HSPICE Introduction

Meng-Hung Shen (沈孟弘)
Chieh-Hung Chen (陳界弘)
Po-Chiun Huang

VER. 2012.7

Outline

- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Outline

- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Spice Overview

spice

spice
[spaIs; spais]
《源自拉丁文"商品, (尤指) 香料與藥"的意思》<<名詞>>
1 (C)[集合稱為(U)]佐料,香辣調味料
2 (U) [又作 a ~] 氣味,有幾分…的地方,意味,趣味,情趣
The fact gave [lent] ~ to the story.
該事實給那一則故事添加了趣味 [妙趣]
There was a ~ of malice in his words.
他的話中帶有幾分惡意
<<及物動詞>>
1 加調味料于…中<up>
2 a. 給…添加趣味<up>
b. [用…]給…調私[with]

- Simulation Program with Integrated Cir
  - Developed in 1970's at UC Berkeley
  - HSPICE is a robust industry standard
- Written in FORTRAN for punch-card ma
  - Circuits elements are called cards
  - Complete description is called a SPICE deck

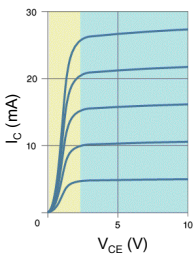
Idea

Netlist

```
* CMOS NAND Gate
.prot
.lib 'cic018.l' TT
.unprot
.option RELTOL=1e-7 POST=1
****   Netlist   ****
.subckt nand x y out
M1 out x vdd vdd P W=1u L=1u
M2 out y vdd vdd P W=1u L=1u
M3 out x 1   0   N W=1u L=1u
M4 1   y 0   0   N W=1u L=1u
.ends
```

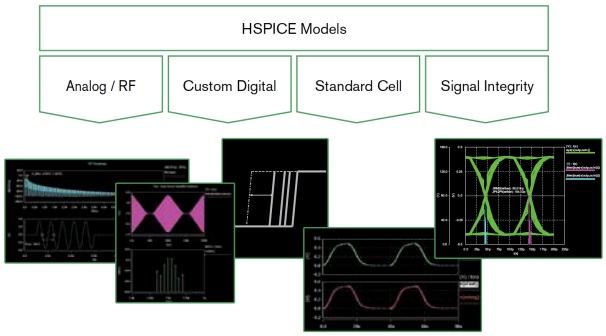
Simulation

Result

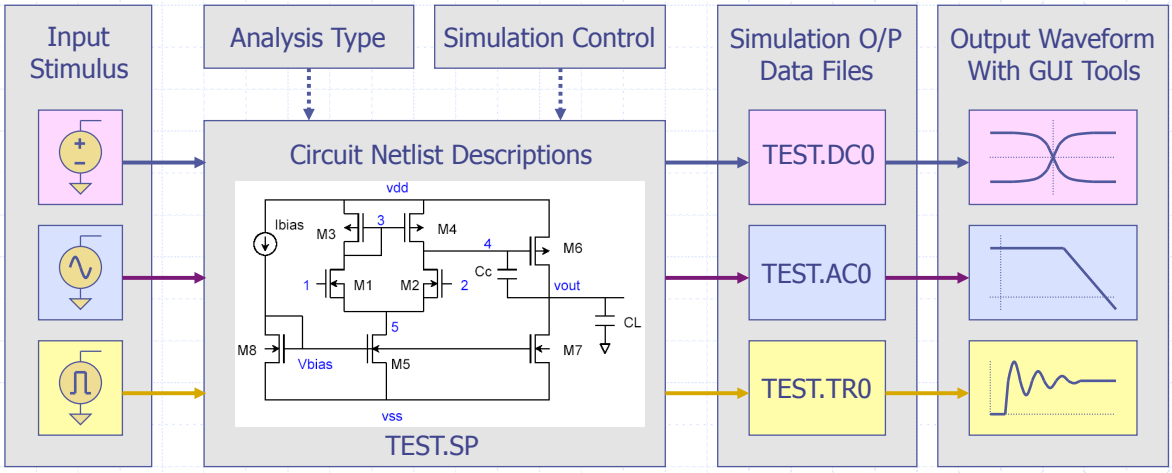


HSPICE Overview

- HSPICE is the gold standard for accurate circuit simulation
  - Now is an product in Synopsis Inc.
- Key features
  - Foundry-certified models
  - Behavioral modeling with Verilog-A
  - IC cell characterization
  - RF analyses functions
  - Design for Yield



General Descriptions of SPICE



Prepare for Simulation in Synopsys HSPICE

- Library File: cic018.l (CIC virtual 0.18um CMOS technology)
  - SPICE model for CIC 0.18um 1.8V virtual process
  - Different processes have different library files
    - Ex. "rf018.l" for TSMC 0.18um process
  - Different library files for different simulation tools
    - Can't use "cic018.l" in Cadence Spectre
- Simulation File: xxx.sp (write it on your own)
  - Devide into 3 parts
    - Setup
    - Main circuit description
    - Analyses
  - Can be from texteditor or schematic tool (Cadence Composer, …)

Can be separated as a single netlist file (xxx.spi). Then use .include 'xxx.spi' in the main simulation file.

CIC 0.18-um Model

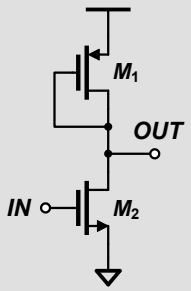
- Syntax: .lib "cic018.l" <TT/FF/SS/SF/FS>

```
* ********************************************
* *             CIC SPICE MODEL              *
* ********************************************
*
*    PROCESS :  CIC 0.18um 1.8V/3.3V 1P6M virtual Mixed Mode/RFCMOS Process
*    MODEL   :  BSIM3
*    DOC. NO.:
*    VERSION :  1.2
*    DATE    :  Aug. 15, 2005
*    HSPICE VERSION : For H98.2 & H98.4 only
*
*****************************************************************************
**History
*****************************************************************************
* V1.0  : Original
* V1.1  : change minute width for N_18, P_18, N_BPW_18, N_33, P_33, N_BPW_33
* V1.2  : updata sf conner data
*
*****************************************************************************
*       IN THIS MODEL LIB CONTAINS :                                        *
*                                                                           *
*       LIB   TT                                                            *
*       LIB   SS                                                            *
*       LIB   FF                                                            *
*       LIB   SF                                                            *
*       LIB   FS                                                            *
*                                                                           *
*       Example  :                                                          *
*       .lib 'cic018.l' tt                                                  *
*       XC net1 net2 MIMCAPM L=xxx W=xxx                                    *
*****************************************************************************
```

SPICE Netlist Overview

P_18 / N_18 為 cic018.l 定義之 1.8V model, 不同製程命名會有差異. 需查閱 foundry 給予的 model. 如 T 製程常為 pch /nch

```
* Diode-Connected Common Source Amplifier
.protect
.lib "cic018.l" TT
.unprotect
.option ABSTOL=1e-7 RELTOL=1e-7
+ POST=1 CAPTAB ACCURATE=1

****  Netlist  ****
M1  OUT OUT VDD vdd  P_18  W=1u L=0.18u M=1
M2  OUT IN   VSS  vss  N_18  W=1u L=0.18u M=2

****  Sources  ****
Vsupp  VDD VSS  DC=1.8
Vsupn  VSS  0    DC=0
Vin    IN   VSS  DC=0.7

****  Analysis  ****
.OP
.DC  VIN 0 1.8 0.1
.PRINT V(OUT)
.PLOT  V(OUT) V(IN)
.end
```



Setup

Main Circuit

Analyses

SPICE Netlist Overview

```
* Diode-Connected CS Amplifier
.protect
.lib "cic018.l" TT
.unprotect
.option ABSTOL=1e-7 RELTOL=1e-7
+ POST=1 CAPTAB ACCURATE=1

****  Netlist  ****
M1  OUT OUT VDD vdd  P_18  W=1u L=0.18u M=1
M2  OUT IN   VSS  vss  N_18  W=1u L=0.18u M=2

****  Sources  ****
Vsupp  VDD VSS  DC=1.8
Vsupn  VSS  0    DC=0
Vin    IN   VSS  DC=0.7

****  Analysis  ****
.OP
.DC  VIN 0 1.8 0.1
.PRINT V(OUT)
.PLOT  V(OUT) V(IN)
.end
```

模擬時第一行會被忽略

加進本file所吃的library檔

增加/修改 模擬條件

一行寫不下可加個"+"
在下一行繼續寫

大小寫沒有區分，且netlist中
前後順序沒有差別

Net 0 代表 global ground

在Netlist中註解
可用"*"來表示(僅限一行的開頭),
或者"$"(用在行中)

分析電路偏壓點

結尾一定要有"end"

Node Naming Conventions

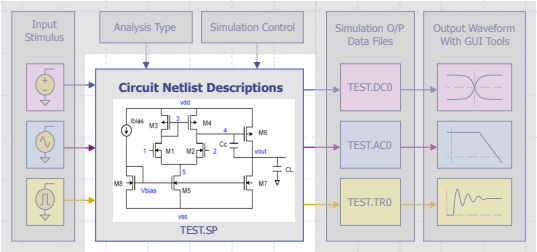
- Either numbers or words (e.g. C1, 99, NOD1)
- No differences between upper and lower case
- Trailing alphabetic character are ignored in node number(e.g. 1a=1b=1)
- 0, GND, GND!: always refer to the global ground
- All nodes are LOCAL node
  - Same name in different subcircuit represents different nodes
- Node names can be GLOBAL by a .GLOBAL statement
  - Example ".GLOBAL VDD VSS"

Numbers and Scale Factor

- Numbers can be
  - Integer / Floating point
  - Floating point with integer exponent
  - Integer or floating point with one scale factor
- Numbers can use:
  - Exponential format
  - Engineering key letter format
  - Not both
    - $1e^{-12}$ or 1p, but not $1e^{-6}$u

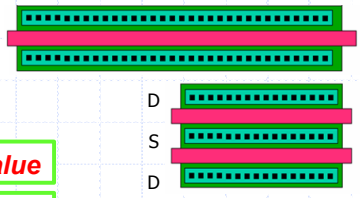
Prefix	Scale Factor	Multiplying Factor
Tera	T	1e+12
Giga	G	1e+9
Mega	MEG or X	1e+6
Kilo	K	1e+3
Milli	M	1e-3
Mikro	u	1e-6
Nano	n	1e-9
Pico	p	1e-12
Femto	f	1e-15
Atto	a	1e-18

Outline



- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Device Type



- Passive Devices
  - Resistor – R `Rxxx node1 node2 value`
  - Capacitor – C `Cxxx node1 node2 value`
  - Inductor – L `Lxxx node1 node2 value`
- Active Devices
  - BJT – Q `Qxxx C B E mname`
    C: Collector B: Base E: Emitter
    mname: model name
  - Diode – D
  - MOSFET – M `Mxxx D G S B mname w=val l=val m=val`
    D: Drain G: Gate S: Source B: Body
    mname: model name
    w: width l: length m: 並聯個數
- Other Devices
  - Behavioral – E, G, H, F, B
  - Transmission Lines – T, U, O
  - Subcircuit – X `Xxxx node 1 node2 … nodeN Xname`
    Xname: subcircuit name

Elements & Device Models – MOSFET

- Syntax:

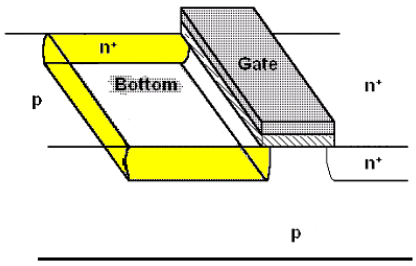
```
Mxxx  drain gate source body  mname  W=val  L=val
+ <AS=val> <AD=val> <PS=val> <PD=val>  <M=val>
```

- Example:

```
M1  3 4 5 VDD  P_18 W=10u L=1u
M2  OUT IN VSS VSS  N_18 W=2u L=0.18u M=2 AD=50p AS=50p
```

- How to decide the values of AS/AD?

CMOS Small-Signal Capacitances

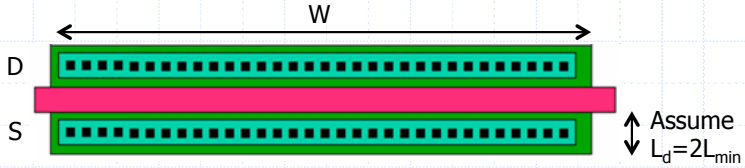


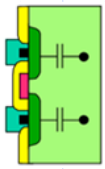

```
* CMOS INV Characterization
****  Define Parameters  ****
.param Wn = 1.0u
+ Wp = '2*Wn'
+ Lmin = 0.18u
+ Ld = '2*Lmin'

****   Netlist   ****
Mp1 out in  vdd vdd  P_18 W=Wp L=Lmin
+  AD='Wp*Ld'   AS='Wp*Ld'
+  PD='Wp+2*Ld' PS='Wp+2*Ld'
Mn1 out in  0   0    N_18 W=Wn L=Lmin
+  AD='Wn*Ld'   AS='Wn*Ld'
+  PD='Wn+2*Ld' PS='Wn+2*Ld'
```

- AS and AD are the areas of the source/drain junctions
- PS and PD are the source/drain perimeters excluding the sides adjacent to channel
- These values can be formulated by designer, or automatically calculated by advanced version by setting CALCACM with other device parameters.

Parasitics in Transistors

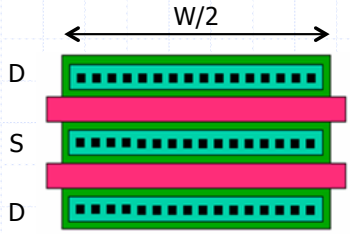
- Analog transistors with large W/L ratio



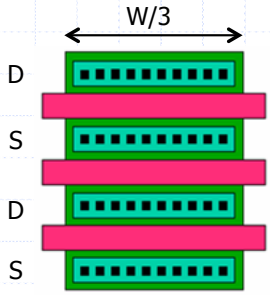


$$AS = AD = W \cdot L_d \qquad PS = PD = W + 2 \cdot L_d$$

- Use multi finger layout



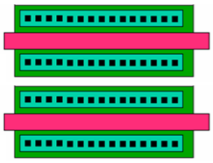
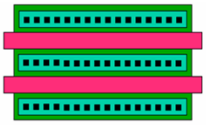
$$AS = \frac{W}{2} \cdot L_d \qquad AD = 2 \cdot \frac{W}{2} \cdot L_d$$



$$AS = \frac{2W}{3} \cdot L_d$$

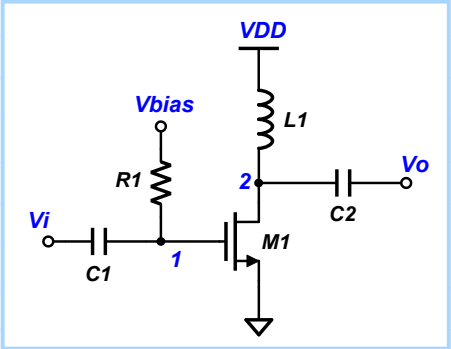
$$AD = \frac{2W}{3} \cdot L_d$$

Layout Dependent Transistor Description

	`M1 d g s b pmos W=Wp L=Lmin`
	`M1 d g s b pmos W='Wp/2' L=Lmin M=2`
	`M1 d g s b pmos W=Wp L=Lmin nf=2`

- "nf" means number of fingers (not supported in BSIM 3)
- For the generation before 0.18um, these three types are almost the same.
- For technologies using BSIM4, layout style should be considered.

Components Example




```
** Components
C1   Vi     1    0.1u
C2   2      Vo   0.5u
R1   Vbias  1    1k
L1   VDD    2    1n
M1   2 1 0 0    N_18   W=10u
+ L=0.18u
```


Subcircuits

- Subciruits for commonly-used circuit
- Use hierarchical structure to simplify complex connection
- Definition with .SUBCKT and .ENDS
- Use X<subcircuit_name> to call a subcircuit

- Subckt Example:

```
.subckt inv in out vdd vss
M1   out in vdd vdd   P_18   W=2u L=0.18u
M2   out in vss vss   N_18   W=1u L=0.18u
.ends inv
```

optional

```
X1   1 2 vdd vss   inv
X2   2 3 vdd vss   inv M=2
```

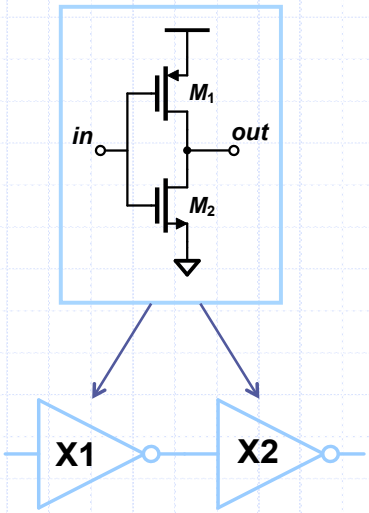
Example of Subcircuit and .GLOBAL

- Without .GLOBAL

```
.subckt inv in out vdd vss
M1   out in vdd vdd   P_18   W=2u L=0.18u
M2   out in vss vss   N_18   W=1u L=0.18u
.ends inv

X1   1 2 vdd vss   inv
X2   2 3 vdd vss   inv   M=2
```

- With .GLOBAL

```
.GLOBAL vdd vss
.subckt inv in out
M1   out in vdd vdd   P_18   W=2u L=0.18u
M2   out in vss vss   N_18   W=1u L=0.18u
.ends inv

X1   1 2   inv
X2   2 3   inv   M=2
```



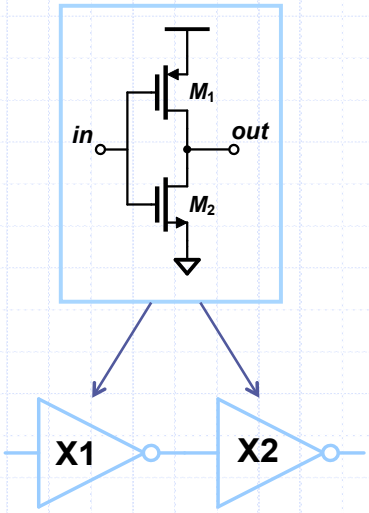
Example of Parameter Substitution

- Without parameter substitution

```
.subckt inv in out vdd vss WP=2u WN=1u
M1   out in vdd vdd   P_18   W=WP L=0.18u
M2   out in vss vss   N_18   W=WN L=0.18u
.ends inv

X1   1 2 vdd vss   inv
X2   2 3 vdd vss   inv   M=2
```

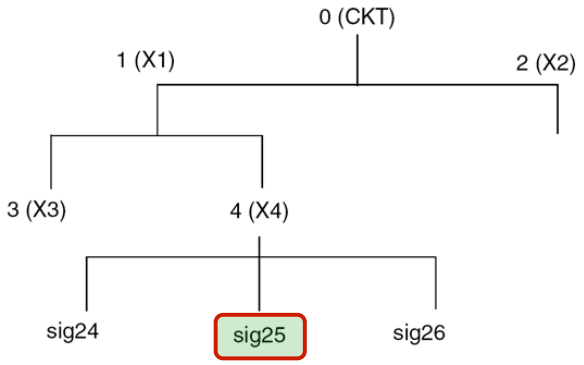
- With parameter substitution

```
.GLOBAL vdd vss
.subckt inv in out WP=2u WN=1u
M1   out in vdd vdd   P_18   W=WP L=0.18u
M2   out in vss vss   N_18   W=WN L=0.18u
.ends inv

X1   1 2   inv   WP=10u WN=5u
X2   2 3   inv   WP=20u WN=10u
```



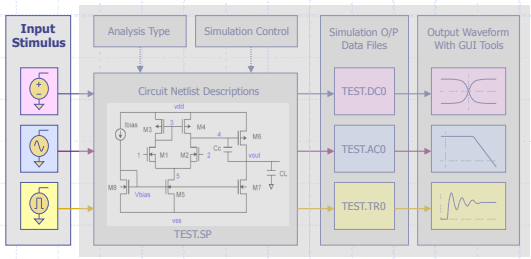
Subcircuit Hierarchy

- Access of nodes in subcircuits over (.) extension
- Concatenation of circuit path name with the node name



- Path name of the sig25 node in X4 subcircuit is: X1.X4.sig25
- E.g. can be used to print: .PRINT V(X1.X4.sig25)

Outline



- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Input Source Type

- Independent Source Elements
  - AC / DC
  - Transient Source (PULSE / SIN / PWL / LFSR / .....)
  - Others
- Dependent Source Elements
  - Voltage-Control Voltage Source (VCVS)
  - Voltage-Control Current Source (VCCS)
  - Current-Control Voltage Source (CCVS)
  - Current-Control Current Source (CCCS)

Independent Source Elements – DC/AC

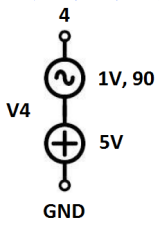
- Syntax:

```
Vxxx n+ n-  < <DC=>dcval> <tranfun> <AC=acmag, <acphase>>
Iyyy n+ n-  < <DC=>dcval> <tranfun> <AC=acmag, <acphase> <M=val>
```

- Example:

```
V1  1  0  DC=5V
V2  2  0  5V
I3  3  0  5mA
```

```
V4  4  0  DC=5V  AC=1V  90
V5  5  0  AC  1.0  180
```



  - DC sweep range is specified in .DC analysis statement
  - AC frequency sweep range is specified in .AC analysis statement

Independent Source Elements – Transient

- The most commonly used transient forms:
  - SIN – sinusoidal waveform
  - PULSE – pulse waveform
  - PWL – piece-wise linear waveform

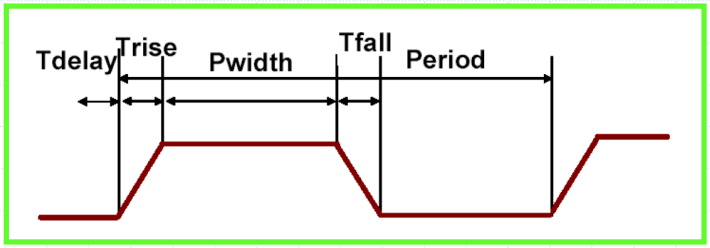
Independent Source Elements – PULSE

- Syntax:

```
PULSE ( V1 V2 < Tdelay Trise Tfall Pwidth Period > )
```

- Example:

```
Vin 1 0 PULSE ( 0V 5V 10ns 10ns 10ns 40ns 100ns )
```



Independent Source Elements – SIN

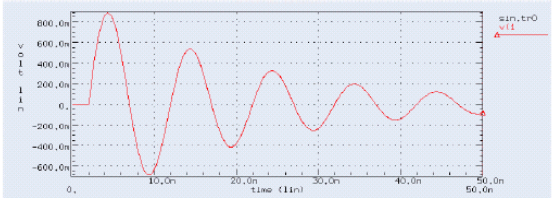
- Syntax:

SIN (Voffset Vacmag < Freq Tdelay Dfactor >)

Voffset + Vacmag* $e^{-(t-TD)*Dfactor}$ * sin(2π Freq(t-TD))

- Example:

Vin 3 0 SIN (0V 1V 100Meg 2ns 5e7)



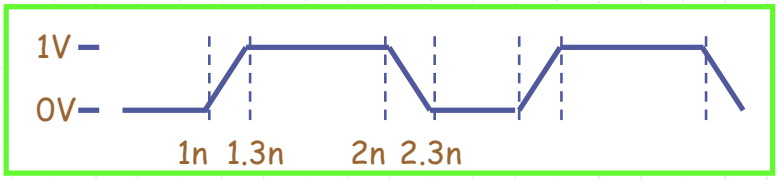
Independent Source Elements – PWL

- Syntax:

PWL (<t1 v1 t2 v2 … > <R=repeat time>)

- Value at intermediate is determined by linear interpolation.
- Example:

V3 10 5 PWL 0n 0V, 1n 0V, 1.3n 1V, 2n 1V, 2.3n 0V, R 0



Independent Source Elements – LFSR

- Syntax:

```
LFSR ( VL VH <Tdelay Trise Tfall Rate Seed <Taps> <Rout=val>> )
```

- Generate a pseudo-random binary sequence
  - Output is like: 1000010101110110001111100.....
  - An example [23, 22, 21, 20, 19, 7] denotes a 23-stage LFSR
- Example:

```
Vin  IN  0  LFSR ( 0V 1V 10ns 10ns 10ns 10Meg 1 [5, 2] rout=10)
```


Dependent Source Elements

- Four typical linear controlled sources:

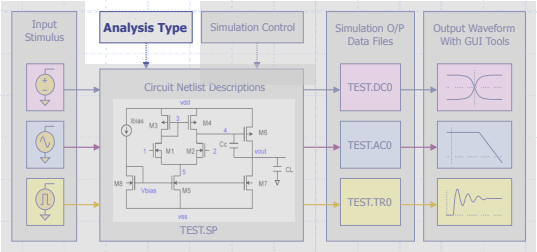
Voltage Controlled Voltage Sources (VCVS) --- E Elements
Voltage Controlled Current Sources (VCCS) --- G Elements
Current Controlled Voltage Sources (CCVS) --- H Elements
Current Controlled Current Sources (CCCS) --- F Elements

- Used for behavior modeling
  - Fasten execution time
- Example:

Controlled nodes Controlling nodes

E(name)	N+	N-	NC+	NC-	(Voltage Gain Value)
Eopamp	3	4	1	2	1e6
Ebuf	2	0	1	0	1.0

Outline



- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Common Analysis Types

- DC Operating Point Analysis
  - .OP
  - .IC
- DC Sweep and Small Signal Analysis
  - .DC
- AC Sweep and Small Signal Analysis
  - .AC
  - .NOISE
- Transient Analysis
  - .TRAN
  - .FOUR

Analysis Overview

- Definition of analysis type (OP, DC, transient, AC, …)
- At begin of analysis: Determination of DC operating point values for all nodes and sources:
  1. Calculation of all values
  2. Setting values specified in .NODESET and .IC statements
- Then: Iteratively searching of exact solution
  - End of search: error smaller than reltol/abstol
- At transient analysis: resulting DC operating point is initial estimate to solve the next timestep
- Initial estimates close to exact solution increase likelihood of convergent solution and lower simulation time

Analysis Types – OP Analysis

- Operating Point Analysis is the core of HSPICE
- .OP statement prints out following items in *.lis

  (1) Node Voltages (2) Source Currents (3) Power Dissipation

  (4) Semiconductors Device Currents, Conductance, Capacitance

Analysis Types – DC Analysis

- Sweep for Power Supply, Temp., Param., & Transfer Curves
- .DC Analysis: Syntax

```
.DC var1 start1 stop1 incr1 < var2 start2 stop2 incr2 >
.DC var1 start1 stop1 incr1 < SWEEP var2 DEC/OCT/LIN np start2 stop2 >
```

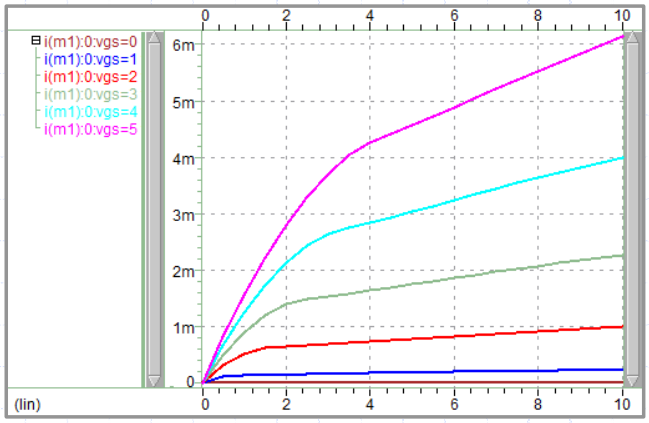
- Example:

```
.DC  VIN  0.25  5.0  0.25
.DC  VDS  0  10  0.5  VGS  0  5  1
.DC  xval  1k  10k  0.5k  SWEEP  TEMP  LIN   5  25  125
```


Analysis Types – DC Analysis Example

```
*** DC sweep example
m1 vd vg vs 0 nch l=1u w=10u
*** mos model
.model nch nmos level=2
*** voltage
vds vd vs
vgs vg vs
.dc vds  0  10  0.5  vgs  0  5  1
.probe dc i(m1)
.end
```

設太大會使得波形不平滑



Analysis Types – DC Small Signal Analysis

- .TF Analysis: Syntax

```
.TF  V(OUT)  VIN      **output-variable, input-source
**NOTE: Compute Transfer Function v(out)/vin , Rin ,Rout
```

- Example:

```
mn1 out in vss vss nch l=1u w=10u
mp1 out in vdd vdd pch l=1u w=20u
.global vdd vss
vdd vdd 0 5
vss vss 0 0
vin in 0
.tf v(out) vin
.end
```

```
****  small-signal transfer characteristics
   v(out)/vin =   -2.4582n
   input resistance at vin   =  1.000e+20
   output resistance at v(out)  =    1.1902k
```


Analysis Types – DC Small Signal Analysis

- .PZ Analysis: Syntax

```
.PZ  V(OUT)  VIN      **output-variable, input-source
**NOTE: Compute information about pole/zero analysis
```

- Example:

```
******   pole/zero analysis  tnom=  25.000 temp=  25.000
....
     poles (rad/sec)              poles ( hertz)
*********************************************************************
   real          imag            real          imag
  -1.0393x        0.            -165.4096k       0.
 -12.7888g        0.              -2.0354g       0.
.....
     zeros (rad/sec)              zeros ( hertz)
*********************************************************************
   real          imag            real          imag
 -12.7834g        0.              -2.0345g       0.
 -45.4697g    20.0122g            -7.2367g       3.1850g
```

Analysis Types – AC Analysis

- Syntax:

```
.AC  DEC/OCT/LIN  np  fstart  fstop
.AC DEC/OCT/LIN  np  fstart  fstop < SWEEP var start stop incr >
```

- Example:

```
.AC  DEC  10  1K  100MEG
```

Freq sweep 10 points per decade for 1KHz to 100MHz

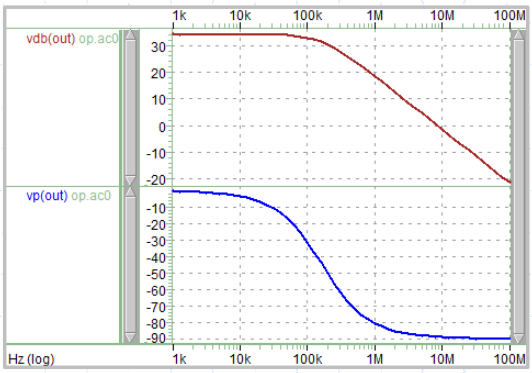
```
.AC  LIN  100  1  100Hz
```

Linear Sweep 100 points from 1Hz to 100Hz. 1Hz per AC point.

Analysis Types – AC Analysis Example

```
** differential amp
.subckt dopa va vb out
M1 net1 va   net2 0    nch w=25u l=1u
M2 out  vb   net2 0    nch w=25u l=1u
M3 net1 net1 vdd vdd pch w=8u    l=1u
M4 out  net1 vdd vdd pch w=8u    l=1u
M5 net2 net3 0    0    nch w=123u  l=1u
M6 net3 net3  vdd vdd pch w=1u  l=1u
M7 net3 net3  0    0    nch w=100u  l=1u
Cl out 0 5p
.ends
.model nch nmos level=2
.model pch pmos level=2
.GLOBAL vdd vss
vdd vdd 0 3.3
vss vss 0  0
```

```
xopa1 ina inb out dopa
vr1   inb 0    dc 1.65
vin  ina inb  ac 1  ** input signal
.ac dec 10 1k 100meg
.probe vdb(out) vp(out)
.option post
.end
```



Analysis Types – Transient Analysis

- Syntax:

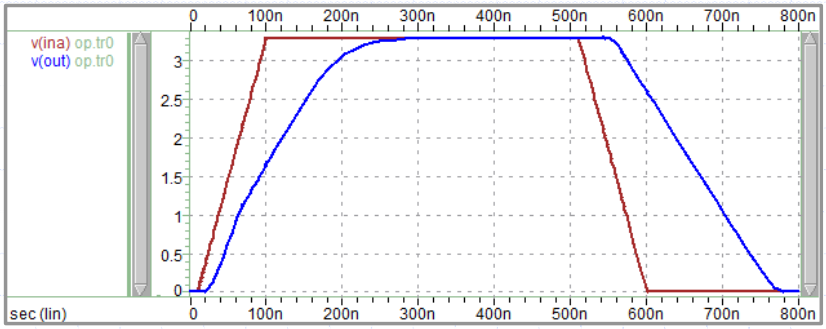
```
.TRAN  tincr1  tstop1 < tincr2  tstop2 ..... > < START=val>
.TRAN  tincr1  tstop1 < tincr2  tstop2 ..... > < START=val> UIC <SWEEP..>
```

- Example:

```
.TRAN   1NS  100NS
.TRAN  10NS  1US  UIC
.TRAN  10NS  1US  UIC  SWEEP  TEMP  -55  75  10  $ step=10
```


Analysis Types – Transient Analysis Example

```
vin2 ina 0 pwl (0 0 10n 0 100n  3.3 510n  3.3 600n 0)
.tran 10n 800n
.option post
.end
```



Analysis Types – Transient Analysis

- .FOUR Statement

```
.FOUR  100K  V(5)  V(7,8)          $  fundamental-freq , output-variable1,2,.....
```

Note: Determines DC and first Nine AC Harmonics & Reports THD (%)

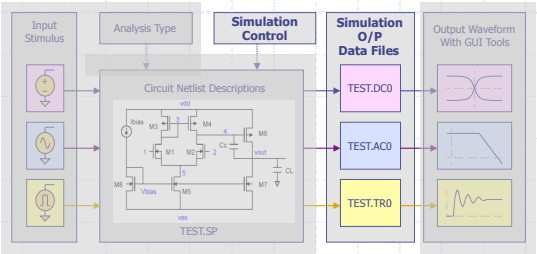
- .FFT Statement

```
.FFT v(1,2) np=1024 start=0.3m stop=0.5m freq=5K window=Kaiser alfa=2.5
```

Note1: Window Types : RECT, BLACK, HAMM, GAUSS, KAISER, HINN....

Note2: Determines DC and first Ten AC Harmonics & Reports THD (%)

Outline



- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

Output File Summary

Output File Type	Extension
Run Status	.st0
Output Listing (for debugging)	.lis
Measurement Results (AC/DC/TRAN)	.m*#
DC Analysis Results	.sw#
AC Analysis Results	.ac#
Transient Analysis Results	.tr#
FFT Analysis Results (from .fft statement)	.ft#

Contains result of: .print & .plot .op (operating point) .options (result)

Depends on .option post

#: index of simulation count, an integer

Simulation Output

- Output Commands
  - .PRINT – Print numeric analysis results in .lis file
  - .PLOT – Generates low resolution plot in .lis file
  - .PROBE – Allows save output variables only into the graphic data files
  - .MEAS – Print numeric results of measured specifications

Output Variable Examples

- DC & Transient Analysis:
  - Nodal Voltage Output : V(1), V(3,4), V(X3.5)
  - Current Output (Voltage Source) : I(VIN), I(X1.VSRC)
  - Current Output (Element Branches) : I2(R1), I1(M1), I4(X1.M3)
- AC Analysis:
  - AC : V(2), VI(3), VM(5,7), VDB(OUT), IP(9)
- Element Templates:
  - mn1[vth] → LV9(mn1)
  - mn1[gds] → LX8(mn1)
  - mn1[gm] → LX7(mn1)
  - mn1 gate caps (cgs+cgd+cgb) → LX18(mn1)

R : Real
I : Imaginary
M : Magnitude
P : Phase
DB : Decibels

Parametric Statement

- Algebraic Expressions for Output Statements:
  - .PRINT DC V(IN) V(OUT) PAR('V(OUT)/V(IN)')
  - .PROBE AC Gain=PAR('VDB(5)-VDB(2)') Phase=PAR('VP(5)-VP(2)')
- Other Algebraic Expressions:
  - Parameterization : .PARAM WN=5u LN=10u VDD=5.0V
  - Algebra : .PARAM X='Y+5'
  - Functions : .PARAM Gain(IN, OUT)='V(OUT)/V(IN)'
  - Algebra in Element : R1 1 0 r='ABS(V(1)/I(M1))+10'
- Built-In Functions:

sin(x) cos(x) tan(x) abs(x) sqrt(x)
log(x) log10(x) exp(x) db(x) min(x,y) max(x,y) ...

Ex: .print tran Gain=PAR('log (V(out)/V(in)) ')

Simulation Output – .PRINT/.PROBE /.PLOT

- Syntax:

```
.PRINT anatype ov1 <ov2 ov2...>
.PROBE anatype ov1 <ov2 ov2...>
.PLOT  anatype ov1 <ov2 ov2...>
```

- Example:

```
.PRINT  TRAN  V(4)  V(X3.3)
.PRINT  AC  VM(4,2)  VP(6)  VDB(3)
.PROBE  DC  V(2)  I(VSRC)  V(37,29)  I1(M7)
.PLOT  TRAN  V(5,3) (2,5)  V(8) I(VIN)
```


Simulation Output – .MEAS

RISE, FALL, and DELAY

- Syntax:

```
.MEASURE  DC|AC|TRAN  result_var  TRIG ... TARG ...
```

- ***result_var*** *: Name Given the Measured Value in HSPICE Output*
- ***TRIG ...*** *: TRIG trig_var VAL=trig_value <TD=time_delay> <CROSS=n>*
  + <RISE=r_n> <FALL=f_n|LAST>
- ***TRIG ...*** *: TRIG AT=value*
- ***TARG ...*** *: TARG targ_var VAL=targ_value <TD=time_delay>*
  + <CROSS=n|LAST> <RISE=r_n|LAST> <FALL=f_n|LAST>

- Example:

```
.meas  TRAN  tprop  trig  v(in)  val=2.5  rise=1  targ  v(out)  val=2.5  fall=1
```

Simulation Output – .MEAS

AVG , RMS, MIN, MAX, PP

- Syntax:

```
.MEASURE DC|AC|TRAN result FUNC out_var <FROM=val1> <TO=val2>
```

  - *result : Name Given the Measured Value in HSPICE Output*
  - *FUNC : AVG ----- Average MAX ----- Maximun PP ---- Peak-to-Peak*
    MIN ------ Minimum RMS ----- Root Mean Square
  - *out_var : Name of the Output Variable to be Measured*

- Example:

```
.meas TRAN minimal MIN V(1,2) from=25ns to=50ns
.meas TRAN tot_power AVG power from=25ns to=50ns
.meas TRAN rms_power RMS power
.meas DC Max_range PP V(out)
.meas AC Gainmax MAX VDB(out)
```


Simulation Output – .MEAS

FIND, WHEN

- Syntax:

```
.measure DC|AC|TRAN result WHEN ...
.measure DC|AC|TRAN result FIND out_var1 WHEN ...
.measure DC|AC|TRAN result_var FIND out_var1 AT=val
```

  - *result : Name Given the Measured Value in HSPICE Output*
  - *WHEN ... : WHEN out_var2=val|out_var3 <TD=time_delay>*
    + <CROSS=n|LAST> <RISE=r_n|LAST> <FALL=f_n|LAST>

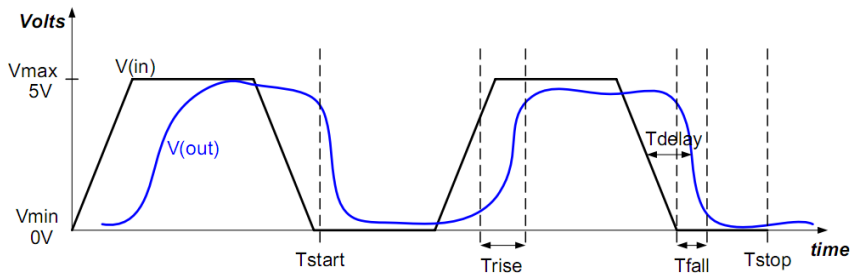
- Example:

```
.meas TRAN fifth WHEN v(osc_out)=2.5V rise=5
.meas TRAN result FIND v(out) WHEN v(in)=2.5V rise=1
.meas TRAN vmin FIND v(out) AT=30ns
```

Simulation Output – .MEAS Example

- Rise, Fall, and Delay Calculations:

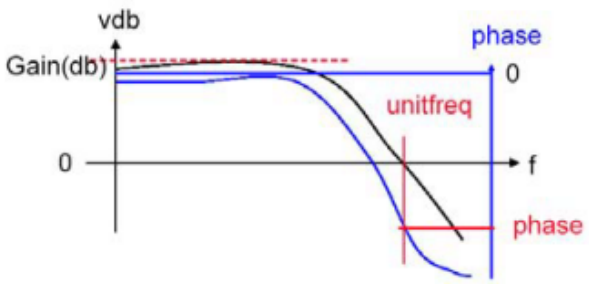
```
.meas TRAN Vmax MAX   v(out)  FROM=TDval            TO=Tstop
.meas TRAN Vmin MIN   v(out)  FROM =TDval           TO =Tstop
.meas TRAN Trise TRIG v(out)  VAL='Vmin+0.1*Vmax'   TD=TDval   RISE=1
+                TARG v(out)  VAl='0.9*Vmax'                   RISE=1
.meas TRAN Tfall TRIG v(out)  VAL='0.9*Vmax'        TD=TDval   FALL=2
+                TARG v(out)  VAl='Vmin+0.1*Vmax'              FALL=2
.meas TRAN Tdelay TRIG v(in)  VAL=2.5               TD=TDval   FALL=1
+                TARG v(out)  VAL=2.5                           FALL=2
```



Simulation Output – .MEAS Example

- Unit-gain Freq, Phase Margin, and DC gain(db/M):

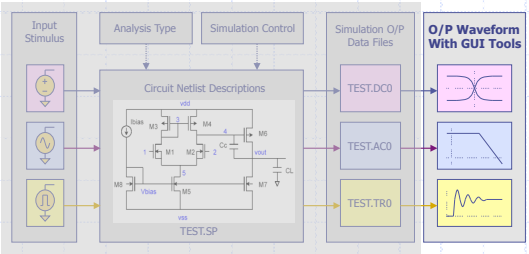
```
.meas  AC  unitfreq  WHEN  VDB(out)=0  <FALL=1>
.meas  AC  phase  FIND  VP(out)  WHEN  VDB(out)=0
.meas  AC  GainDB  MAX  VDB(out)
.meas  AC  phase_margin  param='180+phase'
```



Temperature & Control Options

- Temperature: .temp 100 (default is 25°C)
- Options: .option <options>
  - post: 儲存波形資料, 才能用waveform viewer開啟
  - captab: 儲存節點電容值
  - accurate=1: 增加模擬精確度
  - Nomod: 縮減.lis內的model parameter敘述
  - abstol: 當節點絕對值(V/I)小於此值即視為收斂
  - reltol: 當節點在iteration間的相對值小於此值即視為收斂
  - probe: 有此指令的話, 僅會儲存有probe的波形, 否則所有節點的電壓電流皆會被儲存
- Example: **.option post captab abstol=1e-7**

Outline



- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

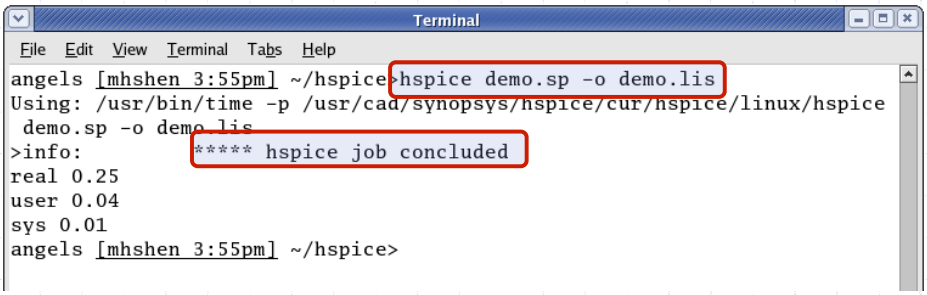
Simulation Platform

- 使用學校工作站 (Unix/Linux)
  - 在電機系4F工作站室執行
  - 遠端連線到工作站執行
    - 需用到GUI介面時, 需另外使用X-window System
  - HSPICE可在Linux或Unix機器跑, 但awaves只有Unix版本
  - Linux機器可用Custom WaveView/Cosmos Scope/nWave看圖
- 使用Windows版
  - 在個人電腦端執行
    - No license, 且大電路會拖慢電腦速度

Using HSPICE

預設輸出xxx.lis,
但也可以透過-o更改

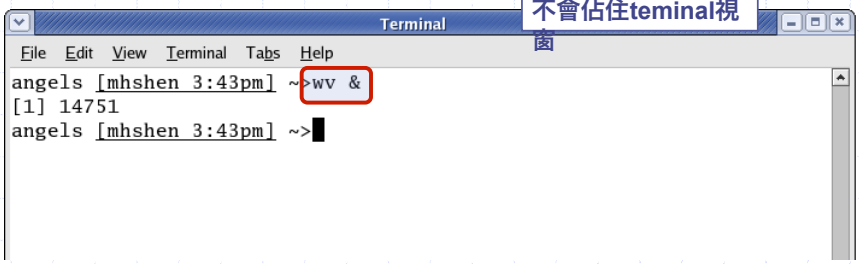
- Invoking hspice xxx.sp <–o xxx.lis>
  - HSPICE job concluded -> if circuit is correctly descirbed
  - HSPICE job aborded -> if circuit has something wrong
- If you get a "concluded" message, your simulation was successful and you are ready to view the results.



Viewing HSPICE Results

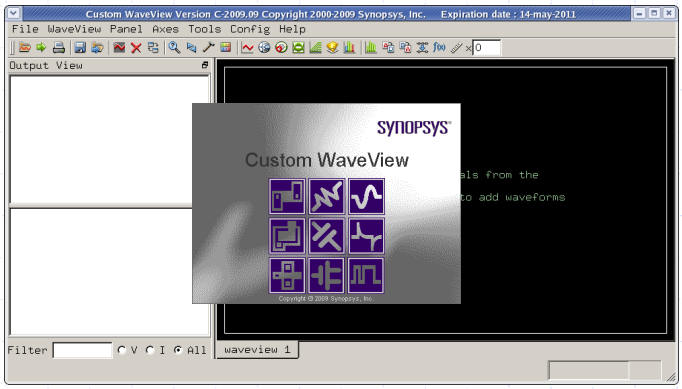
- To view the text results:
  - Use vim/gvim/joe editor: vim demo.lis
    - If job concluded: 包含detail電路資訊
    - If job aborted: 包含錯誤訊息以供debug
- To view the graphic results:
  - Use awaves: awaves & (only in Unix, already phase out)
  - Use Custom WaveView: wv &

代表背景執行,
不會佔住teminal視窗

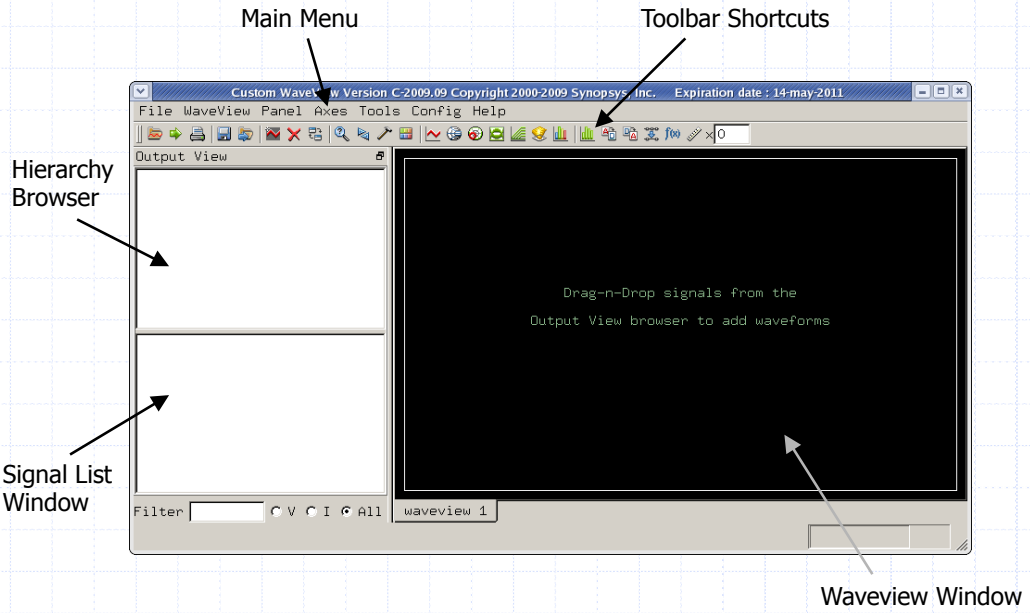


Result Browser – Custom WaveView

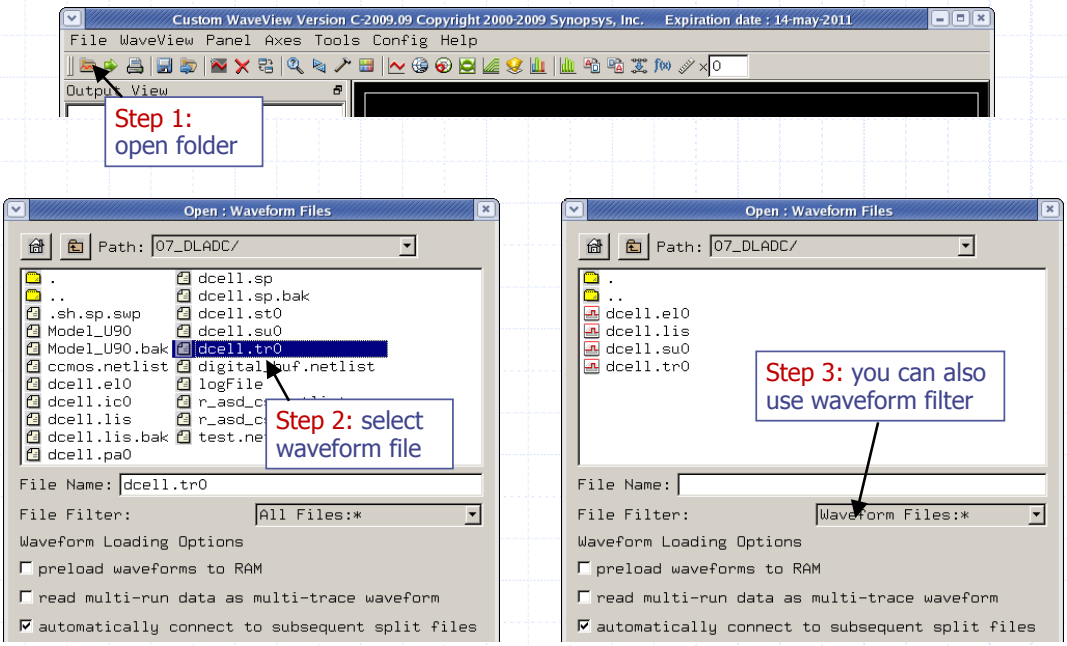
- A graphical waveform viewing and analysis program for analog/ mixed-signal IC design simulations
  - Formerly known as WaveView Analyzer
  - Developed by Sandwork Inc., and acquired by Synopsys in 2007
- Has been the default waveform browser of Synopsys AMS platform



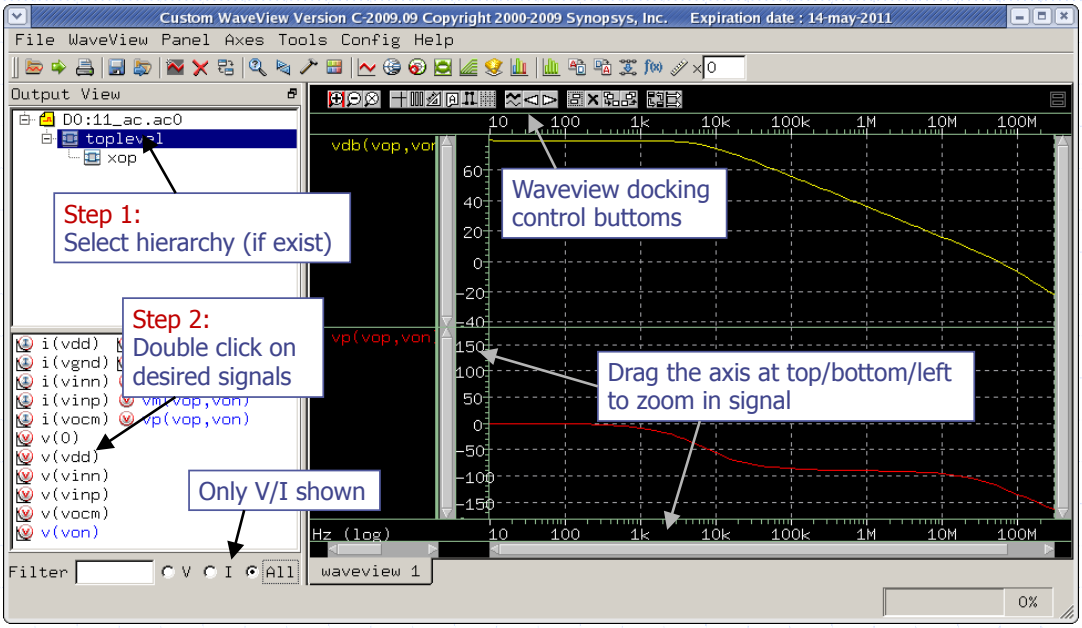
Result Browser – Custom WaveView



Result Browser – Custom WaveView

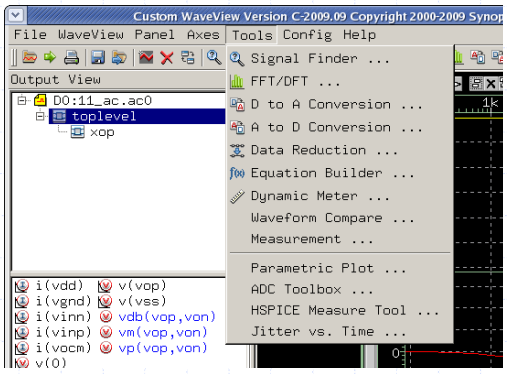


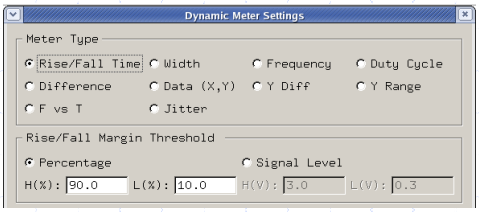
Result Browser – Custom WaveView

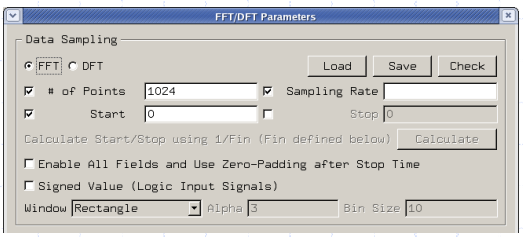


Result Browser – Custom WaveView

- There are many useful toolboxs
  - FFT/DFT ➡ evaluate spectrum performance fastly
  - Dynamic Meter ➡ measure rise/fall time, pulse width, …..
  - ADC Toolbox ➡ evaluate ADC spec directly
  - Jitter vs. Time tool ➡ PLL design







Outline

- General Descriptions
- Elements and Device Models
- Input Sources and Stimulus
- Analysis Types
- Simulation Output and Controls
- Simulation Steps and Graphic Tools
- Examples

HSPICE .SP Example – NAND Gate

```
* CMOS NAND Gate Characterization

.prot
.lib 'cic018.l' TT
.unprot
.option ABSTOL=1e-7 RELTOL=1e-7 POST=1 CAPTAB ACCURATE=1

****  Define Parameters  ****
.param vdd = 1.8
.param Wn = 1.0u
+ Wp = Wn
+ Lmin = 0.18u

****  Define Global  ****
.global vdd

****   Netlist   ****
.subckt nand x y out
Mp1 out x  vdd vdd  P_18  W=Wp L=Lmin
+   AD='2*Wp*Lmin'        AS='2*Wp*Lmin'
Mp2 out y  vdd vdd  P_18  W=Wp L=Lmin
+   AD='2*Wp*Lmin'        AS='2*Wp*Lmin'
Mn1 out x  1   0    N_18  W=Wn L=Lmin
+   AD='2*Wn*Lmin'        AS='2*Wn*Lmin'
Mn2 1   y  0   0    N_18  W=Wn L=Lmin
+   AD='2*Wn*Lmin'        AS='2*Wn*Lmin'
.ends
```

Define parameters with .param

Can define parameter based on other parameters or expressions.

Use '+' to continue long lines.

Subcircuits are SPICE's way of defining modules repeated in your design.

First line in a SPICE deck is always a comment (with or without *).

Here's how you can do arithmetic on your parameters.

HSPICE .SP Example – NAND Gate

```
.subckt inverter in out strength=Wn
Mn1 out in 0   0    N_18  W=strength L=Lmin
+   AD='2*strength*Lmin'  AS='2*strength*Lmin'
Mp1 out in vdd vdd  P_18  W=strength L=Lmin
+   AD='2*strength*Lmin'  AS='2*strength*Lmin'
.ends

.subckt buffer in out strength=Wn
Xinv1 in  mid inverter strength=strength
Xinv2 mid out inverter strength=strength
.ends

Xbuff1 a_source a buffer strength='Wn'
Xbuff2 b_source b buffer strength='Wn'
Xnand1 a b out nand

Cl out 0 0.1p  $ Load capacitance

****   Sources   ****
vdd vdd 0 vdd
Va  a_source 0 pulse (0 vdd 20n 0.1n 0.1n 40n 60n)
Vb  b_source 0 pulse (0 vdd 10n 0.1n 0.1n 20n 40n)
```

You can even parametrize values in your module and select these values when you instantiate your module.

Here is the subcircuit parameters and their default values

Instantiate modules like so.

First vdd is the voltage source.
Second vdd is the node.
Third vdd is the global parameter (=1.8 V) .

HSPICE .SP Example – NAND Gate

```
****   Analysis  ****
.tran 50p 80n sweep wn lin 3 1.0u 3.0u
.meas tran tplh_b1ar trig v(a) td=20ns val='vdd/2' cross=
+ targ v(out) td=20ns val='vdd/2' cross=1
.meas tran tplh_a1bf trig v(b) td=30ns val='vdd/2' cross=
+ targ v(out) td=30ns val='vdd/2' cross=1
.meas tran tplh_a1br trig v(b) td=50ns val='vdd/2' cross=1
.meas tran avgpower AVG power from=1ns to=80ns
.meas tran peakpower MAX power from=1ns to=80ns


****   Alter Parameters   ****
.alter case 2: Wp=2Wn
.param Wp = '2*Wn'

.alter case 3: use fast corner
.del lib TT
.lib 'cic018.l' FF

.alter case 4: increase Vdd by 10%
.param vdd = '1.8*1.1'

.alter case 5: change temperature
.TEMP 100
.end
```

We can sweep parametrized values using 'sweep'.

Here 'lin' means linear. Alternatives are 'dec' and 'oct'.

Temperature may be swept by using temp instead of sweep par

Measure propagation delays accurately using the '.meas' statement.

Outputs are written to .mtn files, where n=0,1,… (alteration number)

Here we trigger when the voltage at node 'a' crosses vdd/2, and measure the time until the output crosses vdd/2. Td = 20ns makes sure were on the right part of the waveform.

.ALTER statements allows us to modify the circuit and run again.

They must be before the final .end statement.

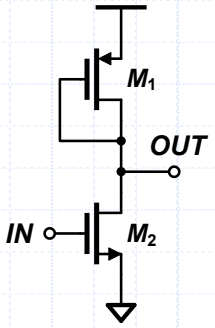
Note: ALTER blocks are incremental!!!

We can even change the models.

Temperature may be altered as well.

HSPICE Example – Diode-Connected CSA

```
* Diode-Connected CS Amplifier
.protect
.lib "cic018.l" TT
.unprotect
.option ABSTOL=1e-7 RELTOL=1e-7 POST=1 CAPTAB ACCURATE=1

****   Netlist   ****
M1  OUT OUT VDD VDD  P_18  W=1u L=0.18u M=1
M2  OUT IN  VSS VSS  N_18  W=1u L=0.18u M=2

****   Sources   ****
Vsupp  VDD  VSS  DC=1.8   $Voltage source of power supply
Vsupn  VSS  0    DC=0
Vin    IN   VSS  DC=0.7

****   Analysis  ****
.OP
.DC  IN 0 1.8 0.1
.PRINT V(OUT)
.PLOT  V(OUT) V(IN)
.end
```



HSPICE .lis Example – I

.PRINT output

```
 * HSPICE heading and input netlist omitted

 ******
 * diode-connected cs amplifier
  ******  dc transfer curves              tnom=  25.000 temp=  25.000
 ******
x
        volt    voltage
                    out
    0.            1.7117
  100.00000m      1.6016
  200.00000m      1.4818
  300.00000m      1.3622
  400.00000m      1.2452
  500.00000m      1.1008
  600.00000m    890.3282m
  700.00000m    637.3656m
  800.00000m    396.7913m
  900.00000m    240.8799m
    1.00000     174.0191m
    1.10000     139.0655m
    1.20000     117.2608m
    1.30000     102.3319m
    1.40000      91.5053m
    1.50000      83.3408m
    1.60000      77.0088m
    1.70000      71.9951m
    1.80000      67.9627m
```

HSPICE .lis Example – II

.PLOT output

```
legend:

  a: v(out)
  b: v(in)

      volt        v(out)
 (ab      )        0.          500.0000m       1.0000        1.5000       2.0000
                     +            +            +             +            +
   0.          1.712 -b------+------+------+------+------+------+-----a+------+-
 100.0000m     1.602  +  b   +      +      +      +      +      +  a   +      +
 200.0000m     1.482  +     b+      +      +      +      +     a+      +      +
 300.0000m     1.362  +      +b     +      +      +      +  a   +      +      +
 400.0000m     1.245  +      +   b  +      +      +      a      +      +      +
 500.0000m     1.101  +      +      b      +      +  a   +      +      +      +
 600.0000m   890.328m +      +      +  b   +   a  +      +      +      +      +
 700.0000m   637.366m +      +      +   a b+      +      +      +      +      +
 800.0000m   396.791m +      +   a  +      +b     +      +      +      +      +
 900.0000m   240.880m +      a      +      +   b  +      +      +      +      +
   1.0000    174.019m-+----a-+------+------+------b------+------+------+------+-
   1.1000    139.065m +   a  +      +      +      +  b   +      +      +      +
   1.2000    117.261m +  a   +      +      +      +     b+      +      +      +
   1.3000    102.332m +  a   +      +      +      +      +b     +      +      +
   1.4000     91.505m +  a   +      +      +      +      +   b  +      +      +
   1.5000     83.341m + a    +      +      +      +      +      b      +      +
   1.6000     77.009m + a    +      +      +      +      +      +  b   +      +
   1.7000     71.995m + a    +      +      +      +      +      +     b+      +
   1.8000     67.963m + a    +      +      +      +      +      +      +b     +
                      +            +            +             +            +
```


HSPICE .lis Example – III

Operating Information

```
******  operating point information      tnom=  25.000 temp=  25.000
******
***** operating point status is all       simulation time is     0.
   node    =voltage      node    =voltage      node    =voltage

+0:in      = 700.0000m 0:out     = 637.3656m 0:vdd     =   1.8000
+0:vss     =   0.
```

Cap Table

```
maximum nodal capacitance=   1.088E-14         on node     0:vss

    nodal  capacitance table

    node    =    cap      node    =    cap      node    =    cap

+0:in      =   3.7200f 0:out     =   5.9908f 0:vdd     =   5.2525f
+0:vss     =  10.8804f
```

Source Table

```
****  voltage sources

subckt
element  0:vsupp    0:vsupn    0:vin
 volts      1.8000     0.       700.0000m
 current  -93.3365u   71.2185a    0.
 power    168.0057u    0.          0.

    total voltage source power dissipation=  168.0057u       watts
```

HSPICE .lis Example – IV

Element Table

```
**** mosfets

 subckt
 element  0:m1        0:m2
 model    0:p_18.1    0:n_18.1
 region     Saturati    Saturati
  id       -93.3365u    93.3365u
  ibs      1.702e-20 -2.782e-20
  ibd      101.8856a -182.6666a
  vgs       -1.1626   700.0000m
  vds       -1.1626   637.3656m
  vbs        0.          0.
  vth     -527.5957m  498.4979m
  vdsat   -574.0249m  196.5956m
  vod     -635.0388m  201.5021m
  beta     497.7671u    3.9853m
  gam eff  557.0832m  507.4486m
  gm       207.4457u  664.6417u
  gds       15.8666u   35.9343u
  gmb       69.0015u   91.5388u
  cdtot      1.1888f    2.9023f
  cgtot      1.8997f    3.7200f
  cstot      2.8414f    5.5505f
  cbtot      2.4111f    5.3299f
  cgs        1.4259f    2.6662f
  cgd      354.0700a  720.7667a
```

- What is the difference between Vod and Vdsat?
- In the list file, Vgs-Vth=Vod
  - For long channel device
    - Vod = Vdsat
  - For short channel device
    - |Vod| > |Vdsat| (usually)
    - Due to second order effect
- If you use Vod to design circuit, you will get more margin to ensure the mos in sat region

HSPICE .lis Example – V

Accounting Stats

```
***** job concluded
1****** HSPICE -- A-2008.03 32-BIT (Feb 26 2008) winnt ******
 ******
 * diode-connected cs amplifier
  ******  job statistics summary           tnom=  25.000 temp=  25.000
 ******

           total memory used        194 kbytes

  # nodes =       9 # elements=       6
  # diodes=       0 # bjts    =       0 # jfets   =      0 # mosfets =      2
  # va device =       0

    analysis           time    # points   tot. iter  conv.iter

    op point           0.01           1           8
    dc sweep           0.01          19          76
    readin             0.08
    errchk             0.02
    setup              0.01
    output             0.00
           total cpu time          0.18 seconds
           total elapsed time         1 seconds
               job started at  10:55:14 03/02/2009
               job ended   at  10:55:15 03/02/2009
```

Reference

- HSPICE training course material from
  "國研院國家晶片系統設計中心(CIC)"
- HSPICE manual
- HSPICE tutorial by Y. Mortazavi, M.-F. Chang

This material is for educational purpose only.
Data is for reference without guarantee.
All copyrights belong to their owner.