

UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Tecnologia
Curso de Engenharia Elétrica

SIMULAÇÃO DE CIRCUITOS ELETRÔNICOS EM COMPUTADORES

Ewaldo Luiz de Mattos Mehl ^[1]

SUMÁRIO

	página
1. Porquê simular circuitos em computadores?.....	1
2. Revisão Histórica da Simulação de Circuitos	2
3. O Programa "SPICE"	3
3.1. Tipos de Análise Disponíveis	4
3.2. Características da Análise DC.....	4
3.3. Características da Análise de Transientes.....	4
3.4. Características da Análise AC.....	4
4. Utilização do Programa "SPICE"	5
4.1. Descrição do Circuito.....	5
4.2. Principais Comandos.....	8
4.3. Exemplo de Modelagem: Resistores	9
5. Análise de Circuitos Retificadores.....	9
5.1. Fontes de Tensão Senoidais	9
5.2. Modelagem de Diodos	10
5.3. Análise Transiente	11
5.4. Exemplo de um Retificador de Onda Completa	11
6. Análise de Circuitos com Transistores Bipolares.....	13
6.1. Parâmetros de Transistores Bipolares.....	13
6.2. Sintaxe dos Comandos para Transistores Bipolares.....	13
6.2. Exemplo para Simulação	14
7. Conclusões.....	15
8. Referências Bibliográficas.....	15
Anexo 1: Parâmetros SPICE dos Transistores Bipolares.....	16
Anexo 2: Resultados de Simulações	17

^[1] Engenheiro Eletricista (UFPR-1980), Mestre em Ciências em Engenharia de Materiais (COPPE/UFRJ-1987); Doutor em Engenharia Elétrica (UFSC-1996)
Professor Adjunto no Departamento de Eletricidade da Universidade Federal do Paraná.

SUMÁRIO:

O presente trabalho apresenta as principais vantagens da simulação de circuitos eletrônicos através de programas computacionais, detendo-se com maior atenção nos programas SPICE e PSpice. É feita uma breve revisão histórica sobre a simulação de circuitos em computadores e citam-se as principais características do programa PSpice, com descrição dos tipos de análises disponíveis e forma de entrada de dados. Mostra-se algumas das aplicações do programa PSpice através de exemplos com circuitos simples.

1. PORQUÊ SIMULAR CIRCUITOS EM COMPUTADORES?

A necessidade de se utilizar programas computacionais para análise das condições de funcionamento de circuitos eletrônicos é evidente para todos que se dedicam à tarefa de projetar circuitos integrados. No caso de circuitos eletrônicos “convencionais”, ou seja, constituídos de elementos discretos, é geralmente possível (apesar de caro e demorado) que sejam fabricados um ou mais protótipos do circuito projetado, com a finalidade de se analisar o projeto. Nestes protótipos são realizadas medições e testes, muitas vezes acompanhadas de trocas de componentes, até se obter uma “versão final” cujo funcionamento aproxima-se do desejado. Levando idêntico raciocínio para o caso do projeto de circuitos integrados, observa-se a impossibilidade de se seguir o mesmo procedimento. Uma vez que a fabricação de circuitos integrados envolve uma série de etapas com tecnologia complexa e cara, a fabricação de protótipos deve ser minimizada. Uma limitação adicional é o fato de ser impossível “trocar” um transistor dentro de um circuito integrado ou fazer uma ligação esquecida. Além disso, as medições de parâmetros de dispositivos dentro de um circuito integrado, apesar de serem possíveis, envolvem problemas de acesso e necessita-se instrumentação com grande grau de complexidade. Ou seja, é imperativo que os projetistas de circuitos integrados saibam como seus projetos vão se comportar antes de tentar fabricá-los.

Apesar da maioria dos programas para simulação de circuitos eletrônicos terem sido originalmente desenvolvidos tendo em vista a análise de sistemas de razoável complexidade [3], é evidente que se pode simular com tais programas qualquer circuito. No entanto, os primeiros programas para simulação necessitavam de computadores de grande porte, o que limitava sua utilização a grandes empresas ou centros de pesquisa. A partir da disponibilidade de microprocessadores e programas para simulação de circuitos que funcionam em microcomputadores de custo reduzido, as vantagens da simulação abriram-se aos projetistas de qualquer sistema eletrônico. Entre tais vantagens, citam-se [6]:

- O projetista pode verificar, dentre várias configurações disponíveis, qual terá desempenho mais adequado à sua aplicação.
- Como não há necessidade de se construir diversos protótipos e fazer testes e medições em laboratório, é também desnecessário adquirir ou obter componentes para montagens experimentais.
- Todos os “componentes” do simulador “funcionam” e “não queimam”. Quando se faz a montagem de um protótipo no laboratório, é possível que se tenha a má sorte de se usar componentes defeituosos ou com características diversas daquelas constantes nos catálogos. Desta forma, ao contrário da simulação, se o protótipo não funcionar adequadamente não se saberá com certeza se tal se deve a:
 - Projeto errado;
 - Montagem errada;
 - Componentes danificados ou fora de especificação.
- É possível fazer todo o tipo de “medida” no circuito simulado, mesmo que tais medições sejam difíceis (gerariam ruído ou carregariam excessivamente o circuito), inconvenientes (o circuito poderia ser destruído) ou impossíveis (não se tem o equipamento necessário).
- É perfeitamente viável com um simulador fazer “experiências”, tais como trocar valores dos componentes ou até mesmo tentar novas configurações, por mais “estranhas” que sejam, sem receio de danificar o circuito ou os equipamentos de teste.

A disponibilidade dos modernos simuladores em microcomputadores tem tido notáveis implicações também no ensino da eletrônica. Desenvolvidos originalmente como ferramentas para pesquisas e projetos avançados de circuitos integrados, os simuladores de circuitos eletrônicos estão atualmente sendo largamente utilizado no ensino de Engenharia. Um grande número de Universidades tem adotado programas de simulação como auxiliares didáticos em disciplinas de análise de circuitos e de projeto de circuitos eletrônicos [2,9]. O uso de simuladores para o ensino de análise de circuitos, longe de se considerar um modismo passageiro nos cursos de graduação de Engenharia Elétrica, é resultado da crescente tendência de se enfatizar o ensino de técnicas de projeto, nas quais os microcomputadores e *workstations* são ferramentas de inestimável valor.

2. REVISÃO HISTÓRICA DA SIMULAÇÃO DE CIRCUITOS

Até 1960 só era possível a análise do funcionamento das grandes redes elétricas em regime senoidal com o auxílio de computadores analógicos, compostos de um conjunto de amplificadores e elementos lineares, interligados por cabos e interruptores. Com a disponibilidade de computadores digitais e principalmente de linguagens de programação mais “amigáveis”, tal como FORTRAN (FORmula TRANslation Language), foram feitas diversas tentativas de utilização de computadores para a solução dos sistemas de equações relacionados com circuitos elétricos. Nessas tentativas pioneiras, verificou-se que a análise de circuitos elétricos através de equações obtidas pelo método das correntes de malhas, apesar de conveniente para a análise “manual”, revela-se de difícil implementação em um programa de computador. Já no caso da utilização de análise nodal, obtém-se de modo mais direto um sistema matricial, fazendo com que a análise do circuito se torne basicamente um problema de solução numérica do sistema de equações obtido [5].

Apresenta-se à seguir uma breve descrição dos principais programas simuladores apresentados no período de 1959 a 1990:

PETAP (1959) e **PETAP-A** (1960): Programas desenvolvidos na empresa IBM que permitiam a análise de transitórios no chaveamento de transistores. Apesar de ser um sistema interessante, seu uso ficou restrito à IBM.

PREDICT (1964): Programa desenvolvido pela IBM à partir do PETAP, porém destinado a uso militar, para determinar o desempenho de sistemas eletrônicos frente a radiações nucleares.

NET1 (1964): Foi desenvolvido no laboratório norte-americano de Los Alamos (onde foi desenvolvida a primeira bomba atômica, durante a Segunda Guerra Mundial), igualmente resultante do interesse militar em se verificar o efeito de radiações nucleares nos circuitos eletrônicos.

ECAP1 (1965): Também produto da IBM, como evolução do programa PETAP, porém para aplicações “civis”. Possuía defeitos comuns a programas de primeiras gerações: pouco “amigável” e difícil de ser utilizado. Tinha tendências de não—convergência na solução das equações. Apesar desses problemas, foi colocado à disposição dos usuários de equipamentos IBM.

SPECTRE (1967): Considerado como evolução do programa PREDICT, desenvolvido pela IBM para aplicações na área aeroespacial. Assim como o programa NET1 de Los Alamos, realizava a solução de equações diferenciais não-lineares por meio de sucessivas integrações numéricas em pequenos “passos”. Esse fato, aliado às baixas velocidades dos computadores de então, fazia com que o tempo de processamento se tornasse extremamente longo.

TRAC (1968): Desenvolvido na empresa aeroespacial norte-americana Rockwell, através de uma equipe de engenheiros assessorada por matemáticos. Utilizado um método de integração que foi chamado de **Método de Euler Reverso** (*Backward-Euler Method*). Neste método as equações integro-diferenciais são transformadas a cada passo de integração em um conjunto de equações algébricas, ou seja, de mais fácil solução e conseqüentemente com menor tempo de processamento. O uso desses programa, no entanto, permaneceu restrito à área militar.

TIME (1969): O programa TRAC possuía uma série de subrotinas escritas em linguagem “de máquina” (*assembly*), fato que o tornava não-transportável para outros computadores. O programa TIME foi desenvolvido na Universidade da Califórnia em Berkeley com base no programa TRAC, porém com todas as subrotinas em linguagem FORTRAN, de modo a funcionar em computadores de diferentes fabricantes.

CIRPAC (1969): Desenvolvido nos Laboratórios Bell, utilizava esquema de cálculo semelhante aos programas TRAC e TIME, porém pequenas melhorias nos métodos de integração faziam-no mais rápido que seus antecessores.

BIAS (1970) e **BIAS3** (1971): Desenvolvidos na Universidade da Califórnia em Berkeley à partir de um problema prático, onde se desejava observar o efeito da variação de temperatura em circuitos transistorizados simples. Utilizavam o modelo de Ebers-Moll para modelamento dos transistores e obtinha a solução do sistema de equações pelo método de Newton-Raphson.

ASTAP (1973): Programa desenvolvido na IBM. Apresentava como principal novidade o uso de técnicas inéditas para otimização da velocidade de processamento, tomando vantagem do fato das matrizes que descrevem os circuitos elétricos serem esparsas (possuem muitos elementos nulos). Este programa foi amplamente utilizado pelos projetistas da própria IBM e também esteve disponível para usuários externos à empresa.

CANCER (1971) e **SLIC** (1971): Foram resultados de um trabalho desenvolvido na Universidade da Califórnia em Berkeley por Ronald A. Rohrer, coordenando uma equipe de alunos de pós-graduação, que estudaram técnicas de descrição de circuitos e de solução dos sistemas de equações obtidos. Estes

programas utilizavam o método de **Análise Nodal Modificada**. Antes de se tornar professor em Berkeley, Rohrer havia trabalhado na empresa Fairchild, onde desenvolveu em 1968 um programa chamado **FAIRCIRC** para simulação dos primeiros circuitos integrados produzidos pela empresa.

SPICE (1972), **SPICE2** (1975) e **SPICE3** (1980): Evoluções do programa CANCER, desenvolvidos também na Universidade da Califórnia em Berkeley. O programa SPICE foi apresentado na Tese de Doutorado de Laurence W. Nagel [4], mas apresentava algumas limitações. Nagel posteriormente modificou o programa original, surgindo a versão SPICE2. Estes programas foram colocados à disposição do público e ganharam reconhecimento internacional pela versatilidade dos modelos utilizados e velocidade de processamento [4,6]. O programa SPICE2, codificado em FORTRAN, foi fornecido gratuitamente pela UC-Berkeley a diversas universidades e centros de pesquisa ao longo da década de 1980 e compilado para os mais variados computadores. Nos anos seguintes, diversos alunos de Berkeley encarregaram-se de re-codificar o programa SPICE2 em linguagem "C", dando origem ao SPICE3, igualmente distribuído gratuitamente a usuários universitários.

ICD (1975): Programa da IBM muito parecido com o SPICE2 nos métodos de solução empregados. Foi escrito em linguagem APL com características interativas com o usuário.

ADVICE (1980): Desenvolvido por Laurence W. Nagel (o mesmo autor do SPICE2), nos Laboratórios Bell. É considerado uma versão mais "amigável" do programa SPICE2.

PSpice (1985): Foi o primeiro programa comercial desenvolvido exclusivamente para ser usado em microcomputadores. É basicamente o programa SPICE2, que foi adaptado pela empresa MicroSim para uso em microcomputadores IBM-PC e posteriormente para *workstations* com sistema operacional UNIX. A grande vantagem em relação ao programa SPICE2 de Berkeley foi a apresentação de um programa associado chamado PROBE, de modo a permitir a visualização dos resultados da simulação de modo mais interessante que as listagens originais do SPICE2. As versões mais recentes do programa PSpice incluem a possibilidade de se efetuar simulação lógica do circuito simultânea à simulação analógica. Estão também disponíveis diversas "bibliotecas" de dispositivos semicondutores comerciais pré-modelados [6]. Há versões para microcomputadores (sistemas MS-DOS e Windows) e para diversos tipos de *workstations*.

IG-SPICE (1989): Programa comercial também baseado no SPICE2, para microcomputadores, produzido pela Empresa INTUSOFT. Tem como principal característica o interfaceamento com um programa gráfico que permite ao usuário "desenhar" seu circuito, extraíndo-se automaticamente de tal "desenho" o arquivo de descrição do circuito necessário para simulação. Esta característica foi também incorporada pela MicroSim em versões posteriores do PSpice para o sistema Windows.

É necessário esclarecer que muitos outros programas para simulação foram desenvolvidos no período citado. A relação anterior mostra apenas aqueles de maior importância. De mesmo modo, continuamente vê-se o lançamento de simuladores com diversas características. É inegável porém que o programa SPICE tem se constituído em um padrão *de facto* entre os simuladores utilizados pelos projetistas de circuitos eletrônicos.

3. O PROGRAMA SPICE

SPICE significa *Simulation Program with Integrated Circuits Emphasis* (Programa de Simulação com Ênfase em Circuitos Integrados). Concebido por Laurence W. Nagel na Universidade da Califórnia em Berkeley, o programa SPICE teve sua primeira versão finalizada em 1972 [4]. Tornou-se de domínio público, mais de cem cópias do programa foram fornecidas gratuitamente a outras universidades e a diversas indústrias eletrônicas. Um acompanhamento feito pelo autor junto a estes usuários revelou a necessidade de melhoramentos no programa, o que levou ao desenvolvimento do SPICE2, a segunda versão do programa.

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, destinada a detectar erros "grosseiros" (fontes em curto-circuito, por exemplo). Diversos parâmetros do circuito podem ser omitidos, adotando-se neste caso valores-padrões (*default*) previamente definidos.

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

3.1. TIPOS DE ANÁLISES DISPONÍVEIS

A simulação de um circuito eletrônico normalmente envolve a combinação de três análises: ponto de operação (análise DC), análise de sinais transientes no domínio do tempo e análise para pequenos sinais AC. O programa SPICE (e também o PSpice) permite que sejam feitas as seguintes análises:

- | | |
|---------------------|---|
| Análise DC: | <ul style="list-style-type: none">- Ponto de Operação DC;- Parametrização linearizada dos Modelos;- Função de Transferência para pequenos sinais;- Sensibilidade para pequenos sinais;- Curvas de Transferência DC. |
| Análise Transiente: | <ul style="list-style-type: none">- Resposta no domínio do tempo;- Análise de sinais em série de Fourier. |
| Análise AC: | <ul style="list-style-type: none">- Resposta a pequenos sinais no domínio da frequência;- Análise de ruído;- Análise de distorção. |

3.2. CARACTERÍSTICAS DA ANÁLISE DC

Através da **Análise DC** determina-se o ponto de operação quiescente do circuito. Nesta análise os capacitores são considerados como circuitos abertos e os indutores como curto-circuitos. Ao final desta análise o programa SPICE fornece as tensões em todos os nós do circuito, as correntes nas fontes de tensão independentes e a potência total quiescente dissipada no circuito.

Para a simulação da operação do circuito sob pequenos sinais, os elementos não-lineares do circuito são substituídos por modelos linearizados, obtendo-se assim a resposta do circuito. Os parâmetros dos modelos lineares dependem do ponto de operação calculado. Especificando-se uma variável de saída e uma fonte de entrada, o programa SPICE permite obter também a função de transferência do circuito para pequenos sinais. Além disso, o programa determina a resistência de entrada e de saída do circuito para pequenos sinais. Através de um sub-programa, ainda é possível obter a sensibilidade de uma determinada variável de saída com relação a qualquer parâmetro do circuito.

O ponto de operação do circuito pode ser calculado repetidas vezes, para uma série de valores de uma determinada fonte, obtendo-se uma “varredura” da qual se obtém as características de transferência DC do circuito. O usuário determina a faixa de valores dentro da qual a fonte varia, bem como os “passos de cálculo” ou incrementos de variação.

3.3. CARACTERÍSTICAS DA ANÁLISE TRANSIENTE

A **Análise Transiente** permite determinar a resposta do circuito em função de sinais variáveis no domínio do tempo. O comportamento no tempo “zero” é obtido pela Análise DC; portanto, a Análise DC será sempre realizada antes da Análise Transiente, mesmo que o usuário não a solicite.

O intervalo de tempo da simulação é determinado pelo usuário, bem como os incrementos ou “passos” de tempo. Na verdade o programa determinará internamente um “passo de cálculo” variável, com o objetivo de garantir a precisão dos resultados, sendo que os resultados de saída são interpolados de modo a se acomodarem aos intervalos de tempo determinados pelo usuário. As fontes do circuito podem ser puramente senoidais ou com diferentes formas de variação no tempo.

Uma sub-rotina incorporada ao programa permite a obtenção dos coeficientes da série de Fourier de um sinal específico. Com isso, pode-se avaliar a taxa de distorção harmônica (*THD – total harmonic distortion*) de um sinal, desde que tal nível de distorção seja relativamente elevado.

3.4. CARACTERÍSTICAS DA ANÁLISE AC

A **Análise AC** tem como objetivo determinar a resposta de frequência do circuito. Normalmente especifica-se uma série de valores de frequência, de modo que a análise seja feita de forma repetitiva dentro desta faixa de variação. Ao final da análise, os resultados podem ser apresentados em magnitude absoluta ou relativa (dB) ou em diagramas com fase, parte real e parte imaginária (diagramas de Bode e de Nichols).

Através da Análise AC é também possível verificar as características de ruído e de distorção de um circuito eletrônico. Os resistores e dispositivos semicondutores do circuito são considerados como fontes de ruído térmico, fazendo-se o modelamento matemático por meio da inclusão de fontes de ruído a cada um destes elementos. A contribuição de cada uma destas fontes de ruído é computada separadamente, obtendo-se o ruído total pela soma dos valores eficazes das fontes de ruído individuais.

Uma vez que a análise por série de Fourier não produz bons resultados para níveis baixos de distorção de sinais, os programas SPICE e PSpice permitem avaliação de distorção harmônica no domínio da frequência. Obtém-se com isso valores de distorção harmônica de segunda e terceira ordem, bem como valores da distorção de intermodulação de segunda e terceira ordem. A avaliação da distorção harmônica é feita de modo semelhante à análise de ruído, considerando-se cada elemento não-linear como fonte geradora de distorção. A distorção total é obtida através da soma das componentes vetoriais de cada uma destas contribuições.

4. UTILIZAÇÃO DO PROGRAMA PSpice

Ao usuário acostumado com programas com interface gráfica, a maneira como se usa o programa PSpice pode parecer estranha. Devido a característica de cálculo por meio de manipulação de grandes matrizes numéricas, é normal que a simulação seja lenta, mesmo em microcomputadores equipados com rápidos microprocessadores. Apesar de que na maioria dos casos ter-se os resultados das simulações em poucos segundos, circuitos complexos podem demorar horas para serem simulados. A simulação envolve basicamente três passos:

- Descrição do circuito a ser simulado através de um arquivo tipo texto que contém uma lista dos elementos existentes (ramos do circuito) e seus respectivos nós de ligação. Este arquivo é conhecido como *netlist* e pode ser criado "manualmente" pelo usuário, a partir de algumas regras de sintaxe simples, ou gerado de modo automático através de um sistema de desenho do circuito. Neste trabalho, limita-se à descrição da criação "manual" do arquivo de *netlist*. No caso do programa PSpice, o próprio arquivo de *netlist* contém também comandos que descrevem o tipo de análise desejada.
- Simulação propriamente dita do circuito, realizada informando-se ao programa PSpice o arquivo de *netlist* criado anteriormente. Durante a simulação, o programa gera um arquivo texto contendo mensagens para o usuário e, não existindo erros na descrição do circuito é também gerado um arquivo de resultados em formato adequado para ser lido pelo programa PROBE.
- Análise dos resultados, através do programa PROBE, na forma de gráficos de tensões nos nós e correntes que percorrem os ramos do circuito.

4.1. DESCRIÇÃO DO CIRCUITO OU NETLIST

Para a criação do arquivo de descrição do circuito o usuário deve inicialmente numerar todos os nós do circuito, sendo obrigatória a presença de um nó "zero", que será considerado como **nó de referência** (usualmente o nó de "terra" do circuito), ao qual serão referenciadas todas as tensões calculadas. As versões mais recentes do SPICE, codificadas em linguagem "C", aceitam que os nós recebam **nomes** em vez de somente **números**, mas mesmo assim um dos nós deve ser o "zero". O mesmo ocorre com o programa PSpice. Cada elemento do circuito, por sua vez, deve ter um **nome único** que o identifique, sendo que a primeira letra desse nome especificará o **tipo** de elemento. A sintaxe geral da descrição dos elementos de circuito é:

<nome-do-elemento> <nó+> <nó-> <valor>

A versão original do SPICE aceitava somente os elementos relacionados na tabela seguinte:

Letra de referência	Elemento	Exemplos
R	Resistor	R1, Rload
C	Capacitor	C2, Cac
L	Indutor	L3, Lfiltro
K	Indutância Mútua	K4, Kmutual
V	Fonte de Tensão Independente	V1, Vbias
I	Fonte de Corrente Independente	I1, Igate
G	Fonte de Corrente controlada por tensão (fonte linear)	G1, Gcon
N	Fonte de Corrente controlada por tensão (fonte não-linear)	N4, Nsis
D	Diodo	D1, Da
Q	Transistor Bipolar	Q1, Q2N3055
J	J-FET	J5, Jbxt
M	MOSFET	M6, Mout

Versões posteriores incluíram outros dispositivos:

Letra de referência	Elemento	Exemplos
E	Fonte de tensão controlada por tensão	E1, Eload
H	Fonte de corrente controlada por corrente	H3, Hf
B	FET de arsenieto de gálio (MESFETs)	B1, Bb
T	Linha de transmissão sem perdas	G1, Gcon
S	Chave controladas por tensão	S4, Ssis
W	Chave controladas por corrente	W9, Wb

Os programas SPICE e PSpice não diferenciam letras maiúsculas de minúsculas. Assim, os nomes

Rentrada e **rEnTrAdA**

referem-se ao mesmo resistor. O programa original SPICE limitava o nome dos elementos a oito caracteres. Não há qualquer limitação neste aspecto, no caso de versões mais recentes do programa e também no programa PSpice. É óbvio, porém, que nomes extremamente longos para os elementos podem ser confusos e implicarão em maior trabalho de digitação por parte do operador.

No caso de resistores, capacitores e indutores, raramente o usuário saberá de antemão qual dos nós é o positivo. Tal desconhecimento, no entanto, não terá importância, pois se forem usados os nós invertidos a única consequência será a obtenção da tensão neste elemento com sinal negativo. Já no caso de fontes, deve-se tomar cuidado, assinalando o **nó positivo** ao **polo positivo** das fontes de tensão. Para as fontes de corrente, a corrente "sai" pelo nó negativo!

O valor do elemento é dado na unidade do SI correspondente. Valores fracionários são fornecidos com ponto decimal (e não vírgula!). Potências de dez podem ser fornecidas com a letra **E** seguida do número representando o expoente de dez.

Assim, um resistor R1 de 1000 ohms ligado entre os nós 4 e 5 de um circuito pode ser representado indiferentemente por um dos seguintes comandos:

```
R1 4 5 1000
ou: R1 4 5 1000.0
ou: R1 4 5 1E3
ou: R1 4 5 1.0E+3
```

O programa SPICE permite também o uso de sufixos representando os múltiplos e sub-múltiplos do SI. Como não há diferenciação entre letras maiúsculas e minúsculas, os sufixos não correspondem exatamente aos padronizados no SI. Na tabela seguinte acham-se os sufixos do programa SPICE referentes às potências de dez:

Potência de dez	Nome SI	Sufixo do SPICE
-15	femto	f ou F
-12	pico	p ou P
-9	nano	n ou N
-6	micro	u ou U
-3	mili	m ou M
+3	kilo	k ou K
+6	mega	meg ou MEG
+9	giga	g ou G
+12	tera	t ou T

Observar que a letra **M** significa *mili* (10^{-3}) e não *mega*! As letras que seguirem os sufixos são ignoradas. Isso é útil para tornar o arquivo mais legível. Assim, um capacitor C2 de 10 pF ligado entre os nós 7 e 12 pode ser descrito de várias maneiras:

```
C2 7 12 10p
ou C2 7 12 10pF
ou C2 7 12 10picoFarads
ou C2 7 12 10E-12
```

Observar que, se o capacitor C2 fosse descrito como:

```
c2 7 12 10E-12Farads
```

seria cometido um grave erro de descrição, pois a letra **F** da palavra **Farad** seria interpretada como o sufixo *femto* (10^{-15})!

Na descrição do circuito muitas vezes é conveniente incluir comentários. Existem duas maneiras de serem inseridos comentários. Uma linha inteira pode ser um comentário, se digitarmos como primeiro caráter de tal linha um asterisco. Por exemplo:

```
*Esta é uma linha de comentário
```

Pode-se também iniciar um comentário em uma linha qualquer, usando-se um ";" (ponto-e-vírgula). Tudo o que seguir o símbolo ";" nesta linha será ignorado, como:

```
R5 40 41 23.5ohms ; Resistor de polarização negativa
```

Cada linha do arquivo de descrição do circuito pode ter até 80 caracteres. Em caso de necessidade, uma linha pode ter continuações nas linhas seguintes, usando-se o sinal de soma na primeira posição das linhas-de-continuação. Por exemplo:

```
L4 1 2 5mH ; O indutor L4 tem o valor de 5 milihenrys
+ e está conectado entre os nós 1 e 2 do meu circuito
+ de teste
```

Encontra-se na Figura 4.1 um circuito simples, contendo apenas uma fonte de tensão e dois resistores.

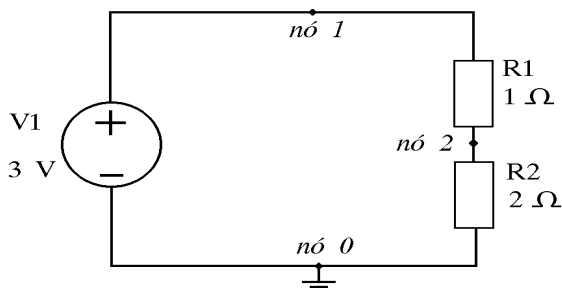


Figura 4.1:
Circuito simples para exemplo.

A descrição do circuito da Figura 4.1 para o programa SPICE é:

```
Exemplo 1 (Figura 4.1)
V1 1 0 3.0
R1 1 2 1.0
R2 2 0 2.0
```

A primeira linha contém o título ou descrição do circuito e será sempre considerada como comentário. Observar na Figura 4.1 que o nó zero é o “terra” do circuito. Aos outros dois nós foram dados os números 1 e 2 (**poderia ser qualquer número!**). Tem-se então os seguintes elementos descritos:

- Resistor **R1**, de 1 Ω, ligado entre os nós 1 e 2;
- Resistor **R2**, de 2 Ω, ligado entre os nós 2 e 0;
- Fonte de Tensão **V1**, de 3 volts, com o “polo positivo” ligado ao nó 1 e o “polo negativo” ligado no nó 0.

Outro exemplo, de um circuito em ponte, pode ser visto na Figura 4.2. Nesse circuito, como não se sabia de antemão qual é o nó de “terra”, arbitrou-se que o nó zero é aquele no qual acha-se conectado o terminal negativo da fonte. O arquivo de descrição será então:

```
Circuito em ponte (Figura 4.2)
Vin 1 0 10
R1 1 2 2
R2 1 3 1
R3 2 0 1
R4 3 0 2
R5 3 2 2
```

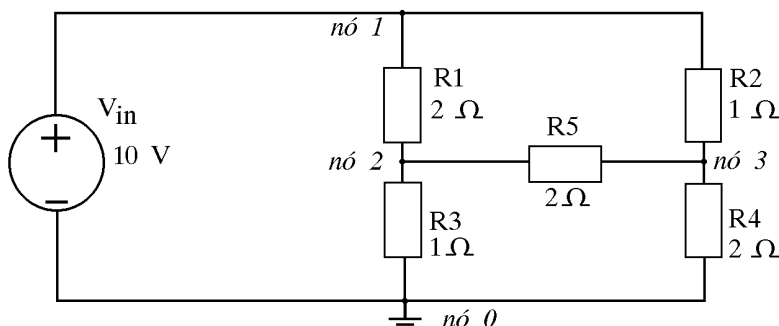


Figura 4.2:
Circuito em ponte.

Observe que optou-se por não digitar os nomes das unidades. O programa SPICE identifica automaticamente que o valor **10** da fonte de tensão **Vin** é dado em *volts* e que os valores dos resistores estão em *ohms* (Ω).

4.2. PRINCIPAIS COMANDOS

Além da descrição do circuito, o arquivo de entrada do programa SPICE deve informar o tipo de análise a ser feita e as informações que são desejadas.

.OP

Comanda o cálculo do "ponto de operação" do circuito. Se incluímos este comando, o programa SPICE produz uma tabela onde constam as tensões em todos os nós e as correntes e potências dissipadas em cada uma das fontes de tensão do circuito. A convenção usada no SPICE assume que a corrente *positiva* vai do nó(+) para o nó(-). Desta forma, as fontes de tensão ativas vão ter suas correntes com o sinal **NEGATIVO**, pois na verdade a corrente estará saindo pelo nó(+) da fonte.

.DC <fonte> <Vinicial> <Vfinal> <passo>

Este comando produz uma "varredura" ou variação da tensão de uma fonte de tensão especificada <fonte> desde um valor inicial <Vinicial> até <Vfinal> com incrementos sucessivos <passo>. Por exemplo, supondo no circuito da Figura 4.1 anterior que a fonte V1 deve variar de 0 a 3 volts, em incrementos ou "passos" de 0,1 volts. Usa-se o comando:

```
.DC V1 0 3 0.1
```

.PRINT DC <lista>

Este comando produz uma tabela de valores para a varredura solicitada no comando **.DC**. Utiliza-se a letra *I* para correntes e *V* para tensões. Alguns exemplos:

```
.PRINT DC V(7)      ; lista a tensão no nó 7;
.PRINT DC V(3,6)   ; lista a diferença de potencial entre os nós 3 e 6;
.PRINT DC V(R1)    ; lista a tensão no resistor R1;
.PRINT DC I(R4)    ; lista a corrente através do resistor R4;
```

Se necessário, é possível produzir uma tabela única - com tensões e correntes, como por exemplo:

```
.PRINT DC V(3) V(4) I(R2)
```

.OPTIONS

Este comando permite escolher uma série de opções do SPICE. As principais delas são:

LIST	produz uma lista de elementos existentes no circuito;
NODE	produz uma lista de nós do circuito;
NOECHO	suprime a listagem do arquivo de entrada no arquivo de saída;
NOPAGE	suprime paginação e cabeçalhos no arquivo de saída;
NUMDGT=	especifica número de dígitos dos resultados. O número máximo é 8 dígitos significativos e o valor <i>default</i> é 4 dígitos.
WIDTH=	especifica o número de colunas do formulário de impressão do arquivo de saída. O número máximo é 132 e o <i>default</i> é 80.

.PLOT

Este comando produz no arquivo de saída um "gráfico" simplificado de valores de corrente e/ou tensão, obtidos a partir do comando **.DC**. Este comando é remanescente das primeiras versões do programa, feitas numa época em que as impressoras não tinham capacidade gráfica e é raramente usado atualmente

.TEMP <valor>

Este comando especifica a temperatura de operação do circuito. Isto é útil para verificar o funcionamento de um circuito em temperaturas muito altas ou muito baixas, situações onde podem ocorrer variações de comportamento dos componentes. No caso geral, os resistores são considerados ideais, ou seja, suas resistências não variam com a temperatura. Se não existir o comando **.TEMP**, a análise é feita na temperatura de 27 °C.

.PROBE

Este comando não existe no programa SPICE de Berkeley, sendo específico do programa PSpice para microcomputadores. 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.

.END

Este comando é obrigatório na última linha do arquivo de entrada. Se o arquivo de entrada só contiver a descrição do circuito e o comando **.END**, o programa SPICE produz um arquivo de saída onde só constará uma lista das tensões em todos os nós.

4.3. EXEMPLO DE MODELAGEM: RESISTORES

Uma das principais características do programa SPICE é a possibilidade do usuário definir o comportamento dos elementos do circuito, através da especificação dos parâmetros dos modelos matemáticos utilizados na montagem do sistema de equações. Para exemplificar a característica de modelagem do programa SPICE, apresenta-se a seguir um exemplo referente a resistores.

Conforme já foi citado, o programa SPICE considera normalmente que todos os resistores são ideais, ou seja, que não sofrem qualquer variação de resistência com a temperatura. Sabe-se, no entanto, que isso não corresponde com a realidade: os resistores feitos com fios ou filmes metálicos *umentam* sua resistência com o aumento da temperatura, enquanto que os de grafite ou filme de carbono *diminuem* a resistência com a aumento da temperatura. A variação de resistência com a temperatura não ocorre de modo linear, podendo ser analisada segundo a expressão {1}:

$$R_t = R_{nom} \cdot [1 + TC_1 \cdot (T - T_{nom}) + TC_2 \cdot (T - T_{nom})^2] \quad \{1\}$$

Na expressão anterior tem-se:

R_t	=	Resistência na temperatura T
R_{nom}	=	Resistência nominal, à temperatura T_{nom}
TC_1	=	Coeficiente linear de temperatura do material
TC_2	=	Coeficiente quadrático de temperatura do material

Para resistores feitos com materiais metálicos, tanto TC_1 como TC_2 são positivos, enquanto que para resistores de grafite estes coeficientes são negativos.

A expressão {1} está “embutida” no programa SPICE. Assim, pode-se simular resistores com comportamento semelhante ao dos resistores reais. Por exemplo, suponha-se que estão sendo usados resistores do “tipo” XYZ, que possuem a seguinte característica:

$$TC_1 = +0,02 / ^\circ C$$

$$TC_2 = +0,005 / (^\circ C)^2$$

Para “informar” as características destes resistores, inclui-se o comando **.MODEL** no arquivo de entrada do programa SPICE, da seguinte forma:

```
.MODEL XYZ RES (R=1 TC1=0.02 TC2=0.005)
```

Os resistores que seguirem este “modelo” devem ser referenciados como tal na descrição do circuito. Supondo que os resistores R1 e R2 do circuito da Figura 4.1 sejam ambos do “tipo” XYZ, pode-se agora completar a descrição do circuito com as seguintes linhas:

```
V1 1 0 3volts
R1 1 2 XYZ 1ohm
R2 2 0 XYZ 2ohms
```

Isto feito, se for repetida a análise, os resultados serão idênticos aos obtidos anteriormente. Porém, se for especificada uma temperatura diferente de 27 °C através do comando **.TEMP**, o programa SPICE inicialmente calculará os novos valores dos resistores nesta temperatura e só então fará a análise solicitada.

5. ANALISE DE CIRCUITOS RETIFICADORES

A utilização do programa SPICE para análise de circuitos retificadores simples com diodos permite apresentar duas características de grande importância no uso deste programa de simulação: o da modelagem de semicondutores e a análise transiente.

5.1. FONTES DE TENSÃO SENOIDAIS

Desejando-se analisar retificadores, é necessário inicialmente mostrar como especificar fontes de tensão alternada senoidal. A descrição deste tipo de fonte é feita de forma semelhante à de uma fonte de tensão contínua:

```
V<nome> <Nó+> <Nó->
```

Após <Nó-> escreve-se a expressão:

```
SIN ( <Voffset> <Vmax> <freq> <delay> <damp> <phase> )
```

Os parâmetros dessa expressão tem o seguinte significado:

<Voffset> Tensão de “offset” (volts).

<Vmax>	Amplitude ou valor “de pico” da tensão (volts).
<freq>	Frequência (hertz).
<delay>	Atraso inicial (segundos).
<damp>	Fator de amortecimento da oscilação (s ⁻¹).
<phase>	Defasamento em relação a t=0 (graus).

Considere-se como exemplo uma fonte AC senoidal de 60 Hz com 127 V de valor eficaz. Deve-se então inicialmente calcular o valor “de pico” da tensão:

$$V_{\max} = 127\sqrt{2} = 179,6 \text{ V}$$

Supondo que tal fonte esteja ligada entre os nós **0** e **2**, sua descrição para o programa SPICE será:

```
Vfonte 2 0 SIN (0 179.6 60 0 0 0)
```

Uma vez que especificou-se defasamento inicial nulo, a oscilação senoidal se dará de modo positivo no primeiro semi-ciclo.

5.2. MODELAGEM DE DIODOS

Ao contrário do que ocorre com resistores, capacitores e indutores, geralmente é inconveniente utilizar um “modelo ideal” no programa SPICE para os demais dispositivos. Assim, se o usuário deseja simular um circuito que contenha diodos, deve “informar” as características dos diodos que utilizará. Os parâmetros fornecidos são utilizados pelo programa SPICE para a montagem de um sistema de equações não-lineares que descrevem matematicamente o funcionamento do dispositivo. No caso de diodos, o programa SPICE utiliza como modelo matemático a equação de Shockley [4], que descreve o comportamento de uma fonte de corrente não-linear:

$$I_d = I_s \left\{ e^{\left(\frac{V_d}{nV_t} \right)} - 1 \right\}$$

onde:

I_d	é a corrente fornecida pela fonte-equivalente;
I_s	é a corrente de saturação do diodo;
V_d	é a tensão da junção;
V_t	é a tensão térmica ($k \cdot T/q$), sendo k a constante de Boltzmann, T a temperatura absoluta em kelvin e q a carga elétrica. À temperatura ambiente, tem-se para V_t o valor de 25,85 mV.

Para operação em polarização reversa, o programa SPICE considera cada diodo como uma pequena condutância, para simular a passagem de uma pequena corrente. O valor *default* usado como condutância reversa de todos os diodos do circuito é 10^{-12} siemens (1 pS) e pode ser alterado com o comando:

```
.OPTIONS GMIN <valor>
```

A forma geral de descrição de um diodo para o programa SPICE é:

```
D<nome> <nó-anodo> <nó-catodo> <nome-do-modelo>
```

Exemplo:

```
D1 2 3 D1N914
```

Devemos incluir no arquivo de descrição do circuito um comando **.MODEL** com os parâmetros necessários para que o programa SPICE estabeleça o sistema de equações que será utilizado para simular o diodo. A sintaxe do comando é:

```
.MODEL <nome-do-modelo> D <lista de parâmetros>
```

No comando **.MODEL**, a letra **D** indica que se trata de parâmetros para um modelo de diodo. Os parâmetros estão listados na tabela à seguir, sendo que sempre que se omite um parâmetro o programa SPICE considera tal parâmetro igual ao valor *default*.

Símbolo	Significado do Parâmetro	Default	Unidade
Is	Corrente de saturação	10 ⁻¹⁴	ampère
N	Coefficiente de emissão	1.0	

Rs	Resistência parasita	zero	ohm
CJO	Capacitância da junção PN com polarização nula	zero	farad
Vj	Tensão da junção	1.0	volt
M	Coefficiente do gradiente da junção	0.5	
FC	Coefficiente da capacitância de depleção em polarização direta	0.5	
Tt	Tempo de transição	zero	segundo
Bv	Tensão de ruptura reversa	infinito	volt
Ibv	Corrente de ruptura reversa	10 ⁻¹⁰	ampère
Eg	Barreira de potencial do material semiconductor utilizado (o valor <i>default</i> refere-se ao silício)	1.11	eV
Xti	Coefficiente de temperatura da corrente de saturação	3.0	
Kf	Coefficiente de ruído	zero	
Af	Expoente de ruído	1.0	

À título de exemplo, mostra-se a seguir a descrição do modelo SPICE de um diodo 1N4148.

```
.MODEL D1N4148 D (Is=0.1p Rs=16 CJO=2p Tt=12n Bv=100 Ibv=0.1p)
```

Quando se utiliza o programa PSpice para microcomputadores, pode-se ter o modelo já descrito em um arquivo ou “biblioteca” de modelos. Normalmente estes modelos estão em um arquivo em disco com a extensão *.LIB*. É comum aos projetistas que trabalham com o programa SPICE “coleccionarem” modelos, formando sua própria “biblioteca”. Nesta hipótese, supondo que o usuário possui num diretório **C:\PSPICE\MODELOS** em seu sistema um arquivo de nome *DIODOS.LIB* que contém modelos pré-definidos de diodos, o arquivo de descrição do circuito deverá incluir um comando:

```
.LIB C:\PSPICE\MODELOS\DIODOS.LIB
```

5.3. ANÁLISE TRANSIENTE

No caso de retificadores, é de se supor que haverá interesse em obter curvas que mostrem o funcionamento do circuito ao longo de um certo intervalo de tempo. Conforme citado anteriormente, no programa SPICE a observação do comportamento do circuito no domínio do tempo é feita através da análise transiente. Para que o programa realize esta análise, deve-se incluir na descrição do circuito o seguinte comando:

```
.TRAN <passo> <tempo-final>
```

Neste comando, *<passo>* refere-se ao intervalo de tempo em segundos usado para imprimir ou “plotar” os resultados da análise. Usando como exemplo retificadores que funcionam com 60 Hz, pode-se utilizar para *<passo>* o valor 5.0 ms (cinco mili-segundos). Quanto menor o valor de *<passo>*, mais pontos serão gerados pelo programa, o que melhora a resolução das respostas porém aumenta o tempo de processamento. Deve-se ressaltar que tal “passo” refere-se apenas à saída de resultados; internamente o programa utiliza um “passo de cálculo” variável, de modo a atingir a precisão desejada. O parâmetro *<tempo-final>* estabelece que a análise será feita desde *TEMPO=0* até *TEMPO=<tempo-final>*. Para retificadores de 60 Hz, um intervalo de tempo de 100 ms permitirá a visualização de seis ciclos completos. Assim, à título de exemplo, pode-se usar o comando:

```
.TRAN 5m 100m
```

5.4. EXEMPLO DE UM RETIFICADOR DE ONDA COMPLETA

Com o intuito de exemplificar o uso de modelos de diodos e da análise de transientes, será mostrada a simulação de um circuito retificador de onda completa com um transformador do tipo *center tap*, conforme a Figura 5.1.

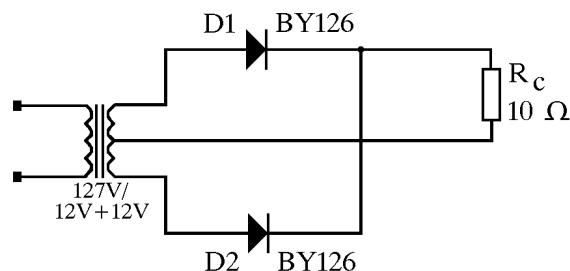


Figura 5.1:
Retificador de onda completa
com transformador "center tap".

Os diodos indicados na Figura 5.1 como *D1* e *D2* são do tipo BY126, para os quais supõe-se os seguintes parâmetros:

$$\begin{aligned} I_s &= 0,5 \mu\text{A} \\ R_s &= 0,2 \Omega \\ C_{JO} &= 2,0 \text{ pF} \\ T_t &= 50 \text{ ns} \\ B_v &= 450 \text{ V} \\ I_{bv} &= 100 \text{ nA} \end{aligned}$$

Para os demais parâmetros dos diodos BY126 serão utilizados os valores *default*. Com o intuito de simplificação, considera-se neste exemplo que o transformador se comporta como fonte senoidal perfeita, fornecendo tensão eficaz de 12 volts a cada um dos terminais do secundário, medida em relação ao *center tap*. É necessário portanto obter o valor “de pico” da tensão fornecida pelo transformador:

$$V_{\max} = 12\sqrt{2} = 16,97 \text{ V}$$

Isso posto, o circuito equivalente, que será objeto da simulação, é mostrado na Figura 5.2.

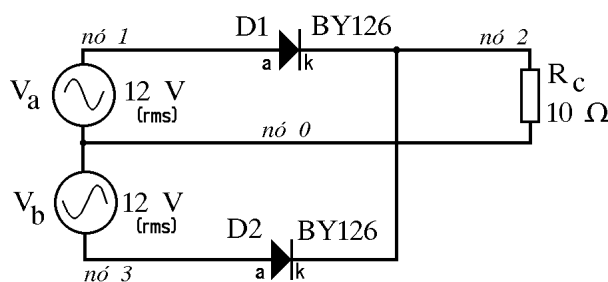


Figura 5.2:
Circuito do retificador de onda completa com transformador "center tap", simplificado para a simulação.

O arquivo de descrição do circuito da Figura 5.2, para uso com o programa PSpice, é:

```
Retificador AC/DC de onda completa:
Va 1 0 SIN(0 16.97 60 0 0 0)
Vb 3 0 SIN(0 -16.97 60 0 0 0)
D1 1 2 DBY126
D2 3 2 DBY126
Rc 2 0 10
.MODEL DBY126 D (Is=0.5u Rs=0.2 CJO=2.0p Tt=50n Bv=450 Ibv=100n)
.TRAN 5.0m 100m
.PROBE
.END
```

Há duas maneiras para especificar que a “fonte” V_b está defasada 180° em relação à “fonte” V_a . No exemplo, a amplitude foi especificada com um valor negativo. A segunda opção seria especificar a amplitude positiva e o valor 180 para o parâmetro `<phase>` da “fonte” V_b .

Encontra-se no Anexo 2 os resultados obtidos com a simulação do circuito da Figura 5.2, através do programa PSpice. As curvas foram obtidas com o programa PROBE, mostrando a forma-de-onda de tensão sobre a carga R_c .

6. ANÁLISE DE CIRCUITOS COM TRANSISTORES BIPOLARES

O programa SPICE permite a simulação de circuitos que empregam transistores bipolares e de “efeito de campo” (*J-FET*, *MOSFET* e *MESFET*). Para isso deve-se incluir, no arquivo de descrição do circuito, uma série de parâmetros elétricos referentes aos dispositivos utilizados, parâmetros estes que serão empregados pelo programa SPICE para modelar matematicamente os transistores existentes no circuito. A introdução dos parâmetros é feita através do comando `.MODEL`, de modo semelhante ao que foi descrito anteriormente para diodos. Os parâmetros que não forem fornecidos serão substituídos por valores *default*.

Apresenta-se à seguir alguns aspectos da utilização de transistores bipolares (NPN ou PNP) em circuitos a serem simulados com o programa SPICE.

6.1. PARÂMETROS DE TRANSISTORES BIPOLARES

O programa SPICE considera cada transistor bipolar existente no circuito como um circuito equivalente, formado por fontes de corrente, resistores e capacitores, regidos por um conjunto de equações não-lineares. É utilizado o modelo de Gummel-Poon [7,8] para transistores bipolares, com algumas modificações. Os parâmetros SPICE referentes a tal modelo de transistores bipolares acham-se relacionados no Anexo 1.

Conforme se pode verificar, a lista de parâmetros é bastante extensa. Para transistores operando com pequenos sinais em baixas frequências, pode-se utilizar os valores *default* dos parâmetros, sem grande prejuízo dos resultados da simulação. Já no caso de transistores usados em circuitos mais “complexos” (chaveamento em altas frequências, por exemplo), é necessário ter-se parâmetros mais próximos do dispositivo real. Alguns dos parâmetros podem ser obtidos a partir das especificações técnicas dos transistores, encontradas nos manuais dos fabricantes (*Databooks*). Outros parâmetros, no entanto, normalmente não são fornecidos pelos fabricantes, devendo ser obtidos através de testes e medições, diretamente sobre amostras do dispositivo em pauta. A obtenção experimental dos parâmetros de dispositivos semicondutores é bastante complexa [7]; no entanto, devido a grande divulgação do programa SPICE entre os projetistas de sistemas eletrônicos, muitos fabricantes já estão fornecendo os parâmetros SPICE dos dispositivos que comercializam. Nos casos em que os fabricantes não fornecem (**ainda!**) os parâmetros SPICE, uma outra alternativa é utilizar-se os parâmetros de transistores equivalentes, alterando-se alguns dos valores de modo a buscar uma melhor aproximação com o dispositivo à ser usado. Um dos parâmetros principais, que deve ser tão próximo quanto possível do valor real do transistor à ser usado no circuito real, é o ganho estático em corrente direta, conhecido como *beta* (β) do transistor. Nos parâmetros do programa SPICE, este parâmetro é designado como **BF** (*Beta forward*).

6.2. SINTAXE DOS COMANDOS PARA TRANSISTORES BIPOLARES

A descrição de transistores bipolares obedece a seguinte sintaxe:

```
Q<nome> <nó-coletor> <nó-base> <nó-emissor> <nome-do-modelo>
```

O *<nome-do-modelo>* deve estar em concordância com um comando **.MODEL** correspondente, onde são fornecidos os valores dos parâmetros do modelo. Por exemplo, se em um circuito existir um transistor **Q1**, do tipo **BC549** ligado nos nós 4 (coletor), 5 (base) e 6 (emissor), sua descrição será:

```
Q1 4 5 6 BC549
```

Os parâmetros necessários para a montagem do modelo de Gummel-Poon correspondente devem ser fornecidos com o comando **.MODEL**, da forma:

```
.MODEL <nome-do-modelo> <polaridade> ( <lista de parâmetros> )
```

Neste comando, *<polaridade>* será **NPN** ou **PNP**, conforme o transistor em pauta. Para o exemplo citado, teremos:

```
.MODEL BC549 NPN (BF=500 RE=1 IS=1p CJC=3p VJE=0.5 RC=2)
```

Os parâmetros que não forem fornecidos valores serão substituídos pelos seus valores *default*, conforme listados no Anexo 1.

6.3. EXEMPLO PARA SIMULAÇÃO

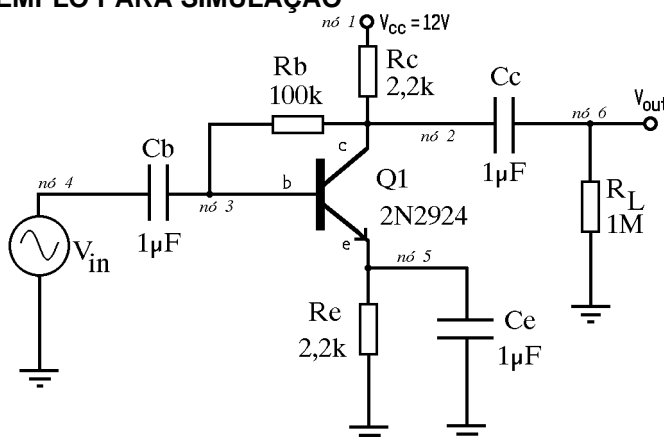


Figura 6.1:
Amplificador com
um transistor.

Com o objetivo de demonstrar a simulação de circuitos com transistores bipolares, encontra-se na Figura 6.1 a representação de um circuito amplificador com um transistor NPN. Deseja-se verificar o sinal que se obtém no resistor R_L , quando se fornece um sinal de entrada senoidal de 100 kHz, com 1 mV de amplitude. Deseja-se também saber o consumo de potência do circuito no ponto quiescente, ou seja, na situação na qual o circuito acha-se com sua tensão de alimentação normal, porém sem qualquer sinal presente na entrada.

As condições do circuito no ponto quiescente são obtidas através do comando `.OP` do programa SPICE. A operação com o sinal de entrada (V_{in}) de amplitude igual a 1 mV e frequência de 100 kHz é obtida através da análise transiente. Para o comando `.TRAN` do programa SPICE, um intervalo de tempo de 30 μ s é suficiente para se ter uma boa visualização dos sinais de entrada e saída. Da mesma forma, escolheu-se um "passo" de simulação igual a 100 ns; um "passo" menor dará curvas mais "suaves", porém o tempo de processamento será conseqüentemente aumentado. Para o modelamento do transistor 2N2924, recorreu-se ao catálogo de um fabricante (no caso, *National Semiconductor*), onde são fornecidos os parâmetros já adequados para os programas SPICE e PSpice. O arquivo que descreve o circuito da Figura 6.1 e fornece os comandos ao programa PSpice é:

Exemplo com Transistor Bipolar

```
Vcc 1 0 12volts
Rc 1 2 2.2k
Rb 2 3 100k
Re 5 0 2.2k
Rl 6 0 1MEG
Cb 3 4 1u
Ce 5 0 1u
Cc 2 6 1u
Vin 4 0 SIN(0 1m 100k 0 0)
Q1 2 3 5 Q2N2924
.MODEL Q2N2924 NPN(Is=12.03f Xti=3 Eg=1.11 Vaf=37.37 Bf=295.7 Ne=1.971
+ Ise=4.272p Ikf=.1072 Xtb=1.5 Br=4.379 Nc=2 Isc=0 Ikr=0 Rc=1
+ Cjc=5.777p Mjc=.3199 Vjc=.75 Fc=.5 Cje=8.307p Mje=.384
+ Vje=.75 Tr=721.6p Tf=385.4p Itf=.17 Vtf=3 Xtf=8 Rb=10)
*Fonte: Catálogo da National Semiconductor Corporation
.OP
.TRAN 100n 30u
.OPTIONS NOPAGE LIST NODE
.PROBE
.END
```

No Anexo 2 são mostradas as curvas obtidas com o programa PROBE, onde verifica-se o sinal de entrada (V_{in}) em comparação com a tensão sobre o resistor de carga R_L . Verifica-se com isso que o circuito tem desempenho conforme previsto.

7. CONCLUSÕES

A utilização de um sistema computacional para análise de circuitos representa notáveis vantagens sob o ponto de vista de facilidade de projeto e avaliação de desempenho dos circuitos eletrônicos. No caso específico do projeto de circuitos integrados, a simulação é uma das mais importantes ferramentas de que dispõe os projetistas. Esta ferramenta pode ser utilizada, no entanto, para a simulação de qualquer circuito eletrônico.

O programa SPICE, devido à qualidade dos modelos empregados e contínuos aperfeiçoamentos ocorridos desde sua apresentação, vem se firmando como um padrão para a simulação de circuitos, sendo atualmente largamente utilizado tanto pelos estudantes que estão iniciando-se na atividade de projeto de circuitos eletrônicos como pelos projetistas de circuitos avançados.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CARTER, Harold W. Computer-aided design of integrated circuits. IN: IEEE Computer Magazine, April 1986, p. 19-36.
- [2] HMURCIK, Lawrence V.; HETTINGER, Mathias; GOTTSCHALCK, Kenneth S.; FITCHER, Franklin C. SPICE applications to an undergraduate electronics program. IN: IEEE Transactions on Education, vol. 33, nº 2 (May 1990), p. 183-189.
- [3] McCALLA, William J. & PEDERSON, Donald O. Elements of computer-aided circuits analysis. IN: IEEE Transactions on Circuit Theory, vol. 18, nº 1 (January 1971), p. 14-25.
- [4] NAGEL, Laurence W. SPICE2; a computer program to simulate semiconductor circuits. Berkeley, University of Califórnia, 1975. 233p. (Thesis ERL-M520).
- [5] PEDERSON, Donald O. A historical review of circuit simulation. IN: IEEE Transactions on Circuits and Systems, vol. 31, nº 1 (January 1984), p. 103-111.
- [6] TUINEGA, Paul W. A guide to circuit simulation and analysis using PSpice. New Jersey, Prentice Hall, 1988. 200p.
- [7] GETREU, Ian E. Modeling the bipolar transistor. Amsterdam, Elsevier, 1978. 261 p.
- [8] ANTOGNETTI, Paolo & MASSOBRIO, Giuseppe Semiconductor device modeling with SPICE. New York, McGraw-Hill, 1988. 389 p.
- [9] MEHL, Ewaldo Luiz de Mattos Experiência da UFPR no uso do programa SPICE para o ensino de engenharia elétrica. Trabalho apresentado no 10º Congresso Brasileiro de Ensino de Engenharia - João Pessoa, Novembro de 1991.

NOTA DE "COPYRIGHT":

PSpice e *PROBE* são marcas registradas da **MicroSim Corporation** (20 Fairbanks, Irvine, California 92718, E.U.A.). As simulações citadas no texto foram obtidas com as "versões estudantis" dos citados programas, que podem ser copiadas e utilizadas livremente para fins didáticos.

ELMM
e:\documents\pspice\apostilapspice.doc
353 kB
7/04/99 18:07:00h

ANEXO 1: Parâmetros SPICE dos Transistores Bipolares
(Modelo de Gummel-Poon Modificado)

Símbolo	Significado do parâmetro	Unidade	Default	Exemplo	
1	IS	transport saturation current	A	1.0E-1 A	1.0E-15
2	BF	ideal maximum forward beta	-	100	100
3	NF	forward current emission coefficient	-	1.	1
4	VAF	forward Early voltage	V	infinite	200
5	IKF	corner for forward beta high-current roll-off	A	infinite	0.01
6	ISE	Base-Emitter leakage saturation current	A	0	1.0E-13
7	NE	Base-Emitter leakage emission coefficient	-	1.5	2
8	BR	ideal maximum reverse beta	-	1	0.1
9	NR	reverse current emission coefficient	-	1	1
10	VAR	reverse Early voltage	V	infinite	200
11	IKR	corner for reverse beta high-current roll-off	A	infinite	0.01
12	ISC	Base-Collector leakage saturation current	A	0	1.0E-13
13	NC	Base-Collector leakage emission coefficient	-	2	1.5
14	RB	zero bias base resistance (maximum)	Ω	0	100
15	IRB	current where base resistance falls halfway to its minimum value	A	infinite	0.1
16	RBM	minimum base resistance at high currents	Ω	RB	10
17	RE	emitter ohmic resistance	Ω	0	1
18	RC	collector ohmic resistance	Ω	0	10
19	CJE	Base-Emitter zero-bias depletion capacitance	F	0	2 p
20	VJE	Base-Emitter built-in potential	V	0.75 V	0.6
21	MJE	Base-Emitter junction exponential factor	-	0.33	0.33
22	TF	ideal forward transit time	s	0	0.1 n
23	XTF	coefficient for bias dependance of TF	-	0	
24	VTF	voltage describing VBC dependence of TF	V	infinite	
25	ITF	high-current parameter for effect on TF	A	0	
26	PTF	excess phase at freq=1.0/(TF*2p) Hz	deg	0	
27	CJC	Base-Collector zero-bias depletion capacitance	F	0	2 p
28	VJC	Base-Collector built-in potential	V	0.75 V	0.5
29	MJC	Base-Collector junction exponential factor	-	0.33	0.5
30	XCJC	fraction of Base-Collector depletion capacitance connected to internal base node	-	1	
31	TR	ideal reverse transit time	s	0	10 n
32	CJS	zero-bias collector-substrate capacitance	F	0	2 p
33	VJS	substrate junction built-in potential	V	0.75	
34	MJS	substrate junction exponential factor	-	0	0.5
35	XTB	forward and reverse beta temperature exponent	-	0	
36	EG	energy gap for temperature effect on IS	eV	1.11	
37	XTI	temperature exponent or effect on IS	-	3	
38	KF	flicker-noise coefficient	-	0	
39	AF	flicker-noise exponent	-	1	
40	FC	coefficient for forward-bias depletion capacitance formula	-	0.5	

ANEXO 2: Resultados de Simulações

A2.1: Simulação do Retificador de Onda Completa (Circuito da Figura 5.2):

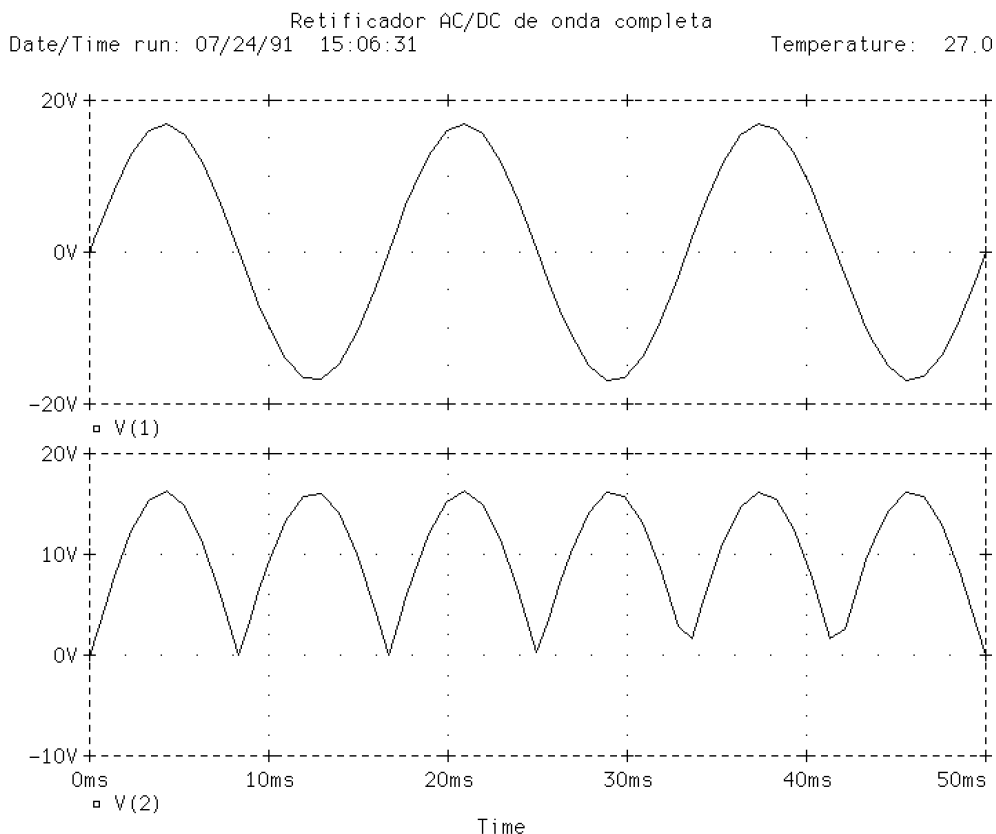


Figura A2.1: Sinais presentes nos nós 1 e 2 do circuito retificador de onda completa.

A2.2: Simulação do Amplificador com um Transistor (Circuito da Figura 6.1):

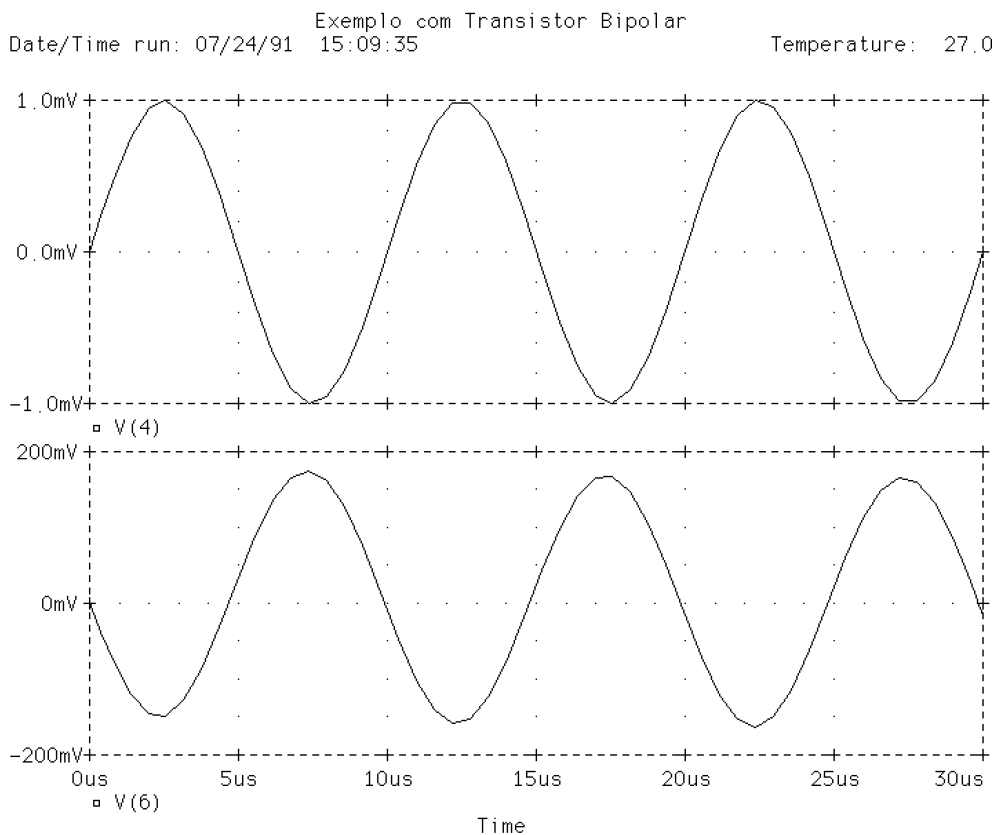


Figura A2.2: Sinais presentes nos nós 4 e 6 do circuito amplificador com um transistor.