以下に、schottky.py プログラムのマニュアルを生成します。

# schottky.py マニュアル

## プログラムの目的

schottky.py は、ショットキー障壁ダイオードの電流-電圧 (I-V) 特性をシミュレーションし、その結果をプロットするPythonプログラムです。熱電子放出、拡散、ショットキー障壁低下、電子トンネリング、熱電子電界放出 (TFE) といった複数の物理モデルをサポートしています。これにより、ユーザーはデバイスの温度、材料定数、幾何学的パラメータなどを変更して、それぞれのモデルがI-V特性に与える影響を比較検討できます。

## 原理

schottky.py は、以下の主要な物理定数を使用してショットキーダイオードの動作をモデル化します。

* **ボルツマン定数** (
* )
* **素電荷** (
* )
* **真空の誘電率** (
* )
* **電子質量** (
* )
* **ディラック定数** (
* )

### 1. 基本パラメータの計算 (calculate\_params関数)

プログラムはまず、以下のデバイスおよび材料に関する基本パラメータを計算します。

* **リチャードソン定数 ()** 半導体中の電子の有効質量比 (
* ) に基づいて計算される、修正されたリチャードソン定数です。
* **伝導帯の有効状態密度 ()** 半導体内の伝導帯における有効状態密度。温度 (
* ) および電子の有効質量比 (
* ) に依存します。
* **半導体のフェルミ準位 ()** 半導体側の伝導帯端 (
* 、電子親和力に相当)、ドナー濃度 (
* )、および有効状態密度 (
* ) から決定されます。
* **ショットキー障壁高さ ()** 金属の仕事関数 (
* ) と半導体の電子親和力 (
* ) から理想的なショットキー障壁高さが計算されます。
* **内蔵電位 ()** 金属と半導体間の熱平衡状態におけるフェルミ準位の差から内蔵電位が計算されます。
* **理想飽和電流密度 ()** リチャードソン・ダッシュマンの式に基づき、温度 (
* ) とショットキー障壁高さ (
* ) から理想的な飽和電流密度が計算されます。
* **半導体の直列抵抗 ()** 半導体の電気伝導度 (
* ) と厚さ (
* )、デバイス面積 (
* ) から直列抵抗が計算されます。
* ここで
* は電子移動度です。

### 2. ショットキー電流の計算 (calculate\_schottky\_current関数)

印加電圧 (

) に対するダイオード電圧 (

) および電流 (

) は、以下の方程式をニュートン法に似た反復計算で解くことで求められます。

ここで

は選択された物理モデルに基づいたダイオード電流です。

各物理モデルにおける電流の式は以下の通りです。

* **Simple モデル (理想ダイオードモデル)** 最も基本的なモデルで、熱電子放出のみを考慮します。理想因子 (
* ) はプログラム中で1.0に固定されています。
* **Diffusion モデル (拡散モデル)** 半導体中の少数キャリア拡散が電流を支配すると仮定するモデルです。
* ここで
* は半導体の誘電率です。
* **Schottky モデル (ショットキー障壁低下モデル)** 高電界下でのイメージ力によるショットキー障壁低下を考慮したモデルです。障壁低下量 (
* ) の近似として、以下の項が使用されます。
* **Tunneling モデル (簡易トンネルモデル)** 電子のトンネリング効果を簡易的に考慮したモデルです。逆方向バイアスや高ドーピング濃度で顕著になります。
* ここで
* はトンネリング理想因子であり、
* は室温 (
* ) での熱電圧 (
* ) の近似値です。
* **TFE モデル (熱電子電界放出モデル)** 熱電子放出 (Thermionic Emission) と電界放出 (Field Emission) の両方を考慮するモデルです。高ドーピングや低温で重要になります。
* ここで
* は電子の有効質量です。
* **順方向 ():**
* **逆方向 ():** 逆方向バイアスでは、電界による障壁の薄層化を考慮し、トンネル透過率が増加します。

## 使い方 (usage)

プログラムはコマンドライン引数を受け付けて動作します。schottky.py を実行するには、以下のコマンド形式を使用します。

python schottky.py [オプション]

利用可能なオプションは以下の通りです。

| オプション | 型 | デフォルト値 | 説明 |
| --- | --- | --- | --- |
| --model | string | Simple | 使用する物理モデルを選択します。Simple, Diffusion, Schottky, Tunneling, TFE から選択。 |
| --temp | float | 300.0 | 温度 [K] |
| --area | float | 0.01 | デバイス面積 [cm^2] |
| --efm | float | 4.4 | 金属のフェルミ準位 (仕事関数) [eV] |
| --ecn | float | 4.05 | 半導体の電子親和力 [eV] |
| --ndn | float | 1.0e16 | 半導体のドナー濃度 [cm^-3] |
| --mun | float | 1500.0 | 半導体の電子移動度 [cm^2/Vs] |
| --eps\_r | float | 11.9 | 半導体の比誘電率 |
| --dn | float | 1000.0 | 半導体の厚さ [nm] |
| --men | float | 0.19 | 半導体の電子有効質量比 (  ) |
| --n\_tunnel | float | 2.0 | Tunneling モデルにおけるトンネリング理想因子 |
| --v0 | float | -2.0 | 印加電圧の開始点 [V] |
| --v1 | float | 1.0 | 印加電圧の終了点 [V] |
| --step | float | 0.02 | 電圧ステップ [V] |

## 使用例

### 1. デフォルト設定でシミュレーションを実行

最も基本的な使用法です。Simple モデルが選択され、デフォルトのパラメータが使用されます。

python schottky.py

このコマンドを実行すると、デフォルトパラメータでのショットキーダイオードのI-V特性がプロットされ、コンソールには計算された基本パラメータが表示されます。

### 2. TFE モデルと異なるドナー濃度でシミュレーションを実行

半導体ドナー濃度を高く設定し、TFE モデルを用いてI-V特性をシミュレーションします。

python schottky.py --model TFE --ndn 1.0e18 --v0 -5.0

この例では、TFE モデルを選択し、ドナー濃度を 1.0e18 cm^-3 に変更しています。また、逆方向電圧範囲を -5.0 V まで広げています。高濃度ドナーでは電界放出効果が顕著になるため、TFE モデルがより適切です。

### 3. 特定の温度とデバイス面積でシミュレーションを実行

低温での特性や、デバイスサイズの影響を調べる場合に使用します。

python schottky.py --temp 77.0 --area 0.001 --model Diffusion

この例では、温度を 77.0 K に、デバイス面積を 0.001 cm^2 に変更し、Diffusion モデルを使用しています。