Homework 8

Jacob Taylor Cassady

Semiconductor Development Fundamentals

March 20, 2020

NOTE: I TRIED TO COPY AND PASTE CODE IN AS IT WAS RELEVANT. I ALSO COPIED THE ENTIRE SOURCE CODE AT THE VERY END.

1 CONSIDER A GERMANIUM P-N JUNCTION WITH NA = 1015 CM-3 AND ND = 1015 CM-3 . THE MINORITY CARRIER LIFETIME ON THE P-SIDE IS 50 µS, AND THE MINORITY CARRIER LIFETIME ON THE NSIDE IS 50 µS

```
# Given
Na = 1e15
Nd = 1e15
t_p = 50e-6
t_n = 50e-9

# Property Constants. mu_n and mu_p values gathered from https://www.ecse.rpi.edu/~schubert/Educational-resources/Materials-Semiconductors-Si-and-Ge.pdf
n_i = 1.79e13
mu_p = 1900
mu_n = 3900
```

1.1 WHAT IS THE BUILT-IN VOLTAGE, Vbi?

```
def calculate_built_in_voltage(n_i, Nd, Na):
    return KBT_Q*math.log(Na * Nd / n_i**2)
```

```
Vbi = calculate_built_in_voltage(n_i, Nd, Na)
print("What is the built-in voltage, Vbi?", Vbi)
```

0.208389 [V]

1.2 WHAT IS THE EXCESS ELECTRON CONCENTRATION AT $x = -xp$, FOR $Vapp = -3\ V$?

```
def calculate_excess_electron_concentration(n_i, Na, Vapp):
    return  n_i**2 / Na * (math.exp(Vapp / KBT_Q) - 1)
```

-320,410,000,000 [cm ^ -3]

1.3 WHAT IS THE EXCESS ELECTRON CONCENTRATION AT $x = -xp$, FOR $Vapp = 0.5\ V$?

7.75835e+19 [cm ^ -3]

1.4 WHAT IS THE REVERSE SATURATION CURRENT DENSITY, JS?

```
def calculate_D(mu):
    return KBT_Q * mu
```

```
def calculate_L(D, t):
    return (D * t)**0.5
```

```
def calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na):
    return q * n_i**2 * ((D_p / (L_p * Nd)) + (D_n / (L_n * Na)))
```

```python
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, t_p)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, t_n)

Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
print("What is the reverse saturation current density, JS?", Js)
```

0.001681168 [A / cm^2]

1.5 WHAT IS THE CURRENT DENSITY FOR $Vapp = -3\ V$?

```python
def calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na):
    Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
    return Js * (math.exp(Vapp/KBT_Q) - 1)
```

```python
Vapp = -3
current_density = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)
print("What is the current density for Vapp = -3 V?", current_density)
```

-0.001681168 [A / cm^2]

1.6 WHAT IS THE CURRENT DENSITY FOR $Vapp = 0.5\ V$?

```python
Vapp = 0.5
current_density = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)
print("What is the current density for Vapp = 0.5 V?", current_density)
```

407075.0774 [A / cm^2]

For the following problems, assume:

- *The cross-sectional area is 250 µm x 250 µm.*
- *The minority carrier lifetime on the p-side is 100 ns.*
- *The minority carrier lifetime on the n-side is 10 µs*

```python
# GIVEN
AREA = 250e-4 * 250e-4 # Area in CM-2
T_P = 100e-9
T_N = 10e-6
```

2 Consider a Si p-n junction with NA = 1018 cm-3 and ND = 1017 cm-3 .

```python
# Given
Na = 1e18
Nd = 1e17

# material properties
n_i = 1e10
mu_n = 261
mu_p = 331
```

2.1 What is the reverse saturation current, Is?

```python
# Perform intermediate calculations
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, T_P)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, T_N)

Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
Is = Js * AREA
print("What is the reverse saturation current, Is?", Is)
```

1.748085672905745e-16 [A]

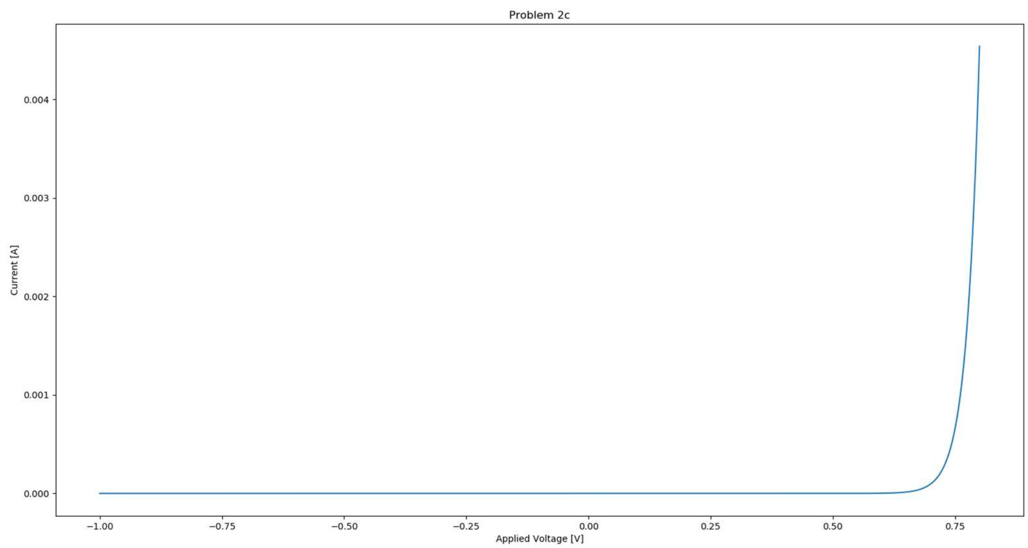
2.2 What is the current for $Vapp$ = 0.5 V ?

```python
Vapp = 0.5
J = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)
current = J * AREA
print("What is the current for Vapp = 0.5?", current)
```

4.2327836289216224e-08 [A]

2.3 Using a computer, plot the current versus applied voltage, ranging from -1 V to 0.8 V. Turn in your code.

```
print("Using a computer plot the current versus applied voltage ranging from -1 to 0.8 V.")
applied_voltages = np.linspace(-1, 0.8, 2000)
currents = list([])
for Vapp in applied_voltages:
    currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)

currents = np.array(currents)
plt.plot(applied_voltages, currents)
plt.xlabel("Applied Voltage [V]")
plt.ylabel("Current [A]")
plt.title("Problem 2c")
plt.show()
```



2.4 What is the apparent turn-on voltage?

Approximately 0.75 [V]0

3 CONSIDER A GAN P-N JUNCTION WITH NA = 1018 CM-3 AND ND = 1017 CM-3 .

```
# Given
Na = 1e18
Nd = 1e17

# material properties
n_i = 1.77e-10
mu_n = 551
mu_p = 142
```

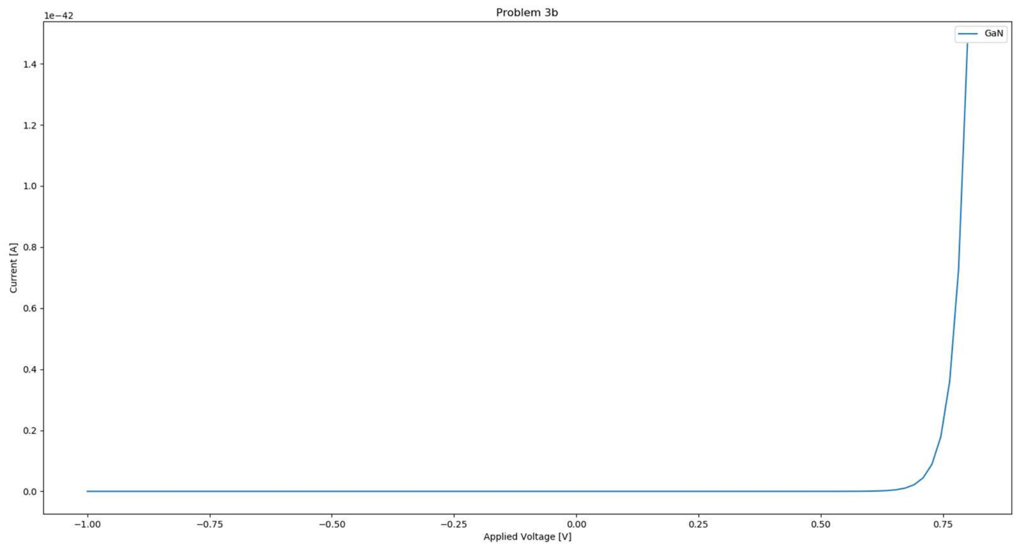
3.1 WHAT IS THE REVERSE SATURATION CURRENT, IS?

```
# Perform intermediate calculations
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, T_P)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, T_N)

Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
Is = Js * AREA
print("What is the reverse saturation current, Js?", Is)
```

5.6425327326520155e-56 [A]

3.2 USING A COMPUTER, PLOT THE CURRENT VERSUS APPLIED VOLTAGE, RANGING FROM -1 V TO 0.8 V. TURN IN YOUR CODE.

```
print("Using a computer, plot the current versus applied voltage ranging form -1 V to 0.8 V.")
applied_voltages = np.linspace(-1, 0.8, 100)
currents = list([])
for Vapp in applied_voltages:
    currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)

currents = np.array(currents)
plt.plot(applied_voltages, currents, label="GaN")
plt.xlabel("Applied Voltage [V]")
plt.ylabel("Current [A]")
plt.title("Problem 3b")
plt.legend()
plt.show()
```



3.3 What is the apparent turn-on voltage?

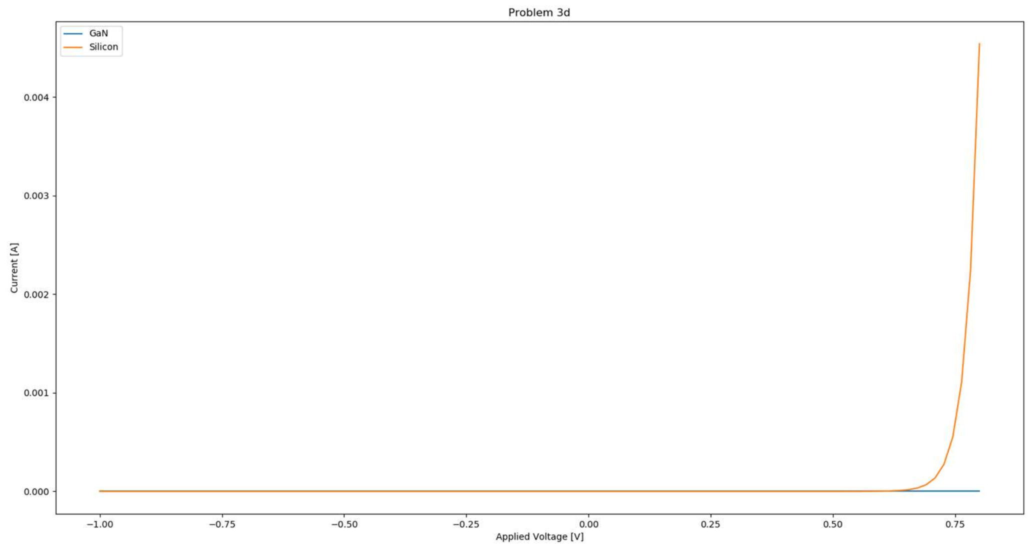
0.75 [V]

3.4 Compare this graph with the graph from problem #2. For the same voltage, which diode has greater current: the silicon diode or the GaN diode?

The silicon diode has a much greater current than the GaN diode.

3.5 Change the x-axis so that the forward current is similar to that found for the silicon diode. Superimpose the graphs for the silicon diode and the GaN diode. Make a single, nice graph. Turn in your code.

```
# material properties
n_i = 1e10
mu_n = 261
mu_p = 331

# Perform intermediate calculations
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, T_P)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, T_N)

print("Change the X-axis so that the forward current is similar to that found in the silcon diode.  Superimpose the graphs for the silicon diode and the GaN diode.")
problem2_currents = list([])
for Vapp in applied_voltages:
    problem2_currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)
problem2_currents = np.array(problem2_currents)

plt.plot(applied_voltages, currents, label="GaN")
plt.plot(applied_voltages, problem2_currents, label="Silicon")
plt.legend()
plt.xlabel("Applied Voltage [V]")
plt.ylabel("Current [A]")
plt.title("Problem 3d")
plt.show()
```



3.6 What is the apparent turn-on voltage for the silicon diode and the GaN diode?

0.75 [V]

4 Consider a Si p-n junction with NA = 1018 cm-3 and ND = 1017 cm-3 . The length of the n-region is 200 µm, and the length of the p- region is 20 µm. What are:

```
# Given
Na = 1e18
Nd = 1e17
x_n = 200e-4
x_p = 20e-4

# material properties
n_i = 1e10
e_s = 11.8
epsilon_crit = 3e5
```

4.1 The breakdown voltage considering only avalanche breakdown?

```
def calculate_avalanche_breakdown(n_i, Nd, Na, e_s, epsilon_crit):
    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    return epsilon_crit**2 * (e_s * e_0 / (2 * q)) * ((Nd + Na)/(Na*Nd)) + Vbi
```

```
Vbr_avalanche = calculate_avalanche_breakdown(n_i, Nd, Na, e_s, epsilon_crit)
print("What is the breakdown voltage considering only avalanche breakdown?", Vbr_avalanche)
```

4.126817683628187 [V]

4.2 THE BREAKDOWN VOLTAGE CONSIDERING ONLY PUNCH-THROUGH ON THE N-SIDE?

```
def calculate_punch_through_n(n_i, Nd, Na, e_s, x_n):
    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    return q * Nd / (2 * e_s * e_0) * ((Na + Nd) / Na) * x_n**2 - Vbi
```

```
Vbr_punch_through_n = calculate_punch_through_n(n_i, Nd, Na, e_s, x_n)
print("What is the breakdown voltage considering only punch-through on the n-side?", Vbr_punch_through_n)
```

3369155.246689906 [V]

4.3 THE BREAKDOWN VOLTAGE CONSIDERING ONLY PUNCH-THROUGH ON THE P-SIDE?

```
def calculate_punch_through_p(n_i, Nd, Na, e_s, x_p):
    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    return q * Na / (2 * e_s * e_0) * ((Nd + Na) / Nd) * x_p**2 - Vbi
```

```
Vbr_punch_through_p = calculate_punch_through_p(n_i, Nd, Na, e_s, x_p)
print("What is the breakdown voltage considering only punch-through on the p-side?", Vbr_punch_through_p)
```

3369155.246689906 [V]

4.4 THE OVERALL BREAKDOWN VOLTAGE?

```
print("What is the overall breakdown voltage?", Vbr_avalanche + Vbr_punch_through_n + Vbr_punch_through_p)
```

6738314.620197495 [V]

```
import math

import numpy as np

import matplotlib.pyplot as plt


KBT_Q = 0.0259

q = 1.6e-19

e_0 = 8.854e-14


def calculate_excess_electron_concentration(n_i, Na, Vapp):

    return  n_i**2 / Na * (math.exp(Vapp / KBT_Q) - 1)


def calculate_built_in_voltage(n_i, Nd, Na):

    return KBT_Q*math.log(Na * Nd / n_i**2)


def calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na):

    return q * n_i**2 * ((D_p / (L_p * Nd)) + (D_n / (L_n * Na)))


def calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na):

    Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)

    return Js * (math.exp(Vapp/KBT_Q) - 1)


def calculate_D(mu):

    return KBT_Q * mu


def calculate_L(D, t):

    return (D * t)**0.5


def calculate_avalanche_breakdown(n_i, Nd, Na, e_s, epsilon_crit):

    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
```

```
    return epsilon_crit**2 * (e_s * e_0 / (2 * q)) * ((Nd + Na)/(Na*Nd)) + Vbi


def calculate_punch_through_n(n_i, Nd, Na, e_s, x_n):
    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    return q * Nd / (2 * e_s * e_0) * ((Na + Nd) / Na) * x_n**2 - Vbi


def calculate_punch_through_p(n_i, Nd, Na, e_s, x_p):
    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    return q * Na / (2 * e_s * e_0) * ((Nd + Na) / Nd) * x_p**2 - Vbi


def problem_1():
    print("\nProblem 1")
    print("Consider a Germanium p-n junction with NA = 1015 cm-3 and ND = 1015 cm-3. The minority carrier lifetime on the p-side is 50 micro-s, and the minority carrier lifetime on the nside is 50 micro-s.")
    # Given
    Na = 1e15
    Nd = 1e15
    t_p = 50e-6
    t_n = 50e-9

    # Property Constants. mu_n and mu_p values gathered from https://www.ecse.rpi.edu/~schubert/Educational-resources/Materials-Semiconductors-Si-and-Ge.pdf
    n_i = 1.79e13
    mu_p = 1900
    mu_n = 3900

    Vbi = calculate_built_in_voltage(n_i, Nd, Na)
    print("What is the built-in voltage, Vbi?", Vbi)
```

```
    Vapp = -3
    excess_electron_concentration = calculate_excess_electron_concentration(n_i, Na, Vapp)
    print("What is the excess electron concentration at 𝑥 = −xp, for Vapp = −3 𝑉?", excess_electron_concentration)

    Vapp = 0.5
    excess_electron_concentration = calculate_excess_electron_concentration(n_i, Na, Vapp)
    print("What is the excess electron concentration at 𝑥 = −xp, for Vapp = 0.5 𝑉?", excess_electron_concentration)

    D_n = calculate_D(mu_n)
    L_n = calculate_L(D_n, t_p)

    D_p = calculate_D(mu_p)
    L_p = calculate_L(D_p, t_n)

    Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
    print("What is the reverse saturation current density, 𝐽𝑆?", Js)

    Vapp = -3
    current_density = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)
    print("What is the current density for 𝑉𝑎𝑝𝑝 = −3 𝑉?", current_density)

    Vapp = 0.5
    current_density = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)
    print("What is the current density for 𝑉𝑎𝑝𝑝 = 0.5 𝑉?", current_density)

# GIVEN
AREA = 250e-4 * 250e-4 # Area in CM-2
```

```
T_P = 100e-9

T_N = 10e-6


def problem_2():

    print("\nProblem 2:")

    print("Consider a Si p-n junction with NA = 1018 cm-3 and ND = 1017 cm-3")

    # Given

    Na = 1e18

    Nd = 1e17


    # material properties

    n_i = 1e10

    mu_n = 261

    mu_p = 331


    # Perform intermediate calculations

    D_n = calculate_D(mu_n)

    L_n = calculate_L(D_n, T_P)


    D_p = calculate_D(mu_p)

    L_p = calculate_L(D_p, T_N)


    Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)

    Is = Js * AREA

    print("What is the reverse saturation current, Is?", Is)


    Vapp = 0.5

    J = calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na)

    current = J * AREA
```

```
    print("What is the current for Vapp = 0.5?", current)

    print("Using a computer plot the current versus applied voltage ranging from -1 to 0.8 V.")
    applied_voltages = np.linspace(-1, 0.8, 2000)
    currents = list([])
    for Vapp in applied_voltages:
        currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)

    currents = np.array(currents)
    plt.plot(applied_voltages, currents)
    plt.xlabel("Applied Voltage [V]")
    plt.ylabel("Current [A]")
    plt.title("Problem 2c")
    plt.show()

    print("What is the apprent turn-on voltage?")
    print("Approximately 0.75V")

def problem_3():
    print("\nProblem 3:")
    print("Consider a GaN p-n junction with Na = 1018 cm-3 and Nd = 1017 cm-3")
    # Given
    Na = 1e18
    Nd = 1e17

    # material properties
    n_i = 1.77e-10
    mu_n = 551
    mu_p = 142
```

```
# Perform intermediate calculations
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, T_P)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, T_N)

Js = calculate_Js(n_i, D_p, L_p, Nd, D_n, L_n, Na)
Is = Js * AREA
print("What is the reverse saturation current, Js?", Is)

print("Using a computer, plot the current versus applied voltage ranging form -1 V to 0.8 V.")
applied_voltages = np.linspace(-1, 0.8, 100)
currents = list([])
for Vapp in applied_voltages:
   currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)

currents = np.array(currents)
plt.plot(applied_voltages, currents, label="GaN")
plt.xlabel("Applied Voltage [V]")
plt.ylabel("Current [A]")
plt.title("Problem 3b")
plt.legend()
plt.show()

print("What is the apparent turn-on voltage?")
print("Approximately 0.75V")
```

```
# material properties
n_i = 1e10
mu_n = 261
mu_p = 331

# Perform intermediate calculations
D_n = calculate_D(mu_n)
L_n = calculate_L(D_n, T_P)

D_p = calculate_D(mu_p)
L_p = calculate_L(D_p, T_N)

print("Change the X-axis so that the forward current is similar to that found in the silcon diode. 
Superimpose the graphs for the silicon diode and the GaN diode.")
problem2_currents = list([])
for Vapp in applied_voltages:
    problem2_currents.append(calculate_J(Vapp, n_i, D_p, L_p, Nd, D_n, L_n, Na) * AREA)
problem2_currents = np.array(problem2_currents)

plt.plot(applied_voltages, currents, label="GaN")
plt.plot(applied_voltages, problem2_currents, label="Silicon")
plt.legend()
plt.xlabel("Applied Voltage [V]")
plt.ylabel("Current [A]")
plt.title("Problem 3d")
plt.show()

print("What is the apparent turn-on voltage for the silicon diode and the GaN diode?")
print("Approximately 0.75V")
```

```
def problem_4():

    print("\nProblem 4:")

    print("Consider a Si p-n junction with NA = 1018 cm-3 and ND = 1017 cm-3 . The length of the n-region is 200 micro-m, and the length of the p- region is 20 micro-m. ")

    # Given

    Na = 1e18

    Nd = 1e17

    x_n = 200e-4

    x_p = 20e-4


    # material properties

    n_i = 1e10

    e_s = 11.8

    epsilon_crit = 3e5


    Vbr_avalanche = calculate_avalanche_breakdown(n_i, Nd, Na, e_s, epsilon_crit)

    print("What is the breakdown voltage considering only avalanche breakdown?", Vbr_avalanche)


    Vbr_punch_through_n = calculate_punch_through_n(n_i, Nd, Na, e_s, x_n)

    print("What is the breakdown voltage considering only punch-through on the n-side?", Vbr_punch_through_n)


    Vbr_punch_through_p = calculate_punch_through_p(n_i, Nd, Na, e_s, x_p)

    print("What is the breakdown voltage considering only punch-through on the p-side?", Vbr_punch_through_p)


    print("What is the overall breakdown voltage?", Vbr_avalanche + Vbr_punch_through_n + Vbr_punch_through_p)
```

```
if __name__ == "__main__":
    problem_1()
    problem_2()
    problem_3()
    problem_4()
```