

Circuitos Elétricos II – 1º semestre de 2016 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Escrever um programa que analise circuitos lineares contendo transistores MOS, encontrando o ponto de operação e a resposta em frequência para pequenos sinais.

O programa deverá analisar um circuito composto pelos elementos:

- Fontes de corrente e de tensão independentes.
- Resistores, capacitores, indutores e transformadores.
- As quatro fontes controladas.
- Amplificadores operacionais ideais, de 4 terminais.
- Transistores MOS, com modelo de “nível 1”.

O programa deve ler um netlist, e realizar uma análise nodal modificada de ponto de operação, e após isto deve realizar análises no estado permanente senoidal do modelo de pequenos sinais, em uma faixa de frequências especificada. Uma linha de comando no netlist define os parâmetros necessários. O resultado deverão ser duas tabelas em arquivos, uma listando o ponto de operação encontrado (o programa pode apenas listar os valores do ponto de operação sem salvar a tabela, mas de forma que todos possam ser examinados), e outra tabela tendo a frequência como primeira coluna, e todas as tensões nodais e correntes nas fontes de tensão e curtos-circuitos nas outras colunas, em módulo e fase, plotável com outro programa. A primeira linha desta tabela deve listar os nomes de todas as variáveis na tabela, com correntes citadas como “j” seguidas do nome do elemento onde estão e sufixos m e f para indicar módulo e fase. Ex: f 1m 1f 2m 2f 3m 3f jH1m jH1f jF2m jF2f.

O programa pode ser baseado no programa exemplo MNA1, que implementa a análise nodal modificada para um circuito resistivo linear. Uma versão gráfica (para o Borland C++ builder 6) está também disponível:

<http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>
<http://www.coe.ufrj.br/~acmq/cursos/mna1gr.zip>

Formato para o netlist:

O netlist pode ser gerado pelo programa Edfil. Veja os programas em <http://www.coe.ufrj.br/~acmq/cursos>

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

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

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

Indutor: L<nome> <nó1> <nó2> <Indutância>

Acoplamento entre indutores: K<nome> <La> <Lb> <k> (La e Lb nomes de indutores já declarados.)

Capacitor: C<nome> <nó1> <nó2> <Capacitância>

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-> <módulo> <fase (graus)> <valor contínuo>

Fonte de tensão: V<nome> <nó+> <nó-> <módulo> <fase (graus)> <valor contínuo>

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

Transistor MOS: M<nome> <nód> <nóg> <nós> <nób> <NMOS ou PMOS> L=<comprimento> W=<largura> <K> <Vt0> <λ> <γ> <φ> <Ld>

Comentário: *<comentário>

(Notar que <xxx> significa colocar o valor xxx sem <>.)

O programa exemplo permite 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.

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.

.AC <LIN ou OCT ou DEC> <pontos> <início> <fim>

Como no SPICE, o número de pontos é por década ou por oitava com as opções DEC e OCT. Senão é o total com a opção LIN. As frequências são em Hz. Pode haver uma opção interna no programa para usar frequências em rad/s.

Opcionalmente, o próprio programa pode plotar seus gráficos. O programa MNARF pode ser usado para plotar os gráficos. Este programa faz exatamente a mesma análise, mas sem o cálculo do ponto de operação. Ele trata transistores MOS, mas usando um modelo de pequenos sinais.

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

O programa deve ser escrito preferencialmente em uma linguagem compilada como C, C++ ou Pascal. O programa deve preferencialmente rodar em ambiente gráfico Windows. Evite usar apenas uma interface de console. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes, arquivo executável, documentação, bibliotecas e exemplos não deve ter mais de 4 Mbytes. Evite sistemas de desenvolvimento que requerem extensas bibliotecas instaladas.

Sugere-se partir do programa exemplo MNA1, que já tem o algoritmo completo da análise nodal modificada em C, implementar a análise do ponto de operação com transistores MOS, e então a análise no estado permanente senoidal, operando com variáveis complexas. Note que a montagem do sistema de equações e sua resolução são similares em todas as análises, com as mesmas variáveis. Embora seja ineficiente, a solução do sistema complexo pode ser usada no cálculo do ponto de operação também. Em C, é simples usar variáveis complexas.

O programa deve tratar as capacitâncias dos transistores MOS, que dependem da região de operação do transistor. O valor de C_{ox} pode ser obtido de $K=\mu C_{ox}/2$, com $\mu=0.05$ para NMOS e 0.02 para PMOS. W e L são as dimensões (largura e comprimento) do transistor, e L_d a superposição efetiva entre g e d ou s . Os valores usuais para estas capacitâncias são:

	Corte, $v_{gs} < V_t$	R. ôhmica, $v_{ds} < v_{gs} - V_t$	Saturação, $v_{ds} \geq v_{gs} - V_t$
C_{gs}	$C_{ox} W L_d$	$C_{ox} W L_d + \frac{1}{2} C_{ox} W L$	$C_{ox} W L_d + \frac{2}{3} C_{ox} W L$
C_{gd}	$C_{ox} W L_d$	$C_{ox} W L_d + \frac{1}{2} C_{ox} W L$	$C_{ox} W L_d$
C_{bg}	$C_{ox} W L$	0	0

Neste programa, não é necessário considerar as capacitâncias dos diodos bs e bd , que dependem da geometria. Estes diodos podem também ser ignorados no cálculo do ponto de operação.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado, completamente funcional, por todo o grupo, e entregue com um relatório com comentários e exemplos significativos e originais verificados, até (entenda-se antes de) duas semanas antes da segunda prova.

Note-se que o trabalho é bastante extenso, e deve ser começado imediatamente.