

Circuitos Elétricos II – 1º semestre de 2013 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos no tempo, incluindo transistores bipolares com efeito Early. O programa deverá analisar um circuito composto pelos elementos:

- Fontes de corrente e de tensão independentes (DC, pulso, senóide).
- Resistores, capacitores e indutores lineares invariantes no tempo.
- As quatro fontes controladas, lineares.
- Amplificadores operacionais ideais, de 4 terminais.
- Diodos.
- Transistores bipolares NPN e PNP.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados por uma linha de comando no netlist. O resultado deverá ser uma tabela em arquivo, tendo o tempo como primeira coluna, e todas as tensões nodais e correntes nas fontes de tensão nas outras colunas, plotável com outro programa. A primeira linha da tabela deve listar os nomes de todas as variáveis calculadas, com correntes citadas como “j” seguidas do nome do elemento onde estão. Ex: t 1 2 3 4 5 6 jH1 jF2.

O método de integração numérica a usar é o dos trapézios com passo fixo. Inicialmente deverá ser feita uma análise usando um passo muito menor que o normal (dividir por 10^9) usando o método backward de Euler, para completar a solução em $t=0$ a partir das condições iniciais sobre indutores e capacitores, e dos valores em $t=0$ das fontes independentes.

A análise deverá ser feita usando análise nodal modificada. O programa pode ser baseado no programa exemplo MNA1, que implementa o algoritmo para um circuito resistivo linear:

<http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>

Formato para o netlist:

O netlist pode ser gerado pelo programa Edfil, e deve ser compatível com o aceito pelo programa exemplo MNAE. Veja os programas em <http://www.coe.ufrj.br/~acmq/cursos>

Primeira linha: Comentário, ignorar (o editor Edfil coloca o número de nós nesta linha).

Linhas seguintes: Descrição do circuito, com um elemento por linha. A primeira letra (ou símbolo) determina o tipo de elemento.

Resistor: R<nome> <nó1> <nó2> <Resistência>

Indutor: L<nome> <nó1> <nó2> <Indutância> [IC=<corrente inicial>]

Capacitor: C<nome> <nó1> <nó2> <Capacitância> [IC=<tensão inicial>]

Fonte de tensão controlada a tensão: E<nome> <nóV+> <nóV-> <nóv+> <nóv-> <Av>

Fonte de corrente controlada a corrente: F<nome> <nóI+> <nóI-> <nói+> <nói-> <Ai>

Fonte de corrente controlada a tensão: G<nome> <nóI+> <nóI-> <nóv+> <nóv-> <Gm>

Fonte de tensão controlada a corrente: H<nome> <nóV+> <nóV-> <nói+> <nói-> <Rm>

Fonte de corrente: I<nome> <nó+> <nó-> <Parâmetros>

Fonte de tensão: V<nome> <nó+> <nó-> <Parâmetros>

Amplificador operacional ideal: O<nome> <nó saída+> <nó saída-> <nó entrada+> <nó entrada->

Diodo: D<nome> <nó+> <nó-> [<Is> <Vt>]

Trans. BJT: Q<nome> <nóc> <nób> <nóe> <tipo> [<alfa> <alfar> <Isbe> <Vtbe> <Isbc> <Vtbc> <VA>]

Comentário: *<comentário>

(Notar que <xxx> significa colocar o valor xxx sem <>, e [xxx] indica que xxx é opcional, sem [].)

O diodo segue a regra $j=I_s(e^{(v/V_t)}-1)$. Se não forem dados I_s e V_t , usar $V_t=25e-3$ e $I_s=3.7751345e-14$. Assim o diodo conduz $j=1$ mA com $v=0.6$ V.

Os transistores bipolares seguem o modelo de Ebers-Moll, com uma modificação para incluir o efeito Early, que é modelado fazendo a corrente de coletor valer a corrente normal do modelo multiplicada por $(1+V_{ce}/V_A)$, para o transistor NPN. O tipo pode ser NPN ou PNP. Se os parâmetros não forem dados, usar diodos como o padrão, $\alpha=0.99$, $\alpha_f=0.5$, e $V_A=100$ V.

Os programas exemplo permitem nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algoritmo (código no programa MNA1), ou admitir apenas números. Neste caso a primeira linha gerada pelo editor Edfil pode ser usada.

As direções para fontes são de acordo com a ordem dos nós e as direções convencionais associadas, sendo o primeiro nó o positivo. Os parâmetros para as fontes podem ser:

DC <valor>

SIN <nível contínuo> <amplitude> <frequência (Hz)> <atraso> <atenuação> <ângulo> <número de ciclos>

PULSE <amplitude 1> <amplitude 2> <atraso> <tempo de subida> <tempo de descida> <tempo ligada> <período> <número de ciclos>

A fonte senoidal, antes do atraso ou após o número de ciclos vale: nível contínuo+amplitude*sen(ângulo* π /180). No resto do tempo vale: nível contínuo+amplitude*exp(-t0*atenuação)*sen(2* π *frequência*t0+ângulo* π /180), onde t0=tempo-atraso.

A fonte pulsada começa na amplitude 1, e fica aí até o fim do tempo de atraso. Então muda para a amplitude 2 variando linearmente dentro do tempo de subida, fica na amplitude 2 durante o tempo ligada, volta à amplitude 1 dentro do tempo de descida, e repete tudo com o período e o número de ciclos especificados. Termina na amplitude 1. Cuidado com como tratar tempos de subida e de descida nulos. Pode ser usado o tempo do passo.

O programa deve ler as instruções de como tratar o netlist de uma linha de comando no próprio netlist, no formato abaixo. Não deve ser necessário fornecer qualquer outro parâmetro ao programa além do arquivo de entrada. Os passos internos permitem aumentar a precisão da análise, com alguns passos entre os valores que vão à tabela de saída. O passo interno usado é então o passo dado dividido pelo número de passos internos.

.TRAN <tempo final> <passo> TRAP <passos internos> UIC

O parâmetro TRAP indica o método dos trapézios (o programa exemplo MNAE aceita outros). O comando UIC significa “use initial conditions”, pois normalmente programas de análise no tempo fazem uma análise de ponto de operação e acham dela as condições iniciais antes de iniciar a análise.

O programa MNAE pode ser usado para plotar os gráficos de saída e para verificação.

O programa deve contar quantas vezes o ciclo de Newton-Raphson é executado, e se o número passar de um valor razoável, tentar outra aproximação inicial para a solução. Deve contar também quantas vezes faz isto, e se o número passar de um valor razoável, abortar a análise.

O programa deve ser escrito preferencialmente em uma linguagem compilada como C, C++ ou Pascal. O programa deve preferencialmente rodar em ambiente gráfico Windows. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes, arquivo executável, documentação e exemplos não deve ter mais de 3 Mbytes. O programa MNAE foi escrito usando o Borland C++ Builder.

Sugere-se partir do programa exemplo MNA1, que já tem o algoritmo completo da análise nodal modificada, implementar a análise no tempo com as fontes de sinal, capacitores e indutores, e então o método de Newton-Raphson e os elementos não lineares.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado por todo o grupo, e entregue com um relatório com comentários e exemplos significativos e originais verificados, até uma semana antes da segunda prova. Note que o trabalho é bastante extenso, e deve ser começado imediatamente. Atrasos na entrega atrapalham o fim do semestre, e implicarão em redução significativa da nota.