

HSPICE 简明教程



復旦大學

专用集成电路与系统国家重点实验室

RFIC

整理者	版本号	日期	说明
官志超	1.0	2007.4.7	本文档内容以常用 HSPICE 指令为主, 主要目的为便于学习与查询, 详细了解请参阅参考文献

版权所有，不得侵犯！传播与修改请保留版权信息。

目录

第一章 概述	5
§1.1 HSPICE 简介	5
§1.2 常数	5
§1.3 输入输出文件及后缀	5
§1.4 一个简单例子	6
§1.5 符号说明	7
第二章 仿真输入及控制的设置	8
§2.1 输入网表概要	8
§2.2 网表文件中的元素	8
第三章 器件及电源	15
§3.1 器件	15
§3.2 独立源	16
3.2.1 直流源	16
3.2.2 交流源	16
3.2.3 瞬态源	16
3.2.4 混合源	21
§3.3 受控源	22
3.3.1 压控电压源 E ELEMENTS	22
3.3.2 压控电流源 G ELEMENTS	23
第四章 参数、函数及仿真设置	25
§4.1 参数	25
4.1.1 参数定义	25
4.1.2 .PARAM 声明	25
4.1.3 指令行内定义	25
4.1.4 代数表达式定义输出参数	25
4.1.5 倍乘参数 M (THE MULTIPLY PARAMETER)	25
4.1.6 参数作用范围	26
§4.2 函数	27
4.2.1 用户定义函数	27
4.2.2 内置函数	27
4.2.3 保留变量	29
§4.3 仿真设置	29
4.3.1 设置控制选项 (CONTROL OPTIONS)	29
4.3.2 基本控制选项	29
第五章 输出设置	31
§5.1 输出指令	31
§5.2 输出参数	31
5.2.1 直流和瞬态分析输出参数	31

5.2.2	功率	32
5.2.3	交流分析输出参数	32
5.2.4	网路相关参数	33
5.2.5	噪声和谐波分析输出参数	33
5.2.6	器件参数输出	34
第六章	常用分析	35
§6.1	直流初始化及工作点分析	35
6.1.1	电路初始化	35
6.1.2	工作点分析 (OPERATING POINT) .OP 声明	35
§6.2	直流扫描分析	36
6.2.1	.DC 声明	36
6.2.2	例子	36
6.2.3	其他直流分析声明	37
§6.3	瞬态分析	38
6.3.1	瞬态分析的初始化	38
6.3.2	瞬态分析 .TRAN 声明	38
6.3.3	例子	38
6.3.4	傅立叶分析	38
§6.4	交流分析	40
6.4.1	交流分析 .AC 声明	40
6.4.2	例子	40
6.4.3	其他交流分析	41
第七章	统计分析及优化	43
§7.1	用户定义的分析	43
7.1.1	.MEASURE 声明	43
7.1.2	上升、下降和延迟(RISE FALL AND DELAY)	43
7.1.3	FIND 和 WHEN 函数	44
7.1.4	方程计算	45
7.1.5	平均值、均方根值、最大最小值和峰峰值测量	45
7.1.6	积分函数	46
7.1.7	微分函数	46
7.1.8	误差函数	47
§7.2	温度分析	48
§7.3	最坏情况分析	48
7.3.1	标准统计名词定义	48
7.3.2	最坏情况分析介绍	49
7.3.3	模型歪斜参数及工艺角文件	49
§7.4	蒙特卡罗分析	50
7.4.1	蒙特卡罗分析概要	50
7.4.2	定义分布函数 .PARAM 声明	51
7.4.3	蒙特卡罗分析的例子	52

7.4.4 最差情况和蒙特卡罗分析的例子.....	53
§7.5 优化	58
7.5.1 优化概要	58
7.5.2 优化相关声明	59
7.5.3 优化的例子	60
备注:	63
参考文献:	63

第一章 概述

§1.1 Hspice 简介

Hspice是电路模拟仿真的工具。其前身可追溯到1972年诞生于美国加利福尼亚大学柏克莱分校的SPICE (Simulation Program with Integrated Circuit Emphasis)。Hspice可在直流到100GHz的频率范围内对电路进行准确的仿真、分析及优化。其主要特征如下:

- 极佳的收敛性。
- 精确的模型。
- 对模型及单元的电路优化,可同步优化多种参数。
- 支持蒙特卡罗和最差情况分析。
- 参数化单元的输入、输出及行为级代数描述。
- 等等。

§1.2 常数

F=1e-15 P=1e-12 N=1e-9 U=1e-6 M=1e-3
T=1e12 G=1e9 MEG=X=1e6 K=1e3

§1.3 输入输出文件及后缀

1) Hspice 输入文件:

输出配置文件	meta.cfg
初始化文件	hspice.ini
直流工作点初始化文件	<design>.ic
输入网表文件	<design>.sp
库输入文件	<library_name>
模拟转移数据文件	<design>.d2a

2) Hspice 输出文件:

输出列表	.lis 或由用户自己定义
瞬态分析结果	.tr#+
瞬态分析测量结果	.mt#
直流分析结果	.sw#+
直流分析测量结果	.ms#
交流分析结果	.ac#+
交流分析测量结果	.ma#
硬拷贝图形数据	.gr#++
数字输出	.a2d
FFT 分析图形数据	.ft#+++
子电路交叉列表	.pa#
输出状态	.st#
工作点节点电压 (初始条件)	.ic

- # : 代表扫描分析序号或者硬拷贝文件序号,一般从 0 开始。

- +: 表示在用.POST 语句产生图形数据后该文件才被确立。
- ++: 表示该文件需要一个.GRAPH 语句或有一个针对 meta.cfg 文件中存在的文件的地址计数器。该文件在 HSPICE 的 PC 版中不产生。
- +++: 表示只有当应用了.FFT 语句后该文件才被确立。
- 测量结果都是.measure 产生的文件。

§1.4 一个简单例子

图 1.1 是一个有直流交流电压源的简单电阻电容网络，电路由电阻 R1 和 R2，电容 C1，电源 V1 构成。节点 1 连接电源正极和电阻 R1，节点 2 连接电阻 R1 和 R2，电容 C1，节点 0 在 Hspice 中默认为接地节点。

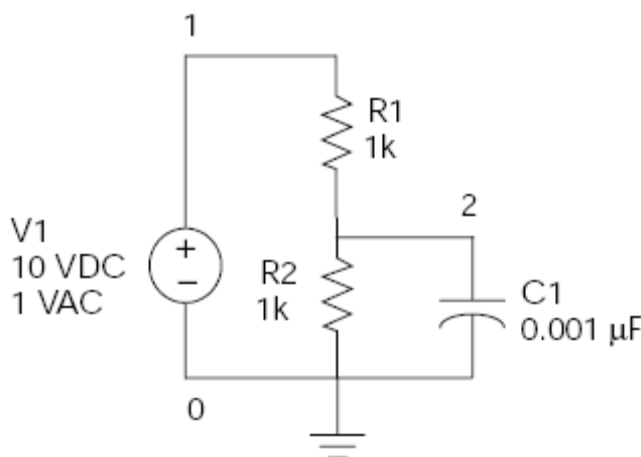


图 1.1 RC 网络电路

下面是Hspice的输入网表:

```
A SIMPLE AC RUN
.OPTIONS LIST NODE POST
.OP
.AC DEC 10 1K 1MEG
.PRINT AC V(1) V(2) I(R2) I(C1)
V1 1 0 10 AC 1
R1 1 2 1K
R2 2 0 1K
C1 2 0 .001U
.END
```

对这个电路进行交流分析:

1. 将上面的网表输入到文件quickAC.sp中。
2. 输入下面的命令启动Hspice:

```
hspice quickAC.sp > quickAC.lis
```

执行完成后出现如下提示:

```
>info: ***** hspice job concluded
```

以及运行时间等信息。

在运行目录中出现如下新文件:

```
quickAC.ac0
```

quickAC.ic
quickAC.lis
quickAC.st0

3. 用编辑器查看.lis和.st0文件来检查仿真结果及状态。
4. 启动 cscope 打开文件 quickAC.ac0 查看波形。查看图形输出启动命令为 cscope 或 Cscope。

§1.5 符号说明

语法中的“< >”表示可有可无的内容。

第二章 仿真输入及控制的设置

§2.1 输入网表概要

1) 输入网表内容

输入网表名格式<design>.sp, 输入网表内容包括:

- 电路网表
- 声明要用到的库(可选)
- 说明使用何种分析(可选)
- 说明期望的输出(可选)

输入网表中的指令顺序可随意, 最后以.end 结尾, 若不以.end 结尾, 会报错。

2) 行指令格式

不区分大小写, 一行写不开可以“+”开始另起一行。

3) 名称

名称必须以字母开始。

4) 节点

节点名称可以是数字, 也可以是字母, 或其组合。

0, GND,GND!,GROUND 都指全局地。

5) 器件名

必须以器件关键字开始, 如电容必须以 C 开始, 电阻用 R, MOS 管用 M, 二极管用 D, 等等。

6) 数字

可以使用科学计数如 1e-12, 或 p, 但不能两者共用。

§2.2 网表文件中的元素

2.2.1 .title

声明网表名称, 并非必要, 默认情况下 Hspice 不会执行网表的第一行, 而是将其解释为网表名称。

2.2.2 注释

以星号“*”开始一行, 或在句中美元符号“\$”开始。

例子:

```
*RF=1K GAIN SHOULD BE 100
$ MAY THE FORCE BE WITH MY CIRCUIT
VIN 1 0 PL 0 0 5V 5NS      $ 10v 50ns
R12 1 0 1MEG              $ FEED BACK
```

2.2.3 器件声明

主要描述三点:

- 器件类型及名称。
- 连接到哪些节点。
- 器件的电学参数。

常用的器件类型有:

C	电容
D	二极管
I	电流源
K	互感
L	电感
M	MOS管
Q	三极管
R	电阻
T,U,W	传输线
V	电压源
X	子电路

例子:

```
M1 ADDR SIG1 GND SBS N1 10U 100U
```

MOS管 M1, 漏、栅、源、衬底分别连接节点 ADDR, SIG1, GND, SBS, 采用模型 N1, length=10um, width=100um。

2.2.4 子电路定义与调用

1) 子电路定义声明

.subckt 或 .macro

语法:

```
.SUBCKT subnam n1 < n2 n3 ... > < parnam=val ... >
```

或

```
.MACRO subnam n1 < n2 n3 ... > < parnam=val ... >
```

说明:

subnam 子电路名。

n1,n2... 子电路接口节点名。

parnam=val 子电路参数, 应指定初值。

2) 子电路定义结束声明

ends. 或 .eom

3) 子电路调用声明

语法:

```
Xyyy n1 <n2 n3 ...> subnam <parnam=val ...> <M=val>
```

Xyyy 子电路名, 必须以“X”开始。

n1,n2... 子电路连接到的外部节点。

subnam 所调用的子电路名。

parnam=val 为调用的子电路参数赋值。

4) 例子

```
*FILE SUB2.SP TEST OF SUBCIRCUITS
```

```
.OPTIONS LIST ACCT
```

```
*
```

```
V1 1 0 1
```

```
.PARAM P5=5 P2=10
```

```
*
```

```

.SUBCKT SUB1 1 2 P4=4
R1 1 0 P4
R2 2 0 P5
X1 1 2 SUB2 P6=7
X2 1 2 SUB2
.ENDS
*
.MACRO SUB2 1 2 P6=11
R1 1 2 P6
R2 2 0 P2
.EOM
*
X1 1 2 SUB1 P4 =6
X2 3 4 SUB1 P6=15
X3 3 4 SUB2
*
.MODEL DA D CJA=CAJA CJP=CAJP VRB=-20 IS=7.62E-18
+ PHI=.5 EXA=.5 EXP=.33
*
.PARAM CAJA=2.535E-16 CAJP=2.53E-16
.END

```

上面的例子定义了两个子电路 **sub1** 和 **sub2**, 都是将电阻值参数化的电阻分压网络, 其中子电路 **sub1** 的定义中调用了子电路 **sub2**。x1,x2,x3 声明调用了这些子电路, 由于每次调用都为电阻赋了不同的值, 三次调用产生了不同的子电路。

5) 子电路节点调用

表示子电路中的器件, 将电路层次以“.”连接:

如: X1.XBIAS.M5

表示子电路中的节点:

如: .PRINT v(X1.X4.sig25) , sig25是子电路x4中的节点。

2.2.5 .global 声明

.global全局性地定义节点名, 不管处于电路中的什么层次上, 只要与.global中定义的节点名称相同, 他们就连接在一起。global通常用来定义电源连接。

例: .GLOBAL VDD input_sig

电路中所有与VDD重名的节点都连接在一起, 所有与input_sig重名的节点也连接在一起。

2.2.6 .temp 声明

声明电路运行的温度。默认为 25℃。

例: .TEMP -55.0 25.0 125.0

2.2.7 .data 声明

可用于.dc,.ac,.tran 等分析中的参数扫描, .data 可定义任意数目的参数, 使用的参

数必须事先定义，.data 可定义三种数据：

- 网表内数据(inline data)
- 外部文件串接数据(data concatenated from external files)
- 外部文件并接数据(data column laminated from external files)

这里只介绍网表内数据的定义

语法：

```
.DATA datanm pnam1 <pnam2 pnam3 ... pnamxxx >
+ pval1<pval2 pval3 ... pvalxxx>
+ pval1' <pval2' pval3' ... pvalxxx'>
.ENDDATA
```

说明：

Datanm 数据名，供仿真分析中调用使用。

pnam1... 参数名。

pval1... 参数值，可以有很多组。

例子：

```
.TRAN 1n 100n SWEEP DATA=devinf
.AC DEC 10 1hz 10khz SWEEP DATA=devinf
.DC TEMP -55 125 10 SWEEP DATA=devinf
.DATA devinf width length thresh cap
+          50u   30u   1.2v  1.2pf
+          25u   15u   1.0v  0.8pf
+          5u    2u    0.7v  0.6pf
+ ... ..
.ENDDATA
```

上面的分析中.data 定义了网表内数据 devinf，由于 Hspice 根据.data 中使用的参数数目，自动对数据分组，并给参数赋值，因此像例子中将数据与参数对准并非必要。上例中也说明了.data 定义的数据调用的方式，即用 data=dataname 的方式，扫描时参数值一组组地采用。

2.2.8 .include 声明

文件包含声明，也可简写作.inc

语法：

```
.INCLUDE '<filepath> filename'
```

说明：

filepath 文件路径。

filename 文件名。

2.2.9 .model 声明

定义器件模型。

语法：

```
.MODEL mname type <VERSION=version_number>
+ <pname1=val1 pname2=val2 ...>
```

说明：

mname	模型名。
type	类型，如：
AMP	运算放大器
C	电容
D	二极管
L	磁芯互感
NMOS	NMOS管
NPN	nnp三极管
PMOS	PMOS管
PNP	pnp三极管
R	电阻
pname1=val...	参数赋值。
version	版本。

例子：

```
.MODEL MOD1 NPN BF=50 IS=1E-13 VBF=50 AREA=2 PJ=3 N=1.05
```

2.2.10 .lib 声明

可以用.lib 声明将常用的命令、器件模型、子电路分析及定义放在一个库文件中。可以在主网表文件中调用这些库。

1) 库文件调用声明

语法：

```
.LIB '<filepath> filename' entryname
```

说明：

filepath 库文件路径。

filename 库文件名。

entryname 库名，一个库文件可定义多个库，每个库都有自己的名称。

例子：

```
.LIB 'MODELS' cmos1
```

2) 库文件定义声明

语法：

```
.LIB entryname1
.$ ANY VALID SET OF Hspice STATEMENTS
.ENDL entryname1
.LIB entryname2
.$ ANY VALID SET OF Hspice STATEMENTS
.ENDL entryname2
.LIB entryname3
.$ ANY VALID SET OF Hspice STATEMENTS
.ENDL entryname3
```

每个定义都以.lib entryname 开始，.endl entryname 结束，中间是任意有效的 Hspice 指令。

例子：

```
.LIB TT
```

```

$TYPICAL P-CHANNEL AND N-CHANNEL CMOS LIBRARY
$ PROCESS: 1.0U CMOS, FAB7
$ following distributions are 3 sigma ABSOLUTE GAUSSIAN
.PARAM TOX=AGAUSS(200,20,3)    $ 200 angstrom +/- 20a
+ XL=AGAUSS(0.1u,0.13u,3)      $ polysilicon CD
+ DELVTON=AGAUSS(0.0,-2V,3)    $ n-ch threshold change
+ DELVTOP=AGAUSS(0.0,.15V,3)   $ p-ch threshold change
.INC '/usr/meta/lib/cmos1_mod.dat'  $ model include file
.ENDL  TT
.LIB  FF
$HIGH GAIN P-CH AND N-CH CMOS LIBRARY 3SIGMA VALUES
.PARAM TOX=220 XL=-0.03 DELVTON=-.2V DELVTOP=-0.15V
.INC '/usr/meta/lib/cmos1_mod.dat'  $ model include file
.ENDL  FF

```

模型应包含于文件 /usr/meta/lib/cmos1_mod.dat. 中:

```

.MODEL NCH NMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTON .....
.MODEL PCH PMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTOP .....

```

2.2.11 .alter 声明

可以用这个声明改变参数甚至电路网表自动进行多次仿真。网表主文件中可以包含多个.alter 声明。程序第一次仿真第一行和第一个.alter 之间的指令，第二次应用第一个.alter 和第二个.alter 之间的指令更改参数后再仿真，以此类推，最后应用最后一个.alter 和.end 之间的指令。

更改设计参数及子电路的规则:

- 如果参数、器件、模型、子电路等的名称与之前声明的相同，之前的会被替换。
- 之前文件或.alter 块中的控制选项(options)会失效。
- 可以将电路拓扑结构的更改等信息保存在库中，然后用.lib 声明调用，用.del lib 声明删除。
- .alter 过程不能修改.include 声明调用的文件中的.lib 声明调用的库，可以修改.lib 声明调用的库中的.include 声明包含的文件。

例子:

```

FILE1: ALTER1 TEST CMOS INVERTER
.OPTIONS ACCT LIST
.TEMP 125
.PARAM WVAL=15U VDD=5
*
.OP
.DC VIN 0 5 0.1
.PLOT DC V(3) V(2)
*
VDD 1 0 VDD
VIN 2 0
*

```

```

M1 3 2 1 1 P 6U 15U
M2 3 2 0 0 N 6U W=WVAL
*
.LIB 'MOS.LIB' NORMAL
.ALTER
.DEL LIB 'MOS.LIB' NORMAL $removes LIB from memory
$PROTECTION

.PROT $protect statements below .PROT
.LIB 'MOS.LIB' FAST $get fast model library
.UNPROT
.ALTER
.OPTIONS NOMOD OPTS $suppress model parameters printing
* and print the option summary
.TEMP -50 0 50 $run with different temperatures
.PARAM WVAL=100U VDD=5.5 $change the parameters
VDD 1 0 5.5 $using VDD 1 0 5.5 to change the
$power supply VDD value doesn't
$work
VIN 2 0 PWL 0NS 0 2NS 5 4NS 0 5NS 5
$change the input source
.OP VOL $node voltage table of operating
$points
.TRAN 1NS 5NS $run with transient also
M2 3 2 0 0 N 6U WVAL $change channel width
.MEAS SW2 TRIG V(3) VAL=2.5 RISE=1 TARG V(3)
+ VAL=VDD CROSS=2 $measure output
*
.END

```

2.2.12 Hspice 的启动方法

一种是直接输入 hspice,按确定后进入命令行模式。

另一种常用的示例如下:

```
hspice demo.sp -n 7 > demo.out
```

demo.sp 是输入网表, -n 7 表示输出文件计数从 7 开始, demo.out 是输出列表文件。

第三章 器件及电源

§3.1 器件

1) 电阻

R1 1 2 100

电阻 R1 连接在 1 正节点 2 负节点之间, 阻值 100 欧姆。

RC1 12 17 R=1k TC1=0.001 TC2=0

电阻 RC1, 连接在节点 12 与 17 之间, 阻值 1K 欧姆, 温度系数 0.001 和 0。

Rterm input gnd R='sqrt(HERTZ)'

电阻 Rterm, 连接于节点 input 和 gnd 之间, 阻值是频率的平方根(只对交流分析非零)。

Rxxx 98999999 87654321 1 AC=1e10

电阻 Rxxx, 对直流及瞬态分析阻值是 1 欧姆, 交流分析阻值是 10G 欧姆。

2) 电容

C1 1 2 20p

电容 C1, 连接于节点 1 和 2 之间, 容值 20pF。

Cshunt output gnd C=100f M=3

三个电容并联于节点 output 和 gnd 之间, 每个的容值是 100fF。

Cload driver output C='1u*v(capcontrol)' CTYPE=1 IC=0v

电容 Cload 的容值是节点 capcontrol 电压乘上 1e-6, 电容两端初始电压 0V, ctype=1 表示容值是电容两端节点电压的函数。

C99 in out POLY 2.0 0.5 0.01

电容 C99 的容值由多项式 $C=c_0 + c_1*v + c_2*v*v$ 决定, v 是电容两端的电压。

3) 电感

L1 1 2 100n

电感 L1 连接于节点 1 和 2 之间, 感值 100nH。

Lloop 12 17 L=1u TC1=0.001 TC2=0

上例是具有温度系数的电感。

L99 in out POLY 4.0 0.35 0.01 R=10

电感 L99, 感值由多项式 $L=c_0 + c_1*i + c_2*i*i$ 决定, i 是通过电感的电流, 电感具有直流电阻 10 欧姆。

4) 互感

K1 Lin Lout 0.9

Kxfmr Lhigh Llow K=COUPLE

0.9 及 k=couple 是耦合系数, 取值 -1 ~ 1 之间, 如果是负值, 表示绕向是相反的。

5) 二极管

D1 1 2 diode1

6) 三极管

Q1 1 2 3 model_1

集电极、基极、发射极分别连接到 1, 2, 3, 使用模型 model_1。

7) MOS 管

Mopamp1 d1 g3 s2 b 1stage L=2u W=10u

漏、栅、源及衬底分别连接到 d1,g3,s2,b, 使用模型 1stage, 长宽分别为 2um,10um。

§3.2 独立源

语法:

电压源

Vxxx n+ n- <<DC=> dcval> <tranfun> <AC=acmag, <acphase>> ...

电流源

Ixxx n+ n- <<DC=> dcval> <tranfun> <AC=acmag, <acphase>> ...

说明:

Vxxx 电压源名, xxx 可为任意符号。

Ixxx 电流源名, xxx 可为任意符号。

n+ 正极。

n- 负极。

DC=dcval 直流电压值, 单位V, 默认为0V。

tranfun 瞬态电压源, 如AM,DC, EXP, PE, PL, PU, PULSE, PWL, SFFM,SIN。

acmag

acphase

..... 瞬态电压源参数。

3.2.1 直流源

V1 1 0 DC=5V

V1 1 0 5V

I1 1 0 DC=5mA

I1 1 0 5mA

3.2.2 交流源

V1 1 0 AC=10V, 90

VIN 1 0 AC 10V 90

10是幅度, 90表示相位。

3.2.3 瞬态源

瞬态源主要有:

Pulse (PULSE function)

脉冲源

Sinusoidal (SIN function)

正弦源

Exponential (EXP function)

指数源

Piecewise linear (PWL function)

分段线性源

Single-frequency FM (SFFM function)

调频源

Single-frequency AM (AM function)

调幅源

1) 脉冲源

语法:

Vxxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>> <)>

or

`lxxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>> <)>`

时间及波形定义如下:

Time	Value
0	v1
td	v1
td + tr	v2
td + tr + pw	v2
td + tr + pw + tf	v1
tstop	v1

例子:

`VIN 3 0 PULSE (-1 1 2NS 2NS 2NS 50NS 100NS)`

低压-1V, 高压1V, 延迟2ns, 上升下降时间都是2ns, 脉冲宽度50ns, 周期100ns。

V1	99	0	PU	lv	hv	tdlay	tris	tfall	tpw	tper
				↓	↓	↓	↓	↓	↓	↓
				低压	高压	延迟	上升时间	下降时间	脉冲宽度	周期

2) 正弦源

语法:

`Vxxx n+ n- SIN <(> vo va <freq <td <θ<φ >>>> <)>`

or

`lxxx n+ n- SIN <(> vo va <freq <td <θ<φ >>>> <)>`

说明:

vo	直流幅度。
va	交流幅度。
freq	频率。
td	延迟, 正弦源开始前的延迟时间, 默认为 0。
θ	衰减因子, 单位 1/s, 默认为 0。
φ	相位延迟, 默认为 0。

例子:

`VIN 3 0 SIN (0 1 100MEG 1NS 1e10)`

直流幅度 0, 交流幅度 1V, 频率 100MHz, 初始延迟 1ns, 衰减因子 1e10。波形如图 3.1 所示。

3) 指数源

`Vxxx n+ n- EXP <(> v1 v2 <td1 <τ1 <td2 <τ2 >>>> <)>`

or

`lxxx n+ n- EXP <(> v1 v2 <td1 <τ1 <td2 <τ2 >>>> <)>`

例子:

`VIN 3 0 EXP (-4 -1 2NS 30NS 60NS 40NS)`

初始电压-4V, 最终电压-1V, 2ns 开始上升, 时间常数 30ns, 60ns 时开始下降, 时间常数 40ns。波形如图 3.2 所示。

4) 分段线性源

语法:

通常格式:

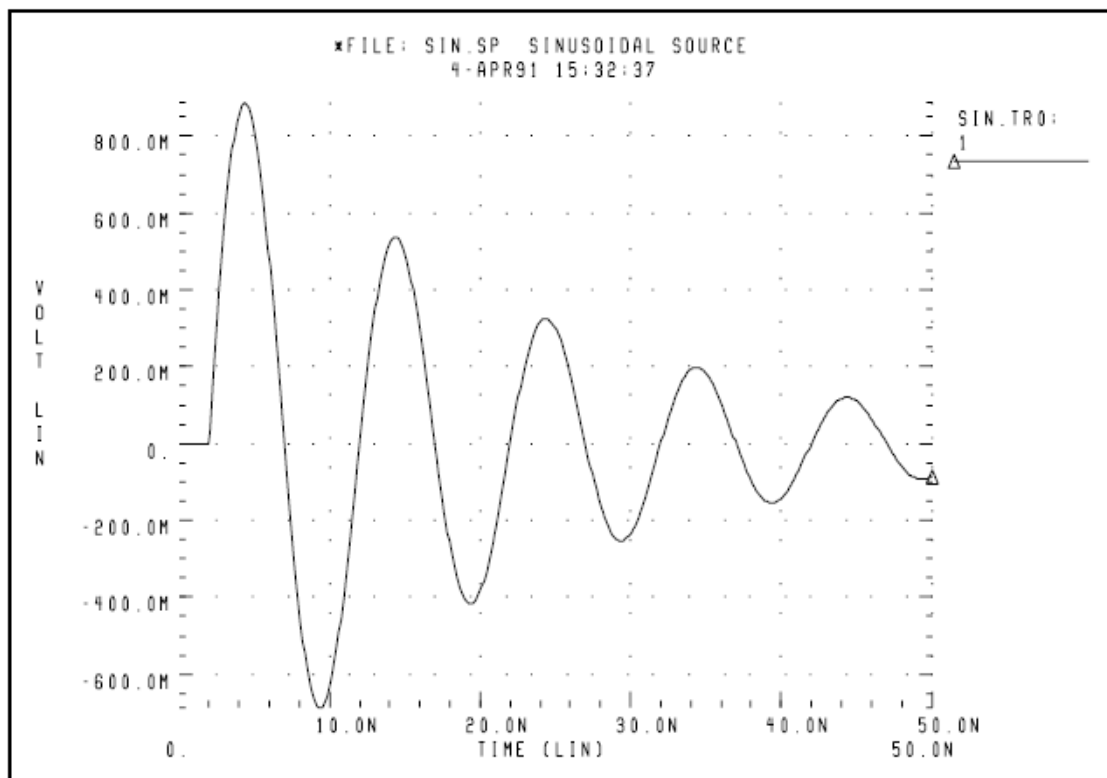


图 3.1 正弦源输出波形

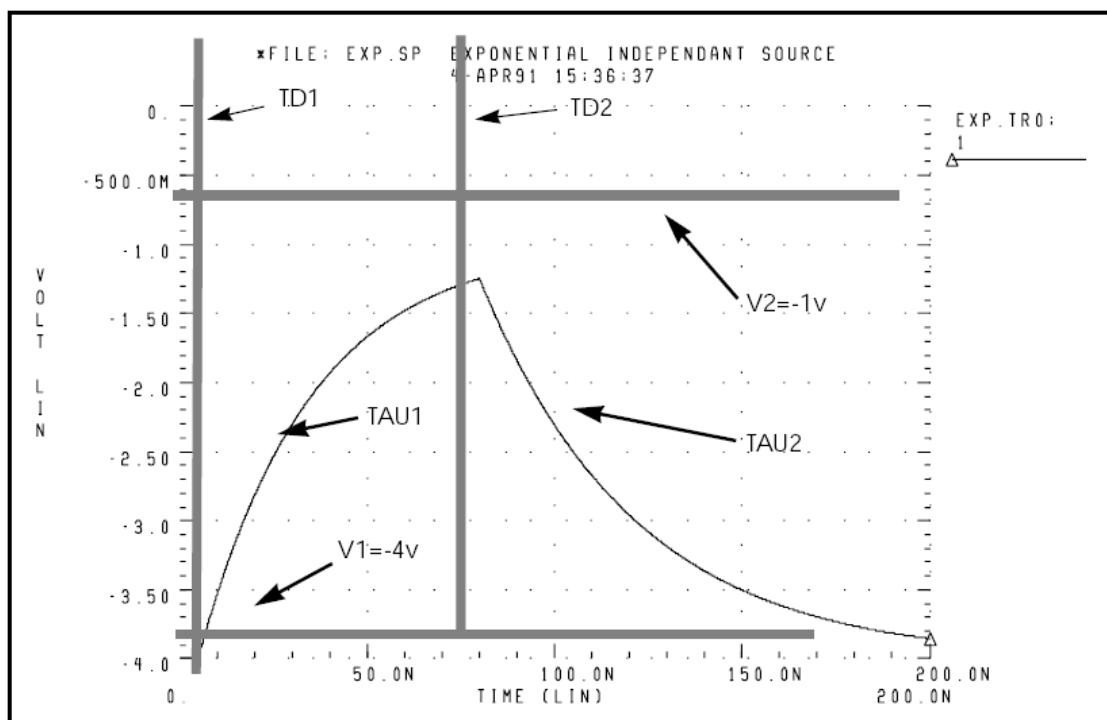


图 3.2 指数源输出波形

Vxxx n+ n- PWL (> t1 v1 <t2 v2 t3 v3...> <R <=repeat>>
+ <TD=delay> <)>

or

Ixxx n+ n- PWL (> t1 v1 <t2 v2 t3 v3...> <R <=repeat>>
+ <TD=delay> <)>

MSINC和ASPEC格式:

```
Vxxx n+ n- PL <(> v1 t1 <v2 t2 v3 t3...> <R <=repeat>>
+ <TD=delay> <)>
```

or

```
lxxx n+ n- PL <(> v1 t1 <v2 t2 v3 t3...> <R <=repeat>>
+ <TD=delay> <)>
```

说明:

(t1 v1),...(ti vi) 每个时间点对应一个电压。时间点之间线性插值填充。

R=repeat 设定循环的起始时间点, repeat 必须是 t1,t2, ... 中的某一个, 即必须是设定的时间点, 但不能是最大的那个时间点。若没有这个命令, 电压维持最后一个时间点的电压。

Td=delay 设定函数开始的延迟。

例子:

```
*FILE: PWL.SP THE REPEATED PIECEWISE LINEAR SOURCE
*ILLUSTRATION OF THE USE OF THE REPEAT FUNCTION "R"
*file pwl.sp REPEATED PIECEWISE LINEAR SOURCE
.OPTION POST
.TRAN 5N 500N
V1 1 0 PWL 60N 0V, 120N 0V, 130N 5V, 170N 5V, 180N 0V, R 0N
R1 1 0 1
V2 2 0 PL 0V 60N, 0V 120N, 5V 130N, 5V 170N, 0V 180N, R 60N
R2 2 0 1
.END
```

上例中定义了两个分段线性源, 唯一的不同是循环的起始点不同。注意声明中使用了 PWL 和 PL 两种关键字, 输出波形如图 3.3 所示。

5) 单调频源

```
Vxxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>
```

or

```
lxxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>
```

vo 直流幅度。

va 交流幅度。

fc 主频率。

mdi 调制系数, 一般取 1 到 10, 默认为 0。

fs 调制频率。

输出波形由下面的表达式定义:

Sourcevalue = vo + va×SIN [2×pi×fc×Time+mdi×SIN(2×pi×fs×Time)]

例子:

```
*FILE: SFFM.SP THE SINGLE FREQUENCY FM SOURCE
.OPTIONS POST
V 1 0 SFFM (0, 1M, 20K, 10, 5K)
R 1 0 1
.TRAN .0005M .5MS
.END
```

上例定义了一个单调频源, 直流幅度 0, 交流幅度 1mV, 载波频率 20KHz, 信号频率

5KHz，调制系数 10（最大波长大约是最小波长的十倍）。输出波形如图 3.4 所示。

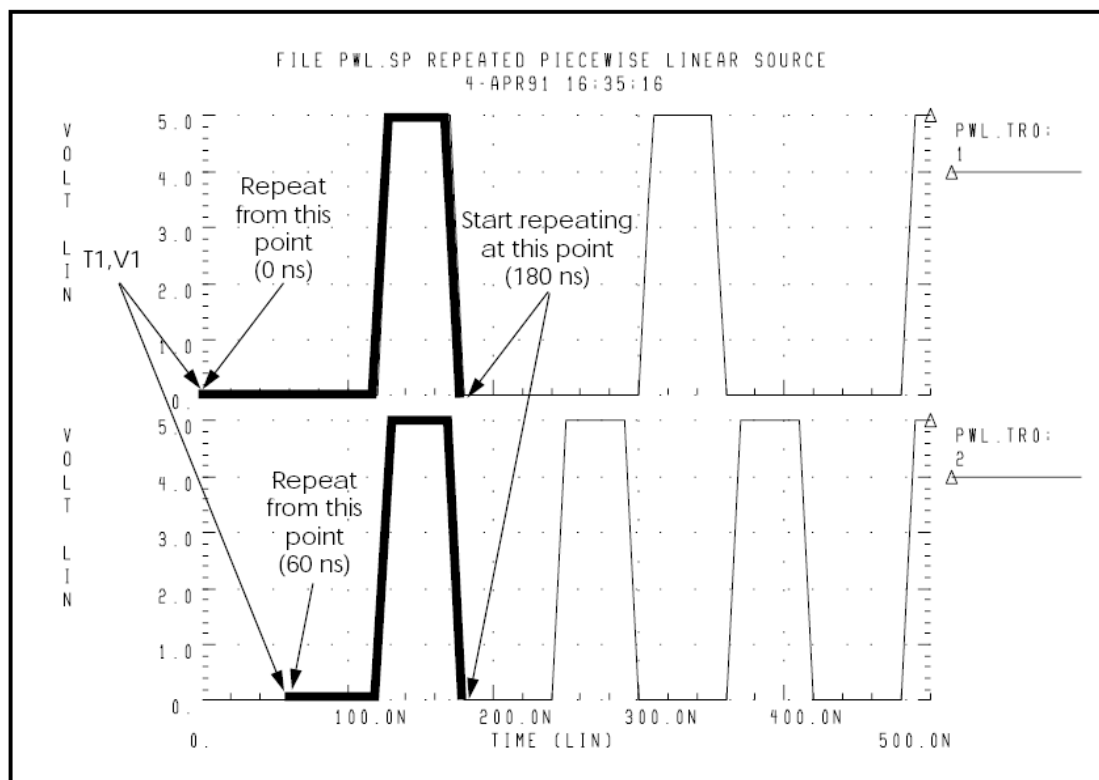


图3.3 分段线性源输出波形

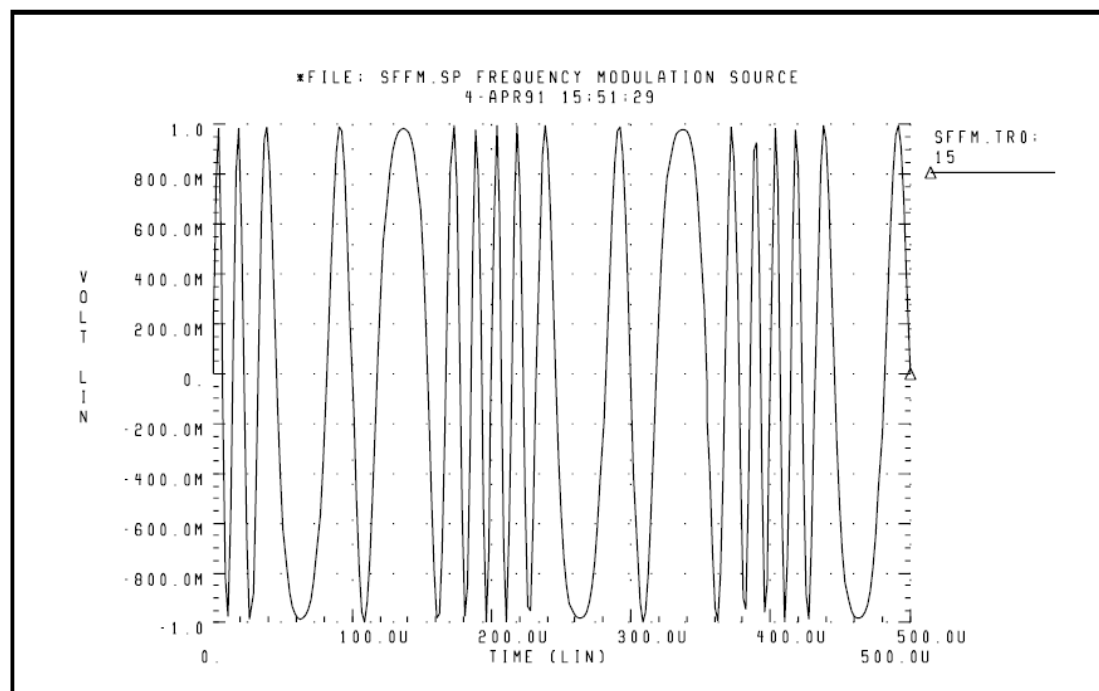


图 3.4 单调频源输出波形

6) 调幅源

Vxxx n+ n- AM <(> vo va fm fc <td> <)>

or

Ixxx n+ n- AM <(> vo va fm fc <td> <)>

说明:

vo 信号幅度。
 va 失调常数, 默认为 0。
 fm 调制频率。
 fc 载波频率。
 td 初始延迟。

输出信号由下面的公式定义:

Sourcevalue = vo · {va + SIN [2×pi×fm×(Time-td)]}
 ×SIN[2×pi×fc×(Time-td)]

例子:

```
.OPTION POST
.TRAN .01M 20M
V1 1 0 AM(10 1 100 1K 1M)
R1 1 0 1
V2 2 0 AM(2.5 4 100 1K 1M)
R2 2 0 1
V3 3 0 AM(10 1 1K 100 1M)
R3 3 0 1
.END
```

上面的网表包含三个幅度调制电压源, 第一个幅度 10V, 失调常数 1, 调制频率 100Hz, 载波频率 1KHz, 初始延迟 1ms, 第二个幅度 2.5V, 失调常数 4, 调制频率 100Hz, 载波频率 1KHz, 初始延迟 1ms, 第三个幅度 10V, 失调常数 1, 调制频率 1KHz, 载波频率 100Hz, 初始延迟 1ms。输出波形如图 3.5 所示。

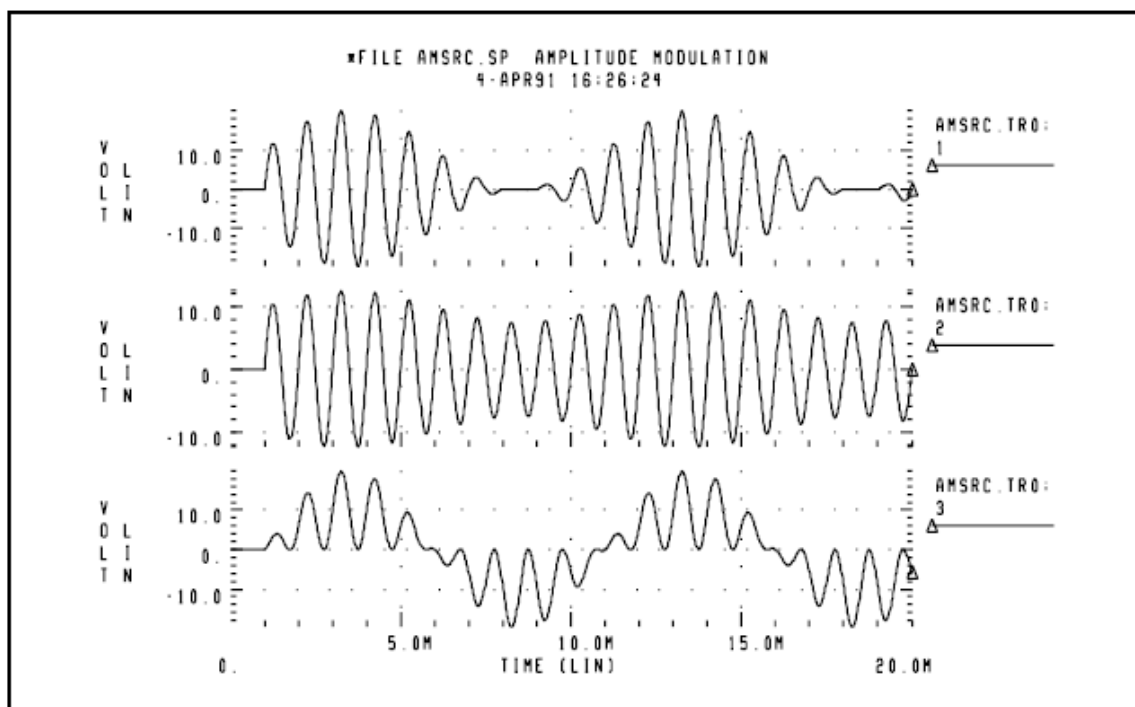


图 3.5 调幅源输出波形

3.2.4 混合源

1) 交直流混合源

VH 3 6 DC=2 AC=1,90

直流值 2V, 交流值 1V, 初始相位 90°。

2) 直流瞬态混合源

VCC in out VCC PWL 0 0 10NS VCC 15NS VCC 20NS 0

直流偏置是参数 VCC, 同时又是一个分段线性瞬态源

3) 交直流瞬态混合源

VIN 13 2 0.001 AC 1 SIN (0 1 1MEG)

直流偏置 0.001V, 交流幅度 1V, 正弦瞬态源直流幅度 0, 交流幅度 1V, 频率 1MHz。

§3.3 受控源

3.3.1 压控电压源 E elements

源定义中主要使用两种函数, 分段线性函数(piecewise linear function), 多项式 (polynomial functions)。

1) 多项式 (polynomial functions)

主要有:

poly(1) 单变量多项式

poly(2) 双变量多项式

poly(3) 三变量多项式

单变量多项式表达式:

$$FV = P0 + (P1 \times FA) + (P2 \times FA^2) + (P3 \times FA^3) + (P4 \times FA^4) + (P5 \times FA^5) \dots$$

双变量多项式表达式:

$$FV = P0 + (P1 \times FA) + (P2 \times FB) + (P3 \times FA^2) + (P4 \times FA \times FB) + (P5 \times FB^2) + (P6 \times FA^3) + (P7 \times FA^2 \times FB) + (P8 \times FA \times FB^2) + (P9 \times FB^3) + \dots$$

三变量多项式表达式:

$$FV = P0 + (P1 \times FA) + (P2 \times FB) + (P3 \times FC) + (P4 \times FA^2) + (P5 \times FA \times FB) + (P6 \times FA \times FC) + (P7 \times FB^2) + (P8 \times FB \times FC) + (P9 \times FC^2) + (P10 \times FA^3) + (P11 \times FA^2 \times FB) + (P12 \times FA^2 \times FC) + (P13 \times FA \times FB^2) + (P14 \times FA \times FB \times FC) + (P15 \times FA \times FC^2) + (P16 \times FB^3) + (P17 \times FB^2 \times FC) + (P18 \times FB \times FC^2) + (P19 \times FC^3) + (P20 \times FA^4) + \dots$$

2) 例子:

E1 5 0 POLY(1) 3 2 1 2.5

其含义是 $V(5,0) = 1 + 2.5 \times V(3,2)$ 。

E1 1 0 POLY(2) 3 2 7 6 0 3 0 0 0 4

其含义是 $V(1,0) = 3 \times V(3,2) + 4 \times V(7,6)^2$ 。

下面是一个三变量多项式的例子, 要产生如下的压控电压源:

$$V_{10} = 3V(3,2) + 4V(7,6)^2 + 5V(9,8)^3$$

三个变量及多项式的系数分别为:

FA = V(3,2)

FB = V(7,6)

FC = V(9,8)

P1=3

P7=4

P19 = 5

所以这个压控源的网表描述为:

```
E1 1 0 POLY(3) 3 2 7 6 9 8 0 3 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 5
```

3) 几种典型的压控电压源

理想运算放大器

```
Eopamp 2 3 14 1 MAX=+5 MIN=-5 2.0
```

一个对输出幅度有限制的运算放大器。其运算关系是 $V(2,3) = V(14,1) * 2$ 。Eopamp的最大幅度是5V，最小幅度是-5V。如果输入-4V，输出-5V，而不是-8V。

电压加法器

```
V(13,0)+V(15,0)+V(17,0)
```

用如下语句，产生上面的受控电压:

```
EX 18 0 POLY(3) 13 0 15 0 17 0 0 1 1 1  
+ IC=1.5,2.0,17.25
```

上面的语句中用了三变量的多项式。并且指定在直流工作点的分析中，将V(13,0), V(15,0), V(17,0) 的电压初始化为1.5V, 2.0V, 17.25V。

零延迟反相门

```
Einv out 0 PWL(1) in 0 .7v,5v 1v,0v
```

输入小于0.7V，输出5V，输入大于1V，输出0V。

理想变压器

```
Etrans out 0 TRANSFORMER in 0 10
```

电压关系是 $V(out)=V(in)/10$ 。

压控振荡器(VCO)

```
Evco out 0 VOL='v0+gain*SIN(6.28*freq*ycontrol)  
+ *TIME)'
```

输出电压由方程 $VOL='v0+gain*SIN(6.28*freq*v(control)*TIME)'$ 定义。注意VOL='equation'的这种形式可以定义任意电压。

3.3.2 压控电流源 G elements

几种典型的压控电流源:

开关

```
Gswitch 2 0 VCR PWL(1) 1 0 0v,10meg 1v,1m
```

压控电阻具有基本的开关特性。当节点1, 0间的电压从0V变为1V时节点2, 0间的电阻从10兆欧姆线性变为1毫欧姆。超出电压范围，电阻分别保持10兆欧姆和1毫欧姆。

开关MOS管

```
Gnmos d s VCR NPWL(1) g s LEVEL=1 0.4v,150g  
+ 1v,10meg 2v,50k 3v,4k 5v,2k
```

漏源掉换时电阻不变。

压控电容

```
Gcap out 0 VCCAP PWL(1) ctrl 0 2v,1p 2.5v,5p
```

当节点ctrl, 0间的电压从2V变为2.5V时节点out, 0间的电容从1pF线性变为5pF。超出电压范围, 电容分别保持1pF和5pF。

零延迟门

```
Gand out 0 AND(2) a 0 b 0 SCALE='1/rload' 0v,0a 1v,.5a
+ 4v,4.5a 5v,5a
```

上面的语句定义了一个二输入的与门。输出电压是输出电流乘上scale, 此处的scale定义为输出电阻的倒数。

延迟器件

```
Gdel out 0 DELAY in 0 TD=5ns SCALE=2 NPDELAY=25
```

延迟器件是一种低通器件。上例中, 节点out和0之间的电流由节点in和0之间的电压决定。电流值是scale*V(in,0), 延迟5ns。

二极管

```
Gdio 5 0 CUR='1e-14*(EXP(V(5)/0.025)-1.0)'
```

二极管是一个电压控制电流器件。控制方程由上面的语句定义。

二极管击穿

```
Gdiode 1 0 PWL(1) 1 0 -2.2v,-1a -2v,-1pa .3v,.15pa
+ .6v,10ua 1v,1a 2v,1.2a
```

二极管击穿可用分段线性源来模拟。如上面的语句所定义, 当控制电压超出范围(-2.2V,2V), 电流将分别维持不变(-1a,1.2a)。

三极管

```
gt i_anode cathode poly(2) anode,cathode grid,cathode 0 0
+0 0 .02
gt i_anode cathode
+cur='20m*v(anode,cathode)*v(grid,cathode)'
```

可以用压控电流源来实现基本的三极管功能。上面的两条定义语句是等价的, 第一个采用 poly(2)定义, 第二个采用方程定义。

第四章 参数、函数及仿真设置

§4.1 参数

4.1.1 参数定义

主要的参数定义方式有如下几种:

<code>.PARAM <SimpleParam> = 1e-12</code>	简单定义
<code>.PARAM <AlgebraicParam> = 'SimpleParam*8.2'</code>	代数定义
<code>.PARAM <MyFunc(x, y)> = 'Sqrt((x*x)+(y*y))'</code>	用户定义函数
<code>.SUBCKT <SubName> <ParamDefName> = + <Value></code>	子电路定义
<code>.MEASURE <DC AC TRAN> result TRIG ... + TARG ... <GOAL=val> <MINVAL=val> + <WEIGHT=val> <MeasType> <MeasParam></code>	<code>.measure</code> 声明

4.1.2 .param 声明

例子:

```
.PARAM TermValue = 1g
.PARAM Pi= '355/113'
.PARAM Pi2= '2*Pi'
.PARAM npRatio= 2.1
.PARAM nWidth= 3u
.PARAM pWidth= 'nWidth * npRatio'
```

上面是简单的定义及代数定义。

```
.PARAM CentToFar (c) = '(((c*9)/5)+32)'
.PARAM F(p1,p2) = 'Log(Cos(p1)*Sin(p2))'
.PARAM SqrdProd (a,b) = '(a*a)*(b*b)'
```

上面是函数定义方式。

4.1.3 指令行内定义

```
r1 n1 0 R='1k/sqrt(HERTZ)' $ Resistance related to frequency.
```

上面是指令行内定义。

4.1.4 代数表达式定义输出参数

语法:

```
PAR('algebraic expression')
```

例子:

```
.PRINT DC v(3) gain=PAR('v(3)/v(2)') PAR('v(4)/v(2)')
```

上面是在输出指令中定义参数, 使用关键字 `par`。

4.1.5 倍乘参数 M (the multiply parameter)

倍乘参数 `M` 可用于任何器件 (除电压源), 它将内部参数值乘上倍乘参数 `M`, 来产

生 M 个器件或子电路并联的效果，如需仿真两个输出缓冲同时切换的效果，只须调用一次子电路，如下：

X1 in out buffer M=2

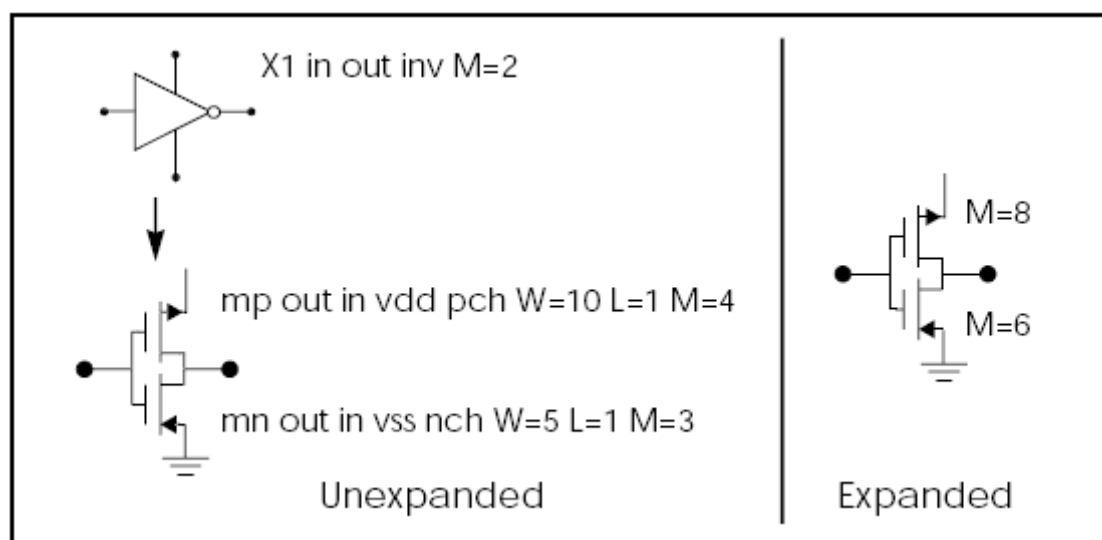


图 4.1 M 参数作用示意

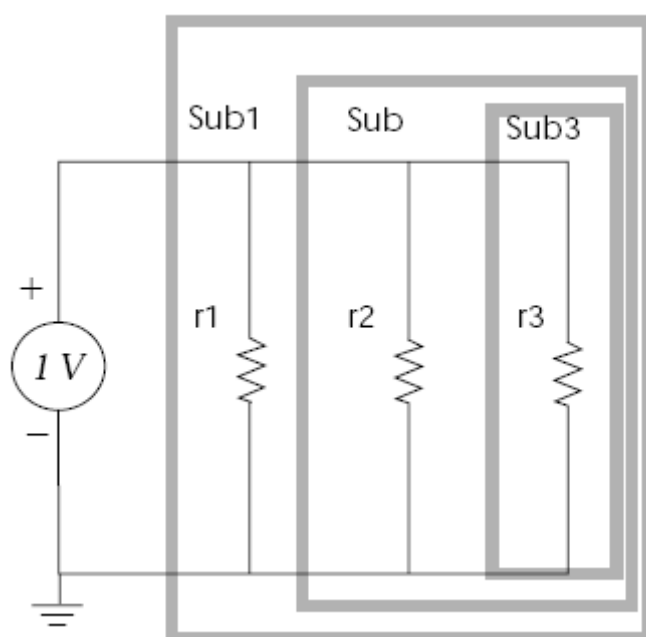
4.1.6 参数作用范围

语法：

.OPTION PARHIER = < GLOBAL | LOCAL >

当定义了参数并设置了初值，又在电路中碰到相同名称的参数并对其赋值（如对子电路赋初值）时，应该格外关注参数的作用范围，不同情况下的结果是完全不同的。当使用 .option parhier=global 声明时，全局参数将作用于子电路，将子电路初始值替换掉。

当使用 .option parhier=local 声明时，局部参数赋值将会获得高优先级，不会被全局参数的初值替换掉。下面的例子会说明这两种情况。



TEST OF PARHIER

```
.OPTION list node post=2
+ ingold=2
+ parhier=<Local|Global>
.PARAM Val=1
x1 n0 0 Sub1
.SubCkt Sub1 n1 n2 Val=1
r1 n1 n2 Val
x2 n1 n2 Sub2
.Ends Sub1
.SubCkt Sub2 n1 n2 Val=2
r2 n1 n2 Val
x3 n1 n2 Sub3
.Ends Sub2
.SubCkt Sub3 n1 n2 Val=3
r3 n1 n2 Val
```

.Ends Sub3

电阻网络通过子电路嵌套定义，主电路用.PARAM Val=1定义了参数Val及其值。如果使用.option global声明，子电路定义中的Val值将全部被替换，最终三个电阻值都是1Ω，电阻网络的阻值是0.3333Ω。如果使用.option local声明，子电路定义中的Val值将不会被替换，最终三个电阻值分别是1Ω、2Ω、3Ω，电阻网络的阻值是0.5455Ω。

下面是另一个例子

```
.Param Wid = 5u                                $ Default Pulse Width for source
v1 Pulsed 0 Pulse ( 0v 5v 0u 0.1u 0.1u Wid 10u )
...
* Subcircuit default definition
.SubCkt Inv A Y Wid = 0                        $ Inherit illegals by default
mp1 <NodeList> <Model> L=1u W='Wid*2'
mn1 <NodeList> <Model> L=1u W=Wid
.Ends
* Invocation of symbols in a design
x1 A Y1 Inv                                    $ Incorrect width!
x2 A Y2 Inv Wid = 1u                          $ Incorrect! Both x1 and x2
                                              $ simulate with mp1=10u and
                                              $ mn1=5u instead of 2u and 1u.
```

默认情况下全局参数具有最高优先级，可以替换掉后面的以及子电路中同名的参数的值。上例中参数Wid用于定义脉冲宽度，可是与子电路中的器件宽度同名，于是参数Wid的取值统统变为5u。

子电路定义中默认Wid=0的目的是要求调用时必须给Wid重新赋值，否则仿真会异常中断，这是一种很重要的技术。本来需要仿真在X1处中断，X2处赋值使Wid=1u，但是实际上X1处不会中断，X2处也不会赋值Wid=1u，X1，X2都为Wid赋值为5u。

为防止参数名取值相同带来的赋值混乱，仿真错误，应采取以下措施

- 确保子电路及主电路中参数名的唯一性。
- 设置两次仿真，一次设置.option parhier=global，一次设置.option parhier=local，检查两次仿真的结果是否相同。
- 在调用大量不同人编写的子电路及库的情况下，为确保子电路与设计时一致，设置.option parhier=local。

§4.2 函数

4.2.1 用户定义函数

语法：

```
fname1 (arg1, arg2) = expr1 (fname2 (arg1, ...) = expr2)
```

例子：

```
f(a,b)=POW(a,2)+a*b
g(d)=SQRT(d)
h(e)=e*f(1,2)g(3)
```

4.2.2 内置函数

除了算术运算符(+, -, *, /)之外, Hspice 提供了很多内置函数。如表 1 所示。

表 1 Hspice 内置函数

HSPICE Form	函数	类型	描述
sin(x)	sine	trig	返回x(弧度)的正弦值
cos(x)	cosine	trig	返回 x(弧度)的余弦值
tan(x)	tangent	trig	返回 x(弧度)的正切值
asin(x)	arc sine	trig	返回 x(弧度)的反正弦值
acos(x)	arc cosine	trig	返回 x(弧度)的反余弦值
atan(x)	arc tangent	trig	返回 x(弧度)的反正切值
sinh(x)	hyperbolic sine	trig	返回 x(弧度)的双曲正弦值
cosh(x)	hyperbolic cosine	trig	返回 x(弧度)的双曲余弦值
tanh(x)	hyperbolic tangent	trig	返回 x(弧度)的双曲正切值
abs(x)	absolute value	math	返回 x 的绝对值 x
sqrt(x)	square root	math	返回 x 的平方根
pow(x,y)	absolute power	math	返回值是 $x^{\text{integer part of } y}$
pwr(x,y)	signed power	math	返回值是(sign of x) x ^y
log(x)	natural logarithm	math	返回值是(sign of x)log(x)
log10(x)	base 10 logarithm	math	返回值是(sign of x)log ₁₀ (x)
exp(x)	exponentia	math	返回值是 e^x
db(x)	decibels	math	返回值是(sign of x)20log ₁₀ (x)
int(x)	integer	math	返回小于等于 x 的最大整数
sgn(x)	return sign	math	x>0 返回 1,x<0 返回-1,x=0 返回 0
sign(x,y)	transfer sign	math	返回值是(sign of y) x
min(x,y)	smaller of two args	control	返回 x,y 中较小值
max(x,y)	larger of two args	control	返回 x,y 中较大值
lv(<Element>) or lx(<Element>)	element templates	various	返回仿真中的不同器件值
v(<Node>), i(<Element>)...	circuit output variables	various	返回仿真中的不同电路值

4.2.3 保留变量

Hspice 还保留了三个变量，供某些器件如 E,G,R,C,L 等调用。但不能用于其他目的（如.param 声明），如表 2 所示。

表 2 Hspice 保留变量

HSPICE Form	函数	类型	描述
time	current simulation time	control	瞬态分析中的当前仿真时间参数
temper	current circuit temperature	control	当前仿真温度参数
hertz	current simulation frequency	control	交流分析中的当前频率参数

§4.3 仿真设置

4.3.1 设置控制选项（control options）

.options 声明

语法：

.OPTIONS opt1 <opt2 opt3 ...>

例子：

.OPTIONS BRIEF \$ Sets BRIEF to 1 (turns it on)

* Netlist, models,

...

.OPTIONS BRIEF=0 \$ Turns BRIEF off

4.3.2 基本控制选项

1) 输入输出选项

ACCT 在输出报告文件结尾增加任务计数及仿真时间统计，此功能默认是打开的。**ACCT** 的选项有：

0 取消报告

1 允许报告

2 允许矩阵统计报告

brief 简化仿真报告

CO=x 设置输出中的列数。可以为80列或132列，默认为80。

ingold = x 设定输出数据格式，默认 ingold=0，设置为 2 可以与 SPICE 工具兼容，ingold 的选项有：

ingold=0 工程格式，指数被表示成单个字母：

1G=1e9 1X=1e6 1K=1e3 1M=1e-3

1U=1e-6 1N=1e-9 1P=1e-12 1F=1e-15

ingold=1 固定与指数共用格式，数值为 0.1 到 999 之间时，直接表示。小于 0.1 或大于 999 表示为指数形式

ingold=2 纯指数格式，可与数据后处理工具兼容。

注意，将.options measdgt 与 ingold 共同使用来控制.measure 的输出数据格式。

list

产生器件数目及关键参数值的摘要。

node

列出跟每一个节点相连的所有器件。

nomod

不输出模型参数。

search

设置库和包含文件的搜索路径。

2) 界面选项

post

允许保存图形界面的数据。post=1,保存为二进制格式。post=2,保存为 ASCII 格式。post=3,保存为新波形二进制格式。默认为 1。

probe

限制输出数据为.print, .plot, .probe, graph 中指定的变量。默认情况下，Spice 输出所有的电压、电流数据，再加上输出命令中指定的数据。用 probe 可以显著减小输出文件大小。

3) 仿真选项

parhier

设置参数优先级，应用于不同层级电路中的重名参数。选项有：

local 较低层级的电路参数具有高优先级

global 较高层级的电路参数具有高优先级

第五章 输出设置

§5.1 输出指令

<code>.print</code>	在输出列表文件(或屏幕提示)中输出数值结果。如果使用了 <code>.option post</code> 也会输出图形数据。
<code>.plot</code>	在输出列表文件(或屏幕提示)中输出低分辨率的点。如果使用了 <code>.option post</code> 也会输出图形数据。
<code>.graph</code>	输出高分辨率的图形数据。不仅应用于图形处理的后处理程序, 在没有图形处理程序的时候, 也可以用某些设备打印(如 HP 激光打印机)。
<code>.probe</code>	输出图形数据, 但不在输出列表文件(或屏幕提示)中输出数据。可用 <code>.option probe</code> 来限制输出, 即只输出 <code>.probe</code> 中指定的数据。
<code>.measure</code>	在输出列表文件(或屏幕提示)中输出用户定义的特定分析的数据。如果使用了 <code>.option post</code> 也会输出图形数据。

例子:

`.print` 声明

```
.PRINT TRAN V(4) I(VIN) PAR('V(OUT)/V(IN)')
```

输出瞬态分析中的 4 节点电压, VIN 节点电流, OUT 和 IN 节点电压的比值。

```
.PRINT AC VM(4,2) VR(7) VP(8,3) II(R1)
```

输出交流分析中的相关结果, VM(4,2)是 4 节点和 2 节点之间的电压幅度(magnitude), VR(7)是 7 节点电压实部(real part), VP(8,3)是 8 节点和 3 节点之间的电压相位差(phase), II(R1)是通过电阻 R1 的电流虚部(imaginary part)。

```
.PRINT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)
```

输出交流分析中的相关结果, 输入阻抗 ZIN, 输出导纳的相位 YOUT(P), 还有 S 及 Z 参数。此条指令需配合`.AC`, `.NET` 指令使用。

```
.PRINT DC V(2) I(VSRC) V(23,17) I1(R1) I1(M1)
```

输出直流分析中相关结果, 电压源 VSRC, MOS 管 M1 上的电流等等。

```
.PRINT NOISE INOISE
```

输出等效输入噪声。

```
.PRINT pj1=par('p(rd) +p(rs)')
```

输出指定的函数值。

`.plot` 声明

```
.PLOT AC VM(5) VM(31,24) VDB(5) VP(5) INOISE
```

`.probe` 声明

```
.PROBE DC V(4) V(5) V(1) beta=PAR('I1(Q1)/I2(Q1)')
```

§5.2 输出参数

5.2.1 直流和瞬态分析输出参数

1) 节点电压

语法:

```
V (n1<,n2>)
```

2) 电压源电流

语法:

I (Vxxx)

例子:

```
.PLOT TRAN I(VIN)
.PRINT DC I(X1.VSRC)
.PLOT DC I(XSUB.XSUBSUB.VY)
```

3) 器件支路电流

语法:

In (Wwww)

说明:

n 指器件声明中的节点序号, 如果器件有四个节点, 则 I3 指声明中的第三个节点, 如果不指定 n, 则默认为 1。

例子:

I1(R1)

电阻 R1 的 1 节点的电流。

I4(X1.M1)

子电路 X1 中 MOS 管 M1 的第四个节点的电流。

5.2.2 功率

Hspice 可以计算无源及有源器件消耗及存储功率, 对于半导体器件, 只计算消耗功率。Hspice 也可以计算整个电路的功率, 以及子电路的功率。

可以用 .print 和 .plot 声明输出器件功率及总功率。

语法:

.PRINT <DC | TRAN> P(element_or_subcircuit_name)POWER

功率计算只对瞬态分析及直流扫描分析有效。measure 声明可以用来计算平均、平方根、最大、最小功率等等。POWER 关键字指示输出总消耗功率。

例子:

```
.PRINT TRAN P(M1) P(VIN) P(CLOAD) POWER
.PRINT TRAN P(Q1) P(DIO) P(J10) POWER
.PRINT TRAN POWER      $ Total transient analysis power
* dissipation
.PLOT DC POWER P(IIN) P(RLOAD) P(R1)
.PLOT DC POWER P(V1) P(RLOAD) P(VS)
.PRINT TRAN P(Xf1) P(Xf1.Xh1)
```

5.2.3 交流分析输出参数

表 3 是交流分析输出变量类型的通用符号。

1) 节点电压

语法:

Vx (n1,<,n2>)

说明:

X 指定输出类型。

例子:

```
.PLOT AC VM(5) VDB(5) VP(5)
```

表 3 交流分析输出变量类型

Type Symbol	Variable Type
DB	decibel
I	imaginary part
M	magnitude
P	phase
R	real part
T	group delay

2) 独立电压源电流

语法:

```
Iz (Vxxx)
```

说明:

Z 指定输出类型。

例子:

```
.PLOT AC IR(V1) IM(VN2B) IP(X1.X2.VSRC)
```

3) 器件支路电流

语法:

```
Izn (Wwww)
```

说明:

z 指定输出类型。

n 器件节点序号。

例子:

```
.PRINT AC IP1(Q5) IM1(Q5) IDB4(X1.M1)
```

5.2.4 网路相关参数

语法:

```
Xij (z), ZIN(z), ZOUT(z), YIN(z), YOUT(z)
```

说明:

X 设为 Z 指阻抗, Y 指导纳, H 指混合参数, S 指散射参数。

ij 二端口网络序号。

z 输出类型, 如果省略, 则指幅度。

ZIN 输入阻抗。

ZOUT 输出阻抗。

YIN 输入导纳。

YOUT 输出导纳。

例子:

```
.PRINT AC Z11(R) Z12(R) Y21(I) Y22 S11 S11(DB)
```

```
.PRINT AC ZIN(R) ZIN(I) YOUT(M) YOUT(P) H11(M)
```

```
.PLOT AC S22(M) S22(P) S21(R) H21(P) H12(R)
```

5.2.5 噪声和谐波分析输出参数

语法:

ovar <(z)>

说明:

ovar 噪声和谐波分析参数。如果是ONNOISE指输出噪声，INOISE指等效输入噪声。或是任意的谐波分析参数(HD2,HD3,SIM2,DIM2,DIM3)。

Z 输出类型。

例子:

```
.PRINT DISTO HD2(M) HD2(DB)
.PLOT NOISE INOISE ONNOISE
```

5.2.6 器件参数输出

语法:

Elname:Property

说明:

Elame 器件名

Property 器件的属性名，如用户定义的参数，状态变量，存储的电荷，电容，电流，变量的导数等等（具体可查阅 Hspice 用户手册）

例子:

```
.PLOT TRAN V(1,12) I(X2.VSIN) I2(Q3) DI01:GD
.PRINT TRAN X2.M1:CGGBO M1:CGDBO X2.M1:CGSBO
```

第六章 常用分析

§6.1 直流初始化及工作点分析

6.1.1 电路初始化

电路仿真求解时，从一组初始值开始不断迭代获得精确解。因此初始值会很大的影响迭代次数及仿真时间。尽管 Hspice 会用默认估计值对电路初始化，某些时候需要人为设置某些节点的初始值，以使电路求解能够收敛或进入期望的工作状态。

.op, .dc 扫描, .ac 及 .tran 分析的第一步都是设置直流工作点。

使用 .ic 及 .nodeset 声明来设置工作点初值。

1) 瞬态分析初始化

默认情况下，瞬态分析从直流分析计算所得工作点开始，但当在 .tran 声明中包含了 UIC 关键字之后，瞬态分析会使用 .ic 声明中指定的初始值。

2) 器件声明中的 ic 参数

在器件声明中使用 IC=<val> 来为器件设置初值。

例子:

```
HXCC 13 20 VIN1 VIN2 IC=0.5, 1.3
```

上例中是电压控制的电流源，VIN1 和 VIN2 的初始电流分别设为 0.5mA 和 1.3mA。

3) .IC and .DCVOLT 初始条件声明

.IC 和 .DCVOLT 声明是等效的，用来设定瞬态分析中的初始条件。如果 .tran 声明中使用 UIC 关键字，则以 .IC 声明中的值为零时刻的初始值。如果 .tran 声明中没有使用 UIC 关键字，则计算直流工作点作为初始值。

语法:

```
.IC V(node1) = val1 V(node2) = val2 ...
```

or

```
.DCVOLT V(node1) = val1 V(node2) = val2 ...
```

例子:

```
.IC V(11)=5 V(4)=-5 V(2)=2.2
```

```
.DCVOLT 11 5 4 -5 2 2.2
```

上面两条声明是等效的。

4) .nodeset 声明

用来设定节点电压初始值。

语法:

```
.NODESET V(node1)=val1 <V(node2)=val2 ...>
```

or

```
.NODESET node1 val1 <node2 val2>
```

例子:

```
.NODESET V(5:SETX)=3.5V V(X1.X2.VINT)=1V
```

```
.NODESET V(12)=4.5 V(4)=2.23
```

```
.NODESET 12 4.5 4 2.23 1 1
```

上面三条声明等效。

6.1.2 工作点分析 (operating point) .op 声明

.op 声明会计算电路的直流工作点，对于需要直流工作点的分析，如瞬态分析，不管有没有.op 声明，都会事先计算直流工作点。.op 也可以输出瞬态分析时的工作点。
例：

```
.op
产生全部直流工作点，包括直流功耗。
.op .5ns cur 10ns vol 17.5ns 20ns 25ns
计算直流工作点，以及瞬态分析中 10ns 时的电流，17.5ns, 20ns, 25ns 时的电压。
```

§6.2 直流扫描分析

6.2.1 .dc 声明

语法(未详细列出):

扫描或参数扫描:

```
.DC var1 START=<param_expr1> STOP=<param_expr2>
+ STEP=<param_expr3>
or
.DC var1 start1 stop1 incr1 <SWEEP var2 type np start2 stop2>
```

蒙特卡罗分析 (Monte Carlo) :

```
.DC var1 type np start1 stop1 <SWEEP MONTE=val>
or
.DC MONTE=val
```

说明:

type 设定扫描类型，可以是以下四种:

- DEC — decade variation
- OCT — octave variation
- LIN — linear variation
- POI — list of points

MONTE=val 随机产生val个参数，供分析使用。参数的分布可以是高斯分布 (Gaussian)、单调分布(Uniform)、随机限制分布(Random Limit)。

6.2.2 例子

```
.dc vin 0.25 5.0 0.25
从 0.25V 到 5.0V 扫描 vin，步长 0.25V。
.dc vds 0 10 0.5 vgs 0 5 1
扫描两个参量的组合，从 0 到 10V，步长 0.5V，扫描 vds，对每一个 vds，从 0 到 5V，步长 1V，扫描一遍 vgs。
.dc temp -55 125 10
温度扫描，从 -55℃到 125℃，步长 10℃
.dc temp poi 5 0 30 50 100 125
温度扫描，针对 5 个点，分别是 0℃，30℃，50℃，100℃，125℃。
.dc xval 1k 10k .5k sweep temp
+ lin 5 25 125
对电阻 xval 和温度进行组合扫描，从 1k 到 10k，步长 0.5k 扫描电阻 xval，从 25℃到
```

125℃线性地取 5 个点扫描温度。

```
.dc data=dataname sweep par1 dec 10 1k 100k
```

对两个参量的组合进行扫描，dataname 是在.data 指令中定义的一组数据，另外从 1k 到 100k 每 10 倍取 10 个点扫描变量 par1。 .data 指令定义数据举例如下：

```
.data def width length thresh cap
+      50u    30u    1.2V    1.2pf
+      25u    15u    1.0V    0.8pf
+      5u     2u     0.8V    0.6pf
+      .....
.enddata
```

上例中 def 即为 dataname，每组数据包含 width, length, thresh, cap 四个变量，每组取值即为每行的值，如 (50u, 30u, 1.2V, 1.2pf) 等等。

```
.dc par1 dec 10 1k 100k sweep monte=30
```

进行 30 次蒙特卡罗(monte carlo)分析，对于每次分析，都从 1k 到 100k 每 10 倍取 10 个点扫描变量 par1。

6.2.3 其他直流分析声明

下面的分析都使用电路的直流等效模型。对.pz 分析，电感电容都包括在等效电路中。

.pz 零极点分析。
.sens 使用直流小信号模型，电路参数对于指定的输出变量的敏感性分析。
.tf 计算直流小信号模型传输函数的值（输出变量对输入源的比值）。

1) .sens 声明

计算的是输出变量对某个电路参数的偏微分值，并将其归一化，因此对所有电路参数敏感性分析的值的和是 1。

语法：

```
.SENS ov1 <ov2 ...>
```

例子：

```
.SENS V(9) V(4,3) V(17) I(VCC)
```

2) .tf 声明

直流小信号传输函数分析，当使用.tf 声明时，将计算直流小信号传输函数的值（输出/输入），输入阻抗，输出阻抗。

语法：

```
.TF ov srcnam
```

例子：

```
.TF V(5,3) VIN
.TF I(VLOAD) VIN
```

3) .pz 声明

零极点分析

语法：

```
.PZ ov srcnam
```

说明：

ov 输出变量。
srcnam 输入源。

例子:

```
.PZ V(10) VIN
.PZ I(RL) ISORC
```

§6.3 瞬态分析

6.3.1 瞬态分析的初始化

瞬态分析的初始值采用直流工作点，对于振荡器或某些闭环电路，可能不存在稳定的工作点。这种情况下，需要将环断开计算直流工作点，或者用.ic 声明对电路初始化。

6.3.2 瞬态分析 .tran 声明

语法: (未全部列出)

Single-point analysis:

```
.TRAN var1 START=start1 STOP=stop1 STEP=incr1
```

or

```
.TRAN var1 START=<param_expr1> STOP=<param_expr2>
+ STEP=<param_expr3>
```

Double-point analysis:

```
.TRAN var1 START=start1 STOP=stop1 STEP=incr1
+ <SWEEP var2 type np start2 stop2>
```

or

```
.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN>
+ <START=val> <UIC> + <SWEEP var pstart
+ pstop pincr>
```

6.3.3 例子

```
.tran 1ns 100ns
```

进行 100ns 的瞬态分析，步长 1ns。

```
.tran .1ns 25ns 1ns 40ns start=10ns
```

前 25ns 步长 0.1ns，后 40ns 步长 1ns，从 10ns 的时候开始文本及图形数据输出。

```
.tran 10ns 1us uic
```

进行 1us 的分析，步长 10ns，uic 表示忽略初始直流工作点计算，采用.ic 声明中设定的初始节点电压来计算初始条件。

```
.TRAN 10NS 1US UIC SWEEP TEMP -55 75 10
```

从 -55℃ 到 125℃，步长 10℃ 扫描温度，在每一个温度值下进行瞬态分析。

```
.TRAN 10NS 1US SWEEP load POI 3 1pf 5pf 10pf
```

在不同的电容值 load 下进行分析，load 可以取三个值，分别是 1pf, 5pf, 10pf。

```
.tran data=dataname
```

用.data 中定义的数据或文件进行瞬态分析。

6.3.4 傅立叶分析

.fft 声明

语法:

```
.FFT <output_var> <START=value> <STOP=value> <NP=value>
+ <FORMAT=keyword> <WINDOW=keyword> <ALFA=value> <FREQ=value>
+ <FMIN=value> <FMAX=value>
```

说明:

表 4 .fft 分析参数说明

参数	默认值	描述
output_var	---	任意有效输出变量
START	参考描述	设定对输出变量分析的初始时刻，默认值是.tran 声明中设定的 START 值，START 默认值是 0
FROM	参考 START	STAT 的别名
STOP	参考描述	设定对输出变量分析的结束时刻，默认值是.tran 声明中的 TSTOP 值
TO	参考 STOP	STOP 的别名
NP	1024	设定 FFT 分析中采样点的数目，必须是 2 的幂
FORMAT	NORM	设定输出数据格式: NORM 归一化幅度 UNORM 非归一化幅度
WINDOW	RECT	设定使用的输出窗口 RECT=simple rectangular truncation window BART=Bartlett (triangular) window HANN=Hanning window HAMM=Hamming window BLACK=Blackman window HARRIS=Blackman-Harris window GAUSS=Gaussian window KAISER=Kaiser-Bessel window
ALFA	3.0	设定 GAUSS, KAISER 窗口中的相关参数
FREQ(重要)	0(Hz)	指定感兴趣的频率。如果是非零值，则输出限制在这个频率的谐波，频率在 FMIN 和 FMAX 之间，这些谐波的 THD 也会输出
FMIN	1.0/T(Hz)	设定输出的最小频率 $T=(STOP-START)$
FMAX	$0.5*NP*FMIN$ (Hz)	设定输出的最大频率

例子:

```
.fft v(1)
.fft v(1,2) np=1024 start=0.3m stop=0.5m freq=5.0k
+ window=kaiser alfa=2.5
.fft l(rload) start=0m to=2.0m fmin=100k fmax=120k
+ format=unorm
.fft par('v(1) + v(2)') from=0.2u stop=1.2u window=harris
```

注意:

- FFT 分析中只能有一个输出变量。
`.fft v(1) v(2) np=1024`
 错误的声明
`.fft v(1) np=1024`
`.fft v(2) np=1024`
 正确的声明
- 输出频率的步长 = $1.0/(\text{STOP} - \text{START})$ ，为获得更好的频率分辨率，最大化时间窗口。
- FMIN和FMAX对图形输出文件 `.ft0`, `.ft1`, ..., `.ftn` 没有作用。

§6.4 交流分析

交流分析的输出是频率的函数。分析前先解直流工作点，以此建立起线性小信号模型。允许交直流分析使用不同的电阻值，如果在电阻声明中使用 `AC=<value>`，则电阻在交直流分析中会采用不同的阻值。

6.4.1 交流分析 `.ac` 声明

语法(未全部列出):

Single/double sweep:

```
.AC type np fstart fstop
or
.AC type np fstart fstop <SWEEP var start stop incr>
or
.AC type np fstart fstop <SWEEP var type np start stop>
or
.AC var1 START= <param_expr1> STOP= <param_expr2>
+ STEP = <param_expr3>
or
.AC var1 START = start1 STOP = stop1 STEP = incr1
```

6.4.2 例子

```
.AC DEC 10 1K 100MEG
频率扫描，从 1KHz 到 100MHz，每十倍频取十个点。
.AC LIN 100 1 100HZ
从 1 到 100Hz 线性取 100 个点。
.AC DEC 10 1 10K SWEEP cload LIN 20 1pf 10pf
交流分析中对电容 cload 的值进行线性扫描，从 1pF 到 10pF 之间线性取 20 个点。
.AC DEC 10 1 10K SWEEP rx POI 2 5k 15k
交流分析中对电阻 rx 进行扫描，rx 可以取两个值，5KΩ 和 10KΩ。
.AC DEC 10 1 10K SWEEP DATA=datanm
交流分析中对 DATA=datanm 中定义的参数进行扫描。
.AC DEC 10 1 10K SWEEP MONTE=30
交流分析中进行 30 次蒙特卡罗分析，每次都从 1 到 10KHz 之间每十倍频取 10 个点扫描频率。
```


6.4.3 其他交流分析

1) 交流小信号谐波分析 .disto 声明

程序会计算负载电阻上的五种谐波, 谐波计算采用两个基频, 第一个基频 F1, 是.ac 分析中设定的扫描频率, 第二个频率 F2, 根据参数 skw2 确定, $skw2=F2/F1$ 。

DIM2	二阶交调($F1 - F2$)
DIM3	三阶交调($2*F1-F2$)
HD2	二阶交调 $2*F1$ (ignoring F2)
HD3	三阶交调 $3*F1$ (ignoring F2)
SIM2	二阶交调($F1 + F2$)

语法:

```
.DISTO Rload <inter <skw2 <refpwr <spwf>>>>
```

说明:

Rload	负载电阻。
inter	谐波分析的摘要输出间隔。设定一些交流分析输出的频率点。如果忽略或设为零, 不会输出任何摘要。可用来限制输出信息的多少。
skw2	$skw2=F2/F1$, $1e-3 < skw2 \leq 1$, 默认为0.9。
refpwr	输出的参考功率, 默认为1mW, 必须大于 $1e-10$ 。
spwf	频率F2的幅度, $\geq 1e-3$, 默认为1。

例子:

```
.DISTO RL 2 0.95 1.0E-3 0.75
```

2) 噪声分析 .noise 声明

需要与.ac 声明配合使用

语法:

```
.NOISE ovv srcnam inter
```

说明:

ovv	指定噪声相加的输出节点。
srcnam	指定计算等效输入噪声的参考源。
inter	指定输出摘要的间隔, 若忽略或设为零则不输出任何噪声摘要信息。

例子:

```
.NOISE V(5) VIN 10
```

3) 交流网络分析 .net 声明

需要与.ac 声明配合使用

计算网络的阻抗矩阵, 导纳矩阵, 混合矩阵, 散射矩阵。

语法:

单端口网络:

```
.NET input <RIN=val>
```

or

```
.NET input <val >
```

双端口网络:

```
.NET output input <ROUT=val> <RIN=val>
```

说明:

input 交流输入源。

output 输出端口，可以是V(n1,n2)或I(source)或I(element)。

RIN 输入或源电阻，默认为 1Ω 。

ROUT 输出或负载电阻，默认为 1Ω 。

例子:

单端口网络:

```
.NET VINAC RIN=50
```

```
.NET IIN RIN=50
```

双端口网络:

```
.NET V(10,30) VINAC ROUT=75 RIN=50
```

```
.NET I(RX) VINAC ROUT=75 RIN=50
```

第七章 统计分析及优化

§7.1 用户定义的分析

7.1.1 .measure 声明

.measure 声明用来定义用户指定的分析，在电路的优化、模型参数拟合等方面有特别的应用。.measure 处理的是仿真输出的数据，因此减小输出数据的精度可能会引起.measure 处理时的误差。

基本的测试功能有：

- 上升、下降和延迟。
- 检索特定条件。
- 方程计算。
- 平均值、均方根值、最大最小值和峰峰值。
- 积分计算。
- 微分计算。
- 相对误差。

测量参数类型 (measure parameter types)

.measure 不能调用子电路中的参数。.measure 中的参数不能和标准参数重名。如果.measure 中定义的参数与.param 中的重名会报错。另外不同类型的参数重名不会引起错误。

7.1.2 上升、下降和延迟(rise fall and delay)

上升、下降和延迟测试模式可以计算起始值和目标值之间的时间、电压、频率等。瞬态分析中的上升下降时间，转换速率，交流分析中的带宽等等。

语法：

```
.MEASURE <DC|AC|TRAN> result TRIG ... TARG ...
+ <GOAL=val> <MINVAL=val> <WEIGHT=val>
```

说明：

result	输出结果名。起始值和目标值之间测得的变量值。瞬态分析中是时间，交流分析中是频率，直流扫描分析中是扫描变量。
TRIG...TARG...	分别指定起始值与目标值。
<DC AC TRAN>	指定仿真类型，如果省略，则采用最后一次仿真。
GOAL	指定优化的目标值,误差用下式计算 $ERRfun = (GOAL - result)/GOAL$ 。
MINVAL	如果GOAL值小于MINVAL，则GOAL值会被MINVAL值取代，默认为 $1.0e-12$ 。
WEIGHT	加权值，在优化中会用WEIGHT值乘以计算所得的误差。默认为1。

TRIG (Trigger) 语法：

```
TRIG trig_var VAL=trig_val <TD=time_delay> <CROSS=c> <RISE=r>
+ <FALL=f>
```

or

```
TRIG AT=val
```

TARG (Target) 语法:

```
TARG targ_var VAL=targ_val <TD=time_delay> <CROSS=c | LAST>
+ <RISE=r | LAST> <FALL=f | LAST>
```

说明:

TRIG	设定测量起始值的关键字。
VAL=trig_val	设定测量起始值。
trig_var	指定需测量的参数。用它设定测量起始值。如果在起始值之前碰到目标值, 则.measure 会报告一个负值。
TARG	设定测量目标值的关键字。
VAL=targ_val	设定测量目标值。
targ_var	指定需测量的参数。用它设定测量目标值。
TD=time_delay	设定测量开始前的延迟时间, 默认为 0。
CROSS=c	RISE 指上升, FALL 指下降, CROSS 指上升或下降。指定需测量的第几次上升或下降。
RISE=r	
FALL=f	
LAST	在最后一次 CROSS,FALL 或 LAST 事件发生的时候进行测量。
AT=val	设定测量起始值。

例子:

```
.MEASURE TRAN tdlay TRIG V(1) VAL=2.5 TD=10n RISE=2
+ TARG V(2) VAL=2.5 FALL=2
```

上例用节点 1 和 2 的电压对瞬态分析的结果进行测量。测量起始值是这样设置的, TD=10n 指延迟 10ns 开始计数, 当到 V(1)到第二个上升, 电压值达到 2.5V 的时候开始测量。测量目标值是这样设置的, 当 V(2)到达第二个下降沿, 电压值达到 2.5V 的时候测量结束。输出结果是 tdlay=<value>。

```
.MEASURE TRAN riset TRIG I(Q1) VAL=0.5m RISE=3
+ TARG I(Q1) VAL=4.5m RISE=3
.MEASURE pwidth TRIG AT=10n TARG V(IN) VAL=2.5 CROSS=3
最后一个例子使用 TRIG 的精简格式, AT=10n 指测量从 10ns 开始。
```

7.1.3 FIND 和 WHEN 函数

FIND 和 WHEN 函数允许当某些事件发生的时候, 测量任何独立变量(时间、频率、参数), 非独立变量(电压、电流等)或其微分值。对于测量单位增益带宽、相位等是有用的。

语法:

```
.MEASURE <DC|TRAN|AC> result WHEN out_var = val <TD = val>
+ < RISE=r | LAST > < FALL=f | LAST > < CROSS=c | LAST >
+ <GOAL=val> <MINVAL=val> <WEIGHT=val>
```

or

```
.MEASURE <DC|TRAN|AC> result WHEN out_var1=out_var2 < TD=val >
+ < RISE=r | LAST > < FALL=f | LAST > < CROSS=c| LAST >
+ <GOAL=val> <MINVAL=val> <WEIGHT=val>
```

or

```
.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2=val <
```

```

TD=val >
+ < RISE=r | LAST > < FALL=f | LAST >
+ < CROSS=c | LAST > < GOAL=val> < MINVAL=val> < WEIGHT=val>
or
.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2 = out_var3
+ < TD=val > < RISE=r | LAST > < FALL=f | LAST >
+ < CROSS=c | LAST > < GOAL=val> < MINVAL=val> < WEIGHT=val>
or
.MEASURE <DC|TRAN|AC> result FIND out_var1 AT=val < GOAL=val>
+ < MINVAL=val> < WEIGHT=val>

```

说明:

CROSS=c	RISE 指上升, FALL 指下降, CROSS 指上升或下降。指定需测量的第几次上升或下降。
RISE=r	
FALL=f	
<DC AC TRAN>	指定仿真类型, 如果省略, 则采用最后一次仿真。
FIND	选择 FIND 函数。
GOAL	指定优化的目标值, 误差用下式计算 $ERRfun = (GOAL - result)/GOAL$ 。
LAST	在最后一次 CROSS, FALL 或 LAST 事件发生的时候进行测量。
MINVAL	如果 GOAL 值小于 MINVAL, 则 GOAL 值会被 MINVAL 值取代, 默认为 $1.0e-12$ 。
out_var(1,2,3)	设定测量条件的参数。
result	输出结果名。
TD	设定测量开始前的延迟时间。
WEIGHT	加权值, 在优化中会用 WEIGHT 值乘以计算所得的误差。默认为 1。
WHEN	选择 WHEN 函数。

7.1.4 方程计算

用这个声明来计算方程, 方程的变量是 **measure** 声明所得结果, 不能是节点电压或支路电流。

语法:

```

.MEASURE <DC|TRAN|AC> result PARAM='equation'
+ < GOAL=val> < MINVAL=val>

```

7.1.5 平均值、均方根值、最大最小值和峰峰值测量

峰峰值指感兴趣范围内最大值最小值的差。

语法:

```

.MEASURE <DC|AC|TRAN> result func out_var <FROM=val> <TO=val>
+ < GOAL=val> < MINVAL=val> < WEIGHT=val>

```

说明:

<DC AC TRAN>	指定仿真类型, 如果省略, 则采用最后一次仿真。
FROM	设定“func”计算的起点。在瞬态分析中是时间。
TO	设定“func”计算的终点。

GOAL	指定优化的目标值, 误差用下式计算 $ERRfun = (GOAL - result)/GOAL$ 。
MINVAL	如果GOAL值小于MINVAL, 则GOAL值会被MINVAL值取代, 默认为1.0e-12。
func	从如下的声明中指定一种: AVG(average) 计算out_var的平均值。 MAX(maximum) 报告out_var的最大值。 MIN (minimum) 报告out_var的最小值。 PP (peak-to-peak) 计算out_var在指定的间隔里最大值和最小值的差。 RMS (root mean squared) 计算out_var在指定的间隔里的均方根值。
result	输出结果名。
out_var	需测量的参数。
WEIGHT	加权值, 在优化中会用WEIGHT值乘以计算所得的误差。默认为1。

例子:

```
.MEAS TRAN avgval AVG V(10) FROM=10ns TO=55ns
```

上例中计算了从10ns到55ns的V(10)的平均值, 输出到结果avgval。

```
.MEAS TRAN MAXVAL MAX V(1,2) FROM=15ns TO=100ns
```

上例中计算从15ns到100ns的V(1,2)的最大值, 输出到结果MAXVAL。

```
.MEAS TRAN MINVAL MIN V(1,2) FROM=15ns TO=100ns
```

```
.MEAS TRAN P2PVAL PP I(M1) FROM=10ns TO=100ns
```

7.1.6 积分函数

语法:

```
.MEASURE <DC|AC|TRAN> result INTEGRAL out_var <FROM=val>  
+ <TO=val> <GOAL=val> <MINVAL=val> <WEIGHT=val>
```

语法与平均值、均方根值、最大最小值和峰峰值测量的语法完全相同。func 定义为 integral 或 integ。

例子:

```
.MEAS TRAN charge INTEG I(cload) FROM=10ns TO=100ns
```

上例计算 I(cload)从 10ns 到 100ns 之间的积分值, 输出到结果 charge。

7.1.7 微分函数

根据分析类型求某时刻、某频率或某扫描变量指定值上某参数的微分值。也可以求指定事件发生时, 参数的微分值。

语法:

```
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var AT=val <GOAL=val>  
+ <MINVAL=val> <WEIGHT=val>
```

or

```
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var WHEN var2=val
+ <RISE=r | LAST> <FALL=f | LAST> <CROSS=c | LAST>
+ <TD=tdval> <GOAL=goalval> <MINVAL=minval> <WEIGHT=weightval>
or
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var WHEN var2=var3
+ <RISE=r | LAST> <FALL=f | LAST> <CROSS=c | LAST>
+ <TD=tdval> <GOAL=goalval> <MINVAL=minval> <WEIGHT=weightval>
```

说明:

AT=val	在此值处求微分。
CROSS=c	RISE 指上升, FALL 指下降, CROSS 指上升或下降。指定需测
RISE=r	的第几次上升或下降。
FALL=f	
<DC AC TRAN>	指定仿真类型, 如果省略, 则采用最后一次仿真。
DERIVATIVE	测量类型为求微分, 也可以写为DERIV。
GOAL	指定优化的目标值, 误差用下式计算 $ERRfun = (GOAL - result)/GOAL$ 。
LAST	在最后一次 CROSS, FALL 或 LAST 事件发生的时候进行测量。
MINVAL	如果GOAL值小于MINVAL, 则GOAL值会被MINVAL值取代, 默认为 $1.0e-12$ 。
result	输出结果名。
TD	设定测量前的延迟时间。
var(2,3)	设定测量条件的参数。
WEIGHT	加权值, 在优化中会用WEIGHT值乘以计算所得的误差。默认为1。
WHEN	设定when函数。

例子:

```
.MEAS TRAN slewrate DERIV V(out) AT=25ns
上例计算V(out)在25ns时的微分值。
.MEAS TRAN slew DERIV v(1) WHEN v(1)='0.90*vdd'
计算当V1='0.90*vdd'时V(1)的微分值。
.MEAS AC delay DERIV 'VP(output)/360.0' AT=10khz
计算当频率等于10KHz时'VP(output)/360.0'的微分值。
```

7.1.8 误差函数

相对误差函数报告两个输出变量的差别。这种格式经常用于对测量数据的优化及曲线拟合。设定.param中需测量的变量, 用ERR, ERR1, ERR2, 或ERR3 函数计算两变量的相对误差。可以设定一组变量, 并改变他们以使测量值跟目标值相符合。

语法:

```
.MEASURE <DC|AC|TRAN> result ERRfun meas_var calc_var <MINVAL=val>
+ <IGNORE | YMIN=val> <YMAX=val> <WEIGHT=val> <FROM=val> <TO=val>
<DC|AC|TRAN> 指定仿真类型, 如果省略, 则采用最后一次仿真。
result        输出结果名。
ERRfun        指定误差函数的类型, 可以为: ERR, ERR1, ERR2 或 ERR3。
meas_var      数据声明中的任何参数。M 代表误差函数中的 meas_var。
```

calc_var	输出参数。C 代表误差函数中的 calc_var。
IGNOR YMIN	如果 meas_var 的绝对值小于 IGNOR 的值，在误差函数的计算中这个点会被忽略。默认为 1.0e-15。
FROM	设定误差计算的起点。默认为扫描参数的第一个值。
WEIGHT	加权值，在优化中会用 WEIGHT 值乘以计算所得的误差。默认为 1。
YMAX	如果 meas_var 的绝对值大于 YMAX 的值，在误差函数的计算中这个点会被忽略。默认为 1.0e15。
TO	设定误差计算的终点。默认为扫描参数的最后一个值。
MINVAL	如果 meas_var 的绝对值小于 MINVAL，则 meas_var 值会被误差函数中分母上的 MINVAL 值取代，以避免小值主导误差函数，默认为 1.0e-12。

误差函数举例 ERR(其余省略):

$$ERR = \left[\frac{1}{NPTS} \cdot \sum_{i=1}^{NPTS} \left(\frac{M_i - C_i}{\max(MINVAL, M_i)} \right)^2 \right]^{1/2}$$

§7.2 温度分析

Hspice 中有三种温度:

模型温度	用 TREF(或 TEMP 或 TNOM)定义, 指模型参数测量和提取的温度。
电路温度	用 .temp 声明指定, 是所有器件仿真的温度。默认值是 TNOM。
器件温度	可用 DTEMP 参数定义器件温度与电路温度的差别, 仿真时器件的实际温度为: 电路温度 + DTEMP。

例子:

```
.TEMP 100
D1 N1 N2 DMOD DTEMP=30
D2 NA NC DMOD
R1 NP NN 100 TC1=1 DTEMP=-30
.MODEL DMOD D IS=1E-15 VJ=0.6 CJA=1.2E-13 CJP=1.3E-14
+ TREF=60.0
```

上例中电路温度是 100℃, D1 仿真温度是 100℃+30℃=130℃, 模型参考温度是 60℃, 故 D1 对模型的温度校正是 130℃-60℃=70℃。D2 的仿真温度是 100℃, 对模型的温度校正是 100℃-60℃=40℃。R1 的仿真温度是 100℃-30℃=70℃。

§7.3 最坏情况分析

7.3.1 标准统计名词定义

$$\text{平均值(mean)} = \frac{x_1 + x_2 + \dots + x_N}{N}$$

$$\text{方差(variance)} = \frac{(x_1 - \text{mean})^2 + \dots + (x_N - \text{mean})^2}{N - 1}$$

$$\text{标准偏差(sigma)} = \sqrt{\text{variance}}$$

$$\text{绝对偏差(average deviation)} = \frac{|x_1 - \text{mean}| + \dots + |x_N - \text{mean}|}{N-1}$$

7.3.2 最坏情况分析介绍

常用于分析 MOS 管及三极管电路。仿真时所有参数取最坏情况值,即两倍的 **sigma**,或三倍的 **sigma**。实际电路工艺中所有变量同时取最差值是不可能的,因此这种分析有点太过悲观,但却比较快捷。

7.3.3 模型歪斜参数及工艺角文件

器件物理模型中会存在歪斜参数(skew parameters),之所以称之为歪斜参数,是因为他们会从平均值处歪斜(skew)掉。如图 7.1 是 MOS 管一个歪斜参数的历史分布。

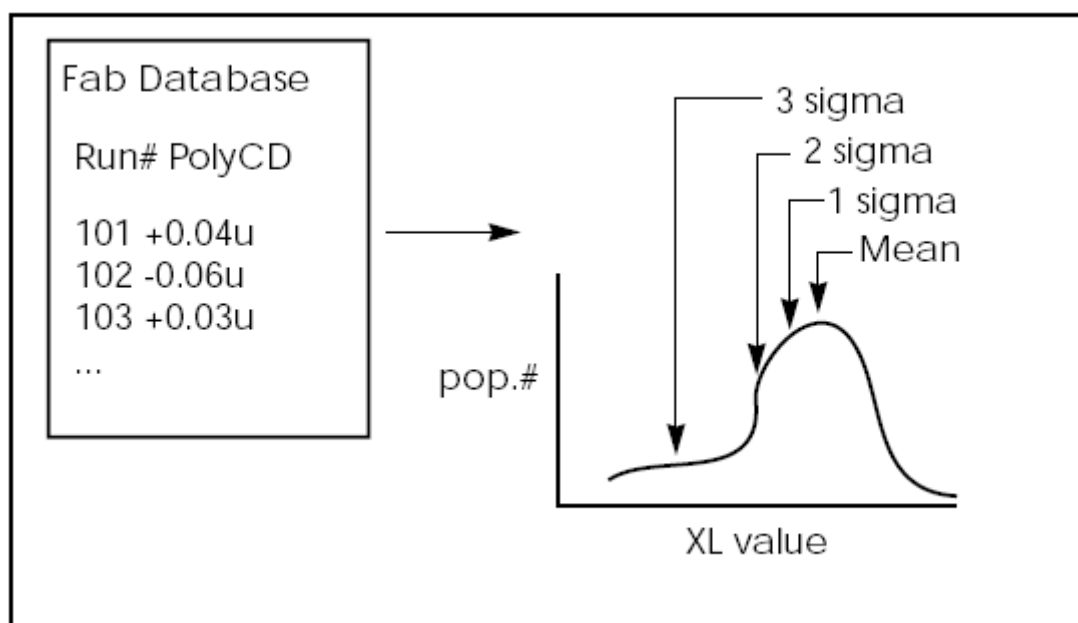


图 7.1 MOS 管某歪斜参数的历史分布

可以将歪斜参数写入一个用于最差情况分析的工艺角文件中,歪斜参数会替换掉模型参数,这个文件通常为一库文件,下面是一个工艺角文件的例子。

```
.LIB TT
$TYPICAL P-CHANNEL AND N-CHANNEL CMOS LIBRARY DATE:3/4/91
$ PROCESS: 1.0U CMOS, FAB22, STATISTICS COLLECTED 3/90-2/91
$ following distributions are 3 sigma ABSOLUTE GAUSSIAN
.PARAM
$ polysilicon Critical Dimensions
+ polycd=agauss(0,0.06u,1) xl='polycd-sigma*0.06u'
$ Active layer Critical Dimensions
+ nactcd=agauss(0,0.3u,1) xwn='nactcd+sigma*0.3u'
+ pactcd=agauss(0,0.3u,1) xwp='pactcd+sigma*0.3u'
$ Gate Oxide Critical Dimensions (200 angstrom +/- 10a at 1
$ sigma)
+ toxcd=agauss(200,10,1) tox='toxcd-sigma*10'
```

```

$ Threshold voltage variation
+ vtoncd=agauss(0,0.05v,1) delvton='vtoncd-sigma*0.05'
+ vtopcd=agauss(0,0.05v,1) delvtop='vtopcd+sigma*0.05'
.INC '/usr/meta/lib/cmos1_mod.dat'$ model include file
.ENDL TT
.LIB FF
$HIGH GAIN P-CH AND N-CH CMOS LIBRARY 3SIGMA VALUES
.PARAM TOX=230 XL=-0.18u DELVTON=-.15V DELVTOP= 0.15V
.INC '/usr/meta/lib/cmos1_mod.dat'$ model include file
.ENDL FF

```

模型应包含于文件/usr/meta/lib/cmos1_mod.dat中

```

.MODEL NCH NMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTON .....
.MODEL PCH PMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTOP .....

```

§7.4 蒙特卡罗分析

7.4.1 蒙特卡罗分析概要

参数在使用时会根据其分布计算一随机值。如果没有为参数指定任何分布，那么会采用名义值(nominal value)，分布函数只应用于蒙特卡罗分析，其他分析只采用名义值。可为模型参数指定分布函数，于是使用同一模型的器件会有同样的分布函数。

1) 几种分布函数

高斯分布(Gaussian Distribution)

均匀分布(Uniform Distribution)

随机限制分布(Random Limit Distribution)

2) 蒙特卡罗分析的设置

- .param 声明，将模型或器件参数设为某种分布
- 在.dc, .ac, .tran 分析中设置 monte 关键字
- .measure 声明，计算平均值，方差，标准偏差等等。

语法：

工作点：

```
.DC MONTE=val
```

直流扫描：

```
.DC vin 1 5 .25 SWEEP MONTE=val
```

交流扫描：

```
.AC dec 10 100 10meg SWEEP MONTE=val
```

瞬态扫描：

```
.TRAN 1n 10n SWEEP MONTE=val
```

val 表示蒙特卡罗分析的重复次数。常用的值是 30，其统计学含义是：如果电路在 30 次重复分析中都能正常工作，意味着电路有 99% 的可能性在 80% 的取值下正常工作。

3) 蒙特卡罗分析的输出

.measure 声明是最方便的概括输出结果的方式。

.print 声明能够产生列表结果并输出蒙特卡罗分析中用过的参数值。

.graph 声明为每次重复产生高分辨率的图形。

7.4.2 定义分布函数 .param 声明

用 .param 声明定义参数并确定其分布函数，供蒙特卡罗分析使用。

语法：

```
.PARAM xx=UNIF(nominal_val, rel_variation <, multiplier>)
or
.PARAM xx=AUNIF(nominal_val, abs_variation <, multiplier>)
or
.PARAM xx=GAUSS(nominal_val, rel_variation, sigma <,
+ multiplier>)
or
.PARAM xx=AGAUSS(nominal_val, abs_variation, sigma <,
+ multiplier>)
or
.PARAM xx=LIMIT(nominal_val, abs_variation)
```

说明：

xx	参数名，其值可用分布函数计算。
UNIF	使用相对偏差的均匀分布函数。
AUNIF	使用绝对偏差的均匀分布函数。
GAUSS	使用相对偏差的高斯分布函数。
AGAUSS	使用绝对偏差的高斯分布函数。
LIMIT	使用绝对偏差的随机限制分布函数。
nominal_val	用于蒙特卡罗分析的名义值或其他分析的默认值。
abs_variation	AUNIF 和 AGAUSS 用 +/- abs_variation 改变 nominal_val 的值，来确定参数的变化范围。
rel_variation	UNIF 和 GAUSS 用 +/- (nominal_val * rel_variation) 改变 nominal_val 的值，来确定参数的变化范围。
sigma	abs_variation 或 nominal_val*rel_variation 与标准偏差的比值。用来确定分布函数的具体形状。如 sigma=3，那么标准偏差就等于 abs_variation 除以 3。
multiplier	默认值为 1。设定重复计算的次数并保存最大偏差时的结果。结果是双峰分布的，即趋向于取两端的值。

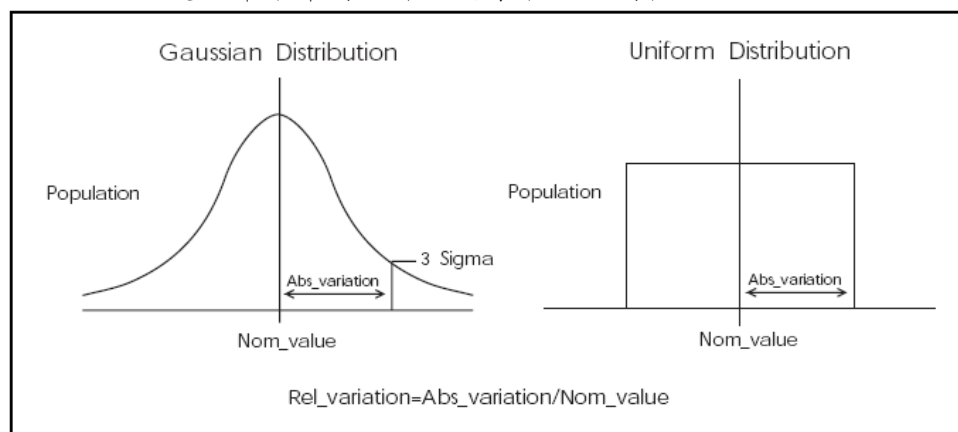


图 7.2 高斯分布和均匀分布

7.4.3 蒙特卡罗分析的例子

1) 例一: Gaussian, Uniform, and Limit Functions

Test of monte carlo gaussian, uniform, and limit functions

```
.options post
.dc monte=60
* setup plots
.model histo plot ymin=80 ymax=120 freq=1    $this model is used for .graph
.graph model=HISTO aunif_1=v(au1)
.graph model=HISTO aunif_10=v(au10)
.graph model=HISTO agauss_1=v(ag1)
.graph model=HISTO agauss_10=v(ag10)
.graph model=HISTO limit=v(L1)
* uniform distribution relative variation +/- .2
.param ru_1=unif(100,.2)
lu1 u1 0 -1
ru1 u1 0 ru_1
* absolute uniform distribution absolute variation +/- 20
* single throw and 10 throw maximum
.param rau_1=aunif(100,20)
.param rau_10=aunif(100,20,10)
lau1 au1 0 -1
rau1 au1 0 rau_1
lau10 au10 0 -1
rau10 au10 0 rau_10
* gaussian distribution relative variation +/- .2 at 3 sigma
.param rg_1=gauss(100,.2,3)
lg1 g1 0 -1
rg1 g1 0 rg_1
* absolute gaussian distribution absolute variation +/- .2 at
3 sigma
* single throw and 10 throw maximum
.param rag_1=agauss(100,20,3)
.param rag_10=agauss(100,20,3,10)
lag1 ag1 0 -1
rag1 ag1 0 rag_1
lag10 ag10 0 -1
rag10 ag10 0 rag_10
* random limit distribution absolute variation +/- 20
.param RL=limit(100,20)
IL1 L1 0 -1
rL1 L1 0 RL
.end
```

2) 例二: Major and Minor Distribution

在工艺中,有些参数变化较大,有较大的离散分布,如晶片到晶片之间,有些参数变化较小,有较小的离散分布,如晶体管到晶体管之间,可以用蒙特卡罗分析处理这些现象。

下面的例子 LEFF 只有 5% 的变化范围, PHOTO 却有 30% 的变化范围。

```
File: MONDC_A.SP
.DC VDD 4.5 5.5 .1 SWEEP MONTE=30
.PARAM LENGTH=1U LPHOTO=.1U
.PARAM LEFF=GAUSS (LENGTH, .05, 3)
+ XPHOTO=GAUSS (LPHOTO, .3, 3)
.PARAM PHOTO=XPHOTO
M1 1 2 GND GND NCH W=10U L=LEFF
M2 1 2 VDD VDD PCH W=20U L=LEFF
M3 2 3 GND GND NCH W=10U L=LEFF
M4 2 3 VDD VDD PCH W=20U L=LEFF
.MODEL NCH NMOS LEVEL=2 UO=500 TOX=100 GAMMA=.7 VTO=.8
+ XL=PHOTO
.MODEL PCH PMOS LEVEL=2 UO=250 TOX=100 GAMMA=.5 VTO=-.8
+ XL=PHOTO
.INC Model.dat
.END
```

7.4.4 最差情况和蒙特卡罗分析的例子

1) HSPICE 输入文件

分成如下几个部分:

分析设置部分

仿真用AUTOSTOP选项来加速,其含义是.MEASURE到达目标值后会停止计算。

```
$ inv.sp sweep mosfet -3 sigma to +3 sigma, then Monte Carlo
.option nopage nomod acct
+ autostop post=2
.tran 20p 1.0n sweep sigma -3 3 .5
.tran 20p 1.0n sweep monte=20
.option post co=132
.param vref=2.5
.meas m_delay trig v(2) val=vref fall=1
+ targ v(out) val=vref fall=1
.meas m_power rms power to=m_delay
.param sigma=0
```

电路网表部分

```
.global 1
vcc 1 0 5.0
vin in 0 pwl 0,0 0.2n,5
x1 in 2 inv
```

```

x2 2 3 inv
x3 3 out inv
x4 out 5 inv
.macro inv in out
mn out in 0 0 nch W=10u L=1u
mp out in 1 1 pch W=10u L=1u
.eom

```

模型中的歪斜参数重新覆盖部分

```

* overlay of gaussian and algebraic for best case worst case and
* monte carlo
* +/- 3 sigma is the maximum value for parameter sweep
.param
+ mult1=1
+ polycd=agauss(0,0.06u,1) xl='polycd-sigma*0.06u'
+ nactcd=agauss(0,0.3u,1) xwn='nactcd+sigma*0.3u'
+ pactcd=agauss(0,0.3u,1) xwp='pactcd+sigma*0.3u'
+ toxcd=agauss(200,10,1) tox='toxcd-sigma*10'
+ vtoncd=agauss(0,0.05v,1) delvton='vtoncd-sigma*0.05'
+ vtopcd=agauss(0,0.05v,1) delvtop='vtopcd+sigma*0.05'
+ rshncd=agauss(50,8,1) rshn='rshncd-sigma*8'
+ rshpcd=agauss(150,20,1) rshp='rshpcd-sigma*20'

```

MOS管模型

```

* level=28 example model for high accuracy model
.model nch nmos
+ level=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ xl=xl xw=xwn tox=tox delvto=delvton rsh=rshn
+ ld=0.06u wd=0.2u
+ acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rdc=0 rsc=0
+ js=3e-04 jsw=9e-10
+ cj=3e-04 mjd=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8 fc=.5
+ capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=1.4e-03
* dc model
+ x2e=0 x3e=0 x2u1=0 x2ms=0 x2u0=0 x2m=0
+ vfb0=-.5 phi0=0.65 k1=.9 k2=.1 eta0=0
+ muz=500 u00=.075
+ x3ms=15 u1=.02 x3u1=0
+ b1=.28 b2=.22 x33m=0.000000e+00
+ alpha=1.5 vcr=20
+ n0=1.6 wfac=15 wfacu=0.25

```

```

+ lvfb=0 lk1=.025 lk2=.05
+ lalpha=5
.model pch pmos
+ level=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ xl=xl xw=xwp tox=tox delvto=delvtop rsh=rshp
+ ld=0.08u wd=0.2u
+ acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rdc=0 rsc=0 rsh=rshp
+ js=3e-04 jsw=9e-10
+ cj=3e-04 mj=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8 fc=.5
+ capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=-1.7e-03
* dc model
+ x2e=0 x3e=0 x2u1=0 x2ms=0 x2u0=0 x2m=5
+ vfb0=-.1 phi0=0.65 k1=.35 k2=0 eta0=0
+ muz=200 u00=.175
+ x3ms=8 u1=0 x3u1=0.0
+ b1=.25 b2=.25 x33m=0.0
+ alpha=0 vcr=20
+ n0=1.3 wfac=12.5 wfacu=.2
+ lvfb=0 lk1=-.05
.end

```

2) 仿真

第一步只运行最差情况扫描，在网表中用*使蒙特卡罗分析失效，输入：

```
hspice *.sp > worst.lis
```

*.sp 指输入文件名。

第二步只运行蒙特卡罗扫描，修改网表后输入：

```
hspice *.sp -n 1 > monte.lis
```

-n 1 表示文件从 1 开始计数。

3) 仿真结果

瞬态扫描结果

图 7.3 是 sigma 从-3 到 3 瞬态分析 V(2)和 V(out)的输出结果。

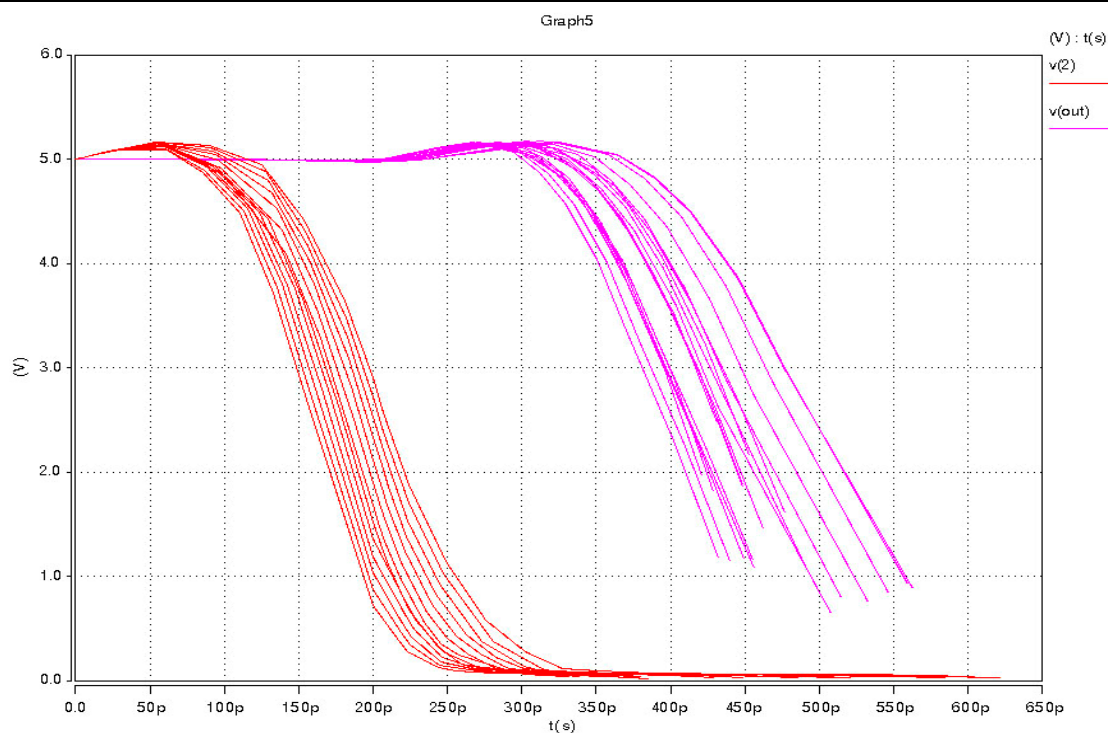


图 7.3 sigma 从-3 到 3 瞬态分析扫描结果

图 7.4 是利用 .measure 测量数据得到的结果。延迟和总消耗功率相对于 sigma 的变化。

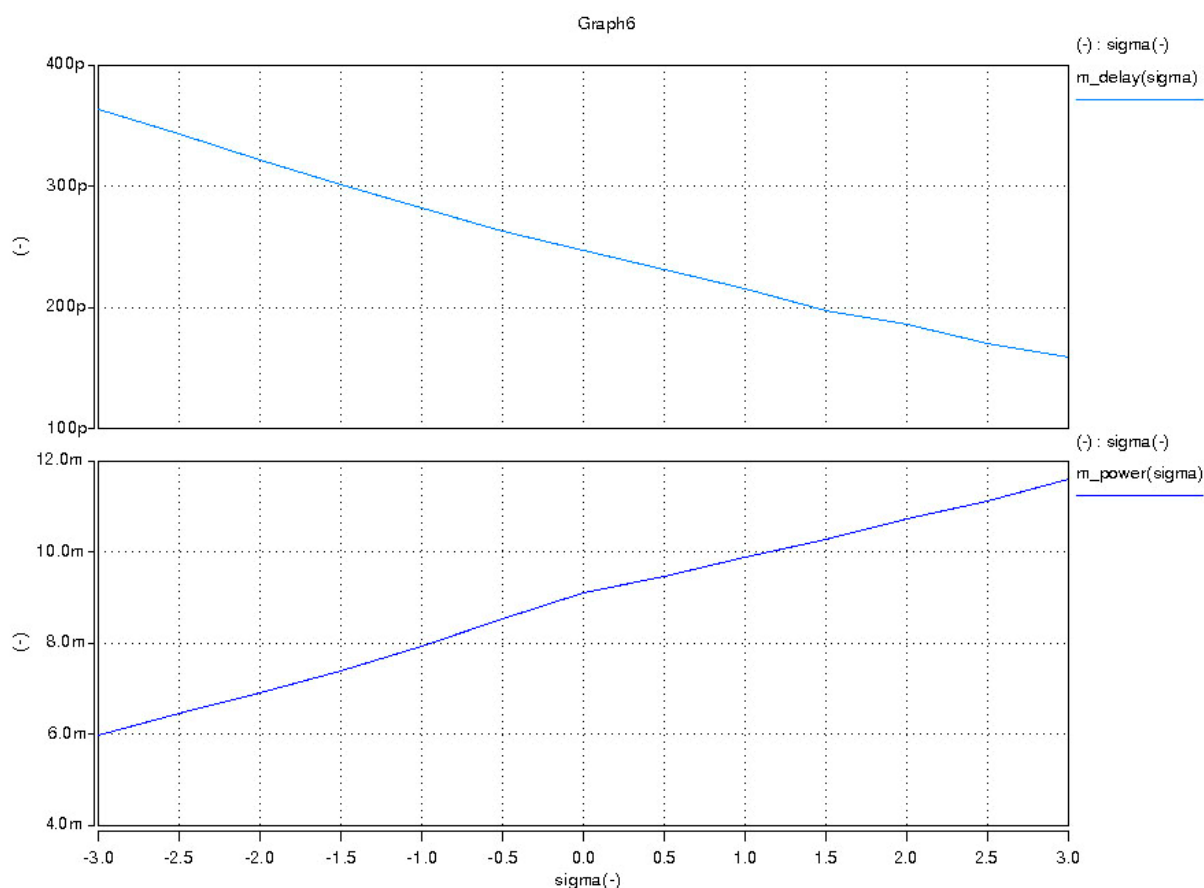


图 7.4 延迟和功率相对于 sigma 的变化

蒙特卡罗分析结果

图 7.5 是延迟随分析次数的分布。

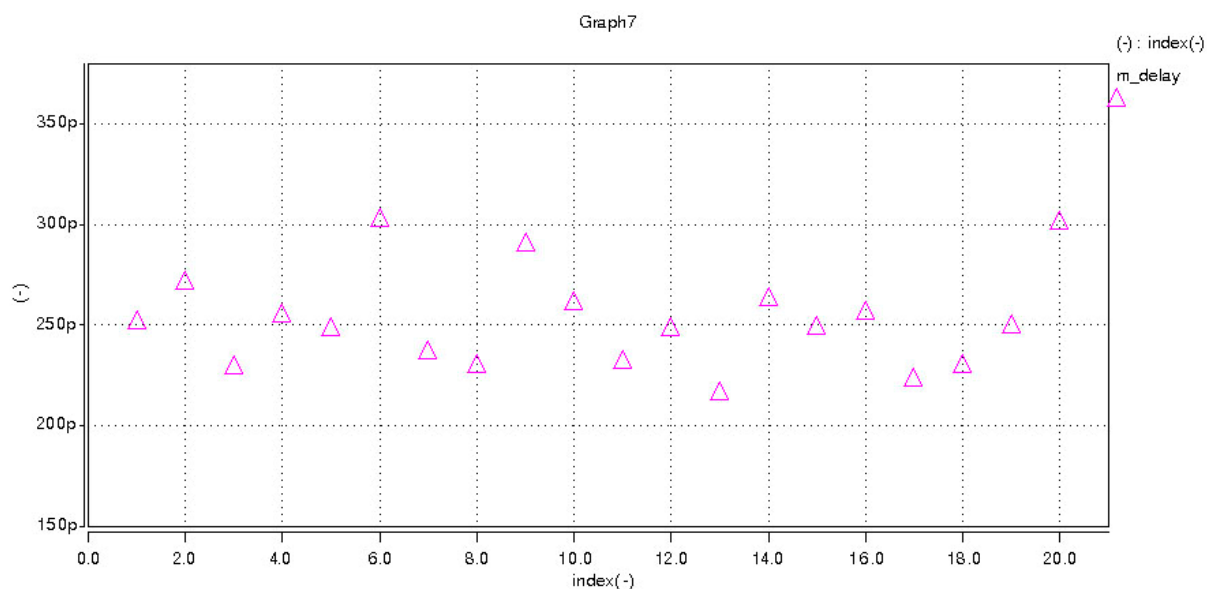


图 7.5 延迟随分析次数的分布

比较重要的信息是哪些参量对性能的影响较大，图 7.6 是延迟相对于沟道长度改变的变化，显示了延迟对于沟道长度的敏感度。

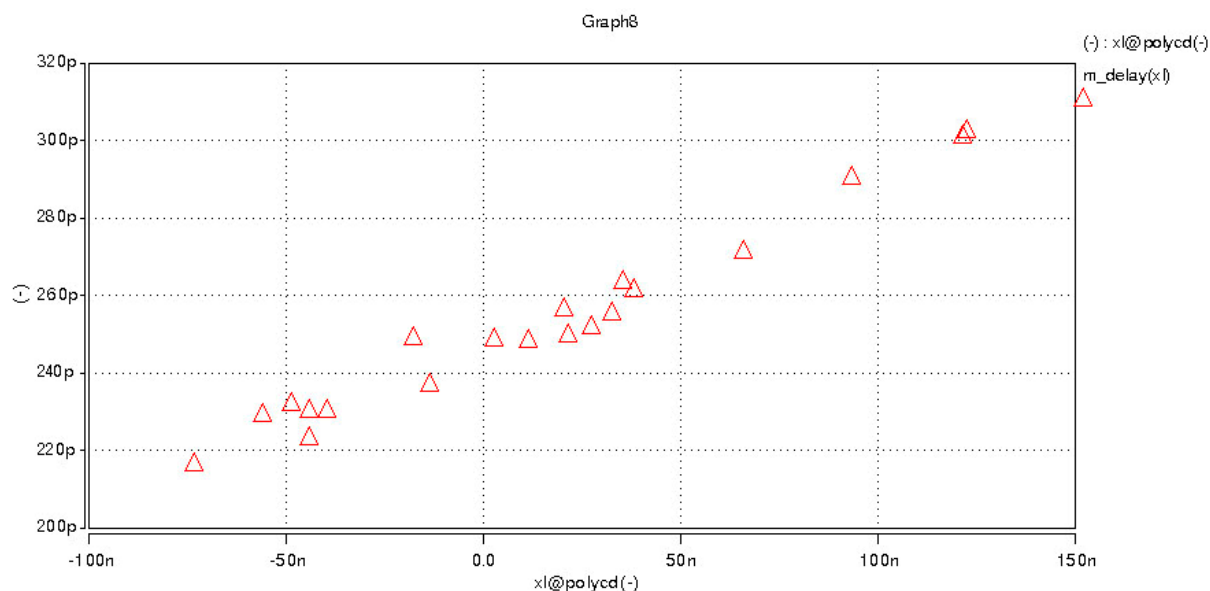


图 7.6 延迟对于沟道长度的敏感性分析

图 7.7 是蒙特卡罗分析与最差情况分析下延迟的对比，可见最差情况分析是有点过于悲观了。

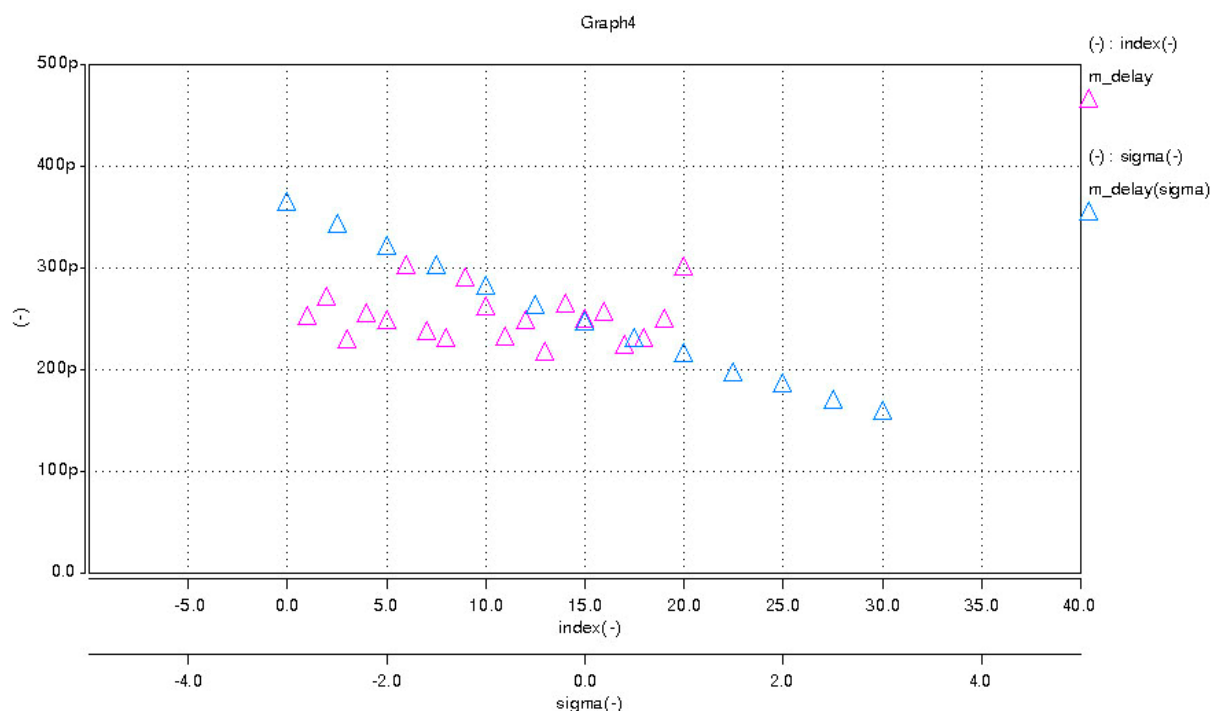


图 7.7 最差情况分析与蒙特卡罗分析对比

图 7.8 是延迟关于功耗的分布图。BIN 1 象限是功耗最少延迟最小的象限，BIN 4 是功耗最多延迟最大的象限，等等。通过蒙特卡罗分析提供的分布图我们可以预计将来电路的性能分布。

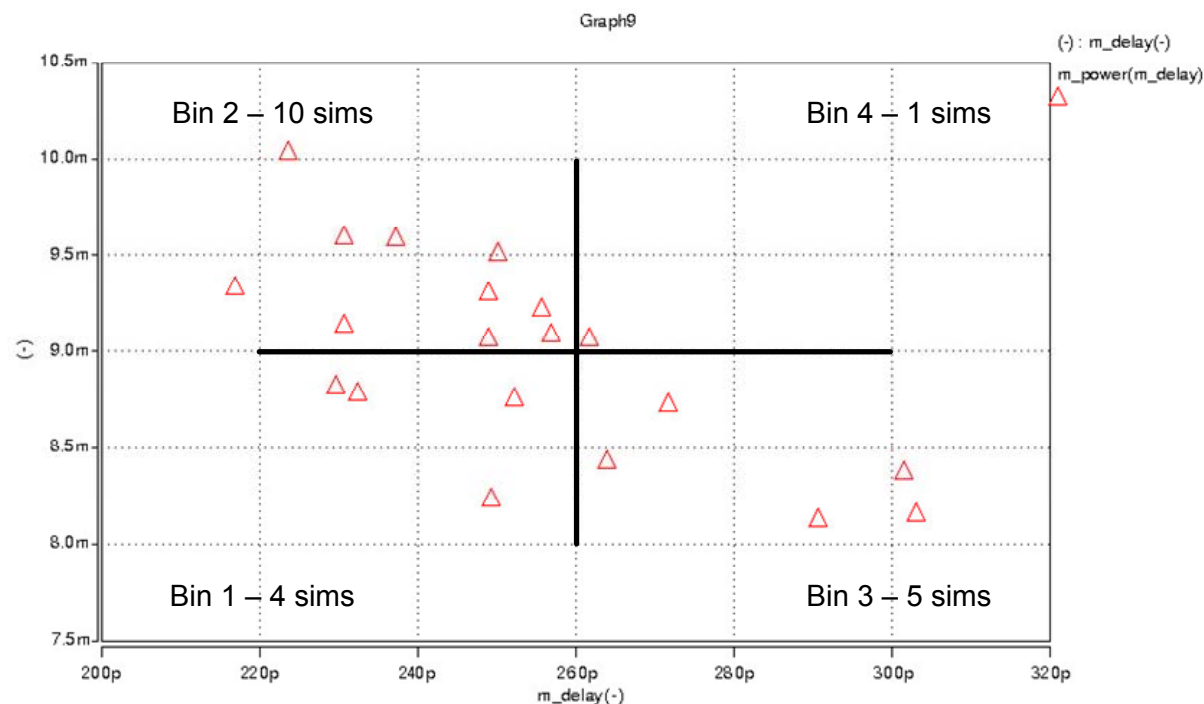


图 7.8 延迟、功耗性能分布图

§7.5 优化

7.5.1 优化概要

Hspice 可以根据设定的目标或测量数据优化模型参数值或器件参数值。Star-Hspice

在电路优化相关算法及用户接口方面拥有十年以上研究积累。优化函数集成到仿真核心，效率高。优化目标是 `.measure` 声明的一部分，可优化的参数在 `.param` 中定义，优化设置用 `.model` 定义。

1) 目标优化

目标优化需要在 `.measure` 中定义 `goal` 关键字。

2) 曲线拟合

可以用优化来拟合用户定义的直流、交流、瞬态分析数据，如根据 `.data` 定义的数据进行优化。优化定义需用 `.measure` 中的误差函数，通常用 `ERR1`，优化中会反复选取参数值进行仿真，直到与待拟合的曲线最接近，或满足误差容限。

3) 优化设置

- 设置优化选项 `.MODEL modname OPT ...`
- 定义待优化的参数 `.PARAM parameter=OPTxxx (init, min, max)`
- 定义分析类型，如 `.DC`, `.AC`, 或 `.TRAN`，并使用 `MODEL=modname`, `OPTIMIZE=OPTxxx`, 和 `RESULTS=measurename` 关键字。
- 用 `.MEASURE measurename ... <GOAL = | < | > val>` 声明设定优化函数、优化目标等。

一旦 `.DC`, `.AC`, `.TRAN` 分析中带有 `OPTIMIZE` 关键字，就只能用于优化。想获得这些分析的输出必须重新进行声明。合理的设置顺序是这样的：

1. 带有 `OPTIMIZE` 关键字的分析声明
2. `.measure` 声明设定优化目标及误差函数
3. 普通分析声明
4. 输出声明

7.5.2 优化相关声明

1) 优化控制

用 `.model` 定义收敛标准、迭代次数等，一般 10 到 30 次迭代就足够获得精确结果。
语法：

```
.MODEL mname OPT <parameter=val ...>
```

说明：

`mname` 模型名。用这个名称相关于这些设置。

`ITROPT` 设定最大的迭代次数。默认为 20。

(其余参数省略)

2) 直流、交流、瞬态分析声明

语法：

```
.DC <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1 ...  
+ ierrn MODEL=optmod
```

or

```
.AC <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1 ...  
+ ierrn MODEL=optmod
```

or

```
.TRAN <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1  
+ ... ierrn MODEL=optmod
```

说明：

DATA	设定优化中要使用的网表内数据。
OPTIMIZE	指示优化分析的关键字，也定义了优化分析的名称，所有与此名称关联的参数都会参与运算。
MODEL	优化控制选项的关联名称，于.model 中定义。
RESULTS	与测量相关的名称，在.measure 中定义。RESULTS 用来向.measure 传递分析数据。

3) 参数声明

语法:

```
.PARAM parameter=OPTxxx (initial_guess, low_limit, upper_limit)
or
.PARAM parameter=OPTxxx (initial_guess, low_limit, upper_limit,
+ delta)
```

说明:

OPTxxx 参数相关联的优化分析的名称。

parameter 参数名。

例子:

```
.PARAM vtx=OPT1(.7,.3,1.0) uox=OPT1(650,400,900)
```

上例中名为 OPT1 的优化分析将调用 vtx 和 uox, vtx 的猜测值为 0.7, 范围为 0.3 到 1.0 之间。uox 的猜测值为 650, 范围为 400 到 900 之间。

7.5.3 优化的例子

MOS Level 3模型直流优化

这个例子展示了用 I-V 数据来优化 Level 3 MOS 模型。数据含有栅漏曲线(ids 关于 vgs), 漏极曲线(ids 关于 vds)。会对 Level 3 参数 VTO, GAMMA, UO, VMAX, THETA 和 KAPPA 进行优化。优化后会将模型与原始数据对比。

Level 3 模型直流优化输入网表

```
$level 3 mosfet optimization
$tighten the simulator convergence properties
.OPTION nomod post=2 newtol relmos=1e-5 absmos=1e-8
.MODEL optmod OPT itropt=30
```

输入电路

```
vds 30 0 vds
vgs 20 0 vgs
vbs 40 0 vbs
m1 30 20 0 40 nch w=50u l=4u
$process skew parameters for this data
.PARAM xwn=-0.3u xln=-0.1u toxn=196.6 rshn=67
$the model and initial guess
.MODEL nch NMOS level=3
+ acm=2 ldif=0 hdif=4u tlev=1 n=2
+ capop=4 meto=0.08u xqc=0.4
$note capop=4 is ok for H8907 and later, otherwise use
$Capop=2
```

```

$fixed parameters
+ wd=0.15u ld=0.07u
+ js=1.5e-04 jsw=1.8e-09
+ cj=1.7e-04 cjsw=3.8e-10
+ nfs=2e11 xj=0.1u delta=0 eta=0
$process skew parameters
+ tox=toxn rsh=rshn
+ xw=xwn xl=xln

```

优化参数

```

+ vto=vto gamma=gamma
+ uo=uo vmax=vmax theta=theta kappa=kappa
.PARAM
+ vto = opt1(1,0.5,2)
+ gamma = opt1(0.8,0.1,2)
+ uo = opt1(480,400,1000)
+ vmax = opt1(2e5,5e4,5e7)
+ theta = opt1(0.05,1e-3,1)
+ kappa = opt1(2,1e-2,5)

```

优化扫描

```

.DC DATA=all optimize=opt1 results=comp1 model=optmod
.MEAS DC comp1 ERR1 par(ids) i(m1) minval=1e-04 ignor=1e-05

```

直流扫描

```

.DC DATA=gate
.DC DATA=drain

```

直流扫描数据

```

$data
.PARAM vds=0 vgs=0 vbs=0 ids=0
.DATA all vds vgs vbs ids
1.000000e-01 1.000000e+00 0.000000e+00 1.655500e-05
5.000000e+00 5.000000e+00 0.000000e+00 4.861000e-03
.ENDATA
.DATA gate vds vgs vbs ids
1.000000e-01 1.000000e+00 0.000000e+00 1.655500e-05
1.000000e-01 5.000000e+00 -2.000000e+00 3.149500e-04
.ENDDATA
.DATA drain vds vgs vbs ids
2.500000e-01 2.000000e+00 0.000000e+00 2.302000e-04
5.000000e+00 5.000000e+00 0.000000e+00 4.861000e-03
.ENDDATA
.END

```

上面的输入网表包含的内容有：

- 用.options 指定了更严格的约束条件。
- “.MODEL optmod OPT itropt=30”限制迭代次数为 30 次。

- 电路只有一个晶体管。
- 在.param 声明中将工艺可变参数 XL, XW, TOX, RSH 设置为常数。
- 模型中定义了可优化的参数。在“GAMMA= GAMMA”中，左边是模型参数名，右边是.param定义的参数名。
- 长.param声明中定义了可优化的参数，他们的猜测值，最大最小值。
- 第一个.dc 声明指定了网表内数据“.data all”模块，优化参数名opt1(本例中包含了所有待优化参数)，误差函数名comp1(与.measure中定义相符)，模型名optmod(设定了迭代限制)。
- .measure 设定了误差函数。par(ids)与i(m1)的差值会除以par(ids)和minval=10e-6中的较大值。minval的使用避免了小的电流主导误差函数。
- 剩下的.dc声明用于优化后的仿真和输出。
- “.PARAM VDS=0 VGS=0 VBS=0 IDS=0”只是声明这些量是参数。

输出结果:

optimization results

residual sum of squares	=	1.008464E-10
norm of the gradient	=	2.089366E-04
marquardt scaling parameter	=	2.225853E-04
no. of function evaluations	=	21
no. of iterations	=	7

**** optimized parameters opt1

*		%norm-sen	%change
.param vto	= 796.2617m	\$ 73.1393	-3.3141m
.param gamma	= 981.3552m	\$ 2.5745	17.8649m
.param uo	= 469.9599	\$ 20.0696	-2.8397m
.param vmax	= 135.5013k	\$ 2.1617	-18.4567m
.param theta	= 60.3725m	\$ 2.0542	19.5475m
.param kappa	= 10.0000m	\$ 718.4750u	0.

关于输出结果的说明:

residual sum of squares 和 norm of the gradient 都是总误差的量度，他们越小越好。
marquardt scaling parameter 代表参数迭代中的收敛性，越小越好。

备注:

表5 常用声明的通用写法

声明	等价写法
.options	.option
.include	.inc
.measure	.meas

参考文献:

Avant! Star-Hspice Manual, Release 1999.2, June 1999.