

Circuitos Elétricos II – 2º semestre de 2012 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos no tempo contendo portas lógicas com modelos lineares por partes. 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.
- Portas lógicas: Inversor, AND, NAND, OR, NOR, XOR, e NXOR de 2 entradas.

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 “backward” de Euler com passo fixo. Inicialmente deverá ser feita uma análise usando um passo muito menor que o normal, 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->
Inversor: ><nome> <nó entrada> <nó saída> <Parâmetros>
AND:)<nome> <nó entrada> <nó entrada> <nó saída> <Parâmetros>
NAND: (<nome> <nóentrada> <nó entrada> <nó saída> <Parâmetros>
OR: }<nome> <nó entrada> <nó entrada> <nó saída> <Parâmetros>
NOR: {<nome> <nó entrada> <nó entrada> <nó saída> <Parâmetros>
XOR:]<nome> <nó entrada> <nó entrada> <nó saída> <Parâmetros>
NXOR: [<nome> <nó entrada> <nó entrada> <nó saída> <Parâmetros>
Comentário: *<comentário>

Os programas exemplo permitem nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algoritmo, 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.

Os parâmetros para as portas lógicas são: <V> <R> <C> <A>

A tensão V é a tensão de saída máxima, R a resistência da saída, C a capacitância de entrada, para cada entrada, e o ganho A é a derivada da tensão de saída em aberto em relação à variação da tensão em qualquer entrada, antes da saturação da saída. A saída da porta pode ser modelada como uma fonte de tensão com resistor em série, ou pelo equivalente Norton, o que economiza incógnitas. O valor da tensão de saída em aberto, no caso do inversor, deve valer:

$V_{out} = -A(V_{in} - V/2) + V/2$, limitada entre 0 e V.

Para o caso das portas lógicas, o critério deve ser:

AND com entradas V1 e V2:

Se $V1 > V2$, $V_{out} = A(V2 - V/2) + V/2$, senão $V_{out} = A(V1 - V/2) + V/2$, limitada entre 0 e V.

OR com entradas V1 e V2:

Se $V1 > V2$, $V_{out} = A(V1 - V/2) + V/2$, senão $V_{out} = A(V2 - V/2) + V/2$, limitada entre 0 e V.

NAND com entradas V1 e V2:

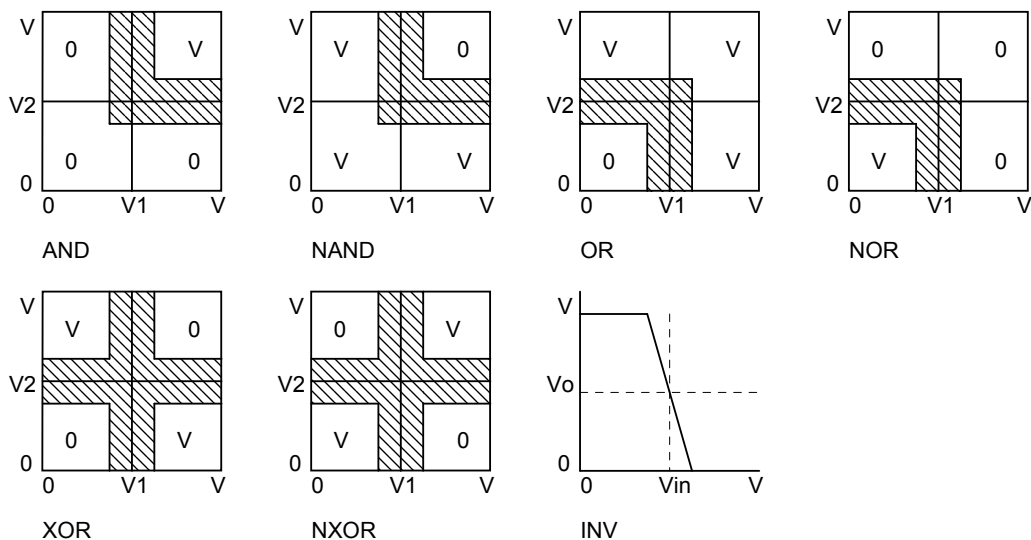
Se $V1 > V2$, $V_{out} = -A(V2 - V/2) + V/2$, senão $V_{out} = -A(V1 - V/2) + V/2$, limitada entre 0 e V.

NOR com entradas V1 e V2:

Se $V1 > V2$, $V_{out} = -A(V1 - V/2) + V/2$, senão $V_{out} = -A(V2 - V/2) + V/2$, limitada entre 0 e V.

Para fazer o XOR e o NXOR, basta combinar as expressões acima adequadamente, separando a região determinada por V1 e V2 em quatro áreas pelas diagonais. Com as portas modeladas assim, é possível modelar circuitos lógicos tipo CMOS, e circuitos analógicos montados a partir de portas lógicas, como osciladores, monoestáveis, etc. Fica faltando o modelamento correto das entradas que deveriam limitar a tensão a pouco acima de V e pouco abaixo de 0, devido aos diodos de proteção que normalmente existem. A limitação poderia ser implementada por resistores não lineares por partes em paralelo com as entradas (não requerido, mas útil).

Notar que estas expressões equivalem a funções lineares por partes de 2 dimensões para as portas de 2 entradas. O fator de ganho como nas fórmulas acima cria transições lineares:



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> BE <passos internos> UIC

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, versão 3.0.

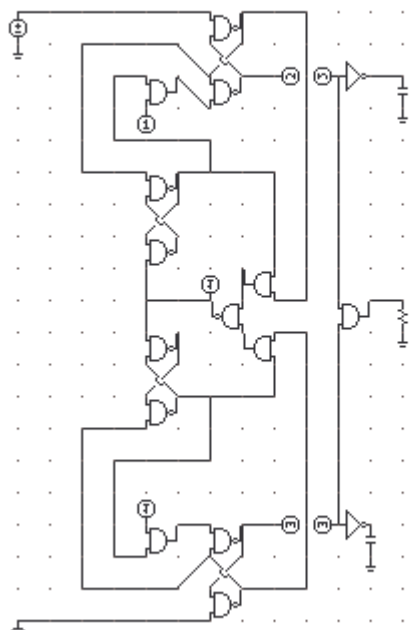
Sugere-se partir do programa exemplo MNA1, que já tem o algoritmo completo da análise nodal modificada, implementar a análise no tempo com as fontes de sinal, capacitores e indutores, e então o método de Newton-Raphson e as portas lógicas.

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.

Exemplo:

O comparador de fase II do integrado CMOS 4046, com saídas filtradas, comparando sinais que diferem em fase e em frequência (exemplo 4046^a do MNAE). As saídas sobre os capacitores primeiramente indicam diferença de fases negativa, mas logo passam a indicar a diferença de frequências positiva.

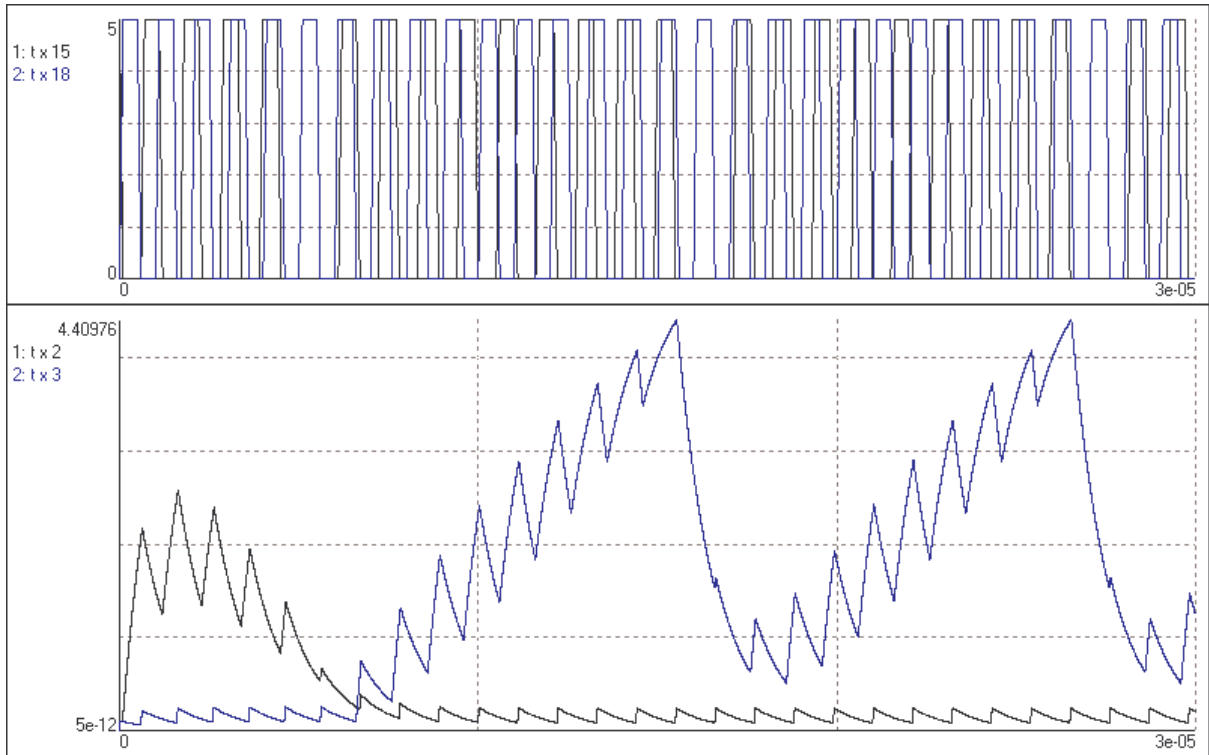
* Comparador de fase II do 4046
 * Comparando fase e frequência com saídas filtradas



```

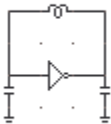
Comparador de fase II do 4046
R0100 1 0 4000
C0200 2 0 1000E-12
C0300 3 0 1000E-12
>0402 4 2 5 1000 10e-12 10
>0503 5 3 5 1000 10e-12 10
)0607 6 7 8 5 1000 10e-12 10
)0907 9 7 10 5 1000 10e-12 10
)0504 5 4 1 5 1000 10e-12 10
)1112 11 12 13 5 1000 10e-12 10
)1106 11 6 14 5 1000 10e-12 10
(0415 4 15 9 5 1000 10e-12 10
(0809 8 9 4 5 1000 10e-12 10
(1609 16 9 7 5 1000 10e-12 10
(0607 6 7 16 5 1000 10e-12 10
(1310 13 10 6 5 1000 10e-12 10
(1106 11 6 17 5 1000 10e-12 10
(1217 12 17 11 5 1000 10e-12 10
(1214 12 14 5 5 1000 10e-12 10
(1805 18 5 12 5 1000 10e-12 10
V1500 15 0 PULSE 0 5 -0.5E-6 0.1E-6 0.1E-6 0.4E-6 1.1E-6 100
V1800 18 0 PULSE 0 5 0 0.1E-6 0.1E-6 0.4E-6 1E-6 100
.TRAN 30E-6 1E-9 BE 1 UIC
  
```

• TRAN 30E-6 1E-9 BE 1 UIC



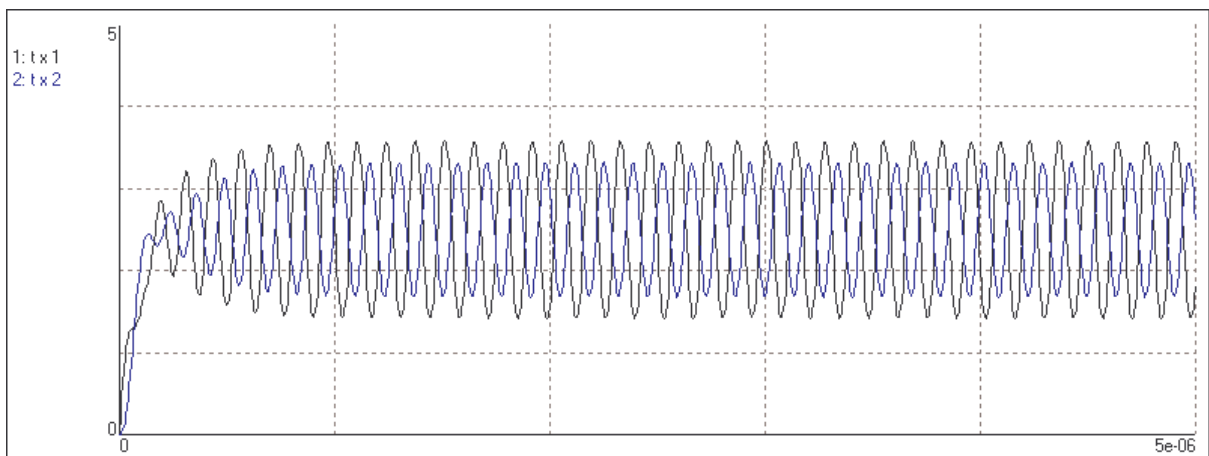
Exemplo: Oscilador Colpitts com inversor.

* Oscilador Colpitts



```
Oscilador Colpitts
L0102 1 2 10E-6
C0200 2 0 100E-12
C0100 1 0 100E-12
>0201 2 1 5 1000 10e-12 10
.TRAN 5E-6 5E-9 BE 1 UIC
```

• TRAN 5E-6 5E-9 BE 1 UIC



Esta análise está certa? Tente mudar o tamanho do passo, ou acrescentar passos internos.