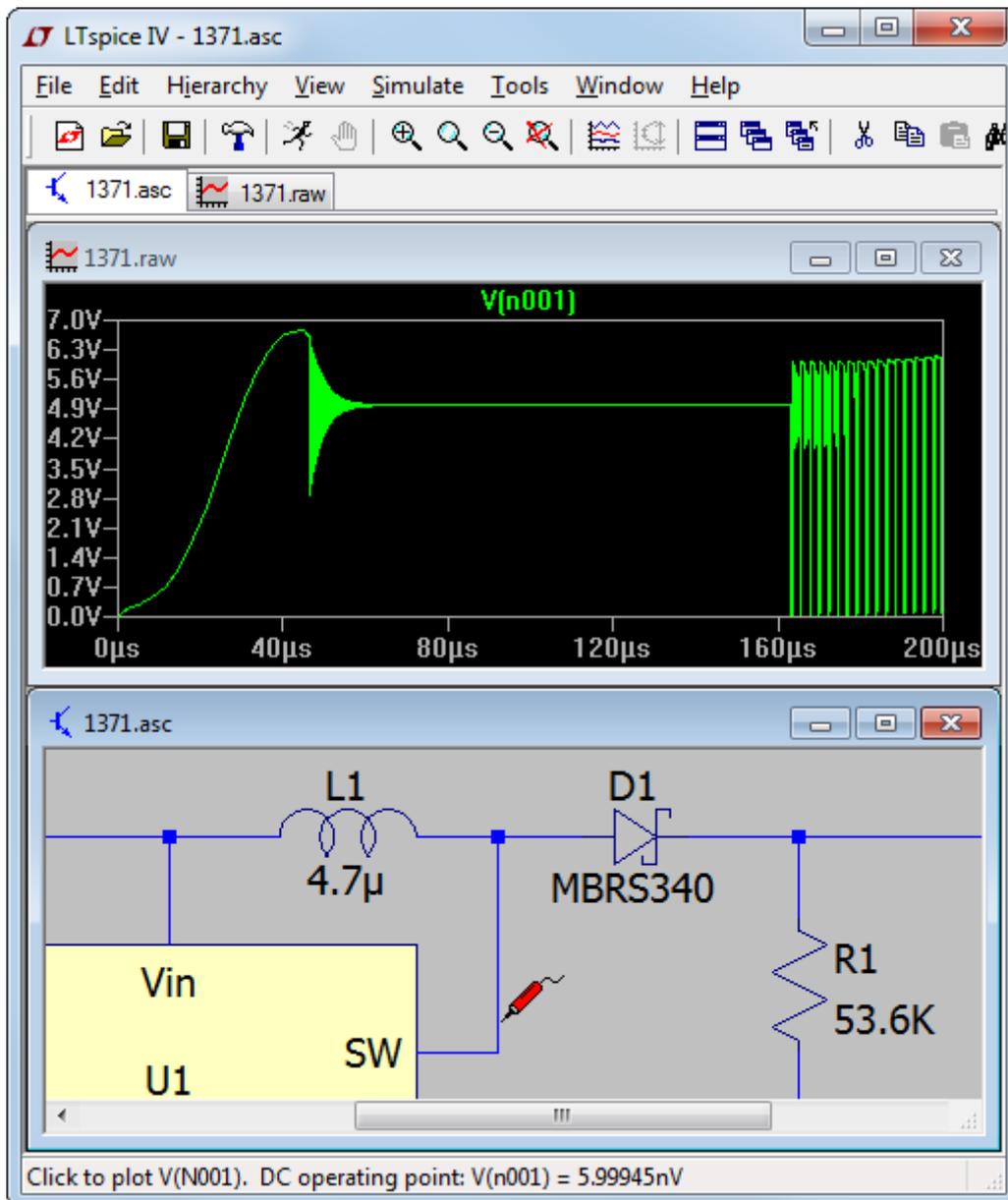
データ・トレースの選択

プロットされたトレースを選択する方法は基本的に3つあります。

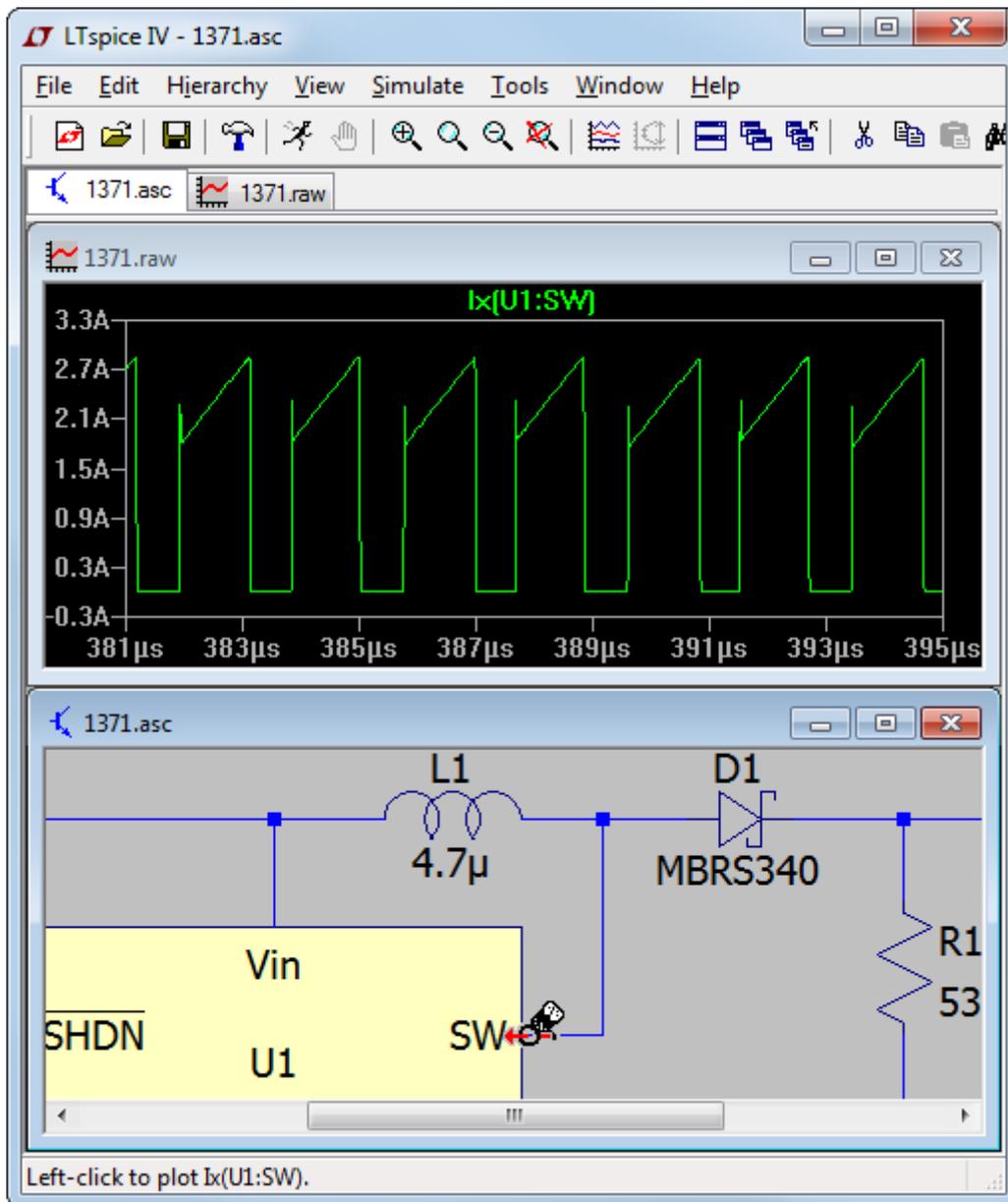
1. 回路図から直接プロービング
2. メニューコマンド「Plot Settings」=>「Visible Traces」
3. メニューコマンド「Plot Settings」=>「Add Trace」

undo コマンドおよび redo コマンドを使用すると、選択方法がどれであっても別のトレース選択方法で再調査できます。

1. 回路図から直接プロービング:

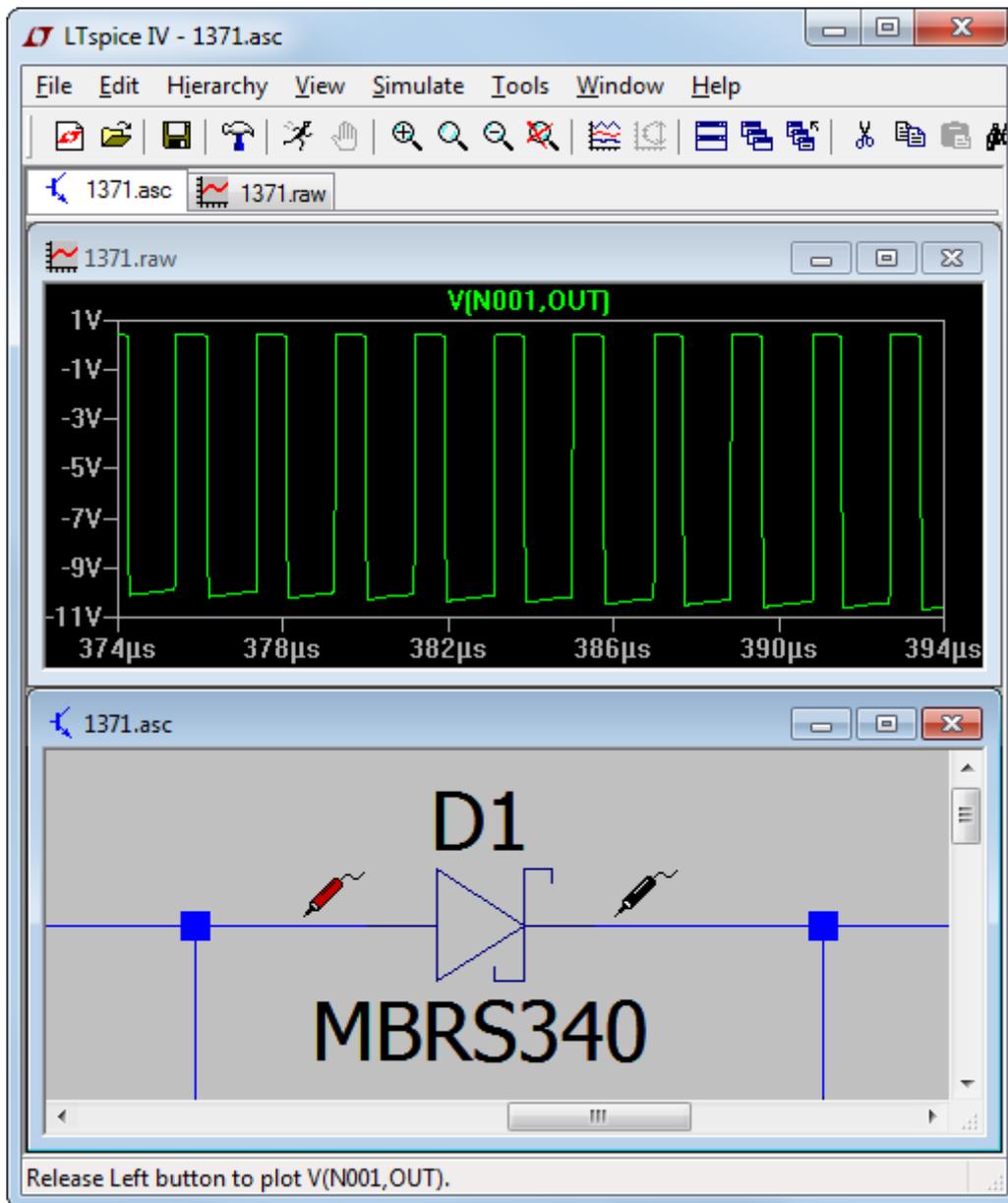


最も簡単な方法は、回路図をそのままプローブで調べる方法です。素線を単純にポイントしてクリックし、その素線の電圧をプロットします。(抵抗、コンデンサ、インダクタのような)2箇所を持つ部品の本体をクリックすることにより、その部品を流れる電流をプロットします。サブ回路の電圧と電流をすべて保存している場合、この方法は回路階層のすべてのレベルで機能します。また、3ピン以上の部品の特定の接点に流れる電流をプロットするには、シンボルの該当のピンをクリックします。同じ電圧または電流をダブルクリックすると、その他のすべてのトレースは消去され、ダブルクリックしたトレースが単独でプロットされます。delete コマンドを選択した後にトレースのラベルをクリックすれば、トレースを個別に削除できます。次のスクリーン・ショットは、1つのピンに流れる電流の方向を示す方法を示しています。マウス・カーソルは、プロットできる電流を指しているときは、スナップオン・アンメータのように見えるアイコンに変わり、赤の矢印が正の電流方向を示すので注意してください。

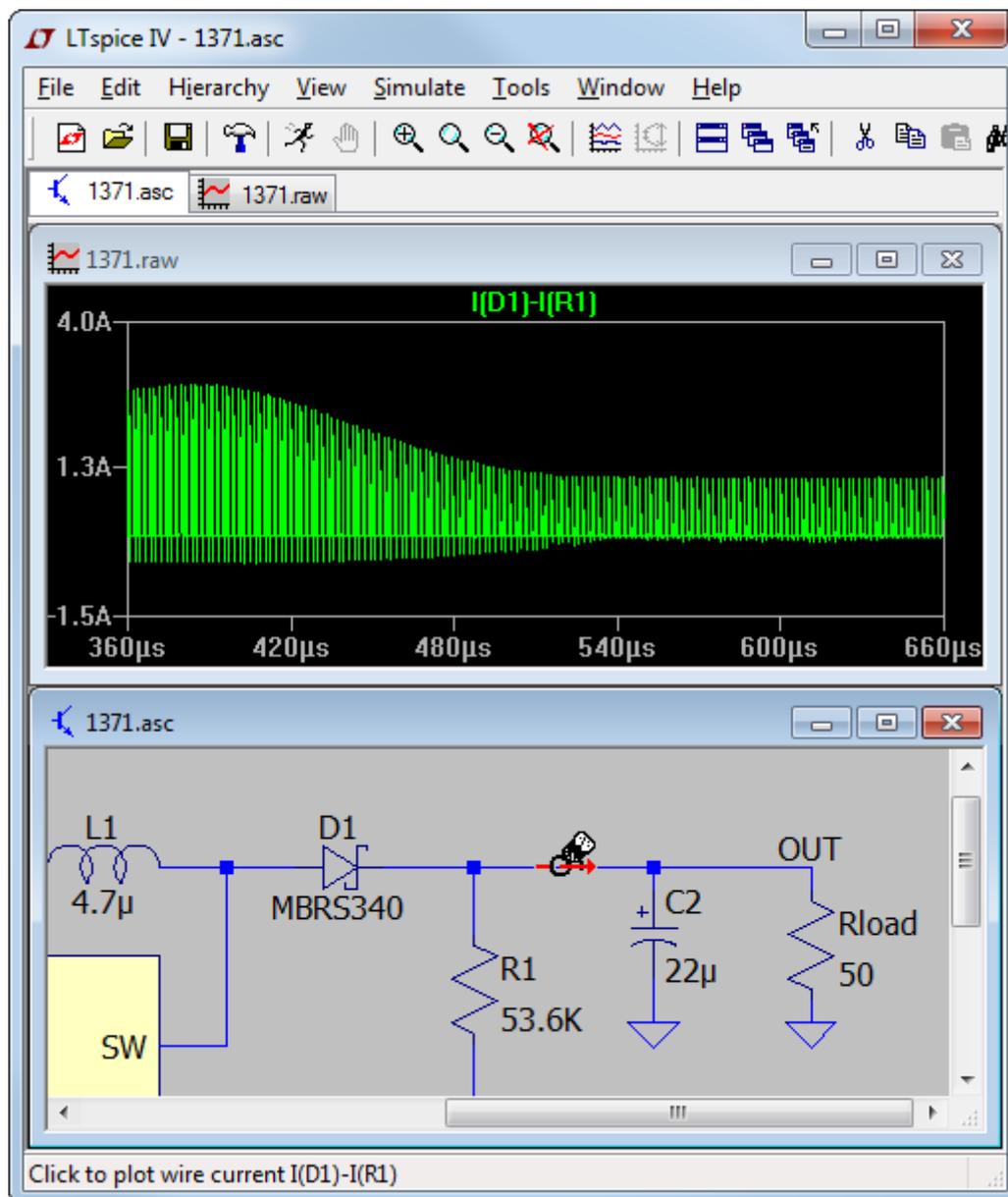


また、マウスを使用して電圧差を示すこともできます。あるノードをクリックし、別のノードにドラッグすることができます。最初のノードに赤の電圧プローブがあり、2番目のノードに黒のプローブがあることが分かります。こうすると、電圧の差分をプロットできます。

しかしながら、もう1つの回路図プロービング技法は部品の瞬時電力損失をプロットすることです。こうするには、Altキーを押したまま、部品のシンボルの本体をクリックします。瞬時電力損失は電圧と電流の式としてプロットされます。ワット(W)を単位とした独自の縮尺でプロットされます。マウス・カーソルは、プロットできる電力損失を指しているときは、温度計のように見えるアイコンに変わります。Ctrlキーを押しながらトレース・ラベルをクリックすれば、平均電力損失を表示できます。

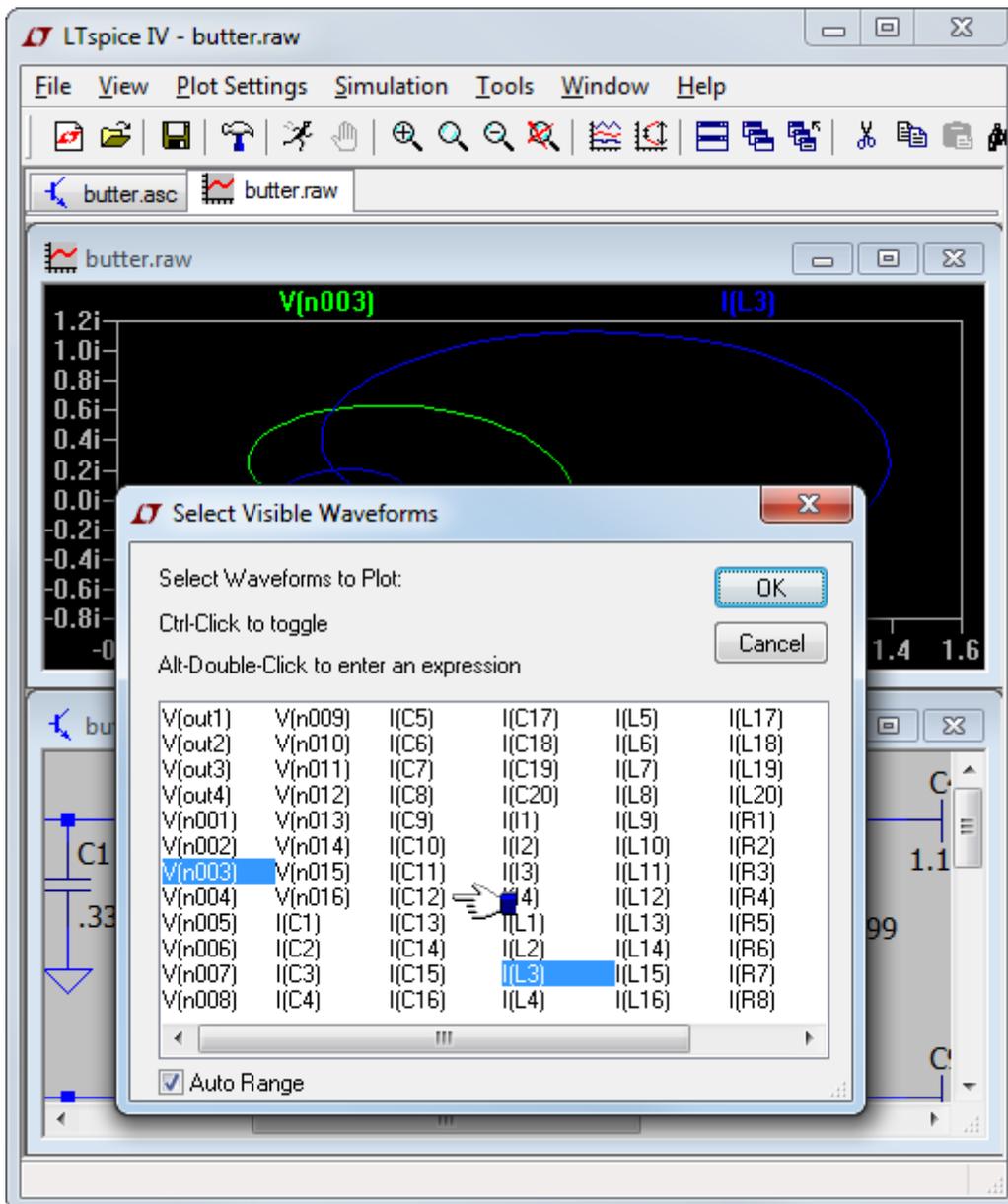


また、素線を流れる電流を調べることもできます。こうするには、Alt キーを押したまま素線をクリックします。マウス・カーソルはスナップオン・アンメータに変化してこの電流を示しており、赤の矢印は正の電流方向を示しています。



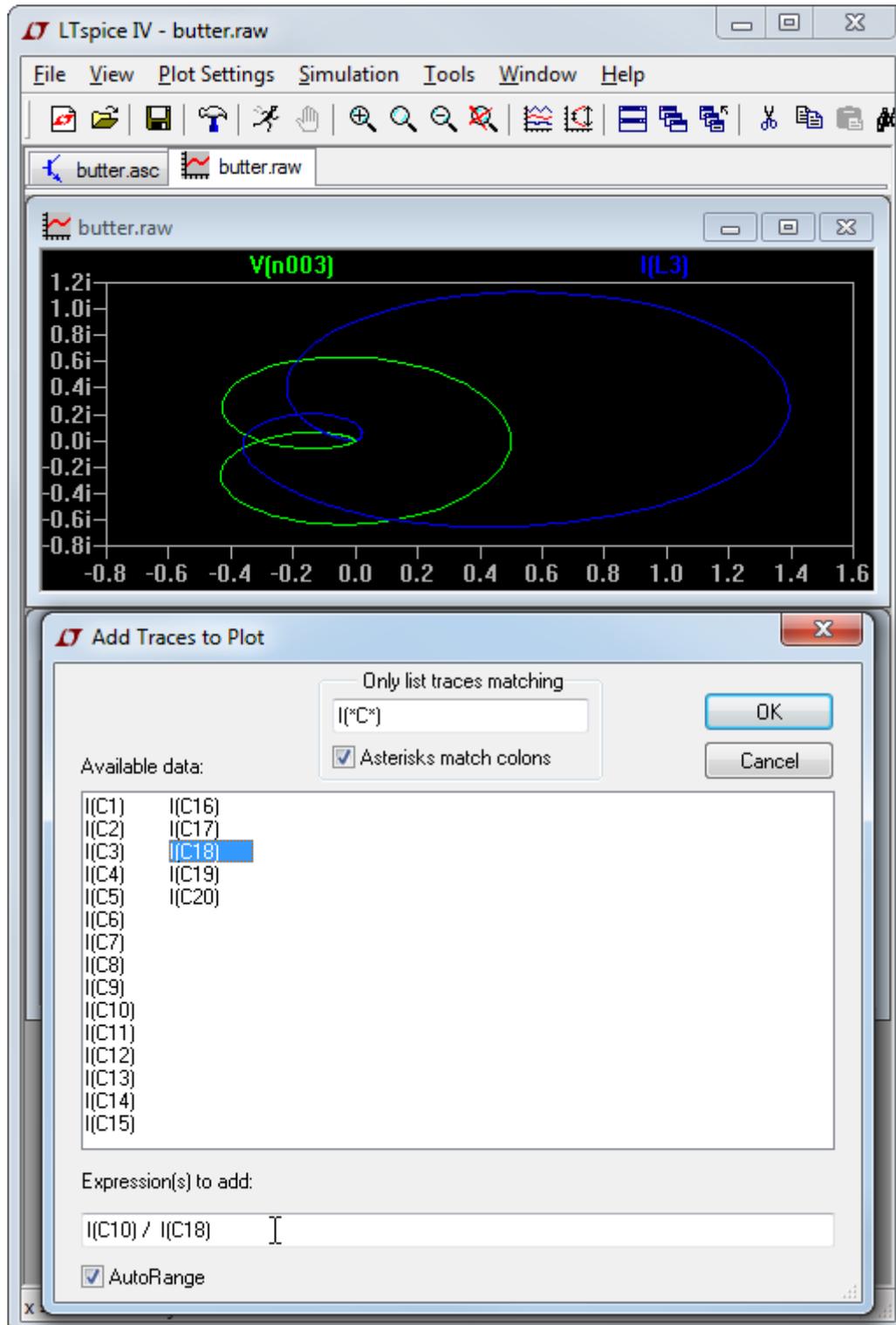
2. メニューコマンド「Plot Settings」=>「Visible Traces」:

メニューコマンド「Plot Settings」=>「Visible Traces」は、シミュレーションからのデータをプロットするとき最初に現れるダイアログです。ここでは、プロットを開始する最初のトレースを選択します。プロットしたトレースの全リストにランダムにアクセスすることもできます。



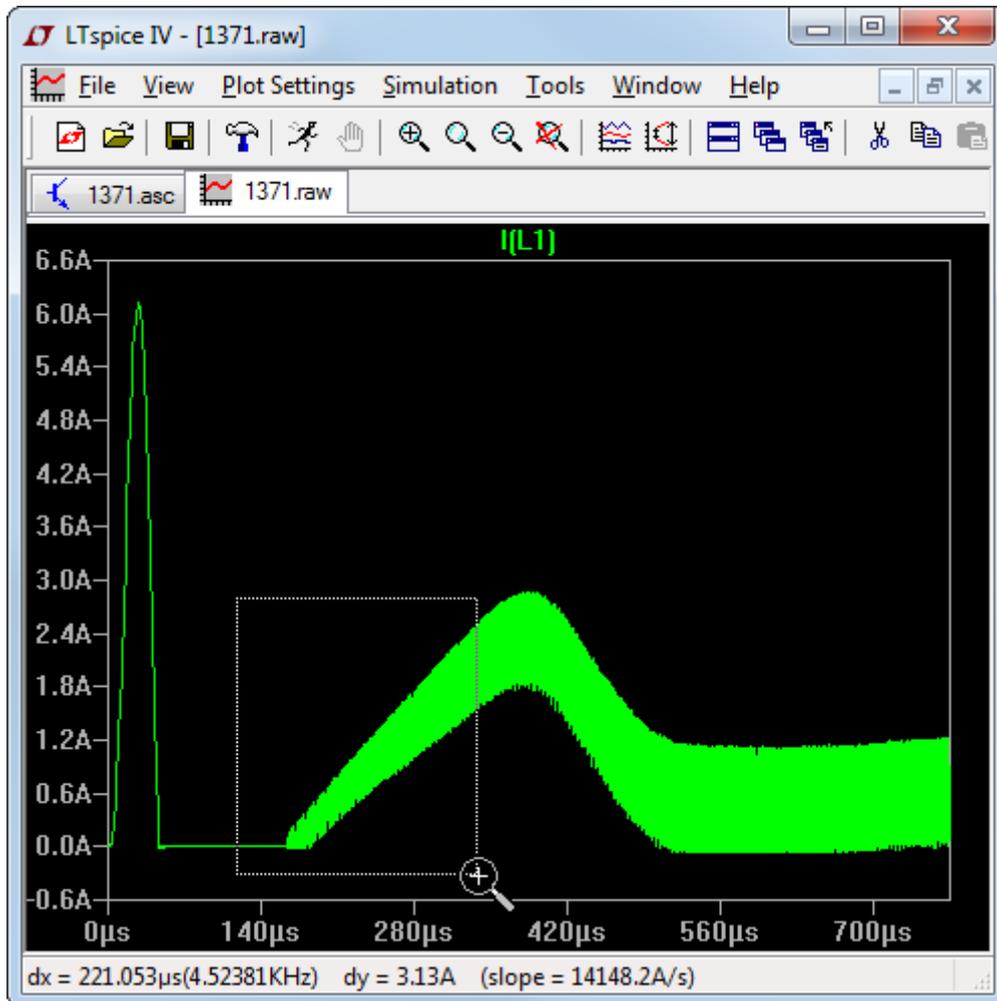
3. メニューコマンド「Plot Settings」=>「Add Trace」:

「Plot Settings」=>「Add Trace」コマンドは、「Plot Settings」=>「Visible Traces」コマンドと同様です。ただし、このコマンドを使用して既に表示されているトレースを削除することはできません。このコマンドには有用な機能が2つあります。1つは、文字のパターンを入力するためのダイアログの上部付近にある編集ボックスです。ダイアログに表示されるのは、パターンと一致するトレース名だけです。これが非常に役立つのは、トレース名を一部しか思い出せないときにトレースを探すときです。また、名前を入力する代わりにダイアログで名前をクリックできるので、トレース・データで式を作成するのが少し簡単になります。



ズーム

LTspice IV では、プロットする新しいデータがある場合は必ず、ズームが自動的に行われます。ある領域上で拡大するには、描画を拡大して見る領域を囲むボックスをドラッグすれば済みます。



添付カーソルを設定せずに差異を素早く測定できるように、ズーム・ボックスのサイズが下部のステータス・バーに表示されることに注意してください。縮小、パン、ズーム範囲自動設定への復帰に対応するツールバー・ボタンおよびメニューコマンドがあります。undo コマンドおよび redo コマンドを使用すると、使用したのと異なるズームを検討できます。

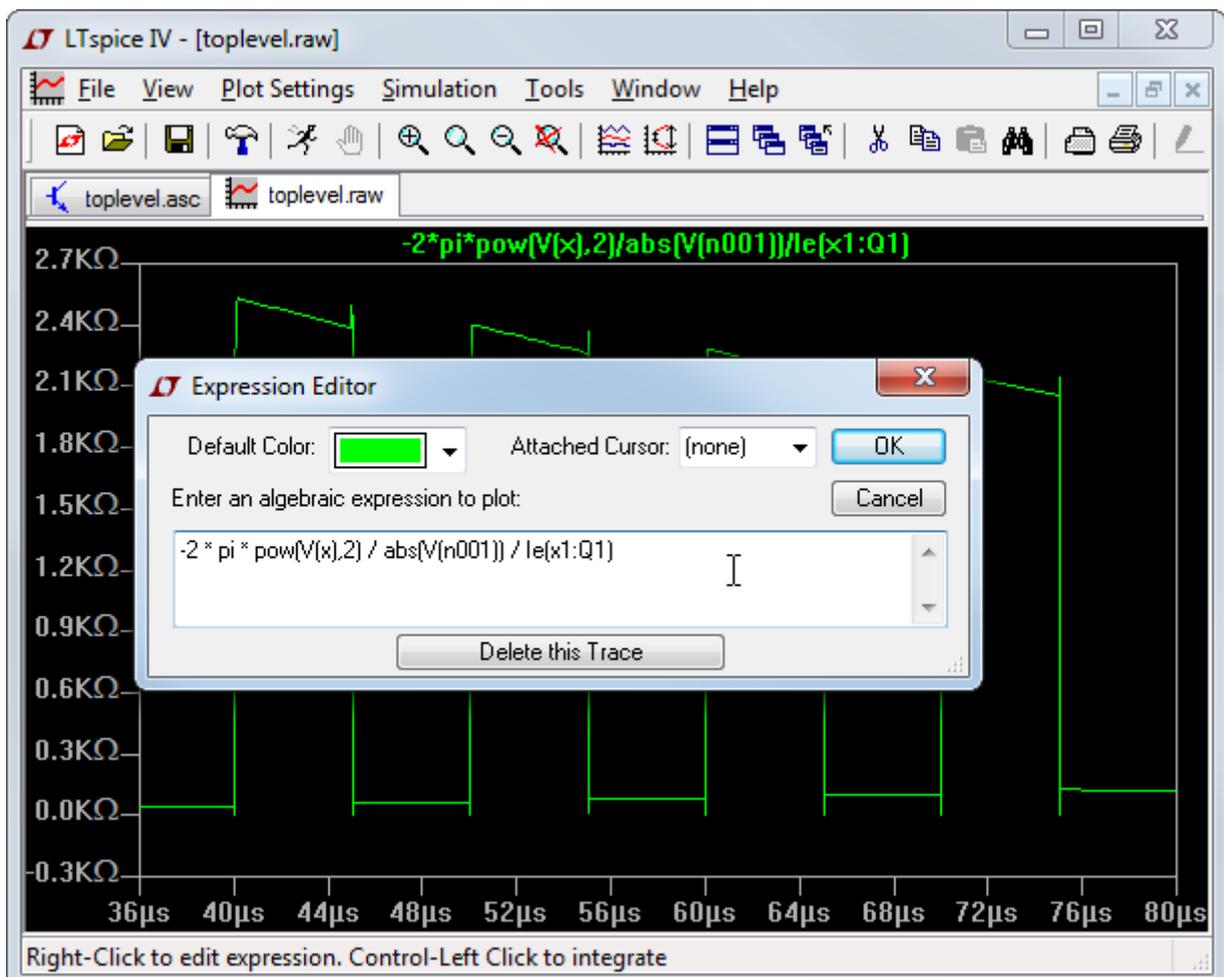
波形の算術演算

波形データに対して実行できる数学的演算は、次の 3 種類です。

1. トレースの式をプロットする
2. トレースの平均または RMS を計算する
3. トレースのフーリエ変換を表示する

1. トレースの式をプロットする

「View」=>「Visible Traces」コマンドまたは「View」=>「Add Trace」コマンドを使用すると、どちらもデータの式を入力できます。使用可能なシミュレーション・データ・トレースの式をプロットするもう 1 つの方法は、トレースのラベルにマウスを移動して、右クリックすることです。このダイアログボックスでは、トレースの色を設定することや、波形にカーソルを添付できます。LTspice は式の次元解析を実行し、その結果を、該当の単位でラベル付けされた垂直軸に対してプロットします。たとえば、下図では LTspice が $-2 * \pi * \text{pow}(V(x), 2) / \text{abs}(V(n001)) / \text{le}(x1:Q1)$ の次元を Ω と識別していることが分かります。同じ単位の描画ペイン内にあるすべての波形は同じ軸でプロットされます。



2つの電圧の差、たとえば $V(a) - V(b)$ は、 $V(a,b)$ と書き換えることができます。実数データには以下の関数を使用できます。

関数名	概要
abs(x)	x の絶対値
acos(x)	x の逆余弦
arccos(x)	acos()と同義
acosh(x)	逆双曲線余弦
asin(x)	x の逆正弦
arcsin(x)	sin()と同義
asinh(x)	逆双曲線正弦
atan(x)	x の逆正接
arctan(x)	atan()と同義
atan2(y, x)	y/x の 4 象限逆正接
atanh(x)	逆双曲線正接
buf(x)	x>.5 の場合は 1、そうでない場合は 0
ceil(x)	x 以上の整数
cos(x)	x の余弦
cosh(x)	x の双曲線余弦
d()	有限の差分に基づく導関数
exp(x)	e の x 乗
floor(x)	x 以下の整数
hypot(x,y)	$\sqrt{x^{**2} + y^{**2}}$
if(x,y,z)	x>.5 の場合は y、そうでない場合は z
int(x)	x を整数に変換
inv(x)	x>.5 の場合は 0、そうでない場合は 1
limit(x,y,z)	x、y、z の中間値
ln(x)	x の自然対数
log(x)	ln()の代替構文
log10(x)	10 を底とする対数
max(x,y)	x と y のうち大きい方の値
min(x,y)	x と y のうち小さい方の値
pow(x,y)	x^{**y}
pwr(x,y)	$\text{abs}(x)^{**y}$
pwrsgn(x,y)	$\text{sgn}(x) * \text{abs}(x)^{**y}$
rand(x)	x の整数値に依存する 0 と 1 の間の乱数
random(x)	rand()と同様だが、値の間を滑らかに遷移

round(x)	x に最も近い整数
sgn(x)	x の符号
sin(x)	x の正弦
sinh(x)	x の双曲線正弦
sqrt(x)	x の平方根
table(x,a,b,c,d,...)	一連の点の組として与えられた参照表に基づいて x の値を内挿する
tan(x)	x の正接
tanh(x)	x の双曲線正接
u(x)	単位ステップ、つまり $x > 0$ の場合は 1、そうでない場合は 0
uramp(x)	$x > 0$ の場合は x、そうでない場合は 0
white(x)	-0.5 と 0.5 の間の乱数が値の間を random() より滑らかに遷移する

複素数データの場合、関数 atan2(), sgn(), u(), buf(), inv(), uramp(), int(), floor(), ceil(), rand(), min(), limit(), if(), および table(...) は使用できません。関数 Re(x) および Im(x) は複素数データに対して使用可能であり、それぞれ引き数の実数部または虚数部に等しい実数部と、0 に等しい虚数部を持つ複素数を返します。関数 Ph(x) および Mag(x) も複素数データに対して使用可能であり、それぞれ引き数の位相角または振幅に等しい実数部と、0 に等しい虚数部を持つ複素数を返します。また、関数 conj(x) は複素数データに対して使用可能であり、x の複素共役を返します。

評価の優先順位の逆の順に要約した以下の演算が、実数データに対して使用可能です。

オペランド	概要
&	式の両辺をブール値に変換後、論理積をとる
	式の両辺をブール値に変換後、論理和をとる
^	式の両辺をブール値に変換後、排他的論理和をとる
>	式の左辺が式の右辺より大きい場合は TRUE、そうでない場合は FALSE
<	式の左辺が式の右辺より小さい場合は TRUE、そうでない場合は FALSE
>=	式の左辺が式の右辺以下である場合は TRUE、そうでない場合は FALSE
<=	式の左辺が式の右辺以上である場合は TRUE、そうでない場合は FALSE
+	加算
-	減算

*	乗算
/	除算
**	左辺を右辺で累乗する
!	次の式をブール値に変換して反転する
@	ステップ選択演算子

TRUE は数値的には 1 に等しく、FALSE は 0 に等しくなります。ブール値への変換により、値が 0.5 より大きい場合、値は 1 に変換され、そうでない場合、値は 0 に変換されます。

ステップ選択演算子「@」は、.step、.temp または.dc 解析時にシミュレーションを複数回実行できる場合に役立ちます。この演算子は、特定の実行回からデータを選択します。

たとえば、プロット対象としてどのステップを選択した場合でも、V(1)@3 では、3 回目の実行からのデータがプロットされます。複素数データの場合、使用できるのは+、-、*、/、**、@だけです。また、複素数データに関しては、ブール XOR 演算子(^)は指数演算子(**)を意味すると解釈されます。

以下の定数は内部で定義されています。

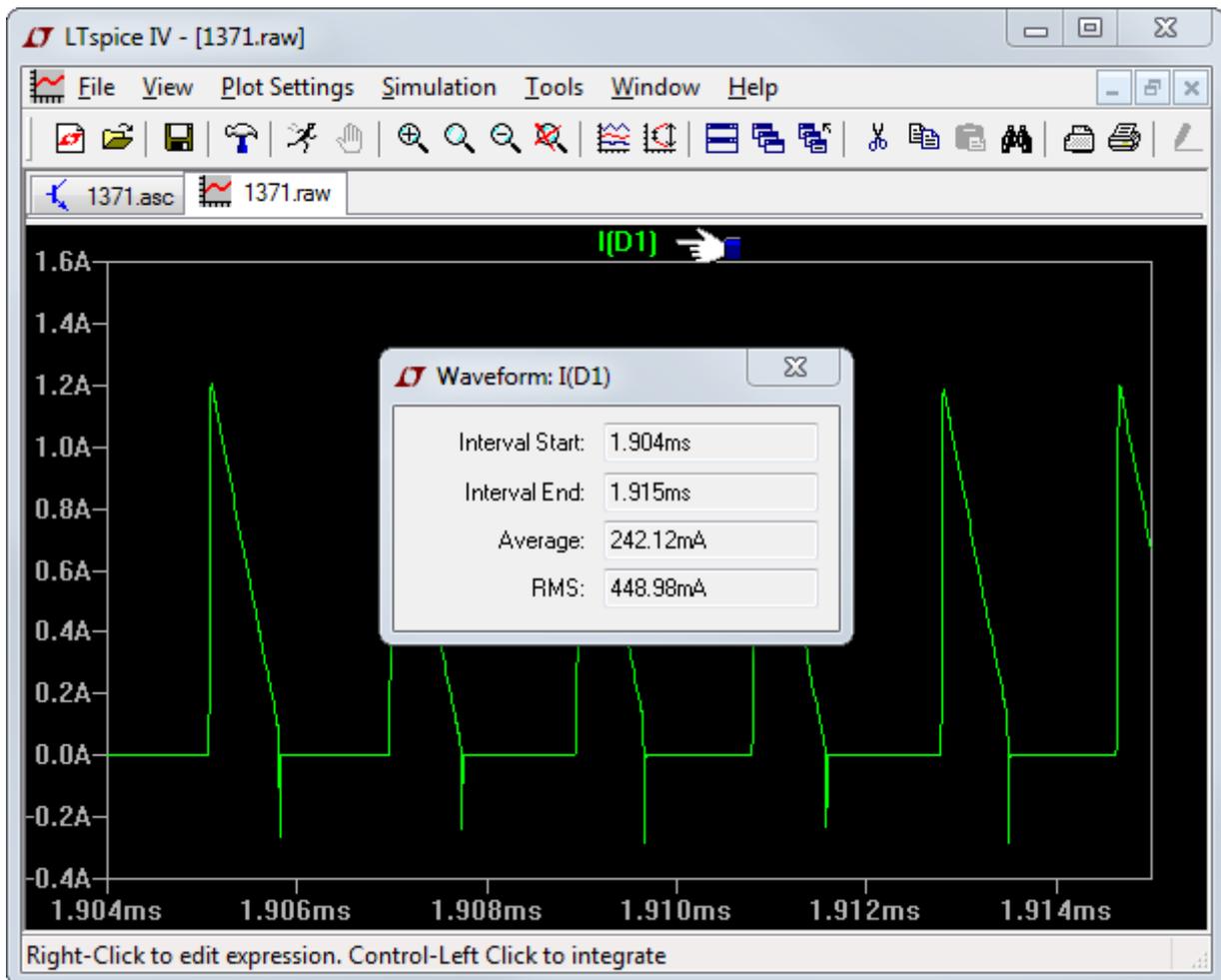
名前	値
E	2.7182818284590452354
pi	3.14159265358979323846
K	1.3806503e-23
Q	1.602176462e-19

キーワード「time」は、トランジェント解析波形データをプロットするときに解釈されます。同様に、「freq」および「omega」は、AC 解析のデータをプロットするときに解釈されます。「w」は omega の同義語として使用できます。

2. トレースの平均または RMS を計算する

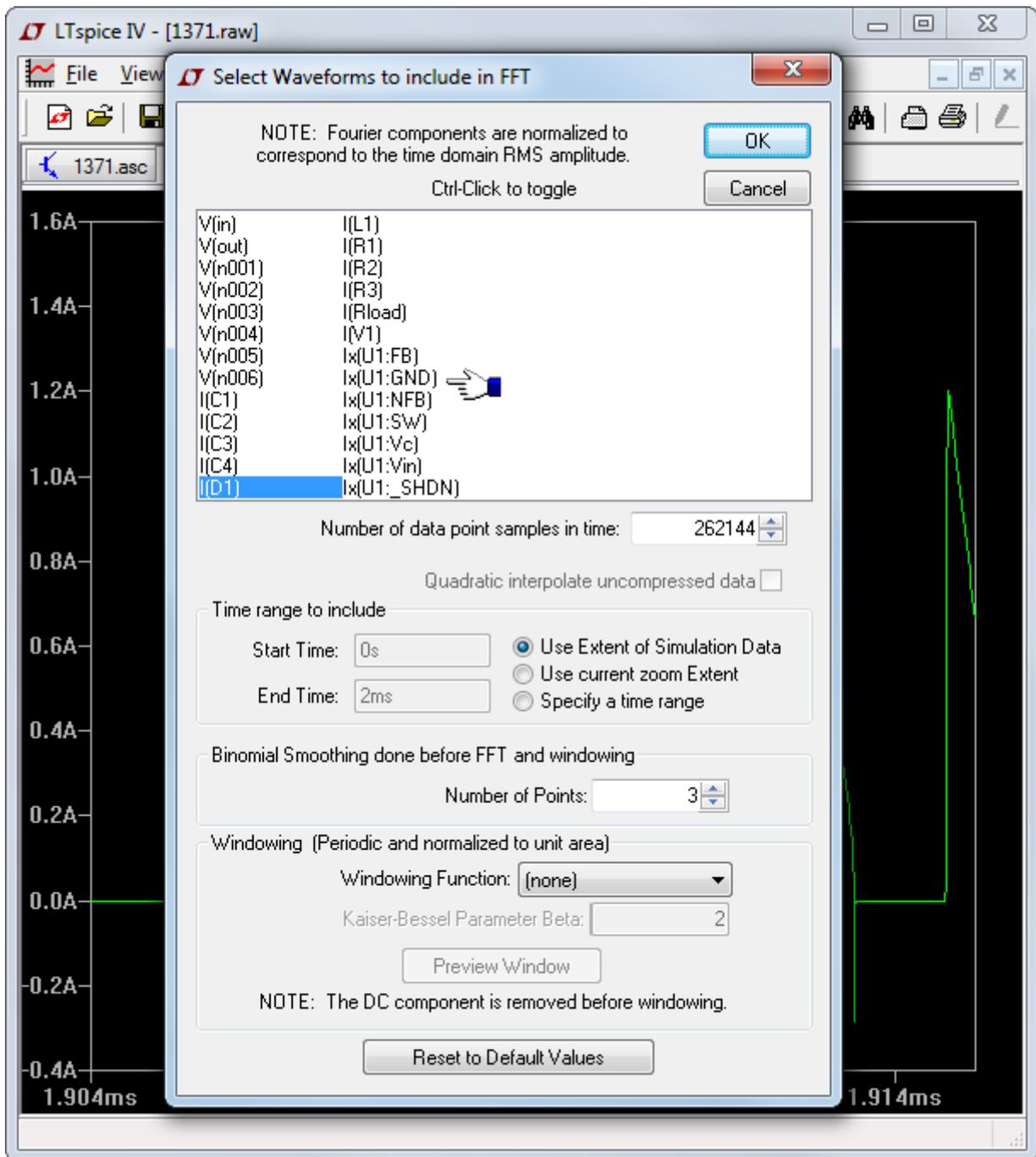
波形ビューワは、トレースを取り込んで、表示領域での平均値および RMS 値を求めることができます。

まず、目的の領域まで波形を拡大/縮小し、次に、トレースのラベルにマウスを移動して、Ctrl キーを押したまま左マウス・ボタンをクリックします。



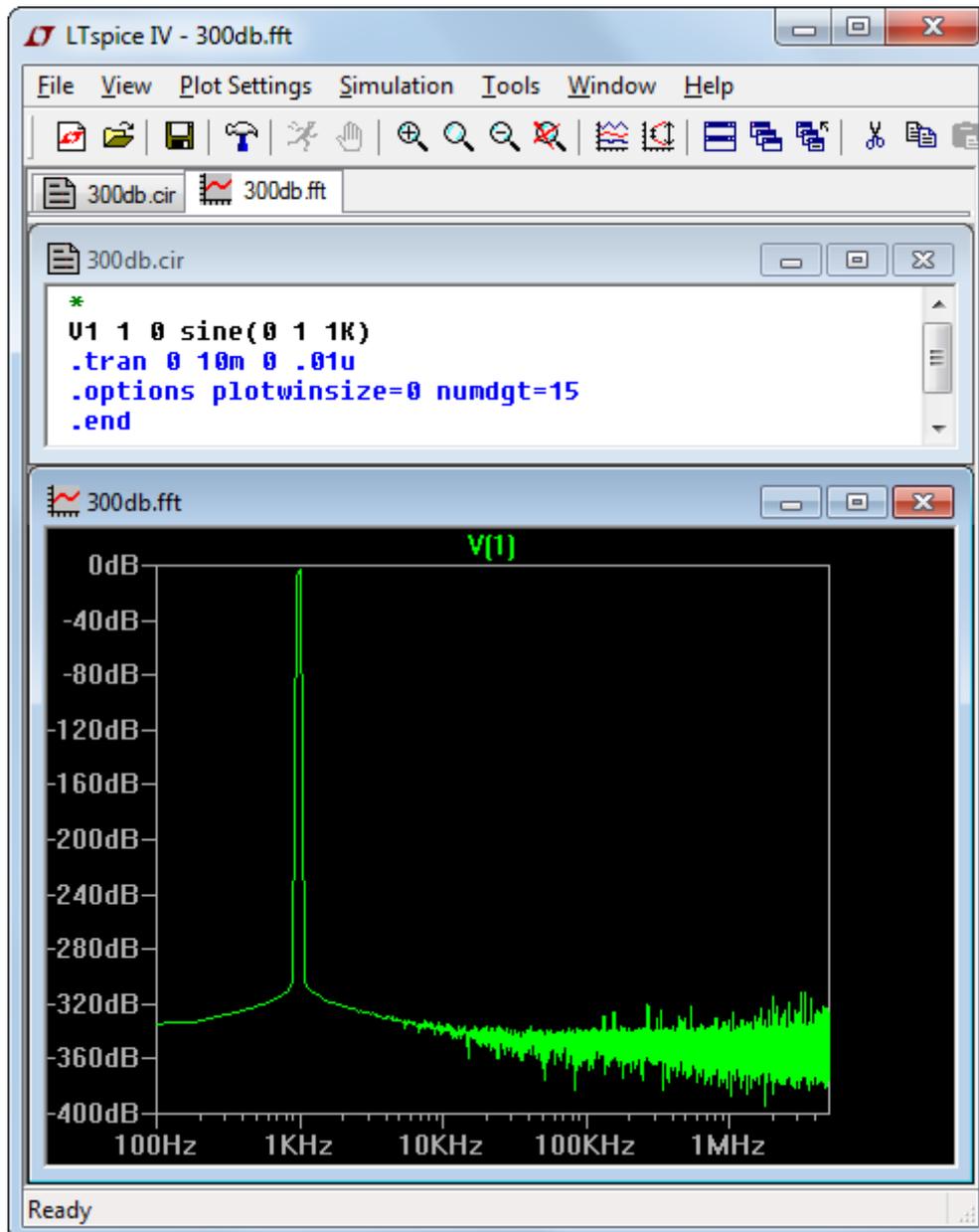
3. トレースのフーリエ変換を表示する。

メニューコマンド「View」=>「FFT」を使用して、さまざまなデータ・トレースの高速フーリエ変換を実行できます。



LTspice は、任意の数のデータ点を使用できる、つまりデータ点が 2 の累乗に制限されない、独自の FFT アルゴリズムを使用します。

シミュレーション・データの FFT を実行する場合は、波形の圧縮をやめ、最大の時間刻みを規定し、場合によっては倍精度の波形ファイル形式を使用して数値ノイズフロアを低減することも必要になるでしょう。次のネットリストは、LTspice の FFT アルゴリズムの固有ノイズフロアが 300dB を超えていることを示しています。



ユーザ定義関数

メニューコマンド「Plot Settings」=>「Edit Plot Defs File」を実行すると、波形ビューワで使用する独自の関数定義およびパラメータ定義を入力できます。これらの関数は、LTspiceIV の実行可能ファイル scad3.exe と同じディレクトリ内にあるファイル plot.defs に保管されます。

さらに、構文は、パラメータ化された回路に使用される.param 文や.func 文と同じです。たとえば、次の行

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

では、関数 Pythag()が、その 2 つの引き数の和の平方根であると定義されます。

同様に、次の行

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

では、twopi が 6.28318530717959 であると定義されます。この行では、波形ビューワの内部で既に定義されている定数 pi が使用されていることに注意してください。

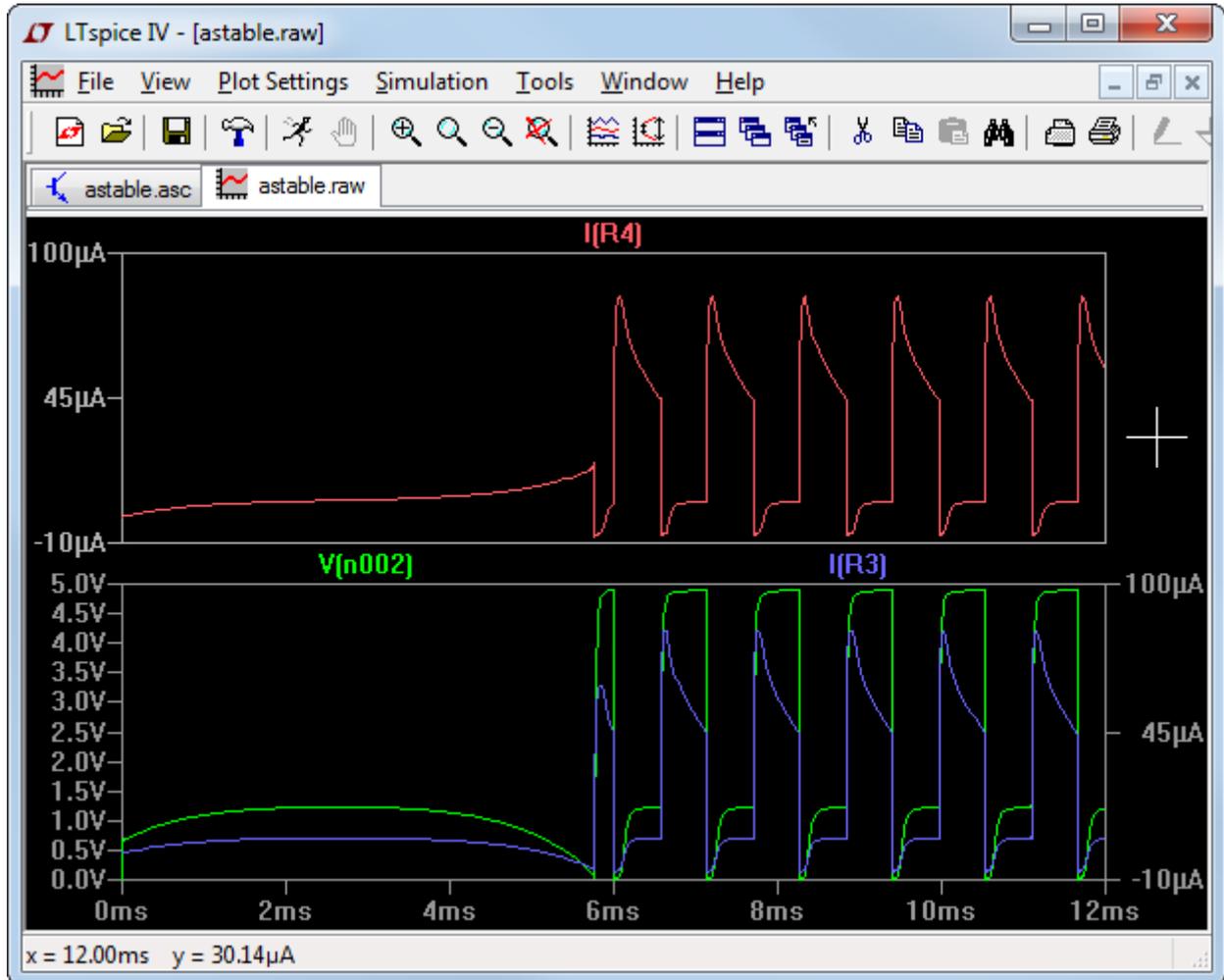
軸の制御

データ・プロット領域を越えてマウス・カーソルを動かすと、カーソルは定規に変わります。これは、その軸の属性を指していることを示しています。左クリックするとダイアログが表示され、その軸の範囲とプロットの性質を手動で入力できます。たとえば、実数データの場合、画面の下部にマウスを移動して左クリックするとダイアログが表示され、プロットされる水平方向の数を変更できます。これにより、パラメータのプロットが可能です。複素数データの場合は、右の垂直軸に対してプロットするパラメータを位相、群遅延、プロットなしの中から選択できます。

複素数データの左の垂直軸にマウスを移動すれば、複素数データの表現を「Bode」から「Nyquist」または「Cartesian」に変更できます。

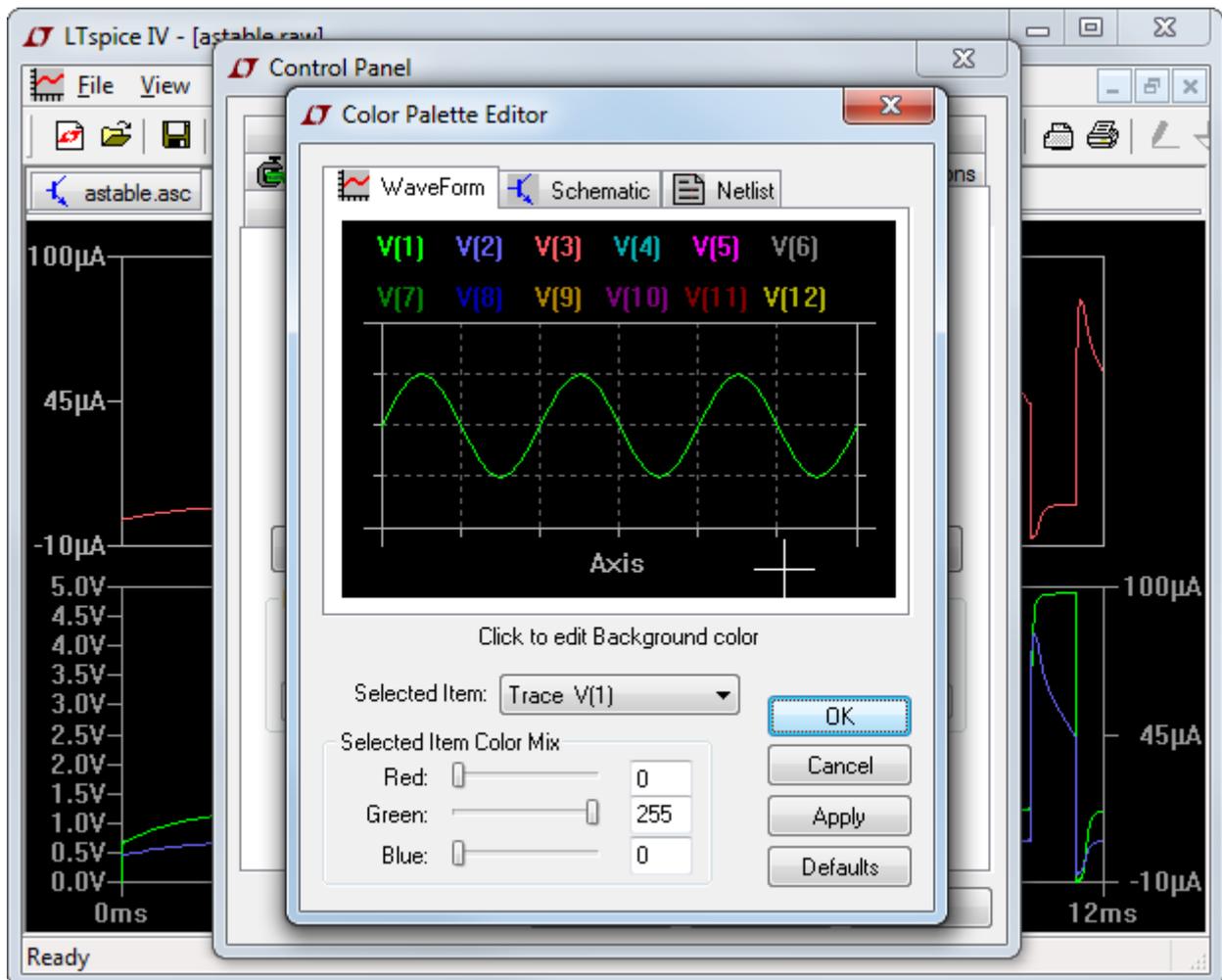
プロット・ペイン

1つのウィンドウに複数のプロット・ペインを表示できます。これにより、トレース間の区切りが良好になり、異なるトレースの目盛を個別に自動調整できます。ラベルをドラッグすれば、トレースをペイン間でドラッグできます。マウス・ボタンを放すときに Ctrl キーを押したままにすることにより、別のペインにトレースをコピーすることができます。



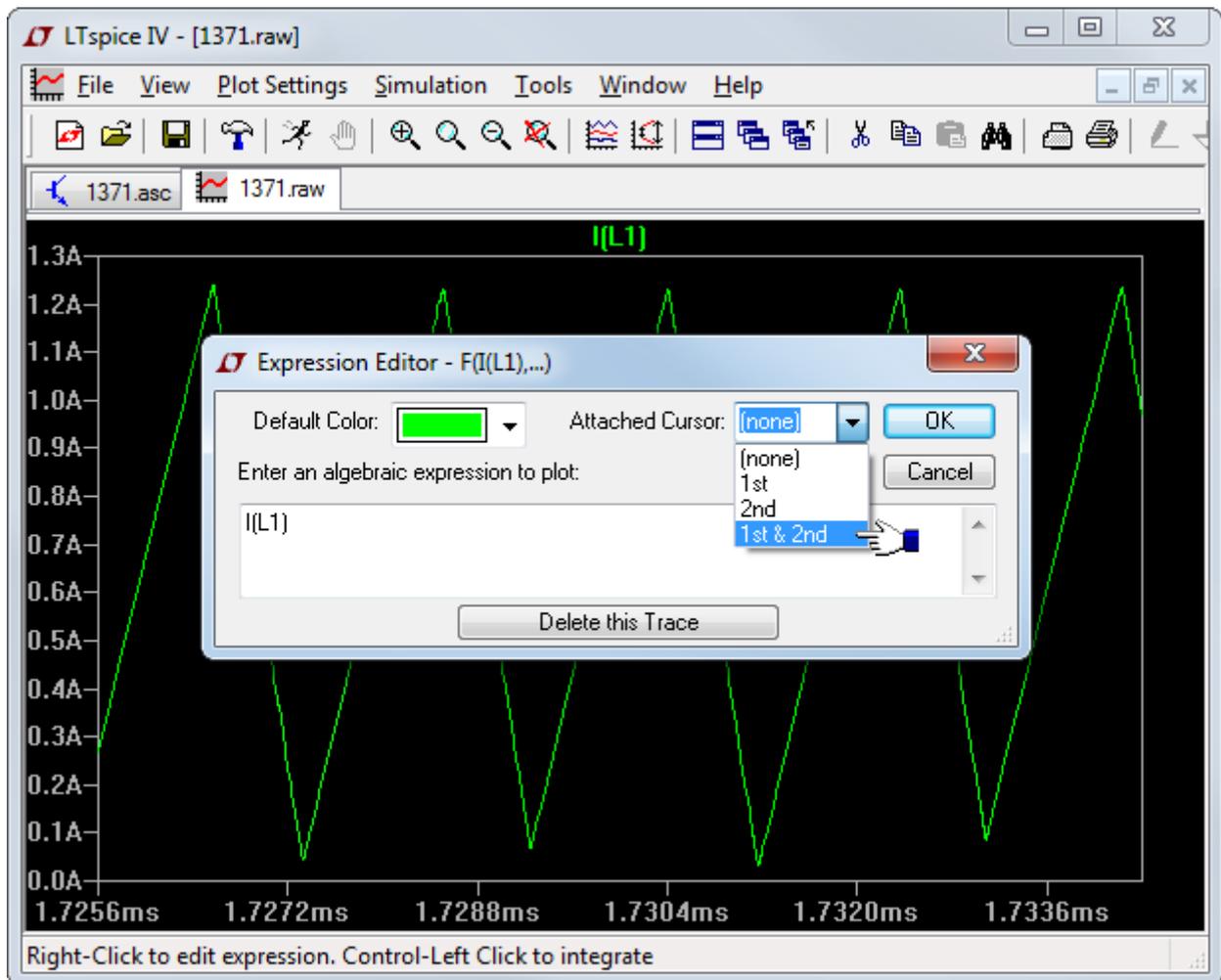
色の制御

メニューコマンド「Tools」=>「Color Preferences」を使用すると、データをプロットするときに使用する色を設定できます。サンプル・プロット内のオブジェクトをクリックし、赤のスライダ、緑のスライダ、および青のスライダを使用して、好みに合わせて色を調整します。

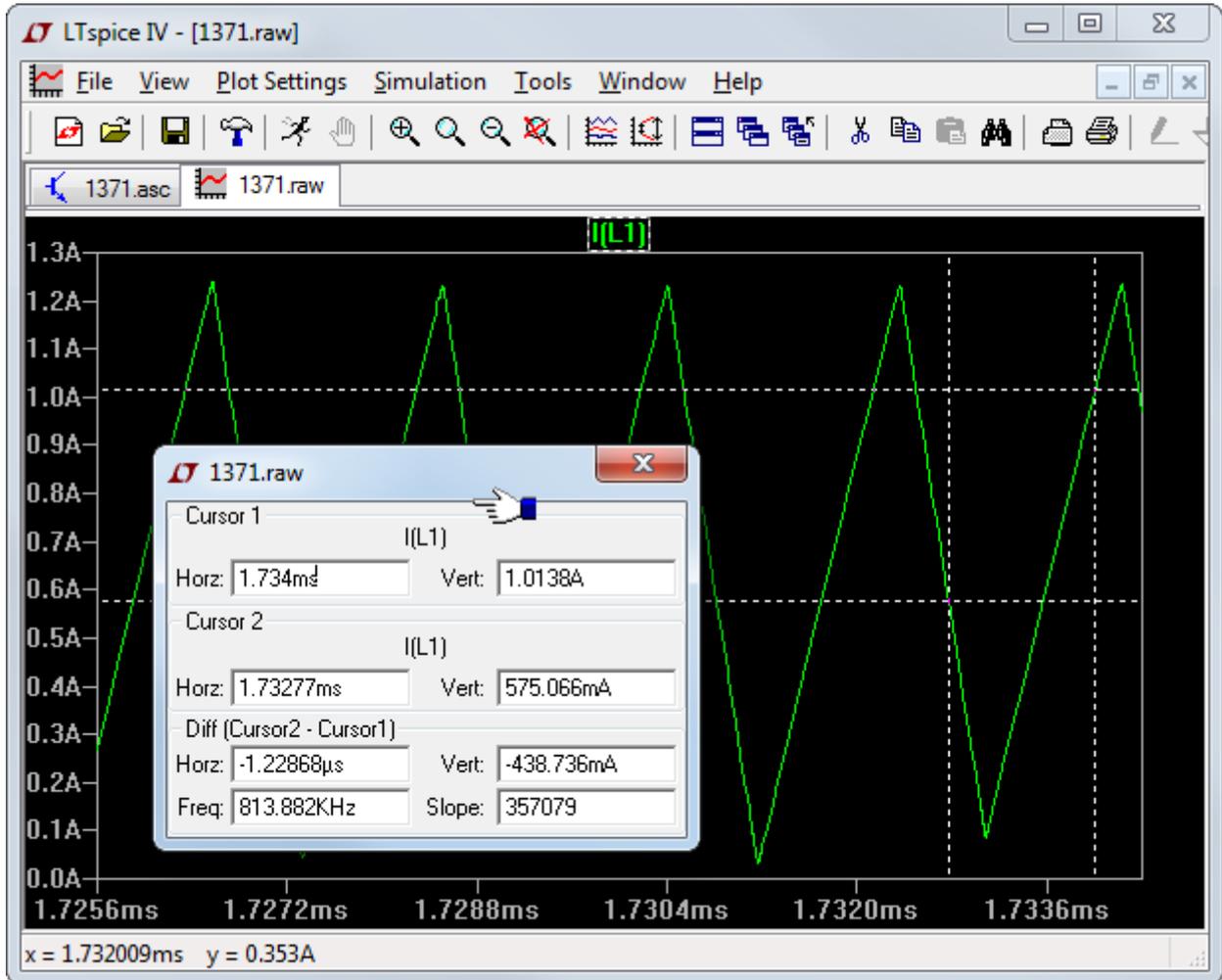


添付カーソル

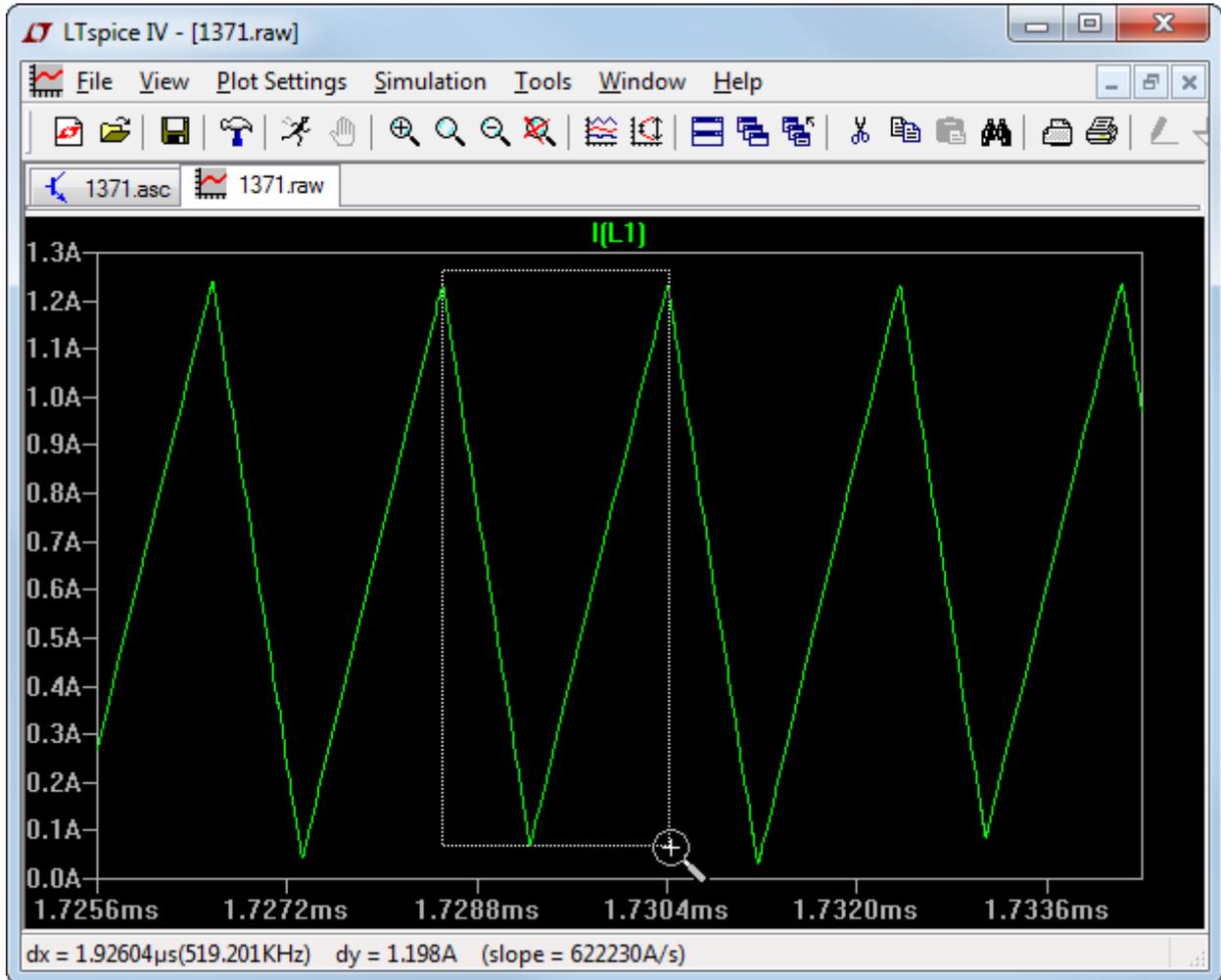
使用可能な添付カーソルは最大で2つあります。マウスの左ボタンでトレース・ラベルをクリックすれば、トレースにカーソルを添付できます。トレース・ラベルを右クリックして「1st & 2nd」を選択することにより、1つのトレースに2つのカーソルを添付できます。また、任意のトレースのラベルを右クリックして「Attached Cursor」ドロップダウン・ボックスを使用することにより、第1のカーソル、第2のカーソル、または両方のカーソルをトレースに添付できます。添付カーソルはマウスでドラッグするか、カーソル・キーで移動できます。



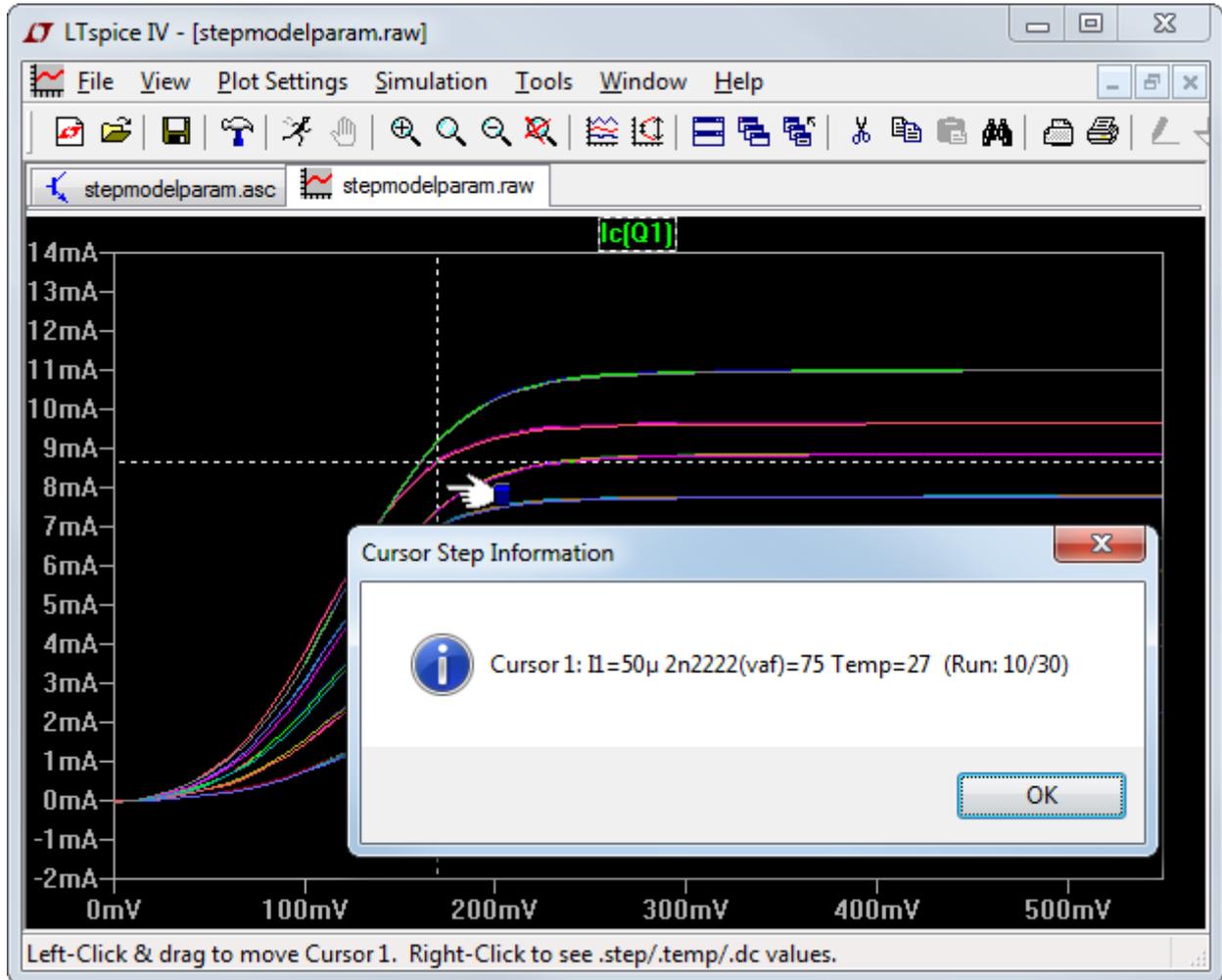
アクティブな添付カーソルがある場合は、カーソルの場所と差を通知する読み取り表示部が表示されるようになります。



マウス・カーソル読み取り値は上記の添付カーソル読み取り値とは無関係であることにも注意してください。波形ウィンドウ上でマウスを動かすのに応じて、マウスの位置がステータス・バー上に読み出されます。ズームを行うかのようにマウスをドラッグすると、ボックスのサイズがステータス・バーに表示されます。こうすると、マウス・カーソルで差を素早く測定できます。水平軸が時間である場合は、この時間差も周波数に変換されます。



左マウス・ボタンを放す前に Esc キーまたは右マウス・ボタンを押すことにより、ズームを行わずにこの要領で差を測定できます。



添付カーソルは、一連の.step/.dc/.temp シミュレーションのどの実行回にどのトレースが属するかを読み取るのに使用することもできます。キーボードの上下のカーソル・キーを使用してデータセット間でカーソルを移動し、さらに、カーソルを右クリックして、該当する実行回のステップ情報を参照できます。

プロット構成情報の保存

メニューコマンド「Plot Settings」=>「Save Plot Settings/Open Plot Settings Files」を使用すると、プロット構成を読み取ってディスクに書き込むことができます。プロット設定ファイルは、ファイル拡張子が.plt の ASCII ファイルです。デフォルトのファイル名は、データ・ファイルの「.raw」拡張子を「.plt」に置き換えることにより、データ・ファイルの名前を基に割り出されます。最初にデータ・ファイルを開いたときに、そうしたファイル名が存在する場合は、そのプロット設定ファイルが初期プロット構成のために読み取られます。

各解析タイプ(.tran、.ac、.noise など)には、プロット設定ファイル内にそれ専用の項目があります。ある解析タイプから別の解析タイプに設定を読み込むことはできません。ただし、同じ解析タイプの別のシミュレーションからのプロット設定ファイルを使用しても構いません。

Fast Access ファイル形式

LTspice がシミュレーション中に通常使用するファイル形式は、残りのファイルを変更せずにシミュレーション・データを追加できる圧縮バイナリ・ファイル形式です。ただし、いったんシミュレーションが完了すると、単一の新しいプロット・トレースをファイルから追加するために、このファイル形式はアクセス速度が低下する可能性があります。

この時間を短縮するため、ファイルを別の形式 (Fast Access) に変換することができます。この形式に変換できるのは、シミュレーション完了後、ファイルに追加する新しいデータがないときだけです。ただし、いったんこの形式にファイルを変換すると、新しいトレースの読み込み時間が短縮され、通常はファイル内に保存されているデータ・トレースの数で割った値になります。たとえば、データ・トレース数が 2000 の 5GB のファイルがある場合、新しいトレースを追加するのに 4 分かかることがあります。しかし、ファイルを Fast Access 形式に変換後、この 4 分の読み込み時間は 1 秒に短縮されます。これにより、大型のシミュレーションデータ・ファイルを対話式に使用して大規模な回路を相互に調査できます。Fast Access 形式ファイルからのトレースを読み込むのにかかる正確な時間は、ハードディスクの速度より装着している物理メモリの大きさに依存します。

波形ウィンドウを Fast Access 形式に変換し、波形ウィンドウをアクティブ・ウィンドウにするには、メニューコマンド「Files」=>「Convert to Fast Access」を実行します。変換処理では、変換するファイル・サイズと等しい空きディスク容量が必要ですが、変換後のファイルは元のファイルよりわずかに 11 バイト大きくなります。変換処理には長い時間がかかり、最大で物理メモリの 4 分の 1 を使用する可能性があります。事実、ファイルを Fast Access 形式に変換するのに、最初のシミュレーションで必要だった時間より長い時間がかかることがあります。

変換処理に必要な正確な時間は、ハードディスク断片化の状態、装着している物理メモリの量などの要因に依存します。変換中、マシンがマウスとキーボードにあまり反応しないことに気づくことがあります。以下のコマンド・ライン構文を使用して、バッチ・コマンドでファイルを変換することができます。

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

ここで<file>は、Fast Access 形式に変換する.raw ファイルの名前です。この形式のサポート対象は実数データだけであり、.ac 解析によって得られる複素数データは対象外です。

メモリ、RAM、およびアドレス空間

LTspice は PC ベースの最初の SPICE プログラムで、独自の 64 ビット・アドレスをハードディスク上に実装して、ファイル・サイズが実質的に無制限の波形データ・ファイルを表示できます。LTspice は、多 GB のデータを収容したデータ・ファイルをアドレス指定し、一度に最大 4GB をページインして波形ビューワでプロットできます。

ただし、Microsoft Windows のほとんどのエディションでは、実際には 32 ビットで 4GB を一意にアドレス指定できるにもかかわらず、許容しているアドレス空間は最大で 2GB です。よくあるように、物理 RAM を 4GB 保有している場合は、Windows を構成してこのデフォルト値を変更できます。Windows XP とバリエーションの環境では、boot.ini ファイルを編集して、「/3GB」のオプションをオペレーティング・システム行に追加することができます。http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.msp で、以下の内容を参照します。

標準的な boot.ini ファイルの場合：

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)¥WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)¥WINNT="????"/3GB
```

ここで、「????」は、以下のいずれかのプログラム名です。

```
Windows XP Professional
Windows Server 2003
Windows Server 2003, Enterprise Edition
Windows Server 2003, Datacenter Edition
Windows 2000 Advanced Server
Windows 2000 Datacenter Server
Windows NT Server 4.0, Enterprise Edition
```

Windows Vista は異なるので、ユーティリティ bcdedit.exe を使用してください。例を次に示します。

```
C:¥Windows¥System32>bcdedit.exe /set IncreaseUserVa 3072
```

システムが再起動するまで変更は行なわれません。

LTspice® IV

LTspice IV は、回路図方式の回路シミュレーション・プログラムです。LTspice シミュレータの原型は、数年前の Berkeley SPICE 3F4/5 が基本になっています。LTspice シミュレータは、その性能を向上し、バグを修正して、シミュレータを拡張するために完全な書き直しを行いました。これにより、このシミュレータは業界標準の半導体モデルおよび挙動モデルを実行できるようになりました。共同シミュレーションを含むデジタル・シミュレーション機能が追加されました。並行処理、SPARSE マトリクス・ソルバでのアセンブリやオブジェクトの動的なコード生成など、アナログ SPICE シミュレータを大幅に拡張したことにより、LTspice IV は業界最高のアナログ・シミュレータになりました。

リニアテクノロジーの多くの製品は、カスタム・マクロモデルによって現実的な動作を正確にカプセル化する専用のビルディング・ブロックあるいは専用のハードウェア記述言語によってモデル化されます。これにより、SMPS のプロトタイプをシミュレーションによって迅速に作成できます。

LTspice は汎用の SPICE シミュレータとして使用できます。新しい回路の下書きは、組み込みの[回路図取り込み機能](#)を使用して作成できます。シミュレーションのコマンドおよびパラメータは、確立された [SPICE 構文\(SPICE syntax\)](#)を使用して、回路図にテキストとして配置されます。回路ノードとデバイス電流の波形をプロットするには、シミュレーション中またはシミュレーション後に回路図内のノード上でマウスをクリックします。

本書を補完するきわめて貴重な参考文献は、『Semiconductor Device Modeling with SPICE』(Giuseppe Massobrio と Paolo Antognetti の共著、McGraw Hill 社、1993 年および後年の再版)です。この本では、各種の商用 SPICE プログラムで使用されている半導体デバイスの式や拡張機能(本書で使用したものを含む)が記述されています。BSIM 3 および 4 のデバイスについては、UC Berkeley CAD グループから入手できる関連文書を参照してください。

LTspice はリニアテクノロジー社の登録商標です。

回路の記述

回路はテキスト・ネットリストによって定義されます。ネットリストは、一連の回路素子とそのノード、モデル定義、およびその他の SPICE コマンドで構成されます。

ネットリストは、通常、図面で入力します。新しい回路図を作成するには、メニュー項目「File」=>「Open」を選択します。Windows のファイル・ブラウザが表示されます。既存の回路図を選択して新しい名前でも保存するか、新しい名前を入力して新しい空白の回路図ファイルを作成します。LTspice は、さまざまなタイプのファイルおよび文書を使用します。ファイル名拡張子が「.asc」のファイルを作成する必要があります。回路図取り込みコマンドは「Edit」メニューの下にあります。コマンドのキーボード・ショートカットは「Schematic Editor Overview」で一覧表示されます。

回路図をシミュレートすると、回路図の図表情報からネットリスト情報が抽出され、名前が回路図と同じでファイル拡張子が「.net」のファイルに情報が取り込まれます。LTspice はこのネットリストを読み込みます。

また、手動で生成するか外部で生成したテキスト・ネットリストを開いてシミュレートし、編集することもできます。拡張子が「.net」、「.cir」、または「.sp」のファイルは、LTspice によってネットリストと認識されます。

ヘルプ文書のこのセクションでは、ネットリストで使用される構文を記述していますが、ときには回路図レベルのアドバイスが得られます。

全般的な構造および規則

解析の対象回路はネットリストと呼ばれるテキスト・ファイルで記述されます。ネットリスト内の最初の行は無視されます。つまり、その行はコメントとみなされます。ネットリストの最終行は、通常は「.END」だけの行ですが、これは省略できます。「.END」行の後の行はすべて無視されます。

コメント行と最終行の間にある行の順序は意味がありません。行はコメント、回路素子宣言、シミュレーション指令のいずれにもなります。次の例から始めましょう。

```
* この先頭行は無視されます
* 下記の回路は 1MHz の方形波信号で駆動される
* RC 回路を表します
R1 n1 n2 1K ; ノード n1 と n2 の間の 1KΩ の resistor\(抵抗\)
C1 n2 0 100p ; ノード n2 とグラウンドの間の 100pF の capacitor\(コンデンサ\)
V1 n1 0 PULSE\(電圧源\) (0 1 0 0 0 .5μ 1μ) ; 1MHz の方形波
.tran 3μ ; 3μs のトランジェント解析を実行する
.end
```

先頭の 2 行はコメントです。「*」で始まる行はすべてコメントであり、無視されます。「R1」で始まる行は、ノード n1 と n2 の間に 1K の抵抗が接続されていると宣言しています。セミコロン「;」を使用して、行の途中からコメントを開始できることに注意してください。「C1」で始まる行は、ノード n2 とグラウンドの間に 100pF のコンデンサがあると宣言しています。ノード「0」は回路全体の共通グラウンドです。

LTspice の辞書の概要を以下に示します。

- 大文字と小文字、先頭のスペース、空白、およびタブは無視されます。
- 行の先頭の非空白文字は、回路素子のタイプを定義します。

先頭文字	行のタイプ
*	コメント
A	特殊機能デバイス
B	任意の動作信号源
C	コンデンサ
D	ダイオード
E	電圧依存型の電圧源
F	電流依存型の電流源
G	電圧依存型の電流源
H	電流依存型の電圧源
I	独立した電流源

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	損失の大きい伝送線路
Q	バイポーラ・トランジスタ
R	抵抗
S	電圧制御型スイッチ
T	損失の小さい伝送線路
U	均一な RC 線路
V	独立した電圧源
W	電流制御型スイッチ
X	サブ回路の呼び出し
Z	MESFET トランジスタ
-	シミュレーション指令。例: .options reitot=1e-4
+	前の行からの継続。「+」は削除され、行の残りは前の行の一部とみなされます。

数値は科学的記数法(例: 1e12)だけでなく、工学的乗数を使用して表現することもできます。つまり、1000.0 または 1e3 も 1K と書くことができます。解釈される乗数の表を以下に示します。

接尾部	乗数
T	1.00E+12
G	1.00E+09
Meg	1.00E+06
K	1.00E+03
Mil	2.54E-05
M	1.00E-03
u(or μ)	1.00E-06
n	1.00E-09
p	1.00E-12
f	1.00E-15

接尾部は大文字と小文字を区別しません。数値または工学的乗数の直後に認識されない文字があると無視されます。したがって、10、10V、10Volts、および 10Hz はすべて同じ数を表わします。また、M、MA、MSec、および MMhos はすべて同じ倍率(.001)を表わします。よくある誤りは、1MΩ(メガオーム)の抵抗のつもりで抵抗の値を 1M と下書きすることですが、1M は 1mΩ(ミリオーム)の抵抗と解釈されます。これは標準 SPICE 方式との互換性確保のために必要です。

LTspice は、6K34 という書式で書かれた数値を 6.34K の意味で解釈します。これは上記のいずれの乗数の場合でも機能します。この動作をオフするには、「Tools」=>「Control Panel」=>「SPICE」に移動して、「Accept 3K4 as 3.4K」チェックボックスをオフにします。

ノード名は任意の文字列にすることができます。GND は特別な同義語ですが、グローバルな回路共通ノード（グラウンド）は「0」です。ノードは文字列なので、「0」と「00」は別個のノードであることに注意してください。

このマニュアルの以下のセクション全体にわたって、具体的な情報を入力する必要のあるデータ・フィールドの両側には角括弧が付きます。たとえば、<srcname>はある特定の供給源の名前になります。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ます。

シミュレータ指令 - ドット・コマンド

シミュレーションを実行するには、回路を定義するだけでなく、実行の対象となる解析のタイプも定義する必要があります。異なる 6 種類の解析（線形化された [small signal AC\(小信号 AC\)](#)、[DC sweep\(DC 掃引\)](#)、ノイズ、[DC operating point\(DC 動作点\)](#)、small signal DC transfer function（小信号 DC 伝達関数）、[transient analysis\(トランジェント解析\)](#)）があります。これらの 6 つの解析のうちいずれか 1 つを明確に指定する必要があります。

回路の配置は、通常は図式的に下書きされるのに対して、コマンドは通常、回路図にテキストとして配置されます。こうしたコマンドはすべてピリオドで始まるので、「ドット・コマンド」と呼ばれます。

AC – DC 動作点を中心に線形化された小信号 AC 解析を実行する

LTspice の小信号 (線形) AC 部は、複素数の AC ノード電圧を周波数の関数として計算します。まず、回路の DC の動作点が分かります。次に、この動作点に対して、回路内のすべての非線形デバイスに対する線形化された小信号モデルが分かります。最後に、最終的な線形化回路は、個々の電圧源および電流源を駆動信号として使用して、指定の周波数範囲での周波数領域で解決されます。

このモードの解析は、フィルタ解析、回路網解析、安定性解析およびノイズの検討に役立ちます。

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

周波数は周波数 StartFreq と EndFreq の間で掃引されます。ステップ数は、次の表に従ってキーワード「oct」、「dec」または「lin」と Nsteps で定義されます。

キーワード	Nsteps
oct	1 オクターブ当たりのステップ数
dec	1 デイケード当たりのステップ数
lin	StartFreq と EndFr の間で直線状に間隔を置いたステップの総数

.BACKANNO – ポート電流にサブ回路のピン名を注釈付けする

構文: `.backanno`

この指令は、LTspice IV が回路図から生成するすべてのネットリストに自動的に組み込まれます。この指令により、LTspice は、ピン名でポート電流を参照するために使用できる.raw ファイルに情報を組み込むよう指示されます。こうすると、シンボルのピンをマウスでクリックすることにより、シンボルのピンに流れる電流をプロットできます。

.END -- ネットリストの終端

この指令は、テキスト形式のネットリストの終端に印を付けます。この行以降のすべての行が無視されます。この指令は終端でネットリスト・エクストラクタによって入力されるので、回路図にテキストとして配置しないようにしてください。

.DC – DC 信号源による掃引解析を実行する

信号源の DC 値を掃引している間に直流解析を行いません。

アンプの DC 伝達関数を計算することや、トランジスタの特性曲線をプロットしてモデルを検証するのに役立ちます。

```
構文:. dc <srcnam> <Vstart> <Vstop> <Vincr>  
      + [<srcnam2> <Vstart2> <Vstop2> <Vincr2>]
```

<srcnam>は、<Vstart>から<Vstop>までステップ幅<Vincr>で掃引する独立した電圧源または電流源です。次の例では、デフォルトの BSIM3v3.2.4 特性曲線がプロットされます。

```
* 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
```

.END -- ネットリストの終端

この指令は、テキスト形式のネットリストの終端に印を付けます。この行以降のすべての行が無視されます。この指令は終端でネットリスト・エクストラクタによって入力されるので、回路図にテキストとして配置しないようにしてください。

.ENDS -- サブ回路定義の終端

この指令は、サブ回路定義の終端に印を付けます。詳細については、[.SUBCKT](#)を参照してください。

.FOUR -TRAN 解析後にフーリエ成分を計算する

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

例: `.four 1kHz V(out)`

このコマンドはトランジェント解析後に実行されます。このコマンドが用意されているのは、旧来の SPICE シミュレータとの互換性を維持するためです。このコマンドからの出力は.log ファイルに出力されます。出力を表示するには、メニュー項目「View」=>「Spice Error Log」を使用します。ほとんどの目的では、波形ビューワに組み込まれている FFT 機能の方が便利です。

整数 Nharmonics を指定すると、その次数の高調波が解析に組み込まれます。指定しなかった場合、高調波のデフォルトの次数は 9 になります。

整数 Nperiods を Nharmonics の後に指定しなかった場合、フーリエ解析は最終時刻 (Tend) から Tend より 1 周期前までの期間に実行されます。Nperiods を -1 と指定すると、フーリエ解析はシミュレーション・データ範囲全体にわたって実行されます。

.FUNC -- ユーザ定義関数

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

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

.func 指令を使用すると、ユーザによるパラメータ化回路と動作信号源と組み合わせてユーザ定義関数を作成できます。これは、関数に名前を関連付けて分かりやすくすることと、サブ回路をパラメータ化して抽象回路をライブラリに保存できるようにするのに役立ちます。

.func 文をサブ回路定義の内部に組み込むことにより、関数の有効範囲を、対象のサブ回路と、そのサブ回路が呼び出す複数のサブ回路に制限できます。

これらのユーザ定義関数を使用してパラメータ置換評価と式の評価を呼び出すには、式を中括弧で囲みます。囲まれた式は浮動小数点値に置き換えられます。

.func 文と.param 文の両方を使用する例を以下に示します。

```
.func 文を使用した deck の例
.func myfunc(x,y) {sqrt(x*x+y*y)}
.param u=100 v=600
V1 a 0 pulse(0 1 0 1n 1n .5μ 1μ)
R1 a b {myfunc(u,v/3)}
C1 b 0 100p
.tran 3μ
.end
```

パラメータ置換評価がすべて完了してからシミュレーションが始まります。

.FERRET - 指定の URL でファイルをダウンロードする

このコマンドを使用すると、URL を指定することにより、ファイルをバッチ・モードでダウンロードできます。これが便利なのは、すべてのファイルにブラウザを向ける必要はない場合や、ブラウザがファイルを変換しないようにする場合です。ダウンロードしたファイルは、元の回路図またはネットリストと同じディレクトリに入ります。このコマンドはシミュレーションに対しては効力がありません。

.GLOBAL – グローバル・ノードを宣言する

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

例: `.global VDD VCC`

.global コマンドを使用すると、サブ回路に記述された特定のノードがサブ回路に限定されず、絶対的なグローバル・ノードであると宣言できます。グローバルな回路共通ノードはノード「0」で、.global 文が必要ないことに注意してください。また、「\$G_」という書式のノード名も、.global 文で記述しないグローバル・ノードです。

.IC – 初期条件を設定する

.ic 指令を使用すると、トランジェント解析の初期条件を指定できます。ノード電圧およびインダクタ電流を指定する場合があります。DC 解法は、初期条件を制約条件として使用して実行します。インダクタは他の SPICE プログラムの DC 解法では短絡とみなされますが、初期電流を指定した場合、LTspice では無限インピーダンスの電流源とみなされることに注意してください。

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

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

.INCLUDE – 別のファイルを組み込む

構文: `.include <filename>`

この指令を使用すると、指定のファイルが `.include` コマンドの代わりにネットリストに入力されたかのように組み込まれます。これはモデルまたはサブ回路のライブラリを組み込むのに役立ちます。

ファイル名として絶対パス名を入力してもかまいません。そうしなかった場合、LTspice は、まず `<LTspiceIV> %lib%sub` ディレクトリを参照し、その後、呼び出し側のネットリストが入っているディレクトリを参照します。ここで、`<LTspiceIV>` は `scad3.exe` 実行可能ファイルが入っているディレクトリで、通常は `C:%Program Files%LTC%LTspiceIV` としてインストールされています。

ファイル名拡張子は想定されません。ファイルを「`myfile.lib`」で呼び出す場合は、「`.inc myfile`」ではなく「`.inc myfile.lib`」を使用する必要があります。

ファイル名として次の書式の URL を指定できます。

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

ファイル「`library.lib`」は回路ディレクトリに `http` 転送され、組み込まれます。

その後のシミュレーションでは、シミュレーションを実行するたびにファイルをダウンロードしないように、`.inc` 文を編集して `.inc library.lib` にすることができます。

指定した URL が存在しない場合、ほとんどの Web サーバはエラーを返しません、そのエラーについて説明する HTML Web ページを返し、それが Web ブラウザに表示されることに注意してください。

LTspice は必ずしもこれらのページをエラー状態と解釈できないので、有効な SPICE 構文としてシミュレーションに組み込まれている HTML 言語エラー・ページをシミュレーションが続行しようとする、不可解なエラー・メッセージが表示されることがあります。

.LIB -- ライブラリを組み込む

構文: .lib <filename>

この指令を使用すると、指定のファイルが.lib コマンドの代わりにネットリストに入力されたかのように、そのファイルのモデル定義とサブ回路定義が組み込まれます。グローバル・スコープでは回路素子が無視されます。

ファイル名として絶対パス名を入力してもかまいません。そうしなかった場合、LTspice は、まず <LTspiceIV>%lib%cmp ディレクトリを参照し、その後<LTspiceIV>%lib%sub を、さらに呼び出し側のネットリストが入っているディレクトリを参照します。ここで、<LTspiceIV>は scad3.exe 実行可能ファイルが入っているディレクトリで、通常は C:%Program Files%LTC%LTspiceIV としてインストールされています。

ファイル名拡張子は想定されません。ファイルを「myfile.lib」で呼び出す場合は、「.lib myfile」ではなく「.lib myfile.lib」を使用する必要があります。ファイル名として次の書式の URL を指定できます。

<http://www.company.com/models/library.mod>

ファイル「library.mod」は回路ディレクトリに http 転送され、ライブラリとして組み込まれます。その後のシミュレーションでは、シミュレーションを実行するたびにファイルをダウンロードしないようにするために、.lib 文を編集して.lib library.mod にすることができます。

指定した URL が存在しない場合、ほとんどの Web サーバはエラーを返しません、そのエラーについて説明する HTML Web ページを返し、それが Web ブラウザに表示されることに注意してください。LTspice は必ずしもこれらのページをエラー状態と解釈できないので、有効な SPICE 構文としてシミュレーションに組み込まれている HTML 言語エラー・ページをシミュレーションが続行しようとする、不可解なエラー・メッセージが表示されることがあります。

暗号化ライブラリ

LTspice は、特殊な形式の暗号化ライブラリの生成および読み取りが可能です。これにより、あるユーザがライブラリを準備し、それを別のユーザがシミュレーションで使用できます。このとき、ライブラリの実装を明らかにする必要はありません。無許可の関係者による暗号化ライブラリの解読を困難にすることを無理のない範囲で試みましたが、ソフトウェアに暗号化ライブラリが実装されているという理由だけで完全に安全であるとみなすことはできません。

暗号化ライブラリを準備するには、コマンド・ライン・オプション「-encrypt」を使用して、コマンド・ラインから LTspice を起動する必要があります。ライブラリは暗号化バージョンに置き換えられるので、まずライブラリをバックアップする必要があります。暗号化ライブラリを平文に戻すユーティリティはありません。2 つの手順の概要を以下に示します。

1. ライブラリのバックアップ・コピーを作成します。暗号化したバージョンは削除されます。
2. コマンド・ラインで、次のコマンドを入力します。

```
scad3.exe -encrypt <filename>
```

ファイル<filename>は、暗号化バージョンに置き換えられます。暗号化処理には数分かかります。

この処理が終了すれば、暗号化 ASCII ファイルが得られます。「* Begin:」行の上に著作権表示を追加することは可能ですが、ファイルの先頭の 9 行は未変更のままにして、追加する著作権表示の各行は文字「*」から始める必要があります。

つまり、LTspice によって作成された次の暗号化ファイルを考えると、

```
* LTspice Encrypted File
*
* This encrypted file has been supplied by a 3rd
* party vendor that does not wish to publicize
* the technology used to implement this library.
*
* Permission is granted to use this file for
* simulations but not to reverse engineer its
* contents.
* Begin:
50 3E 46 0F FA 6E 67 FF B8 4D D9 62 14 32 60 24
36 71 35 0B 66 4F AD 52 B8 F5 9E 22 9F C0 18 8B
FB FE 1D...
```

これを次のように変更できます。

```
* LTspice Encrypted File
*
* This encrypted file has been supplied by a 3rd
* party vendor that does not wish to publicize
* the technology used to implement this library.
*
* Permission is granted to use this file for
* simulations but not to reverse engineer its
* contents.
*
```

* Copyright © 2005 Acme SPICE Modeling

* For additional information, see

* <http://www.acmespicemodels.com/>

*

* Begin:

50 3E 46 0F FA 6E 67 FF B8 4D D9 62 14 32 60 24

36 71 35 0B 66 4F AD 52 B8 F5 9E 22 9F C0 18 8B

FB FE 1D...

.LOADBIAS – 以前に解かれた DC 解法を読み込む

構文: `.loadbias <filename>`

.loadbias コマンドは、.savebias コマンドの補完コマンドです。まず、.savebias コマンドを実行するシミュレーションを実行します。その後、.savebias コマンドを.loadbias コマンドに変更します。

.MEASURE – ユーザ定義の電気量を評価する

異なる 2 種類の基本的な .MEASURE 文があります。横軸方向の 1 点を指す .MEASURE 文 (水平軸、つまり .tran 解析の時間軸に沿ってプロットされた独立変数) と、横軸上の範囲を指す .MEASURE 文です。1 種類目の横軸上の 1 点を指す文は、特定の 1 点のデータ値または式、あるいは条件を満たした場合のデータ値または式を出力するために使用します。次の構文を使用します。

```
構文:  .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 文が当てはまる解析のタイプは自由に記述できます。これにより、特定の .MEAS 文を特定の解析タイプに対してのみ使用できます。結果に対して、他の .MEAS 文で使用できるパラメータ名を付ける場合は、name が必要です。横軸方向の 1 点を指す .MEAS 文の例を以下に示します。

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

t=5ms での V(out)の値を res1 とラベル付けして出力します。

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

条件 $V(x)=3*V(y)$ を初めて満たしたときに式 $V(out)*I(Vout)$ の値を出力します。これは res2 とラベル付けされます。

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

条件 $V(x)=3*V(y)$ を 3 回満たしたときに V(out)の値を出力します。これは res3 とラベル付けされません。

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

V(x)が増加して $3*V(y)$ に近づいている場合で、条件 $V(x)=3*V(y)$ を最後に満たしたときに V(out)の値を出力します。これは res4 とラベル付けされます。

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

条件 $V(x)=3*V(y)$ を満たした 3 回目に V(out)の値を出力しますが、時間が 1ms 経過するまではカウントを開始しません。これは res5 とラベル付けされます。

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

$3*res1/res2$ の値を出力します。この書式は他の .meas 文の式の結果を出力するのに役立ちます。V(3)のような直接のシミュレーション・データに基づいた式が評価対象の式に存在することを意味する

わけではありませんが、そうした式が存在する場合、データは最後のシミュレーション実行点から得られます。この結果は res6 とラベル付けされます。

上記の例では、横軸方向の 1 点を指している間、要求結果は縦軸のデータ(従属変数)に基づくので注意してください。縦軸情報が要求されない場合は、その後、測定条件が生じる横軸上の点が .MEAS 文によって出力されます。

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

条件 $V(x)=3*V(y)$ を満たした最初の時刻を出力します。これは res6 とラベル付けされます。

別のタイプの .MEAS 文は、横軸上の範囲を指します。次の構文を使用します。

```
構文: .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>]
      + [TARG <lhs2> [[VAL]=<rhs2>] [TD=<val2>]
      + [<RISE|FALL|CROSS>=<count2>]
```

横軸上の範囲は、「TRIG」および「TARG」で定義した点を使用して指定します。TRIG 点を省略した場合は、シミュレーションの開始がデフォルトになります。同様に、TARG 点を省略した場合は、シミュレーションデータの終了がデフォルトになります。TRIG、TARG、前の WHEN の 3 点をすべて省略した場合、.MEAS 文はデータの全範囲にわたって動作します。ある一定の間隔にわたって実行できる測定動作のタイプは以下のとおりです。

キーワード	一定の間隔にわたって行われる動作
AVG	<expr>の平均を計算します。
MAX	<expr>の最大値を検索します。
MIN	<expr>の最小値を検索します。
PP	<expr>のピーク・トゥ・ピーク値を検索します。
RMS	<expr>の 2 乗平均を計算します。
INTEG	<expr>を積分します。

測定動作を指定しなかった場合、.MEAS 文の結果は、TRIG 点と TARG の点の間の横軸方向の距離になります。間隔 .MEAS 文の例を以下に示します。

```
.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
```

1.1us 経過後に V(NS05)が初めて 1.5V に低下するまで、および 1.1us 経過後に V(NS03)が初めて 1.5V に低下するまでの V(NS01)の平均値を出力します。これは res7 とラベル付けされます。

.AC 解析では、複素数データの条件式が、式の複素数値の実数部だけを考慮することにより、実数の条件に変換されます。

また、.MEAS 文の結果を別の .MEAS 文で使用することもできます。次の例では、3dB 帯域幅が計算されます。

.MEAS AC tmp max mag(V(out)); ピーク応答を見つけて、それを「tmp」と呼ぶ

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

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

ピーク応答点から 3dB 低下した点までの 2 点間の周波数の差を出力します。

注記: .AC 解析から得られるデータは複素数であり、.measurement 文の結果も同様です。ただし、等式は複素数の実数部のみを表します。つまり、「mag(V(out))=tmp/sqrt(2)」は「Re(mag(V(out)))=Re(tmp/sqrt(2))」と等価です。

ノイズは求積法により全周波数にわたって大幅に積分されるので、.NOISE 解析では、AVG 動作、RMS 動作、INTEG 動作の解析タイプが異なります。したがって、AVG と RMS ではどちらも RMS ノイズ電圧が得られ、INTEG では全積分ノイズが得られます。

したがって、SPICE 指令を追加した場合は、「when <cond1> = <cond2>」のような条件をテストする場合に必要なのは、条件が等式を満たすことではなく、条件を等式で検証することです。このことは、数値の保管に使用する有限の歳差により、浮動小数点式が必要ないという事実に関係があります。

.MEAS NOISE out_totn INTEG V(onoise)

.MEAS NOISE in_totn INTEG V(inoise)

.noise 解析では、入力および出力を基準とした全積分 RMS ノイズが .log ファイルに出力されます。

.MEAS 文は、シミュレーションが完了した後の後処理で実行されます。これにより、.MEAS 文のスクリプトを作成して、データセット上で実行することができます。このためには、波形ウィンドウをアクティブ・ウィンドウにして、メニューコマンド「File」=>「Execute .MEAS Script」を実行します。シミュレーション後の後処理で行われる .MEAS 文の別の結果は、圧縮後に .MEAS 文出力の精度が波形データの精度によって制限されていることを示しています。より正確な .MEAS 文を出力するために圧縮設定を調整することが必要な場合があります。

.MEAS 文は、通常は SPICE 指令として回路図にそのまま記述されます。あるいは残りのシミュレーション・コマンドと回路定義を使用してネットリスト内に記述されます。出力は.log ファイルに格納されますが、このファイルはメニューコマンド「View」=>「SPICE Error Log」で表示できます。シミュレーションに.step コマンドが含まれている場合、.measure 文はステップごとに実行され、結果は.log ファイル内に表として出力されます。.measure 文の結果に対応するこれらの表は、以下の手順によって正規波形のようにプロットできます。

- i) シミュレーションが完了したら、メニューコマンド「View」=>「SPICE Error Log」を実行します。
- ii) .log ファイルの内側で右クリックして、コンテキスト・メニュー・コマンド「Plot .step'ed .meas data」を実行します。

.MODEL – SPICE モデルを定義する

ダイオード、トランジスタ、スイッチ、損失の大きい伝送線路、または均一な RC 線路のモデルを定義します。

一部の回路素子(たとえば、トランジスタ)には、多くのパラメータがあります。トランジスタのすべての具体例に対してすべてのトランジスタ・パラメータを定義する代わりに、トランジスタをモデル名によって分類し、パラメータを共通にしています。同じモデルのトランジスタに異なるサイズを設定可能であり、電気的な動作は具体例のサイズに対応します。

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

パラメータ・リストは、モデルのタイプにより異なります。モデル・タイプのリストを以下に示します。

タイプ	関連付けられた回路素子
SW	電圧制御型スイッチ
CSW	電流制御型スイッチ
URC	均一分布の RC 線路
LTRA	損失の大きい伝送線路
D	ダイオード
NPN	NPN バイポーラ・トランジスタ
PNP	PNP バイポーラ・トランジスタ
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
PMOS	P チャネル MOSFET
NMF	N チャネル MESFET
PMF	P チャネル MESFET
VDMOS	縦型二重拡散パワーMOSFET

具体例に固有のパラメータと、モデルにとって一般的なパラメータを示すリストについては、回路素子の説明を参照してください。

.NET – .AC 解析で回路網パラメータを計算する

この文は小信号(.AC)解析と組み合わせて使用し、2ポート回路網の入力アドミタンス、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パラメータを計算します。また、この文を使用して1ポート回路網の入力アドミタンスおよびインピーダンスを計算することもできます。この文は、回路網解析の周波数掃引を決定する.AC文と組み合わせて使用する必要があります。

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

回路網入力は、独立した電圧源<Vin>または独立した電流源<lin>を指定します。オプションの出力ポートは、ノード V(out)または抵抗 I(Rout)のいずれかで指定します。ポートは抵抗 Rin および Rout で終端します。終端インピーダンスは、指定しなかった場合、デフォルトで 1Ω になりますが、Rser を指定した電圧源の場合、または抵抗を指定した出力ポートの場合は除きます。これら2つの場合、終端抵抗はデフォルトでデバイス・インピーダンスになります。.NET 文で指定した終端値は、.NET 計算ではデバイス・インピーダンスより優先されますが、通常の.AC ノード電圧および電流では優先されません。つまり、.AC 解析の一部として計算される通常の電圧および電流の場合は、.NET 文によって回路網に終端インピーダンスが加わることはありません。

通常は C:\Program Files\LTC\LTspiceIV\examples\Educational\S-param にインストールされるサンプル・ファイルを参照してください。このファイルでは、Rser を目的の信号源インピーダンスに設定した電圧源(V4)と、「.net I(Rout) V4」をそのまま読み取る.NET 文で出力終端を設定する抵抗(Rout)を使用することを推奨しています。.net 文で指定した Rin または Rout の値と入出力デバイスによってデフォルトの終端値が決まることはありません。この配置により、.AC 解析のノード電圧および電流は、.NET 文の場合と同じ方法で終端される回路網と一致するようになります。

.NODESET -- 最初の DC 解法のヒントを示す

.nodeset 指令は、DC 動作点を見つけるためのヒントを示します。フリップフロップなどのように回路に複数の DC 状態が考えられる場合、DC 解法を見つけるための繰り返し過程は収束しない可能性があります。nodeset 指令を使用して、回路をいずれかの状態にすることができます。基本的に、nodeset 指令で指定した電圧で解法を 1 回実行した後は、その後の繰り返し処理に備えて制約条件が取り除かれます。

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

.NOISE – ノイズ解析を実行する

これは、ジョンソン・ノイズ、ショット・ノイズ、およびフリッカ・ノイズによるノイズを計算する周波数領域の解析です。出力データは単位帯域幅の平方根当たりのノイズ・スペクトラム密度です。

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

`V(<out>[,<ref>])`は全出力ノイズを計算するノードです。これを $V(n1, n2)$ と表現して、2つのノード間の電圧を表わすことができます。`<src>`は入力ノイズの基準となる独立した信号源の名前です。`<src>`はノイズのない入力信号です。パラメータ`<oct, dec, lin>`、`<Nsteps>`、`<StartFreq>`、および`<EndFreq>`は、.AC 指令で使用される方法で対象の周波数範囲と分解能を定義します。

出力データ・トレース `V(onoise)`は、上記の構文で出力として指定したノードを基準にしたノイズ・スペクトラム電圧密度です。入力信号を電圧源として与えた場合、データ・トレース `V(inoise)`は入力換算のノイズ電圧密度になります。入力を電流源として指定した場合、データ・トレース `inoise` は入力電流源信号を基準にしたノイズになります。各部品のノイズ寄与分はグラフ化することができます。これらの寄与分は出力を基準にしています。データ・トレース「gain」で除算することにより、寄与分を入力基準にすることができます。

波形ビューワで帯域幅のノイズを積分するには、対応するデータ・トレース・ラベルを、<Ctrl>キーを押しながら左マウス・ボタンでクリックします。

.OP – DC 動作点を見つける

コンデンサを開回路にしてインダクタを短絡し、DC 動作点解決法を実行します。通常、DC 解法は回路の動作点を見つけるために別の解析の一部として行なわれます。この動作点だけを見つける場合は、.op を使用しません。結果はダイアログボックスに表示されます。.OP シミュレーション後にノードまたは電流を指し示すと、.OP 解決法がステータス・バーに表示されます。

ニュートン・ラプソン反復法で行われるように、連続線形近似で一般的な非線形回路の動作点を見つけることができるという保証はありません。直接ニュートン反復法ではうまくいかない場合、LTspice は、他のいくつかの方法で動作点を見つけようとします。使用する方法と、特定の方法を無効にするのに必要なオプション設定の表を以下に示します。

キーワード	データ・タイプ	デフォルト値	概要
abstol	Num.	1pA	電流誤差の絶対許容差
baudrate	Num.	(なし)	アイ・ダイアグラムに使用されます。横軸の時間を回り込ませてビット遷移に重ね合わせる方法を波形ビューワに指示します。
chgtol	Num.	10fC	電荷の絶対許容差
cshunt	Num.	0	各ノードとグラウンドの間に追加されるオプションの容量
cshuntintern	Num.	cshunt	各デバイス内部のノードとグラウンドの間に追加されるオプションの容量
defad	Num.	0	MOS ドレインのデフォルトの拡散面積
defas	Num.	0	MOS ソースのデフォルトの拡散面積
defl	Num.	100 μ m	MOS のデフォルトのチャンネル長
defw	Num.	100 μ m	MOS のデフォルトのチャンネル幅
delay	Num.	0	アイ・ダイアグラムに使用されます。
fastaccess	flag	FALSE	シミュレーションの終了時に Fast Access ファイル形式に変換します。
flagloads	flag	FALSE	外部電流源に負荷としてフラグを立てます。
Gmin	Num.	1.00E-12	収束しやすくするためにすべての PN 接合に追加される導電率
gminsteps	Num.	25	最初の DC 解法に対して Gmin ステッピングを行わないようにする場合はゼロに設定します。
gshunt	Num.	0	各ノードとグラウンドの間に追加されるオプションの導電率
itl1	Num.	100	DC 反復回数の限度

itl2	Num.	50	DC 伝達曲線の反復回数の限度
itl4	Num.	10	トランジェント解析時点での反復回数の限度
itl6	Num.	25	最初の DC 解法に対して信号源ステップングを行わないようにする場合はゼロに設定します。
srcsteps	Num.	25	itl6 の代替名
maxclocks	Num.	Infin.	保存するクロック・サイクルの最大値
maxstep	Num.	Infin.	トランジェント解析での最大ステップ幅
meascplxfmt	string	bode	.meas 文の結果の複素数形式。「polar」、「cartesian」、「bode」のいずれかです。
measdgt	Num.	6	.measure 文の出力に使用される有効数字の数
method	string	trap	数値積分法、台形法または Gear 法
minclocks	Num.	10	保存するクロック・サイクルの最小値
MinDeltaGmin	Num.	1.00E-04	適応 Gmin ステップングの終了に対して限度を設定します。
nomarch	flag	FALSE	行進型波形をプロットしないようにします。
noopiter	flag	FALSE	Gmin ステップングに直接移動します。
numdgt	Num.	6	これまで、「numdgt」は出力データに使用した有効数字の数を設定するために使用していました。LTspice では、「numdgt」を 6 より大きい値に設定すると、従属変数データには倍精度数が使用されます。
pivrel	Num.	1.00E-03	最大の列項目と許容できるピボット値の間の相対比
pivtol	Num.	1.00E-13	ピボットとして認められるマトリクス項目の絶対最小値
reltol	Num.	0.001	相対誤差の許容差
srcstepmethod	Num.	0	開始時に使用する信号源ステップング・アルゴリズム
sstol	Num.	0.001	定常状態を検出するための相対誤差
startclocks	Num.	5	定常状態を探す前に待つクロック・サイクルの数
temp	Num.	27°C	温度を指定しない回路素子インスタンスのデフォルトの温度
tnom	Num.	27°C	デフォルトの温度を規定していないモデルに対して、デバイス・パラメータを測定したときの温度
topologycheck	Num.	1	フロート・ノード、電圧源のループ、および非物理的なトランス巻線配置の検査をスキップする場合は 0 に設定します。

trtol	Num.	1	一時的な誤差の許容差を設定します。このパラメータは実際の切り捨て誤差が過大評価される係数の推定値です。
trytocompact	Num.	1	0 以外の場合、シミュレータは LTRA 伝送線路の入力電圧および電流の履歴を圧縮しようとします。
vntol	Num.	1 μ V	絶対電圧誤差の許容差を設定します。
plotreltol	Num.	0.0025	波形圧縮に対する相対誤差の許容差を設定します。
plotvntol	Num.	10 μ V	波形圧縮に対する絶対電圧誤差の許容差を設定します。
plotabstol	Num.	1nA	波形圧縮に対する絶対電流誤差の許容差を設定します。
plotwinsize	Num.	300	1つのウィンドウ内で圧縮するデータ点の数。圧縮を無効にする場合は0に設定します。
ptrantau	ptrantau	0.1	減衰疑似トランジェント解析で動作点を見つけるための特性信号源の起動時間。疑似トランジェント解析を無効にする場合は0に設定します。
ptranmax	Num.	0	0以外に設定した場合は、回路が安定しているかどうかにかかわらず、減衰疑似トランジェント解析のその時間が動作点として使用されます。

.PARAM – ユーザ定義パラメータ

.param 指令を使用すると、ユーザ定義変数を作成できます。これは、値に名前を関連付けて分かりやすくすることと、サブ回路をパラメータ化して抽象回路をライブラリに保存できるようにするのに役立ちます。

.param 文をサブ回路定義の内部に組み込むことにより、パラメータ値の有効範囲を対象のサブ回路に制限できます。

パラメータ置換評価と式の評価を呼び出すには、式を中括弧で囲みます。囲まれた式は浮動小数点値に置き換えられます。

サブ回路の呼出し行に.param 文と直接渡すパラメータの両方を使用する例を以下に示します。

```
*
*   これは回路定義です
.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 .5μ .5μ 0 1μ)

*   これはサブ回路の定義です
subckt divider n1 n2 n3
r1 n1 n2 {top}
r2 n2 n3 {bot}
.ends
*

.tran 3μ
.end
```

パラメータ置換方式は記号宣言型言語です。パラメータは評価済みの値としてはサブ回路に渡されませんが、式自体および関係自体によって渡されます。中括弧がある場合、括弧で囲まれた式は、有効範囲で使用可能なすべての関係に基づいて評価され、浮動小数点値に換算されます。

以下の関数および演算を使用できます。

関数名	概要
abs(x)	x の絶対値

acos(x)	x の逆余弦の実数部。たとえば、acos(-5)の結果は 3.14159 であり、3.14159+2.29243i ではありません。
arccos(x)	acos()と同義
acosh(x)	x の双曲線余弦の実数部。たとえば、acosh(.5)の結果は 0 であり、1.0472i ではありません。
asin(x)	x の逆正弦の実数部。たとえば、asin(-5)の結果は-1.57080, であり、-1.57080+2.29243i ではありません。
arcsin(x)	asin()と同義
asinh(x)	逆双曲線正弦
atan(x)	x の逆正接
arctan(x)	atan()と同義
atan2(y, x)	y/x の 4 象限逆正接
atanh(x)	逆双曲線正接
buf(x)	x>.5 の場合は 1、そうでない場合は 0
cbt(x)	(x)の立方根
ceil(x)	x 以上の整数
cos(x)	x の余弦
cosh(x)	x の双曲線余弦
exp(x)	e の x 乗
fabs(x)	abs(x)と同じ
flat(x)	x と一様分布がある x の間の乱数
floor(x)	x 以下の整数
gauss(x)	シグマが x のガウス分布を基にした乱数
hypot(x,y)	sqrt(x**2 + y**2)
if(x,y,z)	x>.5 の場合は y、そうでない場合は z
int(x)	x を整数に変換
inv(x)	x>.5 の場合は 0、そうでない場合は 1
limit(x,y,z)	x、y、z の中間値
ln(x)	x の自然対数
log(x)	ln()の代替構文
log10(x)	10 を底とする対数
max(x,y)	x と y のうち大きい方の値
mc(x,y)	一様分布での x*(1+y)と x*(1-y)の間の乱数
min(x,y)	x と y のうち小さい方の値
pow(x,y)	x**y の実数部。たとえば、pow(-.5,1.5)の結果は 0 であり、0.353553i ではありません。

pwr(x,y)	abs(x)**y
pwr(x,y)	sgn(x)*abs(x)**y
rand(x)	x の整数値に依存する 0 と 1 の間の乱数
random(x)	rand()と同様だが、値の間を滑らかに遷移
round(x)	x に最も近い整数
sgn(x)	x の符号
sin(x)	x の正弦
sinh(x)	x の双曲線正弦
sqrt(x)	x の平方根の実数部。たとえば、sqrt(-1)の結果は 0 であり、0.707107i ではありません。
table(x,a,b,c,d,...)	一連の点の組として与えられた参照表に基づいて x の値を内挿する
tan(x)	x の正接
tanh(x)	x の双曲線正接
u(x)	単位ステップ、つまり x>0 の場合は 1、そうでない場合は 0
uramp(x)	x>0 の場合は x、そうでない場合は 0

以下の演算は評価の優先順位の逆の順に要約しています。

オペランド	概要
&	式の両辺をブール値に変換後、論理積をとる
	式の両辺をブール値に変換後、論理和をとる
^	式の両辺をブール値に変換後、排他的論理和をとる
>	式の左辺が式の右辺より大きい場合は true、そうでない場合は false
<	式の左辺が式の右辺より小さい場合は true、そうでない場合は false
>=	式の左辺が式の右辺以下である場合は true、そうでない場合は false
<=	式の左辺が式の右辺以上である場合は true、そうでない場合は false
+	浮動小数点の加算
-	浮動小数点の減算
*	浮動小数点の乗算
/	浮動小数点の除算
**	左辺を右辺で累乗する。実数部だけが返される。たとえば、-2**1.5 の結果は 0 であり、2.82843i ではありません。

パラメータ置換評価がすべて完了してからシミュレーションが始まります。

.SAVE – 保存済みデータの数量を制限する

一部のシミュレーション、特に時間領域のシミュレーションでは、大量のデータが生成されることがあります。.save 指令を使用して、対象となる特定のノード電圧およびデバイス電流だけを保存することにより、出力の量を制限できます。

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

指令.save I(Q2)は、バイポーラ・トランジスタ Q2 のベース電流、コレクタ電流、およびエミッタ電流を保存します。単一の端子電流を保存するには、Ic(Q2)を指定します。

ワイルドカード文字「*」および「?」を使用してパターンと一致するデータ・トレースを指定できます。たとえば、「.save V(*) Id(*)」を実行すると、すべての電圧とすべてのドレイン電流が保存されます。

キーワード「dialogbox」を指定すると、ダイアログボックスが表示され、その中に選択可能なすべてのデフォルト・ノードおよび電流のリストが表示されるので、ユーザはどれを保存するかをリストから選択できます。ネットリストが回路図から生成されていた場合は、回路図内でノードとデバイスをポイントしてクリックし、ダイアログボックスで選択したように強調表示することができます。

.SAVEBIAS-動作点をディスクに保存する

構文: `.savebias <filename> [internal]`
`+ [temp=<value>] [time=<value> [repeat]] [step=<value>]`
`+ [DC1=<value>] [DC2=<value>] [DC3=<value>]`

このコマンドを実行すると、この後のシミュレーションでの `.loadbias` コマンドによって再度読み込まれるテキスト・ファイルがディスクに書き込まれます。解くのが困難な DC 動作点がある回路の場合は、残りのシミュレーションに移る前に次の解析が DC 解法を見つける時間を節約できるように、DC 解法をディスクに保存できません。

キーワード「`internal`」を追加して、いくつかのデバイスの内部ノードも維持することを示し、より完成度の高い DC 解法が維持されるようにすることができます。

`.tran` 解析からの特定の DC 動作点を保存する場合は、時刻を指定できます。指定した時刻の後、最初に解いた時点が書き込まれます。修飾子「`repeat`」を指定すると、この時刻で指定されたすべての期間の後に DC 解法が書き込まれます。ファイルに入るのは直前に解かれた DC 動作点だけです。DC1、DC2、および DC3 を指定すると、`.dc` 掃引解析から単一の動作点を抽出できます。

`savebias` コマンドでは、テキスト・ファイルが `.nodeset` コマンドの形式で書き込まれます。

`nodeset` 文は解法の推奨にすぎないことに注意してください。つまり、ソルバは、`nodeset` 文で指定されたノード電圧で解法の繰り返しを開始しますが、解法が有効であるという条件を満たすまで繰り返し続けます。DC 動作点から `.tran` 解法を再開する場合は、ファイルを `.nodeset` から `.ic` に編集して、ソルバがこの DC 状態から開始するよう強制できます。

すべての回路リアクタンスの積分状態が `.savebias` ファイルに保存されるわけではないので、この技法で成功するかどうかは確実ではありません。

.STEP – パラメータの掃引

このコマンドを実行すると、温度、モデルパラメータ、グローバル・パラメータ、または独立した信号源を 1 ステップずつ変化させる間に解析が繰り返し行なわれます。ステップは線形でも対数でも構いません。値のリストとして指定することもできます。

例: `.step oct v1 1 20 5`

独立した電圧源 V1 を 1 から 20 まで 1 オクターブ当たり 5 点の割合で対数的に 1 ステップずつ変化させます。

例: `.step I1 10u 100u 10u`

独立した電流源 I1 を 10uA から 100uA まで 10uA 刻みで増やします。

例: `.step param RLOAD LIST 5 10 15`

グローバル・パラメータ Rload を 5、10、および 15 に設定してシミュレーションを 3 回実行します。

例: `.step NPN 2N2222(VAF) 50 100 25`

NPN モデルパラメータ VAF を 50 から 100 まで 25 の刻みで変化させます。

例: `.step temp -55 125 10`

温度を-55°Cから 125°Cまで 10°C刻みで変化させます。ステップ掃引は最大 3 階層までネストできます。

.SUBCKT -- サブ回路を定義する

回路の定義を支援する方法として、繰り返し出現する回路をサブ回路定義で囲み、同じ回路内で複数のインスタンスとして使用することができます。シミュレーションが動作する前に、サブ回路の各呼び出しをサブ回路定義内の回路素子に置き換えることにより、回路はフラットなネットリストに展開されます。サブ回路のサイズまたは複雑さに制限はありません。

サブ回路定義の最後は.ends 指令にする必要があります。

サブ回路を使用した例を以下に示します。

```
*
* これは回路定義です
X1 a b 0 divider
V1 a 0 pulse(0 1 0 .5μ .5μ 0 1μ)
* これはサブ回路の定義です
.subckt divider n1 n2 n3 r1 n1 n2 1k
r2 n2 n3 1k
.ends
.tran 3
.end
```

このシミュレーションは以下の拡張後に動作します。

```
* X1 を 2 つの抵抗回路網に展開します
r:1:1 a b 1k
r:1:2 b 0 1k
*
v1 a 0 pulse(0 1 0 .5μ .5μ 0 1μ)
.tran 3μ
.end
```

サブ回路の展開によって挿入される回路素子に対しては、サブ回路名およびサブ回路定義の素子名に基づいて固有の名前が作成されることに注意してください。

.TEMP-- 温度の掃引

これは温度に対するステップ・コマンドの古い書式です。示した温度ごとにシミュレーションを実行します。

次の構文

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

は、次の構文と等価です。

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

.TF – DC 小信号伝達関数を見つける

これは、個別の信号源の小幅なばらつきにより、ノード電圧または分岐電流の DC 小信号伝達関数を見つける解析モードです。

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

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

.TRAN – 非線形のトランジェント解析を実行する

トランジェント解析を実行します。これは回路の最も直接的なシミュレーションです。この解析では、基本的に、回路に電源を投入したときに何が起こるかを計算します。テスト信号は、多くの場合、独立した信号源として入力されます。

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

最初の書式は従来の.tran SPICE コマンドです。Tstep は、波形のプロット時の増分ですが、最初のステップ幅の推測値としても使用されます。LTspice は波形の圧縮を使用するので、このパラメータの値はきわめて低く、省略しても 0 に設定してもかまいません。Tstop はシミュレーションの期間です。

トランジェント解析は、必ず 0 に等しい時刻で開始します。ただし、Tstart を指定した場合、0 から Tstart までの間の波形データは保存されません。これは、起動時の過渡現象を無視できるようにすることで、波形ファイルのサイズを管理する方法です。最後のパラメータ dTmax は、回路の式を積分する間にかげられる最大の時間刻みです。Tstart または dTmax を指定する場合は、Tstep を指定する必要があります。

.tran 行にはいくつかの [modifiers\(修飾子\)](#)を置くことができます。

.WAVE – 選択したノードを.Wav ファイルに記述する

LTspice は.wav オーディオ・ファイルを記述できます。これらのファイルは、後で聞くことも、別のシミュレーションの入力として使用することもできます。

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

例: `.wave C:%output.wav 16 44.1K V(left) V(right)`

<filename.wav>は、作成する.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またはネットリストが入っているディレクトリを基準に計算した相対パスです。二重引用符を使用してスペースを含むパスを指定できます。<Nbits>はサンプリング・ビットの数です。有効な範囲は 1~32 ビットです。

<SampleRate>はシミュレーションの所要時間(秒)当たりの記述対象サンプル数です。有効な範囲は 1~4294967295 サンプル/秒です。残りの構文は、保存するノードを示しています。各ノードは.wav ファイル内の独立したチャンネルになります。チャンネルの数は最小で 1、最大で 65535 となる可能性があります。デバイス電流(例: Ib(Q1)) やノード電圧を記述できます。.wav アナログ/デジタル・コンバータのフルスケール・レンジは-1 ~+1(V または A)です。

PC のコーデックの制限が原因で、チャンネルの数、サンプル・レート、またはビット数のためにユーザの PC サウンド・システムで再生できない.wav ファイルを記述できることに注意してください。ただし、これらの.wav ファイルは、別のシミュレーションの入力として LTspice で引き続き使用できます。LTspice シミュレーションでの.wav ファイルの再生について詳しくは、「LTspice」=>「回路素子」=>「V. 電圧源」および「I. 電流源」のセクションを参照してください。.wav ファイルを PC のサウンド・カードで再生する場合、より一般的にサポートされている.wav ファイル形式は、チャンネル数が 1 または 2、ビット/チャンネルが 8 または 16、サンプル・レートが 11025、22050、または 44100Hz です。

.TRAN 修飾子

- [UIC:](#) DC 解法をスキップして、ユーザ指定の初期条件を使用します。
- [steady:](#) 定常状態に達したら、シミュレーションを停止します。
- [nodiscard:](#) 定常状態に達する前に、過渡シミュレーションの部分を削除しません。
- [startup:](#) 個別の電圧源および電流源をオフにして最初の動作点を解きます。その後、トランジェント解析を開始し、シミュレーションの最初の 20us 以内に電圧源および電流源をオンにします。
- [step:](#) 回路のステップ応答を計算します。

UIC

初期条件を使用します。通常は、トランジェント解析を開始する前に DC 動作点解析を実行します。この指令では、この初期設定が抑止されます。一部の回路素子の初期条件は、個々のインスタンスに応じて指定できます。UIC は特に推奨される SPICE の機能ではありません。DC 動作点解析をスキップすると、非物理的な初期条件になります。たとえば、静電容量と並列に接続した電圧源を考えます。ノード電圧は、指定しなかった場合、0 になります。その後、最初の時間刻みで、コンデンサを充電するのに無限の電流が必要です。シミュレータは、電流を非特異にする十分に短い時間刻みを見つけることができないので、「time step too small convergence fail」(時間刻みが短すぎて収束できない)というメッセージが表示されます。

startup

これは SPICE のオリジナルの「[uic](#)」と同様です。これは、シミュレーションの最初の 20 μ s の間に個々の電源の電圧が上昇する必要があるという意味です。ただし、DC 動作点解析は、.ic 指令で指定した制約条件を使用して実行されます。

steady

定常状態に達したら、シミュレーションを停止します。これは効率計算レポートでは必須です。定常状態の検出は、SMPS マクロモデルに書き込まれます。書き込まれる理由は、通常、クロック・サイクル期間での平均値が 0 のエラーアンプの出力電流を探すためです。このアルゴリズムでは、エラーアンプの出力コンプライアンス範囲が考慮されます。電流が 0 とみなされるピーク電流のほんの何分の 1 かの値は sstol オプションで指定します。

定常状態の自動検出は、限界に近すぎる場合やあまり限界に近くない場合は失敗することがあります。定常状態は以下の方法で対話式に指定できます。シミュレーションが始まったらすぐに、メニューコマンド「Simulate」=>「Efficiency Calculation」=>「Mark Start」を実行します。このコマンドを初めて実行するときは、一体化の制限値を手動で指定する予定であることを LTspice に伝達します。回路が定常状態に達したように見えたら、そのコマンドを再度実行します。これにより、履歴が消去され、効率計算が再開されます。その後しばらくしたら(10 サイクルを超えるクロック・サイクルを確認したら)、「Simulate」=>「Efficiency Calculation」=>「Mark End」を実行します。「Simulate」=>「Efficiency Calculation」=>「Mark Start」を実行するたびに、効率計算を再開して波形履歴を消去します。この方法は、データ・ファイルが大きくなりすぎて、プロット速度が低下しないようにするという点で優れているので、効率計算の一体化に組み込まない本質的なデータを蓄積したことが明確な場合は常に「Simulate」=>「Efficiency Calculation」=>「Mark Start」を定期的に行うことを推奨します。

ノード電圧とインダクタ電流を指定して、定常状態を検出するのに必要なトランジェント解析の長さを短くする場合は、.ic 指令を使用してください。

nodiscard

定常状態に達する前に、トランジェント・シミュレーションの部分を削除しません。

step

回路のステップ応答を計算します。この機能は、電流源を一連のステップ電流を備えた負荷として使用することで機能します。手順は以下のとおりです。

1. nodiscard を設定していない場合は、定常状態を計算して、履歴を破棄します。
2. ステップ負荷を電流のリスト内の次の値に、 $20\text{A}/\mu\text{s}$ の割合で増やします。
3. 定常状態になるまで計算します。
4. ステップ負荷をリスト内の次の値に変更します。次の値がない場合は停止します。

回路が複雑なために、ステップの自動遷移を検出できないことがあります。この状況では、.TRAN コマンドを使用して、過渡シミュレーションを実行し、目的のステップ負荷応答の開始期間と終了期間を観察するのが最善です。出力負荷電流を設定して、目的の期間に異なるレベルに切り替えるには、PWL コマンドを使用します。たとえば、次のように記述します。

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

負荷電流は時刻 0 では 0.5A から始まり、1ms では 0.5A のまま推移し、1.01ms で 0.1A に切り替わり、3ms まで 0.1A のまま推移して、3.01ms で 0.5A に切り替わり、0.5A を維持します。

PWL では、ほぼ無制限のシーケンス対(時間、値)が可能です。

回路素子のクイック・リファレンス

部品	構文
特殊機能	Axx n1 n2 n3 n4 n5 n6 n7 n8 + <model> [extra parameters]
任意の動作信号源	Bxx n+ n- <V=... または I=...>
Capacitor(コンデンサ)	Cxx n+ n- <capacitance> + [ic=<val>][Rser=<val>] + [Lser=<val>][Rpar=<val>] + {Cpar=<val>}[m=<val>]
Diode(ダイオード)	Dxx A K <model> [area]
電圧依存型の電圧源	Exx n+ n- nc+ nc- <gain>
電流依存型の電流源	Fxx n+ n- <Vnam> <gain>
電圧依存型の電流源	Gxx n+ n- nc+ nc- <transcond.>
電流依存型の電圧源	Hxx n+ n- <Vnam> <transres.>
独立した電流源	Ixx n+ n- <current>
JFET トランジスタ	Jxx D G S <model> [area] [off] +[IC=<Vds,Vgs>][temp]
相互インダクタンス	Kxx L1 L2 L3 <coeff.>
インダクタンス	Lxx n+ n- <inductance> + [ic=<val.>] [Rser=<val.>] + [Rpar=<val.>] + [Cpar=<val.>] [m=<val.>]
MOSFET トランジスタ	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>]
損失の大きい伝送線路	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][temp=<T>]
Resistor(抵抗)	Rxx n1 n2 <value>
電圧制御型スイッチ	Sxx n1 n2 nc+ nc- <model> + [on,off]
損失の小さい伝送線路	Txx L+ L- R+ R- ZO=<value> + TD=<value>

均一な RC 線路	Uxx n1 n2 ncommon <model> + L=<len> [N=<lumps>]
独立した電圧源	Vxx n+ n- <voltage>
電流制御型スイッチ	Wxx n1 n2 <Vnam> <model> + [on,off]
サブ回路	Xxx n1 n2 n3...<subckt name>
MESFET トランジスタ	Zxx D G S model [area] [off] + [IC=<Vds,Vgs>]

A. 特殊機能

シンボル名: INV、BUF、AND、OR、XOR、SCHMITT、SCHMTBUF、SCHMTINV、
DFLOP、VARISTOR、および MODULATE

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

これらはリニアテクノロジー社独自の特殊機能/混成モード・シミュレーション・デバイスです。これらのデバイスは LTspice で利用できる一連の各新モデルとともにしばしば変化するので、これらのデバイスとその動作のほとんどは文書に記載されていません。しかしながら、ここでは、これらの一般的な興味のためにその一部を文書化します。

INV、BUF、AND、OR、および XOR は、一般的な理想化動作ゲートです。すべてのゲートは 8 端子でネットリスト化されます。これらのゲートには外部電源は必要ありません。電流は相補出力(端子 6 および 7)から流出または吸い込まれ、デバイス共通ノードである端子 8 を通じて戻ります。端子 1~5 は入力です。未使用の入力および出力は端子 8 に接続されます。デジタル・デバイス・コンパイラは、対象の端子が使用されていないことをフラグとして認識し、その端子をシミュレーションから削除します。こうすると、入力を接地した場合と 0V にした場合で AND ゲートの動作が異なるという混乱状況になる可能性があります。グラウンドがゲートの共通ノードである場合、接地入力はロジックの false 条件ではなく、単純にシミュレーションの一部ではないということです。これらのゲートがこのように実装される理由は、こうすると 1 つのデバイスが、true 出力、反転出力、または相補出力を備えた 2 つ、3 つ、4 つまたは 5 つの入力ゲートとして動作できるだけでなく、未使用の端子によってシミュレーション速度が犠牲にならないからです。つまり、AND デバイスは 12 種類の異なる AND ゲートとして動作します。ゲートはデフォルトで 0V/1V のロジックになり、そのロジックしきい値は 0.5V で、伝播遅延がなく、出力インピーダンスは 1Ω です。出力特性は以下のインスタンス・パラメータを使用して設定します。

名前	デフォルト	概要
Vhigh	1	ロジック“H”のレベル
Vlow	0	ロジック“L”のレベル
Trise	0	立ち上がり時間
Tfall	Trise	立ち下がり時間
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	ロジック“H”レベルのインピーダンス
Rlow	Rout	ロジック“L”レベルのインピーダンス

同じインスタンス上ですべてのパラメータを同時に指定できるとは限らないことに注意してください。たとえば、出力特性は、スルーイング立ち上がり時間または RC 時定数のいずれかであり、その両方ではありません。

伝播遅延はデフォルトでは 0 であり、インスタンス・パラメータ Td を使用して設定します。入力保持時間は伝播遅延と同じ値です。

入力ロジックしきい値はデフォルトでは $0.5 \cdot (V_{high} + V_{low})$ になりますが、インスタンス・パラメータ Ref を使用して設定できます。保持時間は伝播遅延と同じです。

使用する入力が 2 つより多いと、排他的論理和 (XOR) デバイスの動作は標準的ではなくなります。

出力が trueなのは、すべての入力のうち厳密に 1 つだけが true の場合です。複数の XOR デバイスとの XOR の関連特性を使用して、入力数が 2 より多い XOR ブロックを実装します。

シュミット・トリガ回路デバイスの出力特性は、ゲートと同様です。これらのデバイスの作動点は、インスタンス・パラメータ Vt および Vh で指定します。低い方の作動点は $V_t - V_h$ であり、高い方の作動点は $V_t + V_h$ です。ゲートとシュミット・トリガ回路デバイスは、デフォルトではシミュレーション・エンジンに時間刻みの情報を提供しません。つまり、状態をいつ変化させようとしているかを確認せず、状態変化前後のどちら側に近い時間刻みがあるかを確認できません。インスタンス・パラメータ tripdt を設定して、シミュレータが状態変更の前後でとる最大の時間刻みサイズを規定できます。

VARISTOR は電圧制御型のバリスタです。そのブレークダウン電圧は、端子 1 と 2 の間の電圧で設定します。そのブレークダウン・インピーダンスは、インスタンス・パラメータ rclamp で指定します。回路図例 `¥examples¥Educational¥varistor.asc` を参照してください。

MODULATE デバイスは、電圧制御発振器です。回路図例 `¥examples¥Educational¥PLL.asc` を参照してください。瞬間的な発振周波数は、FM 入力の電圧で設定します。

電圧から周波数への変換は線形であり、2 つのインスタンス・パラメータ、mark と space で設定します。mark は FM 入力 が 1V のときの周波数で、space は FM 入力 が 0V のときの周波数です。AM 入力を未使用の場合 (MODULATE の共通ノードに接続した場合)、振幅は AM 入力の電圧によって 1V に設定され、デフォルトでは 1V になります。

LTspice の回路図取り込みの観点から、これらのデバイスのシンボルは特別な方法でネットリスト化されます。接続されていないすべての端子は、自動的に端子 8 に接続されます。また、端子 8 が未接続な場合は、ノード 0 に接続されます。

B. 任意の動作電圧源または電流源

シンボル名: BV、BI

構文: 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>]]

最初の構文では動作電圧源を指定しており、次の構文では動作電流源を指定しています。電流源の場合は、Rpar インスタンス・パラメータを使用して並列抵抗を指定できます。

tripdv および tripdt は、ステップ除去率を制御します。電圧源両端の電圧が tripdt 秒間に tripdv ボルトを超えて変化した場合、そのシミュレーション時間刻みは除去されます。

式には以下を含めることができます。

- ノード電圧。たとえば、V(n001)
- ノード電圧の差。たとえば、V(n001, n002)
- 回路素子電流。たとえば、スイッチ S1 を流れる電流 I(S1)または Q1 のベース電流 Ib(Q1)。ただし、回路素子電流は準静的に変化しているとみなされます。つまり、参照デバイスを流れる電流と動作電流源出力間に瞬間的な帰還はありません。同様に、こうしたデバイス電流の AC 成分は、小信号の線形 AC 解析では 0 であるとみなされます。
- キーワード「time」シミュレーションにおける現在の時刻を意味します。
- キーワード「pi」3.14159265358979323846 を意味します。
- 以下の関数:

関数名	概要
abs(x)	x の絶対値
absdelay(x,t[,tmax])	遅延時間が t のときの x。オプションの最大遅延通知 tmax
acos(x)	x の逆余弦の実数部。たとえば、acos(-5)の結果は 3.14159 であり、3.14159+2.29243i ではありません。
arccos(x)	acos()と同義
acosh(x)	x の双曲線余弦の実数部。たとえば、acosh(.5)の結果は 0 であり、1.0472i ではありません。
asin(x)	x の逆正弦の実数部。たとえば、asin(-5)の結果は -1.57080 であり、-1.57080+2.29243i ではありません。
arcsin(x)	asin()と同義
asinh(x)	逆双曲線正弦
atan(x)	x の逆正接
arctan(x)	atan()と同義
atan2(y, x)	y/x の 4 象限逆正接
atanh(x)	逆双曲線正接
buf(x)	x>.5 の場合は 1、そうでない場合は 0
ceil(x)	x 以上の整数
cos(x)	x の余弦
cosh(x)	x の双曲線余弦
ddt(x)	x の時間導関数
delay(x,t[,tmax])	absdelay()と同じ
exp(x)	e の x 乗
floor(x)	x 以下の整数
hypot(x,y)	$\sqrt{x^2 + y^2}$
idt(x[,ic[,a]])	x を積分、オプションの初期条件 ic、a が true の場合はリセット
idtmod(x[,ic[,m[,o]]])	x を積分、オプションの初期条件 ic、母数が m に達したらリセット、出力を o だけオフセット補正
if(x,y,z)	x>.5 の場合は y、そうでない場合は z
int(x)	x を整数に変換
inv(x)	x>.5 の場合は 0、そうでない場合は 1
limit(x,y,z)	x、y、z の中間値
ln(x)	x の自然対数
log(x)	ln()の代替構文
log10(x)	10 を底とする対数

max(x,y)	x と y のうち大きい方の値
min(x,y)	x と y のうち小さい方の値
pow(x,y)	x^{**y} の実数部。たとえば、 $\text{pow}(-1,.5)=0$ であり、 i ではありません。
pwr(x,y)	$\text{abs}(x)^{**y}$
pwr(x,y)	$\text{sgn}(x)*\text{abs}(x)^{**y}$
rand(x)	x の整数値に依存する 0 と 1 の間の乱数
random(x)	rand()と同様だが、値の間を滑らかに遷移
round(x)	x に最も近い整数
sdt(x[,ic[,assert]])	idt()の代替構文
sgn(x)	x の符号
sin(x)	x の正弦
sinh(x)	x の双曲線正弦
sqrt(x)	x の平方根
table(x,a,b,c,d,...)	一連の点の組として与えられた参照表に基づいて x の値を内挿する
tan(x)	x の正接
tanh(x)	x の双曲線正接
u(x)	単位ステップ、つまり $x>0$ の場合は 1、そうでない場合は 0
uramp(x)	$x>0$ の場合は x、そうでない場合は 0
white(x)	-0.5 と 0.5 の間の乱数が値の間を random()より滑らかに遷移する
!(x)	inv(x)の代替構文
~(x)	inv(x)の代替構文

以下の演算は評価の優先順位の逆の順に要約しています。

オペランド	概要
&	式の両辺をブール値に変換後、論理積をとる
	式の両辺をブール値に変換後、論理和をとる
^	式の両辺をブール値に変換後、排他的論理和をとる
>	式の左辺が式の右辺より大きい場合は true、そうでない場合は false
<	式の左辺が式の右辺より小さい場合は true、そうでない場合は false

>=	式の左辺が式の右辺以下である場合は true、そうでない場合は false
<=	式の左辺が式の右辺以上である場合は true、そうでない場合は false
+	浮動小数点の加算
-	浮動小数点の減算
*	浮動小数点の乗算
/	浮動小数点の除算
**	左辺を右辺で累乗する。実数部だけが返される。たとえば、 $-1^{**}1.5$ の結果は 0 であり、 i ではない
!	次の式をブール値に変換して反転する

true は数値的には 1 に等しく、false は 0 に等しくなります。ブール値への変換により、値が 0.5 より大きい場合、値は 1 に変換され、そうでない場合、値は 0 に変換されます。

LTspice では、ブール XOR 演算には脱字記号 (^) を使用し、指数演算には「**」を使用することに注意してください。また、LTspice は指数演算 ($x^{**}y$) と関数 $\text{pwr}(x,y)$ を区別します。一部のサードパーティ・シミュレータは指数演算動作の実装に誤りがあり、 $-3^{**}3$ の評価結果が -27 ではなく、誤って 27 になります。これはおそらく、負の数を累乗した結果が整数の累乗以外の値になる問題を回避するためです。LTspice では、指数の結果の実数部を返すことにより、この問題に対処します。たとえば、 $-2^{**}1.5$ は 0 と評価されますが、これは正しい答えである $2.82842712474619i$ の実数部です。この意味は、サードパーティ・シミュレータを対象とするサードパーティ・モデルをインポートした場合は、 x^y のような構文を $x^{**}y$ や、さらに $\text{pwr}(x,y)$ に変換することが必要な可能性があるということです。

オプションのラプラス変換を定義した場合、その変換は動作電流または動作電圧の結果に適用されます。

ラプラス変換は s だけの関数である必要があります。ブール XOR 演算子 (^) は、ラプラスの式の中で使用した場合は指数 (**) を意味すると解釈されます。周波数 f での周波数応答は、 s の代わりに $\text{sqrt}(-1)*2*\pi*f$ を使用することによって求められます。時間領域動作は、瞬間的な電流(または電圧)と、インパルス応答によるこの電流(または電圧)の履歴の畳み込み積分を加算することで分かります。時間領域インパルス応答に対するラプラス伝達関数の数値の極性反転は、潜在的に計算限界の処理であり、現在の数値研究の議題の 1 つです。LTspice では、周波数領域応答で離散的な設定点の FFT を行うことによってインパルス応答が得られます。この処理では、離散 FFT では一般的なスペクトルの漏れや ピケット・フェンシングのような FFT の通常のアーティファクトが発生する傾向があります。LTspice は独自のアルゴリズムを使用しており、周波数領域応答の正確な解析式を備えていることを利用し、こうしたアーティファクトを 0 まで正確に回折させる設定点と期間を選択します。ただし、LTspice では、適切な周波数範囲および分解能を推定する必要があります。まず、LTspice がこれを推定できるようにすることを推奨します。使用された期間の長さおよび FFT 波形データ点の数は、ログ・ファイル内で報告されます。その後、 nfft および期間の長さを明示的に設定することにより、アルゴリズムの選択肢を調整できます。期間の値の逆数は、周波数の分解能です。 nfft の値をこの分解能で乗算した値が、考

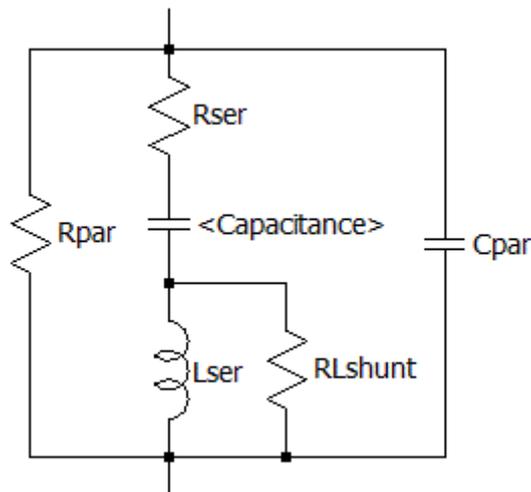
慮される最大周波数です。動作信号源によるインパルス応答の畳み込み積分も、潜在的に計算限界の処理であることに注意してください。

C. コンデンサ

シンボル名: CAP、POLCAP

構文: Cnnn n1 n2 <capacitance> [ic=<value>]
 + [Rser=<value>] [Lser=<value>] [Rpar=<value>]
 + [Cpar=<value>] [m=<value>]
 + [RLshunt=<value>] [temp=<value>]

等価の直列抵抗、直列インダクタンス、並列抵抗、および並列分流容量を指定することが可能です。等価回路を以下に示します。



コンデンサのインスタンス・パラメータ

名前	概要
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 両端の分流抵抗
m	並列単位の数
temp	インスタンスの温度(対応する.model 文での tempcos に対応)
ic	初期電圧(.tran カード上で uic にフラグが立てられる場合にのみ使用)

コンデンサ内部に存在する寄生成分 R_{par} 、 R_{ser} 、 R_{Lshunt} 、 C_{par} 、および L_{ser} は、下図に明示するより組み込む方が計算上適切です。LTspice では、独自の回路シミュレーション技術により、このモデルの物理的コンデンサを内部ノードなしでシミュレートします。これにより、シミュレーション・マトリクスは小さくなり、解明速度が向上するので、短い時間刻みで特異解になることが少なくなります。

コンデンサ素子にはこれらの寄生成分が含まれているので、それが圧電結晶の基本をマクロモデル化するのに役立つことに注意してください。

一般的な非線形コンデンサも使用できます。静電容量を指定する代わりに、充電の式を記述します。LTspice はこの式を整理し、すべての変数について記号的に微分して、静電容量に対応する偏導関数を見つけます。

構文: $C_{nnn} \ n1 \ n2 \ Q=<expression> \ [ic=<value>] \ [m=<value>]$

デバイス両端の電圧を意味する特殊変数(x)があります。したがって、100pF の一定の静電容量を次のように記述できます。

$C_{nnn} \ n1 \ n2 \ Q=100p*x$

0V で 100p から 300p まで急激に変化する静電容量は次のように記述できます。

$C_{nnn} \ n1 \ n2 \ Q=x*if(x<0,100p,300p)$

このデバイスは、仮説に基づく新しい充電モデル(例:トランジスタ)の動作を素早く評価するのに役立ちます。

D. ダイオード

シンボル名: DIODE、ZENER、SCHOTTKY、VARACTOR.

構文: Dnnn anode cathode <model> [area] [off] [m=<val>] [n=<val>] [temp=<value>]

例: D1 SW OUT MyIdealDiode

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

```
D2 SW OUT dio2
```

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

インスタンス・パラメータ M は並列デバイスの数を設定しますが、インスタンス・パラメータ N は直列デバイスの数を設定します。

ダイオードでは、その特性を指定するのに.model カードが必要です。2 種類のダイオードを使用できます。1 つは導通領域的に線形のモデルで、理想化ダイオードの計算的軽量表現が得られます。このモデルには、導通の線形領域が 3 つ(オン、オフ、逆方向ブレークダウン)あります。順方向導通と逆方向ブレークダウンは、Ilimit(revlimit)で電流制限を指定すれば非線形にすることができます。tanh()を使用して、順方向導通の傾斜を電流制限値に合わせます。パラメータ epsilon および repsilon を指定することにより、オフ状態と導通状態を滑らかに切り替えることができます。ダイオードの IV 曲線が値と傾斜の点で途切れないようにするために、オフ状態とオン状態の間は 2 次関数が適しています。また、オフから順方向導通に遷移する電圧は epsilon の値で指定し、オフから逆方向ブレークダウンに遷移する電圧は repsilon の値で指定します。

この種のダイオードのモデルパラメータを以下に示します。

名前	概要	単位	デフォルト値
Ron	順方向導通時の抵抗	Ω	1
Roff	オフ時の抵抗	Ω	1./Gmin
Vfwd	導通状態に入る順方向しきい値電圧	V	0
Vrev	逆方向ブレークダウン電圧	V	Infin.
Rrev	ブレークダウン・インピーダンス	Ω	Ron
Ilimit	順方向電流制限	A	Infin.
Revlimit	逆電流制限	A	Infin.
Epsilon	2 次領域の幅	V	0
Repsilon	逆方向の 2 次領域の幅	V	0

モデル内で Ron、Roff、Vfwd、Vrev、Rrev のいずれかを指定すると、この理想化モデルが使用されます。

使用可能なもう一方のモデルは標準的な Berkeley SPICE 半導体ダイオードですが、より詳細なブレイクダウン動作および再結合電流を処理するために拡張されています。指定したモデルの等価並列デバイスの数は、領域係数によって決まります。このダイオードのダイオード・モデル・パラメータを以下に示します。

名前	概要	単位	デフォルト値	例
Is	飽和電流	v	1.00E-14	1.00E-07
Rs	オーミック抵抗	Ω	0.	10.
N	発光係数	-	1.	1.
Tt	遷移時間	sec	0.	2n
Cjo	ゼロバイアス接合容量	F	0.	2p
Vj	接合部電位	V	1.	.6
M	濃度勾配係数	-	0.5	0.5
Eg	活性化エネルギー	eV	1.11	1.11 Si 0.69 Sbd 0.67 Ge
Xti	飽和電流の温度指数	-	3.0	3.0 jn 2.0 Sbd
Kf	フリッカ・ノイズ係数	-	0.	
Af	フリッカ・ノイズ指数	1.	1.	
Fc	順バイアス時の空乏層容量式の係数	-	1.	
BV	逆方向ブレイクダウン電圧	V	Infin.	40.
nbv	逆方向ブレイクダウン放射係数	-	1.	2.
Ibv	ブレイクダウン電圧時の電流	A	1.0E-10	
Ibvl	低レベルの逆方向ブレイクダウン屈曲点電流	A	0.	
nbvl	低レベルの逆方向ブレイクダウン放射係数	-	1.	
Tnom	パラメータ測定温度	$^{\circ}\text{C}$	27	50
Isr	再結合電流パラメータ	A	0.	
Nr	Isr 放射係数	-	2	
Ikf	高注入屈曲点電流	A	Infin.	
Tikf	Ikf の 1 次温度係数	$/^{\circ}\text{C}$	0.	
Trs1	Rs の 1 次温度係数	$/^{\circ}\text{C}$	0.	
Trs2	Rs の 2 次温度係数	$/^{\circ}\text{C}^2$	0.	
Tbv1	ブレイクダウン電圧の温度係数	$/^{\circ}\text{C}$	0.	

Tbv2	ブレークダウン電圧の 2 次温度係数	$/^{\circ}\text{C}^2$	0.	
Tikf	Ikf の温度係数	$/^{\circ}\text{C}$	0.	
Perim	デフォルトの周囲長	m	0.	
Is	サイドウォールの Is	A	0.	
ns	サイドウォールの放射係数	-	1.	
Rsw	サイドウォールの直列抵抗	Ω	0.	
Cjsw	サイドウォールの Cjo	F	0.	
Vjsw	サイドウォールの Vj	V	Vj	
mjsw	サイドウォールの mj	-	0.33	
Fcs	サイドウォールの Fc	-	Fc	

モデルに電圧定格と電流定格を注釈付けすることができます。この情報はデバイスの選択を支援するために回路図取り込み GUI に表示されますが、シミュレーションでの電氣的動作には直接影響しません。以下のパラメータを指定できます。

名前	概要	単位
Vpk	ピーク電圧定格	V
Ipk	ピーク電流定格	A
Iave	平均電流定格	A
Irms	RMS 電流定格	A
diss	最大電力損失定格	W

E. 電圧依存型の電圧源

シンボル名: E, E2

3種類の電圧依存型電圧源回路素子があります。

構文: `Exxx n+ n- nc+ nc- <gain>`

この回路素子は、ノード `nc+` と `nc-` の間の入力電圧に依存するノード `n+` と `n-` の間の出力電圧をアサートします。これは固定利得単独で指定する線形の電圧依存型電圧源です。

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

参照表を使用して伝達関数を指定します。参照表は、一連の数値対です。制御電圧が対の最初の値に等しい場合、対の2番目の値が出力電圧です。制御電圧が指定点の間にある場合、出力は直線補間されます。制御電圧が参照表の範囲を超えている場合、出力電圧は参照表の最後の点の定電圧として外挿されます。

構文: `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、そのラプラス変換によって指定されます。ラプラス変換は s だけの関数である必要があります。周波数 f での周波数応答は、 s の代わりに $\sqrt{-1} * 2 * \pi * f$ を使用することによって求められます。時間領域動作はインパルス応答により得られますが、この応答は周波数領域の応答をフーリエ変換することで得られます。LTspice は適切な周波数範囲および分解能を推定する必要があります。応答は高周波で低下する必要があり、そうでない場合はエラーが報告されます。まず、LTspice がこれを推定できるようにして、その後、`reitol` を小さくするか、または `nfft` と期間を明示的に設定することにより、精度をチェックすることを推奨します。期間の値の逆数は、周波数の分解能です。`nfft` の値をこの分解能で乗算した値が、考慮される最大周波数です。ブール XOR 演算子「`^`」は、ラプラスの式の中で使用した場合は指数「`**`」を意味すると解釈されます。

構文: `Exxx n+ n- value={<expression>}`

これは、動作信号源(任意の動作電圧源、B)の代替構文です。

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

これは多項式による任意の挙動モデリングの古い方法です。
 従来のオペアンプ・モデルを実行するのに役立ちます。

注記: 抵抗で分流した G の電流源を使用して E の電圧源に近づける方が、 E の電圧源を使用するより適しています。抵抗で分流した電圧制御型電流源は、電圧制御型電圧源より計算が高速で、収束の問題が少なく済みます。また、結果として出力インピーダンスがゼロ以外の値になるので、より実際的な回路を表します。

F. 電流依存型の電流源

シンボル名: F

構文: Fxxx n+ n- <Vnam> <gain>

この回路素子は、ノード n+ と n- の間に電流を流します。流れる電流は、<Vnam> で指定した電圧源を流れる電流に利得を掛けた値と等しくなります。

構文: Fxxx n+ n- value={<expression>}

これは、動作信号源(任意の動作電圧源、B)の代替構文です。

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

これは多項式による任意の挙動モデリングの古い方法です。従来の挙動モデルを実行するのに役立ちます。

G. 電圧依存型の電流源

シンボル名: G、G2

3種類の電圧依存型電流源回路素子があります。

構文: Gxxx n+ n- nc+ nc- <gain>

この回路素子は、ノード nc+と nc-の入力電圧に依存するノード n+と n-の間の出力電流をアサートします。これは固定利得単独で指定する線形の電圧依存型電圧源です。

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

ここでは、参照表を使用して伝達関数を指定します。参照表は、一連の数値対です。制御電圧が対の最初の値に等しい場合、対の2番目の値が出力電流です。制御電圧が指定点の間にある場合、出力は直線補間されます。制御電圧が参照表の範囲を超えている場合、出力電流は参照表の最後の点の定電流として外挿されます。

構文: Gxxx 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、そのラプラス変換によって指定されます。ラプラス変換は s だけの関数である必要があります。周波数 f での周波数応答は、s の代わりに $\sqrt{-1} \cdot 2 \cdot \pi \cdot f$ を使用することによって求められます。時間領域動作はインパルス応答により得られますが、この応答は周波数領域の応答をフーリエ変換することで得られます。LTspice は適切な周波数範囲および分解能を推定する必要があります。応答は高周波で低下する必要があり、そうでない場合はエラーが報告されます。

まず、LTspice がこれを推定できるようにして、その後、reltol を小さくするか、または nfft と期間を明示的に設定することにより、精度をチェックすることを推奨します。期間の値の逆数は、周波数の分解能です。nfft の値をこの分解能で乗算した値が、考慮される最大周波数です。ブール XOR 演算子「^」は、ラプラスの式の中で使用した場合は指数「**」を意味すると解釈されます。

構文: Gxxx n+ n- value={<expression>}

これは、動作信号源(任意の動作電圧源、B)の代替構文です。

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

これは多項式による任意の挙動モデリングの古い方法です。従来の挙動モデルを実行するのに役立ちます。

H. 電流依存型の電圧源

シンボル名: H

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

この回路素子は、ノード n+ と n- の間に電圧を印加します。印加される電圧は、電圧源 <Vnam> を流れる電流にトランスレジスタンス (transresistance) を掛けた値と等しくなります。

構文: Hxxx n+ n- value={<expression>}

これは、動作信号源 (任意の動作電圧源、B) の代替構文です。

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

これは多項式による任意の挙動モデリングの古い方法です。既存のリニアテクノロジー挙動モデルを実行するのに役立ちます。

I. 電流源

シンボル名: CURRENT

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

この回路素子は、ノード n+ と n- の間に定電流を供給します。電流源を負荷と設定した場合、電流源は強制的に放散されます。つまり、ノード n+ と n- の間の電圧が 0 または負の値になると、電流も 0 になります。このオプションの目的は、出力電圧が 0 である場合に電流を流さない電源の電流負荷をモデル化することです。

AC 解析では、AC の値を解析周波数での信号源の振幅として使用します。

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

時間依存性のあるパルス電流源

名前	概要	単位
Ioff	初期値	A
Ion	パルス値	A
Tdelay	遅延	sec
Tr	立ち上がり時間	sec
Tf	立ち下がり時間	sec
Ton	オン時間	sec
Tperiod	周期	sec
Ncycles	サイクル数(自走パルス関数の場合は省略)	cycles

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

時間依存性のある正弦波電流源

名前	概要	単位
Ioffset	DC オフセット	A
Iamp	振幅	A
Freq	周波数	Hz
Td	遅延	秒
Theta	減衰係数	1/秒
Phi	正弦波の位相	度

Ncycles	サイクル数(自走パルス関数の場合は省略)	cycles
---------	----------------------	--------

Td より短い時間か、Ncycles の完了後の時間に動作させた場合、出力電流は $I_{offset} + I_{amp} \cdot \sin(\pi \cdot \phi / 180)$ で与えられます。それ以外の場合、出力電流は次式で与えられます。

$$I_{offset} + I_{amp} \cdot \exp(-(time - T_d) \cdot \Theta) \cdot \sin(2 \cdot \text{Freq} \cdot (time - T_d) + \phi / 180)$$

減衰係数(Theta)は、減衰時定数の逆数です。

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

時間依存性のある指数電流源

名前	概要	単位
I1	初期値	A
I2	パルス値	A
Td1	立ち上がり遅延時間	秒
Tau1	立ち上がり時定数	sec
Td2	立ち下がり遅延時間	sec
Tau2	立ち下がり時定数	sec

時間が Td1 より短い場合、出力電流は I1 です。時間が Td1 と Td2 の間の場合、出力電流は次式で与えられます。

$$I1 + (I2 - I1) \cdot (1 - \exp(-(time - T_{d1}) / \tau_{1}))$$

時間が Td2 より長い場合、出力電流は次式で与えられます。

$$I1 + (I2 - I1) \cdot (1 - \exp(-(time - T_{d1}) / \tau_{1})) - (I2 - I1) \cdot (1 - \exp(-(time - T_{d2}) / \tau_{2}))$$

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

時間依存性のある単一周波数 FM 電流源

名前	概要	単位
Ioff	DC オフセット	A
Iamp	振幅	A
Fcar	搬送周波数	Hz
MDI	変調指数	-

Fsig	信号周波数	Hz
------	-------	----

電流は次式で与えられます。

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

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

また、電流は、参照表を使用して出力ノード両端の電圧の関数として指定することもできます。これは負荷の特性をモデル化するのに役立ちます。

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

これは電流源用の特殊な書式です。電流は、ステップ負荷応答のトランジェント解析で使用する電流のリストとして指定します。このモードでは、リスト内の最初の電流<value1>で定常状態に達するまで、シミュレーションが計算されます。その後、電流はリスト内の次の値<value2>に進みます。その電流で定常状態に達するまで、シミュレーションは進みます。その後、電流は次の値に進み、リストを使い果たすまでこの処理が繰り返されます。.tran コマンドで「step」を指定していない場合は、元の<value>が使用されます。

構文: `Ixxx n+ n- R=<value>`

これは電流源ではなく、抵抗です。これが使用されるのは、負荷が電流源としてネットリスト化されている場合に、抵抗性負荷をモデル化するためです。

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

任意の区分線形電流源

t1 より前の時刻では、電流は i1 です。t1 と t2 の間の時刻では、電流は i1 と i2 の間で直線的に変化します。任意の数の時刻点、電流点を指定できます。最終時刻後の時刻では、電流は最後の電流になります。

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

これを使用すると、.wav ファイルを LTspice への入力として使用できます。<filename>は.wav ファイルの絶対パス(フルパス)か、シミュレーション回路図またはネットリストが入っているディレクトリから計算した相対パスです。二重引用符を使用してスペースを含むパスを指定できます。.wav ファイルには最大 65536 チャンネルが含まれている場合があり、0~65535 の番号が付いています。Chan を設定すると、どのチャンネルを使用するかを

指定できます。デフォルトでは、最初のチャンネル(番号 0)が使用されます。.wav ファイルは-1A から 1A までのフルスケール・レンジがあると解釈されます。

この信号源は.tran 解析でのみ意味があります。

J. JFET トランジスタ

シンボル名: NJF、PJF

構文: 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 トランジスタでは、その特性を指定するのに、[.model](#) カードが必要です。モデル・カード・キーワード NJF および PJF がトランジスタの極性を指定することに注意してください。指定したモデルの等価並列デバイスの数は、領域係数によって決まります。

JFET モデルは Shichman と Hodges の FET モデルに由来しており、ゲート接合部の再結合電流および衝撃イオン化を組み込むために拡張されています。DC 特性はパラメータ VTO、BETA で規定されます。VTO および BETA はゲート電圧によるドレイン電流の変化を決定します。LAMBDA は出力導電率を決定します。Is は 2 つのゲート接合部の飽和電流です。2 つのオーミック抵抗 (Rd および Rs) が組み込まれています。電荷の蓄積は、2 つのゲート接合部の非線形空乏層容量によってモデル化されます。この容量は接合電圧の -1/2 乗に比例し、パラメータ Cgs、Cgd、および PB によって定義されます。適合パラメータ B が追加されています。A. E. Parker, D. J. Skellern 共著の『An Improved FET Model for Computer Simulators』(IEEE Trans CAD, vol. 9, no. 5, pp. 551-553, May 1990) を参照してください。

名前	概要	単位	デフォルト値	例
Vto	しきい値	V	-2.0	-2.0
Beta	トランスコンダクタンス・パラメータ	A/V ²	1.0E-04	1.0E-03
Lambda	チャネル長調整パラメータ	1/V	0	1.0E-04
Rd	ドレインのオーミック抵抗	Ω	0.	100
Rs	ソースのオーミック抵抗	Ω	0.	100
Cgs	ゼロバイアスでの G-S 接合容量	F	0.	5p
Cgd	ゼロバイアスでの G-D 接合容量	F	0.	1p
Pb	ゲートの接合部電位	V	1.	0.6
m	ゲート接合部の濃度勾配係数	-	.5	0.8
Is	ゲート接合部の飽和電流	A	1.00E-14	1.00E-14
B	ドーピングのテール・パラメータ	-	1	1.1

KF	フリッカ・ノイズ係数	-	0	
Nlev	ノイズ式セレクト	-	0	3
Gdsnoi	nlev=3 の場合のショット・ノイズ係数	-	1.	2.
AF	フリッカ・ノイズ指数	-	1	
Fc	順バイアスでの空乏層容量の係数	-	.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・パラメータの指数温度係数	%/°C	0	
VtoTc	しきい値電圧の温度係数	V/°C	0	
N	ゲート接合部の放射係数	-	1.	
Isr	ゲート接合部の再結合電流パラメータ	A	0.	
Nr	Isr の放射係数	-	2	
alpha	イオン化係数	1/V	0	
Vk	イオン化屈曲点電圧	V	0	
Xti	飽和電流の温度係数	-	3	
mfg	メーカーの注釈	-	3	ACME Semi Ltd.

K. 相互インダクタンス

シンボル名: なし。回路図にテキストとして配置されます。

構文: Kxxx L1 L2 [L3 ...]<coefficient>

L1とL2は回路内のインダクタの名前です。相互結合係数は-1~1の範囲内にある必要があります。

次の行

K1 L1 L2 L3 L4 1.

は、以下の6行と同義です。

K1 L1 L2 1.

K2 L2 L3 1.

K3 L3 L4 1.

K4 L1 L3 1.

K5 L2 L4 1.

K6 L1 L4 1.

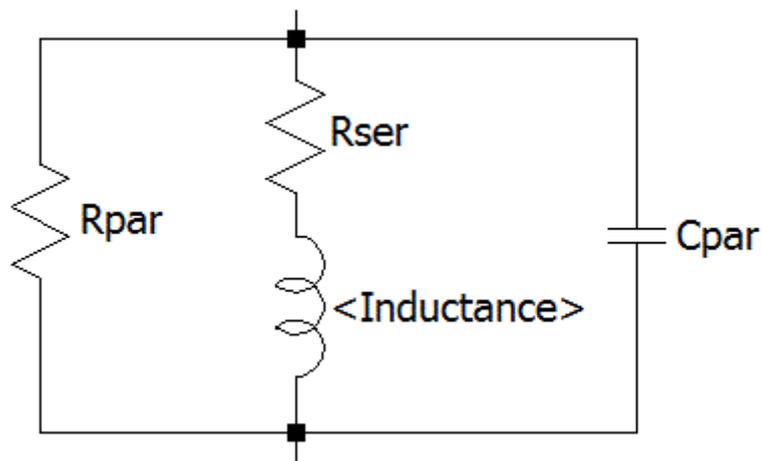
相互結合係数を1にして設計を開始することを推奨します。こうすると、減衰を行わない場合に極端な高周波でリングングが発生したり、シミュレーション速度が低下したりする原因となる漏れインダクタンスを除去できます。

L. インダクタ

シンボル名: IND、IND2

構文: Lxxx n+ n- <inductance> [ic=<value>]
 + [Rser=<value>] [Rpar=<value>]
 + [Cpar=<value>] [m=<value>] [temp=<value>]

等価の直列抵抗、直列インダクタンス、並列抵抗、および並列分流量を指定することが可能です。等価回路を以下に示します。



インダクタのインスタンス・パラメータ

名前	概要
Rser	等価直列抵抗
Rpar	等価並列抵抗
Cpar	等価並列容量
m	並列単位の数
ic	初期電流(.tran カード上で uic にフラグが立てられる場合にのみ使用)
tc1	インダクタンスの 1 次温度係数
Tc1	インダクタンスの 2 次温度係数
temp	インスタンスの温度

インダクタ内部に存在するデバイスの寄生成分 Rpar、Rser、および Cpar は、下図に明示するより組み込む方が適切です。LTspice では、独自の回路シミュレーション技術により、この物理的インダクタを内部ノードなしでシミュレートします。これにより、シミュレーション・マトリクスは小さくなり、計算速度が向上するので、時間刻み幅全体で特異解になることが少なくなります。

デフォルトでは、LTspice はインダクタに損失を与えて SMPS トランジェント解析を支援します。SMPS では、これらの損失は通常は重要ではありませんが、必要に応じてオフにすることができます。「Tools」=>「Control Panel」=>「Hacks!」ページで、「Supply a min. inductor damping if no Rpar is given.」のチェックボックスをオフにします。この設定はプログラムを複数回起動した場合も引き継がれます。相互インダクタンスの文では言及されていないインダクタには、 $1\text{m}\Omega$ のデフォルト直列抵抗もあります。この Rser を使用すると、LTspice IV では、回路の線形化マトリクスのサイズを低減するために、テブナン等価回路の代わりにノートン等価回路としてインダクタンスを積分することができます。LTspice がこの最小抵抗を導入しないようにする場合は、対象のインダクタに対して明示的に Rser=0 を設定する必要があります。こうすると、LTspice はより煩雑なインダクタのテブナン等価回路をトランジェント解析時に使用することが必要になります。

LTspice では、非線形インダクタの 2 つの書式を使用できます。1 つは磁束の式で指定された動作インダクタンスです。インダクタの電流は、式ではキーワード「x」で参照します。ネットリストでの例を以下に示します。

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

この例では、インダクタンスをノード N001 の電圧として読み取ることができるように、I1 によって di/dt は 1 になります。

LTspice で使用できる別の非線形インダクタは、『IEEE Transactions On Computer-Aided Design』(Vol. 10.No. 4, April 1991) で John Chan らによって最初に提案されたモデルに基づくヒステリシスのコア・モデルですが、米国特許 7,502,723 の方法によって拡張されました。このモデルは、以下の 3 つのパラメータだけでヒステリシス・ループを定義します。

名前	概要	単位
Hc	保磁力	アンペア回数/メートル (AT/m)
Br	残留磁束密度	Tesla
Bs	飽和磁束密度	Tesla

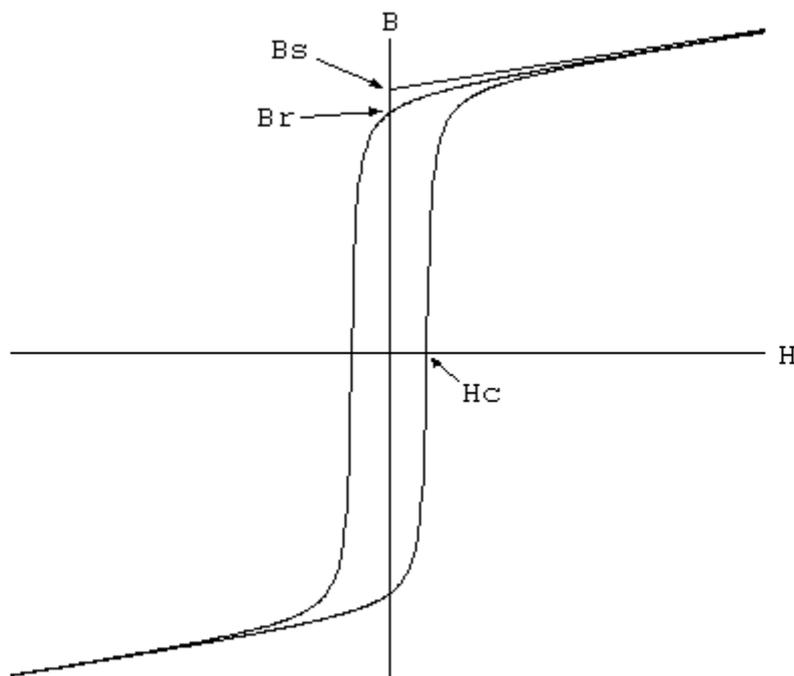
ヒステリシス主要ループの上側と下側の分岐は以下の式で与えられます。

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

および

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

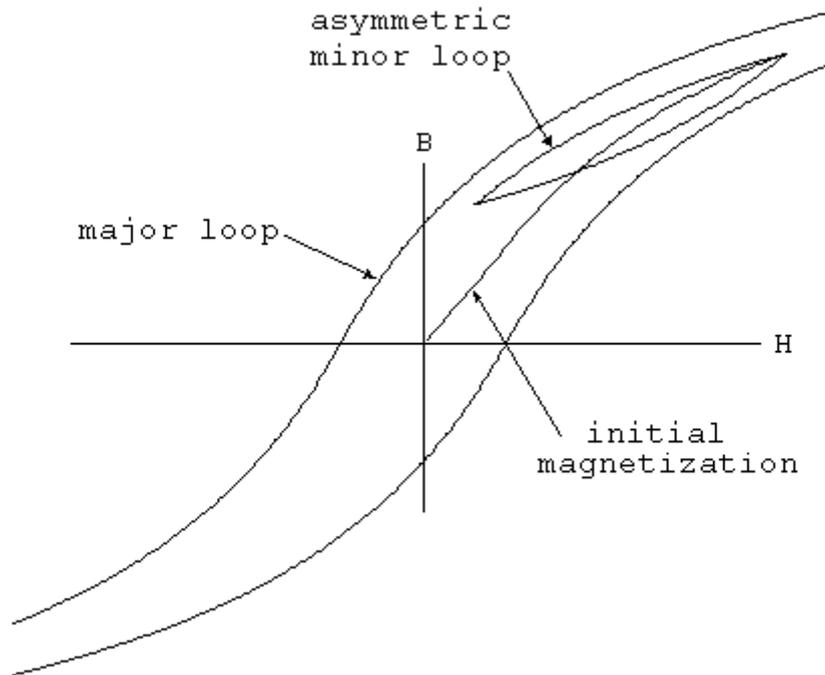
これらの関数は次の図にプロットされています。Hc および Br は、主要ヒステリシス曲線と H 軸および B 軸との交点です。Bs は、H が無限遠に向かうにつれて近づく漸近線 ($B_{sat}(H) = B_s + \mu_0 \cdot H$) の B 軸との交点です。



初磁化曲線は次式で与えられます。

$$B_{mag}(H) = .5 \cdot (B_{up}(H) + B_{dn}(H))$$

マイナー・ループは、引用した参考文献に従ってこの等式をさまざまな形に変換することによって得られます。コアの絶対透磁率および微分透磁率は、H と H の値の履歴の関数です。以下のプロットは、標準的なパワー・フェライト ($H_c = 16 \text{ AT/m}$, $B_s = .44 \text{ T}$, $B_r = .10 \text{ T}$) の場合に非対称のマイナー・ループによって得られる経路を示します。



コアの特性パラメータ H_c 、 B_r 、 B のほかに、コアの機械的寸法が必要です。

名前	概要	単位
L_m	磁気長(隙間を除く)	m(メートル)
L_g	隙間の長さ	m(メートル)
A	断面積	m^2 (平方メートル)
N	巻数	-

0 以外の隙間を指定する場合、磁界(H)は巻線内の電流に比例しないことに注意してください。LTspice は、断面積が均一であるか、隙間が薄いか均等に分散しているという前提で、コアおよび隙間での磁界の値を求めます。

以下は、L1(隙間のあるコアに巻きつけたインダクタ)のインダクタンス対電流を示す例です。I1によって dl/dt は 1 になるので、インダクタンスは $V(n001)$ として読み取ることができます。コアは初磁化曲線に従うので、まず、電流が増加するにつれて透磁率は初期値から増加し、その後電流が飽和すると減少することが分かります。インダクタンスは隙間によってコアの厳密な透磁率に対する感度が低くなるので、インダクタンスの増加を確認するには $V(n001)$ を実際に拡大する必要があります。ピークは、コア内部の H がその H_c と等しくなるときです。

*

L1 N001 0 $H_c=16$.

```
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
```

M. MOSFET

シンボル名: NMOS、NMOS3、PMOS、PMOS3

LTspice には、タイプの異なる基本的な MOSFET が 2 つあります。それは、モノリシック MOSFET と、新しい縦型二重拡散パワーMOSFET モデルです。

モノリシック MOSFET:

構文: 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>]

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

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

縦型二重拡散パワーMOSFET:

構文: 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 Cgdmin=50p Cgs=3.1n Cjo=1n
 + Is=5.5p Rb=5.7m)

MOSFET のモデル・カードは、どのタイプを対象とするかを指定します。モデル・カード・キーワード NMOS および PMOS は、N チャネルまたは P チャネルのモノリシック MOSFET トランジスタを指定します。モデル・カード・キーワード VDMOS は、縦型二重拡散パワーMOSFET を指定します。

モノリシック MOSFET は 4 端子のデバイスです。Nd、Ng、Ns、Nb は、ドレイン、ゲート、ソース、バルク(つまり、基板)の各ノードです。L および W はチャンネル長とチャンネル幅(m)です。

AD および AS は、ドレインとソースの拡散面積(平方メートル)です。接尾辞 u は μm を指定し、p は平方 μm を指定することに注意してください。L、W、AD、AS のいずれも指定しなかった場合は、デフォルト値が使用されます。PD および PS は、ドレイン拡散とソース拡散の周囲長(メートル)です。NRD と NRS は、ドレインとソースの拡散面積と等価な正方形の数を指定します。これらの値は、.MODEL 制御行で指定されるシート抵抗 RSH と乗算します。PD および PS のデフォルト値は 0 ですが、NRD および NRS のデフォルト値は 1 です。OFF は、DC 解析でのデバイスの初期条件を示します。IC=VDS、VGS、VBS を使用して初期条件を指定するのは、静止した動作点以外の動作点からトランジェント解析を始めるときに、.TRAN 制御行の UIC オプションを指定する場合があります。オプションの TEMP 値はこのデバイスが動作する温度であり、.OPTION 制御行の温度指定より優先されます。この温度指定はレベル 1、2、3、および 6 の MOSFET にのみ有効で、レベル 4、5、または 8 の BSIM デバイスには当てはまりません。

LTspice には、7 種類の異なるモノリシック MOSFET と、1 種類の縦型二重拡散パワーMOSFET が収録されています。

モノリシック MOSFET デバイス・モデルは 7 つあります。モデルパラメータ LEVEL は、使用するモデルを指定します。デフォルトのレベルは 1 です。

レベル モデル

レベル	モデル
1	Shichman-Hodges
2	MOS2 (A. Vladimirescu, S. Liu 共著、『The Simulation of MOS Integrated Circuits Using SPICE2, ERL Memo No. M80/7』、Electronics Research Laboratory University of California, Berkeley, October 1980 参照)
3	MOS3、半経験的モデル(レベル 2 の参考文献を参照)
4	BSIM (B. J. Sheu, D. L. Scharfetter, P. K. Ko 共著、『SPICE2 Implementation of BSIM』、ERL Memo No. ERL M85/42, Electronics Research Laboratory University of California, Berkeley, May 1985 参照)
5	BSIM2 (Min-Chie Jeng 著、『Design and Modeling of Deep-Submicrometer MOSFETs』、ERL Memo Nos. ERL M90/90, Electronics Research Laboratory University of California, Berkeley, October 1990 参照)
6	MOS6 (T. Sakurai, A. R. Newton 共著、『A Simple MOSFET Model for Circuit Analysis and its application to CMOS gate delay analysis and series-connected MOSFET Structure』、ERL Memo No. ERL M90/19, Electronics Research Laboratory, University of California, Berkeley, March 1990 参照)

- 8 BSIM3v3.3.0: University of California, Berkeley が発表、2005 年 7 月 29 日時点
- 9 BSIMSOI3.2(シリコン・オン・インシュレータ): University of California, Berkeley の BSIM Research Group が発表、2004 年 2 月
- 12 EKV 2.6: Ecole Polytechnique Federale de Lausanne が発表したコードに基づく。
<http://legwww.epfl.ch/ekv> および『The EPFL-EKV MOSFET Model Equations for Simulation, Version 2.6』、M. Bucher、C. Lallement、F. Theodoloz、C. Enz、F. Krummenacher 共著、EPFL-DE-LEG, June 1997 を参照。
- 14 BSIM4.6.1: University of California, Berkeley の BSIM Research Group が発表、2007 年 5 月 18 日。
- 73 HiSIMHV 1.2: 広島大学、STARC が発表

レベル 1 からレベル 3 の MOSFET の DC 特性は、デバイス・パラメータ VTO、KP、LAMBDA、PHI、および GAMMA によって定義されます。これらのパラメータはプロセス・パラメータ(NSUB、TOX...)が与えられている場合に計算されますが、ユーザが指定した値が常に優先されます。VTO は、N チャネル(P チャネル)デバイスの場合、エンハンスメント・モードでは正(負)であり、デプレッション・モードでは負(正)です。電荷の蓄積は以下の各要素によってモデル化されます。(1)オーバーラップ容量を表わす 3 つの固定容量コンデンサ、CGSO、CGDO、および CGBO、(2)ゲート、ソース、ドレイン、およびバルク領域間に分散した非線形の薄い酸化膜容量、(3)底面と周囲に分けられた 2 つの基板接合部の非線形の空乏層容量、それぞれ接合部電圧の MJ 乗および MJSW 乗に比例する。また、電荷の蓄積はパラメータ CBD、CBS、CJ、CJSW、MJ、MJSW、および PB によって決まります。電荷蓄積効果は、Meyer によって提唱された区分線形電圧依存の静電容量モデルによってモデル化されます。薄い酸化膜の電荷蓄積効果の扱いは、レベル 1 のモデルの場合はわずかに異なります。これらの電圧依存静電容量が含まれるのは、Tox を指定した場合に限られます。

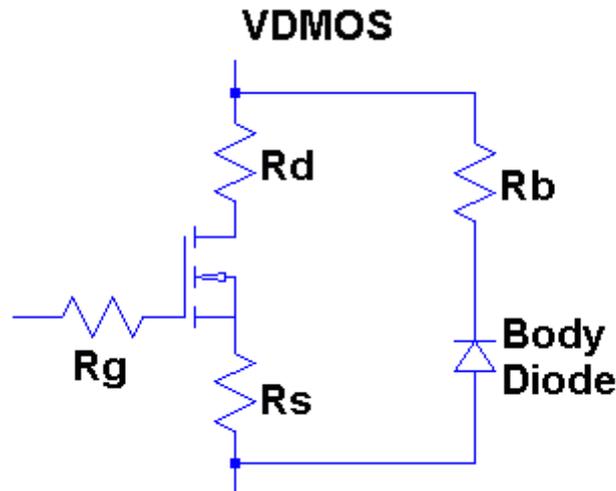
接合部を記述しているパラメータ間には多少の重複部分があります。たとえば、逆方向電流は Is[Amp]または Js[Amp/m/m]で指定できます。最初のパラメータは絶対値ですが、2 番目のパラメータには Ad および As を掛けて、それぞれドレイン接合部およびソース接合部の逆方向電流を示します。また、ゼロバイアスでの接合容量についても同じ考え方が当てはまり、一方は CBD および CBS[Farad]で、もう一方は CJ[Farad/m/m]です。ドレインおよびソースの寄生直列抵抗は、RD および RS[Ω]、または RSH[Ω/□]と表すことができます。後者には、デバイス行の正方形の数 NRD および NRS の入力を乗算します。

MOSFET レベル 1、2、および 3 のパラメータを以下に示します。

名前	概要	単位	デフォルト値	例
Vto	ゼロバイアスでのしきい値電圧	V	0	1.0
Kp	トランスコンダクタンス・パラメータ	A/V ²	2E-05	3E-05
Gamma	バルクしきい値パラメータ	V ^{1/2}	0.	0.37
Phi	表面反転電位	V	0.6	0.65
Lambda	チャンネル長の調整(レベル 1 および 2)	1/V	0.	0.02
wd	横方向拡散幅の減少量	m	0.	0.5u
Rd	ドレインのオーミック抵抗	Ω	0.	1.
Rs	ソースのオーミック抵抗	Ω	0.	1.
Rg	ゲートのオーミック抵抗	Ω	0.	1.
Rb	バルクのオーミック抵抗	Ω	0.	1.
Rds	ドレイン-ソース間分流抵抗	Ω	0.	1Meg
Cbd	ゼロバイアスでの B-D 接合容量	F	0.	20f
Cbs	ゼロバイアスでの B-S 接合容量	F	0.	20f
Is	バルク接合部の飽和電流	A	1E-14	1E-15
N	バルク・ダイオードの放射係数	-	1.	
Pb	バルクの接合部電位	V	0.8	0.87
tt	バルク接合部の遷移時間	s	0	1n
Cgso	チャンネル幅 1m 当たりのゲート-ソース間オーバーラップ容量	F/m	0.	4E-11
Cgdo	チャンネル幅 1m 当たりのゲート-ドレイン間オーバーラップ容量	F/m	0.	4E-11
Cgbo	チャンネル幅 1m 当たりのゲート-バルク間オーバーラップ容量	F/m	0.	2E-10
Rsh	ドレイン拡散とソース拡散のシート抵抗	Ω	0.	10.
Cj	接合面積 1 平方メートル当たりのゼロバイアス時バルク接合部底面の容量	F/m ²	0.	2E-04
Mj	バルク接合部底面の濃度勾配係数	-	0.5	0.5
Cjsw	接合周囲長 1 メートル当たりのゼロバイアス時バルク接合部サイドウォールの容量	F/m	0.	1p
Mjsw	バルク接合部サイドウォールの濃度勾配係数	-	.50 level 1 .33 level 2,3	
Js	接合面積 1 平方メートル当たりのバルク接合部飽和電流	A/m ²	0.	1u
Jssw	サイドウォール 1 メートル当たりのバルク接合部飽和電流	A/m	0.	1n
Tox	酸化膜厚	m	1E-07	1E-07

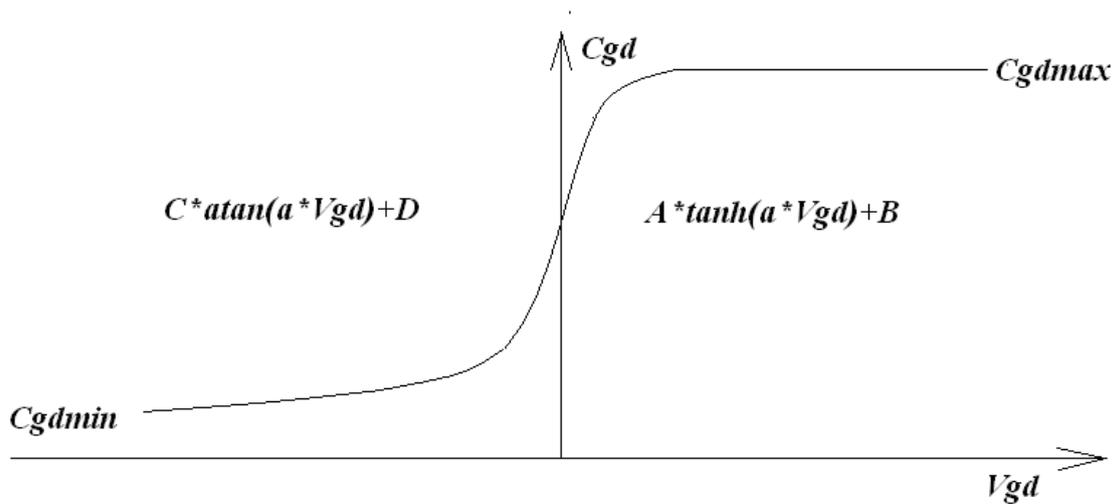
Nsub	基板のドーピング濃度	1/cm ³	0.	4E+15
Nss	表面準位密度	1/cm ²	0.	1E+10
Nfs	速い表面準位(レベル 2 および 3)	1/cm ²	0.	1E+10
Xd	空乏層幅(レベル 3)	m	0.	100n
TPG	ゲート材料の型: 基板と反対の場合は+1 基板と同じ場合は-1 Al ゲートの場合は 0	-	1	
Xj	冶金学上の接合深さ(レベル 2 および 3)	m	0.	1μ
Ld	横方向拡散	m	0.	0.8μ
Uo	表面移動度	cm ² /V/s	600	700
Ucrit	移動度低下の臨界磁界(レベル 2)	V/cm	1E+04	1E+04
Uexp	移動度低下時の臨界磁界指数(レベル 2)	-	0.	0.1
Utra	横磁界係数(レベル 2)	-	0.	0.3
Vmax	最大キャリア流動速度(レベル 2 および 3)	m/s	0.	5E+04
Neff	全チャネル電荷指数(レベル 2)	-	1.	5.
Kf	フリッカ・ノイズ係数	-	0.	1E-26
Af	フリッカ・ノイズ指数	-	1.	1.2
Nlev	ノイズ式セレクト	-	0.	1.
Gdsnoi	nlev=3 の場合のショット・ノイズ係数	-	1.	2.
Fc	順バイアス時の空乏層容量式の係数	-	0.5	
Delta	しきい電圧に対する幅の影響(レベル 2 および 3)	-	0.	1.
Theta	移動度調整(レベル 3)	-	0.	0.1
Eta	静的帰還(レベル 3)	-	0.	1.
Kappa	飽和磁界(レベル 3)		0.2	0.5
Tnom	パラメータ測定温度	°C	27	50
L	デフォルトの長さ	m	defl	20u
W	デフォルトの幅	m	defw	20u
Ad	デフォルトのドレイン面積	m ²	defad	200p
As	デフォルトのソース面積	m ²	defas	200p
Pd	デフォルトのドレイン周囲長	m	0.	20u
Ps	デフォルトのソース周囲長	m	0.	20u
Nrd	デフォルトのドレイン面積等価正方形数	-	0.	1
Nrs	デフォルトのソース面積等価正方形数	-	0.	1
Nrg	デフォルトのゲート面積等価正方形数	-	0.	1
Nrb	デフォルトのバルク面積等価正方形数	-	0.	1
Lmin	ピン長さの下限	m	0.	10u

Lmax	ピン長さの上限	m	0.	20u
Wmin	ピン幅の下限	m	0.	10u
Wmax	ピン幅の上限	m	0.	20u



ボード・レベルのスイッチ・モード電源で広く使用されるディスクリートの縦型二重拡散 MOSFET トランジスタ (VDMOS) は、前述のモノリシック MOSFET モデルとは動作の性質が異なります。特に、(i) VDMOS トランジスタのボディ・ダイオードは、モノリシック MOSFET の基板ダイオードとは外部端子との接続が異なり、(ii) ゲート・ドレイン間容量 (C_{gd}) の非線形性は、モノリシック MOSFET モデルの簡単な傾斜型静電容量ではモデル化できません。VDMOS トランジスタでは、 C_{gd} によってゲート・ドレイン間電圧 (V_{gd}) は突然約 0V に変化します。 V_{gd} が負の場合、 C_{gd} の物理的な基盤となるのは、ゲートを一方の電極とし、ダイ裏面のドレインをもう一方の電極とするコンデンサです。この静電容量は、非導通状態のダイの厚さにより、かなり低い値になります。しかし、 V_{gd} が正の場合はダイが導通しており、 C_{gd} の物理的基盤となるのはゲート酸化膜の厚さからなるコンデンサです。

パワー MOSFET の動作を再現するために、従来から精巧なサブ回路が使用されてきました。計算速度、収束の信頼性、およびモデル記述の簡略化のために、この動作をカプセル化する新しい固有の SPICE デバイスが作成されました。DC モデルはレベル 1 のモノリシック MOSFET と同じですが、トランスコンダクタンスをスケールリングなしで直接指定できるように、長さと幅がデフォルトで 1 になるという点が異なります。AC モデルは以下のとおりです。ゲート・ソース間容量は一定と解釈されます。ゲート・ソース間電圧が負にならない場合は、これがパワー MOSFET に対する良好な近似であることが経験的に判明しました。ゲート・ドレイン間容量は、経験的に判明した以下の形状に従います。



V_{gd} が正の場合、 C_{gd} は V_{gd} の双曲線正接として変化します。 V_{gd} が負の場合、 C_{gd} は V_{gd} の逆正接として変化します。モデルパラメータ a 、 C_{gdmax} 、および C_{gdmin} は、ゲート-ドレイン間容量をパラメータ化します。ソース-ドレイン間容量は、ソース抵抗とドレイン抵抗の外側でソース電極とドレイン電極の間に接続されているボディ・ダイオードの傾斜型静電容量によって与えられます。

名前	概要	単位	デフォルト値	例
Vto	しきい値	V	0	1.0
Kp	トランスコンダクタンス・パラメータ	A/V ²	1.	.5
Phi	表面反転電位	V	0.6	0.65
Lambda	チャネル長調整	1/V	0.	0.02
mtriode	三極管領域での導電率乗数(三極管領域と飽和領域の個別の適合が可能になります)	-	1.	2.
subthres	2乗から指数関数的なしきい値以下の導通に切り替わる(V_{ds} 1V 当たりの)電流	A/V	0.	1n
BV	V_{ds} のブレイクダウン電圧	V	Infin.	40
IBV	$V_{ds}=BV$ のときの電流	A	100pA	1u
NBV	V_{ds} ブレイクダウンの放射係数	-	1.	10
Rd	ドレインのオーミック抵抗		0.	1.
Rs	ソースのオーミック抵抗		0.	1.
Rg	ゲートのオーミック抵抗		0.	2.
Rds	ドレイン-ソース間分流抵抗		Infin.	10Meg
Rb	ボディ・ダイオードのオーミック抵抗		0.	.5
Cjo	ゼロバイアスでのボディ・ダイオード接合容量	F	0.	1n

Cgs	ゲート・ソース間容量	F	0.	500p
Cgadmin	非線形 G-D 間容量の最小値	F	0.	300p
Cgdmax	非線形 G-D 間容量の最大値	F	0.	1000p
A	非線形の Cgd 容量パラメータ	-	1.	.5
Is	ボディ・ダイオードの飽和電流	A	1E-14	1E-15
N	バルク・ダイオードの放射係数	-	1.	
Vj	ボディ・ダイオードの接合電位	V	1.	0.87
M	ボディ・ダイオードの濃度勾配係数	-	0.5	0.5
Fc	順バイアス時の空乏層容量式のボディ・ダイオード係数	-	0.5	
oneway	電流がチャネルの一方向にのみ流れることができることを示す挙動モデリング・フラグ	-	-	-
tt	ボディ・ダイオードの遷移時間	sec	0.	10n
Eg	Is に対する温度の影響に対応するボディ・ダイオード活性化エネルギー	eV	1.11	
Xti	ボディ・ダイオードの飽和電流温度指数	-	3.	
L	長さのスケーリング	-	1.	
W	幅のスケーリング	-	1.	
Kf	フリッカ・ノイズ係数	-	0.	
Af	フリッカ・ノイズ指数	-	1.	
nchan[*]	N チャネル VDMOS	-	(true)	-
pchan[*]	P チャネル VDMOS	-	(false)	-
Tnom	パラメータ測定温度	°C	27	50
Lmin	ビン長さの下限	m	0.	10u
Lmax	ビン長さの上限	m	0.	20u
Wmin	ビン幅の下限	m	0.	1
Wmax	ビン幅の上限	m	0.	10

*] モデル名 VDMOS は N チャネル・デバイスと P チャネル・デバイスの両方に使用されます。極性はデフォルトで N チャネルになります。P チャネルを指定するには、キーワード「pchan」を使用してモデルにフラグを立てます。たとえば、「.model xyz VDMOS(Kp = 3 pchan)」と記述すると、P チャネル・トランジスタが定義されます。

モデルに電圧定格と公称性能を注釈付けすることができます。この情報はデバイスの選択を支援するために回路図取り込み GUI に表示されますが、シミュレーションでの電氣的動作には直接影響しません。以下のパラメータを指定できます。

名前	概要	単位
Vds	ドレイン・ソース間の電圧定格	V

Ron	公称のオン抵抗	Qg
Qg	Ron に達するために必要な公称のゲート電荷	C
mfg	メーカーの名前	-

0. 損失の大きい伝送線路

シンボル名: LTLIN

構文: Oxxx L+ L- R+ R- <model>

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

これは損失の大きい単一導体の伝送線路です。N1 と N2 はポート 1 にあるノードです。N3 と N4 はポート 2 にあるノードです。この回路素子の電気的特性を定義するためには、モデル・カードが必要です。

損失の大きい伝送線路のモデルパラメータ

名前	概要	単位/タイプ	デフォルト値
R	単位長当たりの抵抗	Ω	0.
L	単位長当たりのインダクタンス	H	0.
G	単位長当たりの導電率	$1/\Omega$	0.
C	単位長当たりの静電容量	F	0.
Len	単位長の数	-	0.
Rel	区切り点を設定する導関数の相対変化率		1.
Abs	区切り点を設定する導関数の絶対変化率		1.
NoStepLimit	時間刻みを線路の遅延より短い値に制限しない	(flag)	設定なし
NoControl	複雑な時間刻み制御を試みない	(flag)	設定なし
LinInterp	直線補間を使用する	(flag)	設定なし
MixedInterp	2次補間が失敗するように見える場合は直線補間を使用する	(flag)	設定なし
CompactRel	履歴圧縮時の Reltol		RELTOL
CompactAbs	履歴圧縮時の Abstol		ABSTOL
TruncNr	時間刻み制御にニュートン-ラフソン法を使用する	(flag)	設定なし
TruncDontCut	インパルス応答誤差を小さく保つために時間刻みを制限しない	(flag)	設定なし

Q. バイポーラ・トランジスタ

シンボル名: NPN、PNP、NPN2、PNP2

構文: 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・トランジスタでは、その特性を指定するのにモデル・カードが必要です。モデル・カード・キーワード NPN および PNP は、トランジスタの極性を示します。

バイポーラ接合トランジスタ・モデルは、Gummel と Poon の一体型電荷制御モデルを改変したものです。この修正版 Gummel-Poon モデルでは、オリジナル・モデルを拡張して、高バイアス・レベルでのいくつかの効果、準飽和状態、および基板の導電率を組み込んでいます。特定のパラメータを指定しなかった場合、モデルは自動的に Ebers-Moll モデルに単純化されます。DC モデルは、順方向の電流利得特性を決定するパラメータ I_s 、 B_f 、 N_f 、 I_{se} 、 I_{kf} 、および N_e 、逆方向の電流利得特性を決定するパラメータ I_s 、 B_r 、 N_r 、 I_{sc} 、 I_{kr} 、および N_c 、さらに順方向領域と逆方向領域の出力導電率を決定するパラメータ V_{af} および V_{ar} によって定義されます。3つのオーミック抵抗 R_b 、 R_c 、および R_e が組み込まれますが、 R_b は電流に大きく依存する可能性があります。ベース電荷の蓄積は、以下の各要素によってモデル化されます。(1) 順方向と逆方向の遷移時間 T_f および T_r 、(2) バイアス依存性のある順方向遷移時間 T_f (必要な場合)、(3) 非線形の空乏層容量 (B-E 接合の場合は C_{je} 、 V_{je} 、および M_{je} によって決まり、B-C 接合の場合は C_{jc} 、 V_{jc} 、 M_{jc} によって決まり、コレクタ-基板間接合の場合は C_{js} 、 V_{js} 、および M_{js} によって決まります)。飽和電流の温度依存性 (I_s) は、エネルギー・ギャップ (E_g) と飽和電流温度指数 (XTI) によって決まります。さらに、ベース電流の温度依存性は、新しいモデルの電流増幅率温度指数 XTB によってモデル化されます。指定の値は温度 T_{NOM} で測定されているとみなされます。これは、OPTIONS 制御行で指定するか、.model 行での指定によって変更することができます。修正版 Gummel-Poon モデルで使用される BJT パラメータを以下に示します。

修正版 Gummel-Poon の BJT パラメータ

名前	概要	単位	デフォルト値
I_s	輸送飽和電流	A	1E-16
I_{bc}	ベース-コレクタ間飽和電流	A	I_s
I_{be}	ベース-エミッタ間飽和電流	A	I_s
B_f	理想的な順方向電流増幅率の最大値	-	100
N_f	順方向電流の放射係数	-	1.
V_{af}	順方向アーリー電圧	V	Infin.

lkf	順方向電流増幅率の大電流時ロールオフ特性のコーナー	A	Infin.
nk	大電流時のロールオフ係数	-	.5
lse	B-E 間漏れ電流の飽和電流	A	0.
Ne	B-E 間漏れ電流の放射係数	-	1.5
Br	理想的な逆方向電流増幅率の最大値	-	1.
Nr	逆方向電流の放射係数	-	1.
Var	逆方向アーリー電圧	V	Infin.
lkr	逆方向電流増幅率の大電流時ロールオフ特性のコーナー	A	Infin.
isc	B-C 間漏れ電流の飽和電流	A	0
Nc	B-C 間漏れ電流の放射係数	-	2
Rb	ゼロバイアスでのベース抵抗		0
lrb	ベース抵抗が途中でその最小値に低下する電流	A	Infin.
Rbm	大電流時のベース抵抗の最小値	Ω	Rb
Re	エミッタ抵抗	Ω	0.
Rc	コレクタ抵抗	Ω	0.
Cje	ゼロバイアス時の B-E 間空乏層容量	F	0.
Vje	B-E 間拡散電位	V	0.75
Mje	B-E 接合部の指数因子	-	0.33
Tf	理想的な順方向遷移時間	sec	0.
Xtf	Tf のバイアス依存性の係数	-	0.
Vtf	Tf の Vbc 依存性を示す電圧	V	Infin.
ltf	Tf に対する影響に対応する大電流パラメータ	A	0.
Ptf	freq=1/(Tf*2*)Hz での過剰位相	$^{\circ}$	0.
Cjc	ゼロバイアス時の B-C 間空乏層容量	F	0.
Vjc	B-C 間拡散電位	V	0.75
Mjc	B-C 接合部の指数因子	-	0.33
Xcjc	内部ベース・ノードに接続された B-C 間空乏層容量の一部	-	1.
Xcjc2	内部ベース・ノードと外因性コレクタの間に接続された B-C 間空乏層容量の一部	-	0
extsub	基板容量の電荷分割に使用される、より本質的なコレクタ・ノードの外因性	-	0
Tr	理想的な逆方向遷移時間	sec	0.
Cjs	ゼロバイアスでのコレクタ-基板間容量	F	0.
Xcjs	内部で Rc に接続されている Cjs の一部	F	0.

Vjs	基板接合部の拡散電位	V	0.75
Mjs	基板接合部の指数因子	-	0.
Xtb	順方向と逆方向の電流増幅率温度指数	-	0.
Eg	Is に対する温度の影響に対応するエネルギー・ギャップ	eV	1.11
Xti	Is に対する影響の温度指数	-	3.
Kf	フリッカ・ノイズ係数	-	0.
Af	フリッカ・ノイズ指数	-	1.
Fc	順バイアス時の空乏層容量式の係数	-	0.5
subs	LPNP を使用しない場合の形状セクタ: 1 は縦型を意味し、2 は横型を意味する	-	NPN:1 PNP:2
BVcbo	コレクタ-ベース間ブレイクダウン電圧	-	Infin.
nBVcbo	コレクタ-ベース間ブレイクダウン電圧の係数	-	5
BVbe	ベース-エミッタ間ブレイクダウン電圧	V	Infin.
lbvbe	ベース-エミッタ間ブレイクダウン電圧時の電流	A	1E-10
nbvbe	ベース-エミッタ間ブレイクダウン電圧の係数	-	1.
Tnom	パラメータ測定温度	°C	27
Cn	正孔移動度の準飽和状態温度係数		2.42 NPN 2.2 PNP
D	散乱が制限された正孔キャリア速度の準飽和状態温度係数		.87 NPN .52 PNP
Gamma	エピタキシャル領域のドーピング率		1E-11
Qco	エピタキシャル領域の電荷係数	Coul	0.
Quasimod	温度依存性の準飽和状態フラグ	-	(設定なし)
Rco	エピタキシャル領域の抵抗	Ω	0.
Vg	0° K での準飽和状態を外挿したバンドギャップ電圧	V	1.206
Vo	キャリア移動度の屈曲点電圧	V	10.
Tre1	Re の 1 次温度係数	1/°C	0.
Tre2	Re の 2 次温度係数	1/°C ²	0.
Trb1	Rb の 1 次温度係数	1/°C	0.
Trb2	Rb の 2 次温度係数	1/°C ²	0.
Trc1	Rc の 1 次温度係数	1/°C	0.
Trc2	Rc の 2 次温度係数	1/°C ²	0.
Trm1	Rmb の 1 次温度係数	1/°C	0.
Trm2	Rmb の 2 次温度係数	1/°C ²	0.
Iss	基板接合部の飽和電流	A	0.

Ns	基板接合部の放射係数	-	1.
Tvaf1	Vaf の 1 次温度係数	1/°C	0.
Tvaf1	Vaf の 2 次温度係数	1/°C ²	0.
Tvar1	Var の 1 次温度係数	1/°C	0.
Tvar1	Var の 2 次温度係数	1/°C ²	0.
Tikf1	Ikf の 1 次温度係数	1/°C	0.
Tikf1	Ikf の 2 次温度係数	1/°C ²	0.
Trbm1	Rbm の 1 次温度係数	1/°C	0.
Trbm2	Rbm の 2 次温度係数	1/°C ²	0.
Tbvcbo1	BVcbo の 1 次温度係数	1/°C	0.
Tbvcbo1	BVcbo の 2 次温度係数	1/°C ²	0.

モデルにデバイスの定格を注釈付けすることができます。この情報はデバイスの選択を支援するために回路図取り込み GUI に表示されますが、シミュレーションでの電氣的動作には直接影響しません。

以下のパラメータを指定できます。

名前	概要	単位
Vceo	ベースがフロート状態のコレクタ-エミッタ間電圧の最大値	V
Icrating	最大コレクタ電流	A
mfg	メーカーの名前	-

モデルパラメータ「level」を使用して、LTspice で別のタイプの BJT を指定できます。

NXP(Philips) の MEXTRAM 504 トランジスタを使用するには Level=504 を設定します。

DAnalyse GmbH(ドイツ、ベルリン)の Dietmar Warning 博士により、寛大にもソース・コードを寄贈していただいたので、LTspice には、VBIC の一種が組み込まれています。代替デバイスを使用するには、Level=9 を設定します。Level 4 は Level 9 の同義語です。Warning 博士によって次の資料が提供されました。

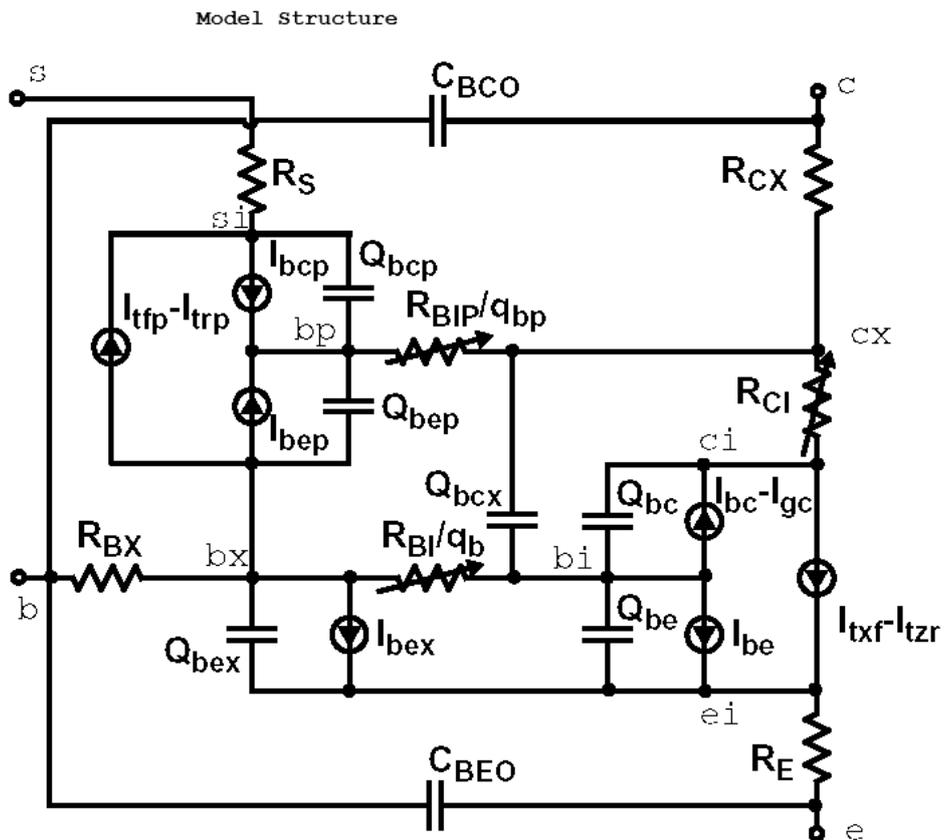
VBIC - Vertical Bipolar Inter Company model

VBIC モデルは、現代の半導体技術における一体型バイポーラ・トランジスタに焦点を合わせた Standard Gummel-Poon(SGP)モデルの拡張開発版です。Kull と Nagel による修正版 Quasi-Saturation モデルを実装したので、スイッチング・トランジスタの特殊な出力特性をモデル化することも可能です。このモデルは、シリコン、SiGe、および III-V HBT デバイスでは、SGP モデルの代わりに広く使用されています。

Standard Gummel-Poon モデルと比較した VBIC の機能

- 集積化プロセスでの寄生デバイスの一体化基板トランジスタ
- 弱い電子なだれモデルおよびベース-エミッタ間ブレークダウン・モデル
- 改善されたアーリー効果モデリング
- I_c と I_b の物理的な分離
- 改善された空乏層容量モデル
- 改善された温度モデリング

モデル構造



VBIC パラメータ

VBIC モデルは SGP モデルに基づいているので、SGP パラメータから始めて、いくつかの変換を実行できます。以下のパラメータは VBIC バージョン 1.2 からのものですが、このバージョンは、過剰位相回路網がなく自己発熱の影響がない 4 端子バージョンで LTSpice に実装されています。SGP から VBIC に切り替えるには、別途パラメータ・レベルを 9 に設定してください。

名前	概要	単位	デフォルト値
Tnom	パラメータ測定温度	°C	27.
Rcx	外部コレクタ抵抗	Ω	0.1
Rci	固有コレクタ抵抗	Ω	0.1
Vo	エピ領域のドリフト飽和電圧	V	Infin.
gamm	エピ領域のドーピング・パラメータ		0.0
hrcf	大電流の RC 係数		Infin.
Rbx	外部ベース抵抗	Ω	0.1
Rbi	固有ベース抵抗	Ω	0.1
Re	固有エミッタ抵抗	Ω	0.1
Rs	固有基板抵抗	Ω	0.1
Rbp	寄生ベース抵抗	Ω	0.1
Is	輸送飽和電流	A	1E-16
nf	順方向の放射係数		1.
nr	逆方向の放射係数		1.
Fc	順バイアス時の空乏層容量制限値		0.9
Cbeo	外因性 B-E 間オーバーラップ容量	F	0.0
Cje	ゼロバイアス時の B-E 間空乏層容量	F	0.0
pe	B-E 間拡散電位	V	0.75
me	B-E 間接合の濃度勾配係数		0.33
Aje	B-E 間容量の平滑化係数		-0.5
Cbco	外因性 B-C 間オーバーラップ容量	F	0.
Cjc	ゼロバイアス時の B-C 間空乏層容量	F	0.
Qco	エピ領域の電荷パラメータ	C	0.
Cjep	B-C 間の外因性ゼロバイアス時容量	F	0.
pc	B-C 間拡散電位	V	0.75
mc	B-C 間接合の濃度勾配係数		0.33
Ajc	B-C 間容量の平滑化係数		-0.5
Cjcp	ゼロバイアス時の S-C 間容量	F	0.
ps	S-C 間接合の拡散電位	V	0.75
ms	S-C 間接合の濃度勾配係数		0.33

Ajs	S-C 間容量の平滑化係数		-0.5
lbei	理想的な B-E 間飽和電流	A	1E-18
wbe	Vbei からの IBEL の一部、Vbex からは 1-WBE		1.
nei	理想的な B-E 間放射係数		1.
lben	非理想的な B-E 間飽和電流	A	0.
nen	非理想的な B-E 間放射係数		2.
ibci	理想的な B-C 間飽和電流	A	1E-16
Nci	理想的な B-C 間放射係数		1.
ibcn	非理想的な B-C 間飽和電流	A	0.
ncn	非理想的な B-C 間放射係数		1.
avc1	avc1	1/V	0.
avc2	B-C 間の弱い電子なだれパラメータ 2	1/V	0.
isp	寄生輸送飽和電流	A	0.
wsp	ICCP の一部		1.
nfp	寄生の順方向放射係数		1.
lbeip	理想的な寄生 B-E 間飽和電流	A	0.
ibenp	非理想的な寄生 B-E 間飽和電流	A	0.
ibcip	理想的な寄生 B-C 間飽和電流	A	0.
ncip	理想的な寄生 B-C 間放射係数		1.
lbcnp	非理想的な寄生 B-C 間飽和電流	A	0.
ncnp	非理想的な寄生 B-C 間放射係数		2.
Vef	順方向アーリー電圧		Infin.
Ver	逆方向アーリー電圧		Infin.
lkf	順方向屈曲点電流	A	Infin.
ikr	逆方向屈曲点電流	A	Infin.
lkp	寄生屈曲点電流	A	Infin.
tf	理想的な順方向遷移時間	s	0.
Qtf	ベース幅の調整による TF の変化		0.
Xtf	TF のバイアス依存性の係数		0.
Vtf	TF の VBC 依存性を示す電圧	V	Infin.
Itf	TF の大電流依存性	A	Infin.
tr	理想的な逆方向遷移時間	sec	0.
Td	順方向の過剰位相遅延時間	sec	0.
kfn	B-E 間フリッカ・ノイズ係数		0.
afn	B-E 間フリッカ・ノイズ指数		1.
bfm	B-E 間フリッカ・ノイズの 1/f 依存性		1.0
Xre	RE の温度指数		0.

Xrbi	RBI の温度指数		0.
Xrci	RCI の温度指数		0.
Xrs	RS の温度指数		0.
Xvo	VO の温度指数		0.
Ea	IS に対する活性化エネルギー	V	1.12
Eaie	IBEI に対する活性化エネルギー	V	1.12
Eaic	IBCI/IBEIP に対する活性化エネルギー	V	1.12
Eais	IBCIP に対する活性化エネルギー	V	1.12
Eane	IBEN に対する活性化エネルギー	V	1.12
Eanc	IBCN/IBENP に対する活性化エネルギー	V	1.12
Eans	IBCNP に対する活性化エネルギー	V	1.12
Xis	IS の温度指数		3.
Xii	IBEI、IBCI、IBEIP、IBCIP の温度指数		3.
Xin	IBEN、IBCN、IBENP、IBCNP の温度指数		3.
Tnf	NF の温度指数		0.
Tavc	AVC2 の温度指数		0.
rth	熱抵抗	K/W	0.
Cth	熱キャパシタンス	Ws/K	0.
Vrt	内部 B-C 接合のパンチスルー電圧	V	0.
Art	リーチスルーの場合の平滑化パラメータ		0.1
Ccso	C-S 間固定容量	F	0.
qbm	SGP qb 表現の選択		0.
nkf	大電流時の電流増幅率ロールオフ特性		0.5
Xikf	IKF の温度指数		0.
Xrcx	RCX の温度指数		0.
Xrbx	RBX の温度指数		0.
Xrbp	RBP の温度指数		0.
Isrr	順方向と逆方向で別個の IS		1.
Xisr	ISR の温度指数		0.
dear	ISRR の活性化エネルギーの差分		0.
Eap	ISP の励起エネルギー		1.12
Vbbe	B-E 間ブレークダウン電圧	V	0.
nbbe	B-E 間ブレークダウンの放射係数		1.
lbbe	B-E 間ブレークダウン電流		1E-06
Tvbbe1	VBBE の 1 次温度係数		0.
Tvbbe2	VBBE の 2 次温度係数		0.
Tnbbe	NBBE の温度係数		0.

ebbe	$\exp(-V_{BBE}/(N_{BBE} \cdot V_{tv}))$		0.
dtemp	局所的な温度差	°	0.
Vers	改訂バージョン		1.2
Vref	参照バージョン		0.

参考文献

C. C. McAndrew 他、『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 年

C. C. McAndrew 他、VBIC95、『The Vertical Bipolar Inter-Company Model』、IEEE Journal of Solid State Circuits, vol. 31, No. 10、1996 年 10 月

C. C. McAndrew、VBIC Model Definition, Release 1.2、1999 年 9 月 18 日

R. 抵抗

シンボル名: RES、RES2

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

抵抗は、ノード n1 と n2 の間に単純な線形抵抗を実現します。パラメータ tc を指定することにより、抵抗インスタンスごとに温度依存性を定義できます。抵抗 (R) は次式で表されます。

$$R = R0 * (1. + dt * tc1 + dt**2 * tc2 + dt**3 * tc3 + ...)$$

ここで、R0 は公称温度での抵抗であり、dt は抵抗の温度と公称温度との差です。

S. 電圧制御型スイッチ

シンボル名: SW

構文: 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 nc-の間の電圧により、ノード n1 と n2 の間のスイッチのインピーダンスを制御します。スイッチの動作を定義するためには、モデル・カードが必要です。回路図ファイル

.¥examples¥Educational¥Vswitch.asc を参照して、SPICE 指令として回路図に直接配置されているモデル・カードの例を確認してください。

電圧制御型スイッチのモデルパラメータ

名前	概要	単位	デフォルト値
Vt	しきい値	V	0.
Vh	ヒステリシス電圧	V	0.
Ron	オン抵抗	Ω	1.
Roff	オフ抵抗	Ω	1/Gmin
Lser	直列インダクタンス	H	0.
Vser	直列電圧	V	0.
Ilimit	電流制限	A	Infin.

ヒステリシス電圧(Vh)の値に応じて、スイッチには3つの異なる電圧制御モードがあります。Vhが0である場合は、入力電圧がしきい値より高いかどうかによって、スイッチは必ず完全にオンまたはオフになります。Vhが正の場合、スイッチは、あたかも作動点が Vt-Vh および Vt+Vh のシュミット・トリガ回路によって制御されるかのようにヒステリシスを示します。Vhは作動点間の電圧の半分であり、一般的な学術用語とは異なることに注意してください。Vhが負の場合、スイッチはオン状態のインピーダンスとオフ状態のインピーダンスを滑らかに切り替えます。切り替わるのは、制御電圧が Vt-Vh と Vt+Vh の間です。滑らかな切り替わりは、スイッチの伝導率の対数に一致する低次の多項式に従います。

レベル1スイッチの改良型で負のヒステリシスを持つレベル2の電圧制御スイッチもあります。レベル2のスイッチは、完全にオンすることも完全にオフすることはありません。制御電圧 Vc の関数としての伝導率は次のとおりです。

$$g(Vc) = \exp(A * \operatorname{atan}((Vc - Vt)/\operatorname{abs}(Vh)) + B)$$

ここで、

$$A = \log(R_{\text{off}} / R_{\text{on}}) / \pi$$

$$B = \log(1 / (R_{\text{off}} * R_{\text{on}})) / 2$$

また、レベル 2 のスイッチは、急激にではなく徐々に電流制限状態に移行します。制御電圧が固定の場合、I-V 曲線は次式で与えられます。

$$I(V) = I_{\text{limit}} * \tanh(g(V_c) * V)$$

ここで I_{limit} は、レベル 2 のスイッチの場合、デフォルトで 10A になります。

レベル 2 のスイッチは、フラグ「oneway」を指定するか、パラメータ V_{ser} で電圧降下を指定することにより、一方向だけで導通するオプションをサポートします。順方向の導通と逆方向の開回路との間を滑らかに移行するよう指定するには、パラメータ ϵ をゼロ以外の値に指定します。

U. 均一な RC 線路

シンボル名: URC

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

N1 と N2 は RC 線路が接続する 2 つの要素ノードであるのに対して、Ncom は容量の接続先となるノードです。MNAME はモデル名であり、LEN は RC 線路の長さ(メートル)です。lumps は(指定した場合)、RC 線路をモデル化するとき使用するひとかたまりの線分の数です。lumps を指定しなかった場合は、適当な数のかたまりを使用する推測が行われます。

URC モデルは 1974 年に L.Gertzbergg によって提唱されたモデルに由来します。このモデルを完成させるには、サブ回路型の URC 線路を内部生成ノードを持つひとかたまりの RC 部分の回路網に拡張します。RC 部分は、K を比例定数とし、URC 線路の中央に向かって増加する等比数列になっています。

ISPERL パラメータをゼロ以外の値に指定しない限り、URC 線路は厳密に抵抗部分とコンデンサ部分で構成されます。その場合、コンデンサは、逆バイアスされたダイオードと置き換えられますが、このダイオードのゼロバイアス時の接合容量は置き換え対象の容量と等しい値で、飽和電流は伝送線路 1m 当たり ISPERL(A) であり、オプションの直列抵抗は RSPERL(Ω/m)と等しい値です。

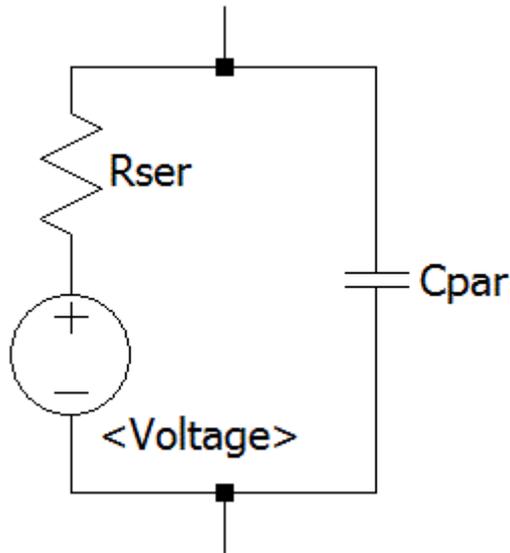
名前	概要	単位	デフォルト値
K	伝搬定数	-	2.
Fmax	目的とする最大の周波数	Hz	1G
Rperl	単位長当たりの抵抗	Ω	1K
Cperl	単位長当たりの静電容量	F	1E-15
Isperl	単位長当たりの飽和電流	A	0.
Rspert	単位長当たりのダイオードの抵抗	Ω	0.

V. 電圧源

シンボル名: VOLTAGE、BATTERY

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

この素子は、ノード n+ と n- の間に定電圧を供給します。AC 解析では、AC の値を解析周波数での信号源の振幅として使用します。直列抵抗と並列容量は設定できます。等価回路を以下に示します。



電圧源は、SPICE では伝統的に電流計として使用されており、電流制御素子の電流センサとして使用されません。Rser を指定した場合、F、H、または W の各素子の検出素子として電圧源を使用することはできません。ただし、電圧源を含むすべての回路素子の電流はプロットできます。

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

時間依存性のあるパルス電圧源

名前	概要	単位
Voff	初期値	V
Von	パルス値	V
Tdelay	遅延	sec
Tr	立ち上がり時間	sec
Tf	立ち下がり時間	sec
Ton	オン時間	sec
Tperiod	周期	sec
Ncycles	サイクル数(自走パルス関数の場合は省略)	cycles

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

時間依存性のある正弦波電圧源

名前	概要	単位
Voffset	DC オフセット	V
Vamp	振幅	V
Freq	周波数	Hz
Td	遅延	sec
Theta	減衰係数	1/sec
Phi	正弦波の位相	度
Ncycles	サイクル数(自走パルス関数の場合は省略)	cycles

Td より短い時間か、Ncycles の完了後の時間に動作させた場合、出力電圧は次式で与えられます。

$$\text{Voffset} + \text{Vamp} * \sin(\pi * \text{Phi} / 180)$$

それ以外の場合、出力電圧は次式で与えられます。

$$\text{Voffset} + \text{Vamp} * \sin((2. * \pi * \text{Fcar} * \text{time}) + \text{MDI} * \sin(2. * \pi * \text{Fsig} * \text{time}))$$

減衰係数(Theta)は、減衰時定数の逆数です。

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

時間依存性のある指数電圧源

名前	概要	単位
V1	初期値	V
V2	パルス値	V
Td1	立ち上がり遅延時間	sec
Tau1	立ち上がり時定数	sec
Td2	立ち下がり遅延時間	sec
Tau2	立ち下がり時定数	sec

時間が Td1 より短い場合、出力電圧は V1 です。時間が Td1 と Td2 の間の場合、出力電圧は次式で与えられます。

$$V1+(V2-V1)*(1-\exp(-(time-Td1)/Tau1))$$

時刻が Td2 より後の場合、出力電圧は次式で与えられます。

$$V1+(V2-V1)*(1-\exp(-(time-Td1)/Tau1))-(V2-V1)*(1-\exp(-(time-Td2)/Tau2))$$

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

時間依存性のある単一周波数 FM 電圧源

名前	概要	単位
Voff	DC オフセット	V
Vamp	振幅	V
Fcar	搬送周波数	Hz
MDI	変調指数	-
Fsig	信号周波数	Hz

この電圧は次式で与えられます。

The voltage is given by $Voff+Vamp*\sin((2*\pi*Fcar*time)+MDI*\sin(2.*\pi*Fsig*time))$

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

任意の区分線形電圧源

t1 より前の時刻では、電圧は V1 です。t1 と t2 の間の時刻では、電圧は v1 と v2 の間で直線的に変化します。任意の数の時刻点、電圧点を指定できます。最終時刻後の時刻では、電圧は最後の電圧になります。

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

これを使用すると、.wav ファイルを LTspice への入力として使用できます。<filename>は、.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またはネットリストが入っているディレクトリを基準に計算した相対パスです。二重引用符を使用してスペースを含むパスを指定できます。.wav ファイルには最大 65536 チャンネルが含まれている場合があります、0~65535 の番号が付いています。Chan を設定すると、どのチャンネルを使用するかを指定できます。デフォルトでは、最初のチャンネル(番号 0)が使用されます。.wav ファイルは-1V から 1V までのフルスケール・レンジがあると解釈されます。

この信号源は.tran 解析でのみ意味があります。

W. 電流制御型スイッチ

シンボル名: CSW

構文: 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 lh=-.5)

指定した電圧源を流れる電流により、スイッチのインピーダンスを制御します。電流制御型スイッチの動作を定義するためには、モデル・カードが必要です。

電流制御型スイッチのモデルパラメータ

名前	概要	単位	デフォルト値
It	しきい値電流	A	0.
lh	ヒステリシス電流	A	0.
Ron	オン抵抗	Ω	1.
Roff	オフ抵抗	Ω	1/Gmin

ヒステリシス電流(lh)の値に応じて、スイッチには3つの異なる電流制御モードがあります。lhが0である場合は、制御電流がしきい値より大きいかどうかによって、スイッチは必ず完全にオンまたはオフになります。lhが正の場合、スイッチは作動点電流が $It - lh$ および $It + lh$ のヒステリシスを示します。lhが負の場合、スイッチはオン状態のインピーダンスとオフ状態のインピーダンスを滑らかに切り替えます。

切り替わるのは、制御電流が $It-lh$ と $It+lh$ の間です。滑らかな切り替わりは、スイッチの伝導率の対数に一致する低次の多項式に従います。

X. サブ回路

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

サブ回路を使用すると、回路を定義してライブラリ内に格納し、後で名前を検索できるようになります。分圧器を定義して呼び出し、回路内で起動する例を以下に示します。

```
*
* 以下のコードで回路を呼び出します
X1 in out 0 divider top=9K bot=1K
V1 in 0 pulse(0 1 0 .5m .5m 0 1m)
*
* 以下はサブ回路の定義です
.subckt divider A B C
R1 A B {top}
R2 B C {bot}
.ends divider
*
.tran 3m
.end
```

Z. MESFET トランジスタ

シンボル名: MESFET

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

MESFET トランジスタでは、その特性を指定するのにモデル・カードが必要です。モデル・カード・キーワード NMF および PMF は、トランジスタの極性を指定します。MESFET モデルは、H.Statz らが『GaAs FET Device and Circuit Simulation in SPICE』（IEEE Transactions on Electron Devices, V34, Number 2, February, 1987 pp160-169）で記述した GaAs FET モデルに由来します。

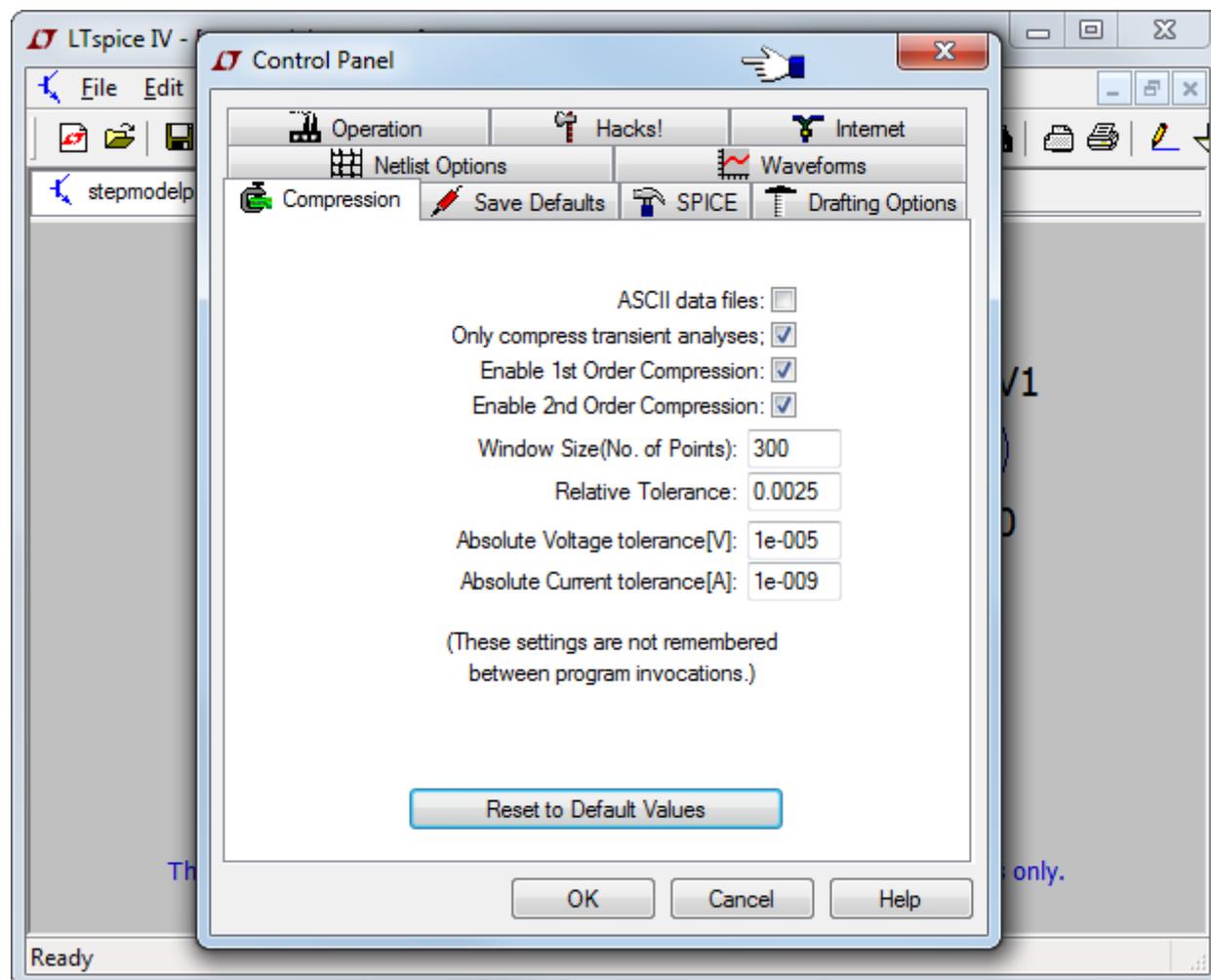
2つのオーミック抵抗 (R_d および R_s) が組み込まれています。電荷の蓄積は、ゲート-ドレイン間電圧とゲート-ソース間電圧の関数としての全ゲート電荷によってモデル化され、パラメータ C_{gs} 、 C_{gd} 、および P_b によって定義されます。

名前	概要	単位	デフォルト値
Vto	ピンチオフ電圧	V	-2.
Beta	トランスコンダクタンス・パラメータ	A/V^2	1E-04
B	ドーピングのテール拡張パラメータ	1/V	0.3
Alpha	飽和電圧パラメータ	1/V	2.
Lambda	チャンネル長調整	1/V	0.
Rd	ドレインのオーミック抵抗	Ω	0.
Rs	ソースのオーミック抵抗	Ω	0.
Cgs	ゼロバイアスでの G-S 接合容量	F	0.
Cgd	ゼロバイアスでの G-D 接合容量	F	0.
Pb	ゲートの接合部電位	V	1.
Kf	フリッカ・ノイズ係数	-	0.
Af	フリッカ・ノイズ指数	-	1.
Fc	順バイアス時の空乏層係数	-	0.5
Is	接合部の飽和電流	A	1E-14

Control Panel へのアクセス

「Control Panel」を表示するには、メニューコマンド「Tools」=>「Control Panel」を使用します。
ここでは、LTspice IV の多くの外観を設定できます。

Compression



LTspice は、生データ・ファイルを生成時のまま圧縮します。圧縮ファイルは、圧縮していないファイルの 1/50 に小さくすることができます。これは不可逆圧縮です。「Control Panel」のこのペインでは、圧縮実行時に生じる損失の程度を制御できます。

Window Size(No. of Points):

2 つの端点間に圧縮できる点の最大数。

Relative Tolerance:

圧縮データと非圧縮データの間で許容される相対誤差。

Absolute Voltage tolerance[V]:

圧縮アルゴリズムによって許容される電圧誤差。

Absolute Current tolerance[A]:

圧縮アルゴリズムによって許容される電流誤差。

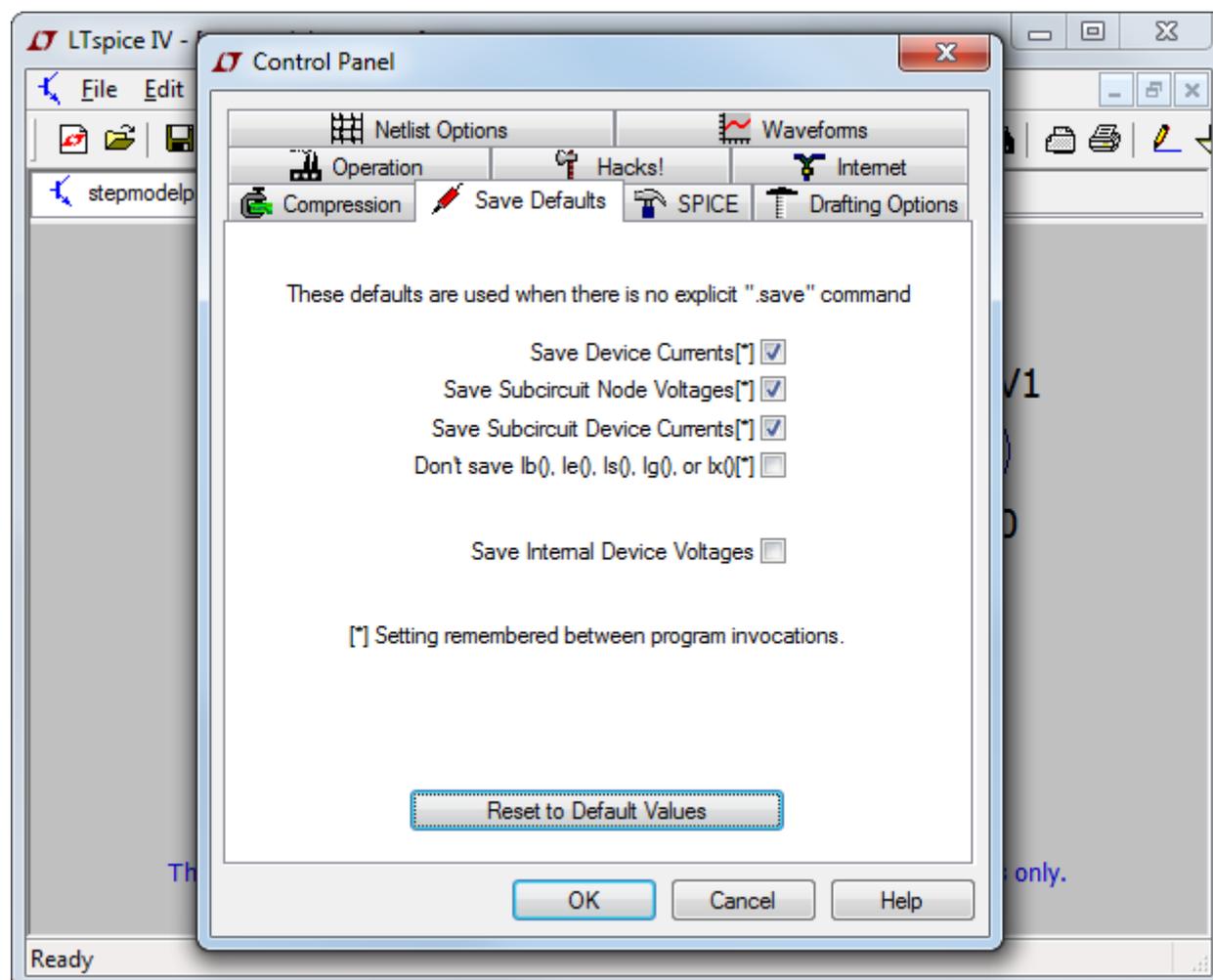
これらの圧縮設定は、デフォルト設定の使用を促すため、プログラムを複数回起動した場合は引き継がれません。これらの設定が「Control Panel」で使用できるのは、診断に利用するためです。

上記の許容差 (tolerance) と「Window Size」は、回路図に SPICE 指令として配置された .option 文でオプション・パラメータ plotreltol、plotvntol、plotabstol、および plotwinsize を使用して指定できます。.four 文を使用するか、データの解析後に SPICE 指令を使用して FFT を実行するときは、圧縮をオフにする必要がある場合があります。

```
.options plotwinsize=0
```

Save Defaults

これらの設定が使用されるのは、シミュレーション時にどのノードを保存するかを明示的に指定しなかった場合です。役に立つ設定は、「Save Device Currents」、「Save Subcircuit Node Voltages」、および「Save Subcircuit Device Currents」です。デバイス電圧および内部デバイス電圧は、内部プログラム開発専用です。



Save Device Currents:

デバイス電流と端子電流をプロットできるようにするには、このチェックボックスをオンにします。また、電力損失をプロットできるようにする場合もオンにする必要があります。

Save Subcircuit Node Voltages:

階層設計回路内の電圧をプロットするには、このチェックボックスをオンにする必要があります。

Save Subcircuit Device Currents:

階層設計回路内の電流をプロットするには、このチェックボックスをオンにする必要があります。

Don't save Ib(), Ie(), Is(), Ig():

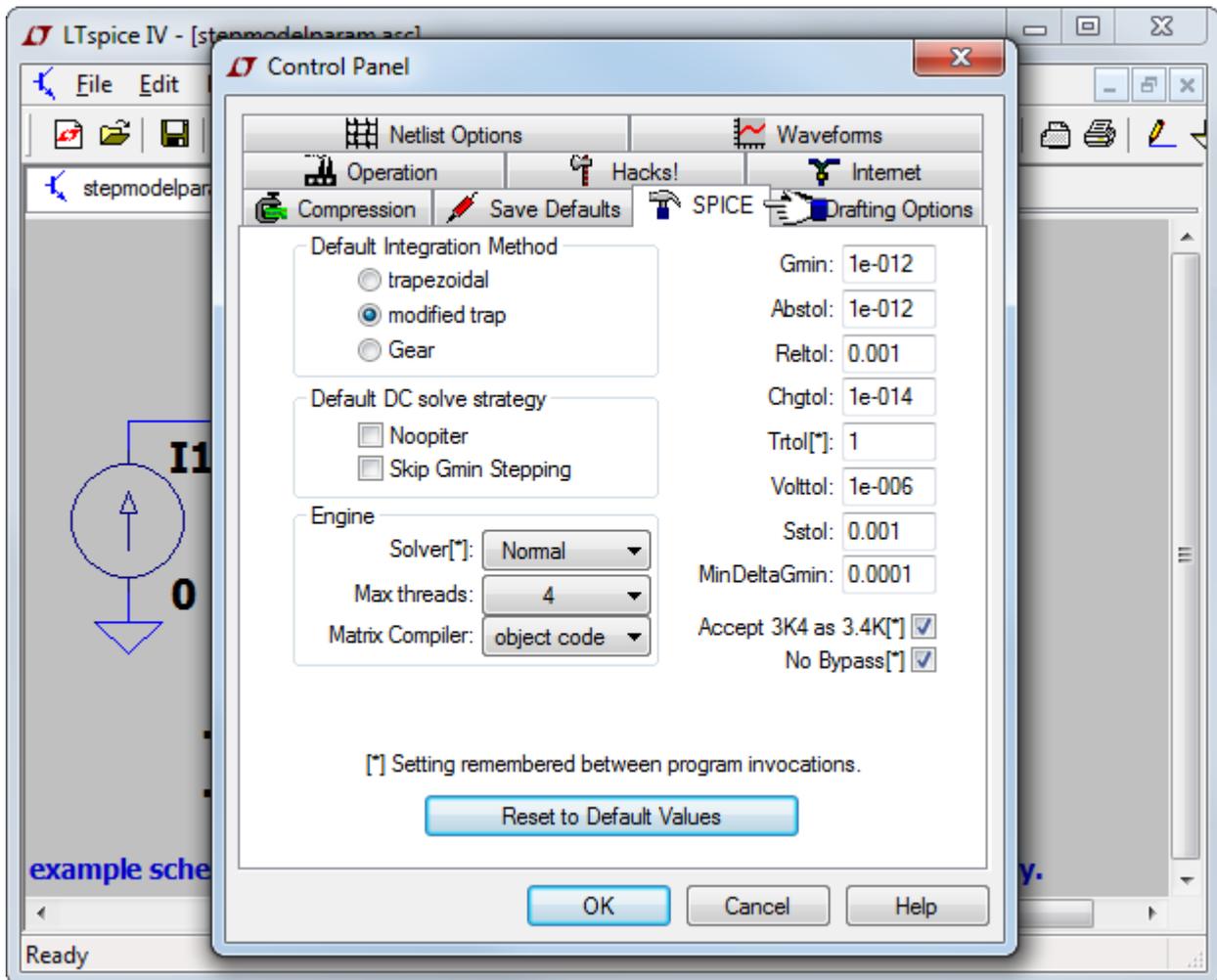
これは、出力.data ファイルのサイズを低減するため、トランジスタのコレクタ(ドレイン)電流だけを保存する設定です。これは IC 設計に役立ちますが、この設定を使用すると、トランジスタの電力損失を計算するためのデータが不十分になることを意味します。

Save Device Currents :

デバイス電流と端子電流をプロットできるようにするには、このチェックボックスをオンにします。また、電力損失をプロットできるようにする場合もオンにする必要があります。

SPICE

このペインでは、LTspice のさまざまなデフォルト設定を定義できます。これらのデフォルト設定を任意のシミュレーションで無効にするには、該当のシミュレーションでオプションを指定します。通常は、これらのオプションをそのままにしておいてかまいません。Web を介してプログラムを頻繁に更新している場合は、「Reset to Default Values」を押して、最新の推奨設定にリセットすることが必要な場合があります。



変更する必要があると考えられるデフォルト設定の 1 つは trtol です。大半の商用 SPICE プログラムでは、このデフォルト値は 7 です。LTspice ではこのデフォルト値は 1 になっており、SMPS マクロモデルを使用するシミュレーションで、波形にシミュレーション・アーティファクトを生じる可能性が低くなるようにしています。trtol では、シミュレーション精度に対する直接的な影響よりも時間刻みに対する影響のほうが大きくなります。トランジスタ・レベルのシミュレーションでは、1 より大きな値にするのが、通常は全体的に良好な解決策です。trtol を大きくすれば、シミュレーション精度に悪影響を及ぼさずに 2 倍の速度が得られることが分かります。trtol の設定値は、プログラムを複数回起動した場合も引き継がれます。ただし、従来の SPICE 許容差パラメータの大半 (gmin、abstol、reltol、chgtol、vntol) は、デフォルト値の使用を促すため、プログラムを複数回起動した場合は引き継がれません。デフォルト値以外の値を使用する場合は、使用する値を指定する.option 文を記述して回路図に配置するか、設定をファイル内に保存してそのファイルを組み込みます。

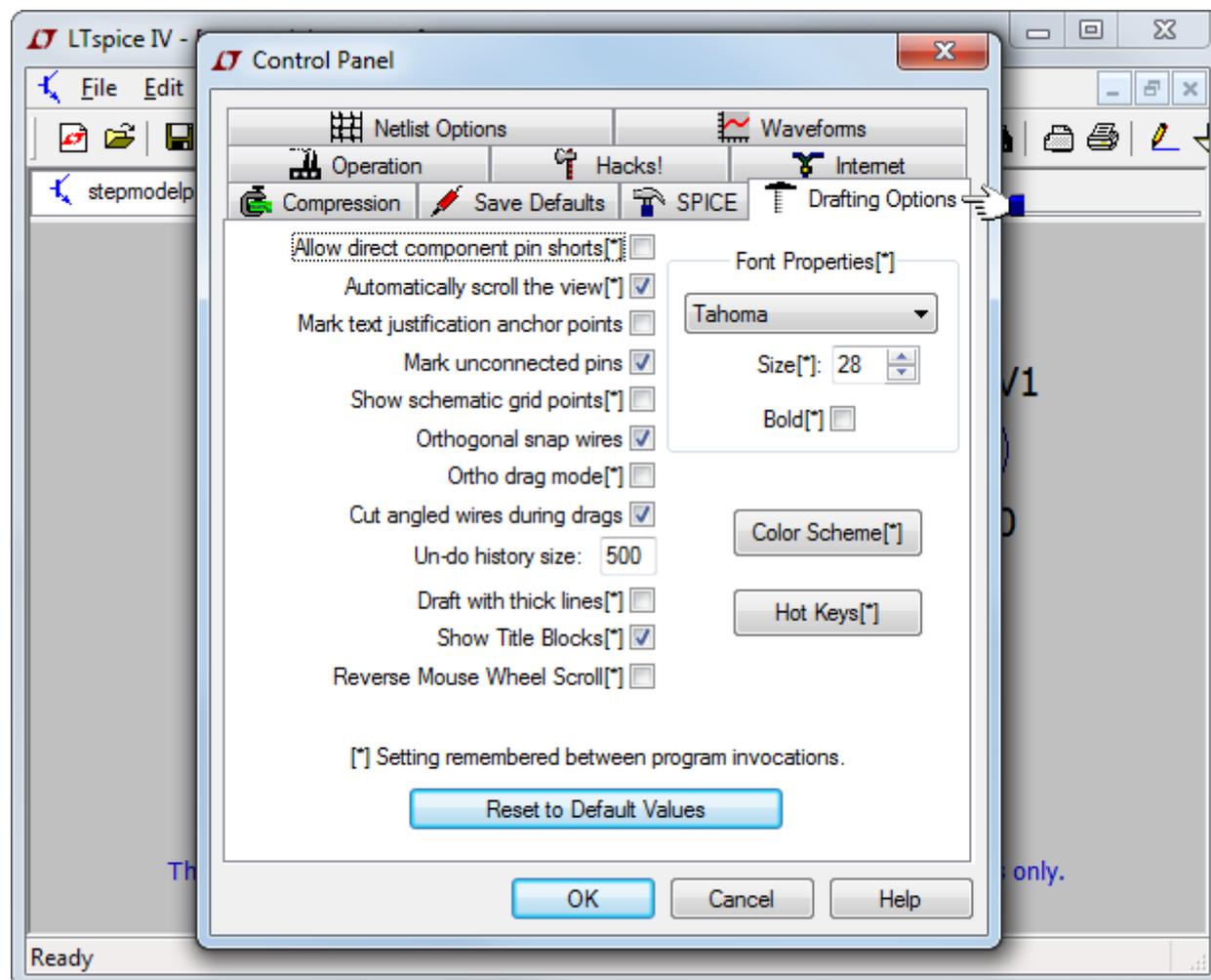
また、どのソルバが使用されるかも興味深い点です。LTspice は、SPICE の完全版を 2 種類内蔵しています。一方は標準ソルバと呼ばれ、もう一方は代替ソルバと呼ばれます。代替ソルバは、丸め誤差の少ない別の疎行列パッケージを使用しています。通常、代替ソルバのシミュレーション速度は標準ソルバの半分ですが、内部精度は 1000 倍高精度です。これは、役立つ診断プログラムとして利用できます。どのソルバを使用するかを指定する option 文はありません。2 つのソルバは異なるパーサーを使用するので、ネットリストが解析される前に選択しておく必要があります。

シミュレーションに使用するスレッドの最大数は、OS と CPU ハードウェアがサポートするスレッドを同時に実行する最大数に設定します。ある特定のシミュレーションで使用される実際の数値は、回路の性質により異なります。LTspice は、結局のところシミュレーションの実行速度を向上しない追加スレッドとは連携しません。

行列コンパイラは、デフォルトでオブジェクト・コードになります。これが意味するのは、LTspice は、回路を解くときに、回路に合わせて最適化されたアセンブリ言語の一覧を実行中に記述するということです。その後、通常の LU 因数分解コードの代わりに、このコードをアセンブルして、リンクを張り、実行します。通常のコードは、C、C++および手動でコード化したアセンブリを組み合わせて記述されていますが、回路に固有の性質についての知識なしで記述されています。

「Accept 3K4 as 3.4K」の横にあるチェックボックスをオンにして、4K99 と記述された数値が 4.99K に等しいと LTspice に強制的に解釈させます。通常の SPICE 手法ではこうすることができませんが、LTspice では、この要求が多かったので使用可能になっています。

Drafting Options



Allow direct component pin shorts:

通常は、部品を直接貫いて素線を描画することが可能であり、ピンを短絡する素線部分は削除されます。このチェックボックスをオンにすると、短絡素線が自動的に削除されなくなります。

Automatically scroll the view:

このチェックボックスをオンにすると、回路図の編集にマウスを動かしてエッジに近づけたときに、それに応じて回路図がスクロールされます。

Mark text Justification anchor points:

テキスト・ブロックの基準点を示す小さな円を描画します。

Mark unconnected pins:

個々の未接続ピンに小さな正方形を描画し、それらのピンが未接続であるというフラグを立てます。

Show schematic grid points:

可視グリッドを最初から有効にします。

Orthogonal snap wires:

描画時に素線を垂直部分と水平部分に強制的に描画します。このチェックボックスをオフにすると、素線を任意の角度で描画可能になり、任意のグリッドに位置を合わせることができます。Ctrl キーを押したままにすると、素線の描画中に電流の設定が瞬時に切り替わります。

Ortho drag mode:

ドラッグ時に素線を垂直部分と水平部分に強制的に描画します。このチェックボックスをオフにすると、素線を任意の角度でドラッグできます。Ctrl キーを押したままにすると、素線のドラッグ中に電流の設定が瞬時に切り替わります。

Cut angled wires during drags:

「Drag」コマンドの実行中、中間の素線に沿ってクリックすると、非直交素線は 2 本の接続素線に分解されます。

Undo history size:

取り消し/やり直しバッファのサイズを設定します。

Draft with thick lines:

すべての線幅を太くします。発行用の画像を生成するのに役立ちます。

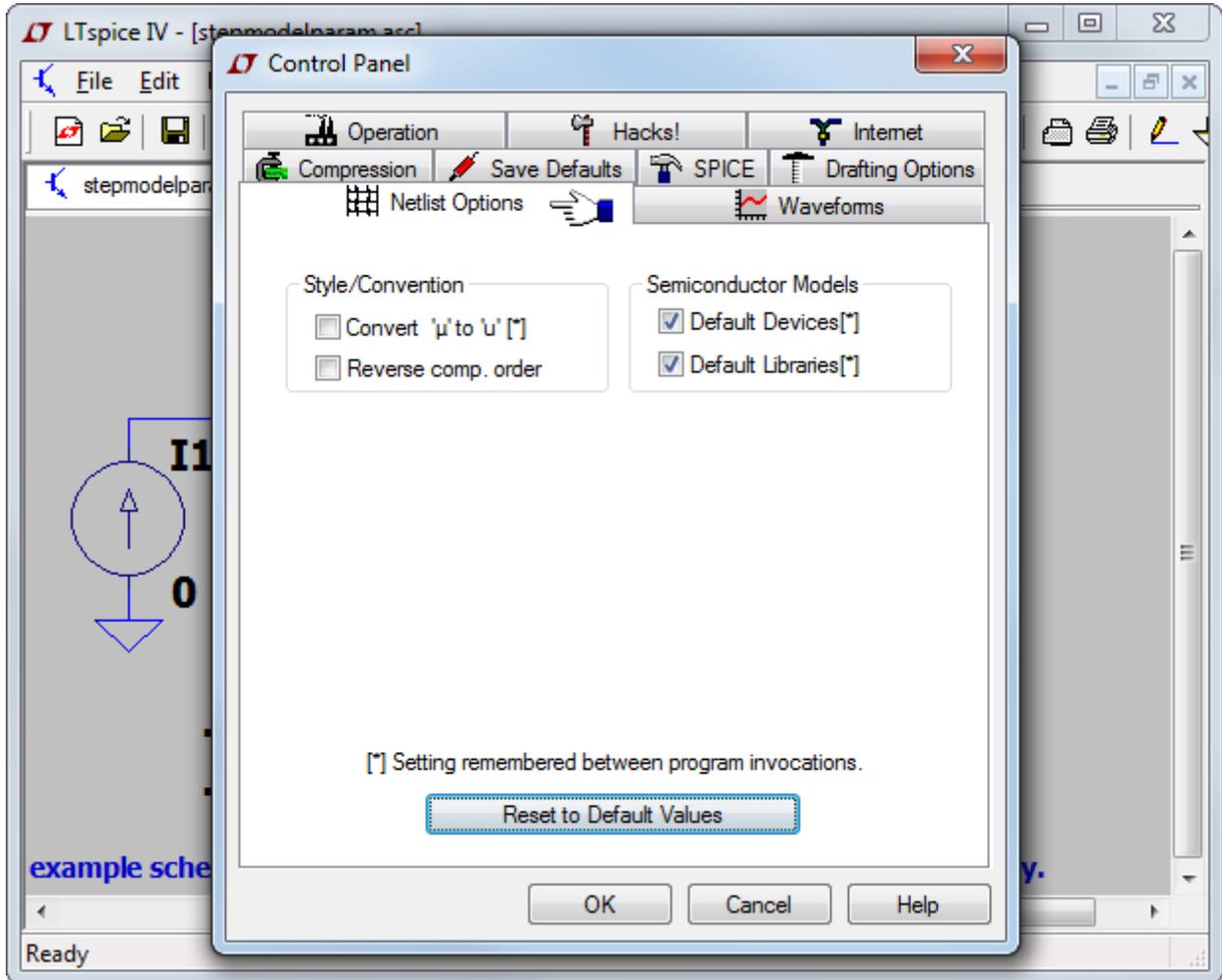
Show Title Block:

内部使用目的のオプション。

Reverse Mouse Wheel Scroll:

通常は、マウス・ホイールを自分の方に動かすと画面が近づく方向に表示されます。このオプションのチェックボックスをオンにした場合、マウスで同じ動作をすると頭が画面から遠ざかるように表示されます。

Netlist Options



Convert 'μ' to 'u':

「μ」のすべてのインスタンスを「u」に置き換えます。(i)ご使用の MS Windows インストール環境では、(たとえば、一部の中国語版 Windows にはデフォルトのフォントが付属していないので)ギリシャ文字の「ミュー」を表示できない場合、および(ii)「μ」という文字を 1e-6 を表すメートル乗数と解釈しないネットリストを SPICE シミュレータに対して生成する場合に役立ちます。

Reverse comp. order:

回路素子は、通常は回路図に加えられた順にネットリスト化されます。このチェックボックスをオンにすると、この順序が逆になります。

Default Devices:

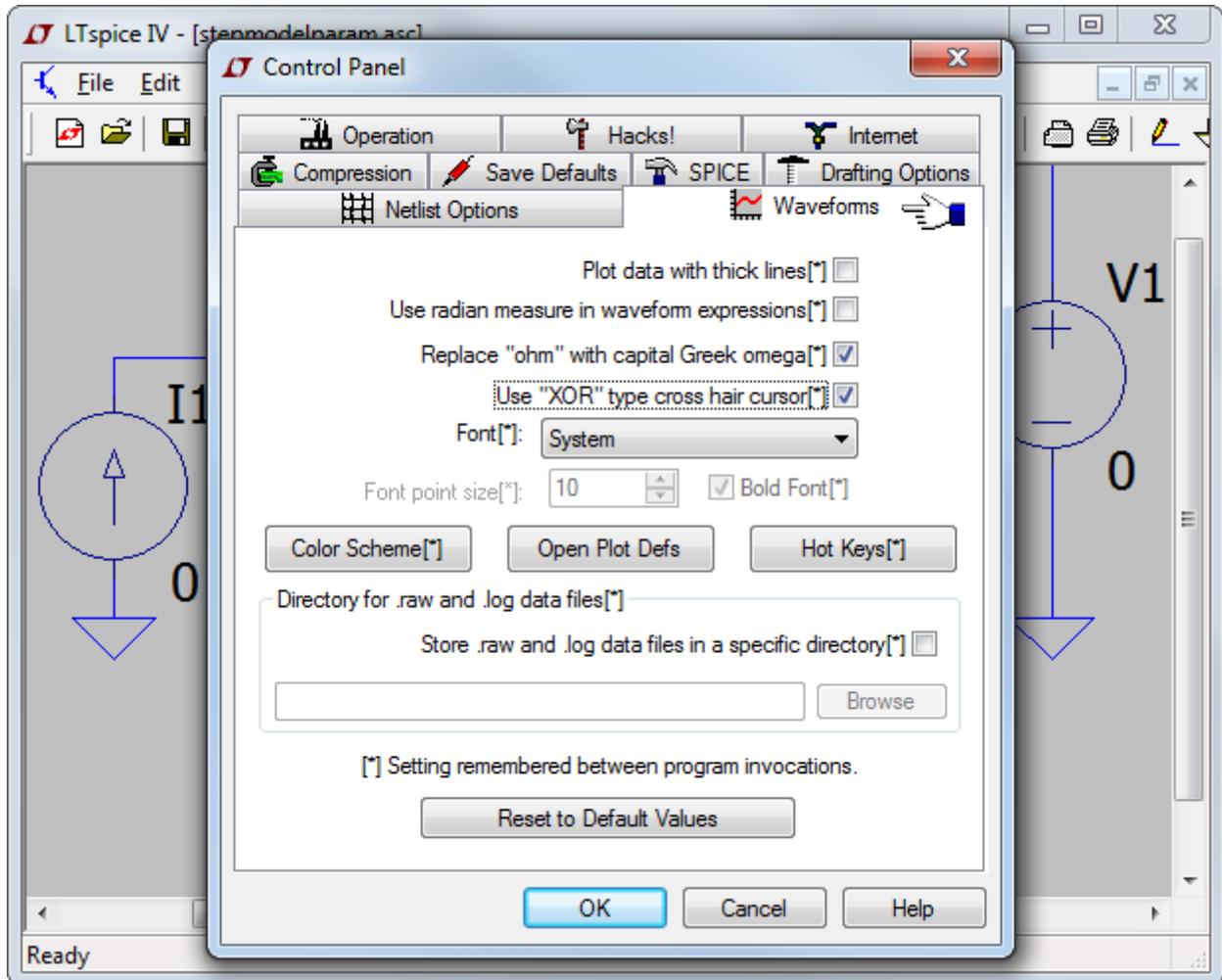
LTspice の回路図にダイオードを使用する場合は必ず、デフォルトの.model 文「.model D D」がネットリストに追加され、デフォルトのモデルの使用に関するメッセージが表示されないようになります。このオプションのチェックボックスをオフにすると、この行の組み込みだけでなく、バイポーラ・トランジスタ、MOSFET、および JFET の類似の.model 文の組み込みも行わない設定になります。

Default Libraries:

LTspice の回路図にダイオードを使用する場合は必ず、.lib 文によってデフォルトのライブラリ (standard.dio) がシミュレーションに組み込まれます。このオプションのチェックボックスをオフにすると、このライブラリの組み込みだけでなく、バイポーラ・トランジスタ、MOSFET、および JFET の類似の .lib 文の組み込みも行わない設定になります。

Waveforms

このペインでは、波形ビューワのいくつかの外観を設定できます。



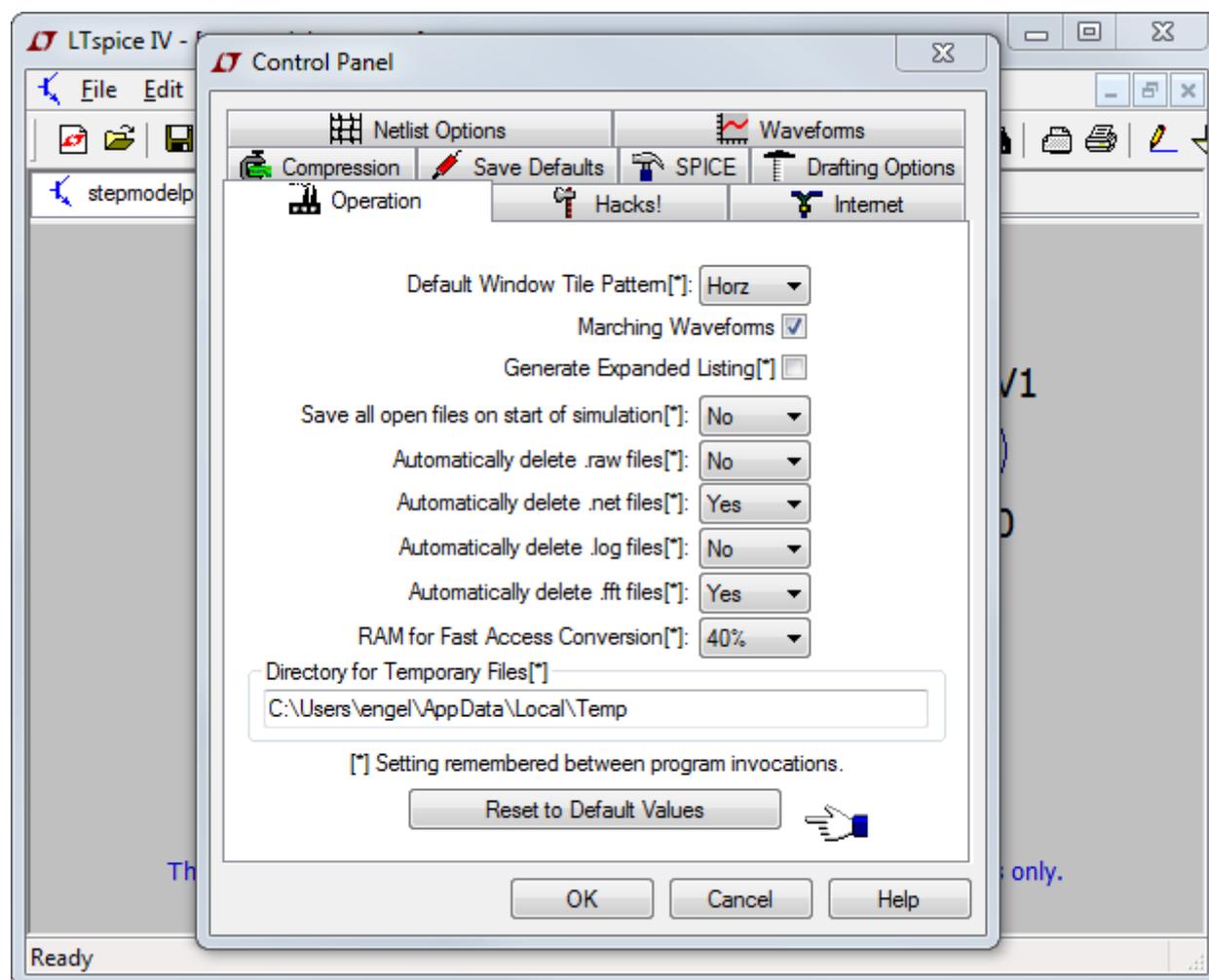
デフォルト設定からの変更を検討する可能性がある2つの設定は、次のとおりです。

- i) 「Use "XOR" type cross hair cursor」チェックボックスをオンにします。

XOR カーソルは、カーソルの背景画面の色が何であっても目に見える十字マークです。たとえば、背景が黒の場合、カーソルは白になり、逆の場合も同様です。ただし、このカーソルの機能は実際には XOR ではないことに注意してください。なぜなら、カーソルは灰色の背景に対しても相変わらず目立っていますが、灰色の背景で XOR をとった値がすべて反転ビットになっても、色は区別できるほど変わらないからです。「XOR」カーソルの使用は非常に望ましいですが、すべてのビデオ・ハードウェアおよびドライバがこれを正しく取得するとは限りません。したがって、LTspice は波形ウィンドウの背景に対するコントラストが高い不透明なカーソルをサポートしますが、プロット・データに対するコントラストは必ずしも高いわけではありません。一部のハードウェアは「XOR」カーソルを適切に実行できないので、このあまり望ましくないカーソルはインストール時のデフォルト設定です。

- ii) 波形画像を.wmf ファイルとして保存する場合は、「Font」を「System」以外に変更してください。
「System」フォントはビットマップ化されたフォントであり、デスクトップ・パブリッシングで使用される.wmf ファイルのベクター・グラフィック形式では拡大縮小できません。

Operation



アスタリスク[*]の付いた設定は、プログラムを複数回起動した場合も引き継がれます。

Default Window Tile Pattern:

2つのモニタを横に並べて使用する場合は、一方のモニタを回路図に使用し、もう一方のモニタを波形データに使用するように、このオプションを「Vert」に設定すると便利ことがあります。

Marching Waveforms:

このチェックボックスをオンにすると、シミュレーション中にシミュレーション結果を徐々にプロットできます。

Generate Expanded Listing:

サブ回路を拡張した後にフラットなネットリストを SPICE Error Log ファイルにダンプします。

Save all open files on start of simulation:

LTspice は、ディスク上の回路図ではなく、メモリ内の回路図をシミュレートします。このオプションを

設定すると、2つの回路図は、シミュレーションを開始するたびに同期状態を強制されます。

Automatically delete .raw files :

このオプションを設定すると、シミュレーションの終了後に、波形データ・ファイルを自動的に削除できます。こうすると、LTspice によって使用されるディスク容量が劇的に低減しますが、シミュレーションを再開する場合には、シミュレーションを再実行する必要があります。

Automatically delete .net files :

このオプションを設定すると、回路図を閉じたときは必ず、回路図のネットリストを自動的に削除できます。これらのファイルは小さな一時ファイルとみなすことができるので、これらのファイルを削除することにより、ディレクトリはより整然と探索されます。これらのファイルは、LTspice シミュレータに対して回路図の電氣的接続性を定義しています。ユーザによってはこれらのファイルをさらに活用するので、削除しないほうがいい場合があります。

Automatically delete .log files :

このオプションを設定すると、シミュレーションを終了したときは必ず、シミュレーション・ログを自動的に削除できます。これらのファイルには、シミュレーション中の経過時間などの各種のシミュレーション統計情報、警告メッセージ、エラー・メッセージ、および .step/.temp/.dc 解析に使用されるステップ・パラメータが含まれています。

Automatically delete .fft files :

このオプションを設定すると、シミュレーションの終了後に、波形データ・ファイルの FFT を自動的に削除できます。

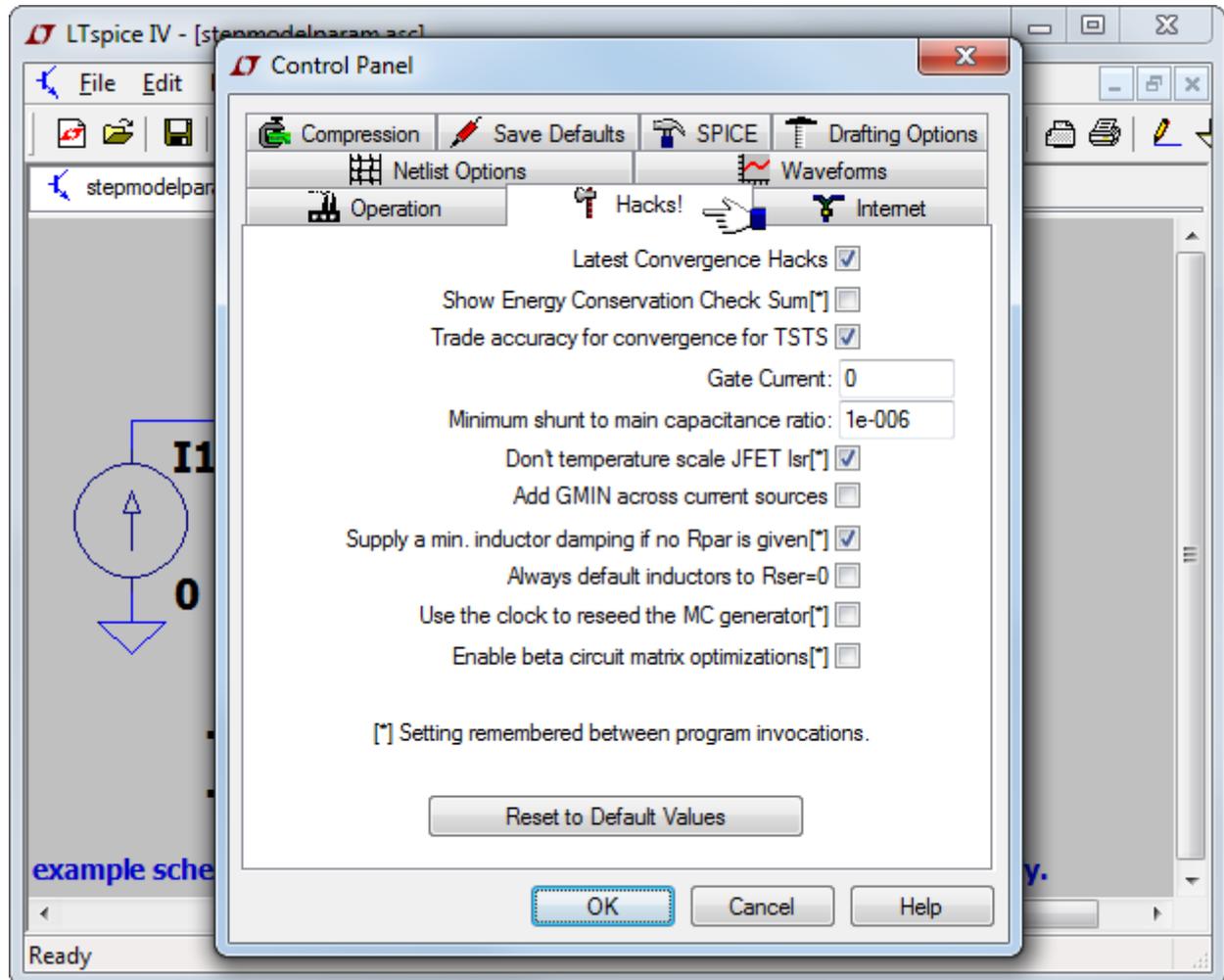
RAM for Fast Access Conversion :

このオプションを設定すると、波形データを Fast Access ファイル形式に変換するときにメモリ使用量を調整できます。

Directory for Temporary Files :

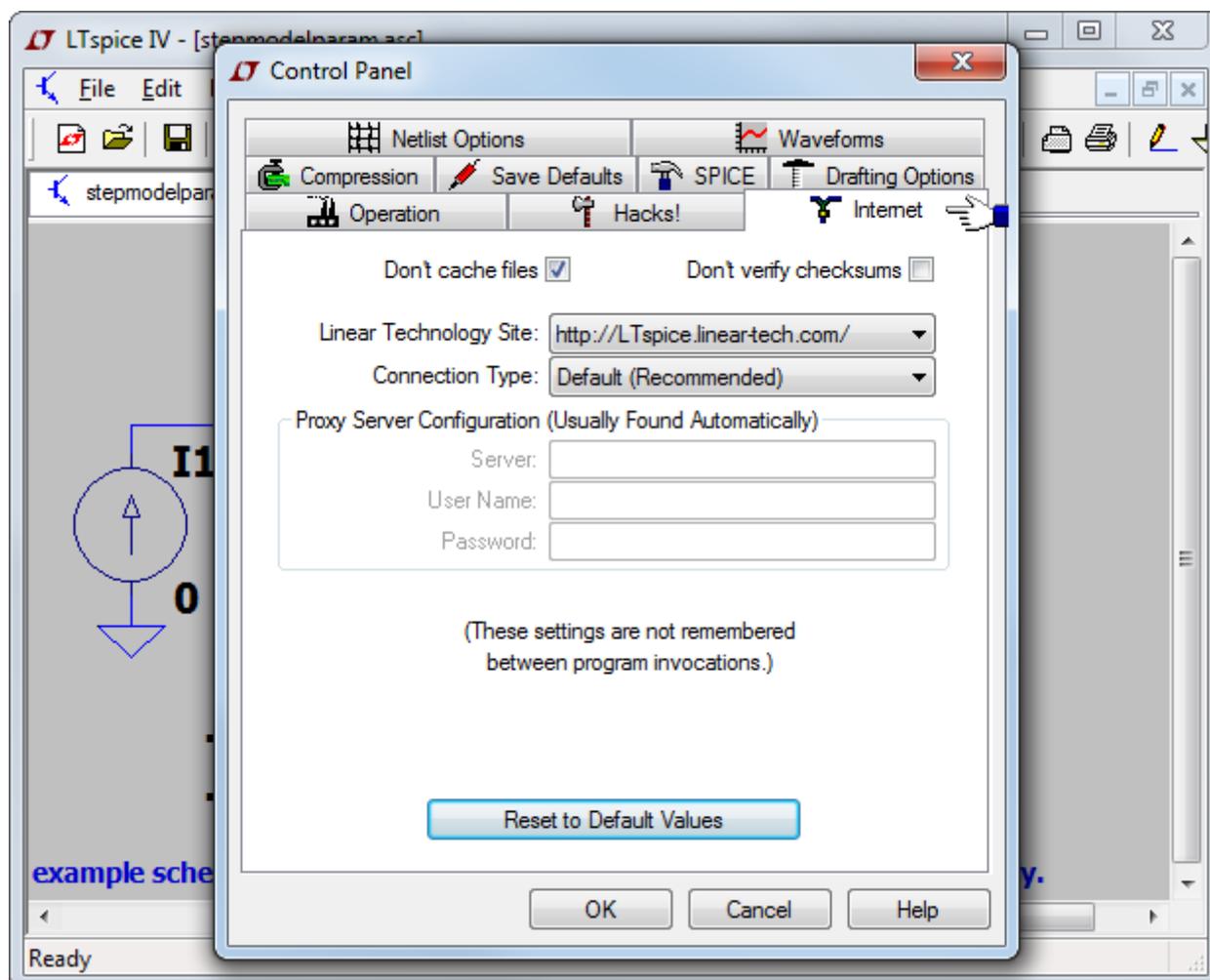
「Sync Release」の実行時にダウンロードした更新ファイルの一時記憶領域のディレクトリです。

Hacks



このペインは社内でのプログラム開発に使用されていましたが、現在はほとんど使用されていません。

Internet Options



「Control Panel」のこのペインは、Web で取得した増分更新データのために使用されます。LTspice は、新しい機能およびモデルで頻りに更新されます。最新バージョンに更新するには、メニューコマンド「Tools」=>「Sync Release」を使用します。数か月間更新しなかった場合、LTspice は更新データの有無について確認するかどうかを尋ね始めます。LTspice では、ユーザの許可を求めずに Web にアクセスすることはありません。LTspice にはスパイウェアは含まれていませんし、更新に必要なファイルを取得する間にどんな種類のデータも送信しません。

Don't cache files:

更新時に弊社マシンのキャッシュに格納されたファイルをキャッシュに格納することも使用することもしません。

Don't verify checksums:

セキュリティ上の理由により、LTspice は、独自で社外秘の 128 ビット・チェックサム・アルゴリズムを使用して、更新のために Web から受信するファイルを認証します。そのアルゴリズムに誤りがあつ

た場合には、この認証が無効になることがあります。ただし、これまでにこの問題が報告されたことはないので、このセキュリティ機能を常に無効にすることは推奨しません。

LTspice では、そのインターネット・アクセスに上位のオペレーティング・システム呼び出しだけを使用します。プロキシ・サーバーとパスワードを指定する必要があるというまれな場合以外には、これらの設定を調整する必要はありません。インターネット・アクセスを管理しているのは LTspice ではなく、ご使用のコンピュータとオペレーティング・システムであるからです。このペインでの設定は、プログラムを複数回起動した場合は引き継がれません。

よくある質問(FAQ):

[インストールの問題](#)

[プログラムの更新](#)

[トランスのシミュレーション](#)

[サードパーティ・モデル](#)

[インダクタ・モデル](#)

[MOSFET モデル](#)

[ライセンスと配布](#)

[効率計算](#)

[カスタム・シンボル](#)

[メモリの問題](#)

[モデルの互換性](#)

[SPICE ネットリスト](#)

[波形データのエクスポート](#)

[SMPS ボーデ線図](#)

[Linux 環境での動作](#)

[紙のマニュアル](#)

[ユーザグループ](#)

インストールの問題

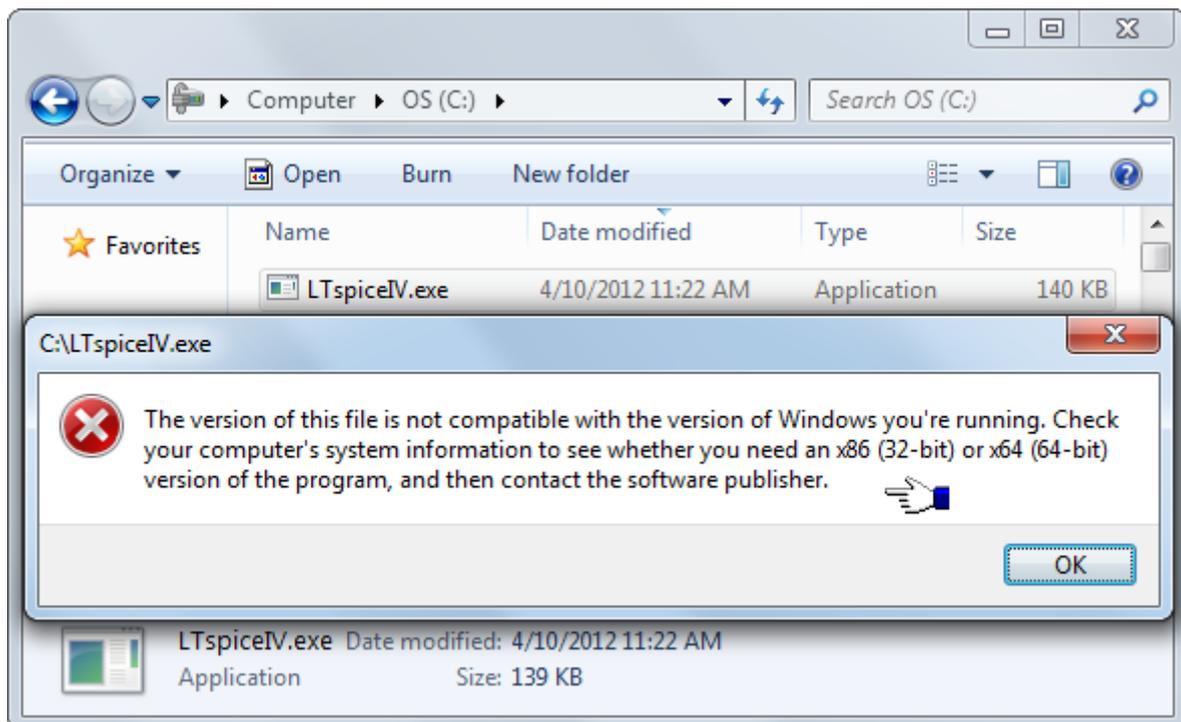
Q LTspice IV のインストール方法を教えてください。

1. <http://www.linear-tech.co.jp> にアクセスし、LTspiceIV.exe ファイルを PC の一時ディレクトリにダウンロードします。
2. インストール対象の LTspiceIV.exe ファイルを実行します。
Windows Vista または Windows 7 では、管理者として実行することになるでしょう。

Q 中国語版の Windows を実行しています。ギリシャ文字の「ミュー」が正しく表示されません。どうすればいいですか。

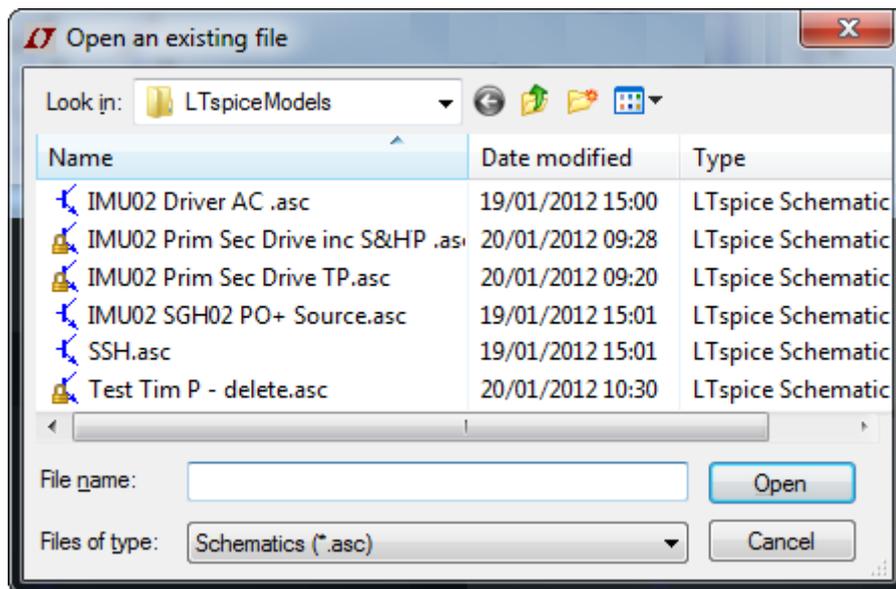
その問題は LTspice IV の最新バージョンで修正する必要があります。ただし、「Tools」=>「Control Panel」=>「Netlist Options」の順にメニュー項目を進めて、「Convert 'μ' to 'u」チェックボックスをオンにすることができます。このオプションはネットリストに適用されるだけでなく、ギリシャ文字の「ミュー」が画面に表示される場合にも必ず適用され、「u」と描画されるようになりました。

Q LTspice インストール・プログラムをダウンロードしましたが、使用中の Windows のバージョンとは互換性がないと表示されます。下記のエラー・メッセージを参照してください。LTspice は Windows のすべてのバージョンで動作すると聞いていました。Windows 7 Pro x64 で動作するプログラムを入手できますか。



インストール・ファイル(LTspiceIV.exe)がダウンロード中に破損しています。この状況が発生する可能性があるのは、低速の接続経路でインストール・ファイルをダウンロードしている間に(Webサイトのメンテナンスにより)サーバのファイルが改訂される場合です。ファイルを再度ダウンロードする必要があります。Windows 7 x64は、現在 LTspice IV の最も一般的なプラットフォームです。

Q 回路図に小さなロック・アイコンがあるのはなぜですか。



「ユーザ アカウント制御(UAC)」をオフにするのを忘れていました。これは Windows 7 の問題です。

プログラムの更新

Q 最新バージョンの入手方法を教えてください。

いったんインストールしたら、最新バージョンの入手方法は2つあります。[インストールの問題](#)で説明したように、プログラムはいつでも再インストールすることができます。インストーラでは、好みの設定が失われないように、オプションでフル・インストールの代わりに更新を実行できます。インストール前に古いバージョンを削除する必要はありません。最新のリリースを入手する別の方法として、単純に [Sync Release\(*update 参照\)](#) 機能を使用する方法があります。

Q 追加された新機能の内容を知る方法を教えてください。

ファイルを最新バージョンに更新した後は、ルート・ディレクトリ内にある「changelog.txt」ファイル(通常は C:\Program Files\LTC\LTspiceIV\Changelog.txt)に、詳細なプログラム改訂リストが入っています。

Q Sync_Release の実行後に古いバージョンに戻ることはできますか。

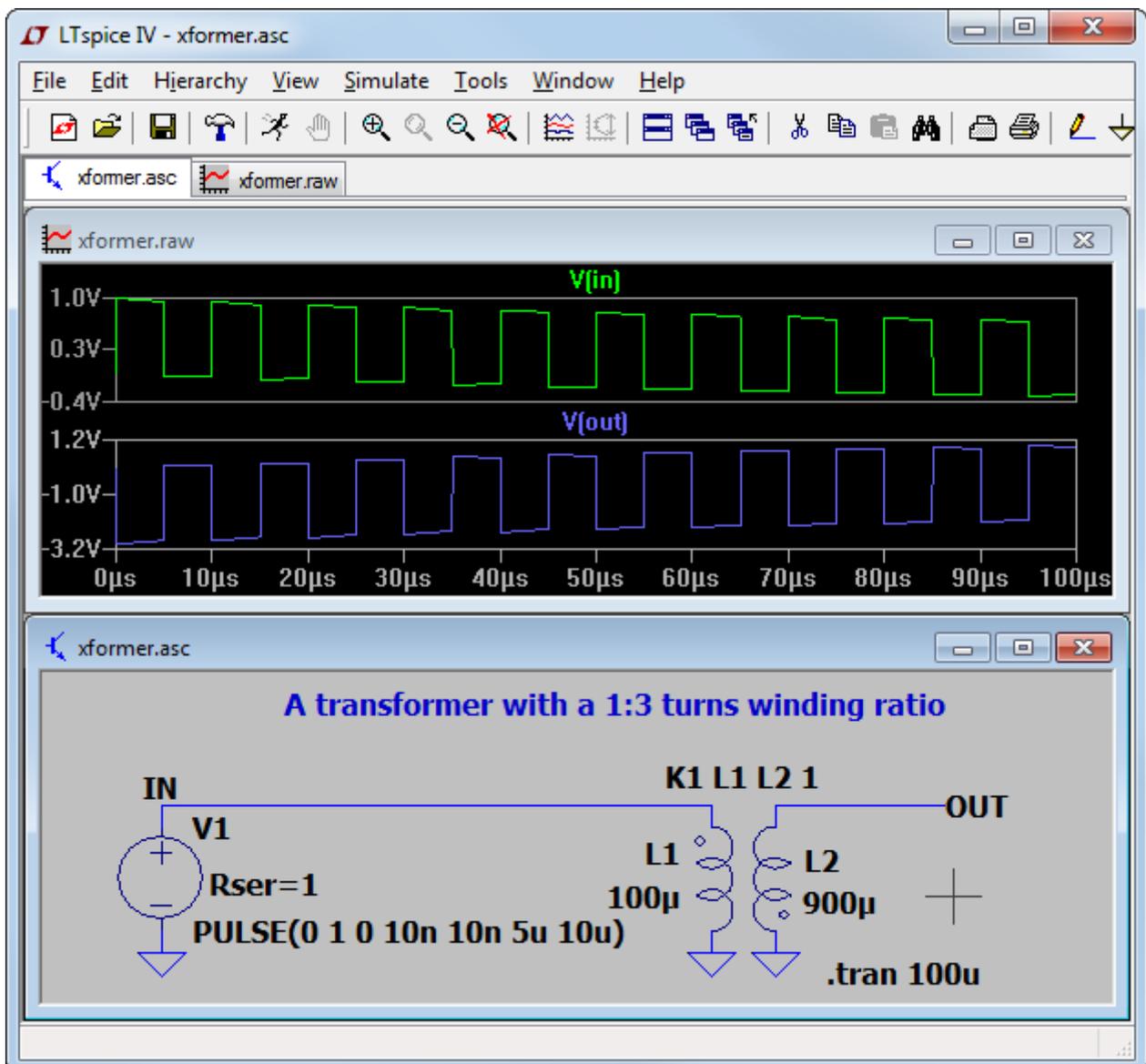
いいえ。シンボル、モデル、およびプログラムはすべて最新バージョンに更新されます。部品データベース(standard.*)は新しいデータベースと自動的にマージされます。新しいインダクタまたはコンデンサを追加した場合、それまでに作成したデバイスは保護され、更新により新しいデバイスとマージされます。

トランスのシミュレーション

Q トランスのシミュレーション方法を教えてください。

結合したインダクタを単独のインダクタとして下書きし、次に、SPICE 指令として回路図上に配置された相互インダクタンス文を追加します。詳細については、相互インダクタンスのセクションを参照してください。相互インダクタンスに関するインダクタは、位相合わせのドットを付けて描画されます。

次の例では、巻数比が 1:3 (インダクタンス比は 1:9) のトランスに正弦波を入力した実例を示して 100 μ s の間シミュレートします。相互結合係数を 1 に設定して、漏れインダクタンスのないトランスをモデル化しています。



Q いや、それでは簡単すぎます。もっと複雑なものはありませんか。

もちろんあります。

- i) 直流の LCR メーターで巻線の L を測定します。これらの値をシミュレーションで直接使用します。
- ii) 巻線の ESR をオーム計で測定します。抵抗は周波数が高いほど DC での値より大きくなるので、シミュレーションでは測定値の 2 倍の値を使用します。経験上、フェライトとスイッチャを使用した場合では、通常は 2 倍近くになります。
- iii) 最も誘導性の高い巻線以外すべての巻線を短絡し、直流の LCR メーターで漏れインダクタンスを測定します。結合係数を調整してこれと一致させます。あるいは、2 本の巻線の場合は以下のようにします。

$$K = \sqrt{1 - L_{\text{leak}} / \sqrt{L_1 * L_2}}$$

$$L_{\text{leak}} = \sqrt{L_1 * L_2} * (1 - K * K)$$
- iv) トランスの共振周波数および Q を求めます。この共振周波数と一致する最も誘導性の高い巻線の Cpar および Rpar を指定します。
- v) シミュレーションを楽しんでください。

Q 非線形のトランスはどうですか。

デフォルトの場所 C:\Program Files (x86)\LTC\LTspiceIV\examples\Educational\NonLinearTransformer.asc にインストールされている例を参照してください。

サードパーティ・モデル

このセクションでは、サードパーティ・モデルを LTspice IV に追加する場合の基本について説明します。基本的には、2 種類のサードパーティ SPICE モデルがあります。.MODEL 文で記述されたモデルと、.SUBCKT で定義されたモデルです。

.MODEL 文として与えられたモデルは、ダイオードやトランジスタのような固有 SPICE デバイスが対象です。 .MODEL 文では、特定の部品に対してパラメータが指定されます。SPICE はデバイスの動作を既に認識しているため、パラメータを指定する必要があるのは、部品の電気的特性の指定を終了するためだけです。

その一方で、.SUBCKT 文によって与えられるモデルは、モデル化された部品を固有 SPICE デバイスの一連の回路によって定義します。たとえば、オペアンプの SPICE モデルはサブ回路として与えられます。

LTspice にモデルを組み込む方法は、モデルを.MODEL 文または.SUBCKT のどちらとして与えるかによって異なります。

.MODEL 文で定義された NPN トランジスタの例を以下に示します。

1. シンボル NPN のインスタンスを回路図に追加します。
2. 値「NPN」を「BC547C」に編集して、目的の.MODEL 文で使用する名前と一致させます。
3. ここで、以下のいずれかを行います。
 - 3a) .MODEL BC547C...文を SPICE 指令として回路図に追加します。

または、

- 3b) .MODEL BC547C...が収容されている bipolar.lib ファイルがある場合は(他のモデルもこのファイル内にある可能性あり)、SPICE 指令「.INCLUDE bipolar.lib」を回路図に追加します。「bipolar.lib」は任意のファイル拡張子を持つ完全な名前にする必要があります。Windows エクスプローラはデフォルトではファイル拡張子を表示しないことに注意してください。したがって、メモ帳で編集/表示できる「bipolar.lib.txt」と呼ばれるファイルがあり、Windows エクスプローラでこのファイルが「bipolar.lib」として存在すると表示される場合、このファイルを組み込む SPICE 指令は「.inc bipolar.lib.txt」です。「.inc bipolar.lib」を使用した場合は、ファイルが見つからないというエラー・メッセージが表示されます。

または、

- 3c) 標準では C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt としてインストールされているファイルに.MODEL BC547C...文を追加できます。この文を追加すると、対象となる NPN トランジスタのモデルが自動的に表示されます。この standard.bjt ファイルを LTspice の外部で編集した場合は、LTspice を再起動して、ファイルが変更されたことを認識させる必要があります。

5 ピンのオペアンプの例これは以下のようにして.SUBCKT 文で定義します。

1. シンボル opamp2 のインスタンスを回路図に追加します。
2. 回路図で値「opamp2」を「TL072」に編集して、.SUBCKT.の名前と一致させます。
3. 以下のいずれかを行います。

- 3a) 「.SUBCKT TL072ENDS」定義を 1 つの複数行 SPICE 指令として回路図に貼り付けます。

または、

- 3b) (「.SUBCKT TL072...」で始まる行のような)サブ回路 TL072 の定義が収容されている「TI.lib」と呼ばれるファイルがある場合は、SPICE 指令「.INCLUDE TI.lib」を回路図に追加します。新しいシンボルを作成し、そのシンボルを回路図上で使用するときは必ず、必要なモデルをシミュレーションに自動的に組み込むようプログラムすることができます。ヘルプのセクション[「回路図の取り込み」=>「新しいシンボルの作成」](#)を参照してください。

任意のサードパーティ・モデルに照らして正しくネットリスト化される新しいシンボルを自動的に生成し、そのシンボルが回路図に現われる場合は必ず、必要なモデルをシミュレーションに組み込むように、シンボルをプログラムすることができます。ヘルプのセクション[「回路図の取り込み」=>「新しいシンボルの作成」](#)を参照してください。サブ回路として定義された新しいモデルを追加する場合、ほとんどのユーザにとっては、これが検討に値する唯一の方法です。詳細が漏れなく処理されるからです。

.SUBCKT 文で定義された 3 ピン NPN トランジスタの例を以下に示します。

- 1) シンボル NPN のインスタンスを回路図に追加します。
- 2) 新たに配置された NPN シンボル・インスタンスの本体の上にカーソルを移動します。<Ctrl>キーを押したまま右マウス・ボタンをクリックします。ダイアログボックスが表示されます。「Prefix:QN」を「Prefix:X」に変更します。これにより、このシンボルのインスタンスは、固有のバイポーラ・トランジスタの代わりにサブ回路としてネットリスト化されます。

3) 値「NPN」を「BFG135」に編集して、.SUBCKT 行に指定されている名前と一致させます。

4) その後、以下のいずれかを行います。

4a) .SUBCKT BFG135 行を回路図に追加します。

または、

4b) .SUBCKT BFG135が含まれているファイル Phil.lib がある場合は(他のモデルもこのファイル内にある可能性があります)、SPICE 指令 INCLUDE Phil.lib を追加する必要があります。

.SUBCKT モデルを LTspice に追加する場合は、シンボルを使用してサブ回路を呼び出し、モデルが同じピン/ポート・ネットリスト順序に一致する必要があります。上記の例では、追加するサードパーティ・モデルが一般的なピン配列の規則に従うと仮定しています。

詳細な関連情報は、ヘルプ・セクション「[回路図の取り込み](#)」および「[LTspice](#)」にあります。シミュレータ(LTspice)が読み込むネットリストを回路図取り込みプログラムが生成するというのが基本的な考え方です。サードパーティ・モデルのインポートに関するどの局面も、SPICE ネットリスト構文を理解し、回路図取り込みプログラムがその構文をどのように生成するかを理解することで解決できます。このトピックに関して準備されたチュートリアルもあり、独立したユーザグループ(<http://tech.groups.yahoo.com/group/LTspice>)でアーカイブに保管されています。

インダクタ・モデル

Q 結合インダクタの設計方法を教えてください。

まず、(i) インダクタを 2 つ以上描画し、次に(ii) 2 つのインダクタ間の係数 K を定義します。詳細については、[相互インダクタンスのセクション](#)を参照してください。

Q インダクタの寄生抵抗を制御する方法を教えてください。

デフォルトでは、LTspice はインダクタに損失を与えて SMPS トランジェント解析を支援します。SMPS では、これらの損失は通常は重要ではありませんが、必要に応じてオフにすることができます。「Tools」=>「Control Panel」=>「Hacks!」ページで、「Supply a min. inductor damping if no Rpar is given」のチェックボックスをオフにします。この設定はプログラムを複数回起動した場合も引き継がれます。相互インダクタンスの文では言及されていないインダクタには、 $1\text{m}\Omega$ のデフォルト直列抵抗もあります。この R_{ser} を使用すると、LTspice IV では、回路の線形化マトリクスのサイズを低減するために、テブナン等価回路の代わりにノートン等価回路としてインダクタンスを積分することができます。LTspice がこの最小抵抗を導入しないようにする場合は、対象のインダクタに対して明示的に $R_{ser}=0$ を設定する必要があります。こうすると、LTspice はより煩雑なインダクタのテブナン等価回路をトランジェント解析時に使用することが必要になります。

Q 独自のインダクタ・モデルを追加/編集できますか。

標準では `C:\Program Files\LTC\LTspiceIV\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・モデルを追加または編集します。

MOSFET モデル

Q LTspice IV の MOSFET モデルと標準 SPICE の MOSFET モデルの違いは何ですか。

LTspice IV では、標準 SPICE の MOSFET モデルの他に、他の SPICE プログラムには実装されていない独自の MOSFET モデルも組み込んでいます。LTspice IV では、縦型二重拡散 MOS トランジスタの充電動作を直接カプセル化しています。これにより、他の SPICE プログラムの場合と同様に、サブ回路の代わりに LTspice 固有の VDMOS デバイスを使用してパワー・デバイスをモデル化できます。詳細については、[「MOSFET」のセクション](#)を参照してください。

Q 独自の MOSFET モデルを追加できますか。

はい。ファイル C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.mos に独自のモデルを追加できます。このファイルは.model 文で定義したデバイスだけが対象であり、サブ回路として定義したデバイスは対象外です。サブ回路を使用する場合は、以下の手順に従ってください。

1. シンボルの部品インスタンスの「Prefix」属性を「X」に変更します。シンボルは変更しないでください。回路図上の部品としてシンボルのインスタンスだけを変更します。Ctrl キーを押したまま部品の本体を右クリックすれば、この属性にアクセスできます。
2. 部品の「Value」属性を編集して、使用するサブ回路の名前と一致させます。

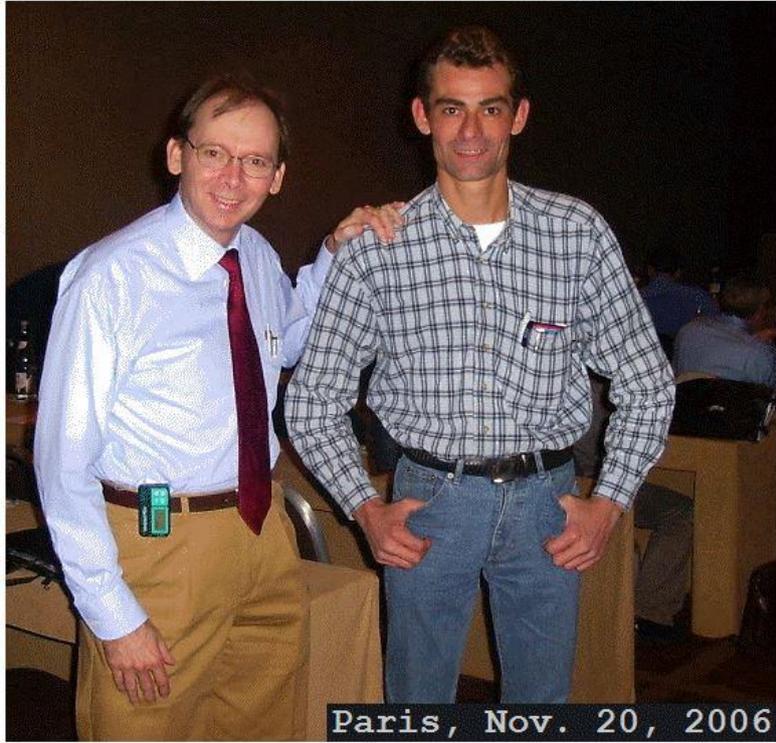
「.inc filename」などの SPICE 指令を回路図に追加します。ここで、filename はサブ回路の定義が含まれているファイルの名前です。これは任意のファイル拡張子を持つ完全な名前にする必要があります。Windows エクスプローラはデフォルトではファイル拡張子を表示しないことに注意してください。したがって、メモ帳で編集/表示できる「mylib.sub.txt」と呼ばれるファイルがあり、Windows エクスプローラでこのファイルが「mylib.sub」として存在すると表示される場合、このファイルを組み込む SPICE 指令は「.inc mylib.sub.txt」です。「.inc mylib.sub」を使用した場合は、ファイルが見つからないというエラー・メッセージが表示されます。

Q サブ回路の代わりに固有の LTspice VDMOS MOSFET モデルを生成するツールはありますか。

はい。Hendrik Jan Zwerver は、独立したユーザグループ <http://tech.groups.yahoo.com/group/LTspice> の「Files」セクションで配布されている無料の VDMOS ツールを開発しました。

Q Hendrik Jan Zwerver とは誰ですか。

写真の右側の男性です。



ライセンスと配布

Q このソフトウェアは再配布してもかまいませんか。

はい。リニアテクノロジーのお客様であるかどうかに関係なく、このソフトウェアは無償で配布できます。詳細については、[「使用許諾契約/免責事項」のセクション](#)を参照してください。

Q このソフトウェアは、シェアウェア、フリーウェア、デモのうちどれですか。

このプログラムは、シェアウェアでもデモでもありません。完全に機能するフリーウェアです。このソフトウェアの目的は、弊社のお客様が弊社製品を使用する際の支援を行うことです。また、回路図取り込み機能と SPICE シミュレーション機能を備えた汎用の回路設計パッケージとして使用することもできます。弊社では、学生のみなさんがこのプログラムを使用してアナログ設計工程に精通することを奨励しています。弊社では、リニアテクノロジー以外のことに関連したプログラムの使用法に対するサポートは保証できませんが、一般的なプログラム・バグはすべて修正いたしますし、そのようなご連絡には感謝いたします。弊社は広範囲な社内テストを実施しており、このプログラムには優れた収束機能があると信じています。既知の未解決バグはありません。

Q リニアテクノロジーの誰に連絡すれば支援を得ることができますか。

ソフトウェアに関するすべての問題については、LTspice@linear.com に電子メールを送信してください。

リニアテクノロジー製 IC に関する追加のアプリケーション情報など、ハードウェアに関するすべての問題については、通常の営業時間内に(408)954-8400 でリニアテクノロジーのアプリケーション部門にお電話ください。

回路の効率計算

Q 回路の効率レポートを作成するにはどうしたらいいですか。

.tran コマンドにキーワード「steady」を追加する必要があります(例:「.TRAN <time> steady」)。プログラムはスイッチャ・マクロモデルの内部状態を検査することにより、定常状態を検出します。定常状態検出機能はモデル内に実装されているので、スイッチング・レギュレータ部分が回路の一部でない場合には機能しません。スイッチング・サイクルの間に積分されるので、通常はエラーアンプから流れ出る電流が 0 まで低下するのを調べることで検出します。回路内の電圧源は厳密に 1 つである必要があります。これは入力と識別されます。回路内の電流源は厳密に 1 つである必要がありますが、代わりに Rload という名前の抵抗を使用してもかまいません。これは負荷と識別されます。シミュレーションが完了したら、「View」メニューの「Efficiency Report」を選択して、回路図に関するレポートを参照できます。

カスタム・シンボル

Q 独自のシンボルを作成できますか。

はい、独自のシンボルを作成できます。

Q 独自のシンボルの作成方法を教えてください。

メニューコマンド「File」=>「New Symbol」から始めます。

Q 独自のスイッチング・レギュレータ・モデルを作成できますか。

あまり簡単ではありません。LTspice IV に付属のスイッチング・レギュレータ・モデルは、新しいハードウェア記述言語および新しい固有 SPICE デバイスを使用して、LTC のスイッチング・レギュレータ製品の動作をカプセル化するように設計されています。標準の SPICE 基本要素を使用してモデルの作成に成功したとしても、シミュレーションの実行速度は数桁低下します。ただし、一部のユーザは標準の SPICE デバイスを使用してそのようなスイッチング・レギュレータ・モデルを作成しています。LTspice はこれらのモデルを実行することが可能であり、対象としたシミュレータより通常は性能が優れています。

メモリの問題

Q プログラムを実行するために必要なメモリの量を教えてください。

Windows を実行できれば、LTspice IV を実行できます。弊社はこのプログラムの所要メモリ量を最小限に抑えるために多大な労力を費やしました。メモリ・リークはありません。ただし、波形データにはメモリが必要なので、そこで問題が発生します。x64 OS はこの点で最良の選択です。

Q 波形はシミュレーション中どこに保管されますか。

波形データはすべてディスクに保管されます。プロットされたトレースだけは、RAM に読み込まれます。行進型波形をオフにすると、必要なメモリ量を低減できます。ほとんどの種類の解析では、ファイル・サイズの制限は特になくことに注意してください。GB 数が非常に大きいサイズの.raw ファイルを生成して表示できません。

Q 長時間のシミュレーション時に十分なディスク容量がない場合はどうしたらいいですか。

波形データは圧縮されていますが、やはり実行時間と保存されたトレース数に比例します。メモリを節約する最も簡単な方法は、シミュレーションを開始する前に、目的のトレースだけを保管の対象として選択することです。やるべきことはすべて実行しましたが、それでもメモリを使い果たしそうです。

Q では、どうすればいいですか。

トランジェント解析中に、「0」キーを押すことにより過去の波形を対話的に破棄することができます。そうすると、シミュレーション時間はもう一度現在時刻として $t=0$ になります。

モデルの互換性

Q スイッチング・レギュレータ・モデルは PSpice モデルやその他のモデルと互換性がありますか。

LTspice SMPS マクロモデルは、新しい独自の固有 LTspice デバイスと独自のハードウェア記述言語を組み合わせで実装されています。汎用の SPICE または PSpice マクロモデルを開発することは、原則としては可能ですが、得られるシミュレーション速度は実用的ではありません。ただし、LTspice は PSpice 半導体モデルおよび挙動モデルを実行可能であり、通常はかなり性能の高いシミュレータなので、Pspice シミュレーションを LTspice に移すことが考えられます。多くのユーザが PSpice から LTspice にアップグレードします。

SPICE ネットリスト

Q SPICE ネットリストの作成方法を教えてください。

ネットリストは、ASCII ファイルを生成可能なすべてのテキスト・エディタで作成できます。コマンド「View」=>「SPICE netlist」を使用して、任意の回路図の SPICE ネットリストを LTspice IV で表示できます。このビューで、すべてのテキストを選択し、Ctrl キーを押しながら C を入力すれば、クリップボードにネットリストをコピーして、ネットリストを別のエディタに移動できます。

Q ネットリストの実行方法を教えてください。

テキスト・ファイルを単に開き、実行します。ファイル拡張子が「. cir」である場合、LTspice IV はファイルをネットリストと認識します。

波形データのエクスポート/マーキング

Q 波形データは他のアプリケーションにエクスポートできますか。

波形ウィンドウをアクティブ・ウィンドウにして、Ctrl キーを押しながら C を入力すれば、プロットをビットマップとしてコピーできます。次に、Word やペイントのように、クリップボードからのビットマップの貼り付けを受け付けるアプリケーションで、Ctrl キーを押しながら V を入力します。なお、この方法は回路図のビットマップについても機能します。これらの画像も(メニューコマンド「Tools」=>「Write to a .wmf file」により)Windows メタファイルとしてエクスポートできます。このコマンドでは、さまざまなデスクトップ・パブリッシング・ツールでインポートできる.wmf ファイルにベクター・グラフィックスとして画像が書き込まれます。波形データのメタファイルをエクスポートする場合は、まず「Tools」=>「Control Panel」=>「Waveform」=>「Font」に移動して、「Arial」を選択します。デフォルトの「System」は、ディスプレイ上では非常に読みやすいフォントですが、メタファイルでは正確に拡大縮小できない固定フォントです。

Q ビットマップでは機能することが分かりましたが、データ自体を Excel のようなアプリケーションに移すことはできますか。

データを ASCII ファイルにエクスポートできるエクスポート・ユーティリティ(「Waveform」メニュー:「File」=>「Export」)があります。Helmut Sennewald が作成したサードパーティの無料ユーティリティもあります。このユーティリティは独立したユーザグループ <http://tech.groups.yahoo.com/group/LTspice> から入手できます。このユーティリティでは、別のシミュレーション実行回で得られた波形をマージする機能など、さまざまな形式のデータ操作が可能です。

Q Helmut Sennewald とは誰ですか。

写真の右側の男性です。



スイッチ・モード電源ループ利得の抽出と、通常はその必要がない理由

実際の目標が発振器の作成でない限り、負帰還ループの利得を周波数とともに低下させ、位相シフトが大きくなり過ぎないうちに単位利得より小さくする必要があります。[1]この考え方は、スイッチ・モード電源(SMPS)の安定性解析に当てはめることができます。SMPSが本来は小信号の線形等価回路を持たない非線形回路であるとしても、通常は、フィルタ処理された切り替え後の出力上で動作するアナログ帰還ループがあります。

SMPSのループ利得を求める場合には次の2つの問題が関係しています。(i)閉ループ系から開ループ利得を求めることと、(ii)スイッチング・サイクル間の平均をとるか、フーリエ解析を使用してスイッチング周波数成分を無視すること(あるいはその両方)により、スイッチング波形を無視することです。最初の問題は帰還ループのほとんどの安定性解析では一般的です。安定性解析は開ループ応答に基づいていますが、開ループ応答を直接測定するために帰還ループを開くと回路はもはや動作しないので、これが帰還を最初に使用したそもそもの理由です。2番目の問題は、SMPSが本質的に非線形回路であり、線形帰還理論は、基本的にスイッチング・サイクルにわたって平均化された仮想波形に制限されるという事実に起因します。

線形の閉ループ系の開ループ応答の割り出しは、Middlebrookの方法[2]によって十分に解明された問題です。その方法は、閉ループ系に注入したテスト信号を使用して、電圧利得と電流利得を別々に解くものです。これら2つの利得が互いに複雑に関係することで、真のループ利得が得られます。低インピーダンスが高インピーダンスを駆動する、帰還ループ内の点を特定できる場合、電流利得は0になるので、電圧利得だけを測定し、その値をループ利得であると認識すれば十分です。電源出力がエラーアンプ入力を駆動しているため、そのような点は、通常はSMPS内にあります。

SMPSループ利得の実験室での測定は、Venable Corporationが最初に開発した商用計測機器によって自動化され、現在では他社からも入手できるようになりました。注入されたテスト信号とフーリエ解析を使用する技術は、周波数応答解析(FRA)と呼ばれます。この方法は実験室では日常的なものですが、この方法をシミュレーションでどう使用するかを必ずしも誰もが分かっているとは限りません。この記事では、LTspice IVでFRAを行う方法について説明します。この方法では、Middlebrook法の電圧利得の部分、フーリエ変換を行う.measure文、周波数を掃引するステップ文、および.measure文の結果をプロットできるようにするLTspiceの機能を使用します。以下の手順を通読する場合には、実用的なFRAの例を参照すると分かりやすいことがあります。この例は一般的なLTspice IVリリースの一部であり、通常はディレクトリ

C:\Program Files\LTC\LTspiceIV\examples\Educational\FRA\にインストールされています。

- ステップ 1: 低インピーダンス電圧源が高インピーダンス入力を駆動している、SMPS 帰還ループ内の点を特定します。これに役立つのは2箇所、SMPS コントローラの帰還ピンと直列になるところか、帰還ピンに向かう抵抗分割器の上側に接続する出力の間です。
- ステップ 2: ここに電圧源を挿入します。これは帰還ループを乱す時間領域正弦波になります。これに対して「SINE(0 10m {Freq})」という値を指定します。選択した振幅(ここ

では 10mV) は、この方法の精度と信号対ノイズ比に影響します。振幅が小さいほど、信号対ノイズ比も小さくなります。しかし、振幅が大きすぎると、周波数も影響を受けるため、系が直線的に動作しなくなり、周波数応答の関連性は低くなります。

ステップ 3: この電圧源の両端のノードにラベル「A」および「B」を付けます。帰還の方向はノード A からノード B の方向にします。たとえば、電圧源を帰還ピンに直接接続する場合、ノード B は帰還ピン側であり、ノード A は電圧源の反対側になります。

ステップ 4: 以下の .measure 文を SPICE 指令として回路図に貼り付けます。

```
.meas Aavg avg V(a)
.meas Bavg avg V(b)
.meas Are avg (V(a)-Aavg)*cos(360*time*Freq)
.meas Aim avg -(V(a)-Aavg)*sin(360*time*Freq)
.meas Bre avg (V(b)-Bavg)*cos(360*time*Freq)
.meas Bim avg -(V(b)-Bavg)*sin(360*time*Freq)
.meas GainMag param 20*log10(hypot(Are,Aim)/hypot(Bre,Bim))
.meas GainPhi param mod(atan2(Aim,Are)-atan2(Bim,Bre)+180,360)-180
```

これらの .measure 文により、ノード A および B のフーリエ変換が行われ、その後、結果の複素数電圧の比が計算されます。結果は系の複素開ループ利得になります。振幅は GainMag(dB) で与えられ、位相は GainPhi(°) として与えられます。

ステップ 5: シミュレーション・コマンドの以下の内容を SPICE 指令として回路図に貼り付けます。

```
.param t0=.2m
.tran 0 {t0+10/freq} {t0}
```

パラメータ t0 は、系が定常状態になるのに必要な時間の長さです。おそらく数回はシミュレーションを実行して、適切な t0 の値を決定する必要があります。t0 は .tran コマンドの3番目のパラメータとして現れますが、これはシミュレータがデータの保存を開始する時刻という意味です。これにより、ステップ 4 の .meas 文が解析中にこのデータを使用するのを防ぎます。この処理が行われる理由は、最初のトランジェント状態が、レギュレーション状態からの小規模な乱れの範囲内(小信号応答とみなすことができる範囲内)で動作していない可能性があるからです。

.tran コマンドの 2 番目と 3 番目のパラメータの両方に t0 があることに注意してください。2 番目のパラメータは停止時刻です。開始時刻と停止時刻の差は、 $10/\text{freq}$ (つまり擾乱サイクルの整数倍)として選ばれました。擾乱サイクルとスイッチング・サイクルの両方の整数倍となる期間にわたってフーリエ解析を行うのが理想ですが、これは必ずしも可能とは限りません。ループ利得はスイッチング周波数の数分の 1 の周波数で 1 未満に低下する必要があるため、スイッチング・サイクルは擾乱サイクルよりも必ず多くなります。そこで、擾乱サイクルの整数倍を使用して、スイッチング・サイクルの整数倍以外からの誤りが小さくなることを期待します。多くのスイッチング・サイクルは組み込まれるからです。

ステップ 6: 解析を行なう周波数(複数可)を選択します。単一の周波数で実行するには、単に次の SPICE 指令を追加します。

```
.param Freq=15K
```

そして、シミュレーションを実行します。.meas 文の出力は、シミュレーションを実行した後にメニューコマンド「View」=>「SPICE Error Log」を使用して表示できるエラー・ログにあります。回路図上に次の SPICE 指令を配置すれば、複数の周波数でシミュレーションを実行できます。

```
.step oct param freq 50K 100K 5
```

この指令により、LTspice は、50kHz から 100kHz までの周波数で、1 オクターブ当たり 5 点を使用してシミュレーションを実行するよう指示されます。シミュレーション完了後、これをボーデ線図としてプロットするには、メニューコマンド「View」=>「SPICE Error Log」を実行して、メニュー「Plot .step'ed .meas data」を右クリックします。この時点ではボーデ線図にプロットするデータがないので、もう一度右クリックし、メニューコマンド「Visible Traces」を実行して、利得を選択します。

上記の技術を身に付けたことで、帰還ループのボーデ解析によって SMPS の設計を進めて攻略できるという感触を持つかもしれません。その気持は理解できます。帰還ループを横断してポールおよびゼロを発生させた部品を識別し、どのゼロを動かしてどのポールを打ち消すかを入念に計画し、補償回路網の部品の値を合成して、安定した帰還ループを実現できた場合には、意義があります。しかし、この手順はまさに、この技術ではできないことであり、これ以外のどの周波数領域技術を駆使してもできないことなのです。その理由を説明しましょう。

C:\Program Files\LTC\LTspiceIV\examples\Educational\FRA\Eg3.asc

にあるような、標準的な固定周波数、ピーク電流モードのスイッチャを考えます。

コントローラは、クロック・パルスによって設定されるフリップフロップを使用して、インダクタ電流を増加するス

スイッチをオンします。ピーク・スイッチ電流がエラーアンプの出力の電圧に比例するようになると、フリップフロップはリセットされ、スイッチはオフになり、次のクロック・パルスが再びフリップフロップを設定するまでの間、コントローラはアイドル状態のままになります。平均電流は幾何学的因子までピーク電流に比例するので、1 クロック・サイクルの間で平均した場合、このフリップフロップ制御スイッチはトランスコンダクタンス回路のように動作します。それはスイッチを流れる電流であり、エラーアンプの出力の電圧に比例します。さて、帰還経路に沿ってたどると、スイッチ電流と直列にインダクタが接続されています。スイッチは電流源なので、インダクタの直列インピーダンスがリアクタンスである場合でも、位相シフトは発生しません。実際にはこのことが電流モード制御の要点であり、このコントローラを購入する理由です。帰還経路をたどっていくと、今度は SMPS の出力に到達しました。出力フィルタ・コンデンサ(C4)により、ポールが 1 つ発生します。その後、出力は帰還抵抗分割器で分割され、帰還ピンでリファレンス電圧と比較されます。分割された出力とリファレンス電圧との差は誤差電圧です。この誤差電圧はエラーアンプによって増幅され、電流になってエラーアンプから流れ出します。ただし、この電圧はエラーアンプ出力の電圧であり、エラーアンプから流れ出る電流(スイッチ電流を決定する電流)ではないので、帰還ループの横断を完了するには、その電流を電圧に変換する必要があります。抵抗を使用すれば電流を電圧に変換できますが、コンデンサ(C1)を使用する方がはるかに優れています。コンデンサを使えば開ループ DC 利得が最大になり、出力が固定電圧に安定化された状態で維持されるからです。そのコンデンサにより、2 番目のポールが形成されます。

ここで、各ポールによって 90° に限りなく近い位相シフトが生じる可能性があるので、コントローラは何らかの付加的な遅延を発生させる必要があることから、安定した帰還ループを確保する回路設計が必要と考えるでしょう。しかし、特にアルミニウム電解コンデンサの出力フィルタ・コンデンサを使用する場合には該当しません。このコンデンサには ESR があり、それによって応答にゼロが生じるからです。また、補償コンデンサ C1 と直列抵抗 R1 を購入するので、これによっても応答に別のゼロが生じます。さらに、コントローラによる遅延はスイッチング周波数と比較するとごくわずかです。ループのクロスオーバー周波数以下の周波数では、遅延は無視できます。このことはすべて、ループが安定していることを意味しており、ループはすべての部品の値で安定しているので部品の値は合成できません。この議論で基本的に指摘しているのは、電流モード SMPS の帰還ループによって安定化された信号が、1 回のスイッチング・サイクルの間に平均化された電流によって十分に描写されるとすぐに、そのループは安定するということです。

出力フィルタ・コンデンサがアルミニウム電解コンデンサではなくセラミック・コンデンサである場合、セラミック・コンデンサの ESR は SMPS の安定性に大きく影響するほど大きくありませんしたがって、ループ応答は、前述の議論に従うと、2 つのポールと 1 つのゼロが形成されているので、エラーアンプ出力に接続された出力フィルタ・コンデンサまたは RC 回路の特定の部品の値とは関係なく、引き続き安定しているはずですが、上記の解析の適用範囲の許容限度について議論することは目的にはなっています。電流モード SMPS の安定性に関する上記の説明の精度から低下させる影響があります。デューティ・サイクル(したがってリップル電流)は出力電圧とともに変化するので、平均電流は出力電圧が変動する間ピーク電流に比例しません。このため、コントローラのフリップフロップを作動させるピーク電流が同じでも、出力電圧が変化する間の平均電流は同じにはなりません。これは、エラーアンプの出力中の電圧からインダクタに流れ込む電流までの伝達関数は完全にはトランスコンダクタンスとは表現されないが、何らかの実インピーダンスにより分流されたトランスコンダクタンスと表現されることを意味します。

このインピーダンスは標準で数オームです。MOSFET のオン抵抗と比較すると非常に大きな値ですが、無限大よりは小さな値です。これは安定性の観点から望ましいことではありません。インダクタには電流源から電流が供給されず、そのリアクタンスによって位相シフトが生じる可能性があるからです。この状況は、スロープ補償によってさらに悪化します。スロープ補償は、高いデューティ・サイクルで動作する固定周波数電流モード・コントローラで発生する分数調波振動に対する調整です。この技法では、測定したスイッチ電流に偽電流を加算し、その量を使用してコントローラのフリップフロップをリセットする必要があります。電流以外の量を使用してフリップフロップをリセットする影響により、インダクタに電流を供給する電流源のインピーダンスが減少するので、インダクタのリアクタンスによってさらに多くの位相シフトが生じます。全般的に見て、電流モード SMPS を不安定にするのはかなり難しいことが分かります。たとえば、値が 1 桁または 2 桁高いインダクタンスを使用した場合、インダクタのリップル電流は非常に小さくなるので、スロープ補償の偽電流がフリップフロップのリセットを制御するようになります。これにより、このインダクタンスを駆動する切り替え電流源のインピーダンスは MOSFET の $R_{ds(on)}$ のインピーダンスまで減少します。このため、インダクタはループ内に別のポールを形成するので、それによって不安定性が生じます。ただし、その状況では、電流モード・コントローラを使用している場合でも、電源は実際には電圧モードで動作します。電圧モード電源の小信号線形解析はきわめて有益です。帰還ループを考案していずれかのポールを打ち消さない限り、電源は発振し、最初の電源投入時に自爆してしまう可能性があるからです。電流モード電源はまったく異なります。電流モード・スイッチャの小信号線形解析を行うことは可能ですが、電源が実際に電流モードで動作している限り帰還ループは安定しているので、この方法で実現する技術はあまり多くありません。

お伝えできる最後のアドバイスは、SMPS が安定していて電流モードで動作していると確信できる方法の答えです。答えは、データシートの表紙の回路図から始めることです。そこでの重要な情報は、インダクタンスの値、出力フィルタ容量、および外部補償部品の値です。一部のデータシートには、これらの値を計算するための式が記載されていますが、まずはこれらの値から始めて、時間領域シミュレーションを使用して調整し、応答を評価します。結論として、周波数領域の解析全般で重要なことは、時間領域の応答を改善することです。電流モード・スイッチャでは、安定性は既に実現されているので、通常は時間領域シミュレーションに直接移行してオーバーシュートを確認します。

1] この説明は最小位相系に限定されます。

2] R. David Middlebrook 著、『Measurement of Loop Gain in Feedback Systems』、International Journal of Electronics (vol 38, no. 4, 485~512 ページ、1975 年 4 月)

Linux 環境での動作

Q このプログラムの Linux 版はありますか。

独立した Linux 版はありませんが、WINE 環境で動作します。このプログラムは、Linux RedHat 8.0 と WINE バージョン 20030219 の組み合わせ、RedHat 9.0 と WINE 20040716 の組み合わせ、および SuSE 9.1 と 20040716 の組み合わせでテストされています。

Q WINE の使用経験がないのですが、インストール方法を教えてください。

<http://www.winehq.com> を調べて、ご使用のシステムでの WINE の現行バージョンを探してください。

該当する.rpm ファイルをマシンにコピーして、Nautilus で開きます。

ファイル LTspiceIV.exe を <http://www.linear-tech.co.jp> から入手します。xterm で、「wine LTspiceIV.exe」を実行して LTspice をインストールします。

GNOME デスクトップにリニアテクノロジーのロゴが表示されるようになります。このロゴをダブルクリックするか、xterm から「wine LTspiceIV.exe」と入力してプログラムを起動します。

Q WINE 環境では、回路図のフォントの拡大/縮小が Windows 環境ほど滑らかではありません。それはなぜですか。

WINE は、検出したフォントに対しては、最大限の処理を行っています。ファイル arial.ttf および cour.ttf を Windows システムから検出する方法を WINE に指示すれば、動作は改善されます。

Q WINE 環境では PWL 追加点エディタが正しく動作しているように見えません。

ライセンスを得ている Windows システムからネイティブの Windows .dll を使用してみてください。その後、WINE から LTspice を起動するコマンド・ラインは、wine -dll commctrl,comctl32=n scad3.exe になります。

Q Windows 環境と WINE/Linux 環境では LTspice がわずかに異なる動作しているように見えます。それはなぜですか。

LTspice は、LTspice 自体が WINE 環境で動作しているかどうかを検出します。WINE 環境で動作している場合

は、数件の WINE の問題を回避します。コマンド・ライン・スイッチ wine を使用して、WINE 環境で動作しているように LTspice を認識させることができます。WINE 問題の対処に興味がある場合は、コマンド・ライン・スイッチ nowine を使用して、WINE 環境で動作していないように認識させることができます。

Q Linux 環境での動作時の性能を Windows 環境での動作時と比較するとどうですか。

質問された Linux ユーザは全員、Linux 環境の方が Windows 環境より LTspice が良好に動作すると答えるでしょう。

紙のマニュアル

ヘルプ・ページがマニュアルです。

ヘルプ・ページ式は印刷できるように準備されていますが、そうする前に環境について考えてください。

Q 関連書籍はありますか。



アマゾンをチェックすると、本のタイトルに LTspice が含まれる本は 10 冊あります。著者が渋谷道夫、Gilles Brocard、および Mihail Pushkarev (表紙は上図には示さず) の書籍は高品質です。このヘルプ文書のタイトル・ページの図は渋谷道夫によるものです。

ユーザグループ

独立したユーザグループ <http://tech.groups.yahoo.com/group/LTspice> でチュートリアル、ライブラリ、事例のファイルを公開しています。