EE341 Fall 2019 HW 1
Lewis Collum
Updated: September 24, 2019

Document Org source code: github.com/LewisCollum/microelectronics

---

3.1
Solution:

```python
import api.homework_1 as api
unit = api.unit

t = {
    "-55C": unit.Quantity(-55, unit.celsius).to('kelvin'),
    "0C": unit.Quantity(0, unit.celsius).to('kelvin'),
    "20C": unit.Quantity(20, unit.celsius).to('kelvin'),
    "75C": unit.Quantity(75, unit.celsius).to('kelvin'),
    "125C": unit.Quantity(125, unit.celsius).to('kelvin')
}

for key, kelvin in t.items():
    intrinsicConcentration = api.Silicon.intrinsicConcentrationFromKelvin(kelvin)
    ionizationRatio = intrinsicConcentration / api.Silicon.densityOfAtoms

    print(f"Case {key} ({kelvin:.5}): {intrinsicConcentration:.2E}")
    print(f"Fraction of atoms ionized: ni/N = {ionizationRatio:.2E}\n")
```

Answer:

```
Case -55C (218.15 kelvin): 2.72E+06 / centimeter ** 3
Fraction of atoms ionized: ni/N = 5.43E-17 / centimeter ** 3

Case 0C (273.15 kelvin): 1.53E+09 / centimeter ** 3
Fraction of atoms ionized: ni/N = 3.07E-14 / centimeter ** 3

Case 20C (293.15 kelvin): 8.64E+09 / centimeter ** 3
Fraction of atoms ionized: ni/N = 1.73E-13 / centimeter ** 3

Case 75C (348.15 kelvin): 3.71E+11 / centimeter ** 3
Fraction of atoms ionized: ni/N = 7.42E-12 / centimeter ** 3

Case 125C (398.15 kelvin): 4.73E+12 / centimeter ** 3
Fraction of atoms ionized: ni/N = 9.46E-11 / centimeter ** 3
```

Thermal excitation of electrons in the valence band is a consequence of raising the temperature. As such, the number of possible conducting states increase, as does the probability that those states are occupied by electrons; this implies a greater intrinsic density.

3.3
Solution:

```python
import api.homework_1 as api
unit = api.unit

acceptorDopantConcentration = 5e18 / unit.centimeters**3
systemKelvin = 300 * unit.kelvin
intrinsicConcentration = api.Silicon.intrinsicConcentrationFromKelvin(300*unit.kelvin)
electronConcentration = intrinsicConcentration**2/acceptorDopantConcentration

print(f"P-type Electron Concentration: {electronConcentration:.4}")
print(f"P-type Hole Concentration: {acceptorDopantConcentration:.4}")
```

Answer:

```
P-type Electron Concentration: 44.04 / centimeter ** 3
P-type Hole Concentration: 5e+18 / centimeter ** 3
```

3.5

Solution:

```python
import api.homework_1 as api
from api import unit

donorDopantConcentration = 10e17 / unit.centimeters**3
print(f"N-type Electron Concentration: {donorDopantConcentration:.4} @ 27C and 125C")

t = {
    '27C': unit.Quantity(27, unit.celsius).to('kelvin'),
    '125C': unit.Quantity(125, unit.celsius).to('kelvin')
}

for label, kelvin in t.items():
    intrinsicConcentration = api.Silicon.intrinsicConcentrationFromKelvin(kelvin)
    holeConcentration = intrinsicConcentration**2/donorDopantConcentration
    print(f"N-type Hole Concentration: {holeConcentration:.4} @ {label}")
```

Answer:

```
N-type Electron Concentration: 1e+18 / centimeter ** 3 @ 27C and 125C
N-type Hole Concentration: 225.4 / centimeter ** 3 @ 27C
N-type Hole Concentration: 2.238e+07 / centimeter ** 3 @ 125C
```

3.21

Solution:

```python
import api.homework_1 as api
unit = api.unit

carrierConcentration = {
    '300K': api.Silicon.intrinsicConcentrationFromKelvin(300 * unit.kelvin),
    '305K': api.Silicon.intrinsicConcentrationFromKelvin(305 * unit.kelvin)
}

factor = carrierConcentration['305K']**2 / carrierConcentration['300K']**2
print(f"Factor: {factor.magnitude:.3}")
```

Answer:

```
Factor: 2.14
```

4.18

$$v = V_T \ln \frac{i}{I_S}$$
$$v = V_T \ln \frac{10000 I_S}{I_S}$$
$$v = V_T \ln 10000 \text{ assume } V_T = 25\text{mV}$$
$$\boxed{v = 0.025 \cdot \ln 10000 = 0.23\text{V}}$$

$$
\begin{aligned}
i &= I_S \cdot e^{v/V_T} \\
i &= I_S \cdot e^{0.7/0.025} \\
&\boxed{i = 1.45 \times 10^{12} I_S}
\end{aligned}
$$

4.19

$$
\begin{aligned}
\frac{I_2}{I_1} &= e^{(V_2 - V_1)/V_T} \\
I_2 &= e^{(0.5-0.7)/0.025)} \cdot 1\text{mA} \\
&\boxed{I_2 = 335\mu\text{A}}
\end{aligned}
$$

4.23

The voltage across each diode, V_d, is $V_o/3 = 0.67$.

$$
\begin{aligned}
I &= I_S \cdot e^{V_d/V_T} \\
I &= 10^{-14}\text{A} \cdot e^{0.67/0.025} \\
&\boxed{I = 4.4\text{mA}}
\end{aligned}
$$

If 1mA is drawn, $I = 4.4\text{mA} - 1\text{mA} = 3.4\text{mA}$

$$
\begin{aligned}
V_o &= V_T \cdot \ln \frac{I}{I_S} \cdot 3 \\
V_o &= 0.025 \cdot \ln \frac{0.0034}{10^{-14}} \cdot 3 \\
V_o &= 1.991\text{V}
\end{aligned}
$$

Output voltage changed by $2 - 1.991 = \boxed{9.0\text{mV}}$

4.25

$$
\begin{aligned}
I_{D1} &= I_{S1} \cdot e^{V_D/V_T} \\
I_{D2} &= I_{S2} \cdot e^{V_D/V_T} \\
V_D &= V_T \ln \frac{I_{D1}}{I_{S1}}
\end{aligned}
$$

4.28

$$
\begin{aligned}
I &= I_1 + I_2 \text{ by KCL} \\
I_1 &= V/R \text{ by Ohm's law} \\
V &= V_2 - V_1 \text{ by KVL} \\
e^{(V_2-V_1)/V_T} &= e^{V/V_T} = \frac{I_2}{I_1} = \frac{I - I_1}{I_1} = \frac{I}{I_1} - 1 \\
\frac{I \cdot R}{V} - 1 &= e^{V/V_T} \to R = \frac{V}{I} \cdot (e^{V/V_T} + 1) \\
&\boxed{R = \frac{0.05\text{V}}{0.01\text{A}} (e^{0.05\text{V}/0.025\text{V}} + 1) = 42\Omega}
\end{aligned}
$$

4.29

According to the text book:

```
At a given constant diode current, the voltage drop across the diode
decreases by approximately 2 mV for every 1°C increase in temperature.
```

Case $T = -20°C$:

$$\begin{aligned}\Delta T &= -40°\text{C}\\ V &= 690\text{mV} + 2\text{mV} \cdot 40°\text{C}\\ &\boxed{= 770\text{mV}}\end{aligned}$$

Case $T = +85°C$:

$$\begin{aligned}\Delta T &= +65°\text{C}\\ V &= 690\text{mV} - 2\text{mV} \cdot 65°\text{C}\\ &\boxed{= 560\text{mV}}\end{aligned}$$

Appendix: Code

```python
from __future__ import annotations
import math
import pint

unit = pint.UnitRegistry()

def densityOfStates(materialConstant: float, kelvinOfSystem: float) -> float:
    return materialConstant/unit.kelvin**(3/2)/unit.centimeters**3 * kelvinOfSystem**(3/2)

class Boltzmann:
    @classmethod
    def probability(cls, stateEnergy: float, kelvinOfSystem: float) -> float:
        return math.exp(-stateEnergy*unit.eV/(kelvinOfSystem * unit.boltzmann_constant))

class Silicon:
    densityOfStatesMaterialConstant = 7.3e15
    densityOfAtoms = 5e22
    holeMobility = 480 * unit.centimeters**2 / (unit.volts*unit.seconds)
    electronMobility = 1350 * unit.centimeters**2 / (unit.volts*unit.seconds)

    @classmethod
    def intrinsicConcentrationFromKelvin(cls, kelvin: float) -> float:
        siliconDensityOfStates = densityOfStates(
            materialConstant=Silicon.densityOfStatesMaterialConstant,
            kelvinOfSystem=kelvin)
        distribution = Boltzmann.probability(
            stateEnergy=1.12,
            kelvinOfSystem=kelvin)**(1/2)
        return siliconDensityOfStates * distribution
```