



PLECS Tutorial

Introduction to PLECS Spice

PLECS Spice入門

- PLECSモデルにSPICEネットリストを組み込む方法の学習を開始します -

Tutorial Version 1.1

1 はじめに

PLECSは理想スイッチを用いることで、複雑なパワーエレクトロニクスシステムの高速かつ効率的なシミュレーションを実現できます。これらのモデルは、シミュレーション時間を短縮し安定した数値挙動を実現しますが、一般的に、コンポーネントの寄生効果などの物理的効果の詳細を捉えきれません。

より詳細な解析が必要な場合、例えば部品の寄生要素の影響を解析する場合などは、SPICEモデルを使用することができます。これらのモデルは、部品の物理的挙動をより正確に再現することを目的としていますが、その代償として、シミュレーション時間と計算負荷が大幅に増加します。

このチュートリアルでは、PLECS Spiceの基本的な機能とワークフローについて学びます。典型的なフライバックコンバータの理想化PLECSのモデルを、SPICEベースのコンポーネントで拡張します。段階的に、PLECS Spiceの機能を理想的な設計に統合していきます。

学習目標:

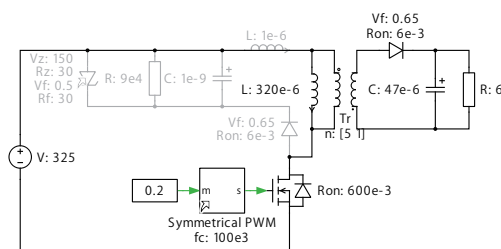
- ・ 既存のSPICEネットリストをPLECS回路図にインポートする方法
- ・ データシートのパラメータに基づいてシンプルなSPICEネットリストを作成する方法
- ・ SPICEコンポーネントのモデル参照ブロックを活用する方法
- ・ 回路図をPLECS Spice対応にする方法
- ・ コンポーネントプローブの定義方法
- ・ PLECSモデルとSPICEモデルをシームレスに切り替えるための可変サブシステムの作成方法

始める前に

- 1 Pleximの[Tutorials](#)ページからチュートリアル"PLECS: Introduction to PLECS Spice"の.zipファイルをダウンロードし、ローカルドライブに解凍してください。
- 2 input_dataフォルダには、参考資料としてSPICEネットリスト(.libファイル)とコンポーネントのデータシートが含まれています。
- 3 flyback_converter_1...5.plecsファイルは、チュートリアルの中段階を表し、参照モデルとして機能します。

このチュートリアルは、フライバックコンバータの回路図を含むファイルflyback_converter_0.plecsから始まります。参考として、[図1](#)に回路図を示します。

図1: PLECSでモデル化されたフライバックコンバータ(チュートリアル出発点)



注意 一部のコンポーネントは意図的にコメントアウトされており、チュートリアルの後の段階で有効化されます。



あなたのタスク: PLECSモデルflyback_converter_0.plecsを開き、**シミュレーション** -> **開始**をクリックしてシミュレーションを実行し、シミュレーションが期待どおりに実行されることを確認します。スコープでシミュレーション結果を調べます。

2 MOSFET用のSPICEネットリストを追加

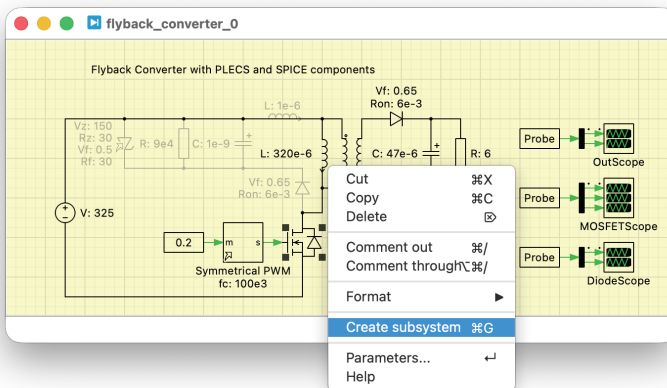
チュートリアルはこの段階では、特定のコンポーネントのSPICEネットリストを使用して、MOSFETの代替記述を作成します。



あなたのタスク:

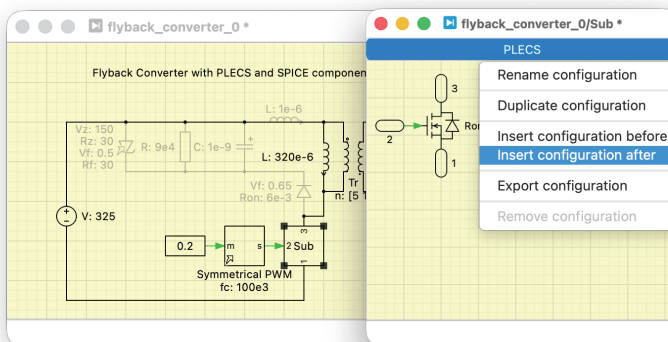
- 1 ダイオード内蔵MOSFETコンポーネントを**右クリック**して**サブシステムの作成**を選択するか、**Ctrl+G**を押します。[図2](#)を参照してください。

図2: サブシステムの作成



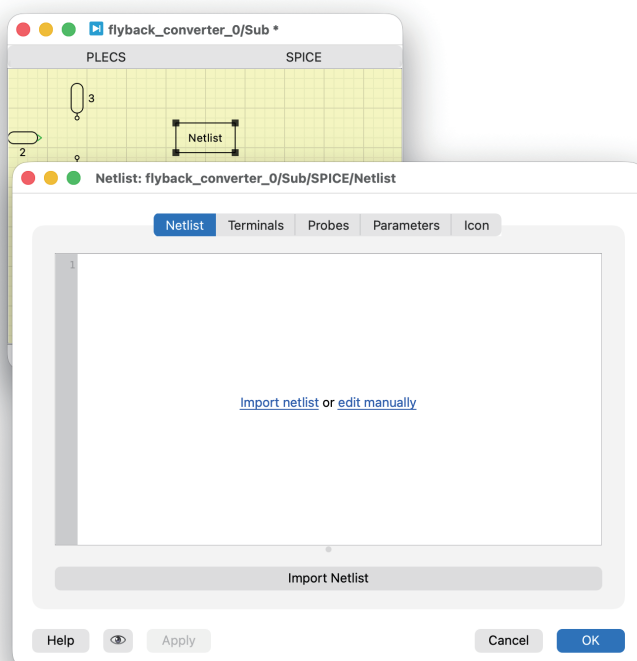
- 2 新しく作成したサブシステムを**右クリック**し、**サブシステム -> 可変サブシステムへ変換**を選択します。オプション: *Sub* ラベルをダブルクリックして新しい名前(例: ConfigMOSFET)を入力することで、可変サブシステムの名前を変更できます。コンポーネント名を非表示にするには、**Ctrl+Shift+N**キーを押すか、コンポーネントを右クリックして**ブロックの配置 -> 名前の表示**を選択します。
- 3 可変サブシステムを開きます(**Ctrl+U**を押すか、右クリックして**サブシステム -> サブシステムのモデル表示**を選択します)。
- 4 既存の設定の名前を変更するには、*Configuration 1*を**ダブルクリック**します。分かりやすい名前を入力してください (例: *PLECS*)。
- 5 既存の設定の後に新しい設定を追加し(設定を右クリックして**回路を後に挿入**を選択)、新しい設定の名前を例えばSPICEに変更します ([図3](#)を参照)。

図3: 回路を後に挿入を選択



- 6 この2番目の構成はまだ空です。コンポーネントライブラリのSPICEセクションからネットリストコンポーネントを配置してください。
- 7 図4に示すように、ネットリストコンポーネントをダブルクリックしてネットリストウィンドウを開きます。ネットリストインポートをクリックし、目的のネットリストライブラリファイル(例: Infineon_CoolMOS_C3_600V_Spice.lib)を選択します。

図4: ネットリストウィンドウ



- 8 図5のようなウィンドウが表示されます。下部のドロップダウンメニューでは、ネットリストライブラリにあるサブ回路の中から目的のサブ回路を選択できます。ドロップダウンメニューから、IPD60R600CM8_L0コンポーネントを選択してください。選択されたサブ回路とすべての依存関係がインポートされ、ネットリストコンポーネントに組み込まれます。

図5: ネットリストインポート

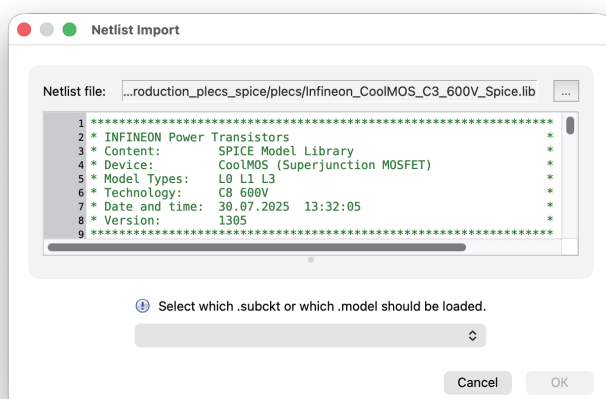
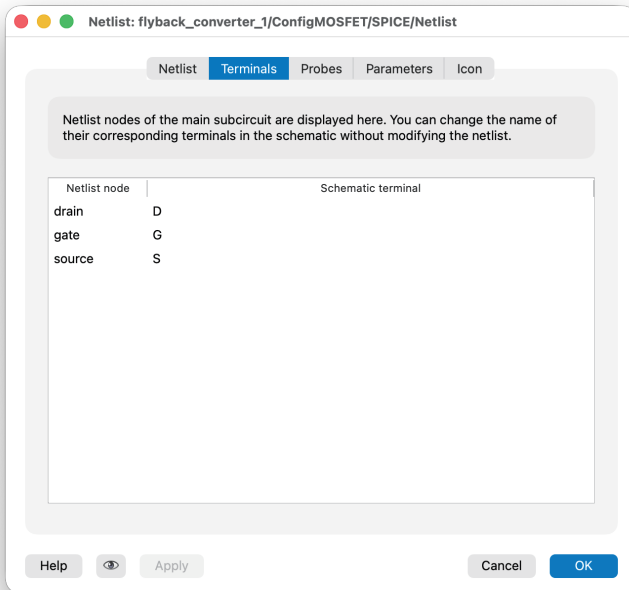
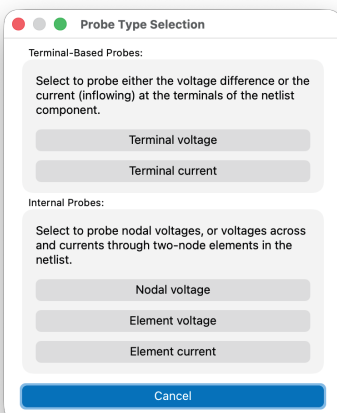


図6: ネットリストの端子タブ



- 9 ネットリストウィンドウの**端子**タブから、各回路図端子にカスタム名を割り当てることができます。図6に示すように、ネットリスト・ノードのdrain、drain、sourceにそれぞれカスタム端子名D、G、Sを割り当てることができます。

図7: プローブタイプの選択ウィンドウ



- 10 ネットリストウィンドウの**プローブ**タブから、ネットリストコンポーネントにプローブ信号を追加できます。+アイコンをクリックすると、図7に示すメニューが開きます。

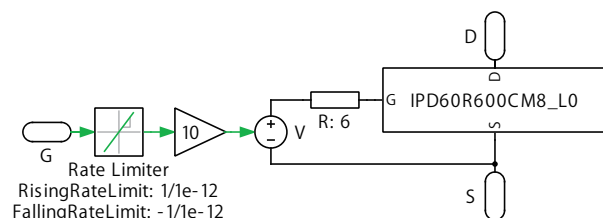
- MOSFETのチャネル電圧を測定するには、端子Dと端子S間に**端子 電圧**プローブを追加します。信号名を入力してください(例: V_{ds})。
- MOSFETのドレイン端子に流入する電流を測定するために、端子Dに**端子 電流**プローブを追加します。信号名を入力してください(例: I_d)。
- MOSFETのゲート電圧を測定するために、端子Gと端子S間に**端子 電圧**プローブを追加します。信号名を入力してください(例: V_{gs})。

11 MOSFETのモデリングを完了し、PLECS Spiceに対応させるには、ゲート駆動回路を含める必要があります。最もシンプルなゲートドライバは、わずか4つの部品で実現できます:

- ・ **入力制限:** 制御信号からの矩形パルス平滑化し、有限の立ち上がり時間をモデル化します。この例では、立ち上がりおよび立ち下がりスルーレートを1V/1ps(つまり、1e12V/s)に設定します。この値は、ほぼ理想的な遷移に近い値です。
- ・ **ゲイン(利得):** 論理制御信号を適切なゲート電圧レベルにスケールリングします。ゲインを10に設定することで、0から1までの入力信号が0から10Vまでのゲート電圧に変換されます。
- ・ **電圧源(可変):** 信号を電圧に変換します。
- ・ **抵抗器:** ゲート駆動抵抗をモデル化します。この抵抗はMOSFETのゲート容量の充放電速度を制御するため、スイッチング速度、スイッチング損失、およびリング挙動に影響を与えます。この例では、抵抗値を6Ωに設定します。

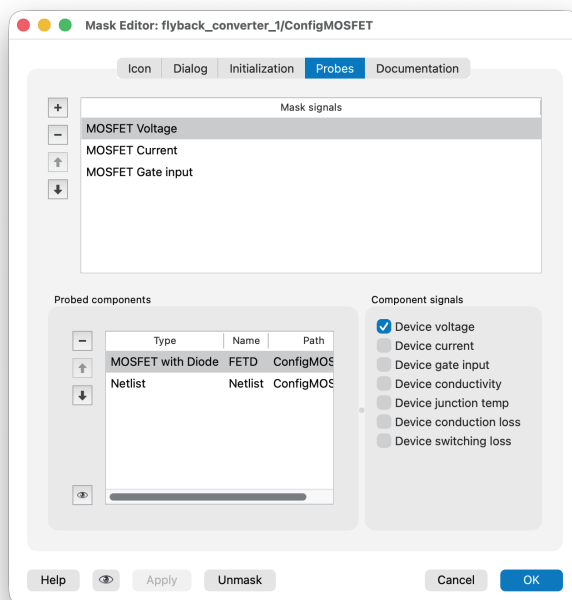
12 オプション: 図8の回路図のように、サブシステムの信号入力ポートとポートD、G、Sの名前を変更します。

図8: SPICE MOSFET構成の回路図



この段階では、MOSFETのSPICE構成の回路図は図8に示すものと同じになっているはずです。

図9: 可変MOSFETサブシステム: プローブタブ



13 ステップ10で定義したプローブ信号は、外部のPLECS回路図からSPICE回路信号にアクセスするために使用されます。したがって、これらの信号は可変サブシステムに渡されなければなりません。これを実現するには、サブシステム用のマスクを作成し、プローブ信号を次のように定義します:

- 可変サブシステムを選択し、コンテキストメニューから**サブシステム -> マスクの編集**を選択するか、**Ctrl+M**を押します。
- マスクエディタの**プローブ**タブで、新しい**マスク信号**を追加し、例えば**MOSFET Voltage**など、意味のある名前を付けます。
- **プローブ・コンポーネント**セクションで、MOSFETコンポーネントのPLECS構成からダイオード内蔵MOSFETコンポーネントを、またSPICE構成からネットリストコンポーネントを**ドラッグアンドドロップ**し、それぞれにプローブ信号を割り当てます。
- プローブが必要な信号ごとに、この手順を繰り返してください。その結果、マスクの**プローブ**タブは図9のようになります。
- **注意:** MOSFETのPLECS構成とSPICE構成をシームレスに切り替えられるようにするには、両方の構成に同じ数のプローブ信号を割り当ててください。
- **オプション:** マスクエディタの**アイコン**タブでは、Lua構文とPlecsIconLibライブラリを使用してアイコンを描画できます。ドキュメントは[オンライン](#)で入手可能です。図11のようなMOSFETを描画するには、以下のコードを使用します。

```
local IconLib = require('PlecsIconLib')
IconLib.mosfetDiode(0, 0)
```

14 アイコンがマスクしたサブシステムの端子位置と合わない場合:]

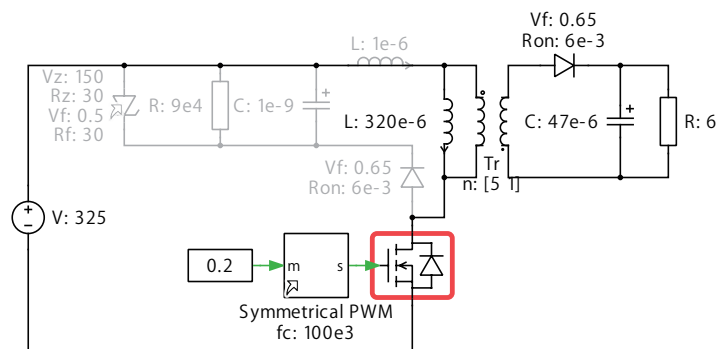
- サブシステムブロックにマスクアイコンを定義すると、PLECSは自動的にそのブロックと、基となる回路図のポートを保護します。サブシステムブロックのサイズ変更や、サブシステムフレーム上の端子の位置変更はできなくなります。
- マスクされたサブシステムブロックを変更したい場合は、サブシステムを右クリックして**サブシステム -> リンクの非保護**を選択することで保護を解除できます。
- モデルが保護されていない状態になったら、**Shift**キーを押しながら**マウスの右ボタン**を押して、マスクアイコンで示された正しい位置に各端子をドラッグできます。]
- 端子がマスクに対して適切な位置に配置されたら、サブシステムのサイズをマスク自体に合うように調整することをお勧めします。
- 後で同じメニューから**リンクの保護**を選択することで、再度保護することができます。
- マスクエディタの**アイコン**タブでサブシステムのフレームを表示の選択を解除するとコンポーネントフレームを非表示にでき、また端子ラベルの非表示を選択すると端子ラベルを非表示にできます。

図10: プローブコンポーネントを"MOSFETScope"に接続



15 ステップ13でマスクしたサブシステムで定義されたプローブは、PLECSまたはSPICE構成からの信号にアクセスするために使用します。たとえば、それらをオシロスコープに表示するために使用されます。"MOSFETScope"に接続されたプローブコンポーネント(図10参照)で、"- "アイコンをクリックしてMOSFETとダイオードコンポーネントを削除し、新しく作成したマスクされたサブシステムを**プローブ**コンポーネントウィンドウにドラッグアンドドロップして、コンポーネント信号内のすべての信号を選択します。

図11: 可変サブシステムのMOSFETを搭載したフライバックコンバータ(チュートリアルステージ1)



 ダイオードコンポーネントを備えた従来のMOSFETは、可変システムに置き換えられました。このサブシステムにより、同一の回路図内でPLECSベースのMOSFETモデルとSPICEベースのMOSFETモデルを切り替えることが可能になります。モデルは、[図11](#)と同じになっているはずです。このモデルはファイルflyback_converter_1.plecsに含まれています。

 **注意** このモデルでは、同じ電気回路内の他の半導体がまだ理想化されたPLECSモデルのみを使用して実装されているため、SPICEシミュレーションに対応できる状態ではありません。同一回路内のPLECSコンポーネントとSPICEコンポーネント間の電気的接続は許可されておらず、シミュレーションが失敗する原因となります。

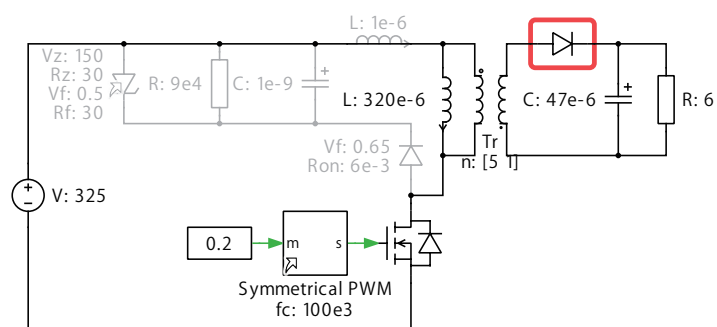
3 ダイオード用のカスタムSPICEネットリストを作成

このチュートリアルはこの段階では、データシートのパラメータに基づいて独自のネットリストを作成し、ダイオードのSPICEベースの記述を作成する方法を学びます。このチュートリアルのinput_dataフォルダのSTPS20120C.pdfに記載されているデータシートを使用して、ダイオードのSPICEネットリストを作成します。

あなたのタスク:

- 1 [図12](#)で強調表示されているフライバックコンバータの二次側ダイオードについて、[第2章](#)の手順1～6を繰り返します。

図12: 可変ダイオードを備えたフライバックコンバータ(チュートリアルステージ2)



- 2 ネットリストウィンドウで**手動で編集**をクリックすると、独自のネットリストを作成できます。
- 3 SPICEサブサーキットを定義するための最小限の構文が表示されます。

- 4 最初の行では、サブサーキットの名前とその端子(回路のノードでもある)を定義します。

```
.subckt CUSTOM_DIODE 1 2
```

ネットリスト端子1と2を持つCUSTOM_DIODEというサブ回路を定義します。

- 5 2行目で、コンポーネントの種類を定義します:

```
d1 1 2 diode
```

ダイオードd1は、ノード1とノード2間にdiodeと呼ばれるカスタムダイオードモデルで接続されています。

- 6 3行目では、SPICEステートメントDを使用してダイオードモデルを指定します:

```
.model diode D(N=1.4 VJ=.6 RS=7.5E-3 IS=350E-9 CJO=550E-12)
```

データシートからこれらのパラメータを抽出するための基本的なガイドラインは、[第8章](#)に記載されています。

- 7 ネットリストウィンドウの**端子**タブから、各回路端子にカスタム名を割り当てることができます。この場合、ノード1と2の名前をアノードとカソードを表すAとCに変更します。

- 8 ネットリストウィンドウの**プローブ**タブから、ネットリストコンポーネントにプローブ信号を追加します。"+"アイコンをクリックすると、[図7](#)のようなメニューが開きます。

- 端子AとC間に**端子電圧**プローブを追加して、ダイオード電圧を測定します。信号名を入力します(例: *Vf*)。
- ダイオードのアノード端子の電流を測定するために、端子Aに**端子電流**プローブを追加します。信号名を入力します(例: *If*)。

図13: SPICEダイオード構成の概略図



 この段階では、SPICE構成の回路図は[図13](#)と同じになっているはずです。

- 9 [ステップ8](#)で定義したプローブ信号は、外部PLECS回路図からSPICE回路信号にアクセスするために使用されます。ダイオードの可変サブシステムのマスクの設定方法については、[第2章のステップ13](#)を参照してください。マスクの**プローブ**タブは、[図14](#)のように表示されます。

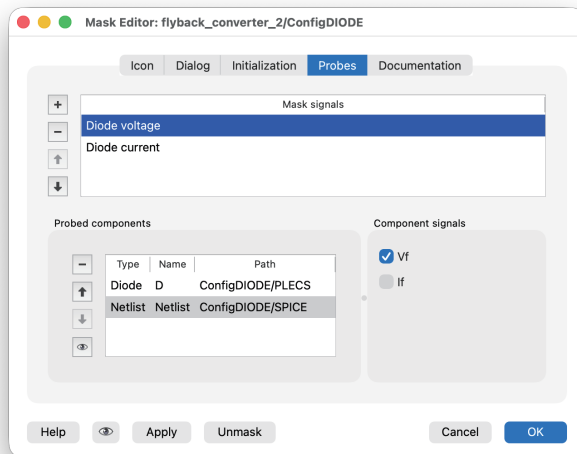
- オプション: [図12](#)のようなダイオードを描画するには、**アイコン**タブで以下のコードを使用します:

```
local IconLib = require('PlecsIconLib')
IconLib.diode(0, 0)
```

- 10 オプション: [図13](#)の回路図に従って、サブシステムのポートAとCの名前を変更します。アイコンがマスクされたサブシステムへの接続と一致しない場合は、[第2章のステップ13](#)に従ってください。

- 11 "DiodeScope"に接続されているプローブコンポーネントで、"- "アイコンをクリックして**ダイオード**コンポーネントを削除し、新しく作成したマスクされたサブシステムを**プローブエディタ**ウィンドウにドラッグアンドドロップして、**Component signals**内のすべての信号を選択します。

図14: 可変ダイオードサブシステム: プローブタブ



 元のダイオードコンポーネントは、可変サブシステムに置き換えられました。このサブシステムにより、同一の回路図内でPLECSベースのダイオードモデルとSPICEベースのダイオードモデルを切り替えることが可能になります。モデルは、[図12](#)と同じになっているはずです。このモデルはファイル flyback_converter_2.plecs に含まれています。

このモデルはまだSPICEシミュレーションに対応していません。詳細については、7ページの[注意](#)を参照してください。

4 PLECS Spiceシミュレーションを実行

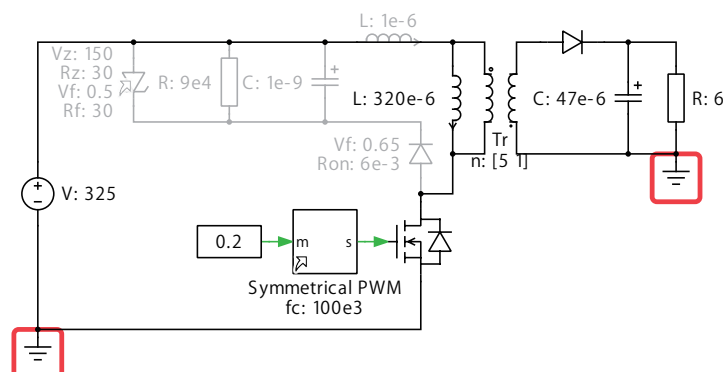
モデルはPLECS Spiceでシミュレーションを行う準備がほぼ整っています。ただし、最終的な調整がいくつか必要です。

4.1 アース(電氣的接地)ポートを追加する



あなたのタスク: フライバックコンバータの一次側と二次側の両方に、アース(電氣的接地)コンポーネントを追加します。

図15: PLECSおよびPLECS Spiceに対応したフライバックコンバータ(チュートリアルステージ3)





作成したモデルは図15に示すモデルと同じになっているはずです。モデルはflyback_converter_3.plecsファイルに含まれています。すべてのサブシステムに対してSPICE構成を選択することで、PLECS Spiceを使用してコンバータのシミュレーションを実行できます。



注意 PLECSモデルとSPICEモデルは、単一接続の電気回路内で混在させることはできません。したがって、シミュレーションが成功するのは、電氣的に接続されたすべての半導体コンポーネントが同じ実装、つまりすべてPLECSまたはすべてSPICEのいずれかを使用する場合に限られます。さらに、変圧器はPLECSとSPICEの電気ドメインを分離しません。なぜなら、ソルバから見ると、両側とも同じ電気ネットワークの一部であるためです。

4.2 変数を使用して構成を制御

可変サブシステムを手動で開いてPLECS構成とSPICE構成を切り替えるのは面倒です。代わりに、シミュレーション開始時に変数を初期化することができます。この1つの変数を設定することで、すべての可変サブシステムがそれに応じて構成を切り替えます。



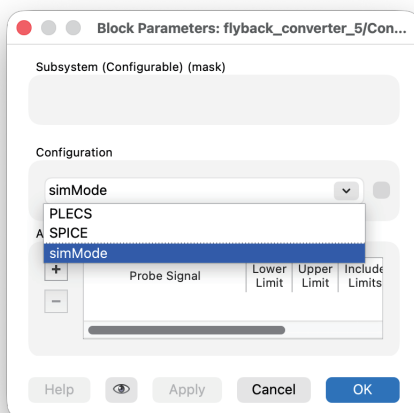
あなたのタスク:

- 1 シミュレーション -> シミュレーション・パラメータメニューを開きます。初期化タブで、構成を切り替えるために使用する変数を定義します。

```
PLECS = 1;
SPICE = 2;
simMode = PLECS;
```

この変数(例: simMode)の値は1または2を前提としており、可変サブシステムの最初の構成(PLECS)または2番目の構成(SPICE)を選択するために使用します。

図16: 可変サブシステムの構成を制御するために使用する変数



- 2 このチュートリアルで作成した可変サブシステムを、ダブルクリックして開きます。回路名ドロップダウンメニューで、<参照>プレースホルダを初期化で定義した変数名(例: simMode)に置き換えます(図16を参照)。

 **注意** 可変サブシステムでは、さまざまな設定項目の一貫した順序を維持することが重要です。これにより、例えば `simMode` などの構成選択変数によって、すべてのサブシステムに対して同じ実装(PLECS またはSPICE)がアクティブ化されることが保証されます。そうしないと、PLECSとSPICEの構成が同じ電気回路内に接続されることになり、これは許可されておらず、シミュレーションが失敗する原因となります。

 **注意** 別の方法として、変数`simMode`を文字列として定義し、例えば"PLECS"や"SPICE"などを指定して、対応するサブシステム構成を選択する方法があります。この場合、構成の順序は重要ではありません。しかし、適切なコンポーネントモデリングを選択するためには、すべてのサブシステムの構成間で表現の一貫性を維持することが極めて重要です。

 この最終調整により、モデルは**シミュレーション -> シミュレーション・パラメータ -> 初期化タブのモデル初期化コマンド**で変数の値を変更するだけで、PLECSシミュレーションとPLECS Spiceシミュレーションを切り替えることができます。PLECSモデルからPLECS Spice対応回路への変換が正常に完了しました。これでシミュレーションを実行し、次のチュートリアルでPLECSとPLECS Spiceの結果を比較することができます。

4.3 結果

このチュートリアルでは、理想化したPLECS構成で得られたシミュレーション結果と、SPICEネットリスト構成で得られたシミュレーション結果を比較します。

あなたのタスク:

- 1 変数(例: `simMode`)を設定して、設定可能なすべてのデバイスをPLECSモデルで構成します。
- 2 PLECSモデルを使用してシミュレーションを実行します。
- 3 17のように、PLECSスコープのツールバーにある**現在のトレースを保持**ボタンをクリックして波形を保持します。スコープウィンドウにツールバーが表示されない場合は、**表示 -> ツールバー**から有効にしてください。

図17: スコープツールバーの現在のトレースを保持ボタン



- 4 モデル初期化コマンド内の変数`simMode`の値を変更して、構成可能なすべてのデバイスをSPICEネットリストベースのモデルで構成します。
- 5 シミュレーションを実行します。今回はSPICEモデルを使用して実行します。

結果は、出力電圧、磁化インダクタンス電流、およびMOSFETの電圧と電流を示す18と19に示されているものと類似します。

 **注意** 19の下グラフは、MOSFETのゲート指令値を示しています。PLECSモデルでは、この信号は論理制御入力であり、0または1となります。それに対し、SPICEモデルでは、これは物理的なゲート-ソース間電圧に対応し、実際のデバイスの V_{GS} を反映しています。

SPICEモデルによって提供される追加情報は、デバイスの寄生効果によって生じる影響を把握するために不可欠です。これらの効果の一つとして、 V_{DS} の低周波振動が挙げられます。これは、磁化インダクタンスとMOSFETの出力容量との間のエネルギー交換の結果です。

図18: PLECS(緑色)およびPLECS Spice(赤色)シミュレーションにおける出力電圧と励磁インダクタ電流

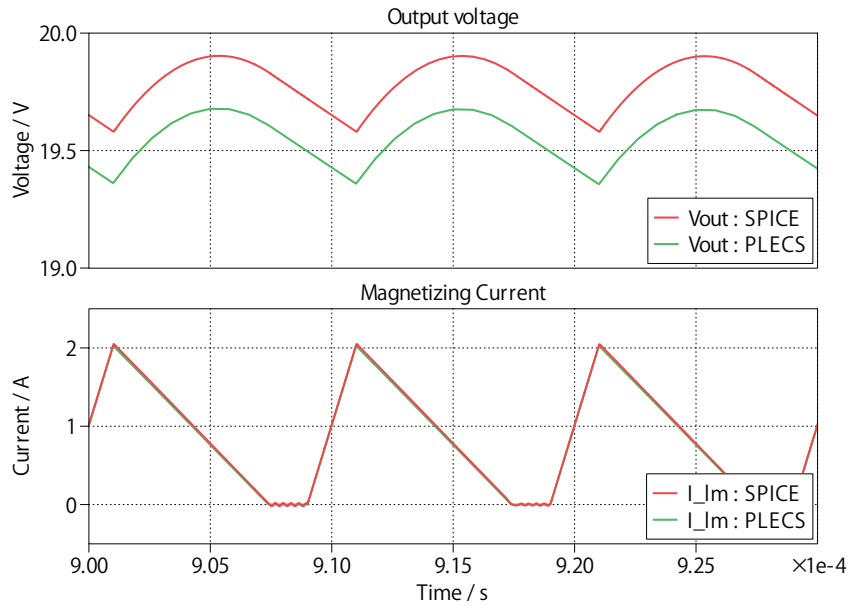
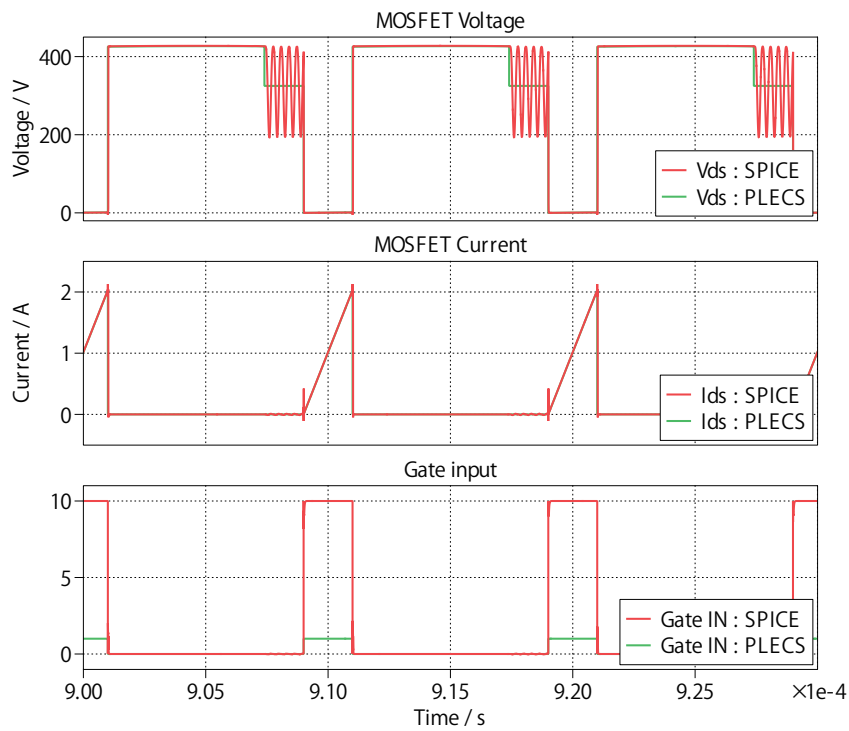


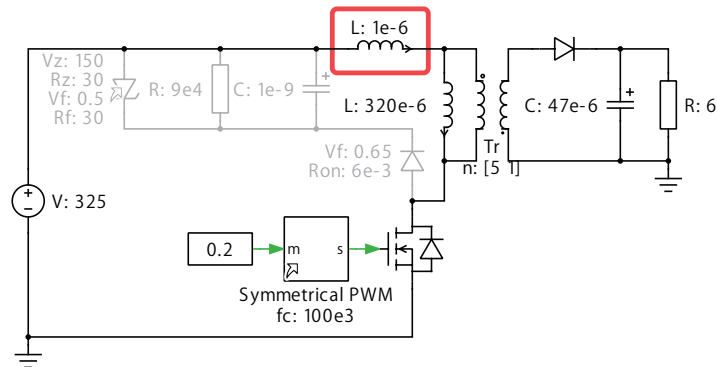
図19: PLECS(緑色)およびPLECS Spice(赤色)シミュレーションにおけるMOSFETの電圧と電流



5 変圧器の漏れインダクタンスとスナバ回路

実際には、フライバックトランスの一次側と二次側は完全に結合しているわけではありません。したがって、回路の解析精度を高めるために、回路モデルに漏れインダクタンスを含めることができます。

図20: 変圧器漏れインダクタンスを有するフライバックコンバータ



あなたのタスク:

- 1 図20のように、漏れインダクタの回路要素を無視を解除します。
- 2 PLECS Spiceシミュレーションを実行し、結果が期待どおりであることを確認します。
- 3 モデルを理想コンポーネントに戻し、PLECSシミュレーションを実行します。

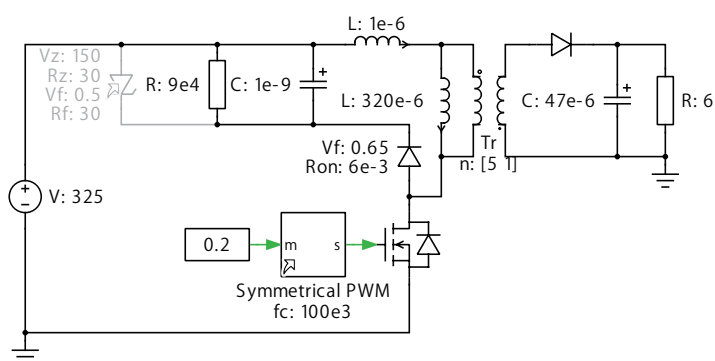
❓ PLECSがエラーメッセージを表示して異常終了するのはなぜですか?(MOSFETの導通状態を確認)。

A MOSFETが開くと、磁化インダクタンス電流が流れる経路がなくなるため、インダクタンス電流は瞬時にゼロに跳ね上がります。これはPLECSでは実行できません。そのため、「スイッチング後の不連続状態によるエラー」が発生します。

💡 **注意** PLECSがこのようなエラーメッセージを表示して異常終了した場合、通常は回路設計に何らかの問題があります。

実際には、オフ状態のMOSFETは、開放回路ではなく、非常に大きな抵抗値を示します。この効果は、MOSFETのSPICEネットリスト内でモデル化されています。したがって、PLECS Spiceを用いたシミュレーションでは問題なく実行できます。PLECSでは、この効果は例えば $R_{off} = 10 \text{ k}\Omega$ といった追加の抵抗器を理想MOSFETに並列接続することでモデル化できます。MOSFETの両端の電圧を見ると、MOSFETがオフのときに非常に高い電圧がかかっていることがわかります。この過電圧は、漏れインダクタに蓄積されたエネルギーがMOSFETチャネルを通して放出されることで発生し、場合によっては部品自体を破壊する可能性があります。図21のように、変圧器の漏れエネルギーによって引き起こされるMOSFETにかかる過電圧を軽減するために、シンプルなスナバ回路の設計を提案します。

図21: シンプルなスナバ回路を備えたフライバックコンバータ





あなたのタスク: フライバック回路の一次側にある抵抗器、キャパシタ、ダイオード(理想モデル)の各コンポーネントのコメントを解除して、スナバ回路を含めるようにします。



注意 この回路をPLECSで実行すると、トランスの漏れエネルギーによって発生するMOSFETチャネル電圧の過電圧スパイクの発生が明らかになります。しかし、新たに組み込まれたダイオードは理想的なモデリングしか行っておらず、PLECSとのみ互換性があります。したがって、現状の回路をSPICEでシミュレーションすると、エラーが発生します。

5.1 スナバダイオードをモデル参照コンポーネントと交換

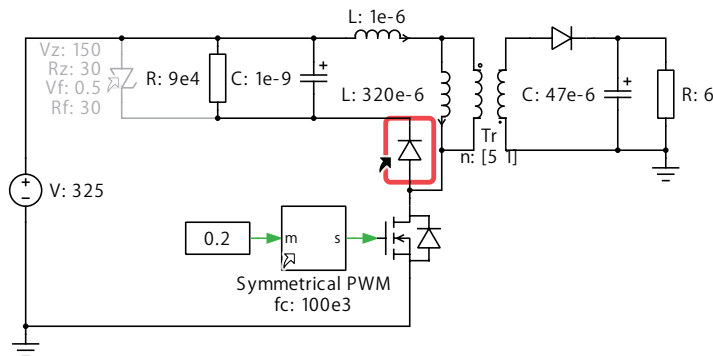
回路の反復部分は、モデル参照コンポーネントを使用して実装できます。このチュートリアルはこの段階では、モデル参照コンポーネントを使用して、フライバックコンバータの一次側のダイオードを置き換え、[第3章](#)で実装した可変サブシステムを参照します。



あなたのタスク:

- 1 PLECSコンポーネントライブラリからモデル参照ブロックを配置します。
- 2 ブロックを**ダブルクリック**してダイアログウィンドウを開きます。
- 3 [第3章](#)で実装した二次側ダイオード用の可変サブシステムを**ドラッグアンドドロップ**します。
- 4 コンポーネントの左下隅にある小さな黒色の矢印で示されるように、モデル参照が作成されました([図15](#)を参照)。元のコンポーネントに加えられた変更は、参照先のサブシステムにも自動的に適用されます。
- 5 元のPLECSダイオードをモデル参照コンポーネントに置き換えることができました。

図22: モデル参照ダイオードを使用したフライバックコンバータ(チュートリアルステージ4)



元のスナバダイオードは、可変サブシステムへの参照に置き換えられました。作成したモデルは[図22](#)に示すモデルと同じになっているはずです。モデルは、flyback_converter_4.plecsに含まれています。

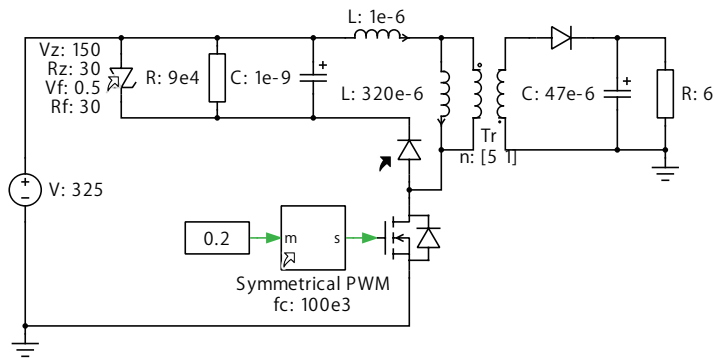
6 ツェナーダイオードを備えたスナバ回路

MOSFETにかかる過電圧を抑制するために、ツェナーダイオードを組み込むことができます。これは、コンバータの起動時に特に有効です。



あなたのタスク: 図23のように、ツェナーダイオードのブロックの無効化を解除します。

図23: モデル参照ダイオードを備えたフライバックコンバータ



注意 第5章で説明したように、新たに導入されたツェナーダイオードの理想的なモデル化のため、この回路はPLECSでのみ実行可能です。したがって、回路をそのままSPICEでシミュレーションすると、エラーが発生します。

6.1 ツェナーダイオード用のSPICEネットリストを追加

この段階で、PLECSコンポーネントをPLECS Spice対応サブシステムに変換するワークフローは明確になります。



あなたのタスク: 次の段階は、PLECSツェナーダイオードを変換します。前章で説明した手順を繰り返しますが、以下の点が異なります:

- 1 ツェナーダイオード(SMAJ1504)のネットリストは、ファイルSMAJ1_series_Spice.libに含まれています。
- 2 オプション: 適切なプローブ信号を定義します。ただし、これはチュートリアルを完了するために必須ではありません。
- 3 オプション: アイコンタブで以下のコードを使用して、図23のようなツェナーダイオードを描画します。

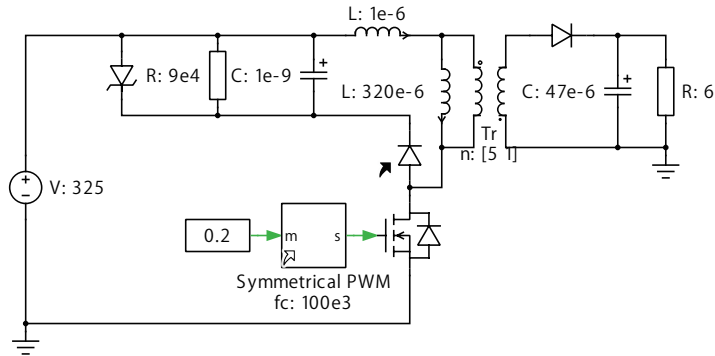
```
Icon:line({-6, -6, 6, -6}, {-7, 7, 0, -7})
Icon:line({3, 6, 6, 9}, {-9, -7, 7, 9})
Icon:line({-10, -6}, {0, 0})
Icon:line({10, 6}, {0, 0})
Icon:line({10, 6}, {0, 0})
```

アイコンがマスクされたサブシステムへの接続と一致しない場合は、第2章のステップ13に従ってください。



この段階で、すべての半導体コンポーネントは可変サブシステムに置き換えられています。これらのサブシステムにより、同一の回路図内でPLECSベースの半導体モデルとSPICEベースの半導体モデルを切り替えることが可能になります。作成したモデルは図24に示すモデルと同じになっているはずです。モデルは、ファイルflyback_converter_5.plecsに含まれています。

図24: 可変ツェナーダイオードを備えたフライバックコンバータ(チュートリアル最終)

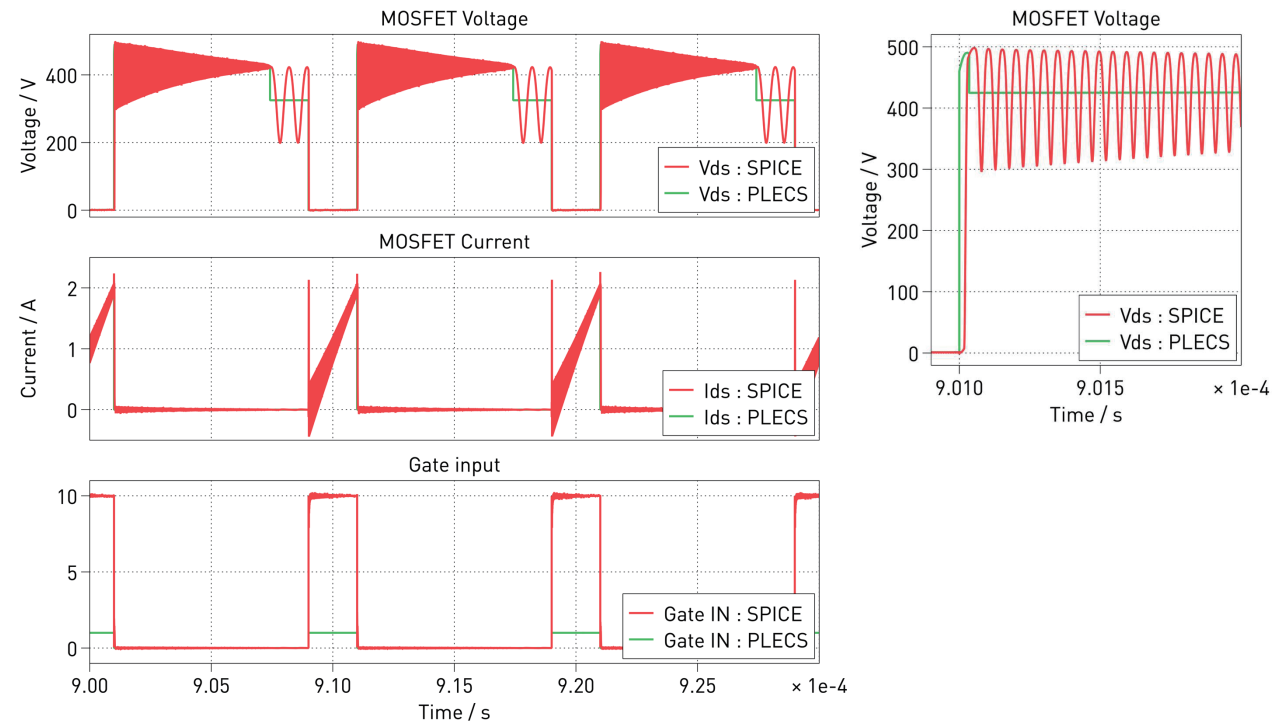


注意 すべての半導体コンポーネントはPLECSとSPICEの両方の実装をサポートする可変サブシステムに置き換えられていますが、このモデルはまだSPICEシミュレーションや両者間のシームレスな切り替えに対応していません。モデル構成のさらなる調整が必要です。

7 最終結果

あなたのタスク: チュートリアル最終では、変圧器の漏れインダクタンスとスナバ回路を含むフライバックコンバータのシミュレーション結果を比較します。そのためには、[4.3](#)で説明した手順を繰り返します。

図25: PLECS(緑色)およびPLECS Spice(赤色)シミュレーションにおけるMOSFETの電圧と電流



結果は、MOSFETの電圧と電流を示す図25と類似します。コンバータは、理想的なPLECSモデルとネットリストベースのSPICEアプローチを組み合わせた構成可能なコンポーネントを使用してすべての部分がモデル化されているため、複数の抽象化レベルでシステムを解析することが可能であり、コンポーネント間の動作と相互作用を段階的に理解することができます。

変圧器の漏れインダクタンスとスナバ回路を組み込むことで、 V_{ds} 波形はSPICEベースのモデルでしか再現できない動的特性を示します。前回のケースと比較すると、この挙動は変圧器の漏れインダクタンスとスナバ回路の存在によって影響を受けます。

8 付録: データシートから基本的なダイオードのネットリストパラメータを抽出

 **注意** ここでの説明は、データシートの値からダイオードのネットリストパラメータを抽出するための網羅的なガイドとなることを意図したものではありません。ただし、このチュートリアル目的には十分であり、PLECS Spiceを用いたシミュレーションの出発点となります。(D)ダイオードステートメントのすべてのパラメータの完全な仕様は、公式の [PLECS documentation](#) に記載されています。

第3章では、以下の SPICEステートメントDでダイオードモデルが指定されています:

```
.model diode D(N=1.4 VJ=.6 RS=7.5E-3 IS=350E-9 CJO=550E-12)
```

N 放出係数であり、指数関数の曲率を決定します。一般的に、ショットキーダイオードの適切な近似値は1.4です。

VJ 接合部電位であり、IV特性曲線をシフトさせます。ショットキーダイオードの場合、0.6Vと仮定するのが妥当です。

IS はダイオードの飽和電流であり、順方向電圧を決定します。ショックレーダイオードの方程式は次のとおりです:

$$I(V_j) = I_S \left(e^{\frac{V_j}{nV_T}} - 1 \right) \quad (1)$$

ここで、 n は上記で定義されており、 $V_T = 26\text{mV}$ は周囲温度における熱電圧です。深伝導近似の下では、飽和電流を分離することが可能です:

$$I_S = I(V_j) e^{-\frac{V_j}{nV_T}} \quad (2)$$

データシートから、IV特性から点(V_F , I_F)を抽出することが可能です。抽出した値を式に代入して I_S を導出します:

$$I_S = I_F \cdot e^{-\frac{V_F - R_s \cdot I_F}{nV_T}} \quad (3)$$

ここで、 R_s は式(5)で以下のように定義されます。データシートには順方向電圧が $V_F = V_j + R_s \cdot I(V_j) = V_j + R_s \cdot I_F$ と記載されていることに注意してください。

RS ダイオードの直列抵抗であり、高電圧/大電流条件下でのダイオードの挙動を決定づける重要な要素です。データシートに明示的に記載されている場合もあります。そうでない場合、IV特性の線形領域から2点を選択し、次の式に代入することができます:

$$I_S = I_1 e^{-\frac{V_1 - R_S \cdot I_1}{nV_T}} = I_2 e^{-\frac{V_2 - R_S \cdot I_2}{nV_T}} \quad (4)$$

これにより、直列抵抗の式を導出できます:

$$R_S = \frac{V_1 - V_2 + nV_T \ln\left(\frac{I_2}{I_1}\right)}{I_1 - I_2} \quad (5)$$

精度を向上させるには、[式\(3\)](#)と[式\(5\)](#)間を繰り返す必要があります。

CJO ゼロバイアス接合容量であり、データシートに記載されている場合があります。あるいは、ダイオードの静電容量—逆電圧特性から妥当な値を推定することもできます。このシミュレーションでは、コンポーネントの大信号特性に焦点を当てているため、接合容量は重要ではありません。

改訂履歴:

Tutorial Version 1.0 初版

Tutorial Version 1.1 中間チェックポイントを導入



☎ +41 44 533 51 00

Pleximへの連絡方法:

Phone

✉ Plexim GmbH

Mail

Technoparkstrasse 1

8005 Zurich

Switzerland

@ info@plexim.com

Email

<http://www.plexim.com>

Web



☎ +81 3 5282 7047

アドバンオートメーションへの連絡方法:

Phone

✉ ADVAN AUTOMATION CO.,LTD Mail

1-9-5 Uchikanda, Chiyoda-ku

Tokyo, 101-0047

Japan

@ info-advan@adv-auto.co.jp

Email

<https://adv-auto.co.jp/>

Web

PLECS Tutorial

© 2002–2026 by Plexim GmbH

このマニュアルで説明されているソフトウェアPLECSは、ライセンス契約に基づいて提供されています。ソフトウェアは、ライセンス契約の条件の下でのみ使用またはコピーできます。Plexim GmbHの書面による事前の同意なしに、このマニュアルのいかなる部分も、いかなる形式でもコピーまたは複製することはできません。

PLECSはPlexim GmbHの登録商標です。MATLAB、Simulink、およびSimulink Coderは、The MathWorks, Inc.の登録商標です。その他の製品名またはブランド名は、それぞれの所有者の商標または登録商標です。

本マニュアルは、Plexim社の英文マニュアルを日本語に翻訳したものです。本マニュアルと英文マニュアルとで差異がある場合、英文マニュアルを正とします。

本マニュアルの内容に基づいて発生した負傷や損害などに対して、Plexim GmbHおよびアドバン オートメーション株式会社は一切責任を負いません。製品とアプリケーションに関連したリスクを最小限に抑えるため、ユーザが適切な設計および保護対策を用意する必要があります。