Digital Integrated Circuits

YuZhuo Fu

contact:fuyuzhuo@ic.sjtu.edu.cn

Office location：417 room

WeiDianZi building,No 800 DongChuan road,MinHang Campus

3.CMOS Inverter

Introduction to CMOS VLSI Design

SPICE Simulation

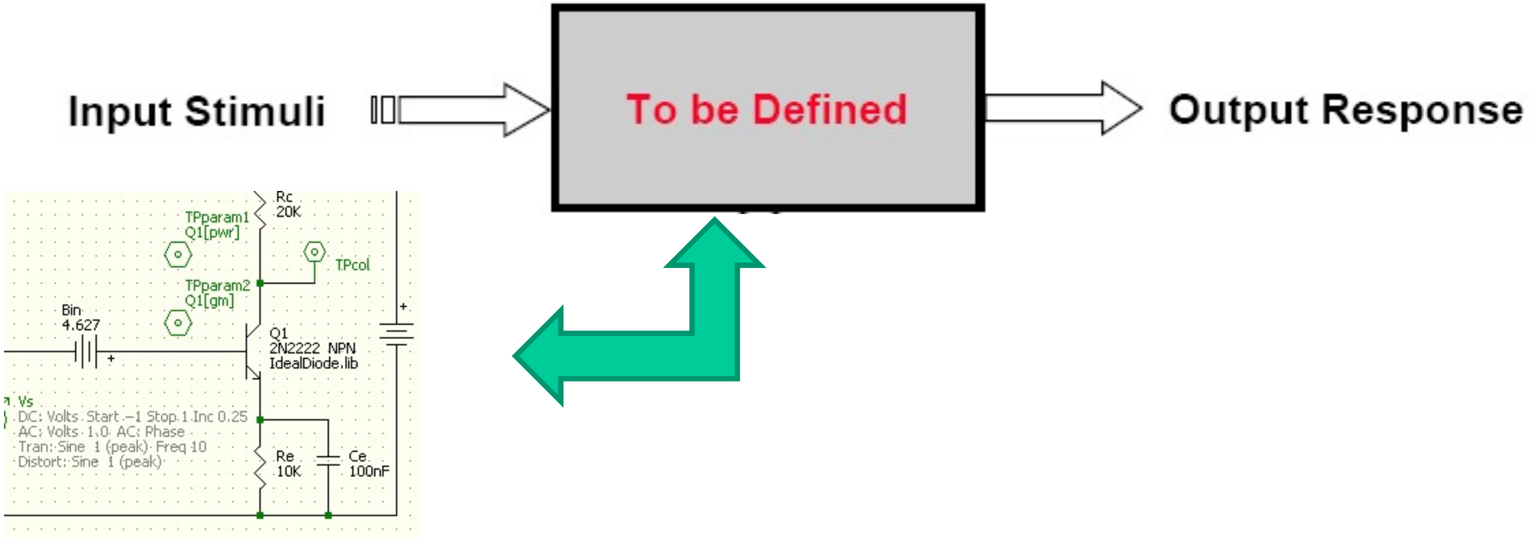
Simulation Program with Integrated Circuit Emphasis

contents

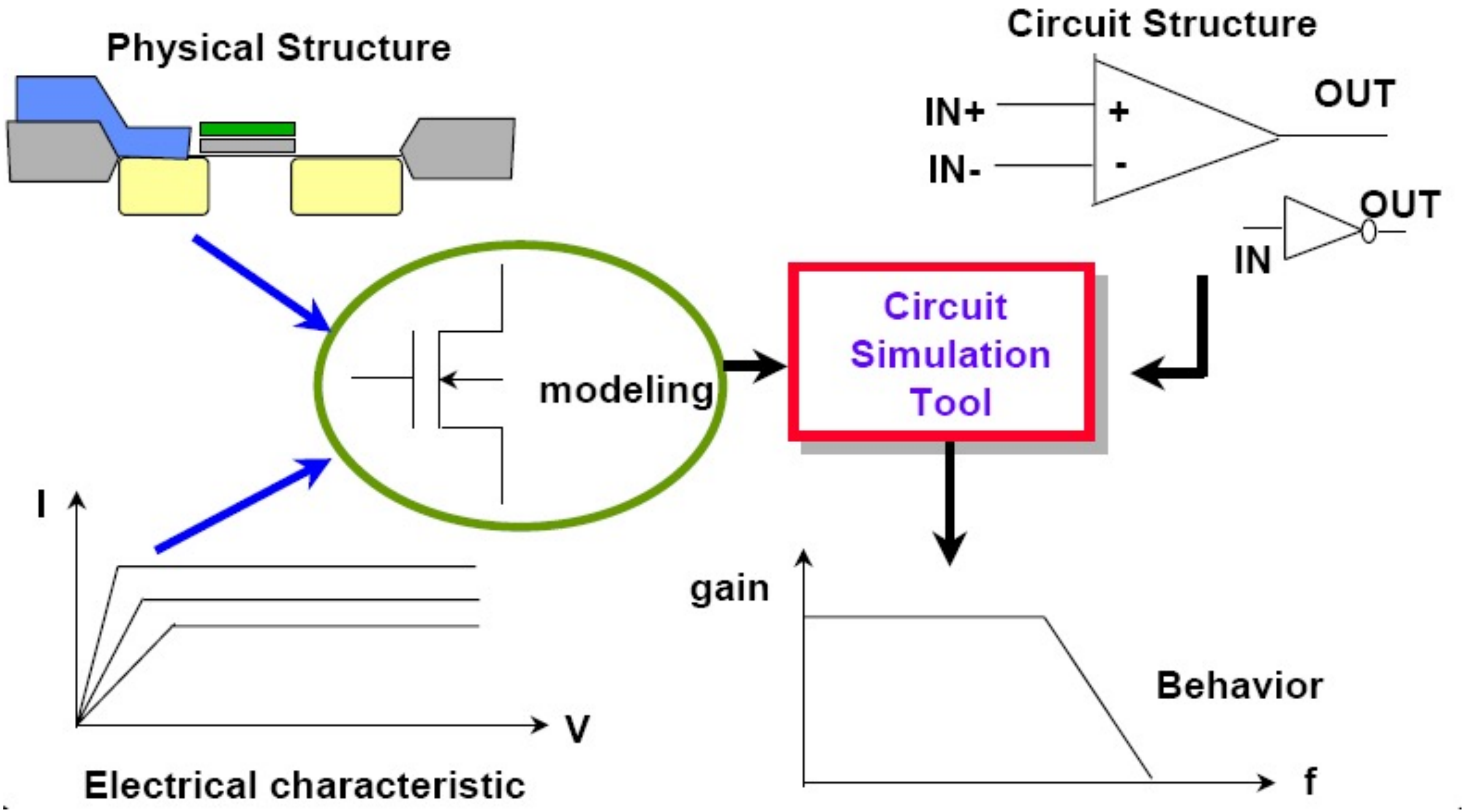
- **SPICE Overview**
- Simulation Input and Controls
- Sources and Stimuli
- Analysis Types
- Simulation Output and Controls
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

Circuit Design Background

- Circuit/System Design:
  - A procedure to construct a physical structure which is based on a set of basic component, and the constructed structure will provide a desired function at specified time/ time interval under a given working condition.



Circuit Simulation Background



Overview of SPICE

- SPICE
  - Numerical Approach to Circuit Simulation
  - 1970’s Developed by UCB
  - Widely Adopted, Become *De Facto Standard*
  - Circuit Node/Connections Define a Matrix
- Rely on Sub-Models for Behavior of Various Circuit Elements
  - Simple (e.g. Resistor)
  - Complex (e.g. MOSFET)
- Written in FORTRAN for punch-card machines
  - Circuits elements are called *cards*
  - Complete description is called a SPICE *deck*

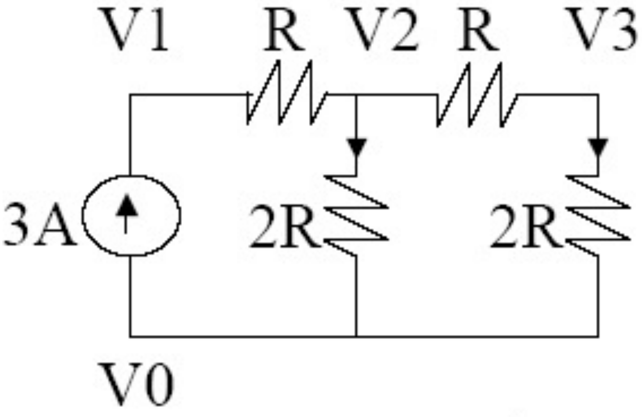
Writing Spice Decks

- Writing a SPICE deck is like writing a good program
  - Plan: sketch schematic on paper or in editor
    - Modify existing decks whenever possible
  - Code: strive for clarity
    - Start with name, email, date, purpose
    - Generously comment
  - Test
    - Predict what results should be
    - Compare with actual
    - *Garbage In, Garbage Out!*

SPICE Background

- SPICE generally is a Circuit Analysis tool for Simulation of Electrical Circuits in Steady-State, Transient, and Frequency Domains
- There are lots of SPICE tools available over the market,SBTSPICE, HSPICE, Spectre, TSPICE, Pspice, Smartspice, ISpice...
- Most of the SPICE tools are originated from Berkeley's SPICE program, therefore support common original SPICE syntax
- Basic algorithm scheme of SPICE tools are similar, however the control of time step, equation solver and convergence control might be different.

Solution for Linear Network



R=5ohm

$$\begin{pmatrix} 0.2 & 0 & -0.1 & -0.1 \\ 0 & 0.2 & -0.2 & 0 \\ -0.1 & -0.2 & 0.5 & -0.2 \\ 0 & 0 & -0.2 & 0.2 \end{pmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 0 \\ 0 \end{bmatrix}$$

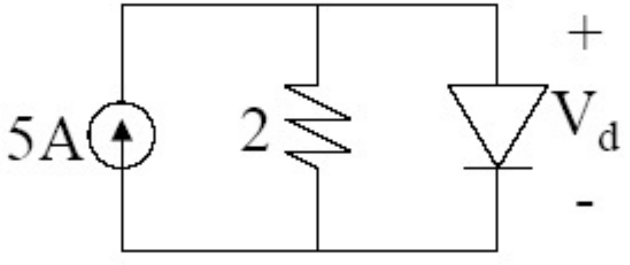
$$\begin{pmatrix} 0.2 & -0.2 & 0 \\ -0.2 & 0.5 & -0.2 \\ 0 & -0.2 & 0.2 \end{pmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} \quad V_0 \text{ ground}$$

With Gaussian elimination

$$\begin{pmatrix} 0.2 & -0.2 & 0 \\ 0 & 0.3 & -0.2 \\ 0 & 0 & 0.25 \end{pmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$$

Results : $V_3 = 12V$, $V_2 = 18V$, $V_1 = 33V$

Iteration and approximation -How solution is obtained



$I_d = 1pA*[\exp(40*V_d)-1]$

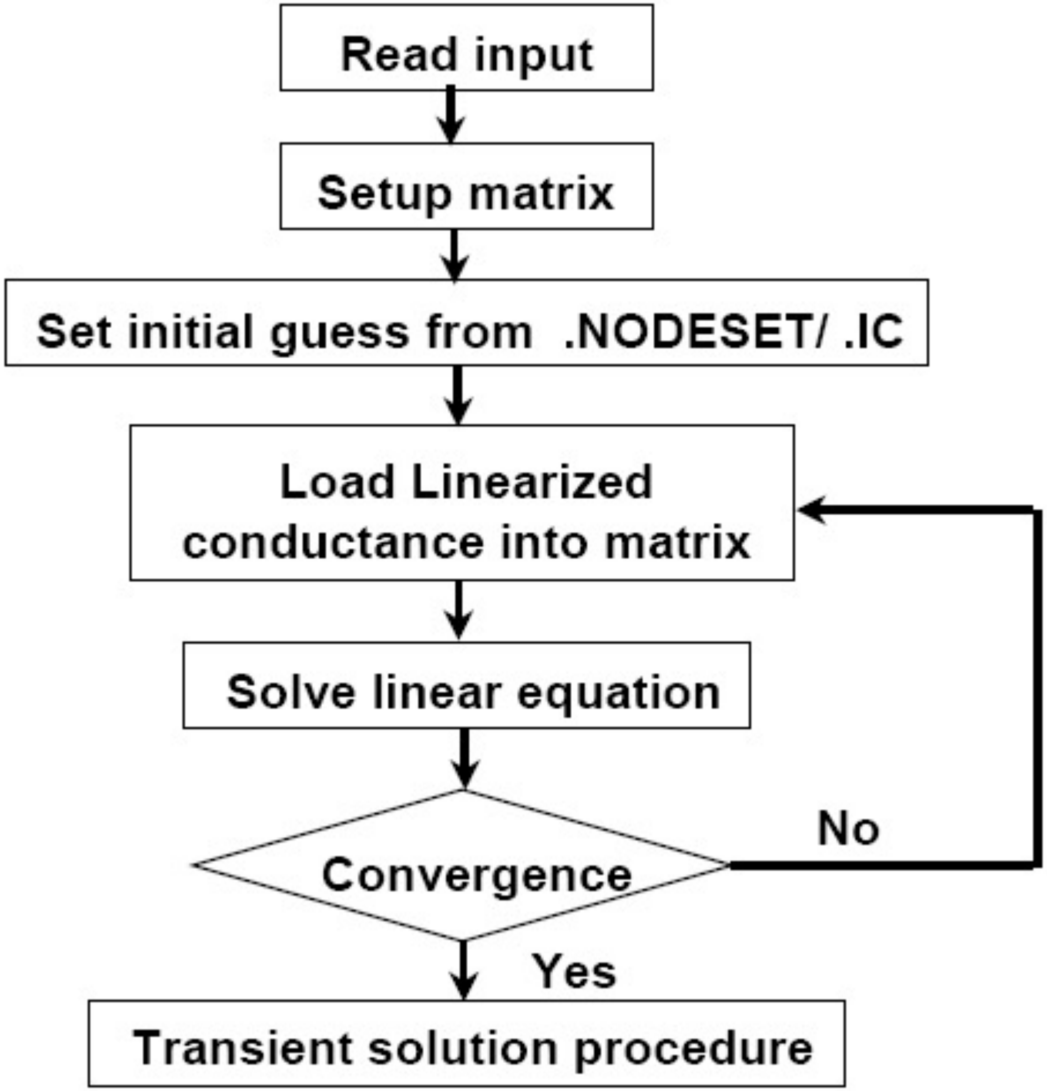
$5 = V_d/2 + I_d$

$5 = V_d/2 + 1pA*[\exp(40*V_d)-1]$

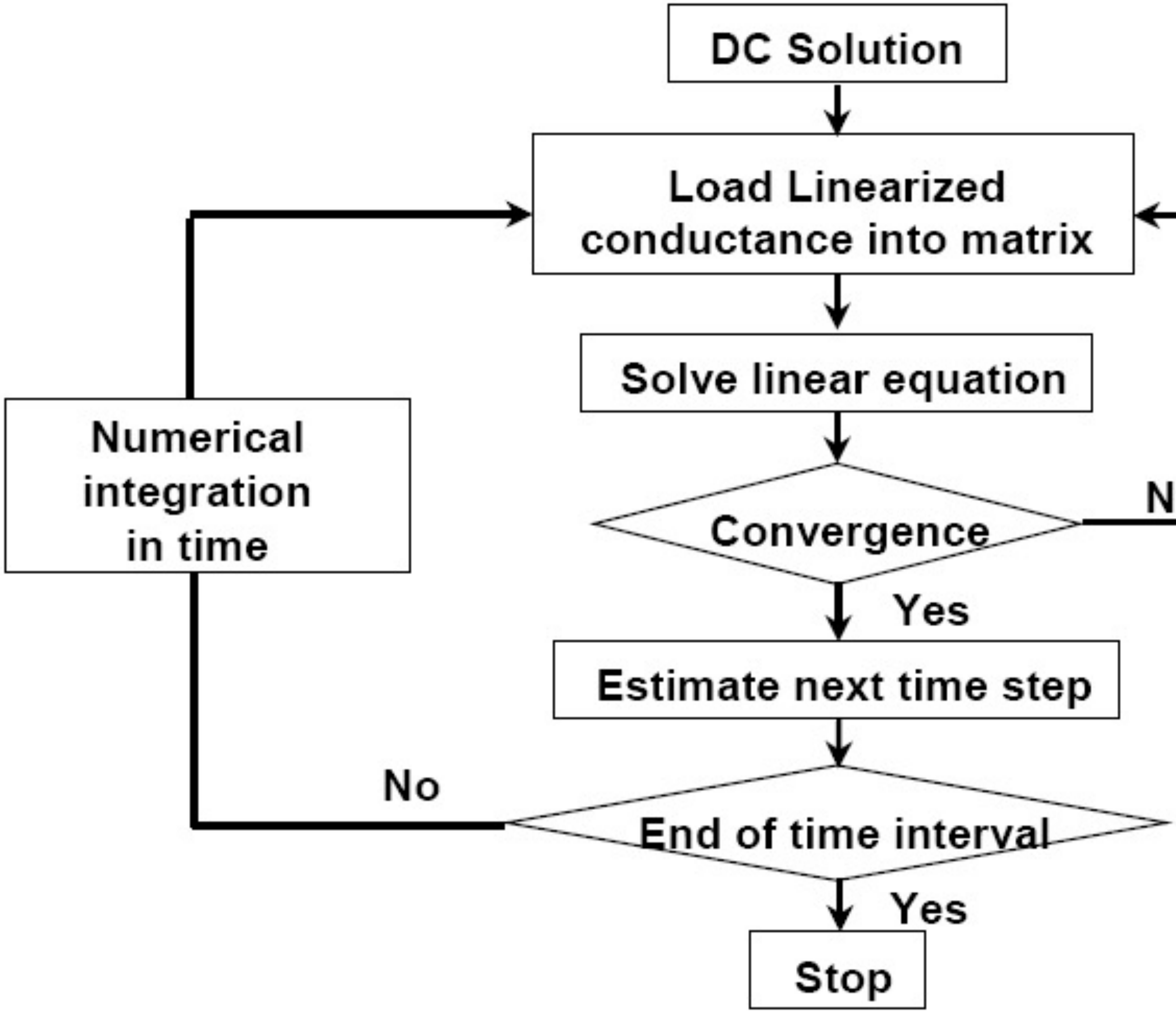
$V_{d+1} = V_d - F(V_d)/F'(V_d)$

	V_d	V_{d+1}	Delta V
1	1	0.975001	0.02499
2	0.975001	0.950002	0.02499
3	0.950002	0.925005	0.02499
4	0.925005	0.900015	0.02499
5	0.900015	0.875041	0.02497
6	0.875041	0.850113	0.02493
7	0.850117	0.825309	0.02481
8	0.825309	0.800838	0.02447
9	0.800838	0.777250	0.02359
10	0.777250	0.755885	0.02136
11	0.755885	0.739445	0.01644
12	0.739447	0.730983	0.00846
13	0.730983	0.729186	0.00179
14	0.729186	0.729119	0.00007

SPICE Simulation Algorithm —— DC


Read input
Setup matrix
Set initial guess from .NODESET/ .IC
Load Linearized conductance into matrix
Solve linear equation
Convergence
No
Yes
Transient solution procedure

SPICE Simulation Algorithm —— Transient



SPICE Elements

Letter	Element
R	Resistor
C	Capacitor
L	Inductor
K	Mutual Inductor
V	Independent voltage source
I	Independent current source
M	MOSFET
D	Diode
Q	Bipolar transistor
W	Lossy transmission line
X	Subcircuit
E	Voltage-controlled voltage source
G	Voltage-controlled current source
H	Current-controlled voltage source
F	Current-controlled current source

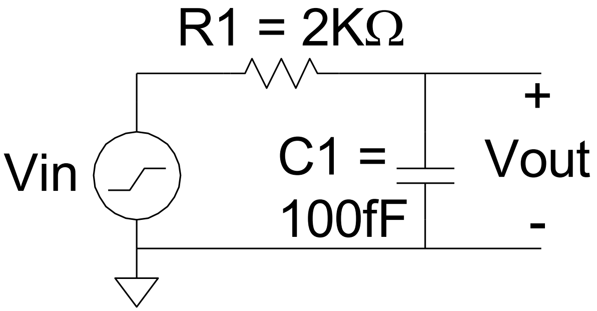
Units

Letter	Unit	Magnitude
a	atto	10^{-18}
f	fempto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
u	micro	10^{-6}
m	mili	10^{-3}
k	kilo	10^{3}
x	mega	10^{6}
g	giga	10^{9}

Ex: 100 femptofarad capacitor = 100fF, 100f, 100e-15

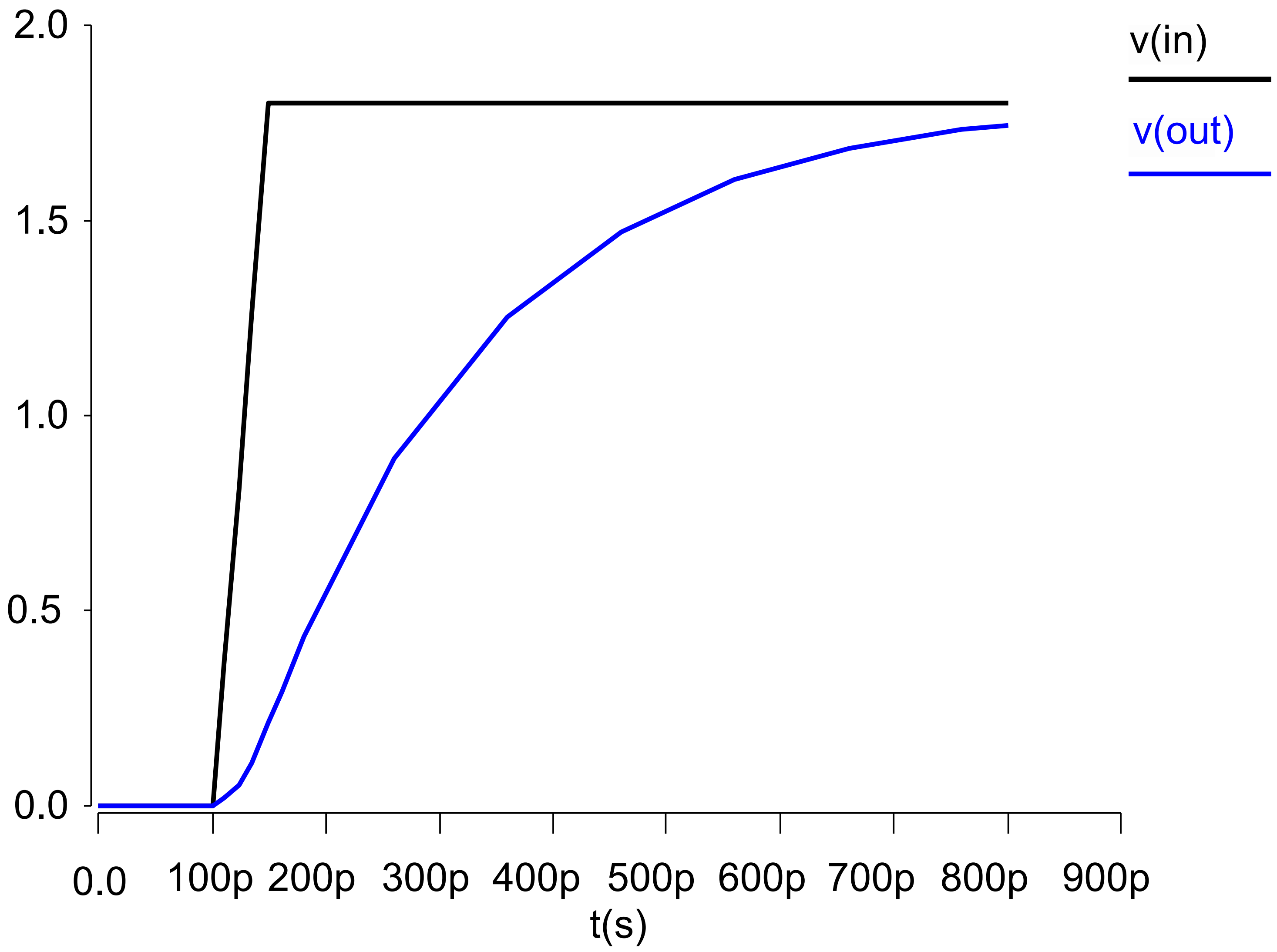
Example: RC Circuit

```
* rc.sp
* David_Harris@hmc.edu 2/2/03
* Find the response of RC circuit to rising input
*---------------------------------------
* Parameters and models
*---------------------------------------
.option post
*---------------------------------------
* Simulation netlist
*---------------------------------------
Vin    in    gnd    pwl    0ps 0 100ps 0 150ps 1.8 800ps 1.8
R1     in    out    2k
C1     out   gnd    100f
*---------------------------------------
* Stimulus
*---------------------------------------
.tran 20ps 800ps
.plot v(in) v(out)
.end
```



Do not forget!

Result (Graphical)



Sources

- *DC* Source

```
Vdd vdd gnd 2.5
```

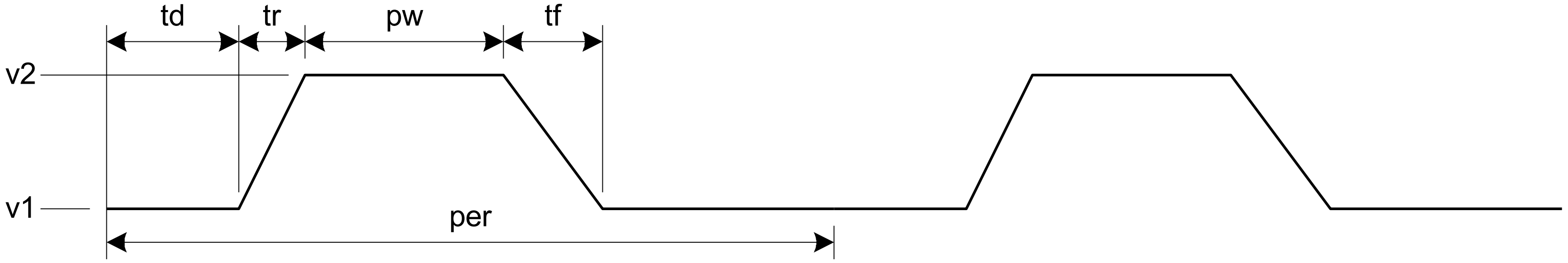
- *Piecewise Linear* Source

```
Vin in gnd pwl 0ps 0 100ps 0 150ps 1.8
  800ps 1.8
```

- *Pulsed* Source

```
Vck clk gnd PULSE 0 1.8 0ps 100ps 100ps
  300ps 800ps
```

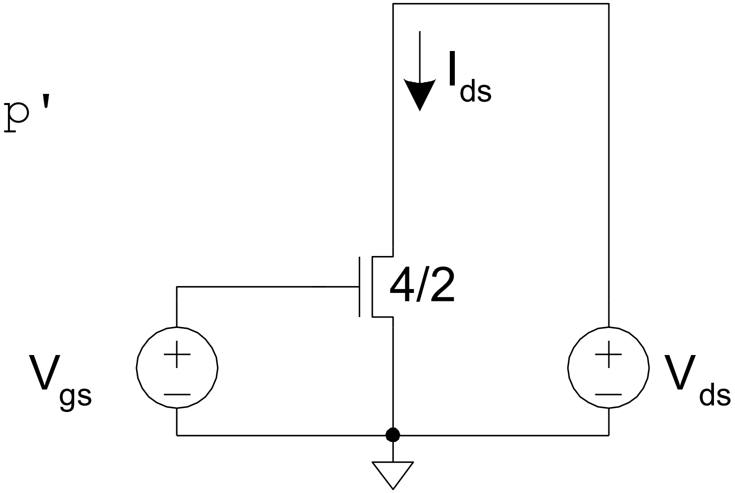
`PULSE v1 v2 td tr tf pw per`



DC Analysis

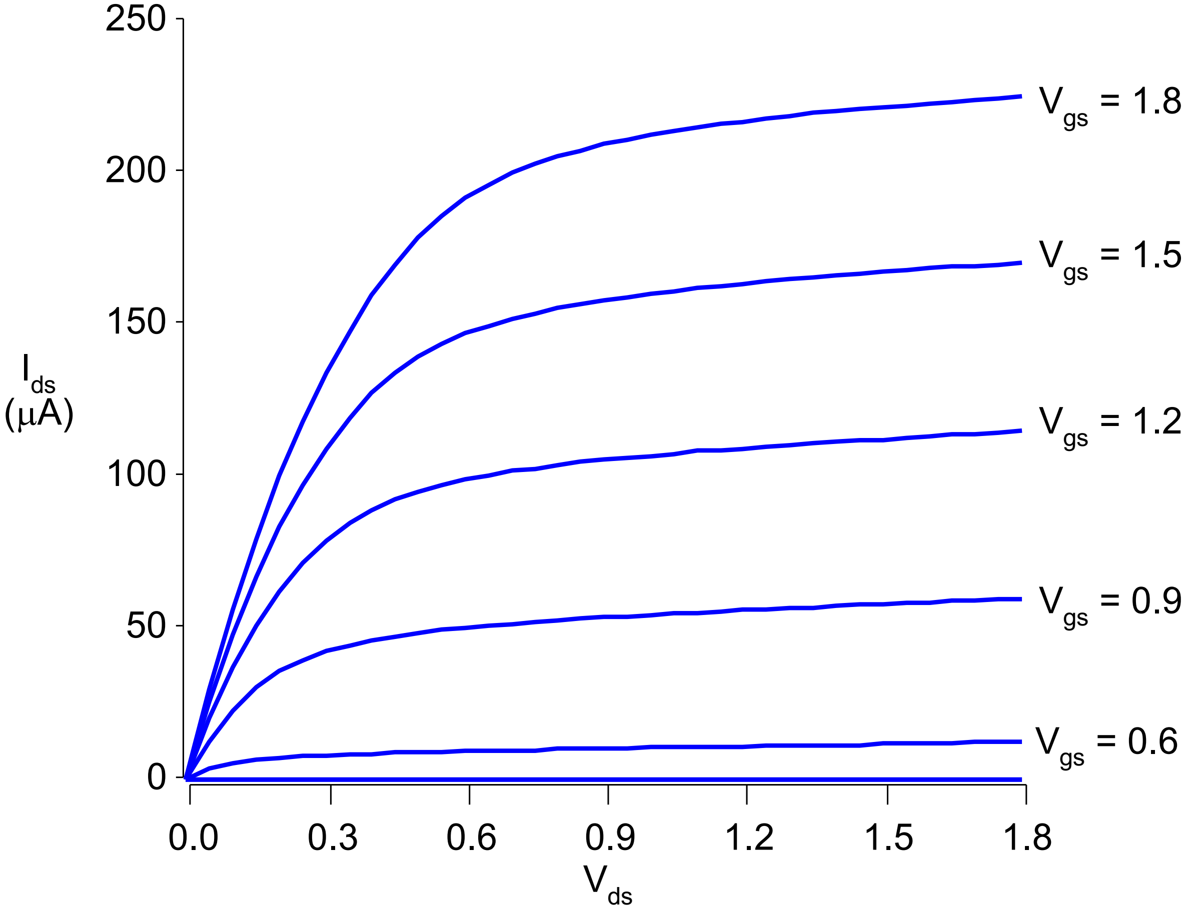
- mosiv.sp

```
*----------------------------------------
* Parameters and models
*----------------------------------------
.include '../models/tsmc180/models.sp'
.temp 70
.option post
*----------------------------------------
* Simulation netlist
*----------------------------------------
*nmos
Vgs    g     gnd    0
Vds    d     gnd    0
M1     d     g      gnd    gnd    NMOS   W=0.36u L=0.18u
*----------------------------------------
* Stimulus
*----------------------------------------
.dc Vds 0 1.8 0.05 SWEEP Vgs 0 1.8 0.3
.end
```



I-V Characteristics

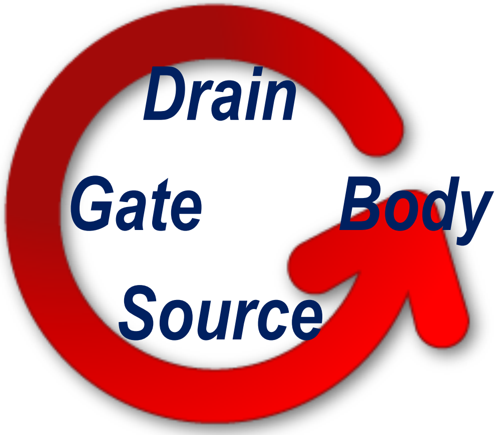
- nMOS I-V



MOSFET Elements

M element for MOSFET

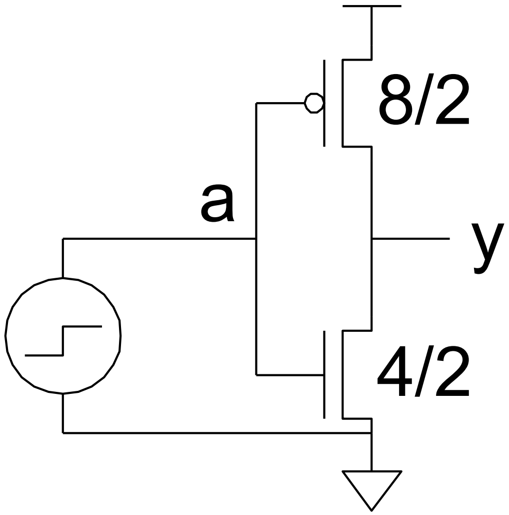
```
Mname drain gate source body type
+ W=<width> L=<length>
+ AS=<area source> AD = <area drain>
+ PS=<perimeter source> PD=<perimeter drain>
```



Transient Analysis

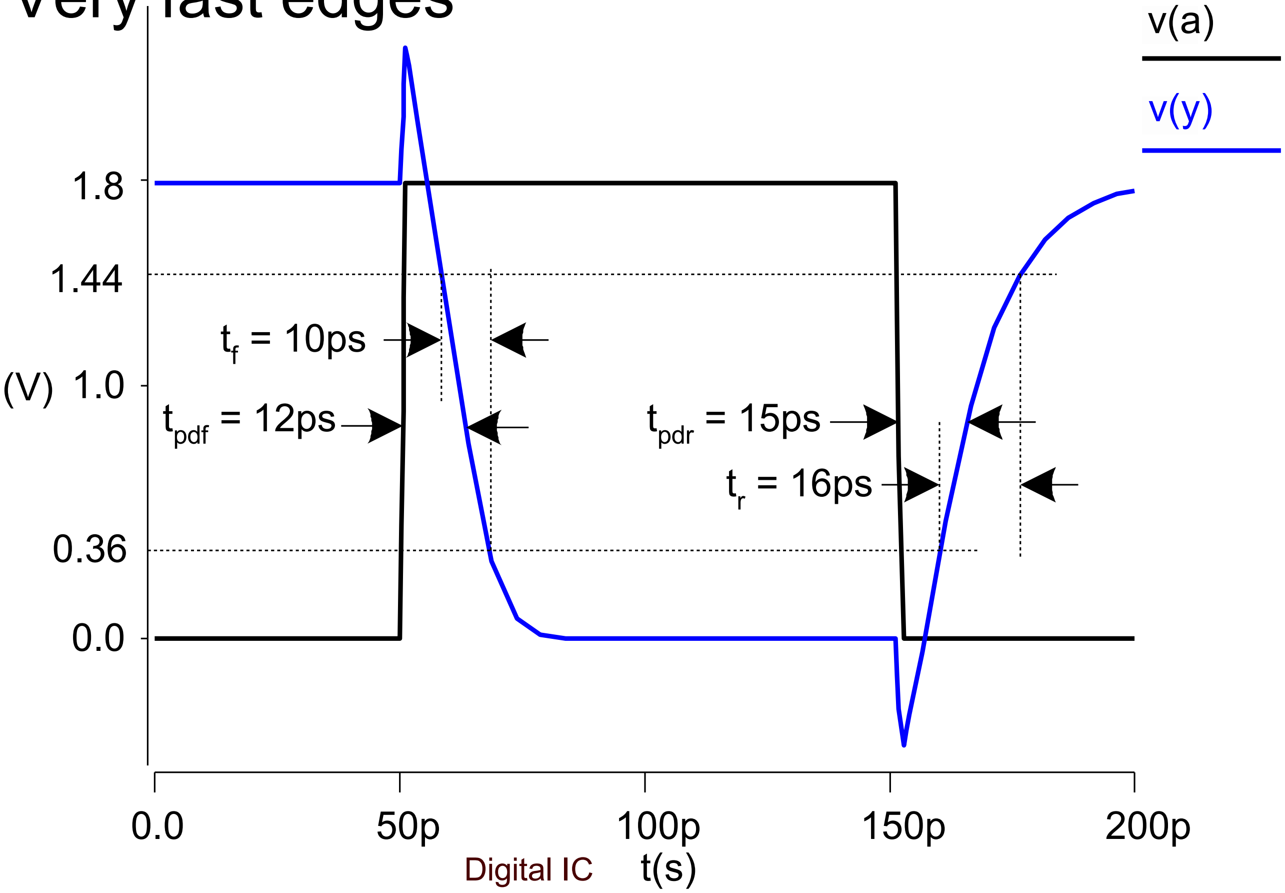
- inv.sp

```
* Parameters and models
*------------------------------------
.param SUPPLY=1.8
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post
* Simulation netlist
*------------------------------------
Vdd    vdd    gnd    'SUPPLY'
Vin    a      gnd    PULSE  0 'SUPPLY' 50ps 0ps 0ps 100ps
200ps
M1     y      a      gnd    gnd    NMOS   W=4    L=2
+ AS=20 PS=18 AD=20 PD=18
M2     y      a      vdd    vdd    PMOS   W=8    L=2
+ AS=40 PS=26 AD=40 PD=26
* Stimulus
*-------------------------------------
.tran 1ps 200ps
.end
```



Transient Results

- Unloaded inverter
  - Overshoot
  - Very fast edges

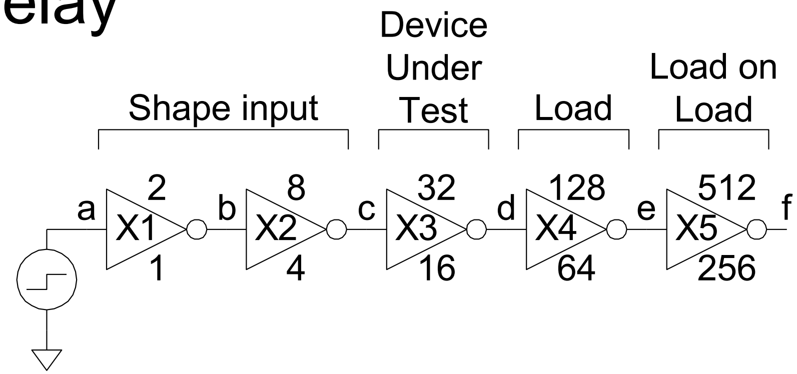


Subcircuits

- Declare common elements as subcircuits

```
.subckt inv a y N=4 P=8
M1 y a gnd gnd NMOS W='N' L=2
+ AS='N*5' PS='2*N+10' AD='N*5' PD='2*N+10'
M2 y a vdd vdd PMOS W='P' L=2
+ AS='P*5' PS='2*P+10' AD='P*5' PD='2*P+10'
.ends
```

- Ex: Fanout-of-4 Inverter Delay
  - Reuse inv
  - Shaping
  - Loading



FO4 Inverter Delay

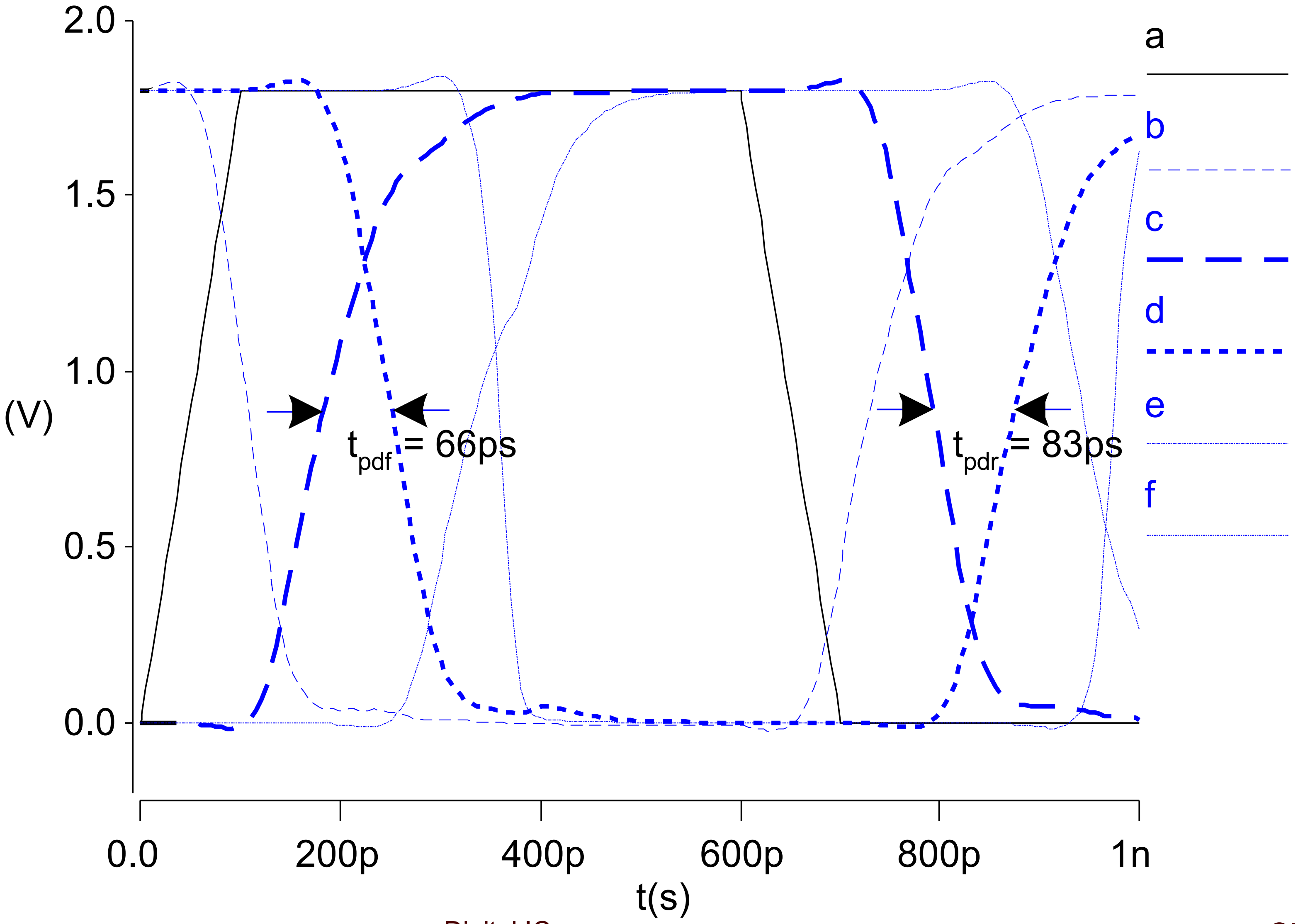
- fo4.sp

```
* Parameters and models
*-------------------------------------------------------------
.param SUPPLY=1.8
.param H=4
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post
* Subcircuits
*-------------------------------------------------------------
.global vdd gnd
.include '../lib/inv.sp’
* Simulation netlist
*-------------------------------------------------------------
Vdd     vdd    gnd     'SUPPLY'
Vin     a      gnd     PULSE  0 'SUPPLY' 0ps 100ps 100ps 500ps
1000ps
X1      a      b       inv             * shape input waveform
X2      b      c       inv    M='H'    * reshape input waveform
```

FO4 Inverter Delay Cont.

```
X3      c      d       inv    M='H**2' * device under test
X4      d      e       inv    M='H**3' * load
x5      e      f       inv    M='H**4' * load on load
* Stimulus
*---------------------------------------------------------
.tran 1ps 1000ps
.measure tpdr                         * rising prop delay
+     TRIG v(c)         VAL='SUPPLY/2' FALL=1
+     TARG v(d)         VAL='SUPPLY/2' RISE=1
.measure tpdf                         * falling prop delay
+     TRIG v(c)         VAL='SUPPLY/2' RISE=1
+     TARG v(d)         VAL='SUPPLY/2' FALL=1
.measure tpd param='(tpdr+tpdf)/2'           * average prop delay
.measure trise                               * rise time
+       TRIG v(d)       VAL='0.2*SUPPLY' RISE=1
+       TARG v(d)       VAL='0.8*SUPPLY' RISE=1
.measure tfall                               * fall time
+       TRIG v(d)       VAL='0.8*SUPPLY' FALL=1
+       TARG v(d)       VAL='0.2*SUPPLY' FALL=1
.end
```

FO4 Results


2.0
1.5
1.0
0.5
0.0
(V)
0.0
200p
400p
600p
800p
1n
t(s)
t_{pdf} = 66ps
t_{pdr} = 83ps
a
b
c
d
e
f

Optimization

- HSPICE can automatically adjust parameters
  - Seek value that optimizes some measurement
- Example: Best P/N ratio
  - We' ve assumed 2:1 gives equal rise/fall delays
  - But we see rise is actually slower than fall
  - What P/N ratio gives equal delays?
- Strategies
  - (1) run a bunch of sims with different P size
  - (2) let HSPICE optimizer do it for us

P/N Optimization

- fo4opt.sp

```
* Parameters and models
*--------------------------------------------------------------
.param SUPPLY=1.8
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post
* Subcircuits
*--------------------------------------------------------------
.global vdd gnd
.include '../lib/inv.sp’
* Simulation netlist
*--------------------------------------------------------------
Vdd     vdd     gnd     'SUPPLY'
Vin     a       gnd     PULSE   0 'SUPPLY' 0ps 100ps 100ps 500ps
1000ps
X1      a       b       inv     P='P1'          * shape input waveform
X2      b       c       inv     P='P1'  M=4     * reshape input
X3      c       d       inv     P='P1'  M=16    * device under test
```

P/N Optimization

```
X4      d      e      inv     P='P1'  M=64    * load
X5      e      f      inv     P='P1'  M=256   * load on load
* Optimization setup
*-------------------------------------------------------------
.param P1=optrange(8,4,16)        * search from 4 to 16, guess 8
.model optmod opt itropt=30       * maximum of 30 iterations
.measure bestratio param='P1/4’   * compute best P/N ratio
* Stimulus
*-------------------------------------------------------------
.tran 1ps 1000ps SWEEP OPTIMIZE=optrange RESULTS=diff
MODEL=optmod
.measure tpdr                         * rising propagation delay
+       TRIG v(c)     VAL='SUPPLY/2' FALL=1
+       TARG v(d)     VAL='SUPPLY/2' RISE=1
.measure tpdf                         * falling propagation delay
+       TRIG v(c)     VAL='SUPPLY/2' RISE=1
+       TARG v(d)     VAL='SUPPLY/2' FALL=1
.measure tpd param='(tpdr+tpdf)/2' goal=0    * average prop delay
.measure diff param='tpdr-tpdf' goal = 0     * diff between
delays
.end
```

P/N Results

- P/N ratio for equal delay is 3.6:1
  - $t_{pd} = t_{pdr} = t_{pdf}$ = 84 ps (slower than 2:1 ratio)
  - Big pMOS transistors waste power too
  - Seldom design for exactly equal delays
- What ratio gives lowest average delay?

```
.tran 1ps 1000ps SWEEP OPTIMIZE=optrange
  RESULTS=tpd MODEL=optmod
```

  - P/N ratio of 1.4:1
  - t_{pdr} = 87 ps, t_{pdf} = 59 ps, t_{pd} = 73 ps

Power Measurement

- HSPICE can measure power
  - Instantaneous P(t)
  - Or average P over some interval

```
.print P(vdd)
.measure pwr AVG P(vdd) FROM=0ns TO=10ns
```

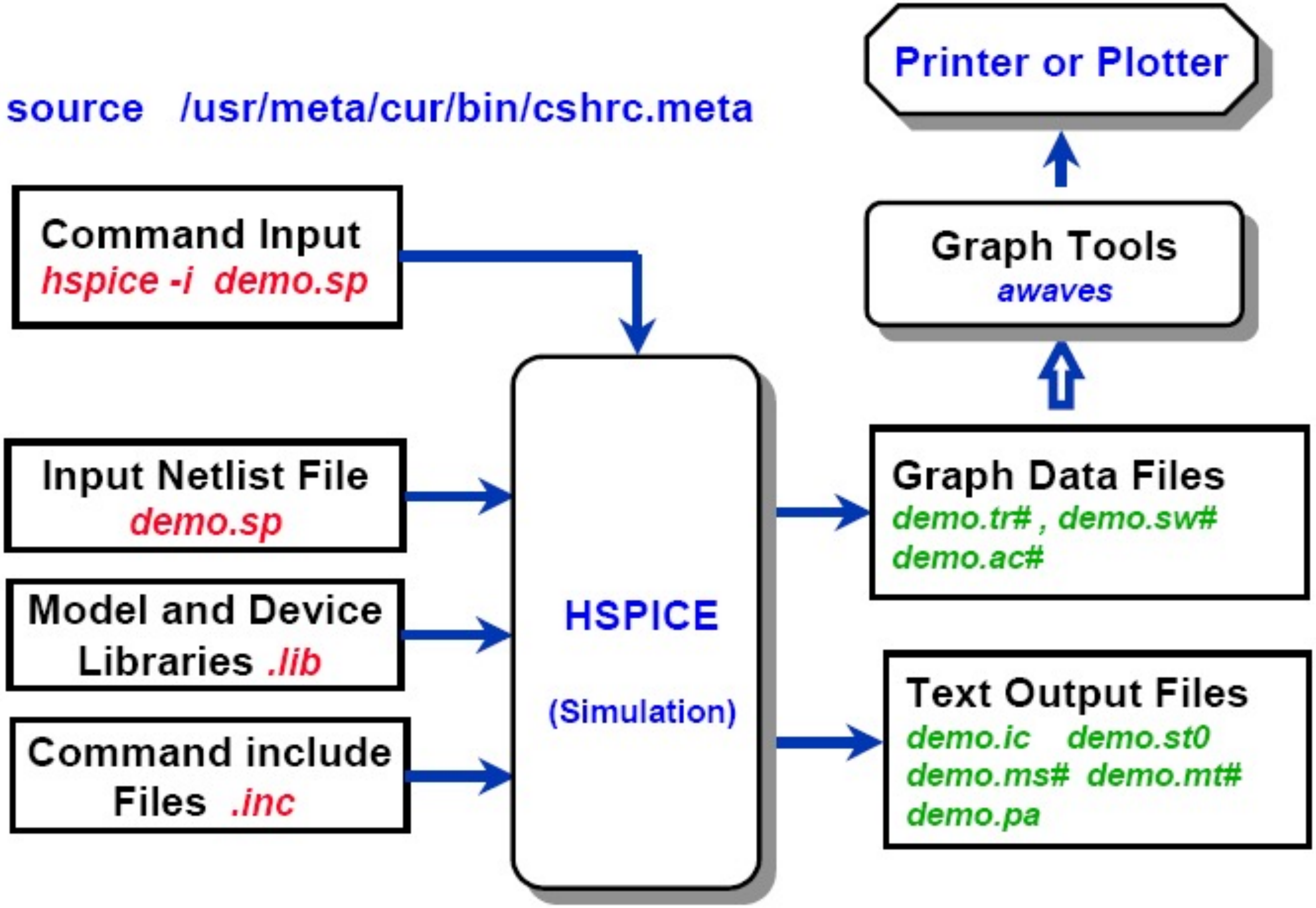
- Power in single gate
  - Connect to separate V_{DD} supply
  - Be careful about input power

contents

- SPICE Overview
- **Simulation Input and Controls**
- Sources and Stimuli
- Analysis Types
- Simulation Output and Controls
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

(1) HSPICE data flow

source /usr/meta/cur/bin/cshrc.meta



Command Input
hspice -i demo.sp

Input Netlist File
demo.sp

Model and Device Libraries *.lib*

Command include Files *.inc*

HSPICE

(Simulation)

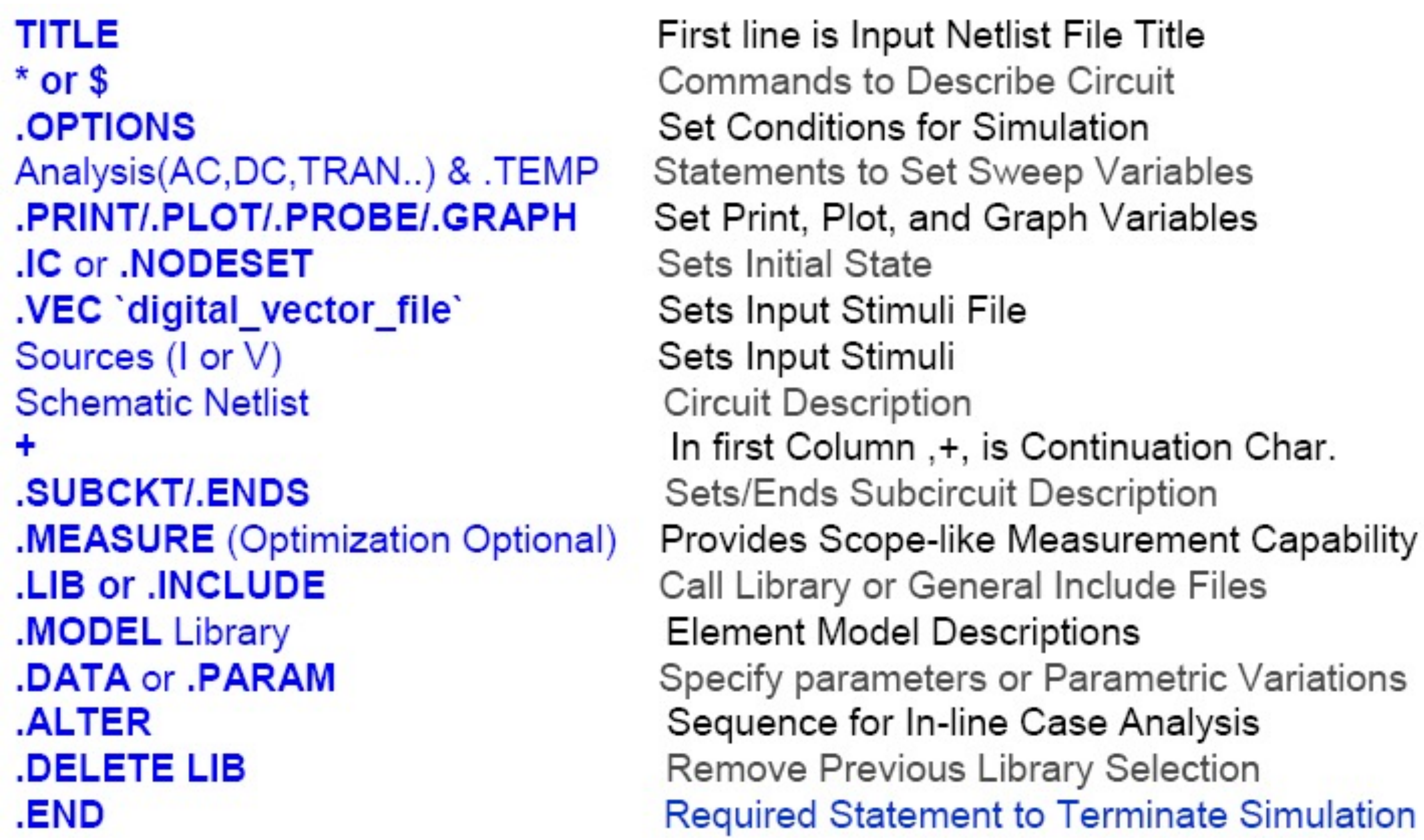
Graph Data Files
demo.tr# , demo.sw#
demo.ac#

Text Output Files
demo.ic demo.st0
demo.ms# demo.mt#
demo.pa

Graph Tools
awaves

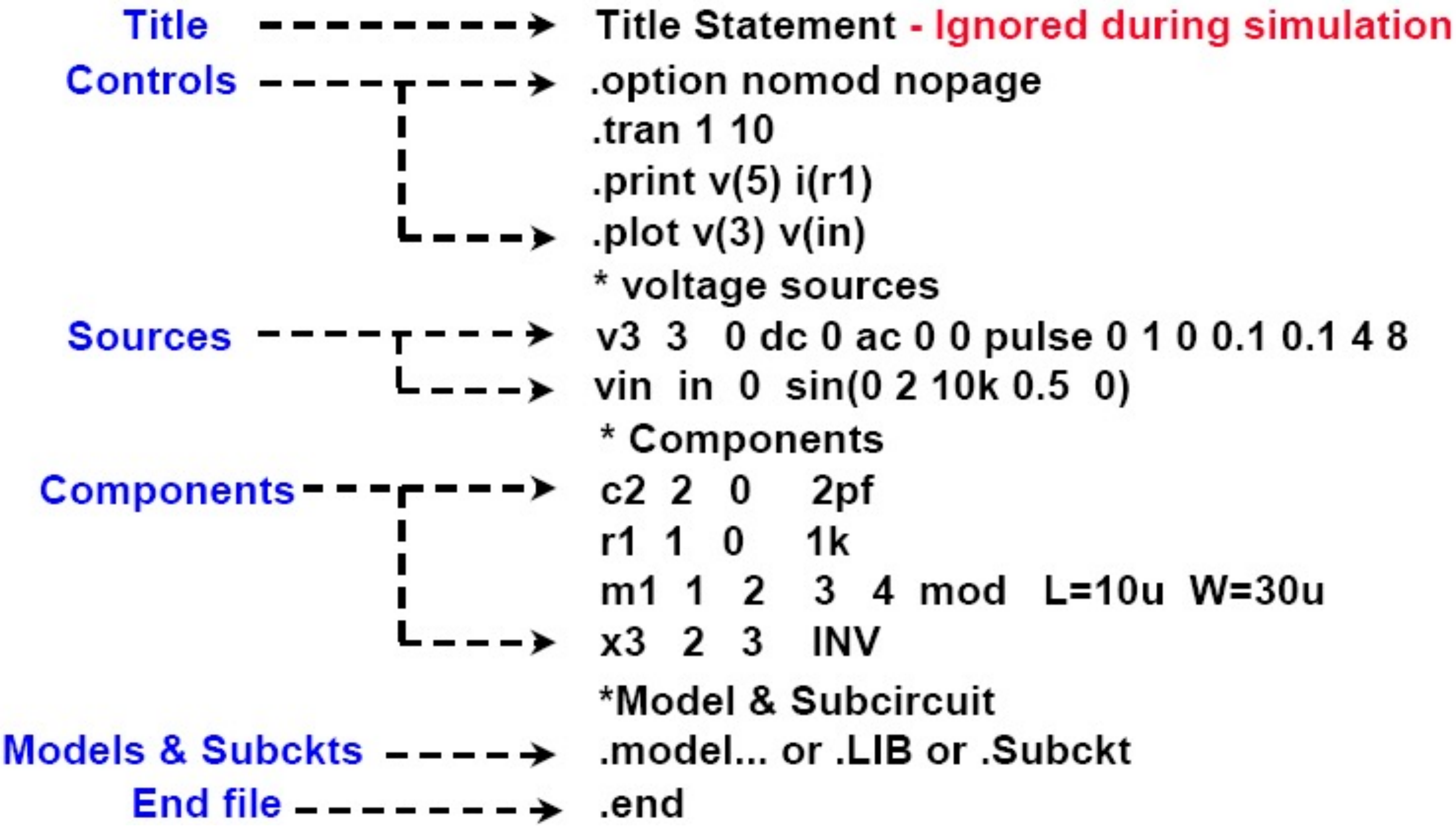
Printer or Plotter

(2)Netlist Statements and Elements



TITLE	First line is Input Netlist File Title
*** or $**	Commands to Describe Circuit
.OPTIONS	Set Conditions for Simulation
Analysis(AC,DC,TRAN..) & .TEMP	Statements to Set Sweep Variables
.PRINT/.PLOT/.PROBE/.GRAPH	Set Print, Plot, and Graph Variables
.IC or **.NODESET**	Sets Initial State
.VEC \`digital_vector_file\`	Sets Input Stimuli File
Sources (I or V)	Sets Input Stimuli
Schematic Netlist	Circuit Description
+	In first Column ,+, is Continuation Char.
.SUBCKT/.ENDS	Sets/Ends Subcircuit Description
.MEASURE (Optimization Optional)	Provides Scope-like Measurement Capability
.LIB or .INCLUDE	Call Library or General Include Files
.MODEL Library	Element Model Descriptions
.DATA or **.PARAM**	Specify parameters or Parametric Variations
.ALTER	Sequence for In-line Case Analysis
.DELETE LIB	Remove Previous Library Selection
.END	Required Statement to Terminate Simulation

(3) Netlist Structure (SPICE Preferred)



Title

Controls

Sources

Components

Models & Subckts

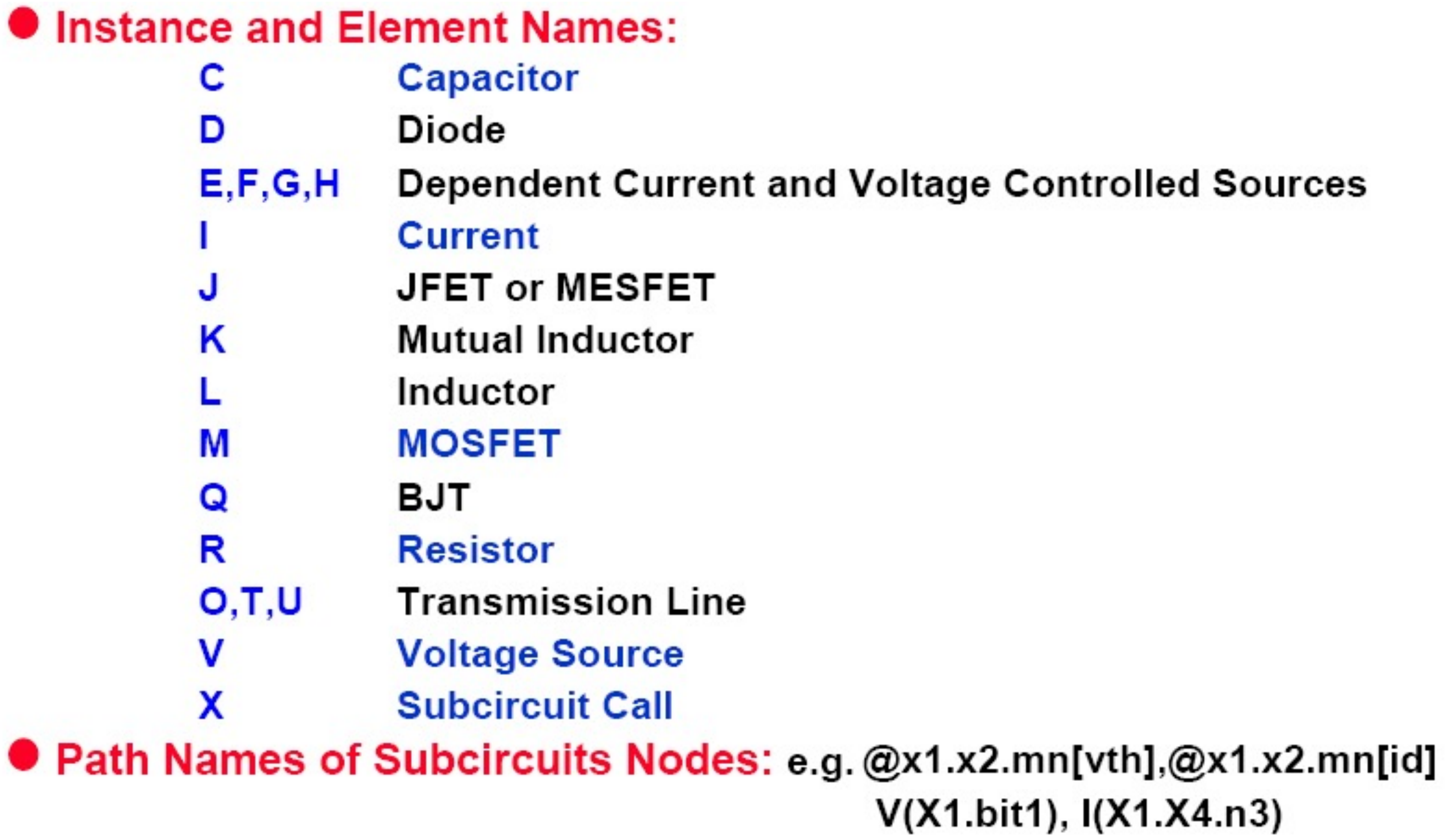
End file

```
Title Statement - Ignored during simulation
.option nomod nopage
.tran 1 10
.print v(5) i(r1)
.plot v(3) v(in)
* voltage sources
v3  3   0 dc 0 ac 0 0 pulse 0 1 0 0.1 0.1 4 8
vin  in  0  sin(0 2 10k 0.5  0)
* Components
c2  2   0     2pf
r1  1   0     1k
m1  1   2    3   4  mod   L=10u  W=30u
x3   2   3    INV
*Model & Subcircuit
.model... or .LIB or .Subckt
.end
```

(4) Element and Node Naming Conventions

- Node and Element Identification:
  - Either Names or Numbers (e.g. data1, n3, 11, ...)
  - 0 (zero) is Always Ground
  - Trailing Alphabetic Character are ignored in Node Number,(e.g. 5A=5B=5)
  - Ground may be 0, GND, !GND
  - All nodes are assumed to be local
  - Node Names can be may Across all Subcircuits by a .GLOBAL Statement (e.g. .GLOBAL VDD VSS)

(4) Element and Node Naming Conventions(cont.)



- Instance and Element Names:

C	Capacitor
D	Diode
E,F,G,H	Dependent Current and Voltage Controlled Sources
I	Current
J	JFET or MESFET
K	Mutual Inductor
L	Inductor
M	MOSFET
Q	BJT
R	Resistor
O,T,U	Transmission Line
V	Voltage Source
X	Subcircuit Call

- Path Names of Subcircuits Nodes: e.g. @x1.x2.mn[vth],@x1.x2.mn[id]
  V(X1.bit1), I(X1.X4.n3)

(5) Units and Scale Factors

- Units:
  - R Ohm (e.g. R1 n1 n2 1K)
  - C Farad(e.g. C2 n3 n4 1e-12)
  - L Henry(e.g. L3 n5 n6 1e-9)
- Scale Factors

F	*1e-15*
P	*1e-12*
N	*1e-9*
U	*1e-6*
M	*1e-3*

K	*1e3*
Meg	*1e6*
G	*1e9*
T	*1e12*
DB	*20log10*

Examples:

1pF
1nH
10MegHz
vdb(v3)

- Technology Scaling:All Length and Widths are in Meters
- Using .options scale=1e-6   L2 W100

(6) Input Control Statements : .ALTER

- .ALTER Statement:Description
- Rerun a Simulation Several Times with Different

Circuit Topology
Models
Elements Statement
Parameter Values
Options
Analysis Variables, etc

- 1st Run:Reads Input Netlist File up to the first .ALTER
- Subsequent:Input Netlists to next .ALTER, etc.

(6) Input Control Statements : .ALTER(cont.)

```
*file2:alter2.sp alter examples    $ Title Statement
.lib ‘mos.lib’ normal
.param wval=50u Vdd=5v
R4 4 3 100
.alter
.del lib ‘mos.lib’ normal          $remove normal lib
.lib ‘mos.lib’ fast                $get fast model lib
.alter
.temp -50 0 50                     $run with different temperature
r4 4 3 1K                          $change resistor value
c3 3 0 10p                         $add the new element
.param wval=100u Vdd=5.5V          $change parameters
.end
```

(6) Input Control Statements : .ALTER(cont.)

- 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

(7). Input Control Statements: .DATA

- .DATA Statement: Inline or Multiline .DATA Example

```
* Inline .DATA example
.Tran 1n 100n SWEEP DATA=devinf
.AC DEC 10Hz 100kHz SWEEP DATA=devinf
.DC TEMP -55 125 10 SWEEP DATA=devinf
*
.DATA devinf Width Length Vth Cap
+               10u   100u   2v   5p
+               50u   600u  10v  10p
+              100u   200u   5v  20p
.ENDDATA
```

```
* Multiline .DATA example
.PARAM Vds=0 Vbs=0 L=1.0u
DC DATA=vdot
.DATA vdot
Vbs       Vds       L
0         0.1       1.0u
0         0.1       1.5u
-1        0.1       1.0u
0         0.5       1.0u
.ENDDATA
```

(8). Input Control Statements: .TEMP

- .TEMP Statement: Description
  - When TNOM is not Specified, it will Default to 25 ^{o}C for HSPICE
  - Example 1:

    .TEMP 30 *Ckt simulated at 30 ^{o}C
  - Example 2:

    .OPTION TEMP = 30 *Ckt simulated at 30 ^{o}C
  - Example 3:

    .TEMP 100

    D1 n1 n2 DMOD DTEMP=130 *D1 simulated at 130 ^{o}C

    D2 n3 n4 DMOD *D2 simulated at 100 ^{o}C

    R1 n5 n6 1K

(9). Input Control Statements:

- **.OPTION**

Listing Formats
Simulation Convergence
Simulation Speed
Model Resolution
Algorithm
Accuracy

- **.OPTION** Statement : Description
  - **.**Option Controls for
  - .Option Syntax and Example

```
.OPTION opt1 <opt2> .... <opt=x>
.OPTION LVLTIM=2 POST PROBE SCALE=1
```

(10). Library Input Statement

- .INCLUDE Statement Copy the content of file into netlist
  - *.INCLUDE ‘$installdir/parts/ad’*
- .LIB Definition and Call Statement File reference and Corner selection

```
.LIB TT
.MODEL nmos_tt(level=49 Vt0=0.7
+TNOM=27…………..)
.ENDL TT
```

```
.LIB ‘users/model/tsmc/logic06.mod’ TT
```

```
.PROTECT
.LIB “~users/model/tsmc/logic06.mod” TT
.UNPROTECT
```

Prevent the listing of included contents

(11)Hierarchical Circuits, Parameters, and Models

- .SUBCKT Statement : Description
  - .SUBCKT Syntax

```
.SUBCKT subname n1 <n2 n3...> <param=val...>
n1 ... Node Number for External Reference; Cannot be Ground node (0)
       Any Element Nodes Appearing in Subckt but not Included in this
       list are Strictly LOCAL, with these Exceptions :
             (1) Ground Node (0)
             (2) Nodes Assigned using .GLOBAL Statement
             (3) Nodes Assigned using BULK=node in MOSFET or BJT Models
param Used ONLY in Subcircuit, Overridden by Assignment in Subckt Call
        or by values set in .PARAM Statement
.ENDS [subname]
```

(11). Hierarchical Circuits, Parameters, and Models (Cont.)

- .SUBCKT Statement : Examples

```
.PARAM VALUE=5V WN=2u WP=8u
*
.SUBCKT INV IN OUT WN=2u WP=8u
M1 OUT IN VDD VDD P L=0.5u W=WP
M2 OUT IN 0 0 N L=0.5u W=WN
R1 OUT 4 1K
R2 4 5 10K
.ENDS INV
*
X1 1 2 INV WN=5u WP=20u
X2 2 3 INV WN=10u WP=40u
```

- Subcircuit Calls (X Element Syntax)

```
Xyyyy n1 <n2 n3...> subname <param=val...> <M=val>
XNOR3 1 2 3 4 NOR WN=3u LN=0.5u M=2
```

(12). Example Circuit

subckt call →

```
Invter gain
.lib ’logs353v.l' TT
.option acct post
.param vref=1.0 Wmask=25u LMask=0.8u vcc=5
.subckt inv out inp d
mn1 out inp 0 0 nch w=Wmask l=Lmask
mp1 out inp d d pch w=Wmask l=Lmask
.ends inv
x1 out inp vdd inv
vdd vdd 0 dc vcc
vin inp 0 dc 0 pulse(0 vcc 0 1ns 1ns 2ns 5ns)
.dc vin 0 vcc 0.01 sweep data=d1
.tran 0.1ns 10ns sweep data=d1
.meas tran tpd trig v(inp) val=2 rise=1
+ targ v(out) val=3 fall=1
.probe v(inp) v(out)
.data d1
Lmask Wmask
0.6u 250u
2.0u 420u
.enddata
.end
```

contents

- SPICE Overview
- Simulation Input and Controls
- **Sources and Stimuli**
- Analysis Types
- Simulation Output and Controls
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

Source types

- Source / Stimuli : drive source of circuit
  - 1. Independent DC Sources(supply fixed voltage/current)
  - 2. Independent AC/TRAN Sources(for input signal)
  - 3. dependent DC/AC/TRAN Sources(for models)
    - 压控电压源(VCVS-Voltage-Controlled Current Sources)
    - 压控电流源(VCCS)
    - 流控电压源(CCVS)
    - 流控电流源(CCCS)

(1). Independent Source Elements: AC, DC Sources

- Source Element Statement :
- Syntax :

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

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

- Examples of DC & AC Sources :

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

```
V4 4 0 AC=10V, 90
V5 5 0 AC 1.0 180
*AC or Freq. Response Provide Impulse Response
```

(2). Independent Source Functions : Transient Sources

- Transient Sources Statement :
  - Types of Independent Source Functions :

Pulse (PULSE Function)
Sinusoidal (SIN Function)
Exponential (EXP Function)
Piecewise Linear (PWL Function)
Single-Frequency FM (SFFM Function)
Single-Frequency AM (AM Function)

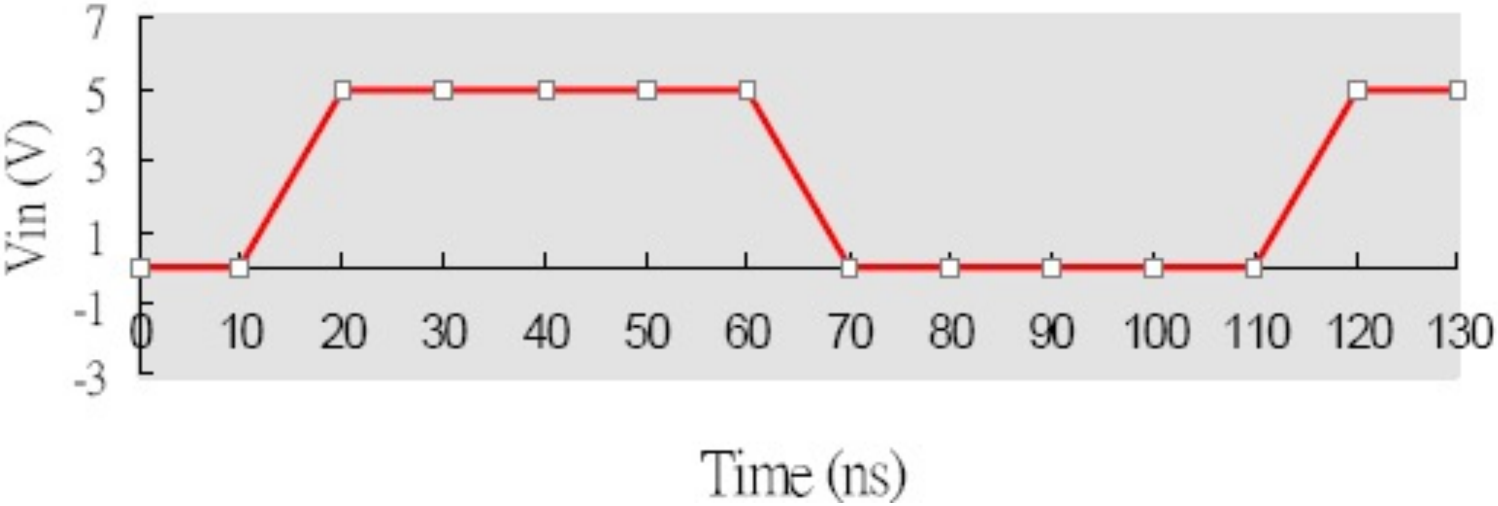
(2). Indep. Source Functions : Transient Sources(Cont.)

- Pulse Source Function : PULSE
- Syntax :

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

- Example :

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



(2). Indep. Source Functions : Transient Sources(Cont.)

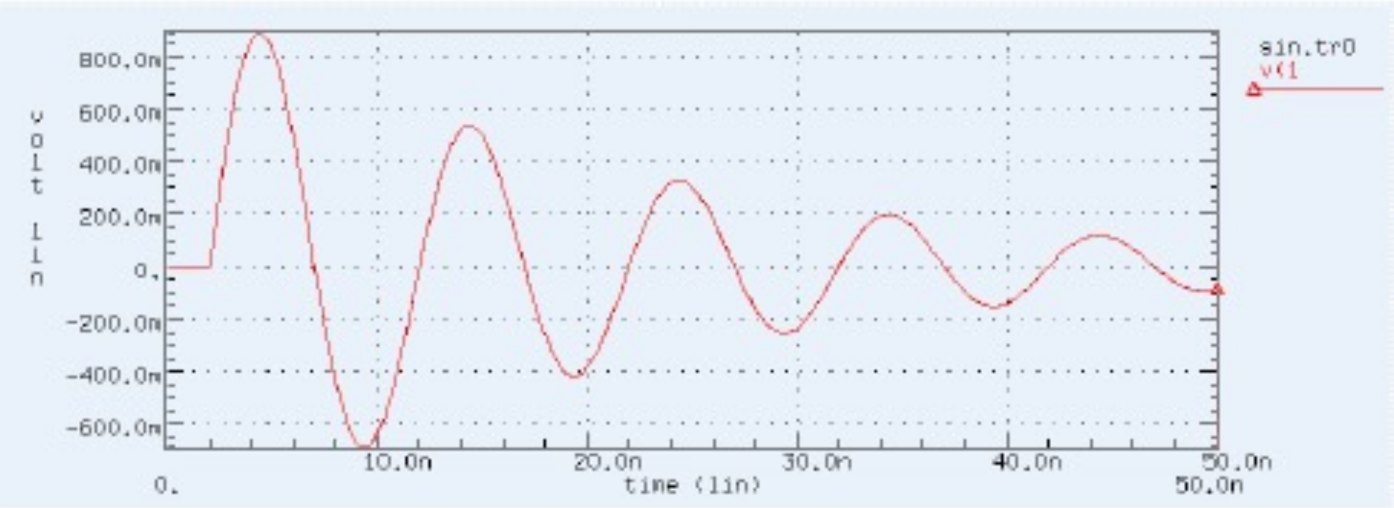
- Sinusoidal Source Function : SIN
- Syntax :

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)



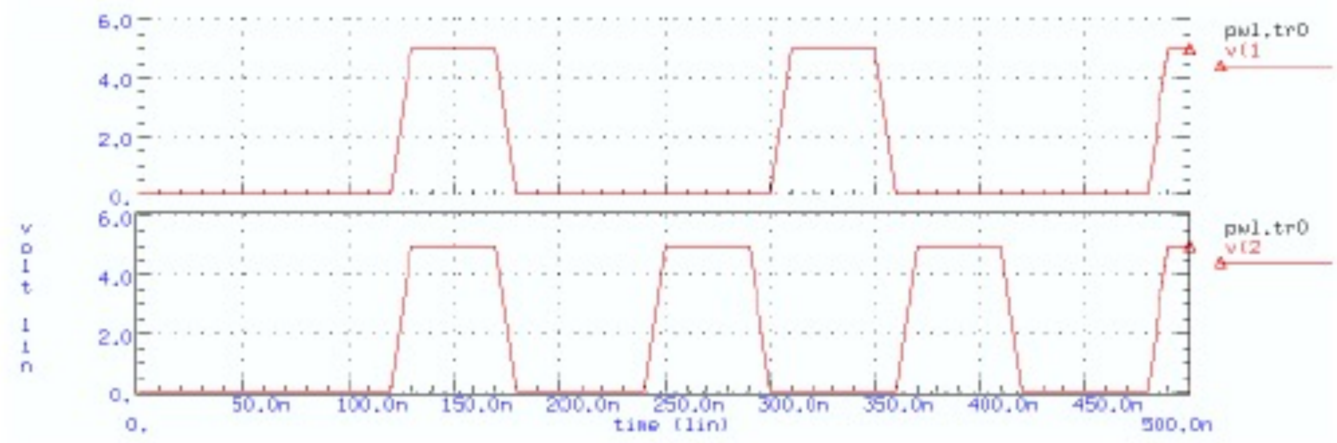
(2). Indep. Source Functions : Transient Sources(Cont.)

- Piecewise Linear Source Function : PWL or PL
- Syntax :

PWL (<t1 v1 t2 v2 ......> <R<=repeat>> <Tdelay=delay>)
$ R=repeat_from_what_time TD=time_delay_before_PWL_start

- Example :

V1 1 0 PWL 60n 0v, 120n 0v, 130n 5v, 170n 5v, 180n 0v, R 0
V2 2 0 PL 0v 60n, 0v 120n, 5v 130n, 5v 170n , 0v 180n , R 60n



(3). Voltage and Current Controlled Elements

- Dependent Sources (Controlled 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

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

contents

- SPICE Overview
- Simulation Input and Controls
- Sources and Stimuli
- Analysis Types
- Simulation Output and Controls
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

(1). Analysis Types & Orders

- Types & Order of Execution :
  - DC Operating Point : First Calculated for ALL Analysis Types

    .OP .IC .NODESET
  - DC Sweep & DC Small Signal Analysis :

    .DC .TF .PZ .SENS
  - AC Sweep & Small Signal Analysis :

    .AC .NOISE .DISTO .SAMPLE .NET
  - Transient Analysis:

    .TRAN .FOUR (UIC)
  - Other Advanced Modifiers :
    - Temperature Analysis, Optimization

(2). Analysis Types : DC Operating Point Analysis

- Initialization and Analysis:
  - First Thing to Set the DC Operating Point Values for All Nodes and Sources : Set Capacitors OPEN & Inductors SHORT
  - Using .IC or .NODESET to set the Initialized Calculation If UIC Included in .TRAN ==> Transient Analysis Started Directly by Using Node Voltages Specified in .IC Statement
  - .NODESET Often Used to Correct Convergence Problems in .DC Analysis
  - IC force DC solutions, however .NODESET set the initial guess.
- OP Statement :
  - .OP Print out :(1). Node Voltages; (2). Source Currents; (3). Power Dissipation; (4). Semiconductors Device Currents, Conductance, Capacitance

(3). Analysis Types : DC Sweep & DC Small Signal Analysis

- DC Analysis Statements :
  - .DC : Sweep for Power Supply, Temp., Param., & Transfer Curves
  - .OP : Specify Time(s) at which Operating Point is to be Calculated
  - .PZ : Performs Pole/Zero Analysis (.OP is not Required)
  - .TF : Calculate DC Small-Signal Transfer Function (.OP is not Required)
- .DC Statement Sweep :
  - Any Source Value Any Parameter Value
  - Temperature Value
  - DC Circuit Optimization
  - DC Model Characterization

(3). Analysis Types : DC Sweep & DC Small Signal Analysis (Cont.)

- .DC Analysis : Syntax

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

- Examples :

```
.DC VIN 0.25 5.0 0.25
.DC VDS 0 10 0.5 VGS 0 5 1
.DC TEMP -55 125 10
.DC TEMP POI 5 0 30 50 100 125
.DC xval 1k 10k 0.5k SWEEP TEMP LIN 5 25 125
.DC DATA=datanm SWEEP par1 DEC 10 1k 100k
.DC par1 DEC 10 1k 100k SWEEP DATA=datanm
```

(3). Analysis Types : DC Sweep & DC Small Signal Analysis (Cont.)

- .DC Analysis : Syntax

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

- Examples :

```
.DC VIN 0.25 5.0 0.25
.DC VDS 0 10 0.5 VGS 0 5 1
.DC TEMP -55 125 10
.DC TEMP POI 5 0 30 50 100 125
.DC xval 1k 10k 0.5k SWEEP TEMP LIN 5 25 125
.DC DATA=datanm SWEEP par1 DEC 10 1k 100k
.DC par1 DEC 10 1k 100k SWEEP DATA=datanm
```

(4). Analysis Types : Transient Analysis (Cont.)

- .TRAN Analysis : Calculate Time-Domain Response
  - Temperature
  - Optimization
  - .Param Parameter

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

- Examples :

```
.TRAN 1NS 100NS
.TRAN 10NS 1US UIC
.TRAN 10NS 1US UIC SWEEP TEMP -55 75 10 $ step=10
.TRAN 10NS 1US SWEEP load POI 3 1pf 5pf 10pf
.TRAN DATA=datanm
```

contents

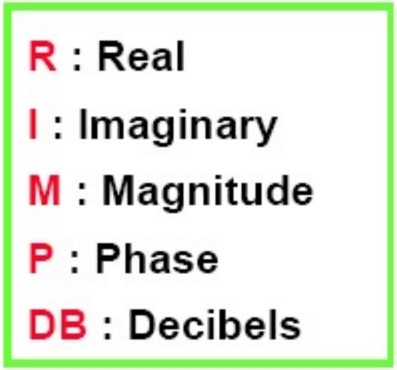
- SPICE Overview
- Simulation Input and Controls
- Sources and Stimuli
- Analysis Types
- **Simulation Output and Controls**
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

(1). Output Files Summary:

Output File Type	Extension
Output Listing	.lis
DC Analysis Results	.sw#
DC Analysis Measurement Results	.ms#
AC Analysis Results	.ac#
AC Analysis Measurement Results	.ma#
Transient Analysis Results	.tr#
Transient Analysis Measurement Results	.mt#
Subcircuit Cross-Listing	.pa#
Operating Point Node Voltages (Initial Condition)	.ic

(3). Output Variable Examples: DC, Transient, AC, Template

- 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), IP4(M4)
- Element Template :
- @x1.mn1[vth]
- @x1.mn1[gds]
- @x1.mn1[gm],@x1.mn1[gbs],@x1.mn1[



R : Real

I : Imaginary

M : Magnitude

P : Phase

DB : Decibels

(4). Regional Analysis of Power for Transient Analysis

- .option rap = x <Rap_Tstart=Tstart><Rap_Tstop=Tstop>
  - 0 < x < 1 , The nodes with average power consumption greater than (1-x)*(total power consumption) will be listed
  - x = 1 will dump all power information of nodes

  - Tstart is the start time for power report, default is 0
  - Tstop is the stop time for power report, default is simulation stop time
  - All RAP output is stored in file .rap

(5). Output Variable Examples: Parametric Statements

- 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) asin(x) acos(x) atan(x) sinh(x) tanh(x) abs(x)*
  - *sqrt(x) log(x) log10(x) exp(x) db(x) min(x,y) max(x,y) power(x,y)...*

(6). Displaying Simulation Results: .PRINT & .PLOT

- Syntax :

  .PRINT anatype ov1 <ov2 ov2...>

  Note : .PLOT with same Syntax as .PRINT, Except Adding <pol1, phi1> to set plot limit

- 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)
```

(7). Output Variable Examples: .MEASURE Statement

- General Descriptions :
  - .MEASURE Statement Prints User-Defined Electrical Specifications of a Circuit and is Used Extensively in Optimization
  - .MEASURE Statement Provides Oscilloscope-Like Measurement Capability for either AC , DC, or Transient Analysis
  - Using .OPTION AUTOSTOP to Save Simulation Time when TRIG-TARG or FIND-WHEN Measure Functions are Calculated
- Fundamental Measurement Modes :
  - Rise, Fall, and Delay (TRIG-TARG)
  - AVG, RMS, MIN, MAX, & Peak-to-Peak (FROM-TO)
  - FIND-WHEN

(8). MEASURE Statement : Rise, Fall, and Delay Syntax

```
.MEASURE DC|AC|TRAN result_var TRIG ... TARG ... <Optimization Option>
```

- 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>
- TARG ... : TARG targ_var VAL=targ_value <TD=time_delay>
- + <CROSS=n|LAST> <RISE=r_n|LAST> <FALL=f_n|LAST>
- TRIG ... : TRIG AT=value
- <Optimization Option> : <GOAL=val> <MINVAL=val> <WEIGHT=val>

- Example:

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

(9). MEASURE Statement : AVG, RMS, MIN, MAX, & P-P

- Syntax :

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

- result_var : 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
- <Optimization Option>: <GOAL=val> <MINVAL=val> <WEIGHT=val>

- Example:

```
.meas TRAN minval MIN v(1,2) from=25ns to=50ns
.meas TRAN tot_power AVG power from=25ns to=50ns
.meas TRAN rms_power RMS power
```

(10). MEASURE Statement : Find & When Function

- Syntax :

  .measure DC|AC|TRAN result WHEN ... <Optimization Option>

  .measure DC|AC|TRAN result FIND out_var1 WHEN ...<Optimization Option>

  .measure DC|AC|TRAN result_var FIND out_var1 AT=val <Optimization Option>

  - 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>
  - <Optimization Option> : <GOAL=val> <MINVAL=val> <WEIGHT=val>

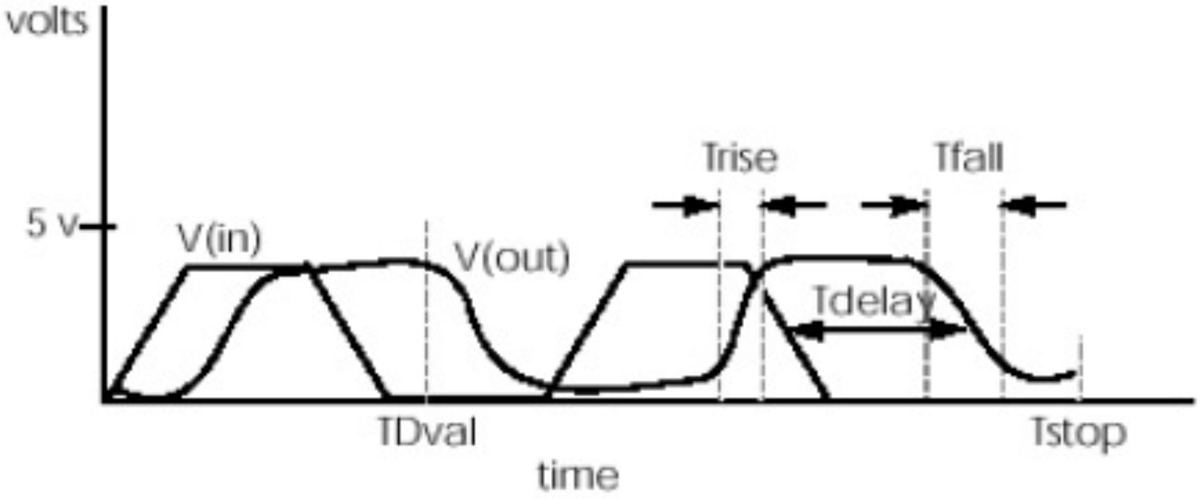
- Example:

```
.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
```

(11). MEASURE Statement : Application Examples

- Rise, Fall, and Delay Calculations :

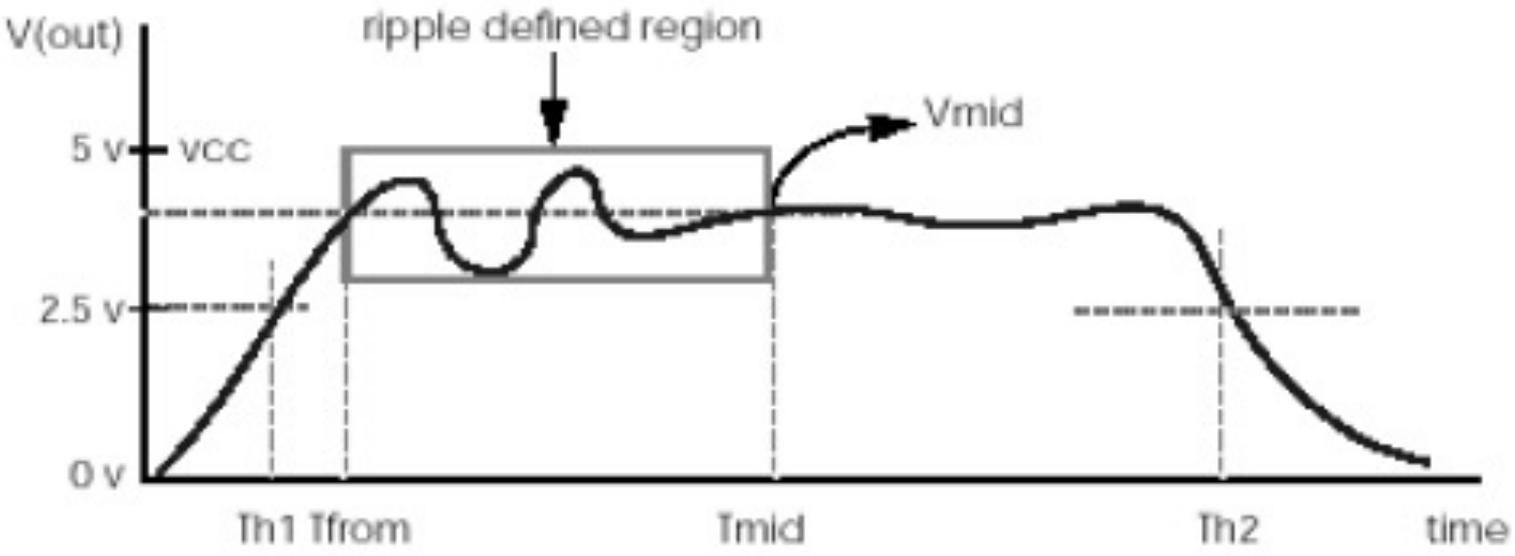
```
.meas TRAN Vmax MAX v(out) FROM=TDval TO=Tstop
.meas TRAN Vmin MIN v(out) FROM =TDval TO =Tstop
.meas TRAN Trise TRIG v(out) VAL='Vmin+0.1*Vmax' TD=TDval RISE=1
+ TARG v(out) VAl='0.9*Vmax' RISE=1
.meas TRAN Tfall TRIG v(out) VAL='0.9*Vmax' TD=TDval FALL=2
+ TARG v(out) VAl='Vmin+0.1*Vmax' FALL=2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



(12). MEASURE Statement : Application Examples(Cont.)

- Ripple Calculation :

```
.meas TRAN Th1 WHEN v(out)='0.5*v(Vdd)' CROSS=1
.meas TRAN Th2 WHEN v(out)='0.5*v(Vdd)' CROSS=2
.meas TRAN Tmid PARAM='(Th1+Th2)/2'
.meas TRAN Vmid FIND v(out) AT='Tmid'
.meas TRAN Tfrom WHEN v(out)='Vmid' RISE=1
.meas TRAN Ripple PP v(out) FROM='Tfrom' TO='Tmid'
```



contents

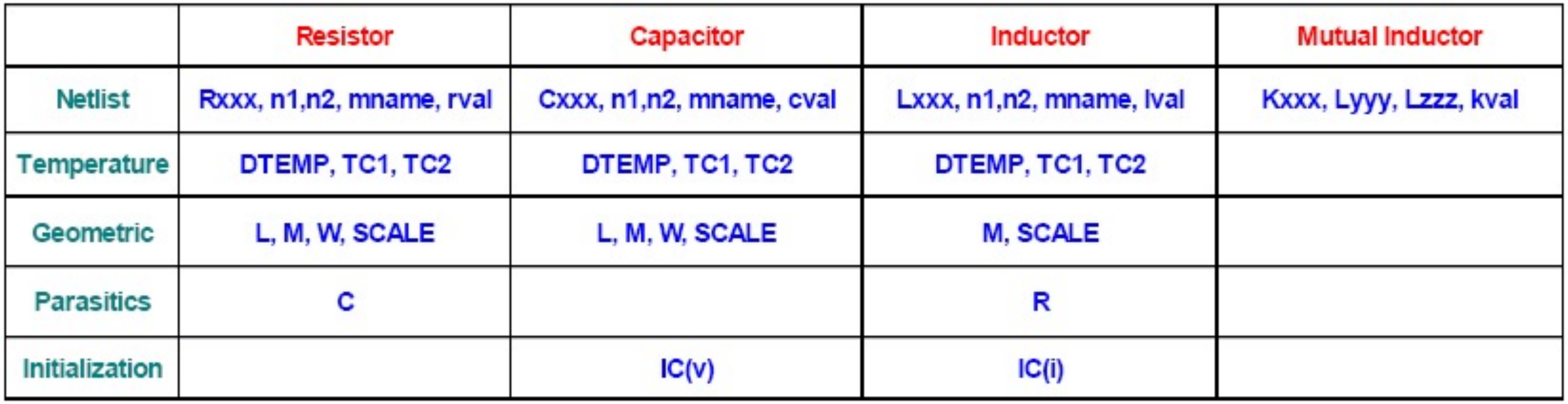
- SPICE Overview
- Simulation Input and Controls
- Sources and Stimuli
- Analysis Types
- Simulation Output and Controls
- **Elements and Device Models**
- Optimization
- Control Options & Convergence
- Applications Demonstration

(1). Types of Elements:

- Passive Devices :
  - R ---- Resistor
  - C ---- Capacitor
  - L ---- Inductor
  - K ---- Mutual Inductor
- Active Devices :
  - D ---- Diode
  - Q ---- BJT
  - J ---- JFET and MESFET
  - M ---- MOSFET
- Other Devices :
  - Subcircuit (X)
  - Behavioral (E,G,H,F,B)
  - Transmission Lines (T,U,O)

(2). Passive Devices : R, C, L, and K Elements

- Passive Devices Parameters :



	Resistor	Capacitor	Inductor	Mutual Inductor
Netlist	Rxxx, n1,n2, mname, rval	Cxxx, n1,n2, mname, cval	Lxxx, n1,n2, mname, lval	Kxxx, Lyyy, Lzzz, kval
Temperature	DTEMP, TC1, TC2	DTEMP, TC1, TC2	DTEMP, TC1, TC2	
Geometric	L, M, W, SCALE	L, M, W, SCALE	M, SCALE	
Parasitics	C		R	
Initialization		IC(v)	IC(i)	

- Examples :

R1 12 17 1K TC1=1.3e-3 TC2=-3.1e-7

C2 7 8 0.6pf IC=5V

LSHUNT 23 51 10UH 0.01 1 IC=15.7mA

K4 Laa Lbb 0.9999

(3). Active Device : MOSFET Introduction

- MOSFET Model Overview :
  - MOSFET Defined by :
    - (1). MOSFET Model & Element Parameters
    - (2). Two Submodel : CAPOP & ACM
  - ACM : Modeling of MOSFET Bulk_Source & Bulk_Drain Diodes
  - CAPOP : Specifies MOSFET Gate Capacitance
- MOSFET Model Levels :
  - Available : All the public domain spice model
  - Level = 4 or 13 : BSIM1
  - Modified BSIM1
  - Level = 5 or 39 : BSIM2
  - Level = 49 : BSIM3.3
  - Level = 8 : SBT MOS8

(4). MOSFET Introduction : Element Statement

- MOSFET Element Syntax :

```
Mxxx nd ng ns <nb> mname <L=val> <W=val> <AD=val> <AS=val>
+ <PD=val> <PS=val> <NRD=val>
+ <NRS=val>
+ <OFF> <IC=vds,vgs,vbs> <M=val>
+ <TEMP=val> <GEO=val> <DELVTO=val>
```

- MOSFET Element Statement Examples:

```
M1 24 2 0 20 MODN L=5u W=100u M=4
M2 1 2 3 4 MODN 5u 100u
M3 4 5 6 8 N L=2u W=10u AS=100P AD=100p PS=40u PD=40u
.OPTIONS SCALE=1e-6
M1 24 2 0 20 MODN L=5 W=100 M=4
```

(5). MOSFET Introduction : Model Statement

- MOSFET Model Syntax :

```
.MODEL mname NMOS <LEVEL=val> <name1=val1> <name2=val2>..........
.MODEL mname PMOS <LEVEL=val> <name1=val1> <name2=val2>..........
```

- MOSFET Model Statement Examples:

```
.MODEL MODP PMOS LEVEL=2 VTO=-0.7 GAMMA=1.0......
.MODEL NCH NMOS LEVEL=39 TOX=2e-2 UO=600..........
```

- Corner_LIB of Models:

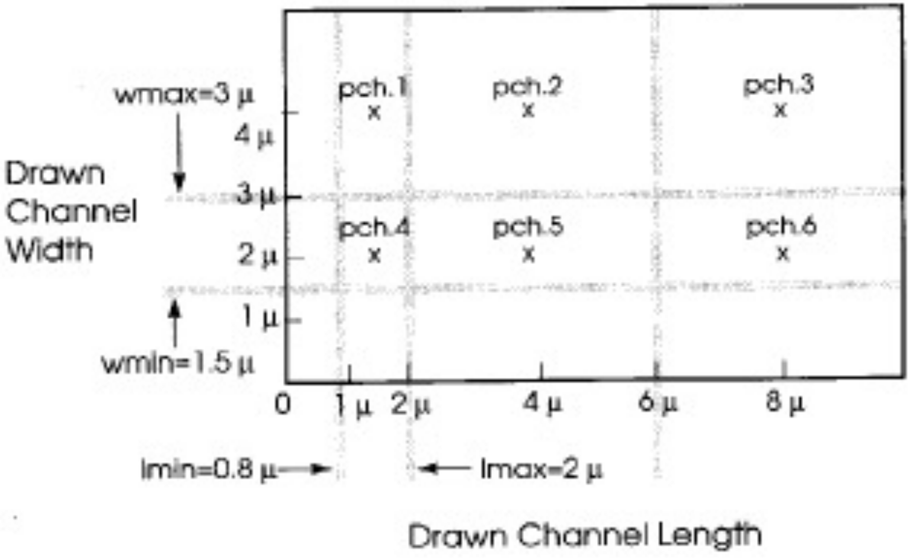
```
.LIB TT or (FF|SS|FS|SF)
.param toxn=0.0141 toxp=0.0148......
.lib '~/simulation/model/cmos.l' MOS
.ENDL TT or (FF|SS|FS|SF)
```

```
.LIB MOS
.MODEL NMOD NMOS (LEVEL=49
+ TOXM=toxn LD=3.4e-8 , ......)
.ENDL MOS
```

(6). MOSFET Introduction : Automatic Model Selection

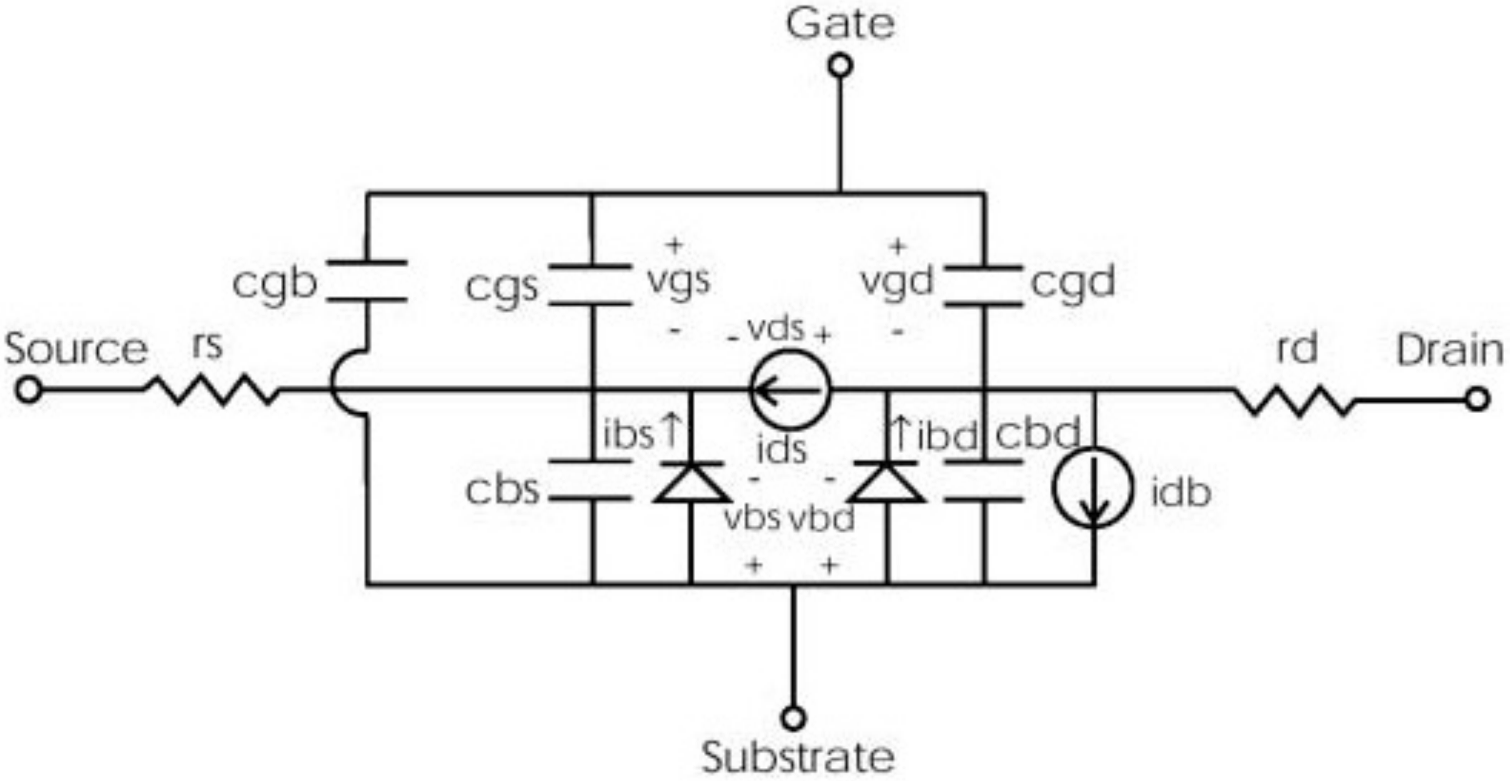
- Automatic Model Selection :
  - HSPICE can Automatically Find the Proper Model for Each Transistor Size by Using Parameters, LMIN,LMAX,WMIN, & WMAX in MOSFET Models

```
.MODEL pch.4 PMOS WMIN=1.5u WMAX=3u LMIN=0.8u LMAX=2.0u
.MODEL pch.5 PMOS WMIN=1.5u WMAX=3u LMIN=2.0u LMAX=6.0u
M1 1 2 3 4 pch W=2u L=4u $ Automatically Select pch.5 Model
```



(7). MOSFET Introduction : Equivalent Circuits

- MOSFET Equivalent Circuit for Transient Analysis:



(8). MOSFET Transistor Basics : Higher-Order Effects

- Geometry and Doping Effects on Vth :
  - Short Channel Effect (Small L)
  - Narrow Channel Effect (Small W)
  - Non-Uniform Doping Effect
- Physical Effects on Output Resistance :
  - Channel Length Modulation (CLM)
  - Substrate Current Induced Body Effects (SCBE)
- Other Physical Effects :
  - Channel Mobility Degradation
  - Carrier Drift Velocity
  - Bulk Charge Effect
  - Parasitic Resistance
  - Subthreshold Current

(9). MOSFET Models : Historical Evolution

- Can Define Three Clear Model “Generations”
- First Generation :
  - “Physical” Analytical Models
  - Geometry Coded into the Model Equations Level 1, Level 2, & Level 3
- Second Generation :
  - Shift in Emphasis to Circuit Simulation
  - Extensive Mathematical Conditioning
  - Individual Device Parameters & Separate Geometry Parameter
  - Shift “Action” to Parameter Extraction (Quality of Final Model is Heavily Dependent on Parameter Extraction)
  - BSIM1, Modified BSIM1, BSIM2

(10). MOSFET Models : Historical Evolution

- Third Generation :
  - “Original Intent” was a Return to Simplicity
  - Scalable MOSFET model
  - 1-st derivative is continuous
  - Attempt to Re-Introduce a Physical Basis While Maintaining “Mathematical Fitness”
  - BSIM3, MOS-8, Other ???

(11). Overview of Most Popular MOSFET Models :

- UCB Level 1 : (Level = 1)
  - Shichman-Hodges Model (1968)
  - Simple Physical Model, Applicable to L> 10um with Uniform Doping
  - Not Precise Enough for Accurate Simulation
  - Use only for Quick, Approximate Analysis of Circuit Performance
- UCB Level 2 : (Level = 2)
  - Physical/Semi-Empirical Model
  - Advanced Version of Level 1 which Includes Additional Physical Effects
  - Applicable to Long Channel Device (~ 10 um)
  - Can Use either Electrical or Process Related Parameters

SPICE : Simulation Program with Integrated Circuit Emphasis

UCB : University of California at Berkeley

(11). Overview of Most Popular MOSFET Models(Cont.) :

- UCB Level 3 : (Level = 3)
  - Semi-Empirical Model Model (1979)
  - Applicable to Long Channel Device (~ 2um)
  - Includes Some New Physical Effects (DIBL, Mobility Degradation by Lateral Field)
  - Very successful Model for Digital Design (Simple & Relatively Efficient)
- BSIM : (Level = 13)
  - First of the “Second Generation” Model (1985)
  - Applicable to Short Channel Device with L~ 1.0um
  - Empirical Approach to Small Geometry Effects
  - Emphasis on Mathematical Conditioning of Circuit Simulation

BSIM : Berkeley Short-Channel IGFET Model

(11). Overview of Most Popular MOSFET Models (Cont.) :

- Modified BSIM1 LEVEL 28 :
  - Enhanced Version of BSIM 1, But Addressed most of the Noted Shortcomings
  - Empirical Model Structure --> Heavy Reliance on Parameter Extraction for Final Model Quality
  - Applicable to Deep Submicron Devices (~ 0.3 - 0.5um)
  - Suitable for Analog Circuit Design
- BSIM 2 : (HSPICE Level = 39)
  - "Upgraded" Version of BISM 1 (1990)
  - Applicable to Devices with (L~ 0.2um)
  - Drain Current Model has Better Accuracy and Better Convergence Behavior
  - Covers the Device Physics of BSIM 1 and Adds Further Effects on Short Channel Devices

(11). Overview of Most Popular MOSFET Models (Cont.) :

- **EKV Model :**
  - Developed at Swiss Federal Institute of Technology in Lausanne (EPFL)
  - A Newly “Candidate” Model for Future Use
  - Description of Small Geometry Effects is Currently Being Improved
  - Developed for Low Power Analog Circuit Design
  - Fresh Approach to FET Modeling
    - Use Substrate (not Source) as Reference
    - Simpler to Model FET as a Bi-Directional Element
    - Can Treat Pinch-Off and Weak Inversion as the same Physical Phenomenon
  - First “Re-Thinking” of Analytical FET Modeling Since Early 1960s.

(12). MOSFET Model Comparison :

- **Model Equation Evaluation Criteria : (Ref: HSPICE User Manual 1996, Vol._II)**
  - Potential for Good Fit to Data
  - Ease of Fitting to Data
  - Robustness and Convergence Properties
  - Behavior Follows Actual Devices in All Circuit Conditions
  - Ability to Simulate Process Variation
  - Gate Capacitance Modeling
- **General Comments :**
  - Level 3 for Large Digital Design
  - HSPICE Level 28 for Detailed Analog/Low Power Digital
  - BSIM 3v3 & MOS Model 9 for Deep Submicron Devices
  - All While Keeping up with New Models

contents

- SPICE Overview
- Simulation Input and Controls
- Sources and Stimuli
- Analysis Types
- Simulation Output and Controls
- Elements and Device Models
- Optimization
- Control Options & Convergence
- Applications Demonstration

(1). SPICE Optimization

- Circuit Level Goal Optimization:
  - A procedure for automatic searching instance parameters to meet design goal
  - Can be applied for both .DC , .AC and .TRAN analysis
  - Optimization implemented in SBTSPICE can optimize one goal
  - Optimization implemented in HSPICE can optimize multi-goal circuit parameter/device model parameter
  - The parameter searching range must differentiate the optimization goal

(2). Optimization Preliminaries

- Circuit Topology Including Elements and Models
- List of Element to be Optimized
  - Initial Guess, Minimum, Maximum
- .Measure Statements for Evaluating Results
  - Circuit Performance Goals
  - Selection of Independent or Dependent Variables Measurement Region
- Specify Optimizer Model

(3). Optimization Syntax : General Form

- Variable Parameters and Components :

```
.PARAM parameter = OPTxxx (init, min, max)
```

- Optimizer Model Statement :

```
.MODEL method_namd OPT <Parameter = val .....>
```

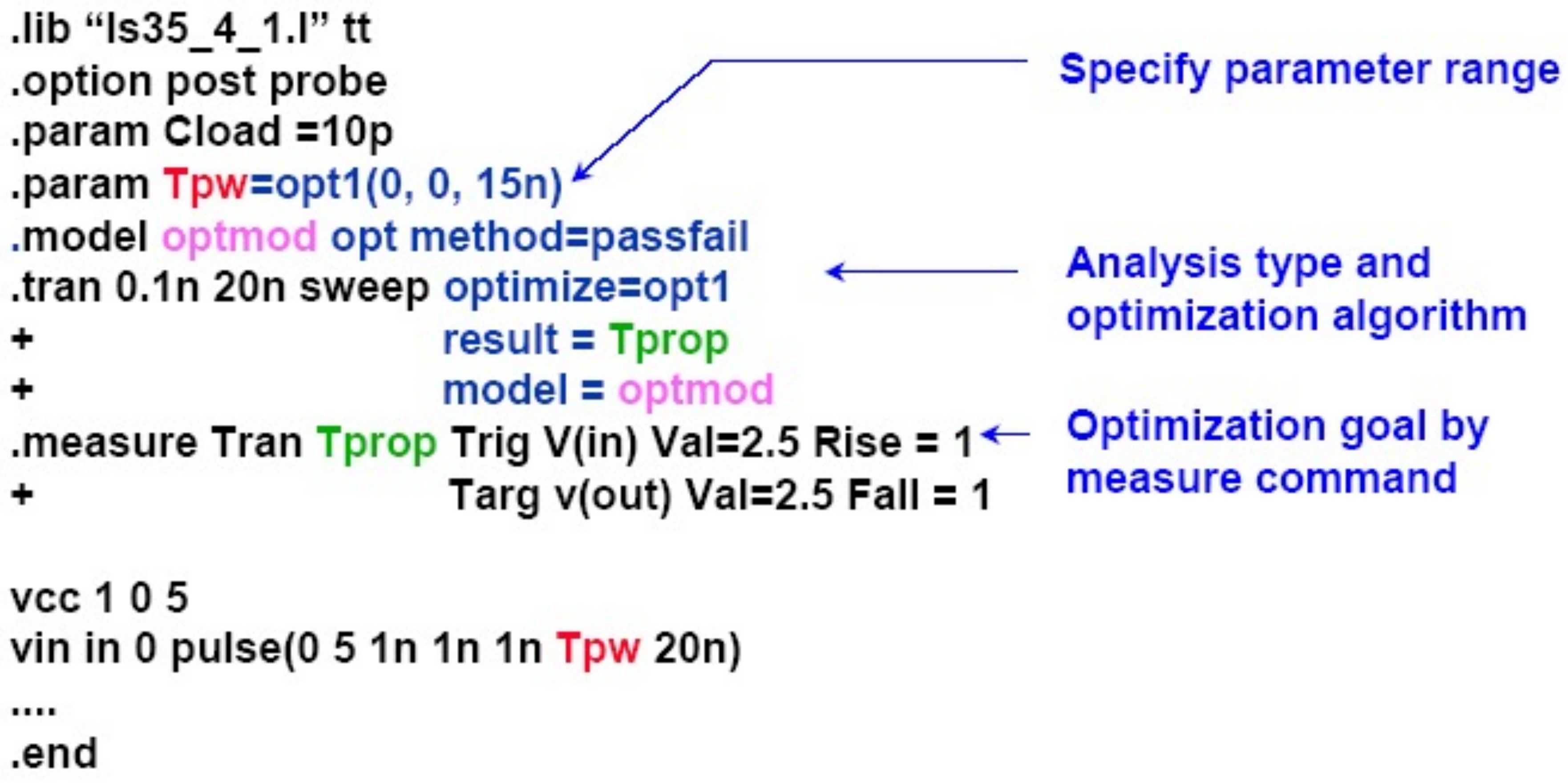
- Analysis Statement Syntax :

```
.DC|AC|TRAN .......<DATA=filement > SWEEP OPTIMIZE = OPTxxx
+ Results = meas_name MODEL = method_name
```

- Measure Statement Syntax :

```
.MEASURE meas_name .......<GOAL=val> <MINVAL=val>
```

(4). Optimization Example



```
.lib “ls35_4_1.l” tt
.option post probe
.param Cload =10p
.param Tpw=opt1(0, 0, 15n)
.model optmod opt method=passfail
.tran 0.1n 20n sweep optimize=opt1
+                     result = Tprop
+                     model = optmod
.measure Tran Tprop Trig V(in) Val=2.5 Rise = 1
+                   Targ v(out) Val=2.5 Fall = 1

vcc 1 0 5
vin in 0 pulse(0 5 1n 1n 1n Tpw 20n)
....
.end
```

Specify parameter range

Analysis type and optimization algorithm

Optimization goal by measure command