

湖北九同方微电子

Spice语法

李聪 (Claire)

congcong@hust.edu.cn

School of Optical and Electronic Information,
Huazhong University of Science and Technology, Wuhan

2014.07.08

主要内容

- Spice简介
- Spice网表的语法详解
 - 网表基本结构
 - 基本词法
 - 基本单位
 - 器件定义
 - 器件模型
 - 分析类型描述语句
 - 控制语句和Options语句
 - 输出格式描述语句
- 实例演示

Spice简介

Spice简介

SPICE: Simulation Program for Integrated Circuits Emphasis

- **Spice**是一种功能强大的通用模拟电路仿真器，已经具有几十年的历史了，该程序主要用于集成电路的电路分析程序中，Spice的网表格式变成了通常模拟电路和晶体管级电路描述的标准，其第一版本于1972年完成，是用Fortran语言写成的，1975年推出正式实用化版本，1988年被定为美国国家工业标准，主要用于IC，模拟电路，数模混合电路，电源电路等电子系统的设计和仿真。
- **Spice**是最为普遍的电路级模拟程序，各软件厂家提供了Vspice、Hspice(Synopsys)、Pspice(Cadence)等不同版spice软件，其仿真核心大同小异，都是采用了由UC Berkeley大学开发的spice模拟算法。

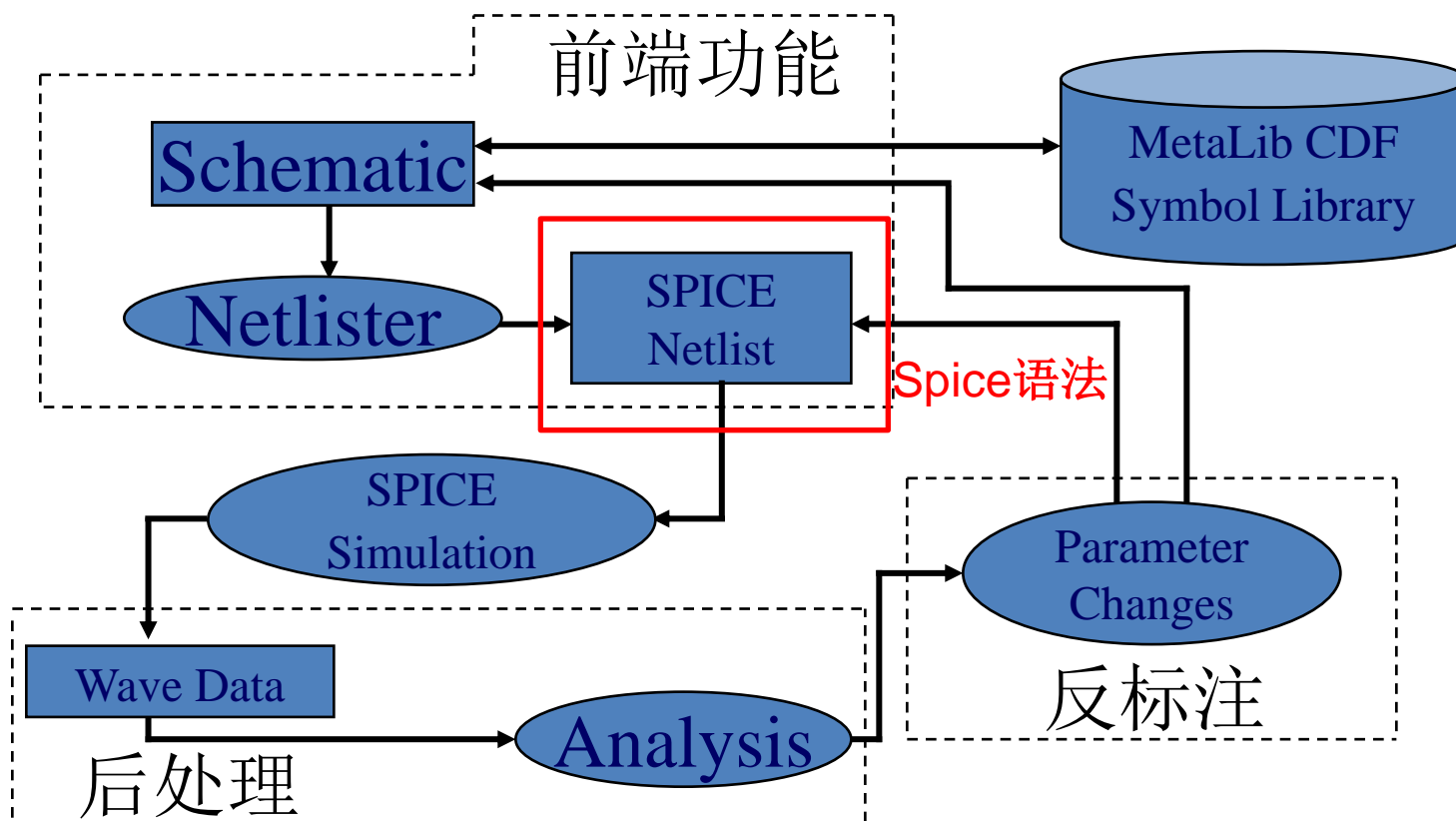
Spice功能

- 电路级和行为级仿真
- 直流特性分析、灵敏度分析
- 交流特性分析、瞬态分析
- 噪声分析、温度特性分析
- 傅立叶分析
- 电路优化（优化元件参数）
- **Monte Carlo**, 最坏情况, 参数扫描, 数据表扫描
- 功耗、各种电路参数（如H参数、T参数、s参数）等可扩展性能分析

Spice功能

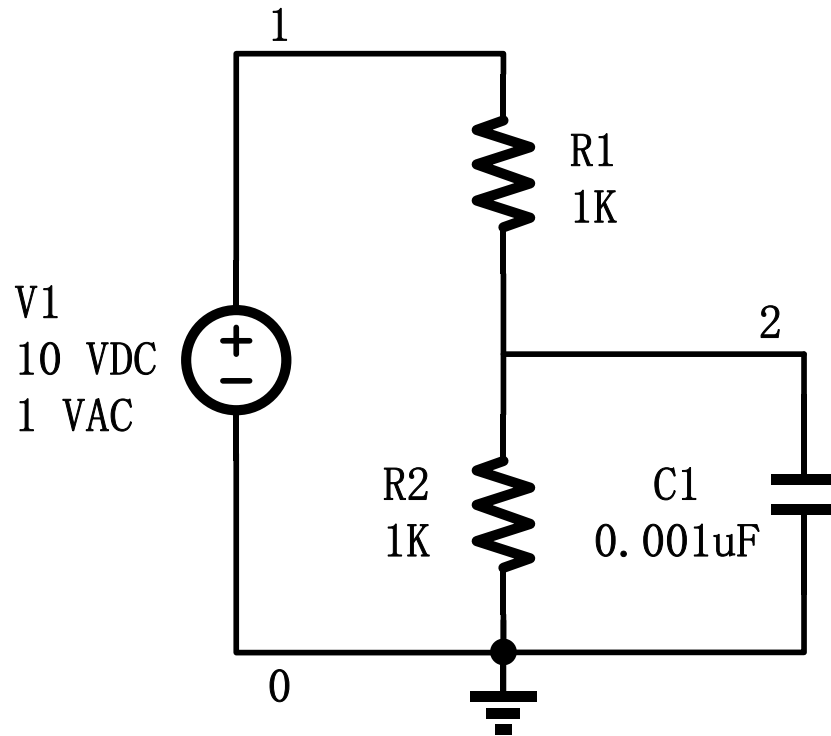
- 在用Spice进行电路仿真之前，应：
 - 了解元件的基本特性
 - 熟悉所设计的电路功能
 - 了解需要验证的电路指标和对应的模拟种类、电路状态
 - 了解电路的输入信号特性
 - 了解电路各项指标的相依性及优先度
 - 了解电路结构、元件参数与各项电路特性的相关性，以便于模拟结果的改进

Spice设计流程



Spice网表语法详解

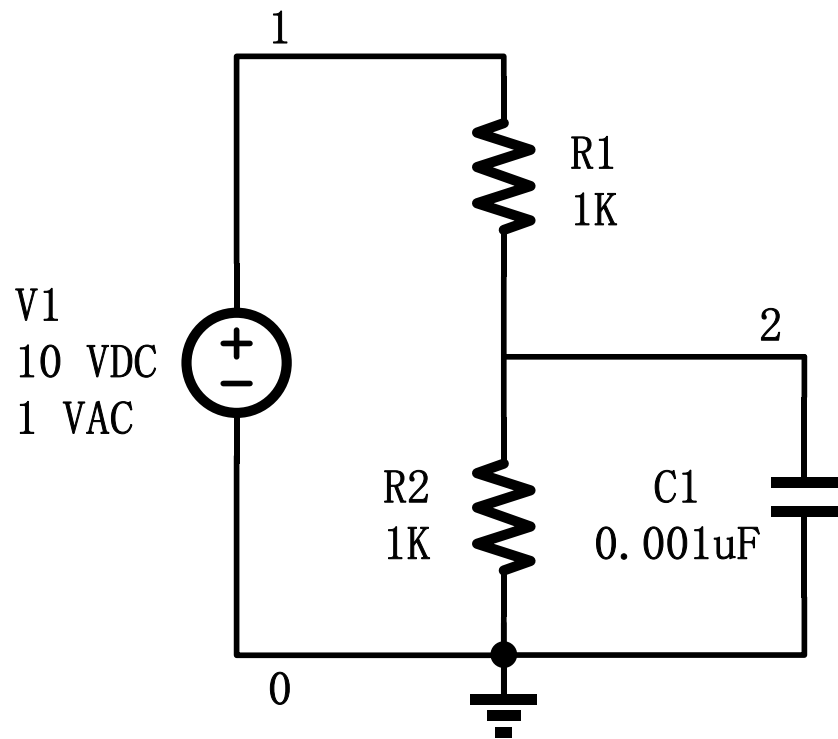
- RC网络AC分析



网表基本结构

例子：A Simple RC network circuit

```
*title 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
```



网表基本结构

例子: A Simple RC network circuit

***title A SIMPLE AC RUN**

→ 标题行, *注释

.OPTIONS LIST NODE POST

→ 设计模拟的可选条件

.OP

.AC DEC 10 1K 1MEG

→ 设置仿真分析类型

.PRINT AC V(1) V(2) I(R2) I(C1)

→ 设置输出结果显示

V1 1 0 10 AC 1

→ 设置输入激励

R1 1 2 1K

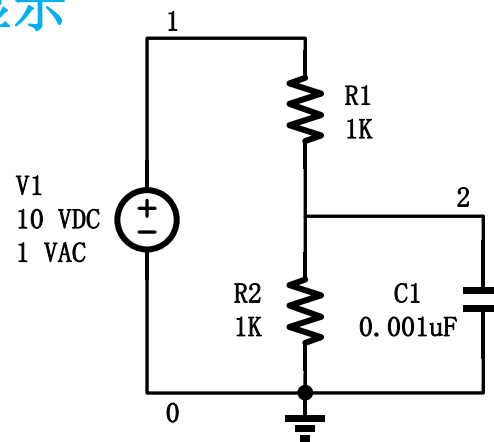
R2 2 0 1K

C1 2 0 .001U

→ 电路网表描述

.END

→ 结束语句



网表基本结构

默认为文件的第一行

*title	输入文件的标题
options	设置模拟的条件
Analysis statement	设置扫描变量、设置分析模式
.print/.plot/.graph/.probe	设置输出结果的显示方式
Sources (I or V)	设置输入激励
netlist	电路网表
.lib	元件库
.model libraries	元件模型描述
.end	结束语句

注：网表里面不区分大小写

基本词法

1. 不区分大小写 (case insensitive)
2. 标题行——第一行
3. 注释: * or \$
4. 续接符: +
5. 非线性器件要用 .MODEL statement
6. 0, GND, GND!, GROUND 表示全局地
7. 不要有悬空节点、不能重复定义节点
8. Output variables: (后面会有详细的讲解)
 - 两点间的电压: $v(n1, n2)$
 - 节点到地电压: $v(n1)$
 - 独立电流源: $i(vin)$
9. Tab键、空格、逗号、等号、括号都是分隔符, 元件属性由冒号分隔 ($M1: beta$), 级别由句号指示 ($X1.A1.V$)
10. 文件名、语句、等式的长度不能超过256字符
11. 表示数量的词尾: 默认是国际单位, 详细单位见下表。

基本单位

F (f)	1.00E-15
P (p)	1.00E-12
N (n)	1.00E-09
U (u)	1.00E-06
M (m)	1.00E-03
K (k)	1.00E+03
Meg (meg)	1.00E+06
G (g)	1.00E+09
T (t)	1.00E+12
DB (db)	20log10

注：关键不要搞错M和MEG！

器件定义

- 无源器件：电阻、电感、电容

1、电阻

RXXX n1 n2 <mname> <R=>resistance <AC=val> 电阻值可以是表达式。

例：R1 1 2 10K

Rac 9 8 1 AC=1e10

Rterm input gnd R='sqrt(HERTZ)'

2、电容

CXXX n1 n2 <mname> <C=>capacitance

例：C1 1 2 1pF

3、电感

LXXX n1 n2 <L=>inductance

例：L1 1 2 1nH

器件定义

- **有源器件：Diode、BJT、JEFET、MOSFET**

1、Diode（二极管）

DXXX N+ N- MNAME<AREA> <OFF> <IC=VD>

可选项：AREA是面积因子，OFF是直流分析所加的初始条件，IC=VD是瞬态初始条件

注：模型中的寄生电阻串联在正极端

2、BJT（双极性晶体管）

QXXX NC NB NE <NS> MNAME<AREA> <OFF> <IC=VBE, VCE>

NC、NB、NE、NS分别是集电极、基极、发射极和衬底节点，缺省时NS接地。后面与二极管相同。

3、JFET（结型场效应晶体管）

JXXX ND NG NS MNAME<AREA> <OFF> <IC=VDS, VGS>

器件定义

4、MOSFET（MOS场效应晶体管）

MXXX ND NG NS NB MNAME <L=VAL> <W=VAL> <Other options>

M为元件名称，ND、NG、NS、NB分别是漏、栅、源和衬底节点。MNAME是模型名，L沟道长，W为沟道宽。

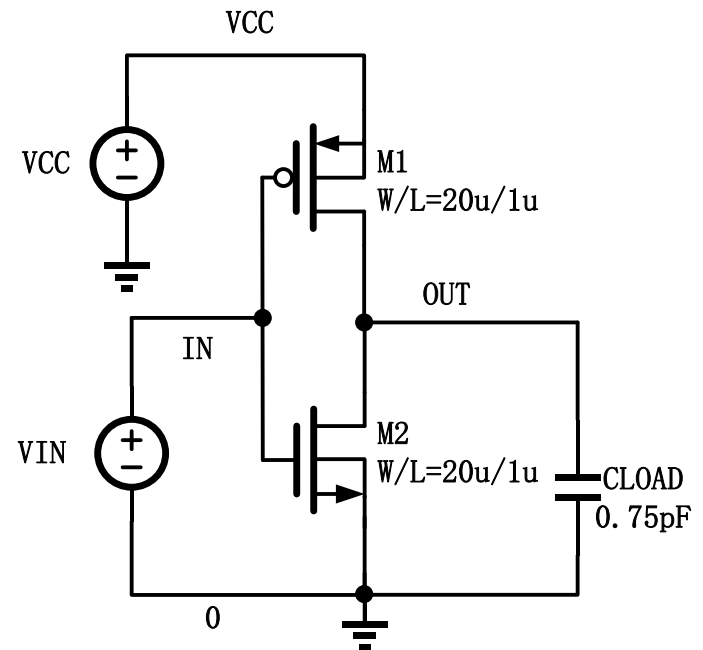
例：下面是一个CMOS反相器网表：

.....

M2 out in 0 0 NMOS W=20u L=1u

M1 out in vcc vcc PMOS W=20u L=1u

.....



器件定义

• 子电路

1、子电路定义开始语句

`.SUBCKT SUBNAM <node1 node2...>`

其中，SUBNAM为子电路名，node1...为子电路外部节点号，不能为零。子电路中的节点号（除接地点），器件名，模型的说明均是局部量，可以和外部的相同。

例： `.SUBCKT OPAMP 1 2 3 4`

2、子电路终止语句

`.ENDS <SUBNAM>`

若后有子电路名，表示该子电路定义结束；若没有，表示所有子电路定义结束。

例： `.ENDS OPAMP / .ENDS`

器件定义

3、子电路调用语句

X***** <node1 node2 ...> SUBNAM

在Spice中，调用子电路的方法是设定以字母X开头的伪元件名，其后是用来连接到子电路上的节点号，再后面是子电路名。

例：

```
.SUBCKT INV IN OUT wn=1.2u wp=1.2u
```

```
Mn out in 0 0 NMOS W=wn L=1.2u
```

```
Mp out in vdd vdd PMOS W=wp L=1.2u
```

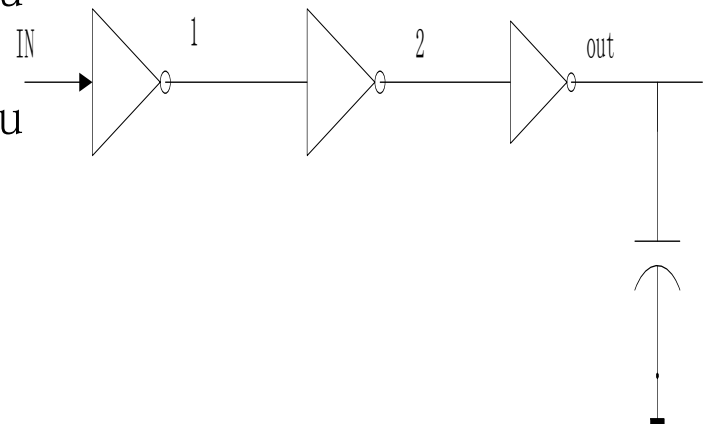
```
.ENDS
```

```
X1 IN 1 INV WN=1.2U WP=3U
```

```
X2 1 2 INV WN=1.2U WP=3U
```

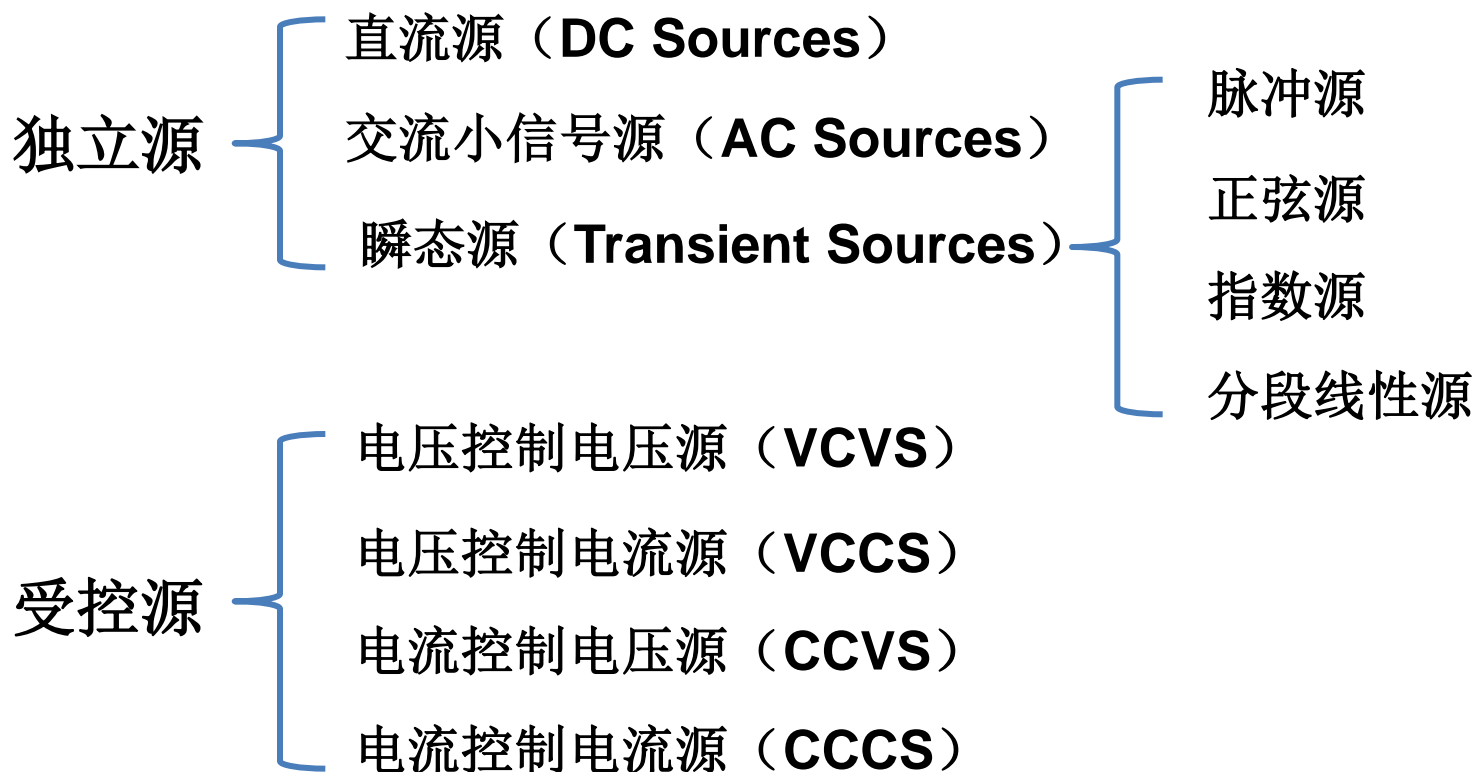
```
X3 2 OUT INV WN=1.2U WP=3U
```

.....



器件定义

- 激励源：独立源和受控源



器件定义

- 独立源

1、直流源（ **DC Sources** ）

VXXX N+ N- DC VALUE

IXXX N+ N- DC VALUE

例： V1 1 0 DC=5V

I1 1 0 DC=5mA

2、交流小信号源（ **AC Sources** ）

VXXX N+ N- AC<ACMAG<ACPHASE>>

IXXX N+ N- AC<ACMAG<ACPHASE>>

其中，ACMAG和ACPHASE分别表示交流小信号源的幅度和相位。

例： V1 1 0 DC=5V AC 1V

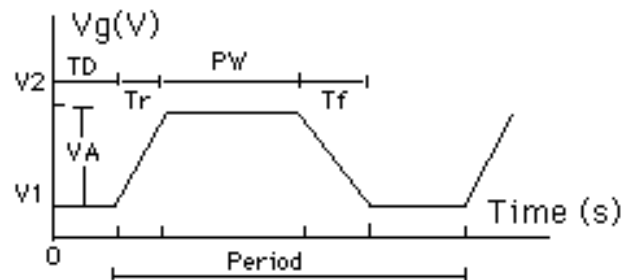
器件定义

3、瞬态源 (Transient Sources)

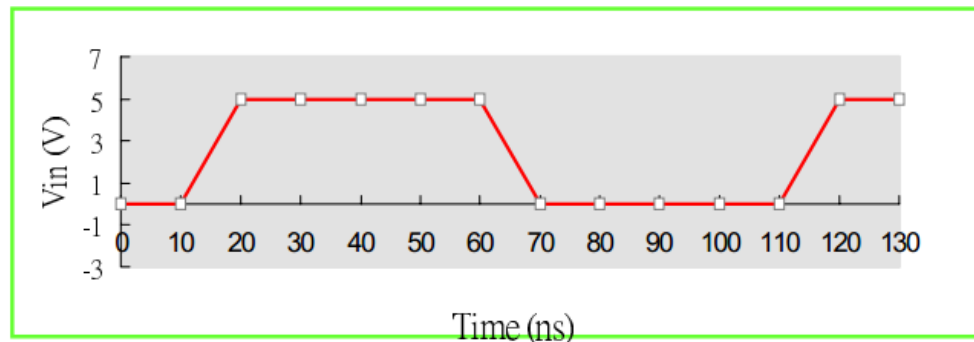
➤ 脉冲源 (又称周期源, PULSE Sources)

VXXX N+ N- PULSE (V1 V2 TD TR TF PW PER)

其中, V1初始值, V2脉动值, TD延时, TR上升时间, TF下降时间, PW脉冲宽度, PER脉冲周期。



例: $V_{in} \ 1 \ 0 \ PULSE \ (0V \ 5V \ 10ns \ 10ns \ 10ns \ 40ns \ 100ns)$



器件定义

➤ 分段线性源 (PWL Sources)

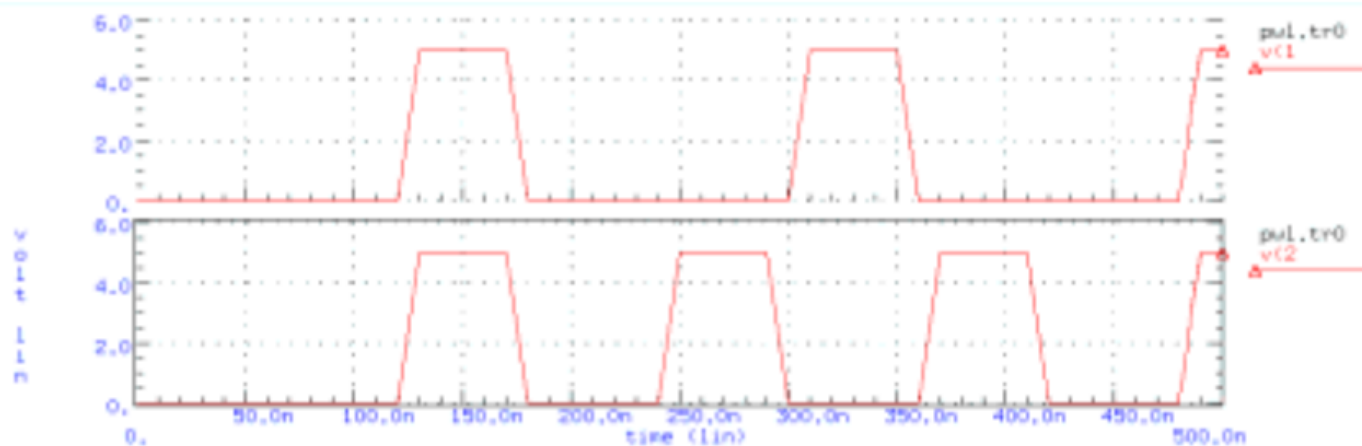
VXXX N+ N- PWL (T1 V1 <T2 V2 T3 V3 ...>) <R<=repeat>> <TD=delay>

\$ R=repeat_from_what_time TD=time_delay_before_PWL_start

其中, V_i 是 T_i 时刻的值, repeat 是开始重复的起始点, delay 是延迟时间

例: V1 1 0 PWL 60n 0V, 120n 0V, 130n 5V, 170n 5V, 180ns 0V, R 0

V2 2 0 PWL 60n 0V, 120n 0V, 130n 5V, 170n 5V, 180ns 0V, R 60n



器件定义

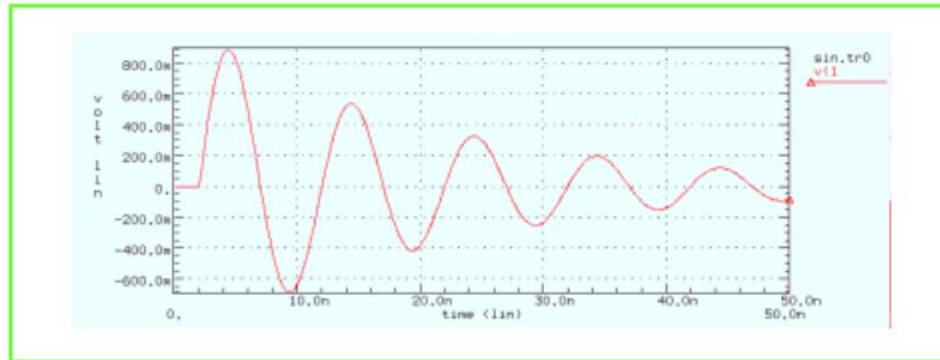
➤ 正弦源 (Sine Sources)

VXXX N+ N- SIN(V0 VA FREQ TD THETA PHASE)

其中, V0偏置, VA幅度, TD延时, THETA阻尼因子, PHASE相位
得到的波形为:

Time=0~TD	$V0 + VA \cdot \sin(2\pi\phi/360)$
Time=TD~瞬态分析的结束时间	$V0 + VA \exp[-(Time - TD) \times PHASE] \cdot \sin\{2\pi \cdot [FREQ(Time - TD) + \phi/360]\}$

例: VIN 3 0 SIN (0V 1V 100MEG 2NS 5e7)



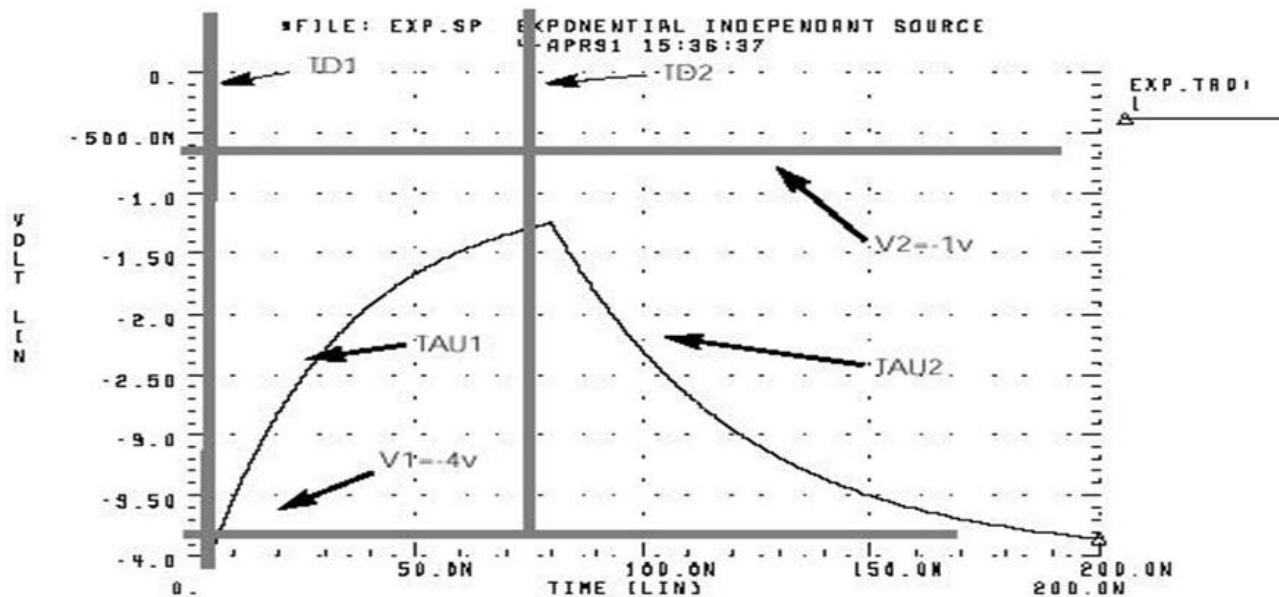
器件定义

➤ 指数源 (EXP Sources)

VXXX N+ N- EXP(V1 V2 TD1 TAU1 TD2 TAU2)

V1是初始值, V2是峰值, TD1是上升延迟时间, TAU1是上升时间常数, TD2是下降延迟时间, TAU2是下降时间常数。

例: VIN 3 0 EXP (-4 -1 5N 30N 40N 80N)



器件定义

• 受控源

1、电压控制电压源（VCVS）

E_{XXX} N+ N- NC+ NC- (Voltage Gain Value)

例：EOPA 3 4 1 2 1E6

EBUF 2 0 1 0 1.0

2、电压控制电流源（VCCS）

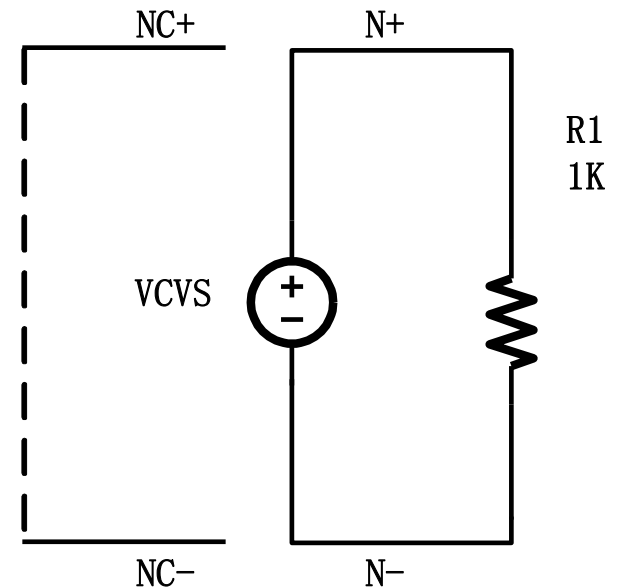
G_{XXX} N+ N- NC+ NC- (Voltage Gain Value)

3、电流控制电压源（CCVS）

H_{XXX} N+ N- NC+ NC- (Voltage Gain Value)

4、电流控制电流源（CCCS）

F_{XXX} N+ N- NC+ NC- (Voltage Gain Value)



器件模型

元器件需要模型语句来定义其参数值。模型语句不同于元器件定义描述语句，它是以“.”开头的语句，由关键字.MODEL，模型名称，模型类型和一组参数组成。

无源器件：定义模型参数即可

◆ 电阻模型

.MODEL 模型名 R keyword=value

NOISE, RX: 热噪声参数, $inr = \text{SQRT}(\text{NOISE} \cdot 4KT/R)$

噪声 = $inr^2 \cdot RX^2$

◆ 电容模型

.MODEL 模型名 C parameter=value

◆ 电感模型

.MODEL 模型名 L parameter=value

器件模型

有源器件：说明都有一个**LEVEL**参数，不同**LEVEL**精度不同

◆ 二极管模型

.MODEL 模型名 D <LEVEL = val> <keyword = val> ...

电阻、电容、电流参数

例：.MODEL D D (CO=2PF, RS=1, IS=1P)

.MODEL DFOWLER D (LEVEL=2, TOX=100, JF=1E-10, EF=1E8)

.MODEL DGEO D (LEVEL=3, JS=1E-4, JSW=1E-8)

.model n d

+level=1 js=1.52e-06 jsw=1.59e-11 n=1.0752 rs=1.28e-08

ik=3.57e+04

+ikr=0.00 vb=100.0 ibv=1.00e-03 trs=-1.72e-04 cta=0.00 ctp=0.00

+eg=1.4235 tcv=0.00 gap1=4.73e-04 gap2=1.11e+03 ttt1=0.00

ttt2=0.00

+tref =25.0 tm1=0.00 tm2=0.00 tpb=0.00 tphp=0.00 xti=-5.0

器件模型

◆ BJT管模型

.MODEL mname NPN <(<pname1 = val1> ... <)>

.MODEL mname PNP <pname1 = val1> ...

模型参数中一般包括LEVEL，说明哪种模型，不同级的模型有不同的模型参数集。

例：上华BJT SPICE模型：

```
.model pnp20 pnp  
+level=1 is=1.01e-16 bf=16.3979  
+nf=0.9909 vaf=261.3346 ikf=1.46e-03  
+ise=4.38e-15 ne=1.9797 br=0.6531  
+nr=0.8241 var=10.0 ikr=1.5  
+isc=1.00e-17 nc=1.5 rb=40.7749  
.....
```

器件模型

◆ MOS管模型

.MODEL 模型名 PMOS <LEVEL=val> <parameters>

.MODEL 模型名 NMOS <LEVEL=val> <parameters>

LEVEL=1 常用于数字电路，精度低、速度快

LEVEL=2 考虑了衬底电荷对电流的影响

LEVEL=13, 39, 49模拟电路，精度高、速度慢

例：1.2um CMOS工艺MOS管SPICE模型：

```
.MODEL NMOS NMOS LEVEL=2 LD=0.15U TOX=200.0E-10 VTO=0.74 KP=8.0E-05  
+NSUB=5.37E+15 GAMMA=0.54 PHI=0.6 U0=656 UEXP=0.157 UCRIT=31444  
+DELTA=2.34 VMAX=55261 XJ=0.25U LAMBDA=0.037 NFS=1E+12 NEFF=1.001  
+NSS=1E+11 TPG=1.0 RSH=70.00 PB=0.58  
+CGDO=4.3E-10 CGSO=4.3E-10 CJ=0.0003 MJ=0.66 CJSW=8.0E-10 MJSW=0.24
```

上华提供的level=49的MOS模型见csmc.lib

器件模型

Level 1 仿真网表:

*Simulation for Level 1

M1 D G S 0 N_1 W=1u L=1u

.model N_1 NMOS

+LEVEL = 1

+L = 1.0E-4 W = 1.0E-4 NRS = 1.0 NRD = 1.0

+VTO = 0 KP = 2.0E-5 GAMMA = 0 UO = 6.0E-02

+PHI = 0.1 LAMBDA = 0 NSUB = 0 TOX = 1.0E-7

+NSS = 0 TPG = 1 IS = 1.0E-14 JS = 1.0E-14

+RS = 0 RD = 0 CBS = 0 CBD = 0

+RSH = 1.0E+10 CJ = 0 MJ = 0.5

+PB = 0.8 CJSW = 0 MJSW = 0.5 CGSO = 0

+CGDO = 0 CGBO = 0 KF = 0 AF = 1

+NSUB = 1E15 PHI = 0.576

VD D 0 3

VS S 0 0

VG G 0 1.5

.DC VD 0 3 0.1 VG 1 3 1

.OP

.PRINT I(M1)

.OPTION post

.END

Level 49 仿真网表:

*Simulation for level 49

M1 D G S 0 N_33 W=1u L=1u

.lib

'E:\cmos_assistant\CMOS_035_Spice_Model.lib' tt

VD D 0 3

VS S 0 0

VG G 0 1.5

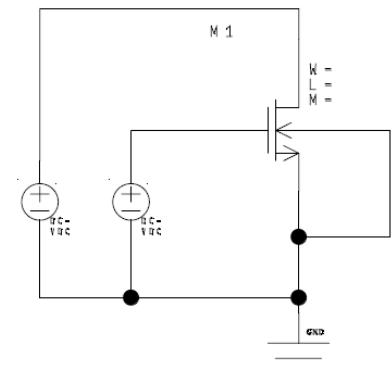
.DC VD 0 3 0.01 VG 1 3 1

.OP

.PRINT I(M1)

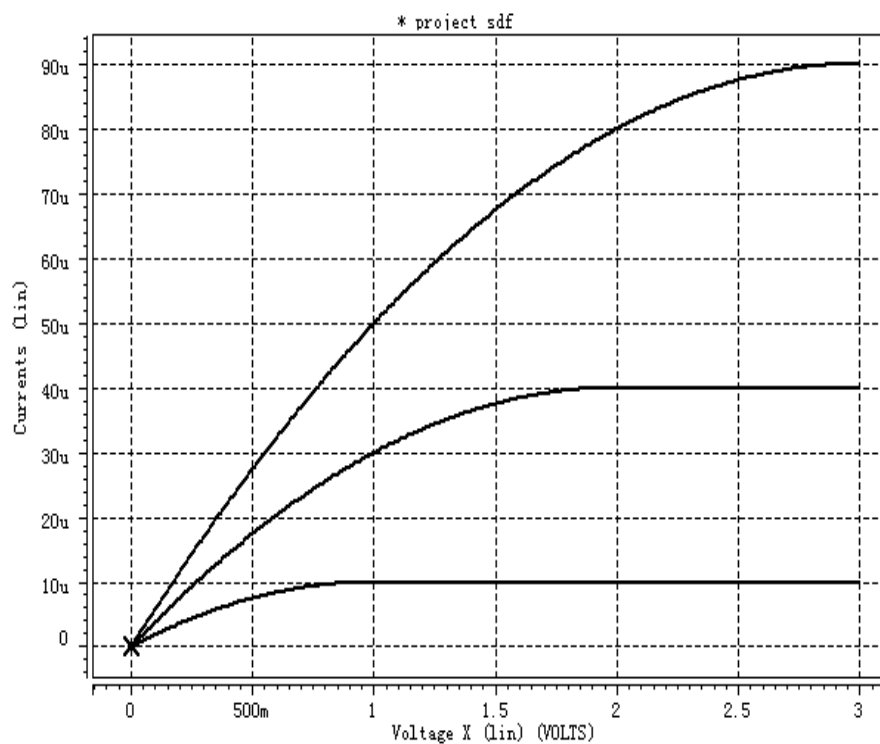
.OPTION post

.END

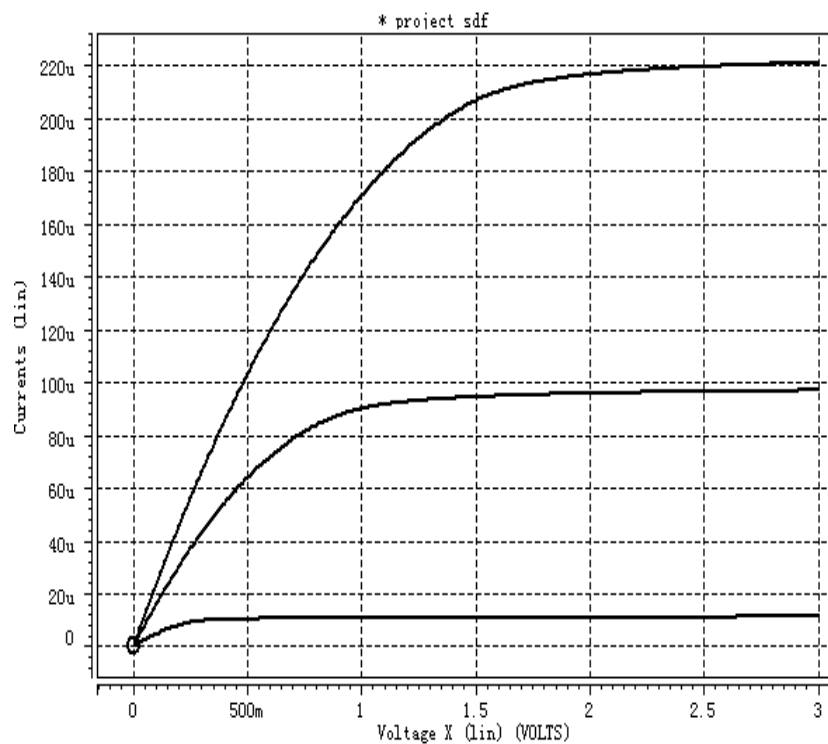


器件模型

仿真结果对比:



Level 1仿真结果



Level 49仿真结果

分析类型描述语句

.OP: 直流工作点分析

- 严格来说，这不算一种仿真类型。计算直流工作点指令，对于分析电路很有用（要会看.lis文件），会在输出文件中列出一些直流参数和各结点的工作点电压与支路电流、静态功耗。
- 一般在任何其它仿真之前都需要计算直流工作点。
- .op time看某一时刻的各个器件和电源的状态。
- .op vol time某一时刻所有节点电压。
- .op cur time某一时刻所有器件和电源的电流。
- 默认状态：.ac和.dc分析是初始工作点，.tran默认时间是0。

分析类型描述语句

.DC: 直流分析

- 可以对参数和独立电源扫描
- .DC var1 START STOP STEP/<SWEEP var2 type np start2 stop2>
其中, type有DEC(十进位)/OCT(倍频)/LIN(线性)/DATA=datanm/POI(列表), Np是单位范围内的点数(依type而定)
- 注意:
 - 1) 对独立源扫描时, var为电源名, 而非节点名
 - 2) 仿真迟滞特性要正、负双向扫描,
如: .DC Vin 0 5 0.1
.DC Vin 5 0 -0.1
 - 3) 注意对两个量的扫描, 后面的是外循环
 - 4) SWEEP后的变量可是电压、电流或温度等变量

分析类型描述语句

例：分析反相器链的直流传输特性和工作点

.....

.global vdd

.SUBCKT INV IN OUT wn=1.2u wp=1.2u

.....

.ENDS

X1 IN 1 INV WN=1.2U WP=3U

X2 1 2 INV WN=1.2U WP=3U

X3 2 OUT INV WN=1.2U WP=3U

CL OUT 0 1PF

VCC VDD 0 5V

VIN IN 0

.DC VIN 0 5V 0.1V / SWEEP TEMP -20 80 1

.OP

.....

.END

分析类型描述语句

.TRAN: 瞬态分析

.TRAN:

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

例: .TRAN 1NS 100NS

注: 以1ns的步长输出到100ns

.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN> <START=val> <UIC>

起始时刻和步长都指的是输出打印的时刻点, 计算的时间步长由hspice自己决定。UIC参数表示使用.IC语句指定的节点初始值。

例: .TRAN .1NS 25NS 1NS 40NS START=10NS

注: 0—25ns, 步长0.1ns, 25ns-40ns, 步长1ns; 从10ns开始输出结果。

分析类型描述语句

.AC: 交流分析

.AC:

.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 = start1 STOP = stop1 STEP = incr1

例: .AC DEC 10 1K 100MEG

- 注: 一共有4种
 - DEC – 十进制的(decade variation)
 - OCT – 八进制的(octave variation)
 - LIN – 线形的(linear variation)
 - POI – 列举的(list of points)

分析类型描述语句

.NOISE: 噪声分析

用来计算各个器件的噪声对输出节点的影响并给出其均方根并输出，可完成.AC语句规定的各频率的计算，应在.AC分析之后。

.NOISE ovv srcnam inter

Ovv—输出变量，srcnam—输入源，inter—频率间隔

例：.title ac sweep example

.OPTIONS POST

R1 in 1 5

C1 1 0 500pf

V1 IN 0 0 AC=10V,37

.AC OCT 10 1 100MEG

.noise v(1) v1 20——分析1点电压的噪声情况，噪声源为V1端口

.END

分析类型描述语句

```
***** Star-HSPICE -- 1999.4 (19991220) 22:12:12 04/16/2002 pcnt
```

```
*****
```

```
.title ac sweep example
```

```
***** noise analysis          tnom= 25.000 temp= 25.000
```

```
***** frequency = 1.0000 hz
```

```
**** resistor squared noise voltages (sq v/hz)
```

```
element 0:r1
```

```
total 8.233e-20
```

```
rx 5.0000
```

```
**** total output noise voltage = 8.233e-20 sq v/hz  
                                = 286.9260p v/rt hz
```

```
transfer function value:    v(1)/v1 = 1.0000
```

```
equivalent input noise at v1 = 286.9260p /rt hz
```

```
**** the results of the sqrt of integral (v**2 / freq)
```

```
from fstart upto 1.0000 hz. using more freq points
```

```
results in more accurate total noise values.
```

```
**** total output noise voltage = 0. volts
```

```
**** total equivalent input noise = 0.
```

```
.....
```

最后给出总的
输出噪声电压
和等效到输入
端的噪声电压

控制语句

.INCLUDE语句

.INCLUDE语句：引用一个文件，被引用的文件置于引用文件前。

例：LNA

```
.include "me98xxxx/model.sp"
```

...

.LIB 语句：

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

该语句根据文件路径和文件名来调用一个库文件，一般该文件包含器件模型。

例：.lib 'MODELS' cmos1

MODELS文件：

...

```
.MODEL CMOS1 nmos ...
```

我们仿真中加库的语句应该是

```
.LIB 'D:\TEST\PROCESS\0.6U BCD V0.1PHASE1.LIB' TT
```


控制语句

.PARAM 语句

.PARAM语句：定义一个变量。

.PARAM <parametername> = '<Expression>'

例子：.PARAM width = 20u

.PARAM length = 'sqrt(width)*1.65'

M1 3 2 0 0 NMOS width length

.DATA 语句

.DATA语句：给一个数组赋值。

例子：.DATA D1 width length RL

+	50u	20u	1K
+	60u	10u	10K
+	100u	25u	1K

.ENDDATA

控制语句

.ALTER语句

.ALTER语句：使用不同的参数和数据以返回一个仿真结果。

语法：.ALTER <title_string>

title_string 是任何最长至72个字母的字符串。.alter运行所需的合适的字符串在每个输出列表文件和图形数据文件（.tr#）的标题部分打印出来。

例：*file2: alter2.sp alter examples \$ Title Statement

● .ALTER Statement : Limitations

■ CAN Include:

- ➔ Element Statement (Include Source Elements)
- ➔ .DATA, .LIB, .INCLUDE, .MODEL Statements
- ➔ .IC, .NODESET Statement
- ➔ .OP, .PARAM, .TEMP, .TF, .TRAN, .AC, .DC Statements

■ CANNOT Include:

- ➔ .PRINT, .PLOT, .GRAPH, or any I/O Statements

.param wval=1000 fdd=0.01 \$ change parameters
.end

控制语句

.GLOBAL 语句

.GLOBAL语句：定义全局变量。

.GLOBAL parametername (XX)

例子：.GLOBAL VDD

.GLOBAL VSS

注：上述语句会把所有命名为XX的节点连接起来，包括子电路里面的节点。

Global的使用需格外小心，一般仅把电源定义成Global

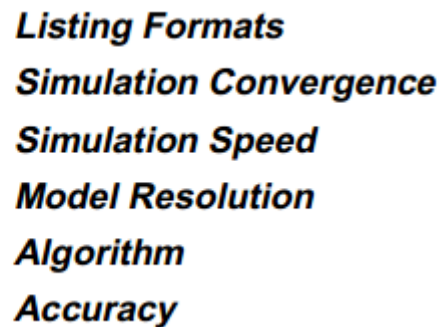
Options语句

.Options 语句

.Options语句：可选设置语句，该语句允许用户重新设置程序的参数或控制程序的功能。

.OPTIONS opt1 <opt2> ... <opt=x>

■ .Option Controls for



Listing Formats
Simulation Convergence
Simulation Speed
Model Resolution
Algorithm
Accuracy

常用的一些如下：

node: 列出个节点的元件端点，便于查错；

post: 使输出数据使用 AvantWaves 浏览(即将数据输出到post processor)

list: 列出输入元件列表；

Options语句

.Options 语句

例: .OPTIONS POST PROBE

AvantWaves只观察.probe语句输出的变量。因为仿真文件会保存所有的电压，造成了仿真文件过大，很多节点电位没有必要知道，所以可以通过选择性的输出自己要的节点电位，从而减小仿真的文件。

```
.OPTIONS PROBE
```

```
.PROBE V(VOUT)
```

以上的命令只输出VOUT节点的电位。

除了以上的还有其它的参数详细Hspiceguide中。

常用的是fast, bypass=0 加速仿真，但是会以牺牲精度为代价，建议一般仿真不要用。

输出格式描述语句

• 输出语句

- **.PRINT**: 在输出的list文件中打印数字的分析结果, 如果.OPTIONS中有POST则同时输出到post-processor中。
- **.PLOT**: 在输出的list文件中打印低分辨率的曲线 (由ASCII字符组成), 如果.OPTIONS中有POST则同时输出到post-processor中。
- **.GRAPH**: 生成用于打印机或PostScript格式的高分辨率曲线。
- **.PROBE**: 把数据输出到post-processor, 而不输出到list文件。
- **.MEASURE**: 输出用户定义的分析结果到mt0文件, 如果.OPTIONS中有POST则同时输出到post-processor中。
- **.OP, .TF, .NOISE, .SENS和.FOUR**都提供直接输出功能。

输出格式描述语句

.PRINT

.PRINT antype ov1 <ov2 ... ov32>

Antype——AC/DC/TRAN;

Ovi——输出变量，可以有以下形式：

V(1): 节点1的电平

V(1, 2): 1、2间的电压

V(R1): 电阻R1的电压;

VM(1): v1的幅值,

VR(1): v1的实部,

VI(1): v1的虚部,

VP(1): v1的相位,

VDB(1): v1的分贝值;

(电流与以上类似)

● Examples :

```
.PRINT TRAN V(4) V(X3.3) P(M1) P(VIN) POWER PAR('V(OUT)/V(IN)')  
.PRINT AC VM(4,2) VP(6) VDB(3)  
.PRINT AC INOISE ONOISE VM(OUT) HD3  
.PRINT DISTO HD3 HD3(R) SIM2  
.PLOT DC V(2) I(VSRC) V(37,29) I1(M7) BETA=PAR('I1(Q1)/I2(Q1)')  
.PLOT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)  
.PLOT TRAN V(5,3) (2,5) V(8) I(VIN)
```

输出格式描述语句

.PLOT

.PLOT: .PLOT antype ov1 <(plo1, phi1)> ... <ov32>
+<(plo32, phi32)>(plo1, phi1) —ov1绘图的上下限。

.PROBE

.PROBE: .PROBE antype ov1 ... <ov32>

*元件电流引用: BJT: I1(Qx)–Ic, I2(Qx)–Ib, I3(Qx)–Ie, I4(Qx)–衬底电流; MOS:I(Mx)–Ids。

◆ .PRINT ac V(1) S11(DB) S21(m) S22(DB)

◆ .TRAN 1N 200N

.PROBE V(OUT)

◆ .NOISE v(out) vin 10

.PRINT noise onoise inoise

.PRINT im(rd)

输出格式描述语句

.MEASURE

- 包括以下测量模式：
 - Rise, fall, and delay (上升, 下降和延迟)
 - Find-when (寻找时间)
 - Equation evaluation (方程计算)
 - Average, RMS, min, max, and peak-to-peak
 - Integral evaluation (相对错误)
 - Derivative evaluation (求导计算)
 - Relative error (积分计算)

– `.MEASURE <DC|AC|TRAN> result TRIG ... TARG ...`

Result是测量结果的名字, TRIG ... TARG表示起始...中止, 对瞬态分析它是时间, 对AC分析它是频率, 对DC分析它是DC扫描变量。如果target在trigger激活以前到达, 结果的值是负的

输出格式描述语句

.MEASURE

例：计算反相器链电路的延迟时间

```
.TITLE 1.2UM CMOS INVERTER CHAIN
```

```
.INCLUDE "models.sp"
```

```
.....
```

```
X1 IN 1 INV WN=1.2U WP=3U
```

```
X2 1 2 INV WN=1.2U WP=3U
```

```
X3 2 OUT INV WN=1.2U WP=3U
```

```
CL OUT 0 1PF
```

```
VCC VDD 0 5V
```

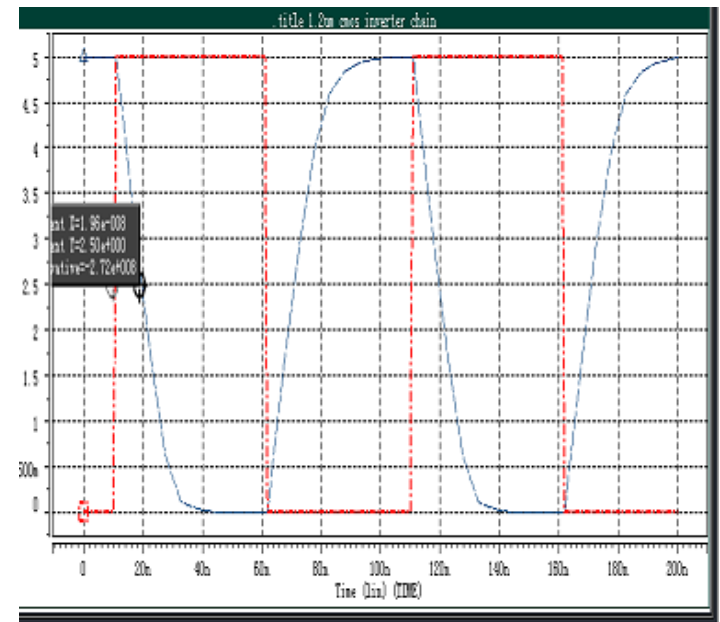
```
VIN IN 0 PULSE(0 5V 10NS 1N 1N 50N 100N)
```

```
.TRAN 1N 200N
```

```
.MEASURE tran tdelay trig v(in) val=2.5 td=8ns rise=1
```

```
+ targ v(out) val=2.5 td=9n fall=1
```

```
.END
```



输出格式描述语句

.MEASURE

➤ 分析功耗

`.print/plot <dc或tran> P(element) power`

其中，**power** 关键词用来计算整个电路的功率。

例：`.print tran P(M1) P(Vin) P(Cload) Power`

`.measure tran p AVG POWER from=0n to=100ns`

这里只计算瞬态分析或直流分析中的瞬时功耗或静态功耗。

➤ 分析效率

`.meas tran pload avg p(IRL) from=700u to=800u`

这个语句是测量700u~800u这一段时间的效率，IRL是输出负载电流，AVG是求平均，效率是负载上的功耗除以输入功耗。

输出格式描述语句

.MEASURE

更多MEASURE例子:

```
.MEASURE TRAN t_delay TRIG v(in) VAL=2.5 CROSS=1 TARG v(out) VAL=2.5  
CROSS=1
```

测量V(out)第一次到达2.5V和V(in)第一次到达2.5V之间的延迟时间。

```
.MEAS t_rise TRIG v(out) VAL=0.5 RISE=1 TARG v(out) VAL=4.5 RISE=1
```

测量V(out)第一次从10%上升90%所需要的时间。

```
.MEAS TRAN max_current MAX I(Vdd)
```

测量Vdd这个电压源流过的最大电流

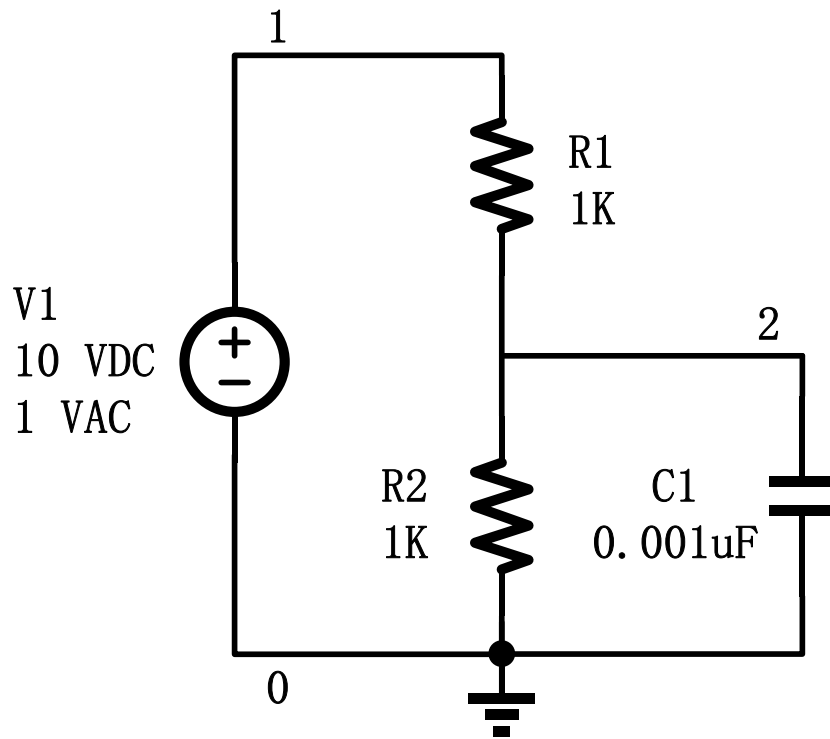
```
.MEAS peak_power PAR='max_current*VIN'
```

计算峰值功耗

实例演示

实例演示

- 1、RC网络AC分析



实例演示

```
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
```

注释：第一行 A SIMPLE AC RUN 为标题行；

第二行 .OPTIONS LIST NODE POST 为可选项设置，LIST 打印出元件总结列表；NODE 打印出 元件节点表 (element node table)；POST 表示用何种格式储存模拟后的数据，以便与其它工具接口。

第三行 .OP 计算直流工作点。

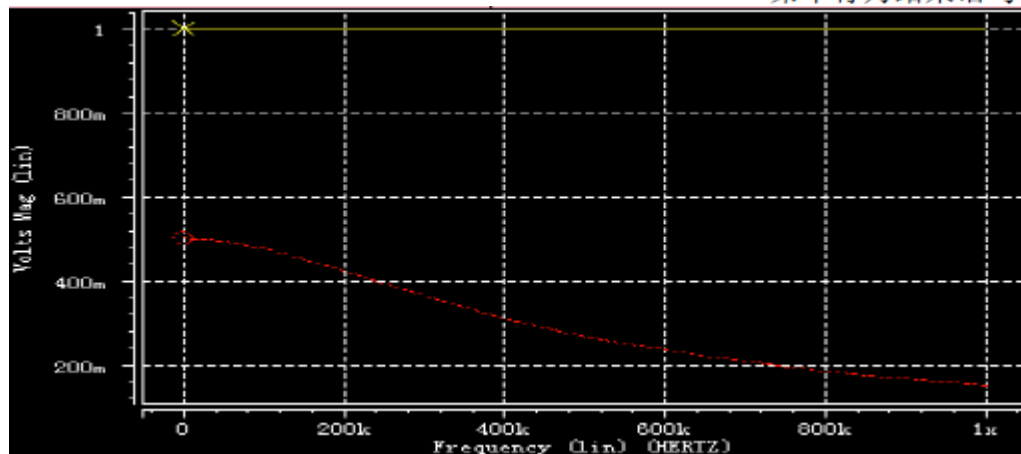
第四行 .AC DEC 10 1K 1MEG (指从 1 到 10KHZ 范围，每个数量级取 10 点，交流小信号分析)

第五行 .PRINT AC V(1) V(2) I(R2) I(C1) 打印交流分析类型的节点 1, 2 的电压，以及 R2, C1 的电流

第六行 V1 1 0 10 AC 1 表示节点 1 与 0 间，加直流电压 10v 和幅值为 1v 的交流电压。

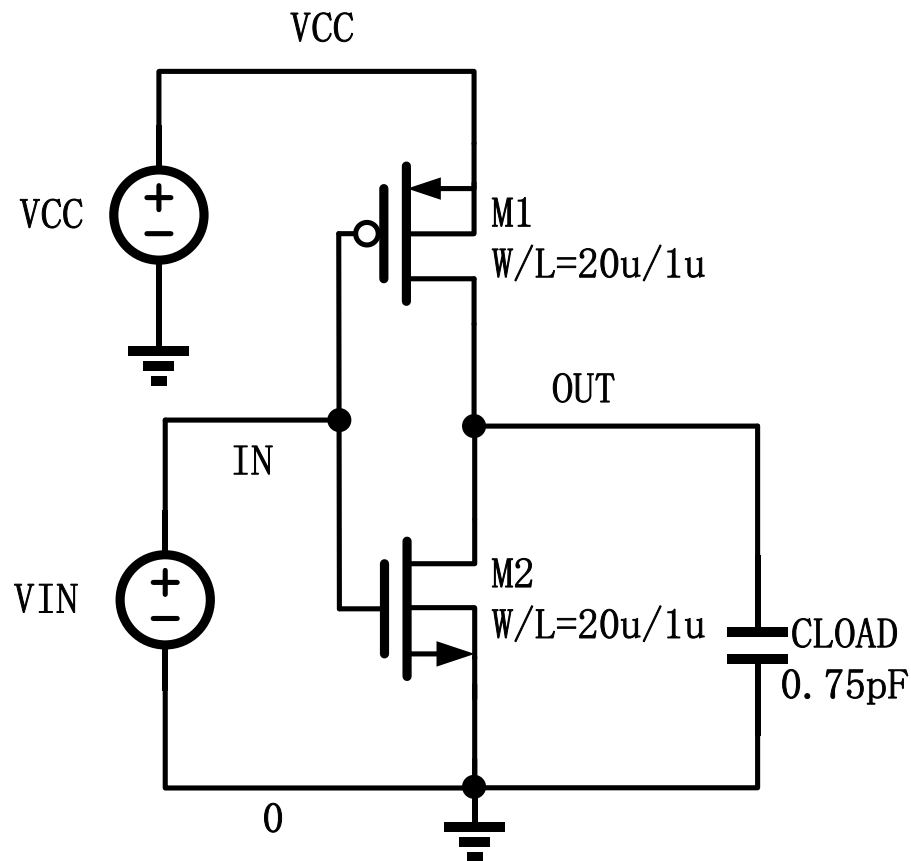
第七至九行为电路描述语句。

第十行为结束语句。



实例演示

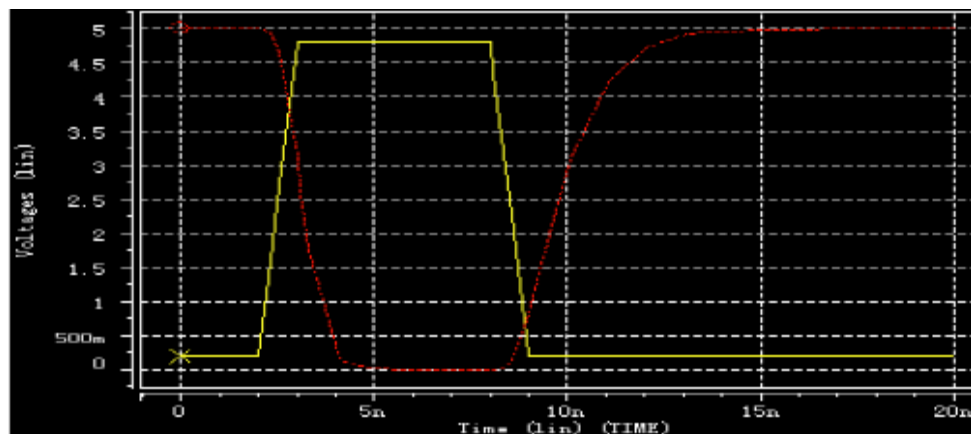
• 2、反相器TRAN分析



VIN: PULSE(0.2 4 0.8
2N 1N 1N 5N 20N)

实例演示

```
Inverter Circuit
.OPTIONS LIST NODE POST
.TRAN 200P 20N
.PRINT TRAN V(IN) V(OUT)
M1 OUT IN VCC VCC PCH L=1U W=20U
M2 OUT IN 0 0 NCH L=1U W=20U
VCC VCC 0 5
VIN IN 0 0 PULSE .2 4.8 2N 1N 1N 5N 20N
CLOAD OUT 0 .75P
.MODEL PCH PMOS LEVEL=1
.MODEL NCH NMOS LEVEL=1
.END
```



注释：第三行.TRAN 200P 20N表示瞬态分析步长为200ps，时间为20ns

第四行.PRINT TRAN V(IN) V(OUT)表示打印节点in, out 电压瞬态分析值

第五，六，九行为电路连接关系描述语句。

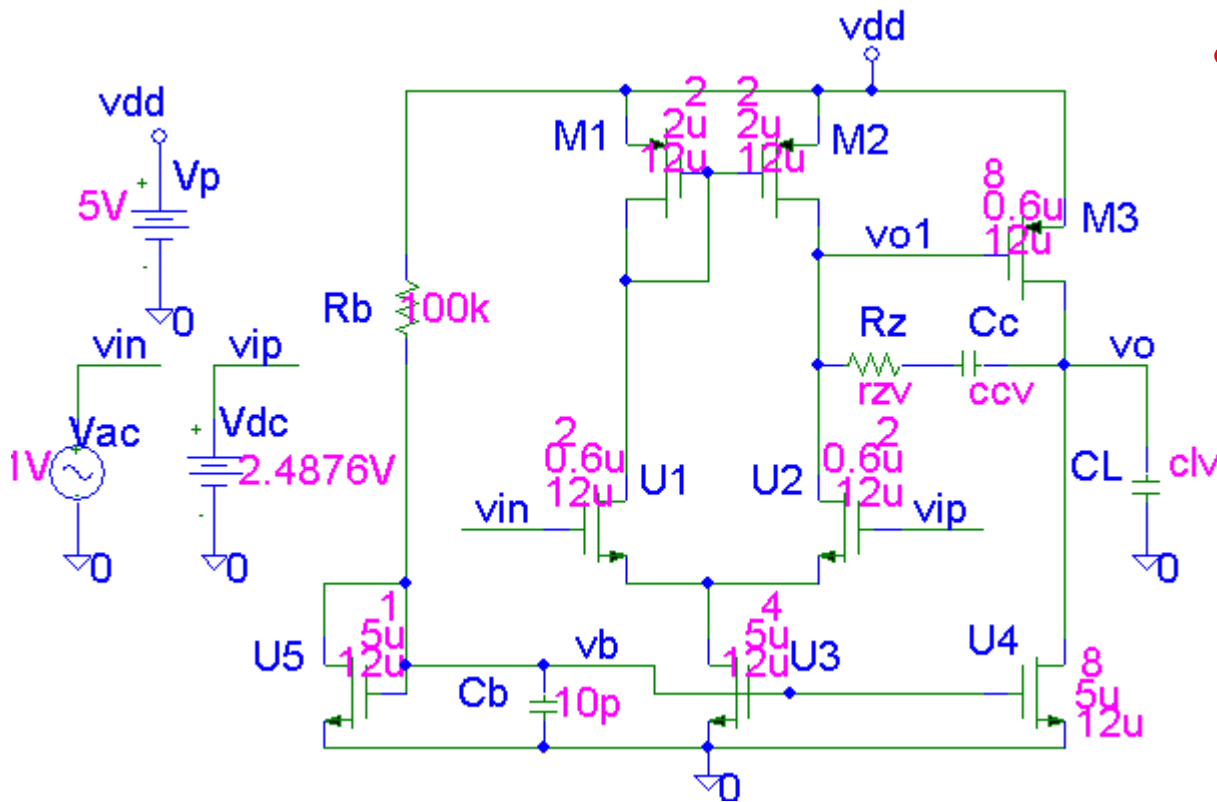
第七行VCC VCC 0 5表示在节点VCC，0之间加5v直流电压。

第八行VIN IN 0 0 PULSE .2 4.8 2N 1N 1N 5N 20N表示在节点IN，0之间加一个脉冲源，低电平0.2v, 高电平4.8v, 延时2ns, 上升沿1ns, 下降沿1ns, 脉冲宽度5ns, 周期20ns

第九，十行为模型语句，表示模型名PCH，管子类型为PMOS，使用的是一级模型。

实例演示

• 3、简单跨导放大器分析



• 指标分析

- 偏置/功耗
- 开环增益
- GBW与PM
- 压摆率
- 噪声
- 工艺corner分
- 温度特性分析

实例演示

V_Vp vdd 0 5V
V_Vac vin 0 DC 2.5V AC 1V 0
V_Vdc vip 0 2.5V

R_Rz vo1 N_0001 rzv

C_Cc N_0001 vo ccv

C_CL 0 vo clv

C_Cb 0 vb 10p

R_Rb vb vdd 100k

M_U2 vo1 vip N_0002 0 nm L=0.6u W=12u M=2

M_M1 N_0003 N_0003 vdd vdd pm L=2u W=12u M=2

M_M3 vo vo1 vdd vdd pm L=0.6u W=12u M=8

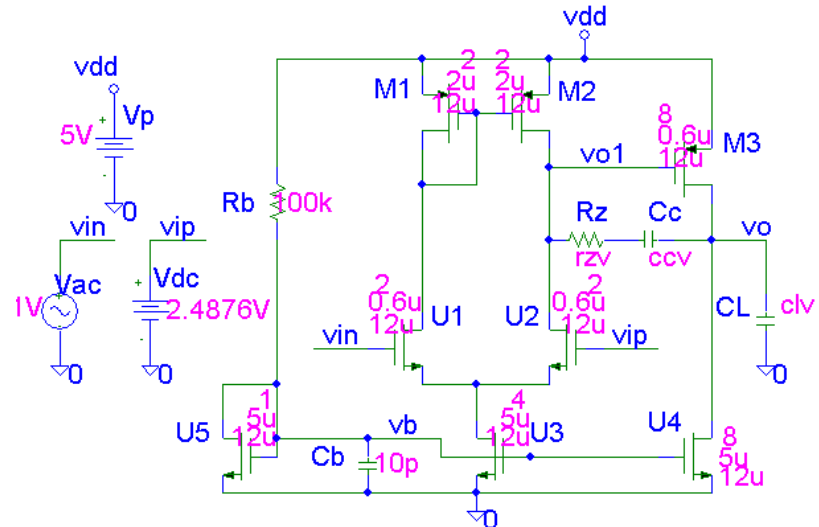
M_U1 N_0003 vin N_0002 0 nm L=0.6u W=12u M=2

M_U4 vo vb 0 0 nm L=5u W=12u M=8

M_U5 vb vb 0 0 nm L=5u W=12u M=1

M_U3 N_0002 vb 0 0 nm L=5u W=12u M=4

M_M2 vo1 N_0003 vdd vdd pm L=2u W=12u M=2



实例演示

*工作点分析

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe ac v(vo1) v(vo) vp(vo)  
.op  
*.dc v_vdc 2.48 2.5 0.0001  
*.trans 10ns 200ns 20ns 0.1ns  
*.ac dec 10 1k 100meg $sweep rzv 0 2k 0.2k  
.para rzv=1k ccv=1p clv=1p  
.include 'NETLIST_PATH\ota.net'  
.end
```

实例演示

- 浏览并分析.lis文件的内容
- .prot与.unprot使用将使得其中的内容不在.lis中出现
- 用oper查找，即可找到operating point information这一段，可看到电路各节点的电压、各元件的工作状态
- 注意此时 $v_o=4.8916$
- 对于提供电源的电压源v_vp，注意其功耗就是电路功耗，因此可查得电路功耗为2.47mW
- 对于MOS管，注意各参量的含义：region、id、vgs、vds、vth、vdsat、gm、gmb、gds.....可查得流过M_U3的偏置电流为149.8uA，并注意到M_M3的region为Linear

实例演示

*直流扫描

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe dc v(vo1) v(vo)  
.op  
.dc v_vdc 2.45 2.55 0.001  
*.dc v_vdc 2.48 2.495 0.0001  
*.trans 10ns 200ns 20ns 0.1ns  
*.ac dec 10 1k 100meg $sweep rzv 0 2k 0.2k  
.para rzv=1k ccv=1p clv=1p  
.inc 'NETLIST_PATH\ota.net'  
.end
```

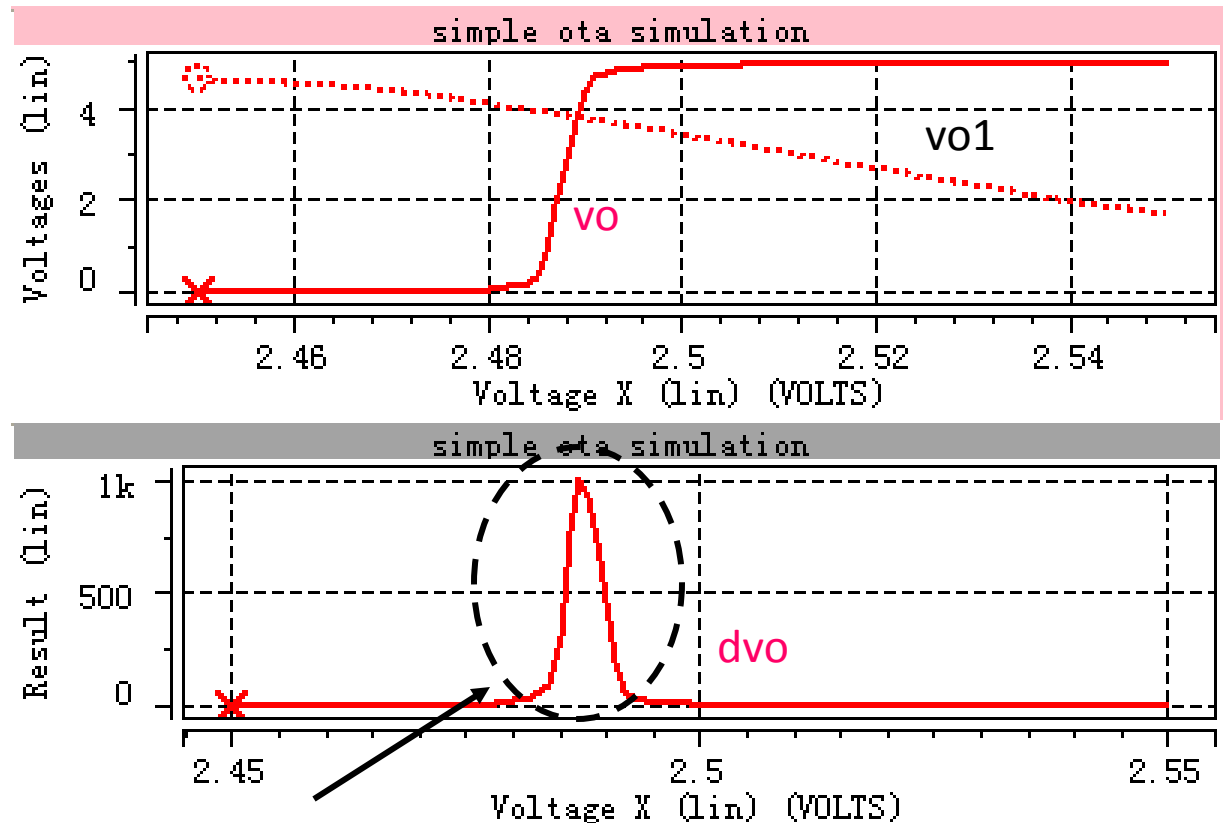
粗扫

精扫

实例演示

对vo求导

小信号增益Gain =
 $d(v_o)/d(v_{vdc})$



确定精扫扫描范围

实例演示

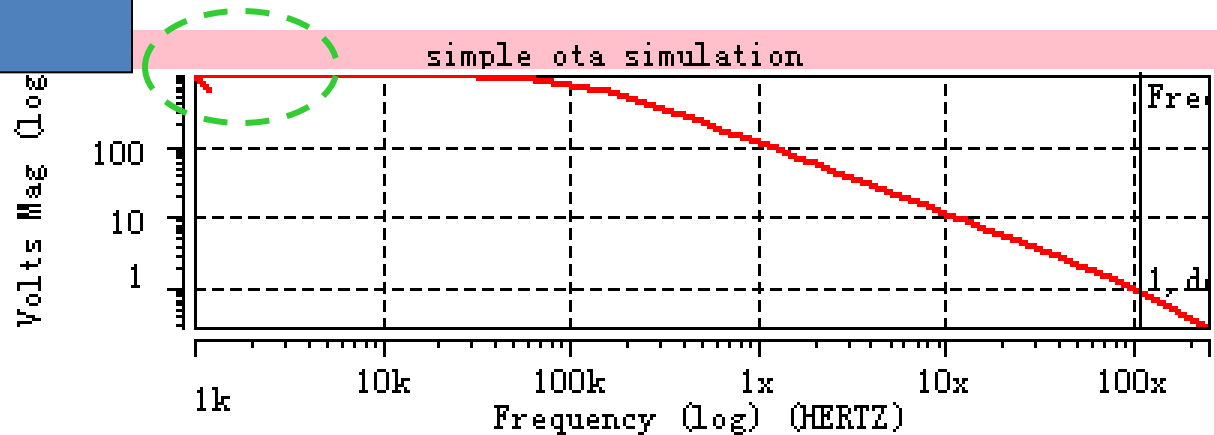
*交流分析

Ota simulation

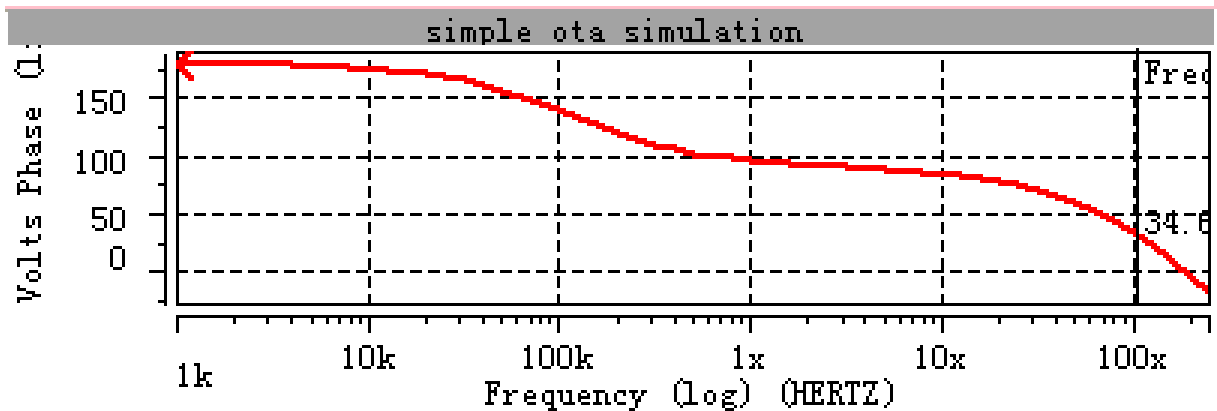
```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe ac v(vo1) v(vo) vp(vo)  
.op  
*.dc v_vdc 2.48 2.495 0.0001  
*.trans 10ns 200ns 20ns 0.1ns  
.ac dec 10 1k 200meg $sweep rzv 0 2k 0.2k  
.para rzv=0 ccv=1p clv=1p  
.inc 'NETLIST_PATH\ota.net'  
.end
```


实例演示

直流small-signal gain



GBW=99.8MHz
相位裕度34.6度



实例演示

*交流扫描

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe ac v(vo1) v(vo) vp(vo)  
.op  
*.dc v_vdc 2.48 2.495 0.0001  
*.trans 10ns 200ns 20ns 0.1ns  
.ac dec 10 1k 200meg sweep rzv 0 2k 0.2k sweep ccv 0 5p 1p  
.para rzv=0 ccv=1p clv=1p  
.inc 'NETLIST_PATH\ota.net'  
.end
```

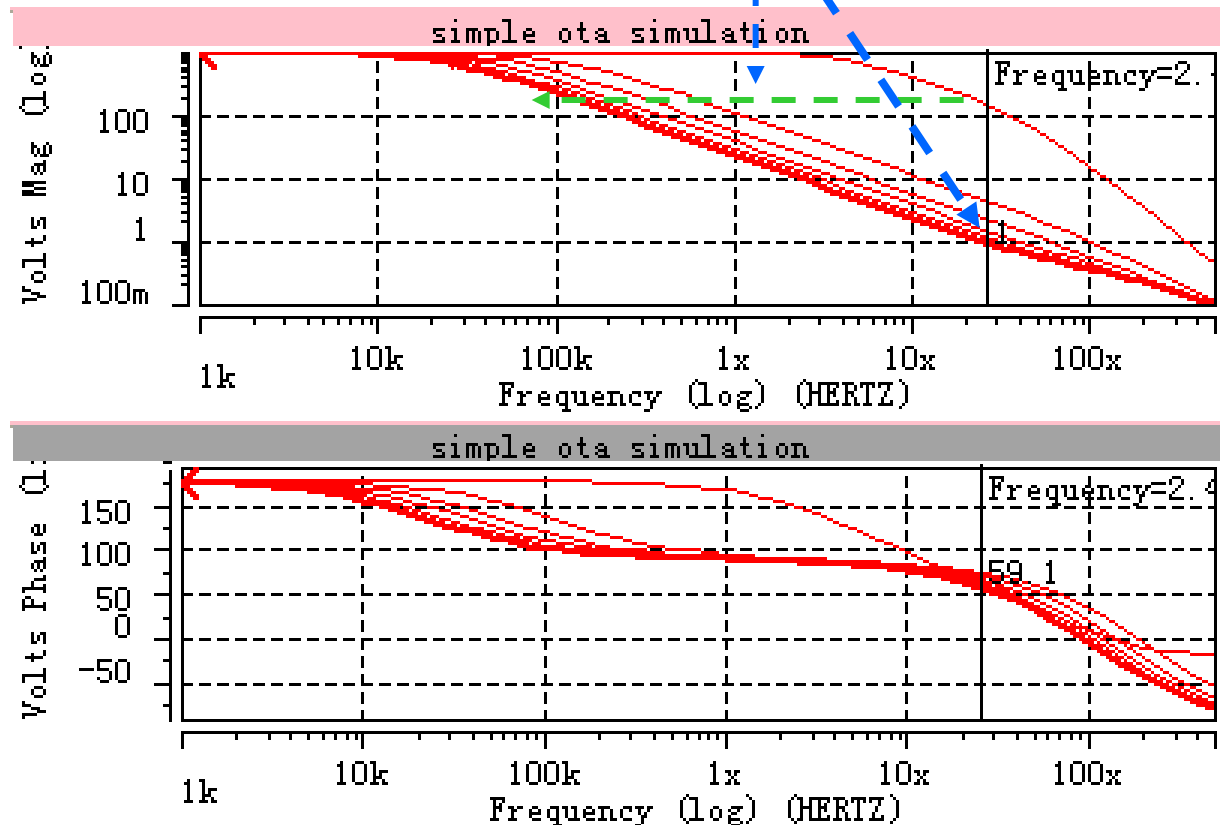
分析miller补偿效应

.ac dec 10 1k 200meg sweep rzv 0 2k 0.2k sweep ccv 0 5p 1p

实例演示

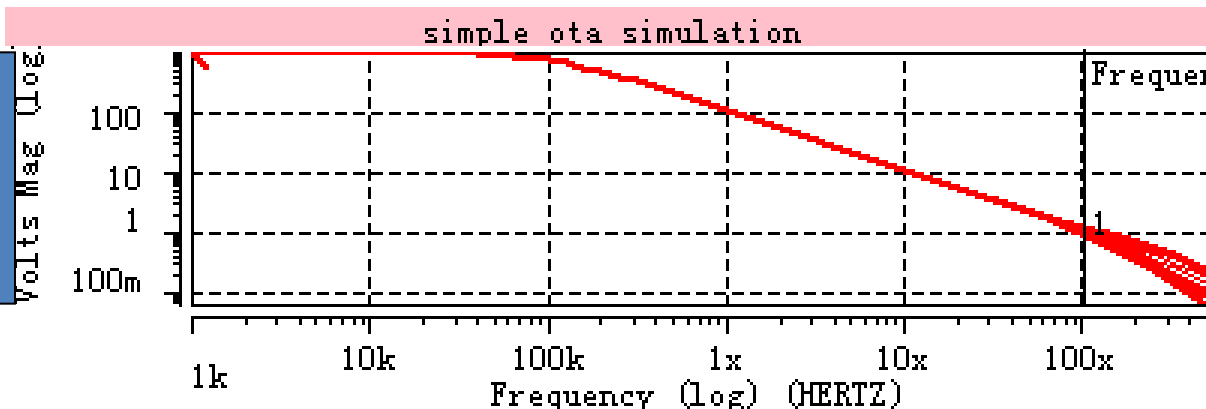
增加 C_c , ω_{p1} 向下移动, GBW减小
相位裕度增加

增加 C_c 到5p时, 相位裕度
增加到约59度, 而GBW已
经减小到24.8MHz!

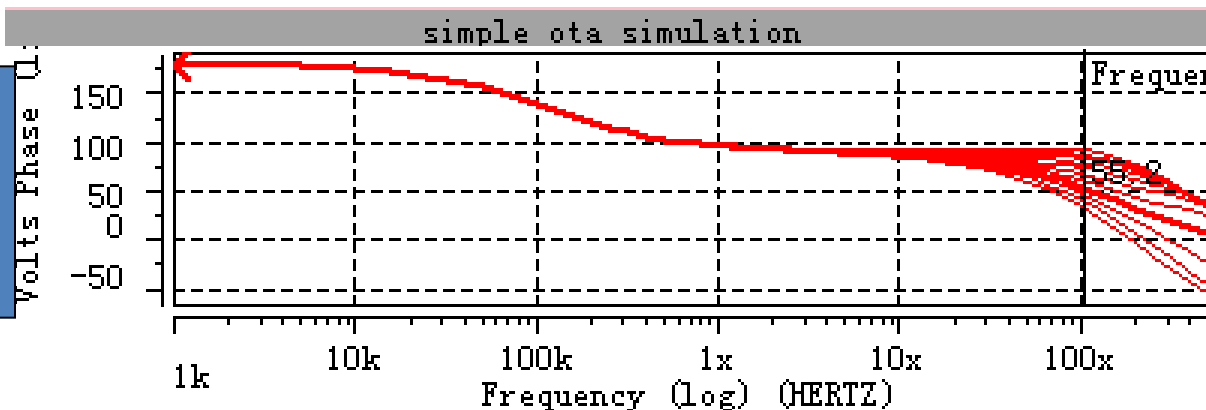


实例演示

R_z 增加到0.6k时，相位裕度增加到约55度，GBW约76MHz



R_z 增加到1时，相位裕度增加到约67度，GBW约103MHz



实例演示

*压摆率分析 (TRAN)

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe tran v(vo1) v(vo)  
.op  
*.dc v_vdc 2.45 2.51 0.001 sweep monte = 30  
.trans 0.1ns 1000ns  
*.ac dec 10 1k 500meg $sweep ccv 0 5p 1p  
*.noise v(vo) v_vac 20  
.para rzv=1k ccv=1p clv=1p $alfa=agauss(0,3,3)  
.inc 'NETLIST_PATH\ota.net'  
.end
```

在ota.net中将V_vac的定义换成:

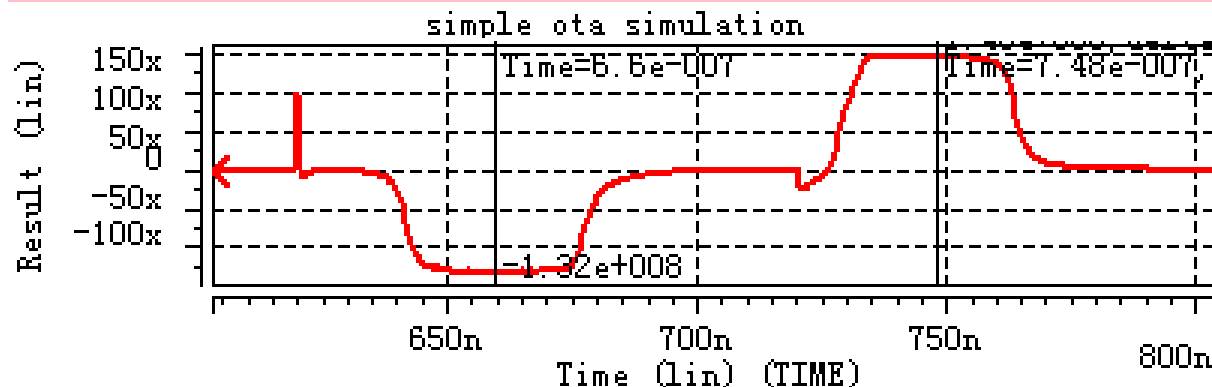
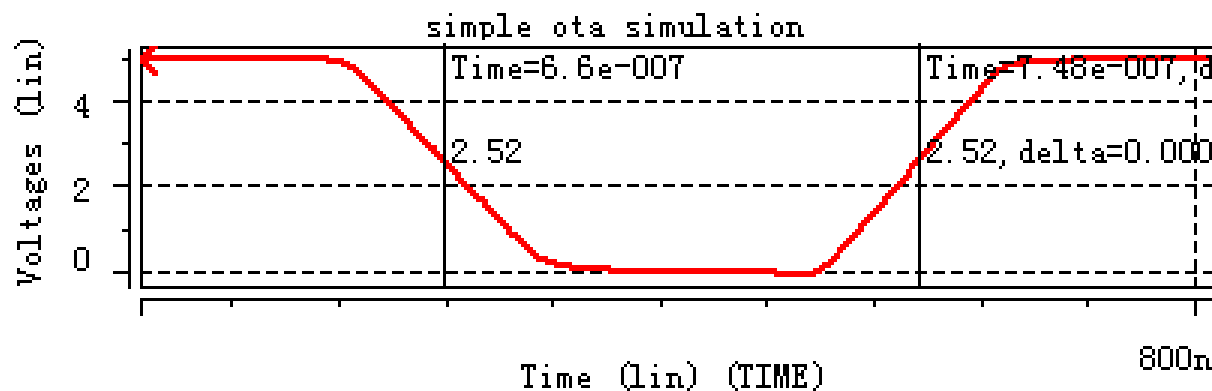
```
V_vpulse vin 0 PULSE 2 3 20ns 0.1n 0.1n  
100n 200n
```

实例演示

由右图可测得ota
的上升和下降压摆
率分别为146V/us
和132V/us

Problem: 在电路图
中如何分析上升和
下降压摆率？将结
果与仿真结果进行
对比

压摆率仿真结果



实例演示

*模型Corner仿真

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe dc v(vo)  
.op  
.dc v_vdc 2.45 2.51 0.0001 $sweep monte = 30  
*.trans 0.1ns 1000ns  
*.ac dec 10 1k 500meg $sweep ccv 0 5p 1p  
*.noise v(vo) v_vac 20  
.para rzv=1k ccv=1p clv=1p $alfa=agauss(0,3,3)  
.inc 'NETLIST_PATH\ota.net'  
.end
```

在ota.net中将V_vpulse的定义换回来:

```
V_Vac vin 0 DC 2.5V AC 1V 0
```

```
.alter  
.lib 'LIB_PATH\csmc.lib' ff  
.alter  
.lib 'LIB_PATH\csmc.lib' fs  
.alter  
.lib 'LIB_PATH\csmc.lib' sf  
.alter  
.lib 'LIB_PATH\csmc.lib' ss
```

实例演示

Ff时增益最小，ss时增益最大，查看csmc.lib，
对这种现象给出解释

$V_o = v_{dd}/2$ 分别对应于
 V_{vdc} 为：

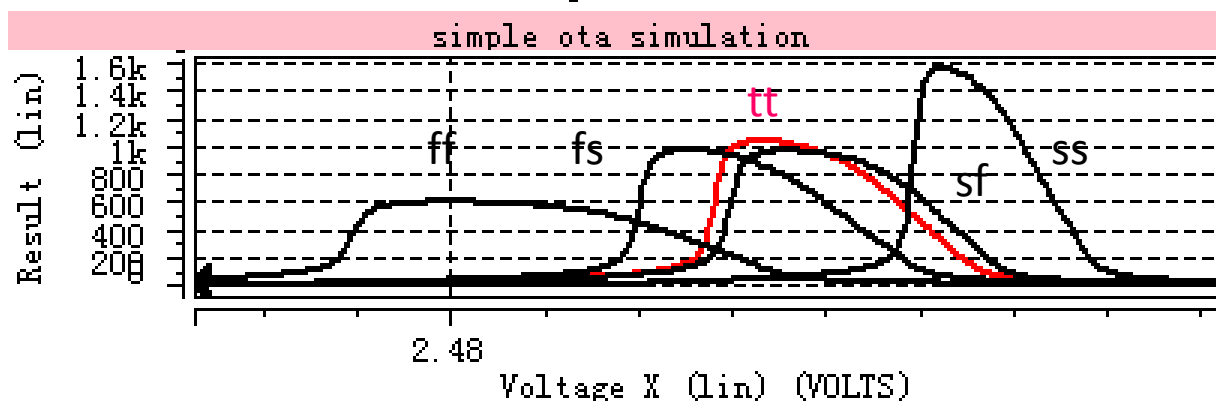
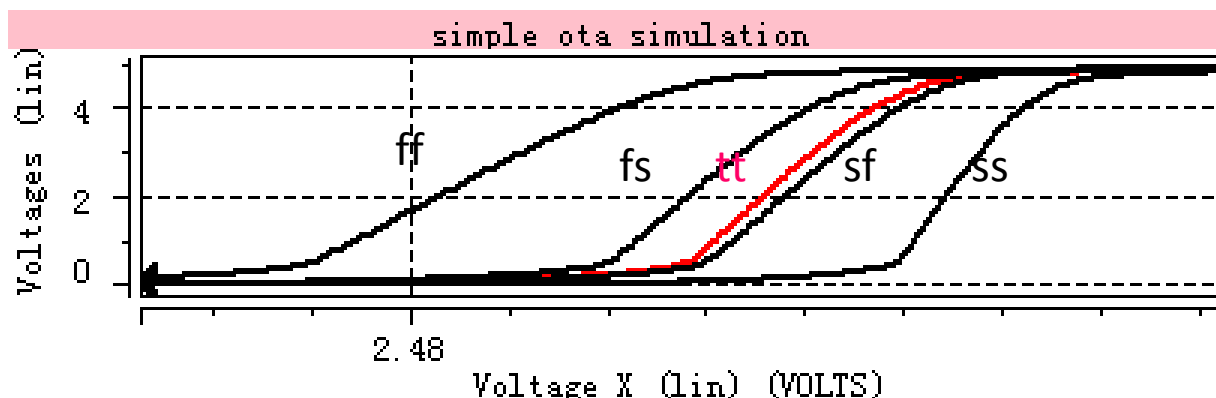
2.4876

2.4814

2.4861

2.4881

2.4912



实例演示

由仿真结果可测得：

	gain	GBW	Phase margin
tt	989	103MHz	67.2
ff	585	122MHz	75.9
fs	922	108MHz	64.8
sf	939	97.9MHz	71.2
ss	1.46k	87.4MHz	62.4

实例演示

*温度分析

Ota simulation

```
.prot  
.lib 'LIB_PATH\csmc.lib' tt  
.unprot  
.option post probe  
.probe dc v(vo)  
.op  
.dc v_vdc 2.45 2.51 0.0001 sweep temp 0 100 20  
*.trans 0.1ns 1000ns  
*.ac dec 10 1k 500meg $ temp 0 100 20  
*.noise v(vo) v_vac 20  
.para rzv=1k ccv=1p clv=1p $alfa=agauss(0,3,3)  
.inc 'NETLIST_PATH\ota.net'  
.end
```

实例演示

温度变化对系统失调和增益的影响

$V_o = v_{dd}/2$ 分别对应于
 V_{vdc} 为:

2.4882

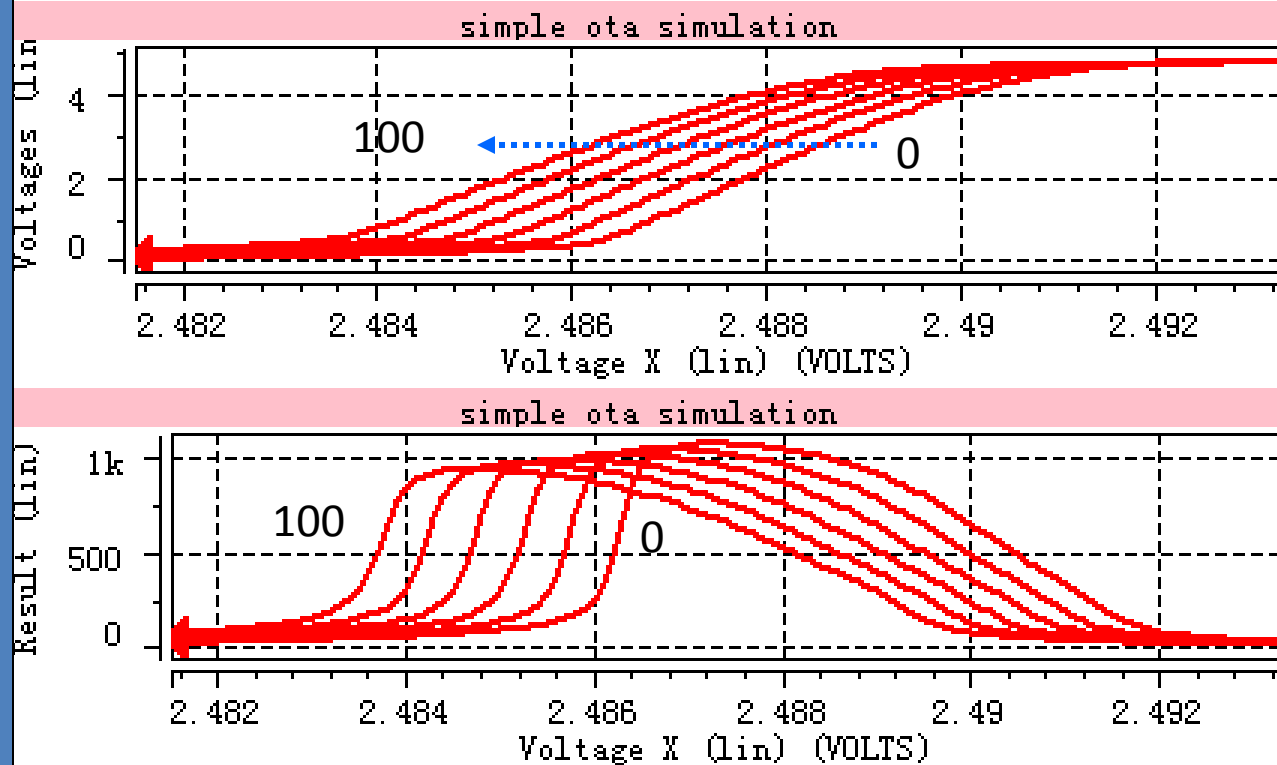
2.4877

2.4873

2.4868

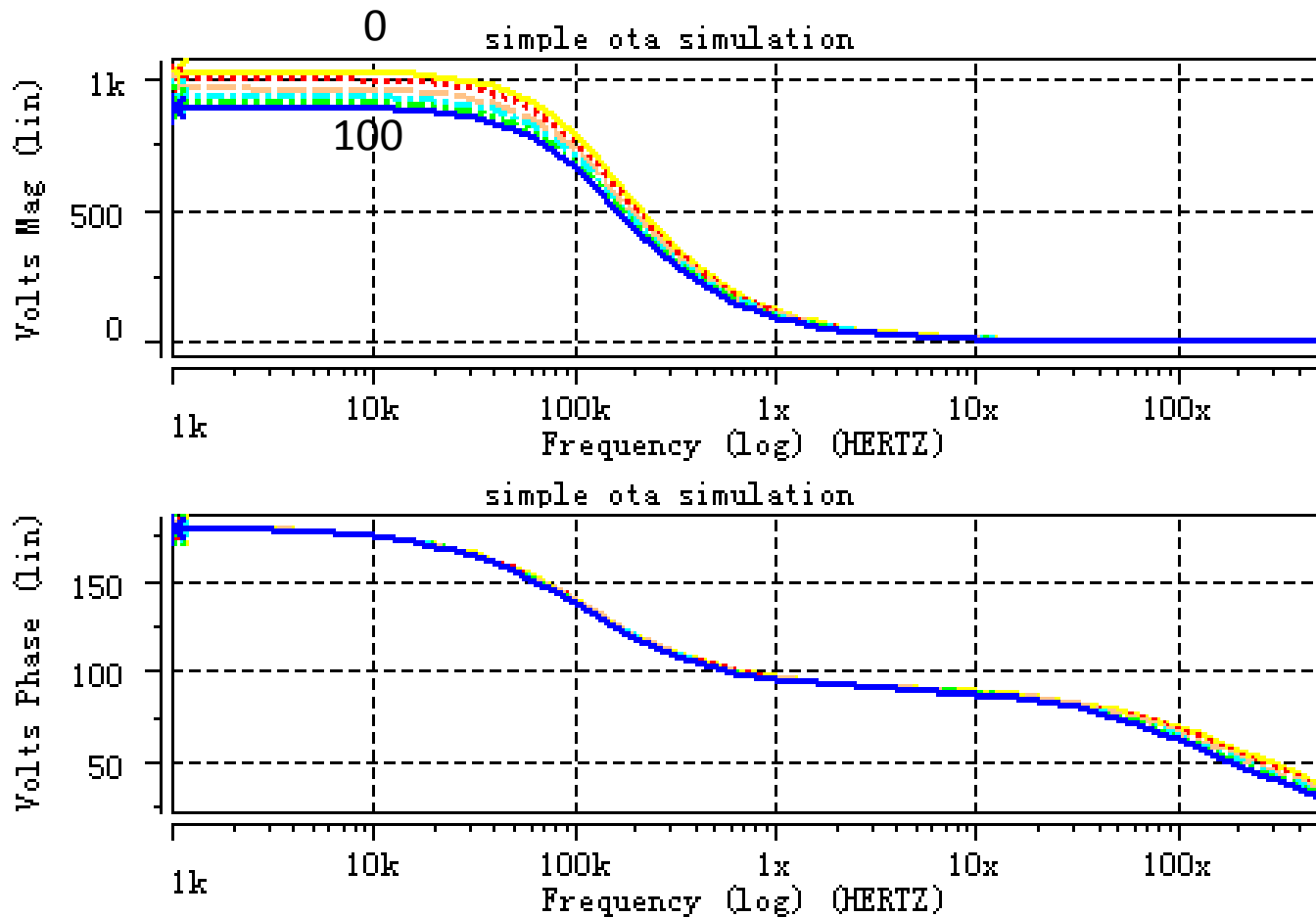
2.4863

2.4858



实例演示

不同温度下的AC分析结果



温度分析

由仿真结果可测得：

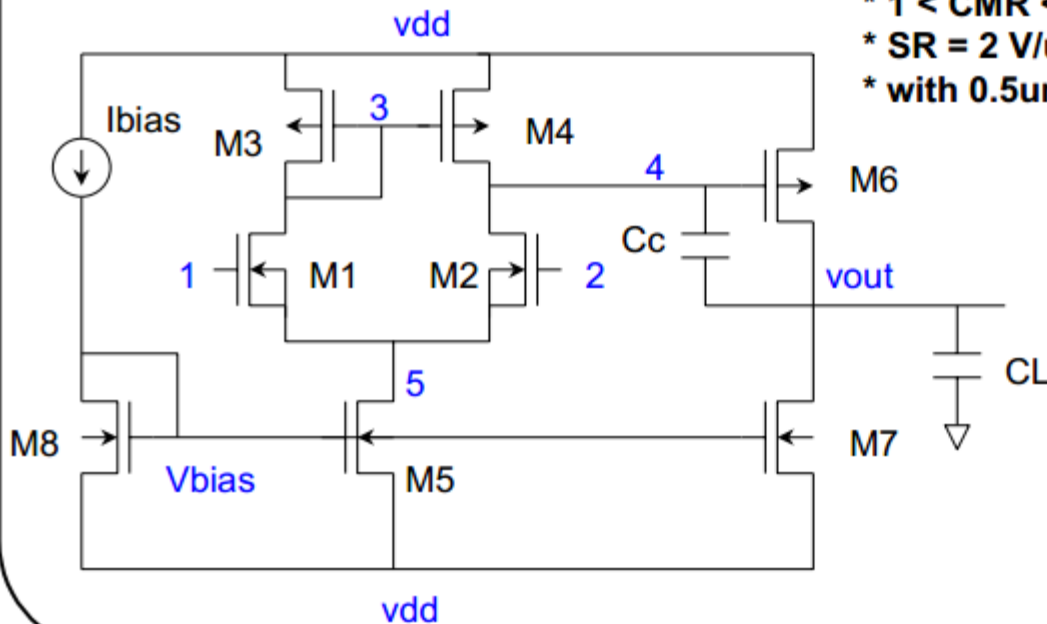
温度升高性能变差！

temp	gain	GBW	Phase margin
0	1.03k	109MHz	68.3
20	998	104MHz	67.4
40	962	98.3MHz	66.8
60	936	94.5MHz	66.2
80	913	90.7MHz	65.7
100	890	86.7MHz	65.4

• 4、两级运放设计

10. Application Demo

- * Target specification :
- * $CL = 4\text{pF}$, $A_v > 4000$,
- * $GB = 2\text{MHz}$
- * $1 < CMR < 4$, $0.8 < V_{out} < 4.2$
- * $SR = 2\text{ V/us}$, $P_{diss} < 10\text{mW}$,
- * with $0.5\mu\text{m}$ UMC process



实例演示

SPICE

10. Application Demo

(2). Netlist

***Two stage OP design**

.lib "umc05.lib" TYP

.options post nomod

.TEMP 27

*** Netlist information**

M1 3 1 5 0 nmos L=2u W=8u AS=18p AD=18p

+ PS=18u PD=18u

M2 4 2 5 0 nmos L=2u W=8u AS=18p AD=18p

+ PS=18u PD=18u

M3 3 3 vdd vdd pmos L=10u W=10u AS=12p AD=12p PS=16u PD=16u

M4 4 3 vdd vdd pmos L=10u W=10u AS=12p AD=12p PS=16u PD=16u

M5 5 vbias vss vss nmos L=2u W=7u AS=49p AD=49p PS=26u PD=26u

M6 vout 4 vdd vdd pmos L=2u W=70u AS=490p AD=490p PS=150u PD=150u

M7 vout vbias vss vss nmos L=2u W=130u AS=930p AD=930p

+ PS=260u PD=260u

M8 vbias vbias vss vss nmos L=2u W=7u AS=49p AD=49p PS=26u PD=26u

*** Feedback CAP**

Cc vout 4 0.44pF

Cl vout 0 4pF

Ibias vdd vbias 8.8u

*** Voltage sources**

vdd vdd 0 5v

vss vss 0 0v

实例演示

SPICE

10. Application Demo

(3). AC Frequency Analysis

```
vin2 2 0 2.5v  
vin1 1 0 DC 2.5v AC 1
```

```
*.OP
```

```
* AC Analysis function
```

```
.ac dec 10 10 100MEG
```

```
.probe ac vdb(vout)
```

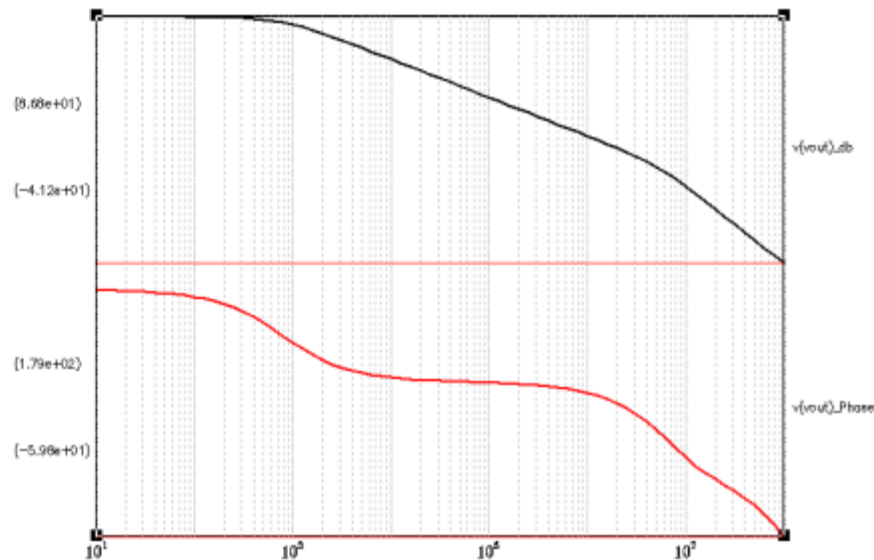
```
+ vp(vout) vdb(4) vp(4)
```

```
.meas ac Unit_gain
```

```
+ when vdb(vout)=0
```

```
.meas ac phase_mar
```

```
+ FIND vp(vout) when vdb(vout)=0
```



实例演示

SPICE

10. Application Demo

(4). Transient Analysis Slew Rate analysis

* Transient analysis section

```
M1 3 vout 5 0 nmos L=2u W=8u AS=18p AD=18p PS=18u PD=18u  
vin2 2 0 pulse(0v 5v 1n 1p 1p 600n 1200n)  
.tran 5n 2u  
.probe tran  
.save all
```

