

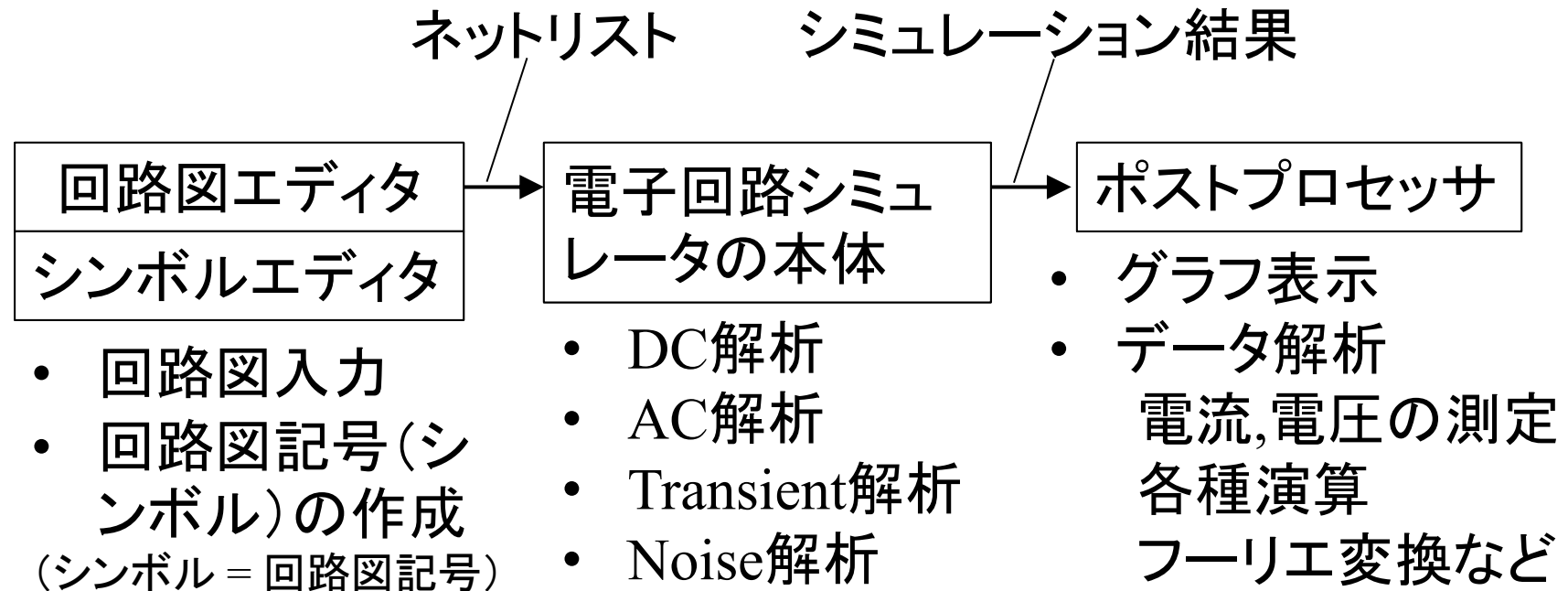
第2章 電子回路シミュレータ

電子回路シミュレータの基礎知識

回路設計の基本的なツールである電子回路シミュレータの仕組みと機能(参考書1-1参照)

2.1 電子回路シミュレータの機能

電子回路シミュレータの構造と機能



(参考) 電子回路シミュレータの本体は、カリフォルニア大学バークレイ校で開発されたSPICE(Simulation Program with Integrated Circuit Emphasis)のソースコードを元に開発されたものが多い。これらの回路シミュレータをSPICE系シミュレータと呼ぶ。これ以外の回路シミュレータは、非SPICEまたはPost-SPICEシミュレータと呼ばれる。

前スライドの専門用語

Schematic:	回路図 (Schematic diagram, Circuit diagramという用語もあるがSchematicと呼ぶのが普通)
Symbol:	回路記号
Netlist:	ネットリスト (回路図の部品接続情報)
Schematic entry:	回路図入力 (Schematic captureも同義)
Circuit simulator:	電子回路シミュレータ
DC analysis:	DC解析 (または直流解析)
AC analysis:	AC解析 (または小信号交流解析または周波数特性解析)
Transient analysis:	過渡応答解析 (またはトランジェント解析)
Noise analysis:	ノイズ解析 (またはノイズパワースペクトラム解析)
Postprocessor:	ポストプロセッサ (データの可視化処理やフォーマット変換などを行うソフト)

電子回路シミュレータの主な解析機能

実測するとき必要な計測器

解析の種類	得られる特性
DC解析	横軸が電圧または電流の特性(マルチメータ)
AC解析	横軸が周波数の特性(ネットワークアナライザ)
Transient解析	横軸が時間の解析(オシロスコープ)
Noise解析	横軸が周波数、縦軸が雑音電圧密度

- 他の解析に比べて、Transient解析は時間がかかる。トランジスタ数が1000個以上の場合、高性能PCを使用しても、数日かかることが多い
- 回路を設計するためには、色々な解析法を組み合わせる必要がある。初心者はTransient解析だけに頼りがちだが、これでは意図通りに動作する回路を設計できない

デバイスのモデルパラメータ

L, C, R

キャパシタンス $C = 10\text{pF}$

インダクタンス $L = 10\text{nH}$

抵抗 $R = 10\text{k}\Omega$

各デバイスの特性は、最低1個のパラメータで表せる。→回路図に書き込んでしまう。

ダイオード

飽和電流 $I_s = 100\text{pA}$

電位障壁 $V_j = 0.65\text{V}$

エミッション係数 $n = 1.2$

ゼロバイアス接合容量 $C_{jo} = 100\text{pF}$

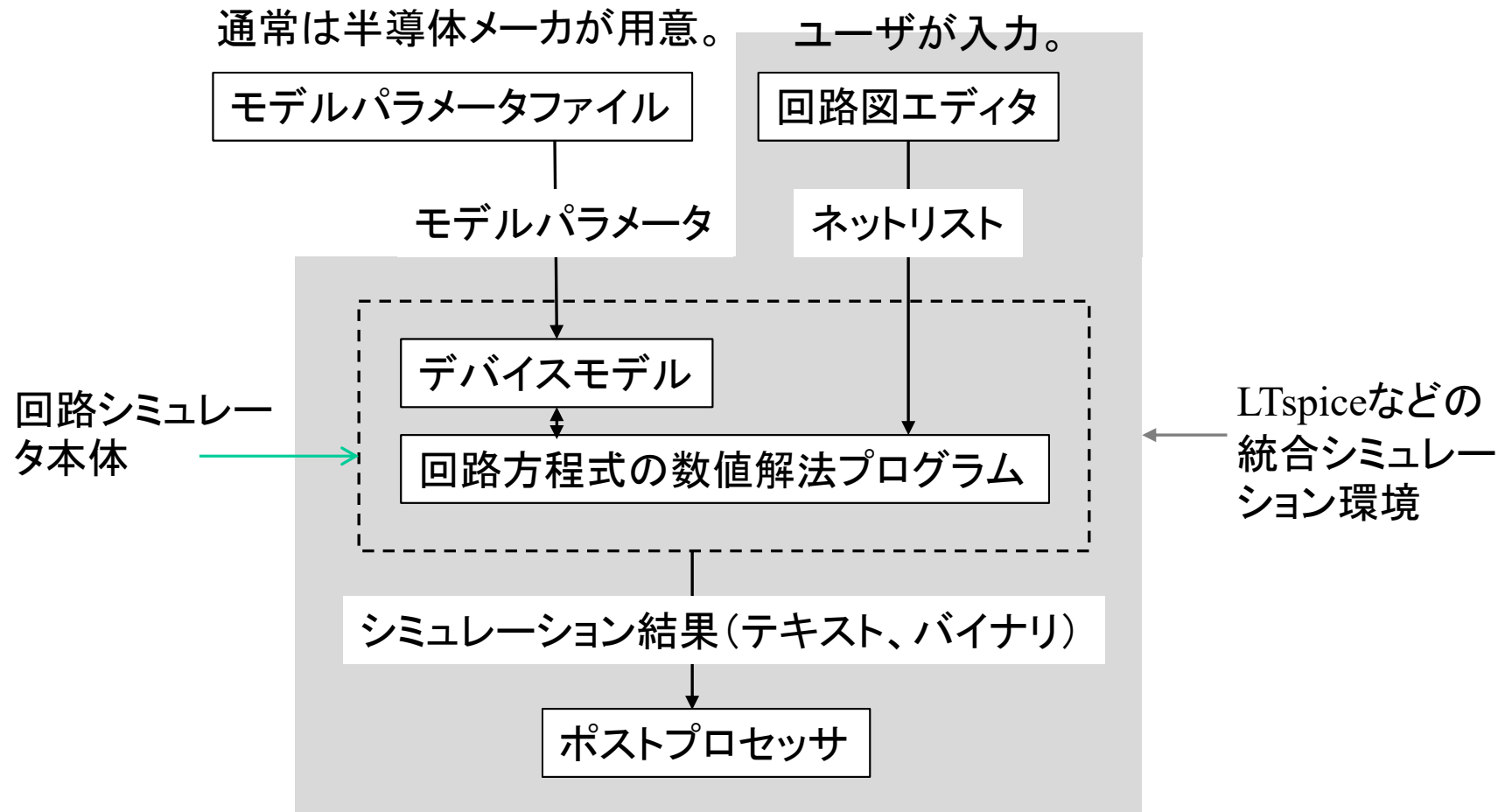
ブレークダウン電圧 $BV = 250\text{V}$

⋮

特性の計算には多くのパラメータが必要。→ [モデルパラメータファイルに記載してインクルード。](#)

トランジスタの特性計算には、数10個のパラメータが必要。

電子回路シミュレータの処理フロー



デバイスモデル(Device model):

半導体デバイスの特性を表す計算式。

モデルパラメータ(Model parameter):

デバイスモデルに含まれるパラメータ。

電子回路シミュレータの限界

- 電子回路シミュレータでシミュレーションできないケース
 - 温度が時間変化する場合や、部品によって異なる温度になる場合はシミュレーションできない
 - 過電圧、過電流によるデバイスの破損はシミュレーションできない
 - **絶対最大定格** (Absolute maximum rating, デバイスが破損する電圧、電流、電力の限界値) を越えてもシミュレーション上では回路が動作する
 - 実際の回路が故障しないためには、絶対最大定格に対する、マージン(設計余裕度)が必要
- 電子回路シミュレータの精度
 - 半導体デバイスのモデルパラメータの精度に依存している
 - 代入していないパラメータはデフォルト値になる
 - モデルパラメータの正確な推定は難しいため、半導体メーカーが全てのパラメータを提供しているとは限らない

回路シミュレータは現実とは違うことを理解した上で使用する必要がある。

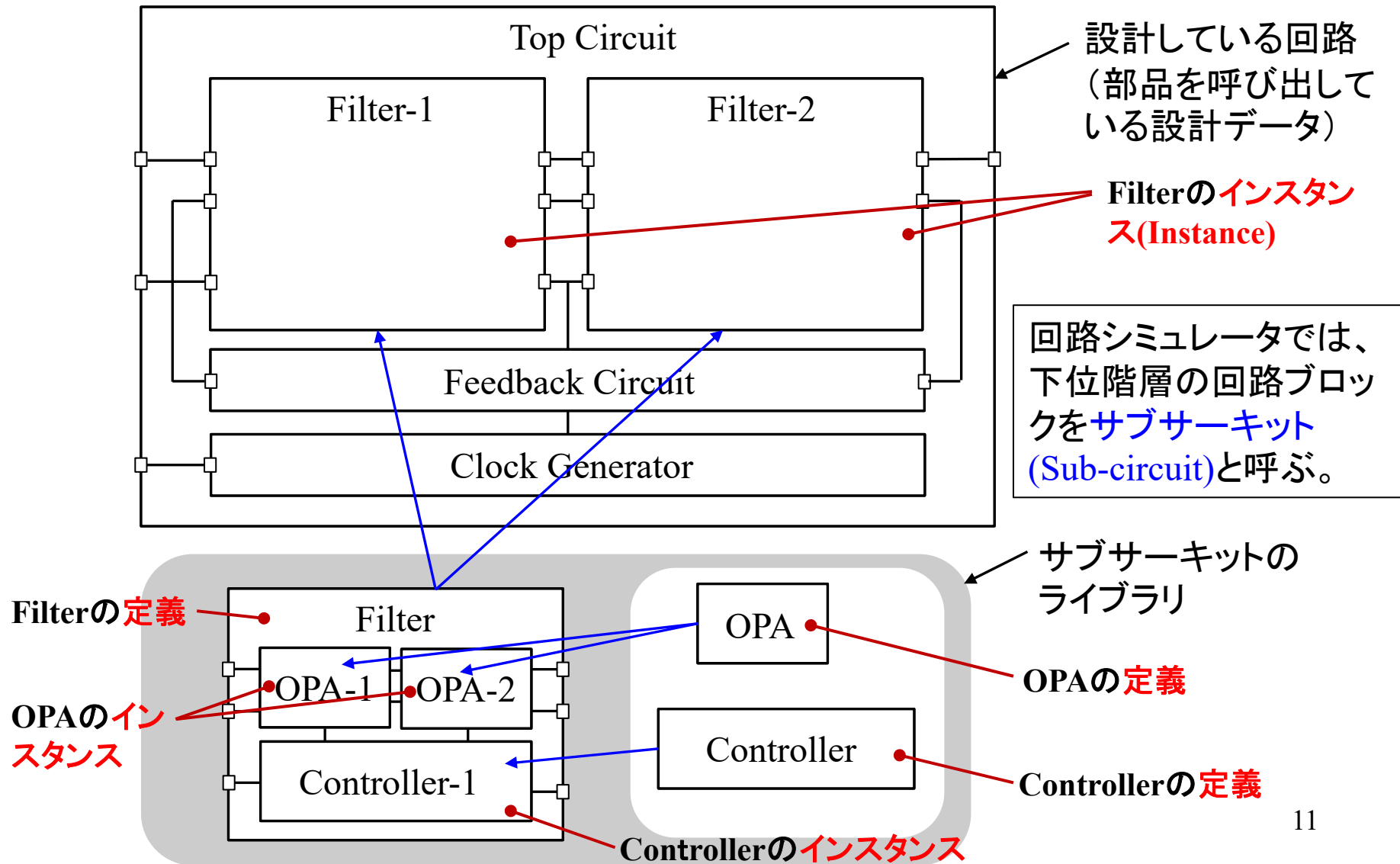
2.1節のまとめ

- 電子回路シミュレータは、回路図エディタ、シンボリエディタ、シミュレータ本体、ポストプロセッサから構成されている
 - 回路図エディタはネットリスト(回路の接続情報)を出力し、回路シミュレータ本体は、ネットリストから回路方程式を作成する
 - シミュレータ本体は、カリフォルニア大学が開発したSPICEをベースとしているものが多い
 - 主な解析機能にはDC解析(直流電圧、直流電流)、AC解析(周波数領域)、Transient解析(時間領域)、Noise解析(周波数領域)などがある
- 回路図の他に、デバイスモデルパラメータが必要
 - デバイス特性の計算のため、モデルパラメータファイルをインクルードする
 - モデルパラメータはデバイス特性の実測値から推定することができるが、デバイスモデルに関する専門知識が必要
 - 値を与えなかったパラメータがある場合、デフォルト値が使用されるためシミュレーション結果が正確ではない

複雑な電子回路を簡単化する方法

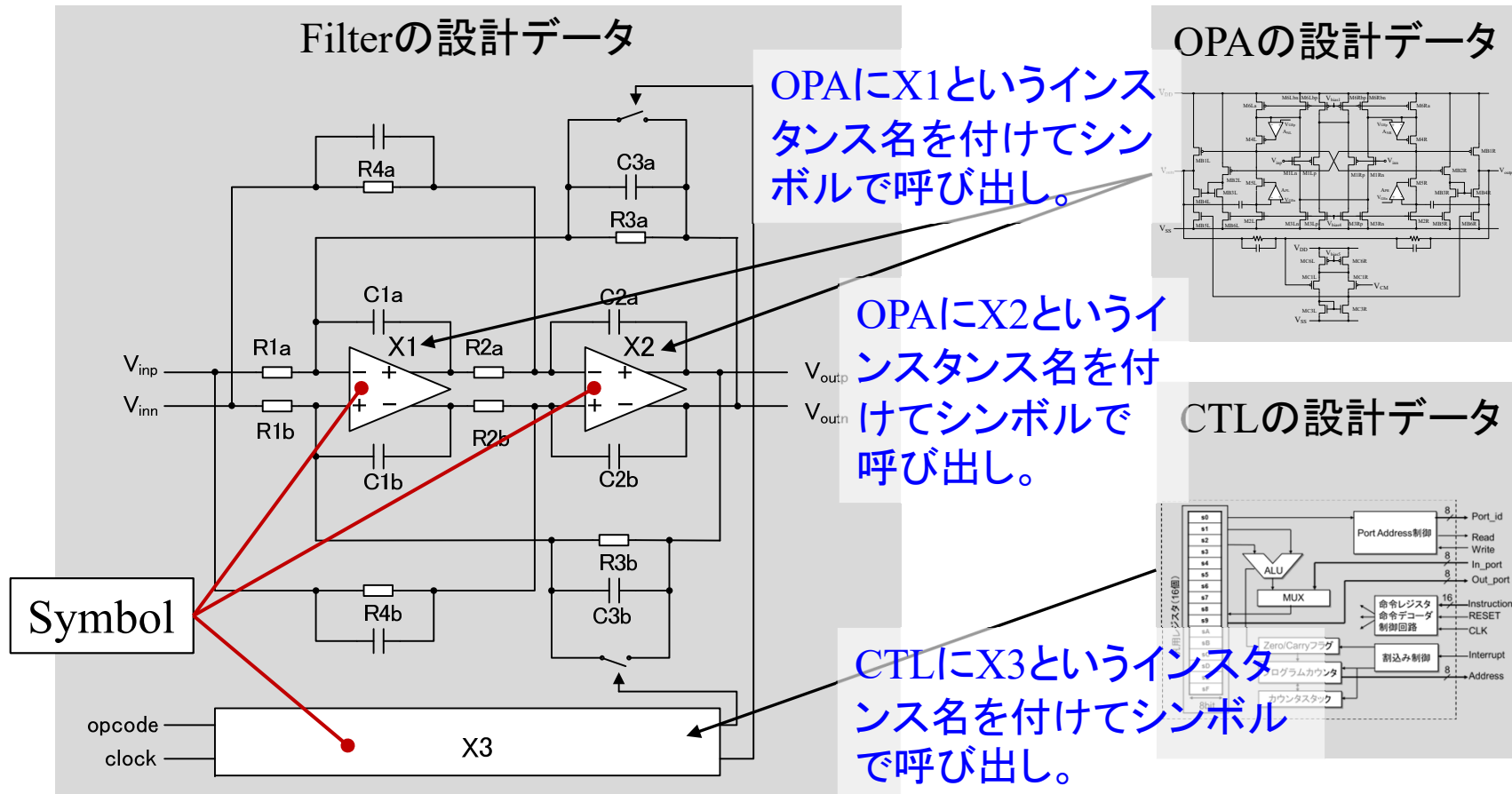
2.2 回路の階層化

回路の階層構造(Hierarchy)



サブサーキットの呼び出し方法

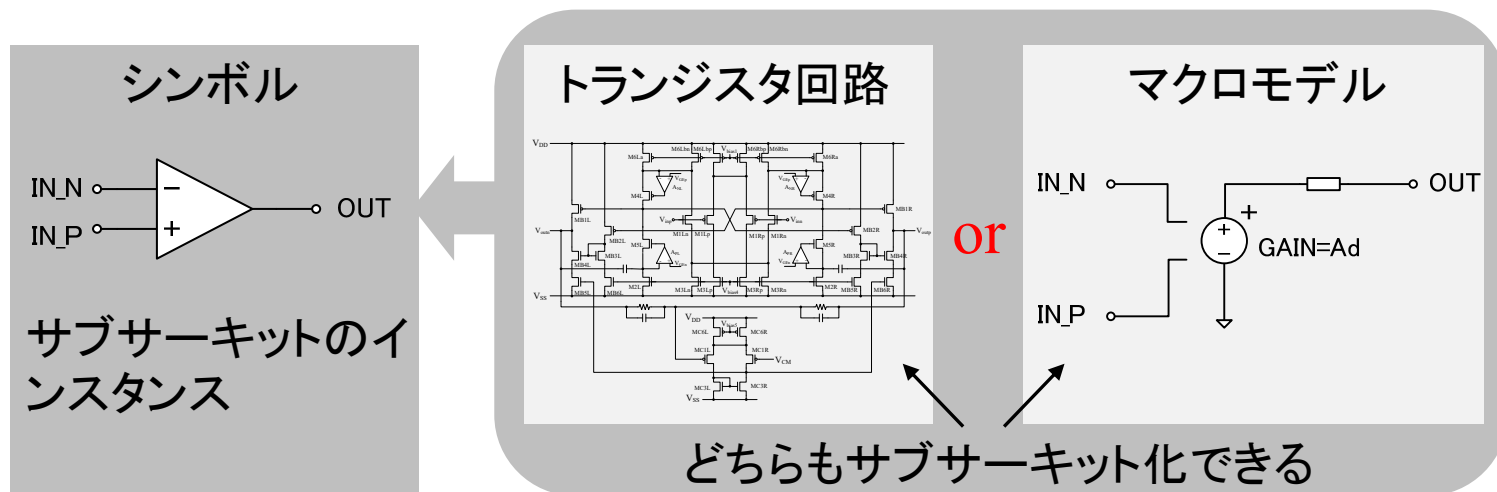
シンボルとして呼び出し ← インスタンス名 (X1, X2, X3) ← 定義 (OPA, CTL)



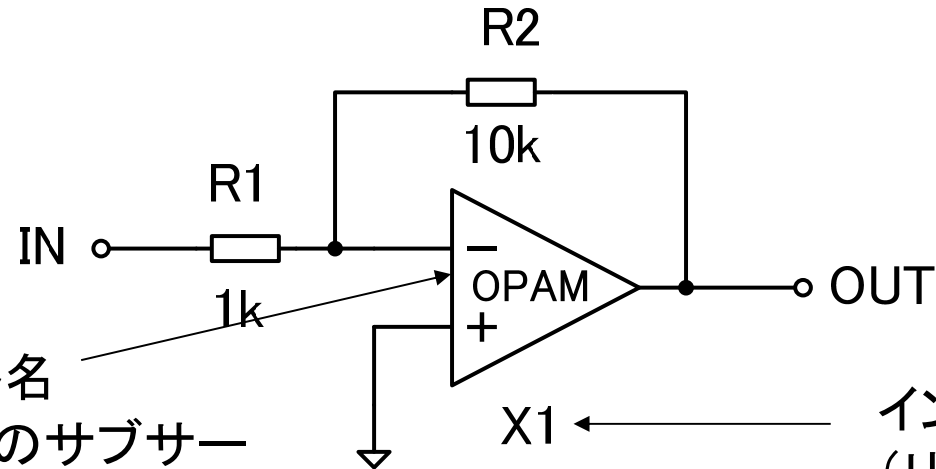
サブサーキットを呼び出すためにはインスタンス名 (リファレンス) とシンボルが必要。

マクロモデル(Macro-model)

- 等価回路や特性計算式で表現したサブサーキットをマクロモデルまたはビヘイビアモデル(Behavior model)と呼ぶ
 - 前ページのFilterのシミュレーションを行う際に、OPAのトランジスタ回路を呼び出してもよいが、OPAに含まれるトランジスタ数が多い場合には、シミュレーションに長い時間がかかる
 - トランジスタ回路の代わりにマクロモデルを使用することで、計算量を大幅に削減することができる



マクロモデルの呼び出し



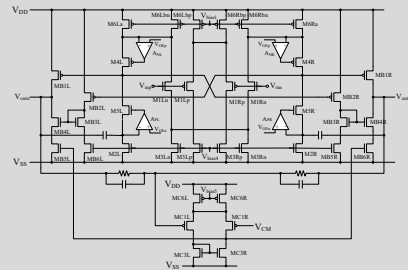
サブサーキット名
(マクロモデルのサブサーキット名を指定した場合)

インスタンス名
(リファレンスとも言う)

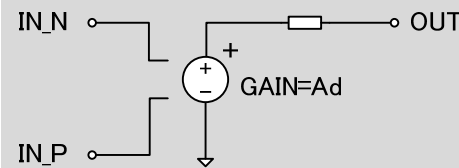
高精度 ← どちらか選択 → 計算量少

自分でOPARを設計した場合は、実回路のサブサーキットに差し替えが可能。

トランジスタ回路 (OPAR)



マクロモデル (OPAM)



通常、半導体メーカーは、マクロモデルのみ提供。

2.2節のまとめ

- 回路を階層設計することにより複雑な回路を簡単に扱える
 - 回路シミュレータは、要素回路をサブサーキットとして定義し、インスタンス化して上位階層に呼び出す仕組みを持っている
 - インスタンスを呼び出すために、インスタンス名(リファレンス)とシンボルを使用する
- サブサーキットの表現にはトランジスタ回路記述とマクロモデルがある
 - トランジスタ回路記述は、トランジスタの特性から計算するため高精度だが、計算時間が長い
 - マクロモデルは、簡単な数式で記述されているので、シミュレーション精度は高くないが、計算時間は短い
 - メーカーの半導体製品は、マクロモデルのサブサーキットが提供されていることが多い
 - トランジスタ回路記述とマクロモデルは、必要に応じて使い分ける

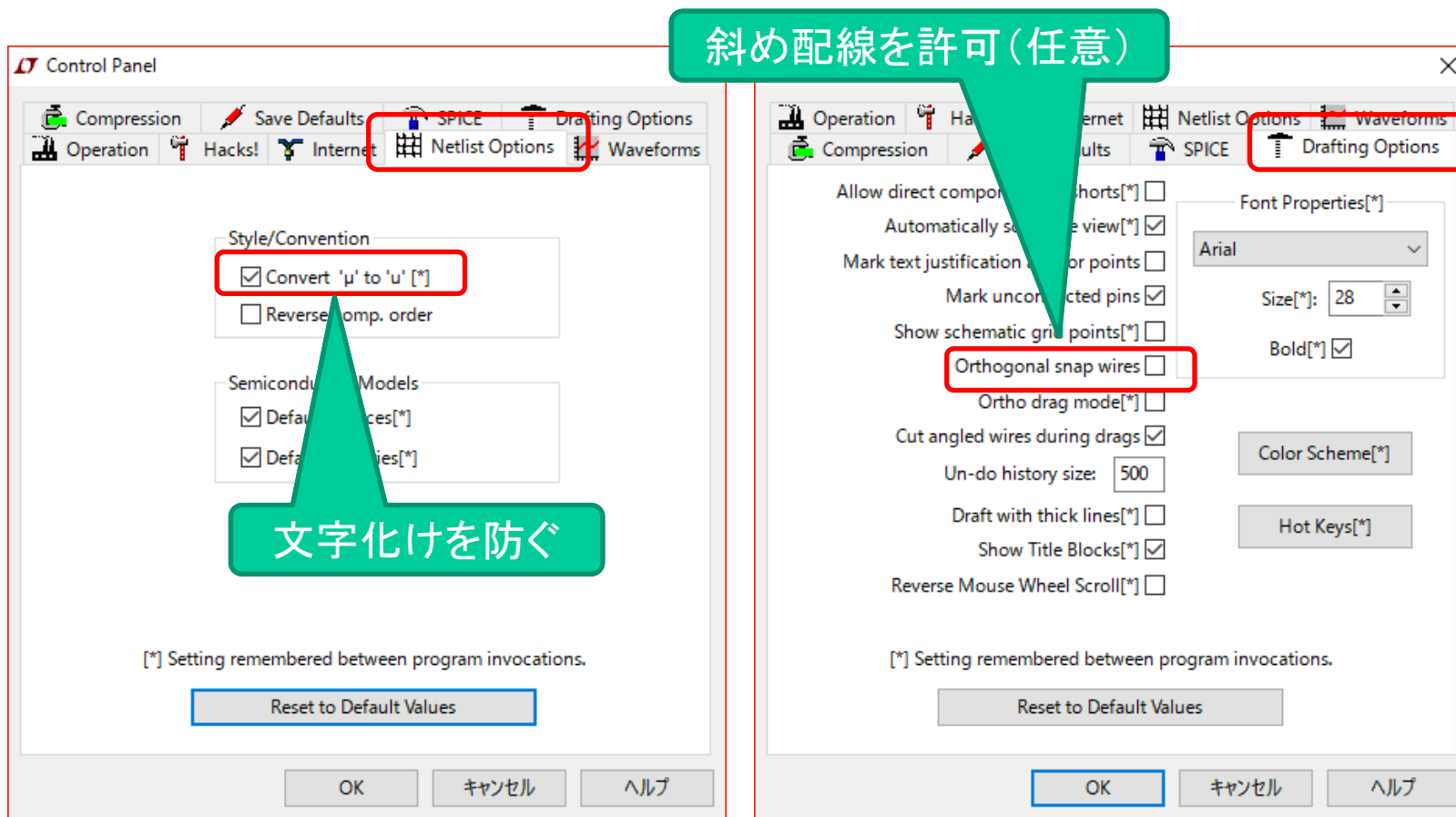
回路図入力～シミュレーション結果の表示まで

2.3 LTSPICEの基本操作

2. 3. 1 初期設定

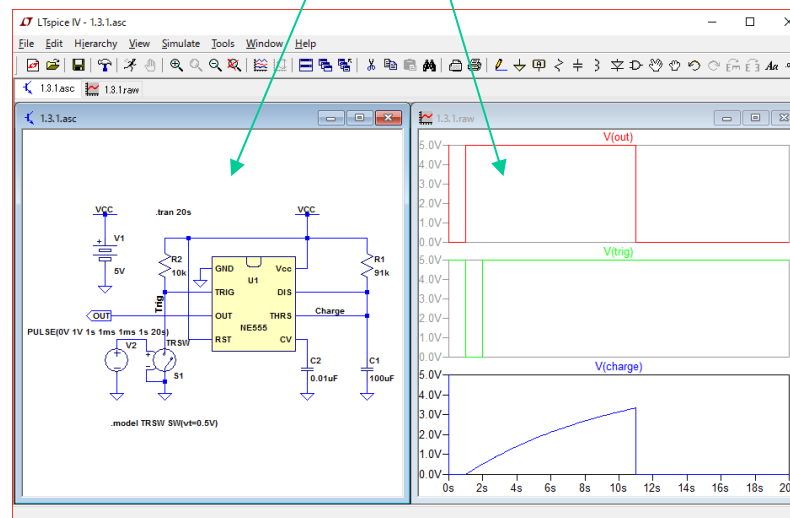
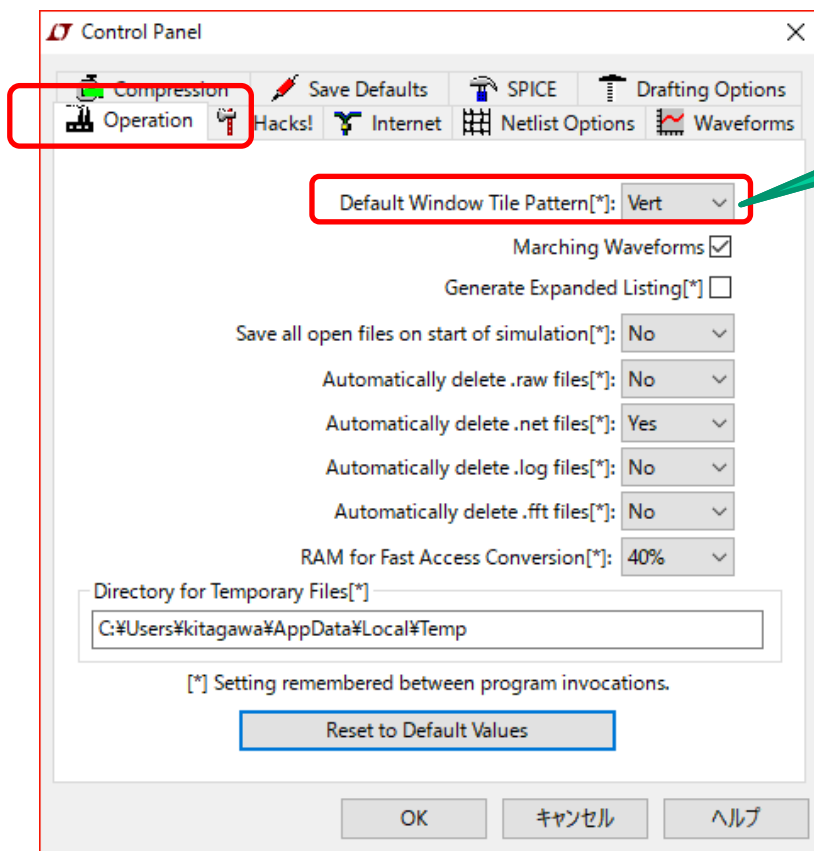
LTspiceを使いやすくするための準備

初期設定1



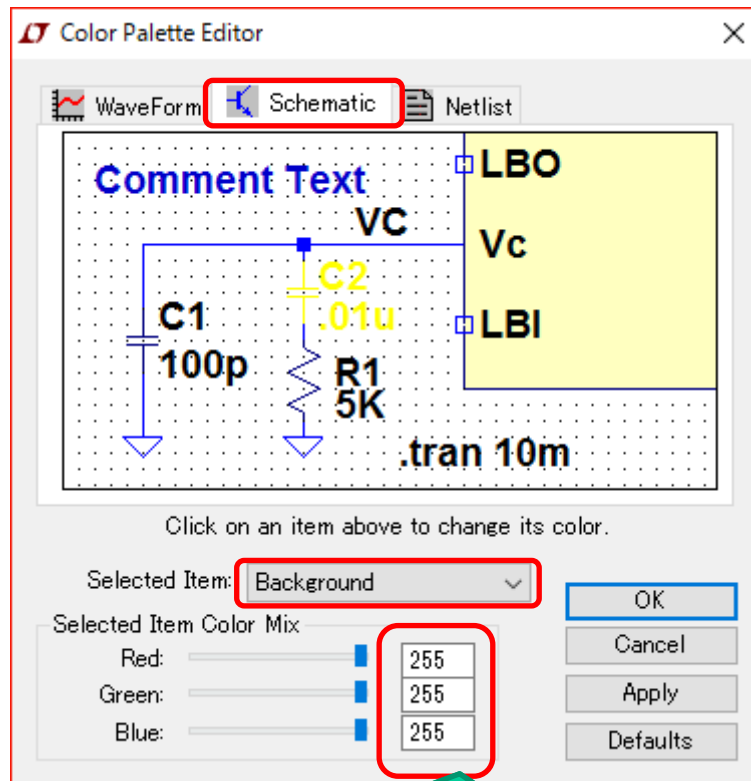
初期設定2

回路図とグラフを横に並べて表示(任意)

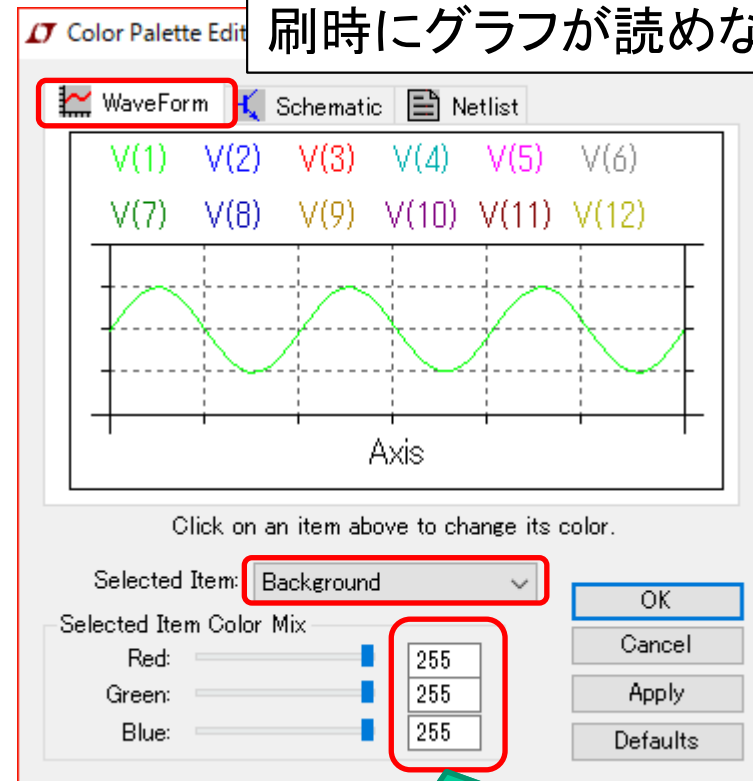


カラー設定1

グラフの背景が黒だと印刷時にグラフが読めない

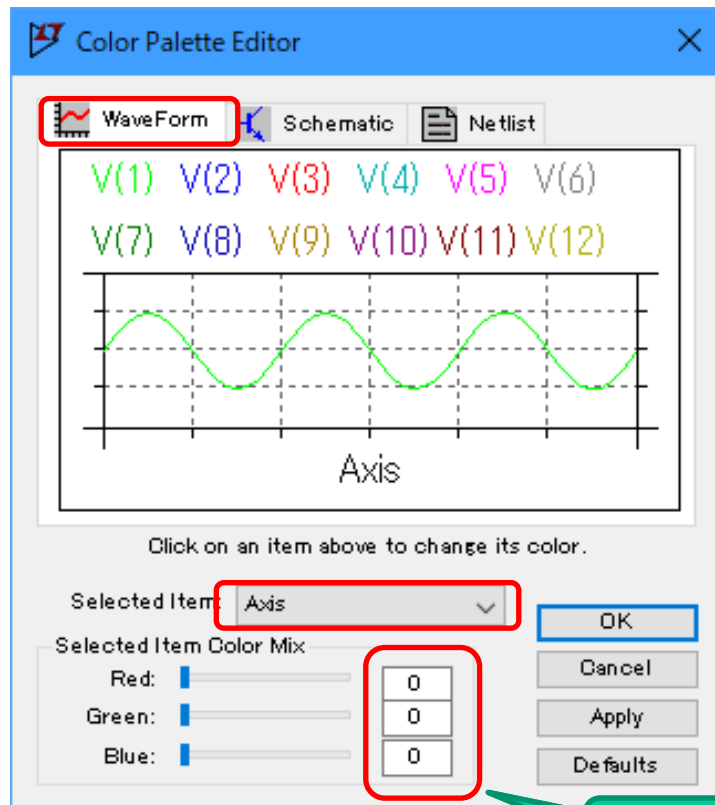


回路図の背景を白に
(任意)



グラフの背景を白に
(必須)

カラー設定2



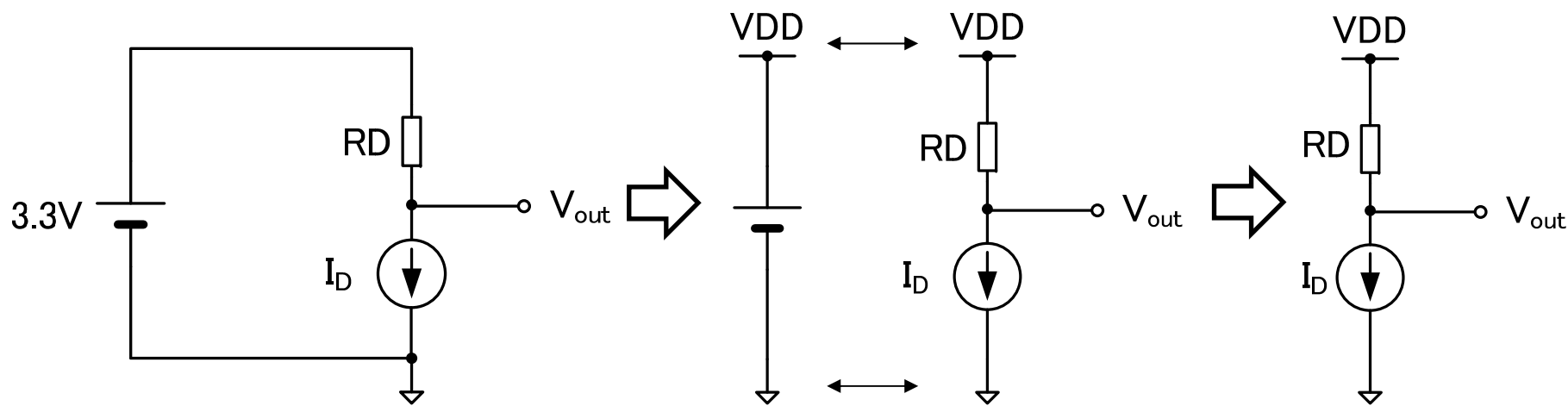
グラフの背景を白にした場合、
軸と目盛り数字を黒にすると、
グラフが読みやすい

グラフの軸を黒に(任意)

2. 3. 2 例題:Lチ力回路

Lチ力回路 = LEDを点滅させる回路

(重要)電源の省略表記



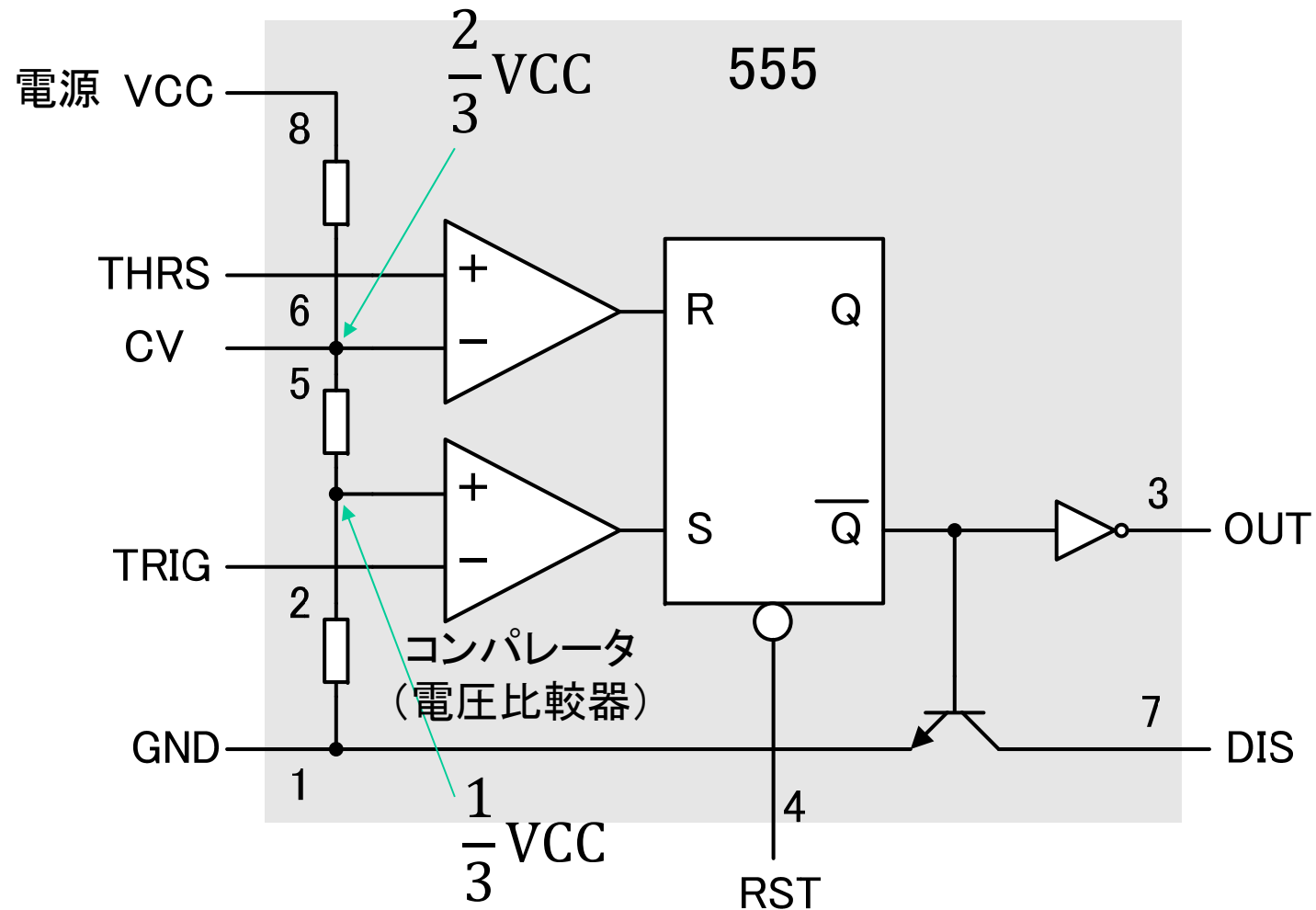
完全な表記

電源配線の省略
(シミュレーション用
の表記)

電源配線と電源の省略
(一般の回路図表記)

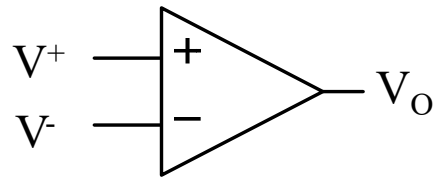
電源の電圧源と電源配線を省略して、電源配線名VDDまたはVCC(負電源の場合はVSSまたはVEE)を付けることにより、回路図を読みやすくする。

NE555 (タイマーIC) の内部回路



NE555の構成要素

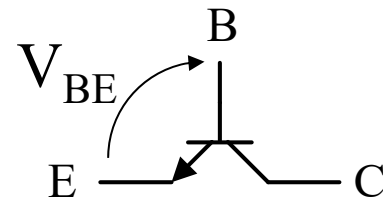
Comparator (電圧比較器)



$$V^+ - V^- \geq 0 \rightarrow V_O = \text{"H"}$$

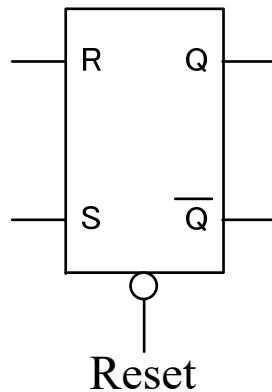
$$V^+ - V^- < 0 \rightarrow V_O = \text{"L"}$$

BJT (Bipolar Junction Transistor)



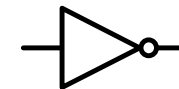
$$V_{BE} \geq 0.6 \rightarrow \text{C-E間がON}$$

RS-FF (Reset-set flip-flop)



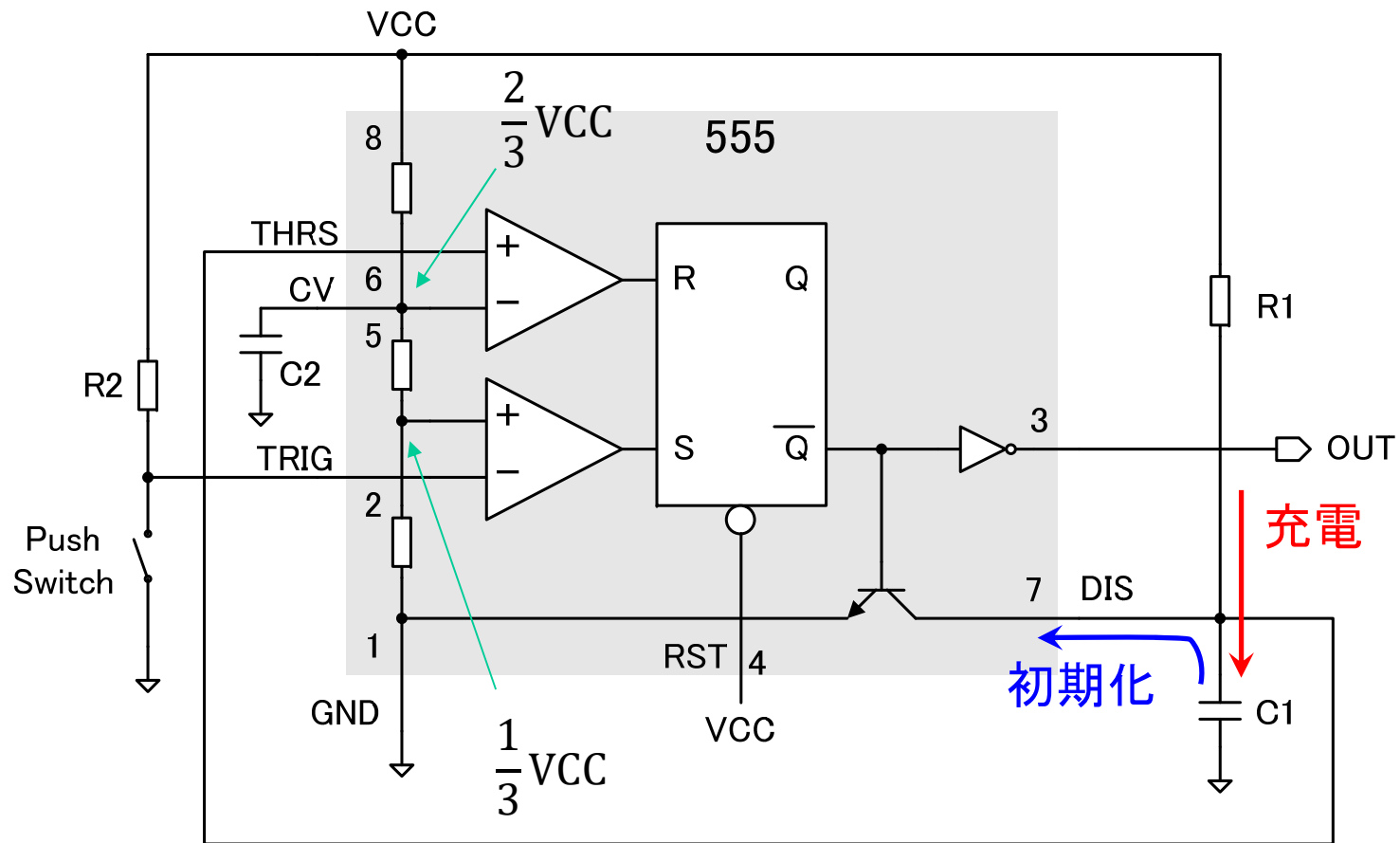
R	S	Q
0	0	Hold
0	1	1
1	0	0

Inverter



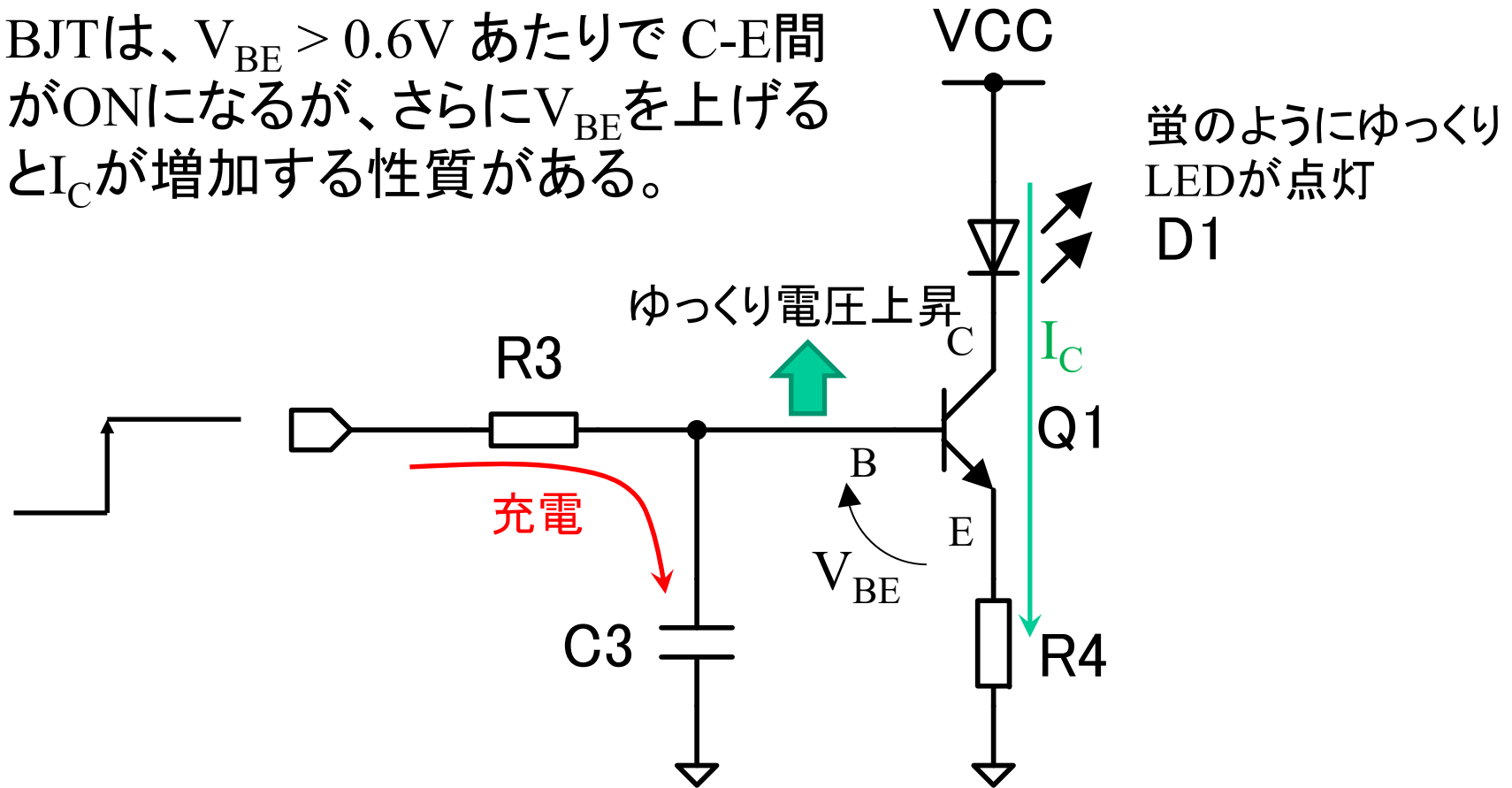
IN	OUT
"L"	"H"
"H"	"L"

NE555を用いたタイマーの例

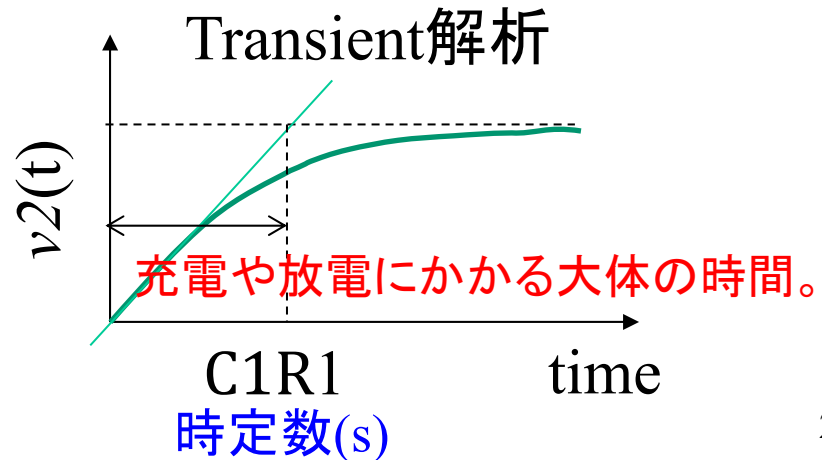
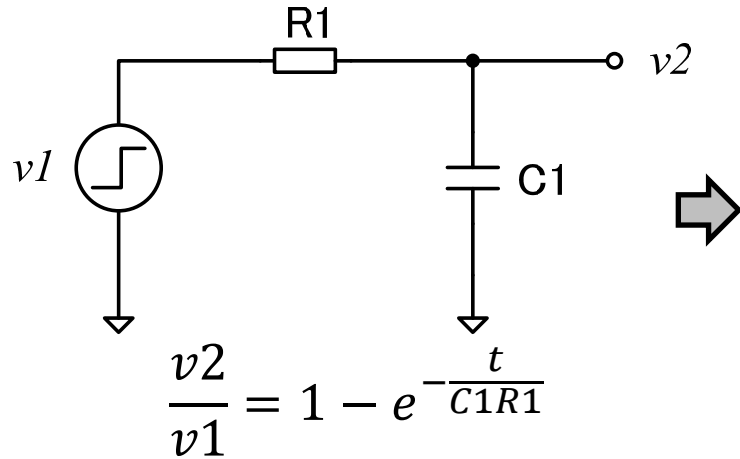
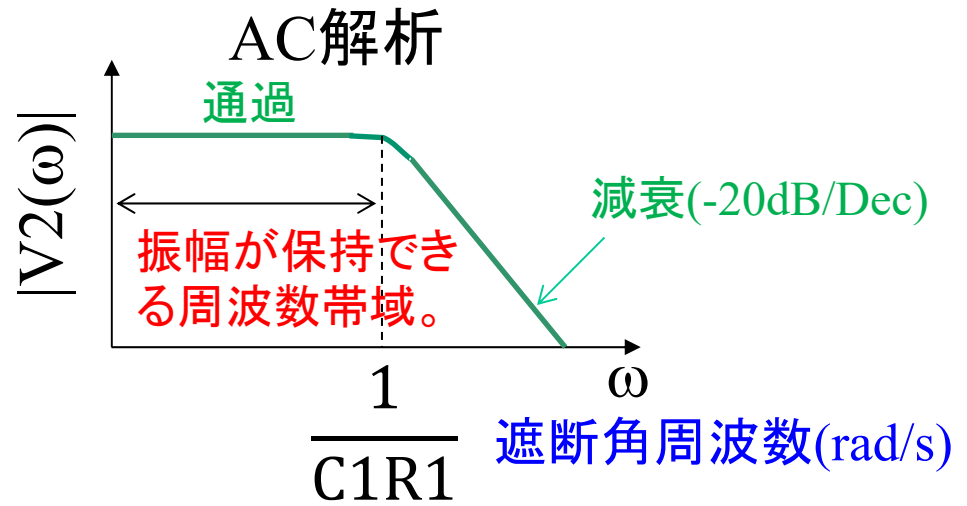
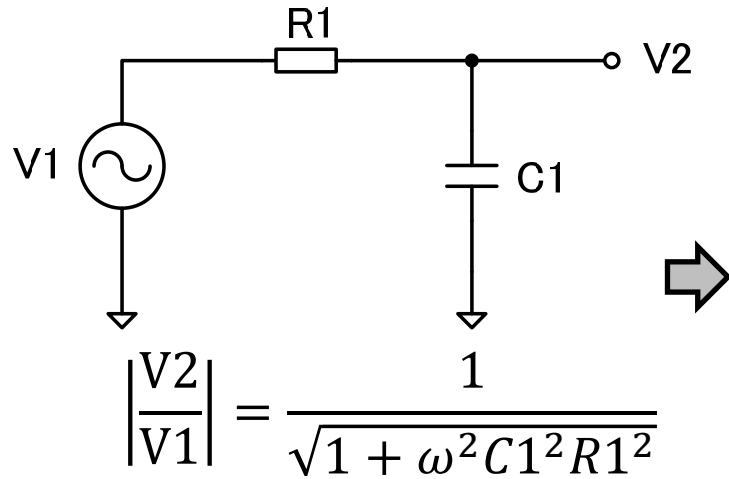


ソフトなLED点灯回路

BJTは、 $V_{BE} > 0.6V$ あたりに C-E間
がONになるが、さらに V_{BE} を上げると
 I_C が増加する性質がある。



(参考) RC回路の特性



回路図の作成

- 参考書2-1を参考にLチカの回路図を入力する
- **注意:** 保存は、ツールバーのディスクアイコンを使用**しない**で、メニューの[File] - [Save As] を使用すること
 - ツールバーから保存すると、ファイル名がDraft1.ascのようなデフォルト名になってしまい、後でファイルを探せなくなる
 - 回路図ファイルの拡張子は、.ascとなる。回路図ファイルをダブルクリックすると、LTspiceを起動することができる

回路定数の書き方

覚えておこう

1. 定数に単位を付けるとき、数値との間にスペースを空けないこと(V, A, F, Hなどの単位は省略可)
 - 例: 100fF(フェムトファラッド), 1mH(ミリヘンリー)
2. マイクロはuで表す(※)
3. メガはMEG, ミリはmとする
 - SPICE系シミュレータでは大文字と小文字は同じものとして扱われるためMとmの見分けがつかないため

※ 回路シミュレータ特有の約束だが、回路分野の文書でもマイクロを表すために u が使用されることが多い。

SI接頭辞

SI: International System of Units

- 電子回路で出てくる数値には、非常に小さい値や大きい値が多い
- SI接頭辞を用いて表記を簡略化する(記憶すること)

記号	読み方	指数部
PET	ペタ	10^{15}
T	テラ	10^{12}
G	ギガ	10^9
MEG	メガ	10^6
k	キロ	10^3
m	ミリ	10^{-3}
u	マイクロ	10^{-6}
n	ナノ	10^{-9}
p	ピコ	10^{-12}
f	フェムト	10^{-15}

トレーニング

1. $1/1\text{MEG} = ?$
2. $1/1\text{m} = ?$
3. $1/1\text{u} = ?$
4. $1/1\text{n} = ?$
5. $1\text{MEG}/1\text{k} = ?$
6. $1\text{k}/1\text{MEG} = ?$
7. $1/(1\text{m} \cdot 1\text{m}) = ?$
8. $1/(1\text{n} + 1\text{n}) = ?$

Transient解析（横軸＝時間）の設定

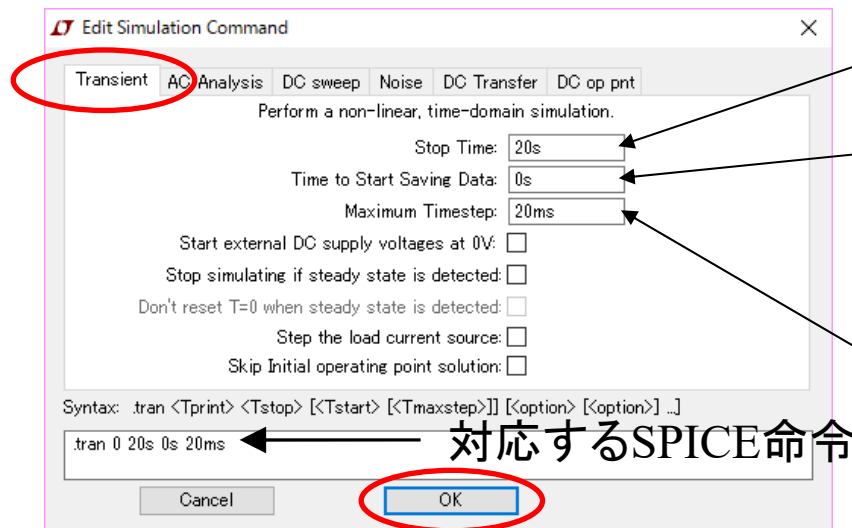
AC解析と同様に、回路図エディタの部品や配線がないスペースを右クリック

→ Edit Simulation Cmd. を選択

→ Transientタブを設定（教科書参照）

→ SPICEコマンドの文字列を回路図エディタ画面に配置

→ 再びスペースを右クリックし、Runでシミュレーション開始



シミュレーションを終了する時間

データの保存を開始する時間
(定常状態の波形を見たい場合などは、最初の方の結果は必要ないため。)

時間刻み幅の上限(シミュレータは時間刻みを自動調整している。)

課題2. 1(1)

1. NE555を用いたタイマーのスイッチを入れた後、出力OUTが 'H' になり続ける時間 T_1 をR1, C1の式で表せ
2. NE555を用いた発振器の発振周期 T_2 をR1, R2, C1の式で表せ

課題2. 1(2)

3. 参考書2. 1節のLチカ回路のシミュレーションを行い、回路図、結果のグラフをPDFにして提出せよ(ワープロ等に貼り付けて、PDF出力すればよい)
 - 結果のグラフには、 $I(D1)$, LEDに加わる電圧, $V(BASE)$, $V(TRIG)$, $V(OUT)$ を示すこと
4. シミュレーション結果から、点灯時間 T_{ON} , 消灯時間 T_{OFF} , 発光周期 T_p を測定し、それぞれの計算値に対する相対誤差(%)を求めよ
 - ただし、 T_{ON} は、 $V(OUT)$ が'HIGH'となる時間、 T_{OFF} は、 $V(OUT)$ が'LOW'となる時間と定義する。

エクスポート

回路図の出力

回路図エディタで作成した回路図をOfficeソフト等を使用するとき

1. メニュー: Tools → Write to a .wmf file
2. 適当なファイル名を付けて保存
3. 保存できる形式はWMF(Windows Meta File)のみ

シミュレーションデータの出力

シミュレーション結果(数値)をExcelや他のソフトウェアで使いたいとき

1. 保存したいグラフのウィンドウをクリック
2. メニュー: File → Export
3. 保存したい電圧、電流を CTRLキーを押しながら選択し、OKボタンをクリック
4. ファイル名は、自動的に"回路名.txt"となる(PRN形式)

画面のコピー

回路図やグラフをコピーしたいとき

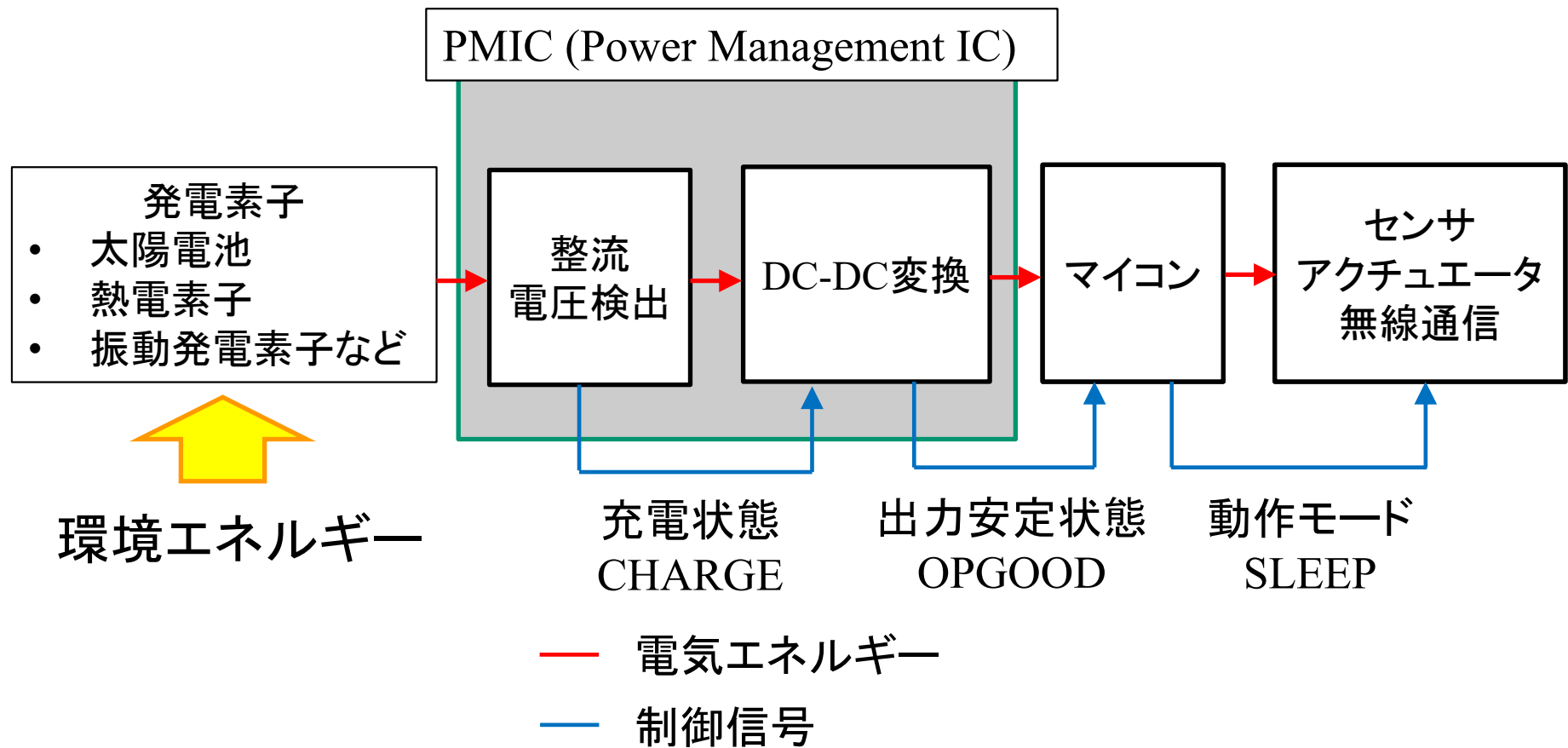
1. 回路図をコピーしたい場合は、回路図のウィンドウをクリック
2. グラフをコピーしたい場合は、グラフのウィンドウをクリック
3. メニュー: Tools → Copy bitmap to Clipboard
4. 貼り付け先でペースト(CTRL + v)を実行

(注意) 表示されているままの形でコピーされるので、予め貼り付け先の大きさに合わせてから実行すること。ビットマップデータなので、後で拡大縮小するときたなくなる。

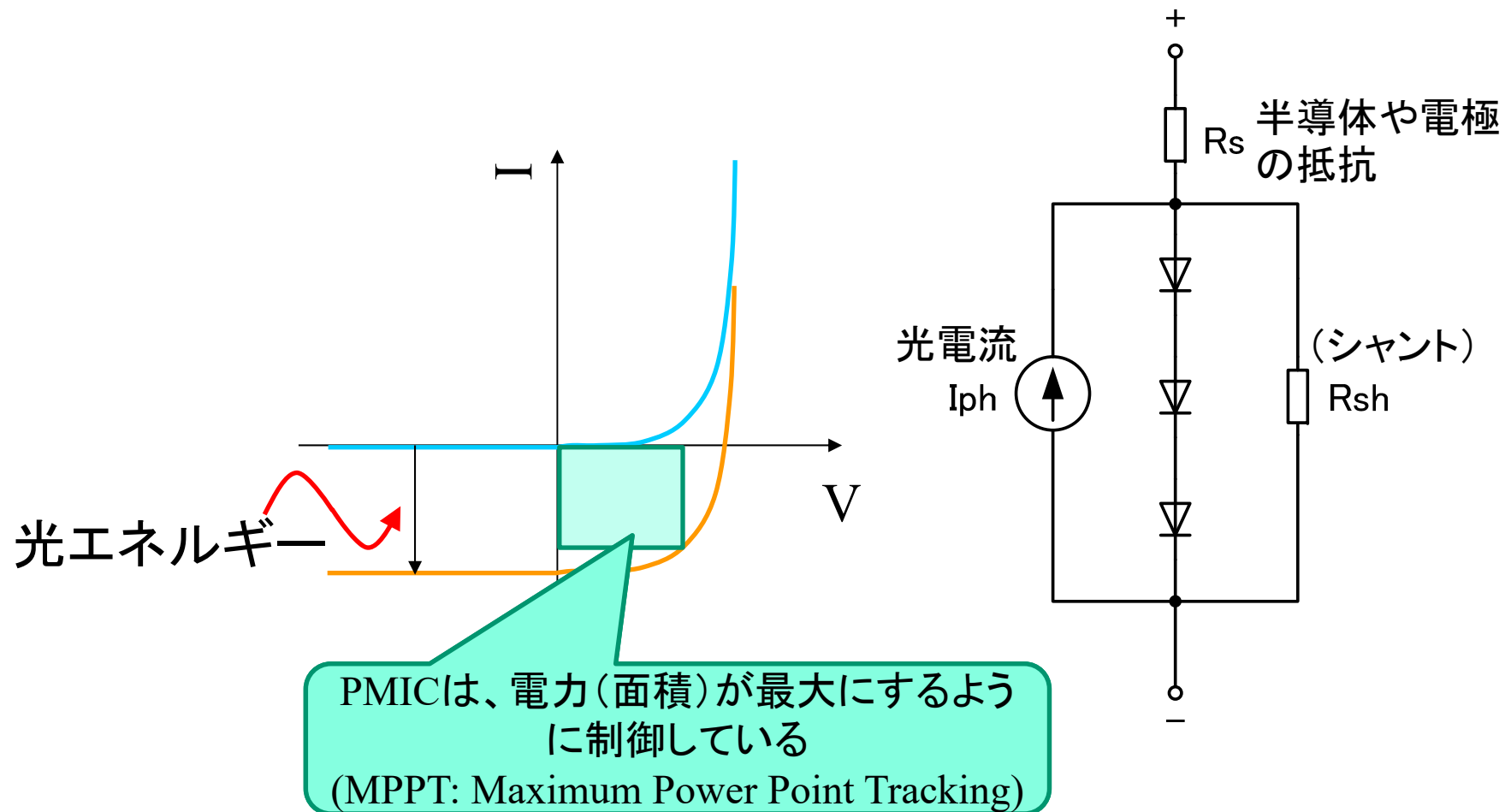
2. 3. 3 例題: エナジーハーベ スティング

エナジーハーベスティング (Energy Harvesting)
= 環境エネルギーの電力への変換

エネルギーハーベスティング回路



太陽電池(Solar cell)の等価回路

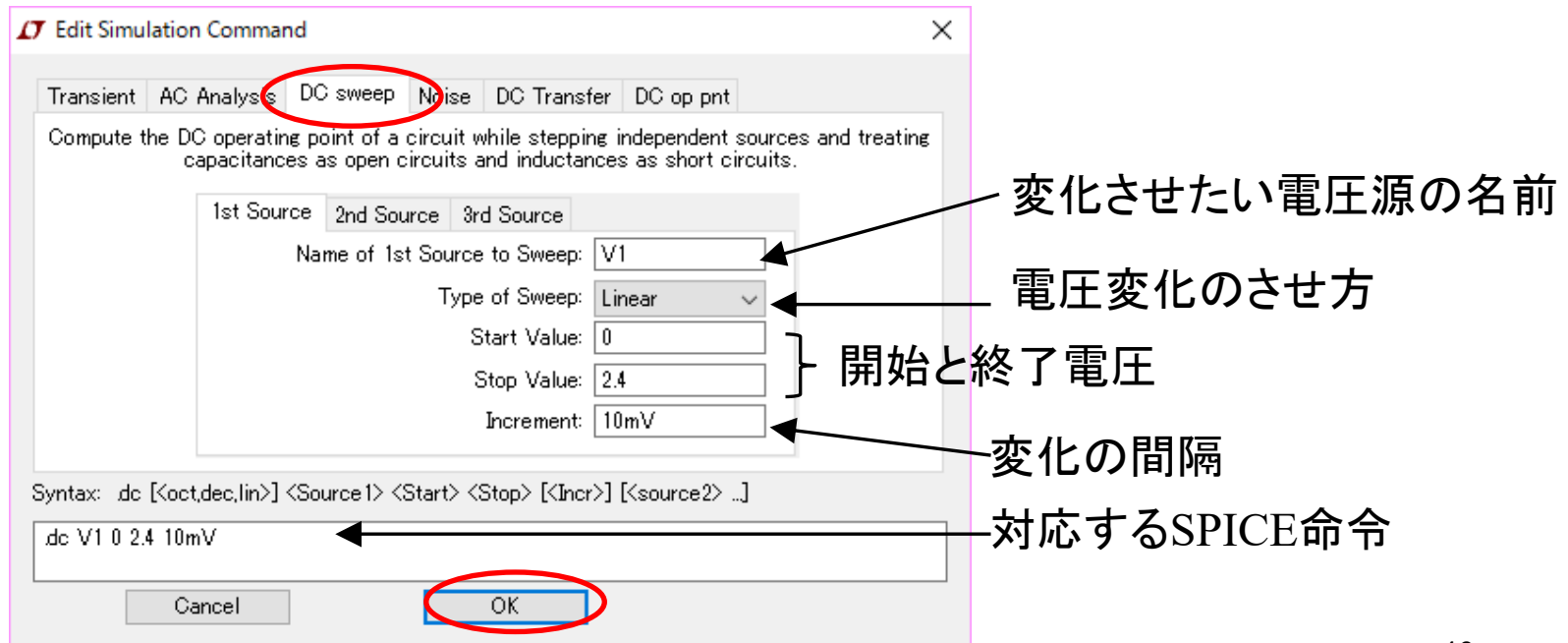


太陽電池のモデル

- 発電素子のような特殊な部品は、回路シミュレーション用のモデルが無い場合が多いため、等価回路を作成して、データシートの特性と合わせ込む必要がある
- 参考書3-1を参考に、太陽電池の等価回路を入力する

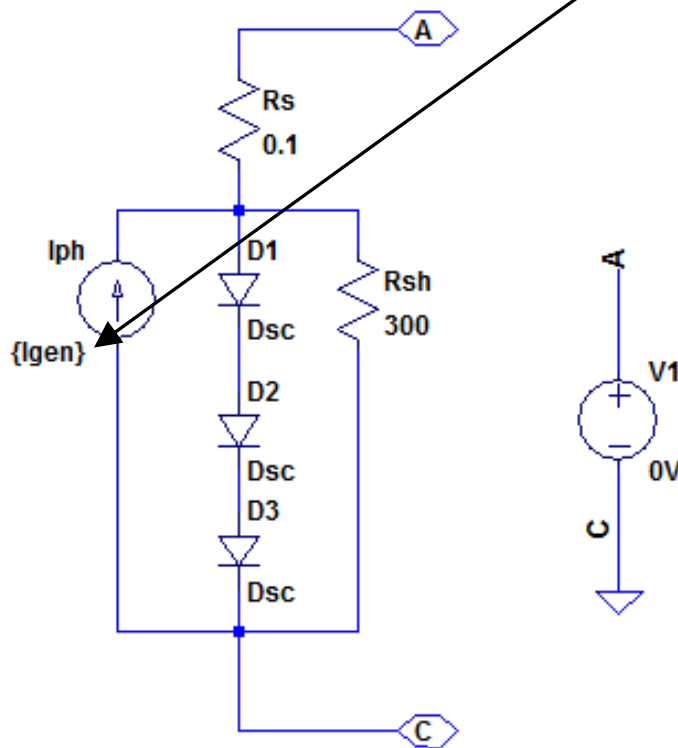
DC解析(横軸=電圧)の設定

- 回路図エディタの部品や配線がないスペースを右クリック
- Edit Simulation Cmd. を選択
- SPICE命令の文字列(.dc~)を回路図エディタ画面に配置
- 再びスペースを右クリックし、Runでシミュレーション開始



パラメータスイープ(Parameter sweep)

```
.dc V1 0 2.4 10mV  
.step param Igen list 0mA 100mA 200mA 300mA  
.meas dc Pmax max I(V1)*V1  
.meas dc Vmax find V1 when I(V1)*v1=Pmax  
.meas dc Imax find I(V1) when I(V1)*v1=Pmax
```



```
.model Dsc D(Is=80n Rs=0.1 N=1.9)
```

.step 命令で変数値を指定する。
変数は、{Igen}のように{}で囲む。

.meas (.measurement) 命令で自動測定を行う。

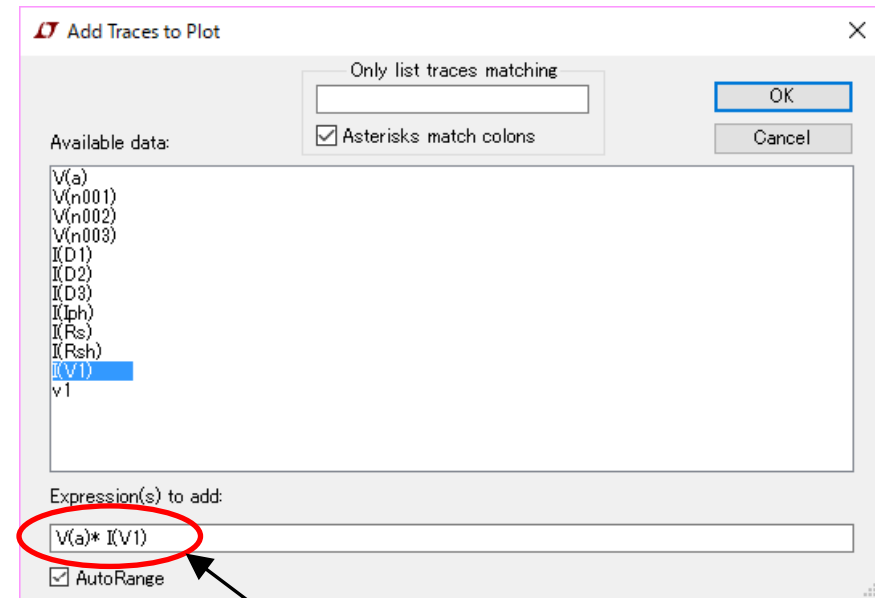
- Pmax 最大電力
- Vmax 最大電力となる電圧
- Imax 最大電力となる電流

素子値を変化させて特性変化を解析することを**パラメータスイープ**と呼ぶ。

命令(ディレクティブ)の入力は、.opアイコンをクリック。

波形演算

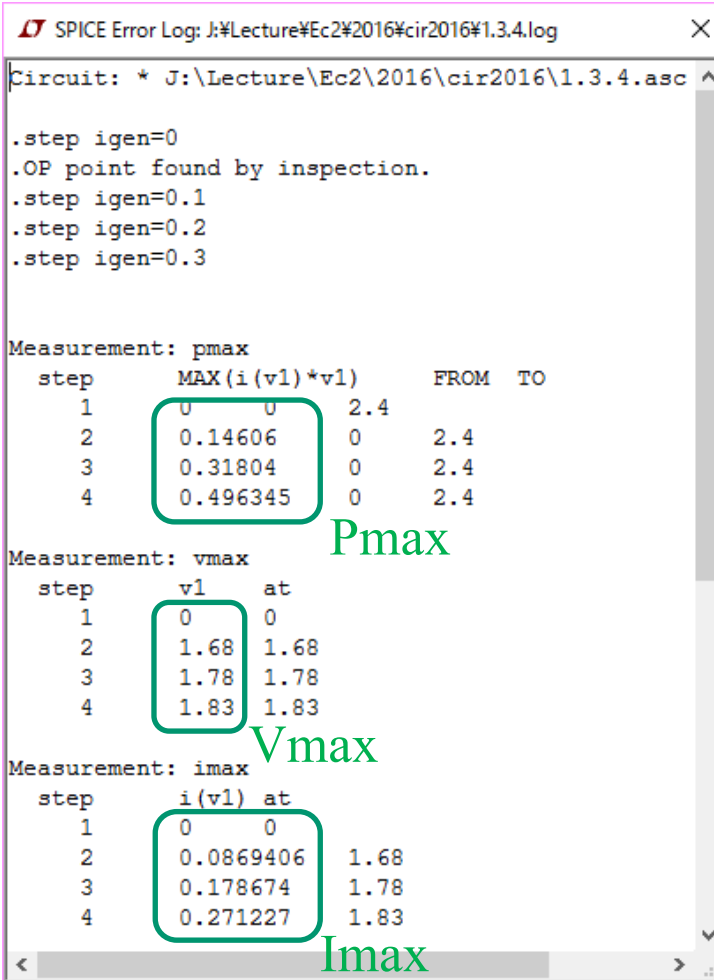
- グラフの上で右クリック
- Add Plot Paneを選択
- 追加された空のグラフ枠内を右クリック
- Add Traceを選択
- Expression(s) to add:欄に $V(A)*I(V1)$ を入力
 - ALTを押しながらV1をクリックでも、電力表示できる
- 縦軸の数字をクリックして、Top = 600mW, Bottom=-100mWに変更



求めたい値の計算式を入力

.measの結果の確認

- グラフまたは回路図の上で右クリック
- ポップアップメニューから [View] - [SPICE Error Log] を選択
- SPICE Error Logウィンドウが開き、結果が表示される



```
SPICE Error Log: J:\Lecture\Ec2\2016\cir2016\1.3.4.log
Circuit: * J:\Lecture\Ec2\2016\cir2016\1.3.4.asc
.step igen=0
.OP point found by inspection.
.step igen=0.1
.step igen=0.2
.step igen=0.3

Measurement: pmax
step    MAX(i(v1)*v1)    FROM TO
1       0 0            2.4
2       0.14606        0    2.4
3       0.31804        0    2.4
4       0.496345       0    2.4

Measurement: vmax
step    v1    at
1       0    0
2       1.68 1.68
3       1.78 1.78
4       1.83 1.83

Measurement: imax
step    i(v1) at
1       0    0
2       0.0869406 1.68
3       0.178674 1.78
4       0.271227 1.83
```

Pmax

Vmax

Imax

ネットリスト

- 回路の接続情報を表すデータを**ネットリスト(Netlist)**と呼ぶ
 - 回路図エディタは、回路図をネットリストに変換する
 - 回路シミュレータは、ネットリストに基づき回路方程式を作成する
- LTspiceでシミュレーションを実行すると、拡張子 `.asc` および `.net` というファイルが作成される
 - `*.asc`: 回路図エディタのデータ(ダブルクリックで回路図エディタ起動)
 - `*.net`: ネットリスト(SPICE形式)
- 保存した `*.net` ファイルをテキストエディタで開いてみよう
 - ネットリストは、終了時に削除される(デフォルト設定の場合)
 - ネットリストを保存したい場合は、メニュー: View → SPICE Netlist
 - ネットリストが表示されるので、ネットリスト上を右クリック → Edit as Independent Netlist で保存(この場合は、拡張子 `.sp` となる)

SPICE ネットリストのフォーマット

```
* J:\Lecture\Ec2\2016\cir2016\1.3.4.asc
Iph 0 N001 {Igen}
Rsh N001 0 300
Rs A N001 0.1
V1 A 0 0V
D1 N001 N002 Dsc
D2 N002 N003 Dsc
D3 N003 0 Dsc
.model Dsc D(Is=80n Rs=0.1 N=1.9)
.dc V1 0 2.4 10mV
.step param Igen list 0mA 100mA 200mA 300mA
.backanno
.end
```

← 行頭の*はコメント行。

← Rshがノード"N001"とノード"0"の間に接続されている。抵抗値は300Ω。ノード番号'0'は、GND(Ground)を表す。

} . で始まる行はシミュレータに渡す命令行 (SPICE Directive)。

(参考) ネットリストを使う場面

- SPICE ネットリストは標準的なフォーマットなので、他の電子回路シミュレータでも読み込める場合が多い
- シミュレーションエラーが発生したときに、ネットリストが読めると、原因の特定がしやすい
- マイクロマシン、論理回路、プログラム、新型デバイスなどを言語記述によりモデリングし、アナログ回路との混在シミュレーションをする場合にSPICE ネットリストが使用できる

課題2. 2

1. 参考書3-2, 3-3を参考にして、環境発電回路のシミュレーションを行い、回路図、結果のグラフをPDFにして提出せよ。グラフには、 $V(VDD)$, $V(IN)$, $V(LDO)$, $V(MPPC)$, $V(PGOOD)$, $V(AUX)$, $I(L1)$, $I(R5)$ を示すこと。

2.3節のまとめ

- LTspiceの基本的な使い方
 - デバイス(シンボル)の配置、配線
 - 素子値の設定
 - 半導体デバイスモデルとモデル名の指定
 - 電圧源、電流源の設定
 - 解析の設定(.DC, .TRAN命令の設定)
 - グラフ表示
 - 回路図とシミュレーション結果のエクスポート
- LTspiceの少し高度な使い方
 - .stepによるパラメトリック解析
 - .measによる自動測定
 - 波形演算機能によるシミュレーション結果に対する演算
 - ネットリストの出力