



TRABALHO DE CONCLUSÃO DE CURSO

**COMPLEMENTO DE UM SIMULADOR GERAL DE
CIRCUITOS (QUCS) COM UM MODELO E MÉTODO
DE EXTRAÇÃO UNIFICADOS (UMEM)**

Arthur Rocha de Almeida Lima

Brasília, dezembro de 2015

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO
**COMPLEMENTO DE UM SIMULADOR GERAL DE
CIRCUITOS (QUCS) COM UM MODELO E MÉTODO
DE EXTRAÇÃO UNIFICADOS (UMEM)**

Arthur Rocha de Almeida Lima

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Stefan Michael Blawid, ENE/UnB
Orientador

Dr. Muthupandian Cheralathan, ENE/UnB
Co-orientador

Prof. Daniel Chaves Café, ENE/UnB
Examinador interno

Prof. Alexandre Ricardo Soares Romariz,
ENE/UnB
Examinador interno

Dedicatória

Dedico este trabalho a minhas avós Nadyr e Clésia.

Arthur Rocha de Almeida Lima

Agradecimentos

Agradeço, em primeiro lugar, a meus pais por serem o alicerce que sustentou toda minha trajetória até aqui. Sou grato também a meus padrinhos, tios e primos pelas conversas e momentos de união. Agradeço aos meus amigos por serem meus psicólogos, conselheiros e comparsas em todos os momentos, bons e ruins. Em especial, meus amigos de curso foram de extrema importância para que eu chegasse até aqui. Me sinto eternamente grato pelas noites mal dormidas de estudo em grupo, pelo conhecimento e aprendizado compartilhados e principalmente pelo amor e união diante de tantos obstáculos durante todo o curso; sem vocês, tenho certeza que eu não teria chegado tão longe na Engenharia. Agradeço a Dr^a. Alejandra Castro-Carranza pelo envio do código computacional utilizado no projeto. Não posso esquecer de agradecer a meu orientador Prof. Dr. Stefan Michael Blawid e meu co-orientador Dr. Muthupandian Cheralatan, por todo o conhecimento transmitido e por toda a paciência. Por último e não menos importante, agradeço a minha namorada Juliana Emiliana por ter acreditado em mim quando nem eu mesmo acreditava e por me fazer enxergar o quão boa a vida pode ser.

Arthur Rocha de Almeida Lima

RESUMO

O presente projeto apresenta a adição de um método unificado de modelagem e extração de parâmetros (UMEM) de transistores orgânicos de filmes finos (OTFTs) a um simulador geral de circuitos (Qucs). Neste trabalho, é apresentado um breve tutorial explicando como se implementa o modelo, através de um código escrito em Verilog-A, no simulador. A infra-estrutura formada pela junção das duas ferramentas é então testada como ferramenta de modelagem de transistores orgânicos de filme fino, analisando-se dois conjuntos de dados experimentais. A primeira é tomada da literatura [1] e representa as características IV de um OTFT em uma plataforma de tecnologia estabelecida. O segundo conjunto de dados ainda não foi publicado [2] e representa o comportamento de um OTFT empregando uma plataforma de tecnologia ainda em desenvolvimento. São feitas comparações dos resultados obtidos pela infra-estrutura Qucs + UMEM com uma infra-estrutura já existente, onde o modelo foi implementado e simulado via código em Mathcad. Foi verificada uma discrepância entre os resultados obtidos via Mathcad e via Verilog-A.

ABSTRACT

This project presents the addition of a unified model and parameter extraction method (UMEM) for organic thin film transistors (OTFTs) in a general circuit simulator (Quite Universal Circuit Simulator). This report includes a brief tutorial explaining how to implement the model in the simulator by means of a code written in Verilog-A. The infrastructure formed by the junction of these two tools is then tested as a modeling tool for organic thin film transistors analyzing two experimental data sets. The first is taken from the literature [1] and represents the IV characteristics of a OTFT manufactured on an established technology platform. The second data set is yet unpublished [2] and represents the behavior of an OTFT employing a technology platform still under development. Comparisons are made between the results of the Qucs + UMEM infrastructure with an existing infrastructure, where the model is implemented and simulated via Mathcad code. A discrepancy between the results obtained via Mathcad and via Verilog-A has been verified.

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | CONTEXTUALIZAÇÃO | 1 |
| 1.2 | OBJETIVOS DO PROJETO | 2 |
| 1.3 | APRESENTAÇÃO DO RELATÓRIO | 3 |
| 2 | Revisão Bibliográfica | 4 |
| 2.1 | INTRODUÇÃO | 4 |
| 2.2 | OTFT | 4 |
| 2.2.1 | DEFINIÇÃO | 4 |
| 2.2.2 | ARQUITETURA DE TRANSISTORES DE FILME FINO | 5 |
| 2.2.3 | FUNCIONAMENTO | 6 |
| 2.2.4 | RESISTÊNCIA DE CONTATO | 8 |
| 2.2.5 | TRANSPORTE DE CARGA | 9 |
| 2.2.6 | TÉCNICAS DE FABRICAÇÃO | 10 |
| 2.2.7 | MATERIAIS UTILIZADOS EM OTFTs | 11 |
| 2.3 | MODELOS COMPACTOS | 11 |
| 2.4 | VERILOG-A | 12 |
| 2.4.1 | VANTAGENS DA LINGUAGEM VERILOG-A | 12 |
| 2.5 | QUCS | 13 |
| 2.6 | UMEM | 14 |
| 2.6.1 | MODELAGEM DE OTFTs COM A PRESENÇA DE CONTATOS NÃO-OHMICOS | 16 |
| 2.6.2 | EXTRAÇÃO DOS PARÂMETROS DO DIODO | 17 |
| 3 | Metodologia | 19 |
| 3.1 | <i>i</i> -MOS | 20 |
| 3.2 | MATCAD | 20 |
| 3.3 | SIMULAÇÃO DE UM MOSFET NO QUCS | 20 |
| 3.4 | IMPLEMENTAÇÃO DO UMEM EM VERILOG-A | 25 |
| 3.5 | CONSTRUÇÃO DE UM MÓDULO NO QUCS UTILIZANDO UM CÓDIGO VERILOG-A | 27 |
| 4 | Resultados e Análises | 30 |
| 4.1 | INTRODUÇÃO | 30 |
| 4.2 | VERIFICAÇÃO DO USO DO QUCS COM O UMEM | 30 |

| | | |
|-----------|---|-----------|
| 4.2.1 | COMPARAÇÃO ENTRE OS DADOS FORNECIDOS PELA LITERATURA E UMA SIMULAÇÃO VIA QUCS | 34 |
| 4.3 | EXTRAÇÃO DOS PARÂMETROS VIA MATHCAD E VERIFICAÇÃO COM QUCS | 37 |
| 4.4 | UTILIZAÇÃO DO UMEM PARA MODELAR UM OTFT..... | 40 |
| 5 | Conclusões..... | 45 |
| 5.1 | RECOMENDAÇÕES PARA TRABALHOS FUTUROS | 46 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 47 |
| | Anexos..... | 49 |
| I | Código em Verilog-A que implementa o UMEM..... | 50 |
| II | Descrição do conteúdo do CD..... | 54 |

LISTA DE FIGURAS

| | | |
|------|---|----|
| 2.1 | Estruturas de OTFTs, ambas com a porta na região inferior. (a) Bottom contact (BC); (b) Top contact (TC). | 6 |
| 2.2 | Curvas de saída (a) e transferência (b) de um típico OTFT. A figura (b) mostra ainda a estrutura molecular do pentaceno, utilizado como o semiconductor desse dispositivo. | 6 |
| 2.3 | Esquema de energia do transistor de pentaceno-ouro | 7 |
| 2.4 | Circuitos equivalentes de um TFT incluindo as resistências dos contatos. O último circuito inclui diodos para modelar resistências não-lineares | 8 |
| 2.5 | Diagrama de níveis de energia da interface ouro-pentaceno, apresentado em [10] | 9 |
| 2.6 | Hierarquia dos modelos computacionais, do menos descritivo ao mais descritivo. | 12 |
| 3.1 | Captura de tela mostrando a aba que contém os componentes não lineares | 21 |
| 3.2 | Captura de tela mostrando o circuito já contendo as fontes e a ponta de prova no Qucs..... | 21 |
| 3.3 | Captura de tela mostrando o circuito com os blocos de simulação e varredura devidamente configurados..... | 22 |
| 3.4 | Captura de tela mostrando a tela de escolha da variável a ser plotada..... | 23 |
| 3.5 | Característica de saída do n-MOS simulado pelo Qucs | 23 |
| 3.6 | Curva de transferência do transistor n-MOS simulado pelo Qucs..... | 24 |
| 3.7 | Curva simulada em azul e curva importada em vermelho..... | 24 |
| 3.8 | Captura de tela mostrando a aba que contém o símbolo do módulo no Qucs | 27 |
| 3.9 | Captura de tela mostrando a janela que deve aparecer após o símbolo ser salvo | 28 |
| 3.10 | Captura de tela mostrando a janela de seleção do módulo a ser carregado..... | 28 |
| 3.11 | Captura de tela mostrando a seção na qual o componente criado está disponível para uso em esquemáticos | 29 |
| 4.1 | Curvas obtidas ao final do processo de extração no Mathcad. Os parâmetros obtidos ao final dessa extração são mostrados na Tabela 4.1 | 31 |
| 4.2 | Curva experimental e curva obtida via simulação no Mathcad. Os parâmetros obtidos ao final dessa extração são mostrados na Tabela 4.1 | 32 |
| 4.3 | Curva experimental e curva obtida via simulação no Qucs. Os parâmetros da Tabela 4.1 foram inseridos no Qucs para que a curva I_{DS} vs V_{DS} fosse gerada..... | 32 |
| 4.4 | Curva experimental e curva obtida via simulação no <i>i</i> -MOS. Os parâmetros da Tabela 4.1 foram inseridos no <i>i</i> -MOS para que a curva I_{DS} vs V_{DS} fosse gerada. | 33 |

| | | |
|------|--|----|
| 4.5 | Curva experimental e curvas obtida via simulação no Mathcad, Qucs e <i>i</i> -MOS. Para gerar essa figura, foram usados os parâmetros mostrados na Tabela 4.1 | 33 |
| 4.6 | Curva característica de saída do transistor T1 obtida com Qucs utilizando os parâmetros fornecidos em [1] e a curva original apresentada em [1]. Esses parâmetros são mostrados na Tabela 4.2..... | 35 |
| 4.7 | Curva característica de saída do transistor T1 obtida com Qucs utilizando parâmetros modificados. Esses parâmetros são mostrados na Tabela 4.3 | 36 |
| 4.8 | Curva característica de saída do transistor T1 obtida com Qucs alterando apenas γ_a em relação aos valores da tabela 4.2. Esses parâmetros são mostrados na Tabela 4.3 | 37 |
| 4.9 | Curva característica de saída do transistor T1 obtida com Qucs alterando apenas V_{aa} em relação aos valores da tabela 4.2. Esses parâmetros são mostrados na Tabela 4.3 | 37 |
| 4.10 | Curva característica de saída do transistor criado via <i>i</i> -MOS e o respectivo ajuste via Mathcad. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4 | 39 |
| 4.11 | Curva característica de saída do transistor criado via <i>i</i> -MOS e o respectivo ajuste via Qucs. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4 | 39 |
| 4.12 | Comparação entre a curva característica de saída do transistor criado via <i>i</i> -MOS e a curva criada via Qucs com os parâmetros extraídos anteriormente via Mathcad. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4 | 40 |
| 4.13 | Ajuste da região linear da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5..... | 41 |
| 4.14 | Ajuste da região de saturação da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5 | 42 |
| 4.15 | Ajuste completo da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5 | 42 |
| 4.16 | Ajuste completo via Qucs e via <i>i</i> -MOS. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5..... | 43 |
| 4.17 | Curva de transferência do transistor utilizando os valores dos parâmetros com os quais se realizou o ajuste completo da curva característica de saída. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5 | 43 |
| 4.18 | Curva de transferência do transistor com o ajuste completo realizado. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.6..... | 44 |

LISTA DE TABELAS

| | | |
|-----|---|----|
| 4.1 | Parâmetros do modelo UMEM extraídos via simulação/processo de extração no Mathcad. Esses dados foram utilizados como base para as figuras 4.1, 4.2, 4.3, 4.4 e 4.5..... | 31 |
| 4.2 | Parâmetros do transistor T1 [1], utilizados nas figuras 4.6, 4.7, 4.8 e 4.9 | 34 |
| 4.3 | Parâmetros do transistor T1 modificados para obtenção da curva mostrada em 4.7, 4.8 e 4.9 | 38 |
| 4.4 | Parâmetros iniciais do transistor inseridos no i -MOS e parâmetros extraídos via Mathcad. Esses dados foram utilizados para gerar as figuras 4.10, 4.11 e 4.12..... | 38 |
| 4.5 | Parâmetros extraídos manualmente utilizando o Qucs referentes as curvas 4.13, 4.14 e 4.15 | 41 |
| 4.6 | Parâmetros utilizados para o ajuste completo da curva de característica de saída e da curva de transferência | 44 |

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

| | | |
|-------------|---|-----------------------|
| MOSFET | <i>Metal Oxide Semiconductor Field Effect</i> | |
| TFT | <i>Thin Film Transistor</i> | |
| LCD | <i>Liquid Crystal Display</i> | |
| OTFT | <i>Organic Thin Film Transistor</i> | |
| Qucs | <i>Quite Universal Circuit Simulator</i> | |
| UMEM | <i>Unified Model and Parameter Extraction Method</i> | |
| T | Temperatura | [K] |
| kT | Energia Térmica | [eV] |
| HOMO | Orbital mais alto ocupado | |
| LUMO | Menor orbital molecular desocupado | |
| μ | Mobilidade do canal | [m ² /V s] |
| ID_{lin} | Corrente na região linear da curva característica de saída | [A] |
| ID_{sat} | Corrente na região de saturação da curva característica de saída | [A] |
| V_G | Tensão de porta | [V] |
| V_D | Tensão de dreno | [V] |
| V_T | Tensão de limiar | [V] |
| R_D | Resistência de contato do dreno | [Ω] |
| R_S | Resistência de contato da fonte | [Ω] |
| <i>PMOS</i> | MOSFET com canal tipo P | |
| <i>NMOS</i> | MOSFET com canal tipo N | |
| I_{DS} | Corrente entre dreno e fonte | [A] |
| V_{DS} | Tensão entre dreno e fonte | [V] |
| V_{GS} | Tensão entre porta e fonte | [V] |
| γ_a | Parâmetro de ajuste da mobilidade do canal no modelo UMEM | |
| V_{aa} | Segundo parâmetro de ajuste da mobilidade do canal no modelo UMEM [V] | |
| W | Largura do canal | [μm] |
| L | Comprimento do canal | [μm] |
| I_0 | Corrente de fuga | [A] |

| | | |
|-------------|---|-------|
| m | Parâmetro de ajuste da região do joelho da curva do modelo UMEM | |
| λ | Modulação do comprimento do canal | [1/V] |
| V_{DSsat} | Tensão de saturação | [V] |
| α_s | Parâmetro de ajuste da região de saturação da curva característica de saída do modelo UMEM | |
| ξ | Parâmetro de ajuste do modelo UMEM relacionado à presença de contatos não-ôhmicos no transistor | |
| TOX | Espessura do dielétrico | [nm] |

Capítulo 1

Introdução

1.1 Contextualização

A história da eletrônica remete aos primeiros anos de século XX, destacando-se a invenção de Lee de Forest's: o triodo a tubo de vácuo, consequência do trabalho anterior de Ambrose Flaming, que trouxe ao mundo o diodo a tubo de vácuo, também chamado de retificador a vácuo [3].

Forest basicamente incluiu um terceiro terminal entre anodo e catodo do dispositivo de Ambrose, sendo assim capaz de controlar o fluxo de carga elétrica entre os dois terminais. O triodo a tubo de vácuo foi base para um outro estudo, realizado anos depois por Edgar Lilienfeld, que registrou uma patente descrevendo um aparato capaz de controlar a corrente elétrica entre dois terminais através de um terceiro terminal [3].

A ideia de Lilienfeld foi o alicerce principal para o desenvolvimento de uma tecnologia que viria a revolucionar a eletrônica e, conseqüentemente, o mundo: o transistor de efeito de campo metal-óxido-condutor, ou em inglês, *metal oxide semiconductor field effect transistor* (MOSFET). Esse dispositivo, desde sua concepção na década de 50, passou a ser largamente utilizado no desenvolvimento das telecomunicações, computação e demais áreas da engenharia elétrica, sendo utilizado em grande escala em praticamente todos os aparelhos eletrônicos que se tornaram parte da vida moderna [3].

O sucesso do MOSFET deve-se muito ao desenvolvimento do manuseio de um material semicondutor específico: o silício. Algumas empresas, como a IBM, chegaram a fabricar MOSFETs com outros semicondutores, sendo um exemplo o germânio e as ligas de silício e germânio (SiGe). Além disso, outros materiais foram testados durante os anos, como o arseneto de gálio, porém o silício mostrou-se o melhor material para ser utilizado no canal de MOSFETs a serem fabricados em larga escala de produção [3].

Além de suas numerosas aplicações tecnológicas, os transistores de efeito de campo também são utilizados no estudo de transporte de carga em materiais sólidos, graças ao acesso direto a mobilidade dos portadores de carga que esse dispositivo permite. Um exemplo disso é o transistor construído a partir da estrutura de silício amorfo hidrogenado (a-Si:H). Esse transistor é concebido

a partir de uma arquitetura diferente dos MOSFETs, sendo chamado de TFT (*thin-film transistor*, ou em português, transistor de filme fino). O TFT difere do MOSFET no processo de fabricação: o TFT é feito depositando-se finas películas do material semiconductor e do material dielétrico, além dos contatos de metal, em cima de um substrato não condutor (geralmente vidro). Já no MOSFET, o material semiconductor é, tipicamente, o substrato. Recentemente, o TFT deixou de ser um dispositivo utilizado apenas para estudos e passou a ser empregado na fabricação de telas de LCD de matriz ativa (AM-LCD) [3].

Devido à sua baixa mobilidade de portadores, semicondutores orgânicos passaram a ser empregados nas tecnologias TFT. Por muitos anos, os transistores orgânicos de filme fino foram deixados de lado devido a sua baixa performance, sendo estudados apenas por alguns grupos acadêmicos, que focaram seus esforços em melhorar a mobilidade do canal desses transistores. A partir do momento em que esse parâmetro foi aprimorado a ponto de ultrapassar os TFTs de silicone amorfo hidrogenado foi que grupos industriais decidiram adentrar na pesquisa dos OTFTs (*organic thin film transistor*) [3].

Um novo conjunto de aplicações pôde surgir com o desenvolvimento dos OTFTs devido ao seu baixo custo de fabricação. Dentre esses novos produtos, destacam-se os identificadores de radiofrequência, capazes de substituir os leitores de código de barras atuais; sensores de baixo custo e *displays* flexíveis.

No decorrer das últimas duas décadas, foi desenvolvido um ramo da eletrônica focado em modelar os diferentes dispositivos semicondutores que vêm sendo fabricados em grande quantidade nessa era. Os modelos concebidos para descrever os dispositivos são muito variados, sendo alguns microscópicos (baseados na física dos dispositivos) e modelos macroscópicos, utilizados para circuitos integrados completos [3].

Os modelos microscópicos são muito complexos, enquanto que os macroscópicos pecam na inexatidão. Para solucionar esse problema foram desenvolvidos os chamados modelos compactos, que tentam reunir exatidão e baixa complexidade. Esses modelos geralmente são implementados em linguagens de descrição de hardware (HDL): Verilog-A e VHDL-A [3].

1.2 Objetivos do projeto

O objetivo deste trabalho é verificar a função de uma nova infra-estrutura que une um simulador geral de circuitos, Qucs, e um método unificado de modelagem e extração de parâmetros (UMEM) para transistores orgânicos de filme fino. Será verificada a aplicabilidade dessa infra-estrutura para uma tecnologia ainda não publicada, bem como a possibilidade de que esse conjunto (QUCS + UMEM) seja utilizado como ferramenta para o Laboratório de Dispositivos e Circuitos Integrados (LDCI) da Universidade de Brasília. Como o UMEM propõe tanto a extração de parâmetros quanto a modelagem de OTFTs, ambas as partes desse modelo serão testadas e será verificado se os parâmetros extraídos, ao serem inseridos em um simulador de circuitos através de um código em Verilog-A, reproduzem com fidelidade os dados experimentais.

1.3 Apresentação do relatório

O capítulo 2 apresenta uma revisão bibliográfica teórica, onde é explicado brevemente como o OTFT é fabricado e como ele se comporta. Além disso, é mostrado como, tradicionalmente, as modelagens são feitas no campo de estudo da eletrônica e explicado o que é um código implementado em Verilog-A, utilizado para as simulações do dispositivo UMEM. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento do projeto. Resultados obtidos são discutidos no capítulo 4, seguido das conclusões no capítulo 5. Os anexos contêm material complementar.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

Neste capítulo será realizada uma breve revisão dos conceitos mais importantes envolvidos no trabalho, permitindo assim uma melhor compreensão das discussões realizadas e dos dados obtidos a serem mostrados nos próximos capítulos.

2.2 OTFT

2.2.1 Definição

Um semicondutor é um material que se comporta como um isolante em baixas temperaturas, mas apresenta uma considerável condutividade à temperatura ambiente. A distinção entre um isolante e um semicondutor não é bem definida, mas pode-se dizer que o semicondutor é um isolante com um gap pequeno o suficiente para que sua banda de condução seja sensivelmente povoada termicamente à temperatura ambiente [3]. O que torna os semicondutores tão importantes para a eletrônica é que suas propriedades eletrônicas podem ser alteradas de forma controlável através da adição de pequenas quantidades de impurezas. Esse é o bastante conhecido processo de dopagem, que está relacionado ao conceito de semicondutores extrínsecos [3].

Essa definição vem da Teoria de Bandas. Essa Teoria faz uma distinção clara entre um metal, que tem uma banda de condução parcialmente preenchida, e um isolante, caracterizado por uma banda de valência preenchida e uma banda de condução vazia. Os isolantes são, no entanto, apenas perfeitamente isolantes em condições de temperatura $T = 0$ K. Assim que a sua temperatura é elevada, os elétrons podem ser termicamente excitados e saírem da banda de valência para a banda de condução. Como a condução elétrica pode ocorrer em bandas parcialmente preenchidas, tanto a banda de valência quanto a banda de condução podem contribuir para a ocorrência desse fenômeno. Por conta do baixo valor da energia térmica ($kT = 25$ meV, à temperatura ambiente), esta condução termicamente ativada só pode ser observada em isolantes de baixa energia de gap, também conhecidos como semicondutores intrínsecos [3].

Dopar um semicondutor resulta em induções de níveis de energia localizados perto do limite da banda de condução (dopagem tipo n) ou da banda de valência (dopagem tipo p). Por conseguinte, a energia necessária para promover um elétron (ou buraco) na banda de condução (ou valência) é substancialmente reduzida a um nível comparável com a energia térmica. Quando se trata da dependência entre condutividade e temperatura, um semicondutor pode estar em três distintas regiões: a região intrínseca, nas altas temperaturas, caracterizada por uma alta dependência entre condutividade e temperatura; a região de saturação (esgotamento), a temperaturas médias, em que a condutividade praticamente não depende da temperatura; e a região de congelamento, nas baixas temperaturas, na qual os portadores estão congelados. Em termos de dispositivos eletrônicos, o único domínio de interesse é o regime de esgotamento, no qual a densidade de portadores de carga é praticamente igual a do dopante [3].

É essencial observar que a dopagem requer elevado grau de pureza. Na realidade, com exceção de alguns compostos inorgânicos utilizados na microeletrônica, a maioria dos semicondutores contém aproximadamente quantidades iguais de impurezas do tipo P e do tipo N; tais materiais são denominados compensados e se comportam como os semicondutores intrínsecos. Quase todos os semicondutores orgânicos pertencem a essa categoria. Em particular, o efeito de dopagem intencional de semicondutores orgânicos os coloca muito mais próximos de condutores do que de semicondutores, tornando-os inúteis para a microeletrônica [3].

Em conclusão, podemos afirmar que os semicondutores orgânicos estão mais perto de isolantes do que de semicondutores. Devido a seu grau de pureza demasiado baixo, a dopagem em semicondutores orgânicos ainda não foi desenvolvida para que estes dispositivos fossem usados de forma útil na microeletrônica. O seu potencial em constituir a base de dispositivos eletrônicos é, no entanto, agora bem estabelecido. Isso tornou-se possível usando um meio alternativo de indução de portadores de carga, chamado de injeção de eletrodos [3].

2.2.2 Arquitetura de transistores de filme fino

A estrutura dos OTFTs é bastante variável. A grosso modo, o OTFT, ou transistor orgânico de filme fino, é um transistor constituído basicamente por três partes: um isolante, uma fina camada de semicondutor orgânico e três eletrodos. Dois desses eletrodos, a fonte e o dreno, são ligados diretamente na camada do semicondutor. O terceiro eletrodo, porta, não tem contato com o semicondutor, pois está ligado ao material isolante [3]. A estrutura de um OTFT depende não só de seu modo de operação, como também da sua fabricação. O esquema básico de fabricação consiste em empilhar finos filmes dos diferentes elementos. Como a maioria dos semicondutores orgânicos são frágeis, a deposição de semicondutores orgânicos no isolante é muito mais fácil do que o inverso. Assim, a grande maioria dos OTFTs atuais são construídos de acordo com a arquitetura *bottom-gate* (porta na parte inferior), que por sua vez se divide em duas alternativas ilustradas na figura 2.1: *top contact* (TC), na qual o contato fica na parte superior e *bottom-contact* (BC), com o contato na parte inferior. Cada uma destas estruturas tem as suas vantagens e desvantagens. Na estrutura BC, os contatos são depositados sobre o isolante; se o último é um óxido inorgânico, por exemplo óxido de silício, os eletrodos podem ser modelados por meio de

técnicas de microlitografia. Isso não é possível com a arquitetura TC; com esta arquitetura os contatos são depositados através de máscaras de sombra, com perda substancial de resolução. Por outro lado, a resistência de contato tem sido relatada como mais baixa nos dispositivos TC do que nos dispositivos BC [3].

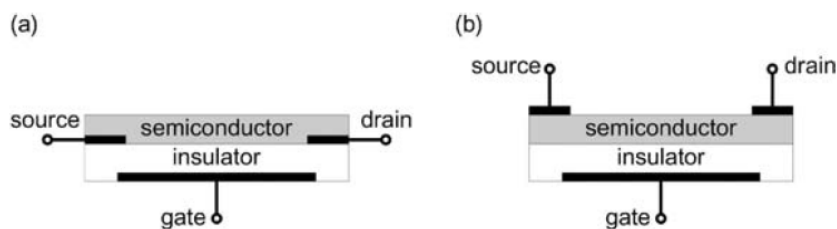


Figura 2.1: Estruturas de OTFTs, ambas com a porta na região inferior. (a) Bottom contact (BC); (b) Top contact (TC).

2.2.3 Funcionamento

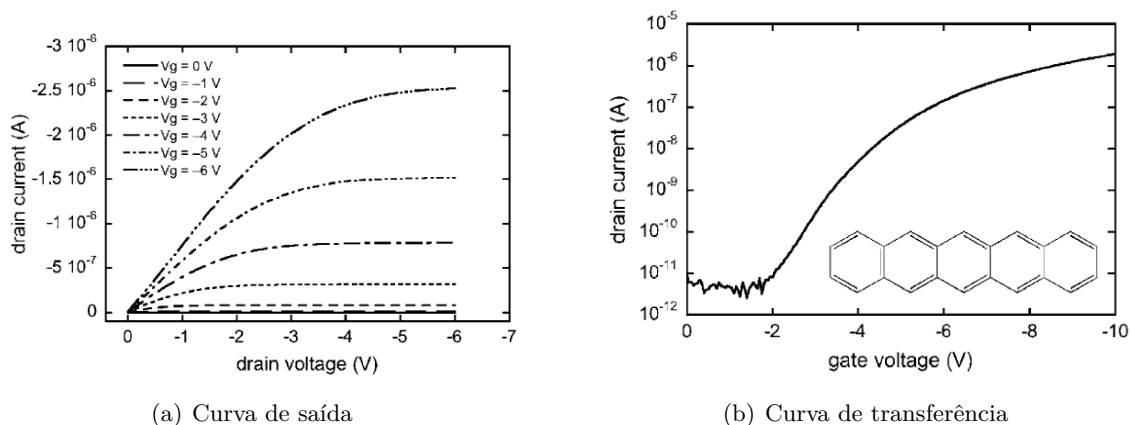


Figura 2.2: Curvas de saída (a) e transferência (b) de um típico OTFT. A figura (b) mostra ainda a estrutura molecular do pentaceno, utilizado como o semicondutor desse dispositivo.

Na figura 2.2 estão apresentadas curvas de saída e uma curva de transferência de um transistor feito de pentaceno, que é a estrutura química presente na inserção, com os contatos da fonte e do dreno feitos de ouro. O esquema de energia na figura 2.3 indica a posição do nível de Fermi do ouro e os orbitais de fronteira: orbital mais alto ocupado, HOMO (do inglês *Highest Occupied Molecular Orbital*), e menor orbital molecular desocupado, LUMO (do inglês *Lowest Unoccupied Molecular Orbital*) do pentaceno. Os dados para o pentaceno foram obtidos em [4]. Para a discussão que se segue, considera-se a fonte como o terminal de referência [3].

Quando uma tensão positiva é aplicada à porta, cargas negativas são induzidas na fonte. Como pode-se observar na figura 2.3, o nível de Fermi do ouro está bem distante do LUMO do pentaceno, com isso, os elétrons têm uma forte barreira de energia a ultrapassar, sendo assim a injeção de elétrons muito pouco provável. Dessa maneira, não ocorre passagem de corrente através da camada

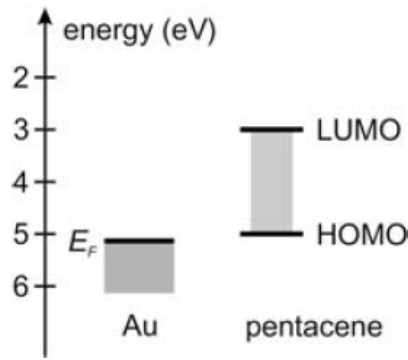


Figura 2.3: Esquema de energia do transistor de pentaceno-ouro

de pentaceno. Os pequenos valores de corrente observados na figura 2.2(a) devem-se a correntes de fuga através do isolante. Em contraste, quando a tensão da porta é revertida para negativa, os buracos são facilmente injetados porque o nível de Fermi está perto do nível HOMO e a altura da barreira de energia é baixa. Um canal condutor é formado na interface isolante-semicondutor, e portadores de carga podem ser conduzidos a partir da fonte para o dreno através da aplicação de uma segunda, independente, tensão de polarização no dreno. Já que buracos são mais facilmente injetados que elétrons, o pentaceno é considerado do tipo p. Simetricamente, um semicondutor orgânico é dito ser do tipo n quando a injeção de elétrons é mais fácil que a injeção de buracos [3].

Basicamente, o OTFT funciona como um capacitor; quando uma tensão é aplicada na porta, uma carga igual (mas de sinal oposto), é induzida em ambos os lados do isolante. No lado do semicondutor essa carga forma um canal de condução, caso os portadores de carga possam ser injetados dentro do material semicondutor. A condutância do canal é proporcional à carga e também proporcional à tensão da porta. Em baixos valores para a tensão de dreno, a corrente segue a lei de Ohm; sendo assim proporcional às tensões de porta e de dreno.

Analisando a figura 2.2(a), nota-se que após um certo valor de tensão de dreno, a corrente passa a ser praticamente independente dessa tensão. Essa região da curva denomina-se região de saturação. A curva na figura 2.2(b) é a característica de transferência do transistor; ela nos mostra que o dispositivo apresenta uma alta relação de ligado-desligado, tornando-o bastante aplicável em circuitos digitais [3].

As correntes na região linear e na região de saturação podem ser encontradas através das expressões (2.1) e (2.2), respectivamente, que são obtidas a partir de duas premissas [3]:

1. O campo elétrico longitudinal do canal é muito menor do que o seu campo elétrico transversal. Isso é a chamada aproximação progressiva de canal, que é válida quando a distância entre a fonte e o dreno é muito maior do que a espessura do isolante.

2. a mobilidade, μ , é constante.

$$I_{Dlin} = \frac{W}{L} \cdot \mu \cdot C_i \cdot (V_G - V_T)V_D, \quad (2.1)$$

$$I_{\text{Dsat}} = \frac{W}{2L} \cdot \mu \cdot C_i \cdot (V_G - V_T)^2, \quad (2.2)$$

onde W e L são, respectivamente, a largura e o comprimento do canal, C_i é a capacitância do isolante por unidade de área, V_G e V_D são as tensões de porta e dreno, respectivamente, e V_T é a tensão de limiar, que leva em consideração diversas quedas de potencial elétrico através da estrutura porta-isolante-semicondutor. Essas equações são bastante simplificadas e possuem muitas limitações ao se analisar OTFTs [3].

2.2.4 Resistência de contato

A resistência dos contatos dos OTFTs é uma grandeza física que deve ser levada em conta no desenvolvimento desses dispositivos, e as limitações criadas por essa grandeza devem ser diminuídas ao máximo. A redução dessas limitações é um ponto chave do desenvolvimento da eletrônica orgânica atualmente [3].

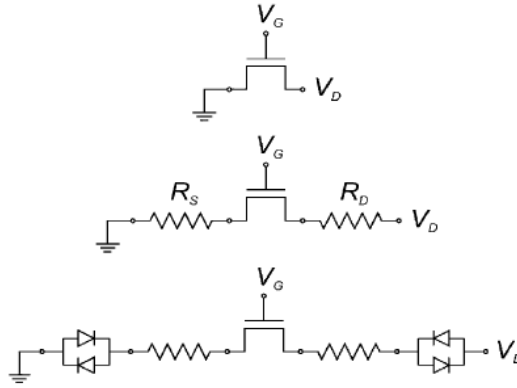


Figura 2.4: Circuitos equivalentes de um TFT incluindo as resistências dos contatos. O último circuito inclui diodos para modelar resistências não-lineares

A figura 2.4 mostra dois circuitos equivalentes para o problema das resistências de contato. No primeiro circuito, existe uma resistência R_D representando a resistência do contato do dreno e R_S representando a resistência do contato da fonte. No segundo circuito, são adicionados diodos em paralelo (com polaridades opostas entre si) em série com essas resistências, no intuito de representar a existência de possíveis resistências não-ôhmicas dos contatos.

Muitos estudos têm sido realizados no intuito de modelar o efeito das resistências de contato, como em [5], [6], [7], [8], [9]. As principais conclusões desses estudos até o momento estão resumidas a seguir:

1. A resistência do contato depende fortemente da natureza do eletrodo, por exemplo sua função trabalho.

2. A resistência do contato é dependente da tensão da porta, apresenta um decréscimo substancial quando a polarização da porta é aumentada.

3. A mobilidade é dependente da polarização da porta.

Segundo o modelo de Mott-Schottky (MS), espera-se que a resistência do contato apresente um comportamento ôhmico quando a função trabalho do metal encontra-se próxima do HOMO ou LUMO do semiconductor, dependendo se ele é do tipo n ou p. Se essa condição não é cumprida, espera-se que uma barreira de potencial se forme na interface metal-semiconductor, tornando o transporte de cargas muito baixo. Por esse ponto de vista, a dupla Au/pentaceno seria um bom candidato a contato devido a uma baixa resistência esperada, porém, experimentos mostraram que a resistência observada dessa interface era bem mais alta que a prevista [3].

Estudos utilizando diodos emissores de luz orgânicos (OLEDs) foram realizados para determinar o mecanismo de formação de barreiras de potencial em interfaces formadas entre metais e semicondutores orgânicos. Tais estudos utilizaram-se das técnicas de espectroscopia de fotoelétrons UV (UPS) e UPS inversa para determinar precisamente os níveis de energia dos dois lados da interface [10].

A figura 2.5 mostra o resultado desses experimentos. Claramente, nota-se que o resultado é bem diferente do esperado pelo modelo MS. A interface possui uma barreira “dipolo” adicional, Δ , que desloca o nível do HOMO para baixo em mais de 1 eV, aumentando assim o tamanho da barreira no mesmo valor [3].

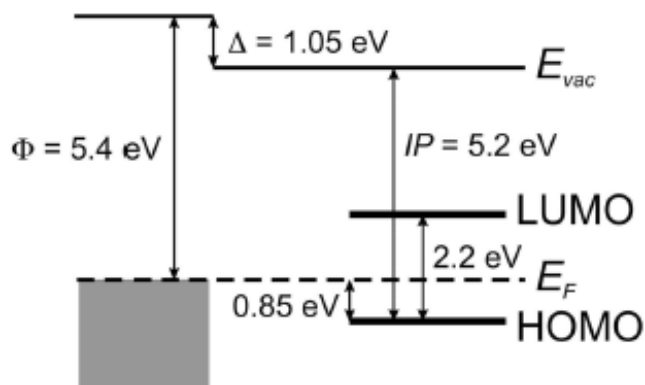


Figura 2.5: Diagrama de níveis de energia da interface ouro-pentaceno, apresentado em [10]

2.2.5 Transporte de carga

O transporte de carga nos materiais orgânicos é um tema ainda bastante controverso, mesmo com o progresso dessa tecnologia ao longo das três últimas décadas. Resumidamente, pode-se dividir esses materiais em duas grandes famílias de modelos de transporte de carga. A primeira delas é a dos materiais desordenados, como os polímeros. Um modelo sobre esse tipo de material é apresentado em [11]. O modelo prevê uma mobilidade ativada termicamente e dependente da tensão de porta seguindo uma relação de potência [3].

A análise de cristais moleculares bem organizados é uma tarefa bem mais difícil. Problemas básicos como encontrar o valor final da mobilidade e a sua dependência da temperatura ainda não

foram resolvidos de forma satisfatória [3].

Um aspecto interessante é a dependência da mobilidade em relação à tensão de porta. Resumidamente, a dependência deriva do fato de que à medida que a tensão de porta aumenta, os portadores de carga injetados tendem a preencher as armadilhas. Com isso, as armadilhas tornam-se menos efetivas e o transporte de carga aumenta. Esse aspecto foi observado tanto em polímeros quanto em componentes feitos de pequenas moléculas [3].

2.2.6 Técnicas de fabricação

Os semicondutores orgânicos são muito diferentes dos inorgânicos em termos de ponto de fusão, solubilidade e propriedades mecânicas. Por isso, é previsível que as técnicas de fabricação convencionais utilizadas na concepção de semicondutores inorgânicos não sejam aplicáveis aos semicondutores orgânicos. Apesar disso, os primeiros componentes microeletrônicos orgânicos utilizavam essas técnicas de fabricação como a oxidação térmica e a fotolitografia [3].

A utilização dessas técnicas de fabricação não é apropriada para que se aproveitem as vantagens dos materiais orgânicos, como o seu baixo custo, grande área e a possibilidade de utilização de um substrato flexível.

Um dos processos utilizados atualmente é o proposto em [12]: A impressão de microcontatos utiliza selos elastoméricos para imprimir camadas, utilizadas tanto para prevenir a remoção do material pré-depositado como para ser um iniciador para a deposição de outro material. A fabricação se inicia com a deposição de ouro sobre um substrato preparado anteriormente. Depois, o ouro desprotegido pelas camadas impressas são gravados, definindo assim os eletrodos do dreno e da fonte. As camadas são então removidas com calor ou luz ultravioleta. O semicondutor, o isolante e o eletrodo da porta são sequencialmente fundidos a partir de uma solução.

Uma segunda técnica é exposta em [13]. Consiste numa adaptação da já conhecida técnica de impressão a jato de tinta, largamente utilizada na fabricação de OLEDs, por exemplo. O principal problema ao se utilizar essa técnica para OTFTs é a resolução de tamanho limitado, entre 20 μm e 50 μm , por conta do espalhamento das gotículas de tinta pelo substrato. A solução proposta em [13] consiste na limitação do espalhamento das gotículas de tinta a base d'água utilizando regiões hidrofóbicas para limitar o dispositivo, fazendo assim com que tais gotículas sejam repelidas e confinadas dentro da região.

O principal problema das técnicas apresentadas acima é a utilização de tintas líquidas, que pode ser um empecilho ao se trabalhar com conjugados de polímeros e moléculas pequenas. Em [14], um processo seco que toma como base imagens térmicas foi proposto como uma possível solução para o problema. Essa técnica utiliza a transferência de uma fina camada de película sólida a partir de um doador para um receptor flexível.

2.2.7 Materiais utilizados em OTFTs

Os materiais orgânicos semicondutores são, em geral, divididos entre polímeros e pequenas moléculas. Os polímeros mais utilizados são os polifluorenos e o poli(3-alquiltiofeno) (P3AT). O desempenho desses polímeros ainda é algumas ordens de grandeza menor do que o desempenho das pequenas moléculas, já que a performance de OTFTs feitos de polímero depende fortemente da ordenação química e estrutural das cadeias na interface isolante-polímero. [3].

Das pequenas moléculas, a mais utilizada na fabricação de OTFTs é o pentaceno. Os oligotiofenos e seus derivados também apresentam sua importância. Para o pentaceno, foram documentadas mobilidades maiores que $6 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ [14], enquanto que para os oligotiofenos os maiores valores encontrados são pouco maiores que $1 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ [15].

2.3 Modelos Compactos

Com o rápido e progressivo desenvolvimento da tecnologia de semicondutores, tornou-se necessário o desenvolvimento de modelos computacionais que fossem capazes de reproduzir as características desses componentes. Na intenção de se utilizarem esses dispositivos em circuitos é necessário: do ponto de vista tecnológico, a possibilidade de sua integração no mesmo substrato; do ponto de vista dos desenvolvedores, ferramentas que realizem a simulação desses circuitos.

Os primeiros modelos possuíam uma complexa estrutura computacional, sendo necessária a utilização de diversos métodos numéricos, que tornavam todo o processo mais lento e trabalhoso. Outros códigos, entretanto, não possuíam tamanha complexidade. Entretanto, os códigos menos complexos apresentavam resultados menos satisfatórios. Tentando unir exatidão dos resultados com um menor esforço computacional a ser empregado surgiram os modelos compactos.

A principal utilidade dos modelos é o estudo do comportamento desses dispositivos, permitindo assim a melhoria deles. Todos os tipos de modelos, inclusive os compactos, possuem as mesmas necessidades para que sejam desenvolvidos: conhecimento das características físicas do dispositivo; aproximações dos aspectos teóricos, com o intuito de facilitar os cálculos dos parâmetros básicos do dispositivo; métodos e ferramentas matemáticas para realizar os cálculos; métodos de extração de parâmetros; verificação da compatibilidade do modelo com um dispositivo real, via experimento ou simulação; aplicação dos resultados.

A figura 2.6 ilustra a hierarquia básica dos modelos computacionais:

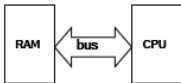
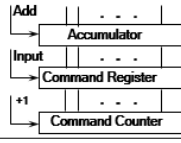
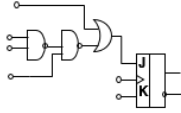
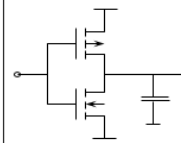
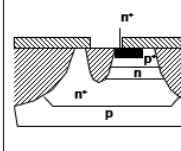
| Nível | Objeto de modelagem | Exemplo | Ferramenta matemática | HDL |
|--------------------------|--|---|---|------------------------|
| Sistema | Circuitos Estruturais |  | Enfileiramento | SystemVerilog, SystemC |
| Registradores-Contadores | Circuitos funcionais no nível de componentes multibits |  | Álgebra Booleana | Verilog, VHDL |
| Portas lógicas | Circuitos no nível de portas lógicas e flip-flops |  | | |
| Circuito | Circuito elétrico |  | Sistema de equações diferenciais | Spice |
| Componentes | Componentes de circuitos integrados |  | Sistema de equações diferenciais com derivadas parciais | |

Figura 2.6: Hierarquia dos modelos computacionais, do menos descritivo ao mais descritivo.

2.4 Verilog-A

Verilog-A é uma linguagem de modelagem de componentes de circuitos analógicos. É a sub linguagem responsável por componentes analógicos da linguagem Verilog-AMS (Verilog-Analog and Mixed Signals)[16].

2.4.1 Vantagens da linguagem Verilog-A

Desde 1985, a principal linguagem utilizada para a implementação de modelos compactos tem sido C [16] . Antes do C se tornar a linguagem líder, FORTRAN era a linguagem mais utilizada para descrever modelos compactos [16]. Como os componentes semicondutores foram se tornando cada vez menores em sua geometria, novos efeitos físicos passaram a afetar o comportamento desses dispositivos [17]. Atualizar os códigos em C ou FORTRAN de acordo com essas mudanças tornou-se muito trabalhoso e demandava muito tempo, já que, nessas linguagens, as equações diferenciais parciais de corrente e carga deveriam ser calculadas e depois traduzidas para a linguagem utilizada [17].

Nos últimos anos, o Verilog-A se tornou cada vez mais adotada como a principal linguagem por desenvolvedores de modelos compactos [16]. O MATLAB também é uma linguagem que pode ser utilizada para modelos compactos, graças a sua simplicidade em relação às C e FORTRAN, e a

eficácia com que dados podem ser manipulados e gráficos podem ser plotados. Porém, os modelos em MATLAB não podem ser utilizados em simuladores de circuitos, como o SPICE. A linguagem Verilog-A é quase tão simples quanto o MATLAB, e seus modelos podem ser usados diretamente nos simuladores de circuito, bem como em softwares de extração de parâmetros [17].

Atualmente, a maioria dos simuladores comerciais de circuitos suportam a linguagem Verilog-A (apesar disso, não é uma tarefa fácil encontrar simuladores gratuitos que suportem a linguagem), tornando-a, além de mais simples que as demais no que diz respeito a modelos compactos, praticamente universal [17]. Isso já é, por si só, uma grande vantagem dessa linguagem em relação as outras existentes, porém a Verilog-A também apresenta outras vantagens. A principal delas é que Verilog-A para modelos compactos livra o desenvolvedor de algumas tarefas trabalhosas, como: leitura dos parâmetros do modelo, checagem da topologia, alocação de memória, entre outros. Os desenvolvedores que utilizam Verilog-A não precisam se preocupar com esses detalhes [17].

Simuladores de Verilog-A computam automaticamente as derivadas parciais das correntes e cargas de um modelo compacto e determinam a inserção apropriada desses valores na matriz Jacobiana utilizada no método numérico de Newton-Raphson, por exemplo. Em C, essa tarefa deve ser feita manualmente pelo desenvolvedor. Assim, mesmo que o desenvolvedor esteja trabalhando com um modelo já existente em C, qualquer alteração será bastante trabalhosa, pois ao se introduzir uma nova dependência no modelo, as derivadas devem ser calculadas e essas derivadas são de difícil verificação. Como resultado, quase todos os códigos em C de modelos compactos apresentam alguns erros nas derivadas na sua primeira versão [17].

Um efetivo sistema de declaração de parâmetros dos modelos também está presente na linguagem Verilog-A. Os valores padrões dos parâmetros podem ser declarados diretamente no código. Por exemplo, alguns parâmetros são diferentes para PMOS ou NMOS e isso pode ser incorporado diretamente nas instruções de declaração, ao invés de requererem um código especial para especificar essas diferenças [17].

2.5 Qucs

O Qucs, abreviação de *Quite Universal Circuit Simulator*, é um simulador de circuito integrado produzido pela Qucs Team. Com ele, o usuário é capaz de estabelecer um circuito através de uma interface gráfica com o usuário (GUI) e simular: comportamento de grandes sinais, pequenos sinais e ruídos do circuito. Após a simulação finalizada, o usuário pode ver os resultados em uma janela onde podem ser plotados gráficos, tabelas, etc.[18].

O Qucs é um software avançado que permite a obtenção de resultados em vários tipos de diagramas. Análise DC, análise AC, ruídos, Parâmetros-S, análise de transientes, tudo isso é possível de ser simulado com essa ferramenta. Além disso, o Qucs é capaz de importar modelos SPICE para serem usados em sua interface. [18].

Qucs é desenvolvido para os sistemas operacionais GNU/Linux OS. Entretanto, o simulador funciona de forma satisfatória em outros sistemas operacionais como: Windows, Solaris, NetBSD,

FreeBSD, MacOS e Cygwin. No caso desse projeto, foi utilizada a versão 0.0.18 do Qucs executada no sistema operacional Windows. [18].

A escolha do Qucs como simulador de circuitos para o presente trabalho se deve a alguns fatores importantes: o Qucs é distribuído através de uma licença *open source* do tipo *General Public License* (GPL), sendo assim uma ferramenta que pode ser instalada em qualquer computador pessoal. Realizar esse projeto em um software pago demandaria recursos financeiros e tempo para a obtenção da licença. Além disso, o Qucs é praticamente o único simulador de circuitos, dentre os gratuitos, a entender a linguagem Verilog-A. [18].

Os modelos compactos de transistores orgânicos estão sempre em constante atualização, por isso, é interessante que se utilize uma ferramenta na qual essas modificações podem ser feitas rapidamente. Utilizando-se os códigos Verilog-A, essas atualizações podem ser feitas de maneira ágil. [18].

A interface gráfica do Qucs é bastante simples, tornando-o um software de fácil utilização e o tempo investido para se aprender a mexer nele é curto se comparado a outros simuladores, facilitando o trabalho do usuário.[18].

2.6 UMEM

O UMEM (*unified model and parameter extraction method*) foi apresentado em [19], tendo sido desenvolvido para modelar e extrair parâmetros de transistores de película fina de a-Si:H, nas regiões linear e de saturação. Modelos compactos implementados em AIM-Spice para descrever esses dispositivos já haviam sido previamente apresentados em outros artigos, porém tais modelos apresentavam mais de 15 parâmetros a serem manipulados durante as simulações, tornando o esforço computacional empregado no processo um empecilho [19].

O AIM-Spice fornece, como um programa adicional, um extrator de parâmetros, que otimiza muitos dos parâmetros do modelo na tentativa de ajustar as curvas simuladas àquelas obtidas experimentalmente. Em muitos casos, esse extrator é problemático e um ajuste simultâneo das curvas $I_{DS}-V_{GS}$ e $I_{DS}-V_{DS}$ simuladas com as experimentais nem sempre é possível para o mesmo conjunto de parâmetros [19].

Através de um conjunto de equações, o UMEM propõe a extração de parâmetros das curvas de transferência e de saída do transistor, sem que seja necessária a utilização de otimização não-linear ou processamento de dados gráficos. Dessa maneira, $I_{DS}-V_{GS}$ e $I_{DS}-V_{DS}$ são calculados usando as expressões do modelo [19].

Em [1], o UMEM foi utilizado para modelar a região acima do limiar de OTFTs com diferentes geometrias, processos de fabricação e materiais dielétricos. Em [19], foi mostrado que a mobilidade de portadores no modelo UMEM é dada por:

$$\mu_{FET} = \mu_0 \left[\frac{(V_{GS} - VT)}{V_{aa}} \right]^{\gamma_a}, \quad (2.3)$$

onde μ_{FET} é a mobilidade para campos elétricos pequenos, longitudinais e perpendiculares, V_T é a tensão de limiar e γ_a e V_{aa} são parâmetros de ajuste, μ_0 é a mobilidade de banda do material utilizado no TFT. γ_a está relacionado com a condutividade, descrevendo assim se a relação entre V_{GS} e a mobilidade é diretamente ou inversamente proporcional, ou seja, γ_a pode ser maior ou menor que 0 [1].

A corrente do dreno, tanto para a região de saturação quanto para a linear é dada por:

$$I_{\text{DS}} = \frac{W}{L} \cdot C_{\text{diel}} \frac{\mu_{\text{FET}} \cdot (V_{\text{GS}} - V_T)}{\left(1 + R \frac{W}{L} \cdot C_{\text{diel}} \mu_{\text{FET}} \cdot (V_{\text{GS}} - V_T)\right)} \times \frac{V_{\text{DS}} (1 + \lambda \cdot V_{\text{DS}})}{\left[1 + \left[\frac{V_{\text{DS}}}{V_{\text{DSsat}}}\right]^m\right]^{\frac{1}{m}}} + I_0 . \quad (2.4)$$

onde W é a largura do canal, L é o comprimento do canal, R é a soma das resistências do dreno e da fonte, I_0 é a corrente de fuga e m e λ são parâmetros de ajuste relacionados com o formato da região do “joelho” da curva e a modulação do comprimento do canal, respectivamente.

A tensão de saturação, V_{DSsat} , é dada por:

$$V_{\text{DSsat}} = \alpha_S (V_{\text{GS}} - V_T) . \quad (2.5)$$

O parâmetro α_S é chamado de parâmetro de modulação de saturação. Esse parâmetro é utilizado para ajustar a tensão de saturação do transistor a ser estudado.

Em [19], foi definida a função $H(V_{\text{GS}})$, dada por:

$$H(V_{\text{GS}}) = \frac{\int_0^{V_{\text{GS}}} I_{\text{DS}}(x) dx}{I_{\text{DS}}(V_{\text{GS}})} . \quad (2.6)$$

Substituindo (2.3) em (2.4) e calculando $H(V_{\text{GS}})$ pela equação acima, chega-se na seguinte expressão:

$$H_a(V_{\text{GS}}) = \frac{1}{2 + \gamma_a} (V_{\text{GS}} - V_T) . \quad (2.7)$$

O passo a passo para extraírem-se os parâmetros de determinado transistor, utilizando o UMEM, segue a seguinte ordem:

Passo no.1 : o valor de V_T é obtido pela interseção, e γ_a pela inclinação de (2.7)

Passo no.2 : Neste passo, utilizam-se apenas os dados experimentais da região linear da curva de transferência. Calcula-se $(I_{\text{DS}})^{\frac{1}{1+\gamma_a}}$ vs. $(V_{\text{GS}} - V_T)$ e sua inclinação Sl . O valor de V_{aa} é então dado por:

$$V_{\text{aa}} = \left[\frac{K V_{\text{DS}}}{Sl^{1+\gamma_a}} \right]^{\frac{1}{\gamma_a}} \quad (2.8)$$

,

onde $K = \frac{W \mu_0 C_{\text{diel}}}{L}$. Dessa maneira, os três parâmetros que determinam a mobilidade do efeito de campo na região acima do limiar são extraídos.

Passo no.3: Usando-se a corrente de saturação característica para $V_{DS} \geq V_{GS} - V_T$, calcula-se a inclinação S_s na região linear do gráfico $I_{DSSat}^{-\frac{1}{2+\gamma_a}}$ vs. $(V_{GS} - V_T)$. Dessa maneira, encontra-se o parâmetro α_s através da expressão:

$$\alpha_s = \frac{S_s^{2+\gamma_a} V_{aa}^{\gamma_a} \sqrt{2}}{K}. \quad (2.9)$$

Passo no.4: O parâmetro m é calculado através de (2.4) na tensão V_{DSSat} para um valor de tensão de porta próxima da máxima medida, negligenciando-se R e γ :

$$m = \log 2 / \log \left[\frac{K \mu_{FET}(V_{GS}) \alpha_s (V_{GS1} - V_T)}{[1 + K \mu_{FET}(V_{GS})(V_{GS} - V_T)] I_{DSSat}(V_{DSSat})} \right]. \quad (2.10)$$

Para determinar $I_{DSSat}(V_{DSSat})$ escolhe-se uma curva de saída característica para um determinado $V_{GS} = V_{GS1}$. Esse valor escolhido deve ser uma tensão de porta próxima do valor que se quer modelar. Em seguida, escolhe-se um valor de V_{DS} na região de saturação não muito afastado do “joelho” da curva selecionada. O valor da corrente de dreno I_{DS1} é conhecida dos valores experimentais. Essa tensão de dreno V_{DS1} é igual a:

$$V_{DS1} = \alpha_s (V_{GS1} - V_T). \quad (2.11)$$

Esse valor de I_{DS1} corresponde ao termo $I_{DSSat}(V_{DSSat})$ da equação (2.10).

Passo no.5: O parâmetro λ é extraído pela equação:

$$\lambda = \left\{ \frac{\frac{(I_{DS2})}{(V_{DS2})^2} \cdot [1 + R \cdot K \cdot \mu_{FET}(V_{GS1}) \cdot (V_{GS1} - V_T)] \cdot \left[1 + \left[\frac{(V_{DS2})}{\alpha_s \cdot (V_{GS1} - V_T)} \right]^m \right]^{\frac{1}{m}}}{K \mu_{FET}(V_{GS1})(V_{GS1} - V_T)}} - \frac{1}{V_{DS2}} \right\}. \quad (2.12)$$

onde V_{DS2} é um valor escolhido não muito longe da tensão de dreno máxima medida na curva a ser modelada.

2.6.1 Modelagem de OTFTs com a presença de contatos não-ohmicos

É sabido que OTFTs podem apresentar contatos não lineares no dreno e na fonte. Os passos para se modelar essa região também foram apresentados em [1]. Na presença de contatos não-ohmicos, a tensão de polarização externa, V_{DSext} será dividida entre o contato não-ohmico (diodo) e o próprio transistor. Temos assim:

$$V_{DSext} = V_{diodo} + V_{DS} = \frac{I_{DS}}{G(V_{GS}, V_{DS})} + \frac{nkT}{q} \cdot \log\left(\frac{I_{DS}}{I_{do}}\right) = \frac{I_{DS}}{G(V_{GS}, V_{DSext}) \cdot \xi} + \frac{nkT}{q} \cdot \log\left(\frac{I_{DS}}{I_{do}}\right), \quad (2.13)$$

onde

$$G(V_{GS}, V_{DSext}) = \frac{K\mu_{FET}(V_{GS}) \cdot (V_{GS} - V_T) \cdot (1 + \lambda \cdot V_{DS})}{[1 + RK \cdot \mu_{FET}(V_{GS}) \cdot (V_{GS} - V_T)][1 + [\frac{V_{DS}}{V_{DSsat}}]^m]^{\frac{1}{m}}}. \quad (2.14)$$

e ξ é um parâmetro de ajuste para levar em conta a tensão real através do transistor quando a resistência do diodo é significativa. Diferente da equação (2.4), em (2.14) o valor de K é dado por $K = \frac{WC_{diel}}{L}$.

Para se obter as curvas de saída do transistor quando na presença de contatos não-ohmicos, é necessário obter os parâmetros do diodo após o quarto passo apresentado na seção 2.6.

2.6.2 Extração dos parâmetros do diodo

Depois que os parâmetros V_T , γ_a , α_S , V_{aa} e m forem determinados, escolhe-se uma curva característica com $V_{GS} = V_{GS2}$ na qual o efeito do contato não-ohmico pode ser claramente observado. Plota-se a curva $\log(I_{DS})$ vs. V_{DSext} e determina-se a inclinação, B , e a interseção, A , dessa curva perto da origem. Após isso utiliza-se as seguintes equações:

$$n = \frac{\log e}{B \cdot \frac{kT}{q}}. \quad (2.15)$$

e

$$I_{do} = 10^A. \quad (2.16)$$

A presença desse diodo reduz a corrente de dreno a pequenos valores, quando a resistência do diodo é alto. Como a tensão de polarização eleva-se além do joelho da curva $I-V$ do diodo, a corrente do dreno aumenta e começa a ser principalmente ditada pelo transistor. Isso significa que na curva característica de saída, próximo ou após a inclinação máxima nessa região de deformação, a maior parte da tensão aplicada começa a cair através do transistor. Por essa razão, o parâmetro ξ é determinado como:

$$\xi = \frac{\left[\left(\frac{I_{DS}(V_{GS2}, V_{ext2})}{G1(V_{GS2})} \right) - \left(\frac{I_{DS}(V_{GS2}, V_{ext1})}{G1(V_{GS2})} \right) \right]}{[V_{DS1} - V_{DS2}]}. \quad (2.17)$$

onde $I_{DS}(V_{GS2}, V_{ext2})$ e $I_{DS}(V_{GS2}, V_{ext1})$ são as correntes medidas para as curvas características de saída selecionadas em duas tensões de polarização externa V_{ext1} e V_{ext2} limitando uma região aproximadamente linear da curva I_{DS} vs. V_{DSext} , na qual a inclinação máxima dessa curva está incluída.

$G1(V_{GS2})$ é a condutância na região linear da curva $V_{GS} = V_{GS2}$:

$$G1(V_{GS2}) = \frac{K \cdot \mu_{FET}(V_{GS2}) \cdot (V_{GS2} - V_T)}{(1 + R \cdot \mu_{FET}(V_{GS2}) \cdot (V_{GS2} - V_T))}. \quad (2.18)$$

Essa condutância é utilizada para calcular o parâmetro ξ na equação 2.17. Nessa equação, R representa a soma das resistências de dreno R_D e de fonte R_S , que são iguais. A condutância G leva em consideração valores de V_{GS} externos, que são a soma de V_{GS} interno com o produto da corrente pela resistência de fonte.

O próximo capítulo explicará a metodologia utilizada neste relatório.

Capítulo 3

Metodologia

O primeiro passo do projeto foi obter uma familiarização com o software QUCS, executando simulações com um MOSFET presente na biblioteca original do simulador. Na sequência, utilizou-se o código em Verilog-A que descreve o modelo UMEM, criando assim um componente para que fosse utilizado num circuito simples, capaz de nos fornecer as características de saída e a curva de transferência do transistor em análise.

O passo seguinte foi a execução da extração de parâmetros do modelo UMEM a partir de um código já existente implementado no software Mathcad. A extração já inclui por si só, devido ao UMEM ser um modelo unificado, a reprodução das curvas que representam o OTFT em estudo.

Feito isso, partiu-se para a verificação de que o componente criado no QUCS realmente reproduzia os mesmos resultados que o Mathcad, dando entrada dos parâmetros extraídos no Qucs. Para reforçar essa verificação, também utilizou-se o simulador online *i*-MOS [20], que também utiliza códigos em Verilog-A para implementar modelos compactos. O UMEM já encontra-se disponível nessa plataforma online, tornando muito fácil a simulação com essa ferramenta. Uma outra parte da verificação consistiu em utilizar o transistor T1, publicado em [1], modelado com UMEM. Foi escolhida apenas a curva de saída para um $V_{GS} = -40V$ e tentou-se reproduzir a mesma curva no Qucs, fazendo uso dos valores dos parâmetros do modelo utilizados na simulação original, também fornecidos em [1]. Durante a tentativa de reprodução da curva de saída, foi observada atentamente a sensibilidade da curva aos parâmetros alterados.

Posteriormente, realizou-se um experimento computacional no *i*-MOS, “criando-se” um transistor novo com um conjunto de valores dos parâmetros do UMEM escolhido. Em seguida, inseriu-se as curvas I_{DS} vs. V_{DS} e I_{DS} vs. V_{GS} criadas pelo *i*-MOS no código de extração em Mathcad, no intuito de checar se os parâmetros pré-estabelecidos seriam corretamente extraídos de volta.

O passo seguinte foi aplicar o modelo a dados experimentais de uma tecnologia ainda não publicada fornecidos pela Technische Universität Dresden, na Alemanha. Os dados fornecidos contêm valores de I_{DS} em função de V_{DS} e valores de I_{DS} em função de V_{GS} . Com esses dados, pode-se plotar as características de saída e a curva de transferência do transistor, respectivamente. Foi feito um ajuste manual entre o modelo e os dados experimentais utilizando apenas o Qucs.

3.1 *i*-MOS

O *i*-MOS, *interactive Modeling and Online Simulation Platform*, é uma plataforma online aberta para que desenvolvedores de modelos e designers de circuitos possam interagir. Desenvolvedores de modelos podem implementar seus modelos na plataforma *i*-MOS para promover a sua aceitação e obter feedback dos usuários. Os designers de circuitos podem usar a plataforma para testar os mais recentes modelos de muitos dispositivos recém-desenvolvidos antes que eles sejam lançados no mercado. A plataforma fornece uma interface padrão para que os usuários possam avaliar e comparar os modelos facilmente. Atualmente, o *i*-MOS só fornece como saída as características dos modelos, mas seus desenvolvedores prometem que, em breve, seja possível realizar simulações online no servidor do *i*-MOS.

Um dos modelos já implementados na plataforma é o UMEM, e essa implementação deu-se através do mesmo código Verilog-A utilizado no presente trabalho, por isso o *i*-MOS foi escolhido como uma segunda ferramenta para checar a funcionalidade do código.

3.2 Mathcad

O Mathcad é um software comercial de matemática, engenharia e outras ciências exatas fornecido pela Parametric Technology Corporation. O Mathcad funciona através de uma planilha, aonde as equações e expressões são inseridas e manipuladas no mesmo formato gráfico em que são apresentadas (na computação, chama-se essa característica de *What You See Is What You Get*).

O Mathcad foi utilizado nesse projeto como ferramenta de extração de parâmetros através de um código já pronto no qual o UMEM foi implementado. Esse código foi escrito pelos mesmos desenvolvedores do UMEM. Portanto, para esse projeto, apenas executou-se o código e obteve-se os parâmetros e curvas, ou seja, num primeiro momento, nenhuma alteração foi feita nesse código.

3.3 Simulação de um MOSFET no Qucs

Essa seção visa introduzir o leitor ao software Qucs. Um exemplo de simulação utilizando um n-MOSFET será realizado com fins didáticos, e as curvas de transferência e características de saída serão plotadas. Um passo a passo explicando o procedimento, da inclusão dos componentes até a obtenção das curvas, será exposto a seguir.

Passo n° 1: Os transistores, diodos e outros componentes analógicos encontram-se na aba “components”, na subseção “nonlinear components”, como mostra a figura 3.1. Para a simulação que se segue, foi escolhido um transistor n-MOS com três terminais de saída (porta, dreno e fonte).

Passo n° 2: Agora, adiciona-se ao esquemático as tensões V_{GS} e V_{DS} e a ponta de prova responsável por medir I_{DS} . É importante que o sentido de I_{DS} esteja correto. Para adicionar as tensões, faz-se uso de fontes DC para representar as tensões do transistor. Essas fontes encontram-se em “sources”. A ponta de prova encontra-se na subseção “probes”. Recomenda-se que as fontes

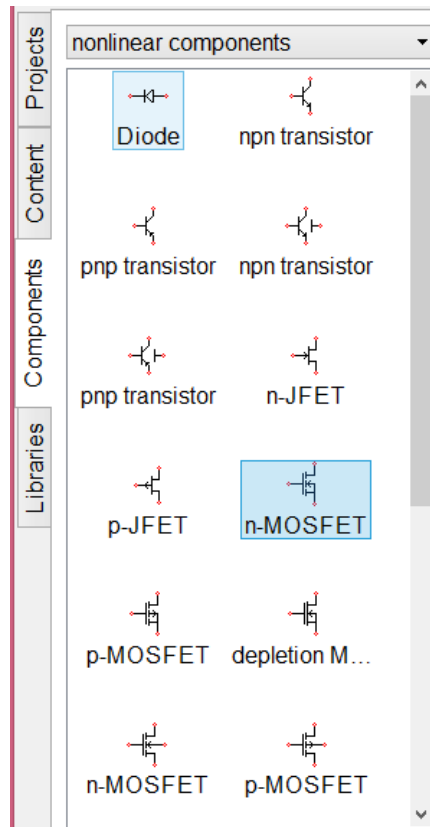


Figura 3.1: Captura de tela mostrando a aba que contém os componentes não lineares

e pontas de prova sejam renomeadas para facilitar a identificação ao se plotar os gráficos. É necessário, ainda, que seja adicionado uma referência para o circuito no esquemático, como mostra a figura 3.2.

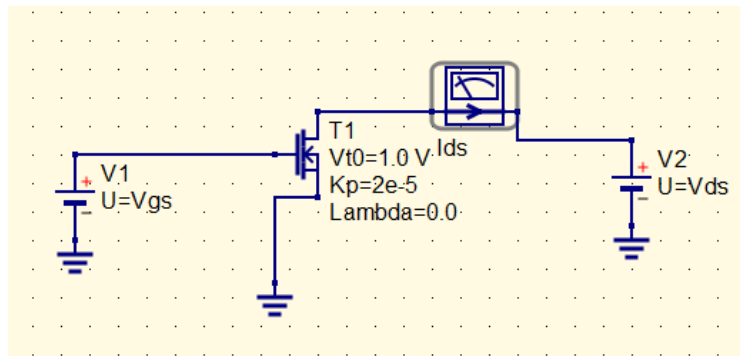


Figura 3.2: Captura de tela mostrando o circuito já contendo as fontes e a ponta de prova no Qucs

Passo nº 3: Nesse passo, são adicionados o bloco que define o tipo de simulação e os que definem a varredura dos parâmetros. Primeiro, clica-se em “simulations”, ainda dentro de “components”. Escolhe-se o componente “DC simulation” e esse é adicionado ao esquemático. Depois, dois blocos de “Parameter sweep” são adicionados, pois serão analisadas tanto a tensão de porta quanto a de dreno. O bloco da simulação pode continuar com o nome original “DC 1”, porém os blocos de varredura devem ser alterados. Se o intuito é plotar uma curva I_{DS} vs. V_{DS} , o bloco “SW1”

deve ser o responsável por varrer o parâmetro V_{DS} e o “SW2” o responsável por varrer V_{GS} . Caso o interesse seja em plotar I_{DS} vs. V_{GS} , as varreduras devem ser trocadas, ou seja “SW1” fica responsável por V_{GS} e o “SW2” o responsável por varrer V_{DS} . Clicando no bloco “SW1”, em “Simulation” deve-se escolher “DC1” e em “Type” escolhe-se linear. No bloco “SW2”, escolhe-se em “Simulation” a opção “SW1”. A figura 3.3 mostra o circuito com os blocos de simulação já adicionados.

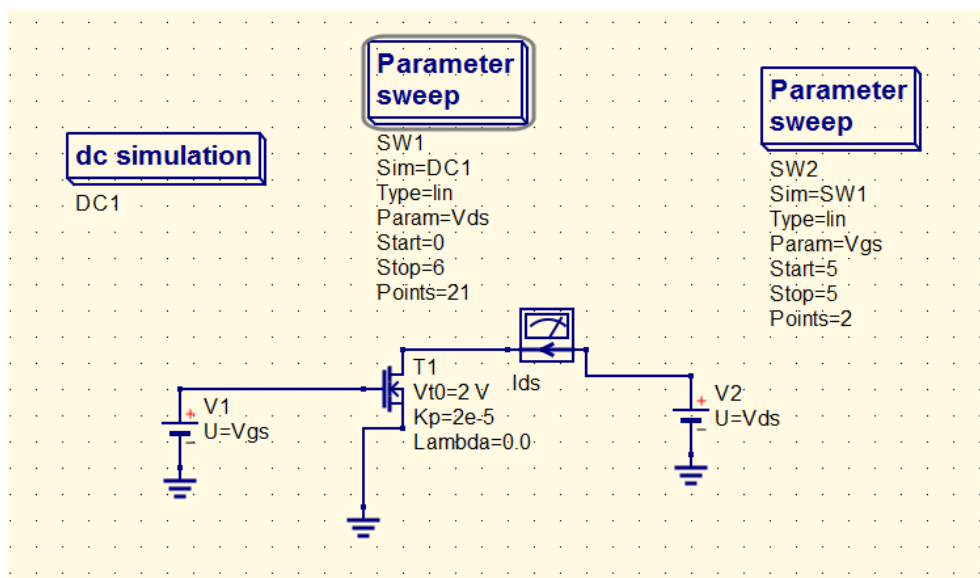


Figura 3.3: Captura de tela mostrando o circuito com os blocos de simulação e varredura devidamente configurados

Passo nº 4: Com o circuito devidamente montado e configurado, basta rodar a simulação. Isso pode ser feito utilizando como atalho a tecla $F2$ do teclado. Uma nova aba se abrirá, com um arquivo de extensão “.dpl” de mesmo nome do esquemático. Essa aba é onde serão exibidos os resultados da simulação. Para adicionar um eixo cartesiano, clica-se em “Cartesian” e insere-se o gráfico no espaço em branco da aba “.dpl”. O Qucs então, pedirá que seja escolhida uma variável dependente a ser plotada. No caso, a variável é I_{DS} , como mostra a figura 3.4. A curva característica de saída para o n-MOS analisado é mostrada na figura 3.5, para um $V_{GS} = 5V$ e V_{DS} variando de $0V$ a $6V$. A curva de transferência é mostrada na figura 3.6, para um $V_{DS} = 2V$ e V_{GS} variando de $0V$ a $6V$. É importante frisar que as duas varreduras só serão necessárias caso se precise plotar várias curvas em um mesmo gráfico.

Passo nº 5: O Qucs dá ao usuário a possibilidade de exportar os dados simulados em um arquivo “.csv”. Esse tipo de arquivo é aceito em softwares populares, como Microsoft Excel e MATLAB. Para realizar a exportação dos dados, o usuário deve clicar sobre a curva que se deseja exportar. Com a curva destacada, clique em “Project” e em sequência “Export to CSV” para salvar o arquivo no diretório desejado.

Além da exportação dos dados obtidos, é possível importar dados de arquivos “.csv” para o ambiente Qucs. Para fazer isso, clique em “Project” e em seguida “Import/Export Data”. Após carregar o arquivo contendo os dados, pode-se adicionar a curva importada ao mesmo *plot* da

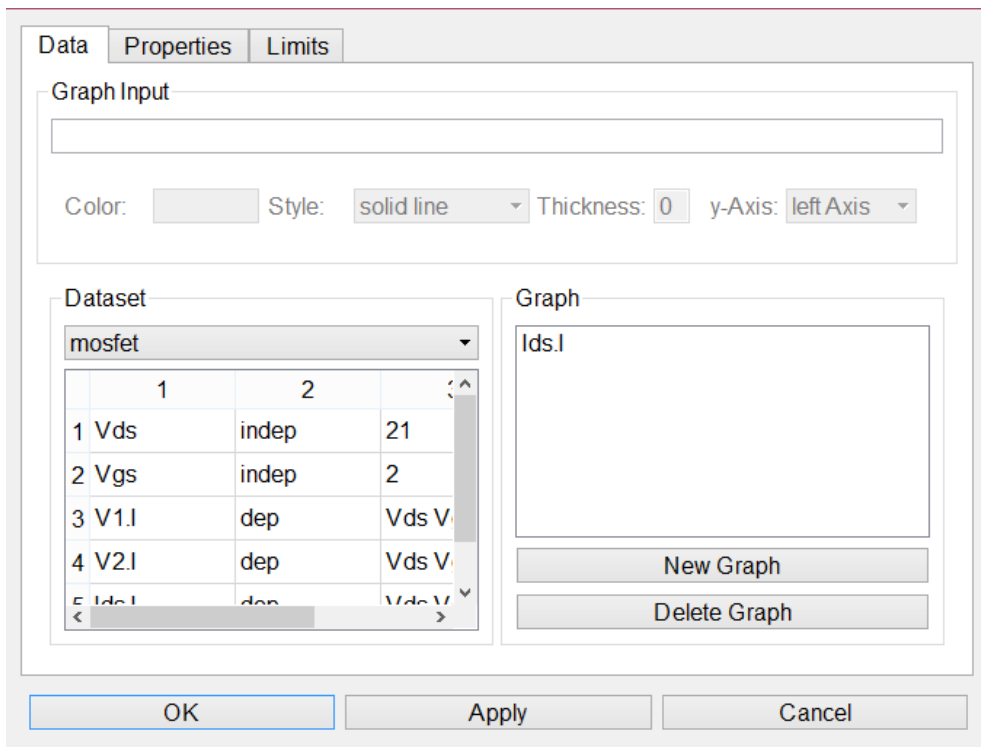


Figura 3.4: Captura de tela mostrando a tela de escolha da variável a ser plotada.

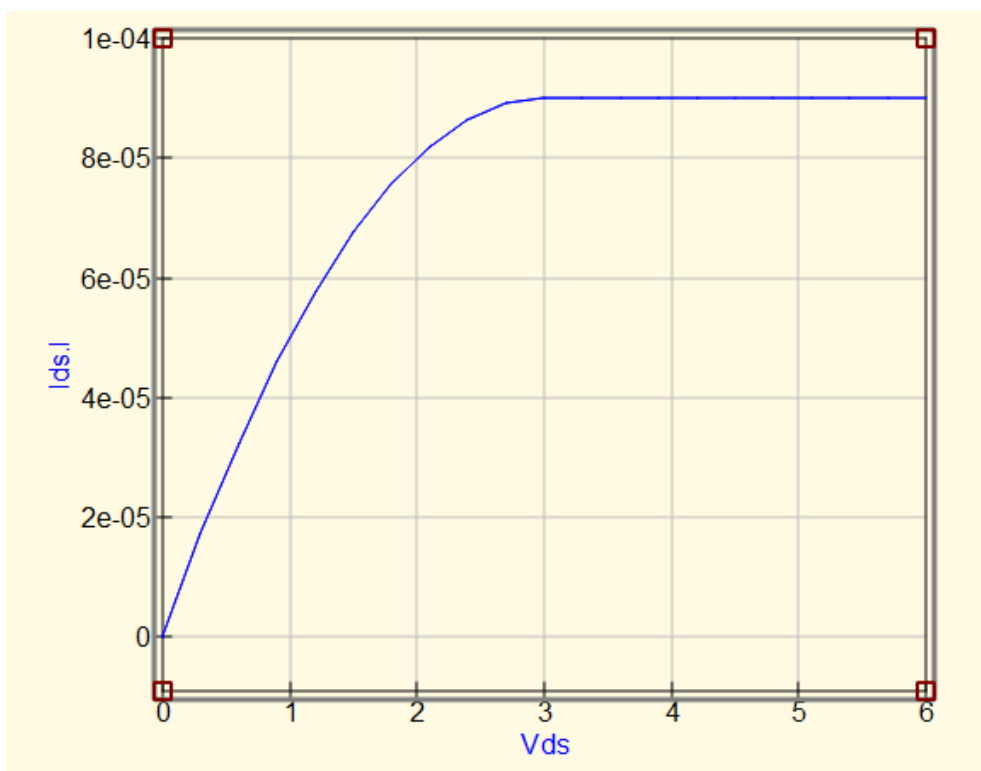


Figura 3.5: Característica de saída do n-MOS simulado pelo Qucs

curva simulada no Qucs, como mostra a figura 3.7, na qual a curva importada é a vermelha e a curva gerada pelo Qucs é a azul. Esse exemplo é a curva de característica de saída de um OTFT

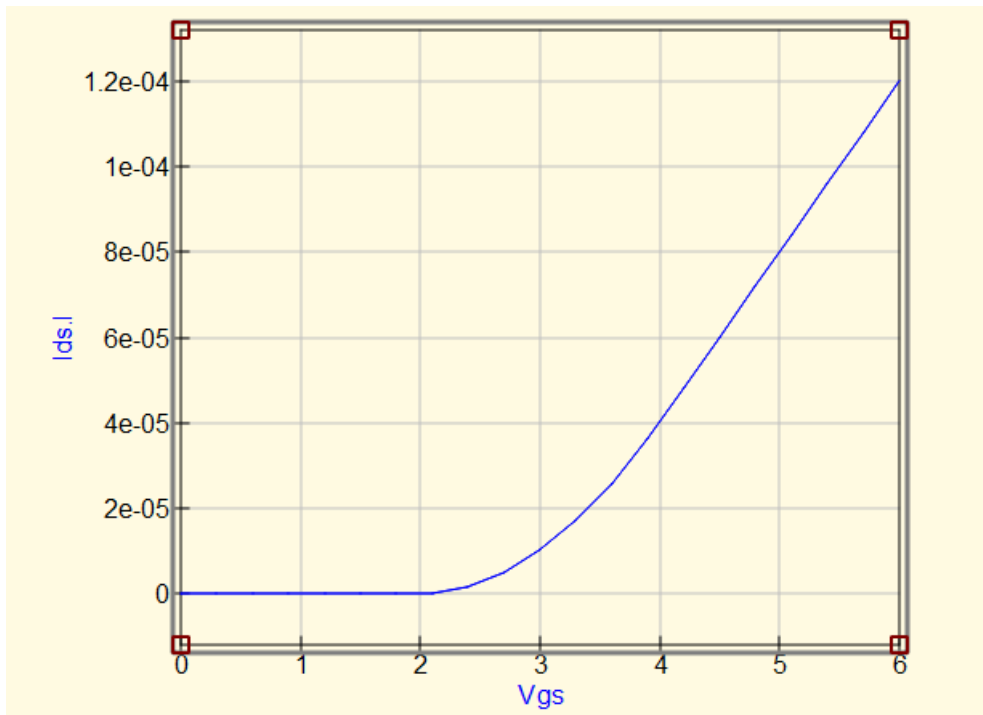


Figura 3.6: Curva de transferência do transistor n-MOS simulado pelo Qucs

do tipo p, por isso as correntes e as tensões são negativas.

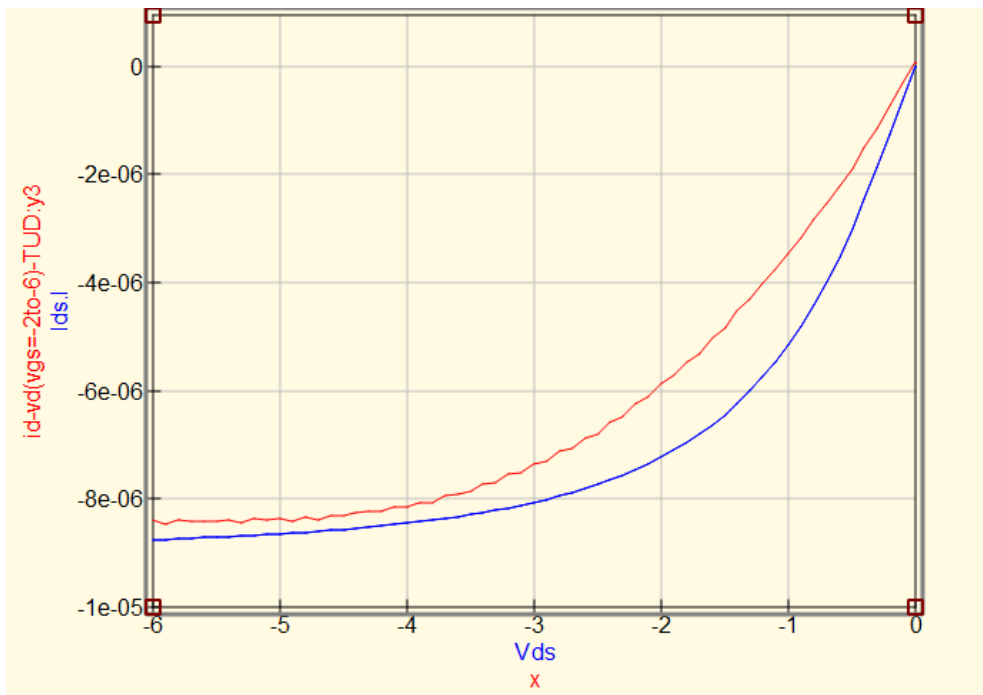


Figura 3.7: Curva simulada em azul e curva importada em vermelho

3.4 Implementação do UMEM em Verilog-A

O código construído em Verilog-A para a implementação do UMEM encontra-se no Anexo I e foi apresentado em [21]. Breves comentários serão feitos nessa seção a respeito desse código.

```
'include "constants.vams"
'include "discipline.vams"
'define $P_K      (1.3806226e-23)$ //Boltzmann constant in eV/K
'define $P_Q      (1.6021918e-19)$ //Charge Cb
'define $M_PI     (3.14159265358979323846)$
'define $P_EO    (8.854214871e-12)$ //VACCUUM PERMITTIVITY IN [F/m]"
```

O trecho “*include constants.vams*” é escrito para que seja incluído o arquivo “constants.vams”, que contém diversas constantes pré-definidas. Algumas delas são utilizadas nesse código, como a constante de Boltzmann, a carga fundamental do elétron, o número π e a permissividade do vácuo. O trecho “*include discipline.vams*” é um arquivo que contém uma coleção de disciplinas (relações entre sinais físicos) da linguagem Verilog-A. É importante ressaltar que a extensão desses arquivos no código original era “.h”, porém o Qucs não reconhece essa extensão, por isso a utilização da extensão “.vams”. Isso se deve ao fato de que a linguagem Verilog A não possui o conceito de cabeçalho (*header*).

Outro trecho notável do código é o seguinte:

```
parameter real $TYPE=1$ from$[-1:1]$ exclude 0;    //$1.0 N-TYPE$, $-1.0 P-TYPE$
parameter real $L = 10e-6$ from$[0.0:inf]$$; //CHANNEL LENGTH IN [m]
parameter real $W = 110e-6$ from$[0.0:inf]$$; //CHANNEL WIDTH IN [m]
```

Nesse trecho, determina-se o tipo do transistor (N ou P) e sua geometria, através dos parâmetros W (largura do canal) e L (comprimento do canal). Vale notar que os valores utilizados para os parâmetros são arbitrários e foram alterados no Qucs durante as simulações, evitando assim que o código fosse alterado entre cada uma das simulações realizadas. Os parâmetros que se seguem também apresentam valores arbitrários:

```
parameter real ALPHASAT=0.572;          SATURATION PARAMETER IN [eV]
parameter real EPS=6.5;  RELATIVE DIELECTRIC CONSTANT OF THE SEMICONDUCTOR
parameter real EPSI=2.6; RELATIVE DIELECTRIC CONSTANT OF THE GATE DIELECTRIC
parameter real GAMMA=1.08;  MOBILITY DEPENDENCE ON (VGS-VT)
parameter real LAMBDA=-1.3e-3; OUTPUT CONDUCTANCE IN [1/V]
parameter real M=1.475; INFLECTION PARAMETER OF OUTPUT CHARACTERISTIC (KNEE)
parameter real RD = 1.474e4; SERIES RESISTANCE AT DRAIN IN [Ohms]
parameter real RS = 1.474e4; SERIES RESITANCE AT SOURCE IN [Ohms]
parameter real TNOM=27; MEASUREMENT TEMPERATURE IN [C]
parameter real TOX=3.7e-7; DIELECTRIC LAYER THICKNESS IN [m]
parameter real VO=0.04; CHARACTERISTIC VOLTAGE OF DEEP STATES IN [V]
parameter real VAA=1.634e3; LOW FIELD MOBILITY PARAMETER
parameter real VT0=2.711; THRESHOLD VOLTAGE IN [V]
parameter real SL=3; SUBTRHESHOLD SLOPE OF LINEAR TRANSFER CURVE
parameter real DVL=2; DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE IN [V]
parameter real QL=0.4; TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE
parameter real IOL=1.006e-10; IOFF FOR LINEAR TRANSFER CURVE IN [A]
parameter real SS=3; SUBTRHESHOLD SLOPE OF TRANSFER CURVE IN SATURATION
```

```

parameter real DVS=2; DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE IN [V]
parameter real QS=0.4; TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE
parameter real IOS=5.144e-9; IOFF FOR SATURATION TRANSFER CURVE IN [A]
parameter real ETAD=0.25; RESISTANCE DRAIN VOLTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real ETAG=0.25; RESISTANCE GATE VOLTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real RCO=1e13; ADJUSTING PARAMETER

```

No trecho acima, são definidos os parâmetros do UMEM, incluindo os parâmetros que modelam a presença de contatos não-ôhmicos. O restante do código, que pode ser visto no anexo I contém a implementação das equações típicas do UMEM em Verilog-A, permitindo o cálculo das grandezas I_{DS} , V_{DS} e V_{GS} e a obtenção, por conseguinte, da característica de saída e da curva de transferência do transistor.

Os trechos seguintes contêm a formalização das equações do modelo, apresentadas na seção 2.6:

```

//CALCULATION OF OHMIC AND NON-OHMIC RESISTANCE AT CONTACTS

R = RS + RD;
RC = RO + (RCO * exp(-ETAD * vds) * exp(-ETAG * vgs));
RT = R + RC;

// EXPRESSION TO CALCULATE CURVES AT VG=VT
vgt = (VMIN/2) * (1+((vgs-VT)/VMIN) + sqrt((DELTA*DELTA)+pow(((vgs-VT)/VMIN)-1),2)));

//CALCULATION OF BELOW THRESHOLD LINEAR CHARACTERISTICS

vgtdL = VT + DVL;
vgtedL = vgtdL - VT;
vsatedL = ALPHASAT * vgtedL;
vdsedL = vds / (pow((1 + (pow((vds/vsatedL),M))), (1 / M)));
nsadL = (pow(vgtedL ,GAMMA)) * vgtedL * (MOS25cg / 'P_Q);
gchidL = 'P_Q * nsadL * MUFET * (W / L);
gchdL = gchidL / (1 + (gchidL * RT));
idcdL = gchdL * vdsedL * (1 + (LAMBDA * vds));
idscL = idcdL * exp(((vgs - VT) / SL) * 2.3);

//CALCULATION OF BELOW THRESHOLD CHARACTERISTICS IN SATURATION

vgtdS = VT + DVS;
vgtedS = vgtdS - VT;
vsatedS = ALPHASAT * vgtedS;
vdsedS = vds / (pow((1 + (pow((vds/vsatedS),M))), (1 / M)));
nsadS = (pow(vgtedS ,GAMMA)) * vgtedS * (MOS25cg / 'P_Q);
gchidS = 'P_Q * nsadS * MUFET * (W / L);
gchdS = gchidS / (1 + (gchidS * RT));
idcdS = gchdS * vdsedS * (1 + (LAMBDA * vds));
idscS = idcdS * exp(((vgs-VT) / SS) * 2.3);

//CALCULATION OF ABOVE THRESHOLD CHARACTERISTICS

vsate = ALPHASAT * vgt;
vdse = vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
nsa = (pow(vgt ,GAMMA)) * vgt * (MOS25cg / 'P_Q);
gchi = 'P_Q * nsa * MUFET * (W / L);
gch = gchi / (1 + (gchi * RT));

```



```
idc = (gch * vdse * (1 + (LAMBDA * vds))) + (vds/CC);
```

3.5 Construção de um módulo no Qucs utilizando um código Verilog-A

Esta seção apresentará um breve tutorial no intuito de auxiliar o leitor a construir um módulo no Qucs utilizando um código Verilog-A.

Passo no.1: escreva o código que descreve seu dispositivo em um novo *script* do Qucs. Salve o código com a extensão “.va”, com o mesmo nome do módulo descrito no código. Por exemplo, no código utilizado nesse projeto, pode-se observar no anexo I que o nome do módulo é “otftmi”. Logo, o código deve ser salvo como “otftmi.va”.

Passo no.2: Com a aba do código aberta, vá em “Project” e em seguida clique em “Build Verilog-A Module”. Isso faz com o que ADMS (que deve ser instalado junto com o Qucs) compile o código e o converta para C++. Cheque se nenhuma mensagem de erro aparece na janela inferior.

Passo no.3: Ainda com a aba do código aberta, vá em “File” e na sequência clique em “Edit Text Symbol”. Nesse momento, uma nova aba se abrirá, com o mesmo nome do arquivo do código porém com a extensão “.sym”, como mostra a figura 3.8. Clique em “Save” para salvar seu símbolo. Uma janela sem mensagens de erros deve aparecer, como na figura 3.9.

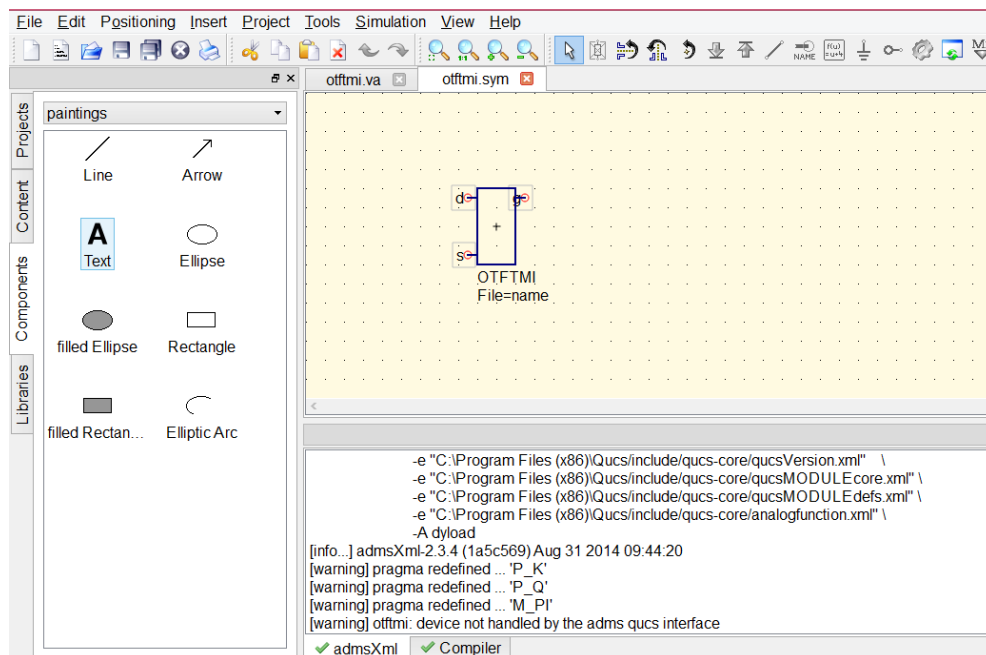


Figura 3.8: Captura de tela mostrando a aba que contém o símbolo do módulo no Qucs

Passo no.4: Para utilizar o seu componente em um esquemático, vá novamente em “Project” e em seguida “Load Verilog-A Module”. Um pop-up avisando sobre a ausência de um ícone aparecerá. A criação de um ícone para o componente é opcional e não será abordada aqui. Uma

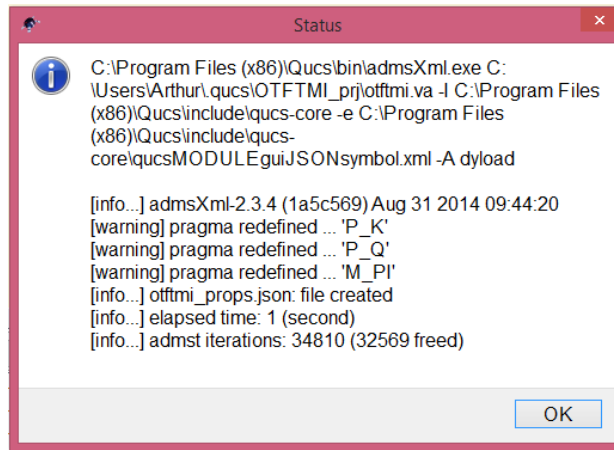


Figura 3.9: Captura de tela mostrando a janela que deve aparecer após o símbolo ser salvo

janela de seleção contendo o seu módulo deve aparecer, como mostra a figura 3.10. Após selecionar o seu módulo, novamente o pop-up alertando sobre a ausência de um ícone aparecerá, ignore-o. Seu módulo agora estará selecionável na seção “Components”, subseção “Verilog-a user devices”, como mostra a figura 3.11.

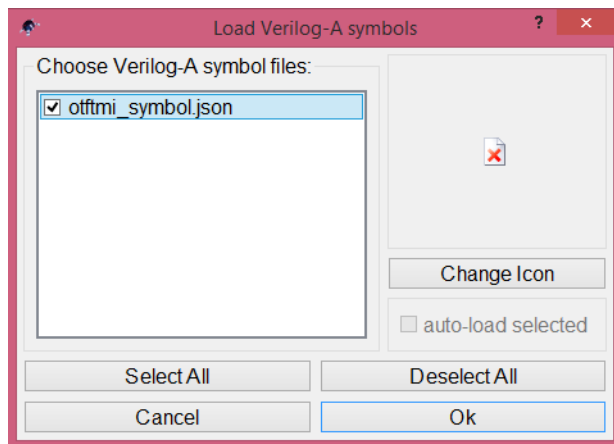


Figura 3.10: Captura de tela mostrando a janela de seleção do módulo a ser carregado

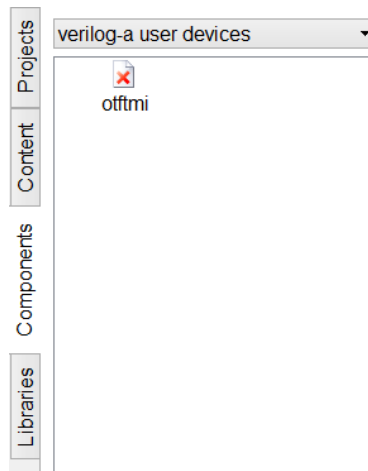


Figura 3.11: Captura de tela mostrando a seção na qual o componente criado está disponível para uso em esquemáticos

Capítulo 4

Resultados e Análises

4.1 Introdução

Neste capítulo serão apresentados os resultados do projeto. Será feita uma verificação do funcionamento do código do UMEM em Verilog-A através de uma comparação entre os resultados obtidos utilizando a simulação/extração via Mathcad com as simulações realizadas no Qucs. Compara-se os resultados do Qucs com os resultados via *i*-MOS. Em seguida, tomando como referência o transistor T1 apresentado em [1], reforça-se a verificação da infra-estrutura Qucs + UMEM. A seção seguinte apresenta a extração dos parâmetros de um transistor “criado” via *i*-MOS através do código em Mathcad que implementa o modelo UMEM e a inserção dos parâmetros extraídos no Qucs. Os resultados dessa inserção são mostrados e comparados. Por último, a combinação Qucs e UMEM é utilizada para realizar o ajuste do modelo com um transistor real ainda não publicado, além dos parâmetros extraídos para essa tecnologia através do código Mathcad e a comparação dos resultados. Basicamente, foram utilizados dois grupos de dados experimentais: os originais do código em Mathcad, fornecido pelo grupo de pesquisa desenvolvedor do UMEM; e outro fornecido pela Technische Universität Dresden.

4.2 Verificação do uso do Qucs com o UMEM

O foco dessa seção é comparar os resultados de uma infra-estrutura já implementada (Mathcad + UMEM) com os resultados da nova infra-estrutura proposta (Qucs + UMEM). O OTFT modelado no Mathcad é do tipo p e possui $W = 210 \mu\text{m}$, $L = 30 \mu\text{m}$, $TOX = 390 \text{ nm}$, constante dielétrica do isolante igual a 4 e constante dielétrica do semicondutor igual a 6.5. São dados de entrada para o procedimento de extração os valores medidos experimentalmente de I_{DS} vs. V_{DS} para as diferentes tensões de porta medidas. Esse experimento foi realizado pelo grupo de pesquisa que forneceu o código em Mathcad para a realização deste trabalho. No caso, essas tensões são $V_{GS} = 0V$, $V_{GS} = -10V$, $V_{GS} = -15V$, $V_{GS} = -20V$, $V_{GS} = -25V$ e $V_{GS} = -30V$.

Ao final da execução do código em Mathcad, obtém-se uma comparação entre as curvas experimentais (pontilhadas) e simuladas (linhas) no próprio ambiente do Mathcad. Essas curvas são

mostradas na figura 4.1. As curvas para a tensão $V_{GS} = -30V$ são as verdes. Como era esperado, já que esse código já havia sido testado antes pelos desenvolvedores do UMEM, as curvas se ajustam satisfatoriamente. Os parâmetros extraídos estão mostrados na tabela 4.1.

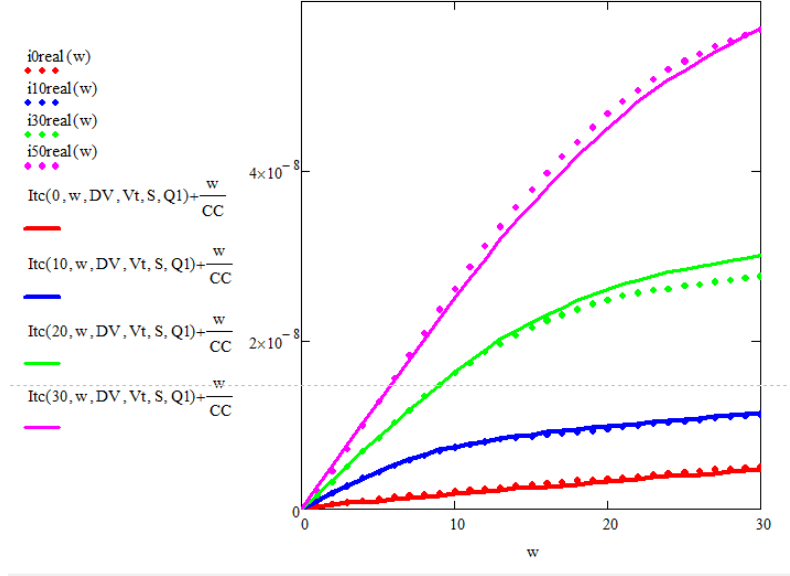


Figura 4.1: Curvas obtidas ao final do processo de extração no Mathcad. Os parâmetros obtidos ao final dessa extração são mostrados na Tabela 4.1

Em seguida, esses parâmetros foram inseridos no Qucs para checar se o código Verilog-A e o código Mathcad produziam o mesmo resultado. Para simplificar a comparação, apenas a tensão $V_{GS} = -30V$ foi escolhida. O resultado foi negativo, como pode ser visto nas figuras 4.2 e 4.3. Ao fazer a comparação entre as figuras 4.2 e 4.3, nota-se que na primeira a corrente para $V_{DS} = -30V$ vale $I_{DS} = -3.5 \times 10^{-8}A$. Para a mesma tensão de dreno, a segunda figura apresenta um valor de $I_{DS} = -3.5 \times 10^{-6}A$, mostrando assim uma considerável discrepância entre os resultados obtidos via MathCad e a infra-estrutura (*Qucs + UMEM*).

Tabela 4.1: Parâmetros do modelo UMEM extraídos via simulação/processo de extração no Mathcad. Esses dados foram utilizados como base para as figuras 4.1, 4.2, 4.3, 4.4 e 4.5

| Parâmetro | Valor |
|---------------------------|-------------------------|
| V_{GS} [V] | -30 |
| V_T [V] | 1.002 |
| γ_a | 0.117 |
| V_{aa} [V] | 3.073×10^{26} |
| \mathbf{R} [Ω] | 3.144×10^6 |
| α_s | 0.847 |
| m | 3.105 |
| λ [1/V] | -6.388×10^{-4} |

Para verificar se o problema apresentado era advindo do Qucs, por exemplo uma má leitura do código feita pelo simulador, deu-se entrada dos mesmos parâmetros no *i*-MOS, que utiliza o

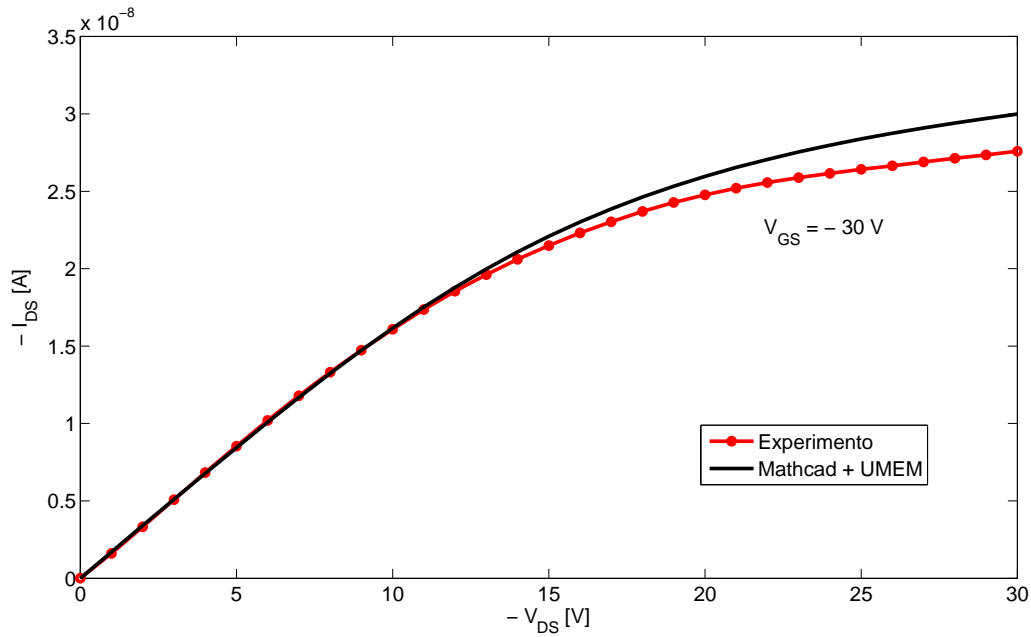


Figura 4.2: Curva experimental e curva obtida via simulação no Mathcad. Os parâmetros obtidos ao final dessa extração são mostrados na Tabela 4.1

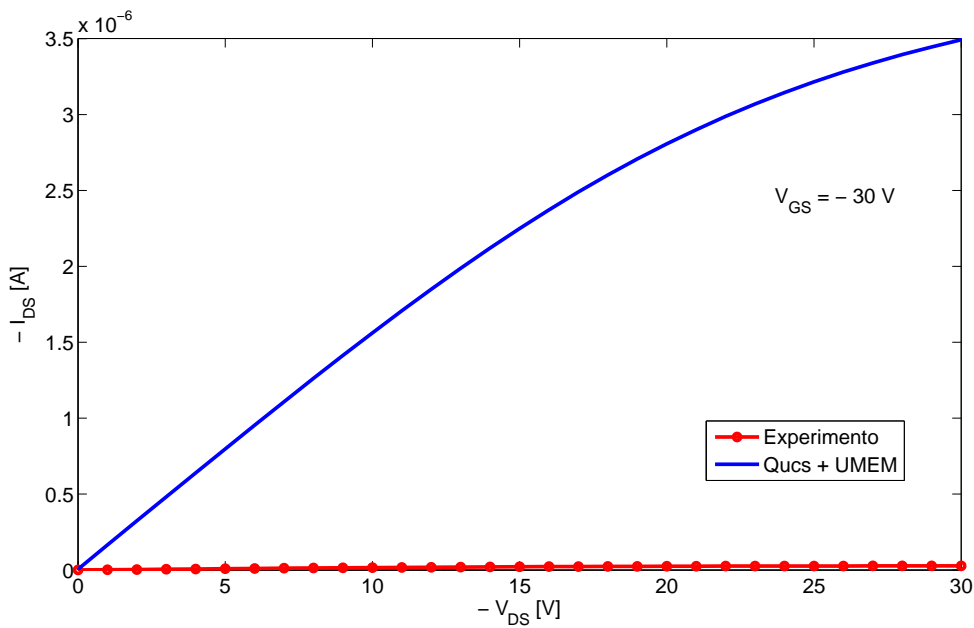


Figura 4.3: Curva experimental e curva obtida via simulação no Qucs. Os parâmetros da Tabela 4.1 foram inseridos no Qucs para que a curva I_{DS} vs V_{DS} fosse gerada.

mesmo código em Verilog-A que o Qucs. Dessa maneira, pode-se checar se o problema estava no Qucs ou numa diferença entre os códigos Verilog-A e Mathcad. A figura 4.4 mostra que o resultado obtido com o *i*-MOS foi exatamente o mesmo do Qucs, ou seja, a discrepância de resultados deve-se realmente a uma diferença entre o código Verilog-A e o código Mathcad. A figura 4.5 mostra

a comparação entre as 3 curvas obtidas via simulações e a curva experimental.

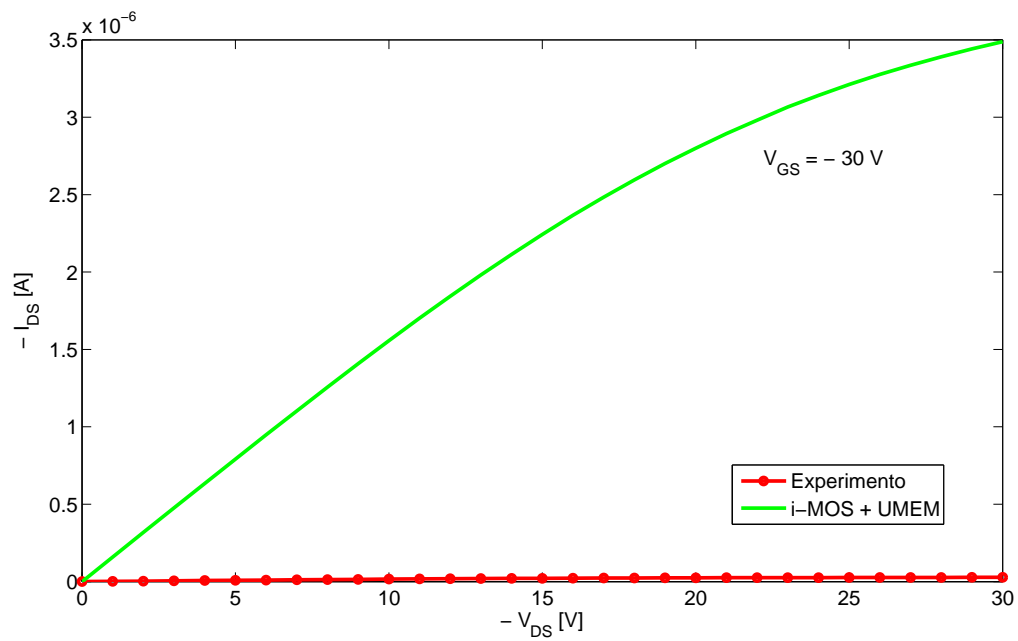


Figura 4.4: Curva experimental e curva obtida via simulação no *i*-MOS. Os parâmetros da Tabela 4.1 foram inseridos no *i*-MOS para que a curva I_{DS} vs V_{DS} fosse gerada.

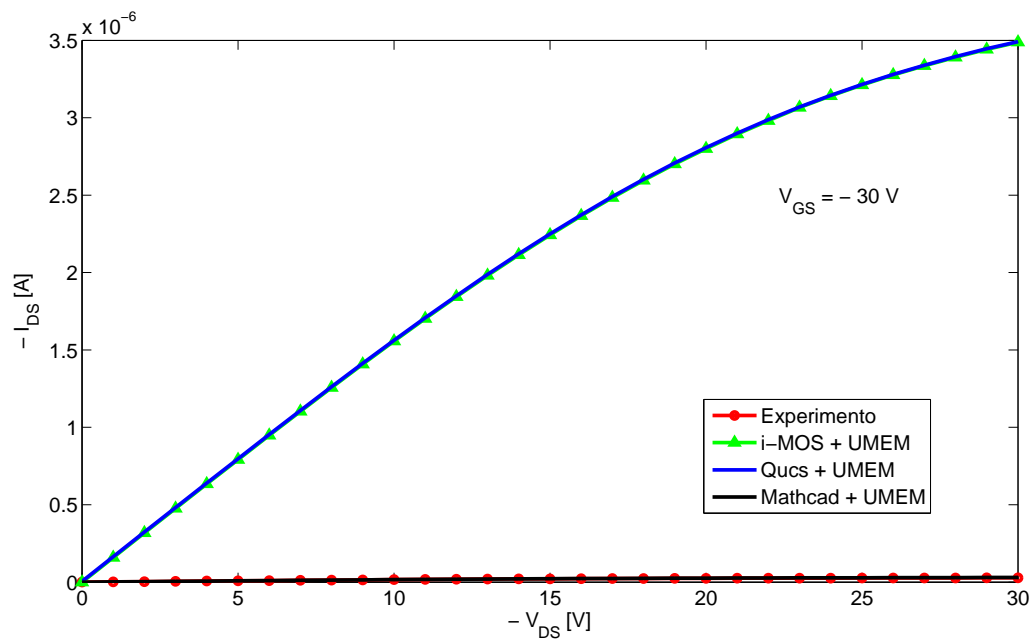


Figura 4.5: Curva experimental e curvas obtida via simulação no Mathcad, Qucs e *i*-MOS. Para gerar essa figura, foram usados os parâmetros mostrados na Tabela 4.1

4.2.1 Comparação entre os dados fornecidos pela literatura e uma simulação via Qucs

O transistor utilizado para verificar o funcionamento do UMEM com o Qucs foi o T1, apresentado em [1]. Esse transistor é um OTFT com porta na parte inferior, canal feito de pentaceno e substrato de vidro. O dielétrico da porta é feito de polimetilmetacrilato (PMMA). Esse transistor foi apresentado originalmente em [22]. A espessura do dielétrico é de 700 nm e a constante dielétrica desse material é 3.9. Esse dispositivo possui largura $W = 600 \mu\text{m}$ e comprimento $L = 120 \mu\text{m}$. Em [1] foi realizada a extração dos parâmetros desse OTFT e o resultado obtido pelos autores encontra-se na tabela 4.2 a seguir:

Tabela 4.2: Parâmetros do transistor T1 [1], utilizados nas figuras 4.6, 4.7, 4.8 e 4.9

| Parâmetro | Valor |
|---------------------------|-----------------------|
| V_{GS} [V] | -40 |
| V_T [V] | -4.1 |
| γ_a | 1.9 |
| V_{aa} [V] | 1.7×10^3 |
| \mathbf{R} [Ω] | 1.1×10^7 |
| α_s | 0.39 |
| m | 1.27 |
| λ [1/V] | -3.5×10^{-3} |
| \mathbf{n} | 0 |
| I_0 [A] | 0 |
| ξ | 1 |

Ao dar entrada desses valores no componente criado via Verilog-A no Qucs, obteve-se a curva em azul mostrada na figura 4.6. A curva em vermelho representa a curva característica de saída do transistor T1 segundo o artigo original [1] (apenas para $V_{GS} = -40V$). Como os dados dessa curva não estão disponíveis de forma digital, a digitalização dos valores originais de I_{DS} foi feita de forma não-automatizada, utilizando-se uma régua.

Percebe-se que o valor da corrente obtida com o Qucs é muito maior que o valor apresentado em [1]. Enquanto a corrente original é da ordem de 10^{-9} , a obtida é da ordem de 10^{-7} .

A figura 4.7 mostra a curva obtida com um novo conjunto valores dos parâmetros essenciais do modelo. As curvas de característica de saída experimental e modelada apresentaram valores muito próximos para a região linear da curva. Esse ajuste foi realizado para ilustrar a possibilidade de ajustar apenas regiões específicas da curva, com o intuito de testar o que cada parâmetro modifica na curva plotada.

O primeiro parâmetro a ser modificado foi α_s , responsável por alterar a tensão a partir da qual ocorre saturação, como mostrado na equação 2.5. Quanto maior o valor de α_s , maior será o valor da corrente I_{DS} na região de saturação. Esse parâmetro é de extrema utilidade caso já se tenha um valor próxima da curva original na região linear.

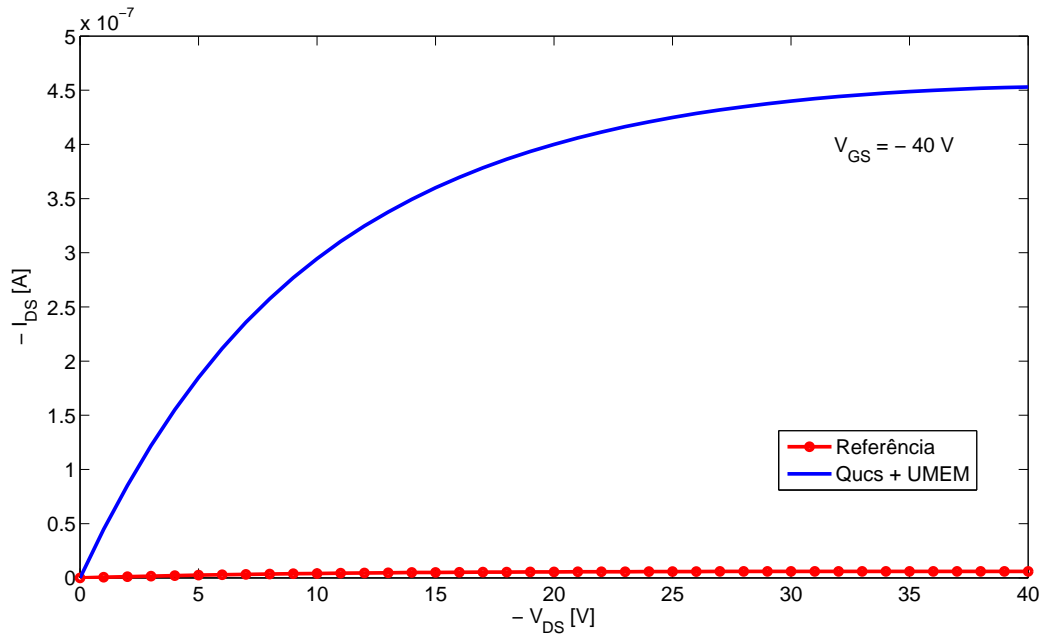


Figura 4.6: Curva característica de saída do transistor T1 obtida com Qucs utilizando os parâmetros fornecidos em [1] e a curva original apresentada em [1]. Esses parâmetros são mostrados na Tabela 4.2

Outro parâmetro que foi alterado foi γ_a . Esse parâmetro, como mostra a equação 2.3, está relacionado a mobilidade do canal do OTFT. A mobilidade μ_{FET} está relacionada com a corrente segundo a equação 2.4. Como a mobilidade também depende de outros fatores, inclusive de V_{aa} , não se pode afirmar se ela aumentará ou diminuirá com o incremento de γ_a . Para o caso do transistor analisado, μ_{FET} diminuirá se γ_a aumentar, mantendo-se os valores originais de V_{aa} e V_T . No caso analisado, tanto V_{aa} quanto V_T foram alterados.

O parâmetro λ foi mantido o mesmo da tabela 4.2 para esse ajuste. Esse parâmetro está relacionado ao formato da curva, bem como o parâmetro m . Porém, apenas o último foi alterado por ser mais simples de ser manipulado. Alguns valores de λ fazem com que a curva fuja de seu comportamento normal, por isso, evitou-se alterar esse parâmetro.

A resistência de contato R é um parâmetro que pode servir para aumentar ou diminuir o valor da corrente em todas as regiões, já que, pela equação 2.4, observa-se que as duas grandezas são inversamente proporcionais. Esse parâmetro também foi alterado para realizar o ajuste da região linear das duas curvas.

Os parâmetros referentes a presença de contatos não-ôhmicos foram mantidos iguais.

Os valores dos parâmetros utilizados para se obter a curva azul da figura 4.7 estão mostrados na tabela 4.3.

Apesar dos valores utilizados na simulação da figura 4.7 produzirem uma curva próxima da original, há discrepâncias entre tais valores e os extraídos em [1]. Por isso, foram realizadas novas simulações focando manter a fidelidade entre valores modelados e valores originais. A figura 4.8

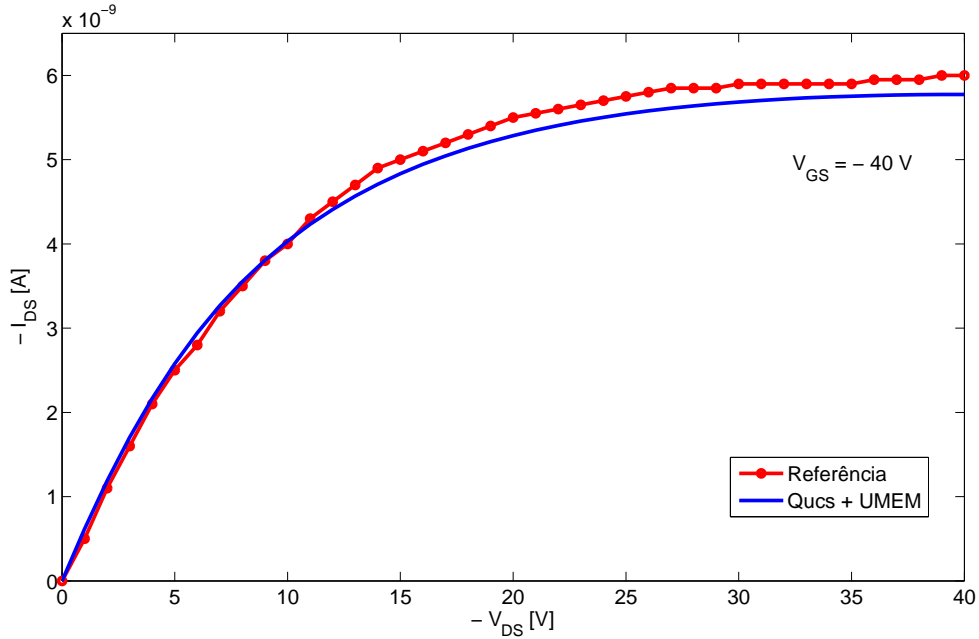


Figura 4.7: Curva característica de saída do transistor T1 obtida com Qucs utilizando parâmetros modificados. Esses parâmetros são mostrados na Tabela 4.3

mostra o ajuste realizado alterando-se apenas um parâmetro: γ_a . O valor original para esse parâmetro era de 1.9 e a curva da figura foi obtida utilizando-se $\gamma_a = 4.2685$. A região de saturação ainda encontra-se ligeiramente diferente da original, porém pode-se dizer que as curvas são praticamente as mesmas.

Um valor tão preciso de γ_a deve-se ao fato de que a curva é bastante sensível a esse parâmetro. Aumentar γ_a de 2 para 4, por exemplo, diminui o valor absoluto da corrente cerca de 10 vezes.

Outra maneira de realizar a sobreposição entre as duas curvas foi alterando-se o parâmetro V_{aa} de 1700 para 8700. Com esse novo valor, obteve-se a curva da figura 4.9, ainda mais próxima da original do que a anterior.

A tabela 4.3 mostra os valores de cada parâmetro para a obtenção das curvas simuladas nas figuras 4.7, 4.8 e 4.9.

Ao final dessas simulações, conclui-se que o código em Verilog-A que implementa o UMEM no Qucs é capaz de reproduzir o comportamento de um OTFT descrito pelo modelo UMEM, desde que alguns parâmetros sejam manipulados manualmente.

É importante ressaltar que o valor de V_T para o ajuste linear, de 26 V, é irreal, porque, além de se tratar de um transistor do tipo n (V_T negativo), uma tensão de 26 V é muito alto. Esse valor foi escolhido para ilustrar que é possível realizar o ajuste mesmo com o uso de valores para alguns parâmetros que fogem da realidade.

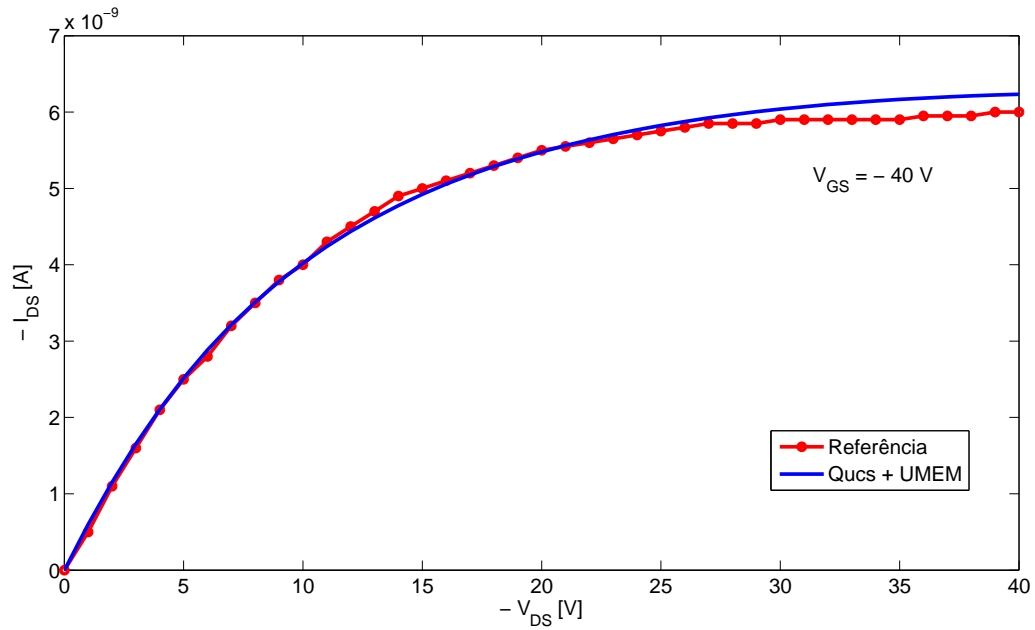


Figura 4.8: Curva característica de saída do transistor T1 obtida com Qucs alterando apenas γ_a em relação aos valores da tabela 4.2. Esses parâmetros são mostrados na Tabela 4.3

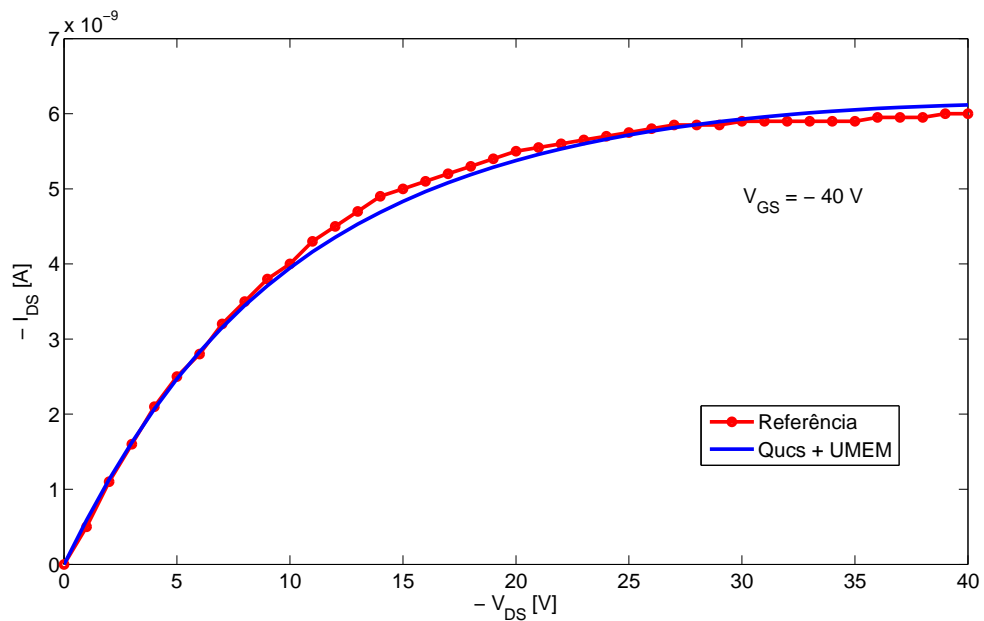


Figura 4.9: Curva característica de saída do transistor T1 obtida com Qucs alterando apenas V_{aa} em relação aos valores da tabela 4.2. Esses parâmetros são mostrados na Tabela 4.3

4.3 Extração dos parâmetros via Mathcad e verificação com Qucs

Nessa seção, escolheu-se um conjunto de valores para os parâmetros UMEM baseados na literatura e deu-se entrada desses valores no *i*-MOS. As dimensões do transistor, do tipo p, são:

Tabela 4.3: Parâmetros do transistor T1 modificados para obtenção da curva mostrada em 4.7, 4.8 e 4.9

| Parâmetro | Original | Figura 4.7 | Figura 4.8 | (Figura 4.9) |
|---------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| V_{GS} [V] | -40 | -40 | -40 | -40 |
| V_T [V] | -4.1 | 26 | -4.1 | -4.1 |
| γ_a | 1.9 | 3 | 4.2685 | 1.9 |
| V_{aa} [V] | 1.7×10^3 | 1.7×10^4 | 1.7×10^3 | 8.7×10^3 |
| \mathbf{R} [Ω] | 1.1×10^7 | 2.5×10^7 | 1.1×10^7 | 1.1×10^7 |
| α_s | 0.39 | 0.18 | 0.39 | 0.39 |
| m | 1.27 | 1.39 | 1.27 | 1.27 |
| λ [1/V] | -3.5×10^{-3} | -3.5×10^{-3} | -3.5×10^{-3} | -3.5×10^{-3} |
| \mathbf{n} | 0 | 0 | 0 | 0 |
| I_0 [A] | 0 | 0 | 0 | 0 |
| ξ | 1 | 1 | 1 | 1 |

$TOX = 320$ nm, $W = 150$ μm $L = 50$ μm e a constante dielétrica desse material é 4. Obteve-se as curvas características de saída para as seguintes tensões de porta: $V_{GS} = 0V$, $V_{GS} = -5V$, $V_{GS} = -10V$, $V_{GS} = -15V$, $V_{GS} = -20V$, $V_{GS} = -25V$ e $V_{GS} = -30V$. Esses dados foram então inseridos no código de extração do Mathcad e os valores dos parâmetros obtidos ao final encontram-se na tabela 4.4. A figura 4.10 mostra que o Mathcad foi capaz de reproduzir com sucesso a curva original (foi escolhido apenas $V_{GS} = -30V$ para fins comparativos). Inserindo os mesmos parâmetros no Qucs, obtém-se um perfeito ajuste entre as curvas, como mostra a figura 4.11.

Tabela 4.4: Parâmetros iniciais do transistor inseridos no *i*-MOS e parâmetros extraídos via Mathcad. Esses dados foram utilizados para gerar as figuras 4.10, 4.11 e 4.12

| Parâmetro | Original | Extraído |
|---------------------------|-------------------------|-------------------------|
| V_{GS} [V] | -30 | -30 |
| V_T [V] | -3.169 | -1.373 |
| γ_a | 0.325 | -0.94 |
| V_{aa} [V] | 6.771 | 546.204 |
| \mathbf{R} [Ω] | 2.488×10^4 | 0 |
| α_s | 0.580 | 0.639 |
| m | 1.733 | 3.116 |
| λ [1/V] | -8.673×10^{-4} | -1.918×10^{-4} |
| I_0 [A] | 5.144×10^{-9} | 5.1×10^{-9} |
| ξ | 100 | - |

Esses novos valores foram então inseridos no Qucs e obteve-se a curva mostrada na figura 4.12, totalmente diferente da curva original. Uma possível razão para uma corrente tão alta é que o valor da resistência \mathbf{R} extraído foi 0, muito distante do valor original. Esse parâmetro possui grande influência no valor da corrente, sendo inversamente proporcional a mesma. Um valor muito

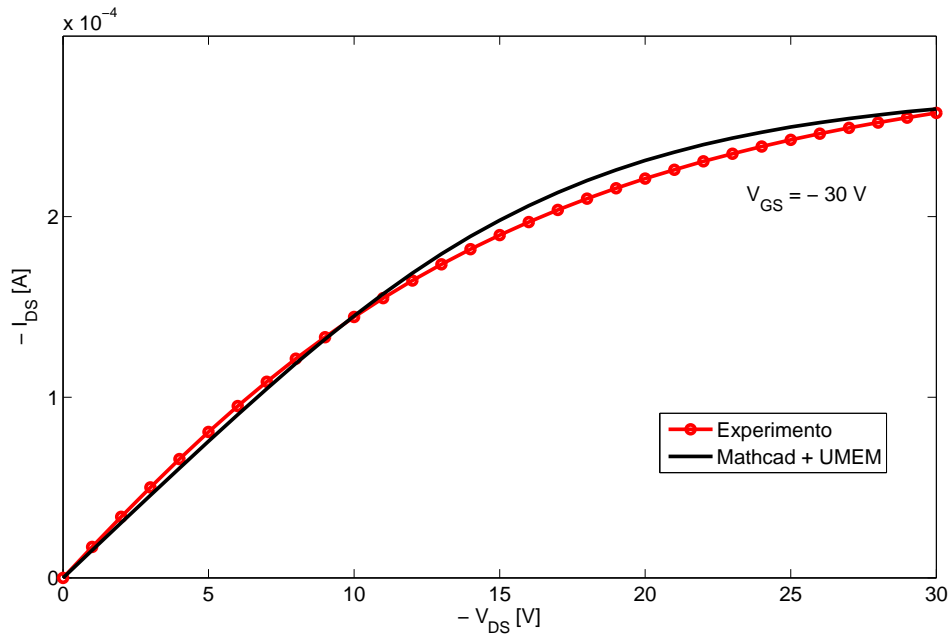


Figura 4.10: Curva característica de saída do transistor criado via *i*-MOS e o respectivo ajuste via Mathcad. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4

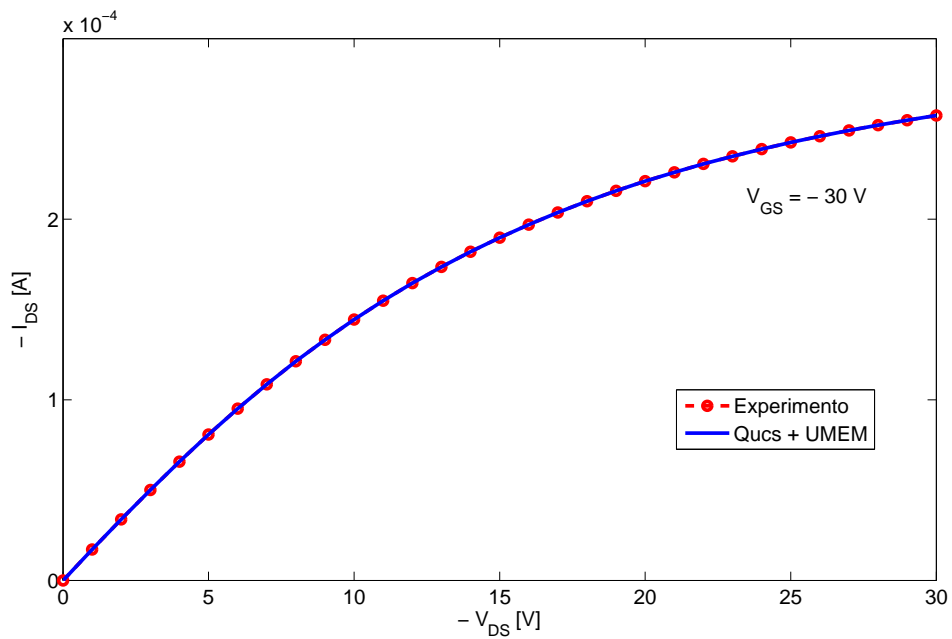


Figura 4.11: Curva característica de saída do transistor criado via *i*-MOS e o respectivo ajuste via Qucs. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4

baixo de resistência gerará um valor alto de corrente e vice-versa.

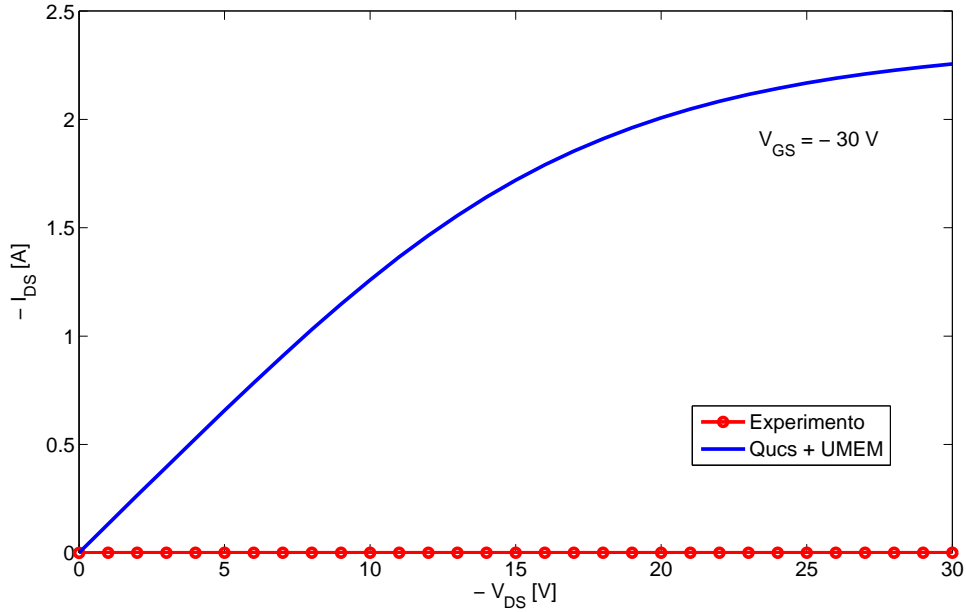


Figura 4.12: Comparação entre a curva característica de saída do transistor criado via *i*-MOS e a curva criada via Qucs com os parâmetros extraídos anteriormente via Mathcad. Os parâmetros usados para plotar essas curvas são mostrados na Tabela 4.4

4.4 Utilização do UMEM para modelar um OTFT

Essa seção inclui a aplicação da infra-estrutura UMEM + QUCS a uma tecnologia ainda não publicada, cujos dados foram fornecidos pela Technische Universität Dresden, na Alemanha. O ajuste foi realizado apenas para I_{DS} vs V_{DS} , sendo $V_{GS} = -6V$ e I_{DS} vs V_{GS} para $V_{DS} = -6V$. As características do transistor são: tipo p, $W = 1000\mu m$, $L = 50\mu m$, $TOX = 30nm$, constante dielétrica do isolante igual a 2.6 e constante dielétrica do semiconductor igual a 9.4.

Para a curva característica de saída, foram realizados três ajustes: um para a região linear, outro para a região de saturação e outro para a curva completa. A tabela 4.5 mostra os valores dos parâmetros obtidos ao final dos ajustes. A estratégia utilizada para realizar o ajuste entre as curvas foi similar ao descrito na seção 4.2.1.

As figuras 4.13, 4.14 e 4.15 mostram os ajustes para a região linear, região de saturação e ajuste completo, respectivamente. Com o intuito de se verificar mais uma vez o funcionamento correto do Qucs, inseriu-se os parâmetros obtidos para o ajuste completo no *i*-MOS. As curvas obtidas via Qucs e via *i*-MOS se sobrepõem, como mostra a figura 4.16

A última parte do trabalho consistiu em realizar o ajuste da curva de transferência do transistor. Utilizou-se o mesmo grupo de parâmetros do ajuste completo da curva de característica de saída. Como pode ser observado na figura 4.17, as curvas experimental e modelada não se sobrepõem, o que já era esperado. Em teoria, o ajuste deveria se realizar, pois nos dois casos, tanto V_{GS} quanto V_{DS} são iguais a - 6 V. Isso não ocorreu pois os dados experimentais apresentam

Tabela 4.5: Parâmetros extraídos manualmente utilizando o Qucs referentes as curvas 4.13, 4.14 e 4.15

| Parâmetro | Figura 4.13 | Figura 4.14 | Figura 4.15 |
|------------------------------|-----------------------|---------------------|---------------------|
| V_{GS} [V] | -6 | -6 | -6 |
| V_T [V] | 2 | -4 | -3.95 |
| γ_a | 5 | 6 | 5 |
| V_{aa} [V] | 10 | 3 | 3 |
| \mathbf{R} [Ω] | 5.3×10^5 | 2.15×10^5 | 3.7×10^5 |
| α_s | 0.85 | 0.7 | 1.165 |
| m | 2.7 | 1.5 | 2.2 |
| λ [$\mathbf{1/V}$] | -1.3×10^{-3} | -3×10^{-3} | -3×10^{-3} |
| ξ | 7 | 7 | 7 |

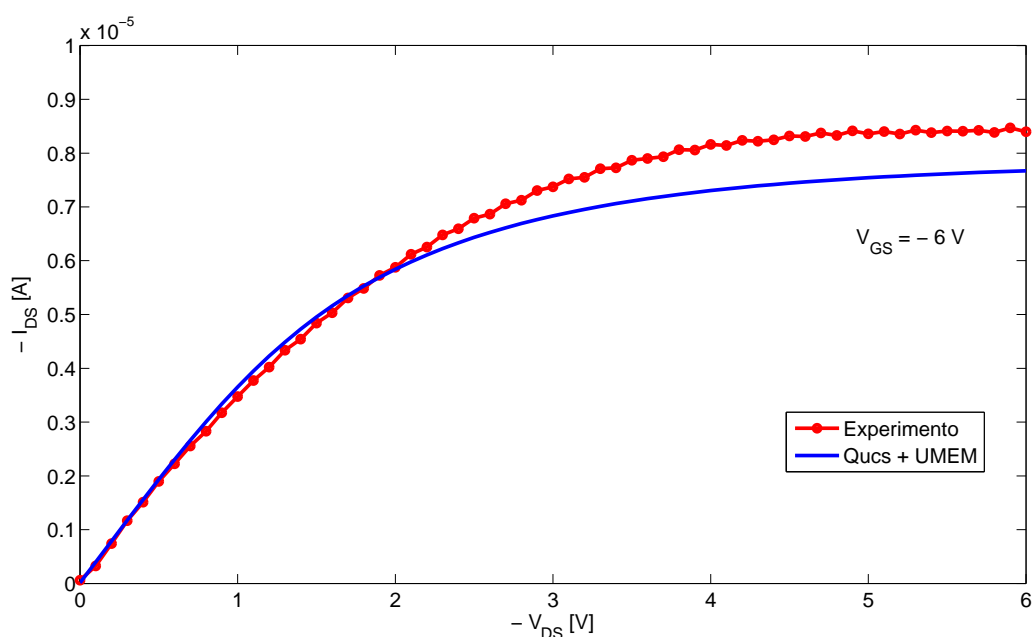


Figura 4.13: Ajuste da região linear da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5

valores diferentes de I_{DS} para $V_{GS} = -6$ V nas medições feitas para a característica de saída e para a curva de transferência. Uma possível explicação para isso são os efeitos da histerese nos OTFTs, ou simplesmente, erros experimentais. Tais discrepâncias são comuns em experimentos com OTFTs. Por essa razão, realizou-se o ajuste com um novo conjunto de valores dos parâmetros do UMEM. O resultado dessa simulação está mostrado na figura 4.18. Esse novo conjunto de valores está mostrado na tabela 4.6.

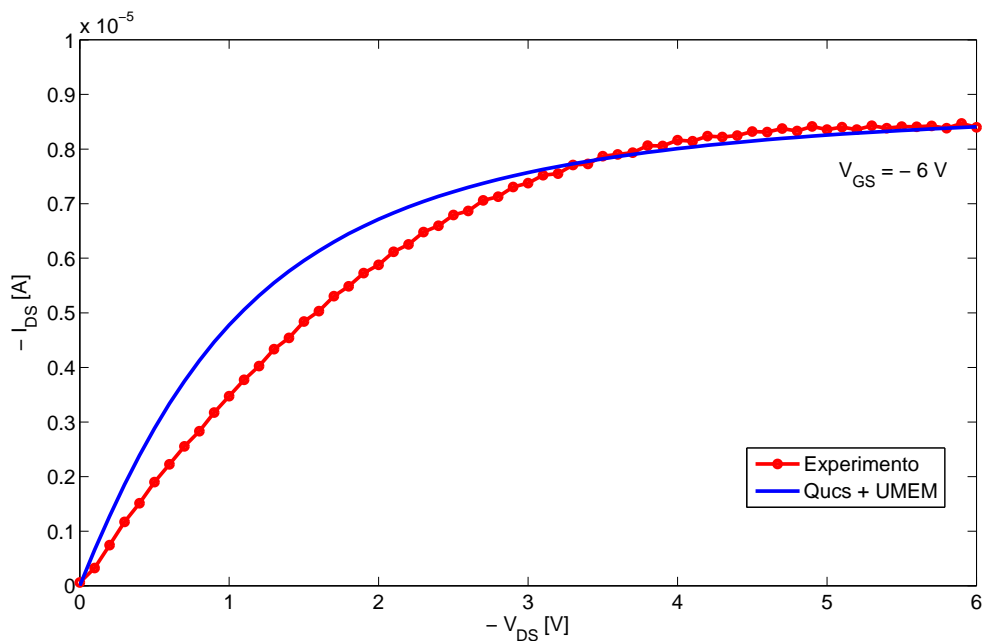


Figura 4.14: Ajuste da região de saturação da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5

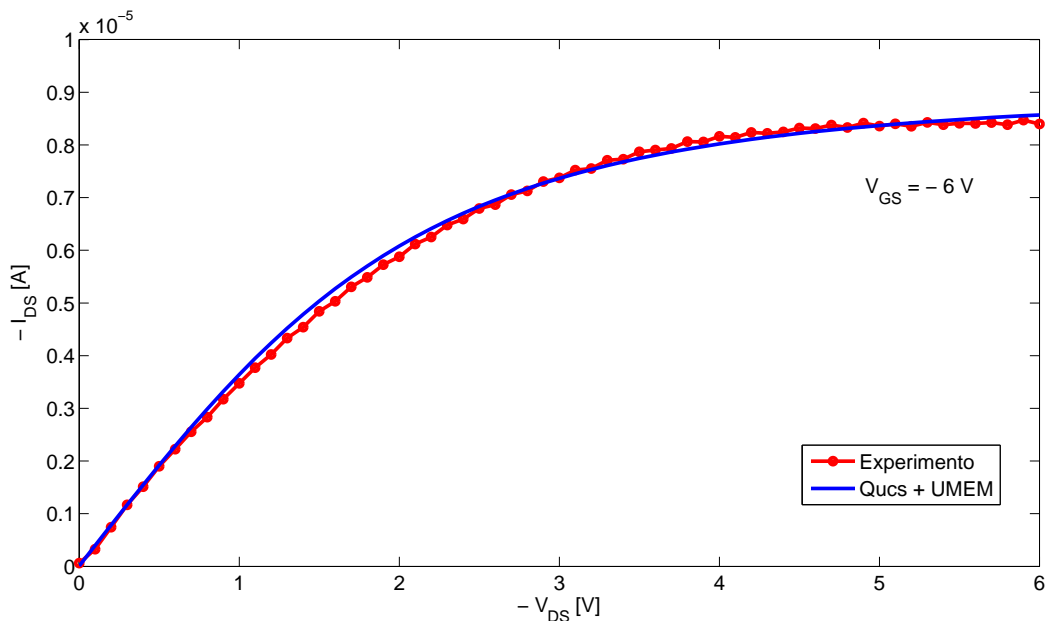


Figura 4.15: Ajuste completo da curva característica de saída do transistor. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5

4.4.0.1 Algoritmo para realizar o ajuste entre curvas do Qucs e curvas experimentais

Um algoritmo foi desenvolvido nesse trabalho, capaz de ajudar o leitor a realizar, utilizando a infra-estrutura Qucs + UMEM, o ajuste entre as curvas simuladas e as curvas experimentais.

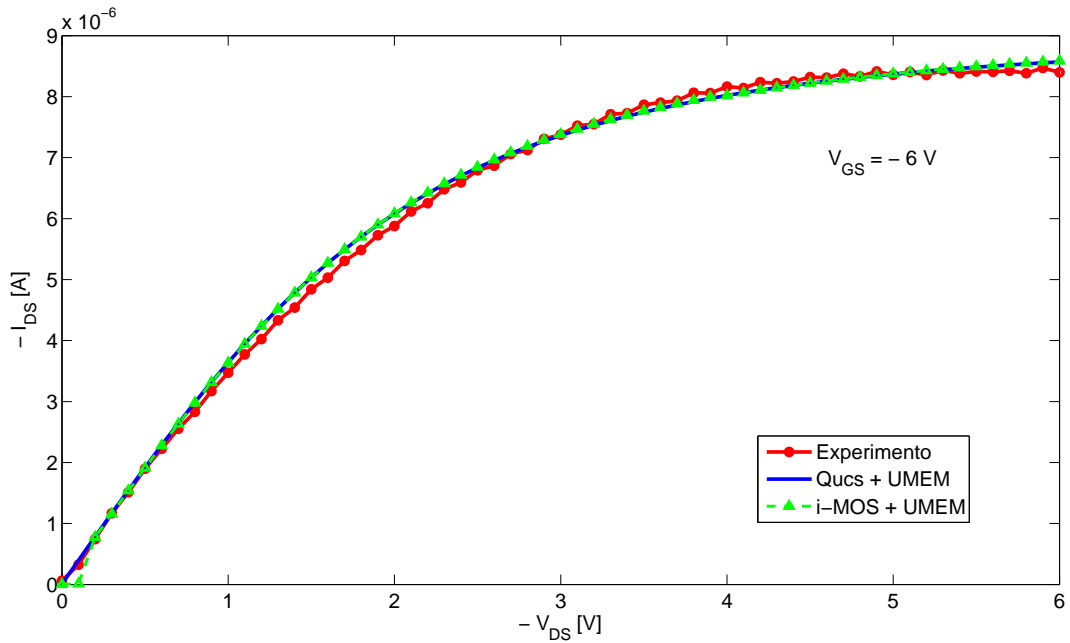


Figura 4.16: Ajuste completo via Qucs e via *i*-MOS. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5

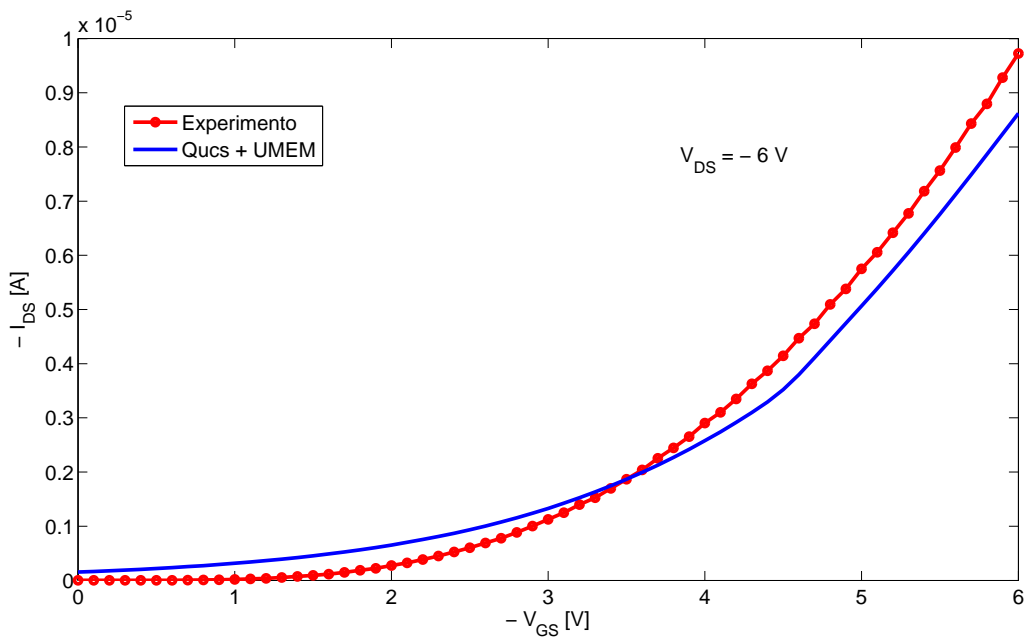


Figura 4.17: Curva de transferência do transistor utilizando os valores dos parâmetros com os quais se realizou o ajuste completo da curva característica de saída. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.5

Primeiro, através do parâmetro m , obtém-se a mesma forma do joelho da curva. O parâmetro R funciona de forma a alterar o módulo da corrente, deslocando a curva modelada para cima ou para baixo, de acordo com o valor inserido. Em sequência, altera-se os parâmetros V_T e α_s para

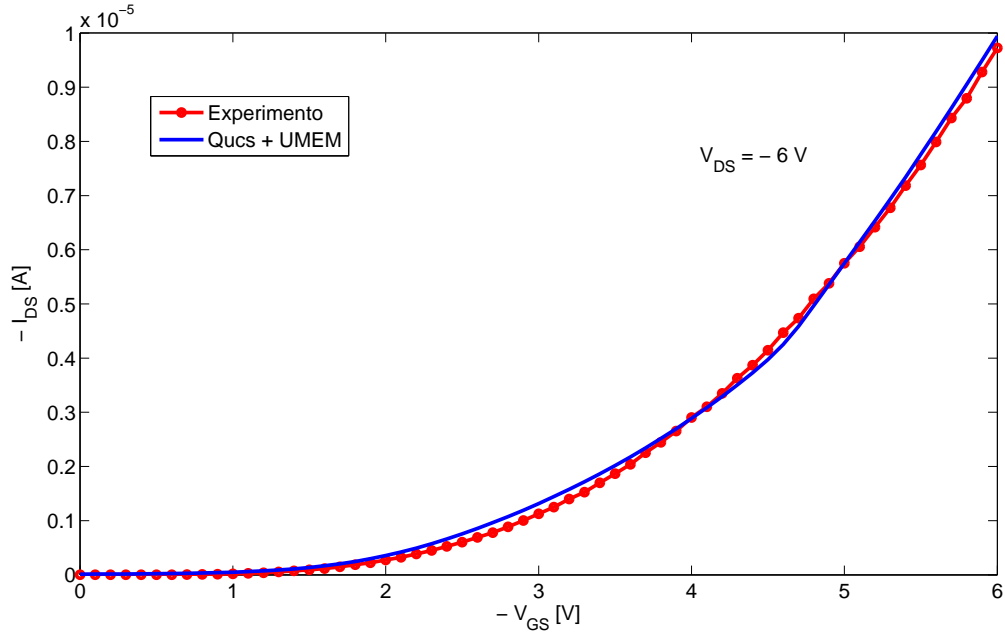


Figura 4.18: Curva de transferência do transistor com o ajuste completo realizado. Os valores dos parâmetros utilizados na realização desse ajuste encontram-se na Tabela 4.6

Tabela 4.6: Parâmetros utilizados para o ajuste completo da curva de característica de saída e da curva de transferência

| Parâmetro | Figura 4.15 | Figura 4.18 |
|------------------|---------------------|---------------------|
| V_{DS} [V] | -6 | -6 |
| V_T [V] | -3.95 | -4 |
| γ_a | 5 | 6 |
| V_{aa} [V] | 3 | 3 |
| R [Ω] | 3.7×10^5 | 2×10^5 |
| α_s | 1.165 | 0.78 |
| m | 2.2 | 1.5 |
| λ [1/V] | -3×10^{-3} | -3×10^{-3} |
| ξ | 7 | 2 |

se alterar a região de saturação. Os parâmetros V_{aa} , γ_a e λ também podem ser alterados para se obter o formato correto da curva, embora, nesse caso específico, eles tenham sido pouco utilizados. O último parâmetro, ξ , apenas reproduz a existência de contatos não-ôhmicos no OTFT. A partir de certo valor desse parâmetro, o efeito não pode ser percebido na curva.

Capítulo 5

Conclusões

Neste trabalho foram apresentados conceitos básicos sobre transistores de filmes finos orgânicos, modelos compactos, Verilog-A, Qucs e UMEM.

Um breve tutorial sobre o Qucs e sobre como criar um novo dispositivo no simulador através de um código Verilog-A foi apresentado, sendo um possível material de consulta para usuários futuros desse simulador, especialmente aos próximos alunos e pesquisadores que possam vir a trabalhar com essa software nos projetos do LDCL.

Foi desenvolvido, ainda, um algoritmo para realizar o ajuste entre as curvas da infra-estrutura Qucs + UMEM e as curvas experimentais fornecidas pela bibliografia. Através desse algoritmo, esse ajuste foi realizado com sucesso.

A infra-estrutura Qucs + UMEM foi implementada com sucesso e, através de ajustes manuais dos parâmetros do modelo, foi possível reproduzir tecnologias existentes de transistores de filmes finos orgânicos. Após se realizar o ajuste manual, pode-se dizer que o transistor está devidamente representado no simulador e, a partir daí, realizar simulações de circuitos maiores contendo esse transistor.

Apesar de ser possível realizar o ajuste entre curvas de característica de saída e curvas de transferência experimentais e simuladas utilizando a infra-estrutura Qucs + UMEM, percebeu-se uma grande discrepância entre as curvas produzidas por essa infra-estrutura ao se utilizar os parâmetros extraídos pelo UMEM através do código em Mathcad e as curvas produzidas pelo próprio Mathcad ou ainda as fornecidas pela literatura. Como forma de checar se o problema era advindo do Qucs em si, utilizou-se outro simulador, *i*-MOS, para checar a saída gerada pelo código em Verilog-A. Os resultados obtidos via Qucs e via *i*-MOS foram os mesmos, provando assim que o problema não encontra-se no Qucs, mas que de fato, há uma discrepância entre o código Verilog-A e o código Mathcad, mesmo que ambos implementem o UMEM com sucesso.

5.1 Recomendações para trabalhos futuros

Para novos trabalhos, seria interessante verificar mais detalhadamente o porquê da discrepância encontrada entre Mathcad e Qucs/*i*-MOS quando os valores extraídos são inseridos nesses simuladores de circuitos, realizando-se novas extrações a partir de dados experimentais. Recomenda-se ainda que o procedimento de extração de parâmetros do UMEM, proposto em [19] e [1], seja implementado em MATLAB. O Mathcad apresenta uma interface do usuário um pouco falha, ocorrendo com frequência diversos erros indesejáveis graças a sua tela de difícil manipulação. Por isso, o MATLAB seria uma boa ferramenta para realizar a extração dos parâmetros. Por fim, o MATLAB é uma ferramenta bem mais disseminada entre os alunos e pesquisadores do nosso departamento, tornando o futuro código de extração de parâmetros do UMEM de uso mais fácil e universal.

O UMEM servirá futuramente, para o grupo do LDCI, como um ponto de referência para avaliar outros modelos compactos de OTFTs fabricados em plataformas de tecnologias emergentes. Devido a essas tecnologias ainda não serem exatamente bem definidas, uma infra-estrutura altamente adaptável é necessária, e a infra-estrutura Qucs + UMEM mostrou-se capaz de realizar essa função.

Além disso, a infra-estrutura implementada Qucs + Modelos em Verilog-A devem ser usados para projetar circuitos eletrônicos baseados em OTFTs, por exemplo, os circuitos que constituem os componentes básicos de um sistema “verde” RFID (*Radio Frequency Identification*).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ESTRADA, M. et al. Accurate modeling and parameter extraction method for organic tfts. *Solid-State Electronics*, v. 49, p. 1009–1016, 2005.
- [2] MOHAMMADI, S. *Untitled Manuscript*. Unpublished.
- [3] KLAUK, H. *Organic Electronics*. [S.l.]: WILEY-VCH, 2006.
- [4] A.KAHN; KOCH, N.; GAO, W. Electronic structure and electrical properties of interfaces between metals and pi-conjugated molecular films. *Journal Of Polymer Scienc Part B: Polymer Physics*, v. 41, p. 2529–2548, 2003.
- [5] R.A.STREET; SALLEO, A. Contact effects in polymers transistors. *Applied Physics Letters*, v. 81(15), p. 2887–2889, 2002.
- [6] HOROWITZ, G.; HAJLAOUI, M.; HAJLAOUI, R. Temperature and gate voltage dependence of hole mobility in polycrystalline oligothiophene thin film transistors. *Journal of Applied Physics*, v. 87 (9), p. 4456–4463, 2000.
- [7] NECLIUDOV, P. et al. Contact resistance extraction in pentacene thin film transistors. *Solid State Electronics*, v. 47(2), p. 259–262, 2003.
- [8] NECLIUDOV, P. et al. Modeling of organic thin film transistors of different designs. *Journal of Applied Physics*, v. 88, p. 6594, 2000.
- [9] ZAUMSEIL, J.; BALDWIN, K. W.; ROGERS, J. A. Contact resistance in organic transistors that use source and drain electrodes formed by soft contact lamination. *Journal of Applied Physics*, v. 93, p. 6117–6124, 2003.
- [10] KOCH, N. et al. Conjugated organic molecules on metal versus polymer electrodes: Demonstration of a key energy level alignment mechanism. *Applied Physics Letters*, v. 82, p. 70–72, 2003.
- [11] VISSENBERG, M. C. J. M.; MATTERS, M. Theory of the field-effect mobility in amorphous organic transistors. *Physical Review B*, v. 57, p. 12964–12967, 1998.
- [12] XIA, Y.; WHITESIDES, G. M. Soft lithography. *Angewandte Chemie*, v. 37, p. 550–575, 1998.

- [13] SIRRINGHAUS, H. et al. High-resolution inkjet printing of all-polymer transistor circuits. *Science*, v. 290, p. 2123–2126, 2000.
- [14] BLANCHET, G. B. et al. Large area, high resolution, dry printing of conducting polymers for organic electronics. *Applied Physics Letters*, v. 82, p. 463–465, 2003.
- [15] KELLEY, T. W. et al. High performance organic thin film transistors. *Mater. Res. Soc. Symp. Proc.*, v. 771, p. 6594, 2003.
- [16] DEPEYROT, G.; POULLET, F.; DUMAS, B. Verilog-a compact model coding whitepaper. *Dolphin Integration*, v. 2, 2010.
- [17] CORAM, G. J. How to (and how not to) write a compact model in verilog-a. In: *IEEE International Behavioral Modeling and Simulation Conference*. [S.l.: s.n.], 2004. p. 97–106.
- [18] Qucs Quite Universal Circuit Simulator. <http://qucs.sourceforge.net/index.html>. Acessado: 2015-11-30.
- [19] A.CERDEIRA et al. New procedure for the extraction of basic a-si:h tft model parameters in the linear and saturation regions. *Solid-State Electronics*, v. 45, p. 1077–1080, 2001.
- [20] ZHANG, L. et al. Standardizing the compact model developments for emerging transistors. *Mixed Design of Integrated Circuits Systems (MIXDES)*, -, p. 70–73, 2014.
- [21] A.CASTRO-CARRANZA. *UMEM-based capacitance model for organic field effect transistors: development and implementation*. Tese (Doutorado) — Universitat Rovira I Virgili, 2013.
- [22] PUIGDOLLERS, J. et al. Pentacene thin-film transistors with polymeric gate dielectric. *Organic Electronics*, v. 5, p. 67, 2004.

ANEXOS

I. CÓDIGO EM VERILOG-A QUE IMPLEMENTA O UMEM

```
//MODEL FOR OTFT BELOW AND ABOVE THRESHOLD
// CINVESTAV-IPN 2012

#include "constants.vams"
#include "discipline.vams"
#define P_K      (1.3806226e-23) //Boltzmann cionstant in eV/K
#define P_Q      (1.6021918e-19) //Charge Cb
#define M_PI     (3.14159265358979323846)
#define P_E0     (8.854214871e-12) //VACCUM PERMITTIVITY IN [F/m]

//Define module with 3 terminals
module otftmi(d, g, s);

//Declare input/output nodes ("inout")
inout d, g, s;

// type of nodes "electrical"
electrical d, g, s;

// Internal nodes
branch (d, s) canal;

//Parameter definition

// TFT geometry
parameter real TYPE=1 from[-1:1] exclude 0; //1.0 N-TYPE, -1.0 P-TYPE
parameter real L = 10e-6 from[0.0:inf]; //CHANNEL LENGTH IN [m]
parameter real W = 110e-6 from[0.0:inf]; //CHANNEL WIDTH IN [m]

// DEFAULT PARAMETERS TO BE CHANGED EXTERNALLY

//DEFAULT MODEL PARAMETERS
parameter real ALPHASAT=0.572; //SATURATION PARAMETER IN [eV]
parameter real EPS=6.5; // RELATIVE DIELECTRIC CONSTANT OF THE SEMICONDUCTOR
parameter real EPSI=2.6; //RELATIVE DIELECTRIC CONSTANT OF THE GATE DIELECTRIC
parameter real GAMMA=1.08; // MOBILITY DEPENDENCE ON (VGS-VT)
parameter real LAMBDA=-1.3e-3; //OUTPUT CONDUCTANCE IN [1/V]
parameter real M=1.475; //INFLECTION PARAMETER OF OUTPUT CHARACTERISTIC (KNEE)
parameter real RD = 1.474e4; //SERIES RESISTANCE AT DRAIN IN [Ohms]
parameter real RS = 1.474e4; //SERIES RESITANCE AT SOURCE IN [Ohms]
parameter real TNOM=27; //MEASUREMENT TEMPERATURE IN [C]
parameter real TOX=3.7e-7; //DIELECTRIC LAYER THICKNESS IN [m]
parameter real VO=0.04; //CHARACTERISTIC VOLTAGE OF DEEP STATES IN [V]
parameter real VAA=1.634e3; //LOW FIELD MOBILITY PARAMETER
parameter real VTO=2.711; //THRESHOLD VOLTAGE IN [V]
parameter real SL=3; //SUBTRHESHOLD SLOPE OF LINEAR TRANSFER CURVE
parameter real DVL=2; //DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE IN [V]
parameter real QL=0.4; //TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF LINEAR TRANSFER CURVE
parameter real IOL=1.006e-10; //IOFF FOR LINEAR TRANSFER CURVE IN [A]
parameter real SS=3; //SUBTRHESHOLD SLOPE OF TRANSFER CURVE IN SATURATION
parameter real DVS=2; //DEFINES VOLTAGE FOR SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE IN [V]
```



```

parameter real QS=0.4; //TANH PARAMETER SEWING BELOW-ABOVE THRESHOLD REGIONS OF SATURATION TRANSFER CURVE
parameter real IOS=5.144e-9; //IOFF FOR SATURATION TRANSFER CURVE IN [A]

// DEFAULT MODEL PARAMETERS FOR NON-OHMIC CONTACT AT D AND S

parameter real ETAD=0.25; //RESISTANCE DRAIN VOTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real ETAG=0.25; //RESISTANCE GATE VOLTAGE DEPENDENCE ADJUSTING PARAMETER
parameter real RCO=1e13; //ADJUSTING PARAMETER

//DEFAULT MODEL PARAMETERS FOR SMICONDUCTOR LAYER CONDUCTANCE BETWEEN D AND S

parameter real CCt=3.5e9; //SEMICONDUCTOR LAYER CONDUCTANCE IN [S]

//INTERNAL VARIABLES
real EPSIO, EPSO;

//OTHER VARIABLES
real MOS25cg, MUFET,CC;

//TEMPERATURE DEPENDENT VARIABLES
real TNOMO;
real VT;

//RESISTANCE RELATED VARIABLES
real R;
real RO;
real RC;
real RT;

// VOLTAGE RELATED VARIABLES
real Vds, Vgs, Vgd;
real vgs, vgd, vds, mode;
real vgt, vgte, vdse, vsate;

// VARIABLES IN BELOW THRESHOLD
real vgtDL, vgtedL, vsatedL, vdsedL;
real nsadL, gchidL, gchdL, idcdL;
real idscL;

// VARIABLES IN ABOVE THRESHOLD
real vgtDS, vgtedS, vsatedS, vdsedS;
real nsadS, gchidS, gchdS, idcdS;
real idscS;

//VARIABLES FOR COMPLETE BELOW AND ABOVE REGIONS
real idc, ids, idsL, idsS, ifinal;

// OTHER VARIABLES REQUIRED
real nsa, gchi, gch, VMIN, DELTA;

analog
begin
//INITIAL CALCULATIONS

RO = 1;
VMIN = 0.01;
DELTA = 1;
TNOMO =TNOM+'P_CELSIUSO;
EPSIO = 'P_EO * EPSI;

```

```

        EPSO = 'P_EO * EPS;
        VT= -VTO;
        MOS25cg = EPSIO / TOX;
        MUFET = 1/pow(VAA,GAMMA);
        CC=Cct/(W/L);

Vds      = TYPE * V(d, s);
Vgs      = TYPE * V(g, s);
Vgd      = TYPE * V(g, d);

//NODE INITIALIZATION
    if (Vds >= 0.0)
        begin
            mode = 1;
            vds = Vds;
            vgs = Vgs;
            vgd = Vgd;
        end
    else
        begin //modo inverso
            mode = -1;
            vds = -Vds;
            vgs = Vgd;
            vgd = Vgs;
        end

//CALCULATIONS
//CALCULATION OF OHMIC AND NON-OMIC RESISTANCE AT CONTACTS

    R = RS + RD;
    RC = R0 + (RCO * exp(-ETAD * vds) * exp(-ETAG * vgs));
    RT = R + RC;

// EXPRESSION TO CALCULATE CURVES AT VG=VT
    vgt = (VMIN/2) * (1+((vgs-VT)/VMIN) + sqrt((DELTA*DELTA)+pow(((vgs-VT)/VMIN)-1),2)));

//CALCULATION OF BELOW THRESHOLD LINEAR CHARACTERISTICS

    vgtgL = VT + DVL;
    vgtgL = vgtgL - VT;
    vsatedL = ALPHASAT * vgtgL;
    vdsedL = vds / (pow((1 + (pow((vds/vsatedL),M))), (1 / M)));
    nsadL = (pow(vgtgL ,GAMMA)) * vgtgL * (MOS25cg / 'P_Q);
    gchidL = 'P_Q * nsadL * MUFET * (W / L);
    gchdL = gchidL / (1 + (gchidL * RT));
    idcdL = gchdL * vdsedL * (1 + (LAMBDA * vds));
    idscL = idcdL * exp(((vgs - VT) / SL) * 2.3);

//CALCULATION OF BELOW THRESHOLD CHARACTERISTICS IN SATURATION

    vgtgS = VT + DVS;
    vgtgS = vgtgS - VT;
    vsatedS = ALPHASAT * vgtgS;
    vdsedS = vds / (pow((1 + (pow((vds/vsatedS),M))), (1 / M)));
    nsadS = (pow(vgtgS ,GAMMA)) * vgtgS * (MOS25cg / 'P_Q);
    gchidS = 'P_Q * nsadS * MUFET * (W / L);
    gchgS = gchidS / (1 + (gchidS * RT));
    idcdS = gchgS * vdsedS * (1 + (LAMBDA * vds));
    idscS = idcdS * exp(((vgs-VT) / SS) * 2.3);

```

```

//CALCULATION OF ABOVE THRESHOLD CHARACTERISTICS

vsate = ALPHASAT * vgt;
vdse = vds / (pow((1 + (pow((vds/vsate),M))), (1 / M)));
nsa = (pow(vgt ,GAMMA)) * vgt * (MOS25cg / 'P_Q);
gchi = 'P_Q * nsa * MUFET * (W / L);
gch = gchi / (1 + (gchi * RT));
idc = (gch * vdse * (1 + (LAMBDA * vds))) + (vds/CC);

//CALCULATION OF CURRENTS

idsL = abs(idscL) * ((1 - tanh((vgs - vgtDL) * QL)) / 2) + abs(idc) * ((1 + tanh((vgs - vgtDL) * QL)) / 2) + IOL;
idsS = abs(idscS) * ((1 - tanh((vgs - vgtDS) * QS)) / 2) + abs(idc) * ((1 + tanh((vgs - vgtDS) * QS)) / 2) + IOS;

if (vgs < VTO)
    ids = idsL;
else
    ids = idsS;

ifinal = ids;

if (mode > 0)
    I(canal) <+ TYPE * ifinal;
else
    I(canal) <+ -TYPE * ifinal;
end

endmodule

```

II. DESCRIÇÃO DO CONTEÚDO DO CD