Riley Worstell

ECE 542

HW # 6

2/16/2019

1)

Riley Worstell ECE 542 HW#6

#1 NMOS $N_A = 10^{17}\,cm^{-3}$, $N_{SS} = 1.5 \cdot 10^{11}\,cm^{-2}$, aluminum gate $\Phi_G = 4.1V$

insulator SiO_2 20nm thick

$20 \cdot 10^{-7}\,cm$

a) $V_{FB} = \Phi_{GS} - \frac{Q_{SS}}{C_i}$; $\Phi_S = \frac{1}{q}\left(\chi + \frac{E_g}{2} + E_{iF}\right)$

$= (4.05 + 0.56 + 0.417) = 5.027V$

$E_{iF} = 0.0259 \ln\left(\frac{10^{17}}{10^{10}}\right) = 0.417$

$\Phi_{GS} = 4.1 - 5.027 = -0.927V$

$C_i = \frac{\varepsilon_i \varepsilon_0}{t_i} = \frac{(3.9)(8.854 \cdot 10^{-14} F/cm)}{20 \cdot 10^{-7} cm} = 173 \cdot 10^{-9}\,F/cm^2$

$Q_{SS} = qN_{SS} = (1.6 \cdot 10^{-19} C)(1.5 \cdot 10^{11}\,cm^{-2}) = 24\,nC/cm^2$

$V_{FB} = \Phi_{GS} - \frac{Q_{SS}}{C_i} = -0.927V - \frac{24}{173} = -0.9409V$

b) $V_T = V_{FB} + \Phi_{SB} + \frac{\sqrt{2\varepsilon_s \varepsilon_0 q N_A \Phi_{SB}}}{C_i}$

$\Phi_{SB} = 2(0.0259)\ln\left(\frac{10^{17}}{10^{10}}\right) = 0.834V$

$V_T = -0.9409 + 0.834 + \frac{1}{\left(173\frac{nF}{cm^2}\right)}\sqrt{2(11.8)(8.854 \cdot 10^{-14} F/cm)(1.6 \cdot 10^{-19})(10^{17})(0.834)}$

$-0.9409V + 0.834V + \frac{167 \cdot \sqrt{\frac{nF}{cm^2}}}{173\frac{nF}{cm^2}}$

$= -0.9409 + 0.834 + 0.965$

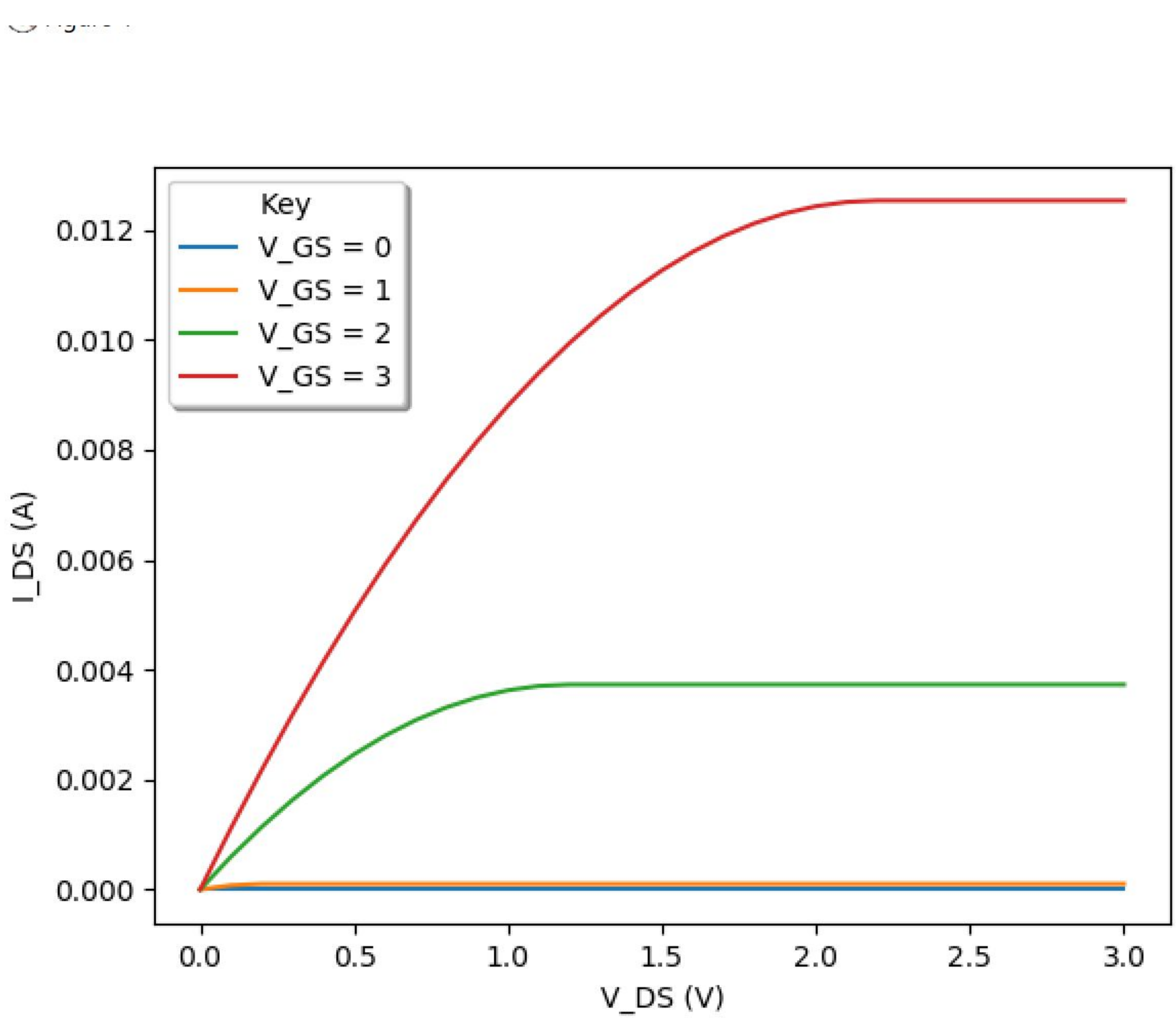
$= 0.8581V$

2) CODE:

```
import matplotlib.pylab as plt

W = 150 * 10**-7
L = 100 * 10**-7
u = 350
V_T = 0.8
V_GS = 0
t_i = 35 * 10**-9
C_i = (3.9 * (8.854*10**-14))/t_i

V_DS = []

x = 0
for i in range(31):
    V_DS.append(x)
    x = x + 0.1

V_GS_0 = 0
V_GS_1 = 1
V_GS_2 = 2
V_GS_3 = 3

V_DS_sat1 = V_GS_1 - V_T
I_DS = []
for i in V_DS:
    x = ((u*C_i*W)/L)*(((V_GS_1-V_T)*i)-(0.5*(i**2)))
    y = ((0.5*u*C_i*W)/L)*((V_GS_1 - V_T)**2)
    if V_GS_1 <= V_T:
        I_DS.append(0)
    else:
        if (V_GS_1 >= V_T) and (i <= V_DS_sat1):
            I_DS.append(x)
        else:
            I_DS.append(y)


V_DS_sat2 = V_GS_2 - V_T
I_DS_2 = []
for i in V_DS:
    x = ((u*C_i*W)/L)*(((V_GS_2-V_T)*i)-(0.5*(i**2)))
    y = ((0.5*u*C_i*W)/L)*((V_GS_2 - V_T)**2)
    if V_GS_2 <= V_T:
        I_DS_2.append(0)
    else:
        if (V_GS_2 >= V_T) and (i <= V_DS_sat2):
            I_DS_2.append(x)
        else:
            I_DS_2.append(y)


V_DS_sat3 = V_GS_3 - V_T
I_DS_3 = []
for i in V_DS:
    x = ((u*C_i*W)/L)*(((V_GS_3-V_T)*i)-(0.5*(i**2)))
    y = ((0.5*u*C_i*W)/L)*((V_GS_3 - V_T)**2)
    if V_GS_3 <= V_T:
        I_DS_3.append(0)
    else:
        if (V_GS_3 >= V_T) and (i <= V_DS_sat3):
            I_DS_3.append(x)
        else:
            I_DS_3.append(y)


V_DS_sat0 = V_GS_0 - V_T
I_DS_0 = []
```

```
for i in V_DS:
    x = ((u * C_i * W) / L) * (((V_GS_0 - V_T) * i) - (0.5 * (i ** 2)))
    y = ((0.5 * u * C_i * W) / L) * ((V_GS_0 - V_T) ** 2)
    if V_GS_0 <= V_T:
        I_DS_0.append(0)
    else:
        if (V_GS_0 >= V_T) and (i <= V_DS_sat0):
            I_DS_0.append(x)
        else:
            I_DS_0.append(y)


plt.plot(V_DS, I_DS_0, label="V_GS = 0")
plt.plot(V_DS, I_DS, label="V_GS = 1")
plt.plot(V_DS, I_DS_2, label="V_GS = 2")
plt.plot(V_DS, I_DS_3, label="V_GS = 3")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.xlabel("V_DS (V)")
plt.ylabel("I_DS (A)")
plt.show()
```

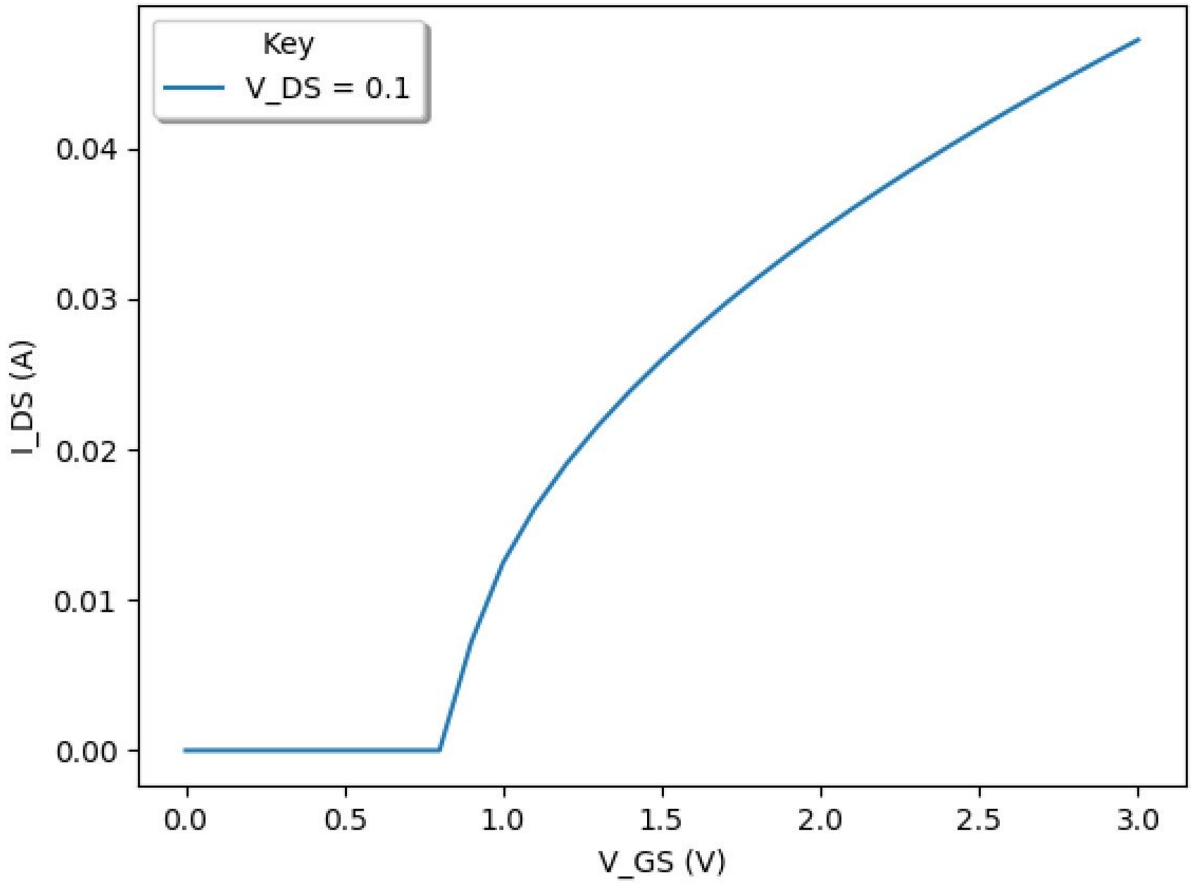
PLOT:



3) CODE:

```
import matplotlib.pylab as plt

W = 150 * 10**-7
L = 100 * 10**-7
u = 350
V_T = 0.8
V_DS = 0.1
t_i = 35 * 10**-9
C_i = (3.9 * (8.854*10**-14))/t_i


V_GS = []
x = 0
for i in range(31):
    V_GS.append(x)
    x = x + 0.1

I_DS = []
for i in V_GS:
    V_DS_sat1 = i - V_T
    x = ((u*C_i*W)/L)*(((i-V_T)*V_DS)-(0.5*(V_DS**2)))
    y = ((0.5*u*C_i*W)/L)*((i - V_T)**2)
    if i <= V_T:
        I_DS.append(0)
    else:
        if (i >= V_T) and (V_DS <= V_DS_sat1):
            I_DS.append(x)
        else:
            I_DS.append(y)

I_DS_sq = []
for i in I_DS:
    i = ((2 * i) ** 0.5)
    I_DS_sq.append(i)


plt.plot(V_GS, I_DS_sq, label="V_DS = 0.1")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.xlabel("V_GS (V)")
plt.ylabel("I_DS (A)")
plt.show()
```

PLOT:



4)

a) Saturation Transconductance for V_GS = 3V:

code:

```
g_m_sat = ((u*C_i*W)/L)*(3-V_T)
print(g_m_sat)
```

Output:

0.00011395098 in (Amps/Volt) or just (A/V)

b) Saturation Output Conductance for V_GS = 3V:

0 (Amps/Volt)(A/V)

c) Triode Transconductance for V_DS = 1V and d) Triode output conductance for V_DS = 1V, V_GS = 3V:

Code:

```
# 4c 4d
g_m_tri = (((u*C_i*W)/L)*1)
print("g_m in triode is: ", g_m_tri, " in Amps per Volt (A/V)")
g_0_tri = (((u*C_i*W)/L)*((3-V_T)-1))
print("g_0 in triode is: ", g_0_tri, " in Amps per Volt (A/V)")
```

Output:

g_m in triode is: 5.1795899999999997e-05 in Amps per Volt (A/V)

g_0 in triode is: 6.215508000000001e-05 in Amps per Volt (A/V)

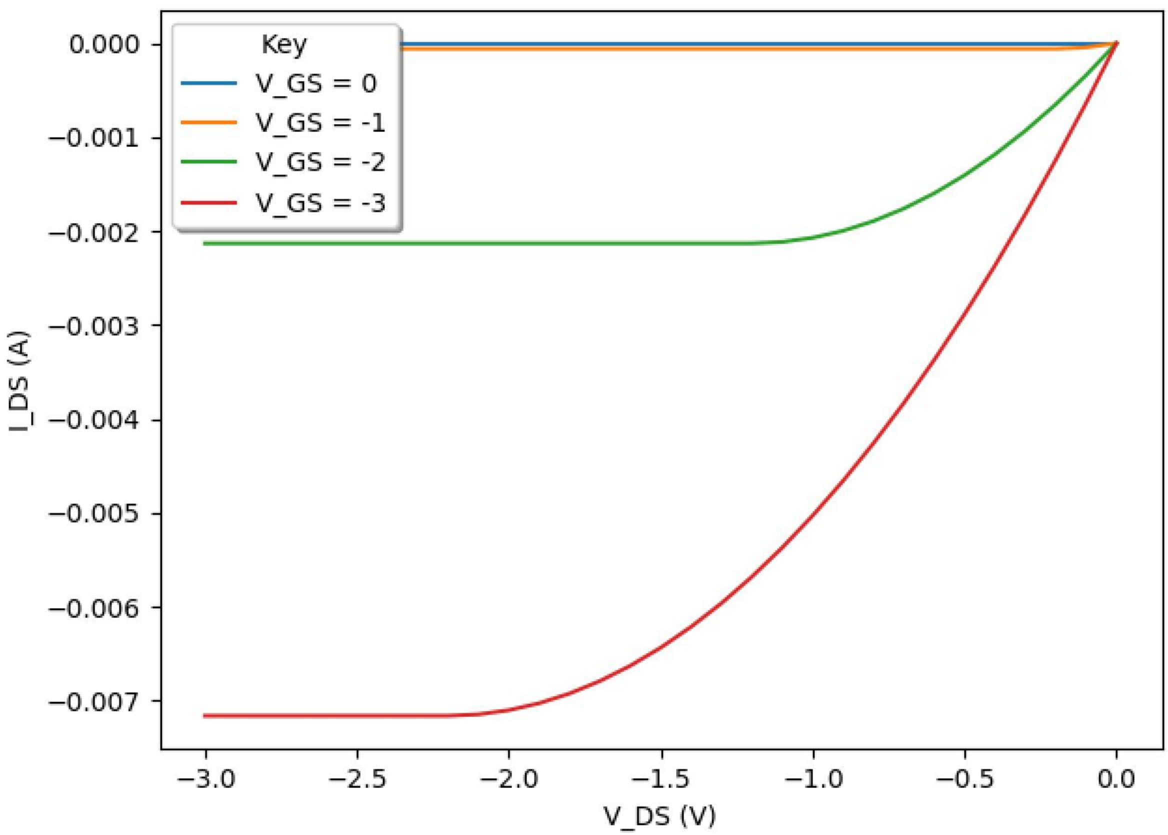
5)

Code:

```
import matplotlib.pylab as plt

W = 150 * 10**-7
L = 100 * 10**-7
u = 200
V_T = -0.8
V_GS = 0.1
t_i = 35 * 10**-9
C_i = (3.9 * (8.854*10**-14))/t_i


V_DS = []
x = 0
for i in range(31):
    V_DS.append(x)
    x = x - 0.1

V_GS_0 = 0
V_GS_1 = -1
V_GS_2 = -2
V_GS_3 = -3

V_DS_sat1 = V_GS_1 - V_T
I_DS = []
for i in V_DS:
    x = ((-u*C_i*W)/L)*(((V_GS_1-V_T)*i)-(0.5*(i**2)))
    y = ((-0.5*u*C_i*W)/L)*((V_GS_1 - V_T)**2)
    if V_GS_1 >= V_T:
        I_DS.append(0)
    else:
        if (V_GS_1 <= V_T) and (i >= V_DS_sat1):
            I_DS.append(x)
        else:
            I_DS.append(y)


V_DS_sat2 = V_GS_2 - V_T
I_DS_2 = []
for i in V_DS:
    x = ((-u*C_i*W)/L)*(((V_GS_2-V_T)*i)-(0.5*(i**2)))
    y = ((-0.5*u*C_i*W)/L)*((V_GS_2 - V_T)**2)
    if V_GS_2 >= V_T:
```

```python
        I_DS_2.append(0)
    else:
        if (V_GS_2 <= V_T) and (i >= V_DS_sat2):
            I_DS_2.append(x)
        else:
            I_DS_2.append(y)


V_DS_sat3 = V_GS_3 - V_T
I_DS_3 = []
for i in V_DS:
    x = ((-u*C_i*W)/L)*(((V_GS_3-V_T)*i)-(0.5*(i**2)))
    y = ((-0.5*u*C_i*W)/L)*((V_GS_3 - V_T)**2)
    if V_GS_3 >= V_T:
        I_DS_3.append(0)
    else:
        if (V_GS_3 <= V_T) and (i >= V_DS_sat3):
            I_DS_3.append(x)
        else:
            I_DS_3.append(y)


V_DS_sat0 = V_GS_0 - V_T
I_DS_0 = []
for i in V_DS:
    x = ((-u * C_i * W) / L) * (((V_GS_0 - V_T) * i) - (0.5 * (i ** 2)))
    y = ((-0.5 * u * C_i * W) / L) * ((V_GS_0 - V_T) ** 2)
    if V_GS_0 >= V_T:
        I_DS_0.append(0)
    else:
        if (V_GS_0 <= V_T) and (i >= V_DS_sat0):
            I_DS_0.append(x)
        else:
            I_DS_0.append(y)


plt.plot(V_DS, I_DS_0, label="V_GS = 0")
plt.plot(V_DS, I_DS, label="V_GS = -1")
plt.plot(V_DS, I_DS_2, label="V_GS = -2")
plt.plot(V_DS, I_DS_3, label="V_GS = -3")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.xlabel("V_DS (V)")
plt.ylabel("I_DS (A)")
plt.show()
```

Plot:



EXTRA CREDIT:

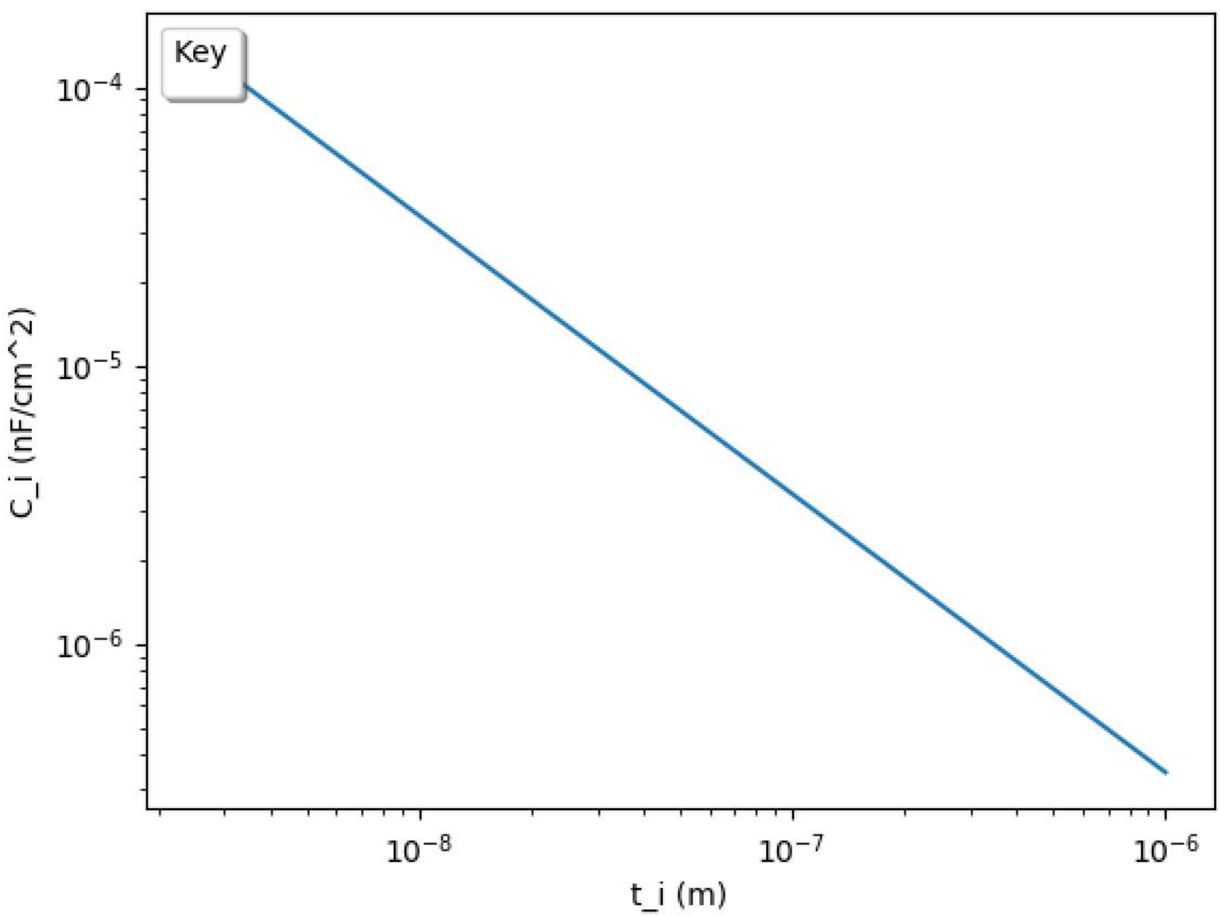
1)

a)

```
import matplotlib.pylab as plt
import math

f = 0.1
t_i_min = 1.5*(10**-9)
t_i_max = 1*(10**-6)
t_i = []

for i in range(1000):
    t_i_min = t_i_min + (1*(10**-9))
    t_i.append(t_i_min)


C_i = []
for i in t_i:
    x = (3.9 * (8.854 * 10 ** -14)) / i
    C_i.append(x)
```

```
plt.xscale('log')
plt.yscale('log')
plt.plot(t_i, C_i, label="")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.xlabel("t_i" (m))
plt.ylabel("C_i" (nF/cm^2))
plt.show()
```

Plot:



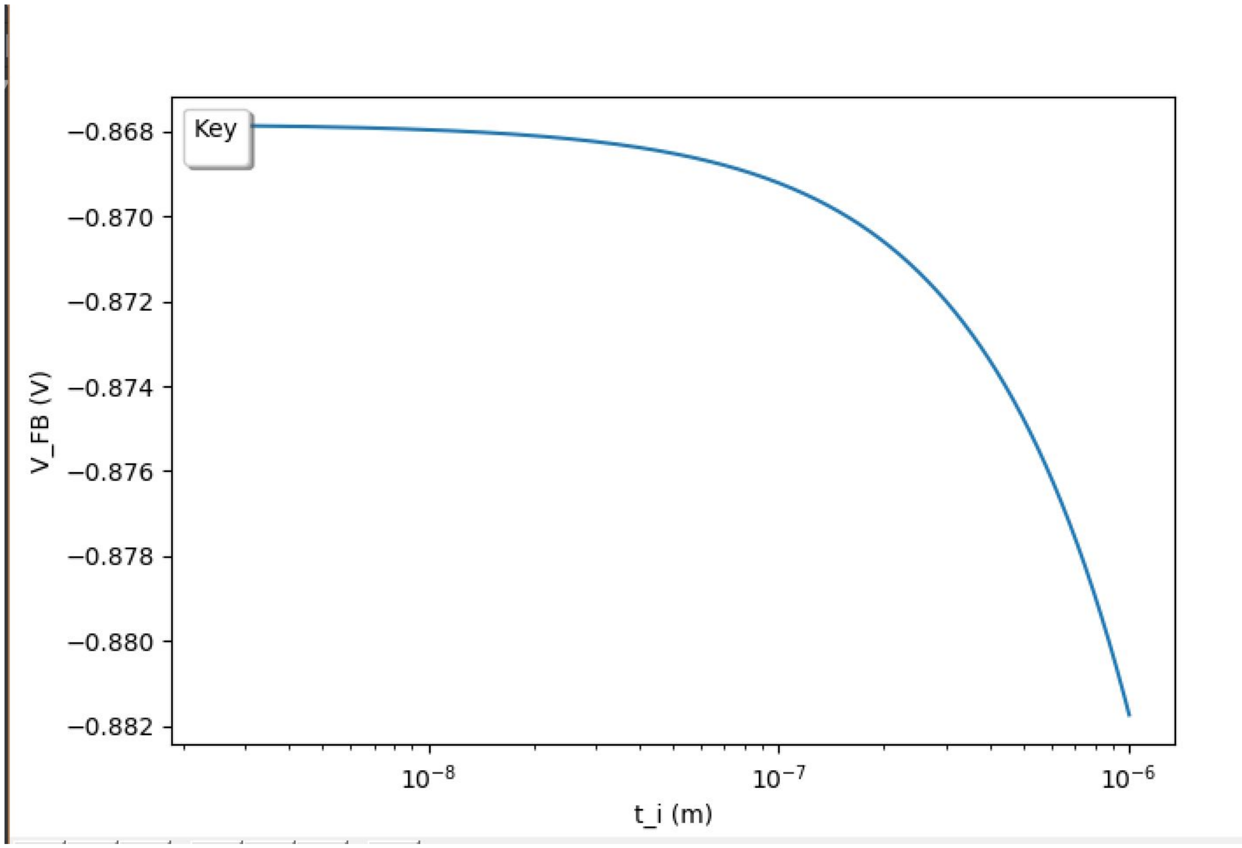
b)

code:

```
import matplotlib.pylab as plt
import math

f = 0.1
```

```
t_i_min = 1.5*(10**-9)
t_i_max = 1*(10**-6)
t_i = []

for i in range(1000):
    t_i_min = t_i_min + (1*(10**-9))
    t_i.append(t_i_min)


X = 4.05
N_A = 10**16
N = 10**10
E_iF = 0.0259*(math.log1p(N_A/N))
Q_ss = (1.6*(10**-19))*(3*(10**10))

phi_B = X+0.56+E_iF
phi_G = 4.1
phi_GB = phi_G - phi_B

# V_FB = phi_GB - (Q_ss/C_i)

V_FB = []
for i in t_i:
    x = (3.9 * (8.854 * 10 ** -14)) / i
    y = phi_GB - (Q_ss/x)
    V_FB.append(y)


plt.semilogx(t_i, V_FB, label="")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.ylabel("V_FB (V)")
plt.xlabel("t_i (m)")
plt.show()
```

PLOT:



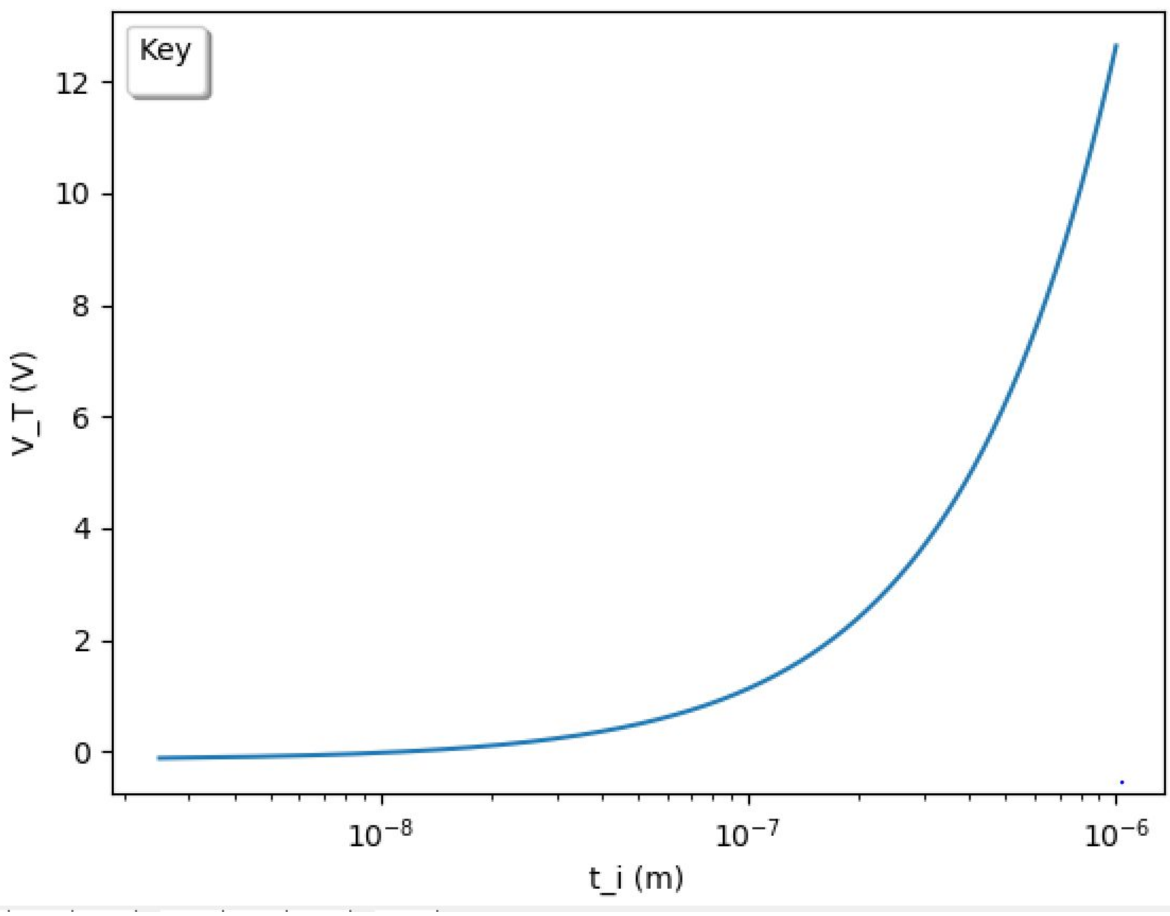
c)

code:

```
import matplotlib.pylab as plt
import math

f = 0.1
t_i_min = 1.5*(10**-9)
t_i_max = 1*(10**-6)
t_i = []

for i in range(1000):
    t_i_min = t_i_min + (1*(10**-9))
    t_i.append(t_i_min)


X = 4.05
N_A = 10**16
N = 10**10
E_iF = 0.0259*(math.log1p(N_A/N))
phi_SB = 2*E_iF
Q_ss = (1.6*(10**-19))*(3*(10**10))

phi_B = X+0.56+E_iF
phi_G = 4.1
phi_GB = phi_G - phi_B
```

```
# V_FB = phi_GB - (Q_ss/C_i)

V_T = []
for i in t_i:
    i = i * (10**2)
    x = (3.9 * (8.854 * 10 ** -14)) / i
    y = phi_GB - (Q_ss/x)
    z = phi_SB + y + (((2*11.8*(8.854*(10**-14))*(10**16)*(1.6*(10**-19))*phi_SB)**0.5)/x)
    V_T.append(z)


plt.semilogx(t_i, V_T, label="")
plt.legend(loc="upper left", shadow=True, title="Key", fancybox=True)
plt.xlabel("t_i (m)")
plt.ylabel("V_T (V)")
plt.show()
```

c)



Extra Credit:

2) V_FB = phi_GB – (qN_ss/C_i) and t_i changes the value of C_i. As t_i gets small C_i gets really large.

This results in V_FB is about = phi_GB.

3) Yes, the values at the very beginning are negative for the threshold voltage V_T.