

## je veux un modele spice du 2N1683

### ChatGPT a dit :

Réflexion durant 23s

Parfait — voici un **modèle SPICE prêt à l'emploi** pour le transistor **2N1683** (version approximative basée sur les fiches techniques trouvées).

Je précise : le 2N1683 est listé comme **transistor germanium PNP** avec  $hFE \approx 50$ ,  $I_c \text{ max} \approx 0.1 \text{ A}$ ,  $V_{ce0} \approx 12 \text{ V}$  et  $f_t \approx 40 \text{ MHz}$  (sources citées en bas). Le modèle ci-dessous est **une approximation utilisable en simulation** ; vous devrez ajuster certains paramètres (IS, TF, Cje, RB, etc.) si vous avez mesures réelles ou une fiche technique complète. [alltransistors.com+1](http://alltransistors.com+1)

---

### Modèle SPICE (à coller dans votre fichier .cir / .asc)

```
* 2N1683 - modèle BJT PNP approximatif (germanium)
* Usage: placez la ligne .model dans votre netlist et utilisez QX C B E 2N1683
.model 2N1683 PNP (
+ IS=5e-7      * saturation current (germanium -> plus élevé que silicium)
+ BF=50       * forward beta (hFE) typique
+ NF=1.2
+ VAF=20      * Early voltage (approx)
+ IKF=0.05
+ ISE=1e-9
+ BR=2
+ NR=2
+ RB=20       * base resistance (Ω) - ajuster selon package
+ RC=1        * collector series resistance (Ω) - petit
+ RE=1        * emitter series resistance (Ω) - petit
+ CJE=6p      * base-emitter junction capacitance (pF) (f.d. ~6 pF)
+ CJC=3p      * base-collector capacitance (pF)
+ TF=2n       * forward transit time (s) -> lié à ft (~40 MHz)
+ XTF=0.5
+ TR=100n     * recombination lifetime (s)
+ ISL=5e-7
+ PTAT=0
)
```

\* Exemple d'utilisation (petit circuit de test)

```
VCC 1 0 DC 6V
Rload 1 2 1k
Q1 2 3 0 2N1683 ; Q1: collector=2, base=3, emitter=0
Vb 3 0 DC -0.2 ; polarisation base (PNP germanium Vbe ≈ -0.2..-0.3 V)
.options abstol=1e-12 reltol=1e-3
.dc Vb -0.5 0 0.01
.print dc I(Q1) IB(Q1) IC(Q1)
.end
```

---

### Remarques / conseils pratiques

- **Polarité PNP** : dans SPICE on déclare `.model <nom> PNP (...)` et on place le transistor `Qname C B E modelname`. Les tensions de base pour germanium sont plus faibles ( $V_{be} \approx 0.15-0.3 \text{ V}$ ), d'où dans l'exemple `Vb` négatif pour polariser correctement.
- **Paramètres à ajuster** : IS, TF, CJE, RB, RC, VAF — si vous avez une fiche technique complète ou mesures (courbe  $I_c/V_{be}$ ,  $f_t$ ), fournissez-les et je peux affiner les valeurs pour mieux coller aux données mesurées.
- **Validation** : faites un sweep DC base/collecteur et comparez  $I_c$ ,  $hFE$  et fréquences de coupure avec les spécifications. Adaptez BF, TF, CJE pour coller à  $hFE$  et  $f_t$  réels.
- **Germanium spécifique** : les modèles SPICE standards sont conçus pour silicium ; pour germanium IS est souvent plus élevé et il peut y avoir des fuites plus importantes — vérifiez les courants de fuite et  $I_{se}/I_{SE}$  si la précision est critique.
- **Si vous utilisez LTspice** : collez la `.model` dans un fichier `.asy` ou directement dans le schéma via `.op/.include`, puis placez un transistor générique et changez son attribut « Value » ou « Spicemodel » en 2N1683.

---

Si vous voulez, je peux :

- vous fournir **une version optimisée** pour LTspice (fichier .asc / .subckt) ;
- **affiner les paramètres** si vous me donnez une courbe  $I_c$  vs  $V_{be}$ ,  $h_{FE}$  typique ou mesures réelles ;
- ou convertir le modèle en **.subckt** comportant résistances série de borne si vous préférez un modèle plus détaillé.

Dites-moi directement ce que vous voulez (par ex. fichier LTspice prêt à ouvrir, modèle plus simple, ou estimation plus précise) et je vous fournis le fichier/modèle immédiatement. [alltransistors.com+1](http://alltransistors.com+1)

**Vous avez dit :**

**un modèle spice d'un transistor TUN utilisé par elektor**

**ChatGPT a dit :**

Réflexion durant 20s

Bonne question — **Elektor n'utilise pas un transistor unique appelé "TUN"** mais une *famille* générique : **TUN = "Transistor Universal NPN"** (un NPN petit signal répondant à des spécifications minimales). Les circuits Elektor marquent TUN quand *n'importe quel* transistor NPN qui respecte ces minima peut être utilisé. Les minima couramment cités sont notamment :  **$V_{ce0} \geq 20\text{ V}$ ,  $I_c \leq 100\text{ mA}$ ,  $h_{FE} (\beta) \geq 100$ ,  $P_{tot} \approx 100\text{ mW}$ ,  $f_T \geq 100\text{ MHz}$** . [Elektor+2wiki.analog.com+2](http://Elektor+2wiki.analog.com+2)

Ci-dessous je vous fournis **un modèle SPICE générique "TUN" (NPN)** que vous pouvez coller directement dans une netlist LTspice / SPICE. Ce modèle vise à respecter les minima Elektor ( $\beta \approx 100$ ,  $f_T \approx 100\text{ MHz}$ ,  $V_{ce0}$  élevé) tout en restant simple et utilisable dans des simulations de petits signaux.

---

### Modèle SPICE générique « TUN »

\* Modèle générique TUN - Transistor Universal NPN (Elektor)

\* Usage : placer la ligne .model dans votre netlist puis utiliser Q1 C B E TUN

```
.model TUN NPN (  
+ IS=1e-14 ; saturation current (silicon small-signal)  
+ BF=100 ; forward beta (hFE) minimum demandé par Elektor  
+ NF=1.0  
+ VAF=50 ; Early voltage -> limite effet Early (approx.)  
+ IKF=0.1  
+ ISE=1e-12  
+ BR=4  
+ NR=2  
+ RB=50 ; résistance interne base (ohm) - ajuster si besoin  
+ RC=0.5 ; résistance série collector  
+ RE=0.5 ; résistance série emitter  
+ CJE=4p ; Cbe (pF) - petit signal  
+ CJC=2p ; Cbc (pF)  
+ TF=1.59e-7 ; forward transit time (s) -> calé pour  $f_t \approx 100\text{ MHz}$  avec  $BF=100$   
+ XTF=0.5  
+ TR=200n  
+ ISC=0  
)
```

\* Exemple de test (DC et AC small-signal)

```
VCC Vc 0 DC 12  
RBias Vb 0 DC 2.7  
R1 Vc Collector 4.7k  
Q1 Collector Base 0 TUN  
Vin Base 0 AC 1 DC 0 ; small-signal AC source for gain measurement
```

\* analyse AC pour estimer  $f_t/h_{fe}$  en fréquence

```
.ac dec 50 10 1e9  
.print ac v(Collector) i(Q1) ib(Q1) ic(Q1)  
.end
```

---

## Explications rapides et conseils

- **Pourquoi  $TF = 1.59e-7$  s ?**

Le paramètre TF (forward transit time) et  $\beta$  déterminent la fréquence de transition approximative par la relation  $f_T \approx \beta 2\pi TF$ . Pour  $\beta=100$  et  $f_T \approx 100$  MHz,  $TF \approx 100 / (2\pi \cdot 100e6) \approx 1.59 \times 10^{-7}$  s (valeur utilisée ci-dessus). Ajustez BF ou TF si vous voulez un  $f_T$  plus élevé.

- **Paramètres à affiner selon le composant réel** : IS, RB, CJE/CJC, TF et VAF. Si vous avez un équivalent commercial (ex : 2N2222, BC547, etc.) qui doit représenter le TUN dans votre schéma, je peux extraire/adapter son modèle pour coller davantage aux comportements réels.
- **Germanium ?** Non — TUNs d'Elektor sont des **transistors silicium** planaires (donc IS bas). Si vous avez besoin d'un modèle germanium, il faut augmenter IS et revoir les fuites. [wiki.analog.com](http://wiki.analog.com)
- **Validation pratique** : lancez un sweep AC et vérifiez le gain en basse fréquence ( $\beta$ ) et la fréquence où  $|h_{21}|$  tombe à 1 ( $\approx f_T$ ). Comparez aussi la tenue  $V_{ce}$  en sweep DC.
- **Si vous utilisez LTspice** : collez la .model dans un fichier texte et Edit → SPICE Directive dans le schéma, ou right-click sur un transistor générique et mettez Value=TUN.

---

Si vous voulez, je peux maintenant (choisissez une option) :

1. **fournir un .subckt / fichier .asy LTspice prêt à ouvrir** (avec symbole et modèle embarqué) ;
2. **adapter ce modèle à un transistor réel** (BC547 / 2N2222 / BC337...) pour qu'il corresponde exactement à un composant de remplacement pratique ;
3. **affiner les paramètres TF / Cje / RB** si vous me donnez une courbe mesurée ( $I_c$  vs  $V_{be}$  ou réponse en fréquence).

Dites-moi quelle option vous intéresse (ou si vous préférez le modèle pour un transistor précis) — je vous prépare directement le fichier/modèle.

**Vous avez dit :**

**TF=1.59e-7**

**et pour un TUP?**

**ChatGPT a dit :**

Réflexion durant 7s

## Modèle SPICE pour TUP (Transistor Universal PNP — « complément » du TUN)

Voici un modèle PNP générique TUP compatible avec l'approche Elektor (petit signal PNP remplaçant générique). J'inclus aussi l'explication sur **TF = 1.59e-7** et la formule utilisée, plus une variante si vous préférez un  $\beta$  (BF) différent.

---

## Calcul de TF (explication numérique)

La relation approximative utilisée est :

$$f_T \approx \beta 2\pi \cdot TF \Rightarrow TF \approx \frac{f_T}{\beta 2\pi} \quad \text{ou} \quad TF \approx \frac{f_T}{2\pi \cdot \beta}$$

Si on vise  $f_T = 100$  MHz =  $100 \times 10^6$  Hz et  $\beta = 100$  :  
 $f_T = 100 \text{ MHz} = 100 \times 10^6 \text{ Hz}$  et  $\beta = 100$  :

Calcul **chiffre par chiffre** :

- $2\pi = 2 \times 3.141592653589793 = 6.2831853071795862$
- $\pi = 3.141592653589793$
- $6.2831853071795862 \times 100 = 628.31853071795862$

- $2\pi \cdot f_T = 6.283185307179586 \times 100,000,000 = 628,318,530.71795862 \pi \cdot f_T = 6.283185307179586 \times 100,000,000 = 628,318,530.7179586$
- $TF = 100 \cdot 628,318,530.7179586 \approx 1.5915494309189535 \times 10^{-7} \text{ s}$   $TF = \frac{100}{628,318,530.7179586} \approx 1.5915494309189535 \times 10^{-7} \text{ s}$

=> **TF ≈ 1.59e-7 s** (valeur que vous aviez mentionnée).

Si vous préférez  $\beta = 60$  (PNP ont parfois un hFE plus faible), alors :

$TF = 60 \cdot 628,318,530.7179586 \approx 9.54929658551372 \times 10^{-8} \text{ s} \approx 9.55e-8 \text{ s}$   
 $TF = \frac{60}{628,318,530.7179586} \approx 9.54929658551372 \times 10^{-8} \text{ s} \approx 9.55e-8 \text{ s}$

### Modèle SPICE (version principale — BF = 100, TF = 1.59e-7)

```
* Modèle générique TUP - Transistor Universal PNP (Elektor-style)
* Usage : Q1 C B E TUP (collector, base, emitter)
.model TUP PNP (
+ IS=1e-14 ; saturation current (silicon small-signal)
+ BF=100 ; forward beta (hFE) ciblé
+ NF=1.0
+ VAF=50 ; Early voltage (approx)
+ IKF=0.1
+ ISE=1e-12
+ BR=4
+ NR=2
+ RB=50 ; résistance de base interne (Ω)
+ RC=0.5 ; résistance série collector (Ω)
+ RE=0.5 ; résistance série emitter (Ω)
+ CJE=4p ; Cbe (pF)
+ CJC=2p ; Cbc (pF)
+ TF=1.59e-7 ; forward transit time (s) -> pour ft ≈100 MHz avec BF=100
+ XTF=0.5
+ TR=200n
+ ISC=0
)
```

### Variante pour PNP avec $\beta = 60$ (TF ajusté)

```
.model TUP60 PNP (
+ IS=2e-14
+ BF=60
+ NF=1.0
+ VAF=40
+ RB=60
+ RC=0.7
+ RE=0.7
+ CJE=5p
+ CJC=3p
+ TF=9.55e-8 ; calculé pour BF=60 et ft≈100MHz
+ XTF=0.6
+ TR=250n
)
```

### Exemple d'utilisation (test simple)

```
VCC 1 0 DC -12V ; notez polarité négative pour PNP alimenté "classiquement"
Rload 1 2 10k
Q1 2 3 0 TUP ; Q1 : collector=2, base=3, emitter=0
Vb 3 0 DC -0.6 ; polarisation base (exemple)
.dc Vb -1 0 0.01
.end
```

## Remarques pratiques

- **Polarité PNP** : les sources et polarités dans les exemples sont adaptées pour un PNP (tensions négatives relatives au 0 V). Adaptez selon votre schéma.
- **TF = 1.59e-7** est valable si vous voulez **ft**  $\approx$  **100 MHz** et **BF**  $\approx$  **100**. Si vous changez BF ou la cible ft, recalculer TF avec la formule ci-dessus.
- **Ajustements recommandés** : IS, RB, CJE/CJC, VAF et TF pour coller à un composant réel. Les PNP petits signaux ont souvent un hFE légèrement inférieur aux NPN équivalents — la variante BF=60 est fournie pour cela.
- **LTspice** : copiez la .model dans une directive SPICE sur le schéma ou dans un fichier .include. Pour un transistor physique, mettez Value=TUP sur le symbole générique PNP.

Si vous voulez que je :

- génère un .subckt LTspice avec symbole prêt à l'emploi, ou
- adapte le modèle à un transistor PNP commercial précis (BC557, 2N3906, BC327...),

je vous fournis ça directement (précisez le composant souhaité).

**Vous avez dit :**

**comment définir la puissance max**

**ChatGPT a dit :**

Réflexion durant 13s

Super — voilà une réponse complète et pratique en français pour **définir la puissance maximale** qu'un transistor peut dissiper (et comment l'utiliser en simulation).

## Principes essentiels

- **Puissance dissipée instantanée** dans le transistor :

$$P = V_{CE} \times I_{CP} = V_{CE} \times I_C$$

(pour un transistor en régime actif/collecteur-émetteur ; si vous connaissez les courants et tensions instantanés, calculez  $P(t) = V_{CE}(t) \cdot I_C(t)$  et intégrez/suivez selon la durée).

- **Limite thermique continue** (ce qui détermine la puissance maximale en continu) : elle dépend de la capacité de dissipation thermique du boîtier et du refroidissement. La formule de base :

$$P_{max,contin} = T_{j,max} - T_a \Theta_{JA} \Rightarrow P_{max,contin} = \frac{T_{j,max} - T_a}{\Theta_{JA}}$$

où

- $T_{j,max}$  : température maximale de jonction (°C) (valeur constructeur, souvent 150 °C pour silicium),
- $T_a$  : température ambiante (°C),
- $\Theta_{JA}$  : résistance thermique jonction → ambiance (°C/W) du boîtier + montage.
- **Safe Operating Area (SOA)** : même si  $P_{max,contin}$  est inférieur à  $P_{max}$ , le transistor peut être détruit par des combinaisons élevées de  $V_{CE}$  et  $I_C$  (effet d'impact, courant de pointe, avalanches). Vérifiez la courbe SOA du datasheet — elle donne limites selon durée d'impulsion.
- **Dérating** : les constructeurs fournissent souvent une courbe de puissance max en fonction de la température ambiante — typiquement  $P_{max}$  diminue linéairement après une température seuil (ex. 25 °C).

## Étapes pratiques pour déterminer la puissance max

1. **Trouver les paramètres constructeur** :  $T_{j,max}$ ,  $\Theta_{JA}$  (ou  $\Theta_{JC}$  si on utilise un dissipateur), et la courbe SOA.
2. **Choisir la température ambiante d'utilisation** ( $T_a$ ).

3. **Calculer la puissance continue max** :  $P_{max,contin} = (T_{j,max} - T_a) / \Theta_{JA}$
4. **Comparer à l'utilisation réelle** : calculez  $P = V_{CE} \cdot I_{CP} = V_{CE} \cdot I_C$  en régime continu. Si  $P < P_{max,contin}$  OK thermiquement (mais vérifiez SOA).
5. **Vérifier transitoires** : pour impulsions, utiliser la courbe SOA correspondant à la durée d'impulsion ( $\mu s$ ,  $ms$ ,  $s$ ).
6. **Sécurité** : appliquer facteur de sécurité (ex. ne pas dépasser 70–80% de  $P_{max,contin}$  si pas de mesures de refroidissement fiables).

### Exemples numériques (illustration)

- Exemple A : petit transistor en boîtier **TO-92** (valeur typique)
  - $T_{j,max} = 150^\circ C$ ,  $T_a = 25^\circ C$ ,  $\Theta_{JA} \approx 200^\circ C/W$
  - $P_{max,contin} = (150 - 25) / 200 = 0,625 \text{ W}$
  - environ **0,6 W** en continu (valeur indicative ; regarder  $\Theta_{JA}$  du datasheet).
- Exemple B : **TO-220** sans radiateur (montage sur PCB)  $\Theta_{JA} \approx 50^\circ C/W$ 
  - $P_{max,contin} = (150 - 25) / 50 = 2,5 \text{ W}$
  - **~2,5 W**. Avec radiateur  $\Theta_{JA}$  tombe beaucoup plus bas =>  $P_d$  augmente.

### En simulation SPICE / LTspice — comment mesurer et vérifier

- **Mesurer la puissance dissipée** par le transistor : multipliez tension C-E par le courant collecteur. Exemple LTspice (nom des noeuds à adapter) :
 

```
* Supposons Q1 Collector=C, Base=B, Emitter=E
.measure P_Q1 param V(C,E)*I(Q1)
.measure P_Q1_max max V(C,E)*I(Q1)
.measure P_Q1_avg avg V(C,E)*I(Q1)
```
- $I(Q1)$  renvoie le courant du transistor (attention orientation selon SPICE ; vérifiez le signe).
- Vous pouvez faire un `.tran` pour voir la puissance en temps, ou un `.param` et `.step` pour sweeper.
- **Calculer Pmax dans la netlist** (ex. paramétrisation) :
 

```
.param Tjmax=150 Ta=40 thetaJA=200
.param Pmax_cont = (Tjmax - Ta)/thetaJA
.print param Pmax_cont
```
- Puis comparez le résultat de `.measure` avec `Pmax_cont`.
- **Electro-thermique** : les simulateurs permettent parfois d'ajouter modèles thermiques ( $R_{th}$ ,  $C_{th}$ ) et couplage électro-thermique pour simuler élévation de  $T_j$ . LTspice n'a pas d'élément thermique natif simple mais on peut faire sous-circuits thermiques ou utiliser outils spécialisés.

### Conseils pratiques / checklist d'ingénieur

- Toujours **vérifier la SOA** pour fortes tensions et courants (surtout commutation).
- Si vous avez des **impulsions courtes**, comparez avec courbes SOA pour la durée de l'impulsion ; souvent un transistor supporte des puissances instantanées bien supérieures pendant  $\mu s$ – $ms$ .
- **Améliorez le refroidissement** (dissipateur, plan de cuivre, ventilation) pour réduire  $\Theta_{JA}$ . Doubler la surface cuivre réduit notablement  $\Theta_{JA}$ .
- **Ajouter marge de sécurité** (par ex. travailler à  $\leq 70\%$  de la valeur calculée).
- **Protéger** : limiter courant (résistance, MOSFETs en driver), ajouter surveillance thermique si nécessaire.

- **Mesurer sur carte** : monitorer température de boîtier en conditions réelles pour valider hypothèses.

## Résumé condensé

- Formule clé :  $P = V_{CE} \cdot I_{CP} = V_{CE} \cdot I_C$ .
- Puissance continue max thermique :  $P_{max,contin} = (T_{j,max} - T_a) / \Theta_{JA}$   $\Theta_{JA} = (T_{j,max} - T_a) / P_{max,contin}$ .
- Toujours vérifier SOA, durées d'impulsion et appliquer un facteur de sécurité.

---

Si vous me dites **le boîtier (TO-92, SOT-23, TO-220, etc.)**, la **température ambiante envisagée**, et si vous avez **les valeurs constructeur** ( $T_{j,max}$ ,  $\Theta_{JA}$ ), je calcule tout de suite la **valeur numérique précise** de  $P_{max,contin}$  et je vous fournis les directives SPICE prêtes à coller pour mesurer/contrôler la puissance dans votre simulation. Voulez-vous que je fasse ce calcul avec un exemple de boîtier (TO-92 ou SOT-23) ?

**Vous avez dit :**  
**et dans le modèle spice?**

**ChatGPT a dit :**

Très bonne remarque 🙌

En fait, dans **SPICE**, la notion de « puissance maximale » n'existe **pas directement** dans le modèle .model d'un transistor BJT.

---

## Pourquoi ?

- Les modèles SPICE BJT (.model ... NPN/PNP) décrivent **le comportement électrique** : courants, gains, capacités, délais de transit, résistances internes, etc.
- La **dissipation de puissance** est une conséquence de la simulation :

$$P(t) = V_{CE}(t) \cdot I_C(t) + V_{BE}(t) \cdot I_B(t) \approx V_{CE}(t) \cdot I_C(t)$$

mais ce n'est pas une **limite incluse** dans le modèle.

- Autrement dit : SPICE vous donne la puissance instantanée dissipée, mais **il ne va pas couper ou griller le transistor tout seul** si vous dépassez  $P_{max}$ .

---

## Comment intégrer la limite de puissance dans une simulation ?

Il y a 3 approches possibles :

### 1. Mesures / vérifications avec directives

Dans LTspice, vous pouvez demander :

```
.measure tran P_Q1 param V(C,E)*I(Q1)
.measure tran P_Q1_max MAX V(C,E)*I(Q1)
.measure tran P_Q1_avg AVG V(C,E)*I(Q1)
```

- I(Q1) donne le courant collecteur (attention au signe).
- Ça vous donne la puissance dissipée, à comparer manuellement à  $P_{max}$  (issu du datasheet ou calcul thermique).

---

### 2. Ajout d'un modèle thermique (subcircuit)

Certains simulateurs SPICE avancés (LTspice, PSpice, Spectre, etc.) permettent de créer un **réseau thermique équivalent RC** pour modéliser la montée de température :

- Résistance thermique ( $\Theta_{JA} \backslash \Theta_{JA} = \text{°C/W}$ )
- Capacité thermique = J/°C
- On relie la puissance dissipée ( $V \cdot I$ ) comme une source de courant dans ce réseau.
- Cela permet de calculer la **température de jonction** et de simuler le **dérating** de la puissance max.

Mais il faut construire ce sous-circuit, car ce n'est pas inclus par défaut dans .model.

---

### 3. Protection artificielle (subckt avec limite)

On peut encapsuler le transistor dans un .subckt qui :

- Calcule la puissance  $P = V_{CE} \cdot I_C = V_{CE} \cdot I_C = V_{CE} \cdot I_C$  avec un élément dépendant,
- Compare à un paramètre  $P_{max}$ ,
- Et coupe ou force un état de défaillance si la limite est dépassée.

→ C'est plus de la simulation de fiabilité que de l'électronique pure, mais c'est faisable si vous voulez « voir » le transistor claquer en simulation.

---

#### En résumé

- Dans un .model SPICE, on **ne définit pas Pmax**.
  - On définit uniquement les paramètres électriques (IS, BF, VAF, TF, CJE, etc.).
  - **La puissance max doit être surveillée par post-traitement** (directives .measure) ou via un **modèle thermique/subckt** si vous voulez modéliser l'échauffement.
  - Donc pour un TUN/TUP Elektor, vous gardez un modèle simple, et c'est à vous de vérifier que  $V_{CE} \cdot I_C$  reste sous la limite calculée  $(T_{jmax} - T_a) / \Theta_{JA}$ .
- 

🔗 Voulez-vous que je vous montre un **exemple de subcircuit SPICE “BJT avec limite de puissance”**, qui arrête le transistor quand  $P > P_{max}$  ? (ça peut être utile si vous voulez « tester » la robustesse en simulation).

**Vous avez dit :**

**non j'ai compris**

**ChatGPT a dit :**

Parfait 😊

Donc à retenir : dans le **.model SPICE**, on ne définit jamais directement la puissance max — c'est à vérifier après coup, ou à ajouter via un subcircuit thermique si besoin.

Si un jour tu veux, je peux te préparer un petit fichier LTspice qui trace automatiquement la dissipation du transistor pour la comparer à la limite thermique calculée.

Veux-tu que je te montre comment tracer la **courbe de puissance dissipée** directement dans LTspice (sans mesures manuelles) ?

**Vous avez dit :**

**oui**

**ChatGPT a dit :**

Super 🙌

Voici la manière la plus simple de **visualiser directement la puissance dissipée d'un transistor** dans **LTspice** :

---

## 1. Formule de puissance dans LTspice

- La puissance dissipée dans un élément est toujours :

$$P(t) = V \cdot I_P(t) = V \cdot I$$

- Pour un transistor BJT **Q1** :
- $V(Q1:c, Q1:e)$  = tension collecteur-émetteur
- $I(Q1)$  = courant dans le collecteur (par convention LTspice)

☞ Donc la dissipation instantanée est :

$$V(Q1:c, Q1:e) \cdot I(Q1)$$

---

## 2. Comment tracer dans LTspice

1. Lance ta simulation .tran (ou .ac si tu veux vérifier en régime AC, mais en général puissance = tran).
  2. Dans la fenêtre de tracé (waveform viewer) :
  3. Clique droit → **Add Trace**
  4. Tape :  
 $V(Q1:c, Q1:e) \cdot I(Q1)$
  5. Ça affiche la **puissance instantanée dissipée**.
  6. Si tu veux la puissance moyenne :
  7. Menu **Plot Settings** → **View** → **FFT/Measure** ou
  8. Ajoute directive .measure tran Pavg AVG V(Q1:c, Q1:e) \* I(Q1)
- 

## 3. Exemple complet de netlist

\* Exemple de transistor dissipant une puissance

```
Vcc N1 0 12
Rc N1 Nc 1k
Q1 Nc Nb 0 TUN ; Transistor NPN
Vb Nb 0 DC 0.7
.model TUN NPN (BF=100 IS=1e-14 VAF=50 TF=1.6e-7)

.tran 0 10m 0 1u
.backanno
.end
```

Ensuite dans le waveform viewer :

- Tracer  $V(Nc, 0) \cdot I(Q1)$
  - → courbe de puissance dissipée instantanément.
- 

## 4. Option bonus : Courbe comparée à la limite

Tu peux aussi ajouter dans le schéma une **ligne horizontale représentant  $P_{max}$**  :

```
.param Pmax=0.6 ; ex : 600 mW
.plot tran V(Nc, 0) * I(Q1) {Pmax}
```

→ Ça affiche la dissipation réelle **et** la limite max en même temps.

---

🔗 Veux-tu que je t'écrive un **fichier LTspice .asc prêt à ouvrir** avec ce tracé déjà configuré (puissance instantanée + ligne Pmax) ?

**Vous avez dit :**

**non c'est bon**

**ChatGPT a dit :**

Parfait 🤖 content que ce soit clair pour toi maintenant.

Tu as donc tout ce qu'il faut pour tracer et comparer la puissance dissipée dans LTspice.

N'hésite pas si tu veux qu'on regarde ensuite un exemple plus poussé (par ex. simulation avec charges variables ou impulsions pour voir les limites en SOA).

Tu veux que je prépare aussi une petite **méthode rapide pour vérifier la SOA d'un transistor** directement en simulation, ou ça va ?

**Vous avez dit :**

**ça va**

**ChatGPT a dit :**

Parfait 🤖

Alors je te laisse avec ça — tu as tout pour surveiller la puissance et rester dans les limites thermiques en SPICE.

Bonne bidouille et bonne simu ! 🚀

Si jamais tu bloques sur un autre point (LTspice, modèles, SOA, etc.), tu peux revenir me voir.

**Vous avez dit :**

**comment est calculé Is?**

**ChatGPT a dit :**

Bonne question 🤖

Dans le modèle SPICE d'un BJT, le paramètre **IS** est le **courant de saturation inverse de la jonction base-émetteur** (parfois noté **ISI\_SIS**, « transport saturation current »).

---

## 1. Rôle physique de ISI\_SIS

- C'est la constante qui fixe le niveau de courant de base/collecteur dans l'équation exponentielle du BJT :

$$I_C \approx I_S \cdot (e^{V_{BE}/(n V_T)} - 1) \approx I_S \cdot \big(e^{V_{BE}/(n V_T)} - 1\big)$$

avec :

- $V_T = kT/q \approx 25.85 \text{ mV}$  à  $300 \text{ K}$   $V_T = kT/q \approx 25.85 \text{ mV}$  à  $300 \text{ K}$  (tension thermique),
  - $n \approx 1$  (facteur d'idéalité),
  - $V_{BE}$  = tension base-émetteur.
  - En pratique, **IS détermine la position de la courbe ICI\_CIC vs VBEV\_{BE}**.  
→ Plus **ISI\_SIS** est grand, plus un petit  $V_{BE}$  suffit pour obtenir un certain **ICI\_CIC**.
- 

## 2. Ordres de grandeur

- Pour des transistors silicium petits signaux :  
 $I_S \sim 10^{-14} \text{ A}$  (valeur courante dans les modèles SPICE).
  - Pour des transistors germanium :  
**ISI\_SIS** est souvent bien plus élevé ( $10^{-8}$  à  $10^{-7} \text{ A}$ ), à cause de la plus faible bande interdite et des fuites plus importantes.
-

### 3. Comment l'estimer à partir d'un datasheet

On utilise la relation :

$$I_S = I_C e^{V_{BE}/V_T} \Rightarrow I_S = e^{V_{BE}/V_T} I_C$$

Exemple :

- Datasheet donne  $I_C = 1 \text{ mA}$  pour  $V_{BE} \approx 0.7 \text{ V}$ .
- À  $T = 300 \text{ K}$ ,  $V_T \approx 25.85 \text{ mV}$ .

$$I_S \approx \frac{I_C}{e^{V_{BE}/V_T}} = \frac{1 \times 10^{-3}}{e^{0.7/0.02585}} \approx \frac{1 \times 10^{-3}}{5.3 \times 10^{11}} \approx 1.9 \times 10^{-15} \text{ A}$$

Donc ici  $I_S \approx 2 \times 10^{-15} \text{ A}$ .

### 4. Dépendance en température

- $I_S$  varie **très fortement** avec la température ( $\approx$  double tous les  $8-10^\circ\text{C}$ ).
- C'est pour cela que  $V_{BE}$  baisse d'environ  $-2 \text{ mV}/^\circ\text{C}$  avec la température : l'augmentation de  $I_S$  entraîne une compensation de la loi exponentielle.

### 5. Dans un modèle SPICE

- Quand vous entrez `.model QX NPN (IS=1e-14 ...)`, vous dites au simulateur : « La jonction BE a un courant de saturation de  $10^{-14} \text{ A}$  à  $27^\circ\text{C}$  ».
- Cela place la courbe  $I_C - V_{BE}$  correctement.
- Si vous voulez coller au datasheet, vous pouvez recalculer  $I_S$  à partir d'un point  $(I_C, V_{BE})$  fourni dans la fiche.

🔗 Veux-tu que je te montre **comment recalculer  $I_S$  directement à partir d'un point du datasheet** d'un transistor particulier (par ex. BC547, 2N3904...), avec une petite formule prête à utiliser dans une calculatrice/Excel ?

**Vous avez dit :**

**oui**

**ChatGPT a dit :**

Réflexion durant 19s

Parfait — je te donne la méthode complète, prête à copier-coller dans Excel/Calc, plus un **exemple numérique détaillé** (calcul **chiffre par chiffre** comme demandé) et la variante « deux points » quand tu as deux couples  $(I_C, V_{BE})$ .

### Formule de base

Pour un BJT en régime actif :

$$I_C \approx I_S e^{V_{BE}/V_T} \Rightarrow I_S \approx \frac{I_C}{e^{V_{BE}/V_T}}$$

=> isoler  $I_S$  :

$$I_S = I_C \exp\left(-\frac{V_{BE}}{V_T}\right)$$

où

- $I_C$  en ampères,

- $V_{BEV\_BE}$  en volts,
- $n$  = facteur d'idéalité (typiquement 1...2 ; prendre 1 si inconnu),
- $V_T = kT/q$  tension thermique  $\approx 25.85 \text{ mV}$  à 300 K ( $\approx 27^\circ \text{C}$ ).

### Étapes pratiques (1 point connu)

1. Choisis  $I_{C\_CIC}$  et  $V_{BEV\_BE}$  (point du datasheet).
2. Choisis  $n$  (si absent, prends  $n=1$ ).
3. Calcule  $V_T$  (à 300 K,  $V_T \approx 0.02585 \text{ V}$ ).
4. Calcule l'exponentiel puis  $I_{S\_SIS}$  avec la formule ci-dessus.

### Exemple numérique (point unique)

Supposons un point datasheet :  $I_C = 1 \text{ mA} = 1.0 \times 10^{-3} \text{ A}$  et  $V_{BE} = 0.700 \text{ V}$ . On prend  $n=1$  et  $T=300 \text{ K}$ .

Calcul **chiffre par chiffre** :

1.  $V_T = 0.02585 \text{ V}$  (valeur usuelle à 300K).
2. Calculer le rapport  $\frac{V_{BE}}{V_T} = \frac{0.700}{0.02585} \approx 27.07$ .
3.  $0.02585 \times 27 = 0.69795$ .
4. Reste pour atteindre 0.700 :  $0.700 - 0.69795 = 0.00205$ .
5.  $0.02585 \times 0.0728 \approx 0.0018612$ .
6.  $0.02585 \times 0.0728 \approx 0.00188068$ .
7. En affinant on obtient environ  $0.700 \approx 27.07 \times 0.02585$  (arrondi raisonnable).

Donc l'exposant vaut  $\approx 27.07$ .

8. Calculer  $e^{27.07}$ . On peut décomposer :
9.  $e^{27.07} = e^{27} \times e^{0.07}$ .
10.  $e^{27} \approx 5.320482 \times 10^{11}$  (valeur connue).
11.  $e^{0.07} \approx 1.072508$ .
12. Produit :  $5.320482 \times 10^{11} \times 1.072508 \approx 5.718 \times 10^{11}$ .

13. Enfin :

$I_S = 1.0 \times 10^{-3} \times 5.718 \times 10^{11} \approx 5.718 \times 10^{-15} \text{ A}$ .

Donc pour cet exemple,  $I_S \approx 5.718 \times 10^{-15} \text{ A}$ .

### Méthode avec deux points (plus robuste — permet d'estimer $n$ aussi)

Si tu as deux points  $(I_{C1}, V_{BE1})$  et  $(I_{C2}, V_{BE2})$ , tu peux trouver  $n$  puis  $I_{S\_SIS}$ .

À partir de  $\ln I_C = \ln I_S + \frac{V_{BE}}{V_T}$  :  $\ln I_{C1} = \ln I_S + \frac{V_{BE1}}{V_T}$  et  $\ln I_{C2} = \ln I_S + \frac{V_{BE2}}{V_T}$

1. Calcul de  $n$  :

$$n = \frac{V_{BE2} - V_{BE1}}{V_T \cdot (\ln I_{C2} - \ln I_{C1})}$$

2. Puis  $IS_{SIS}$  (avec point 1 par exemple) :

$$IS = I_{C1} \exp\left(\frac{I_{C1}}{\exp\left(\frac{V_{BE1}}{n V_T}\right)}\right) = \exp(n V_T V_{BE1}) I_{C1}$$


---

### Formules prêtes pour Excel / Calc

Supposons :

- cellule A2 =  $I_{C1}$  (en A)
- cellule B2 =  $V_{BE}$  (en V)
- cellule C2 =  $n$  (mettre 1 si inconnu)
- cellule D2 = TTT en K (mettre 300 si 27 °C)

**Vt (cellule E2) :**

$$= (8.617333262145E-5 * D2) ; k/q \text{ en eV/K} \rightarrow \text{donne } V_T \text{ si } k \text{ in eV/K}$$

ou plus simple pour 300 K :

$$= 0.02585$$

**IS (cellule F2) :**

$$= A2 / \text{EXP}( B2 / ( C2 * E2 ) )$$

**Deux points (A2,B2) et (A3,B3) :**

**n (cellule C4) :**

$$= ( B3 - B2 ) / ( E2 * ( \text{LN}(A3) - \text{LN}(A2) ) )$$

**IS (cellule F4) :**

$$= A2 / \text{EXP}( B2 / ( C4 * E2 ) )$$

(Remarque : Excel utilise LN pour le logarithme naturel et EXP pour l'exponentielle.)

---

### Température — correction rapide

$IS_{SIS}$  varie fortement avec la température. Une approximation détaillée :

$$IS(T) \propto T^2 \exp(-E_g/kT) \Rightarrow IS(T) \propto T^2 \exp(-kTE_g)$$

Formule exacte pour changement de température ( $T_1 \rightarrow T_2$ ) :

$$IS(T_2) = IS(T_1) \left(\frac{T_2}{T_1}\right)^2 \exp\left(-E_g k \left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right) = IS(T_1) \left(\frac{T_2}{T_1}\right)^2 \exp\left(\frac{-E_g}{k} \left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right)$$

avec  $E_g \approx 1.12$  eV pour le silicium,  $k = 8.617333e-5$  eV/K.

En pratique, règle empirique :  **$IS_{SIS}$  augmente d'environ  $\times 2$  tous les 8–10 °C** (ordre de grandeur).

---

### Conseils pratiques / pièges à éviter

- Toujours travailler en **ampères** et **volts**.
- Vérifie si le point du datasheet est mesuré à une température donnée (souvent 25 °C).
- Si le datasheet donne  $V_{BE}$  pour une valeur de  $I_{C1}$  et une autre condition (ex. courant de base limité), vérifie le contexte.

- Si tu as seulement un point sous courant de fort gain (par ex. Ic à saturation), la méthode devient invalide — il faut un point en régime actif.

Si tu veux, je peux :

- **faire le calcul pour un transistor réel** (donne-moi un point du datasheet ou le modèle : ex. BC547 → par ex. 2 mA à  $V_{be}=0,68$  V) et je te donne  $I_{S,SIS}$  avec tous les détails ;
- ou te fournir un petit **fichier Excel** (je peux générer la formule dans un tableau) prêt à utiliser. Laquelle préfères-tu ?

**Vous avez dit :**

**Qu'est ce que  $kT$**

**ChatGPT a dit :**

Très bonne question 🙏

Le terme  $kT$  apparaît tout le temps en physique des semi-conducteurs, et en particulier dans l'équation exponentielle des diodes et des transistors bipolaires.

### 1. Définition

- **$k$  = constante de Boltzmann**

$$k = 1.380\,649 \times 10^{-23} \text{ J/K} \quad k = 1.380\,649 \times 10^{-23} \text{ J/K}$$

(ou en électronique :  $k = 8.617 \times 10^{-5} \text{ eV/K}$   $k = 8.617 \times 10^{-5} \text{ eV/K}$ ).

- **$T$  = température absolue en kelvin (K).**  
Exemple :  $27^\circ\text{C} \rightarrow T = 300 \text{ K}$   $T = 300 \text{ K}$

Donc :

$$kT = (1.380\,649 \times 10^{-23}) \times T \text{ [J]} \quad kT = (1.380\,649 \times 10^{-23}) \times T \quad kT = (1.380\,649 \times 10^{-23}) \times T \text{ [J]}$$

### 2. Rôle en électronique

- On divise souvent par la **charge élémentaire  $q$**  ( $= 1.602 \times 10^{-19} \text{ C}$ ) pour obtenir la **tension thermique** :

$$V_T = \frac{kT}{q} \quad V_T = \frac{kT}{q}$$

- À 300 K :

$$V_T = \frac{1.380\,649 \times 10^{-23} \times 300}{1.602 \times 10^{-19}} \approx 0.02585 \text{ V} = 25.85 \text{ mV} \quad V_T = \frac{1.380\,649 \times 10^{-23} \times 300}{1.602 \times 10^{-19}} \approx 0.02585 \text{ V} = 25.85 \text{ mV}$$

C'est cette tension  $V_T$  qui apparaît dans l'équation du courant des diodes et des transistors :

$$I \approx I_S e^{V_{BE}/(n V_T)} \quad I \approx I_S e^{V_{BE}/(n V_T)}$$

### 3. Intuition

- $kT$  (ou plutôt  $kT/q$ ) représente l'**énergie thermique moyenne par électron** dans un semi-conducteur.
- Plus la température est élevée, plus  $kT$  est grand → les porteurs sont plus facilement excités → les fuites ( $I_{S,SIS}$ ) augmentent.
- C'est pour ça que  $I_{S,SIS}$  double environ tous les 8–10 °C.

#### 4. Résumé

- $k$  = constante de Boltzmann.
- $T$  = température absolue (K).
- $kT$  = énergie thermique moyenne disponible par particule (en joules).
- $kT/q = VT$  /  $kT/q = V_T$  /  $kT/q = V_T \approx 26$  mV à 300 K = **tension thermique**, fondamentale dans l'équation des jonctions PN.

🔗 Veux-tu que je te fasse un petit **tableau des valeurs de  $VT = kT/q$**  pour différentes températures (ex.  $-40$  °C,  $0$  °C,  $25$  °C,  $100$  °C...) pour visualiser son évolution ?

**Vous avez dit :**

**comment est calculé VAF**

**ChatGPT a dit :**

Super question 🧐 Tu touches là à un paramètre SPICE fondamental : **VAF** (ou  $V_A$ ), la **tension d'Early**.

#### 1. Qu'est-ce que VAF (tension d'Early) ?

- C'est le paramètre qui modélise la dépendance du courant collecteur  $I_{CIC}$  avec la tension collecteur-émetteur  $V_{CE}$  en régime actif.
- Idéalement, dans un BJT parfait,  $I_{CIC}$  ne dépend que de  $V_{BE}$ .
- En réalité, quand on augmente  $V_{CE}$ ,  $I_{CIC}$  augmente légèrement (modulation de largeur de base → effet Early).
- Cette pente est modélisée par la relation :

$$I_C(V_{CE}) \approx I_{C0} \left( 1 + \frac{V_{CE}}{V_A} \right) \approx I_{C0} \left( 1 + \frac{V_{CE}}{V_A} \right)$$

où  $V_A$  est la **tension d'Early** (SPICE : VAF pour NPN, VAR pour PNP).

#### 2. Comment le lire sur une courbe $I_{CIC} - V_{CE}$

- Trace les caractéristiques de sortie du BJT (courbe  $I_{CIC}$  en fonction de  $V_{CE}$  pour un  $I_{BIB}$  constant).
- Ces droites, prolongées vers l'axe des abscisses (vers  $V_{CE} < 0$ ), convergent en un point.
- La valeur de l'abscisse à ce point de convergence est  $-V_A$ .

🔗 Exemple : si les droites se coupent en  $-100$  V, alors  $V_A \approx 100$  V.

#### 3. Formule pratique

On peut relier  $V_A$  à la pente de la courbe :

$$r_o = \frac{\partial I_C}{\partial V_{CE}} \approx \frac{I_C}{V_A} \Rightarrow V_A \approx \frac{I_C}{\frac{\partial I_C}{\partial V_{CE}}}$$

où

- $r_o$  est la résistance de sortie du transistor en régime actif.
- Autrement dit :

$$r_o \approx \frac{V_A}{I_C}$$

#### 4. Exemple numérique

Supposons un NPN avec :

- Pour  $I_B = 20 \mu\text{A}$ , on a  $I_C \approx 2 \text{ mA}$  à  $V_{CE} = 5 \text{ V}$ .
- En augmentant  $V_{CE}$  de  $5 \text{ V} \rightarrow 10 \text{ V}$ ,  $I_C$  passe de  $2.00 \text{ mA}$  à  $2.04 \text{ mA}$ .

Alors :

$$\Delta I_C / \Delta V_{CE} = 0.04 \text{ mA} / 5 \text{ V} = 8 \mu\text{A/V}$$
$$V_A = I_C \partial I_C / \partial V_{CE} = 2.00 \text{ mA} \cdot 8 \mu\text{A/V} = 16 \text{ mV}$$
$$V_A = \partial V_{CE} / \partial I_C = 8 \mu\text{A/V} / 0.04 \text{ mA} = 200 \text{ V}$$

Donc le **VAF**  $\approx 250 \text{ V}$ .

---

#### 5. Dans SPICE

- Dans un modèle .model de BJT :

```
.model QX NPN(IS=1e-14 BF=200 VAF=100)
```

→ ici  $V_{AF}=100$  veut dire qu'on simule un transistor dont les courbes de sortie se coupent à  $-100 \text{ V}$ .

- Plus  $V_{AF}$  est grand, plus le transistor se comporte « comme idéal » (sortie plate).
- Si  $V_{AF} = \infty$ , il n'y a aucune modulation de courant → transistor parfait.

---

Donc :

- **VAF se déduit de la pente des courbes  $I_C(V_{CE})$  en actif.**
- Formule simple :

$$V_A = I_C \partial I_C / \partial V_{CE}$$

à un point donné.

---

Veux-tu que je te montre comment **extraire VAF directement à partir d'un datasheet** (par ex. BC547) avec les courbes fournies ?

**Vous avez dit :**

**oui**

**ChatGPT a dit :**

Réflexion durant 44s

Très bonne question — voici **comment extraire/estimer VAF (tension d'Early)** précisément à partir d'un jeu de courbes ou de données, avec méthodes manuelles, Excel et LTspice + un exemple numérique chiffré étape-par-étape.

---

#### Rappel rapide

La définition utilisée ici :

$$I_C(V_{CE}) \approx I_{C0} (1 + V_{CE}/V_A) \approx I_{C0} \left(1 + \frac{V_{CE}}{V_A}\right)$$

d'où, pour un point donné,

$$\partial I_C / \partial V_{CE} \approx I_{C0} / V_A \Rightarrow V_A \approx I_C \partial I_C / \partial V_{CE} = \frac{I_C}{\partial I_C / \partial V_{CE}} \approx \frac{I_C}{\partial I_C / \partial V_{CE}}$$

Autrement dit :  $VAF = I_C / \text{pente}$  (pente =  $\Delta I_C / \Delta V_{CE}$ ) mesurée sur la même courbe de sortie (même courant de base / même courant de polarisation).

---

### Méthode manuelle (à partir d'une courbe $I_{C1\_CIC}$ vs $V_{CEV_{CE}}V_{CE}$ )

1. Choisis une **même courbe de sortie** (même courant de base ou même  $I_{B1\_BIB}$  / même  $I_{C1\_CIC}$  nominal).
2. Lis deux points proches sur la portion quasi-linéaire (éviter la zone de saturation et la très basse tension) :  $(V_{CE1}, I_{C1})$  et  $(V_{CE2}, I_{C2})$ .

3. Calcule la pente :

$$\text{slope} = \frac{\Delta I_C}{\Delta V_{CE}} = \frac{I_{C2} - I_{C1}}{V_{CE2} - V_{CE1}} \quad \text{slope} = \frac{\Delta V_{CE}}{\Delta I_C} = \frac{V_{CE2} - V_{CE1}}{I_{C2} - I_{C1}} \quad \text{slope} = \frac{V_{CE2} - V_{CE1}}{I_{C2} - I_{C1}} \quad \text{slope} = \frac{V_{CE2} - V_{CE1}}{I_{C2} - I_{C1}}$$

4. Choisis  $I_{C1\_CIC}$  représentatif (par ex. la valeur moyenne  $(I_{C1} + I_{C2})/2$  ou  $I_{C1}$  à  $V_{CE1}$ ) et calcule :

$$V_A \approx I_C \cdot \text{slope} \quad V_A \approx \frac{I_C}{\text{slope}} \quad V_A \approx \text{slope} \cdot I_C$$

(unités :  $A / (A/V) = V$ )

---

### Exemple numérique détaillé (chiffre par chiffre)

Supposons relevés sur une même courbe :

- Point 1 :  $V_{CE1} = 2.0 \text{ V}$ ,  $I_{C1} = 2.000 \text{ mA} = 2.000 \times 10^{-3} \text{ A}$
- Point 2 :  $V_{CE2} = 10.0 \text{ V}$ ,  $I_{C2} = 2.040 \text{ mA} = 2.040 \times 10^{-3} \text{ A}$

Calculs :

1.  $\Delta I_C = 2.040 \times 10^{-3} - 2.000 \times 10^{-3} = 0.040 \times 10^{-3} = 4.0 \times 10^{-5} \text{ A}$
2.  $\Delta V_{CE} = 10 - 2 = 8 \text{ V}$
3.  $\text{slope} = \frac{4.0 \times 10^{-5}}{8} = 5.0 \times 10^{-6} \text{ A/V}$
4. Moyenne de  $I_{C1\_CIC}$  (représentative) :  $I_{C\text{avg}} = (2.000 + 2.040)/2 \text{ mA} = 2.020 \text{ mA} = 2.020 \times 10^{-3} \text{ A}$
5.  $V_A \approx 2.020 \times 10^{-3} / (5.0 \times 10^{-6}) = 404 \text{ V}$

Vérif rapide :  $r_o = 1/\text{slope} = 1/(5.0 \times 10^{-6}) = 200,000 \Omega = 200 \text{ k}\Omega$ . Et  $V_A = r_o \cdot I_{C\text{avg}} = 200,000 \cdot 2.02 \times 10^{-3} \approx 404 \text{ V}$  — cohérent.

---

### Méthode Excel / Google Sheets

Supposons que tu as une colonne  $V_{CE}$  en A2:A### et  $I_C$  en B2:B### (en ampères).

- **Pente entre deux lignes** (ligne 2 et 3) :  
 $= (B3 - B2) / (A3 - A2)$  → slope (A/V)
- **VAF à partir de ces deux lignes** (avec  $I_C$  moyen) :  
 $= ((B2 + B3) / 2) / ((B3 - B2) / (A3 - A2))$
- **Si tu as plusieurs points** : utiliser la régression linéaire (SLOPE) pour obtenir slope moyen :

- $\text{slope} = \text{SLOPE}(B2:B50, A2:A50)$  (y-range = Ic, x-range = VCE)
- $\text{Icmoy} = \text{AVERAGE}(B2:B50)$  (en A)
- $\text{VAF} = \text{Icmoy} / \text{slope}$

(Remarque : attention aux unités — Ic en ampères.)

---

### Extraction dans LTspice (procédure recommandée)

1. Place un .dc sweep sur VCE (par ex. .dc VCE 0 20 0.1) en gardant la même polarisation de base (fixe IB ou VB).
2. Exécute la simulation. Dans la fenêtre trace, affiche la courbe I(Q1) (ou Ic(Q1)) en fonction de V(collector) (ou simplement Ic vs VCE en utilisant le nœud).
3. **Exporter** les données : File → Export → plot data (sauvegarde en CSV).
4. Ouvre le CSV dans Excel / Python et applique la méthode ci-dessus (slope via deux points ou régression).  
— Pourquoi exporter ? Parce que LTspice ne possède pas de commande simple et universelle pour calculer numériquement la dérivée locale d'une courbe DC directement en une ligne (on peut faire des .measure compliqués mais l'export+calc est plus simple et robuste).

Alternative (pour utilisateurs avancés) : on peut écrire une directive .meas pour calculer Ic à 2 VCE différentes et faire la pente manuellement via trois .meas, mais l'export reste plus pratique.

---

### Précisions et bonnes pratiques

- **Choisir la zone** : prends des points dans la zone « actif quasi-linéaire », pas en saturation (VCE trop petit) ni en avalanche (VCE trop grand).
  - **Dépendance en courant** : VAF dépend de ICL\_CIC ; donc mesure autour du courant d'utilisation réel.
  - **Estimation par régression** (plus robuste) : si tu as beaucoup de points (p. ex. 2–10 VCE), fais une régression linéaire pour obtenir une pente moyenne et calcule VAF à partir du courant moyen.
  - **Température** : VAF varie légèrement avec température ; datasheets donnent souvent des courbes à 25 °C.
  - **Unités** : Ic en A, slope en A/V → VAF en V.
- 

Si tu veux, je peux :

- **prendre une image de courbe** (datasheet que tu fournis) et te montrer pas-à-pas l'extraction numérique (je tracerai les points et calculerai VAF) ; ou
- **fournir un petit script Python / feuille Excel** qui prend un CSV issu de LTspice et calcule VAF automatiquement (avec régression et sortie r\_o).

Laquelle préfères-tu ?

**Vous avez dit :**

**est-ce que le calcul de VAR est le même que VAF?**

**ChatGPT a dit :**

Oui 🇧

Le principe est exactement le même — la seule différence est que :

- **VAF** (Forward Early Voltage) → utilisé pour modéliser la modulation de courant **en mode direct** (transistor utilisé normalement : jonction BE polarisée directe, jonction BC polarisée inverse).

- **VAR** (Reverse Early Voltage) → utilisé pour modéliser le comportement **en mode inverse** (transistor « inversé » : jonction BC polarisée directe, jonction BE polarisée inverse, ce qui est rare en pratique sauf dans certains montages spéciaux).

### 1. Formule générale

Dans les deux cas, on a la même équation :

$$I \approx I_0(1 + V/V_A) \approx I_0 \left(1 + \frac{V}{V_A}\right)$$

où  $V$  est la tension de la jonction « collectrice » du mode considéré.

Donc que ce soit en mode direct (VAF) ou inverse (VAR),

$$V_A = I \frac{\partial I}{\partial V} \quad V_A = \frac{I}{\partial I / \partial V} \quad V_A = \frac{\partial I}{\partial V} I$$

### 2. En pratique

- **VAF** est souvent donné ou estimé à partir des courbes  $I_{C1} - I_{C2}$  vs  $V_{CE1} - V_{CE2}$  habituelles des datasheets → facile à mesurer.
- **VAR** est rarement spécifié, car les transistors sont presque toujours utilisés en mode direct.  
→ Dans beaucoup de modèles SPICE, on trouve simplement une valeur générique (ex : VAR=20 V ou 30 V), ou bien VAR = VAF pour simplifier.

### 3. Exemple typique (dans SPICE)

```
.model QX NPN(IS=1e-14 BF=200 VAF=100 VAR=20)
```

- Ici, VAF = 100 V (effet Early en mode direct).
- VAR = 20 V (effet Early en mode inverse, bien plus faible → transistor « moins bon » quand on l'utilise inversé).

Donc oui, **le calcul est le même**, mais :

- VAF s'applique en mode direct → utile, souvent mesuré sur datasheet.
- VAR s'applique en mode inverse → rarement mesuré, souvent choisi par approximation dans les modèles.

Veux-tu que je t'explique **comment SPICE combine VAF et VAR** dans le modèle Gummel-Poon pour gérer la modulation de courant quand le transistor passe de direct à inverse ?

**Vous avez dit :**

**Pourtant je trouve aussi VAR pour les PNP comme dans ce modèle : .model KT646A NPN(IS=17.03f BF=200 BR=4.6 ISE=153.3f ISC=1.35p IKF=0.4095 IKR=0.21 NE=1.368 NC=2 VAF=123 VAR=75 RC=0.35 RB=15 TF=1.8n TR=50n XTF=1 VTF=60 ITF=0.65 CJE=34.4p VJE=0.69 MJE=0.33 CJC=21.24p VJC=0.69 MJC=0.33 FC=0.5 EG=1.11 XTB=1.5 XTI=3 Vceo=60 Icrating=1 mfg=USSR)**

**ChatGPT a dit :**

Très bonne remarque 🙌 Tu as mis le doigt sur un point qui prête souvent à confusion.

### 1. Pourquoi VAR dans un modèle NPN ou PNP « normal » ?

Même si un transistor est utilisé en mode **direct** (jonction BE directe, BC inverse), le modèle SPICE **Gummel-Poon** est symétrique : il inclut aussi le **mode inverse**, où le transistor fonctionne « à l'envers » (la base reste la base, mais le rôle collecteur/émetteur s'inverse au niveau du transport des charges).

- **VAF** = *Forward Early Voltage* → modulation en **mode direct**.
- **VAR** = *Reverse Early Voltage* → modulation en **mode inverse**.

☞ Donc oui, tu peux trouver **VAR dans un modèle NPN ou PNP**, même si, en pratique, ce paramètre a peu d'influence tant que tu n'utilises pas le transistor à l'envers.

Mais le modèle reste complet pour les simulateurs (utile dans certains circuits où un transistor peut momentanément fonctionner en inverse).

---

## 2. Exemple de ton modèle KT646A

.model KT646A NPN( ... VAF=123 VAR=75 ... )

- Ici, **VAF=123 V** → les caractéristiques de sortie en régime direct convergent à  $-123$  V environ.
  - **VAR=75 V** → si tu polarisais le transistor en **mode inverse** (rare, mais possible), les caractéristiques convergeraient à  $-75$  V.
- 

## 3. Différence entre NPN et PNP

- Que le modèle soit NPN ou PNP, les deux paramètres **VAF** et **VAR** existent.
  - Ce n'est pas une question de type (NPN/PNP), mais de **mode d'utilisation** (direct ou inverse).
  - Dans un modèle PNP, tu trouveras aussi VAF et VAR, avec souvent des valeurs différentes car le transistor « inversé » est encore moins performant.
- 

## 4. Résumé

- **VAF** : Early voltage en mode direct (le cas usuel).
  - **VAR** : Early voltage en mode inverse (rare, mais inclus pour un modèle complet).
  - Peu importe que ce soit **NPN ou PNP**, SPICE définit toujours les deux pour couvrir tous les cas.
  - Dans la plupart des simulations « normales », c'est surtout **VAF** qui compte.
- 

☞ Veux-tu que je t'illustre ça avec une petite **simulation SPICE montrant la différence entre VAF et VAR** (un NPN polarisé en mode normal vs en mode inversé) ?

**Vous avez dit :**

**non j'ai compris**

**ChatGPT a dit :**

Parfait ☞ ravi que ce soit clair maintenant !

Donc à retenir : **VAF et VAR existent toujours dans le modèle SPICE**, quel que soit le type (NPN/PNP), mais on n'utilise presque que **VAF** en pratique, puisque le mode inverse est rare.

Si un jour tu veux, je peux te montrer comment lire directement ces valeurs sur les courbes du datasheet (pour comparer avec celles des modèles).

Tu veux qu'on passe à un autre paramètre du modèle SPICE (par ex. **IKF**, **TF**, etc.) ou ça va pour l'instant ?

**je serais curieux de voir un modèle pour un vieux transistor au germanium comme les OC44 ou OC71**

**ChatGPT a dit :**

Réflexion durant 19s

Top — voici des **modèles SPICE pratiques** pour deux transistors germanium anciens très courants : **OC44** (RF PNP) et **OC71** (audio PNP). Je fournis aussi les explications sur les choix de paramètres et comment les ajuster si tu as des mesures/fiches techniques précises.

Je précise d'emblée que ces modèles sont **approximatifs** (les vieux germanium ont de fortes variations, fuites et hFE très dispersés). J'ai choisi des valeurs raisonnables basées sur les datasheets historiques :  $V_{ce0}/V_{cb}$  faibles ( $\approx 12-20$  V),  $I_c$  max  $\approx 10-50$  mA, puissance dissipable basse ( $\sim 0,08-0,125$  W), ft très faible comparé aux silicium. [All Transistors+2](#)  
[All Transistors+2](#)

---

## 1) Modèle SPICE pour OC44 (PNP — transistor germanium, utilisé RF)

Données clés trouvées : PNP germanium,  $V_{ce0} \approx 12$  V,  $P_c \approx 0.083$  W,  $I_c$  max  $\approx 10$  mA, ft  $\approx 8$  MHz (ordre de grandeur). [All Transistors+1](#)

\* Modèle approximatif OC44 (PNP germanium)

\* Usage: Q1 C B E OC44

```
.model OC44 PNP (  
+ IS=5e-8 ; courant de saturation élevé (germanium)  
+ BF=60 ; hFE typique (peut varier beaucoup)  
+ NF=1.2  
+ VAF=35 ; Early voltage (faible à modéré)  
+ IKF=0.05  
+ ISE=5e-9  
+ BR=10  
+ NR=2  
+ RB=60 ; résistance base interne ( $\Omega$ ) - vintage, pas très faible  
+ RC=2 ; résistance série collector ( $\Omega$ )  
+ RE=2 ; résistance série emitter ( $\Omega$ )  
+ CJE=12p ; Cbe ~ datasheet ~12 pF  
+ CJC=5p ; Cbc  
+ TF=1.2e-6 ; transit time -> calé pour ft  $\approx 8$  MHz ( $TF \approx BF/(2\pi \cdot ft)$ )  
+ XTF=0.5  
+ TR=200n  
+ ISC=1e-9  
)
```

Notes :

- IS élevé (5e-8 A) reflète les fuites plus fortes du germanium.
  - TF calculé pour donner un ft  $\approx 8$  MHz avec  $BF \approx 60$  :  $TF \approx BF / (2\pi \cdot ft)$
  - Ajuste BF, IS, CJE si tu disposes d'un relevé ou d'un point datasheet. [All Transistors](#)
- 

## 2) Modèle SPICE pour OC71 (PNP — transistor germanium audio)

Données clé : PNP germanium,  $P_c \approx 0.125$  W,  $V_{ce0} \approx 20$  V (typ.),  $I_c$  max  $\approx 10-50$  mA, ft très faible (quelques centaines de kHz à  $\sim 1$  MHz selon source). hFE typique  $\approx 30$  (très variable). [All Transistors+1](#)

\* Modèle approximatif OC71 (PNP germanium)

\* Usage: Q1 C B E OC71

```
.model OC71 PNP (  
+ IS=1e-7 ; courant de saturation plus élevé (germanium, audio device)  
+ BF=30 ; hFE typique (peut être 10..100 selon pièce)  
+ NF=1.3  
+ VAF=30 ; Early voltage faible  
+ IKF=0.05  
+ ISE=1e-8  
+ BR=8  
+ NR=2  
+ RB=80  
+ RC=3  
+ RE=3  
+ CJE=30p ; datasheets indiquent Cbe relativement élevé ( $\approx 30$  pF)  
+ CJC=10p  
+ TF=5e-6 ; TF pour ft  $\approx 1$  MHz avec  $BF \approx 30$  (ou plus lent si ft  $\approx 0.3$  MHz)
```

+ XTF=0.6  
+ TR=300n  
+ ISC=5e-9  
)

Notes :

- J'ai mis  $IS=1e-7$  et  $CJE=30p$  pour refléter les fuites et capacités plus élevées observées sur OC71.
  - Si ton lot d'OC71 a  $f_t \approx 0.3$  MHz (certaines sources), augmente  $TF \sim 1.6e-5$  pour coller à cela. [All Transistors+1](#)
- 

### 3) Exemples d'utilisation (netlist simple)

```
* Test DC pour OC71
VCC 1 0 DC -12
Rload 1 Col 10k
Q1 Col Base 0 OC71
Vb Base 0 DC -0.25 ; Vbe pour germanium typiquement ~ -0.2..-0.3 V (PNP)
.dc Vb -0.5 0 0.01
.print dc I(Q1) V(Col,0) V(Base,0)
.end
```

(Remarque : pour PNP on met VCC négatif ou polarise la base négativement selon le schéma.)

---

### 4) Pourquoi ces valeurs et comment les affiner

- **IS** : beaucoup plus grand que pour silicium ; ajuste-le à partir d'un point ( $I_c, V_{be}$ ) du datasheet avec la méthode que je t'ai montrée précédemment ( $IS = I_c / \exp(V_{be}/(n \cdot V_t))$ .)
  - **BF (hFE)** : très dispersé d'un exemplaire à l'autre ; il vaut mieux mesurer ton transistor réel et remplacer BF par la valeur mesurée.
  - **TF / ft** : calcule  $TF = BF/(2\pi \cdot ft)$  pour coller à la spécification  $f_t$  donnée par la fiche.
  - **VAF** : les germanium anciens ont souvent un Early faible → met une VAF modérée (20–50 V).
  - **CJE/CJC** : les mesures/datasheets indiquent souvent des capacités assez élevées (OC71  $C_{be} \approx 30$  pF). [All Transistors+1](#)
- 

### 5) Attention — thermique & fuites

- Les germanium ont des **courants de fuite collector-base (ICBO/ICBO)** très sensibles à la température ; en simulation, tu verras des courants de polarité et d'ordre de grandeur instables si tu gardes IS élevé sans tenir compte de la température. Ici ISC et ISE tentent d'approximer ces effets. [Little Diode](#)
  - $V_{be}$  pour germanium  $\approx 0.15\text{--}0.3$  V (beaucoup plus bas que le 0.6–0.7 V des silicium). Utilise donc des sources de polarisation en conséquence.
- 

### 6) Validation / calibration rapide

1. Prends un point du datasheet : ex.  $I_C = 1$  mA à  $V_{BE} = 0.2\text{--}0.25$  V (les datas donnent souvent  $I_c @ V_{be}$ ). Calcule IS (méthode déjà fournie).
  2. Balaye AC/DC : vérifie hFE en bas régime,  $f_t$  (AC small-signal) et compare la courbe  $I_c(V_{CE})$  pour estimer VAF.
  3. Ajuste RB, RC, CJE pour corriger gain/freq et comportement transitoire.
-

### Références utilisées (pour les chiffres cités)

- OC44 datasheet / fiches techniques — paramètres : PNP,  $P_c \approx 0.083$  W,  $V_{ce0} \approx 12$  V,  $f_t \approx 8$  MHz. [All Transistors+1](#)
  - OC71 datasheet / fiches et vendeurs — PNP,  $P_c \approx 0.125$  W,  $V_{ce0} \approx 20$  V,  $h_{FE}$  typ  $\approx 30$ ,  $f_t$  très faible (0.3–1 MHz selon source). [All Transistors+1](#)
- 

Si tu veux, je peux :

- **générer un fichier .subckt LTspice** (ou .asy) prêt à ouvrir ;
- **affiner IS/BF/TF** si tu me donnes un point précis du datasheet ( $I_c, V_{be}$ ) ou une mesure réelle d'un exemplaire ;
- ou **simuler un petit amplificateur** avec OC71/OC44 pour vérifier la bande passante et le gain (je peux te fournir la netlist de test).

Tu préfères que je prépare le .asc LTspice pour l'un des deux (OC44 ou OC71) avec tracés de  $I_c$ ,  $V_{be}$  et  $P_{diss}$  ?