FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

EDA/CAD para Nanoelectrónica

1º Relatório prático ref. ano 2015-2016

Docente: Profª Dra. Maria Helena Fino

As equações devem ser numerada em vez de terem legenda
As legendas das tabelas são colocadas acima destas

Ver comentários no relatório

15

Elaborado pelos alunos de MIEEC:

António João Marques de Andrade Pereira	39971
Filipe Miguel Aleixo Perestrelo	39656
Silvana Regina Ferreira de Oliveira Costa	30159

Índices

Índice Geral

Índice de Tabelas

Índice de Figuras

Índice de Equações

Objectivos:

Este trabalho tem como objectivo o estudo das limitações do modelo de Schokley na caracterização de transístores NMOS em tecnologias sub-micrométricas com o auxilio de ferramentas importantes como o software Cadence para dimensionamento e simulação, E o software Matlab para determinação e cálculos dos parâmetros.

Considerando como objetivo do estudo é a sensibilização para as limitações do escolhido, assim como a determinação dos parâmetros para um modelo em que a variação da corrente *Id* é do tipo:

Na zona de saturação

$$Id(V_{GS}) = K\,(V_{GS} - V_T)^n$$

Equation 1 : Equação da Corrente ID

Para a realização deste trabalho devem ser considerados dois transístores com a mesma relação W/L, como se indica na tabela 1:

	Modelo	W	L
Nmos 1	N_12_LLHVT	4 µ	2u
Nmos 2		800n	400n

Table 1: Relação W/L dos transístores NMOS

Este trabalho é constituido por 3 fases que serao descritas mais adiante.

Introdução Teórica

O Transístor MOS

Os transístores de efeito de campo (Field Effect Transistor, FET) MOS (Metal-Óxido-Semicondutor), logo MOSFETs, assim como os transístores bipolar de junção (TBJ), são dispositivos semicondutores de três terminais. Os primeiros ocupam menor area, apresentam uma resistência de entrada praticamente infinita, funcionam melhor como interruptores e permitem realizar circuitos digitais com menor consumo.São constituídos por um material semicondutor do tipo *p*, no qual se encontram duas regiões do tipo *n*, designadas por dreno (Drain) e fonte (Source).

Nos transístores NMOS, por exemplo, podem operar em três zonas de funcionamento distintas, dependendo das tensões com que está polarizado.

1. Zona de Corte

Não tem corrente a passar pelo dreno. Um transístor se encontra em zona de corte quando:

$$V_{GS} < V_T$$

$$V_{GD} < V_T$$

$$I_D = I_S = I_G = 0$$

Equation 2: Condições em zona de corte

2. Zona de Saturação

Encontra-se activo, com a corrente nula na porta (do inglês “gate”) em todas as zonas ($I_G = 0$). Um transístor se encontra em zona de saturação quando cumpre as seguintes condições:

$$V_{GS} \geq V_T$$

$$V_{GD} \geq V_T$$

$$V_{DS} \geq V_{GS} - V_T$$

$$I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L} [(V_{GS} - V_T)^2] * (1 + \lambda V_{DS})$$

Equation 3: Condições em zona de saturação

3. Zona de Tríodo

Assim como na saturação, também encontra-se activo, mas com um ganho menor do que quando está na zona de saturação. Observa-se um canal entre a fonte e o dreno. Um transístor encontra-se em zona de tríodo quando cumpre as seguintes condições:

$$V_{GS} \geq V_T$$

$$V_{GD} < V_T$$

$$V_{DS} < V_{GS} - V_T$$

$$I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - {V_{DS}}^2 \right] * (1 + \lambda V_{DS})$$

Equation 4: Condições em zona de tríodo

Para um transístor PMOS, altera-se nas equações acima, os parâmetros V_{GS} para V_{SG} e V_{DS} para V_{SD}.

O modelo de Schokley

Já tinha sido introduzido na explicação anterior

O modelo de Schokley, usualmente conhecido por modelo quadrático, dada a variação quadrática da corrente de dreno, I_D, com a tensão entre a porta e a fonte, em inglês V_{GS}, na zona de saturação, tornou-se uma aproximação imprecisa para tecnologias de dimensões reduzidas.

Sendo definido pelas seguintes expressões e condições:

$$I_D = \begin{cases} 0, & Para \quad V_{GS} \leq V_T \\ I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - {V_{DS}}^2 \right] * (1 + \lambda V_{DS}), & Se\ V_{DS} < V_{GS} - V_T \\ I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_T)^2 \right] * (1 + \lambda V_{DS}), & Se\ V_{DS} \geq V_{GS} - V_T \end{cases}$$

Equation 5: Expressões e condições do modelo de Schokley

O modelo n-power

Como descrito acima, para tecnologias com dimensões reduzidas, o modelo de Shockley, tornou-se uma aproximação imprecisa. No entanto, o modelo de Shockley é utilizado no dimensionamento analítico de circuitos MOS por ser um modelo simples, muitas fórmulas terem sido definidas com base neste modelo e as fórmulas derivadas serem frequentemente usadas nos programas de software para dimensionamento de circuitos.

Contudo, o modelo de Shockley não reproduz a característica tensão-corrente dos transístores mais recentes pelo facto destes apresentarem dimensões significativamente reduzidas.

Sendo assim, desenvolveu-se um modelo que, basicamente, é uma extensão da zona de saturação do modelo quadrático de Shockley. Este modelo, designado por modelo n-power, é utilizado no dimensionamento de circuitos. É relativamente simples na aplicação matemática que permite utilizações tanto em simulações como analiticamente, também tem a capacidade de prevenção do comportamento do circuito nas regiões em que possui um tamanho reduzido.

Se tivermos uma configuração de um grau de complexidade maior, este modelo só funcionará para simulações, uma vez que analiticamente seria difícil.

Temos abaixo o modelo n-power descrevendo melhor as curvas I_D e apresentado pelas equações:

$$I_D = \begin{cases} 0, & Para\ \ V_{GS} \leq V_T \\ I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L}[(V_{GS} - V_T)^n] * (1 + \lambda V_{DS}), & Se\ \ V_{GS} \geq V_T\ e\ V_{DS} \geq (V_{GS} - V_T) \\ I_D = \frac{1}{2} * \mu_n C_{ox} \frac{W}{L}[(V_{GS} - V_T)^{0,5n} - 0,5 {V_{DS}}^n] * (1 + \lambda V_{DS}), & Se\ \ V_{GS} \geq V_T\ e\ V_{DS} \leq (V_{GS} - V_T) \end{cases}$$

Equation 6: Equações e condições modelo n-power

Fases do trabalho

Este trabalho foi dividido em 3 fases:

Fase 1: Determinação por utilização do Cadence, das características ID(VDS,VGS).

Através do software cadence, desenvolvemos o seguinte esquemático do circuito:

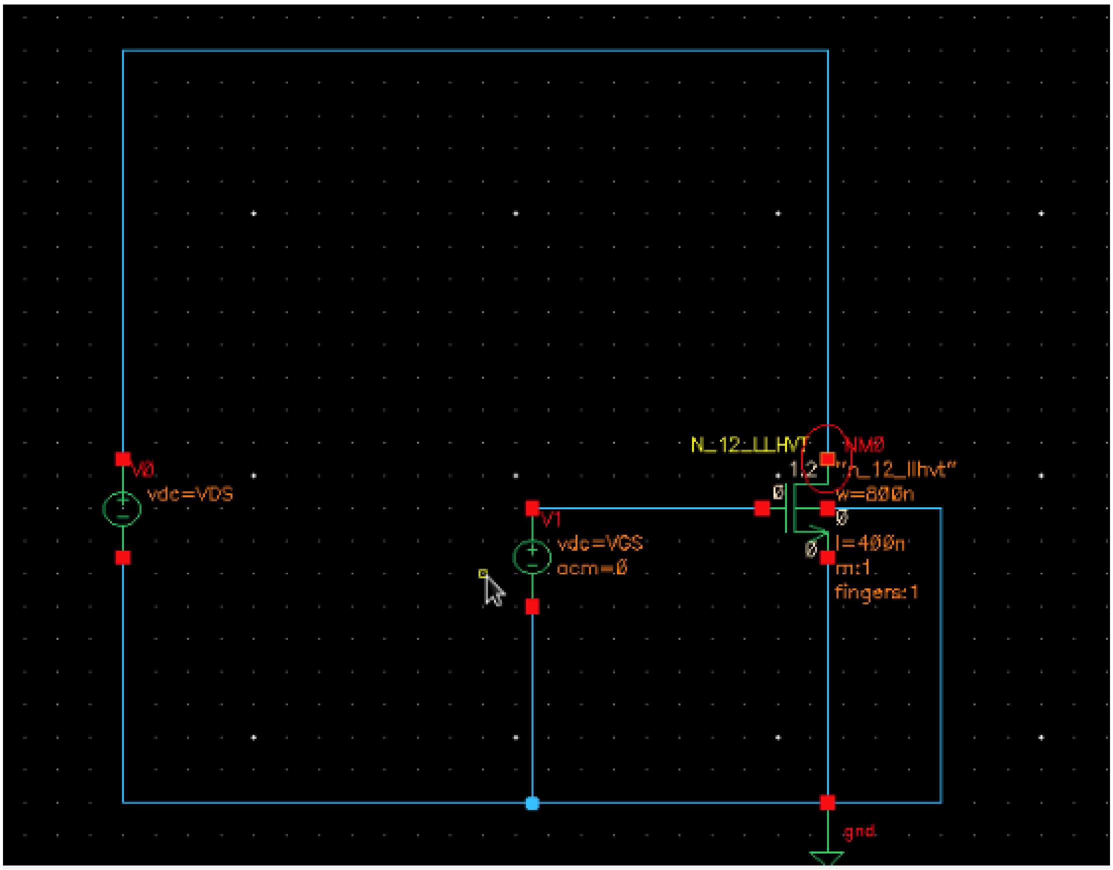

Figure 1: Esquemático do circuito desenvolvido

Background em branco

Após a simulação, Os dados foram exportados para o Matlab onde obtivemos os seguintes gráficos das curvas características ID(VDS,VGS):

Fase 1 alínea b (gráfico de a.1):

Os dados foram exportados para o Matlab onde obtivemos os seguintes gráficos das curvas características ID(VDS,VGS):

Característica $I_D(V_{GS,}V_{DS})\ para\ 0V \leq\ V_{GS}\ \leq 1.2V\ e\ V_{DS} = 1.2V$:

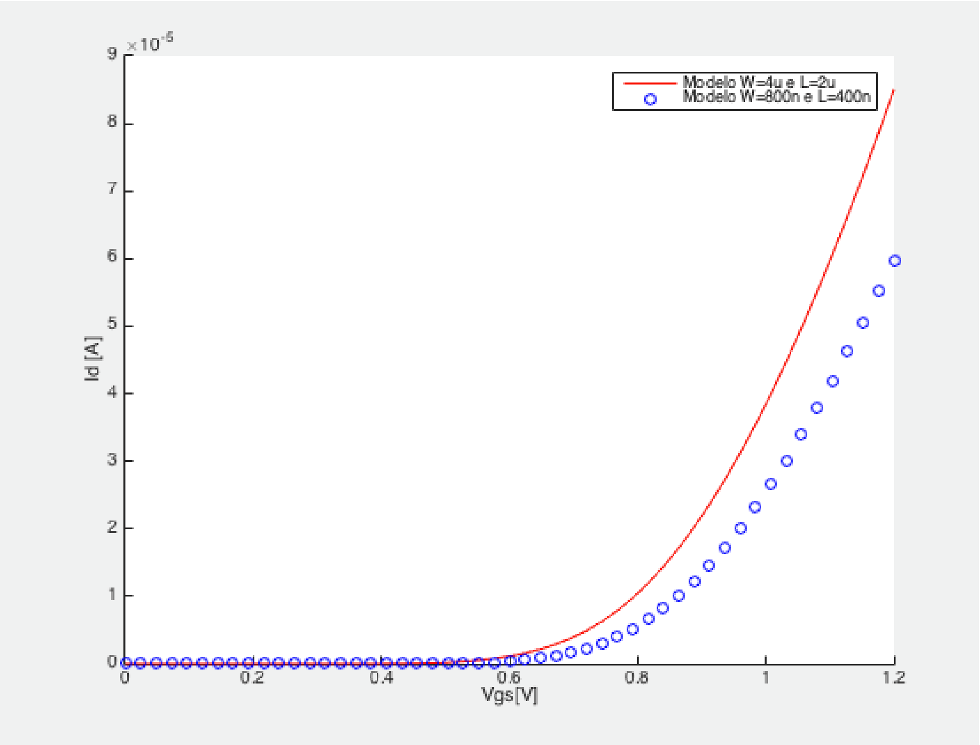

Figure 2: Característica ID (Vgs,Vds) de um transístor com W=4u e L=2u e Vds fixo em 1.2V

Fase 1 alínea c (gráfico de a.2):

Característica $(V_{GS,} V_{DS})\, para\ 0V \leq\ V_{DS}\ \leq 1.2V\ e\ V_{DS} \in \{0.4V, 0.6V, 0.8V, 1.0V\ 1.2V\}$:

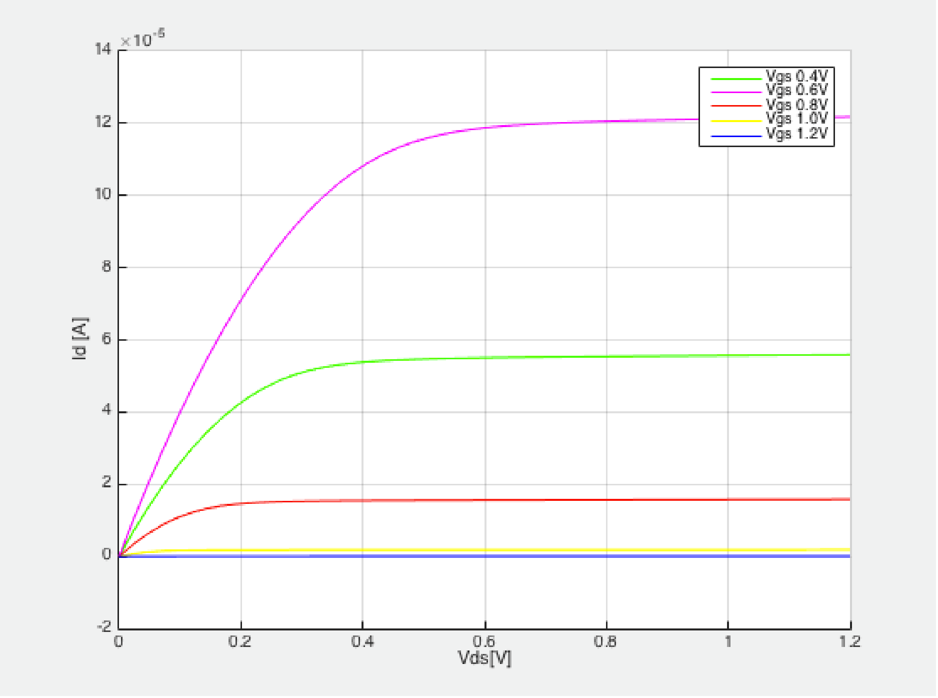

Figure 3: Característica ID(Vgs,Vds) de um transístor com W=800n e L=400n e Vgs com diferentes valores

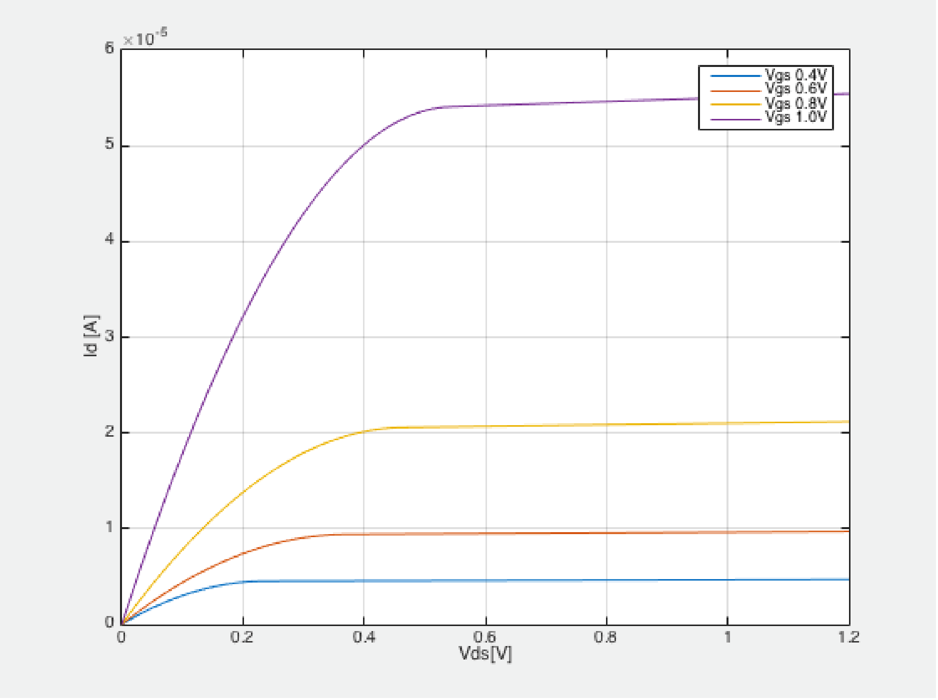

Figure 4: Característica ID(Vgs,Vds) de um transístor com W=4u e L=2u e Vgs com diferentes valores

Fase 1 alínea d (conclusão) :

A necessidade de aplicação de modelos diferentes e cada vez mais reduzidos devem-se a factores importantes para implementação de microcircuitos que possuam características mais potentes com consumos reduzidos.

Fase 1: Comentários:

Os resultados obtidos demonstram a necessidade de utilizar modelos alternativos ao de Schokley uma vez que as curvas ID(VGS) para os dois transístores não coincidem na experiência realizada, algo de inesperado tendo em conta a fórmula de ID(VGS): Sabendo que $I_D = K \times (V_{GS} - V_t)^n$ *e* $K = \mu \times C_{ox} \times \frac{W}{L}$*, em condições normais, para os dois transístores considerados neste trabalho, K deve ser igual para os dois uma vez que a constante Cox é a mesma para ambos bem como a razão (W/L) e μ, os valores de VGS considerados para ambas as experiências foram os mesmos tal como o valor de Vt e m é igual em ambos os ensaios.*

Fase 2: Utilização do modelo *n-power* na caracterização de transistores.

Fase 2 alínea a: Determinar os parâmetros do modelo *n−power* para a tecnologia UMC65, por utilização de metodologia descrita no paper “A Simple MOSFET Model for Circuit Analysis”

Tendo em conta que no modelo *n−power,* na zona de saturação, a corrente de dreno, ID, é descrita através da equação nº 3 acima, torna-se necessário determinar os parâmetros λ,V_t,K,n e B.

Como foram obtidos os parâmetros ?

Os parâmetros determinados foram os seguintes:

Table 2 - Parâmetros Obtidos dos transístores estudados

	Nmos 1 (w=4u e L=2u)	**Nmos 2 (w=800n e L=400n)**
λ	0,0390	0,0906
Vt	0,4658	0,5086
K	0,6303	0,5006
n	2,4488	2,5034
B	1,2414x10^-5	0,6809x10^-5
m	0.9879	0.6825

Fase 2 alínea b: Validar os resultados para transístor individual. Para tal deve gerar características Id(Vds,Vgs) com o modelo *n-power* e fazer um gráfico em que compara com os gráficos resultantes da simulação.

Para o transístor NMOS1 (**W=4u e L=2u)**, obtivemos o seguinte gráfico resultante:

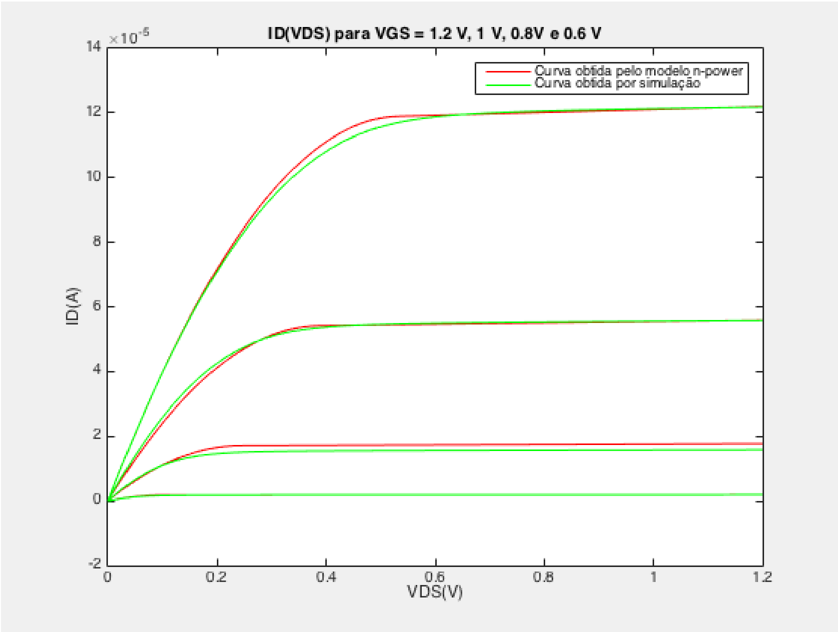

Figure 5: Curva obtida com modelo n-power e simulação do transístor NMOS com w=4u e L=2u

Para o transístor NMOS2 (**W=800n e L=400n)**, obtivemos o seguinte gráfico resultante:

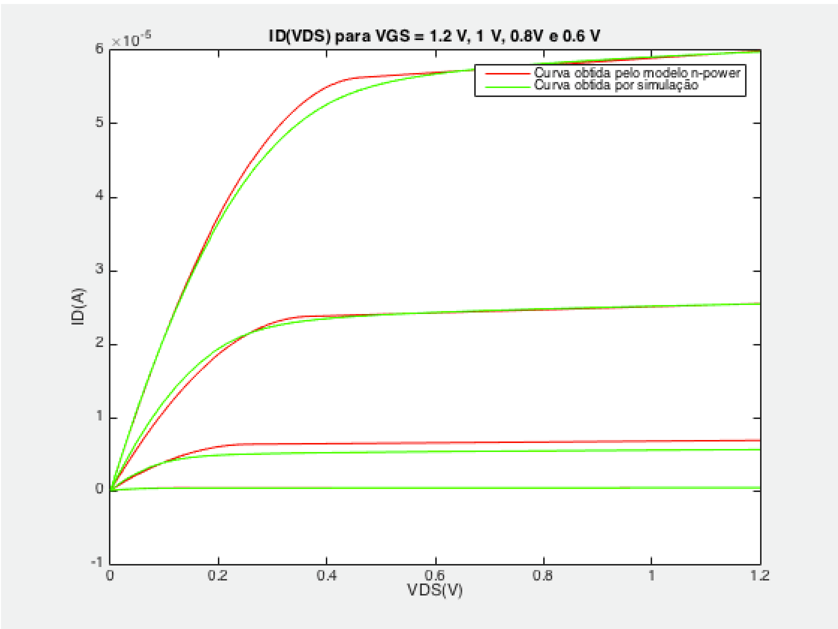

Figure 6: Curva obtida com modelo n-power e simulação do transístor NMOS com w=4u e L=2u

Fase 2: Comentários:

Os resultados provam a precisão superior do modelo n-power, apesar de a aproximação não ser perfeita, comparativamente com o modelo de Schokley, apresenta resultados mais rigorosos e aproximados com o que se verificou em ambiente de simulação. Nota para o facto do modelo n-power ser bastante preciso nas zonas linear e de saturação.

Gráficos de erros relativos ?

Fase 3: Determinação dos parâmetros do modelo *n−power* por utilização de técnicas de "curve-fitting".

Fase 3 alínea a: A partir da característica Id(Vgs) para Vds = 1.2 gerar a característica gm(Vgs).

Para o transístor NMOS1 (**W=4u e L=2u)**, obtivemos o seguinte gráfico resultante:

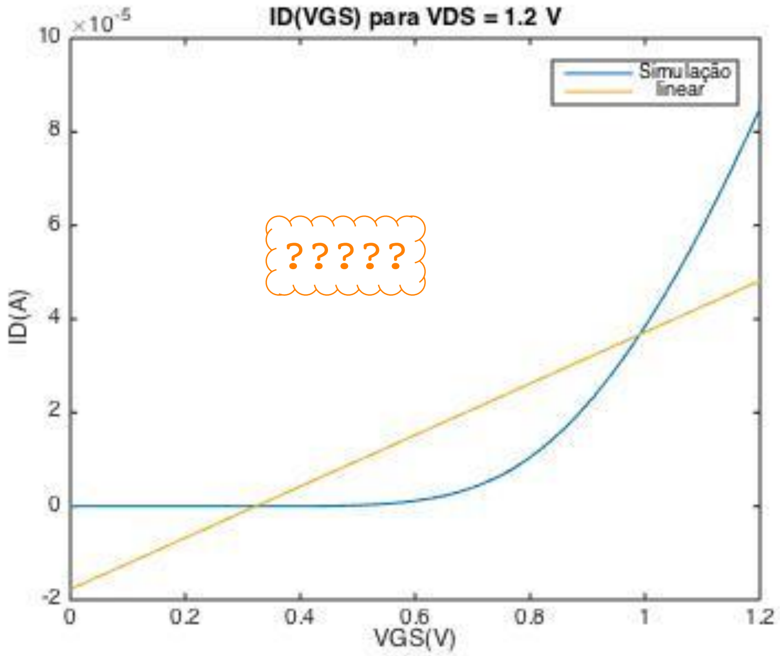

Figure 7: Característica Id(Vgs) para Vds = 1.2V do transístor NMOS com W=4u e L=2u

Para o transístor NMOS2 (**W=800n e L=400n)**, obtivemos o seguinte gráfico resultante:

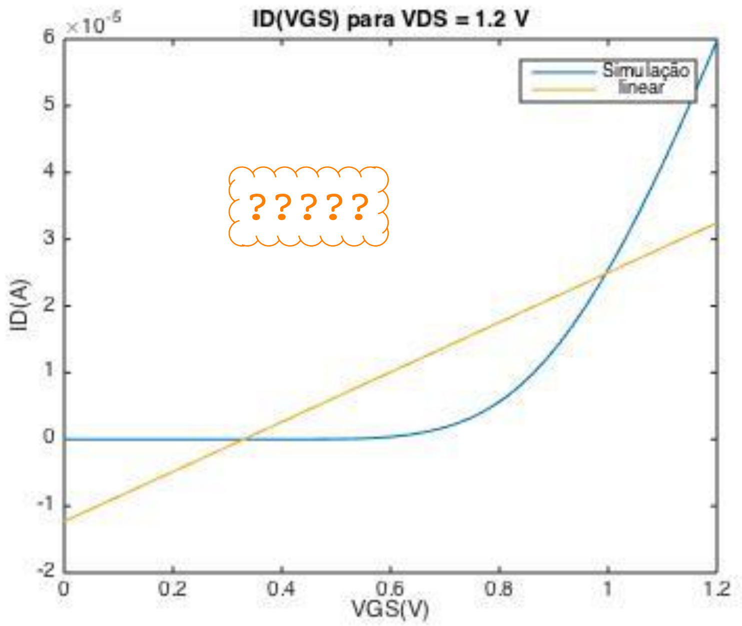

Figure 8: Característica Id(Vgs) para Vds = 1.2V do transístor NMOS com W=800n e L=400n

Fase 3 alínea b: Determinar a característica Id/gm:

Para o transístor NMOS1 (**w=4u e L=2u)**, obtivemos o seguinte gráfico resultante:

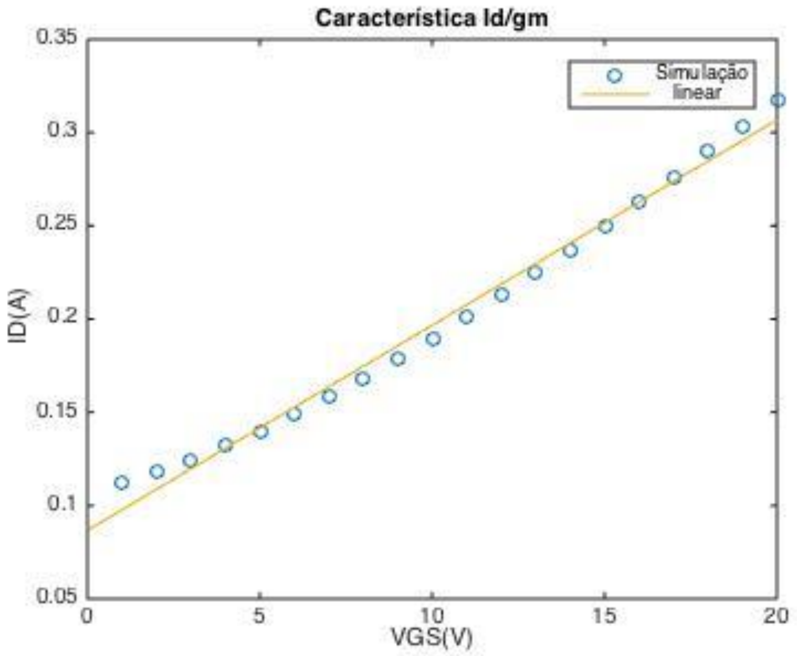

Figure 9: Curve-fitting da característica Id/gm para um transístor com W=4u e L=2u

Para o transístor NMOS2 (**w=800n e L=400n)**, obtivemos o seguinte gráfico resultante:

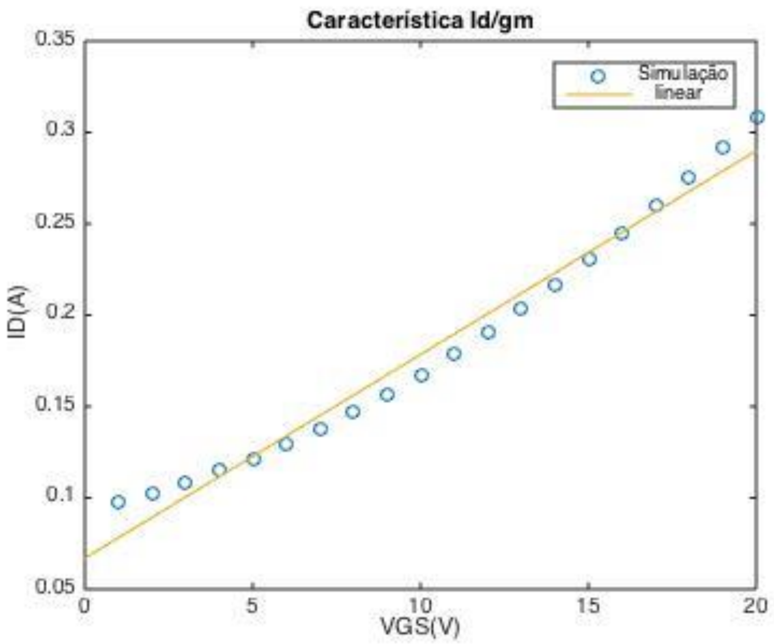

Figure 10: Curve-fitting da característica Id/gm para um transístor com W=800n e L=400n

Para o transístor NMOS1 (**w=4u e L=2u)**, obtivemos os demais gráficos resultantes:

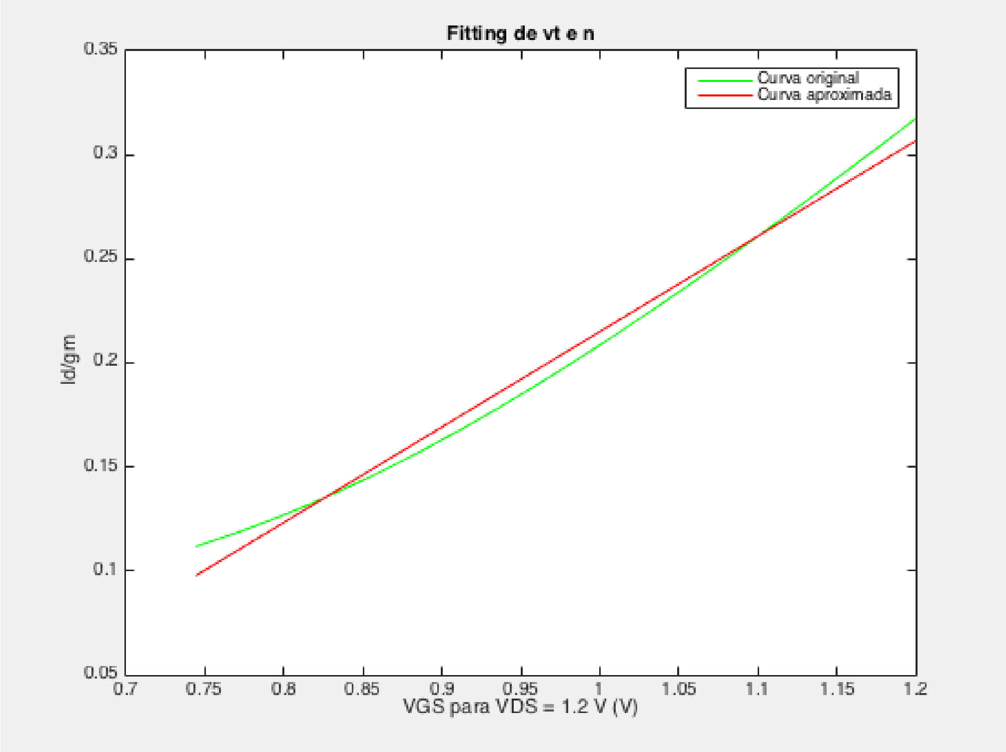

Figure 11 : Curve-fitting para um transístor com W=4u e L=2u

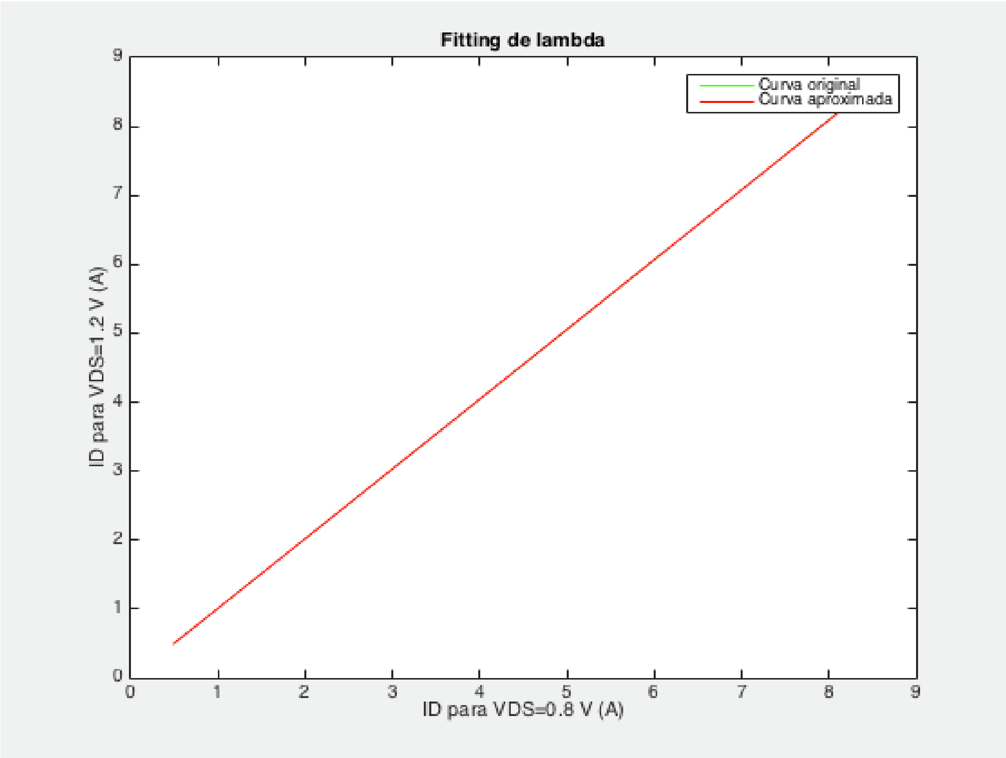

Figure 12: Curve-fitting para um transístor com W=4u e L=2u

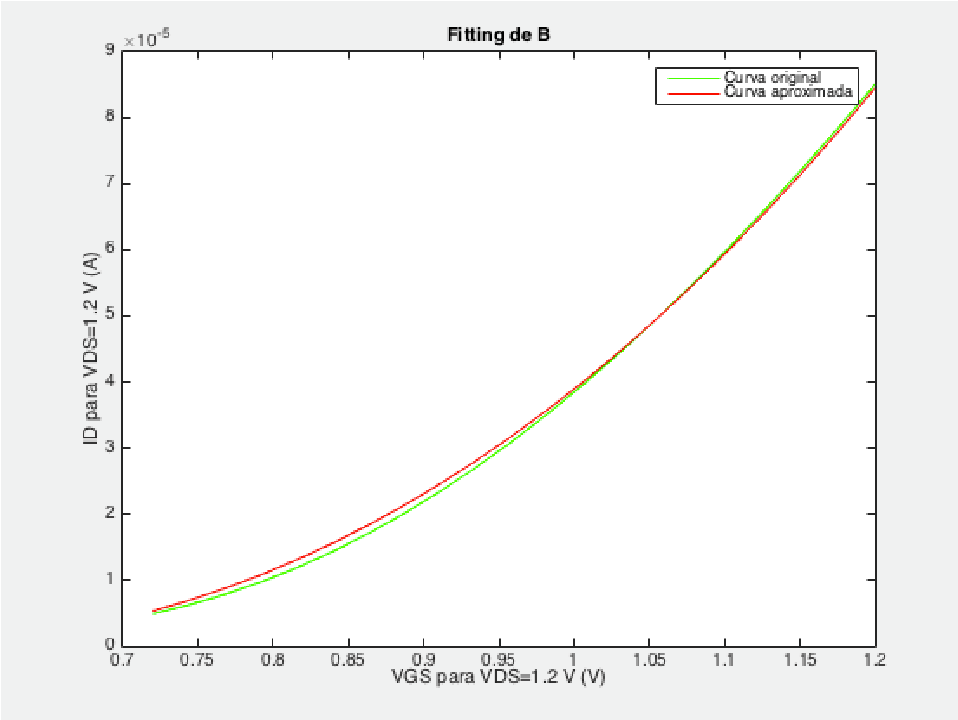

Figure 13: Curve-fitting para um transístor com W=4u e L=2u

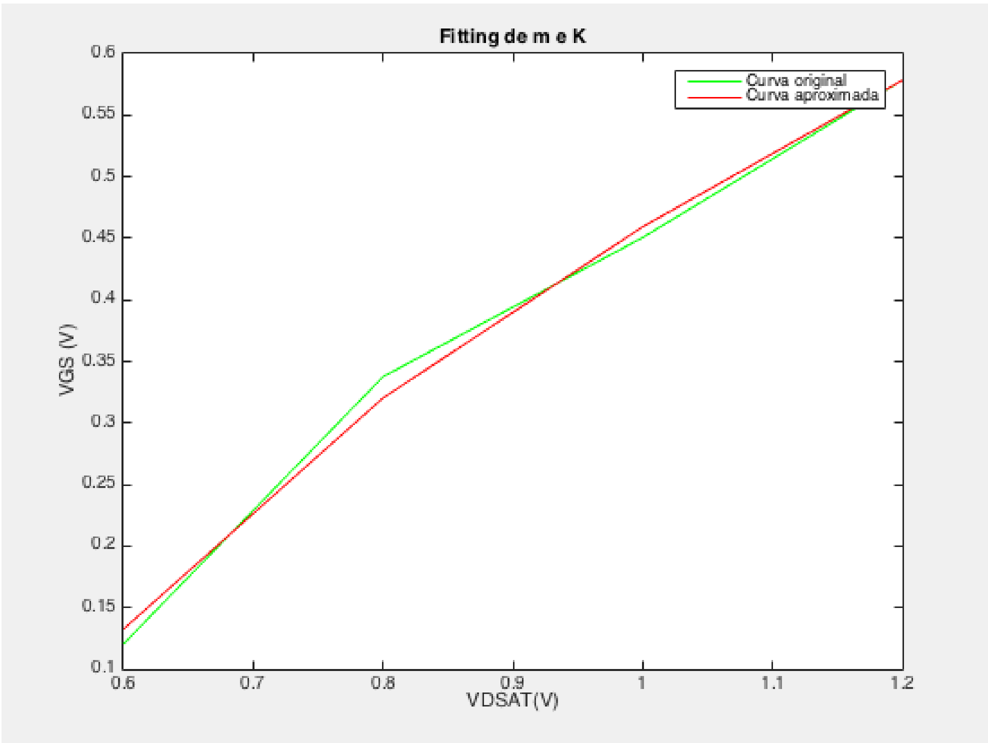

Figure 14: Curve-fitting para um transístor com W=4u e L=2u

Para o transístor NMOS2 (**w=800n e L=400n)**, obtivemos os demais gráficos resultantes

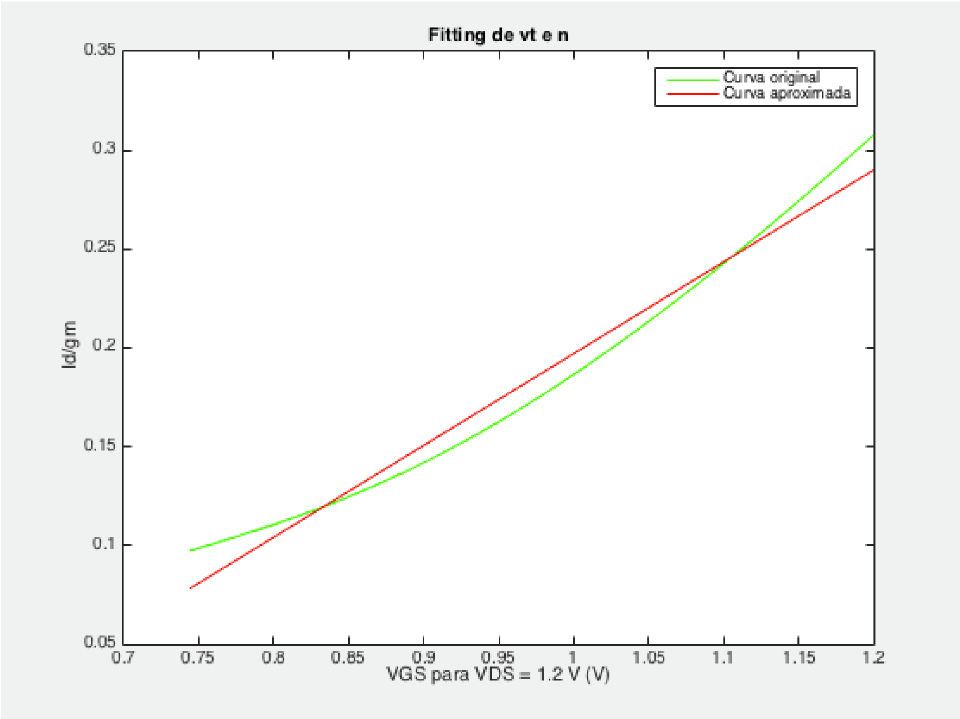

Figure 15: Curve-fitting para um transístor com W=800n e L=400n

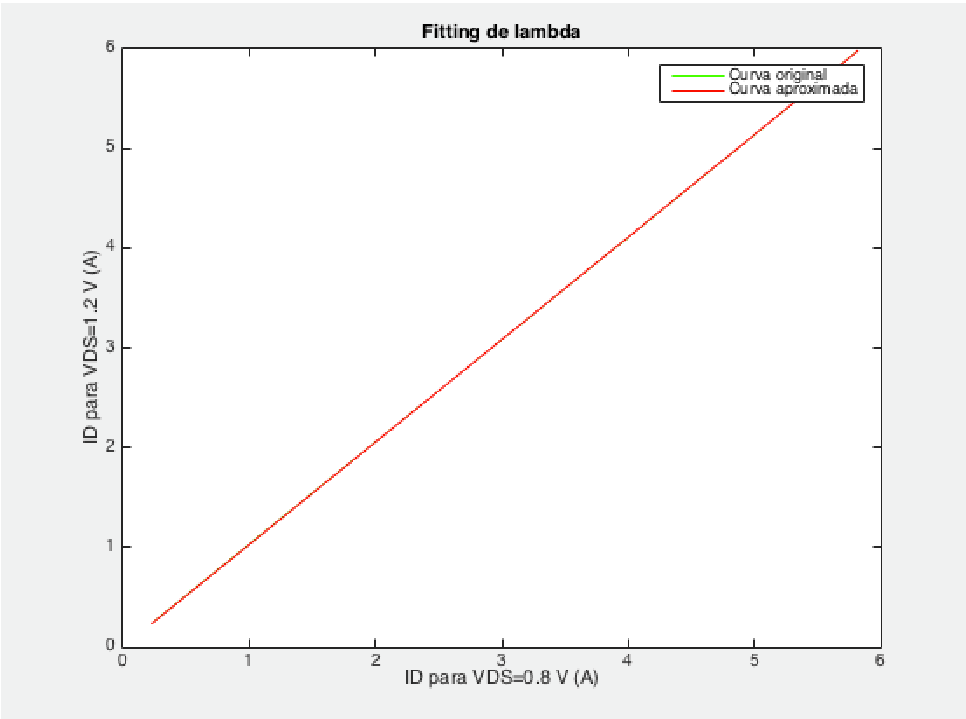

Figure 16: Curve-fitting para um transístor com W=800n e L=400n

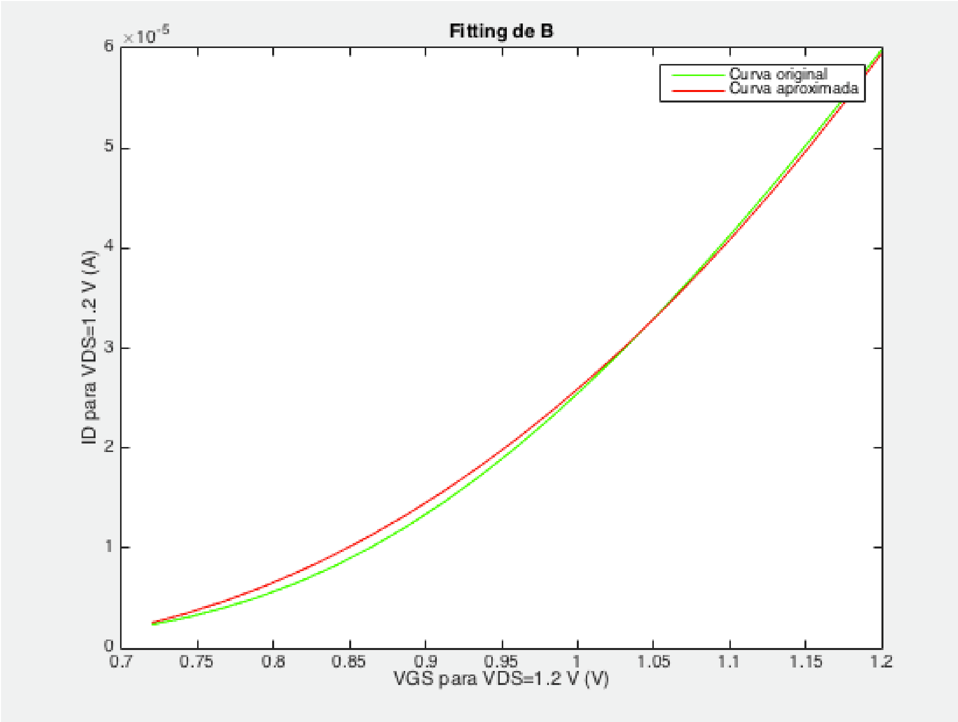

Figure 17: Curve-fitting para um transístor com W=800n e L=400n

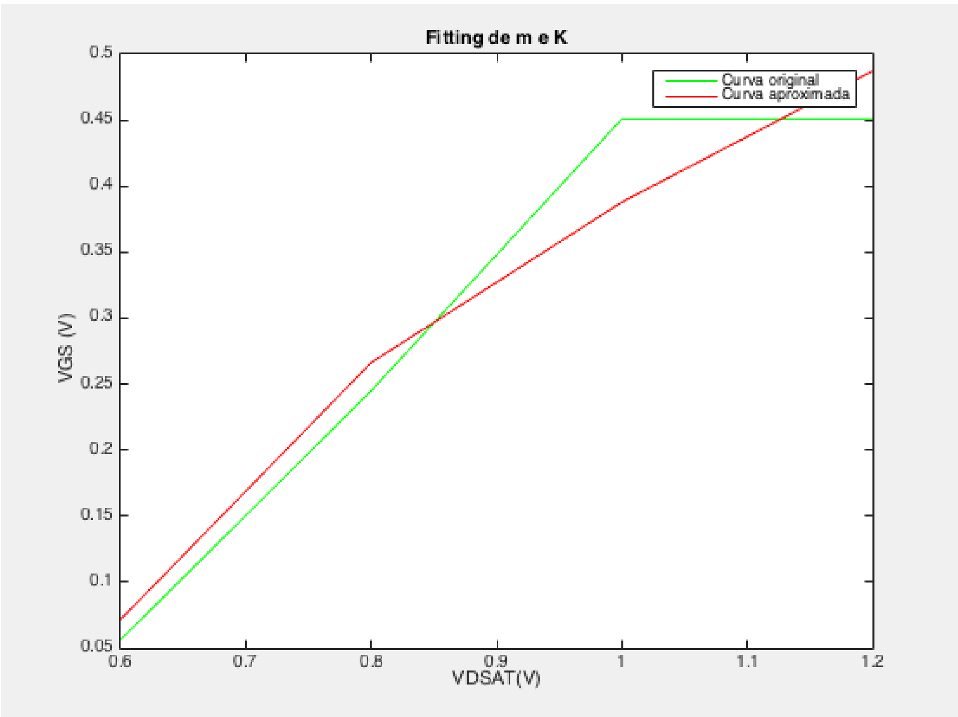

Figure 18: Curve-fitting para um transístor com W=800n e L=400n

Fase 3 alínea c: A partir de b. Determinar valores de Vt e de n

Obtivemos os seguintes valores:

- Para o transístor NMOS1 (**w=4u e L=2u):**

  Vt = 0.531087109842156
  n = 2.180512774235480

- Para o transístor NMOS2 (**w=800n e L=400n)**:

  Vt = 0.575854458793879
  n = 2.152044285997706

Fase 3 alínea d: Sabendo *n* e *Vt* e usando fitting determinar *B*

Obtivemos os seguintes valores:

- Para o transístor NMOS1 (**w=4u e L=2u):**

  B = 0.0001241

- Para o transístor NMOS2 (**w=800n e L=400n)**:

  B = 0.00007540

Fase 3 alínea e: Usando Id(Vds), na zona de saturação, determinar λ:

Quais são os pontos ??? Vgs=? Vds=???

O λ foi determinado através da seguinte equação:

$$\lambda = \frac{I_{D1} - I_{D2}}{V_{DS1} - V_{DS2}}$$

Equação 7: Determinação do Lambda

Obtivemos os seguintes valores:

- Para o transístor NMOS1 (**w=4u e L=2u):**

  lambda = 0.029560132889637

- Para o transístor NMOS2 (**w=800n e L=400n)**:

  lambda = 0.072566480729500

Fase 3 alínea f: Usando Id(Vds), determinar *m* e *k*:

Obtivemos os seguintes valores:

- Para o transístor NMOS1 (**w=4u e L=2u):**

  m = 0.750617133415447
  K = 0.648447045990491

- Para o transístor NMOS2 (**w=800n e L=400n)**:

  m = 0.643325692492985
  K = 0.589720244539865

Fase 3: Comentários:

Os resultados, na generalidade, são coerentes com os valores obtidos na parte 2. Nesta parte, a principal fonte de imprecisões pode ter sido a seleção dos valores de Id para a realização do fitting de Idsat. Procurou-se seleccionar valores de id na zona de saturação, contudo, o critério usado para a selecção desses valores foi meramente prático.

Gráficos de erros relativos?
Variabilidade de parâmetros com variação nas dimensões dos transistores ?

Conclusão

Este trabalho tinha como principal objectivo mostrar que o modelo Schokley não é de facto o melhor modelo a ser aplicado quando se trata de dimensionamento de transístores com dimensões progressivamente mais reduzidas. Para isso, utilizou-se o exemplo de dois transístores com a mesma relação de largura/comprimento do canal do transístor (W/L) mas com tamanhos diferentes (um de W = 4µm e de L = 2 µm e outro de W = 800nm e de L = 400 nm).

Verificou-se na tarefa 1, quando para dois transístores com a mesma relação, se obtiveram diferentes características de corrente *Id(vgs)*, o que levou a que se tivesse de utilizar um modelo mais preciso. Foi-nos sugerido então que se utilizasse o modelo *n-power.*

Na tarefa 2, utilizou-se o modelo *n-power* para se poder dimensionar os dois transístores. Embora este modelo tenha proporcionado melhores características de *Id(vgs)*, as técnicas utilizadas ainda eram, de certa forma, precárias comparadas com as da tarefa 3.

Finalmente, na tarefa 3 (utilizando o mesmo modelo) recorreu-se a técnicas de fitting, proporcionadas pela ferramenta MatLab. Essas técnicas permitiram que se aproximassem as características *Id(vgs)* e *Vgs(vdsat)* de uma zona mais linear, de modo a que os transístores pudessem funcionar na zona de saturação. Não se obteve uma linearização exacta, mas as curvas obtidas aproximaram-se bastante das originais.

Assim conclui-se que o modelo n-power é de facto um melhor modelo a utilizar quando comparado com o modelo de Schokley.

Anexos

Códigos efectuados no MATLAB:

- ❖ Ficheiro: “teste_eda_1_b.m”

```
% trabalho 1 EDA - funcao generica, para todos os trasistores
% para correr digitar o comando:
teste_eda_1_b2('data3_2_4.dat',4*10^-6,2*10^-6)
%                              ou
teste_eda_1_b2('data3_400_800.dat',800*10^-9,400*10^-9).

function teste_eda_1_b(file1,w,l)

close all
clc
data = load(file1);

vdsn=data(:,1);

%PosiÁıes no vector de dados onde vamos buscar os valores de vds e id
npoints=length(vdsn)
vds1=npoints-3;
vds2=npoints-23;
vds6=5;

if (w == 4*10^-6) && (l == 2*10^-6)
    Id1=data(vds1,4)
    Id2=data(vds2,4)
else
    Id1=data(vds1,2)
    Id2=data(vds2,2)
end

if (w == 4*10^-6) && (l == 2*10^-6)
    VDS6 = data(vds6,3);
    VDS1 = data(vds1,3);
    VDS2 = data(vds2,3);
else
    VDS6 = data(vds6,1);
    VDS1 = data(vds1,1);
    VDS2 = data(vds2,1);
end

lambda0 = (Id1-Id2)/((Id2*VDS1)-(Id1*VDS2))

VDS3 = VDS1;
VDS4 = VDS1;
VDS5 = VDS1;

Id3 = Id1;

if (w == 4*10^-6) && (l == 2*10^-6)
    Id4=data(vds1,2);
```

```
else
    Id4=data(vds1,4);
end

Id5=data(vds1,8);

Iz3 = Id3 /(1+(lambda0*VDS3))
Iz4 = Id4 /(1+(lambda0*VDS4))
Iz5 = Id5 /(1+(lambda0*VDS5))

Vgs3 = 1.2;
Vgs4 = 1;
Vgs5 = 0.6;

syms Vt0
equ = log10(Iz3/Iz4)*log10((Vgs4-Vt0)/(Vgs5-Vt0))-
log10(Iz4/Iz5)*log10((Vgs3-Vt0)/(Vgs4-Vt0));

% S = Vt0
S = vpasolve(equ,Vt0,0:1.2)

n = log10(Iz3/Iz4)/log10((Vgs3-S)/(Vgs4-S))

B = (Iz3/((Vgs3-S)^n))*(0.5)

VDS7 = VDS6;
Vgs6 = Vgs3;
Vgs7 = 0.8;

if (w == 4*10^-6) && (l == 2*10^-6)
    ID6 = data(vds6,4)
else
    ID6 = data(vds6,2)
end
ID7 = data(vds6,6)

%Neste caso, assume-se que o comprimento eficaz do transistor È o seu
prÛprio
%comprimento
leff = l;

E6 = ID6 /(B*(w/leff)*((Vgs6-S)^n)*(1+lambda0*VDS6))
E7 = ID7 /(B*(w/leff)*((Vgs7-S)^n)*(1+lambda0*VDS7))

Vdsat6 = VDS6*(1+(sqrt(1-E6)))/E6
Vdsat7 = VDS7*(1+(sqrt(1-E7)))/E7

m = log(Vdsat6/Vdsat7)/(log((Vgs6-S)/(Vgs7-S)))
K = Vdsat6/(Vgs6-S)^m

VGS = 0.4:0.2:1.2;
VDS = 0:0.01:1.2;
%Ciclo para organizar e representar as caracterÌsticas de Id para os
5
%valores de VGS

    if (w == 4*10^-6) && (l == 2*10^-6)
        vgsx12 = data(:,3);
```

```
        vgsy12 = data(:,4);
        vgsx10 = data(:,1);
        vgsy10 = data(:,2);
    else
        vgsx12 = data(:,1);
        vgsy12 = data(:,2);
        vgsx10 = data(:,3);
        vgsy10 = data(:,4);
    end
    vgsx8 = data(:,5);
    vgsy8 = data(:,6);
    vgsx6 = data(:,7);
    vgsy6 = data(:,8);
    for j=2:length(VGS)
        for i=1:length(VDS)
            p(i)=teste_eda_2(VGS(j),VDS(i),w,l,B,n,S,m,K,lambda0);
        end
        p1=plot(VDS,p,'r')
        hold on
    end
    p2=plot(vgsx12,vgsy12,'g');
    plot(vgsx10,vgsy10,'g')
    plot(vgsx8,vgsy8,'g')
    plot(vgsx6,vgsy6,'g')
    title('ID(VDS) para VGS = 1.2 V, 1 V, 0.8V e 0.6 V');
    xlabel('VDS(V)');
    ylabel('ID(A)');
    legend([p1,p2],'Curva obtida pelo modelo n-power','Curva obtida
por simulaÁ„o');
    hold off
%% 3™ parte %%
    %Extraem-se as carcterÌsticas Id(VGS)
    if (w == 4*10^-6) && (l == 2*10^-6)
        dados12 = xlsread('ID_vds_1.2_L4u_W2u.xlsx');
        dados08 = xlsread('ID_vds_0.8_L4u_W2u.xlsx');
    else
        dados12 = xlsread('ID_vds_1.2_L800n_W400n.xlsx');
        dados08 = xlsread('ID_vds_0.8_L800n_W400n.xlsx');
    end

    VGS_12 = dados12(:,1);
    VGS_08 = dados08(:,1);
    ID_12 = dados12(:,2);
    ID_08 = dados08(:,2);
    figure;
    plot(VGS_12,ID_12);
    title('ID(VGS) para VDS = 1.2 V');
    xlabel('VGS(V)');
    ylabel('ID(A)');

    %Excluem-se valores de VGS<0.7, pelo facto de os id serem
demasiado
    %pequenos para a escala de valores do Matlab
    ID_08 = ID_08(find(VGS_08>0.7));
    ID_12 = ID_12(find(VGS_12>0.7));
    VGS_12 = VGS_12(find(VGS_12>0.7));
    gm = diff(ID_12)./diff(VGS_12);
    %charac -> CaracterÌstica Id/gm
    charac = ID_12(2:end)./gm;
    figure;
    plot(charac,'o')
```

```
    title('CaracterÌstica Id/gm');
    xlabel('VGS(V)');
    ylabel('ID(A)');
    val = mylsqftiT(VGS_12(2:end),charac);
    vt_3 = val(1)
    n_3 = val(2)
    lamb = fitting_lambda(double(ID_08),double(ID_12));
    lambda = lamb(1)
    b = fitting_B(VGS_12,ID_12,lambda,n_3,vt_3,w,l);
    B = b(1)

    %fitting de idsat e vdsat

    if (w == 4*10^-6) && (l == 2*10^-6)
        coef12= 3;
        coef10= 1;
    else
        coef12= 1;
        coef10= 3;
    end

    idsat_12 = fitting_idsat(data(end-15:end,coef12),data(end-
15:end,(coef12+1)),lambda)
    idsat_10 = fitting_idsat(data(end-15:end,coef10),data(end-
15:end,(coef10+1)),lambda)
    idsat_08 = fitting_idsat(data(end-15:end,5),data(end-
15:end,6),lambda)
    idsat_06 = fitting_idsat(data(end-15:end,7),data(end-
15:end,8),lambda)
    idsat_04 = fitting_idsat(data(end-15:end,9),data(end-
15:end,10),lambda)

    vdsat_12 =
fitting_vdsat(data(:,coef12),data(:,(coef12+1)),idsat_12,lambda);
    vdsat_10 =
fitting_vdsat(data(:,coef10),data(:,(coef10+1)),idsat_10,lambda);
    vdsat_08 = fitting_vdsat(data(:,5),data(:,6),idsat_08,lambda);
    vdsat_06 = fitting_vdsat(data(:,7),data(:,8),idsat_06,lambda);
    vdsat_04 = fitting_vdsat(data(:,9),data(:,10),idsat_04,lambda);


    ydata = [vdsat_04 vdsat_06 vdsat_08 vdsat_10 vdsat_12]
    xdata = [0.4 0.6 0.8 1 1.2]
    Values = fitting_mk(xdata(2:end),ydata(2:end),vt_3);

    M=Values(1)
    K=Values(2)
end
```

- Ficheiro: "mylsqftiT" (função)

```
function K=mylsqftiT(xdata,ydata)
    Ki = [10,10]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    figure
    plot(xdata,ydata,'g');
    xlabel('VGS[V]');
    ylabel('ID[A]');
    hold on
    newY= (1/K(2))*(xdata-K(1));
    plot(xdata,newY,'r');
    function F = myfun(x,xdata)
        F = (1/x(2))*(xdata-x(1));
    end
end
```

- Ficheiro: "fitting_idsat" (função)

```
function K=fitting_idsat(xdata,ydata,lambda)
    Ki = [100]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    newY= K(1)*(1+(lambda*xdata));
    function F = myfun(x,xdata)
        F = x(1)*(1+(lambda*xdata));
    end
end
```

- Ficheiro: "teste_eda_2" (função)

```
function [Id,Idsat]= teste_eda_2(vgs,vds,w,l,B,n,vt,m,K,lambda0)

Idsat = (w/l)*B*((vgs-vt)^n);
Vdsat = K*(vgs-vt)^m;
    if vds >= Vdsat
        Id=Idsat*(1+lambda0*vds);
    else
        Id=Idsat*(1+lambda0*vds)*(2-(vds/Vdsat))*(vds/Vdsat);
    end
end
```

- Ficheiro: "fitting_vdsat" (função)

```
function K=fitting_vdsat(xdata,ydata,idsat,lambda)
    Ki = [0.0002]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    function [F] = myfun(x,xdata)
        if(xdata>x(1))
            F=idsat*(1+lambda*xdata);
        else
            F=idsat*(1+lambda*xdata).*(2-(xdata/x(1))).*(xdata/x(1));
        end
    end
end
```

❖ Ficheiro: "fitting_B" (função)

```
function K=fitting_B(xdata,ydata,lambda,n_3,vt,w,l)
    Ki = [100]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    figure
    plot(xdata,ydata,'g');
    hold on
    newY= (w/l)*K(1)*(1+lambda*1.2)*((xdata-vt).^n_3);
    plot(xdata,newY,'r')
    function F = myfun(x,xdata)
        F = (w/l)*x(1)*(1+lambda*1.2)*((xdata-vt).^n_3);
    end
end
```

❖ Ficheiro: "fitting_lambda" (função)

```
function K=fitting_lambda(xdata,ydata)
    Ki = [100]; % Starting guess
    ydata=ydata*1e5;
    xdata=xdata*1e5;
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    figure
    plot(xdata,ydata,'g');
    hold on
    newY= xdata*((1+(K(1)*1.2))/(1+(K(1)*0.8)));
    plot(xdata,newY,'r')
    function F = myfun(x,xdata)
        F = xdata*((1+(x(1)*1.2))/(1+(x(1)*0.8)));
    end
end
```

❖ Ficheiro: "fitting_m" (função)

```
function K=fitting_m(xdata,ydata,vt,lambda,w,l,B,n)
    Ki = [100,-1]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    figure
    plot(xdata,ydata,'g');
    hold on
    newY = ((w/l)*B*((xdata-vt).^n)).*(1+lambda*(K(1)*(xdata-
vt).^K(2)));
    plot(xdata,newY,'r')
    function F = myfun(x,xdata)
        F = ((w/l)*B*((xdata-vt).^n)).*(1+(lambda*(x(1)*((xdata-
vt).^x(2)))));
    end
end
```

❖ Ficheiro: "fitting_mk" (função)

```
function K=fitting_mk(xdata,ydata,vt)
    Ki = [1,1]; % Starting guess
    [K,resnorm] = lsqcurvefit(@myfun,Ki,xdata,ydata);
    figure
    plot(xdata,ydata,'g');
    hold on
    newY = K(1)*((xdata-vt).^K(2));
    plot(xdata,newY,'r')
    function F = myfun(x,xdata)
        F = x(1).*((xdata-vt).^x(2));
    end
```

```
end
```