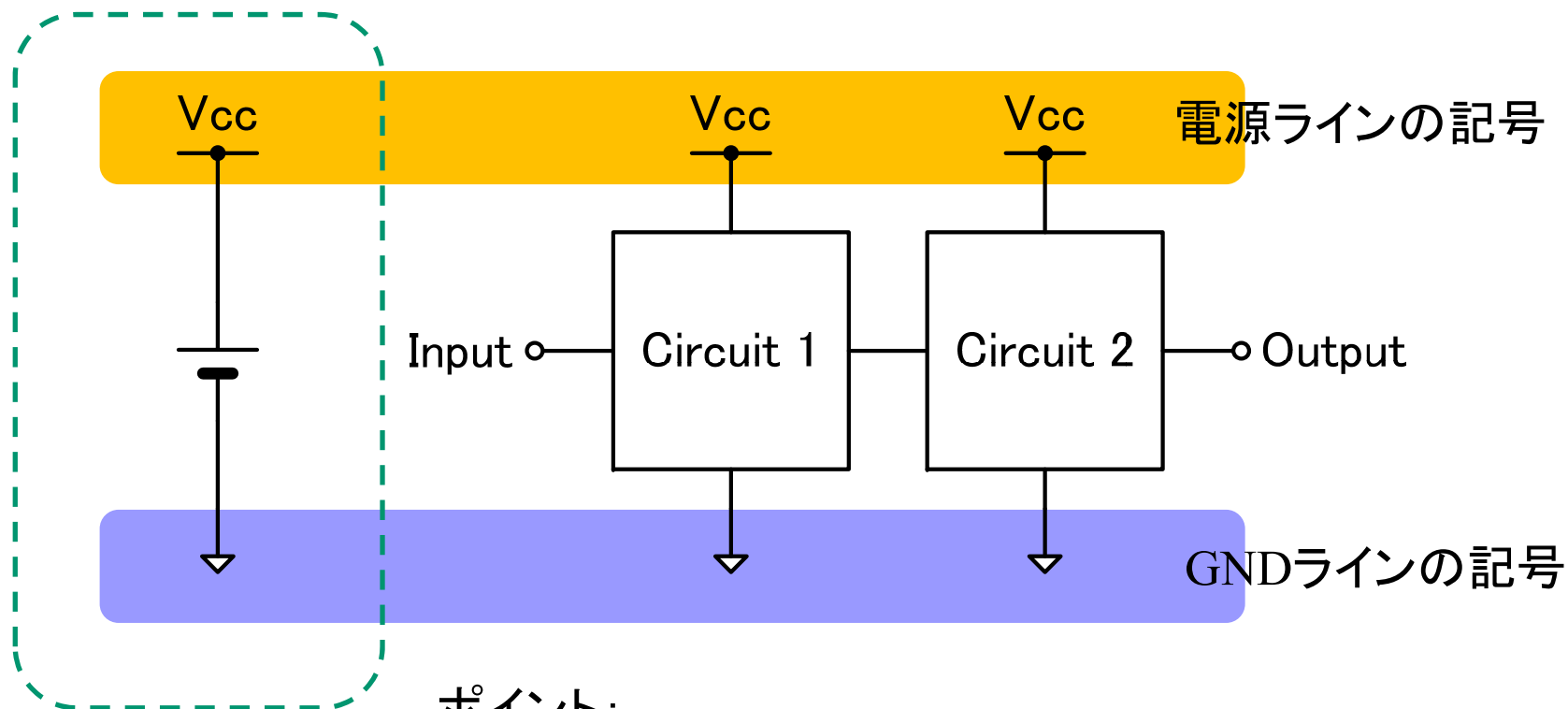


## 3.2 スイッチングの方法

# スマートな回路図表記



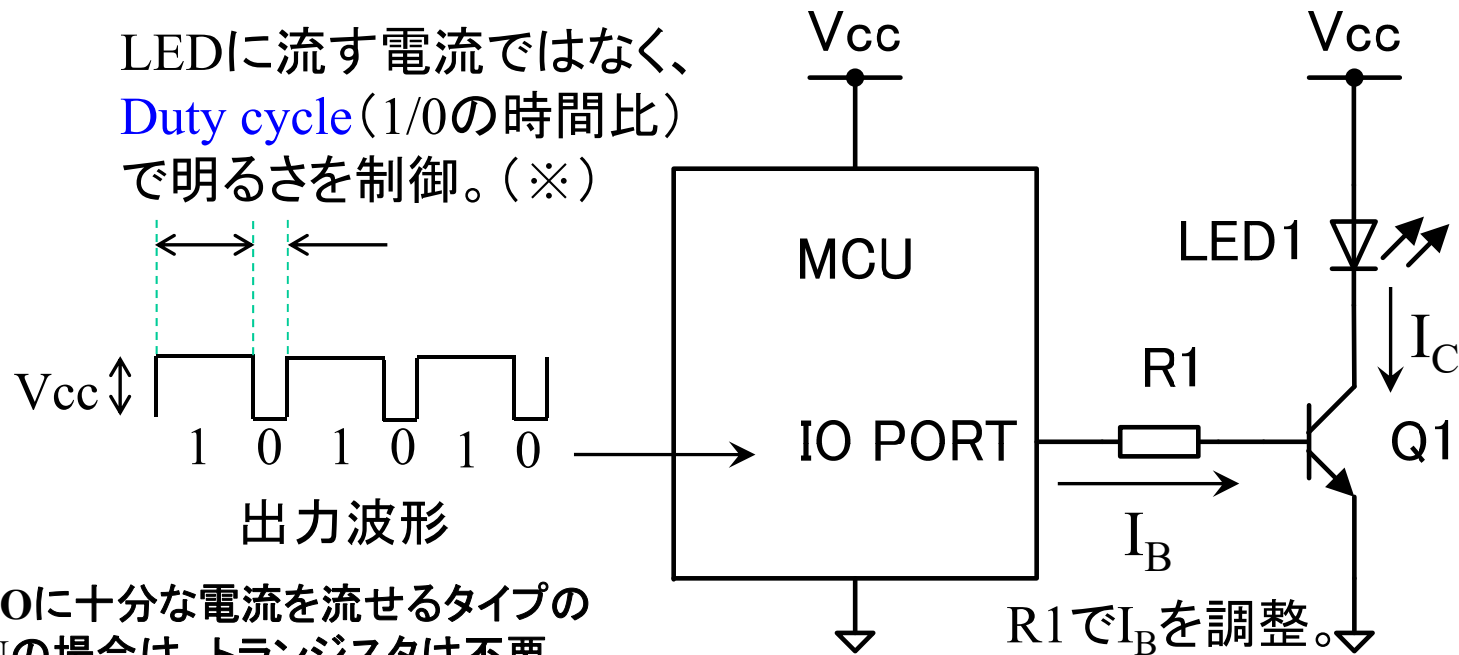
シミュレーションしない場合は、省略してよい。

ポイント:

- 実際には $V_{CC}$ とGND配線が必要だが、線を描かないですっきりした表記にする
- 複数の電源電圧を使用する回路もあるので、電源ラインには $V_{CC}$ などのラベルを付ける

# LEDのスイッチング回路

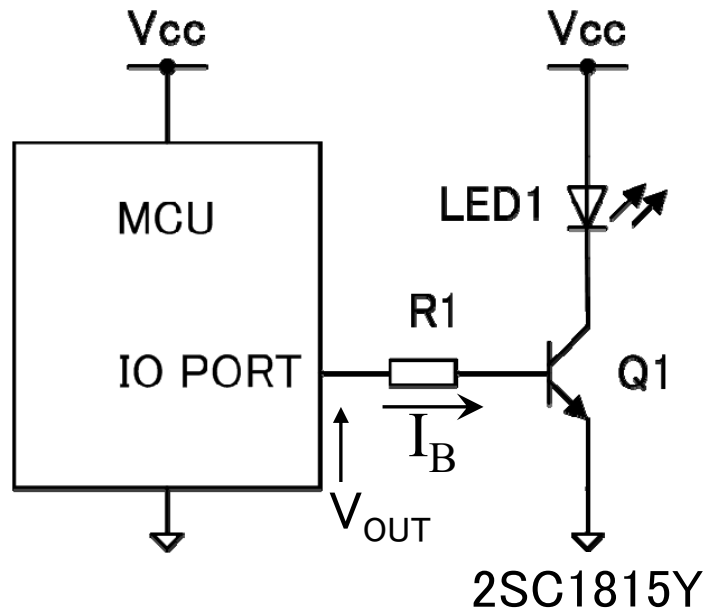
LEDの明るさをMCU(マイコン)で制御する回路。



MCU: MicrocontrollerまたはMicro Control Unit (CPU, メモリ, タイマ, IOなどを1チップに集積化したもの。)

※ 実際には明るさを変えていないが人間の目がだまされる。

# LEDスイッチング回路の設計



電源電圧  $V_{CC} = 3.3V$  (赤/黄/緑)  
LED1順方向電流  $I_F = 15mA$   
を流すように設計してみよう。

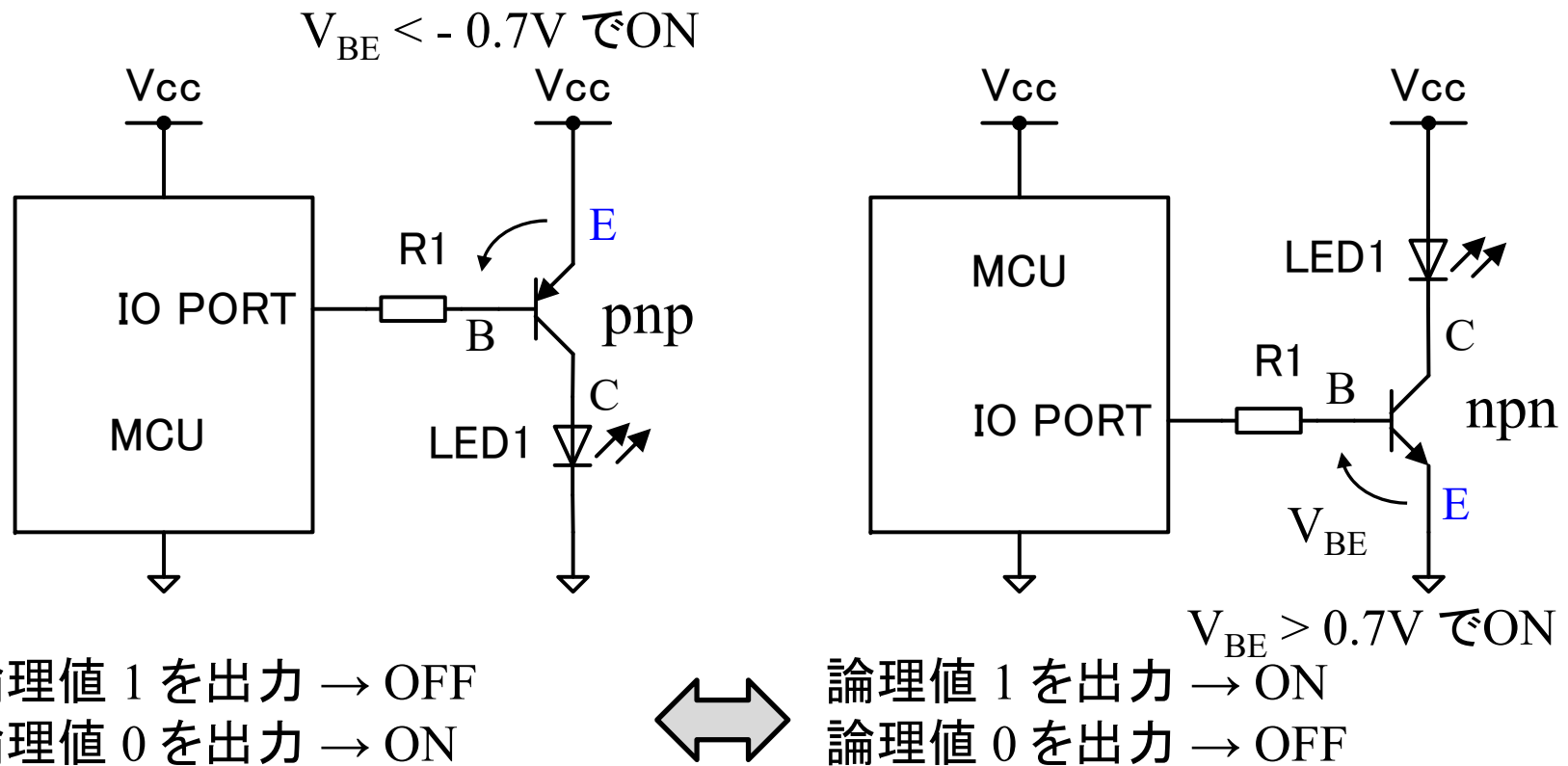
トランジスタの特性は、データシートまたは特性シミュレーションで調べておく。

$$V_{BE}(I_B) \doteq 0.73V (@ I_C = 15mA)$$
$$h_{FE} = 302 (@ I_C = 15mA)$$

IO PORTが論理値'1'の場合、 $V_{OUT} = 3.3V$ が出力される。このとき、 $I_F = 15mA$ となるように、 $R1$ を決定する。

$$\begin{cases} V_{OUT} = V_{BE} + R1 \cdot I_B \\ I_C = I_F = h_{FE} \cdot I_B \end{cases} \quad R1 = \frac{V_{OUT} - V_{BE}}{I_B} = \frac{V_{OUT} - V_{BE}}{I_F / h_{FE}} \cong 51k\Omega$$

# npnトランジスタとpnpトランジスタ

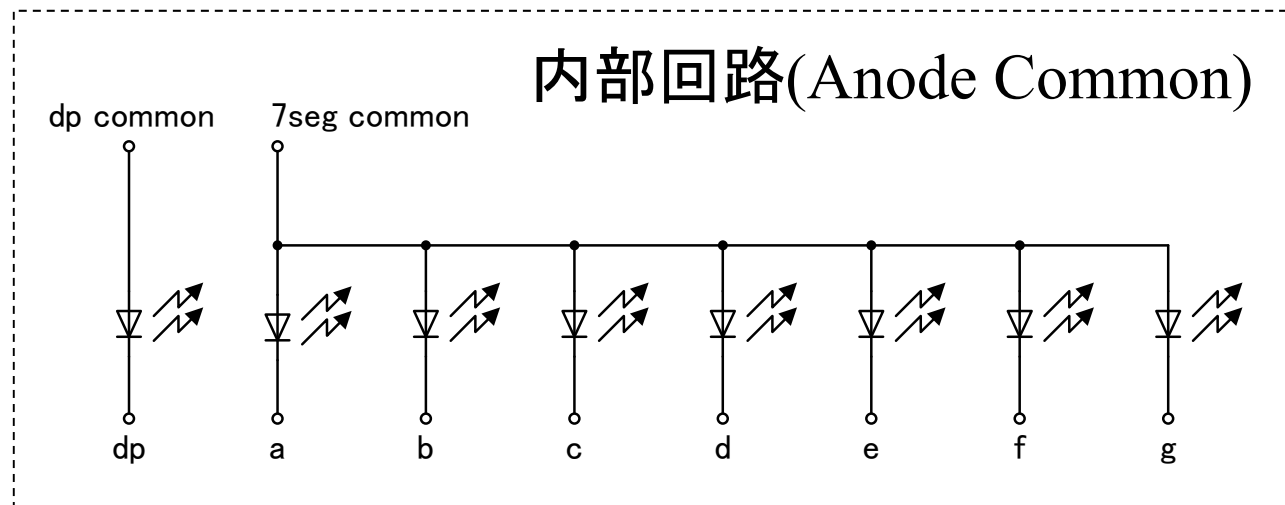
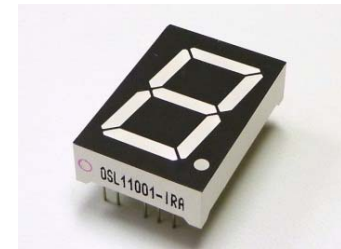
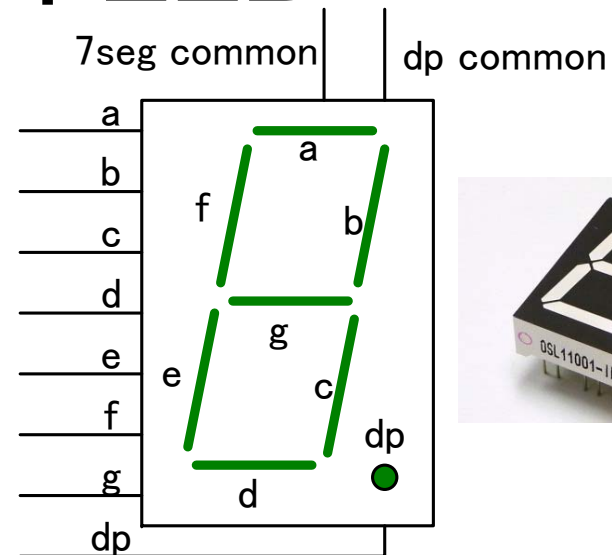


論理値とON/OFFの関係が逆になるので、プログラムの開発時に注意が必要。

# 7セグメントLED

一桁の数字とアルファベットを表示する。  
7seg commonがプラス電圧、a ~ g, dpが  
Ground(GND)のとき、対応するLEDの  
セグメントが発光する。発光には、1セグ  
メント当たり10mA程度の電流が必要。

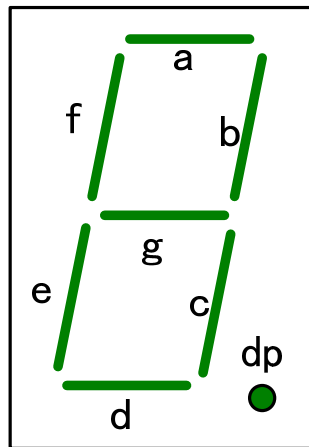
Anode commonタイプ(pnp用)と  
Cathode commonタイプ(npn用)がある。



# 7セグメントデコーダ

7セグメントデコーダの真理値表(※)

LED配置

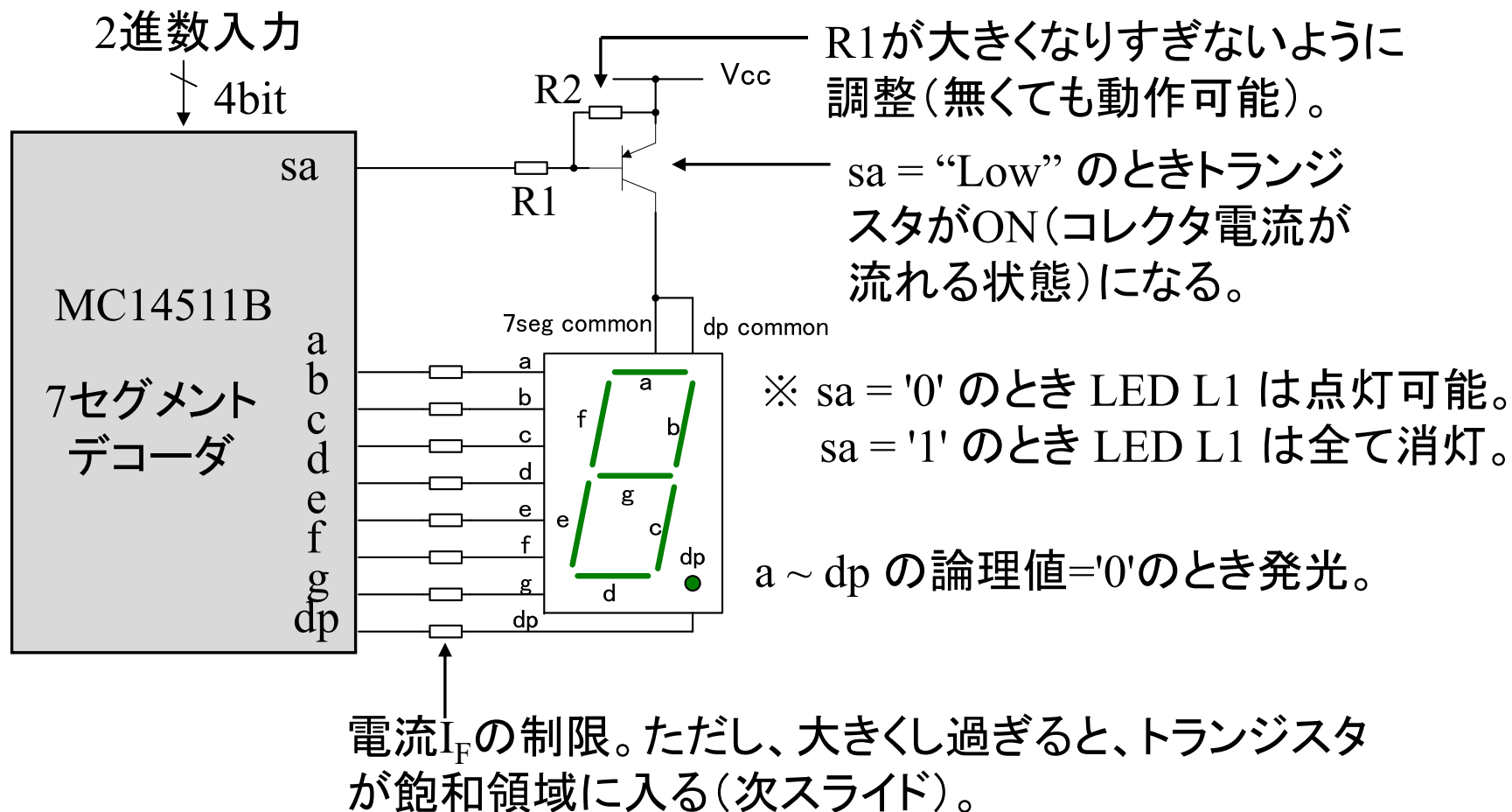


このままでは、使いにくいので、2進数をLEDの発光パターンに変換するロジック(7セグメントデコーダ)と接続して使用する。

表示文字	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0
C	1	1	1	0	0	1	0
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

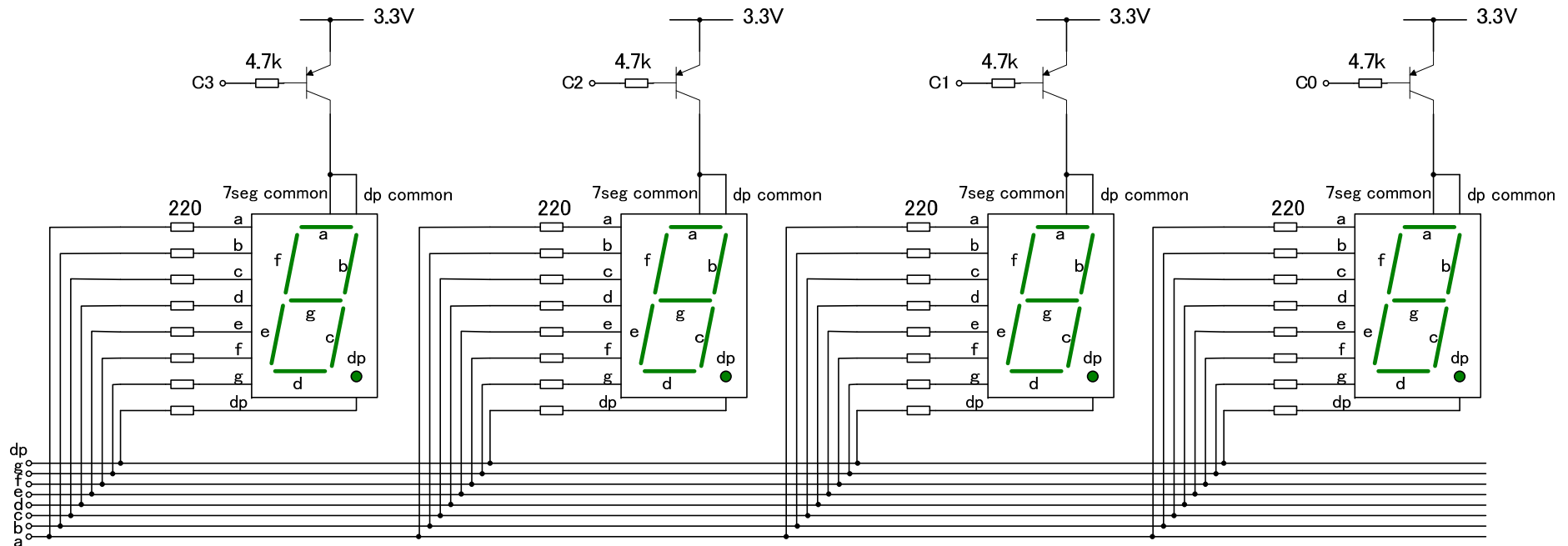
※ 論理値'0'に対応するLEDが発光。

# 7セグメントLEDの制御回路



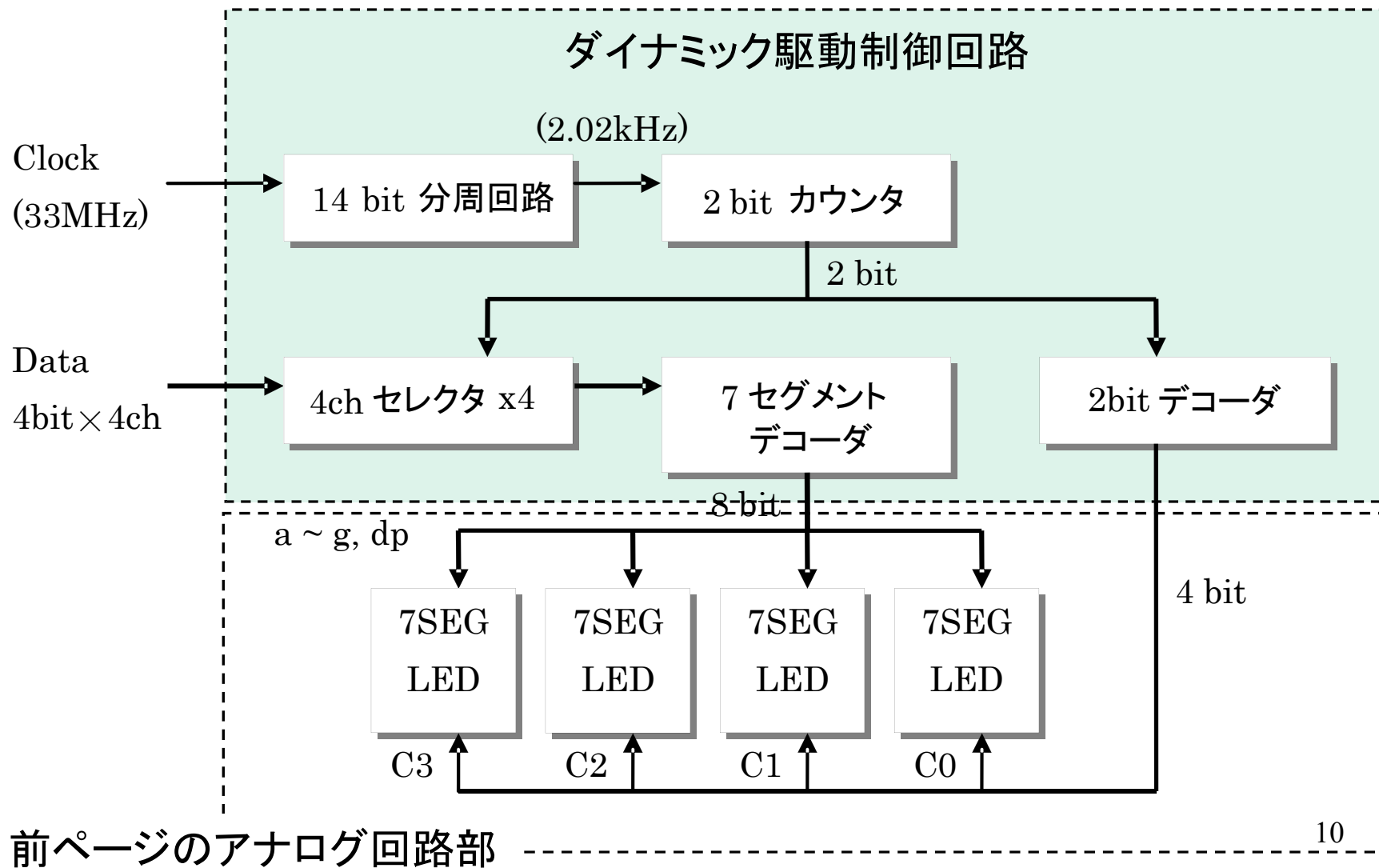


# 複数桁の表示方法

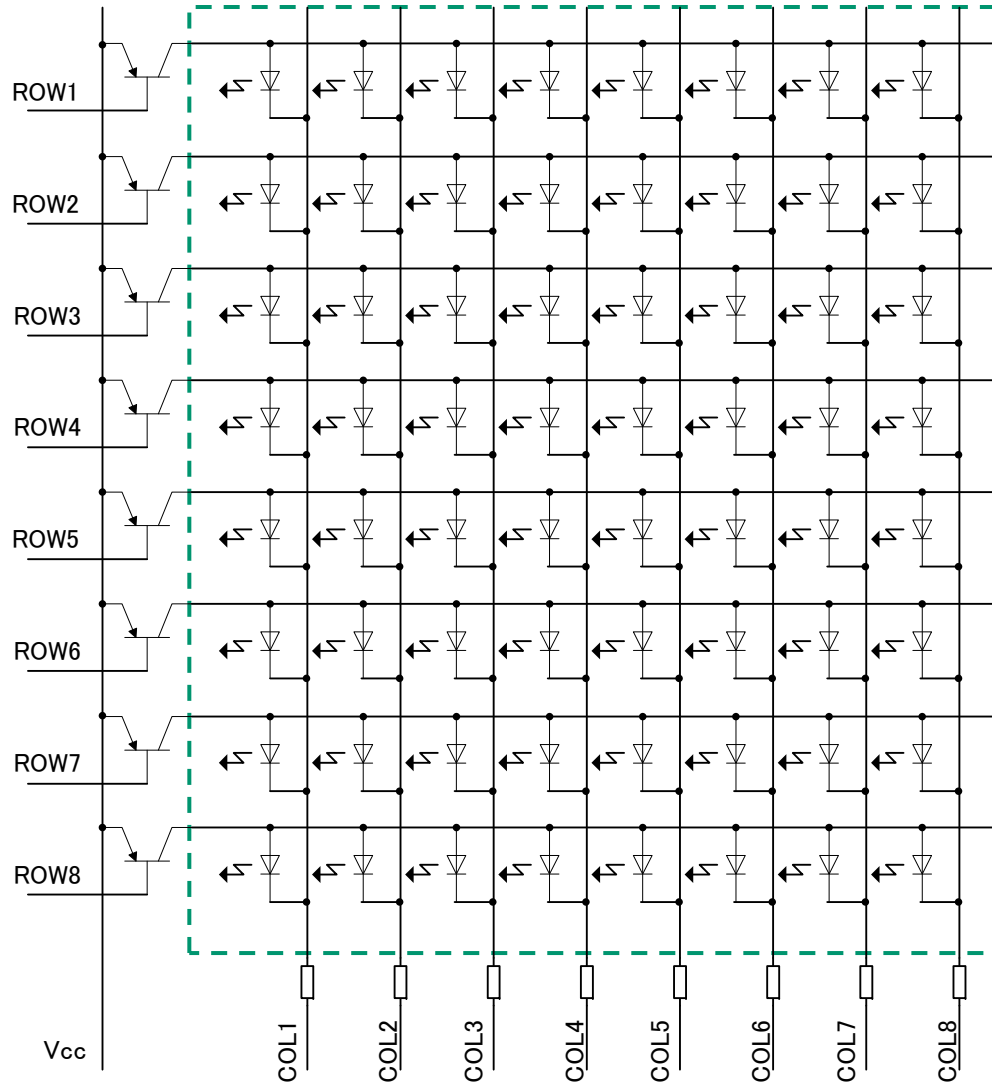


- 直流駆動では  $10 \times 4 = 40$ 本の制御線が必要(多すぎる) → **ダイナミック駆動**により配線を削減
  - C0 ~ C3 を順番に論理値 '0' に切り替え、 $C_n = '0'$  のとき、n番目の7セグメントLEDに表示するデータをa ~ g, dp端子に入力
  - a ~ g, dp, C0 ~ C3の合計12本の制御線で、4桁の7セグLEDを点灯させることができる

# 4桁ダイナミック駆動回路の例



# ドットマトリクスLED

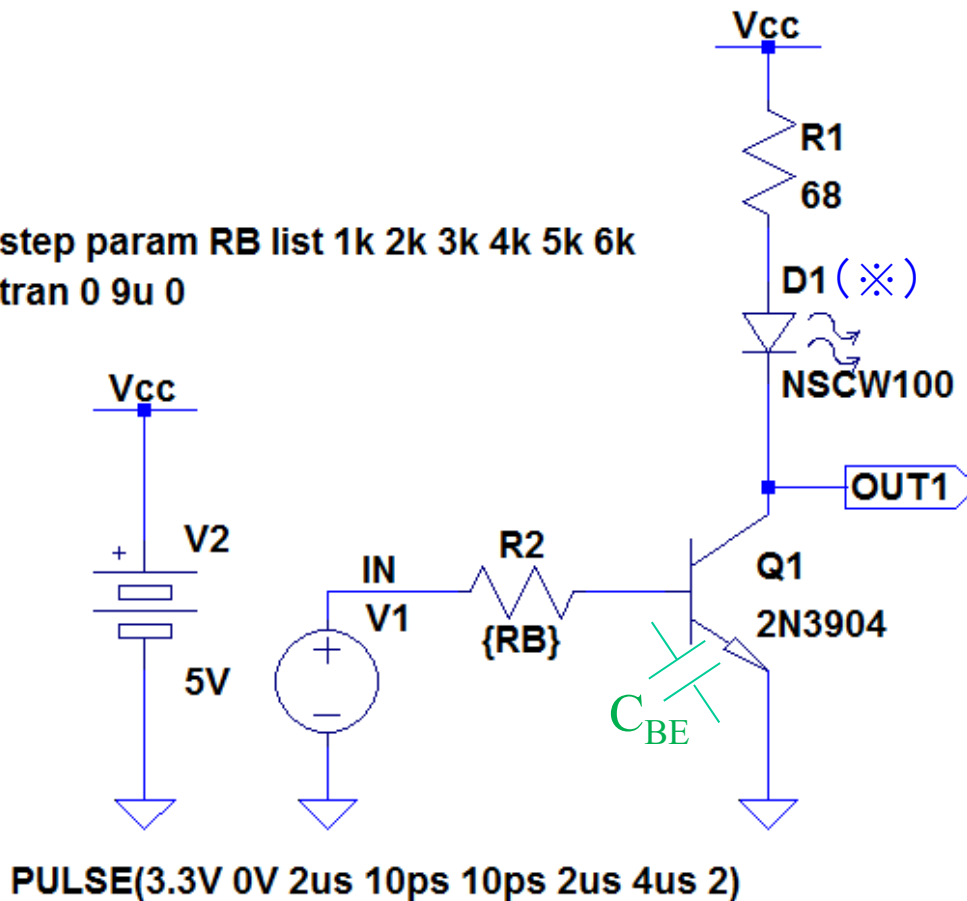


ROW1 ~ ROW8をダイナミック駆動により切り替えて64ドットの画面を表示させることができる。



# LEDフラッシュの実験(回路)

```
.step param RB list 1k 2k 3k 4k 5k 6k  
.tran 0 9u 0
```



※ 市販されている抵抗の値は、一定比率となるようにラインアップされている。”E系列”で検索して調べてみよう。E24系列が入手しやすい。

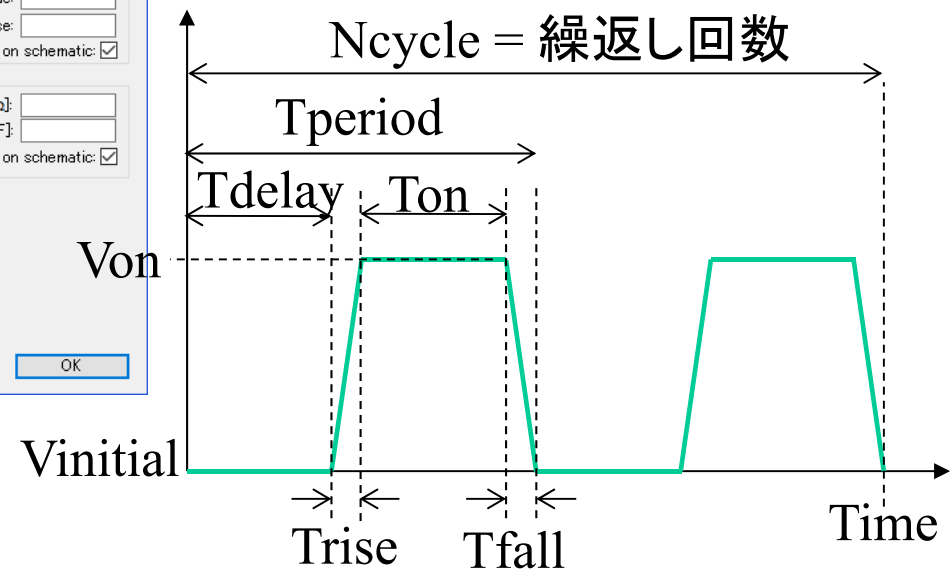
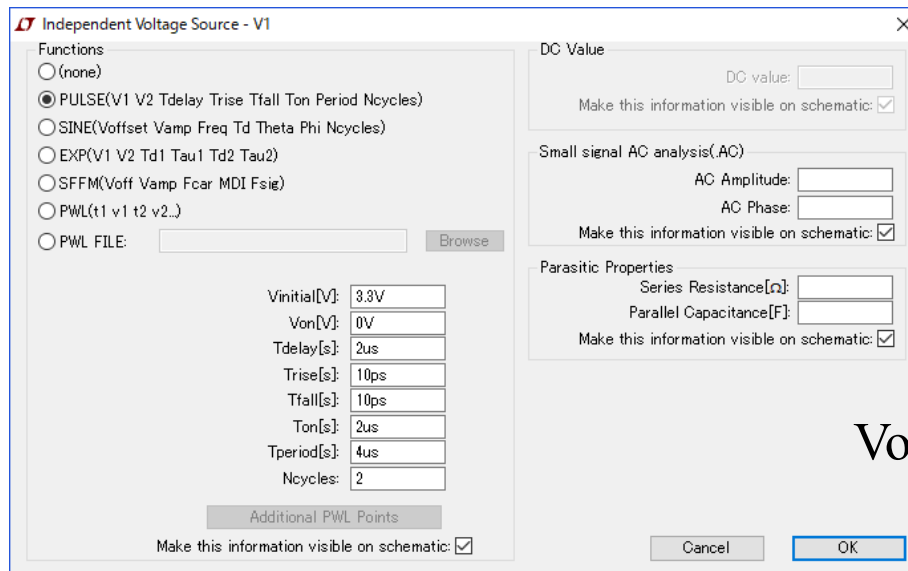
NSCW100(白色LED)

- $V_F = 3.5V$ ,  $I_F = 20mA$
- $V_{cc} = 5V$  (制御回路は3.3V)

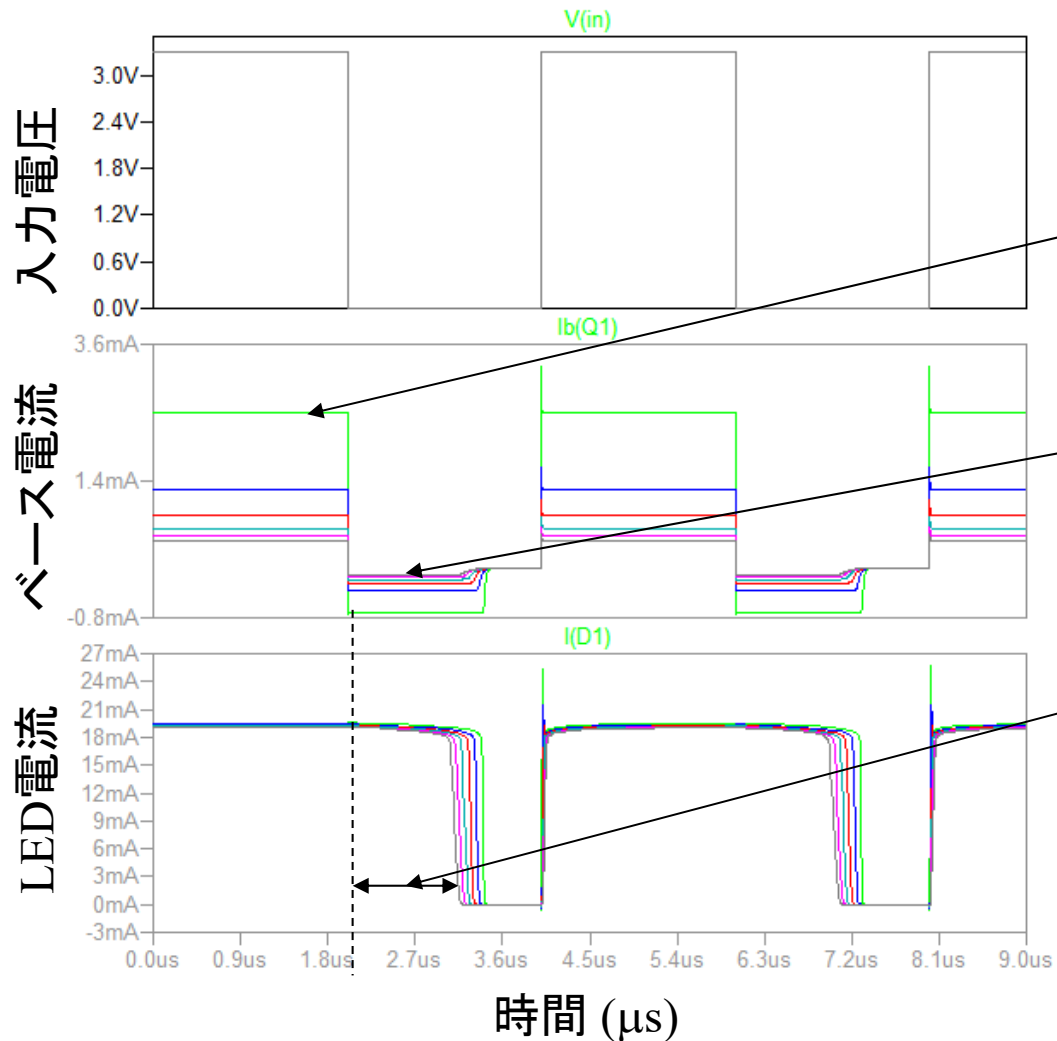
R2を大きくしすぎると、B-E間のpn接合容量 $C_{BE}$ に充放電する時間がかかるため、R1, R2を使用してD1の電流を制限している。

# PULSE波形の設定

- 波形を設定する電流源または電圧源を右クリック
- Advancedボタンをクリック



# LEDフラッシュの実験(結果)

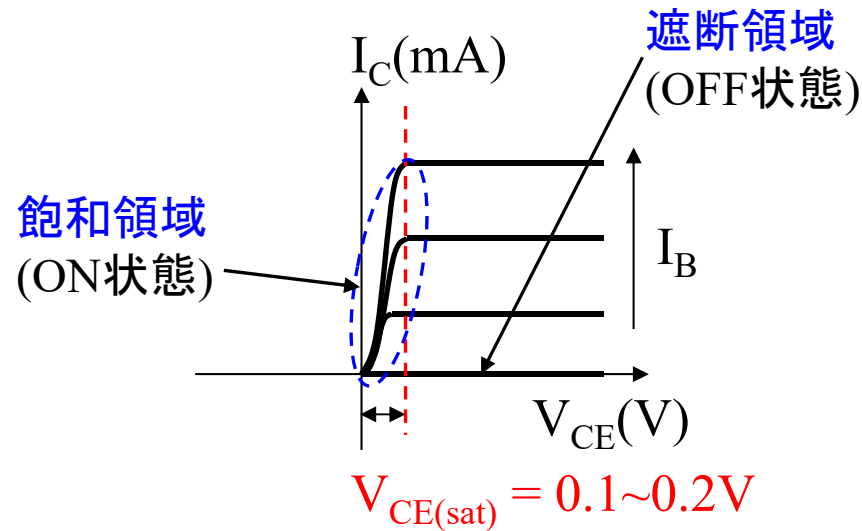


R2の抵抗値が小さいほど、ベース領域に電子が蓄積する。

ベース領域に蓄積した電子を放出する逆方向の電流。

ベース領域に蓄積した電子を放出する時間分だけLEDの消灯が遅れる(短時間フラッシュができない)。

# スイッチング遅延の原因



$V_{BE(sat)} = 0.6 \sim 0.7V$   
 $V_{CE(sat)} = 0.1 \sim 0.2V$   
このとき、 $V_{CB} \doteq -0.5V$  (負電圧)。

B-E間、B-C間ともに順方向電圧がかかる。

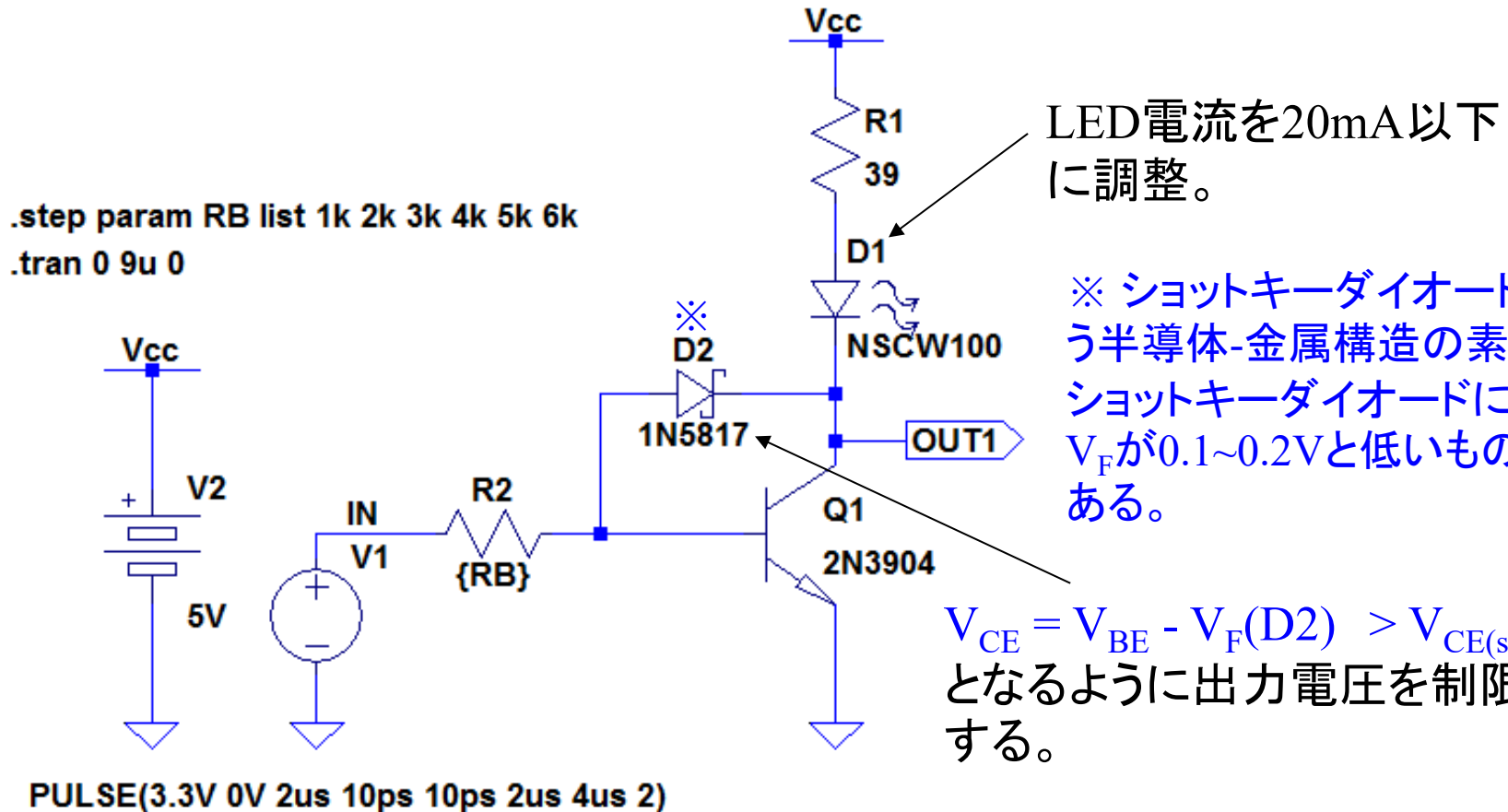
B-C間＝順方向  
B-E間＝順方向  
のとき、エミッタとコレクタの両側から電子が流れ込み、ベース内に蓄積してしまう。

ベースに蓄積した電子が再結合または放電して消えるまでOFF状態にならない。

ON→OFFの時に遅延時間が生じる。

# 飽和領域にならない改良

(ショットキーダイオードクランプ)





## 課題3. 2

1. 前スライドのシミュレーションを実行し、LEDの電流、トランジスタのベース電流、入力電圧V1の波形を調べ、レポートに上記の波形を貼り付けよ
2. ON→OFF時のスイッチング遅延が解消されていることを確認し、 $R1 = 1k\Omega$ のとき、入力電圧が立ち下がってから、LED電流が半分に減少するまでの時間を求めよ
3. トランジスタQ1について、 $V_{CE} > V_{CE(sat)}$  の条件(Q1が飽和領域に入らない条件)を満足していることを、シミュレーション結果のグラフを用いて示せ

## 3. 2節のまとめ

- トランジスタにベース電流を流すと、コレクタ-エミッタ間がONとなり電流が流れる
- トランジスタのベース電流を切ると、コレクタ-エミッタ間がOFFとなり電流が遮断される
- コレクタに流す電流は、ベースに接続した抵抗またはコレクタに接続した抵抗により調整できる
- トランジスタが飽和領域(ON)から遮断領域(OFF)に遷移するときに遅延が生じる
  - 遅延の原因は、トランジスタが飽和領域に入ると、E-B間、C-B間のpn接合が順方向電圧となり、ベースに電子が蓄積するため、ベース領域の電子が無くなるまで電流が流れ続けてしまうためである
- トランジスタが飽和領域( $V_{CE} < V_{CE(sat)}$ )に入らないようにする方法として、ショットキークランプがある