**UNIVERSIDADE FEDERAL DA BAHIA** 17/07/2025
**PPGEE0017 - PROJETO DE CIRCUITOS INTEGRADOS** 2025.1
Aluno(s): André Paiva Conrado Rodrigues (N = 2)

# TRABALHO 2 - AMPLIFICADOR OTA MILLER

**Tabela 1:** Correntes de polarização e níveis de inversão dos transistores

| | $I_D(\mu A)$ | $I_f$ |
|---|---|---|
| $M_{1,2}$ | 9,22461 | 5,221 |
| $M_{3,4}$ | 9,22461 | 18,630 |
| $M_5$ | 18,4492 | 52,208 |
| $M_6$ | 66,0956 | 8,343 |
| $M_7$ | 66,0956 | 93,519 |

**Tabela 2:** Parâmetros de desempenho do OTA Miller

| | **Calculado** | **Simulado** |
|---|---|---|
| $\|A_d(0)\|$ (dB) | <60,915 | 52,538 |
| GBW (Hz) | 52065946 | 11937766 |
| PM (Graus) | 83,241 | 64,995 |
| CMR- | 0,163 | -0,599 |
| CMR+ | 0,592 | 0,444 |
| $V_{OUT(MIN)}$ | -0,269 | -0,579 |
| $V_{OUT(MAX)}$ | 0,443 | 0,591 |
| SR ($V/\mu s$) | 9,225 | (-) 9,680 |
| | | (+) 9,604 |
| \|CMRR(0)\| (dB) | | 68,705 |
| \|PSRR-(0)\| (dB) | | 48,472 |
| \|PSRR+(0)\| (dB) | | 71,262 |

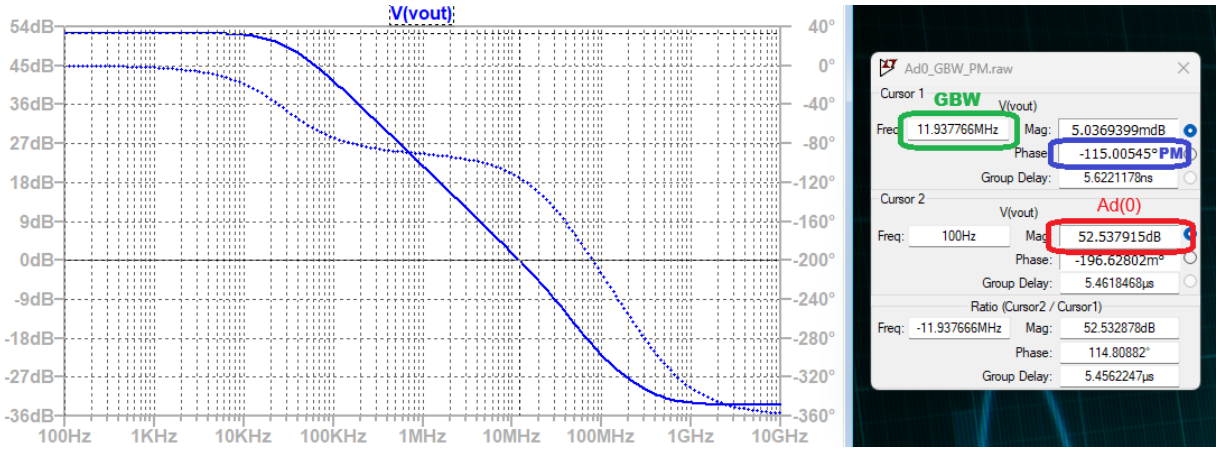


**Figura 1:** Diagrama de Bode para magnitude e fase do ganho de tensão diferencial.

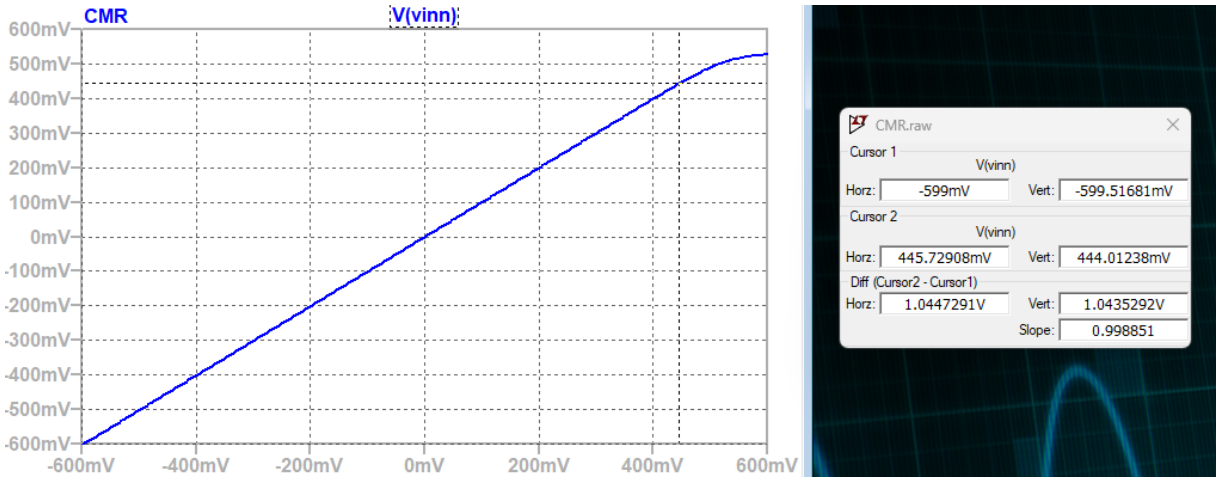


**Figura 2:** Diagrama de simulação DC para análise de CMR.

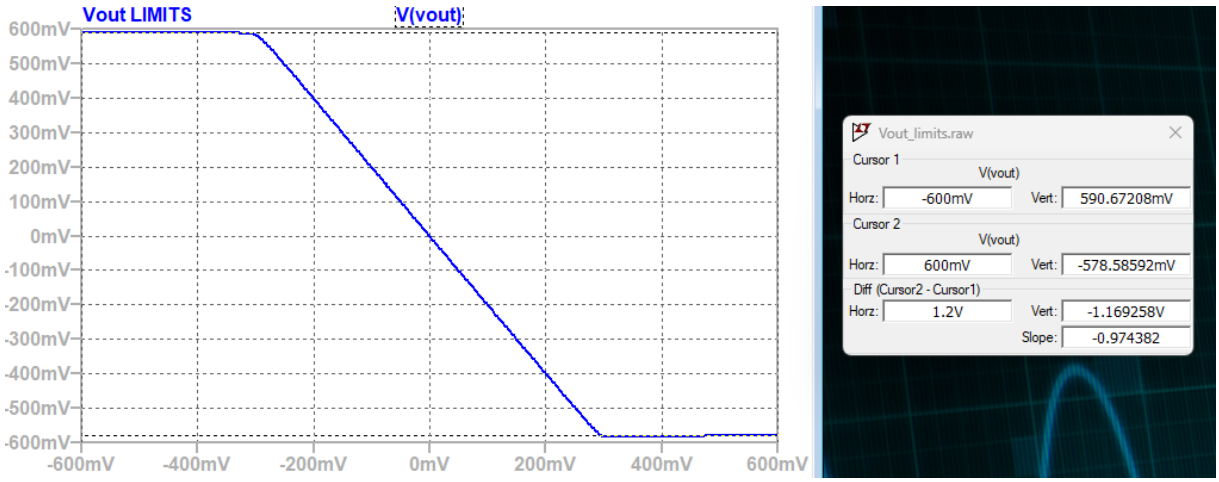


**Figura 3:** Diagrama de simulação DC para análise de excursão de saída.

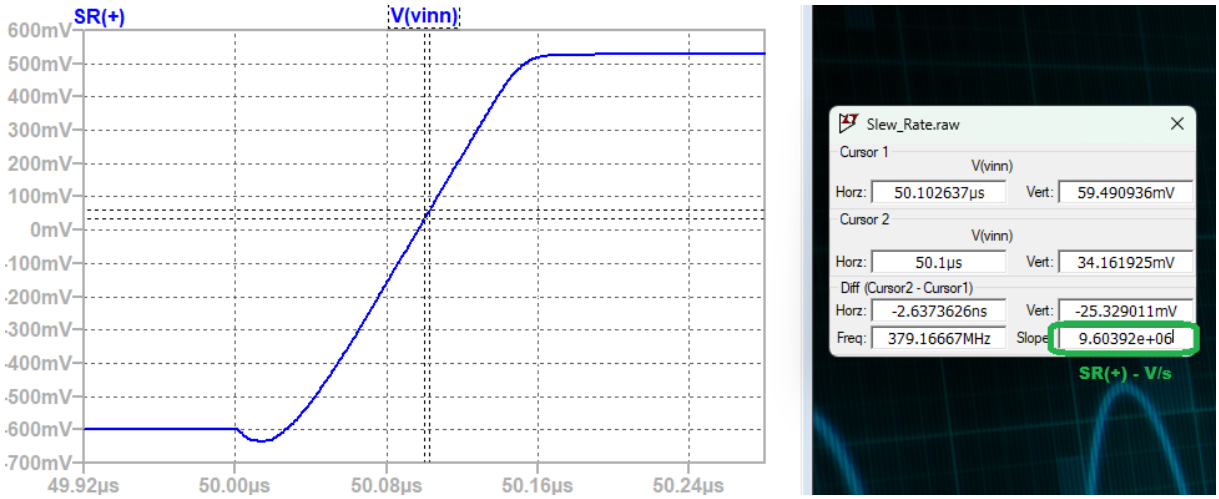


**Figura 4:** Diagrama para análise do *slew rate* (borda de subida).

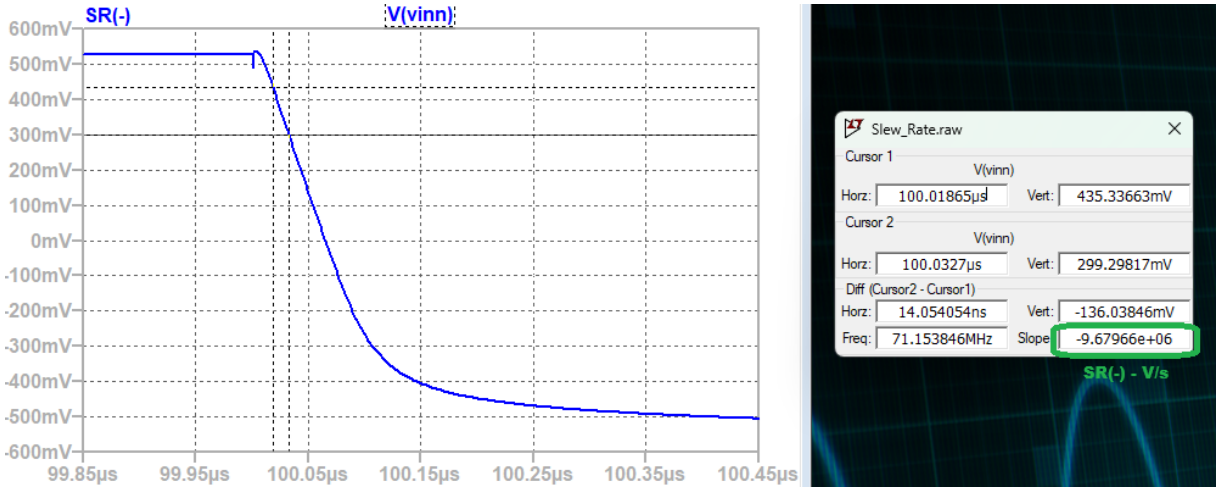


**Figura 5:** Diagrama para análise do *slew rate* (borda de descida).

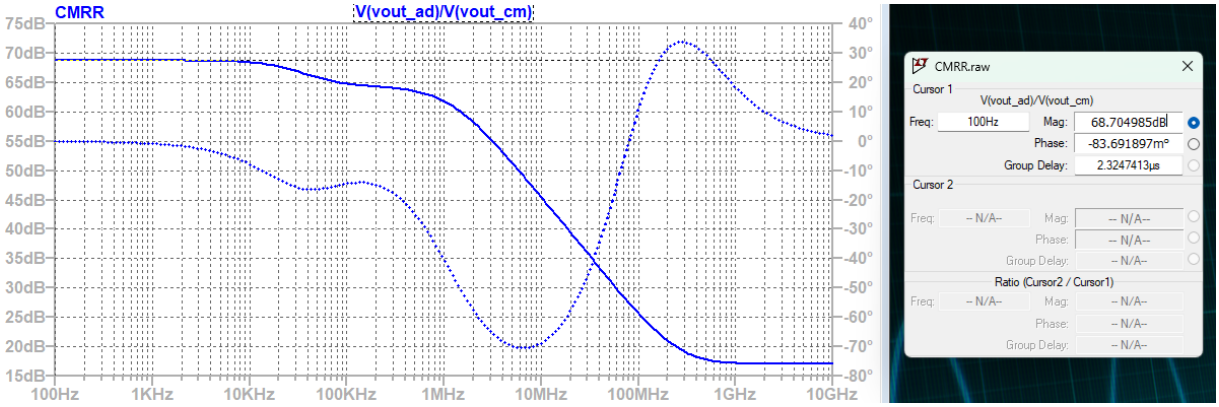


**Figura 6:** Diagrama de magnitude do CMRR.

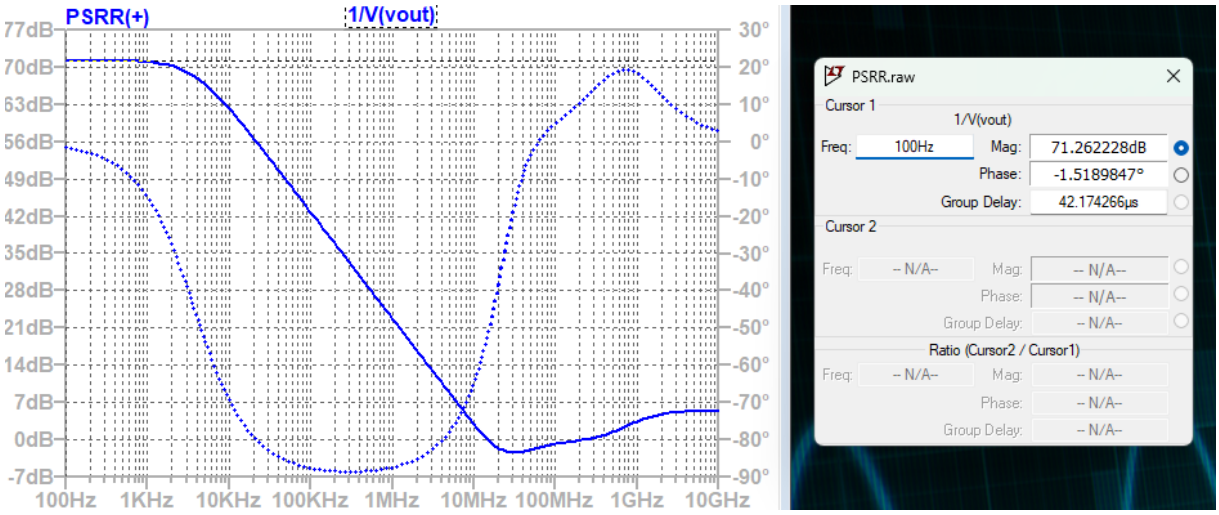


**Figura 7:** Diagrama de magnitude do PSRR(+).

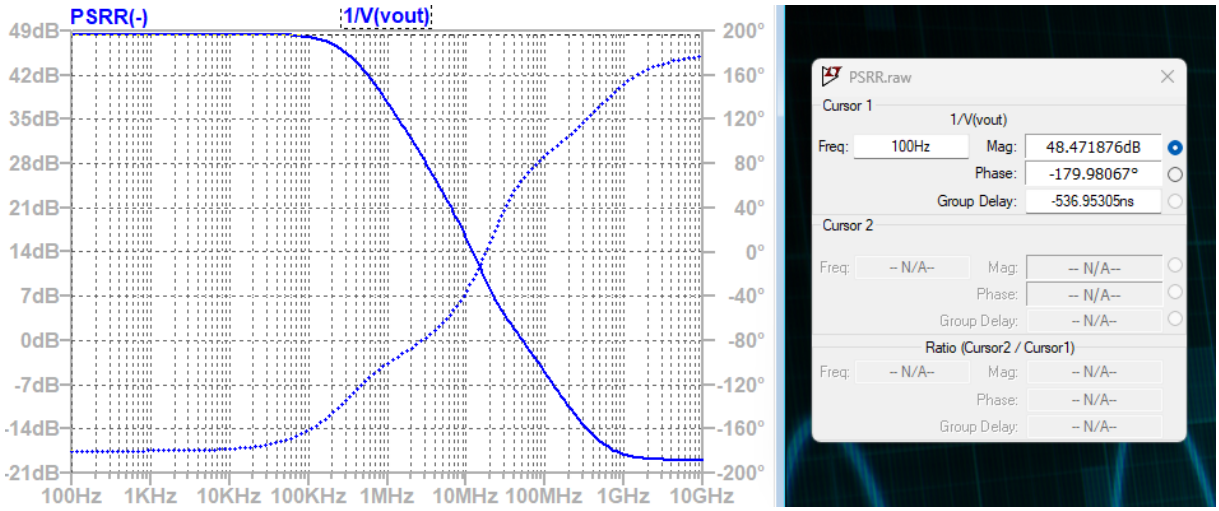


**Figura 8:** Diagrama de magnitude do PSRR(-).

# Trabalho 2 - Memorial de Cálculo

**Disciplina:** PPGEE0017 - Projeto de Circuitos Integrados - PPGEEC/UFBA

**Aluno:** André Paiva Conrado Rodrigues (N=2)

## 1 - Escopo do trabalho

### Especificações do circuito

Extração de características de amplificador OTA Miller com os seguintes parâmetros:

| Parâmetro | Valor | Unidade |
|---|---|---|
| $W_{1,2}$ | 10 | $\mu$m |
| $L_{1,2}$ | 2 | $\mu$m |
| $W_{3,4}$ | 20 | $\mu$m |
| $L_{3,4}$ | 2 | $\mu$m |
| $W_5$ | 1 | $\mu$m |
| $L_5$ | 1 | $\mu$m |
| $W_6$ | 80 | $\mu$m |
| $L_6$ | 0,5 | $\mu$m |
| $W_7$ | 40 | $\mu$m |
| $L_7$ | 20 | $\mu$m |
| $C_L$ | 0,2 | pF |
| $C_M$ | 2,0 | pF |
| $V_{GG5}$ | -0,1 | V |
| $V_{DD}$ | 0,6 | V |
| $V_{SS}$ | -0,6 | V |

## Parâmetros da Tecnologia

| Símbolo | Definição | Unidade | Canal N | Canal P |
|---|---|---|---|---|
| $I_{SQ}$ | Corrente de norm. (quadrado) | nA | 353,38 | 49,514 |
| $n$ | Fator de rampa | - | 1,3790 | 1,2952 |
| $\phi_t$ | Potencial Termodinâmico | mV | 26 | 26 |
| $V_{T0}$ | Tensão de limiar no equilíbrio | V | 0,3618 | -0,3554 |
| $U_{CRIT}$ | Campo Elétrico Crítico | V/m | $1,712 \cdot 10^{6}$ | $6,960 \cdot 10^{6}$ |
| $D_{SUB}$ | Parâmetro de modelo DIBL | - | $4,057 \cdot 10^{-6}$ | $3,781 \cdot 10^{-3}$ |
| $\eta_0$ | Parâmetro de modelo DIBL | - | $2,750 \cdot 10^{-6}$ | $1,151 \cdot 10^{-3}$ |
| $I_{t0}$ | Parâmetro de modelo DIBL | m | $3,1116 \cdot 10^{-8}$ | $3,1116 \cdot 10^{-8}$ |
| $\lambda$ | Parâmetro de modelo CLM | - | 1,9784 | 0,8847 |
| $DL$ | Encurtamento do canal | nm | 10,4 | 14,8 |
| $DW$ | Estreitamento do canal | nm | 0,24 | 0 |

In [1]:
```
# Importação de bibliotecas
import numpy as np
```

In [2]:
```
# Especificações do amplificador OTA Miller
W_12 = 10e-6;    L_12 = 2e-6
W_34 = 20e-6;    L_34 = 2e-6
W_5 = 1e-6;      L_5 = 1e-6
W_6 = 80e-6;     L_6 = 0.5e-6
W_7 = 40e-6;     L_7 = 20e-6
C_L = 0.2e-12
C_M = 2e-12
VGG5 = -0.1
VDD = 0.6
VSS = -0.6
```

In [3]:
```
# Canal dos transistores
pol_M12 = 'n'
pol_M34 = 'p'
pol_M5 = 'n'
pol_M6 = 'p'
pol_M7 = 'n'
```

In [4]:
```
# Parâmetros da tecnologia
Isq_p = 49.514e-9;      Isq_n = 353.38e-9
N_p = 1.2952;           N_n = 1.3790
VT0_p = -0.3554;        VT0_n = 0.3618
Phi_t = 26e-3
Ucrit_p = 6.96e6;       Ucrit_n = 1.712e6
Dsub_p = 3.781e-3;      Dsub_n = 4.057e-6
Eta0_p = 1.151e-3;      Eta0_n = 2.75e-6
It0 = 3.1116e-8
lambda_p = 0.8847;      lambda_n = 1.9784
DL_p = 14.8e-9;         DL_n = 10.4e-9
DW_p = 0;               DW_n = 0.24e-9
```

## 2 - Medições em simulação

In [5]:
```
# Correntes de dreno medidas em simulação
ID_M12 = 9.22461e-06 # Medido com sinal positivo
ID_M34 = 9.22461e-06 # Medido com sinal negativo
ID_M5 = 1.84492e-05 # Medido com sinal positivo
ID_M6 = 6.60956e-05 # Medido com sinal negativo
ID_M7 = 6.60956e-05 # Medido com sinal positivo

# Tensão de offset
Voffset = -259.17e-6
```

## 3 - Cálculo dos níveis de inversão

In [6]:
```
# Função para cálculo da corrente de normalização
def is_calc(W, L, mos_pol):
    if(mos_pol == 'p' or mos_pol == 'P'):
        return Isq_p*W/L
    elif(mos_pol == 'n' or mos_pol == 'N'):
        return Isq_n*W/L
    else:
        return -np.inf

# Função para cálculo de nível de inversão (sat.)
def if_calc(Id, W, L, mos_pol):
    if(mos_pol == 'p' or mos_pol == 'P' or mos_pol == 'n' or mos_pol == 'N'):
        return Id/is_calc(W, L, mos_pol)
    else:
        return -np.inf
```

In [7]:
```
If_M12 = if_calc(ID_M12, W_12, L_12, pol_M12)
If_M34 = if_calc(ID_M34, W_34, L_34, pol_M34)
If_M5 = if_calc(ID_M5, W_5, L_5, pol_M5)
If_M6 = if_calc(ID_M6, W_6, L_6, pol_M6)
If_M7 = if_calc(ID_M7, W_7, L_7, pol_M7)

print(f"I_F\nM1,2: {If_M12:.3f} \nM3,4: {If_M34:.3f} \nM5:   {If_M5:.3f}")
print(f"M6:   {If_M6:.3f}   \nM7:   {If_M7:.3f}")
```

```
I_F
M1,2: 5.221
M3,4: 18.630
M5:   52.208
M6:   8.343
M7:   93.519
```

| | $I_D\,(\mu A)$ | $I_f$ |
|---|---|---|
| $M_{1,2}$ | 9,22461 | 5,221 |
| $M_{3,4}$ | 9,22461 | 18,630 |
| $M_5$ | 18,4492 | 52,208 |
| $M_6$ | 66,0956 | 8,343 |
| $M_7$ | 66,0956 | 93,519 |

# 4 - Cálculo dos parâmetros de desempenho

In [8]:
```
# Cálculo da tensão de early mínima
def early(L, mos_pol):
    if(mos_pol == 'p' or mos_pol == 'P'):
        L_eff = L - (2*DL_p)
        theta_dibl = np.exp(-Dsub_p*L/(2*It0)) + (2*np.exp(-Dsub_p*L/It0))
        sig = theta_dibl * Eta0_p
        Va_min = ((L_eff*Ucrit_p) + Phi_t)/((sig/N_p) + lambda_p)
    elif(mos_pol == 'n' or mos_pol == 'N'):
        L_eff = L - (2*DL_n)
        theta_dibl = np.exp(-Dsub_n*L/(2*It0)) + (2*np.exp(-Dsub_n*L/It0))
        sig = theta_dibl * Eta0_n
        Va_min = ((L_eff*Ucrit_n) + Phi_t)/((sig/N_n) + lambda_n)
    else:
        Va_min = -np.inf
    return Va_min
```

In [9]:
```
# Cálculo da transcondutância de dreno
def g_md(Id, L, mos_pol):
    if(mos_pol == 'p' or mos_pol == 'P' or mos_pol == 'n' or mos_pol == 'N'):
        return Id/early(L, mos_pol)
    else:
        return -np.inf
```

In [10]:
```
# Cálculo da transcondutância de porta
def g_mg(Id, W, L, mos_pol):
    i_s = is_calc(W, L, mos_pol)
    i_f = if_calc(Id, W, L, mos_pol)
    if(mos_pol == 'p' or mos_pol == 'P'):
        return ((2*i_s)/(N_p*Phi_t))*(np.sqrt(1+i_f) - 1)
    elif(mos_pol == 'n' or mos_pol == 'N'):
        return ((2*i_s)/(N_n*Phi_t))*(np.sqrt(1+i_f) - 1)
    else:
        return -np.inf
```

In [11]:
```
# Função para determinar f(i_f)
def f_if(i_f):
    return np.sqrt(i_f + 1) - 2 + np.log(np.sqrt(i_f + 1) - 1)
```

In [12]:
```
# Transcondutâncias úteis
gmg_M2 = g_mg(ID_M12, W_12, L_12, pol_M12)
gmg_M6 = g_mg(ID_M6, W_6, L_6, pol_M6)
gmd_M2 = g_md(ID_M12, L_12, pol_M12)
gmd_M4 = g_md(ID_M34, L_34, pol_M34)
gmd_M6 = g_md(ID_M6, L_6, pol_M6)
gmd_M7 = g_md(ID_M7, L_7, pol_M7)

print(f"G_MG \nM2: {gmg_M2:.3e} \nM6: {gmg_M6:.3e} \n")
print(f"G_MD \nM2: {gmd_M2:.3e} \nM4: {gmd_M4:.3e} \nM6: {gmd_M6:.3e} \nM7: {gmd_M7:
```

```
G_MG
M2: 1.473e-04
M6: 9.677e-04

G_MD
M2: 5.345e-06
M4: 5.954e-07
M6: 1.777e-05
M7: 3.820e-06
```

In [13]:
```python
# Ganho diferencial em baixas frequências
Ad0 = (gmg_M2*gmg_M6)/((gmd_M2 + gmd_M4)*(gmd_M6 + gmd_M7))
Ad0_dB = 20*np.log10(Ad0)

# Produto Ganho-Largura de Banda
GBW = gmg_M2/(np.sqrt(2)*C_M) # Hz

# Margem de fase
omg_p2 = gmg_M6/C_L; omg_z = gmg_M6/C_M
PM = ((np.pi/2) - np.arctan(GBW/omg_p2) - np.arctan(GBW/omg_z))*180/np.pi

# Slew Rate
SR = min((ID_M5/C_M), (ID_M7/C_L))*1e-6 # V/us

# Faixa de modo comum
CMR_p = N_n*(VDD - (N_p*Phi_t*f_if(If_M34)) + VT0_p + (Phi_t*f_if(If_M12)) - VSS - (
CMR_n = N_n*Phi_t*(f_if(If_M12) + np.sqrt(If_M5+1) + 3) + VSS + VT0_n

# Limites da tensão de saída
Vout_min = Phi_t*(np.sqrt(1 + If_M7)+3) + VSS
Vout_max = VDD - Phi_t*(np.sqrt(1 + If_M6)+3)

print(f" Ad0 = {Ad0_dB:.3f} dB \n GBW = {(GBW):.0f} Hz")
print(f"  PM = {PM:.3f}° \n  SR = {SR:.3f} V/us")
print(f"CMR+ = {CMR_p:.3f} \nCMR- = {CMR_n:.3f}")
print(f"{Vout_min:.3f} <= Vout <= {Vout_max:.3f}")
```

```
 Ad0 = 60.915 dB
 GBW = 52065946 Hz
  PM = 83.241°
  SR = 9.225 V/us
CMR+ = 0.592
CMR- = 0.163
-0.269 <= Vout <= 0.443
```