ALMA MATER STUDIORUM – Università di Bologna, Facoltà di Ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Esercitazioni di CIRCUITI ELETTRONICI ANALOGICI L-A

Anno Accademico 2006/2007

Guida all'uso di PSpice®

1 Introduzione

SPICE è un potente e versatile simulatore circuitale, comunemente utilizzato per il progetto di circuiti elettrici ed elettronici e per la verifica del loro comportamento. Questa funzionalità è di particolare importanza nello sviluppo dei circuiti integrati. Fu per questo motivo che nel 1975 il Laboratorio di Ricerca Elettronica dell'Università della California di Berkley diede inizio al suo sviluppo. Il suo nome è l'acronimo di ***S**imulation **P**rogram for **I**ntegrated **C**ircuits **E**mphasis*.

PSpice® è la versione commerciale di SPICE, originariamente prodotta dalla Microsim™ Corporation ed oggi prodotto da Cadence™ e distribuito da EMA™. Nella sua versione attuale, PSpice® è capace di simulare complessi sistemi *mixed-mode*, ovvero contenenti parti sia analogiche che digitali. È in grado di gestire modelli di dispositivi avanzati come IGBT, modulatori di larghezza di impulso e convertitori A/D e D/A. Tramite l'uso di un'interfaccia grafica (GUI) è possibile visualizzare sia i valori numerici calcolati dal simulatore, sia l'andamento temporale di tensioni e correnti relative ai vari rami e nodi del circuito analizzato. Infine, grazie all'uso di funzioni matematiche, è possibile combinare le informazioni contenute nei segnali presenti sul circuito per valutare parametri come l'ampiezza di banda di filtri e la figura di rumore.

Presso il sito di Cadence è possibile scaricare gratuitamente una versione dimostrativa di PSpice come parte della suite di progettazione OrCAD. Questa versione, pur essendo limitata nel numero di nodi di cui può essere composto il circuito da simulare, è interamente funzionante e ampiamente sufficiente per lo scopo di queste esercitazioni. È possibile trovare altre versioni di PSpice®su internet: questa introduzione e le successive esercitazioni fanno riferimento alla versione 9.1 e successive.

2 Notazione

Nel manuale di PSpice® ed in queste esercitazioni verranno utilizzate le seguenti convenzioni di notazione:

Notazione	Esempio	Descrizione
`<>`	`<name>`	*Parametro obbligatorio*. È **sempre** necessario riempire il campo relativo a questo parametro con un singolo valore per permettere il corretto funzionamento del simulatore.
`<>*`	`<value>*`	*Lista di parametri obbligatori*. È **sempre** necessario riempire il campo relativo a questo parametro con uno o più valori per permettere il corretto funzionamento del simulatore.
`<|>`	`<ON|OFF>`	*Scelta di parametri obbligatori*. È **sempre** necessario riempire il campo relativo a questo parametro con un singolo valore a scelta tra quelli indicati per permettere il corretto funzionamento del simulatore.
`[]`	`[name]`	*Parametro opzionale*. È possibile riempire il campo relativo a questo parametro con un singolo valore per specificare la modalità di funzionamento del simulatore.
`[]*`	`[value]*`	*Lista di parametri opzionali*. È possibile riempire il campo relativo a questo parametro con uno o più valori per specificare la modalità di funzionamento del simulatore.
`[|]`	`[ON|OFF]`	*Scelta di parametri opzionali*. È possibile riempire il campo relativo a questo parametro con un singolo valore a scelta tra quelli indicati per specificare la modalità di funzionamento del simulatore.
`*`		*Delimitatore di linea commentata*. Posto come primo carattere di una linea, indica al simulatore di trattare l'intera linea come un commento testuale e di escluderla dai comandi di simulazione.
`;`		*Delimitatore di commento*. Posto all'interno di una linea, indica al simulatore di trattare tutto il testo che si trova alla destra come un commento testuale e di escluderlo dai comandi di simulazione.
`+`		*Delimitatore di linea lunga*. Posto come primo carattere di una linea, indica al simulatore di trattare l'intera linea come proseguimento della linea precedente.

3 Sintassi

Per specificare la topologia del circuito ed il tipo di analisi che PSpice® dovrà effettuare, è necessario fornire in ingresso al simulatore un file di testo con estensione `.cir` contenente la lista degli elementi presenti nel circuito, la loro connettività, i loro parametri ed i comandi di configurazione della simulazione.

Questa lista, unitamente al file che la contiene, viene detta *netlist*. Esistono programmi che permettono di creare la netlist per via grafica, connettendo opportunamente i simboli dei vari componenti elettrici od elettronici e configurandone i parametri. Questa guida e queste esercitazioni fanno riferimento alla creazione della netlist in maniera testuale.

Durante la stesura della netlist è bene tenere sempre a mente che PSpice® è *case insensitive*

ovvero non tiene conto della differenza tra caratteri maiuscoli e minuscoli. Pertanto, a titolo di esempio, le scritture `Vmin` e `VMIN` sono equivalenti ed interpretate allo stesso modo dal simulatore.

3.1 Formato della netlist

La netlist si compone di sezioni distinte che devono essere inserite rigorosamente nell'ordine qui specificato:

- *Titolo della netlist:* la prima linea della netlist è **sempre** considerata dal simulatore come riga di commento, pertanto non viene mai valutata e deve iniziare con il delimitatore di linea commentata come primo carattere. È opportuno riportare come commento una breve descrizione del circuito e delle analisi specificate dalla netlist.

- *Caricamento librerie:* questa sezione è opzionale e permette all'utente di specificare uno o più set di librerie di componenti che il simulatore caricherà prima dell'elaborazione della netlist e che utilizzerà come sorgente delle definizini dei modelli e dei sottocircuti presenti nella netlist, qualora questi non vengano definiti nelle due seziono sottostanti. È utilizzata per includere nella netlist le specifiche dei componenti utilizzati pi di frequente.

- *Definizione dei parametri globali:* questa sezione è opzionale e permette all'utente di specificare uno o più parametri che il simulatore utilizzerà nel corso delle simulazione. Ciò permette di specificare il valore dei componenti o dei loro parametri di modello sotto forma di variabili che possono essere utilizzare nelle simulazioni.

- *Definizioni dei modelli:* questa sezione è opzionale e permette all'utente di specificare uno o più modelli di componenti che il simulatore utilizzerà nel corso dell'elaborazione della netlist. Un modello permette di associare ad un dispositivo già noto a PSpice® (BJT, MOS, diodo, etc.) un insieme di parametri caratteristici, associando ad essi un nome utile per farvi riferimento in seguito.

- *Definizione dei sottocircuiti:* questa sezione è opzionale e permette all'utente di specificare uno o più modelli di sottocircuiti che il simulatore utilizzerà nel corso dell'elaborazione della netlist. Un sottocircuito è un componente generico, con una propria interfaccia, netlist e parametri che può veniere riutilizzato più volte nella nella netlisti come fosse uno dei dispodivi noti a PSpice®.

- *Dichiarazione dei componenti:* questa sezione è obbligatoria e deve contenere la lista dei componenti che costituiscono il circuito, specificandone le connessioni ed i relativi parametri.

- *Comandi di simulazione:* questa sezione è obbligatoria e deve contenere la lista dei comandi che specificano il tipo di simulazione che PSpice® deve eseguire sul circuito specificato nella sezione precedente.

- *Istruzione di post-simulazione:* questa sezione è opzionale e permette all'utente di specificare quali variabili circuitali devono venire mostrate al termine della simulazione. È possibile indicare al simulatore di eseguire operazioni matematiche sui segnali quali calcolo della fase, trasformata di Fourier, etc.

3.2 Caricamento delle librerie

PSpice® carica già automaticamente tutte le librerie di componenti di base, quali resistori, condensatori, induttori, diodi, transistori e generatori di tensione e corrente. Durante queste esercitazioni non avremo necessità di utilizzare nulla al di fuori di questi componenti, pertanto non sarà mai necessario includere librerie esterne. Pertanto, per una discussione di questo argomento, si consulti il manuale ufficiale del simulatore.

3.3 Definizione dei parametri globali

Tramite la parola chiave `.PARAM` è possibile definire una variabile globale che può essere utilizzata nelle simulazioni parametriche per studiare la variazione delle prestazioni in funzione della variazione dei valori dei parametri dei componenti del circuito. La sintassi del comando è la seguente:

```
.PARAM < <nome> = <valore> >*
.PARAM < <nome> = espressione >*
```

È possibile definire il valore di un parametro sia come una costante numerica, sia come un'espressione: in quest'ultimo caso è necessario racchiudere l'espressione entro parentesi graffe. Un'espressione può contenere al suo interno sia costanti che parametri.

Le definizoni di parametri sono indipendenti dall'ordine e possono essere utilizzate anche deltro ai sottocircuiti per definire dei parametri locali. Una volta definito, un parametro può essere utilizzato al posto della maggior parte dei valori numerici presenti nella netlist.

Di seguito riportiamo alcuni esempi di definizione di parametri.

```
.PARAM GAIN=10
.PARAM G2={GAIN*GAIN}
```

3.4 Definizione dei modelli

Tramite la parola chiave `.MODEL` è possibile definire il valore di un insieme di parametri associati ad un componente elettrico od elettronico il cui modello matematico sia già stato definito in PSpice® (BJT, MOS, diodo, ecc.) e di dare a questo insieme un nome simbolico con cui farvi riferimento in seguito. La sintassi del comando è la seguente:

```
.MODEL <nome> <tipo>
+ [ <nome paramentro> = <valore> ]*
```

dove *nome* è il nome che si vuole attribuire al modello e *tipo* è il nome del modello matematico definito in PSpice®. È possibile specificare il valore dei parametri del modello sia sotto forma di costante numerica, sia tramite un parametro globale. Di seguito riportiamo un esempio di modello di transistore BJT ed di diodo che verranno usati nel corso delle esercitazioni.

```
.MODEL NBJTA NPN IS=1E-17 BF=100
.MODEL DMOD D IS={ISvar}
```

3.5 Definizione dei sottocircuiti

Tramite la parola chiave `.SUBCKT` è possibile definire un sottocircuito che può essere inserito nella netlist come uno dei componenti già definiti in PSpice®. È inoltre possibile definire uno o più parametri per fare in modo che le caratteristiche del componente possano essere definite nella fase di inserimento nel circutio invece che durante la sua definizione. La sintassi del comando è la seguente:

```
.SUBCKT <nome> [nodi]*
+ [PARAMS: < <nome> = <valore> >* ]
...
.ENDS
```

dove *nome* nella prima riga è il nome che si dovrà indicare al simulatore per richiamare il circuito e *nodi* è la lista dei nodi di interfaccia del circuito. Questa lista è opzionale in quanto è possibile definire sottocircuiti senza nodi di interfaccia. Infine è possibile definire una lista di parametri opzionali come coppia nome-valore. La definizione di un sottocircuito deve **sempre** terminare con il comando `.ENDS`.

Di seguito riportiamo alcuni esempi di definizioni di sottocircuiti.

```
.SUBCKT OPAMP 1 2 101 102 17
...
.ENDS

.SUBCKT FILTER IN OUT PARAMS: Fc=100kHz BW=10kHz
...
.ENDS
```

3.6 Dichiarazione dei componenti

Tutti i componenti riconosciuti nella netlist vengono inseriti utilizzando la seguente sintassi:

```
<iniziale><nome> <nodi>* [modello] [valore]
```

dove *iniziale* è un codice mnemonico, generalmente di un carattere, che permette a PSpice® di identificare il tipo di componente, *nome* è una stringa alfanumerica che permette di identificare univocamente il componente corrente e *nodi* è la lista dei nomi dei nodi a cui esso è connesso. Il nodo 0 è riservato per la massa del circuito.

Alcuni componenti, come i transistori BJT, richiedono di specificare il modello a cui fa riferimento l'istazna corrente, mentre per la maggior parte dei componenti è sufficiente specificarne il valore, come per i componenti passivi.

Di seguito riportiamo le sintassi ed una sommaria descrizione dei più comuni componenti che incontreremo in queste esercitazioni, comprensive di codice mnemonico.

3.6.1 Resistore (R)

Un resistore può essere dichiarato utilizzando la seguente sintassi:

```
R<nome> <nodo (+)> <nodo (-)> <valore>
```

dove i nodi (+) e (-) definiscono la polarità del resistore quando il resistore è attraversato da una corrente che scorre dal nodo (+) al nodo (-). Si considerino, a titolo di esempio, le seguenti linee di comando:

```
Rload 15 0 2k
R2 1 2 {Rref}
```

che dichiarano, rispettivamente, un resistore chiamato Rload di valore 2kΩ collegato tra i nodi 15 e 0 ed un resistore chiamato R2 collegato tra i nodi 1 e 2 il cui valore è impostato tramite il parametro `Rref`.

3.6.2 Condensatore (C)

Un condensatore può essere dichiarato utilizzando la seguente sintassi:

```
C<nome> <nodo (+)> <nodo (-)> <valore> [IC=valore]
```

dove i nodi (+) e (-) definiscono la polarità del condensatore ed il parametro opzionale `IC` permette di specificare la tensone iniziale sul condensatore per il calcolo del punto di riposo. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
Cload 15 0 20pF
C2 1 2 .2E-12 IC=1.5V
```

che dichiarano, rispettivamente, un condensatore chiamato Cload di valore 20pF collegato tra i nodi 15 e 0 ed un condensatore chiamato C2 di valore 0.2pF collegato tra i nodi 1 e 2 la cui tensione iniziale è pari a 1.5V. È possibile utilizzare come valore un'espressione od un parametro secondo la sintassi già utilizzata in Sez. 3.6.1.

3.6.3 Induttore (L)

Un induttore può essere dichiarato utilizzando la seguente sintassi:

```
L<nome> <nodo (+)> <nodo (-)> <valore> [IC=valore]
```

dove i nodi (+) e (-) definiscono la polarità dell'induttore ed il parametro opzionale `IC` permette di specificare la corrente iniziale sull'induttore per il calcolo del punto di riposo. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
Lload 15 0 20mH
L2 1 2 .2E-6 IC=2mA
```

che dichiarano, rispettivamente, un induttore chiamato Lload di valore 20mH collegato tra i nodi 15 e 0 ed un induttore chiamato L2 di valore 0.2μF collegato tra i nodi 1 e 2 la cui corrente iniziale è pari a 2mA. È possibile utilizzare come valore un'espressione od un parametro secondo la sintassi già utilizzata in Sez. 3.6.1.

3.6.4 Generatori indipendenti di corrente (I) e di tensione (V)

I generatori indipendenti di corrente e di tensione possono essere dichiarati utilizzando la seguente sintassi:

```
I<nome> <nodo (+)> <nodo (-)> [ [DC] <valore> ]
+ [ AC <ampiezza> [fase] ] [ transitorio ]

V<nome> <nodo (+)> <nodo (-)> [ [DC] <valore> ]
+ [ AC <ampiezza> [fase] ] [ transitorio ]
```

dove i nodi (+) e (-) definiscono la polarità del generatore, il parametro `valore` specifica il valore della corrente/tensione fornita in un'analisi DC ed i parametri `ampiezza` e `fase` specificano l'ampiezza e la fase in gradi del segnale sinusoidale generato durante l'analisi AC. Il parametro opzionale `transitorio` verrà discusso in dettaglio nei prossimi paragrafi.

Il valore di default per le correnti/tensioni generate durante le analisi DC, AC e di transitorio è 0: questo significa che, se non specificato altrimenti, un generatore per cui sia indicato solamente il valore AC, genererà corrente/tensione nulla durante un'analisi DC o di transitorio. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
Ibias 13 0 2.3mA
Vac 2 3 AC 1.0
Iacphs 2 3 AC .001 90
```

che dichiarano, rispettivamente, un generatore di corrente chiamato Ibias di valore 2.3mA collegato tra i nodi 13 e 0 utile per l'analisi DC, un generatore di tensione di valore 1.0V collegato tra i nodi 2 e 3 per l'anaisi AC ed un generatore di corrente Iacphs di valore 1mA collegato tra i nodi 2 e 3 per l'analisi AC, la cui fase è pari a 90°.

È possibile utilizzare come valore un'espressione od un parametro secondo la sintassi già utilizzata in Sez. 3.6.1.

Transitorio esponenziale (EXP) Per fare in modo che, in regime transitorio, un generatore di corrente o di tensione produca una forma d'onda esponenziale, è necessario utilizzare la seguente sintassi per il parametro `transitorio`:

```
EXP (<i1> <i2> <td1> <tc1> <td2> <tc2>)
```

dove `i1` è la corrente/tensione di riposo, `i2` la corrente/tensione di picco ed i parametri `td1` e `tc1`, `td2` e `tc2` specificano rispettivamente l'inizio del transitorio di salita e la sua costante di tempo, l'inizio del transitorio di disesca e la sua costante di tempo. Si consideri, a titolo di esempio, la seguente linea di comando:

```
IRAMP 10 5 EXP(1 5 1 .2 2 .5)
```

che dichiara un generatore di corrente tra i nodi 10 e 5 con una corrente di riposo di 1A, una corrente di picco di 5A ed i cui transitori di salita e di discesa iniziano rispettivamente ad 1 e 2 secondi con costanti di tempo pari a 0.2 e 0.5 secondi.

Transitorio sinusoidale (SIN) Per fare in modo che, in regime transitorio, un generatore di corrente o di tensione produca una forma d'onda sinusoidale smorzata, è necessario utilizzare la seguente sintassi per il parametro `transitorio`:

```
SIN (<off> <amp> <freq> <td> <df> <fase>)
```

dove `off` è la corrente/tensione di riposo, `amp` è la corrente/tensione di picco, `freq` è la frequenza della componente sinusoidale, `td` specifica l'inizio del transitorio sinusoidale, `df` specifica il fattore di smorzamento esponenziale e `fase` specifica, in gradi, la fase con cui avrà inizio il transitorio sinusoidale.

È bene ricordare che una fase iniziale diversa da 0 modifica il valore iniziale di corrente/tensione prodotte dal generatore in conformità alla seguente relazione:

$$i_0 = \text{off} + \text{ampl} \cdot sin(2\pi \cdot \text{fase}/360°)$$

Si consideri, a titolo di esempio, la seguente linea di comando:

```
ISIG 10 5 SIN(2 2 5Hz 1sec 1 30)
```

che dichiara un generatore di corrente tra i nodi 10 e 5 con una corrente di riposo di 2A, una corrente di picco di 2A con componente sinusoidale di frequenza pari a 5Hz a partire da 1 secondo, costante di smorzamento pari ad 1 e fase iniziale pari a 30°. Per effetto della fase iniziale non nulla, il valore iniziale di corrente sarà pari a 3A.

Transitorio impulsato (PULSE) Per fare in modo che, in regime transitorio, un generatore di corrente o di tensione produca una forma d'onda impulsata, è necessario utilizzare la seguente

sintassi per il parametro `transitorio`:

```
PULSE (<i1> <i2> <td> <tr> <tf> <pw> <per>)
```

dove `i1` è la corrente/tensione di riposo, `i2` la corrente/tensione di picco ed i parametri `td`, `tr`, `tf`, `pw` e `per` specificano rispettivamente l'istante di inizio del transitorio, la durata della fase di salita, la durata della fase di discesa, la durata della fase di picco ed il periodo degli impusi, ovvero la distanza temporale che separa ciascuno dei fronti di salita dal precedente. Si consideri, a titolo di esempio, la seguente linea di comando:

```
ISW 10 5 PULSE(1A 5A 1sec .1sec .4sec .5sec 2sec)
```

che dichiara un generatore di corrente tra i nodi 10 e 5 con una corrente di riposo di 1A, una corrente di picco di 5A, in cui il primo fronte di salita si ha dopo 1 secondo per una durata di 0.1 secondi, il fronte di discesa dura 0.4 secondi ed avviene dopo 0.5 secondi dal termine del fronte di salita e la ripetizione degli impulsi avviene ogni 2 secondi.

Transitorio lineare a tratti (PWL) Per fare in modo che, in regime transitorio, un generatore di corrente o di tensione produca una forma d'onda lineare a tratti, è necessario utilizzare la seguente sintassi per il parametro `transitorio`:

```
PWL <(<tn>, <in>)>*
```

dove `(tn,in)` rappresenta una coppia ordinata (tempo, valore) tale per cui il generatore produrrà in uscita il valore `in` all'istante di tempo `tn`, congiungendo ciascuno dei punti della lista con un tratto di retta. Le coordinate temporali **devono** essere fornite in ordine strettamente cresente. Per istanti di tempo precedenti alla prima coordinata fornita e per valori successivi all'ultima, il generatore produrrà valori costanti e pari a quanto indicato rispettivamente nella prima e nell'ultima coppia ordinata. Si consideri, a titolo di esempio, la seguente linea di comando:

```
V1 13 4 PWL (1m,0)(1.005m,3.3)
```

che dichiara un generatore di tensione tra i nodi 13 e 4 che produce un fronte di salita di durata pari a 5μs da 0V a 3.3V all'istante 1ms. Per t$\leq$1ms la tensione prodotta è di 0V, mentre è di 3.3V per t$\geq$1.005ms.

3.6.5 Generatori dipendenti di corrente (F,G) e di tensione (E,H)

I generatori dipendenti di corrente e di tensione si suddividono in due categorie: generatori dipendenti controllati in tensione (E,G) e generatori dipendenti controllati in corrente (F,H). I generatori controllati in tensione possono essere dichiarati utilizzando la seguente sintassi:

```
E<nome> <nodo (+)> <nodo (-)>
+ <nodo (C+)> <nodo (C-)> <gain>

G<nome> <nodo (+)> <nodo (-)>
+ <nodo (C+)> <nodo (C-)> <gain>
```

dove i nodi (+) e (-) definiscono la polarità del generatore, i nodi (C+) e (C-) definiscono la differenza di potenziale che pilota il generatore ed il parametro `gain` specifica il guadagno tra tensione di controllo e grandezza prodotta dal generatore. I generatori controllati in corrente possono essere dichiarati utilizzando la seguente sintassi:

```
F<nome> <nodo (+)> <nodo (-)>
```

```
+ <Vcontrollo> <gain>

H<nome> <nodo (+)> <nodo (−)>
+ <Vcontrollo> <gain>
```

dove i nodi (+) e (-) definiscono la polarità del generatore, il parametro `Vcontrollo` specifica il nome del generatore di tensione su cui misurare la corrente, indipendentemente dalla tensione da esso generata, ed il parametro `gain` specifica il guadagno tra corrente di controllo e grandezza prodotta dal generatore. Infine, per i soli generatori dipendenti controllati in tensione, è possibile utilizzare la seguente sintassi:

```
E<nome> <nodo (+)> <nodo (−)> VALUE =  <espressione>
G<nome> <nodo (+)> <nodo (−)> VALUE =  <espressione>
```

dove è possibile specificare il valore della tensione o corrente generata tramite un'espressione che contenga al suo interno solo grandezze misurate in tensione. Nel corso delle esercitazioni quest'ultima sintassi verrà utilizzata, ad esempio, per modellare il comportamento degli amplificatori operazionali. Si considerino infine, a titolo di esempio, le seguenti linee di comando:

```
Ebuff 1 2 10 11 1.0
Fsense 1 2 Vsense 10.0
Esqrt 5 0 VALUE = 5V*SQRT(V(3,2))
```

che dichiarano rispettivamente un generatori di tensione Ebuff tra i nodi 1 e 2 controllato dalla tensione presente fra i nodi 10 ed 11 con guadagno unitario, un generatore di corrente Fsense controllato dalla corrente che scorre sul generatore di tensione Vsense con guadagno pari a 10 ed infine un generatore di tensione che produce una tensione pari a 5 volte la radice quadrata della tensione misurata tra i nodi 3 e 2.

3.6.6 Diodo

Un diodo può essere dichiarato utilizzando la seguente sintassi:

```
D<nome> <nodo (+)> <nodo (−)> <modello>
```

dove i nodi (+) e (-) definiscono la polarità del diodo ed il parametro `modello` permette all'utente di specificare quale fra i diversi modelli di diodi definiti il simulare deve utilizzare. Si rimanda al manuale di PSpice®per la consultazione della lista dei parametri che possono essere specificati nel modello del diodo. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.MODEL DMOD D IS=1F
Dclamp 14 0 DMOD
```

che dichiarano, rispettivamente, un modello di diodo chiamato DMOD la cui corrente di saturazione è impostata a 10^{-15}A, ed un diodo Dclamp connesso tra i nodi 14 e 0 che usa DMOD come modello.

3.6.7 Transistore Bipolare

Un transistore bipolare può essere dichiarato utilizzando la seguente sintassi:

```
Q<nome> <nodo (C)> <nodo (B)> <nodo (E)> <modello>
```

dove i nodi (C), (B) ed (E) definiscono le connessioni di collettore, base ed emettitore ed il

parametro `modello` permette all'utente di specificare quale fra i diversi modelli di transistori bipolari definiti il simulare deve utilizzare. Si rimanda al manuale di PSpice® per la consultazione della lista dei parametri che possono essere specificati nel modello del transistore bipolare. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.MODEL NBJTA NPN (IS=1E-17 BF=100)
.MODEL PBJTA PNP (IS=1E-17 BF=90)
Q1 2 1 15 NBJTA
Q2 2 1 14 PBJTA
```

che dichiarano, rispettivamente, un modello chiamato NBJTA per transistori bipolari di tipo NPN la cui corrente di saturazione è impostata a 10^{-17}A ed il coefficiente β_f a 100, un modello chiamato PBJTA per transistori bipolari di tipo PNP la cui corrente di saturazione è impostata a 10^{-17}A ed il coefficiente β_f a 90, e due transistori bipolari Q1 e Q2 connessi a formare un invertitore di tensione.

3.6.8 Sottocircuito (X)

Un sottocircuito definito tramilte il comando `.SUBCKT` può essere dichiarato utilizzando la seguente sintassi:

```
X<nome> [nodi]* <modello>
+[PARAMS: < <nome> = <valore> >*]
```

dove la lista dei nodi rispecchia, nell'ordine, quanto specificato nella definizione del sottocircuito e `modello` deve essere il nome del sottocircuito stabilito al momento della definizione. È possibile specificare anche il valore dei parametri utilizzati nella definizione del sottocircuito, utilizzando la appropriata coppia nome-valore. Infine, è possibile anche in questo caso sostituireo il valore del parametro con un parametro globale. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
XFOLLOW IN OUT VCC VEE OUT OPAMP
XFELT 1 2 FILTER PARAMS: CENTER=200kHz
```

che dichiarano, rispettivamente, un sottocircuito Xfollow con un'interfaccia a 5 nodi, utilizzando il modello Opamp, ed un sottocircuito Xfelt con un'interfaccia a 2 nodi, utilizzando il modello Filter e ridefinendo per esso il parametro Center.

3.7 Comandi di simulazione

Tutti i comandi di simulazione iniziano con il carattere `.` e permettono all'utente di impostare uno o più profili di simulazione per condurre analisi sul circuito specificato nella netlist. I comandi utilizzati con maggiore frequenza saranno discussi in seguito, mentre per una lista più dettagliata si rimanda alla consultazione del manuale di PSpice®.

3.7.1 Temperatura di simulazione (.TEMP)

Il comando `.TEMP` permette all'utente di specificare al simulatore una o più temperature a cui devono venire condotte le analisi sul circuito. Le temperature sono sempre espresse in gradi centigradi e, in presenza di una lista di temperature, il simulatore ripeterà le analisi per ogni valore di temperature presente nella lista. La sintassi del comando è la seguente:

```
.TEMP <temperatura>*
```

Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.TEMP 125
.TEMP 0 27 125
```

che indicano al simulatore rispettivamente di compiere un'analisi a 125°C e di ripetere le analisi per le temperature di 0°C, 27°C e 125°C.

Si ricordi che il comando `.TEMP` non influenza la temperatura a cui si assumono misurati i parametri dei componenti del circuito. Per modificare questa temperatura utilizzare il comando:

```
.OPTIONS TNOM=<temperatura>*
```

3.7.2 Calcolo del punto di riposo (.OP)

Il comando `.OP` permette all'utente di ottenere dal simulatore informazioni dettagliate circa il punto di riposo del circuito. La sintassi del comando è la seguente:

```
.OP
```

Il punto di riposo viene sempre calcolato dal simulatore, anche in assenza del comando `.OP`, ma senza di esso le uniche informazioni restituite riguardano le tensioni ai nodi, le correnti sui generatori di tensione e la dissipazione totale di potenza del circuito. Al contrario, in sua presenza, oltre alle informazioni già indicate, vengono riportati anche tutti i parametri linearizzati attorno al punto di riposo dei dispositivi non lineari e dei semiconduttori presenti nel circuito.

3.7.3 Calcolo della risposta in frequenza (.AC)

Il comando `.AC` indica al simulatore di calcolare la risposta in frequenza del circuito su uno specificato intervallo di frequenze. La sintassi del comando è la seguente:

```
.AC <LIN | OCT | DEC> <num.  punti>
+ <freq.  iniziale> <freq.  finale>
```

dove il primo parametro permette di scegliere tra una suddivisione rispettivamente lineare, ottale o decadica dell'asse delle frequenze, il secondo parametro indica al simulatore in quanti punti suddividere l'intervallo di frequenze specificato dal terzo e quarto parametro. Entrambe queste frequenze **devono** essere positive e maggiori di zero. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.AC LIN 101 100Hz 200Hz
.AC OCT 10 1kHz 16kHz
.AC DEC 20 1MEG 100MEG
```

che indicano al simulatore rispettivamente di compiere un'analisi in frequenza su un intervallo di frequenze suddiviso linearmente in 101 punti compresi tra 100Hz e 200Hz, su un intervallo di frequenze suddiviso in ottave con 10 punti compresi tra 1kHz e 16kHz e su un intervallo di frequenze suddiviso in decadi con 20 punti compresi tra 1MHz e 100MHz.

È bene ricordare che l'analisi AC è un'analisi lineare: il simulatore calcola la risposta in frequenza linearizzando il circuito attorno al punto di riposo. Inoltre, solo i generatori che hanno specificato il parametro AC vengono presi in considerazione durante questa analisi: la specifica di transitorio SIN non viene presa in considerazione, essendo utilizzato solamente durante le simulazioni di transitorio.

3.7.4 Calcolo della caratteristica statica (.DC)

Il comando `.DC` indica al simulatore di calcolare la caratteristica statica del circuito su uno specificato intervallo di valori. La sintassi del comando è la seguente:

```
.DC <[LIN] | OCT | DEC> <variabile>
+ <val.  iniziale> <val.  finale> <passo>

.DC <variabile> LIST <valori>*
```

dove, nel primo caso, il primo parametro permette di scegliere tra una suddivisione rispettivamente lineare, ottale o decadica dell'intervallo dei valori assunti dalla variabile indicata dal secondo parametro, mentre i restanti parametri permettono di specificare l'intervallo di valori ed il passo di suddivisione. Utilizzando invece la seconda sintassi è possibile indicare al simulatore di eseguire un'analisi solo su una lista ristretta di valori della variabile indicata come primo parametro.

In caso di suddivisione ottale o decadica, il parametro `passo` indica in quante parti suddividere l'intervallo specificato. Nel caso in cui si voglia utilizzare come variabile un parametro globale, è necessario far precedere il nome del parametro stesso dalla parola chiave `PARAM`. Ciò permette di studiare la variazione del punto di riposo in funzione delle variazioni di un parametro di uno dei dispositivi del circuito. Infine, è possibile utilizzare come variabile anche la temperatura di simulazione: per fare ciò è necessario utilizzare come variabile la parola chiave `TEMP`. Il parametro `LIN` può essere omesso.

Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.DC I2 5mA -2mA 0.1mA
.DC PARAM Vsupply 7.5 15 .5
.DC DEC PARAM Rvar 0.9 1.1 11
.DC TEMP LIST 0 20 27 50 80 100
```

che indicano al simulatore rispettivamente di compiere il calcolo della caratteristica statica al variare della corrente fornita dal generatore I2 su un intervallo lineare compreso fra 5mA e -2mA con passo 0.1mA, al variare del parametro Vsupply da 7.5 a 15 con passo .5, al variare del valore del parametro Rvar su un intervallo decadico compreso fra 0.9Ω e 1.1Ω in 10 passi ed al variare della temperatura su una lista discrta di valori.

3.7.5 Calcolo della caratteristica di transitorio (.TRAN)

Il comando `.TRAN` indica al simulatore di calcolare la caratteristica di transitorio su uno specificato intervallo di tempo. La sintassi del comando è la seguente:

```
.TRAN[/OP] <Tstep> <Tstop> [Tstart [Tmax]]
```

dove il primo parametro è opzionale e, se presente, permette di ottenere una stampa dettagliata del punto di riposo, mentre i restanti parametri indicano al simulatore come eseguire l'analisi di transitorio.

In particolare, `Tstep` indica al simulatore il passo temporale con cui vengono registrati su file i risultati della simulazione, `Tstop` specifica l'istante in cui deve terminare la simulazione, `Tstart` indica l'istante in cui il simulatore inzia a registrare su file i risultati della simulazione, infine `Tmax` specifica il limite superiore del passo temporale con cui il simulatore esegue internamente l'analisi di transitorio. È bene specificare sempre quest'ultimo valore ed impostarlo ad un valore non superiore a quello specificato per `Tstep`.

Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.TRAN 1ns 100ns
.TRAN/OP 1ns 100ns 20ns
.TRAN 1ns 100ns 0ns .1ns
```

che indicano al simulatore rispettivamente di compiere un'analisi di transitorio da 0 a 100ns con passo di stampa di 1ns, la stessa analisi del primo caso omettendo il primi 20ns di simulazione ma aggiungendo una descrizione dettagliata del punto di riposto ed infine, la stessa analisi del primo caso ma limitando superiormente il passo interno di simulazione a 0.1ns per ottenere un risultato più affidabile.

3.7.6 Analisi parametrica (.STEP)

Il comando `.STEP` indica al simulatore di compiere un'analisi parametrica, ripetendo tutte le analisi indicate dai comandi `.AC`, `.DC` e `.TRAN` al variare delle condizioni specificate da questo comando. La sintassi è la seguente:

```
.STEP <[LIN] | OCT | DEC> <variabile>
+ <val.  iniziale> <val.  finale> <passo>

.STEP <variabile> LIST <valori>*
```

dove il significato dei parametri è lo stesso rispetto a quanto descritto per il comando `.DC`

È bene ricordare che non è possibile specificare nella stessa netlist un'analisi `.STEP` identica ad un'analisi `.DC`. Il simulatore segnala questa evenienza come errore in fase di lettura della netlist.

Si consideri, a titolo di esempio, il seguente frammento di netlist:

```
.PARAM AMP=0
Vac 1 0 AC AMP
.AC DEC 100 1000 1E6
.STEP PARAM AMP 0 5 1
```

che indica al simulatore di ripetere il calcolo della risposta in frequenza su un intervallo di valori compreso fra 100Hz ed 1kHz suddiviso in maniera decadica in un milione di punti, variando di volta in volta l'ampiezza delle componente sinusoidale generata dal generatore di tensione Vac tramite l'uso del parametro globale AMP.

3.7.7 Calcolo della funzione di trasferimento (.TF)

Il comando `.TF` indica al simulatore di calcolare la funzione di trasferimento fra una variabile ed un generatore di tensione o di corrente linearizzando il circito attorno al punto di riposo. Vengono inoltre calcolate la resistenza di ingresso e quella di uscita. La sintassi del comando è la seguente:

```
.TF <uscita> <ingresso>
```

dove la sintassi del parametro `uscita` è identica a quanto esposto in seguito per il comando `.PROBE`, mentre `ingresso` è l'identificatore di un generatore presente nella netlist.

3.8 Istruzioni di post simulazione

Tutte le istruzioni di post simulazione iniziano con il carattere `.` e permettono all'utente di specificare quali variabile salvare su disco dopo una simulazione o come combinare fra loro i

risultati ottenuti dalla simulazione. I comandi utilizzati con maggiore frequenza saranno discussi in seguito, mentre per una lista più dettagliata si rimanda alla consultazione del manuale di PSpice®.

3.8.1 Selezione delle variabili di analisi (.PROBE)

Il comando `.PROBE` permette all'utente di specificare al simulatore quali variabili memorizzare su disco al termine delle analisi AC, DC e transitorio. Ridurre la liste delle variabili salvate su disco è utile soprattutto nel caso di netlist con molti nodi e componenti, così da ridurre il più possibile l'occupazione su disco. La sintassi del comando è la seguente:

```
.PROBE [variabili]*
```

Omettendo la lista delle varibili, PSpice® salva su disco tutte le variabili utilzzate durante la simulazione.

Per salvare la variabile associata alla corrente che scorre su un componente a due terminali è possibile utilizzare la sintassi `I(<nome>)` dove il parametro `nome` è il nome del componente desiderato. In maniera del tutto analoga, per salvere la variabile associata alla tensione presente su un nodo è possibile utilizzare la sintassi `V(<nome>)` dove il parametro `nome` è il nome del nodo desiderato. Infine, è possibile salvare anche la potenza dissipata da un dispositivo tramite la sintassi `W(<nome>)` dove il parametro `nome` è il nome del dispositivo desiderato. Si considerino, a titolo di esempio, le seguenti linee di comando:

```
.PROBE
.PROBE V(1) I(D1) W(Q1)
```

che salvano su disco, rispettivamente, tutte le variabili utilizzate durante l'analisi e solamente la tensione del nodo 1, la corrente che scorre sul diodo D1 e la potenza dissipata dal transistore bipolare Q1.

Per una lista più completa delle sintassi supportate per il salvataggio delle variabili si rimanda alla guida di PSpice®.

3.8.2 Stampa delle variabili di analisi (.PRINT)

Il comando `.PRINT` permette all'utente di specificare al simulatore che questi deve stampare sotto forma di tabella l'andamento di variabili calcolate durante le analisi in AC, DC e transitorio. La sintassi del comando è la seguente:

```
.PRINT <AC | DC | TRAN> [variabili]*
```

dove la sintassi dell'unico parametro è identica a quanto esposto per il comando `.PROBE`.