

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

CIRCUITOS ELÉTRICOS II

PROFESSOR ANTÔNIO CARLOS MOREIRÃO DE QUEIROZ

**Análise Nodal Modificada Para
Circuitos Lineares com Transistores
MOS: Ponto de Operação e Resposta
em Frequência**

Alunos:

Joao Pedro K. Marques

Maria Luiza C. Vianna

Miguel de Sousa



UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO

1 Introdução

Como trabalho da matéria de Circuitos Elétricos II, o nosso grupo desenvolveu o simulador OrKappes (trocadilho entre o nome do nosso membro e o famoso programa simulador de circuitos). O programa resolve por meio do método da Análise Nodal Modificada circuitos lineares que comportam os seguintes elementos: Resistor, Indutor, Acoplamento entre Indutores, Capacitor, Fonte de tensão controlada a tensão, Fonte de corrente controlada a corrente, Fonte de corrente controlada a tensão, Fonte de tensão controlada a corrente, Fonte de corrente, Fonte de tensão, Amplificador Operacional Ideal e o Transistor MOS(tanto canal N como canal P).

Desenvolvemos o programa em C++ com o VisualStudios 2015 e utilizamos a plataforma GitHub como controle de versão. Escolhemos C++ pela facilidade da orientação a objeto e escolhemos o GitHub pois é uma das ferramentas mais utilizadas hoje em dia para controle de versão.

2 OrKappes

2.1 Funcionamento

O programa principal lê um arquivo do tipo ".net" que contém a netlist a ser processada. Após, ele cria objetos para cada componente de forma dinâmica, isso garante que o nosso programa não tenha um número limite de componentes. A partir da criação do array de componentes, é a hora de montar as suas estampas na matriz de ponto de operação (GMatrix) e processá-la com o algoritmo de Newton-Raphson, caso haja algum componente a ser linearizado. Com a matriz de condutância pronta, utilizamos um resolvidor de sistemas lineares que utiliza o Método de Gauss-Jordan com condensação pivotal para nos informar as tensões e correntes do sistema: o ponto de operação. Vale um comentário a respeito do MOSFET, como ele é um objeto, a cada iteração do Newton-Raphson os parâmetros dele são calculados, até suas capacitâncias parasitas. Isso evita que precisemos calculá-las na hora de realizar a análise espectral.

Depois de calcular o ponto de operação, as tensões e correntes são salvas em arquivos ".dc" e o programa começa a fazer a análise da resposta em frequência. Nessa parte,

montamos a matriz de condutância AC e varremos o sistema desde a frequência mais baixa até a mais alta, variando o passo de acordo com o informado pela netlist. Ao final dessa etapa, o programa salva a solução para cada frequência em um arquivo ".tab".

2.2 Métodos do Programa

2.2.1 Netlist(string netlistPath)

Função recebe o path para o arquivo ".net", lê a net list e armazena o array de componentes. Nesse método calcula-se o número de tensões e de correntes do sistema.

2.2.2 DoConductanceMatrixDC(void)

Utiliza o array de componentes da classe Netlist e compõe a matriz de condutancia, essa matriz é gerada dinamicamete a partir do numero de tensões e de correntes do sistema.

2.2.3 NewtonRaphson(void)

Nesse método, é implementado o conhecido algoritmo de Newton-Raphson.

2.2.4 WriteDCData(void)

Salvamos o ponto de operação em um arquivo ".dc".

2.2.5 ACSweep(void)

Montamos as estampas AC e resolvemos para cada frequência o sistema linear complexo por meio do Método de Gauss-Jordan com condensação pivotal. Por fim, guardamos o resultado em um arquivo ".tab"

2.3 Exemplos

2.3.1 LeapFrog:

- Netlist

```
15
R0100 1 0 1E+3
R0203 2 3 1E+3
R0402 4 2 1E+3
R0504 5 4 1E+3
R0105 1 5 1E+3
R0603 6 3 1E+3
R0701 7 1 1E+3
R0809 8 9 1E+3
R1011 10 11 1E+3
R1208 12 8 1E+3
R1310 13 10 1E+3
R0614 6 14 1E+3
R1206 12 6 1E+3
R1512 15 12 1E+3
R1315 13 15 1E+3
R0713 7 13 1E+3
C0902 9 2 9.32778700555236E-7
C1105 11 5 1.4031540561078E-6
C1503 15 3 1.7274276550171E-6
C0115 1 15 5.79956913281569E-7
C0603 6 3 3.97974815527259E-6
C1504 15 4 5.69845860607148E-6
C0701 7 1 3.51512140879876E-6
C0406 4 6 1.7274276550171E-6
C0704 7 4 5.79956913281569E-7
O0900 9 0 2 0
O1100 11 0 5 0
O0300 3 0 6 0
O0400 4 0 15 0
O0100 1 0 7 0
O1200 12 0 8 0
O1300 13 0 10 0
V1400 14 0 1
.AC DEC 2 0.1 1000
```

- Resultado:

