

SiMetrix

SIMETRIX SIMULATOR REFERENCE MANUAL

(最初の部分の日本語訳)

VERSION 8.3

APRIL 2019

SIMETRIX SIMULATOR REFERENCE MANUAL

COPYRIGHT © SIMETRIX TECHNOLOGIES LTD. 1992-2019

Trademarks:

PSpice is a trademark of Cadence Design Systems Inc.

Hspice is a trademark of Synopsis Inc.

SIMetrix Technologies Ltd.,
78 Chapel Street, Thatcham,
Berkshire
RG18 4QN
United Kingdom

Tel: +44 1635 866395
Fax: +44 1635 868322
Email: support@simetrix.co.uk
Web: <http://www.simetrix.co.uk>



目次

1	はじめに	1
1.1	概要	1
1.2	SIMetrixシミュレータとは何か?	1
1.3	マニュアルの内容	1
2	シミュレータの実行	2
2.1	回路図エディタでシミュレータを使用	2
2.1.1	ネットリスト行の追加	2
2.1.2	ネット名とピン名の表示	2
2.1.3	デバイスパラメータの編集	3
2.1.4	リテラル値の編集 - シフト-F7を使用	3
2.2	非GUIモードでの実行	4
2.2.1	概要	4
2.2.2	重要なライセンス情報	4
2.2.3	構文	4
2.2.4	中止	6
2.2.5	データ読み込み	6
2.3	コンフィグレーション設定	6
2.3.1	グローバル設定	7
2.3.2	データバッファリング	7
2.4	ネットリスト形式	8
2.4.1	ファイル形式	8
2.4.2	言語宣言	9
2.4.3	コメント	9
2.4.4	デバイス行	9
2.4.5	シミュレータステートメント	11
2.5	シミュレータ出力	11
2.5.1	リストファイル	11
2.5.2	バイナリデータファイル	12
2.5.3	出力データ名	12
2.6	保存されるデータの制御	14

3	シミュレータデバイス	15
3.1	概要	15
3.2	XSPICEデバイスの使用	15
3.2.1	ベクトル接続	15
3.2.2	接続タイプ	16
3.3	式の使用	17
3.3.1	概要	17
3.3.2	デバイスパラメータに式を使用する	17
3.3.3	モデルパラメータに式を使用する	17
3.3.4	式の構文	18
3.3.5	例	25
3.3.6	最適化	28
3.4	サブ回路	29
3.4.1	概要	29
3.4.2	サブ回路の定義	29
3.4.3	サブ回路のインスタンス	30
3.4.4	サブ回路にパラメータを渡す	30
3.4.5	サブ回路のネスト	31
3.4.6	グローバルノード	31
3.4.7	サブ回路の前処理	31
3.5	モデルのビンニング	32
3.5.1	概要	32
3.5.2	ビン化モデルの定義	32
3.5.3	例	32
3.6	言語の違い	33
3.6.1	インラインコメント	33
3.6.2	ラベルなしデバイスパラメータ	33
3.6.3	LOG0 と PWR0	34
3.7	デバイスコンフィグレーションのカスタマイズ	34
3.7.1	概要	34
3.8	初期条件	35
3.8.1	ノード初期条件	35
3.8.2	キャパシタ初期条件	35
3.8.3	インダクタ初期条件	36
4	アナログデバイスリファレンス	38

4.1	概要	38
4.2	その他のドキュメント	38
4.3	ACテーブルルックアップ	38
4.3.1	ネットリストエントリ	38
4.3.2	モデル形式	38
4.3.3	ACテーブルの注意	39
4.4	任意ソース	39
4.4.1	ネットリストエントリ	39
4.4.2	任意式の注意	40
4.4.3	電荷デバイスと磁束デバイス	41
4.4.4	任意ソースの例	41
4.4.5	PSpiceとHspiceの構文	43
4.5	バイポーラ接合トランジスタ	44
4.5.1	ネットリストエントリ	44
4.5.2	NPN BJTのモデル構文	44
4.5.3	PNP BJTのモデル構文	44
4.5.4	ラテラルPNP BJTのモデル構文	44
4.5.5	BJTのモデルパラメータ	45
4.5.6	Hspiceの温度パラメータ	47
4.5.7	注意	49
4.6	バイポーラ接合トランジスタ (VBIC without self heating)	49
4.6.1	ネットリストエントリ	50
4.6.2	モデル構文	50
4.6.3	モデルパラメータ	50
4.6.4	注意	53
4.7	バイポーラ接合トランジスタ (VBIC with self heating)	53
4.7.1	ネットリストエントリ	53
4.7.2	モデル構文	54
4.7.3	モデルパラメータ	54
4.7.4	注意	54
4.8	バイポーラ接合トランジスタ (MEXTRAM)	54
4.9	バイポーラ接合トランジスタ (HICUM)	54
4.9.1	ネットリストエントリ	54
4.9.2	NPNモデル構文	55
4.9.3	PNPモデル構文	55
4.9.4	注意	55

4.10	キャパシタ	55
4.10.1	ネットリストエントリ	55
4.10.2	モデル構文	56
4.10.3	モデルパラメータ	56
4.11	制御電流源	56
4.11.1	ネットリストエントリ	57
4.11.2	例	57
4.11.3	多項式の仕様	58
4.12	電流制御電圧源	58
4.12.1	ネットリストエントリ	58
4.13	電流源	59
4.13.1	ネットリストエントリ	59
4.14	ダイオード・Level 1とLevel 3	59
4.14.1	ネットリストエントリ	59
4.14.2	例	60
4.14.3	ダイオードのモデル構文	60
4.14.4	ダイオードのモデルパラメータ・Level = 1	60
4.14.5	ダイオードのモデルパラメータ・Level = 3	62
4.14.6	ダイオードの使用	63
4.15	ダイオード・ソフトリカバリ	63
4.15.1	ネットリストエントリ	63
4.15.2	ダイオードのモデル構文	64
4.15.3	ソフトリカバリダイオードのモデルパラメータ	64
4.15.4	基礎方程式	64
4.15.5	リファレンス	65
4.16	GaAsFET	65
4.16.1	ネットリストエントリ	65
4.16.2	GaAsFETモデル構文	65
4.16.3	GaAsFETモデルパラメータ	65
4.17	インダクタ (理想)	66
4.17.1	ネットリストエントリ	66
4.17.2	関連項目	66
4.18	インダクタ (可飽和)	66
4.18.1	ネットリストエントリ	66
4.18.2	モデル形式 - ヒステリシスのあるJiles-Athertonモデル	67
4.18.3	モデル形式 - ヒステリシスのない単純なモデル	67

4.18.4	Jiles-Athertonパラメータ	67
4.18.5	ヒステリシスのないモデルパラメータ	67
4.18.6	Jiles-Athertonモデルの注意	68
4.18.7	ヒステリシスのないモデルの注意	68
4.18.8	変圧器の実装	68
4.18.9	B-Hカーブのプロット	69
4.18.10	リファレンス.....	69
4.19	インダクタ(テーブルルックアップ)	69
4.19.1	ネットリストエントリ	69
4.19.2	モデル構文	69
4.19.3	境界インダクタンス	70
4.19.4	スムージング関数	70
4.20	IGBT	71
4.20.1	ネットリストエントリ	71
4.20.2	モデル構文	71
4.20.3	注意	72
4.21	ジャンクションFET	72
4.21.1	ネットリストエントリ	72
4.21.2	NチャネルJFET: モデル構文	73
4.21.3	PチャネルJFET: モデル構文	73
4.21.4	JFET: モデルパラメータ	73
4.21.5	例	74
4.22	ラプラス伝達関数 - 集中定数実装	74
4.22.1	ネットリストエントリ	74
4.22.2	接続の詳細	74
4.22.3	モデル形式	74
4.22.4	モデルパラメータ	74
4.22.5	記述	75
4.22.6	例	75
4.22.7	ラプラス式.....	79
4.22.8	係数を使用したラプラス式の定義.....	80
4.22.9	その他のモデルパラメータ	80
4.22.10	制限	80
4.22.11	実装	80
4.23	ラプラス伝達関数- コンボリューション実装	81
4.23.1	ネットリストエントリ	81

4.23.2	モデル構文	81
4.23.3	モデルパラメータ	82
4.23.4	ラプラス伝達関数	83
4.23.5	実装	86
4.23.6	インパルス応答	86
4.23.7	実行時エラー制御	89
4.23.8	PSpice LAPLACEおよびFREQの互換性	89
4.24	損失のある伝送線路	89
4.24.1	ネットリストエントリ	89
4.24.2	モデル構文	90
4.24.3	モデルパラメータ	90
4.24.4	例	91
4.24.5	サブ回路に基づくRLGCモデル	91
4.25	MOSFET	91
4.25.1	ネットリストエントリ	91
4.25.2	NMOSのモデル構文	92
4.25.3	PMOSのモデル構文	92
4.25.4	MOS Levels 1, 2 and 3: モデルパラメータ	92
4.25.5	CJデフォルト	94
4.25.6	ゲート電荷モデル, Levels 1, 2 and 3	95
4.25.7	Level 1, 2 and 3の注意:	95
4.25.8	MOS Level 17: モデルパラメータ	95
4.25.9	Level 17の注意	96
4.26	BSIM3 MOSFET	97
4.26.1	注意	97
4.26.2	バージョンセレクタ	98
4.26.3	モデルパラメータ	98
4.26.4	その他のドキュメント	99
4.26.5	ビニング処理	99
4.27	BSIM4 MOSFETs	99
4.27.1	注意	99
4.27.2	その他のドキュメント	100
4.27.3	ビニング処理	100
4.27.4	HspiceのLevel 54へのマッピング	100
4.28	HiSim HV MOSFET	101
4.28.1	注意	101

4.29	MOSFET GMIN実装	101
4.30	PSP MOSFET	102
4.30.1	ネットリストエントリ	102
4.30.2	NMOSのモデル構文 Version 101.0	102
4.30.3	PMOSのモデル構文 Version 101.0	102
4.30.4	NMOSのモデル構文 Version 102.3	103
4.30.5	PMOSのモデル構文 Version 102.3	103
4.30.6	注意	103
4.31	抵抗	103
4.31.1	ネットリストエントリ	103
4.31.2	注意	104
4.31.3	抵抗のモデル構文	104
4.31.4	抵抗のモデルパラメータ	104
4.31.5	注意	105
4.32	抵抗 - Hspice互換	105
4.32.1	ネットリストエントリ	105
4.32.2	抵抗のモデル構文	106
4.32.3	抵抗の計算	107
4.32.4	キャパシタンスの計算	107
4.32.5	温度スケールリング	108
4.32.6	フリッカーノイズ	109
4.32.7	ACRESMODパラメータ	109
4.32.8	Hspiceの抵抗をデフォルトにする	109
4.33	CMC抵抗	109
4.33.1	ネットリストエントリ	109
4.33.2	モデル形式	109
4.34	サブ回路のインスタンス	109
4.34.1	ネットリストエントリ	110
4.35	伝送線路	110
4.35.1	ネットリストエントリ	110
4.35.2	例	111
4.36	電圧制御電流源	111
4.36.1	ネットリストエントリ	111
4.36.2	PSpiceの構文	111
4.37	電圧制御スイッチ	112
4.37.1	ネットリストエントリ	112

4.37.2	電圧制御スイッチのモデル構文	112
4.37.3	電圧制御スイッチのモデルパラメータ	112
4.37.4	電圧制御スイッチの注意	113
4.38	電圧制御ソース	113
4.38.1	ネットリストエントリ	113
4.38.2	PSpiceの構文	114
4.39	電圧源	114
4.39.1	ネットリストエントリ	114
4.39.2	パルスソース	115
4.39.3	Piece-Wise Linearソース	116
4.39.4	PWLファイルソース	117
4.39.5	正弦波ソース	118
4.39.6	指数関数ソース	119
4.39.7	単一周波数FM	120
4.39.8	ノイズソース	120
4.39.9	拡張PWLソース	120
4.40	相互インダクタ	122
4.40.1	ネットリストエントリ	122
4.40.2	注意	123
4.40.3	例	123
4.41	Verilog-HDLインターフェース (VSXA)	123
4.41.1	概要	123
4.41.2	アナログ入力インターフェース	126
4.41.3	アナログ出力インターフェース	126
4.41.4	データベクトル出力	127
4.41.5	モジュールキャッシュ	127
4.42	NXPコンパクトモデル	128
4.42.1	はじめに	128
4.42.2	SIMKITデバイス	129
4.42.3	SIMKITモデルの注意	134
5	デジタル/混合信号デバイスリファレンス	136
5.1	デバイス概要	136
5.1.1	共通パラメータ	136
5.1.2	遅延	137
5.2	ANDゲート	137

5.2.1	ネットリストエントリ	138
5.2.2	接続の詳細	138
5.2.3	モデル形式	138
5.2.4	モデルパラメータ	138
5.2.5	デバイス動作	138
5.3	D型ラッチ	139
5.3.1	ネットリストエントリ	139
5.3.2	接続の詳細	139
5.3.3	モデル形式	139
5.3.4	モデルパラメータ	139
5.3.5	デバイス動作	140
5.4	D型フリップフロップ	141
5.4.1	ネットリストエントリ	141
5.4.2	接続の詳細	141
5.4.3	モデル形式	141
5.4.4	モデルパラメータ	141
5.4.5	デバイス動作	142
5.5	バッファ	142
5.5.1	ネットリストエントリ	143
5.5.2	接続の詳細	143
5.5.3	モデル形式	143
5.5.4	モデルパラメータ	143
5.5.5	デバイス動作	143
5.6	周波数分割器	144
5.6.1	ネットリストエントリ	144
5.6.2	接続の詳細	144
5.6.3	モデル形式	145
5.6.4	モデルパラメータ	145
5.6.5	デバイス動作	145
5.7	初期条件	146
5.7.1	ネットリストエントリ	146
5.7.2	接続の詳細	146
5.7.3	モデル形式	146
5.7.4	モデルパラメータ	146
5.7.5	デバイス動作	146
5.8	デジタルパルス	146

5.8.1	ネットリストエントリ	146
5.8.2	接続の詳細	146
5.8.3	インスタンスパラメータ	147
5.8.4	モデル形式	147
5.8.5	モデルパラメータ	147
5.8.6	デバイス動作	147
5.9	デジタル信号ソース	147
5.9.1	ネットリストエントリ	148
5.9.2	接続の詳細	148
5.9.3	モデル形式	148
5.9.4	モデルパラメータ	148
5.9.5	デバイス動作	148
5.9.6	ファイル形式	148
5.9.7	例	149
5.10	インバータ	150
5.10.1	ネットリストエントリ	150
5.10.2	接続の詳細	150
5.10.3	モデル形式	150
5.10.4	モデルパラメータ	151
5.10.5	デバイス動作	151
5.11	JKフリップフロップ	151
5.11.1	ネットリストエントリ	151
5.11.2	接続の詳細	151
5.11.3	モデル形式	152
5.11.4	モデルパラメータ	152
5.11.5	デバイス動作	152
5.12	任意の論理ブロック	154
5.12.1	ネットリストエントリ	154
5.12.2	接続の詳細	154
5.12.3	Instance Parameters	155
5.12.4	モデル形式	155
5.12.5	モデルパラメータ	155
5.12.6	デバイス動作	156
5.13	NANDゲート	156
5.13.1	ネットリストエントリ	156
5.13.2	接続の詳細	156

5.13.3	モデル形式	156
5.13.4	モデルパラメータ	156
5.13.5	デバイス動作	157
5.14	NORゲート	157
5.14.1	ネットリストエントリ	157
5.14.2	接続の詳細	157
5.14.3	モデル形式	157
5.14.4	モデルパラメータ	157
5.14.5	デバイス動作	158
5.15	オープンコレクタバッファ	158
5.15.1	ネットリストエントリ	158
5.15.2	接続の詳細	158
5.15.3	モデル形式	158
5.15.4	モデルパラメータ	158
5.15.5	デバイス動作	159
5.16	オープンエミッタバッファ	159
5.16.1	ネットリストエントリ	159
5.16.2	接続の詳細	159
5.16.3	モデル形式	159
5.16.4	モデルパラメータ	159
5.16.5	デバイス動作	159
5.17	ORゲート	160
5.17.1	ネットリストエントリ	160
5.17.2	接続の詳細	160
5.17.3	モデル形式	160
5.17.4	モデルパラメータ	160
5.17.5	デバイス動作	161
5.18	プルダウン抵抗	161
5.18.1	ネットリストエントリ	161
5.18.2	接続の詳細	161
5.18.3	モデル形式	161
5.18.4	モデルパラメータ	161
5.18.5	デバイス動作	161
5.19	プルアップ抵抗	162
5.19.1	ネットリストエントリ	162
5.19.2	接続の詳細	162

5.19.3	モデル形式	162
5.19.4	モデルパラメータ	162
5.19.5	デバイス動作	162
5.20	ランダムアクセスメモリ	162
5.20.1	ネットリストエントリ	162
5.20.2	接続の詳細	162
5.20.3	モデル形式	163
5.20.4	モデルパラメータ	163
5.20.5	デバイス動作	163
5.21	セットリセットフリップフロップ	163
5.21.1	ネットリストエントリ	163
5.21.2	接続の詳細	163
5.21.3	モデル形式	164
5.21.4	モデルパラメータ	164
5.21.5	デバイス動作	165
5.22	SRラッチ	166
5.22.1	ネットリストエントリ	166
5.22.2	接続の詳細	166
5.22.3	モデル形式	166
5.22.4	モデルパラメータ	166
5.22.5	デバイス動作	167
5.23	状態マシン	167
5.23.1	ネットリストエントリ	167
5.23.2	接続の詳細	167
5.23.3	モデル形式	167
5.23.4	モデルパラメータ	167
5.23.5	ファイル構文	168
5.23.6	注意	168
5.24	トグルフリップフロップ	169
5.24.1	ネットリストエントリ	169
5.24.2	接続の詳細	169
5.24.3	モデル形式	169
5.24.4	モデルパラメータ	169
5.24.5	デバイス動作	170
5.25	3ステートバッファ	171
5.25.1	ネットリストエントリ	171

5.25.2	接続の詳細	172
5.25.3	モデル形式	172
5.25.4	モデルパラメータ	172
5.25.5	デバイス動作	172
5.26	Exclusive NORゲート	173
5.26.1	ネットリストエントリ	173
5.26.2	接続の詳細	173
5.26.3	モデル形式	173
5.26.4	モデルパラメータ	173
5.26.5	デバイス動作	173
5.27	Exclusive ORゲート	174
5.27.1	ネットリストエントリ	174
5.27.2	接続の詳細	174
5.27.3	モデル形式	174
5.27.4	モデルパラメータ	174
5.27.5	デバイス動作	175
5.28	アナログデジタルコンバータ	175
5.28.1	ネットリストエントリ	175
5.28.2	接続の詳細	175
5.28.3	モデル形式	175
5.28.4	モデルパラメータ	175
5.28.5	デバイス動作	176
5.29	アナログデジタルインターフェースブリッジ	178
5.29.1	ネットリストエントリ	178
5.29.2	接続の詳細	178
5.29.3	モデル形式	178
5.29.4	モデルパラメータ	178
5.29.5	デバイス動作	178
5.29.6	アナログ入力負荷	179
5.29.7	入力クランプ	179
5.29.8	時間ステップ制御 - TIME_TOLパラメータ	179
5.30	デジタルアナログコンバータ	182
5.30.1	ネットリストエントリ	182
5.30.2	接続の詳細	182
5.30.3	モデル形式	182
5.30.4	モデルパラメータ	182

5.30.5	デバイス動作	182
5.31	デジタルアナログインターフェースブリッジ	185
5.31.1	ネットリストエントリ	185
5.31.2	接続の詳細	185
5.31.3	モデル形式	185
5.31.4	モデルパラメータ	185
5.31.5	DC特性	186
5.31.6	スイッチング特性	187
5.32	制御デジタル発振器	187
5.32.1	ネットリストエントリ	187
5.32.2	接続の詳細	187
5.32.3	インスタンスパラメータ	188
5.32.4	モデル形式	188
5.32.5	モデルパラメータ	188
5.32.6	デバイス動作	188
5.32.7	時間ステップ制御	189
5.33	アナログデジタル シュミットトリガー	189
5.33.1	ネットリストエントリ	189
5.33.2	接続の詳細	189
5.33.3	モデル形式	189
5.33.4	モデルパラメータ	189
5.33.5	デバイス動作	190
6	コマンドリファレンス	191
6.1	概要	191
6.2	一般的なスイープ仕様	192
6.2.1	概要	192
6.2.2	構文	193
6.3	マルチステップ解析	194
6.3.1	概要	194
6.3.2	構文	195
6.4	.AC	196
6.4.1	概要	196
6.4.2	注意	196
6.4.3	例	197
6.4.4	入れ子スイープの例	197

6.5	.ALIAS	197
6.5.1	概要	197
6.5.2	例	197
6.6	.DC	198
6.6.1	概要	198
6.6.2	例	199
6.6.3	入れ子スコープの例	199
6.7	.FILEおよび.ENDF	200
6.7.1	概要	200
6.7.2	例	200
6.7.3	重要な注意	200
6.7.4	SIMPLISの使用	200
6.8	.FUNC	200
6.8.1	例	201
6.8.2	最適化	201
6.9	.GLOBAL	201
6.10	.GRAPH	201
6.10.1	パラメータ	201
6.10.2	多重の.GRAPHステートメントの使用	205
6.10.3	X-Yプロットの作成	205
6.10.4	サブ回路での.GRAPHの使用	206
6.10.5	.GRAPHでの式の使用	206
6.10.6	.GRAPHでのスペクトルのプロット	206
6.11	.IC	207
6.11.1	代替の初期条件実装	207
6.12	.INC	208
6.13	.KEEP	208
6.13.1	オプション設定	209
6.14	.LOAD	211
6.15	.LIB	212
6.15.1	SIMetrixネイティブ形式	212
6.15.2	HSPICE形式	213
6.16	.MAP	213
6.16.1	.MAPの注意	213
6.16.2	デバイスコンフィグレーションファイル	214
6.16.3	全シミュレータデバイスのリスト	215

6.17	.MODEL	216
6.17.1	XSPICEモデルタイプ	216
6.17.2	SPICEモデルタイプ	217
6.17.3	安全動作領域 (SOA) 限界	218
6.17.4	例	218
6.18	.NOCONV	218
6.19	.NODESET	218
6.20	.NOISE	220
6.20.1	注意	220
6.20.2	デバイスベクトル名の添字	221
6.20.3	Noise Info Fileの作成	222
6.20.4	例	222
6.21	.OP	222
6.21.1	'OFF'パラメータ	223
6.21.2	ノードセット	223
6.21.3	初期条件	223
6.21.4	Operating Point出力情報	223
6.22	.OPTIONS	223
6.22.1	シミュレータオプションのリスト	224
6.23	.PARAM	242
6.23.1	例	242
6.23.2	ネットリスト順序	243
6.23.3	サブ回路パラメータ	243
6.23.4	回路図での.PARAMの使用	244
6.23.5	ライブラリでのPARAM	244
6.24	.POST_PROCESS	244
6.24.1	重要な注意	244
6.25	.PRINT	244
6.25.1	注意	245
6.25.2	例	246
6.26	.SENS	246
6.27	.SETSOA	246
6.27.1	例	250
6.28	.SUBCKT and .ENDS	251
6.29	.TEMP	252
6.30	.TF	252

6.30.1	注意	253
6.30.2	例	253
6.31	.TRACE	253
6.31.1	例	254
6.31.2	注意	254
6.32	.TRAN	254
6.32.1	高速スタート	256
6.32.2	スナップショット	256
6.33	リアルタイムノイズ解析	256
6.33.1	例	257
6.33.2	テスト結果	257
6.33.3	リアルタイムノイズの注意	258
7	モンテカルロ, 感度, ワーストケース	262
7.1	概要	262
7.2	モンテカルロ解析	263
7.2.1	マルチステップ	263
7.2.2	シングルステップスイープ	263
7.2.3	モンテカルロログファイル	264
7.2.4	乱数発生器の設定	264
7.3	感度解析とワーストケース解析	265
7.3.1	一般的な操作	265
7.3.2	マルチステップ	265
7.3.3	シングルステップスイープ	267
7.3.4	感度測定関数	267
7.4	公差の指定	267
7.4.1	概要	268
7.4.2	分布関数	268
7.4.3	Hspice分布関数	274
7.4.4	TOL, MATCH, およびLOT デバイスパラメータ	275
8	収束, 精度, および性能	277
8.1	概要	277
8.2	DC動作点	277
8.2.1	概要	277
8.2.2	ソースとGMINのステップング	278

8.2.3	疑似過渡解析	278
8.2.4	ジャンクション初期化反復	280
8.2.5	ノードセットの使用	281
8.3	過渡解析	281
8.3.1	収束しない原因は何か?	281
8.3.2	数値ノイズと反復モード	282
8.3.3	過渡収束の修正と改善	284
8.4	DCスイープ	284
8.5	DC動作点アルゴリズム	284
8.5.1	ジャンクション初期化反復	285
8.5.2	ソースステッピング	285
8.5.3	対角線GMINステッピング	285
8.5.4	ジャンクションGMINステッピング	286
8.5.5	疑似過渡解析	286
8.5.6	DCメソッドシーケンスの制御	286
8.6	特異行列エラー	287
8.7	時間ステップの過小	287
8.8	精度, 積分	287
8.8.1	簡単なアプローチ	287
8.8.2	反復精度	288
8.8.3	時間ステップ制御	288
8.8.4	AC解析の精度	290
8.8.5	公差オプションの要約	290
8.8.6	積分法・METHODオプション	291
8.9	マルチコアシステム	293
8.9.1	シングルステップラン	293
8.9.2	シングルステップランでのマルチコア使用	294
8.9.3	マルチコア, マルチステップシミュレーション	294
8.10	行列ソルバー	294
9	デジタルシミュレーション	296
9.1	概要	296
9.2	論理状態	296
9.2.1	状態解決テーブル	297
9.3	アナログデジタルインターフェース	297
9.3.1	A-Dブリッジの選択方法	299

9.4	ロジックファミリ	299
9.4.1	ロジックファミリ モデルパラメータ	299
9.4.2	ロジック互換テーブル	300
9.4.3	ロジック互換ファイル形式	300
9.4.4	サポートされるロジックファミリ	301
9.4.5	ユニバーサルロジックファミリ	302
9.4.6	内部テーブル	302
9.5	負荷遅延	302
9.5.1	概要	302
9.5.2	出力抵抗	302
9.5.3	入力遅延	303
9.5.4	配線遅延	303
9.6	デジタルモデルライブラリ	303
9.6.1	サードパーティライブラリの使用	303
9.7	任意のロジックブロック - ユーザ定義モデル	303
9.7.1	概要	303
9.7.2	例	304
9.7.3	例 2 - 簡単なマルチプライヤ	306
9.7.4	例 3 - ROMルックアップテーブル	306
9.7.5	例 4 - D型フリップフロップ	307
9.7.6	デバイス定義 - ネットリストエン트리とモデルパラメータ	307
9.7.7	言語定義 - 概要	309
9.7.8	言語定義 - 定数と名前	309
9.7.9	言語定義 - ポート	310
9.7.10	言語定義 - レジスタと変数	311
9.7.11	言語定義 - 代入	313
9.7.12	言語定義 - ユーザ値とデバイス値	315
9.7.13	診断: トレースファイル	316
9.8	混合モードシミュレータ - 使い方	317
9.8.1	イベント駆動型デジタルシミュレータ	317
9.8.2	アナログシミュレータとのインターフェース	317
9.9	XSPICEに対する機能強化	318

1. はじめに

1.1 概要

このマニュアルは、SIMetrixシミュレータの完全なリファレンスドキュメントを提供します。基本的に、シミュレータは入力としてネットリストを受け取り、出力としてバイナリデータファイルとリストファイルを作成します。ネットリストは、回路トポロジを定義し、シミュレータによって実行される解析も指定します。ネットリストには、必要なデバイスモデルを直接含めることも、デバイスモデルライブラリから自動的にインポートすることもできます。

シミュレータは、GUIモードまたは非GUIモードで操作できます。GUIモードは通常の操作方法であり、SIMetrixフロントエンドが必要です。非GUIモードでは、シミュレータは非対話形式でスタンドアローン動作し、バックグラウンドで低い優先度で動作するように設定できます。

1.2 SIMetrixシミュレータとは何か？

SIMetrixシミュレータコアは、イベント駆動型ゲートレベルデジタルシミュレータと密接に結合したダイレクトマトリックスアナログシミュレータで構成されています。この組み合わせは、多くの場合、混合モードまたは混合信号と呼ばれ、アナログ回路とデジタル回路の両方を効率的にシミュレートする機能を備えています。

SIMetrixアナログシミュレータで採用されているコアアルゴリズムは、カリフォルニア大学バークレー校の電気工学・コンピュータサイエンス学科のCAD/ICグループが開発したSPICEプログラムに基づいています。デジタルイベント駆動型シミュレータは、ジョージア工科大学のコンピュータサイエンス・インフォメーションテクノロジー研究所で開発されたXSPICEから派生しています。

1.3 マニュアルの内容

このリファレンスマニュアルには、すべてのシミュレータ解析モードとサポートされるデバイスの詳細な説明が含まれています。

2. シミュレータの実行

2.1 回路図エディタでシミュレータを使用

SIMetrix回路図エディタを使用したシミュレーションの詳細なドキュメントについては、SIMetrixユーザマニュアルをご覧ください。ただし、シミュレータの実行には回路図エディタのいくつかの機能が特に重要であるため、ここで説明を繰り返します。

2.1.1 ネットリスト行の追加

回路図エディタのメニュー**Simulator | Choose Analysis...**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。これは、特にコマンド構文に精通しているユーザーにとっては、GUIを使用するよりも速くて簡単です。

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、**Choose Analysis**ダイアログの設定は同期しているため、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。これはトグルになっており、もう一度押すと非表示になります。**Choose Analysis**ダイアログを使用して既に解析モードを選択している場合は、シミュレータステートメントが既に表示されています。

シミュレータコマンドウィンドウには、右クリックで開くポップアップメニューがあります。一番上の項目の**Edit file at cursor**は、カーソルでポイントされているファイル名、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は、そのウィンドウと回路図描画領域の間の分割バーを使用して、サイズを変更できます。

このウィンドウには、シミュレータコマンドだけでなく、好きなものを追加できます。内容は、シミュレータに提供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、**.PARAM**ステートメント、デバイスモデル、インダクタ結合仕様、**.OPTIONS**ステートメント、または単にコメントを配置できます。**Choose Analysis**ダイアログでは、解析ステートメントと一部の**.OPTIONS**設定が構文解析され、場合によっては変更されますが、その他はすべてそのまま残ります。

2.1.2 ネット名とピン名の表示

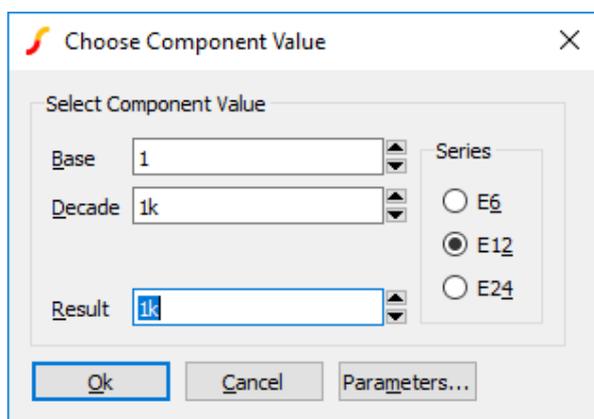
回路図上の特定のネットに使用される名前を知る必要があることがあります。これは、ネットがシミュレータステートメント（.NOISEなど）または任意のソース入力で参照される場合です。

次の2つのアプローチがあります。

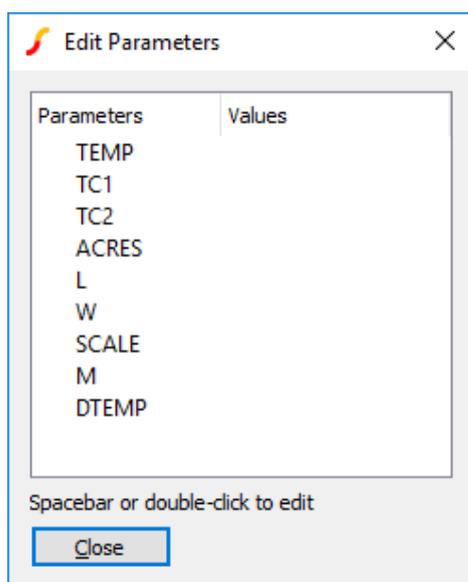
- 回路図エディタのネットリストジェネレータによって生成されたデフォルト名を見つけます。これを行うには、目的のネット上にマウスマウスカーソルを移動し、ステータスバーで“NET=???”の形式のネット名を確認します。
- ユーザが選択したネット名を強制します。これには、TerminalシンボルまたはSmall Terminalシンボルを使用します。これらはメニューの**Place | Connectors**の下にあります。回路図に配置した後、それを選択し、F7を押して名前を編集します。この名前は、接続先のネットに名前を付けるために使用されます。

2.1.3 デバイスパラメータの編集

回路図で追加のパラメータを使用するには、F7または同等のメニューで開いたダイアログボックスでParametersボタンを使用します。たとえば、抵抗を編集するとき次のボックスが表示されます。



Parametersボタンを押すと、パラメータ値を編集できる別のダイアログが開きます。



右クリックメニューの**Edit Additional Parameters...**を使用して、このボックスを直接表示することもできます。

2.1.4 リテラル値の編集 - シフト-F7を使用

前述の方法は、回路図エディタが編集集中のデバイスについて知る必要があるため、確実ではありません。状況によっては、これにはシンボルに特別なプロパティが存在する必要があるため、これらが定義されていない場合があります（たとえば、MOSFETがどのレベルであるかを回路図に伝えるため）。

通常のデバイス編集方法が適さない可能性がある別の状況は、パラメータを式として定義する必要がある場合です。

これらの状況では、**shift-F7**を使用できます。これにより、モデル名を含むデバイスのリテラル値が、ネットリストに配置されるとおりに編集されます。**shift-F7**はすべてのスマートアルゴリズムをバイパスし、生の値を提示します。生の値も指定する必要があります。たとえば、N1と呼ばれるモデルを参照するMOSFETに入力できるものは次のとおりです。

```
N1 L={LL-2*EDGE} W={WW-2*EDGE}
```

モデル名を含める必要があることに注意してください。

2.2 非GUIモードでの実行

2.2.1 概要

シミュレータは、フロントエンドから独立して、非対話型の非GUIモードで実行できます。これは、独自のスクリプトまたは言語またはDOSバッチファイルによって制御されるシミュレーション「バッチ」を実行するのに役立ちます。

Windowsでは、シミュレータは「コンソールモード」アプリケーションとして実行され、GUI要素は作成されません。

このモードで実行すると、シミュレータは指定されたネットリストを読み込み、シミュレーションを実行してから、呼び出しプログラムに制御を戻します。シミュレータはバイナリデータファイルとリストファイルを生成します。

2.2.2 重要なライセンス情報

非GUIモードは、ネットワークライセンスを使用している場合にのみ可能です。ポータブル（ドングル）ライセンスを使用している場合、この機能は使用できません。非ネットワークライセンスを所有していて、非GUIシミュレーションモードを使用したい場合は、サポートにお問い合わせください。

この動作モードは、ライセンス目的で「カウント」されます。つまり、すべてのライセンスが単一のマシン上の単一のユーザによって実行されている場合でも、発行されたライセンスごとに1つの非GUIシミュレーションプロセスしか実行できません。対照的に、GUIを介して手動で開始された通常のシミュレーションはカウントされず、この目的のために同じユーザが同じマシンで任意の数のランを開始できます。

これらの制約は、SIMetrix以外の環境から実行されるシミュレーションでの複数のコアマシンの利用を制限するために作られました。

2.2.3 構文

コマンド構文は次のとおりです。

```
SIM [/config "config_location"] [/gui mode] [/check] [/an  
  "analysis_line"] [/list list_filename] [/options "options"]  
  [/nolist] [/lowPriority] [/nodata] [/k] [/extraline extra_line]  
  netlist_file [data_file]
```

config_location コンフィグレーション設定を保持するファイルの場所。コンフィグレーション設定には、グローバルオプションとグローバルモデルライブ

ラリの場所が含まれます。値は次の形式である必要があります。

PATH;*pathname*

*pathname*には、%EXEPATH%などのシステムシンボリックパス値を使用できます。詳細については、*User's Manual/Sundry Topics/Symbolic Path Names*を参照してください。

指定しない場合、コンフィグレーション設定はBase.sxprjファイルから取得されます。このファイルの場所の詳細については、*User's Manual/Sundry Topics/Configuration Settings/Default Configuration Location*を参照してください。

または、startup.iniファイルの設定を使用して場所を指定できます。SimConfigという値を[Startup]セクションに追加し、次の値を指定します。

PATH;*pathname*

startup.iniファイルは、SIMetrix実行可能バイナリ(WindowsのSIMetrix.exe)と同じディレクトリに配置する必要があります。startup.iniファイルの詳細については、*User's Manual/Sundry Topics/SIMetrix Command Line Parameters/Using startup.ini*を参照してください。

/configスイッチが存在する場合は、常にコマンドの最初の引数の前に表示する必要があることに注意してください。

mode

動作モード。デフォルト=-1。有効な値は-1, 0, 1, 2ですが、スタンドアロン動作で意味があるのは-1と1だけです。0と2は、フロントエンドからシミュレータプロセスを開始するときに使用されます。-1 (/guiを省略するのと同じ) はシミュレータをコンソールモードで実行し、すべてのメッセージをコンソールまたはターミナルウィンドウに出力します。1はGUIモードを有効にし、シミュレータはスタンドアロンモードで動作しますが、メッセージとシミュレータの進行状況を示すグラフィカルステータスボックスを表示します。このモードは、フロントエンドの「非同期」メニューで使用されます。

analysis_line

/anスイッチを指定すると、*analysis_line*は実行する解析を指定し、

ネットリストで指定されたすべての解析ラインを上書きします。

<i>list_filename</i>	リストファイルの名前。デフォルトは、拡張子が.OUTのメインネットリストファイル名です。パス名にスペースが含まれる場合は、パス名を引用符で囲みます。
<i>options</i>	.OPTIONSステートメントに有効なオプションのリスト。
<i>netlist_file</i>	ネットリストのファイル名
<i>data_file</i>	バイナリデータ出力を受け取るファイル
<i>/check</i>	指定すると、ネットリストが読み込まれて構文解析されますが、シミュレーションは実行されません。構文を確認するために使用します。
<i>/nolist</i>	指定すると、リストファイルは作成されません。
<i>/lowPriority</i>	指定すると、シミュレータは優先度の低いプロセスとして、つまりバックグラウンドで実行されます。長時間の実行に推奨されます。
<i>/nodata</i>	.KEEPまたは.PRINTを使用して明示的に指定されたベクトルのみがバイナリファイルに出力されます。ネットリストの「.KEEP /nov /noi /nodig」に相当します。
<i>/k</i>	指定すると、ユーザがEnterキーを押してその旨のメッセージが表示されるまで、プログラムは最終的に終了しません。Windowsでは、プログラムがDOSプロンプトからではなく、別のプログラムから呼び出された場合、メッセージを受信するためのコンソールが作成されます。プログラムが終了すると、ユーザがメッセージを読む前に、コンソールが閉じることがあります。このスイッチはプログラムの終了を遅らせるので、コンソールの消滅を遅らせます。
<i>extra_line</i>	ネットリストに付加される追加行。これにより、ネットリストを簡単にカスタマイズできます。行にスペースが含まれる場合は、二重引用符で囲む必要があります。

2.2.4 中止

cntrl-Cを押すと、確認を求められます。ユーザの応答を待っている間、シミュレーションは一時停止され、「No」を入力すると続行されます。これは、別のタスクにCPUサイクルが必要な場合、またはデータファイルをコピーしたい場合に、実行を一時停止する効果的な手段です。6ページの「データの読み込み」を参照してください。

2.2.5 データの読み込み

通常どおり、シミュレーション結果のデータファイルが作成されます（12ページの「バイナリデータファイル」を参照）。シミュレーションが完了した後、SIMetrixメニューの**File | Data | Load...**を使用して、このファイルを読むことができます。シミュレーションの実行中にこのデータファイルを読み込むこともできますが、最初にcntrl-Cを使用してシミュレーションを一時停止する必要があります。

重要：シミュレーションが完了するか中止される前にデータファイルを読み込んだ場合、各ベクトルのサイズを提供するファイルエントリは埋められません。これは、波形ビューアがベクトルのサイズを確定するためにファイル全体をスキャンする必要があることを意味します。データファイルが大きい場合、これにはかなりの時間がかかります。

2.3 コンフィグレーション設定

コンフィグレーション設定は、多数の永続的なグローバルオプションと、インストールされているモデルライブラリの場所で構成されます。

シミュレータをGUIモードで実行すると、そのコンフィグレーション設定はフロントエンドによって制御され、フロントエンドの設定が保存されている場所に保存されます。詳細については、*User's Manual*を参照してください。

非GUIモードで実行した場合、設定はコンフィグレーションファイルに保存されます。このファイルは、実際には、フロントエンドの設定のデフォルトと同じ場所にデフォルトで保存されます。この場所を変更するには、4ページの「非GUIモードでの実行」で詳しく説明されている/configスイッチを使用します。

コンフィグレーションファイルの形式は次のとおりです。

```
[Options]
option_settings
```

```
[Models]
```

model_libraries

ここで、

option_settings これらは *name = value* の形式であり、いくつかのグローバル設定を指定します。ブール値は、値なしの *name =* の形式です。エントリが存在する場合は **TRUE**、存在しない場合は **FALSE** です。利用可能なグローバル設定の詳細を以下に示します。

model_libraries モデルライブラリの検索場所を指定するエントリのリスト。これらは、*name = value* の形式です。*name* は文字列で、*value* は検索場所です。*name* に使用される文字列は任意ですが、一意である必要があります。エントリは名前に従ってアルファベット順にソートされ、検索順序の決定に使用されます。*value* はパス名であり、ワイルドカード（つまり、「*」と「?」）を含めることができます。

2.3.1 グローバル設定

Name	Type	Default	Description
NoStopOnUnknownParam	String	WARN	Specifies action to be taken in the event of an unknown parameter being encountered in a .MODEL statement. Choices are: TRUE: No action taken, simulation continues normally FALSE: An error will be raised and the simulation will abort WARN: A warning will be displayed but the simulation will continue This will be overridden by a .OPTIONS setting of the same name. See the “List of Simulator Options” on page 224.
MaxVectorBufferSize	Numeric	32760	See “Data Buffering” on page 7.
TotalVectorBufferSize	Numeric	Available RAM/10	See “Data Buffering” on page 7.
TempDataDir	String	%STARTDIR%	Location of temporary binary data file if <i>data_file</i> is not specified on

command line

LibraryDiagnostics	String	Full	Controls output of messages relating to model library search. Specify None to disable
--------------------	--------	------	---

2.3.2 データバッファリング

シミュレータは、データをディスクに書き込む前にバッファします。これにより、バイナリデータファイルをより効率的に編成し、そこからデータを迅速に回復することができます。

デフォルトでは、シミュレータはシステムRAMの10%以上をベクトルバッファに割り当てません。明らかに、大規模な回路を実行して多くのベクトルを保存している場合、バッファサイズは、データの取得が非常に遅くなるレベルまで小さくなる可能性があります。この場合、バッファに割り当てられるメモリを増やすことを検討してください。次の2つのコンフィグレーション設定がベクトルバッファリングを制御します。

- **MaxVectorBufferSize** これにより、個々のベクトルに使用される最大サイズが設定されます。デフォルトは32768バイトです。高性能SCSIディスクシステムを使用している場合は、この値を大きくするとメリットが得られます
- **TotalVectorBufferSize** これにより、すべてのバッファに使用されるメモリの最大量がバイト単位で設定されます。デフォルトは、システムRAMの10%に等しい値です。通常、ほとんどのアプリケーションではこれで十分ですが、非常に大きな回路をシミュレートしていて、十分なRAMがある場合は、この値を増やしてください。

ディスクは、バッファがフルになるまで書き込まれません。オールアナログ回路を使用すると、すべてのバッファが同時にフル状態に達するため、すべてのバッファが同時にディスクに書き込まれます。2GのRAMがあり、大規模な回路をシミュレートしている場合、約200Mのデータが定期的にディスクに書き込まれます。これにより、シミュレーションは一時停止し、大量のディスクアクティビティが生じます。

Setコマンドを使用して、MaxVectorBufferSizeとTotalVectorBufferSizeの両方をフロントエンドから設定できることに注意してください。詳細については、*User's Manual/Sundry Topics/Using the Set and Unset Commands*を参照してください。

2.4 ネットリスト形式

SIMetrix ネットリストの形式は、すべてのSPICEおよびSPICE互換シミュレータで使用される一般的な形式に従います。ただし、非常に多くのSPICE派生物と、SPICE自体の大幅に異なる2つのバージョン（SPICE 2およびSPICE 3）では、標準のSPICE形式を定義することはできません。SIMetrixは、外部ソースから取得できるモデルライブラリと可能な限り互換性があるように開発されました。ディスクリートデバイスの場合、モデルは通常SPICE 2と互換性がありますが、一部はPSpice®用に開発された拡張機能を使用します。IC設計者は通常、製造会社からモデルファイルを受け取り、これらは通常Hspice®を含むさまざまなシミュレータで利用できます。SIMetrixはこれらすべてと互換性がありますが、いくつかの構文の詳細（インラインコメントで使用される文字など）のため、すべての形式との同時互換性は技術的に不可能です。これらの小さな問題を克服するために、ネットリストの先頭と、.INCまたはHspice®バリエーションの.LIBを使用してインクルードされるファイルに、言語宣言を配置することができます。これについては、次のセクションで説明します。

2.4.1 ファイル形式

完全なネットリストは次のもので構成されます。

- タイトル行
- オプションの言語宣言
- デバイス行
- ステートメント行
- コメント行

タイトル行はファイルの最初の行でなければなりません、空でもかまいません。残りの行は、いくつかの例外を除いて、任意の順序で配置できます

他のすべての行は、次のように最初の非空白文字によって定義されます。

- ステートメント行はピリオドで始まります: '.'
- コメント行はアスタリスクで始まります: '*'
- デバイス行は文字で始まります

行は通常、改行文字で終了しますが、「+」継続文字を使用して継続できます。したがって、最初の非空白文字が「+」の場合、その行は前の行の延長と見なされます。SPICEでは、最初の文字に「+」が必要です。SIMetrixでは、空白（スペースまたはタブ）を前に付けることができます。

2.4.2 言語宣言

SIMetrixはPSpice®、Hspice®, およびネイティブのSIMetrixネットリストを読み取ることができますが、場合によってはどの形式のネットリストを読み取るかを指示する必要があります。現在、同時互換性が不可能な3つの領域があります。これらは次のとおりです。

- インラインコメント文字
- ラベルなしのデバイスパラメータ
- 関数LOG()とPWR()の意味

SIMetrixは、言語宣言を使用して、3つの言語のいずれかを使用するように指示できます。これは次のいずれかです。

```
*#SIMETRIX *#HSPICE *#PSPICE
```

言語宣言は、ファイルの最上部のタイトル行のすぐ下に配置する必要があります。また、.INCまたはHSPICE®バージョンの.LIBを使用して参照されるファイルに配置することもできます。この場合、そのファイルとそれが呼び出す他のファイルにのみ適用されます。ファイル内の他の場所に配置された言語宣言は無視されます。詳細については、33ページの「言語の違い」を参照してください。

*#SIMETRIX言語宣言には、デバイスに使用される区別文字を指定するパラメータを設定することもできます。詳細については、9ページの「デバイス行」セクションを参照してください。

2.4.3 コメント

「*」で始まる言語宣言以外の行はコメントとして定義され、無視されます。また、セミコロン「;」（HSPICEモードでは「\$」）から行末まではコメントとして扱われ、無視されます。一部のSPICEシミュレータでは、行の最初の文字に「*」文字が必要です。SIMetrixでは、空白（スペースとタブ）を前に付けることができます。

2.4.4 デバイス行

通常、デバイス行は次の基本的な形式に従いますが、各タイプのデバイスには独自のニュアンスがあります。

```
Name nodelist value [parameters]
```

*value*は、たとえば抵抗などの受動部品の場合には実際の数値、たとえばバイポーラトランジスタなどの半導体デバイスの場合にはモデル名です。モデルは、.MODELステートメント行を使用して定義されます。

*nodelist*はネット名のリストです。これらの数と順序はデバイスに依存します。ネット名自体は、空白と「.」を除く非制御ASCII文字の集まりで構成できます。他のすべてのASCII文字は受け入れられますが、可能であれば次の文字を避けることを推奨します。

\ " \% \& + - * / ^ < > [] ' @ { }

上記の文字のいずれかがネット名で使用されている場合、そのネットの信号電圧をプロットするには特別な構文が必要になります。これについては、12ページの「出力データ名」で説明します。さらに、文字「[」,「]」,「%」,「!」および「」は、XSPICEデバイスで使用する場合に特別な意味を持つため、常に避ける必要があります。

*name*はデバイスの回路参照です。この名前の最初の文字は、次の表に示すようにデバイスのタイプを決定します。

次の表の「Pin Names」列は、デバイスのピン電流の値に使用されるベクトル名に関連しています。12ページの「出力データ名」を参照してください。

Letter	Number pins	Device	Pin Names
A	Any	XSPICE devices	depends on device
B	2	Arbitrary source	P, N
C	2	Capacitor	P, N
D	2	Diode	P, N
E	4	Voltage controlled voltage source	P, N, CP, CN
F	2	Current controlled current source	P, N
G	4	Voltage controlled current source	P, N, CP, CN
H	2	Current controlled voltage source	P, N
I	2	Fixed current source	P, N
J	3	JFET	D, G, S
K	0	Coupling for inductors	
L	2	Inductor	P, N
M	4	MOSFET	D, G, S, B
N	-	Not used	
O	4	Lossy transmission line	P1, N1, P2, N2
P	-	Not used	
Q	3-5	Bipolar transistor	C, B, E, S, DT

R	2	Resistor	P, N
S	4	Voltage controlled switch	P, N, CP, CN
T	4	Lossless transmission line	P1, N1, P2, N2
U	Any	VSXA devices (Verilog-HDL interface), Verilog-A devices, AC Table device	
V	2	Voltage source	P, N
W	-	Not used	
X	Any	Subcircuit	
Y	-	Not used	
Z	3	GaAs FET IGBT	D, G, S C, G, E

このシステムによって課せられる命名制限を除くために、SIMetrixはユーザがすべてのデバイスに任意の名前を使用できるように、上記の拡張機能をサポートしています。デバイス文字の後に「\$」記号が続く場合（デフォルトですが、変更可能-以下を参照してください）、「\$」に続く名前の残りの部分がデバイス名として使用されます。たとえば、Q\$TR23は、TR23という名前のバイポーラトランジスタを定義します。シミュレータによって生成されるすべての出力は、Q\$TR23ではなくTR23を参照します。

上記のメカニズムは無効にできます。また、言語宣言にパラメータを追加することで文字を変更できます（9ページの「言語宣言」を参照）。無効にするには、下記をネットリストの一番上に追加します。

```
*#SIMETRIX sep=none
```

使用する文字を変更するには、

```
*#SIMETRIX sep=character
```

*character*は1文字でなければならず、その他は無視されます。任意の文字が受け入れられますが、数字であってはなりません。

HSPICE言語またはPSPICE言語が指定されている場合、上記のメカニズムも無効になります。

2.4.5 シミュレータステートメント

デバイス定義とコメント以外のシミュレータへの指示はステートメントと呼ばれ、常にピリオド「.」で始まります。

SIMetrixステートメントの完全なドキュメントは、191ページの「コマンドリファレンス」を

参照してください。

2.5 シミュレータ出力

2.5.1 リストファイル

SIMetrixはデフォルトでリストファイルを作成します。これは、モンテカルロログを除くすべてのテキスト出力を受け取ります。これには、動作点の結果、モデルパラメータ、ノイズ解析結果、感度解析結果、極・零点解析結果、および.PRINTで指定された表形式ベクトルが含まれます。

リストファイルは、ネットリストと同じディレクトリに生成されます。名前はネットリストと同じですが、拡張子は.OUTです。

リストファイルの出力を制御する多くのオプションがあります。

Option name	Description
PARAMLOG	Valid values: Full All instance and model parameter values reported Given All user specified model parameters and parameterised instance parameters Brief Parameterised model and instance parameters None None Default = Given
EXPAND	Flag. If specified, the netlist with all sub-circuits expanded will be output to the list file
EXPANDFILE	String. If specified the expand netlist will be output to the specified file rather than the list file
NOMOD	Same as PARAMLOG=none. Model parameters will not be output
WIDTH	Page width in number of characters. (The list file is formatted assuming that it will be read or printed using a fixed width font such as Courier.) The default is 80 but any value may be used not just 80 and 132 as in SPICE 2.
OPINFO	If set DC operating point info file <i>is</i> created for all analyses (except .SENS). Normally it is created only for .OP analyses

2.5.2 バイナリデータファイル

シミュレーションデータはバイナリデータファイルに保存されます。このフォーマットはSIMetrix独自のものであり、SPICEの「生」ファイルとは互換性がありません。

バイナリファイルの名前と場所は、コンフィグレーション設定とシミュレータの実行モードによって異なります。通常、ファイルはTEMPDATADIRコンフィグレーション設定（6ページの「コンフィグレーション設定」を参照）で指定されたディレクトリにあり、解析タイプに従って名前が付けられ、拡張子.sxdatが付加されます。たとえば、tran1.sxdat、ac2.sxdat、dc3.sxdatなどです。名前と場所は、非GUIモードで動作している場合はプログラムのコマンドラインで、GUIモードで実行している場合はフロントエンドのRunコマンドラインで上書きできます。

SIMetrixフロントエンドのみがシミュレータのバイナリデータファイルを読み込むことができます。GUIモードで実行すると、ファイルは自動的にロードされます。ファイルが非常に大きくなりディスクがいっぱいになる場合を除いて、実際にファイルについて知る必要は通常ありません。シミュレータを非GUIモードで実行する場合、実行が完了したときに明示的にデータをフロントエンドにロードする必要があります。これは、コマンドシェルメニューの**File | Data | Load...**で実行できます。データがロードされた後、結果は通常の方法でプロットできます。詳細については *User's Manual/Graphs, Probes and Data Analysis/Saving Data/Restoring Simulation Data* を参照してください。

2.5.3 出力データ名

過渡解析、DC解析、AC解析の場合、SIMetrixは回路のノード電圧とデバイスのピン電流を計算して保存します。これらにはすべて一意の名前が付けられています。フロントエンドの回路図エディタでプロービングテクニックを使用する場合、使用する名前について何も知る必要は通常ありません。ただし、これらの名前がどのように派生したかを知ることが必要または役立つ状況があります。一つの例は、.PRINT文で使用される電圧と電流に関連する式をコンパイルする場合です。別の例は、回路図エディタを使用して生成されなかったネットリストをシミュレートして、作成された結果をプロットする場合です。使用される名前は、次の注意事項に記載されています。

最上位のノード電圧

最上位（つまり、サブ回路ではない）でノード電圧に使用されるベクトル名は、ネットリストで使用されるノードの名前です。

サブ回路のノード電圧

サブ回路内のノードの場合、名前の前にはサブ回路参照と「.」が付きます。たとえば次のとおりです。

```
X1 N1 N2 N3 SubName
X2 N4 N5 N6 SubName

.SUBCKT 1 2 3 SubName
X3 N1 2 N3 SubName2
R1 VIN 0 1k
...
.ENDS

.SUBCKT 1 2 3 SubName2
V1 VCC 0 5
...
.ENDS
```

X1によって参照される定義SubNameの内部ノードVINは、X1.VINと呼ばれます。X2によって参照される同じノードは、X2.VINと呼ばれます。サブ回路SubName2で定義されたノードVCCは、それぞれX1およびX2に対してX1.X3.VCCおよびX2.X3.VCCという名前になります。

非標準の名前を持つノード

非標準のノード名は、数字で始まるか、次の1つ以上の文字を含む名前です。

```
\ " \% \& + - * / ^ < > [ ] ' @ { }
```

これらは適切ですが、伝送する電圧データにアクセスするときに問題が発生します。上記の文字は算術式で使用できるため、ノード名として使用すると競合が発生します。そのような名前のノードの電圧データにアクセスするには、Vec()関数を使用します。

Vec(node_name)

下記は.PRINTとV+というノードの例です。

```
.PRINT TRAN {Vec('V+')}
```

フロントエンドのプロットコマンドを使用する場合、同様の構文が必要です。

デバイスピン電流

デバイスのピン電流は、次の形式で名前が付けられます。

```
device_name#pin_name
```

プリミティブデバイス（つまりサブ回路でない）の場合、`pin_name`は9ページの「デバイス行」の表に準拠する必要があります。たとえば、Q23のコレクタへの電流はQ23#cになります。

サブ回路のピン名は、デバイスのネットリストエントリに *pinnames:* 指定子（109ページの「サブ回路インスタンス」を参照してください）が含まれているかどうかによって異なります。含まれている場合、ピン電流名は *pinnames:* の後にリストされた名前になります。そうでない場合は、1から始まる順に番号が付けられます。順序は、ネットリストのデバイス行に表示される順序と同じです。

たとえば、サブ回路行が下記の場合、

```
X$U10 N1 N2 N3 N4 N5 LM324 pinnames: VINP VINN VP VN VOUT
```

最後のピン（N5に接続）への電流はU10#VOUTになります。

（9ページの「デバイス行」で説明したように、「X\$」は削除されることに注意してください）

ネットリスト行が下記の場合、

```
X$U10 N1 N2 N3 N4 N5 LM324
```

同じ電流は U10#5になります。

内部デバイス値

一部のデバイスには内部ノードまたはソースがあり、これらに関連する電圧または電流がシミュレータによって出力される場合があります。これらはピン電流と同様の方法で命名されています。つまり下記のとおりです。

```
device_name#internal_name
```

*internal_name*はデバイスによって異なります。たとえば、バイポーラトランジスタは、対応する抵抗パラメータを指定する各端子の内部ノードを作成します。したがって、REパラメータを指定すると、emitterと呼ばれる内部ノードが作成されます。

内部デバイス値は、「.KEEP /INTERNAL」ステートメントを使用して明示的に有効にした場合にのみ出力されることに注意してください。208ページの「.KEEP」を参照してください。

2.6 データ処理 - 保存されるデータの制御

12ページの「バイナリデータファイル」で説明されているように、すべてのデータはバイナリディスクファイルに保存されます。デフォルトでは、回路図に表示されるすべての信号が保存されます。つまり、階層の最上位とすべての子回路図にあるすべての信号が保存されます。階層的な回路図によって生成されなかったサブ回路内の信号は保存されません。

SIMetrixには、保存するデータを正確に変更するための包括的な機能があります。一部のシミュレーションでは大量のデータが生成され、マルチコア・マルチステップシミュレーションでは、データが作成される速度がディスクシステムの性能を超える場合があります。したがって、場合によっては、保存されるデータの量を減らすことが望ましいです。

ユーザインターフェイスから実行されるシミュレーションでは、GUIを介してデータ処理機能の一部を使用できます。 *User's Manual/Analysis Modes/Data Handling and Keeps* を参照してください。

.KEEPおよび.OPTIONSを使用して、より包括的な機能を利用できます。詳細については、208ページの「.KEEP」を参照してください。