

SIMetrix / SIMPLIS

SIMPLIS REFERENCE MANUAL

(概要の日本語訳)

VERSION 8.3

APRIL 2019

SIMPLIS REFERENCE MANUAL

COPYRIGHT © SIMPLIS TECHNOLOGIES INC. 1992-2019

SIMPLIS Technologies, Inc.
P.O. Box 40084
Portland, OR 97240-0084
USA

Tel: +1 503 766 3928

Fax: +1 503 296 5674

Email: info@simplistechnologies.com

Web: <http://simplistechnologies.com>



目次

1	はじめに	1
1.1	概要	1
1.2	ユーザーマニュアルの構成	1
2	入力ファイルの構成	3
2.1	概要	3
2.2	入力ファイルの一般規則	3
	文および文の継続	3
	空白文字	3
	空白行	3
	コメント文	4
	インラインコメント	4
	文の継続	4
	デバイス名	4
	モデル名とサブ回路名	5
	大文字と小文字	5
	整数エントリ	6
	浮動小数点エントリ	6
	単位	8
	フィールドと行の長さ	9
2.3	入力ファイルの構成	9
	一般の回路	9
	メイン回路	10
	サブ回路	10
	一般の文	11
3	デバイス文	12
3.1	概要	12
	デバイス文の形式	12
	ノード名	12
	電圧と電流の極性の規則	13
	パラメータの代入	13
	制御デバイス	14
3.2	SIMPLISデバイスの種類	14
	線形抵抗	14
	線形インダクタと線形キャパシタ	14
	独立した電圧ソースと電流ソース	15

三角波ソース	17
方形波ソース	20
立ち上がり時間と立下り時間がゼロのパルスソース	21
正弦波ソース	23
余弦波ソース	25
非周期的指数パルスソース	27
非周期的区分線形ソース	29
相互インダクタンス	30
線形電圧制御ソース	32
線形電流制御ソース	32
理想変圧器	33
単純なスイッチ	34
単純なトランジスタスイッチ	34
区分線形抵抗	35
区分線形インダクタと区分線形キャパシタ	36
単純な論理ゲート	36
サブ回路のコール/インスタンス化	37
4 モデル文	39
4.1 概要	39
4.2 SIMPLISで使用されるデバイスモデル	40
区分線形抵抗モデル	40
区分線形インダクタと区分線形キャパシタのモデル	44
単純なスイッチモデル	45
単純なトランジスタスイッチモデル	47
単純なロジックデバイスモデル	52
5 サブ回路の定義	70
5.1 概要	70
5.2 サブ回路の定義	71
サブ回路の親子関係	71
.SUBCKT文	72
.ENDS文 (サブ回路文の終わり)	72
5.3 定義のスコープ	73
主回路の定義のスコープ	74
サブ回路の定義のスコープ	74
5.4 デバイスとノードの定義のスコープ	74
5.5 外部ノードとローカルノード	75
5.6 サブ回路のコール/インスタンス化	75
6 制御文	78

6.1	概要	78
6.2	オプション文	78
6.3	初期条件の設定	82
	線形およびPWLキャパシタ	82
	線形およびPWLインダクタ	83
	SおよびQスイッチの初期状態の設定	83
	PWL抵抗の初期セグメントの設定	83
	単純な論理ゲートの初期状態の設定	83
	サブ回路のデバイスの初期条件	83
6.4	変数を印刷するための制御文	84
	.PRINT	84
	.KEEP	87
6.5	ノード番号への名前のマッピング	89
6.6	SIMetrixプロットの作成	89
6.7	解析に関連する文	89
	.TRAN - 時間領域過渡解析	90
	.POP - 周期的動作点解析	90
	.AC - 周波数領域解析	90
7	SIMPLISの実行	91
7.1	概要	91
7.2	SIMetrix回路図でSIMPLISの実行	91
	ネットリスト行の追加	91
7.3	外部ネットリストでSIMPLISの実行	92
7.4	スクリプトからのSIMPLISの実行	92
7.5	DOSプロンプトからのSIMPLISの実行	93
7.6	SIMPLISの実行	93
7.7	SIMPLIS実行の中止	93
7.8	SIMPLISによる自動プログラム停止	94
7.9	ネットリストプリプロセッサ	94
	概要	94
	プリプロセッサの起動	95
	ライブラリ検索	95
	パラメータ	96
	サブ回路にパラメータを渡す	96
	条件行	96
	ループ	97
7.10	モンテカルロ解析とマルチステップ解析の実行	97
8	SIMPLISデータファイル	98
8.1	概要	98

8.2	リストデータファイル	98
8.3	エラーメッセージデータファイル	98
8.4	“出口の状態”データファイル	99
8.5	スイッチングインスタンスデータファイル	99
8.6	時間領域データファイル	99
8.7	トポロジ情報ファイル	100
8.8	既存のファイルの活用	100
8.9	POP解析のスイッチングインスタンスデータ	101
8.10	周期的動作点解析のデータ	101
8.11	周波数領域解析の印刷/プロットファイル	101
9	SIMPLIS -TXの例	102
9.1	概要	102
9.2	例: RC負荷のある整流器	102
9.3	例: 抵抗負荷のある3層整流器	104
9.4	例: 飽和のあるオペアンプ	106
9.5	例: 非安定型コンバータ	108
9.6	例: 安定型コンバータ	110
9.7	例: 可飽和インダクタ	113
9.8	例: RL負荷のあるSCR	115
10	SIMPLIS -POP	119
10.1	概要	119
10.2	POP解析に関連する文	120
	POP解析のための.POP文	120
	POP解析に関連するオプション	124
10.3	周期的動作点解析の概要	126
	POP解析中の時間変数	128
	POPは時間変化ソースをどのように扱うか	128
	POP解析のためのPOP_SHOWDATAオプション	131
10.4	POP解析ツールの適用例	133
11	SIMPLIS -FX	137
11.1	概要	137
11.2	AC解析に関連する文	139
	.AC解析文	139
	AC解析に関連するオプション文	140
	小信号ACソースを定義する文	140
11.3	小信号AC解析の概要	141
	小信号ACソースの振幅	142
	小信号ACソースの位相遅延	143

ACソースの連続領域のサンプル波形	143
ACソースの離散領域のサンプル波形	144
連続領域と離散領域の違い	145
時間変化ソースのAC解析動作	146
過渡およびPOPでのACソースの動作	147
AC解析ツールの適用例	147
12 アドバンストデジタルコンポーネント	150
12.1 概要	150
主な利点	150
新しいデジタル機能	150
アドバンストデジタルコンポーネント	151
クラシックコンポーネント	152
クラシックコンポーネントとアドバンストデジタルコンポーネントの類似点 ..	152
クラシックコンポーネントとアドバンストデジタルコンポーネントの違い	152
新しいアドバンストデジタルコンポーネントを活用するための戦略	153
単純なデモ回路	154
12.2 アドバンストコンポーネントのリファレンス	154
はじめに	154
一般的な挙動	154
利用可能な部品の概要	159
D型フリップフロップ	161
SET/RSTのあるD型フリップフロップ	163
S/Rフリップフロップ	166
SET/RSTのあるS/Rフリップフロップ	168
J/Kフリップフロップ	171
SET/RSTのあるJ/Kフリップフロップ	173
Toggleフリップフロップ	176
SET/RSTのあるToggleフリップフロップ	178
ANDゲート	181
NANDゲート	182
ORゲート	183
NORゲート	184
Exclusive-ORゲート	186
コンパレータ	187
バッファ	188
インバータ	190
加算器	191
減算器	193
乗算器	195
除算器	197

固定小数点除算器	199
A/Dコンバータ・動作	202
A/Dコンバータ・パラメータ	204
調整可能な基準電圧のあるA/Dコンバータ・動作	207
調整可能な基準電圧のあるA/Dコンバータ・パラメータ	208
D/Aコンバータ(クロックなし)	211
デジタルパルスソース	213
デジタル信号ソース	214
非対称遅延	215
デジタルコンパレータ	217
デジタル定数	219
デジタルルックアップテーブル	219
入力定義でDon't Careを許可するデジタルルックアップテーブル	220
デジタルマルチプレクサ	221
デジタルデマルチプレクサ	223
アップカウンタ	226
ダウンカウンタ	229
アップ/ダウンカウンタ	232
D型ラッチ	235
S/Rラッチ	237
イネーブルのあるS/Rラッチ	239
データレジスタ	241
シフトレジスタ	245
シフトレジスタ(左)	248
シフトレジスタ(右)	252
シフトレジスタ(マルチビット)	255
バレルシフタ	259
1極離散フィルタ・動作	263
1極離散フィルタ・パラメータ	263
2極離散フィルタ・動作.....	264
2極離散フィルタ・パラメータ	264
PID離散フィルタ・動作	266
PID離散フィルタ・パラメータ	267

1. はじめに

1.1 概要

このマニュアルは、SIMPLIS (SIMulation for Piecewise-LInear System)シミュレーションパッケージの詳細な参考資料を提供します。このソフトウェアパッケージの理解をより深めたい人を対象としています。

SIMPLISは、スイッチング電源のシミュレーションと解析用に特別に設計されたコンピュータソフトウェアパッケージです。典型的なスイッチング電源システムでは、トランジスタとダイオードがスイッチとして機能し、システムを線形回路トポロジの周期的なシーケンスで特徴付けることができます。このようなシステムの反復的な区分線形構造を利用することにより、SIMPLISは効率的かつ正確な方法でシミュレーションを実行できます。

単純な構文エラーに起因する潜在的な問題を回避するために、SIMPLISはユーザが提供した入力ファイルの構文を自動的にチェックします。入力ファイルで構文エラーが検出された場合、エラーメッセージをファイルに記録して、ユーザがエラーを検査して修正できるようにします。この機能により、ユーザはエラーを迅速かつ効率的に検出および修正できます。

SIMPLIS-TXは、2パスの時間領域シミュレータです。シミュレーション実行の最初のパスでは、スイッチングインスタンスでのシミュレーションの状態に関連するデータのみが保存されます。ポストシミュレーション処理実行と呼ばれる2番目のパスでは、最初のパスで生成されたデータから詳細な波形情報が再構築されます。これら2つの個別の操作はSIMPLISによって自動的に実行されるため、ユーザインターフェイスレベルでは区別できません。したがって、SIMPLISは、ユーザにとってワンパスシミュレータとして見えます。この内部の2段階シミュレーション技術は、SIMPLISのシミュレーション速度と汎用性を最適化します。

SIMPLIS-POP (周期的動作点) は、スイッチモード電源システムの定常状態の解への収束を加速する解析ツールです。状態ベクトルの変化に対するスイッチモードシステムのサイクル内間隔のタイミング変動を考慮することにより、POP解析ツールは定常状態の解を正確に計算します。

SIMPLIS-FXは、スイッチング電源の解析用に開発された特別な小信号周波数領域アナライザです。ある周波数範囲で、時間領域の小さな摂動に対する回路の応答を計算し、高速フーリエ解析技術を使用することにより、状態空間の平均等価回路を使用する必要がなくなります。SIMPLIS-FXは、時間領域シミュレーションに使用されるのと同じ回路図から周波数応答を正確に計算します。

2. 入力ファイルの構成

2.1 概要

SIMPLISは、単一のテキスト入力ファイルを使用して以下を定義します。

1. 回路を形成する相互接続とコンポーネント
2. 解析に適用するオプション
3. 実行する特定の解析

この章では、3ページの「[入力ファイルの一般規則](#)」で入力ファイルの一般規則を定義し、9ページの「[入力ファイルの構成](#)」で入力ファイルの構成について説明します。

3. デバイス文

3.1 概要

デバイス文の形式

デバイス文は、回路内でのデバイスの接続方法を定義し、個々のデバイスパラメータの値をリストします。デバイスがデバイスモデルまたは何らかの初期条件を必要とする場合、そのような情報もデバイス文で定義されます。デバイス文の形式は次のように定義されます。

```
DeviceName NodeName {Values|ModelName} [InitConds]
```

ここで

DeviceName	有効なデバイス名
NodeName	一連の有効なノード名
Values	値を表す浮動小数点のエントリまたは一連のパラメータ代入
ModelName	互換性のある有効モデル名。シンボル{Values ModelName}は、ValuesとModelNameが相互に排他的であることを意味します。デバイスにモデル名が必要な場合、デバイス文でValuesは指定されず、逆も同様です。
InitConds	一連の有効な初期条件の仕様。一部のデバイスのみが初期条件を必要とするため、シンボル[InitConds]は、InitCondsフィールドの存在はオプションであることを意味します。

各デバイス文の個々のフィールドは、この章で示されている順序で正確に記述する必要があります。順序が異なると、SIMPLISは文を誤って解釈したり、エラーメッセージを生成しま

す。

ノード名

回路内の各ノードには一意の名前を割り当てる必要があります。有効なノード名は、正の整数またはゼロです。ノード0は従来、グラウンドノードを表すために予約されていることに注意してください。次の点にも注意してください。

1. ユーザがあるグラウンドノードに対して特定のノードの電圧を検査する必要がある限り、SIMPLISはシステムにノード0が存在することを必要としません。
2. SIMPLISでは、システム内のすべてのノードを相互に接続する必要はなく、システムは絶縁されたサブシステムを持つことができます。ただし、ユーザがSIMPLISに2つの電氣的に絶縁されたノード間の電圧を決定するように指示すると、エラーメッセージが生成されます。

電圧と電流の極性の規則

この章で説明する回路素子のほとんどは、2端子素子です。一般的な2端子素子の場合、下図に示すように、正ノードn+と負ノードn-があります。

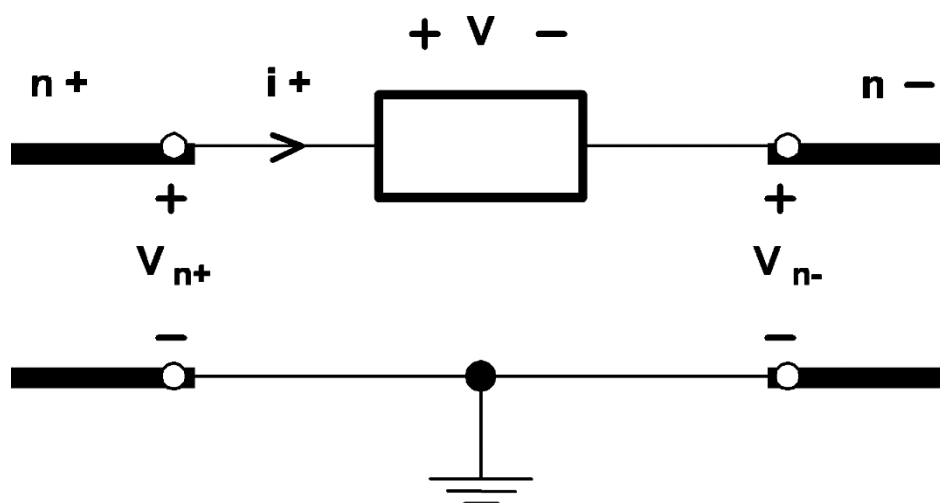


図3.1 2端子素子の電圧と電流の関連付けの定義

このマニュアルで2端子素子の両端の電圧について言及する場合は常に、負のノードの電圧に対して正のノードで測定した電圧を指します。たとえば、図3.1における一般化された2端子デバイスの電圧は

$$V = V_{n+} - V_{n-}$$

のように、n+端子からn-端子までを測定したものです。

このマニュアルで2端子デバイスを流れる電流に言及している場合は、正のノードから素子を通って負のノードに向かう方向で測定された電流を指します。したがって、正の電流は、正のノードへの正味の電荷の流れがあり、2端子素子を通り、負のノードから出ることを示します。

電圧と電流のこれらの符号規則により、正の電圧・電流積は、電力が対応する2端子素子に瞬時に流れていることを示しており、電圧ソース、電流ソース、抵抗、または他の2端子素子であるかどうかに関係ありません。

パラメータの代入

デバイスの初期条件などのパラメータは、パラメータ代入と呼ばれる形式で入力されます。形式は次のとおりです。

```
KEYWORD=value
```

ここで

KEYWORD	対応するキーワードであり、パラメータの説明的な名前を表します。
=	等号
value	パラメータに代入される値

入力ファイルを読みやすくするために、等号(「=」)の前後に空白文字を表示できます。さらに、等号が現在の行の最後の有効な文字である場合、行継続文字であるプラス記号(「+」)を使用して、値を次の行に表示できます。

制御デバイス

4つの制御されたソース、単純なトランジスタスイッチ、および単純なスイッチは、それぞれデバイスの外部にある制御変数によって制御されます。制御変数には3つの種類があります。

1. 同じ回路内の2つのノード間の電圧差
2. 同じ回路内の制御デバイスの正および負のノード間の分岐電圧
3. 同じ回路内の制御デバイスを流れる電流

制御デバイスは、次のデバイスタイプのいずれかです。

線形抵抗

線形インダクタ

線形キャパシタ

すべての種類の独立電圧ソース
すべての種類の独立電流ソース
すべての4種類の線形制御ソース
単純なトランジスタスイッチ
単純なスイッチ
区分線形抵抗
区分線形インダクタ
区分線形キャパシタ

4. モデル文

4.1 概要

単純なデバイスの場合、デバイスのモデル化に必要なパラメータの数は比較的少なく、パラメータはデバイス文と簡単に混合できます。たとえば、線形抵抗の抵抗、インダクタのインダクタンス、およびキャパシタのキャパシタンスは、それぞれデバイス文で定義されます。単純なスイッチ、区分線形素子、単純な論理ゲートなどのデバイスの場合、デバイスの性能を記述するために多数のパラメータが必要です。このような場合、モデル文は、モデルパラメータを定義する便利で体系化された方法を提供します。

モデル文を使用することには、さらに2つの利点があります。多くの場合、検討対象のシステム内の複数のデバイスが同じモデルパラメータを持つ場合があります。このような場合、1つのモデル文で、同じ種類のすべてのデバイスのモデルパラメータを提供できます。この方法のもう1つの利点は、複数のデバイスが同じデバイスモデルで記述されている場合、すべてに共通のモデル文を変更することにより、これらすべてのデバイスのモデル特性を同じ場所に変更できることです。

典型的なモデル文は、次の文例で表すことができます。

```
.MODEL mname mtype param param ...
```

ここで

<code>.MODEL</code>	6文字のキーワード“.MODEL”
<code>mname</code>	5ページの「モデル名とサブ回路名」 で説明されている有効なモデル名。モデル名は、一般の回路内で一意でなければなりません。名前が一般の回路でモデル名として使用されている場合、同じ一

般の回路でサブ回路名として使用することはできません。逆も同様です。

mtype	SIMPLISでサポートされているモデルタイプの1つを表すキーワードです。SIMPLISによって認識されるデバイスモデルのリストを以下の表に示します。
param	13ページの「 パラメータの代入 」に示されている形式のパラメータの代入です。

SIMPLISで使用されるデバイスモデルタイプ

モデルタイプ 説明

VCSW	単純な電圧制御スイッチ
ICSW	単純な電流制御スイッチ
VCQPOS	単純な電圧制御トランジスタスイッチ
VCQNEG	
ICQPOS IC-	単純な電流制御トランジスタスイッチ
QNEG	
VPWLR IP-	区分線形抵抗
WLR	
PWLL	区分線形インダクタ
PWLC	区分線形キャパシタ
INV	単純な論理ゲート, インバータ
COMP	単純な論理ゲート, コンパレータ
XOR	単純な論理ゲート, Exclusive OR
ORk	単純な論理ゲート, k-input OR, kは整数, $2 \leq k \leq 9$
NORk	単純な論理ゲート, k-input NOR, kは整数, $2 \leq k \leq 9$
ANDk	単純な論理ゲート, k-input AND, kは整数, $2 \leq k \leq 9$
NANDk	単純な論理ゲート, k-input NAND, kは整数, $2 \leq k \leq 9$
SRFF	単純な論理ゲート, セット-リセット フリップフロップ
CLK_SRFF	Clocked論理ゲート, Clocked セット-リセット フリップフロップ
CLK_JKFF	Clocked論理ゲート, Clocked JK フリップフロップ

CLK_DFF	Clocked論理ゲート, Clocked D フリップフロップ
CLK_TFF	Clocked論理ゲート, Clocked トグルフリップフロップ
LATCH	ラッチ

キーワード「.MODEL」、モデル名、およびモデルタイプは、示されているとおりの正確な順序で入力する必要があります。これらの3つのフィールドの後には、それぞれがパラメータ代入で構成されたフィールドがいくつかあります。パラメータ代入の実際の数、モデルの種類によって異なります。パラメータ代入の数は、モデルタイプに必要な数と正確に等しくなければなりません。パラメータ代入が余分または不足していると、エラーメッセージが表示されます。ただし、パラメータ代入フィールドのセット内では、フィールドは任意の順序で記述できます。たとえば、次の2つの文はどちらも受け入れられます。

```
.MODEL S1 VCSW TH=2 HYSTWD=2U RON=10m
+ ROFF=10MEG LOGIC=POS
```

または

```
.MODEL S1 VCSW RON=10m ROFF=10MEG
+ TH=2 HYSTWD=2U LOGIC=POS
```

SIMPLISでのパラメータ代入は、デフォルト値を想定していません。したがって、各必須パラメータには、.MODEL文で適切な値を代入する必要があります。

5. サブ回路の定義

5.1 概要

SIMPLISがサポートする基本的なデバイス要素（線形抵抗、線形インダクタ、線形キャパシタ、独立した電圧ソースと電流ソース、相互インダクタンス、4種類の線形制御ソース、理想的なトランス、単純なスイッチ、単純なトランジスタスイッチ、PWL抵抗、PWLインダクタ、PWLキャパシタ、および単純な論理ゲート）は非常に汎用性が高く、さまざまな電子デバイスと回路をモデル化するためのビルディングブロックとして使用できます。たとえば、NPNバイポーラトランジスタは、区分線形Ebers-Mollモデルでモデル化できます。ここで、接合ダイオードに区分線形抵抗を使用し、2つの電流制御電流ソースで電流伝導をモデル化します。さらに、3つの線形抵抗を挿入して、図5.1に示すように接触抵抗をモデル化できます。

図5.1 (a)に示す物理トランジスタには3つのノードがあります。一方、図5.1 (b)の対応するモデルには6つのノードがあり、物理トランジスタの3つのノードと3つの内部ノードに対応しています。図5.1 (a)の物理トランジスタが回路全体に1回しか現れない場合、図5.1 (b)のモデル回路を入力ファイルにそのまま入力して定義できます。回路に同じデバイスの複数のインスタンスが含まれている場合、図5.1 (b)のモデル回路を繰り返し定義する必要があります。出現

ごとに、次のことを確認するように注意する必要があります。

1. ノード名は、他の同様の定義と比較して一意です。
2. 7つの基本素子の各定義で、一意の名前が付けられています。

明らかに、回路のサイズが大きくなると、これは非常に退屈でエラーが発生しやすい作業になります。

図5.1 (b)に示されているネットワークが代わりにサブ回路として定義されている場合、回路内の物理トランジスタのすべての出現は、次の3つのステップで記述できます。

1. 同じタイプのトランジスタの各インスタンスに一意のデバイス名を付けます。
2. 各トランジスタの3つの端子に一意のノード名を定義します。
3. 関連するトランジスタのタイプを定義するサブ回路を参照します。

図5.1 (b)のモデル内の7つのデバイス要素に一意の名前を付け、モデル内の3つの内部ノードに一意の名前を付けるに伴う退屈でエラーが発生しやすい手順は、SIMPLISのサブ回路機能によって自動的に処理されます。SIMPLISのサブ回路機能の定義と使用方法については、この章で詳しく説明します。

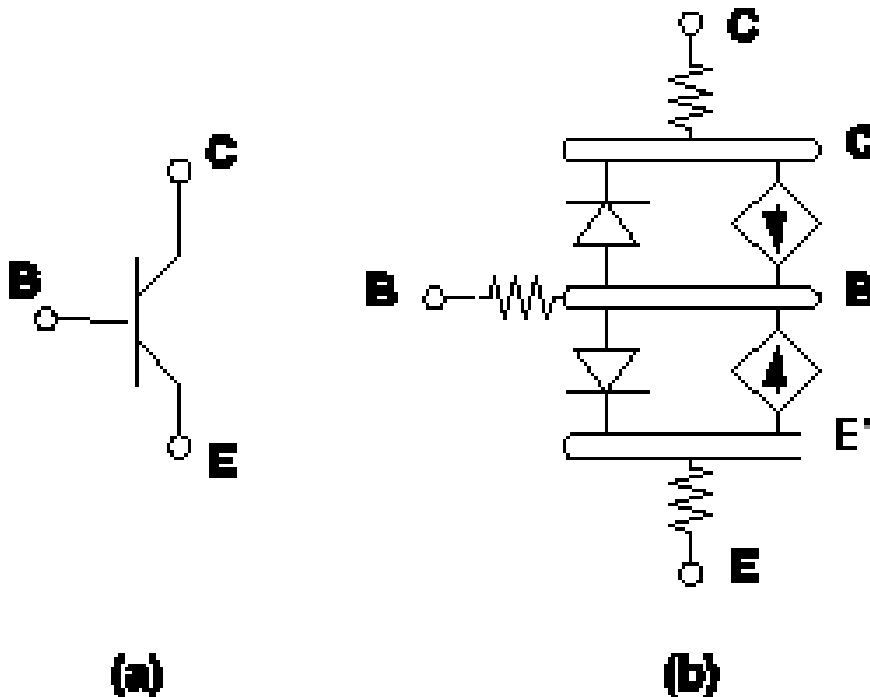


図5.1 NPNトランジスタのEbers-Mollモデル：(a) 回路記号，(b) Ebers-Mollモデル。注：B', C', およびE'はモデリングの目的で導入された内部ノード

6. 制御文

6.1 概要

これまでに説明されているデバイス文、モデル文、およびサブ回路定義文は、すべて回路の定義に関連しています。実行される解析のタイプと、SIMPLISによって生成される出力データのタイプを制御する追加の文があります。これらのタイプの文は、回路の定義とは関係ありません。これらは、まとめて制御文と呼ばれます。

すべての制御文は、文の最初の文字としてピリオド(「.」)で始まります。ただし、ピリオドで始まるすべての文が制御文であるとは限りません。例外は、.MODEL文、.END文、.SUBCKT文、および.ENDS文です。制御文は、次のタイプのいずれかに分類できます。

1. オプション
2. 初期条件
3. 変数の印刷
4. 解析

すべてのタイプの制御文は、メイン回路の定義の範囲内に記述できますが、初期条件の設定と変数の印刷に関連する制御文のみがサブ回路の定義の範囲内に記述できます。一般に、制御文は任意の順序で記述できます。この規則の1つの例外は、解析文の順序です。これは、SIMPLISがさまざまな解析を実行する順序を決定します。

7. SIMPLISの実行

7.1 概要

このマニュアルの対象となるSIMPLISのバージョンは、SIMetrix環境内でのみ使用できます。

通常、回路はSIMetrix回路図エディタを使用して入力されます。このエディタは、このマニュアルの他の部分で説明されている構文の詳細の多くを処理します。詳細については、*SIMetrix User's Manual/Schematic Editor*を参照してください。

手作業で作成したネットリストまたは別の回路図入力ツールを使用して、SIMPLISを実行することもできます。これはSIMetrix環境内で行う必要があります、次のセクションでその方法を説明します。

SIMPLISの実行に役立つSIMetrix回路図エディタのいくつかの機能も、便宜上ここで繰り返

し説明します。

7.2 SIMetrix回路図でSIMPLISの実行

最初に「SIMPLISモード」で回路図を入力する必要があります。SIMPLISモードを選択するには、回路図ウィンドウから**File | Select Simulator**メニューを選択し、SIMPLISを選択します。SIMetrixモードでいくつかのコンポーネントを既に追加している場合、非互換性の警告が表示される場合があります。これは、配置されたコンポーネントの一部がSIMPLISで機能しないことを意味します。これらは、適切な代替品に置き換えるか、再入力する必要があります。

回路図が入力されたら、**Simulator | Choose Analysis...**を使用して、選択した解析モードを選択します。

これで、F9キーまたはメニューの**Simulator | Run**をクリックして、シミュレーションを実行できます。

ネットリスト行の追加

回路図エディタの**Simulator | Choose Analysis...**メニューを使用して選択された解析モードは、回路図のシミュレータコマンドウィンドウにテキスト形式で保存されます。必要に応じて、これを直接編集することができます。シミュレータコマンドウィンドウに入力されたテキストと**Simulator | Choose Analysis...**ダイアログの設定は同期しているので、2つの方法を自由に切り替えることができます。

シミュレータコマンドウィンドウを開くには、回路図を選択してF11キーを押します。これはトグルアクションであり、もう一度押すと非表示になります。**Choose Analysis**ダイアログボックスを使用して既に解析モードを選択している場合は、シミュレータコントロールが既に表示されています。

このウィンドウには、右クリックで選択されるポップアップメニューがあります。一番上の項目**“Edit file at cursor”**は、カーソルによって示されたファイル名または選択されたテキスト項目（存在する場合）で、テキストエディタを開きます。

シミュレータコマンドウィンドウのサイズは、このウィンドウと回路描画領域の間のスプリッターバーを使用して変更できます。

シミュレータコマンドだけでなく、このウィンドウに好きなものを追加できます。内容は、シミュレータに提示される前にネットリストに追加されるだけです。そのため、デバイスモデル、相互インダクタ仕様、.OPTIONコントロール、または単なるコメントを配置できます。**Choose Analysis**ダイアログは、解析コントロールと一部の.OPTIONS設定を構文解析し、場合によっては変更しますが、その他はすべてそのまま残します。

一部の回路図は、SIMPLISとSPICEベースのSIMetrixシミュレータの両方でシミュレートできます。ただし、これらのシミュレータのコマンド（SIMetrixのドキュメントでは「コントロール」と呼ばれます）は互換性がありません。このため、SIMetrixコマンドとSIMPLISコマンドは、.SIMULATORコントロールを使用して分離されます。この構文は次のとおりです。

```
.SIMULATOR SIMPLIS | SIMetrix | DEFAULT
```

SIMPLIS	次の.SIMULATORコントロールまでのすべての行は、SIMPLISモードのネットリストにのみ渡されます。
SIMetrix	次の.SIMULATORコントロールまでのすべての行は、SIMetrixモードのネットリストにのみ渡されます。
DEFAULT	次の.SIMULATORコントロールまでのすべての行は、両方のモードのネットリストに渡されます。

7.3 外部ネットリストでSIMPLISの実行

手作業または別のプログラムで作成されたネットリスト（‘入力デッキ’とも呼ばれる）でSIMPLISを実行することもできます。

SIMetrix GUIからネットリストを実行するには、コマンドシェルメニュー**SIMPLIS | Run Netlist...**を選択し、実行するファイルを選択します。これにより、ネットリストはSIMPLISに提示される前に、ネットリストプリプロセッサを通過することに注意してください。プリプロセッサは、デバイスモデルをインポートおよびパラメータ化するための追加機能を提供します。詳細については、[94ページの「ネットリストプリプロセッサ」](#)を参照してください。完全な構文的に正しいSIMPLISネットリストは、プリプロセッサによって機能的に変更されないことに注意してください。

7.4 スクリプトからのSIMPLISの実行

SIMPLISは、RunSIMPLISコマンドを使用してスクリプトから起動できます。

```
RunSIMPLIS filename
```

<i>filename</i>	SIMPLISネットリストを含むファイルの名前。フルパスが指定されていない場合、ファイル名は現在のディレクトリからの相対パスであると見なされます。ファイルの拡張子は常に指定する必要がありますことに注意してください。デフォルトは想定されていません。
-----------------	---

RunSIMPLISコマンドは、ネットリストを前処理しません。これは、PreProcessNetlistコマンドを使用して個別に実行する必要があります。[94ページの「ネットリストプリプロセッサ」](#)

を参照してください。

RunSIMPLISは、SIMPLISを起動する基本的なSIMetrixコマンドです。ただし、回路図でシミュレーションを実行すると、他の多くのアクティビティが実行されます。これには、回路図エディタで生成されたネットリストの前処理、POP解析のトリガーデバイスの置換も含まれます（119ページの「SIMPLIS-POP」を参照）。Runメニューとまったく同じ方法で回路図をシミュレートする場合は、スクリプトsimplis_runを実行する必要があります。これは、現在開いている回路図をシミュレートします。simplis_runの完全なソースはインストールCDにあります。

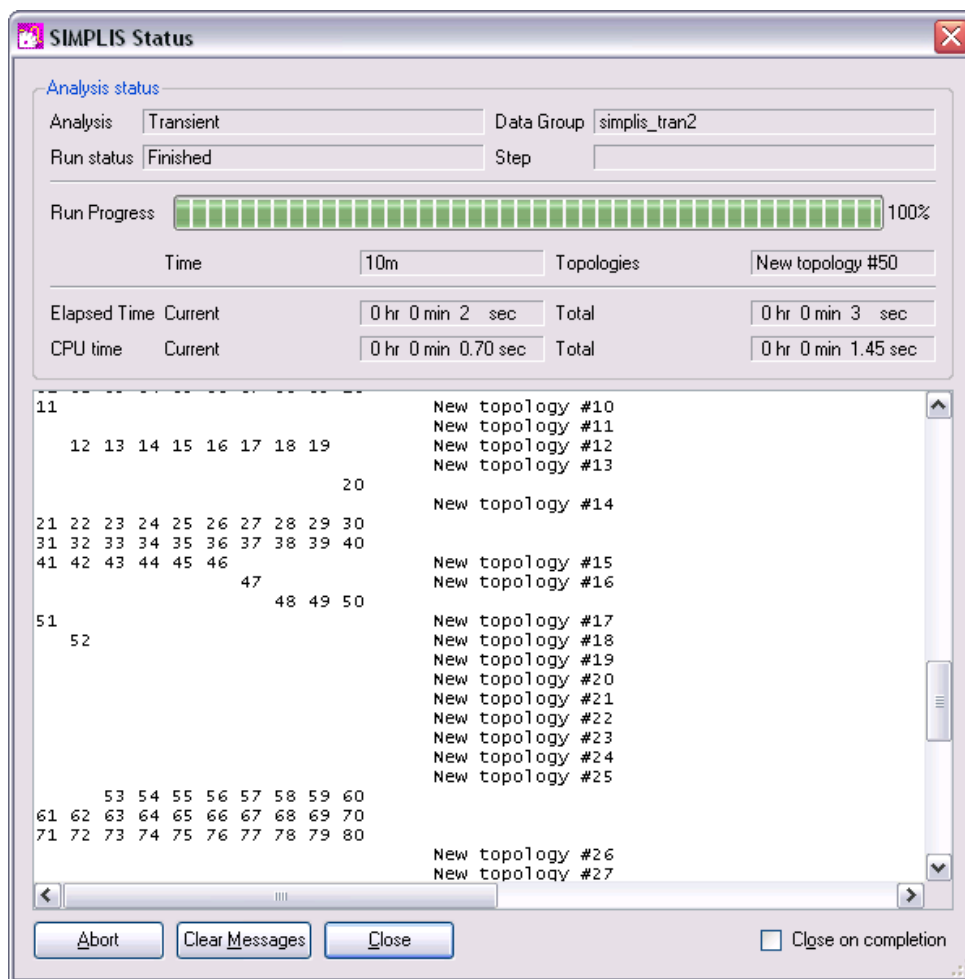
7.5 DOSプロンプトからのSIMPLISの実行

SIMetrix/SIMPLISで提供されるSIMPLISのバージョンは、コマンド（DOS）プロンプトから直接実行できません。

7.6 SIMPLISの実行

SIMPLISは別のプロセスとして実行されますが、実行中にSIMetrixと通信して、実行の進行状況に関する情報を提供します。

この進行状況は、下図に示すようにSIMPLISステータスウィンドウに表示されます。



メッセージウィンドウには、進行状況に関する詳細が表示されます。これは、ステータスウィンドウが表示されない以前のバージョンで表示されていたものと似ています。

7.7 SIMPLIS実行の中止

シミュレーションの実行中はいつでも、シミュレーションを中止できます。コマンドシェルメニューの**SIMPLIS | Abort Run**を選択するか、SIMPLISステータスウィンドウの**Abort**ボタンを押します。これにより、実行中の次の適切な時点でSIMPLISが中止されますが、すぐに中止されるとは限りません。SIMPLISには再開機能がないことに注意してください。

7.8 SIMPLISによる自動プログラム停止

入力回路には、SIMPLISで解決できないスイッチングの競合が含まれている場合があります。たとえば、SIMPLISインバータの入力ノードと出力ノードを接続すると、インバータが動作する有効な論理出力状態を特定できない状況が発生します。このようなスイッチングの競合

が発生すると、シミュレーションは無限ループに入り、時間を進められません。200回の試行で正しい動作状態の特定に成功しない場合、SIMPLISはメッセージウィンドウに次のようなメッセージを出力し、シミュレーションを中止します。

```
Unable to find a starting operating point!!
```

または

```
At t = 1.2345e-06, it is unable to find the correct state of  
operation for some device.
```

```
For example, check the following devices:
```

```
!R1, !R2, and !R3
```

このようなメッセージが表示される場合は、入力ファイルとSIMPLISによって生成された適切なデータファイルを注意深く調べて、スイッチングの競合の原因を特定する必要があります。少し異なる状況では、スイッチングロジックは適切に定義されていますが、シミュレーションが非常に高速で繰り返しスイッチングされるような方法で、いくつかの回路要素が接続されている場合があります。2つのスイッチングイベント間の時間間隔が実際のシミュレーション時間変数に比べて無視できるほど短い場合、SIMPLISはシミュレーションを中止し、次のようなメッセージを表示します。

```
No advance in the time variable
```

```
at t = 3.1425e-05 sec. !!
```

このエラーメッセージの例は、シミュレーションで時間変数 t が 3.1425×10^{-5} 秒に達すると、 3.1425×10^{-5} 秒と比較して無視できる時間内に2つの連続したスイッチングイベントが発生したことを意味します。ある数値は別の数値より16桁小さい場合、無視できると見なされます。この場合も、入力ファイルとSIMPLISによって生成された適切なデータファイルを調べて、問題の原因を特定する必要があります。

7.9 ネットリストプリプロセッサ

概要

ネットリストプリプロセッサは、実際にはSIMPLISではなくSIMetrixの一部です。ただし、SIMetrix (SPICE) ネットリストで使用できますが、もともとSIMPLISで使用するために開発されたので、ここで説明します。ネットリストプリプロセッサは、シミュレータ自体では提供されない追加機能を提供します。これらの機能は次のとおりです。

1. モデルライブラリで未解決のサブ回路を検索し、ネットリストに追加します。
2. パラメータ化された式を評価します。
3. パラメータ化された定義から静的サブ回路を構築します。

4. SIMPLIS互換性のために、グローバルに定義されたサブ回路をローカライズします。

パラメータ化システムには、.IFコントロールおよび.WHILEコントロールを使用する条件機能とループ機能が含まれています。

プリプロセッサの起動

プリプロセッサは、スクリプトコマンドPreProcessNetlistを使用して起動されます。構文は次のとおりです。

```
PreProcessNetlist [/inAppend extraInputLines] [/simulator
SIMPLIS|SIMatrix] [/mc]
+ [/importglobals] [/params paramlist] [/mcseed seed] [/rawdeck]
[/mclogfile mclog
+ inFile outFile
```

ここで

<i>inFile</i>	処理する入力ファイル名
<i>outFile</i>	結果を受け取るファイル
<i>extraInputLines</i>	入力ファイルに追加される行。各行はセミコロン“;”で区切られます。
<i>/mc</i>	存在する場合、モンテカルロ分布関数を有効にします。存在しない場合、これらの関数は1を返します
<i>/importglobals</i>	存在する場合、フロントエンドのグローバルグループで定義された値がインポートされます。これにより、スクリプト内の値をプリプロセッサに渡すことができます。
<i>paramlist</i>	ローカルパラメータを定義するname=valueというペアのセミコロン区切りリスト。 たとえば Let global:param1 = 100.0 param1の値は、前処理されたネットリストで利用可能になります。
<i>seed</i>	モンテカルロシード値を割り当てます。/mcが必要です。
<i>/rawdeck</i>	これにより、拡張子が.rawdeckの中間ファイルが生成されます。このファイルには、モデルとサブ回路がライブラリから収集された後でパラメータ式の置換前の出力が含まれます。これは、暗号化されたパラメータ化モデルの構築を目的としています
<i>mclogfile</i>	指定した場合、モンテカルロログファイルの生成を有効にして、

モンテカルロ分布関数の影響を受けるコンポーネントの値をリストします。/mcが必要です。

/simulatorスイッチを使用すると、ネットリストが対象とするシミュレータの指定が可能になり、ライブラリ検索と.SIMULATORコントロールの効果に影響します。デフォルト値は「SIMPLIS」です。

SIMPLISシミュレーションが開始されると、PreProcessNetlistコマンドが自動的に呼び出されます。通常、ユーザはこれを意識する必要がありません。

ライブラリ検索

SIMetrixライブラリは、サブ回路コールで参照されていて入力ネットリストに存在しないデバイスを検索します。この検索では、SIMPLISモデルとして明示的に指定されたデバイスのみが認識されます。（「/simulator SIMetrix」がPreProcessNetlistコマンドラインで指定されていない限り）。SIMPLISモデルは、.SIMULATORコントロールを使用してモデルファイルの中で識別されます。このコントロールの構文は次のとおりです。

```
.SIMULATOR SIMPLIS | SIMetrix
```

.SIMULATORコントロールは、次の.SIMULATORコントロールまでに表示されるすべてのモデルに適用されます。

ライブラリにデバイスが見つかった場合、そのテキストは入力ネットリストに表示されているかのように入力されます。現在、このライブラリ検索ではサブ回路デバイスのみが置換されます。.MODELを使用して定義されたすべてのプリミティブデバイスは、入力ネットリストで定義する必要があります。

また、現在、SIMetrixグローバルライブラリのみが検索されることに注意してください。.LIBコントロールはサポートされていません。

パラメータ

パラメータは、次の構文を持つ.VARコントロールを使用して定義できます。

```
.VAR parameterName={ parameterExpression }
```

parameterNameには任意の英数字列を使用できますが、文字またはアンダースコアで始まる必要があります。parameterExpressionは、スクリプトリファレンスマニュアルに詳述されている有効なSIMetrix式です。文字列式を使用できます。parameterExpressionは、以前の.VARコントロールで定義されたパラメータを参照することができます。

.VARコントロールに続くネットリストの任意の部分には、中括弧で囲まれたパラメータ式を含めることができます。たとえば次のとおりです。


```
.VAR resval = { 1K }  
R1 1 2 {resval * 2}
```

.VARは、サブ回路の定義内で使用できます。その場合、ローカルスコープがあります。これは、パラメータの定義がそのサブ回路の定義内でのみ有効であることを意味します。このスコープは、ネストされたサブ回路の定義によって継承されないことに注意してください。ただし、最上位で定義されたパラメータにはグローバルスコープがあります。

サブ回路にパラメータを渡す

パラメータは、サブ回路コールを介してサブ回路に渡すことができます。構文は次のとおりです。

```
Xxxx nodes subname vars: param1=value1 param2=value2 ...
```

value1、*2*は、定数または中括弧で囲まれた式です。式は、以前の.VARコントロールで定義された値と、X行自体がサブ回路定義内にある場合、その定義に渡される値を使用できます。ただし、式は、同じX行で他の渡されたパラメータを参照することはできません。たとえば、*value2*は*param1*を参照できません。

条件行

.IFコントロールは、指定された条件が満たされた場合にのみ出力に渡される行を定義するために使用できます。構文は次のとおりです。

```
.IF {expression}  
netlist lines  
...  
[.ELSE  
netlist lines  
...]  
.ENDIF
```

*expression*が非ゼロの値に置換される場合、.ELSEまでの行が出力されます。そうでない場合、.ELSEと.ENDIFの間の行が出力されます。.ELSEはオプションです。.IF/.ELSE/.ENDIFは、任意のレベルにネストできます。

*expression*は、任意の有効な算術式です。これは、サブ回路コールを介して渡されたものを含む、以前に定義されたパラメータを参照することができます。

ループ

.WHILEコントロールを使用して行の繰り返し列を出力するように、プリプロセッサに指示す

ることができます。構文は次のとおりです。

```
.WHILE {expression} [max_loop_count]
netlist lines
...
.ENDWHILE
```

WHILEと.ENDWHILEの間のブロックは、`expression`が最大`max_loop_count`回まで非ゼロである限り繰り返されます。`max_loop_count`を省略すると、デフォルト値の100が使用されます。`max_loop_count`は、無限ループがユーザのディスク容量をいっぱいにならないようにするための、安全対策として意図されています。

7.10 モンテカルロ解析とマルチステップ解析の実行

SIMetrix環境には、複数のSIMPLIS解析を実行する機能があります。パラメータ値をスイープし、モンテカルロのパラメータをランダムに代入する機能が提供されています。

詳細については、*SIMetrix User's Manual/SIMPLIS Analysis Modes/Multi-step and Monte Carlo Analyses*を参照してください。

8. SIMPLISデータファイル

8.1 概要

SIMPLISが解析を実行すると、入力ファイルと同じディレクトリに追加のデータファイルが作成されます。これらのファイルは、主にSIMPLIS自体が使用するために作成されます。誰もがこの章を参照して、これらのデータファイルが何を達成するかを理解し、さまざまな解析中にこれらのデータファイルを利用できるようにすることをお勧めします。[100ページの「既存ファイルの活用」](#)セクションは、時間領域シミュレーションの実行に特に役立ちます。

以下の説明では、入力ファイルの名前は「XXXX」であると想定されています。SIMPLISによって生成された各データファイルの名前は「XXXX.extension」であり、「extension」はデータファイル固有の文字列です。

SIMetrixで提供されるバージョンのSIMPLISは、シミュレーションデータをSIMetrixに送信し、SIMetrixはそれを保存します。このデータは通常、TEMPDATAディレクトリにある.sxdatファイルに保存されます。

9. SIMPLIS -TXの例

9.1 概要

この章では、ユーザがSIMPLISの入力ファイルの構文を理解し、SIMPLIS-TXのさまざまな機能を理解するのに役立つ例を示します。SIMPLIS-POPとSIMPLIS-FXの使用例は、それぞれ第10章と第11章に記載されています。これらの例で検討したシステムは、パワーエレクトロニクスで遭遇するさまざまなシステムを表しています。説明を簡潔にするために、各例は意図的に小さく制限されています。同じ理由で、この例のデバイスモデルは単純に保たれていますが、機能的には適切です。これらの単純なデバイスモデルは、より複雑なモデルを作成するための基本的なビルディングブロックとして機能します。

この章のすべての例は、ディレクトリroot\Work\Examples\SIMPLIS\Manual_Examplesの下に「すぐに実行可能な」SIMetrix回路図としてインストールされます。この章に示す入力ファイルは、実際にはSIMetrixによって作成されたシミュレーションデックですが、プロット文は削除されています。

10. SIMPLIS -POP

10.1 概要

スイッチング区分線形システムの解析では、定常状態の解が重要です。たとえば、安定化スイッチング電源システムのライン/負荷レギュレーションの検討では、関連情報はライン/負荷条件の範囲にわたる定常状態の負荷電圧です。新しいライン/負荷条件の下でシステムが新しい定常状態にどのように落ち着くかに関する過渡的な情報は、それ自体は重要ですが、このような検討の焦点ではありません。このような検討を実施するために、システムが新しいライン/負荷条件の下で新しい定常状態動作に落ち着いた後に、負荷電圧が測定されます。システムの減衰と安定化回路によっては、ライン/負荷条件の変更後にシステムが数百から数千のスイッチングサイクルを経て新しい定常状態動作に落ち着く場合があります。力ずくのシミュレーションでこのような検討を実行することは可能ですが、時間がかかる場合があります。したがって、実際の過渡状態を経ることなく、システムの定常状態動作条件への収束を「加速」できる特別な解析ツールが必要です。

スイッチング区分線形システムの定常状態動作が重要である別の例は、安定化スイッチング電源システムの負荷過渡現象の検討です。このような検討では、負荷の変化に応じてシステムで発生する過渡現象がモニタされます。この検討をシミュレーションで実行する場合、負荷の変更を開始する前に、回路シミュレーションをシステムの初期の定常状態解に初期化する必要があります。定常状態の解を使用してシミュレーションを開始しないと、得られる過渡現象は実験室で測定される過渡現象とは異なります。

スイッチング区分線形システムの定常状態解の計算を高速化するための特別なアルゴリズム

が、特別な解析ツールとしてSIMPLISに組み込まれています。数学的により挑戦的でより計算集約的ですが、スイッチングシステムの定常状態解の計算は、概念的には、DC励起のみで非スイッチングシステムのDC動作点を見つけることに似ています。どちらの場合も、外部励起がない状態でシステムの動作を見つけることに興味があります。「動作点解析」という用語は、DC動作点解析の名前として伝統的かつ広く使用されています。それで、周期的に駆動されるまたは自励発振するスイッチング区分線形システムにおいて、定常状態解の計算を高速化するためのこの特別なアルゴリズムに、「周期的動作点解析」(POP)という名前が与えられています。

SIMPLISの周期的動作点解析ツールは、周期が同等の1つ以上の周期ソースによって自励発振するまたは駆動されるスイッチング区分線形システムにおいて、定常状態解への収束を高速化できます。このような解析を呼び出すには、ユーザは入力ファイルに数行を追加するだけです。120ページの「POP解析に関する文」では、周期的動作点解析に関連する入力文の形式について詳しく説明しています。周期的動作点解析中に何が起こるかについては、126ページの「周期的動作点解析の概要」で説明します。このセクションを読んだ後、ユーザはこの解析ツールの内部動作を理解し、解析ツールをより生産的に使用できるようになります。133ページの「POP解析ツールの適用例」の例は、POP解析の閉ループ安定化スイッチング電源システムへの適用を示しています。

11. SIMPLIS -FX

11.1 概要

小信号をシステムに加えて、その平衡の周りの小信号動作を調査します。結果のデータは、さまざまなグラフィカル形式（半対数ボード線図や極プロットまたは逆極プロットなど）で簡潔に表示できるため、小信号解析はしばしば周波数領域で実行されます。これらのグラフィカルなプロットから、インピーダンス、伝達関数、安定性情報などのさまざまな特性を決定できます。

小信号周波数領域解析をスイッチング区分線形システムに適用することは、非常に大きなチャレンジです。通常、ラプラス変換またはラプラス変換等価回路解析は、非スイッチングシステムに適用されて小信号周波数領域特性を抽出しますが、スイッチング区分線形システムには、固有のスイッチング動作のため、簡単に適用できません。SIMPLIS-FXは、スイッチング区分線形システムの解析用に特別に設計された小信号周波数領域アナライザです。この解析は、回路の平均化や等価な非スイッチングモデルの導出に頼ることなく、スイッチング区分線形システムの時間領域シミュレーションに基づいています。スイッチング動作を削除して小信号周波数領域特性を導出するのではなく、SIMPLIS-FXは小信号周波数領域特性の計算にスイッチング動作を含めます。

SIMPLIS-FXの計算面は面倒で退屈かもしれませんが、それが定式化される数学的基礎は非

常に簡単です。まず、SIMPLIS-POPを使用して、スイッチング区分線形システムの周期的な動作点/軌跡を計算します。この動作点/軌跡は、システムの大信号平衡を表します。その後、SIMPLIS-FXを適用して、大信号平衡の周りのシステムの小信号挙動を調査できます。小信号周波数領域解析は、実際には離散的な解析周波数での一連の解析です。各解析周波数で、解析の手順は次のように要約できます。

1. 調査中のシステムに電圧/電流ソースの形で小信号刺激を適用します。これらの小信号刺激は小信号ACソースと呼ばれ、その波形は時間領域の正弦波です。各解析周波数で、すべての小信号ACソースの周波数は同じ値の解析周波数に設定され、その振幅は非常に小さな値に設定されます。すべての小信号ACソースの周波数は解析周波数に設定されているため、解析周波数は励起周波数とも呼ばれます。
2. 次に、小信号からの摂動の下でシステムの平衡が計算されます。この新しい平衡は、周期的動作点（POP）解析によって計算された大信号平衡からわずかに離れた距離にあります。大信号平衡とは明らかに異なります。大信号平衡は、POP解析で決定されるように、周期的動作周波数またはスイッチング周波数に等しい周波数で、周期的です。一方、この新しい平衡は、解析周波数と周期的動作周波数の間の最大公約数に等しい周波数で、周期的です。
3. 次に、新しい平衡にフーリエ解析を適用して、解析周波数でのシステムの小信号応答を抽出します。

SIMPLIS-FXは時間領域シミュレーションに基づいているため、次の2つの条件が満たされていれば、任意の構造（トポロジ）、任意の動作モード、任意の制御方式のスイッチング区分線形システムを、固定または可変のスイッチング周波数で処理できます。

1. システムは、SIMPLIS-TXを介して時間領域でシミュレーションすることができる。
2. SIMPLIS-POPは、システムの周期的な動作点/軌道を正常に計算することができる。

たとえば、複数のスイッチ、複数の出力、複数のフィードバックループのコンバータは、SIMPLIS-FXで簡単に処理できます。これは、スイッチ、出力、またはフィードバックループの数に関する仮定や制限なしに、一般的なスイッチング区分線形システム用に解析が構築されるためです。

一部のスイッチング電源は、出力電圧のライン/負荷安定性を改善するために、非常に高いDCゲインを持つように設計されています。これらのシステムのループゲインの測定は、システムの高いDCゲインが開ループシステムを駆動して元の閉ループ平衡とは大きく異なる動作平衡で動作する可能性があるため、ループを開いた状態では実行できません。したがって、フィードバックループを閉じた状態でスイッチング電源のループゲインを評価する機能は、非常に便利です。通常、寄生素子の周波数応答への影響は最小限です。少数の寄生素子がシステムの周波数応答に大きな影響を及ぼしていると疑われる場合、SIMPLIS-FXを使用してこのような仮説を検証できます。SIMPLIS-FXは、1スイッチング周期内でシステム変数がゆっくりと変化すると仮定せず、小信号応答を導出するプロセス中にスイッチング動作を削除し

ないため、寄生要素が周波数応答に与える影響を正確に予測できます。

SIMPLIS-FXの背後にあるアルゴリズムは厳密に導出されており、正確でロバストです。たとえば、解析周波数はスイッチング周波数の半分未満に制限されません。スイッチングシステムがスイッチング周波数の半分以上の周波数で励起された場合、間違いなくエイリアシングが存在します。このような状況で、スイッチング周波数より下であろうと上であろうと、SIMPLIS-FXは励起周波数でのシステムの応答を正確に計算できます。理論的には、SIMPLIS-FXの背後にある小信号周波数解析アルゴリズムは、DCから無限大までの各解析周波数で0.5 dBおよび1度以内の精度です。実際には、解析の精度と、適用可能かつ規定の精度を維持できる最高の解析周波数は、物理的コンポーネントがスイッチング区分線形システムでどの程度正確にモデル化されるかに依存します。実験室で測定された周波数応答とSIMPLIS-FXによって生成されたデータとの間に顕著な相違がある場合、SIMPLIS-FXを信頼し、次の点に集中することができます。

1. シミュレートされたシステムが、実験室で測定されたシステムを妥当な正確さで表しているか判断してください。
2. 適切にモデル化されていないと思われるコンポーネントのデバイスモデルを改善してください。
3. スwitchングシステムの小信号励起の実験室測定は本質的にノイズが多く、ノイズは測定エラーを容易に引き起こす可能性があります。このため、実験室の測定セットアップをチェックして、測定が有効であることを確認してください。

要約すると、SIMPLIS-FXの機能は次のとおりです。

1. SIMPLIS-TXを介した時間領域シミュレーションに基づいています。
2. SIMPLIS-POPを使用して、システムの大信号の周期的な動作平衡を計算します。
3. 一般的で汎用性があります。

昇圧、降圧、昇降圧、Cuk、SEPIC、ハーフブリッジ、フルブリッジなどの任意のパルス幅変調（PWM）回路トポロジを処理します。

直列共振、並列共振、準共振、位相シフト共振などの共振回路トポロジを処理します。

連続mmf（CMM）モード、不連続mmf（DMM）モードなど、あらゆる動作モードを処理します。

電圧モード制御、ピーク電流モード制御、平均電流モード制御、電荷制御などの任意の制御方式を処理します。

固定周波数システムと可変周波数システムの両方を処理します。

複数スイッチ、複数出力のコンバータを簡単に処理します。

ループが閉じている状態でシステムのループゲインを評価できます。

寄生素子が周波数応答に与える影響を評価できます。

スイッチング周波数を超える解析周波数において正確です。

精度は0.5 dB以内および1度以内です。

12. アドバンストデジタルコンポーネント

12.1 概要

主な利点

大量のデジタルコンテンツを含むスイッチング電源のシミュレーションをサポートおよび強化するために、SIMPLISに新しいアドバンストデジタルシミュレーション機能を導入しました。

- デジタルコンテンツのレベルに関係なく、パワー変換アプリケーションの混合モードアナログデジタル回路の仮想プロトタイピングを実用的にします。
- アドバンストデジタルライブラリには、シミュレーション作業を簡素化するためのさまざまな新しいデジタル機能が用意されています。
- SIMPLISの以前のバージョンと比較して、基本的なデジタルゲートシミュレーションのシミュレーション速度が10~20倍向上します。

SIMPLISアドバンストデジタルコンポーネントを使用すると、デジタル制御電源の設計者は、ますます複雑化するデジタル制御スキームと完全な電源システムの性能との相互作用を効果的に探索できます。SIMPLISアドバンストデジタルコンポーネントを使用すると、監視回路と保護回路を記述する重要なデジタルコンテンツにより、電源システムのシミュレーション速度も向上します。

新しいデジタル機能

アドバンストデジタルライブラリは、シミュレーション作業を簡素化するためのさまざまな新しいデジタル機能を提供します。

- SIMPLISエンジンに長い間含まれていた基本的な論理ゲートに加えて、ライブラリには次のものが含まれています。
- 加算器
- 減算器
- 乗算器

- コンパレータ
- カウンタ
- ADコンバータ
- フリップフロップとラッチの拡張ライブラリ
- 非対称遅延ブロック

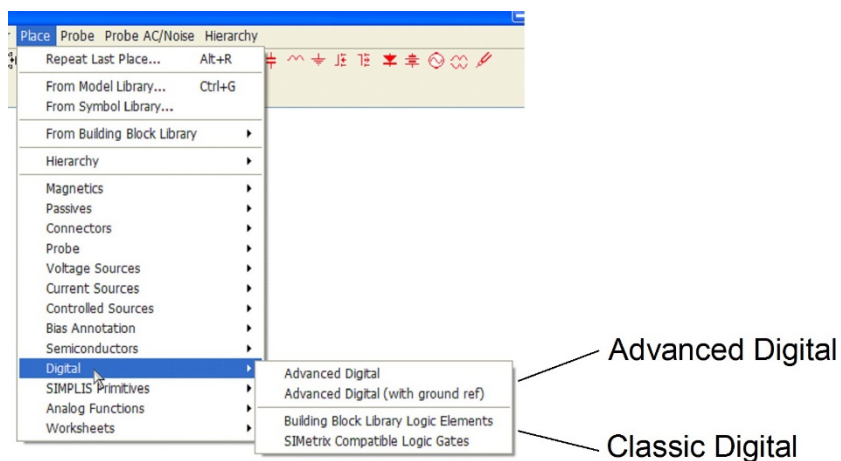
アドバンストデジタルライブラリのすべての新しいロジック機能には、次のように改善された特性があります。

- 入力の慣性遅延。（指定された遅延よりも狭い入力グリッチは、デバイスを介して伝播するのではなく、事実上無視されます。）
- すべてのアドバンストデジタルデバイスでの有限遅延。（これにより、従来のSIMPLIS論理ゲートがゼロ遅延で瞬時に状態を切り替える機能に関連する問題がなくなります。）
- ランダムバスプローブ機能は、階層回路図の任意のレベルで、すべてのアドバンストデジタルノードを含むデジタルバスで使用できるようになりました。
- すべてのアドバンストデジタルデバイスに接続する場合、グランド基準ピンはオプションです。

アドバンストデジタルコンポーネント

SIMPLISアドバンストコンポーネントは、従来のSIMPLISシミュレーションエンジンのデジタルシミュレーション性能を向上させます。この強化されたデジタルシミュレーション能力は、特に新しいアドバンストデジタルコンポーネントのシミュレーションで機能します。SIMPLIS v5.6で導入されたアドバンストデジタルコンポーネントのシミュレーション速度の改善により、検討中のシステムに大量のデジタルコンテンツがある場合、全体的なシミュレーションがはるかに高速かつ効率的になります。

新しいデジタルコンポーネントを「アドバンストデジタルコンポーネント」と呼び、SIMPLISの従来のデジタルモデルを「クラシックデジタルコンポーネント」と呼びます。SIMPLISv5.6以降、クラシックデジタルコンポーネントとアドバンストデジタルコンポーネントの両方がサポートされています。回路図エディタの再編成されたメニューセットを使用して、どちらも回路図上に配置することができます。



すべてのアドバンスデジタルコンポーネントには、シンボルの左下隅に4本の斜め縞があります。たとえば、3入力ANDゲートは次の図のいずれかになります。U1にはグランド基準ピンがあり、U2にはありません。



クラシックコンポーネント

あるコンポーネントは、次のすべての要件を満たしている場合、クラシックコンポーネントと見なされます。

1. アドバンスデジタルコンポーネントではない。
2. 電圧を測定するプローブではない。たとえば、通常の電圧プローブ、バス電圧プローブ、およびボード線図プローブは、すべて電圧を測定するプローブです。
3. 固定ピン電流プローブではない。

したがって、抵抗、キャパシタ、インダクタ、独立した制御ソース、トランス、BJT、MOSFET、光カップラ、固定インライン電流プローブなどはすべて、クラシックコンポーネントと見なされます。クラシックデジタルコンポーネントも、新しく強化されたデジタルシミュレータによってシミュレーション性能が変わらないため、クラシックコンポーネントと見なされます。

クラシックコンポーネントとアドバンスデジタルコンポーネントの類似点

1. クラシックデジタルコンポーネントとアドバンスデジタルコンポーネントはどち

らも、入力動作のモデリングに類似のアナログパラメータを採用しています。通常、各入力ピンは、抵抗 R_{IN} で構成されるA/Dインターフェイスブリッジによってモデル化されます。各入力ピンは、閾値電圧 V_{TH} とヒステリシスウィンドウ幅 $HYSTWD$ と比較した入力電圧の値に応じて、0または1の論理状態によってモデル化されます。

2. クラシックデジタルコンポーネントとアドバンスデジタルコンポーネントの両方は、出力動作のモデリングに類似のアナログパラメータを採用しています。通常、各出力は、電圧ソースと直列の抵抗 R_{OUT} で構成されるD/Aインターフェイスブリッジによってモデル化されます。電圧ソースは、その出力ピンの論理出力状態に応じて、 V_{OL} または V_{OH} の値になります。
3. クラシックコンポーネントとアドバンスデジタルコンポーネントの両方は、いくつかの小さな例外を除き、グランド基準ピンの有無にかかわらずデバイスをサポートします。
4. クラシックコンポーネントとアドバンスデジタルコンポーネントの両方は、立ち上がり時間と立ち下がり時間がゼロの出力のスイッチングをモデル化します。

クラシックコンポーネントとアドバンスデジタルコンポーネントの違い

1. アドバンスデジタルコンポーネントは、入力動作または出力動作をモデリングするためのアナログパラメータをサポートしますが、特定の入力ピンまたは出力ピンがクラシックコンポーネントに接続されている場合にのみ、A/DまたはD/Aインターフェイスブリッジが導入されます。アドバンスデジタルコンポーネントの入力ピンまたは出力ピンが他のアドバンスデジタルコンポーネントのみに接続されている場合、そのようなノードのプロベリングは、時間に対する論理値0、1、または0.5（不定の論理値）の波形を生成します。これは、波形表示ツールの上部にデジタルデータとしてプロットされます。そのようなピンのピン電流をランダムプローブしようとする、ピンの入力動作または出力動作をモデル化するアナログ回路がないため、結果はゼロアンペアの定電流になります。つまり、アドバンスデジタルコンポーネントの入力ピンまたは出力ピンが他のアドバンスデジタルコンポーネントにのみ接続されている場合、そのような入力ピンまたは出力ピンに関連付けられた入力、アナログ空間ではなく論理空間にのみ存在します。
2. アドバンスデジタルコンポーネントについては、少なくとも1つの入力ピンまたは1つの出力ピンがクラシックコンポーネントに接続されている場合、グランド基準ピンが存在する必要があります。アドバンスデジタルコンポーネントのすべての入力ピンとすべての出力ピンが他のアドバンスデジタルコンポーネントにのみ接続されている場合、グランド基準ピンはオプションです。その有無は、シミュレーション結果に影響しません。
3. グランド基準ピンのないクラシックデジタルコンポーネントの場合、各出力は、回路図のグランドノードを基準にしたテブナン等価出力を通じてアナログ電圧を生成します。
4. クラシックデジタルコンポーネントの遅延パラメータはオプションであり、デフォルト値は0.0ですが、アドバンスデジタルコンポーネントの遅延パラメータは必須であり、

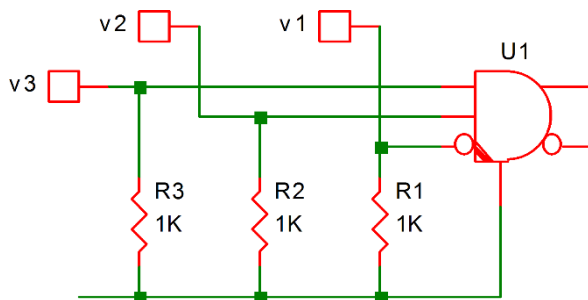
0.0に等しくすることはできません。

5. クラシックデジタルコンポーネントは、「トランスポート」遅延モデルを採用しています。つまり、単純な論理ゲートでは、定義された遅延後に入力グリッチが出力に渡されます。アドバンスデジタルコンポーネントは「慣性」遅延モデルを採用し、出力遅延パラメータより短い入力グリッチはデジタルコンポーネントに吸収され、出力に渡されません。

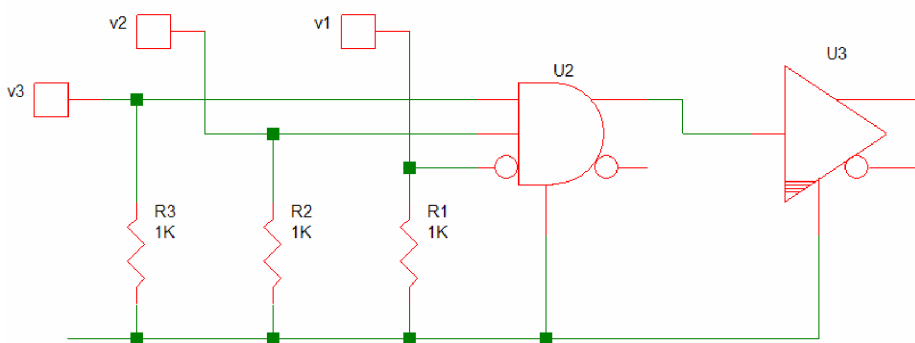
新しいアドバンスデジタルコンポーネントを活用するための戦略

新しいSIMPLISアドバンスデジタルコンポーネントを使用した効率的なシミュレーションの鍵は、アドバンスデジタルコンポーネントの高速シミュレーション時間を最大限に活用しながら、SIMPLIS回路図内の新しいアドバンスデジタルコンポーネントと残りのクラシックコンポーネント間の不要な相互作用を最小限に抑えることの最適なバランスを達成することです。これは、次のガイドラインを使用して実現できます。

1. アドバンスデジタルコンポーネントを可能な限り分離し、クラシックコンポーネントに接続されるI/Oピンの数を極小にします。
2. 単純な論理ゲートへのほとんどまたはすべての入力をクラシックコンポーネントに接続する必要がある場合、アドバンスデジタルコンポーネントライブラリからバッファを駆動するフロントエンドとして、クラシックデジタルコンポーネントを使用すると、シミュレーションが高速に実行されます。たとえば、3入力ANDゲートへのすべての入力がクラシックアナログコンポーネントに接続されている場合、有効なオプションは、3つのアナログ入力を直接検出するアドバンスデジタル3入力ANDゲートを持つことです。



ただし、より効率的で高速なシミュレーションを行うには、回路を再配置し、アドバンスデジタルバッファを駆動するフロントエンドとして、ビルディングブロックライブラリのクラシック3入力ANDゲートを使用する必要があります。従来のシミュレーションエンジンは、U2の出力の論理状態が変化したときにのみアドバンスデジタルシミュレーションを中断するため、このアプローチはより高速です。一方、前者の場合、クラシックシミュレーションエンジンは、ANDゲートの3つの入力のいずれかの論理状態が変わったとき、アドバンスデジタルシミュレーションを中断する必要があります。



この例では、ゼロ遅延をU2（クラシック3入力ANDゲート）に割り当て、非ゼロ遅延をU3（アドバンストデジタルバッファ）に割り当てる必要があります。

3. 少なくとも1つのI/Oピンがアナログノードに接続されているアドバンストデジタルコンポーネントの場合、グランド基準ピンを含むバージョンを使用する必要があります。
4. I/Oピンがすべて他のアドバンストデジタルコンポーネントにのみ接続されているアドバンストデジタルコンポーネントの場合、グランド基準ピンを含むバージョンまたは含まないバージョンを使用できます。選択はユーザの好み次第であり、シミュレーション結果には影響しません。

単純なデモ回路

回路例は、Examples\SIMPLIS\Digital_PWM\SyncBuck_Digital_PWM.sxschに提供されています。これは、PID補償を使用するPWMコントローラによって制御される単純な同期降圧コンバータを表す階層回路図です。このPWMコントローラは、完全に新しいアドバンストデジタルコンポーネントで構成されています。