EE236: Experiment 8
Bipolar Junction Transistor

Param Rathour, 190070049
Spring Semester, 2021-22

Contents

1 Overview of the experiment

This report contains my approach to the experiment, the circuit's design with the relevant circuit designs and output plots & values.

1.1 Aim of the experiment

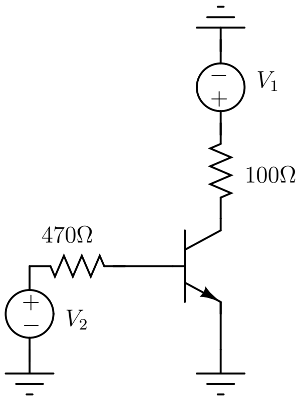
- To plot I_C vs V_{CE} and find BJT parameters in CE configuration
- To plot I_C vs V_{CB} and find BJT parameters in CB configuration
- To plot and understand Gummel Plot
- To calculate small signal Parameters and understand small signal behaviours
- To plot and compare switching characteristics of NMOS and BJT

1.2 Methods

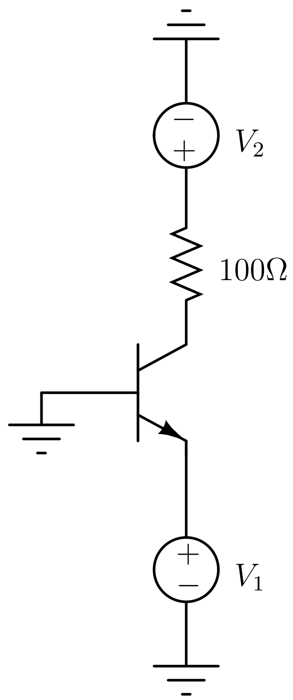
I_C vs V_{CE} characterisitics then by I_C vs V_{CB} characterisitics was done by varying I_B.
Then Gummel Plot was plotted.
Then, small signal characteristics were plotted
Lastly, switching characteristics of NMOS and BJT were compared
All figures & parameters were calculated using hands-on experiment and exporting the results to python

2 Design

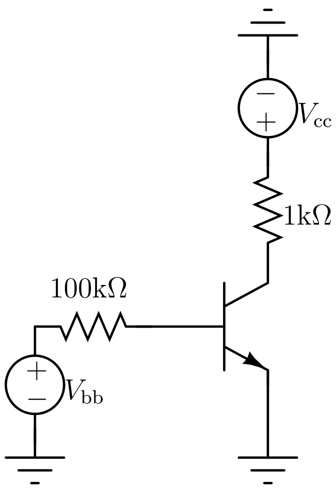
The circuits designed for all parts are shown below with their physical counterparts
For the calculation of reverse β, C and E terminals were interchanged.



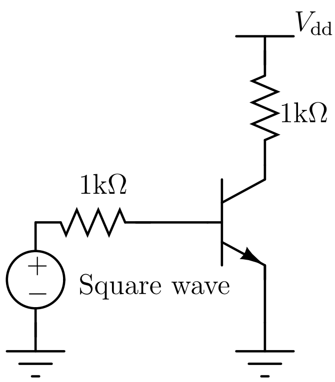
(a) BJT Parameters in CE configuration



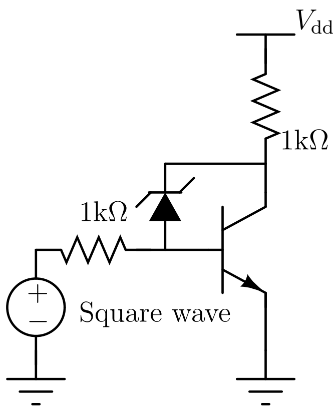
(b) BJT Parameters in CB configuration and Gummel Plot



(c) Small Signal Parameters



(d) Switching behaviour



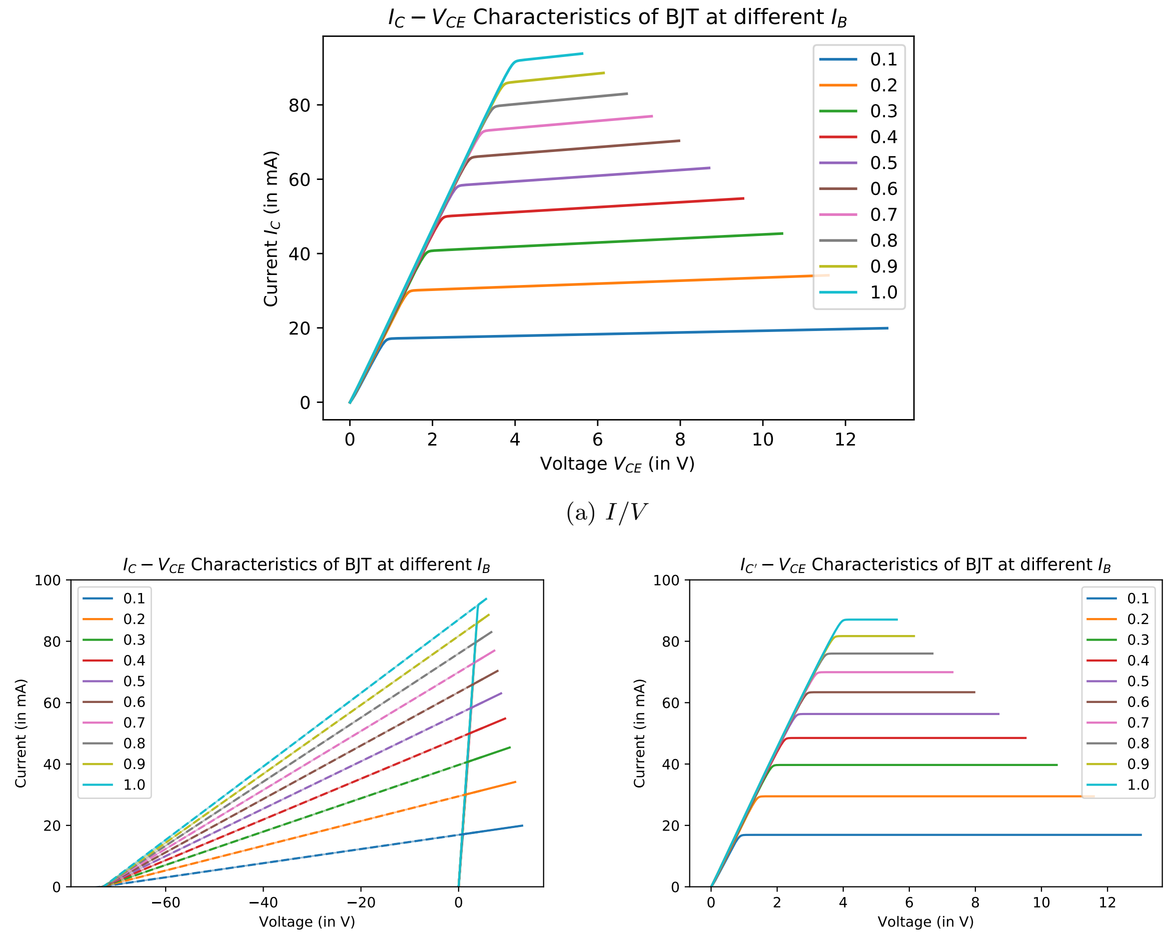
(e) Switching behaviour with Diode

Figure 1: Design circuits

3 Simulation Results

3.1 Plots

3.1.1 BJT Parameters in CE configuration



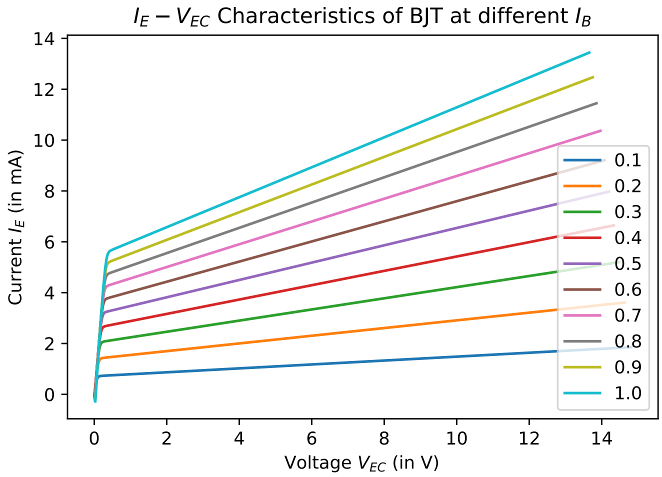
(a) I/V

(b) Early Effect

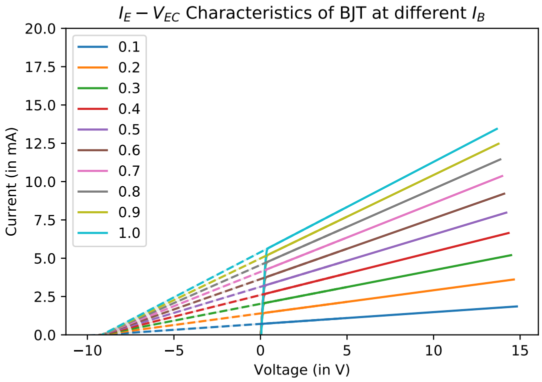
(c) I/V without Early Effect

Figure 2: I_C/V_{CE} characteristics of BJT at different I_B

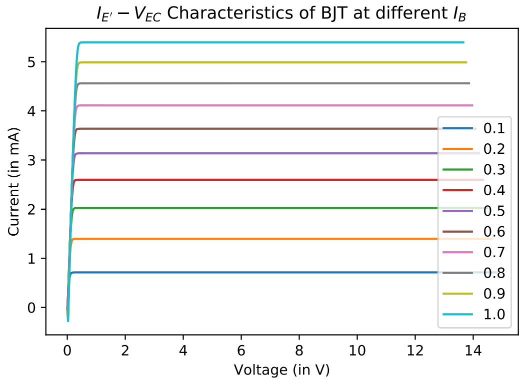
Early voltage $= 72.96948758206197V$
$\alpha = 0.9910952747123186$
$\beta = 116.12236639056505$



(a) I/V



(b) Early Effect

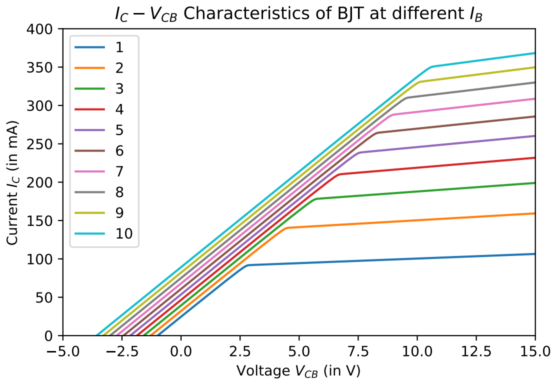


(c) I/V without Early Effect

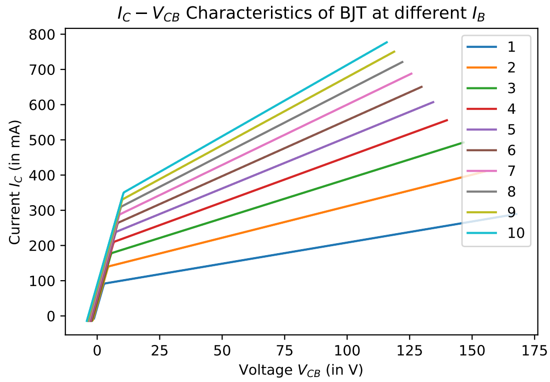
Figure 3: I_E/V_{EC} characteristics of BJT at different I_B (For reverse β)

reverse $\beta = 6.220647099196276$

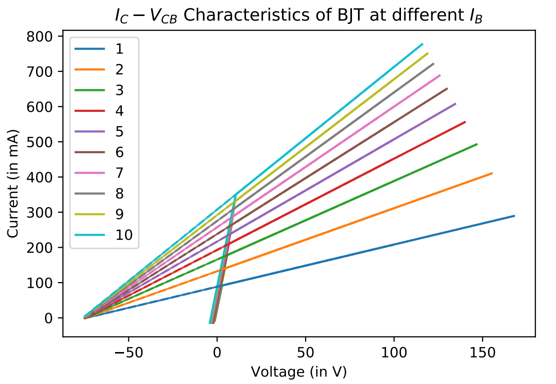
3.1.2 BJT Parameters in CB configuration



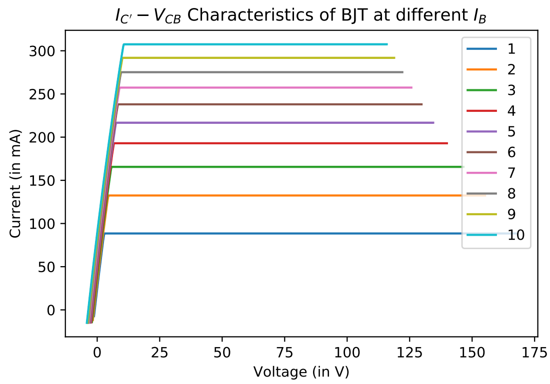
(a) I/V



(b) I/V at Higher Voltage ($\tilde{1}$00V)



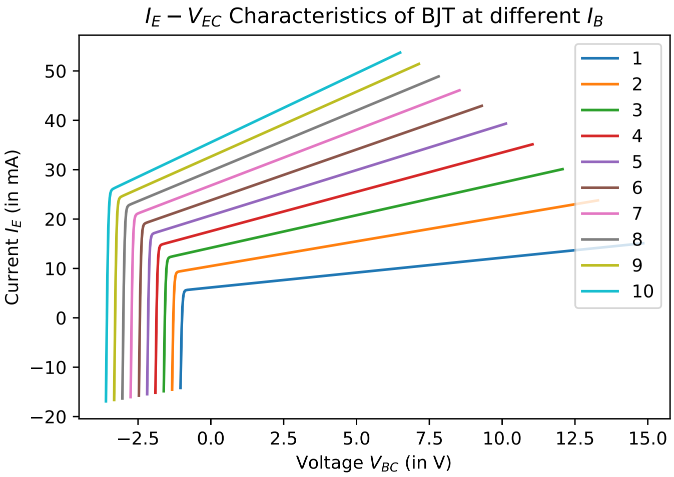
(c) Early Effect



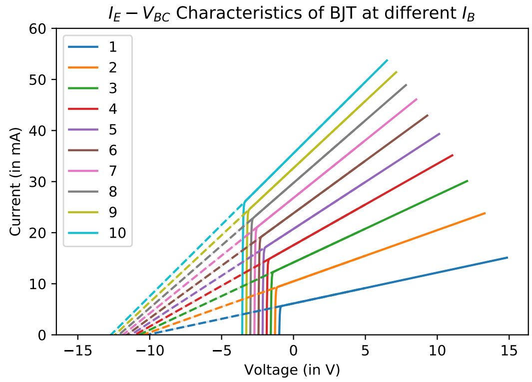
(d) I/V without Early Effect

Figure 4: I_C/V_{CB} characteristics of BJT at different I_B

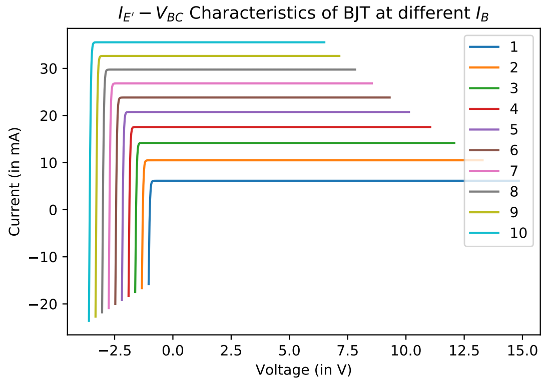
Early voltage = 74.79396681192023
$\alpha = 0.9772565947228588$
$\beta = 47.54815862429751$



(a) I/V



(b) Early Effect

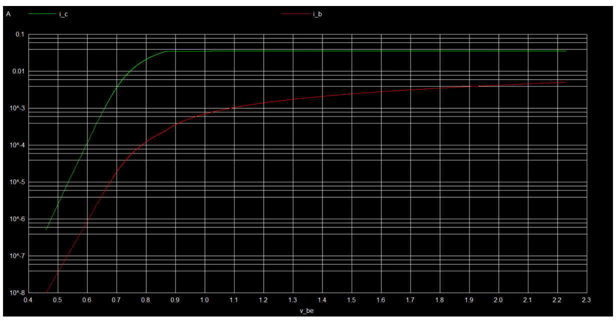


(c) I/V without Early Effect

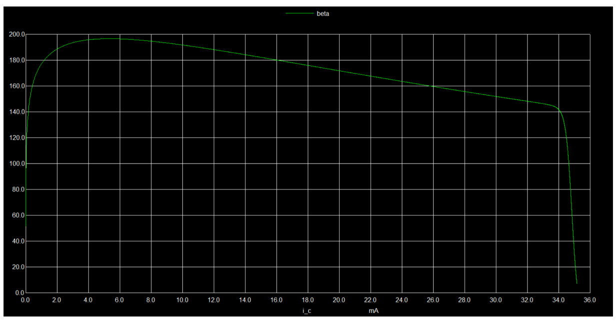
Figure 5: I_E/V_{BC} characteristics of BJT at different I_B

reverse $\beta = 4.335056329929236$

3.1.3 Gummel Plot



(a) $I_C, I_B/V_{BE}$ at a fixed V_{CB}



(b) β_{DC} vs I_C

Figure 6: Gummel Plot

3.1.4 Small Signal Parameters

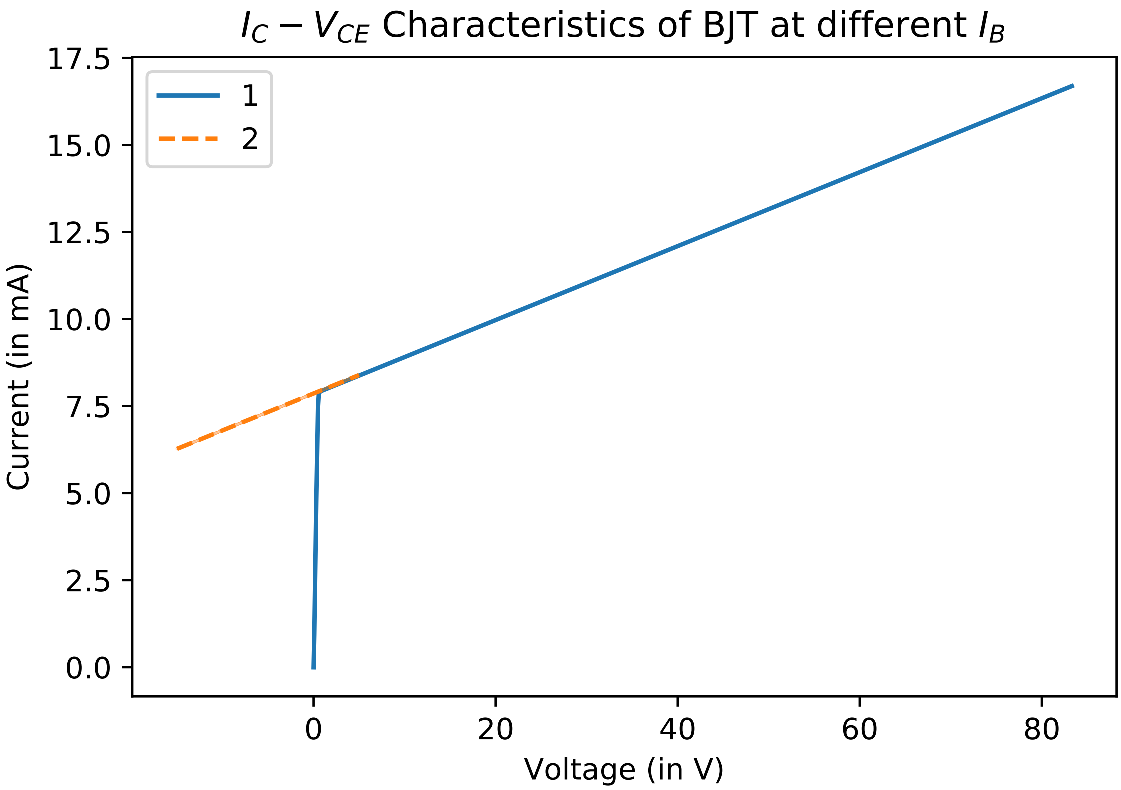
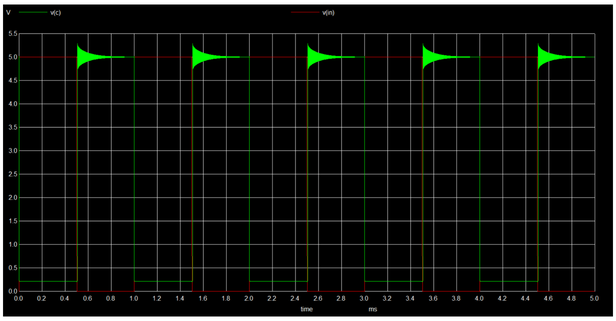


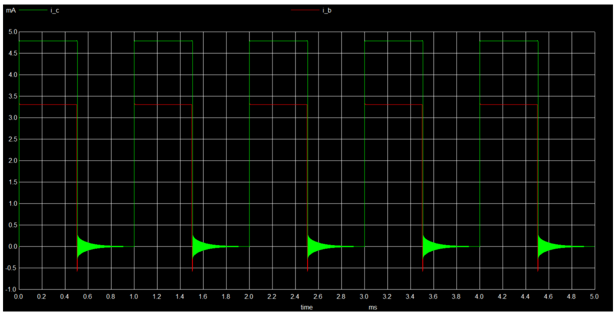
Figure 7: I_C/V_{CE}

$g_m = 1.730769e - 01 = 173m\mho$
$r_{pi} = 6.096300e + 02 = 609\Omega$
$r_o = 1.648099e + 04 = 16.48k\Omega$

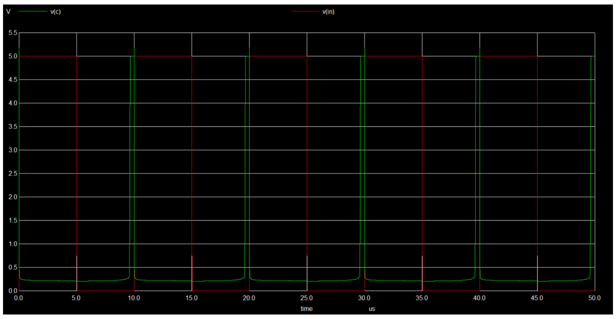
3.2 Switching behaviour



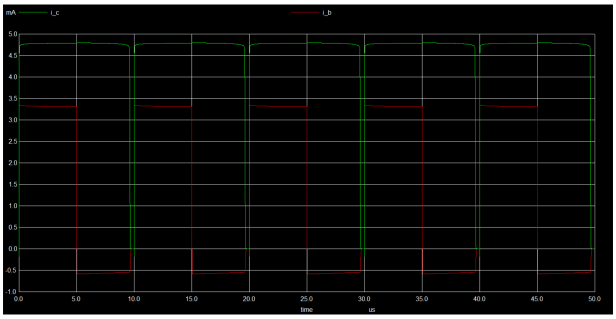
(a) V_{out} vs V_{in} $1KHz$



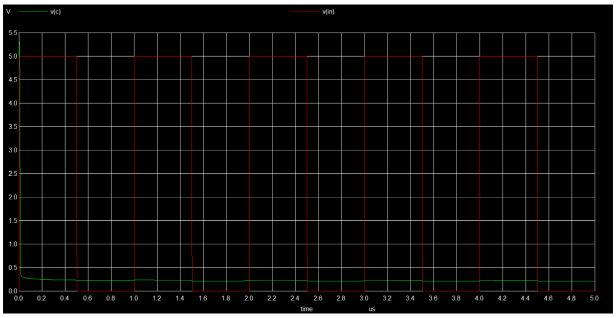
(b) I_C, I_B $1KHz$



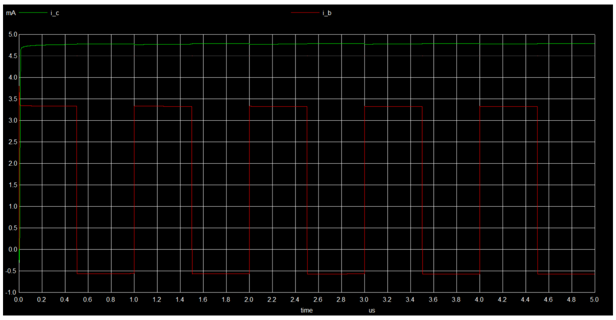
(c) V_{out} vs V_{in} $100KHz$



(d) I_C, I_B $100KHz$



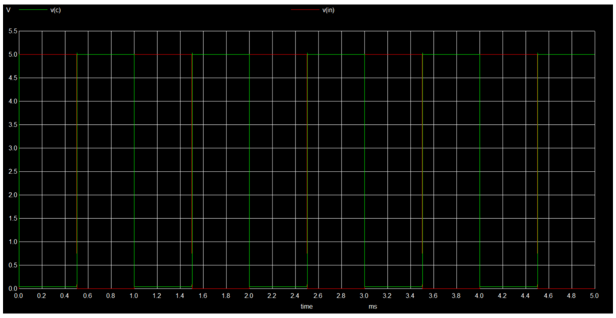
(e) V_{out} vs V_{in} $1MHz$



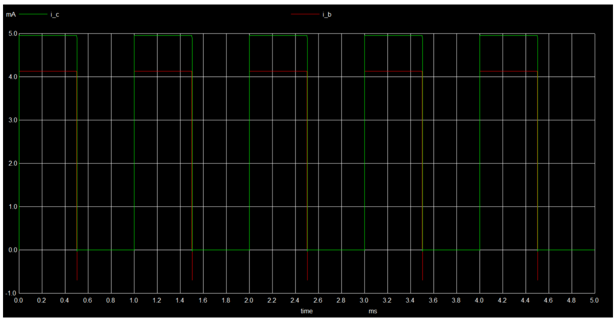
(f) I_C, I_B $1MHz$

Figure 8: BC547a

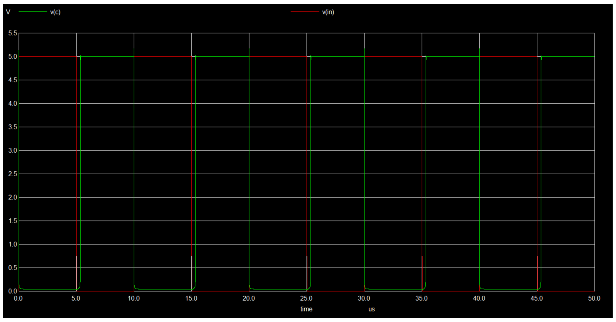
Turn off times
BC547a
$1kHz 5.3\mu s$
$100kHz 4.7815\mu s$
$1MHz No turn off$



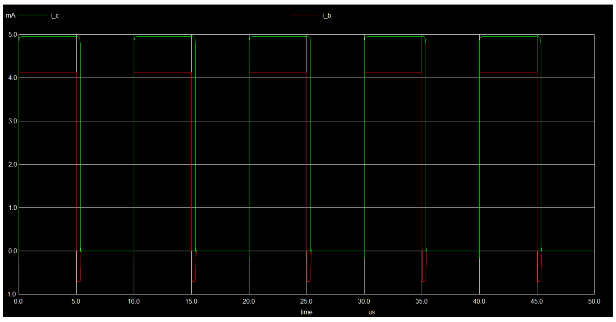
(a) V_{out} vs V_{in} $1KHz$



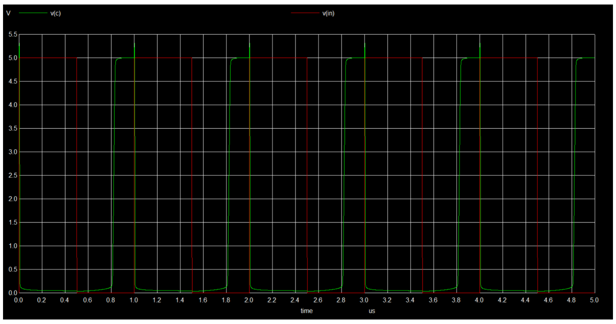
(b) I_C, I_B $1KHz$



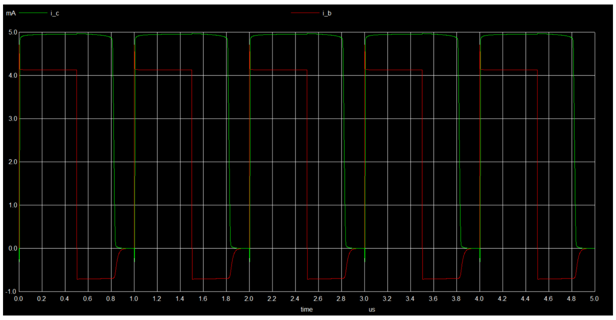
(c) V_{out} vs V_{in} $100KHz$



(d) I_C, I_B $100KHz$



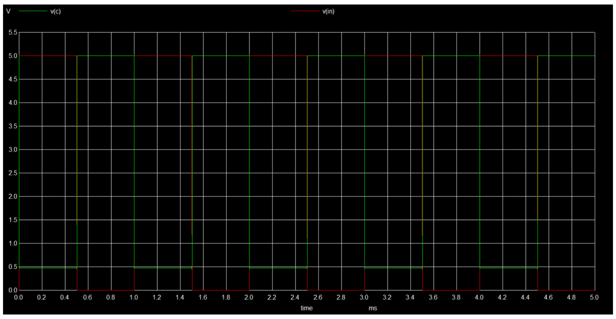
(e) V_{out} vs V_{in} $1MHz$



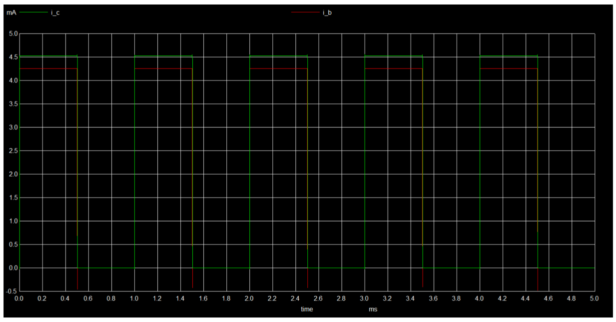
(f) I_C, I_B $1MHz$

Figure 9: 2N3904c

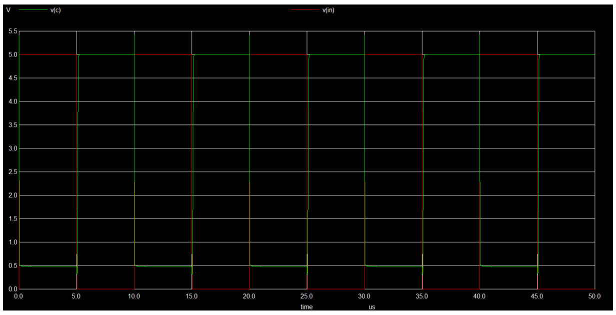
Turn off times
2N3904c
$1kHz 0.36\mu s$
$100kHz 0.4234\mu s$
$1MHz 0.404\mu s$



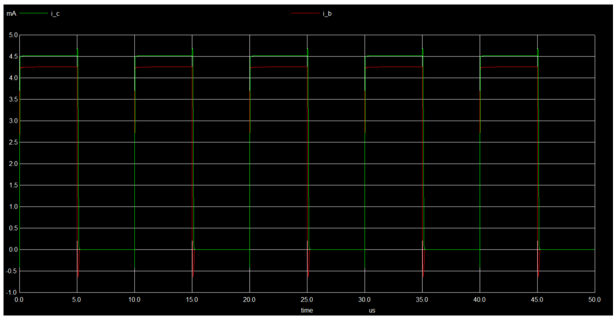
(a) V_{out} vs V_{in} $1KHz$



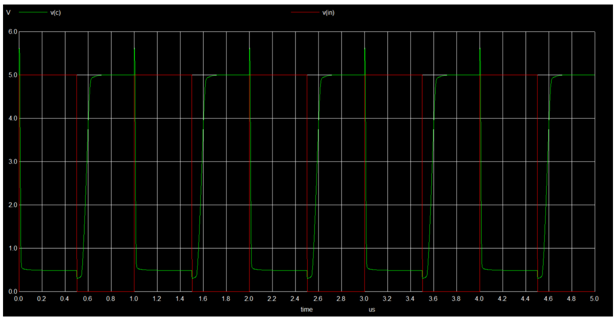
(b) I_C, I_B $1KHz$



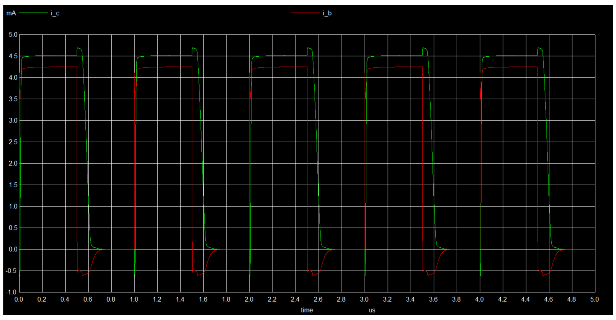
(c) V_{out} vs V_{in} $100KHz$



(d) I_C, I_B $100KHz$



(e) V_{out} vs V_{in} $1MHz$



(f) I_C, I_B $1MHz$

Figure 10: BC547a with Diode

Turn off times
BC547a with Schottky diode
$1kHz 0.36\mu s$
$100kHz 0.42845\mu s$
$1MHz 0.275487\mu s$

3.3 Code snippets

3.3.1 NGSPICE

Note other codes are similar

3.3.1.1 BJT Parameters in CE configuration

```
Param Rathour (190070049), BJT Parameters in CE configuration
.include BC547.txt ; Includes BJT
Q1 c b e bc547a
R1 mid b 470
Ib in2 mid 0.1m
V2 in2 gnd 0
```

```
R2 in1 c 100
V1 in1 gnd 0
Ve e gnd 0
.dc V1 0 15 0.01          ; DC Analysis
.control
run
let I_C = -I(V1)
let V_CE = {V(c) - V(e)}
plot I_C vs V_CE
wrdata 111.txt I_C vs V_CE
.endc
.end
```

3.3.1.2 BJT Parameters in CB configuration

```
Param Rathour (190070049), BJT Parameters in CB configuration
.include BC547.txt ; Includes BJT
Q1 c b e bc547a
R1 mid c 100
Ib gnd b 0
V2 mid gnd 0
V1 e gnd 2
.dc V2 0 200 0.01 Ib 0 10m 1m          ; DC Analysis
.control
run
let I_C = -I(V2)
let V_CB = {V(c) - V(b)}
plot I_C vs V_CB
.endc
.end
```

3.3.1.3 Gummel Plot

```
Param Rathour (190070049), Gummel Plot
.include BC547.txt ; Includes BJT
Q1 c b e bc547a
R1 mid c 100
Ib gnd dummy 0
Vdum dummy b 0
V2 mid gnd 5
V1 e gnd 0
.dc Ib 0.01u 5m 0.1u          ; DC Analysis
* .dc Ib 2u 0.2m 0.1u          ; DC Analysis
* .dc Ib 0.01u 100m 0.1u          ; DC Analysis
.control
run
let I_C = -I(V2)
let I_B = I(Vdum)
let V_BE = {V(b) - V(e)}
let beta = {I_C/I_B}
plot I_C, I_B vs V_BE ylog
plot beta vs I_C ;xlog
.endc
.end
```

3.3.1.4 Small Signal Parameters

```
Param Rathour (190070049), Gummel Plot
.include BC547.txt ; Includes BJT
Q1 c b e bc547a
VCE c e 5
R2 c mid 1k
IC cc mid 4.5m
Vcc cc gnd 10
Vbb bb gnd 5
R1 bb b 100k
Ve e gnd 0
.op
.control
run
let V_t = 26m
let I_C = -I(Vcc)
let I_B = -I(Vbb)
let V_A = {74.16444595209998}
let g_m = I_C/V_t
let beta = I_C/I_B
let ro = V_A/I_C
let rpi = beta/g_m
print g_m, beta, rpi, ro
.endc
.end
* g_m = 1.730769e-01 = 173m mhos
* beta = 1.055129e+02
* rpi = 6.096300e+02 = 609ohms
* ro = 1.648099e+04 = 16.48kohms
```

3.3.1.5 Switching behaviour

```
Param Rathour (190070049), Switching behaviour
.include BC547.txt ; Includes BJT
Q1 c b e bc547a
Rb in b 1k
Vin in gnd pulse(0 5 0 0 0 0.5m 1m)
Vdd dd gnd 5
Rc dd c 1k
Ve e gnd 0
.tran 1u 5m
.control
run
plot V(c) V(in)
let I_C = -I(Vdd)
let I_B = -I(Vin)
plot I_C, I_B
.endc
.end
```

3.3.2 Python Code for Plots and Calculations

Please check the submitted zip file for jupyter notebook with results

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import math
import pandas as pd
import bisect
```

3.3.2.1 BJT Parameters in CE configuration

```python
V_CE = []
I_C = []

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{CE}$ (in V)')
ax1.set_ylabel('Current $I_C$ (in mA)')
ax1.set_title('$I_C-V_{CE}$ Characteristics of BJT at different $I_B$')

for i in range(len(I_B)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab8\\11' +  str(i+1) + '.txt', header = None,
    V_CE.append(data[0])
    I_C.append(data[1])
    ax1.plot(V_CE[i], 1000*I_C[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('11a1.pdf')

y_intercepts = np.zeros(len(I_B))
slope_fit = np.zeros(len(I_B))
for i in range(len(I_B)):
    slope, intercept, r_value, p_value, std_err = stats.linregress(V_CE[i][-2:], I_C[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(I_B))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([0,100])
ax1.set_title('$I_C-V_{CE}$ Characteristics of BJT at different $I_B$')
fig1.set_dpi(150)
intersect = -75
x = np.arange(intersect,5,0.01)
for i in range(len(I_B)):
    ax1.plot(V_CE[i], 1000*I_C[i], '-o', markersize=0.01)
for i in range(len(I_B)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
print(V_A)
print(np.mean(V_A))
ax1.legend(I_B)
fig1.savefig('11a2.pdf')

fig1, ax1 = plt.subplots()
```

```
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([0,100])
ax1.set_title('$I_{C^\prime}-V_{CE}$ Characteristics of BJT at different $I_B$')
I_C1 = [I_C[i]/(1+V_CE[i]/abs(V_A[i])) for i in range(len(I_B))]
for i in range(len(I_B)):
    ax1.plot(V_CE[i], 1000*I_C1[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('11a3.pdf')

beta = np.zeros(len(I_B))
alpha = np.zeros(len(I_B))
for i in range(len(I_B)):
    beta[i] = I_C1[i][-1:]/(I_B[i]*1e-3)
    alpha[i] = beta[i]/(beta[i]+1)
print(beta)
print("beta =", np.mean(beta))
print(alpha)
print("alpha =", np.mean(alpha))

V_EC = []
I_E = []

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{EC}$ (in V)')
ax1.set_ylabel('Current $I_E$ (in mA)')
ax1.set_title('$I_E-V_{EC}$ Characteristics of BJT at different $I_B$')

for i in range(len(I_B)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab8\\11b' +  str(i+1) + '.txt', header = None,
    V_EC.append(data[0])
    I_E.append(data[1])
    ax1.plot(V_EC[i], 1000*I_E[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('11b1.pdf')

y_intercepts = np.zeros(len(I_B))
slope_fit = np.zeros(len(I_B))
for i in range(len(I_B)):
    slope, intercept, r_value, p_value, std_err = stats.linregress(V_EC[i][-2:], I_E[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(I_B))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([0,20])
ax1.set_title('$I_E-V_{EC}$ Characteristics of BJT at different $I_B$')
fig1.set_dpi(150)
intersect = -10
x = np.arange(intersect,5,0.01)
for i in range(len(I_B)):
```

```
    ax1.plot(V_EC[i], 1000*I_E[i], '-o', markersize=0.01)
for i in range(len(I_B)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
print(V_A)
ax1.legend(I_B)
fig1.savefig('11b2.pdf')


fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_title('$I_{E^\prime}-V_{EC}$ Characteristics of BJT at different $I_B$')
I_E1 = [I_E[i]/(1+V_EC[i]/abs(V_A[i])) for i in range(len(I_B))]
for i in range(len(I_B)):
    ax1.plot(V_EC[i], 1000*I_E1[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('11b3.pdf')


rbeta = np.zeros(len(I_B))
for i in range(len(I_B)):
    rbeta[i] = I_E1[i][-1:]/(I_B[i]*1e-3)
print(rbeta)
print("rbeta =", np.mean(rbeta))
```

3.3.2.2 BJT Parameters in CB configuration

```
I_B = np.arange(1, 11, 1)
I_B

V_CB = []
I_C = []

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{CB}$ (in V)')
ax1.set_ylabel('Current $I_C$ (in mA)')
ax1.set_title('$I_C-V_{CB}$ Characteristics of BJT at different $I_B$')

fig2, ax2 = plt.subplots()
ax2.set_xlabel('Voltage $V_{CB}$ (in V)')
ax2.set_ylabel('Current $I_C$ (in mA)')
ax2.set_title('$I_C-V_{CB}$ Characteristics of BJT at different $I_B$')

for i in range(len(I_B)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab8\\12' +  str(i+1) + '.txt', header = None,
    V_CB.append(data[0])
    I_C.append(data[1])
    ax1.plot(V_CB[i], 1000*I_C[i], '-o', markersize=0.01)
    ax2.plot(V_CB[i], 1000*I_C[i], '-o', markersize=0.01)

ax1.set_xlim([-5,15])
ax1.set_ylim([0,400])
ax1.legend(I_B)
fig1.set_dpi(150)
```

```
fig1.savefig('12a11.pdf')
ax2.legend(I_B)
fig2.set_dpi(150)
fig2.savefig('12a12.pdf')

y_intercepts = np.zeros(len(I_B))
slope_fit = np.zeros(len(I_B))
for i in range(len(I_B)):
    slope, intercept, r_value, p_value, std_err = stats.linregress(V_CB[i][-2:], I_C[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(I_B))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_title('$I_C-V_{CB}$ Characteristics of BJT at different $I_B$')
fig1.set_dpi(150)
intersect = -75
x = np.arange(intersect,100,0.01)
for i in range(len(I_B)):
    ax1.plot(V_CB[i], 1000*I_C[i], '-o', markersize=0.01)
for i in range(len(I_B)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
print(V_A)
print(np.mean(V_A))
ax1.legend(I_B)
fig1.savefig('12a2.pdf')

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_title('$I_{C^\prime}-V_{CB}$ Characteristics of BJT at different $I_B$')
I_C1 = [I_C[i]/(1+V_CB[i]/abs(V_A[i])) for i in range(len(I_B))]
for i in range(len(I_B)):
    ax1.plot(V_CB[i], 1000*I_C1[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('12a3.pdf')

beta = np.zeros(len(I_B))
alpha = np.zeros(len(I_B))
for i in range(len(I_B)):
    beta[i] = I_C1[i][-1:]/(I_B[i]*1e-3)
    alpha[i] = beta[i]/(beta[i]+1)
print(beta)
print("beta =", np.mean(beta))
print(alpha)
print("alpha =", np.mean(alpha))

V_BC = []
I_E = []

fig1, ax1 = plt.subplots()
```

```
ax1.set_xlabel('Voltage $V_{BC}$ (in V)')
ax1.set_ylabel('Current $I_E$ (in mA)')
ax1.set_title('$I_E-V_{BC}$ Characteristics of BJT at different $I_B$')

for i in range(len(I_B)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab8\\12b' +  str(i+1) + '.txt', header = None,
    V_BC.append(data[0])
    I_E.append(data[1])
    ax1.plot(V_BC[i], 1000*I_E[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('12b1.pdf')

y_intercepts = np.zeros(len(I_B))
slope_fit = np.zeros(len(I_B))
for i in range(len(I_B)):
    slope, intercept, r_value, p_value, std_err = stats.linregress(V_BC[i][-2:], I_E[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(I_B))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([0,60])
ax1.set_title('$I_E-V_{BC}$ Characteristics of BJT at different $I_B$')
fig1.set_dpi(150)
intersect = -15
x = np.arange(intersect,5,0.01)
for i in range(len(I_B)):
    ax1.plot(V_BC[i], 1000*I_E[i], '-o', markersize=0.01)
for i in range(len(I_B)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
print(V_A)
ax1.legend(I_B)
fig1.savefig('12b2.pdf')

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_title('$I_{E^\prime}-V_{BC}$ Characteristics of BJT at different $I_B$')
I_E1 = [I_E[i]/(1+V_BC[i]/abs(V_A[i])) for i in range(len(I_B))]
for i in range(len(I_B)):
    ax1.plot(V_BC[i], 1000*I_E1[i], '-o', markersize=0.01)
ax1.legend(I_B)
fig1.set_dpi(150)
fig1.savefig('12b3.pdf')


rbeta = np.zeros(len(I_B))
for i in range(len(I_B)):
    rbeta[i] = I_E1[i][-1:]/(I_B[i]*1e-3)
print(rbeta)
```

```
print("rbeta =", np.mean(rbeta))
```

3.3.2.3 Small Signal Parameters

```
data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab8\\3.txt', header = None, skipinitialspace=True,
V_CE = data[0]
I_C = data[1]

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_title('$I_C-V_{CE}$ Characteristics of BJT at different $I_B$')
ax1.plot(V_CE, 1000*I_C, '-o', markersize=0.01)
fig1.set_dpi(150)

slope, intercept, r_value, p_value, std_err = stats.linregress(V_CE[-2:], I_C[-2:])
print(slope, intercept)

ax1.plot(x, 1000*(slope*x+intercept), '--o', markersize=0.01)
V_A = -intercept/slope
print(V_A)
ax1.legend(I_B)
fig1.savefig('31.pdf')
```

4 Experiment completion status

Completed everything in Lab successfully.

5 Questions for reflection

Please, also check by observations section for this.

- At low I_C, due to leakage in the transistor β can be lower. At high I_C, there is β roll off due to high level injection.
- BC547a is worst in switching performance at higher frequencies due to their higher junction capacitance.
- With the addition of Schottky diode, BJTs performance becomes comparable to NMOS. This is done as they have a faster reverse recovery time as seen in previous labs and it will also have less noise.

Overall, the lab was good. Thanks!