

【オープン版】 回路シミュレーションガイドライン

第1.0版
(2025.08.25)

Copyright 2025 TOKAI RIKA CO., LTD

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

本書について

IP62プロセスの回路図作成とシミュレーションを行う方法について説明します

確認ツールバージョン

XSCHEM : 3.4.4

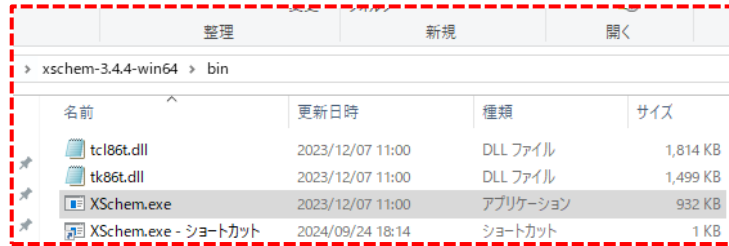
Ngspice : 41

目次

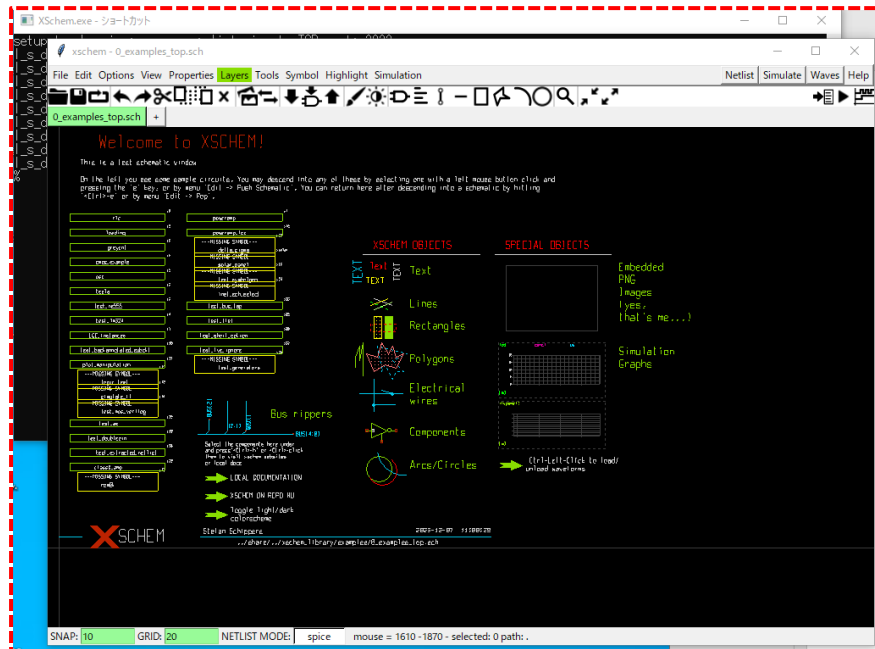
P.3	・・・	1. IP62ライブラリ/共通ライブラリの使い方
P.7	・・・	2. 回路図作成
P.9	・・・	3. 回路シンボル作成
P.10	・・・	4. シミュレーション回路作成
P.11	・・・	5. 回路シミュレーション
P.16	・・・	6. パラメータ初期値とコーナーモデル設定

1. IP62ライブラリ/共通ライブラリの使い方

①XSCHMの起動

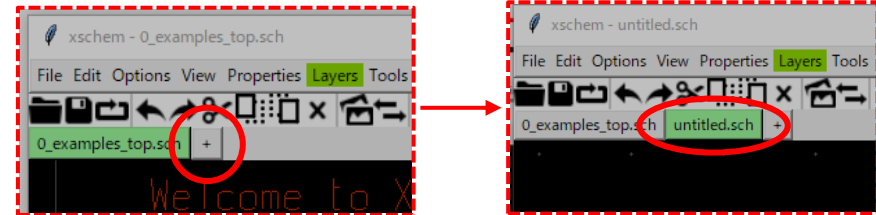


¥xschem-3.4.4-win64¥bin内のXSchem.exeから起動します。

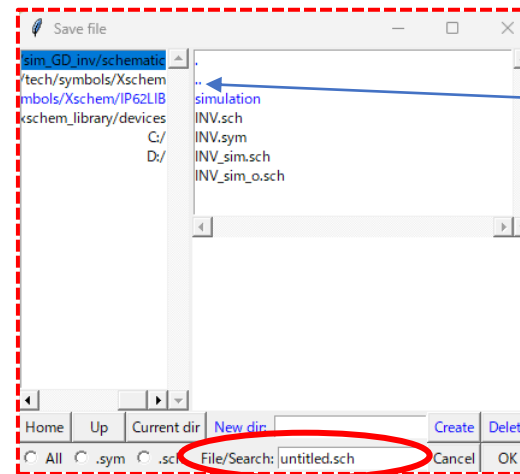


XSCHMとsaltの連携(インストールマニュアルp.3参照)ができていれば、上記のようなWelcome画面になります。

②Schematic(回路図)の作成



十字ボタンをクリックして、新規のschデータ「untitled.sch」を作成します。
ファイル名を変更するには File → save as → 保存するディレクトリを選択して File/searchの名前を変更し、OKを押します。

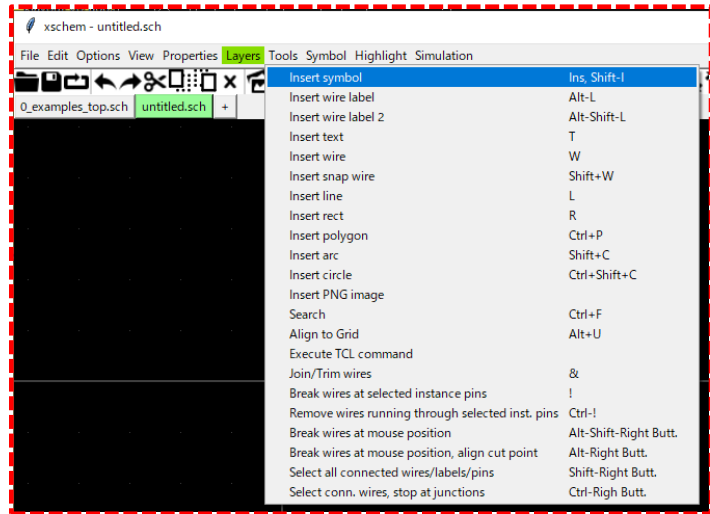


※画面下部に「NUM LOCK SET !」という表示が出ている場合操作が出来ません。
numlockをオフにしてください。

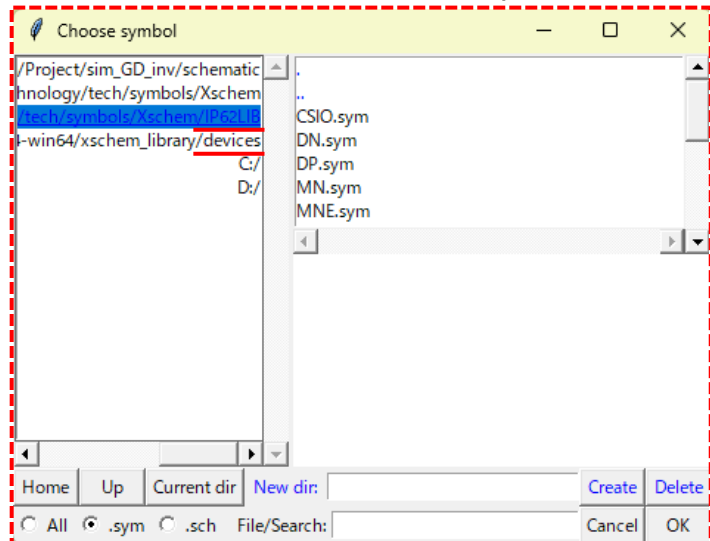


③ シンボルの呼び出し

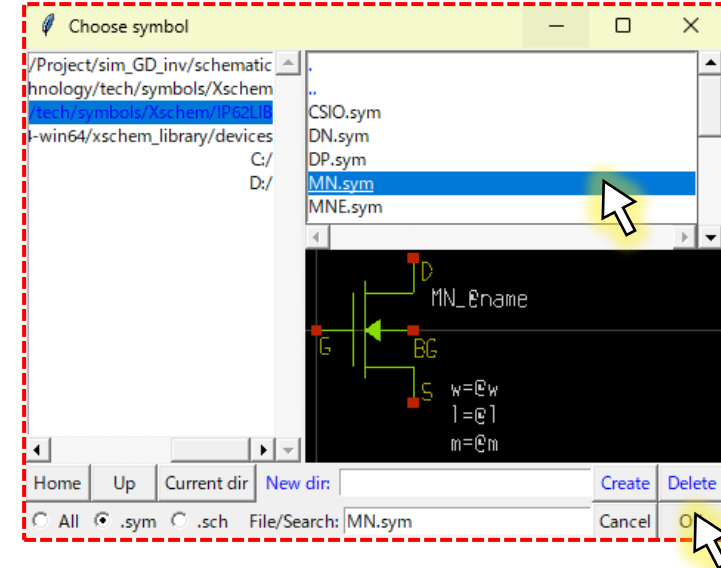
Shift + I、もしくはツールバーのTools→Insert symbolから素子ごとのシンボルを呼び出します。



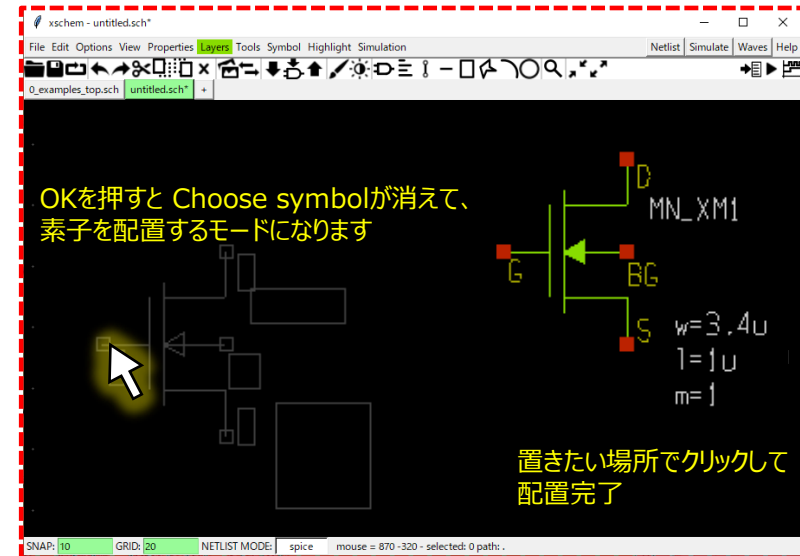
…/IP62LIB からIP62モデル、
 …/devices から通ライブラリが呼び出せます。(⑤シンボル一覧参照)



④ シンボルの配置



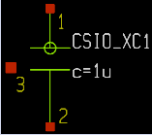
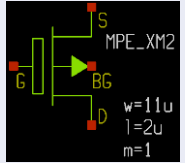
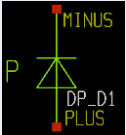
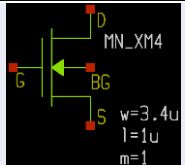
呼び出したい素子を選択、OKを押すと回路図上に素子の配置が可能になります。



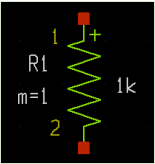
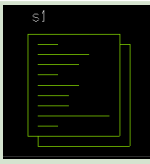

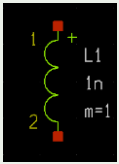
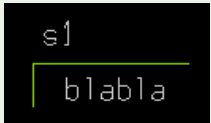

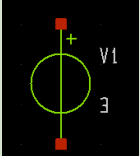


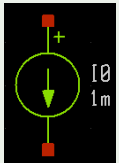

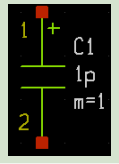

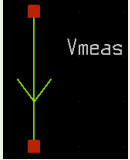

⑤シンボル一覧

シンボルは、IP62ライブラリに加えて、XSCHEMに標準搭載されているもの(共通ライブラリ)も使用します。
 ~\xschem\userConf\xschemrc の xschemrc追加.txt から追加した部分で、シンボルモデルを呼び出しています。

IP62ライブラリ

シンボル	素子名	シンボル	素子名	シンボル	素子名
	MP (5V PMOS)		CSIO (Capacitance)	非公開	
	MPE (ESD保護用 5V PMOS)		DP (ダイオード)		
非公開			DN (ダイオード(ゲート保護用))		RS (poly-si抵抗)
	MN (5V NMOS)	非公開		-	-
	MNE (ESD保護用 5V NMOS)			-	-
非公開			RR (拡散抵抗(P)、中抵抗)	-	-

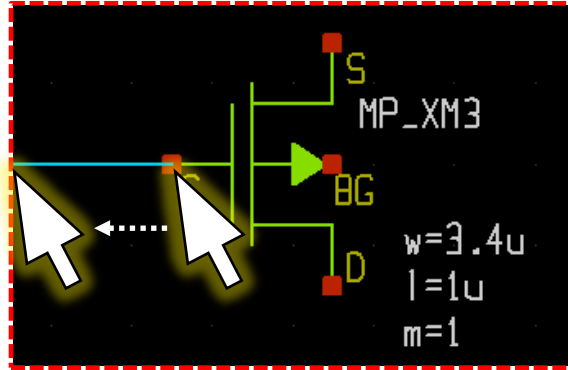
共通ライブラリ(XSCHEM標準搭載)

シンボル	素子名	シンボル	素子名	シンボル	素子名
	res (抵抗)		code (制御文) (ステートメント非表示)		ipin (入力ピン)
	ind (インダクタンス)		code_shown (制御文) (ステートメント表示)		iopin (入出力ピン)
	vsouce (電圧源)		lab_pin (ピン)		opin (出力ピン)
	isource (電流源)		lab_wire (ワイヤ名)	-	-
	capa (コンデンサ)		vdd (電源)	-	-
	ammeter (電流計)		gnd (グラウンド)	-	-

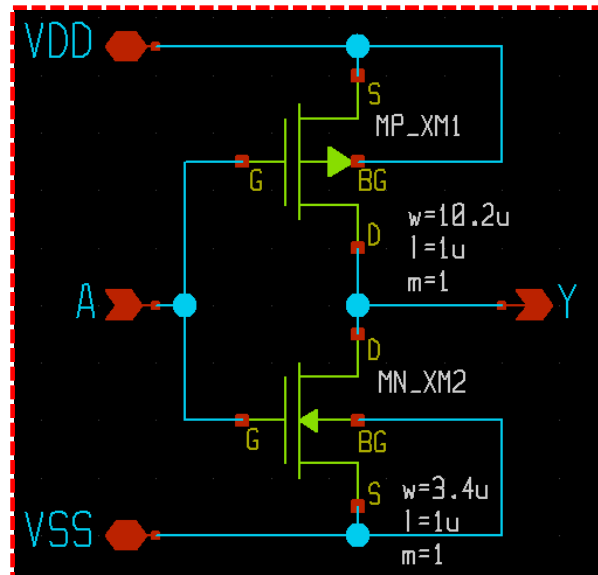
2. 回路図作成

①素子の接続

ワイヤを引き出したい場所にマウスを置いて、wキーでワイヤを引くことができます。(escまたはダブルクリックで終了。)



シンボル(MP,MN)を接続し、入出力と電源GNDにピンを置いたINVの例



②よく使用するXSCHMコマンド

・ シンボルの呼び出し	: shift + i
・ シンボルを消したい場合	: 選択してから delete
・ シンボルをコピーしたい場合	: 選択してから c
・ シンボルを移動させたい場合	: 選択してから m
・ シンボルの情報を見たい/編集したい場合	: 選択してから q、またはダブルクリック
・ シンボルの回転	: alt + r
・ シンボルの左右反転	: alt + f
・ 新しいシンボルの作成	: a
・ ひとつ前の状態に戻る	: u
・ ひとつ後の状態に進む	: shift + u
・ zoom full	: f
・ マウスで指定した範囲を拡大	: z
・ Zoom-in/out	: マウスホイール
・ Zoom-in	: shift + z
・ Zoom-out	: ctrl + z
・ 水平スクロール	: shift + マウスホイール
・ 垂直スクロール	: ctrl + マウスホイール
・ ワイヤを引く	: w
・ 画面の色の反転	: shift + o

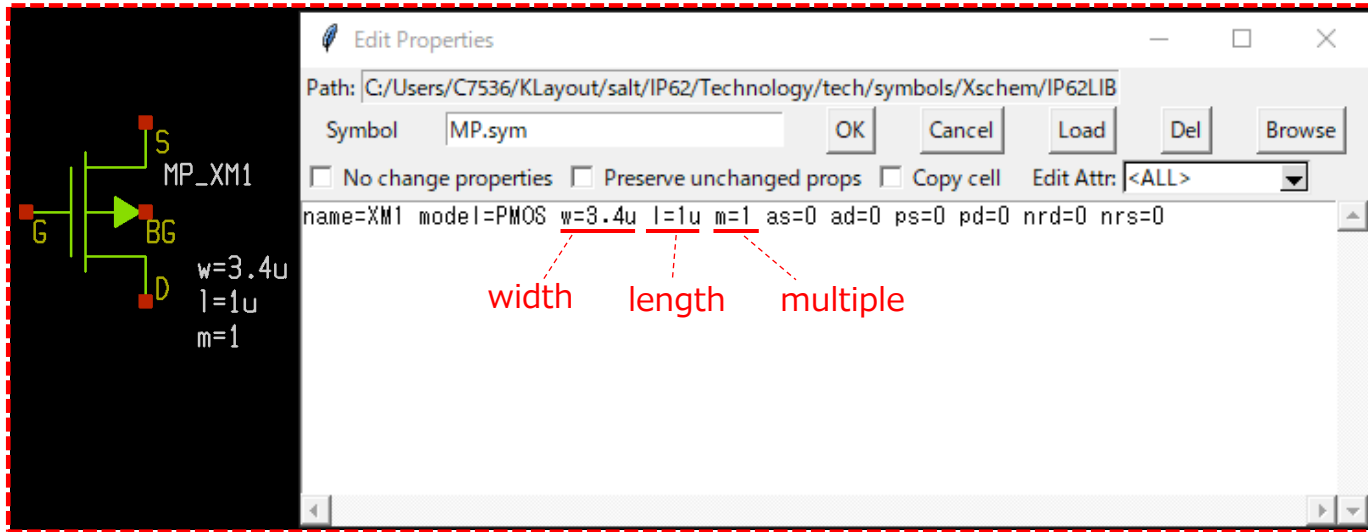
③素子のプロパティ変更

素子を選択してからqキーもしくはダブルクリックで素子のプロパティを表示します。

ここからwidth(w)、length(l)、multiple(m)など素子条件や、電源の電圧値などのsim条件、ラベル名、ピン名などを変更します。

リファレンスマニュアル 2.1 素子リストに示す W、L の範囲で設計してください。

※MOSの場合、w、l の値は、例えば 4um の場合は 4u と入力してください



素子の例

注意： 素子種を変更する場合は、modelを変更後、Saveが必要です。
SaveせずにSPICEファイルを出力する（5-③-2参照）と、model変更が反映されません。

●コールバック機能

<抵抗の場合>

リファレンスマニュアル 2.1 素子リストに示す W（固定値）、L の範囲で設計してください。

抵抗値の計算は、抵抗・容量計算ツールで行えます。

回路図上では、プロパティより w と l を入力すると、抵抗値が R として出力されます。

※抵抗の場合、
w、l の値は、例えば 4um の場合は 4e-6 と入力してください

<容量の場合>

コールバック機能はありません。

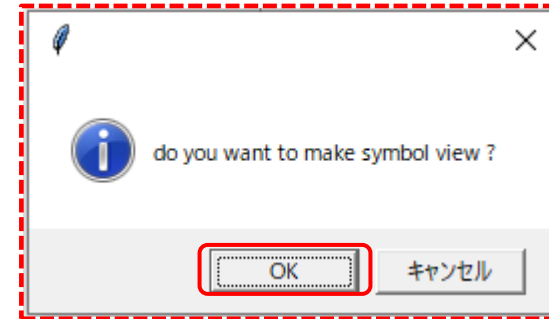
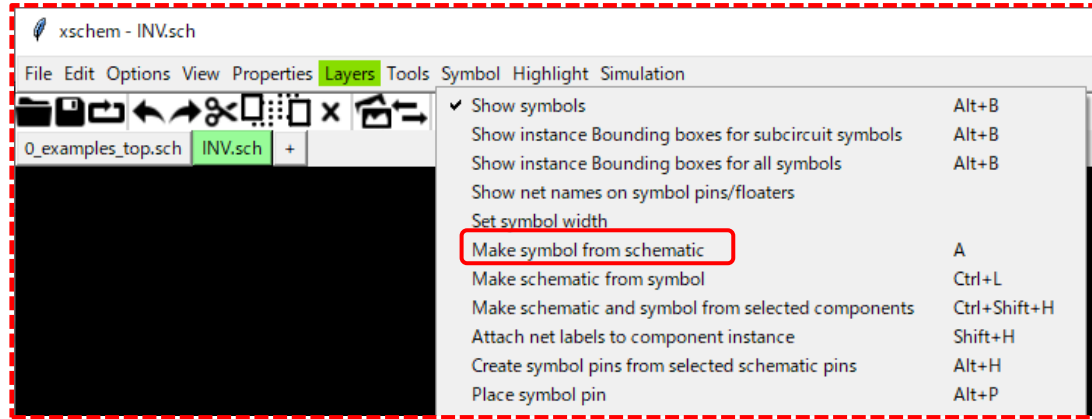
回路シミュレーションで必要なのは容量値のみです。

(レイアウト後に行うLVS検証にも、寸法情報(x,yの値)は不要です。)

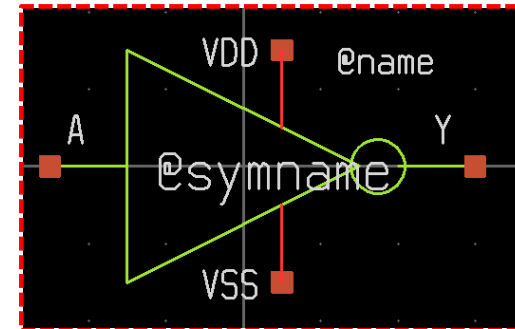
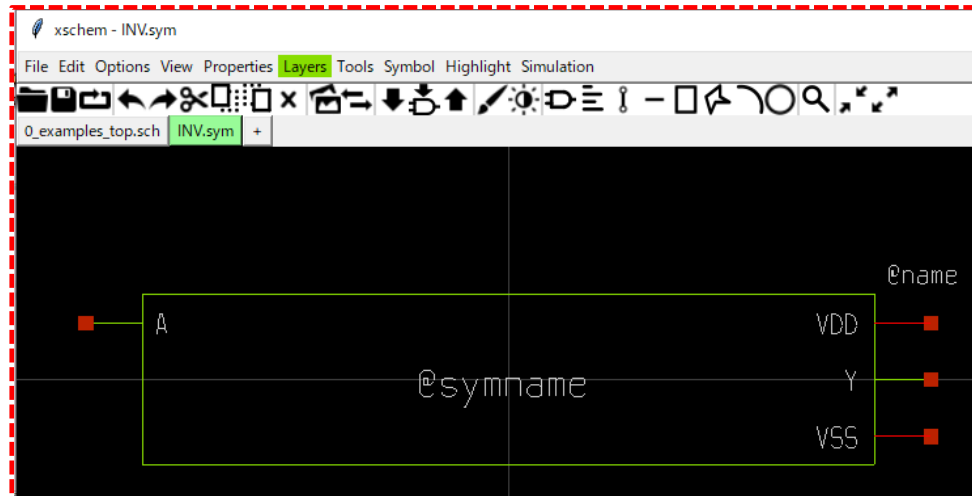
3. 回路シンボル作成

①Symbol > Make symbol from schematic または、ショートカット a を押す → OK

スキマ（例ではINV.sch）の保存先に、シンボル（例ではINV.sym）が保存されます。



②File > Open からシンボルを開くと、自動生成されたシンボルが見えますので、必要に応じて使いやすい形に修正します。

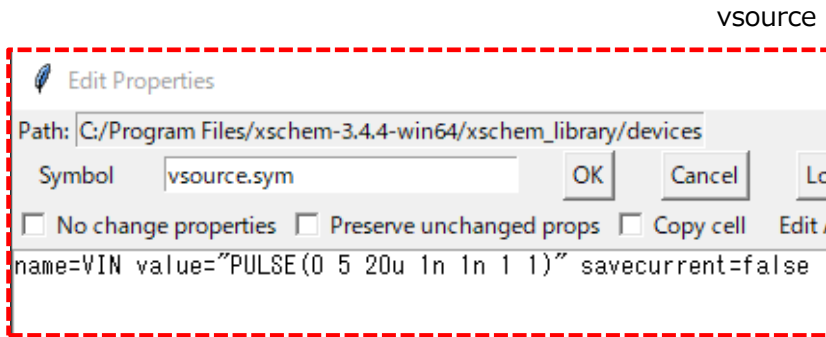


※シンボルを一からマニュアルで作成してxxx.symで保存する作り方も可能です。
 その場合、xxx.symをテキストエディタで開いて、K{ }の記述を以下のように編集する必要があります。
 K {type=subcircuit
 format="@name @pinlist @symname"
 template="name=x1"}

4. シミュレーション回路作成

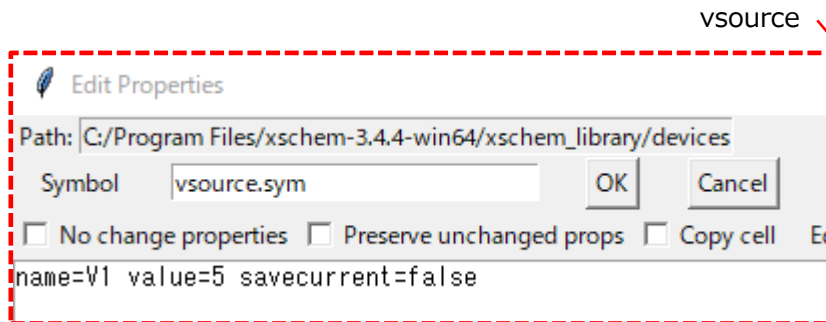
- ①新しい回路図を開きます（+タブ）
- ②shift + i で、シンボルを置き、w でワイヤでつなぎます。必要に応じて shift + i で Lab_wire をワイヤ上に置いてワイヤ名を付けます。
- ③各シンボルの設定を、シンボルをダブルクリックして変更します。

※ワイヤ名などはアルファベットの大文字・小文字が区別されません

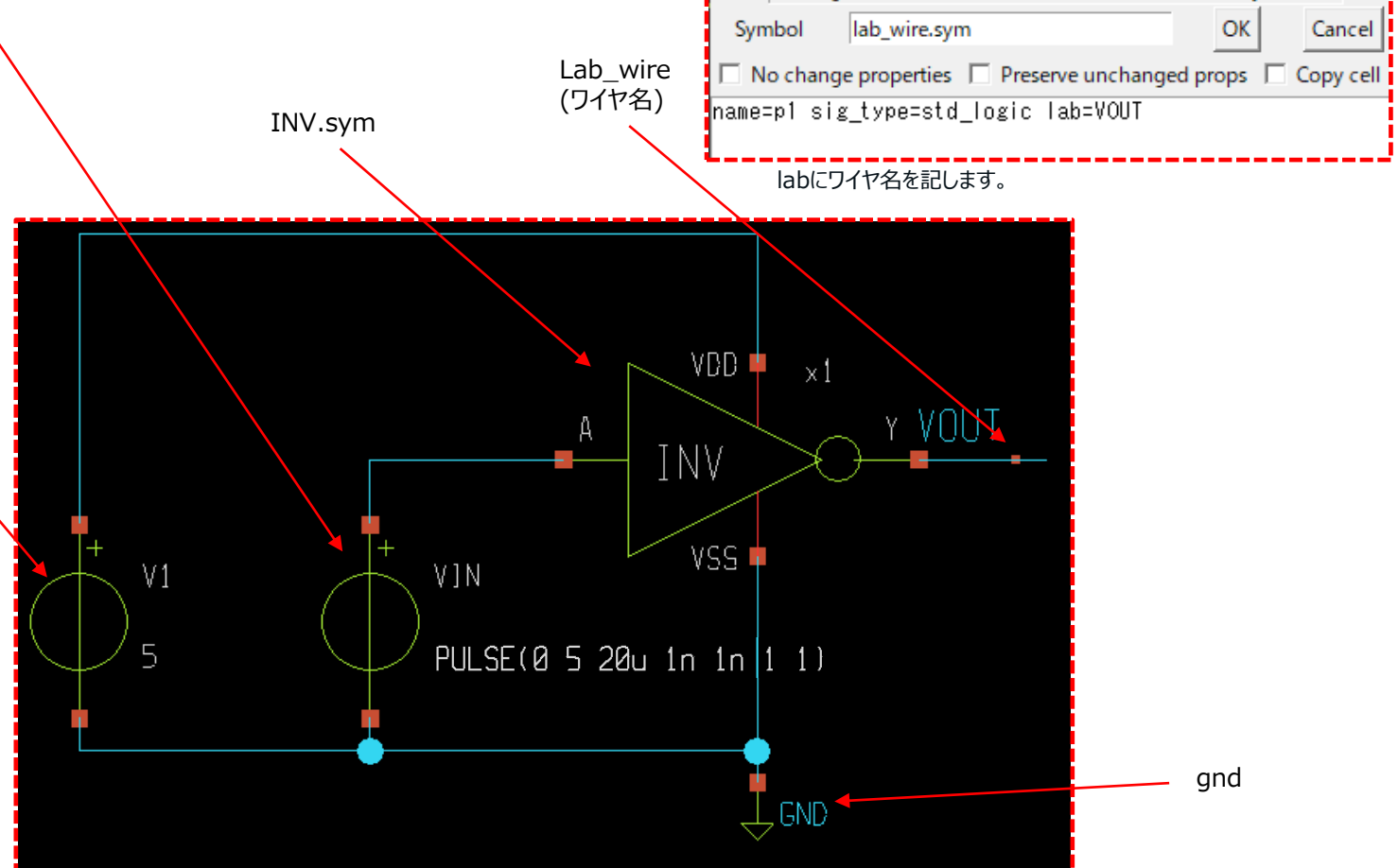


PULSEの中は、初期電圧[V] パルスのピーク電圧[V]
遅延時間[s] tr[s] tf[s] オン時間[s] 周期[s] を示しています。

※スペースを含む場合は""で囲う必要があります。



Valueは5V一定を意味します。

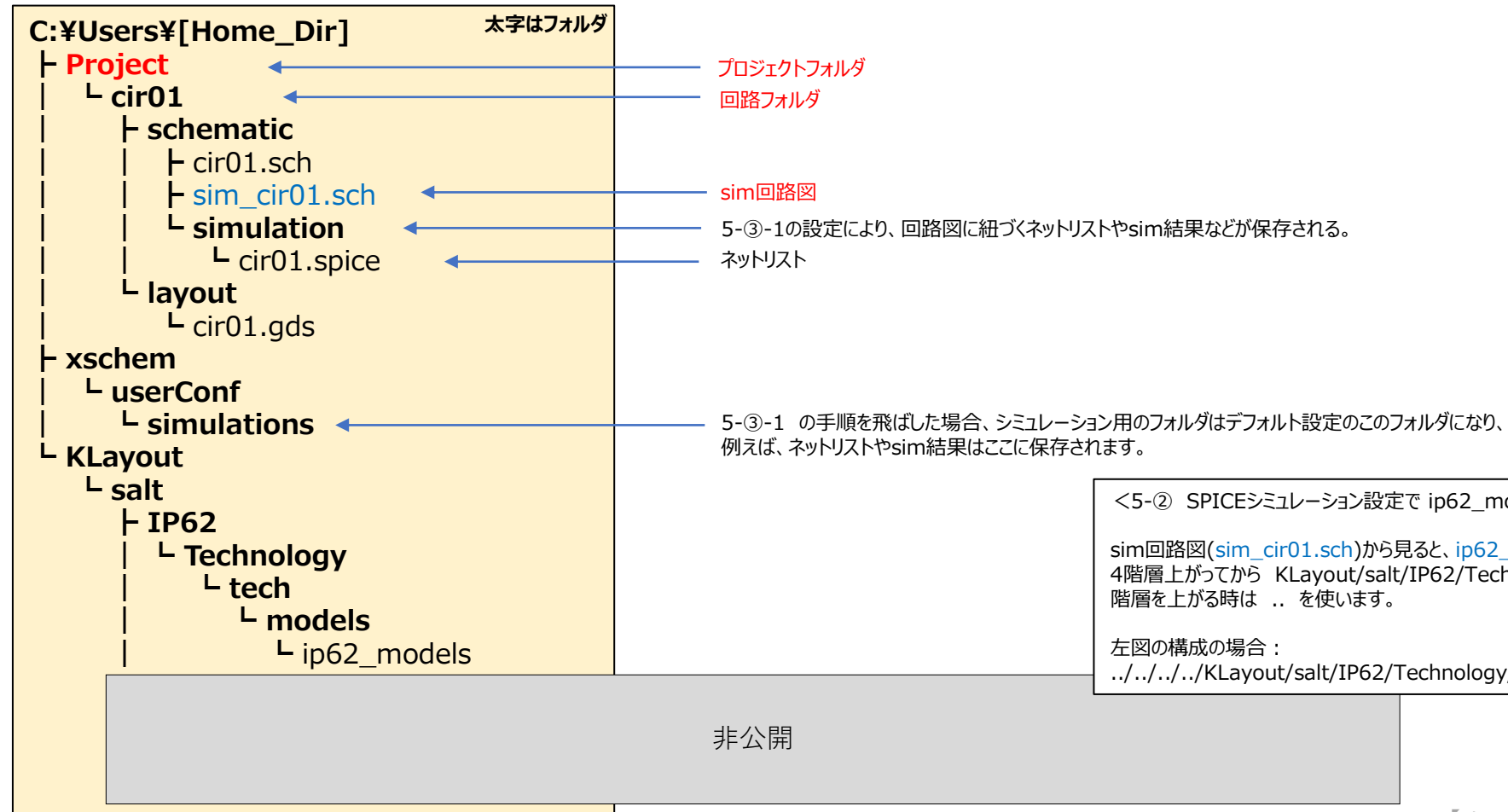


5. 回路シミュレーション

①プロジェクトフォルダを準備（任意）

回路図やsim結果などを管理しやすくするため、プロジェクトフォルダを作成します。フォルダパスに日本語が入らないように注意してください。構成に決まりはありません。

以降、本書では、以下の場所にプロジェクトフォルダ（例、**Project**）を作り、以下の構成で回路図を保存した場合を説明します。



<5-② SPICEシミュレーション設定で ip62_models を読み込む(.inc)時のヒント>

sim回路図(**sim_cir01.sch**)から見ると、**ip62_models**は、4階層上がってから KLayout/salt/IP62/Technology/tech/models/ と降りていくとあります。階層を上げる時は `..` を使います。

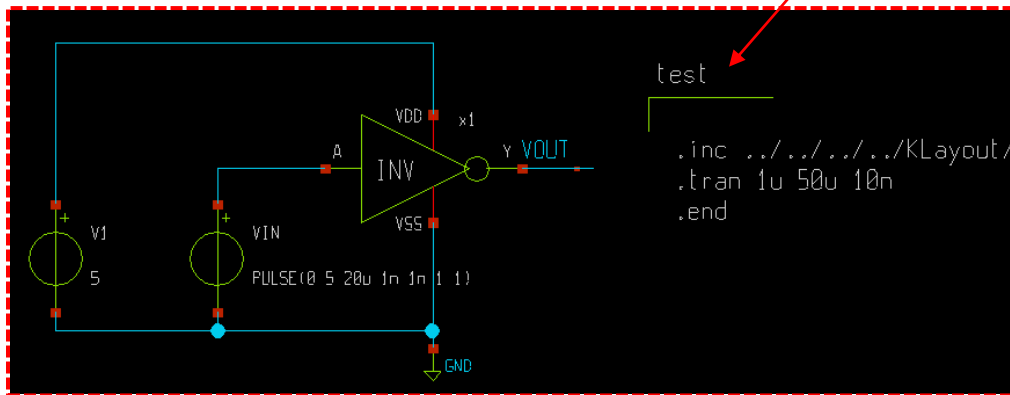
左図の構成の場合：

`../..../KLayout/salt/IP62/Technology/tech/models/ip62_models`

② SPICEシミュレーションの設定

SPICEシミュレーターにどのようなシミュレーションを行うかの指示を出すために、シミュレーションコードの追加が必須です。

シンボル “code_shown” もしくは “code” のどちらかを呼び出して使用します。
code_shown は名前の通りにスキーマ上で条件を見ることができます。
code はプロパティを開くことで条件が見えます。

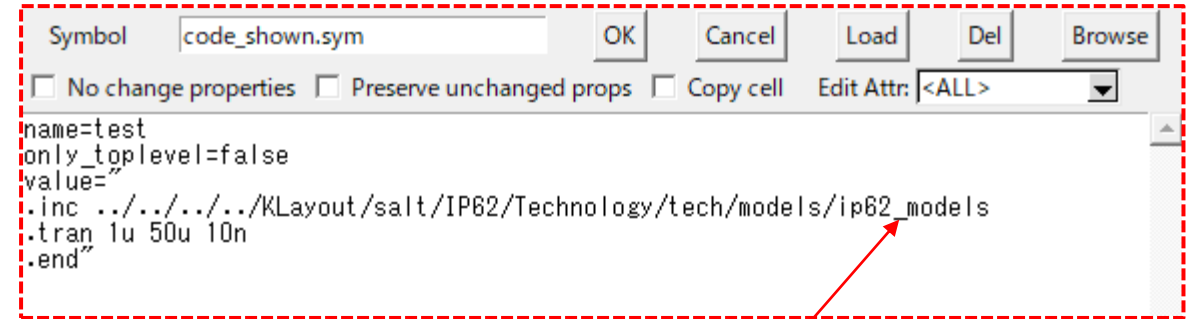


code_shown

test

```
.inc ../../../../../../KLayout/salt/IP62/Technology/tech/models/ip62_models
.tran 1u 50u 10n
.end
```

code_shown をダブルクリックして、シミュレーションコードを編集します。
例は、1usから50usまで10ns刻みでtran解析を行う場合です。



.inc で SPICEモデル (ip62_models) を読み込むため、回路図から見た ip62_models までのパスを入力します。

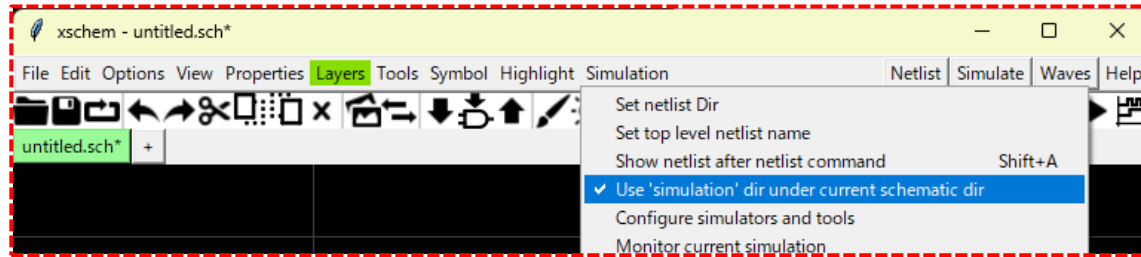
各コードの意味

- Name :** シミュレーションの名前を指定するためのフィールドです。
ここでは仮でtestという名前を付けました。シミュレーションの名前は、結果ファイルの識別や管理に役立ちます。
- only_toplevel :** このフィールドは、シミュレーションがトップレベルの回路だけを対象にするかどうかを指定します。
=true : トップレベルの回路のみをシミュレーションします。
=false : サブサーキットも含めて全体をシミュレーションします。
- Value :** 「」から「」までは、SPICEシミュレーションコードを含むフィールドです。
シミュレーションの内容を定義します。
- .inc :** 外部ファイルを読み込むincludeコマンドです。
主にシミュレーションで使用するモデルやサブサーキットをincludeします。ここではSPICEモデルを読み込んでいます。
- .tran :** トランジェント解析を行うコマンドです。DC解析の場合は.dc、AC解析の場合は.acとなります。
- .end :** シミュレーションの終了を示すコマンドです。これ以降の行は無視されます。

③ シミュレーションの実行

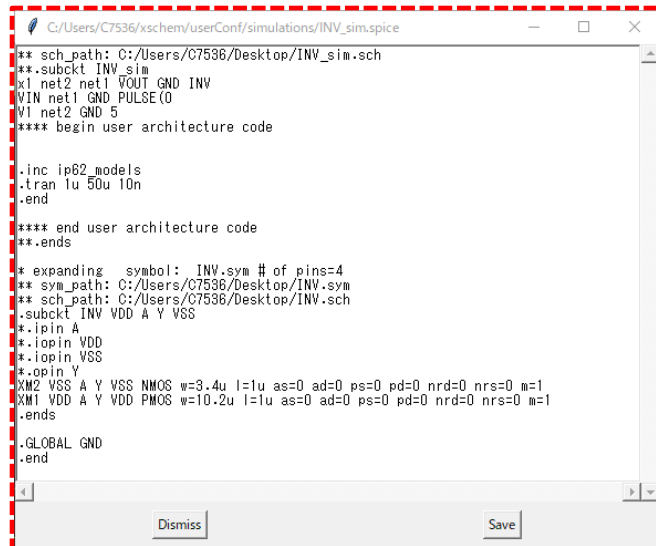
Ngspiceでのシミュレーションを行います。

- ③-1 メニューバー> Simulation> Use 'simulation' dir under current schematic dir をクリックしてチェックを入れると、この後の操作によって作られるネットリストやsim結果が、回路図が保存されているフォルダ下にあるsimulationフォルダに保存されます。

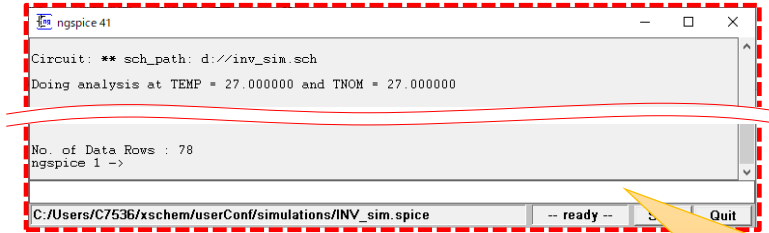


- ③-2 右上のNetlistボタンを押し、SPICEファイル（.spice）を作ります。

※メニューバー> Simulation> Show netlist after command を押してチェックを入れると、作成したSPICEファイルを表示、編集できます。



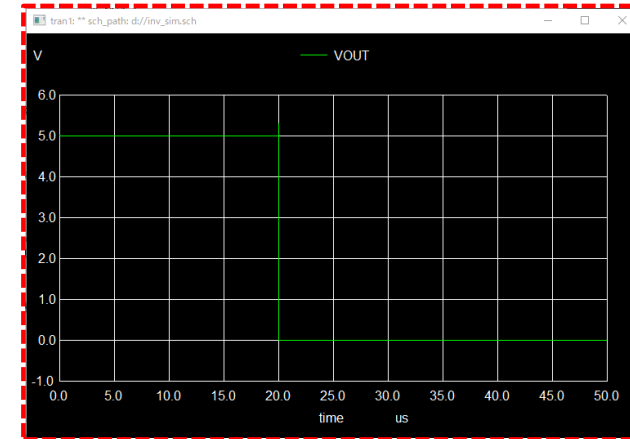
- ③-3 右上のSimulateボタンを押し、シミュレーションを流します。



④、⑤ コマンド入力欄

- ④ シミュレーション結果(波形)の確認

コマンド入力欄に、**plot net名** のコマンドを入れ、ENTERキーを押します。
例では **plot vout** と入力して、VOUTの電圧を確認します。（大文字/小文字は不問）



電流値を見る場合は、**plot i(電圧源名)**を入力します。
例えば、plot i(v1) など

- ④' シミュレーションコードによるシミュレーション結果(波形)の確認

```
GRAPH
.control
run
setplot VIN
plot v(VIN) ylimit -1 12
setplot VOUT
plot v(VOUT) ylimit -1 12
.endc
```

グラフを表示させる例です。
nameは任意です。
Sim後に、コマンド入力しなくても、自動でグラフが表示されます。

⑤シミュレーション結果の確認(波形のデータ)

検証後、波形のデータを見たい場合はコマンド入力欄に **print net名** のコマンドを入れ、ENTERキーを押します。

例えば print vout など

詳細なシミュレーション結果がNgspice上に出力されます。
(文字選択、右クリックでコピーできます)

一部抜粋

Index	time	vout
0	1.000000e-08	5.000000e+00
1	2.000000e-08	5.000000e+00
2	4.000000e-08	5.000000e+00
3	8.000000e-08	5.000000e+00
4	1.600000e-07	5.000000e+00
5	3.200000e-07	5.000000e+00
6	6.400000e-07	5.000000e+00
7	1.280000e-06	5.000000e+00
34	2.000117e-05	-9.09605e-03
35	2.000127e-05	2.231711e-03
36	2.000145e-05	-1.09909e-03
37	2.000182e-05	7.816623e-04
38	2.000257e-05	-6.59443e-04
39	2.000405e-05	6.062178e-04
40	2.000702e-05	-5.80818e-04

⑤' シミュレーションコードによるシミュレーション結果の確認(波形のデータ)

```
DATA
.control
run
wrdata data.raw v(VIN) v(VOUT)
.endc
```

生データを保存させる例です。
nameは任意です。
Sim後に、コマンド入力しなくても、
自動で生データが保存されます。

⑥各解析について

TRAN解析(過度解析) : 時間に対する回路の動作を解析

シミュレーションコードの書き方

.tran <ステップ時間> <シミュレーション時間> [<出力開始時間>] [<最大ステップ時間>]

例：ステップ時間を1ns、シミュレーション時間を1μsに設定する場合：**.tran 1n 1u**
[]は省略可能

AC解析(交流解析) : 回路の周波数特性を調べる解析

シミュレーションコードの書き方

.ac <解析タイプ> <ポイント数> <開始周波数> <終了周波数>

例：線形スイープで10ポイント、開始周波数を1Hz、終了周波数を1kHzに設定する場合：**.ac lin 10 1 1000**

解析タイプには以下があります

lin : 線形スイープ

dec : 対数スイープ (デケード)

oct : 対数スイープ (オクターブ)

DC解析(直流解析) : 回路の動作点を見つける解析

シミュレーションコードの書き方

.dc <ソース名> <開始値> <終了値> <ステップ値>

例：電圧源V1の電圧を0Vから10Vまで1V刻みで変化させる場合：**.dc v1 0 10 1**

その他 (一例)

パラメトリック解析 : 回路パラメータを変化させて、回路の動作を調べる解析。抵抗値や温度を .dcのソース名に指定します。

例：**.dc temp -40 125 5**

温度解析 : TRAN解析、AC解析、DC解析と組み合わせて温度を振ることができます。

例：**.dc v1 0 10 1**

.temp -40 25 100 125

6. パラメータ初期値とコーナーモデル設定

シンボルのパラメータ、及びモデルの値は、モデルのインクルードファイル `~/KLayout/salt/IP62/Technology/tech/models/ip62_models` にて設定します。
デフォルトの値は、以下です。

```
.parameters vthMP      = 0
.parameters vthMN      = 0
.parameters vthMPE     = 0
.parameters vthMNE     = 0
.parameters magRR       = 1
.parameters magRS       = 1
.parameters magCSIO     = 1
.include ngspice/models_IP62_cap_v5p1.lib
.include ngspice/models_IP62_diode_v2.lib
.include ngspice/models_IP62_mos_v2.lib
.include ngspice/models_IP62_res_v5.lib
```

バージョンは更新されることがあります

コーナーモデルは非公開です。

改定履歴

Rev	年月日	改訂内容
1.0	'25.08.25	・初版