

INTRODUÇÃO AO SPICE E AO PSPICE

1. INTRODUÇÃO

A seguir apresentamos breves esclarecimentos das vantagens da simulação de circuitos eletrônicos através do programa PSPICE, bem como uma pequena revisão histórica, principais características deste, descrição dos tipos de análises disponíveis e declarações de entrada e saída de dados. Também, e principalmente, são mostrados neste trabalho, os circuitos estudados e suas características.

2. SPICE E O PROGRAMA PSPICE

Simuladores de circuitos são poderosas ferramentas de software que permitem a análise de sinais elétricos, sem a necessidade da implementação física dos mesmos. Possibilitam uma análise em geral mais rápida, segura e barata do que a montagem física do circuito.

Particularmente para o projeto de circuitos integrados, a simulação é uma ferramenta fundamental, pois a implementação física do chip é um processo caro e demorado. Um circuito integrado só é fisicamente construído quando todas as simulações elétricas do circuito mostram o resultado desejado.

Os simuladores de circuitos começaram a surgir na década de 60 e inicialmente necessitavam de computadores de grande porte.

Na década de 70 (1975) foi desenvolvido na Universidade de Berkeley (Califórnia) um programa de domínio público de grande capacidade, que hoje em dia é um dos simuladores mais utilizados: **SPICE**: Simulation Program with Integrated Circuit Emphasis. O programa SPICE foi escrito originalmente em linguagem FORTRAN IV, totalizando cerca de 10.000 comandos. Para melhorar a velocidade de processamento, algumas subrotinas de manipulação de matrizes foram codificadas em linguagem ASSEMBLY. A sintaxe de entrada de dados possui razoável grau de liberdade, sendo que o programa efetua nos passos iniciais uma verificação dos dados fornecidos, de modo a assegurar a correta descrição do circuito. Diversos parâmetros do circuito podem ser omitidos, adotando-se neste caso valores-padrões (default) previamente definidos.

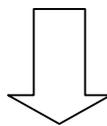
Existem várias versões disponíveis para microcomputadores da linha PC, inclusive uma versão estudante de uso aberto e gratuito. A mais conhecida (e que será usada no

nosso curso) possui o nome **PSPICE**, desenvolvido originalmente pela empresa Microsim e que agora pertence à empresa Cadence / Orcad (<http://www.cadence.com> ou <http://www.orcad.com>), fazendo parte de uma linha de produtos para projetos eletrônicos. Há um site (<http://www.pspice.com>) com diversas informações sobre o Pspice. Outros programas comerciais populares são o Micro-cap (<http://www.spectrum-soft.com>), CircuitMaker (<http://www.microcode.com>). Uma relação mais completa das versões comerciais e gratuitas de Spice para vários sistemas operacionais pode ser vista em <http://www.tu-harburg.de/et1/emv/et1kb/download.html>.

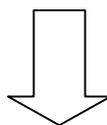
A partir de 1980 tornou-se disponível, a versão SPICE3, cujo programa-fonte pode ser fornecido e codificado em linguagem C para Universidades e Centros de Pesquisa. Uma vez que existem compiladores C para a maioria dos sistemas computacionais, este programa está sendo extensivamente utilizado desde então e tem servido como base de diversos programas comerciais.

3. ESTRUTURA DO SPICE

Arquivo de Descrição do Circuito



Simulador
SPICE



Processador Gráfico
PROBE

A descrição do circuito é realizada através da criação de um arquivo texto o qual tem a função de definir os componentes e suas interligações no circuito. Pode ser composto por fontes de alimentação, resistores, diodos, capacitores, e até mesmo de dispositivos mais complexos como transistores e amplificadores entre outros. Deve ser escrito em um editor de texto e utilizado como entrada do simulador PSPICE.

Algumas versões atuais para *Windows* permitem a geração automática desse arquivo a partir do diagrama esquemático do circuito.

O arquivo criado deve possuir a extensão *.CIR*, como segue o exemplo:

CIRC1.CIR

O simulador SPICE gera um sistema de equações integro-diferenciais a partir da análise nodal ou de malhas do circuito descrito no arquivo e o mesmo resolve este sistema de equações utilizando métodos numéricos.

O arquivo de entrada deve ser compatível com a linguagem do simulador, caso contrário haverá indicação de erro e o circuito não será simulado.

O resultado da simulação é um arquivo de dados que contém as tensões e correntes encontradas nos nós e malhas do circuito respectivamente. Possui o mesmo nome do arquivo de entrada com a extensão *.OUT*, como segue o exemplo:

CIRC1.OUT

Caso seja solicitado no arquivo de entrada, o SPICE gera um arquivo de pontos para ser analisado da forma gráfica "*PROBE*" com extensão *.DAT*. Essa é em geral a melhor forma de analisar os resultados da simulação, que aparecem na tela do computador como seriam vistos no osciloscópio.

O processador gráfico PROBE utiliza o arquivo de dados gerado pelo SPICE para traçar as curvas de tensão e corrente analisadas no circuito. Pode traçar curvas $V \times V$, $V \times I$, $I \times I$, $V \times t$, $V \times f$, entre outros. Os eixos podem ser lineares ou logarítmicos, e suas escalas podem ser alteradas para melhorar a visualização das curvas. Permite a utilização de um cursor que indica a posição exata de cada ponto na curva. O resultado pode ser impresso ou armazenado em arquivo para futura análise.

4. TIPOS DE ANÁLISES DISPONÍVEIS NO SPICE

A simulação de um circuito eletrônico normalmente envolve a combinação de três análises: ponto de operação (análise DC), análise de transientes no domínio do tempo e análise para pequenos sinais AC.

O SPICE pode realizar diversos tipos de análises de circuitos. As mais importantes são:

- Análise DC Não-linear: calcula a curva de transferência DC.
- Análise de transientes Não-linear: calcula a tensão e a corrente em função do tempo quando um sinal grande é aplicado.

- Análise Linear AC: calcula a saída em função da frequência. Um gráfico de bode é gerado.
- Análise de Ruído.
- Análise de Sensibilidade.
- Análise de Distorção.
- Análise de Fourier: calcula e plota o espectro de frequências.
- Análise de Monte Carlo: varia os valores dos componentes segundo uma distribuição estatística.

4.1 CARACTERÍSTICAS DA ANÁLISE DC

A análise das tensões e correntes de um circuito no modo DC é realizada considerando apenas a componente contínua. Fontes AC e indutores são curto-circuitadas, capacitores são circuitos abertos.

É utilizada para o estudo dos seguintes pontos fundamentais:

- Ponto de Operação DC;
- Parametrização linearizada dos Modelos;
- Função de Transferência para pequenos sinais;
- Sensibilidade para pequenos sinais;
- Curvas de Transferência DC.

4.2 CARACTERÍSTICAS DA ANÁLISE AC

Esta é realizada para a observação do circuito no domínio da frequência, onde podemos obter:

- Curva de resposta de filtros que variam com a frequência;
- Análise de ruído e distorção no circuito.

4.3 CARACTERÍSTICAS DA ANÁLISE TRANSIENTE (TRAN)

Esta é realizada para a observar o comportamento do circuito no domínio do tempo. Equivale à análise efetuada com o osciloscópio, por isso a mais utilizada, com o objetivo de obter:

- Resposta de circuitos para sinais alternados ou pulsos;

- Análise de Fourier.

5. DESCRIÇÃO DO CIRCUITO

O texto ou arquivo que descreve o circuito a ser analisado deve ser compatível com a linguagem de entrada do simulador. O SPICE possui linguagem própria, por isso a descrição deve ser feita seguindo algumas exigências que devem ser observadas com cuidado, para que a simulação do circuito a ser analisado seja realizada com sucesso e confiabilidade. São elas:

- Cada nó do circuito deve ser descrito por um número e cada componente por um nome;
- O nó 0 (zero) é sempre a referência ou terra;
- Cada nó deve possuir, pelo menos, duas conexões, exceto os nós de substratos dos MOSFETs e as linhas de transmissão sem terminação;
- A primeira letra do componente define o seu tipo;
- A primeira linha é sempre utilizada como comentário e pode conter qualquer caracter;
- Comentários adicionais podem ser colocados em qualquer parte do arquivo utilizando um asterisco (*) no início da linha e um ponto e vírgula (;) após uma linha do circuito;
- O arquivo deve ser finalizado com “.END”;
- Caracteres maiúsculos ou minúsculos são equivalentes.

Além das recomendações acima, um arquivo de entrada ou *arquivo de circuito* do SPICE contém declarações de título e comentários, declarações de dados, declarações de controle da solução, declarações de controle da saída e uma declaração de fim, os quais são definidos a seguir.

5.1 DECLARAÇÕES DE TÍTULO E COMENTÁRIOS

Uma *declaração de título* é a primeira declaração do arquivo de circuito. Pode conter qualquer texto, mas está restrita a uma linha. Essa declaração é normalmente usada para identificar o circuito em estudo. Um exemplo pode ser

Análise CC do circuito 6.2

Fatores de escala que estão disponíveis para uso com estes números são

$$\begin{array}{lll} F = 10^{-15} & U = 10^{-6} & \text{MEG} = 10^6 \\ P = 10^{-12} & M = 10^{-3} & G = 10^9 \\ N = 10^{-9} & K = 10^3 & T = 10^{12} \end{array}$$

Dessa forma, os seguintes valores são todos equivalentes:

$$1,05E6 \quad 1,05\text{MEG} \quad 1,05E3K \quad 0,00105G$$

As declarações de dados para os vários elementos dos circuitos têm as seguintes formas:

1. Resistor

R <NOME> <(+) NÓ> <(-) NÓ> <VALOR>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres, para designar o elemento.

(+) NÓ e (-) NÓ definem a polaridade da conexão do resistor. Correntes positivas circulam do NÓ (+), através do resistor, ao NO (-).

<VALOR> é o valor não-nulo da resistência (positivo ou negativo), em ohms.

Exemplo RLOAD 12 3 10k

representa um resistor de 10k Ohms, denominado RLOAD, conectado entre os, nós 12 e 3, com corrente circulando do nó positivo 12, através do resistor, para o nó negativo 3.

2. Capacitor

C <NOME> <(+) NÓ > <(-) NÓ> <VALOR> [IC <VALOR INICIAL>]

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+) NÓ e (-) NÓ definem a polaridade da conexão do capacitor. Correntes positivas circulam do NÓ (+), através do capacitor, ao NÓ (-).

<VALOR> é o valor não-nulo (positivo ou negativo), em farads.

<VALOR INICIAL> é opcional e exprime a tensão inicial no capacitor no tempo $t=0$, do NÓ (+) com relação ao NÓ (-), para uma análise da resposta transitória.

Exemplo: CEXT 2 3 10U
 CEXT 2 3 10U IC=4

representa um capacitor de 10uF, denominado EXT, conectado entre os nós 2 e 3. Se IC = 4 for incluído, então existe uma tensão inicial de 4 V no nó 2 com relação ao nó 3, em t = 0. Esse valor é usado apenas na resposta transitória.

OBS: Um capacitor de 225 picofarad pode ser expresso como:
225P, 225p, 225pF; 225pFarad; 225E-12; 0.225N, etc.

3. **Indutor**

L (NOME) <(+) NÓ > <(-) NÓ> <VALOR> [IC = <VALOR INICIAL>]

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+) NÓ e (-) NÓ definem a polaridade da conexão do indutor.

Correntes positivas circulam do NÓ (+), através do indutor, para o NÓ (-).

<VALOR> é o valor diferente de zero (positivo ou negativo), em henrys.

<VALOR INICIAL> é opcional e exprime a corrente inicial no indutor no tempo t = 0, do NÓ (+), através do indutor, ao NÓ (-), para uma análise da resposta transitória.

Exemplo: L12 100 0 10M
 L12 100 0 10M IC=-0,5

representa um indutor de 10mH, denominado 12, conectado entre os nós 100 e 0 (nó de referência). Se IC =0,5 for incluído, então existe uma corrente de -0,5 a partir do nó 100, através do indutor, para o nó 0, no tempo t = 0. Esse valor é usado apenas na resposta transitória.

4. **Transformador linear**

k<NOME> L <INDUTOR A> L <INDUTOR B> <VALOR DO ACOPLAMENTO>

K<NOME> acopla dois indutores A e B usando uma convenção de ponto que é determinada pelas atribuições dos nós dos indutores A e B. A polaridade é determinada pela ordem dos nós nos dispositivos L e não pela ordem dos indutores na declaração K. Os terminais com ponto dos indutores A e B são aqueles ligados aos nós positivos (primeiramente definido), nas suas declarações de definição.

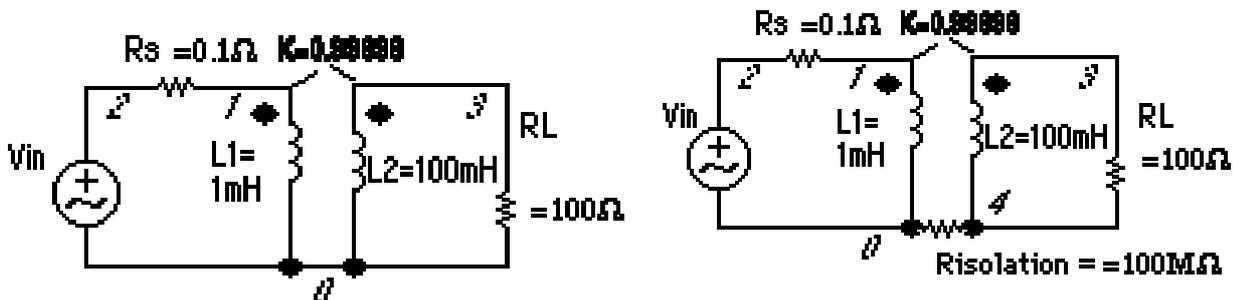
K<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

<VALOR DO ACOPLAMENTO> é o coeficiente de acoplamento mútuo, na faixa de 0 a 1.

Exemplo: LPRI 2 3 500M
LSEC 5 4 400M
KXFRM LPRI LSEC 0.98

representa um transformador linear, tendo acoplamento mútuo de 0,98 entre LPRI e LSEC. Os terminais com ponto, para a polaridade do acoplamento, são o terminal de LPRI conectado ao nó 2 e o terminal de LSEC conectado ao nó 5.

OBS: O SPICE não tem um modelo para um transformador ideal. Um transformador ideal pode ser simulado usando indutâncias mútuas tais que a razão de transformação $N1/N2 = \sqrt{L1/L2}$ e fazendo o fator de acoplamento K próximo a 1 (ex. $K=0.99999$) e escolhendo L tal que $\omega L \gg$ a resistência vista pelo indutor. O circuito do secundário necessita de um caminho DC para terra. Isto pode ser conseguido pela adição de um resistor grande para terra ou dando um nó comum entre o primário e o secundário.



5. Fontes independentes de corrente e tensão

I <NOME> <(+) NÓ> <(-) NÓ> [TYPE <VALOR>] [TRANSIENT SPEC.]

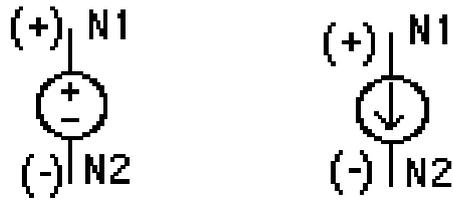
V <NOME> <(+) NÓ> <(-) NÓ> [TYPE <VALOR>] [TRANSIENT SPEC.]

I designa uma fonte independente de corrente.

V designa uma fonte independente de tensão.

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

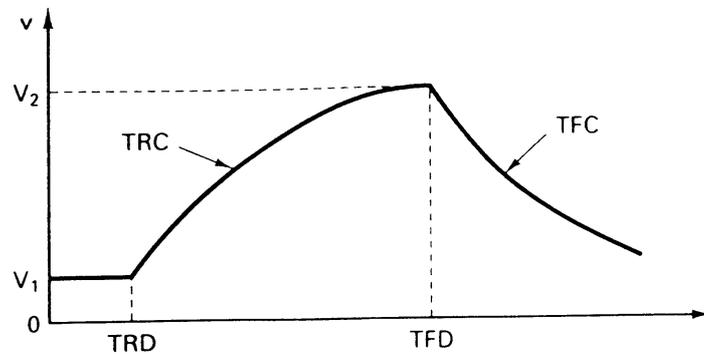
(+) NÓ e (-)NÓ definem a polaridade da fonte. Correntes positivas fluem do NÓ (+), através da fonte, para o NÓ (-). **CUIDADO COM ESSA CONVENÇÃO DE NÓS !!**



TYPE é DC (default) para uma fonte de corrente contínua, AC para uma fonte senoidal em corrente alternada.

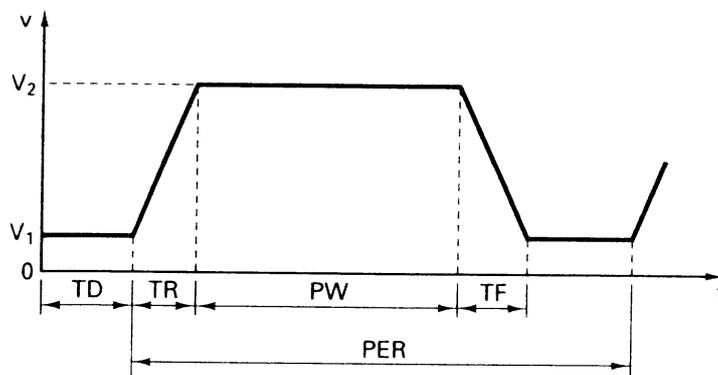
<VALOR> é um valor cc para DC ou um módulo e uma fase (em graus) para AC. Os valores default são zero.

[TRANSIENT SPEC.] é usado apenas na análise de transientes e pode ser um dos seguintes:



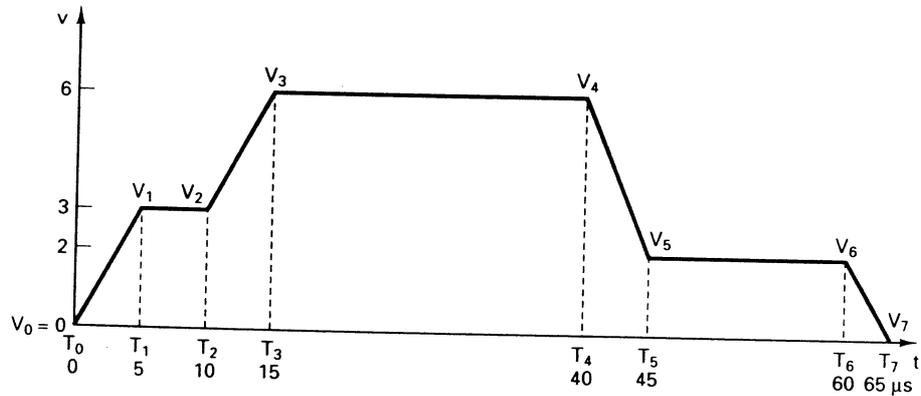
$$EXP(<V1> <V2> <TRD> <TRC> <TFD> <TFC>)$$

A forma EXP faz com que a corrente ou tensão de saída seja <V1> para os primeiros <TRD> segundos. Então, a saída muda exponencialmente de <V1> até <V2>, com uma constante de tempo <TRC>, em <TFD> segundos. Então, a saída decai de <V2> de volta a <V1>, com uma constante de tempo de <TFC>.



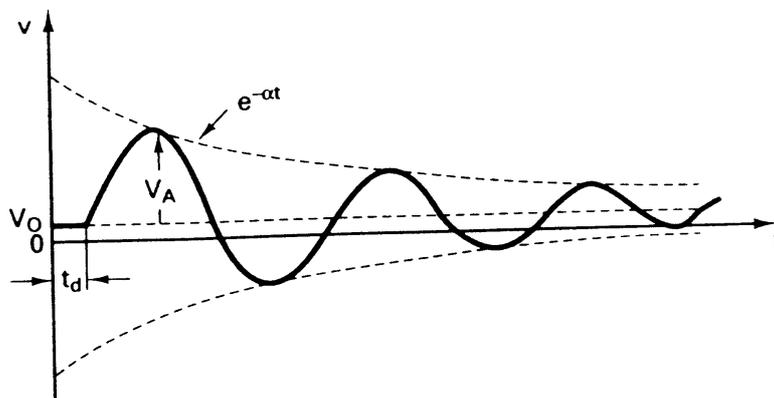
$$PULSE(<V1> <V2> <TD> <TR> <TF> <PW> <PER>)$$

A forma PULSE faz a saída começar em <V1> e permanecer por <TD> segundos. Então, a saída vai linearmente de <V1> a <V2> durante os <TR> segundos a seguir. A saída permanece então em <V2> por <PW> segundos. Retorna então, linearmente, para <V1> durante os <TF> segundos a seguir. Permanece em <V1> por <PER> - (<TR> + <PW>+ <TF>) segundos e o ciclo se repete, excluindo o período inicial e <TD> segundos.



$$PWL(<T1> <V1> <T2> <V2> \dots <TN> <VN>)$$

A forma PWL descreve uma forma de onda de composição linear (ponto a ponto). Cada par de valores tempo-saída especifica uma inflexão da forma de onda. A saída entre esses pontos de inflexão é uma interpolação linear da corrente nesses pontos.



$$SIN (<VO> <VA> <FREQ> <TD> <ALP> <THETA>)$$

A forma SIN faz com que a saída comece em <VO> e permaneça por <TD> segundos. Então, a saída torna-se uma onda senoidal exponencialmente amortecida descrita pela equação

$$VO + VA \cdot \sin\{2\pi [FREQ \cdot (T-TD) - fase/360]\} e^{-(T-td)-\alpha}$$

Exemplos: IG1 2 3 0.2A ou IG1 2 3 DC 0.2A

representa uma fonte de corrente cc, chamada de G1, fornecendo 0,2 A do nó 2 para o nó 3, através da fonte.

ISOURCE 0 5 AC 10 64

representa uma fonte ca $10 \angle 64$ A, chamada de SOURCE, conectada entre os nós 0 e 5, com terminal positivo no nó 0.

12 0 2 EXP(10M 0M 0 0.11)

representa uma corrente exponencial de $10e^{-10t}$ mA, circulando do nó 0 para o 2, através da fonte, no intervalo $0 < t < 0,1$ segundos, para análise transitória.

6. Fonte de tensão controlada por tensão (FTCT)

E <NOME> <(+) NÓ> <(-) NÓ> <(+ CONTROLE) NÓ> <(- CONTROLE) NÓ> <GANHO>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+)NÓ e (-) NÓ definem a polaridade da fonte. Correntes positivas circulam do NÓ (+), através da fonte para o NÓ (-).

(+ CONTROLE) e (- CONTROLE) são em pares e definem um conjunto de tensões de controle que são multiplicadas por <GANHO>. Um nó em particular pode aparecer mais de uma vez e os nós de saída e de controle não precisam ser diferentes.

Exemplo: EBUFF 1 3 11 9 2.4

representa uma FTCT, chamada de BUFF, conectada entre os nós 1 e 3, com tensão no nó 1 com relação ao nó 3 igual a 2,4 vezes a tensão do nó 11 com relação ao nó 9.

7. Fonte de corrente controlada por corrente (FCCC)

F <NOME> <(+) NÓ> <(-) NÓ> <(DISP. DE CONTROLE V) NOME> + <GANHO>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+) NÓ e (-) NÓ definem a polaridade da fonte. Correntes positivas circulam do NÓ (+), através da fonte, para o NÓ (-). A corrente através <(DISP. DE CONTROLE V) NOME>, multiplicada pelo <GANHO>, determina a corrente de saída.

(DISP. DE CONTROLE V) é uma fonte independente de tensão, com uma tensão nos terminais que não pode ser zero.

Exemplo: F23 3 7 VOUT 1.2

representa uma FCCC, chamada de 23, com corrente circulando do nó 3, através da fonte, para o nó 7, tendo um valor igual a 1,2 vezes o da corrente que circula na fonte independente de tensão VOUT (do terminal – para o terminal +).

8. Fonte de corrente controlada por tensão (FCCT)

G<NOME> <(+) NÓ> <(-) NÓ> <(+ CONTROLE)> NÓ> <(- CONTROLE) NÓ>
<TRANSCOND>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+) NÓ e (-) NÓ definem a polaridade da fonte. Correntes positivas circulam do NÓ (+), através da fonte, para o NÓ (-).

(+ CONTROLE) e (- CONTROLE) são em pares e definem um conjunto de tensões de controle que são multiplicadas por <TRANSCOND> (transcondutância). Um nó em particular pode aparecer mais de uma vez e os nós de saída e de controle não precisam ser diferentes.

Exemplo: GAMP 4 3 1 9 1.7

representa uma FCCT, chamada de AMP, conectada entre os nós 4 e 3, com corrente circulando do nó 4 para o nó 3, através da fonte, igual a 1,7 vezes a tensão do nó 1 com relação ao nó 9.

9. Fonte de tensão controlada por corrente (FTCC)

H<NOME> <(+) NÓ> <(-) NÓ> <(DISP. DE CONTROLE V) NOME> <TRANSRESIST>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

(+) NÓ e (-) NÓ definem a polaridade da fonte. Correntes positivas circulam do NÓ (+), através da fonte, para o NÓ (-). A corrente através de <(DISP. DE CONTROLE V) NOME>, multiplicada por <TRANSRESIST> (transresistência), determina a tensão de saída.

(DISP. DE CONTROLE V) é uma fonte de tensão independente, com uma tensão dos terminais que deve ser diferente de zero.

Exemplo: HIN 4 8 VDUMMY 7.7

representa uma FTCC, chamada de IN, com corrente circulando do nó 4, através da fonte, para o nó 8, com valor igual a 7,7 vezes o da corrente circulando na fonte independente de tensão VDUMMY (do terminal - para o terminal +).

10. Declaração de chamada de subcircuito

X<NOME> [NÓ]* <(SUBCIRCUITO) NOME>

<NOME> exprime qualquer cadeia alfanumérica de sete caracteres.

[NÓ]* exprime uma lista de nós necessários para a definição do subcircuito.

<(SUBCIRCUITO) NOME> é o nome da definição do subcircuito (veja a declaração SUBCKT, abaixo). Deve existir na chamada o mesmo número de nós que existe na definição de subcircuito.

Esta declaração faz com que o subcircuito referenciado seja introduzido no circuito, com os nós dados substituindo os nós de argumento na definição. Permite a definição de um bloco de circuito uma vez e, então, o uso desse bloco em vários lugares.

Exemplo: XBUFF 4 1 7 9 UNITAMP

representa a chamada de um subcircuito que substitui a declaração de chamada pelo conteúdo do arquivo UNITAMP, para análise com o SPICE.

11. Dispositivos Semicondutores

Os modelos de dispositivos semicondutores necessitam de muitos parâmetros. Um conjunto de parâmetros do modelo de um dispositivo é definido em uma declaração de modelo .MODEL separada e a ela é atribuído um nome único. Portanto, um dispositivo semicondutor é especificado por duas linhas de comando: um elemento e uma declaração .MODEL.

A sintaxe é:

.MODEL NomeMOD Tipo (valores dos parâmetros)

NomeMOD é o nome do modelo para o dispositivo.

Tipo se refere ao tipo de dispositivo e pode ser:

D: Diodo

NJF: JFET canal N

NPN: transistor bipolar npn

PJF: JFET canal P

PNP: transistor bipolar pnp

PMOS: transistor pmos

NMOS: transistor nmos

Os valores dos parâmetros especificam as características do dispositivo como explicado a seguir:

11.1. Diodo

Linha do Elemento: Dnome N+ N- NomeMOD AREA

Linha do Modelo: .MODEL NomeMOD D (IS= N= Rs= CJO= Tt= BV= IBV=)

O nome do elemento começa com D para indicar que é um diodo, N+ e N- são os números dos nós e NomeMOD é o nome do modelo do diodo que é especificado na linha do modelo. AREA é um fator de escala opcional que indica o número de diodos do tipo em questão que devem ser conectados para formar o tipo desejado.

Os valores especificam: a corrente de saturação IS (default=1E-14A), o coeficiente de emissão N (=1), a resistência série RS (=0 ohm), a capacitância de junção CJO (=0F), tempo de transição TT (=0sec), tensão de ruptura reversa (breakdown voltage) BV (=infinito) e a corrente reversa de ruptura IBV (=1xE-10A). Para uma descrição do modelo do diodo e de todos os seus parâmetros, consulte a seção 3.10 do livro-texto (Sedra/Smith). Se um parâmetro não for especificado, o valor default (entre parênteses) é usado.

Exemplo: Um diodo comercial 1N4148:

```
.model D1N4148 D (IS=0.1PA, RS=16 CJO=2PF TT=12N BV=100 IBV=0.1PA)
```

11.2. Transistores Bipolares

Elemento: Qnome C B E BJT_Nomemod AREA

Modelo: .MODEL BJT_Nomemod NPN(ou PNP) (BF=val IS=val VAF=val)

BF é o ganho de corrente em emissor comum β , IS é a corrente de saturação e VAF é a tensão Early. Se nenhum valor for especificado, os valores default são usados ($\beta=100$; IS=1E-16A e VAF=[infinito]). Outros parâmetros podem ser especificados, incluindo as capacitâncias de junção CJE (0pF) e CJC (0pF), os tempos de transição TT (0sec) e TR (0sec), a resistência de base RB (0 Ohm), de emissor RE (0 Ohm) e de

coletor RC (0 Ohm). Para uma descrição do modelo do transistor bipolar e de todos os seus parâmetros, consulte a seção 4.16 do livro-texto (Sedra/Smith).

Exemplo: transistor NPN 2N2222A:

```
.model Q2N2222A NPN (IS=14.34F XTI=3 EG=1.11 VAF= 74.03 BF=255.9 NE=1.307  
ISE=14.34F IKF=.2847 XTB=1.5 BR=6.092 NC=2 ISC=0 IKR=0 RC=1 CJC=7.306P  
MJC=.3416 VJC=.75 FC=.5 CJE=22.01P MJE=.377 VJE=.75 TR=46.91N TF=411.1P  
ITF=.6 VTF=1.7 XTF=3 RB=10)
```

11.3. MOSFETS

Elemento: Mnome ND NG NS NB MNomeMod

Modelo: .MODEL MNomeMod NomeMod L= W=

O nome do transistor MOS (Mnome) deve começar com M; ND, NG, NS and NB são os números dos nós dos terminais Dreno, Gate, Fonte (Source) and Substrato (Bulk). NomeMod é o nome do modelo do transistor (NMOS ou PMOS).

L e W é o comprimento e largura do canal (em m).

Os parâmetros acima são normalmente suficientes para especificar transistores discretos. Contudo, para circuitos integrados é necessário especificar vários outros parâmetros relacionados com a geometria do transistor. Para uma descrição do modelo do FET e de todos os seus parâmetros, consulte a seção 5.13 do livro-texto (Sedra/Smith).

A forma geral da declaração .MODEL é:

```
.MODEL ModName NMOS (<LEVEL=val> <keyname=val> ... )
```

```
.MODEL ModName PMOS (<LEVEL=val> <keyname=val> ... )
```

LEVEL se refere ao modelo de MOSFET que descreve as características I-V dos transistores. LEVEL 1 é o modelo mais simples e é geralmente suficiente para descrever transistores discretos. Entretanto, para transistores similares de circuitos integrados é preciso usar modelos mais sofisticados. LEVEL 2 inclui modelos de segunda ordem, enquanto LEVEL 3 é um modelo semi-empírico mais adequado para transistores de pequeno canal.

11.4. JFETS

Elemento: Jname ND NG NS NomeMod

ND, NG e NS são os números dos nós dos terminais Dreno, Gate e Fonte.

NomeMod é o nome do modelo do transistor.

Modelo: .MODEL NomeMod NJF (parâmetro=)

.MODEL NomeMod PJF (parâmetro=)

para o JFET N e o JFET P.

Uma lista parcial dos parâmetros: VT0 (tensão de limiar, -2V default para JFET N), BETA (coeficiente de transcondutância, $1E-4 \text{ A/V}^2$), LAMBDA (modulação de comprimento do canal, 0 V^{-1}), IS (corrente de saturação, $1E-14 \text{ A}$), CGD e CGS (capacitâncias de junção gate-dreno e gate-fonte, 0 pf), RD e RS (resistência ôhmicas do dreno e fonte, 0 Ohm). Para uma descrição do modelo do FET e de todos os seus parâmetros, consulte a seção 5.13 do livro-texto (Sedra/Smith).

5.3 DECLARAÇÕES DE CONTROLE DA SOLUÇÃO

Os comandos a serem estudados nessa seção são as declarações .AC, .DC, .FOUR, .IC, .LIB, .SUBCKT, .TF e .TRAN. As declarações de controle da saída .PLOT e .PRINT, descritas na próxima seção, são usadas para se obter os resultados das declarações de controle da solução.

1. Análise cc do ponto de operação: A declaração .OP calcula todas as tensões cc de nós, as correntes cc em todos os ramos e as dissipações de potência das fontes. Tem a fórmula

.OP

2. Análise DC: A declaração .DC faz uma análise em varredura em corrente contínua para o circuito, muito usada para traçar a curva característica de componentes. Tem o formato

.DC <(VARIÁVEL VARREDURA) NOME> <(PARTIDA) VALOR><(FINAL) VALOR>
+ <(INCREMENTO) VALOR>

<(VARIÁVEL VARREDURA) NOME> é o nome de uma fonte independente de tensão ou corrente. É percorrida linearmente de <(PARTIDA) VALOR> até <(FINAL) VALOR>. O valor do incremento é <(INCREMENTO) VALOR>. <(PARTIDA) VALOR> pode ser maior ou menor que <(FINAL) VALOR>; ou seja, a varredura pode se processar em qualquer sentido. <(INCREMENTO) VALOR> deve ser superior a zero. A varredura completa pode especificar apenas um ponto, se desejado.

Exemplos: .DC VIN 10V 10V 1V

representa uma solução cc para um circuito com a fonte independente de tensão VIN= 10V.

.DC IGEN 1M 10M 1M

representa uma varredura em cc para a fonte independente de corrente IGEN, variando de 1 a 10 mA, em passos de 1mA.

3. Análise AC: A declaração .AC é usada para calcular a resposta em frequência de um circuito em uma faixa de frequências. Tem a forma

.AC [LIN] ou [OCT] ou [DEC] <(PONTOS) VALOR> <(FREQ. PARTIDA) VALOR>
+ <(FREQ. FINAL) VALOR>

LIN, OCT, ou DEC são palavras-chave que especificam o tipo de varredura, como segue:

LIN: varredura linear. A frequência varia linearmente da freq. partida até a freq. final. <(PONTOS) VALOR> é o número de pontos da varredura.

OCT: varredura em oitavas. A frequência é percorrida logaritmicamente em oitavas. <(PONTOS) VALOR> é o número de pontos por oitava.

DEC. varredura por décadas. A frequência é percorrida logaritmicamente em décadas. <(PONTOS) VALOR> é o número de pontos por décadas.

Apenas um entre LIN, OCT ou DEC deve ser especificado.

<(FREQ. FINAL) VALOR> não deve ser inferior a <(FREQ. PARTIDA) VALOR>, e ambas devem ser superior a zero. A varredura completa pode especificar apenas um ponto, se desejado.

Exemplo: .AC LIN 1 100HZ 100HZ

representa uma solução em regime permanente ca, para uma rede tendo uma frequência de 100 Hz.

```
.AC LIN 101 100KHZ 200KHZ
```

representa uma resposta em frequência linear, tendo 101 pontos distribuídos uniformemente na faixa de 200 kHz.

4. Análise de Fourier: A análise de Fourier produz uma decomposição em componentes de Fourier, como resultado de uma análise transitória. Uma declaração .FOUR requer uma declaração .TRAN (descrita abaixo). Tem o formato

```
.FOUR <(FREQUÊNCIA VALOR) <(VARIÁVEL DE SAÍDA)>*
```

<(VARIÁVEL DE SAÍDA)> é uma lista de uma ou mais variáveis para as quais se deseja as componentes de Fourier. A análise de Fourier é feita a partir dos resultados da análise transitória para uma ou mais variáveis de saída especificadas. A partir destas tensões ou correntes, a componente cc, a frequência fundamental e as harmônicas de ordem 2 a 9 são calculadas. A frequência fundamental é <(FREQUÊNCIA) VALOR>, que especifica o período para a análise. A análise de transiente deve ter, pelo menos, $1/ <(FREQUÊNCIA) VALOR>$ segundo de duração.

Exemplo:

```
.FOUR 10KHZ V(5) V(6, 7) I(VSENS3)
```

calcula as componentes de Fourier para as variáveis V(5), V(6,7) e I(VSENS3). A frequência fundamental para a decomposição é 10kHz.

5. Análise de transiente: A declaração .TRAN faz com que uma análise de transiente seja efetuada para o circuito. Tem o formato

```
.TRAN <(PASSO TEMPO) VALOR> <(TEMPO FINAL) VALOR>  
+ [TEMPO_SEM_IMPRIMIR] [UIC]
```

A análise de transiente calcula o comportamento do circuito no tempo, iniciando-se em $t = 0$ e indo até <(TEMPO FINAL) VALOR>. <(PASSO TEMPO) VALOR> é o intervalo de tempo empregado para plotagem ou impressão dos resultados da análise. [TEMPO_SEM_IMPRIMIR] é opcional e indica o tempo a partir do qual a saída será impressa. A palavra-chave UIC (*use initial conditions*) faz com que o conjunto de condições iniciais para capacitores e indutores com a especificação IC seja usado.

Exemplo: .TRAN 1n 100n

efetua uma análise de transitório no intervalo de 0 a 100 ns, com saída plotada ou impressa a cada intervalo de 1ns.

6. Condições iniciais: A declaração .IC é usada para estabelecer as condições iniciais para uma análise de transiente. Tem o formato

.IC < V(<NÓ>) = <VALOR> >*

Cada <VALOR> é uma tensão que é associada ao <NÓ> para a tensão inicial no nó, no tempo $t = 0$, para a análise de transiente.

Exemplo: .IC V(2)=5 V(5)=-4 V(101)=10

Estabelece as tensões de nó iniciais nos nós 2, 5 e 101 como 5, -4 e 10V, respectivamente, no tempo $t=0$.

7. Arquivo permanente: A declaração .LIB é usada para referenciar um subcircuito de "biblioteca" a outro arquivo. Tem o formato

.LIB <(ARQUIVO) NOME>

<(ARQUIVO) NOME> é o nome do arquivo do subcircuito.

Exemplo: .LIB OPAMP.LIB

representa que o arquivo do subcircuito é chamado OPAMP .LIB.

8. Definição de subcircuito: A declaração .SUBCKT é usada para definir um subcircuito que é chamado usando a declaração X descrita anteriormente. Tem o formato:

.SUBCKT <NOME> [NÓ]*

A declaração .SUBCKT inicia a definição de um subcircuito. A definição termina com uma declaração .ENDS. Todas as declarações entre .SUBCKT e .ENDS são incluídas na definição. Sempre que o subcircuito é chamado por uma declaração X, todas as declarações na definição substituem a declaração de chamada.

9. Função de transferência: A declaração .TF produz a função de transferência cc. Tem o formato

.TF <(VARIÁVEL DE SAIDA)> <(FONTE ENTRADA) NOME>

O ganho de <(VARIÁVEL DE SAIDA)> em relação a <(FONTE ENTRADA) NOME> é calculado junto com as resistências de entrada e saída. <(VARIÁVEL DE SAIDA)> pode ser uma corrente ou uma tensão: contudo, no caso de uma corrente, fica restrito à corrente através de uma fonte de tensão.

Exemplo: .TF V(3) IIN

produz a função de transferência cc para V(3)/IIN, a resistência de entrada vista dos terminais da fonte de corrente independente IIN e a resistência de saída vista no nó V(3).

5.4 DECLARAÇÕES DE CONTROLE DA SAÍDA

As declarações de controle da saída para plotação ou impressão são as declarações .PLOT, .PRINT e .WIDTH.

OBS 1: O SPICE permite que apenas as correntes fluindo através de fontes independentes de tensão sejam observadas. Tais correntes são especificadas na forma I(Vnome), em que Vnome é o nome da fonte de tensão independente através da qual a corrente está fluindo. Se desejarmos observar uma corrente em um ramo no qual não há fonte de tensão, devemos adicionar uma fonte de tensão de valor zero em série com o ramo e medir a corrente que passa nesta fonte.

OBS 2: Uma variável do tipo tensão é especificada como a tensão diferencial entre dois nós, na forma V(nó1,nó2). Quando um dos nós é omitido, assume-se o nó terra (0).

1. Plotagem: A declaração .PLOT permite que resultados de análise cc, ca e transitória tenham saída na forma de plotagens em *impressoras*. Tem o formato

.PLOT [DC] ou [AC] ou [TRAN] [VARIÁVEL DE SAÍDA]*
([<(LIMITE INF.) VALOR>, <(LIMITE SUP.) VALOR>])

DC, AC e TRAN são os tipos de análise que podem ser plotados. Um único tipo de análise deve ser especificado. [VARIÁVEL DE SAÍDA]* é uma lista das variáveis de saída desejadas para plotagem. É permitido um máximo de oito variáveis de saída em uma declaração .PLOT.

A faixa de valores e o incremento no eixo dos x são determinados pelas características da análise que está sendo plotada. A faixa de valores no eixo dos y pode ser estabelecida acrescentando-se (<(LIMITE INF.) VALOR>, <(LIMITE SUP.) VALOR>) às variáveis de saída com a mesma faixa no eixo dos y. Cada ocorrência define um eixo dos y com a faixa especificada. Todas as variáveis de saída que venham entre esta e a próxima faixa de valores à esquerda são colocadas no seu eixo dos y correspondente. Se nenhum limite no eixo dos y for especificado, o programa determina automaticamente os limites de plotagem.

Exemplos: .PLOT DC V(2) V(3, 5) I(R2)

plota a resposta cc para V(2), V(3,5) e I(R2), a corrente em R2.

.PLOT AC VM(3) VP(3) IR(CI) II(C1)

plota o módulo e a fase de V(3) e as componentes reais e imaginárias da corrente através de CI.

.PLOT TRAN V(5) V(2,3) (0,5V) I(R1) I(VCC (-5MA,5MA))

plota a resposta de transitório de V(5) e V(2,3) entre os limites 0 e 5 V e I(R1) e I(VCC) entre os limites -5 e 5 mA.

2. Impressão: A declaração .PRINT permite que resultados de análise ca, cc e transitória saiam em forma de tabelas. Tem o formato:

.PRINT [DC] ou [AC] ou [TRAN] [(VARIÁVEL DE SAÍDA)]*

DC, AC e TRAN definem o tipo de análise que pode ter saída. Um único tipo de análise deve ser especificada. (VARIÁVEL DE SAÍDA) é uma lista das variáveis de saída desejadas. Não há limite para o número de saída. O formato da saída é determinado pela especificação do comando .WIDTH.

Exemplos: .PRINT DC V(1) I(R12)

Imprime os valores cc para V(1) e I(R12).

.PRINTAC VM(1,5) VP(1,5) IR(12) II(L2)

imprime o módulo e a fase de V(1,5) e as componentes reais e imaginárias de I(L2).

```
.PRINT TRAN V(7) I(L4) I(VCC) V(3, 1)
```

imprime a resposta transitória de V(7), I(L4), I(VCC) e V(3,1).

3. Extensão: A declaração .WIDTH estabelece o tamanho da saída. Tem o formato:

```
.WIDTH OUT = <VALOR>
```

<VALOR> é o número de colunas e deve ser ou 80 ou 132. O valor "default" é 80 colunas.

Exemplo: .WIDTH OUT = 132

5.5 DECLARAÇÕES DE FIM

As declarações de fim para arquivos de subcircuito e de circuito, são, respectivamente, .ENDS e .END.

1. Fim de uma definição de subcircuito: A declaração .ENDS marca o final de uma definição de subcircuito (iniciada com a declaração .SUBCKT). Tem o formato:

```
.ENDS [(SUBCIRCUITO) NOME]
```

Constitui boa prática repetir o nome do subcircuito, embora não seja necessário.

2. Fim do circuito: A declaração .END marca o fim do circuito. Ela tem a forma:

```
.END
```

5.6. PROBE

A presença desse comando no arquivo de descrição do circuito faz com que seja gerado em disco um arquivo com extensão .DAT, que contém dados que são utilizados pelo programa gráfico PROBE. As versões mais recentes do programa PSpice geram o arquivo para o programa PROBE automaticamente, sem a necessidade deste comando.

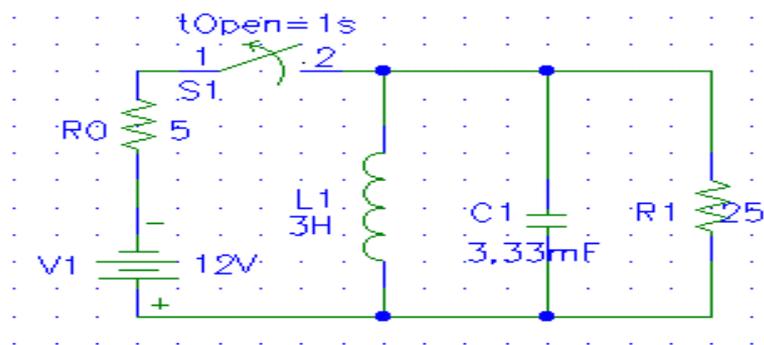
6. RESOLUÇÃO E SIMULAÇÃO DE CIRCUITOS

6.1 RESOLUÇÃO DE CIRCUITOS DE ORDEM ELEVADA

Circuitos contendo um único elemento armazenador de energia é chamado de circuito de primeira ordem, ou seja, são descritos por equações diferenciais de primeira ordem. Já os circuitos de segunda ordem, são aqueles que possuem dois ou mais elementos armazenadores de energia. Estes têm equações representativas que são equações diferenciais de segunda ordem. Assim o SPICE é um programa muito útil na resolução dessas equações.

EXERCÍCIOS:

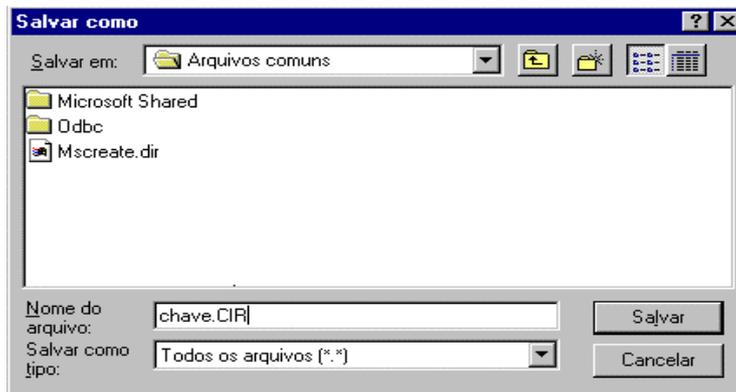
6.1.1 Obter os sinais de corrente no indutor e tensão no capacitor do circuito abaixo.



Tomaremos este circuito como exemplo dos procedimentos à serem seguidos conforme os passos abaixo:

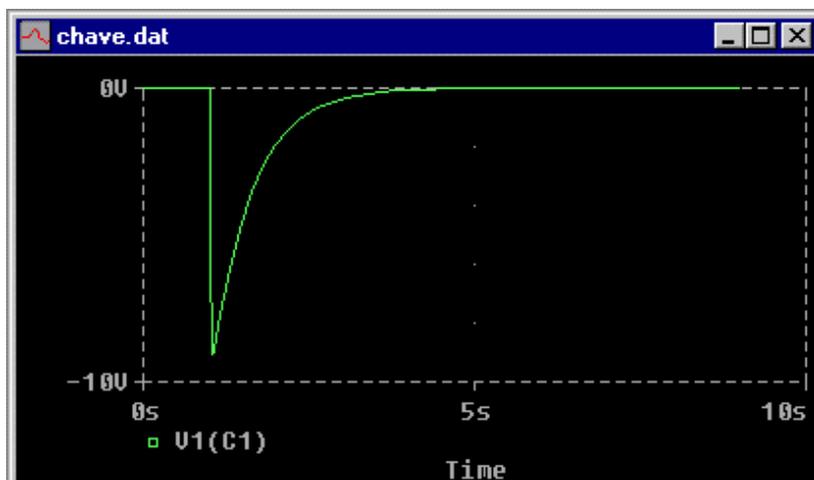
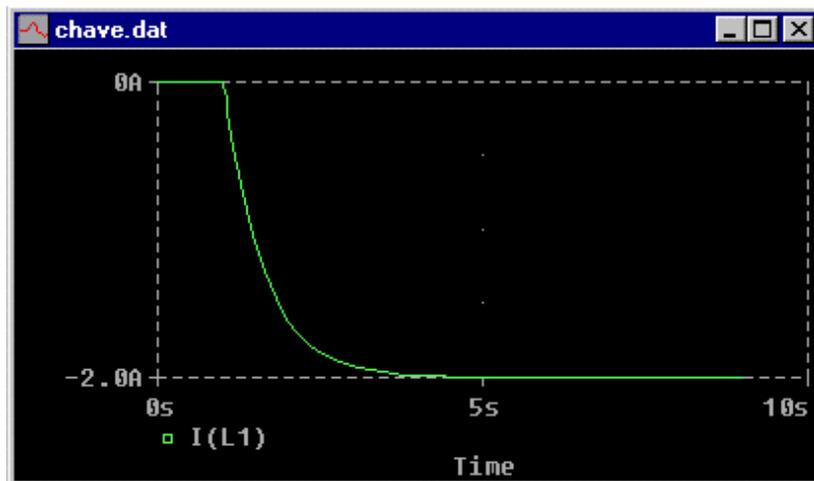
```
Sem nome - Bloco de Notas
Arquivo Editar Pesquisar Ajuda
CIRCUITO-6.1.1
V1 0 1 12
R0 1 2 5
L1 3 0 3
C1 3 0 3.33m
R1 3 0 25
S1 2 3 4 0 CHAVE
US1 4 0 PULSE(0 1 1 2n 2n 9 10)
.MODEL CHAVE USWITCH Ron=1 Roff=1G
.TRAN 5n 9
.PROBE
```

1. Cria-se o arquivo de descrição do circuito, usando o BLOCO DE NOTAS, MS-EDITOR, NOTEPAD, ou qualquer editor de textos disponível;
2. Salva-se o arquivo com a extensão .CIR;

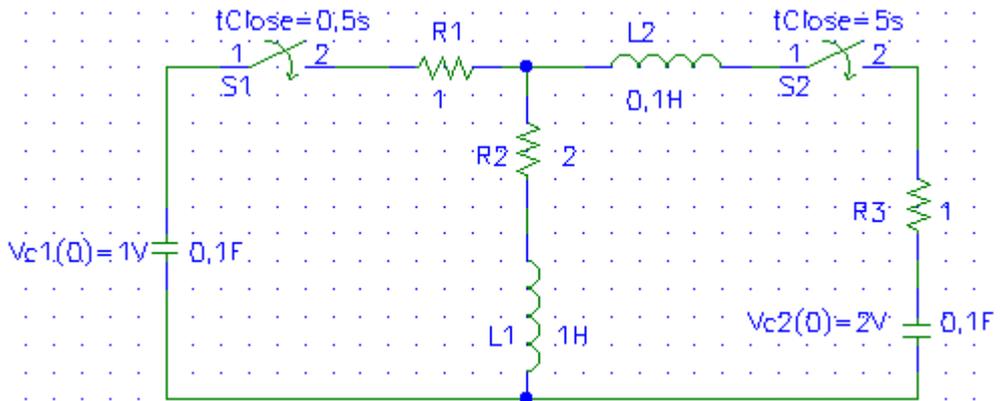


3. Inicia-se o programa SPICE;
4. Abre-se o arquivo a ser simulado (File – Open – Circuit Files (*.cir));
5. Inicia-se a simulação (Simulation – Run);
6. Escolhe-se Trace – Add Trace no programa ou a tecla Insert no teclado, para obter a lista dos elementos a serem analisados;

Para este circuito queremos obter os sinais da corrente no indutor e a tensão no capacitor, assim teremos como resposta:



6.1.2 Obter o sinal $i_{L1}(t)$ e $V_{C1}(t)$.

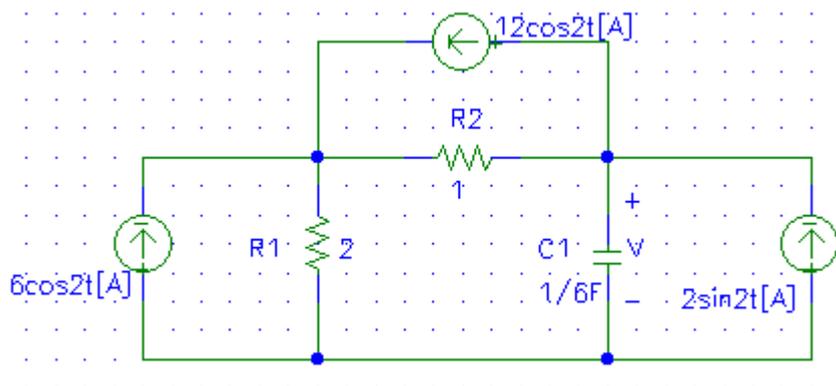


6.2 RESOLUÇÃO DE CIRCUITOS EM REGIME PERMANENTE CA

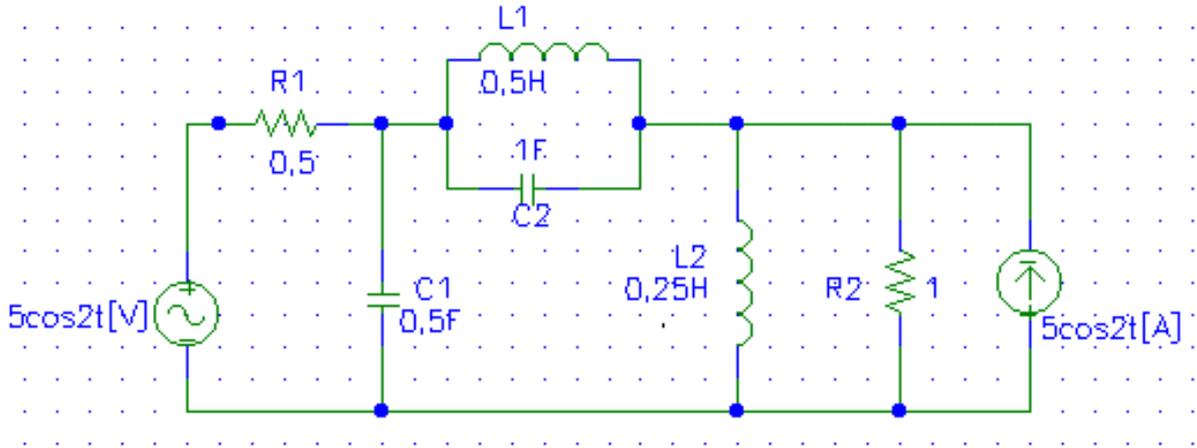
O SPICE é uma ferramenta poderosa na análise de circuitos em regime permanente CA e é particularmente útil no exercício de várias operações tediosas de aritmética complexa associadas à obtenção destas soluções. Utiliza-se a declaração de controle da solução .AC. O SPICE analisa o circuito fasorial para as correntes e tensões fasoriais, resultantes tanto de fontes independentes como de dependentes, que também são expressas como fasores. Visto que o SPICE usa o circuito fasorial no domínio da frequência, todas as fontes CA devem ter a mesma frequência.

EXERCÍCIOS:

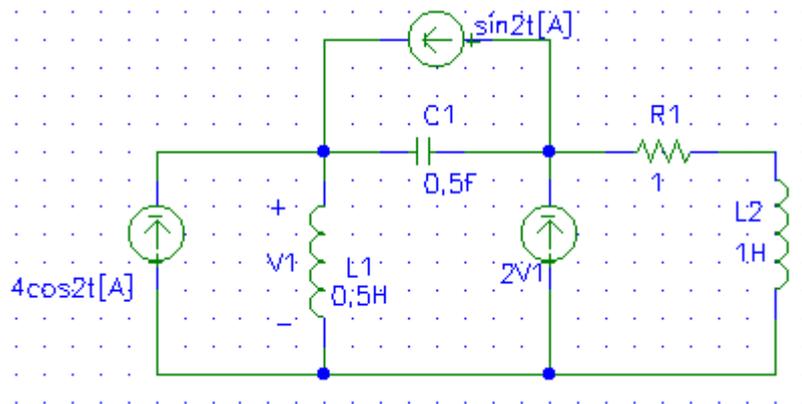
6.2.1 Calcular a tensão no capacitor de 0,5F (C1) na forma polar e a corrente no resistor de 1Ω (R2) na forma retangular.



6.2.2 Use o PSPICE para calcular a representação fasorial de V para $\omega=5$ rad/s.



6.2.3 Calcular a tensão fasorial do indutor de 0,5H para $\omega=6,283$ rad/s.

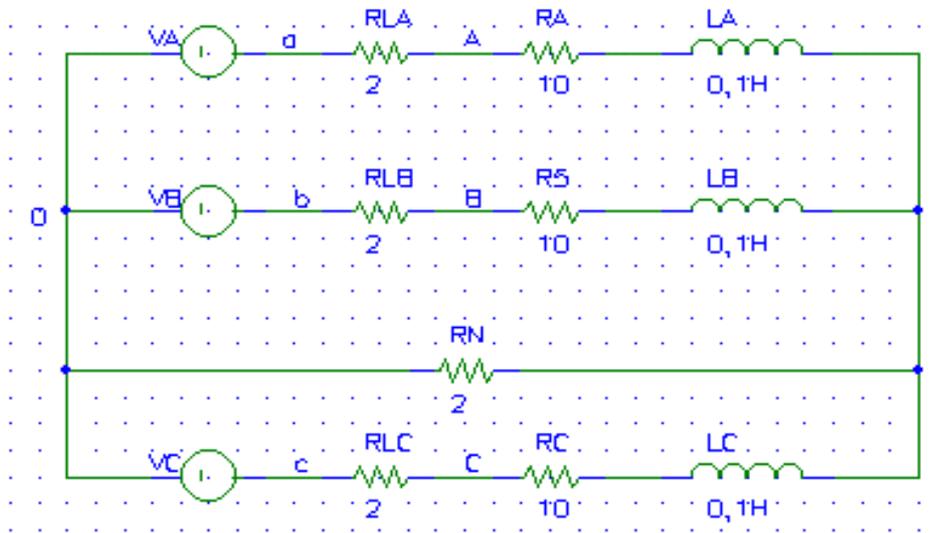


6.3 RESOLUÇÃO DE CIRCUITOS TRIFÁSICOS

Quando as fontes de alimentação de um circuito são geradoras trifásicas, que geram um conjunto equilibrado de tensões, ou seja, três tensões senoidais com a mesma amplitude e frequência, porém defasadas de 120 graus. Desta forma, a fonte trifásica é equivalente a três fontes monofásicas interconectadas, cada uma gerando uma tensão, com diferentes fases. Se as três correntes que saem dos geradores também constituem um conjunto equilibrado, o sistema é dito como um sistema trifásico equilibrado. O uso do SPICE é simples, para circuitos equilibrados ou desequilibrados, quando aplicado ao circuito completo.

EXERCÍCIOS:

6.3.1 Calcular a tensão de linha e a corrente de fase da carga da fase A do sistema da figura abaixo, se a carga de fase C é curto-circuitada (chamado uma falta de fase).



6.3.2 Repetir o exercício anterior, se a linha de neutro de 2Ω for retirada.

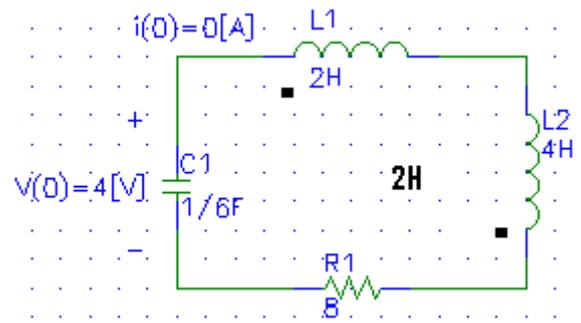
6.4 RESOLUÇÃO DE TRANSFORMADORES

Um sistema de bobinas mutuamente acopladas que são enroladas numa forma, ou núcleo, é normalmente chamado transformador. Transformadores estão disponíveis numa grande variedade de tamanhos e formatos, que são projetados para numerosas aplicações.

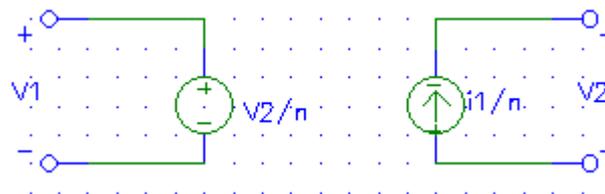
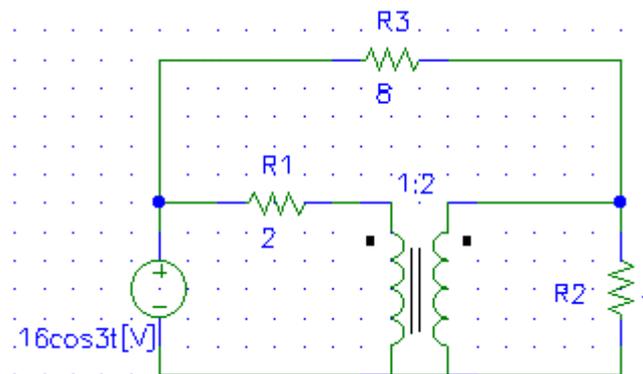
O SPICE pode ser usado para analisar circuitos que contenham transformadores lineares, empregando-se a declaração de dados K (seção 5) para definir o transformador. A declaração K expressa o acoplamento mútuo do transformador.

EXERCÍCIOS:

6.4.1 Usando SPICE, plote i , para $0 < t < 5s$ para o circuito a seguir.

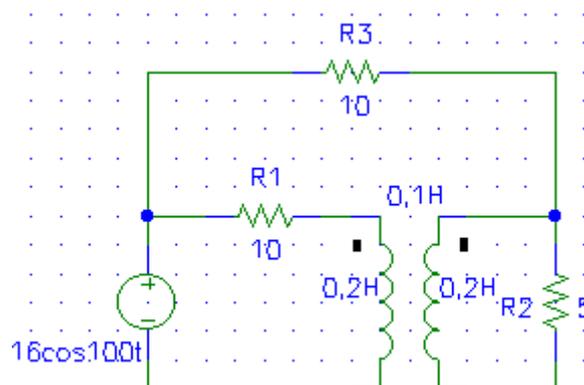


6.4.2 Usando o circuito equivalente indicado abaixo, calcule a tensão no resistor de 8Ω para o circuito a seguir.



(circuito equivalente)

6.4.3 No circuito abaixo, troque o resistor de 10Ω ligado em ponte sobre o transformador por um capacitor de $0,1F$. Use o SPICE para plotar a resposta em frequência do circuito resultante, no intervalo $1 < f < 50$ Hz.



7. PROBLEMAS DE CONVERGÊNCIA

Problemas de convergência podem aparecer quando uma simulação é executada.

Erros comuns:

- Verifique se todas as conexões do circuito são válidas, se as polaridades estão corretas e se existe um caminho DC de qualquer nó para terra.
- Verifique se todos os componentes estão com valores corretos (isto é, MEG em vez de M(mili) para 1E6). Componentes sem um valor atribuído são colocados com valores default determinados pelo simulador.
- Verifique se todos os parâmetros dos modelos são realistas, principalmente se o modelo foi criado ou alterado por você.
- Verifique se todo nó possui duas conexões.
- Verifique se as fontes de tensão ou corrente estão com a sintaxe correta e valores apropriados.
- Use um equivalente série no lugar dos capacitores ou fontes de corrente que estiverem em série.
- Verifique se a letra O não foi usada no lugar de zero (0).
- Verifique se os ganhos das fontes dependentes estão corretos.

Soluções para Convergência DC:

- Coloque `ITL1= 400` na declaração `OPTIONS`. Isto aumentará o número de iterações que o SPICE realizará antes de gerar um aviso de não convergência e abortar a simulação.
- Adicione declarações `.NODESET`. Tensões podem ser atribuídas aos nós no desenho esquemático de mais alto nível. A tentativa inicial feita pelo SPICE para determinar o ponto de operação DC incorpora os `NODESETS`. Isto pode reduzir grandemente o número de iterações necessárias para a convergência, aumentando a probabilidade de uma simulação convergente. Valores impróprios nos `NODESETS` podem resultar em resultados imprecisos. Deve-se tomar muito cuidado com declarações `.NODESET`.
- Use declarações `pulse` para ligar fontes de alimentação DC. Ex: `V1 3 0 5 DC torna-se V1 3 0 PULSE 0 15`. Isto permite ao usuário ligar as fontes de alimentação. Um tempo

de subida (rise time) pode também ser usado para fornecer um comportamento mais realista.

- Coloque Gmin na declaração .OPTIONS. Coloque GMIN=1n ou GMIN=1u. Isto atribui a condutância mínima de todos os dispositivos semicondutores.
- Coloque RSHUNT na declaração .OPTIONS. Esta opção coloca um resistor, com o valor atribuído globalmente por RSHUNT, de todo nó do circuito para terra. Uma solução obtida usando esta técnica de convergência pode ser resultar em um ponto de operação incorreto. Deve ser usado com cautela.
- Coloque ILT6=100 na declaração .OPTIONS. Isto diminui todos os estímulos DC até que um ponto de operação DC seja determinado ou que eles sejam reduzidos para zero volts. As tensões são então gradualmente incrementadas do ponto de operação DC que convergiu (ou de zero volts) até o valor atribuído. O algoritmo usa os aumentos graduais na tensão para estabelecer um novo ponto de operação DC, usando o ponto DC anterior como tentativa inicial. Este processo continua até que um ponto de operação DC tenha sido estabelecido para os valores atribuídos para os estímulos do circuito.

Soluções de Convergência em Transientes:

- Verifique se a convergência DC foi obtida.
- Veja as mensagens de erro para verificar se o problema de convergência pertence exclusivamente à simulação do transiente
- Verifique se o passo de tempo fornece uma resolução apropriada. O passo de tempo deve ser pequeno o suficiente para fornecer uma resolução apropriada para as formas de onda de chaveamento geradas pela simulação. O passo temporal deve ser de ordem de magnitude menor do que o período mais curto na simulação. Por exemplo, em um oscilador de 100KHz o período é 10us. O passo de tempo deve ser de 1us .TRAN 1u 3m. Outros fatores tais como o tempo em ON ou o ciclo útil (duty cycle) devem ser considerados para determinar o passo de tempo. Uma vez que a convergência tenha sido obtida, este valor pode ser maximizado para reduzir o tempo de simulação.
- Para circuitos oscilatórios ou de chaveamento, coloque METHOD=GEAR na declaração .OPTIONS. Isto seleciona o tipo de método de integração que o SPICE usa para resolver as equações de transiente. Deve ser usada para todos os circuitos de

chaveamento. A integração default, trapezoidal, tem a tendência de produzir oscilações.

- Adicione UIC (Use Initial Conditions) na declaração .TRAN. Isto faz com que o SPICE não realize a análise de ponto de operação DC. Como no uso de NODESETS, as condições iniciais podem produzir soluções incorretas. Os resultados devem ser verificados.
- Coloque ITL4=500 na declaração .OPTIONS. Isto aumenta o número de iterações realizadas pelo SPICE antes que um aviso de não-convergência apareça e que a simulação seja abortada.
- Coloque RELTOL=.01 na declaração .OPTIONS. Isto diminui a precisão da simulação incrementando a tolerância de erro relativo necessário para a convergência. Nunca deve ser menor do que .01. O tempo de execução da simulação é também reduzido com o incremento de RELTOL.
- Aumente os tempos de subida e descida de fontes PULSE. Drásticas mudanças na tensão podem resultar em problemas de não-convergência.
- Coloque TRTOL=40 na declaração .OPTIONS. Isto é proporcional ao tamanho do passo usado na simulação de transiente. A precisão da simulação pode ser comprometida pela mudança de TRTOL do valor default de TRTOL=7.
- Reduza a precisão de ABSTOL/VNTOL se os níveis de corrente e/ou tensão permitirem. Os valores (default de ABSTOL=1pA e VNTOL=1uV) devem ser cerca de oito ordens de magnitude abaixo do nível da corrente e tensões máximas.
- Coloque RAMPTIME=10nS na declaração .OPTIONS. Isto faz com que todas as fontes independentes aumentem a partir de zero no início da análise de transiente. Isto é benéfico se a análise de transiente não iniciar. Tome o cuidado de permitir tempo suficiente para as fontes chegarem ao seu valor final para não haver resultados errados.

Soluções de Convergência AC

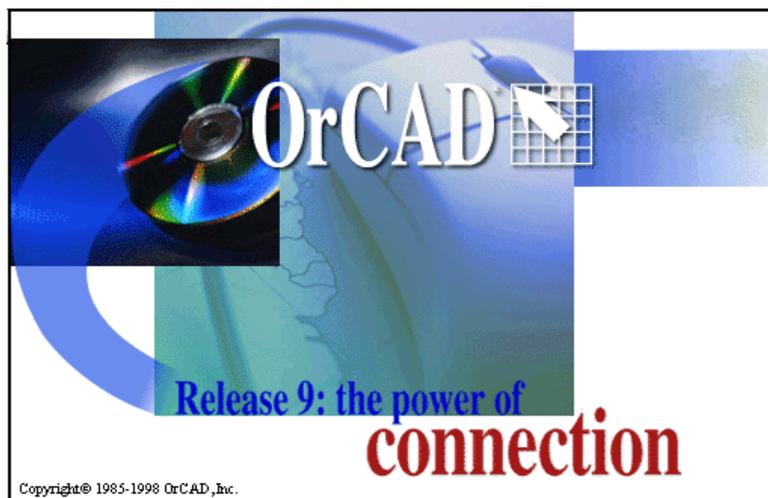
- Não use os passos 3-5 das soluções de convergência DC. Usar estes passos pode não produzir um ponto de operação válido, que é essencial para que o SPICE linearize o circuito. Uma vez que tenha havido a convergência DC, a análise AC também convergirá.

8. USO DE ESQUEMÁTICOS NO PSPICE

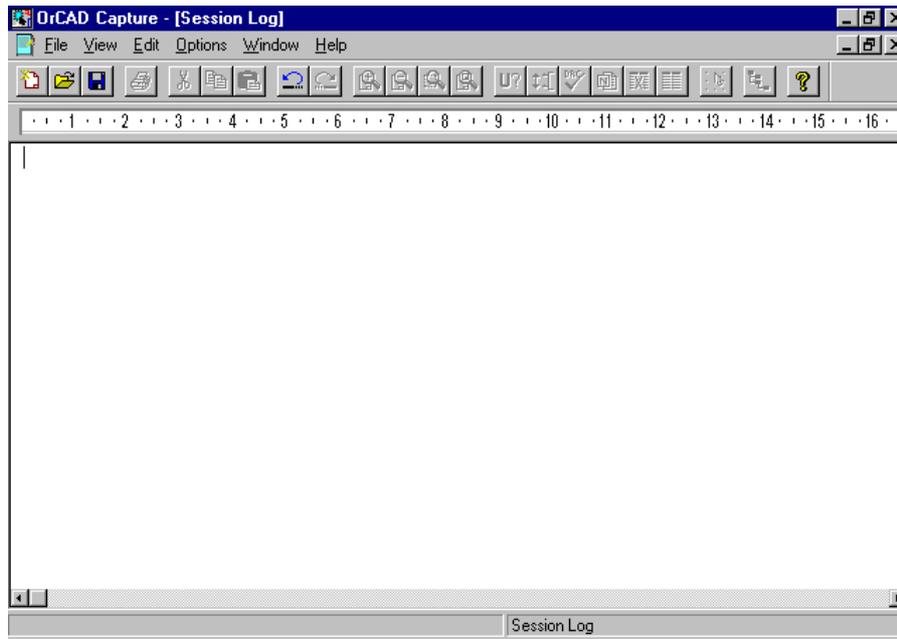
Os modernos programas comerciais de SPICE para Windows, como o PSPICE, permitem a geração da netlist e simulação diretamente a partir do desenho esquemático, sem a necessidade da edição do arquivo .CIR pelo usuário. Isso facilita o uso e permite um projeto de circuitos eletrônicos e simulação deles totalmente feito pelo computador a partir dos desenhos esquemáticos.

A seguir será mostrado um exemplo passo a passo de como simular circuitos a partir do esquemático. Para maiores detalhes de funcionamento, consulte o help do próprio programa e os seus vários manuais, em formato PDF.

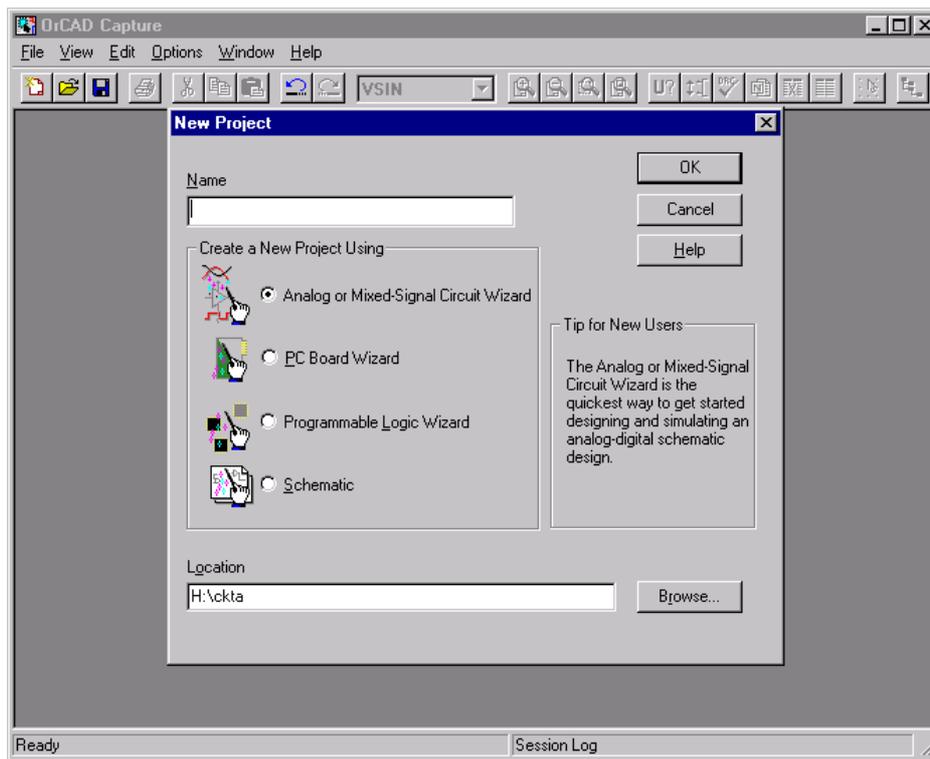
1. No menu de programas do Windows, selecione a opção Pspice Student -> Capture Student. Uma tela de abertura similar a da figura a seguir aparecerá:



2. Quando você abre o programa CAPTURE, esta tela (Session Log) pode aparecer minimizado ou maximizado, como mostra a tela a seguir:

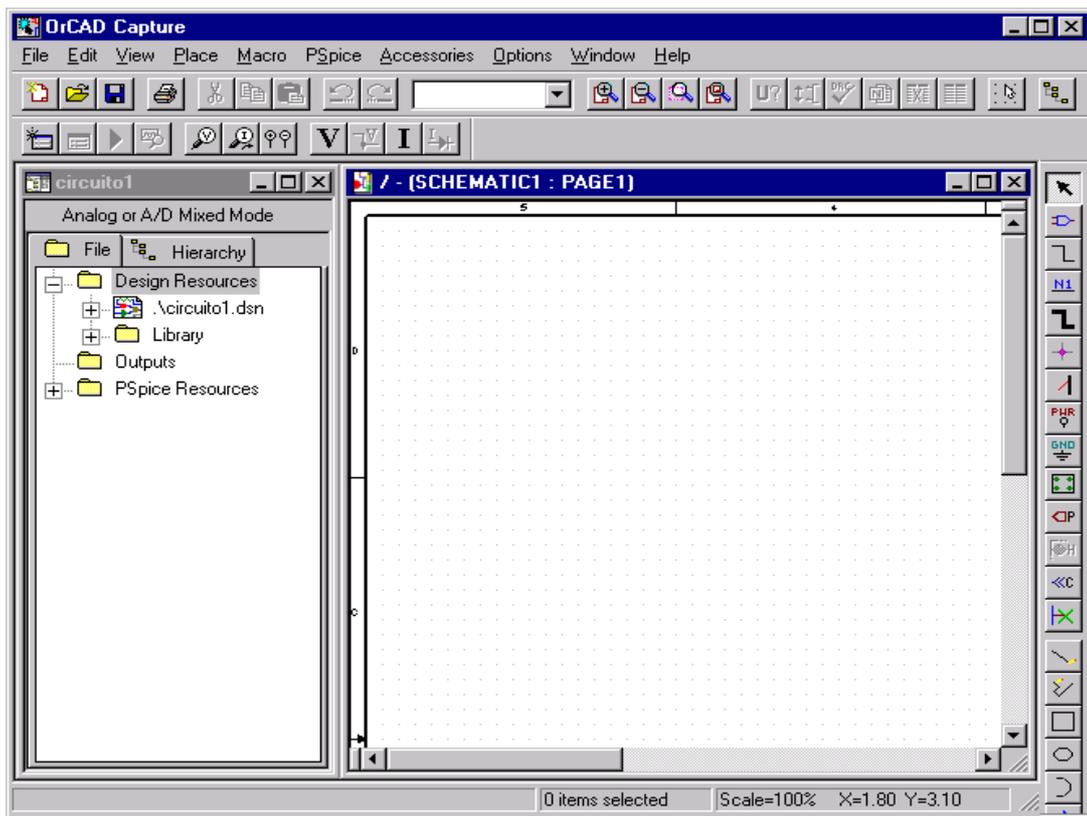


3. Para criar um circuito e simular no SPICE clique em FILE => NEW => PROJECT.

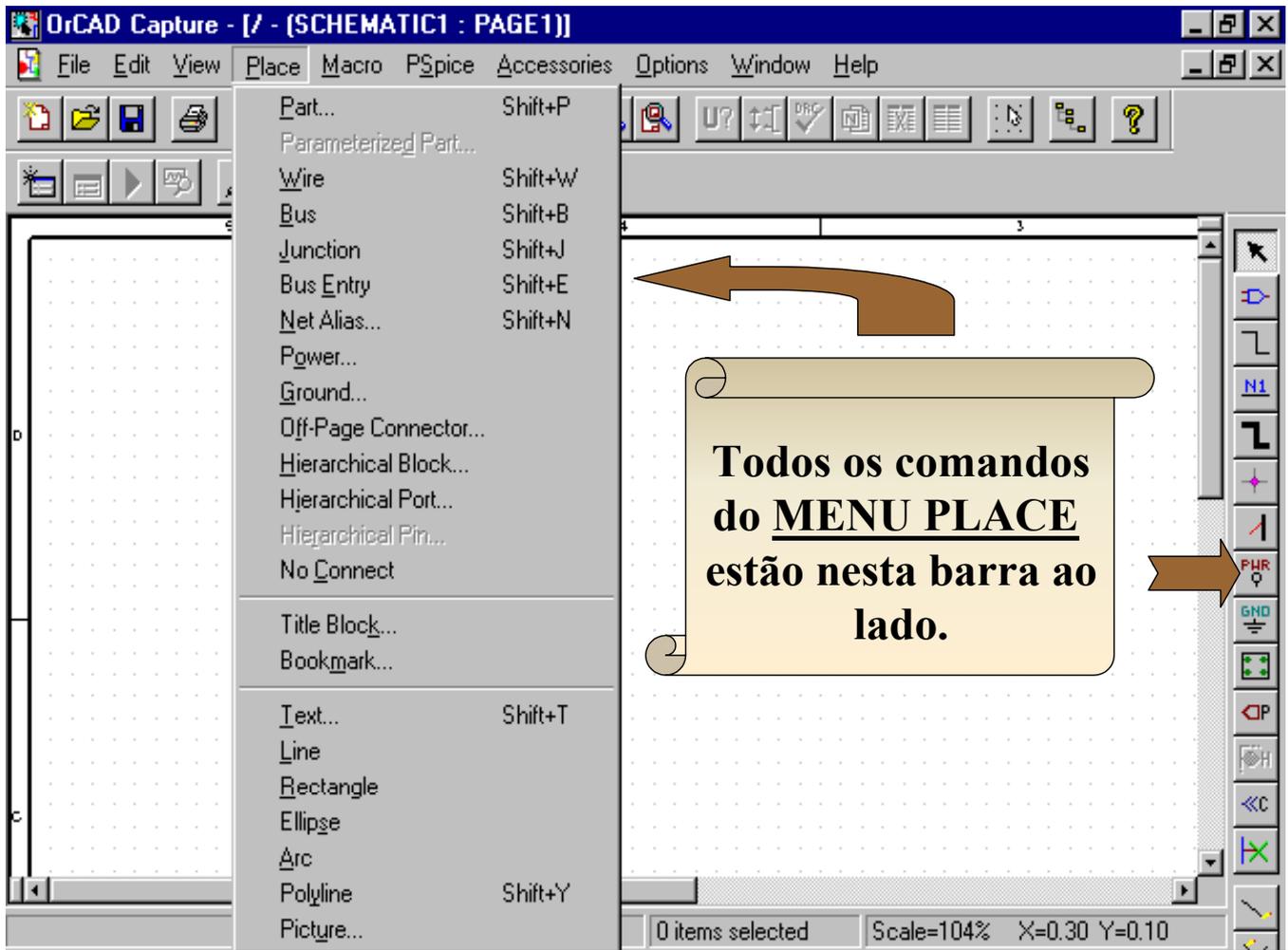


4. O próximo passo é dar um nome a este novo circuito no primeiro campo do formulário (Name), escolher o diretório onde os arquivos serão gravados (Location) e selecionar “Analog or Mixed-Signal Circuit Wizard” para fazer a simulação usando o Pspice. Na próxima tela (não mostrada) escolha “Create a blank project” para criar um projeto em branco. Depois disso aparecerá uma janela pontilhada que

consiste no desenho esquemático em branco e uma janela com uma árvore que contém todos os arquivos relacionados ao projeto (se estiver diferente da figura, use o item Window => Tile Vertically para colocar as janelas lado a lado). Note que, na árvore, há uma pasta SCHEMATIC1 e dentro dela um arquivo PAGE1, que é o correspondente à janela de esquemático vazia. Maximize a janela do esquemático para trabalhar melhor.



5. Inserção dos componentes do circuito:



PART => insere componente

BUS => desenha um barramento

BUS ENTRY => conexão com o barramento

POWER => alimentação

OFF-PAGE CONECTOR => conector <<C

HIERARCHICAL PORT

NO CONNECT => indicar que o pino não é conectado.

FIGURAS => figuras geométricas

WIRE => desenha a conexão

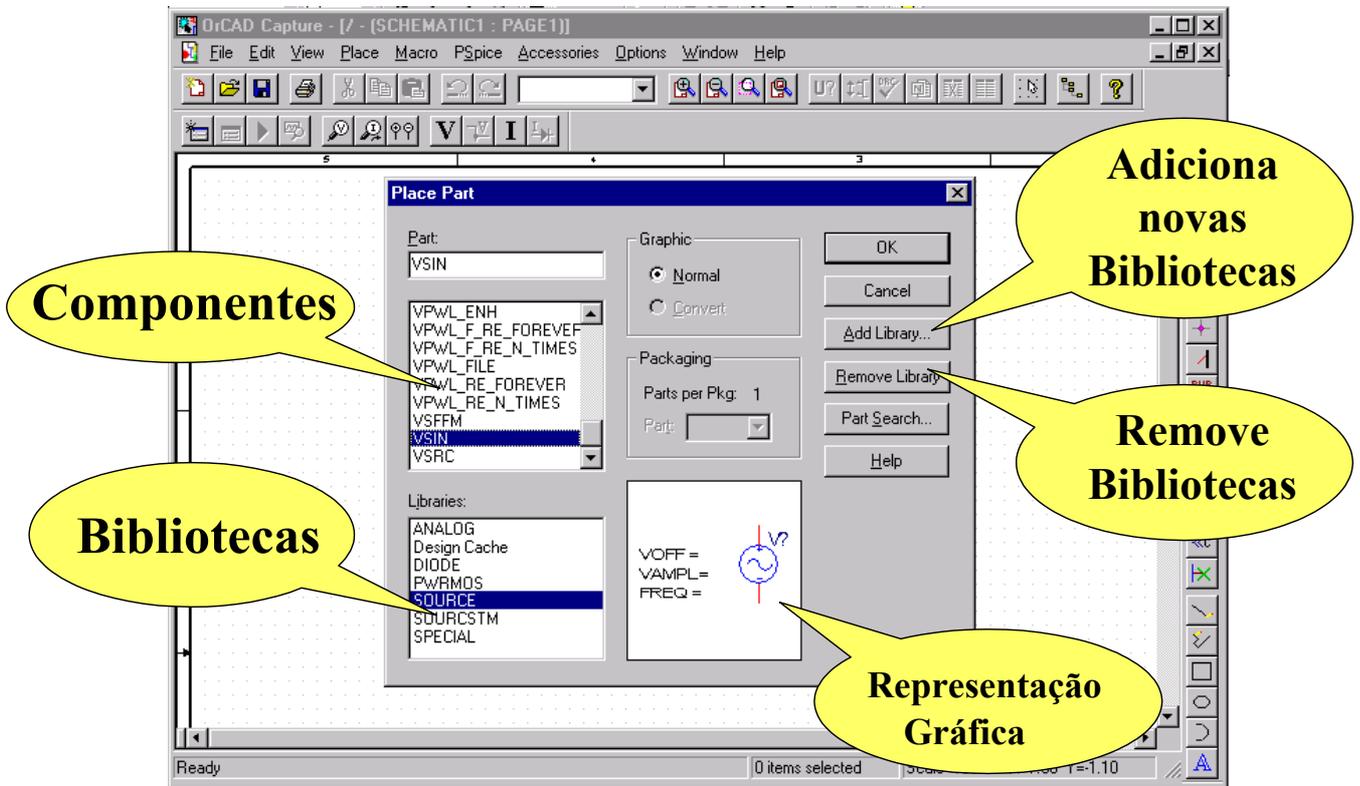
JUNCTION => coloca uma junção

NET ALIAS => labels em nós

GROUND => terra

HIERARCHICAL BLOCK

6. Ao selecionar Place => Part, aparecerá um diálogo mostrado na figura a seguir:

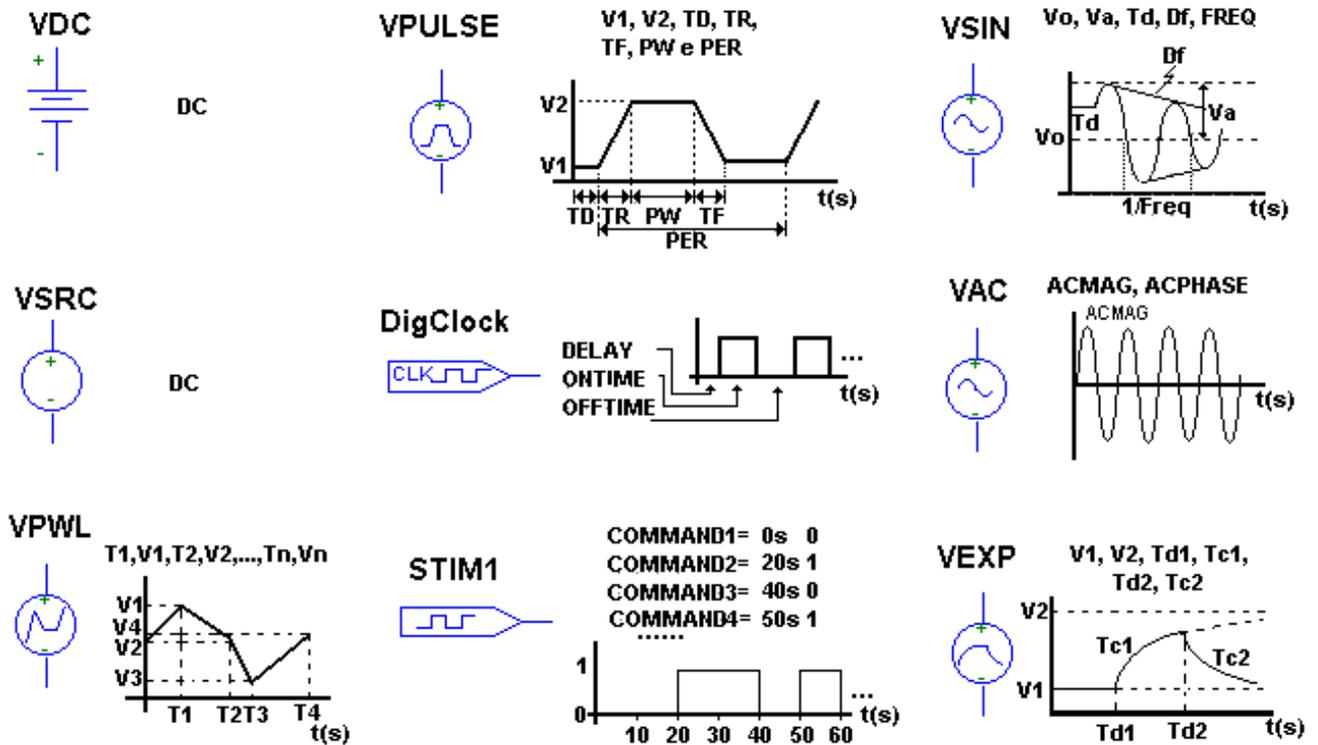


7. As bibliotecas principais de componentes são:

ANALOG => Resistores (R), Capacitores (C), Indutores (L), etc.

SOURCE => Fontes de alimentação (CLOCK, VAC, VDC, VSIN)

A figura a seguir mostra os parâmetros de algumas fontes:



8. Vamos a um exemplo prático:

- 
D1 Diodo =>
D1N4148 Biblioteca - Diode
 Componente - D1N4148

- 
R1 Resistor =>
1k Biblioteca - Analog
 Componente - R

- 
V11 Fonte Senoidal =>
 Biblioteca - Source
 Componente - VSIN

- 
0 Ground =>
 Biblioteca - Source
 Componente - 0/SOURCE

9. Desenhando o circuito:

COMPONENTES

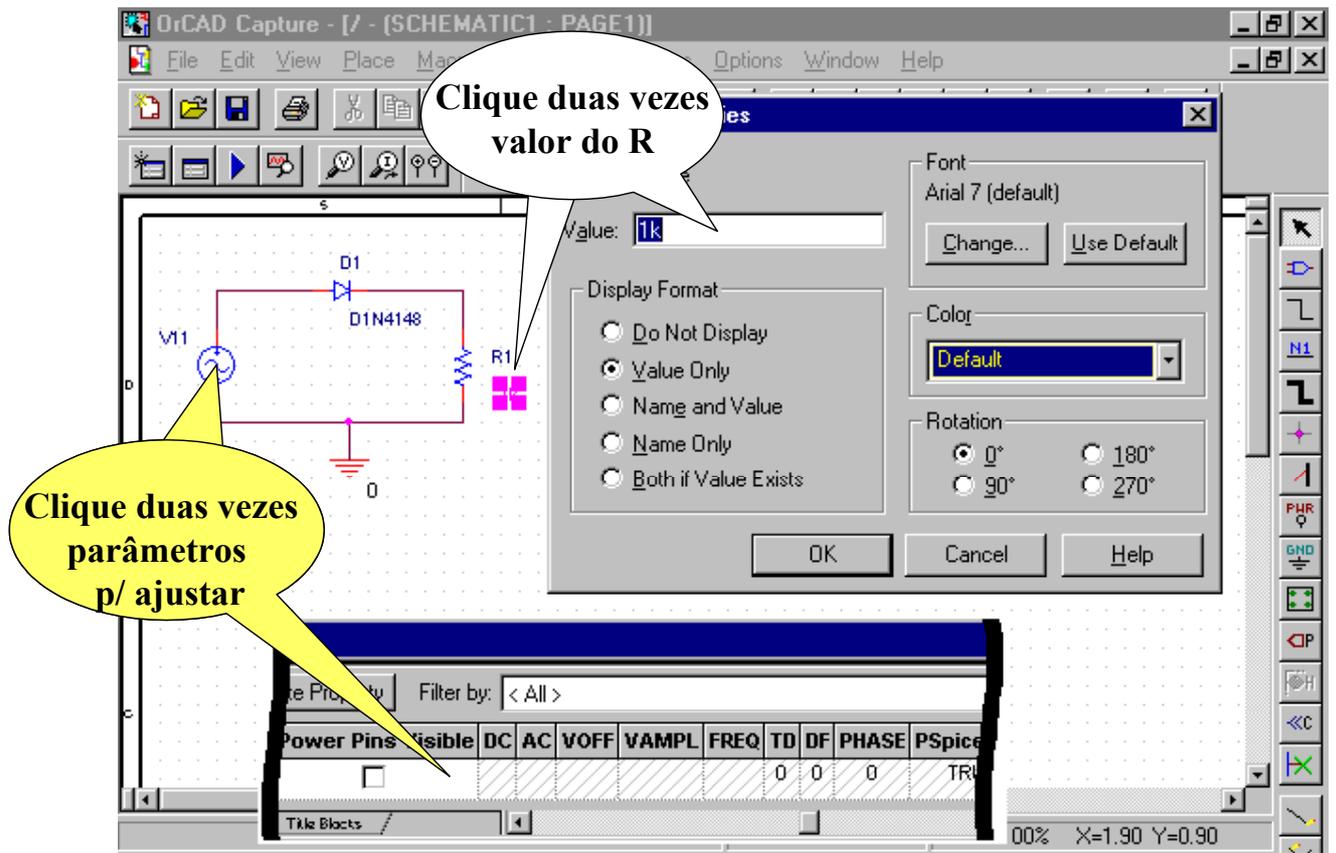
WIRE - FIOS

GROUND

- Para soltar um componente:
clique com o botão direito do mouse e selecione a opção **END MODE**;
- Para rotacionar um componente:
clique sobre ele e aperte a **tecla R**;

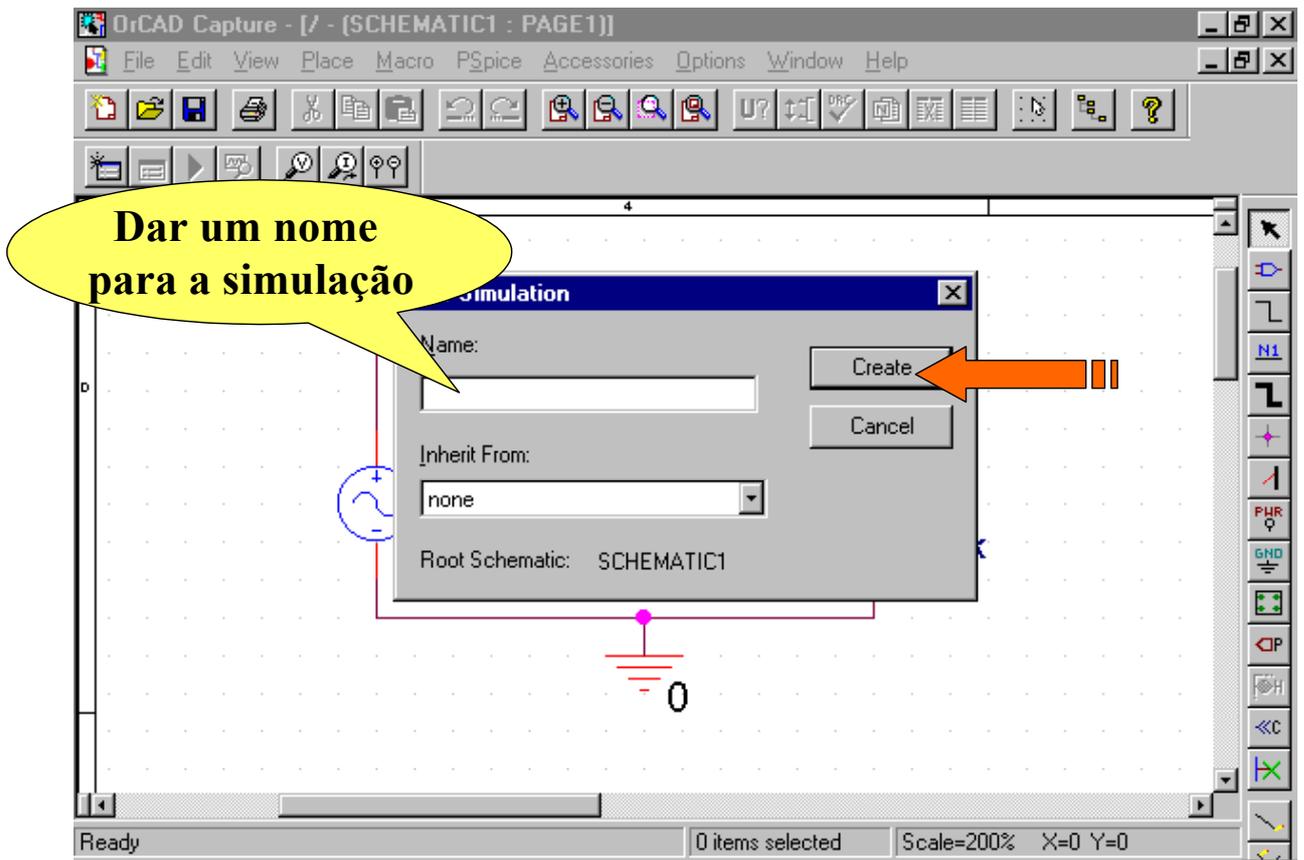
0 items selected Scale=100% X=2.10 Y=2.10

10. Editando valores de componentes e de fontes:

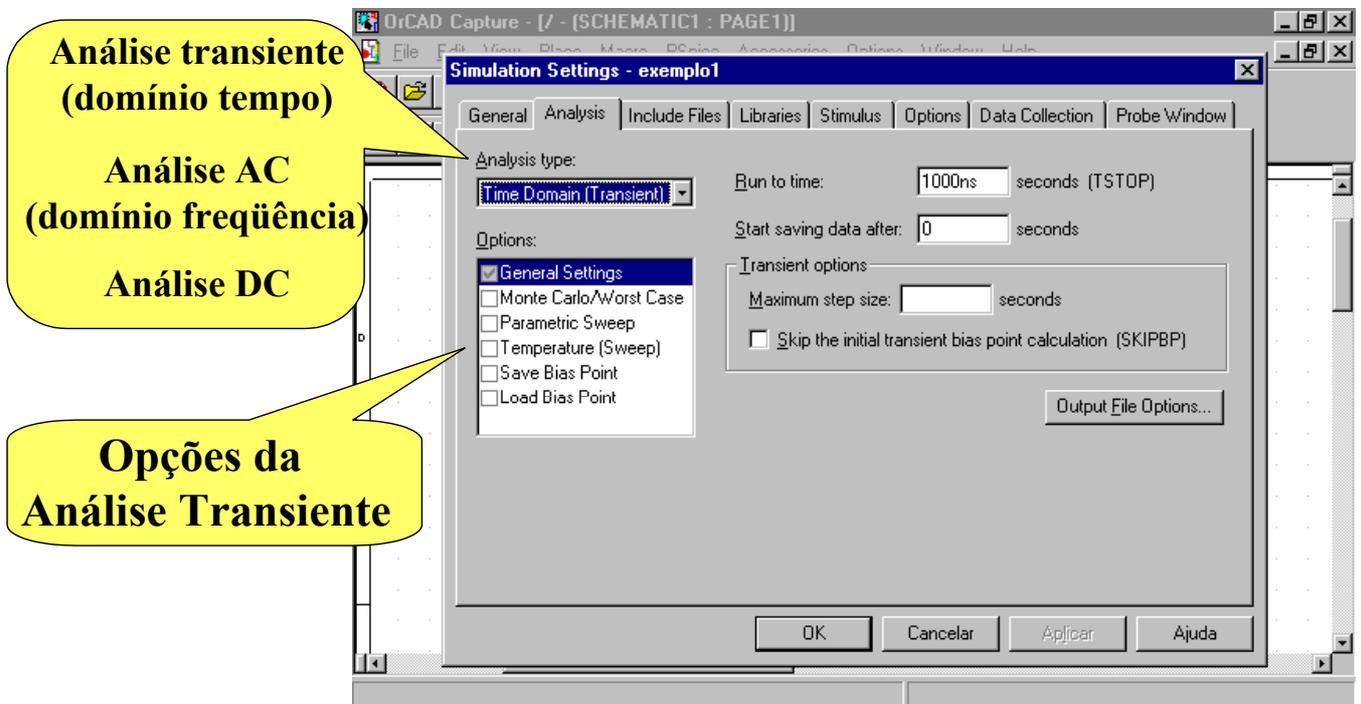


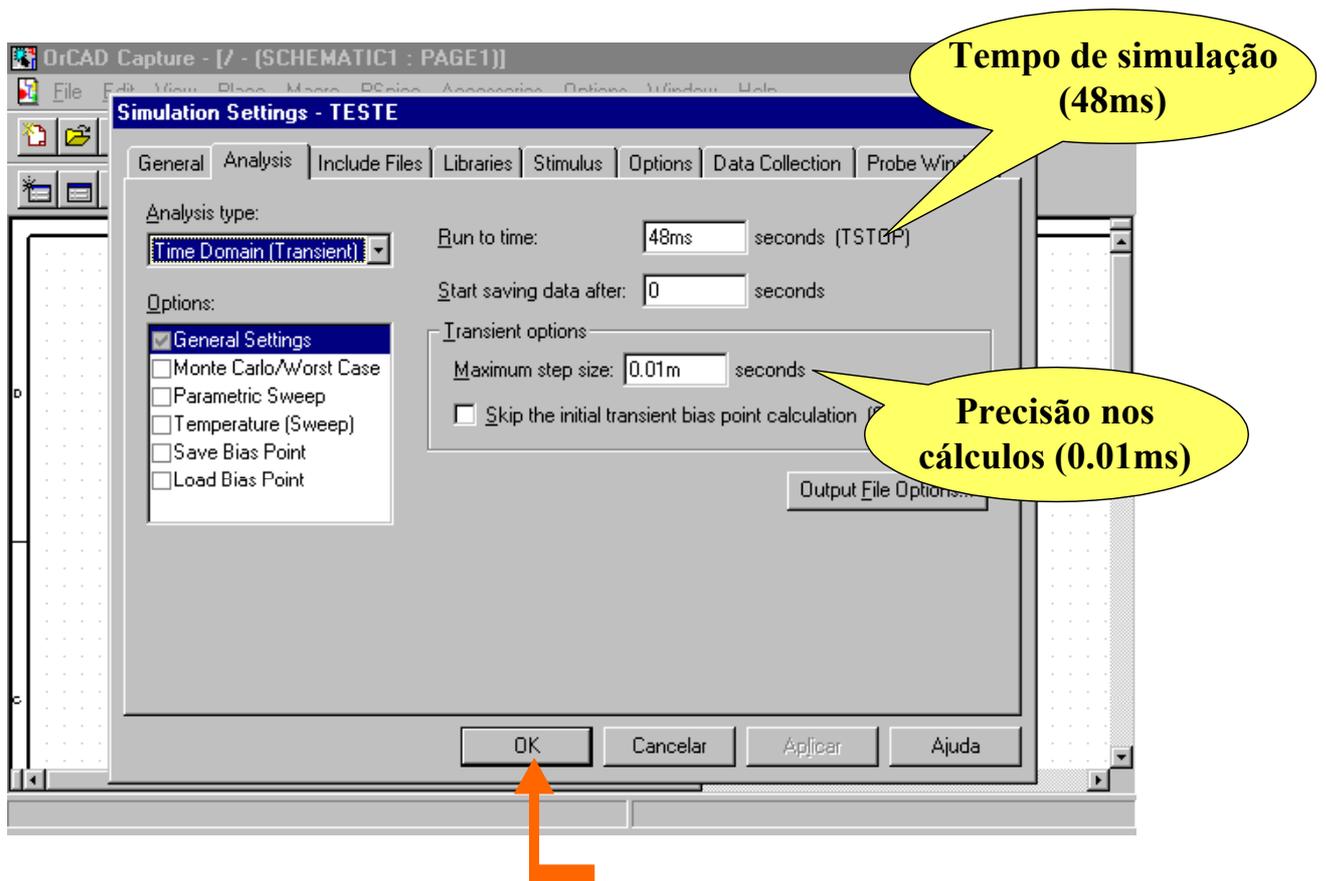
11. Salve a folha de trabalho com File => Save

12. Simulação do circuito: selecionar PSpice => New Simulation Profile. A seguir surgirá o diálogo da figura a seguir. Digite um nome para a simulação.



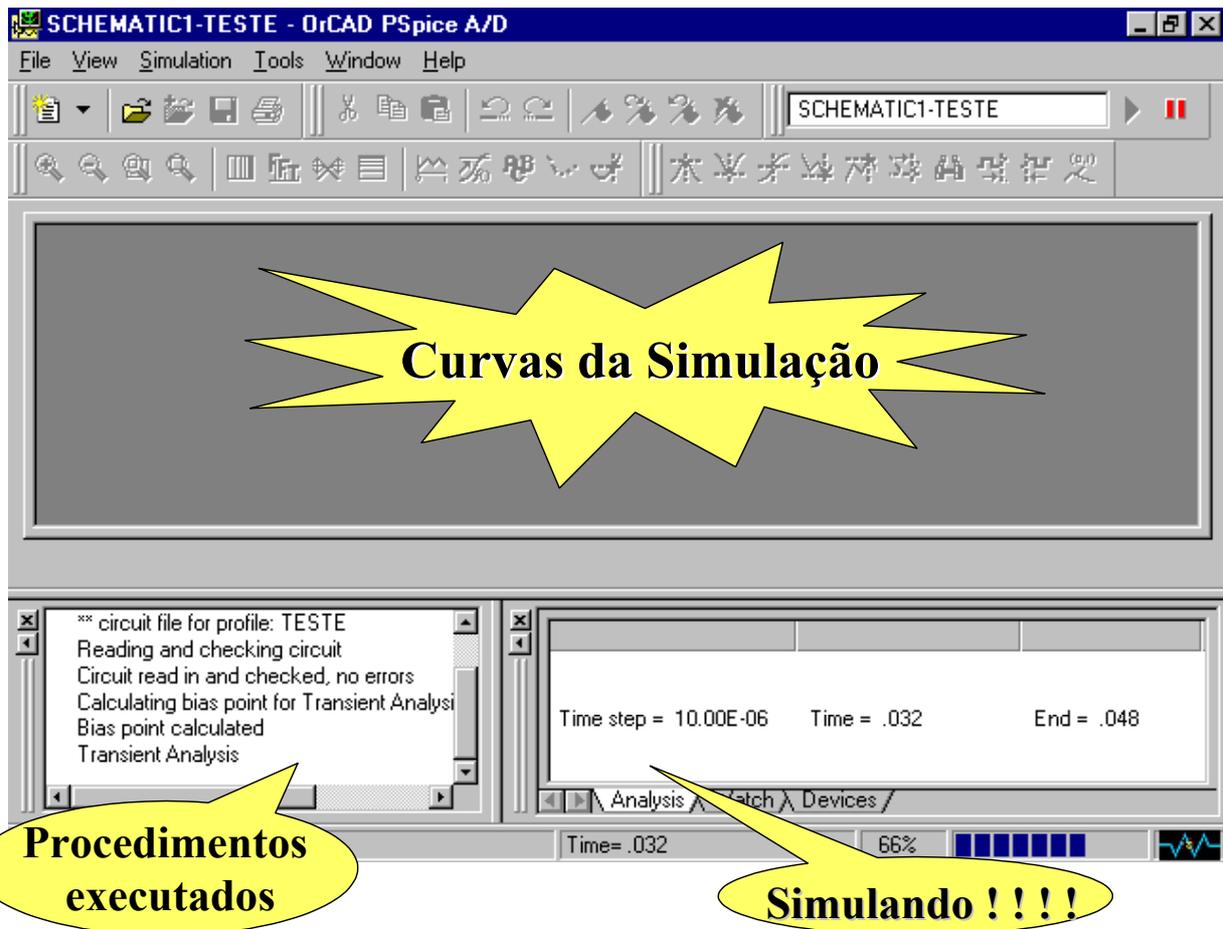
13. Ajustes dos parâmetros de simulação:



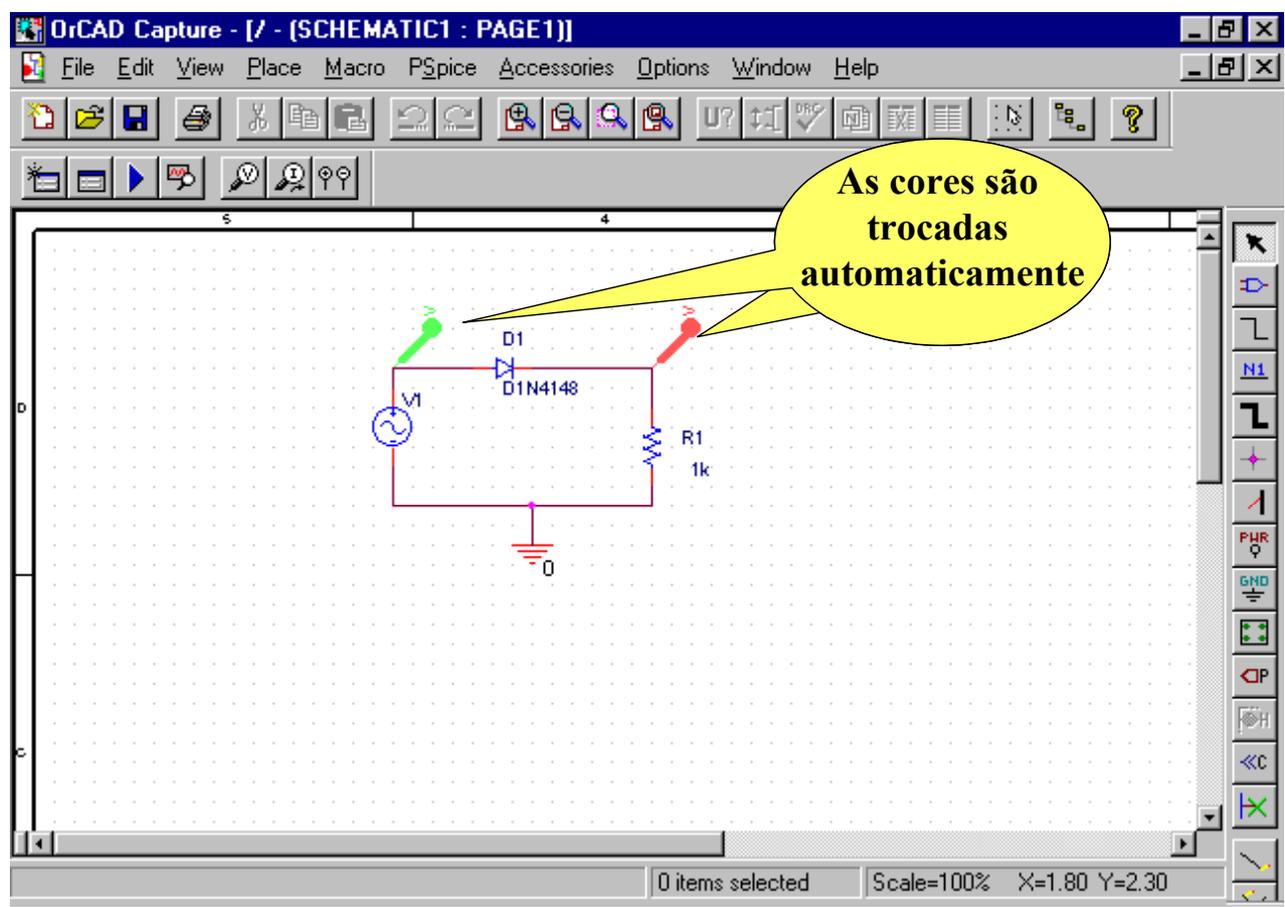
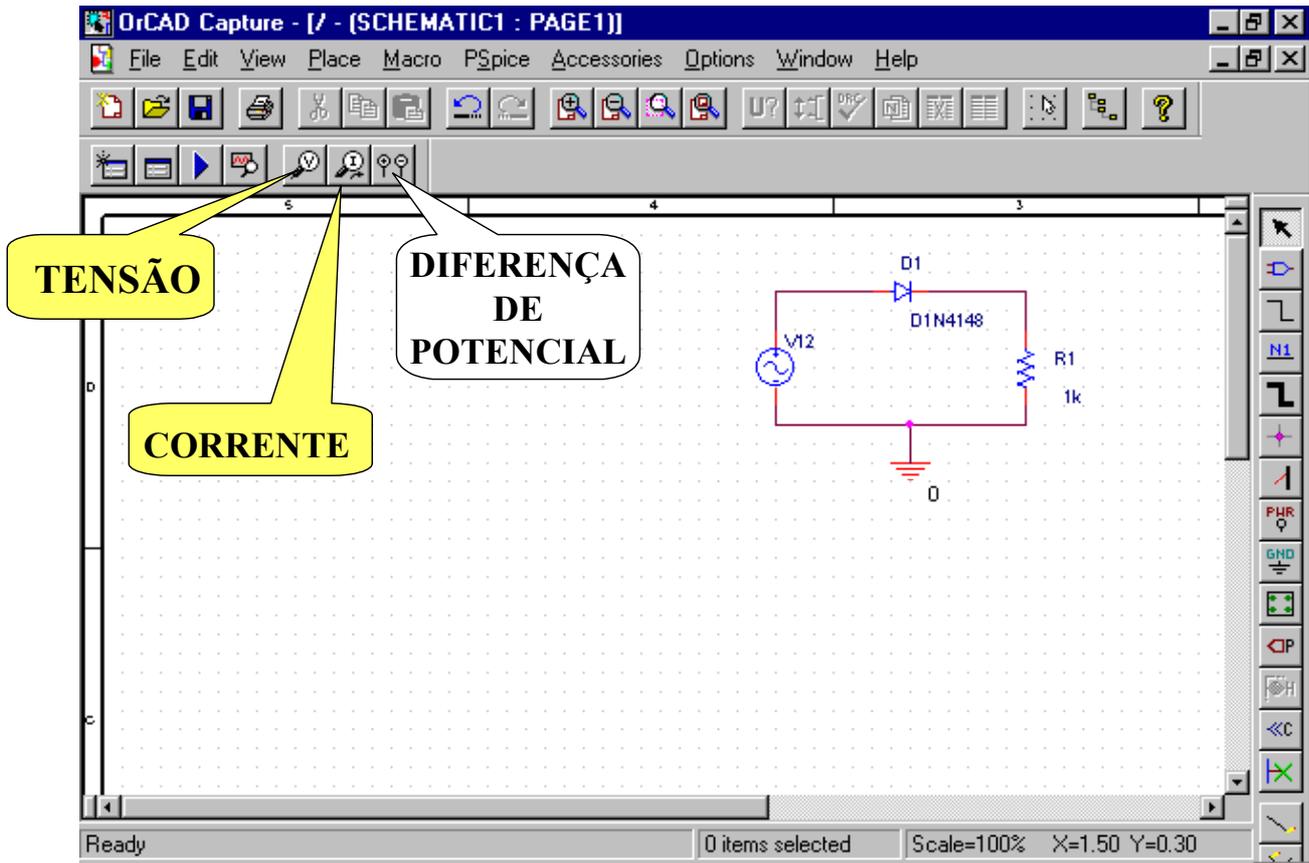


Note que é possível, opcionalmente, variar a temperatura através da opção Temperature (Sweep) e ainda levar em conta a tolerância dos componentes com a opção Monte Carlo/Worst Case.

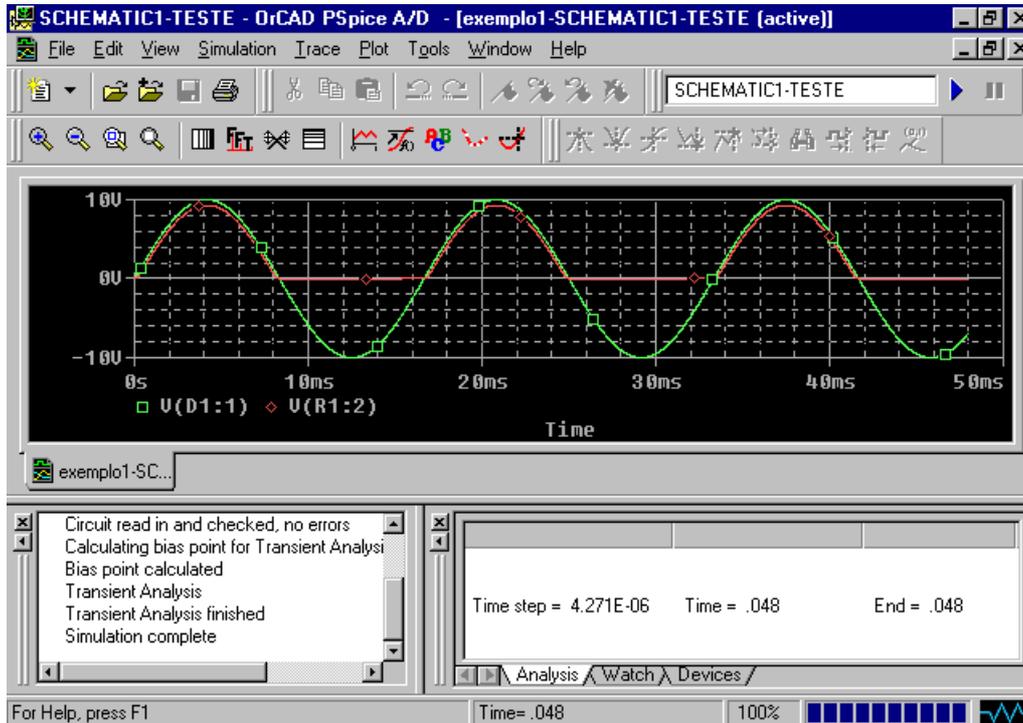
14. Execução da simulação: Escolha a opção Pspice => Run no menu principal. A figura a seguir mostra as divisões da tela do visualizador gráfico dos resultados:



15. Visualização dos gráficos: pode ser feita através da adição de ponteiros de teste, acrescentadas com os botões da barra de ferramentas mostrados na figura a seguir.

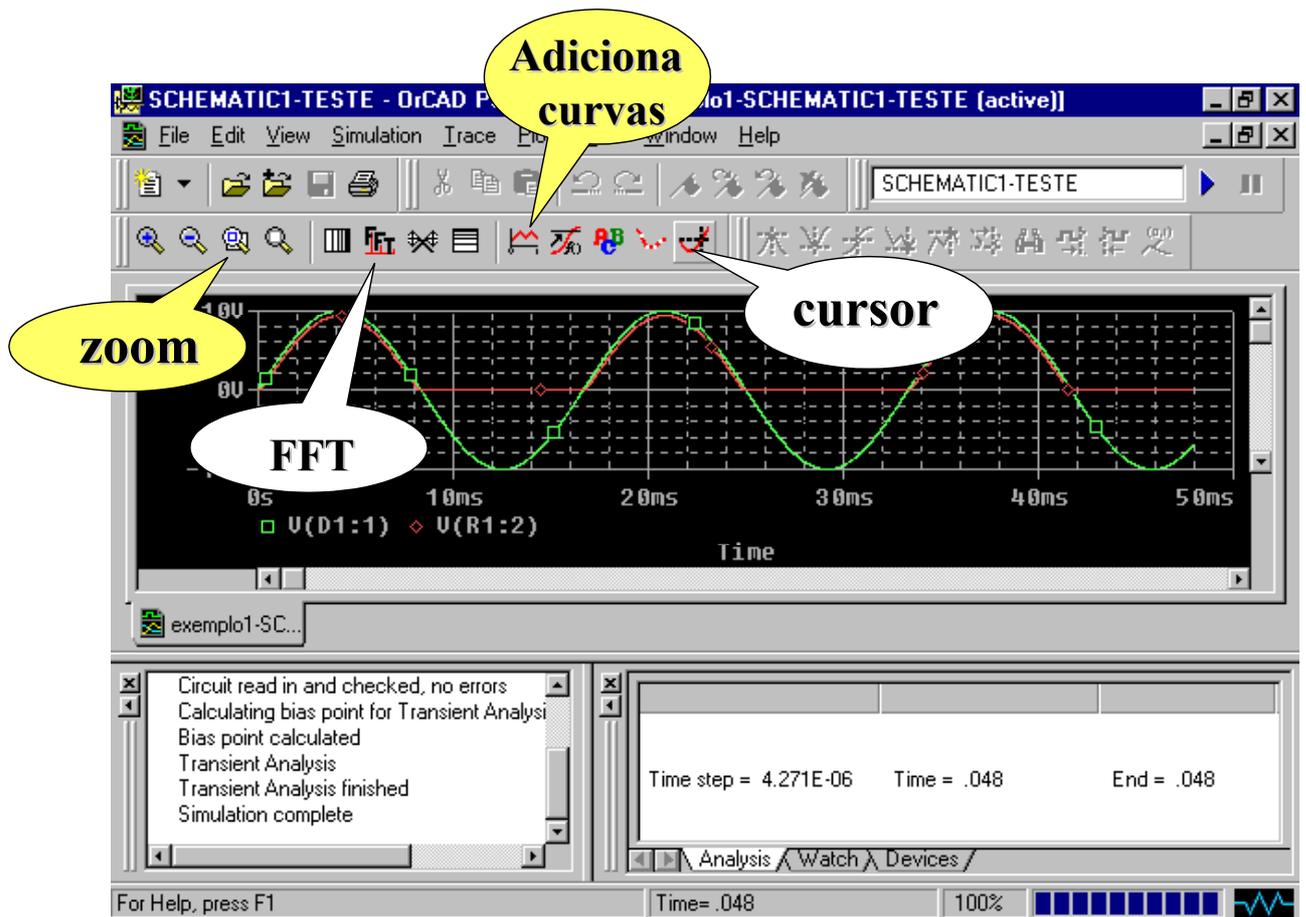


Quando se insere uma ponteira no circuito, após a simulação, as formas de ondas são automaticamente geradas no visualizador gráfico.

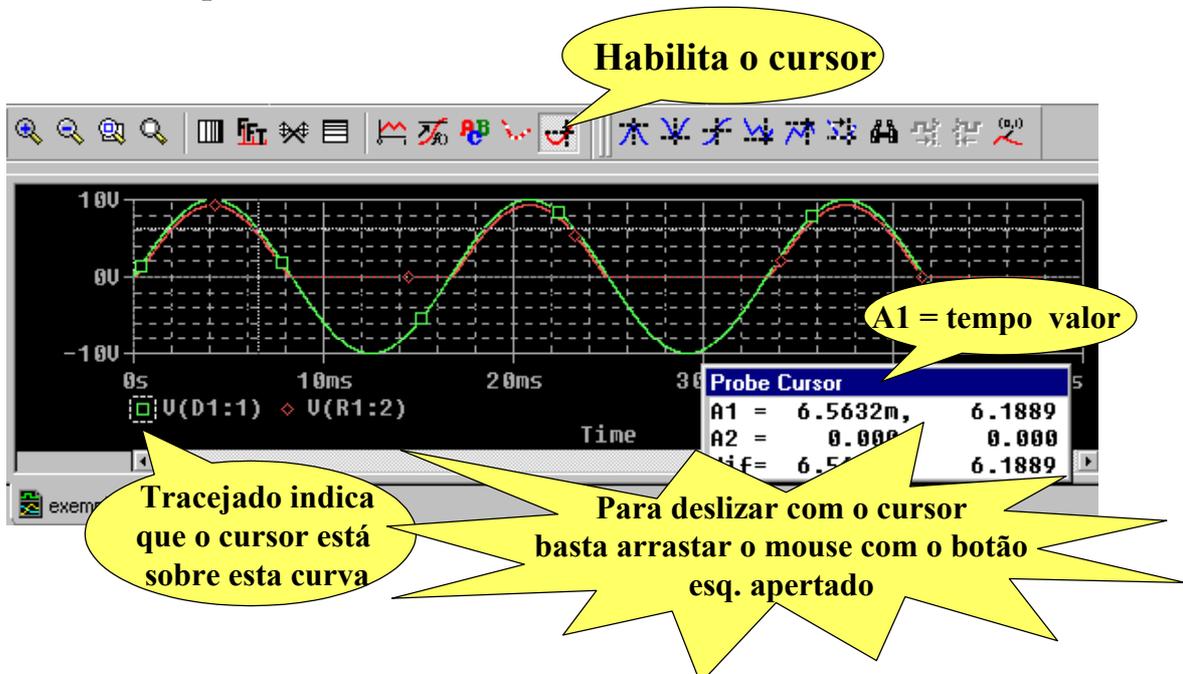


Uma outra forma de adicionar curvas é através da opção Trace => Add Trace e então escolher as variáveis que se deseja observar no diálogo que aparece.

16. Ferramentas do visualizador gráfico: O visualizador possui diversas ferramentas para ampliar (zoom) as curvas, visualizar a transformada de Fourier (FFT) e utilizar cursores para ler com precisão os valores dos pontos das curvas. A figura a seguir mostra estas ferramentas:



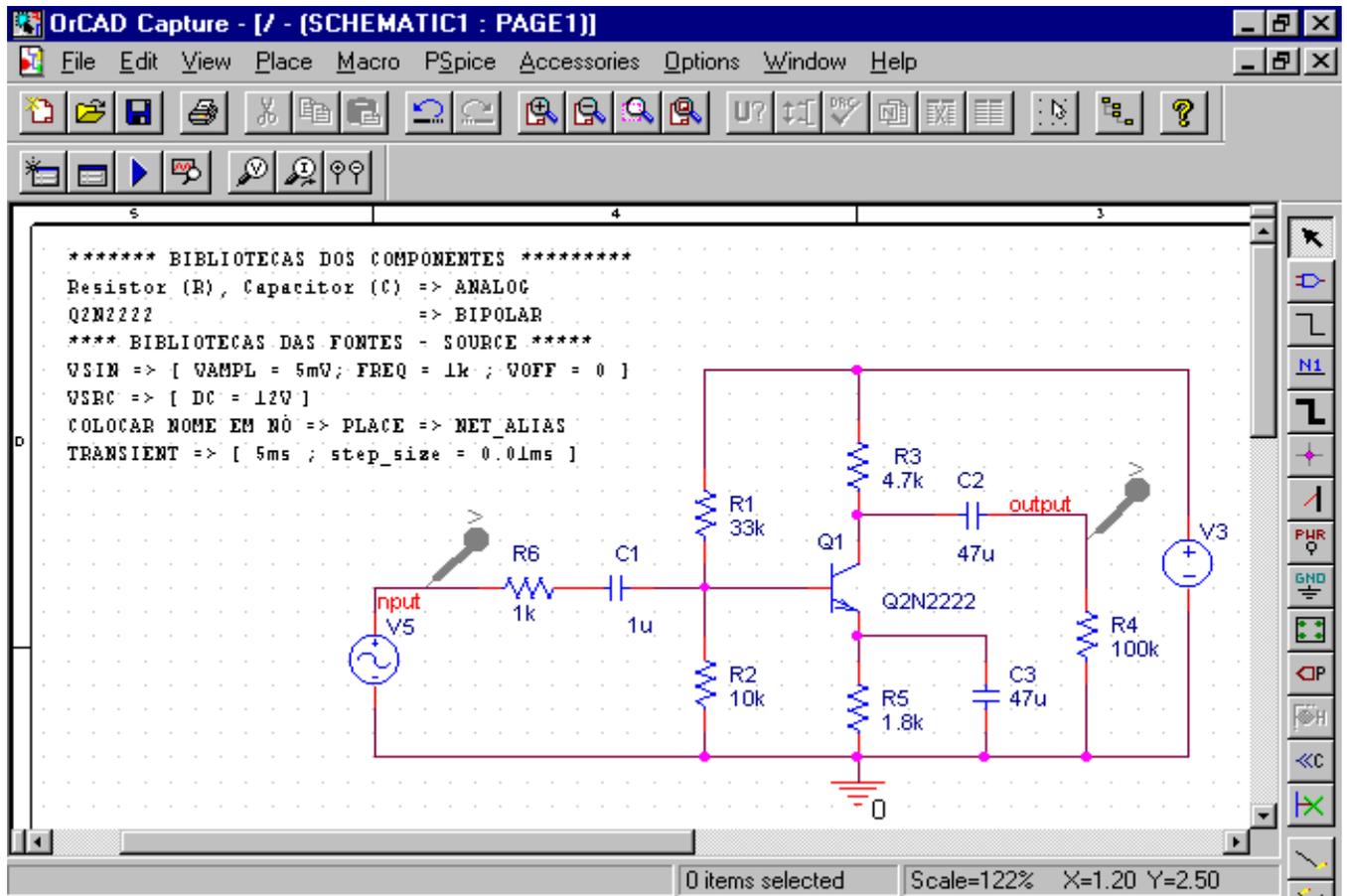
O cursor permite a visualização dos valores numéricos da curva



Obs.: para acrescentar mais um cursor clique na área gráfica com o botão direito do mouse e depois sobre a segunda curva [V(R1:2)], também com o botão direito.

EXEMPLOS PRÁTICOS:

1) AMPLIFICADOR



Dicas para montar o circuito:

FONTES => biblioteca SOURCE, componente VSIN (freq = 1000; v ampl = 5m),
componente VSRC (dc = 12)

COMPONENTES:

biblioteca ANALOG => componentes (R, C)

biblioteca BIPOLAR => componente (Q2N2222)

NOME EM FIOS:

Menu PLACE => NET ALIAS (colocar o nome)

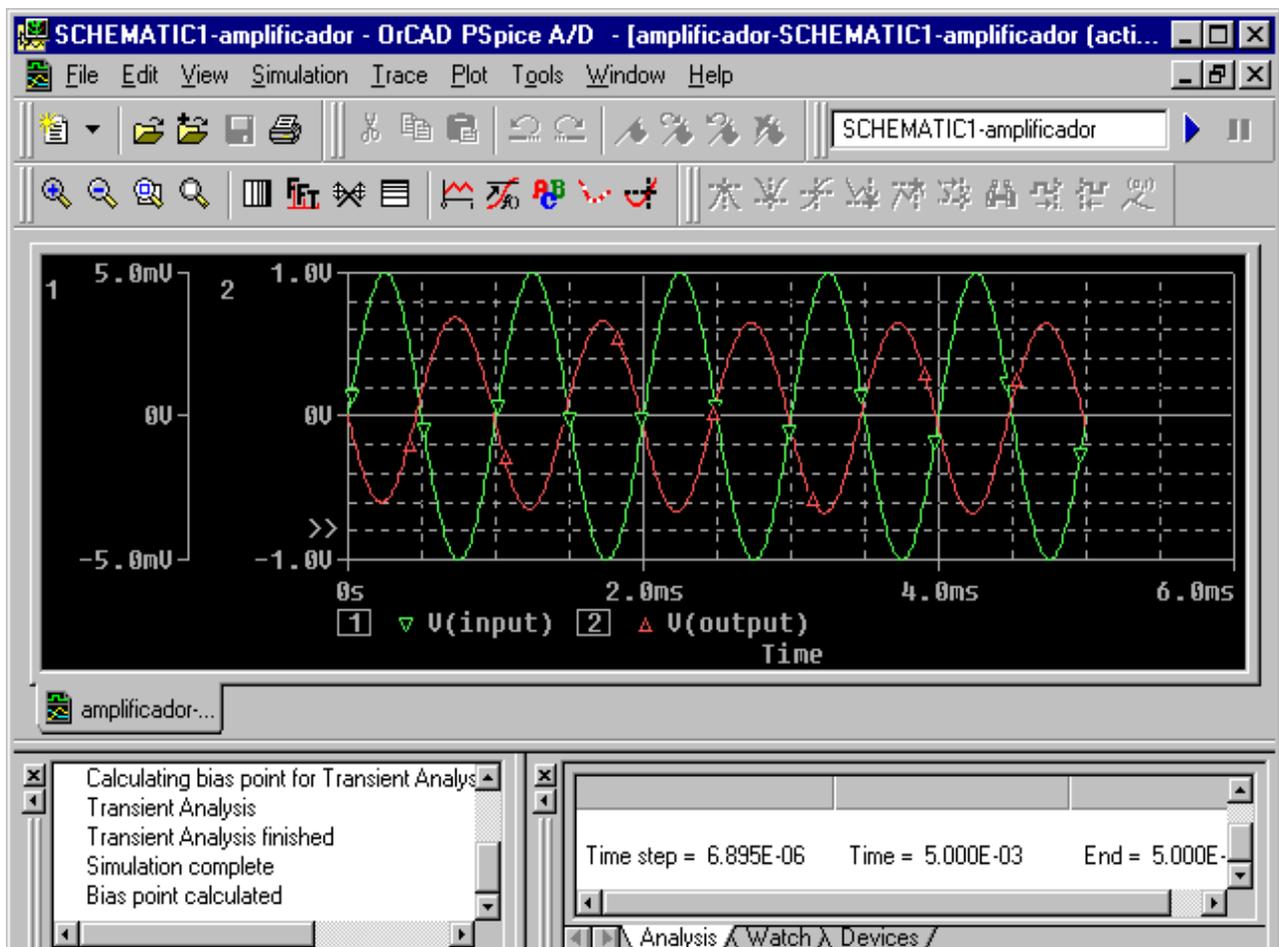
TEMPO DE SIMULAÇÃO => análise no tempo (transiente); run to time = 5ms;
maximum step size = 0.01ms

Dicas no visualizador gráfico:

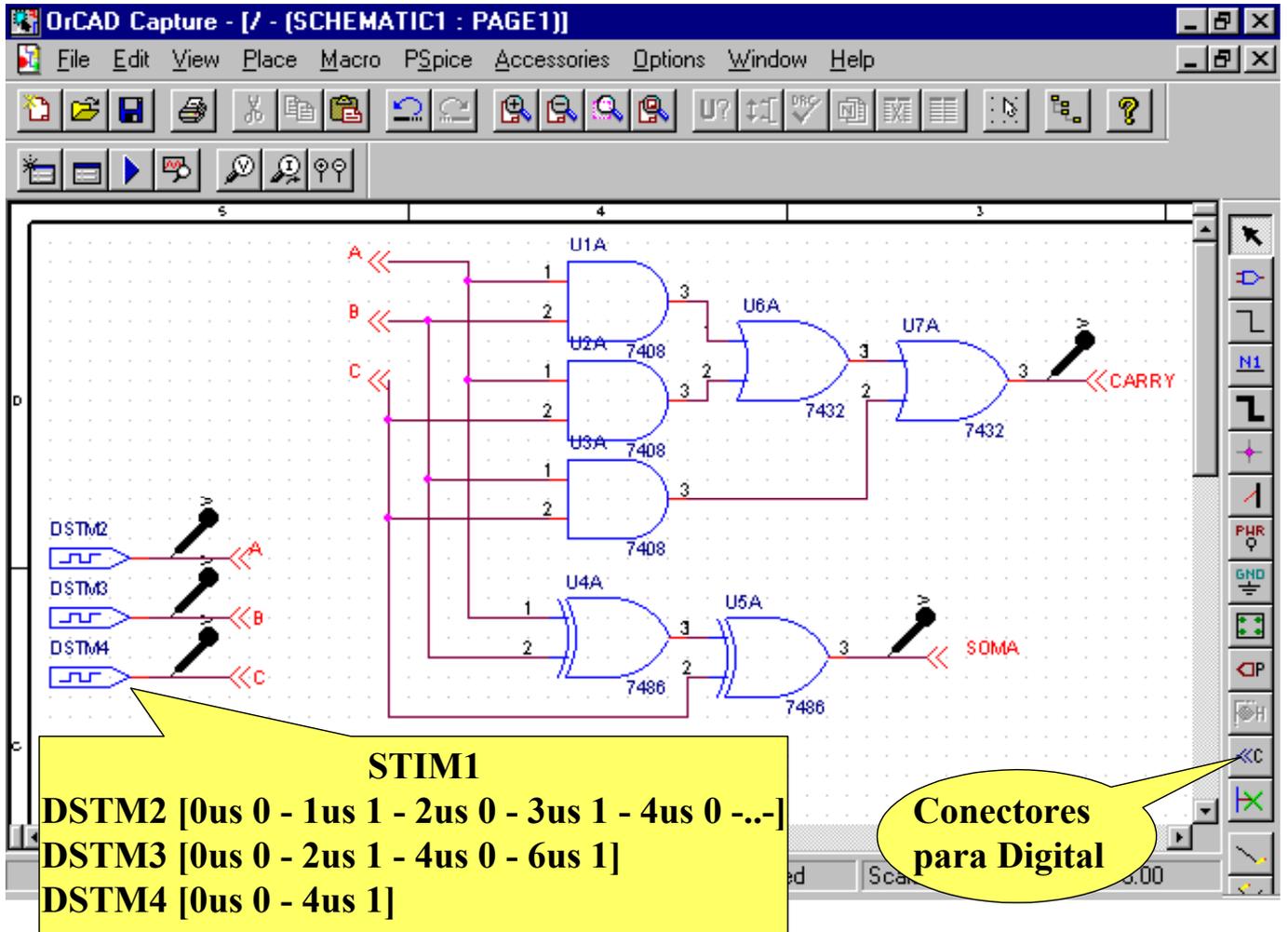
- Devido a diferença de amplitude do sinal de entrada (5mV) e do sinal de saída (1V) aconselha-se o uso de dois eixos Y (Y AXIS). COMO FAZER?

1. Apaga-se todas as curvas (clique sobre o nome e aperte a tecla DELETE).
2. No menu PLOT => ADD Y AXIS (note que aparecerá mais um eixo - clique sobre este eixo)
3. Adicione uma curva para este eixo (TRACE => ADD TRACE) escolha a curva [V(input)].
4. Clique sobre o outro eixo e adicione nova curva (TRACE => ADD TRACE) escolha a curva [V(output)].

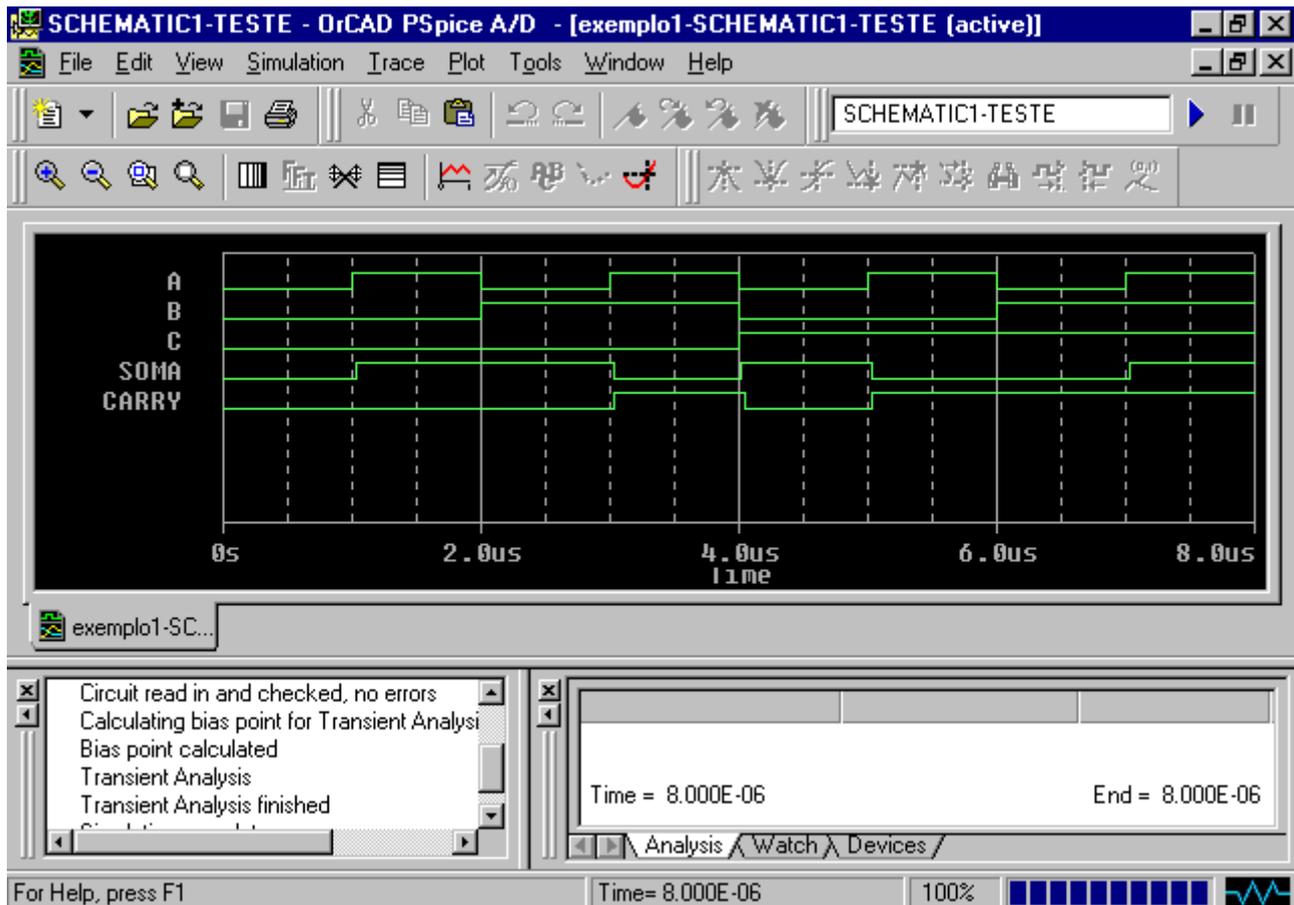
Resultado:



2) Circuito Digital SOMADOR



Resultado:



REFERÊNCIAS BIBLIOGRÁFICAS

1. Costa, Cristian C. G. Estudo de Simulação Computacional destinada à Disciplina de Circuitos Elétricos – “PSPICE”. Universidade Federal do Paraná. Curitiba. Maio de 1999.
2. SPICE – A Brief Overview (<http://www.seas.upenn.edu/~jan/spice/spice.overview.html>). Department of Electrical Engineering. University of Pennsylvania.
3. Sedra/Smith. Microeletrônica - quarta ed., Makron Books Ed. , São Paulo, 2000.
4. Sandler, Steven M. SMPS simulation with SPICE. 1996. McGraw Hill.
5. Kielkowski, Ron. Inside SPICE. 1995. McGraw Hill.