

Passo 1: Preparar o Ambiente no Terminal (FAZER SEMPRE!)

Toda vez que você iniciar uma NOVA sessão de terminal (abrir o Mobaxterm ou uma nova aba), você PRECISA carregar o ambiente Cadence. Isso "ensina" o terminal onde encontrar os comandos como 'spectre' e 'virtuoso'.

1.1. Abra seu terminal (Mobaxterm) e conecte-se ao servidor via SSH.

1.2. Execute o comando de setup específico da sua instituição:

```
source ~/cadence/gpdk045/cds
```

1.3. (Opcional, mas recomendado) Verifique se o comando 'spectre' foi encontrado: `which spectre`, Isso deve retornar o caminho para o executável do Spectre, confirmando que o ambiente foi carregado corretamente.

Passo 2: Criar o Ficheiro Netlist (O "Código" do Circuito)

O netlist é um ficheiro de texto que descreve tudo sobre a sua simulação: o circuito, os modelos dos componentes, e as análises a serem feitas. A extensão padrão para netlists Spectre é '.scs'.

2.1. Navegue até a pasta onde você quer guardar seus ficheiros de simulação:

```
cd ~/simulation (Se a pasta não existir, crie-a com 'mkdir ~/simulation' antes)
```

2.2. Use um editor de texto para criar um novo ficheiro. O 'nano' é simples:

```
nano rc_filtro.scs
```

2.3. Escreva ou cole o seguinte código Spectre dentro do editor 'nano'.

exemplo de netlist:

```
// Este ficheiro é um netlist Spectre completo que simula um componente
// de 4 portas (nport) lendo os seus dados de um ficheiro .s4p.
//
// Pré-requisito:
// Esta simulação PRECISA de um ficheiro de dados S-parameter para ler.
// Certifique-se de que o ficheiro "dummy.s4p" (criado anteriormente)
// está na MESMA pasta que este ficheiro.
//
// -----
//
// --- Declaração do Simulador ---
// Sempre comece declarando a linguagem que estamos a usar.
simulator lang=spectre
//
// --- Definição das Portas de Medição ---
// Para uma análise de Parâmetros S, precisamos de "portas" para
// injetar o sinal e medir a reflexão/transmissão.
// Elas correspondem aos componentes PORT0 a PORT3 do seu esquemático.
```

```

// A impedância de referência (r) é definida como 50 Ohms.
//
// NOTA IMPORTANTE: O parâmetro 'num' (número da porta) DEVE
// começar em 1 e ser um inteiro positivo.

// Porta 1 (ligada ao nó n_p1)
P1 (n_p1 0) port r=50 num=1

// Porta 2 (ligada ao nó n_p2)
P2 (n_p2 0) port r=50 num=2

// Porta 3 (ligada ao nó n_p3)
P3 (n_p3 0) port r=50 num=3

// Porta 4 (ligada ao nó n_p4)
P4 (n_p4 0) port r=50 num=4

// --- Instância do Componente (Device Under Test) ---
// Este é o componente NPORT que queremos analisar.
// O Spectre irá ler o seu comportamento do ficheiro especificado.
//
// Nome da instância: NPORT0
// Nós de conexão: (n_p1 n_p2 n_p3 n_p4) - A ordem DEVE corresponder
// à ordem das portas no ficheiro .s4p
// Tipo de componente: nport
// Ficheiro de dados: "dummy.s4p" (o ficheiro que você criou)
NPORT0 (n_p1 n_p2 n_p3 n_p4) nport file="dummy.s4p"

// --- Definição da Análise de Parâmetros S (sp) ---
// Esta é a análise que corresponde à configuração do seu ADE.
//
// Nome da análise: analiseSP
// Tipo de análise: sp
// Frequência inicial: start=1G (1 GHz)
// Frequência final: stop=10G (10 GHz)
// Tipo de varredura: dec=10 (10 pontos por década, logarítmico)
// Portas a analisar: ports=[P1 P2 P3 P4] (lista das portas que definimos)
//
// ***** COMANDO PARA SALVAR O FICHEIRO DE TEXTO *****
// O parâmetro 'file' diz ao Spectre para criar um ficheiro Touchstone
// (neste caso, .s4p) com os resultados da simulação.
analiseSP sp start=1G stop=10G dec=10 ports=[P1 P2 P3 P4]
file="resultados_simulacao.s4p"

// --- Fim do Netlist ---

```

2.4. Guarde o ficheiro e saia do editor 'nano':

Pressione Ctrl+X
Pressione Y (para Yes)
Pressione Enter (para confirmar o nome do ficheiro)

Varrer Parâmetros

Usando a análise sweep:

sweep: Defina a análise sweep, diga qual parâmetro varrer (param=...), de onde começar (start=...), onde parar (stop=...) e como dividir (lin=... ou dec=...)

Exemplo de Netlist (filtro_com_sweep.scs): Este código faz 10 simulações AC, varrendo Rvalor de 1k a 10k.

```
simulator lang=spectre
```

```
// --- 1. Definir o Parâmetro ---
```

```
parameters Rvalor=1k
```

```
// --- Circuito ---
```

```
Vinput (in 0) vsource dc=0 ac=1 type=dc
```

```
R1 (in out) resistor r=Rvalor // <-- Use o parâmetro aqui
```

```
C1 (out 0) capacitor c=1n
```

```
// --- 2. Definir o Bloco de Sweep ---
```

```
// Nome: sweep_resistencia
```

```
// Varrer: o parâmetro 'Rvalor'
```

```
// De: 1k (1000)
```

```
// Até: 10k (10000)
```

```
// Em: 10 passos lineares
```

```
sweep_resistencia sweep param=Rvalor start=1k stop=10k lin=10 {
```

```
// --- 3. Análise Aninhada ---  
  
// Esta análise 'ac' será executada 10 vezes,  
  
// uma para cada valor de Rvalor.  
  
analise_ac ac start=1 stop=1G dec=10  
  
} // Fim do bloco de sweep
```

Executar em Paralelo

não é um comando do Spectre, mas sim um truque do terminal Linux usando o & (operador de "background").

O '&' no final envia o comando para segundo plano e liberta o terminal

```
spectre filtro.scs &
```

```
spectre oscilador.scs &
```

```
spectre nport_teste.scs &
```

Passo 3: Executar a Simulação Spectre

Agora que o ambiente está carregado e o netlist está pronto, execute o Spectre.

3.1. Certifique-se de que você está na pasta correta ('~/simulation').

3.2. Execute o comando:

```
spectre rc_filtro.scs
```

3.3. Observe a saída no terminal. O Spectre mostrará informações sobre a licença, o progresso das análises (DC, AC) e, no final, um resumo com o número de erros e avisos. Se terminar com "0 errors", a simulação foi bem-sucedida!

Passo 4: Entender os Ficheiros de Saída

Após a execução, o Spectre cria alguns ficheiros e pastas importantes na sua diretoria atual ('~/simulation'):

4.1. Pasta de Resultados (.raw):

Uma nova pasta será criada, geralmente com o nome do netlist + '.raw'. Exemplo: rc_filtro.raw. Esta pasta contém os DADOS BINÁRIOS das formas de onda e resultados das análises, prontos para serem lidos por ferramentas gráficas.

4.2. Ficheiro de Log (.log ou .out):

Um ficheiro de texto (ex: rc_filtro.log ou spectre.out) é criado com TODAS as mensagens que o Spectre imprimiu no terminal, incluindo avisos detalhados, informações de convergência, etc. É útil para depuração.

4.3. (Opcional) Ficheiros de Texto Salvos Diretamente: Se você usou o parâmetro 'file=...' numa análise (para parâmetros salvamos como 'file="resultados.sXp"'), esse ficheiro de texto também será criado aqui.

Passo 5: Visualizar os Resultados Gráficos (Formas de Onda)

Para ver os gráficos (ex: a resposta em frequência do filtro RC), usamos a ferramenta ViVA (Virtuoso Visualization and Analysis).

5.1. No mesmo terminal onde o ambiente Cadence está carregado, inicie o ViVA: virtuoso -viva & O '&' executa em segundo plano, libertando o seu terminal.

5.2. A janela do ViVA irá abrir-se. Vá ao menu: File -> Open Results...

5.3. Navegue até a sua pasta '~/simulation' e SELECIONE A PASTA '.raw':
Exemplo: rc_filtro.raw
Clique em 'Open'.

5.4. O 'Results Browser' dentro do ViVA mostrará as análises disponíveis (ex: 'analiseDC', 'analiseAC'). Clique na análise desejada (ex: 'analiseAC').

5.5. Os sinais que você salvou (ou todos os nós, se não usou 'save') aparecerão. Clique duas vezes no sinal que quer plotar (ex: 'Vout').

5.6. O ViVA perguntará o formato. Para análise AC, escolha por exemplo 'Magnitude (dB20)', para ver o ganho em dB. O gráfico aparecerá.