

# Quarto Relatório de Lab de Eletrônica 1

Bruno França  
Henrique da Silva

4 de dezembro de 2023

## Sumário

### 1 Introdução

### 2 Análise preliminar

2.1	Análise numérica . . . . .	
2.2	Circuito no LTSpice . . . . .	
2.3	Análise de grandes sinais . . . . .	
2.3.1	Tensão nos transistores . . . . .	
2.3.2	Corrente nos transistores . . . . .	
2.4	Análise de pequenos sinais . . . . .	
2.4.1	Calculando parâmetros . . . . .	
2.4.2	Impedâncias e ganhos . . . . .	

### 3 Medições em laboratório

### 4 Análise dos resultados

4.1	Cálculo do ganho $\beta$ . . . . .	
-----	------------------------------------	--

### 5 Conclusões

### 6 Apêndice

# 1 Introdução

Neste relatório, é explorado o comportamento de um circuito em configuração Darlington, que consiste em um circuito com dois transistores TBJ em um circuito amplificador emissor de base comum com fontes simétricas de tensão. Uma análise numérica do circuito será realizada, e o comportamento do mesmo será observado tanto para pequenos quanto para grandes sinais. Além disso, os resultados obtidos serão comparados com os valores medidos em laboratório.

Todos arquivos utilizados para criar este relatório, e o relatório em si estão em: [https://github.com/Shapis/ufpe\\_ee/tree/main/7thSemester/Eletronica2/](https://github.com/Shapis/ufpe_ee/tree/main/7thSemester/Eletronica2/)

## 2 Análise preliminar

Na análise teórica, utiliza-se o software LTspice para realizar a análise numérica do circuito, tanto para os grandes sinais quanto para os pequenos sinais.

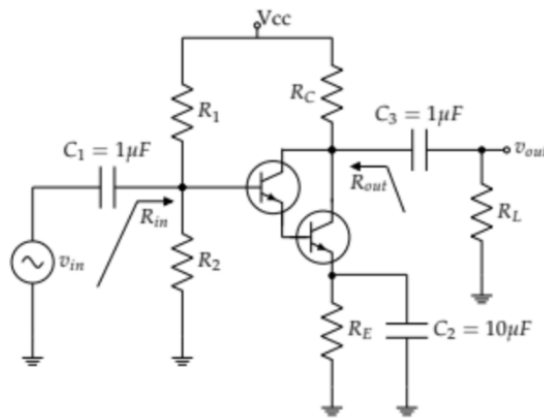


Figura 1: Configuração Darlington (Fonte: o autor)

Figura 1: Circuito em configuração Darlington.

### 2.1 Análise numérica

Nesta etapa, monta-se o circuito no LTSpice e verifica-se as tensões e correntes em pontos críticos específicos do circuito. Com esses dados em mãos, é possível calcular as impedâncias e ganhos associados.

### 2.2 Circuito no LTSpice

Simula-se o circuito no LTSpice, utilizando os valores de componentes do projeto, e em modo *DC operating point* que considera os capacitores do circuito como circuitos em aberto.

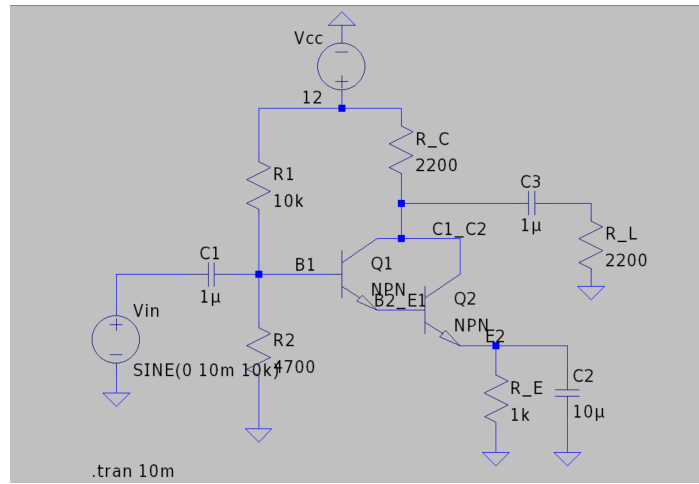


Figura 2: Circuito Darlington simulado no LTSpice.

## 2.3 Análise de grandes sinais

Com o circuito simulado com capacitores em aberto podemos obter os valores das tensões e correntes para grandes sinais. Os valores obtidos estão nas tabelas abaixo.

### 2.3.1 Tensão nos transistores

Tensão	Medida
$V_{C1}$	$6.80V$
$V_{B1}$	$3.84V$
$V_{E1}$	$3.16V$
$V_{C2}$	$6.80V$
$V_{B2}$	$3.16V$
$V_{E2}$	$2.36V$

### 2.3.2 Corrente nos transistores

Corrente	Medida
$I_{C1}$	$23.2\mu A$
$I_{B1}$	$0.232\mu A$
$I_{E1}$	$23.2\mu A$
$I_{C2}$	$2.36mA$
$I_{B2}$	$23.4\mu A$
$I_{E2}$	$2.36mA$

## 2.4 Análise de pequenos sinais

Com os valores da análise de grandes sinais calculados, podemos realizar a análise de pequenos sinais.

### 2.4.1 Calculando parâmetros

Para obter os ganhos globais e as impedâncias do circuito precisamos obter os parâmetros  $g_m$ ,  $r_\pi$ , e  $\beta$  de cada transistor. Para isso, utilizamos as seguintes equações:

$$\begin{aligned}
g_m &= \frac{I_C}{V_T} \\
r_\pi &= \frac{\beta}{g_m} \\
\beta &= \frac{I_C}{I_B}
\end{aligned} \tag{1}$$

E obtemos os seguintes valores:

Parâmetro	Valor
$g_{m1}$	$9.27 * 10^{-4}$
$g_{m2}$	$9.36 * 10^{-2}$
$\beta_1$	100
$\beta_2$	100
$r_{\pi 1}$	$108k\Omega$
$r_{\pi 2}$	$1.07k\Omega$

#### 2.4.2 Impedâncias e ganhos

Com os parâmetros calculados, podemos obter as impedâncias e ganhos do circuito. Para isso, utilizamos as seguintes equações:

$$\begin{aligned}
R_{inB_1} &= r_{\pi 1} + (\beta + 1)r_{\pi 2} \\
R_{in} &= R_1 // R_2 // R_{inB_1} \\
R_{out} &= R_C // R_L \\
A_v &= \frac{-gmR_{out}}{2}
\end{aligned} \tag{2}$$

E obtemos os seguintes valores:

Parâmetro	Valor
$R_{inB_1}$	$216k\Omega$
$R_{in}$	$3.15k\Omega$
$R_{out}$	$1.1k\Omega$
$A_v$	-51.47

## 3 Medições em laboratório

## 4 Análise dos resultados

Os principais resultados obtidos a partir das medições e simulações realizadas no experimento foram analisados, incluindo comparações com os resultados numéricos e as conclusões relevantes.

### 4.1 Cálculo do ganho $\beta$

Calcula-se o  $I_c$  a partir de:

$$I_c = \frac{V_{rc}}{R_c} = \frac{5.87}{470} = 12.5mA \tag{3}$$

Com isto calcula-se o  $I_e$ :

$$I_e = \frac{(V_e - V_{ee})}{R_e} = \frac{(-5.7 - (-8.006))}{178.4} = 12.9mA \tag{4}$$

E temos  $I_b = I_e - I_c = 0.04mA$ .

E por fim o  $\beta$ :

$$\beta = \frac{I_c}{I_b} = \frac{12.5}{0.04} = 31.25 \quad (5)$$

Acredito que a medição do parâmetro  $\beta$  não tenha sido precisa o suficiente, e suspeito que a fonte do problema esteja centralizada na aferição de  $V_{rc}$ .

## 5 Conclusões

Conclui-se que o experimento foi conduzido com sucesso em grande parte, obtendo resultados que se aproximaram das expectativas. No entanto, um ponto crítico que merece atenção é a medição do parâmetro  $\beta$ , a qual não alcançou o nível de precisão desejado. Esse desvio provavelmente se deve a erros de medição relacionados ao  $v_{rc}$ .

Essa imprecisão na determinação de  $\beta$  acabou por inviabilizar uma análise coerente do circuito nas atividades laboratoriais. Isso ocorreu, pois todos os requisitos e cálculos posteriores dependiam diretamente do valor de  $\beta$ .

Portanto, em futuras abordagens experimentais, é crucial aprimorar os procedimentos de medição e garantir uma avaliação mais precisa do  $\beta$  para uma análise mais completa e confiável do desempenho do circuito.

## 6 Apêndice

Abaixo se encontra o código utilizado para a análise numérica do circuito.

```
import matplotlib.pyplot as plt
import sympy as smp
from sympy import *
import numpy as np

# j = smp.symbols('j', imaginary=True)

# Vo, Vi, Vm, V_CO, V_DO, R_m, C, t, R, Vc = smp.symbols(
#     'V_o V_i V_m V_CO V_DO R_m C t R V_c', real=True)

def updateEqs(Vm, V_CO, V_DO, R_m, C, t, R):
    Vo1 = ((V_CO + Vm + V_DO) * np.exp(-t/(R_m * C))) - V_DO
    Vc1 = (V_CO + Vm + V_DO) * np.exp(-t/(R_m * C)) - Vm - V_DO
    Vo2 = (V_CO + Vm) * np.exp(-t/(R * C))
    Vc2 = (V_CO + Vm) * np.exp(-t/(R * C)) - Vm
    return Vo1, Vc1, Vo2, Vc2

# Vo1, Vc1, Vo2, Vc2 = updateEqs(Vm, V_CO, V_DO, R_m, C, t, R)

# estado 1: Vm + Vc0 < - Vd0
# estado 2: Vm + Vc0 > - Vd0

# Exemplo 2

R = float("inf")
# R = 4.7E3
Vm = 5

# Dados
V_DO = 0.5
C = 100E-9
R_m = 330
V_CO = 0

Ts = 500E-6
step = 1E-7
temp = 0

intervalos_tempo = np.arange(0, 10*Ts + step, step)
valores_de_Vo = []
valores_de_Vc = []

for t in intervalos_tempo:
    temp += step
    if temp >= Ts/2:
        Vm = -Vm
        temp = 0

Vo1, Vc1, Vo2, Vc2 = updateEqs(Vm, V_CO, V_DO, R_m, C, step, R)

if (Vm + V_CO) < -V_DO: # estado 1
    Vo = Vo1
    Vc = Vc1
else: # estado 2
    Vo = Vo2
    Vc = Vc2
```

```
valores_de_Vo.append(Vo)
valores_de_Vc.append(Vc)
V_C0 = Vc

# smp.pprint(Vo)

# Plotando os graficos

plotH1 = valores_de_Vo
plotH2 = valores_de_Vc

fig, ax = plt.subplots()

ax.plot(intervalos_tempo, plotH1, color='blue', label='Exemplo 1')
ax.plot(intervalos_tempo, plotH2, color='orange', label='Exemplo 2')
ax.legend(['Vo', 'Vc'])
plt.xlabel('Tempo (s)')
plt.ylabel('Tensao (V)')
plt.title('Grafico de $V_o(t)$ no intervalo de 0 a $10 \cdot T_s$')
plt.show()

print(min(valores_de_Vo))
print(max(valores_de_Vo))
```