Circuitos Elétricos II – 2º semestre de 2009 - 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 lineares invariantes no tempo.
As quatro fontes controladas lineares, invariantes no tempo, e o amplificador operacional ideal.
Transistores MOS, canal N e canal P.
Diodos.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados pelo operador (tempo final, passo fixo, passos internos). 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. A primeira linha deve conter os nomes das variáveis, na mesma ordem das colunas.
O programa deverá primeiramente calcular o ponto de operação do circuito, zerando os valores dos capacitores e indutores. Esta é a solução em $t = 0$. Em vez de zerar, melhor substituir capacitores por resistores grandes (1 GOhm) e indutores por resistores pequenos (1 nOhm). Isto evita problemas com cortes capacitivos e ciclos indutivos.
Os valores numéricos devem ser números (sem multiplicadores literais, como n, u, m u, k, etc). Os nós do circuito podem ter nomes, exceto pelo de referência, que é o "0".

Os diodos devem seguir a expressão:

$$j = I_s\left(e^{v/V_t} - 1\right)$$

Sem parâmetros dados, devem conduzir 1 mA com $v = 0.6$ V, e ter $V_t = 25$ mV.

Os transistores devem ser modelados pelo modelo de Shichman-Hodges (Nível 1 do SPICE). Para o transistor NMOS, as equações são:

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

$$V_t = V_{t0} + \gamma\left(\sqrt{\phi - V_{BS}} - \sqrt{\phi}\right)$$

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):

$W = L = 1\times10^{-6}$ $\quad \lambda = 0.01$

$K = 0.1\times10^{-3}$ $\quad \phi = 0.6$

$V_{t0} = 1$ $\quad \gamma = 0.3$

Para o PMOS, use os mesmos parâmetros, com os mesmos sinais.
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. Diodos entre o substrato e D e S devem ser incluídos no modelo (use os parâmetros "default").

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:

Antonio Carlos M. de Queiroz – 14/10/2009. Atualizado em 28/10/2009.

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

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

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 de Gear de segunda ordem, acoplado ao método de Newton-Raphson, usando análise nodal modificada. A primeira análise (primeiro tempo acima de 0) é feita com o método BE, a partir da solução do ponto de operação, e as demais pelo método de segunda ordem. A parte básica da MNA em C pode ser vista no programa http://www.coe.ufrj.br/~acmq/cursos/mna1.zip

Formato para o netlist:
Primeira linha: Comentário. Ignorar.
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>
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-> <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_{t0}> <λ> <γ> <ϕ>]
Diodo: D<nome> <nó anodo> <nó catodo> [<Is> <Vt>]
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:
Antes do tempo do atraso ou depois do numero de ciclos:

$$nível\_contínuo + \text{sen}\left(\pi\frac{ângulo}{180}\right)$$

Em outros tempos:

$$nível\_contínuo + amplitude\, e^{(t-atraso)\,fator\_de\_atenuação}\,\text{sen}\left(2\pi\ frequência(t-atraso) + \pi\frac{ângulo}{180}\right)$$

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> <**GEAR**> <passos internos>

Os passos internos servem para aumentar a precisão sem ter que gerar tabelas muito grandes. Os resultados só são colocados na tabela a cada número de passos internos. Para ver todos os passos, use 1.
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.

Opcionalmente, outros métodos de análise podem ser também implementados. Os métodos backward de Euler (que já é mesmo usado na primeira análise) e dos trapézios são de simples implementação. O nome do método (BE, TRAP) na linha de comando definiria qual usar.
É interessante também implementar um sistema gráfico para mostrar as formas de onda.

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, dlls e arquivo executável, não deve ter mais de **2 Mbytes**.

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.