Circuitos Elétricos II – 2º semestre de 2007 - Trabalho

Fazer um programa que analise circuitos no tempo, e que aceite, no mínimo, os elementos:

- Fontes de corrente e de tensão independentes (DC, pulso, senóide).
- Resistores, capacitores e indutores.
- As quatro fontes controladas.
- Transistores MOS, canal N e canal P.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados pelo operador (tempo, passo fixo). 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, separadas por espaços, plotável com outro programa (XYplot, Gnuplot).
Capacitores e indutores devem poder ter uma condição inicial.
Os valores numéricos devem ser números (sem multiplicadores literais, como n, u, m u, k, etc). Também não é necessário suportar nomes de nós. Pode-se assumir os nós numerados seqüencialmente a partir de 0 (terra) e dados como números (note que o programa exemplo suporta nós com nomes).
Os transistores devem ser modelados pelo modelo de Shichman-Hodges, com V_t fixo. Para o transistor NMOS, as equações são:

$$i_D = \begin{cases} 0 \quad \text{para } v_{GS} < V_t \\ K\dfrac{W}{L}\left[2(v_{GS}-V_t)v_{DS} - {v_{DS}}^2\right](1+\lambda v_{DS}) \quad \text{para } v_{GS} > V_t \text{ e } v_{DS} < v_{GS} - V_t \\ K\dfrac{W}{L}(v_{GS}-V_t)^2(1+\lambda v_{DS}) \quad \text{para } v_{GS} > V_t \text{ e } v_{DS} \geq v_{GS} - V_t \end{cases}$$

Onde W e L são a largura e o comprimento do canal, dados no netlist como no Spice. Se os parâmetros não forem dados no netlist, use os valores abaixo (ver formato mais adiante):

$$W = L = 1\times10^{-6}$$

$$K = 0.1\times10^{-3}$$

$$V_t = 1$$

$$\lambda = 0.01$$

Para o PMOS, use os mesmos parâmetros, mudando as polaridades onde necessário.
No formato do netlist mostrado abaixo, aparece também o nó b, que seria o substrato. Nesse trabalho, deve ser ignorado, mas estar presente para compatibilidade com futures versões, e com o Spice. Note que o terminal do dreno (D) é o de tensão mais alta entre D e S para o NMOS e o de tensão mais baixa para o PMOS, já que os dispositivos são simétricos.

É interessante adicionar capacitors C_{gs} e C_{gd} ao modelo. Sem eles problemas sérios de convergência podem ocorrer. Os capacitores C_{gs} e C_{gd} são não lineares, mas uma aproximação pode ser feita de considerar ambos iguais e com valor fixo. O valor a usar pode ser obtido da expressão para o parâmetro K:

$$K = \frac{1}{2}\mu C_{OX}$$

$$C_{gs} = C_{gd} = \frac{1}{2}C_{OX}WL = \frac{WLK}{\mu}$$

μ é a mobilidade e C_{OX} a capacitância por unidade de área do gate. Use μ = 0.05 para NMOS e μ = 0.02 para PMOS.

O programa deve poder usar o método “backward” de Euler ou o método de Gear de segunda ordem, acoplado ao método de Newton-Raphson, usando análise nodal modificada. Uma primeira análise com passo pequeno (1/1000 do normal) deve ser feita para achar a solução completa em t=0, usando o método de primeira ordem (BE) e as condições iniciais. A análise seguinte também é feita com o método BE, com as demais pelo método de segunda ordem, se ele for o escolhido. A parte básica da MNA em C pode ser vista no programa http://www.coe.ufrj.br/~acmq/cursos/mna1.zip

Antonio Carlos M. de Queiroz – 10/10/2007. Atualizado em 18/10/2007.

Formato para o netlist:
Primeira linha: Número de nós (ou considerar comentário, se o programa contar os nós)
Linhas seguintes: Descrição do circuito, com um elemento por linha. O formato é livre, com um número qualquer de espaços separando os itens.

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>
Transistor: **M**<nome> <nó d> <nó g> <nó s> <nó b> **NMOS**|**PMOS** [**L=**<comprimento> **W=**<largura>] [<K> <V_t> <λ>]
Comentário: *<comentário>

As fontes F e H tem um ramo de entrada em curto. O formato é diferente do usual do Spice.
Os netlists de entrada podem ser gerados diretamente com o programa editor Edfil (o que também evita que o programa troque os números dos nós, se for usado o algorítmo do programa exemplo), disponível em:
http://www.coe.ufrj.br/~acmq/programs
Os parâmetros para as fontes de tensão ou de corrente devem ser:
DC <valor> *(para fonte DC)*
SIN (<nivel contínuo> <amplitude> <frequencia (Hz)> <atraso> <fator de 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 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.
A fonte senoidal vale (X_i = parâmetro #i):
Antes do tempo de atraso X_4 ou depois do numero de ciclos X_7:

$$X_1 + \text{sen}\left(\pi \frac{X_6}{180}\right)$$

Em outros tempos:

$$X_1 + X_2 e^{(t-X_4)X_5} \text{sen}\left(2\pi X_3 (t - X_4) + \pi \frac{X_6}{180}\right)$$

As direções para fontes e condições iniciais 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, em qualquer lugar, no formato:

.tran <tempo final> <passo> <**BE**|**GEAR**> <passos internos>

Os passos internos servem para aumentar a precisão sem ter que gerar tabelas muito grandes. Entre cada passo listado na tabela é feito o número especificado de passos iguais, com resultados não listados.

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. Em qualquer caso, deve haver meio do usuário interromper a análise.

Opções:

Inclua diodos, transistores bipolares (veja o trabalho do primeiro semestre) o amp. op. ideal e indutores acoplados, Implemente também o método dos trapézios. Plote gráficos (especialmente isso).
O programa deve ser escrito em uma linguagem compilada como C, C++ ou Pascal. Não é aceitável usar Matlab ou similar. **O programa deve rodar em ambiente Windows**, de preferência com interface gráfica. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes e arquivo executável, não deve ter mais de **2 Mbytes**. O programa fonte deve consistir do mínimo número de arquivos permitido no ambiente de desenvolvimento escolhido.

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, que demonstrem que o programa funciona corretamente. Trabalhos não originais ou não funcionando corretamente de acordo com o especificado não serão considerados.