# Getting Started with HSPICE

# Topics

- Starting HSPICE
- Netlist file structure
- HSPICE output files
- Using components, sources and subcircuits to construct a netlist
- HSPICE analysis types
- Simulation controls and options

# Starting HSPICE

- 32-bits
  - Unix only:
    - hspice design.sp > design.lis
  - Unix and Windows:
    - hspice –i design.sp –o design.lis
- 64-bits
  - Unix only:
    - hspice64 design.sp > design.lis
    - hspice –i design.sp –o design.lis

# The output list file

- .lis contains results of
  - .PRINT
  - .OP
  - .OPTIONS

# Files and suffixes

| HSPICE Input | |
|---|---|
| Input netlist | .sp |
| Model libraries | .inc, .lib |
| **HSPICE Output** | |
| Run Status | .st0 |
| Output listing | .lis |
| Analysis data, transient | .tr# (e.g. .tr0) |
| Analysis data, dc | .sw# (e.g. .sw0) |
| Analysis data, ac | .ac# (e.g. .ac0) |
| FFT Analysis data | .ft# (e.g. .ft0) |
| Measure output | .m*# (e.g. .mt0) |
| **Waveform Viewer Input** | |
| All analysis data and measure output | |

# Netlist Structure

| | |
|---|---|
| Title | `First line is always the title` |
| Comment character | `* - comment for a line`<br>`$ - comment after a command` |
| Options | `.option post` |
| Print/Probe/Analysis | `.print v(d) i(rl)`<br>`.probe v(g)`<br>`.tran .1n 5n` |
| Initial Conditions | `.ic v(b) = 0 $ input state` |
| Sources | `Vg g 0 pulse 0 1 0 0.15 0.15 0.42`<br>`* example of a voltage source` |
| Circuit Description | `MN d g gnd n nmos`<br>`RL vdd d 1K` |
| Models | `.model n nmos level = 49`<br>`+ vto = 1 tox = 7n`<br>`* '+' continuation character` |
| End | `.end $ terminates the simulation` |

# Netlist structure

- A netlist consists of one main file and one or more optional added files
- Some commands can be used to structure the netlist:
  - Add a file to your netlist

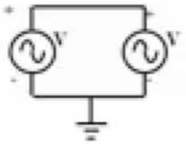
    ```
    .INCLUDE 'filename'
    ```
  - Add a library to the netlist

    ```
    .LIB 'filename' libname
    ```
  - Assign a parameter value

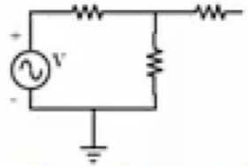
    ```
    .PARAM  res=1
    .PARAM  res2='res+1'
    R1 n1 n2 'res2'
    ```
  - Alter the netlist or parameters and run more simulations

    ```
    .ALTER
    .DATA
    ```
- Elements in a netlist are order independent
  - For Parameters and Options the last definition is used

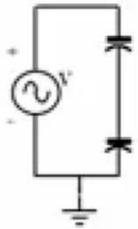
# Netlist Topologies to avoid



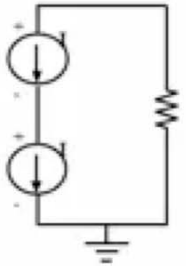
No voltage loops



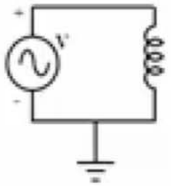
No dangling nodes



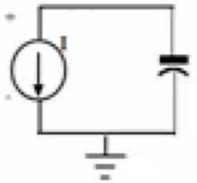
Every node must have a DC path to ground



No stacked current sources



No ideal voltage source in closed inductor loop



No ideal current source in closed capacitor loop

# Node Naming Conventions

- Node Names:
  - Can be up to 1024 characters
  - Either names or numbers (e.g. n1, 33, in1, 100)
  - Numbers: 1 to 9999999999999999 (1 to 1e16)
  - Leading zeros in node names are ignored
  - Can begin with these characters: # _ ! %
  - 0 is ALWAYS ground
  - Global vs. local
  - Do not begin with a "/"
  - May contain:
    + - * / : ; $ # . [ ] ! < > _ %
  - May NOT contain: ( ) , = ' <space>
  - Ground may be either 0, GND, !GND or GROUND

# Element Naming Conventions

- Element Names:
  - Names must begin with an alphabetic character, but thereafter can contain numbers and the following characters:

    ! # $ % * + - / < > [ ] _
  - Names can be up to 1024 characters long
  - Names are not case sensitive
  - Element instances begin with the element key letter
  - Subcircuit instance names begin with X

# Parameter Naming Conventions

- Parameter Names:
  - Follow the name syntax rules except that names must begin with an alphabetic character
  - The other characters must be either a number, or one of these characters:

    ! # $ % [ ] _

# Units and Scale factors

- Element units:
  - R - ohm
  - C - Farad
  - L - Henry
- Technology Scaling:
  - SCALE and SCALM
  - ALL lengths and widths are in METERS
- Common conversion factors
  MIL(S) = 25.4e-6
  FT = .3048 (METERS)
  DB = 20log10

- Scale Factors
  A = 1e-18
  F = 1e-15
  P = 1e-12
  N = 1e-9
  U = 1e-6
  M = 1e-3
  K = 1e3
  MEG = X = 1e6
  G = 1e9
  T = 1e12

# Circuit Elements

- Passive components
  - R, C, L, etc.
- Sources
  - Independent sources
    - DC, AC, TRAN (pulse, pwl, pattern, pseudo random bit stream, sin, AM, exp, single frequency frequency modulated), Mixed, Digital
  - Dependent sources
    - G (voltage controlled current source), E (voltage controlled voltage source), F (current controlled voltage source), H (current controlled current source)
- Active Elements
  - D, M, Q, J (need a .MODEL)

# Subcircuits

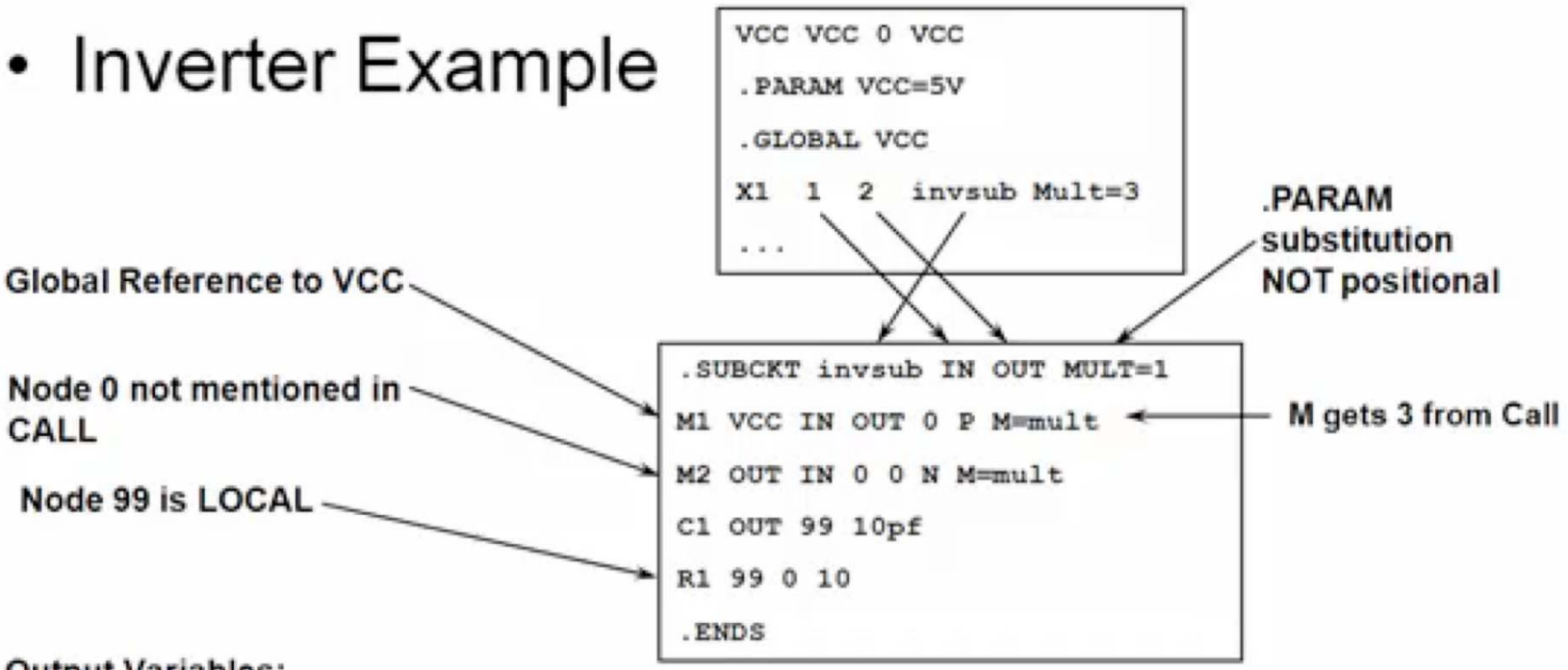
- Subcircuit definition syntax:

  `.SUBCKT subnam n1 n2 [n3 n4 … ] [parnam=val … ]`

  - **`subnam`** – Reference name for the subcircuit model call
  - **`n1, n2 …`** - Node numbers for external reference
    - Any element nodes appearing in the subcircuit, but not included in this list, are strictly local, EXCEPT
      - Ground node (0)
      - Nodes assigned using BULK (MOSFET) or SUBSTRATE (BJT)
      - Nodes assigned using the **`.GLOBAL`** statement
  - **`parnam=val`** - A parameter name set to a value
    - For use only in the subcircuit
    - Overridden by an assignment in the subcircuit call or by a value set in a .PARAM statement that is external to the subcircuit

# Subcircuit Example

- Inverter Example




Output Variables:

```
.PRINT   I(X1.M1)
.PRINT V(X1.99)
.PRINT V(1)   $ IN and OUT replaced by nodes 1 and 2, respectively
.PRINT TRAN ISUB(X1.IN) $ PRINT subcircuit pin current
```

# Analysis Types

- DC Operating Point
- DC Analysis
- AC Analysis
- Transient Analysis
- Temperature Analysis

# Output Commands

- .PROBE:
  - Save output variables in analysis data file without additional output in the output listing file
  - With option probe, can limit *.tr# file size
- .PRINT:
  - Numeric analysis results printed to output listing file
- .MEASURE:
  - Print numeric results of measured electrical specifications for specified analysis

**Examples:**

```
.PROBE tran v(4) i(vin) par('v(out)/v(in)')
.PRINT AC VM(4,2) VP(8,3) VR(7) Ii(R1)
.PRINT LX8(m1)
```

- Print the drain-source conductance of element m1

# Limiting the size of the output data

- Specifying analysis data format:
  - `.OPTION POST` → Generate binary output waveform files
- Limiting the size of the analysis data file:
  - `.OPTION PROBE`
    - ALL nodes plotted by default
    - Limit data in analysis data file to that specified in **`.PRINT`** and **`.PROBE`** statements
  - `.OPTION INTERP`
    - Limit the number of points stored to step size specified in **`.TRAN`** statement
    - Pre-interpolates the output to the interval specified on the **`.TRAN`** statement

# Limiting the size of the output data

- Summary:

| Netlist Input File, *.sp | | Waveform File, *.tr0 |
|---|---|---|
| .option post | → | All nodes and points |
| .option post probe | → | Fewer nodes |
| .option post interp | → | Fewer points per node |

# Output Variables

- Five Groups of Output Variables:
  - DC and transient analysis
    - Displays individual nodal voltages, branch currents, element power dissipation
  - AC analysis
    - Displays imaginary and real components of nodal voltage, branch current
    - Displays phase and impedance parameters
  - Element templates
    - Displays element specific nodal voltages, branch currents, element parameters, and the derivatives of element voltage, current, or charge
  - .MEASURE
    - Displays user-defined variables as specified in the .MEASURE statement
  - Parametric statements
    - Displays mathematically, user-defined expressions operating on nodal voltages, etc

# Output Variables: Examples

- DC and Transient output examples:
  - Standard form is V(node) or I(element)
    - `v(1)`
      - Voltage at node 1
    - `i(Rin)`
      - Current through Rin (direction of I is n1 to n2)
    - `v(1,2)`
      - Voltage between node 1 and node 2 (differential)
  - Complex addressing
    - `i1(xinv1.m3)`
      - Drain current of transistor m3 in subcircuit inv1

# Output Variables: Examples

- DC and Transient output examples (cont.)

  **p(rload)**

  – Power dissipated in rload at point of analysis

  **p(m1)**

  – Power dissipated in transistor m1 at point of analysis

  **Power**

  – Total power dissipation output at point of analysis

  **v(x3.5)**

  – Voltage at INTERNAL node 5 of subckt x3

  **par('p(x1.m1)+p(x2.m2)')**

  – Sum of power in m1 of x1 and m2 of x2

  **i3(2:q2)**

  – Emitter current of q2 in second subcircuit called

# Output Variables: Examples

- AC analysis output examples:

  `vi(2)`

  – Imaginary voltage component at node 2

  `ip1(q4)`

  – The phase of the collector current in q4

  `vdb(2,8)`

  – The voltage ratio between node 2 and 8 in decibels

  `vp(4,6)`

  – The arctangent [vi(4,6)/vr(4,6)]

  `vp(6)`

  – Phase at node 6

# ACOUT option (1/2)

- ACOUT controls the AC output calculation method, for the difference in values of magnitude, phase, and decibels:
  - `ACOUT = 1` (default) selects HSPICE method, which calculates the difference of the magnitudes of the values
  - Real and imaginary

    $VR(N1,N2) = REAL\ [V(N1,0)] - REAL\ [V(N2,0)]$

    $VI(N1,N2) = IMAG\ [V(N1,0)] - IMAG\ [V(N2,0)]$
  - Magnitude

    $VM(N1,0) = [VR(N1,0)^2 + VI(N1,0)^2]^{0.5}$

    $VM(N2,0) = [VR(N2,0)^2 + VI(N2,0)^2]^{0.5}$

    $VM(N1,N2) = VM(N1,0) - VM(N2,0)$
  - Phase

    $VP(N1,0) = ARCTAN[VI(N1,0)/VR(N1,0)]$

    $VP(N2,0) = ARCTAN[VI(N2,0)/VR(N2,0)]$

    $VP(N1,N2) = VP(N1,0) - VP(N2,0)$
  - Decibel

    $VDB(N1,N2) = 20 \cdot LOG10(VM(N1,0)/VM(N2,0))$

# ACOUT option (2/2)

Berkeley SPICE method

- ACOUT = 0 selects SPICE method, which calculates the magnitude of differences
- Real and imaginary

  VR(N1,N2) = REAL[V(N1,0) - V(N2,0)]

  VI(N1,N2) = IMAG[V(N1,0) - V(N2,0)]
- Magnitude

  $VM(N1,N2) = [VR(N1,N2)^2 + VI(N1,N2)^2]^{0.5}$
- Phase

  VP(N1,N2) = ARCTAN[VI(N1,N2)/VR(N1,N2)]
- Decibel

  VDB(N1,N2) = 20·LOG10[VM(N1,N2)]

# ACOUT Example

```
.option ACOUT=1
v1 1 0 ac .5
v2 0 2 ac .5
r1 1 0 1k
r2 2 0 1k
.print ac Vm(1) vm(2) vm(1,2) vp(1,2)
```

- output:
  0.5 0.5 0 180
- If we change to .option ACOUT=0 then

  0.5 0.5 1 0

# Output Variables: Element Templates

- Element Templates:
  - Display element specific nodal voltages, branch currents, element parameters and the derivatives of element voltage, current, and charge

  **Examples:**

  ```
  .print tran lv16(m3)
  ```
  - Prints the effective drain conductance (1/rdeff)

There are over hundred element templates: check documentation

# Output variables: .MEASURE

- .MEASURE
  - Prints user-defined electrical specifications of a circuit
  - Used extensively in optimization
  - Has seven fundamental measurement modes, each with its own form:
    - Rise, fall, and delay
    - Average, RMS, min, max, and P-P
    - Find - When
    - Equation evaluation
    - Derivative evaluation
    - Integral evaluation
    - Relative error

# .MEASURE Examples

- Measure delay from 2nd rising edge of node 1 to 2nd falling edge of node 2:

```
.MEAS TRAN TDELAY TRIG V(1) VAL=2.5 TD=10ns RISE=2
+                 TARG V(2) VAL=2.5          FALL=2
```

- Measure the average voltage of node 10 between 10ns and 55ns:

```
.MEAS TRAN avgval AVG V(10) From=10ns To=55ns
```

- Measure the time of the 5th rise of node "osc_out" at 2.5v and report as "r5":

```
.MEAS TRAN r5 WHEN V(osc_out)=2.5v RISE=5
```

- Measure v(out) when v(in)=40m, then store and print variable "result":

```
.MEAS TRAN result FIND v(out) WHEN v(in)=40m
```

- Make a calculation on previously measured values:

```
.MEAS TRAN my_result PARAM='(result/avgval)'
```

# Output variables: Parametric Output

Parametric Variables are waveform vectors

- Parametric Output Variables:

```
.PRINT tran PAR('algebraic expression')
.PRINT tran out_var=PAR('algebraic expression')
```

**Examples:**

```
.PRINT dc conductance=PAR('i(m1)/v(22)')
.PROBE tran PAR('log10(i(xff4.m1))')
.PRINT tran vds=PAR(vds) vgs=PAR(vgs)
```

# Commonly used .OPTIONS

See the Command Reference for a complete list of options

**Listing**
ACCT
BRIEF
INGOLD
LIST
NODE
NOMOD
NUMDGT
OPTLST

**Interface**
ARTIST
PSF
CSDF
POST
PROBE

**Output**
CAPTAB
DCCAP
UNWRAP

**Error**
BADCHR
DIAGNOSTIC

**Convergence**
CONVERGE
DV
GMIN
GMINDC
GRAMP

**Model**
SCALE
SEARCH
TNOM

**MOSFETs**
SCALM
WL

**DC Matrix**
ITL1

**Transient Accuracy**
RUNLVL
ACCURATE

**Efficiency**
AUTOSTOP
NOELCK
NOTOP

**Time Step**
DELMAX
METHOD

# General Listing Options

`.option LIST`

- Prints a list of netlist elements, node connections and, parameter values
- Prints effective sizes of elements and key values
- Useful in diagnosing topology-related problems

`.option NODE`

- Prints a node connection table
- The nodal cross-reference table lists each node and all the elements connected to it
- Useful in diagnosing topology related non-convergence problems

`.option ACCT`

- Reports job accounting and runtime statistics at the end of the output listing
- Useful in observing simulation efficiency

`.option OPTS`

- Reports settings of all `.OPTIONS` in the listing file

# More listing file options

- Simulation listing can get large:
- . OPTION nomod
  suppress the printout of model parameters
- .OPTION BRIEF
  suppress the printout of the netlist
- But … if you have troubles, remove the nomod and brief options so you can see all information

# More listing file options

`.option INGOLD=value`

- Controls the format of numbers in printouts
  - 0 – engineering format (default)
  - 1 – mix of fixed and exponential format
  - 2 – exponential format

`.option NUMDGT=value`

- Number of significant digits printed for output variables
  - Range is between 1 to 10
  - Defaults is 4

# Output Control Options

- POST (syntax: .OPTION POST)
- PROBE (syntax: .OPTION PROBE)

Save simulation results for a waveform viewer

Limit the waveforms generated

- POSTTOP
  - Limits data written to waveform file to top-level nodes only
- Syntax

  `.OPTION POSTTOP`
- POSTLVL
  - Limits data written to waveform file that only a specified level of nodes signals will be saved
- Syntax

  `.OPTION POSTLVL=n`
  - `n=1` top level only
  - `n=0` all levels, i.e., POSTLVL option is turned off

# Netlist Options (1/2)

- GLOBAL SCALE
  - Scale element geometric instance parameters whose default unit is meters.
- SYNTAX

  `.option scale=value`

  Example:

```
.option scale=1e-6
M1 Vdd 10 20 0 mymodel L=1u w=1u   $ L=1e-12, W=1e-12
M2 Vdd 10 20 0 mymodel L=1 w=1     $ L=1e-6, W=1e-6
C1 Vdd 10 capmod L=0.1 W=0.1       $ L=1e-7, W=1e-7
R1 20 0 resmod L=1 W=1             $ L=1e-6, W=1e-6
```

# Netlist Options (2/2)

- LOCAL SCALE
  - Scale passive device values

**Examples:**

```
R1 1 0 1k scale=10    $ R1=10K
C1 5 0 1u scale=10    $ C1=10u
L1 10 0 1u scale=10   $ L1=10u
```

# Model Options (1/2)

- GLOBAL SCALM
  - Affects model parameters with units in METERS
  - Scales active devices (MOSFETs, diodes, JFETs etc.)
  - Passive devices are not affected by scalm
  - Local scalm takes precedence over global

```
.option scalm
```

- SYNTAX:

```
.option scalm=value
```

Example:

```
.option scalm=1e-6
```

# Model Options (2/2)

- LOCAL SCALM
  - Affects model parameters with the units of meters

  **Example:**

```
M1 Vdd 10 20 0 mymodel L=1u w=1u
.model mymodel nmos scalm=1e-6 level=2
+ tox=1e-1 kp=2.0e-5
```

  - Only model parameters tox will be scaled

```
tox=1e-7
```

  - L and W will not be affected

# Options for Simulation Efficiency

`.option AUTOSTOP`

- Stops a transient analysis in HSPICE after calculating all TRIG-TARG, FIND-WHEN, and FROM-TO measure functions
  - Special interest when testing corners

`.option NOELCK`

- Bypasses element checking to reduce pre-processing time

`.option NOTOP`

- Suppresses topology checks to increase the speed for pre-processing

**NOT RECCOMENDED !!!**

# Circuit Simulation with SPICE

# SPICE Simulation

- Introduction to SPICE
- DC Analysis
- Transient Analysis
- Subcircuits
- Optimization
- Power Measurement
- Design Corners

# Introduction

- **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis
  - Developed in 1970's at Berkeley
  - Many versions are available
  - HSPICE is a robust commercial industry standard
    - Has many enhancements that we will use
- Written in FORTRAN for punch-card machines
  - Circuits elements are called *cards*
  - Complete description is called a SPICE *deck*

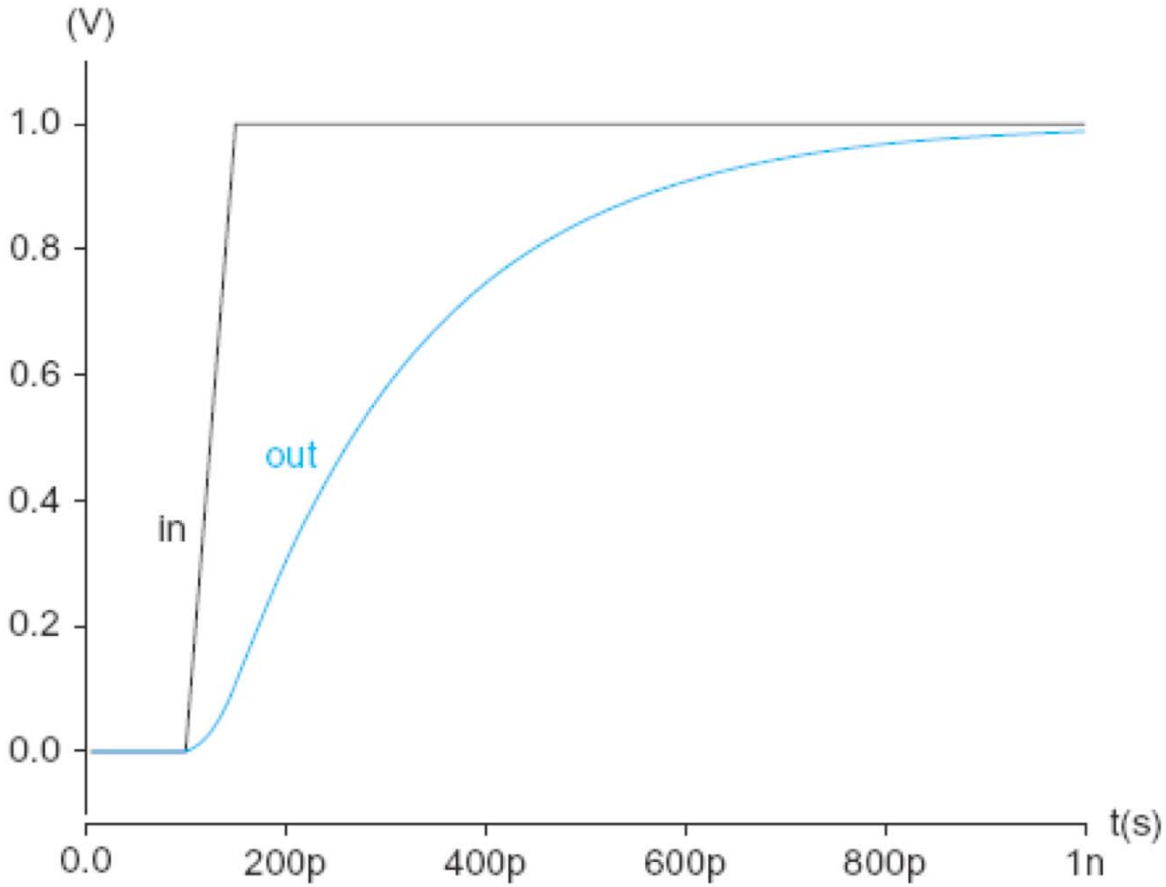
# Writing SPICE Decks

- Writing a SPICE deck is like writing a program
  - Plan: sketch schematic on paper or in editor
    - Whenever possible, “modify” existing decks
  - Code: strive for clarity
    - Start with name, date, purpose
    - Comment, comment, comment …
  - Drive the tool (avoid SPICE monkeying):
    - Predict what results should be
    - Compare with actual
    - *Garbage In, Garbage Out!*

# Example: A simple RC circuit

```
* rc.sp
* Created  David Harris@hmc.edu - 2/2/03,
* Modified ctalarico@ewu.edu - 9/17/11
* Find the response of RC circuit to rising input
*-------------------------------------------------
* Parameters and models
*-------------------------------------------------
.option post brief nomod
*-------------------------------------------------
* Simulation netlist
*-------------------------------------------------
Vin in    gnd  pwl  0ps 0 100ps 0 150ps 1.0 1ns 1.0
R1  in    out  2k
C1  out   gnd  100f
*-------------------------------------------------
* Analysis
*-------------------------------------------------
.tran 20ps 1ns
.plot v(in) v(out)
.end
```

# Result (Graphical)



# Sources (for digital design)

- DC Source

```
Vdd vdd gnd 2.5
```

- Piecewise Linear Source

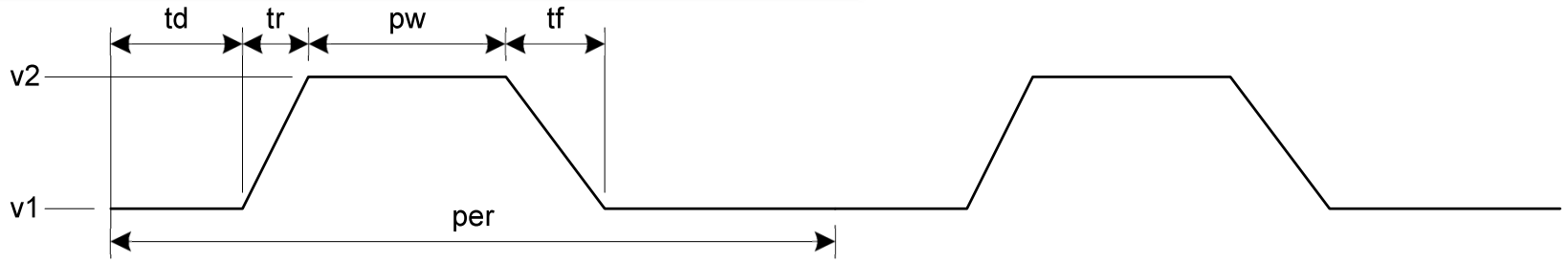
```
PWL t1 v1 t2 v2 …
```

```
Vin in gnd pwl 0ps 0 100ps 0 150ps 1.0 1ns 1.0
```

- Pulsed Source

```
Vck clk gnd PULSE 0 1.0 0ps 100ps 100ps 300ps 800ps
```

```
PULSE v1 v2 td tr tf pw per
```



# 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 |

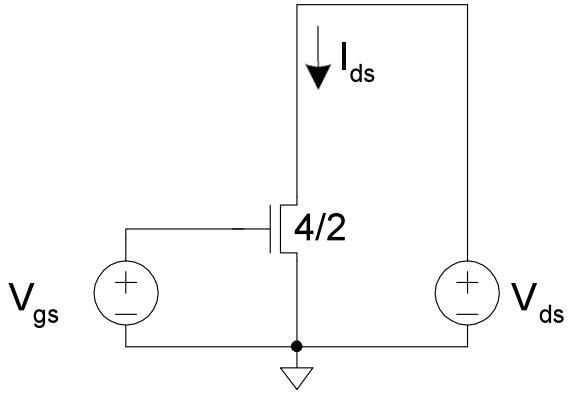
# SPICE Units

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

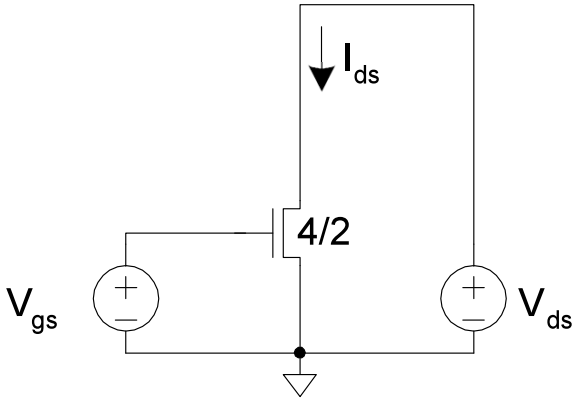
- Example:
  100 femtoFarad capacitor = 100fF = 100f =100e-15

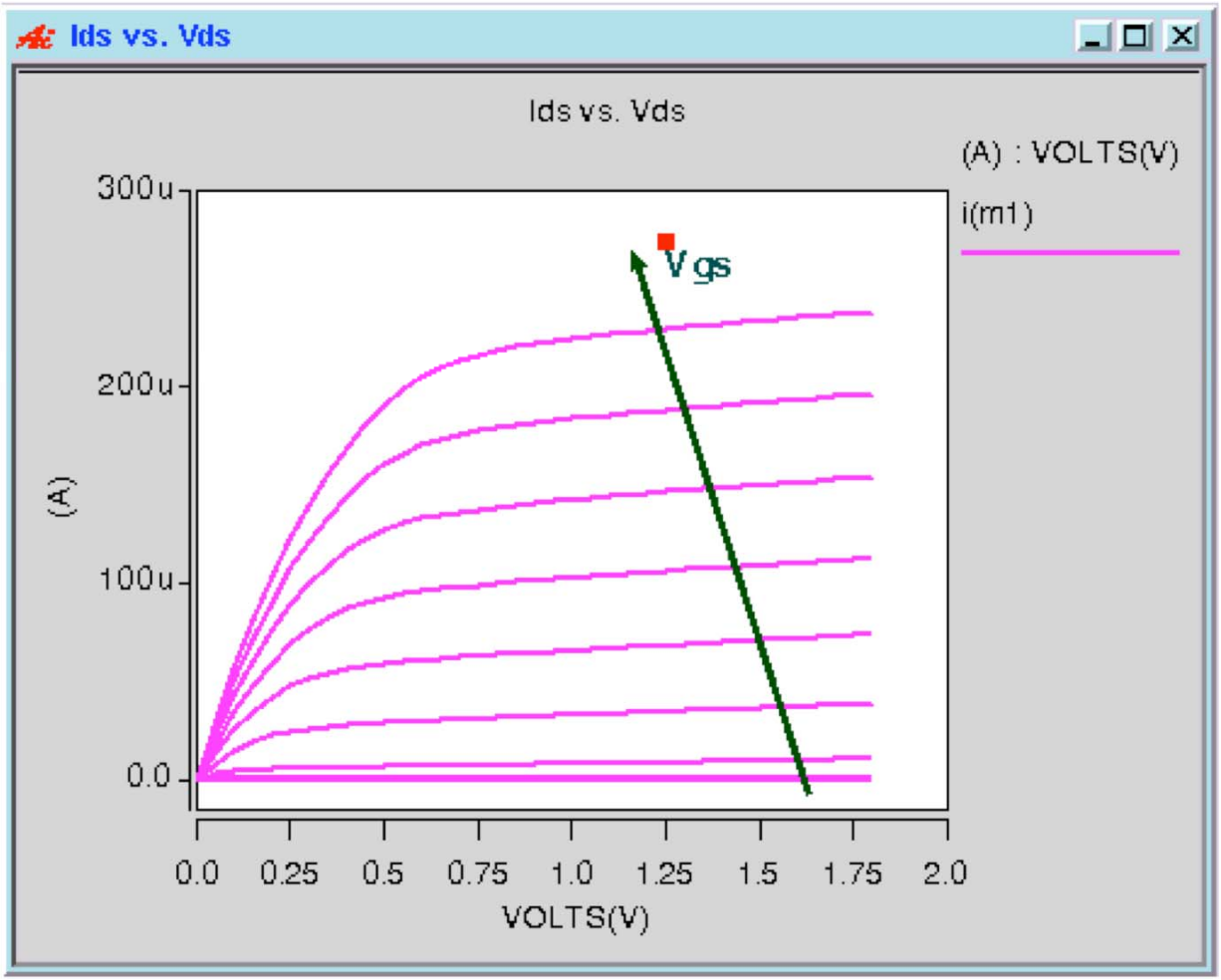
# DC Analysis

```
* mosiv.sp
* nMOS I-V Characteristics

*-----------------------------------------------
* Parameters and models
*-----------------------------------------------
.include '../mosistsmc180/mosistsmc180.sp'
.temp 70
.option post brief nomod

*-----------------------------------------------
* Simulation netlist
*-----------------------------------------------
*nmos
Vgs  g gnd 0
Vds  d gnd 0
M1   d g gnd gnd NMOS W=360n L=180n

*-----------------------------------------------
* Analysis
*-----------------------------------------------
.dc Vds 0 1.8 0.05 SWEEP Vgs 0 1.8 0.2
.probe i(m1)
.end
```



# nMOS I-V Characteristics (Result)



# 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>
```

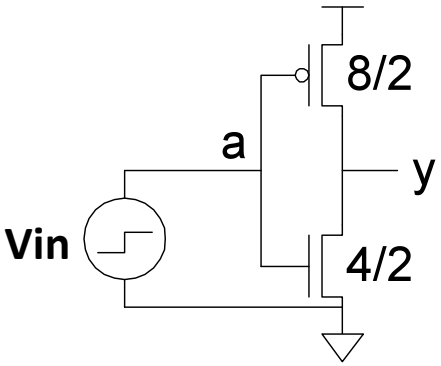
# Transient Analysis

```
* inv.sp
*-----------------------------------------------
* Parameters and models
*-----------------------------------------------
.param SUPPLY=1.8
.option scale=90n
.include '../mosistsmc180/mosistsmc180.sp'
.temp 70
.option post brief nomod

*-----------------------------------------------
* Simulation netlist
*-----------------------------------------------
Vdd vdd gnd 'SUPPLY'
Vin a gnd DC 0 PULSE 0 'SUPPLY' 50ps 1ps 1ps 99ps 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

*-----------------------------------------------
* Analysis
*-----------------------------------------------
.tran 0.1ps 250ps
.end
```

λ=90n

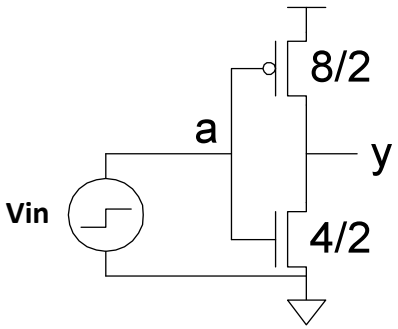
W=360n; L=180n

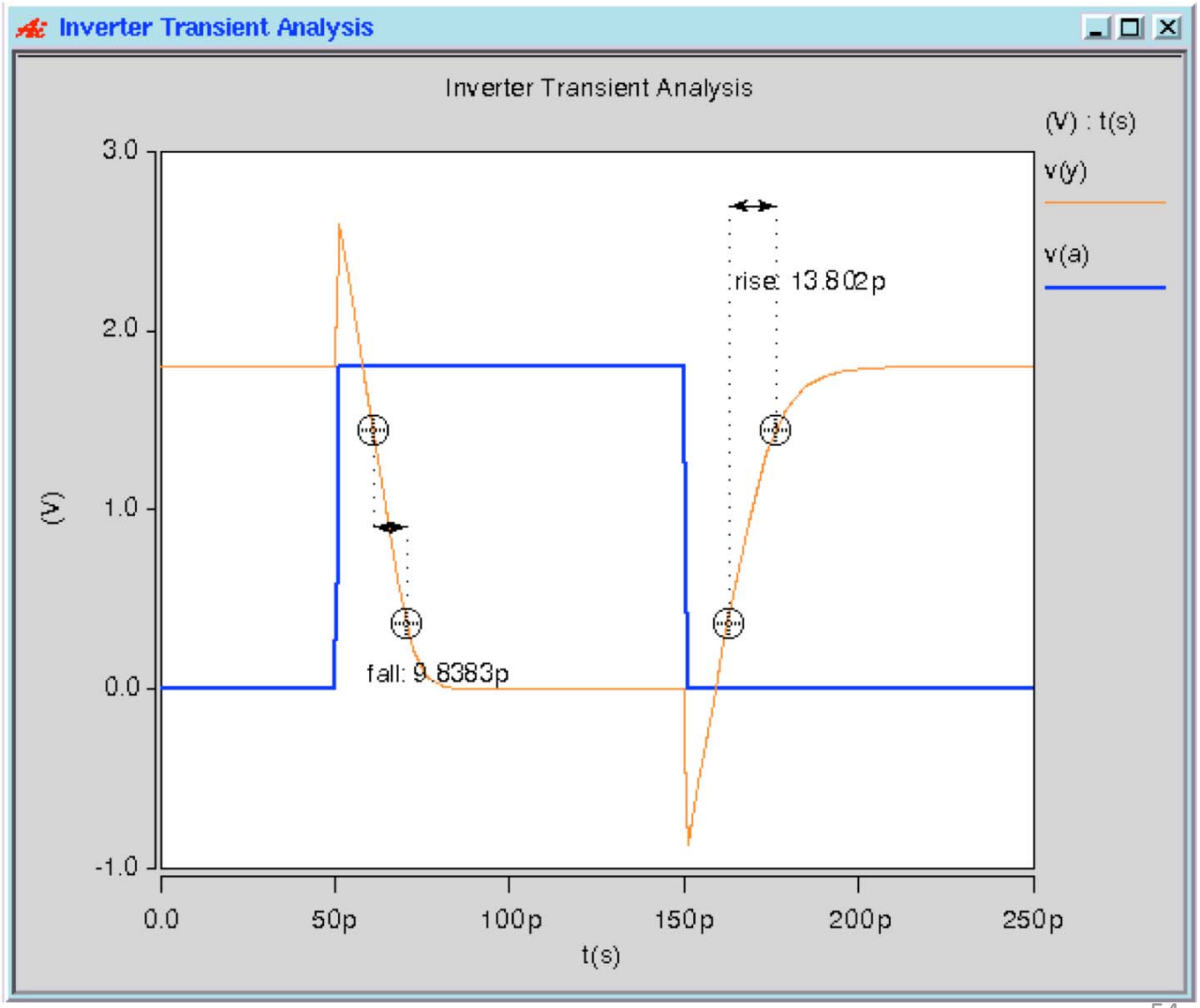
W=720n; L=180n



# Transient Analysis Results

- Unloaded inverter with fast edges
  - Overshoot



# MOSFET Junction Capacitances

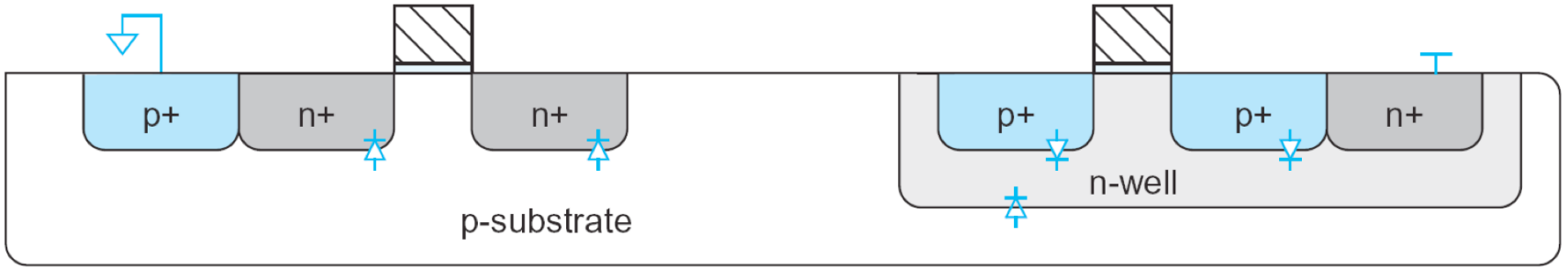
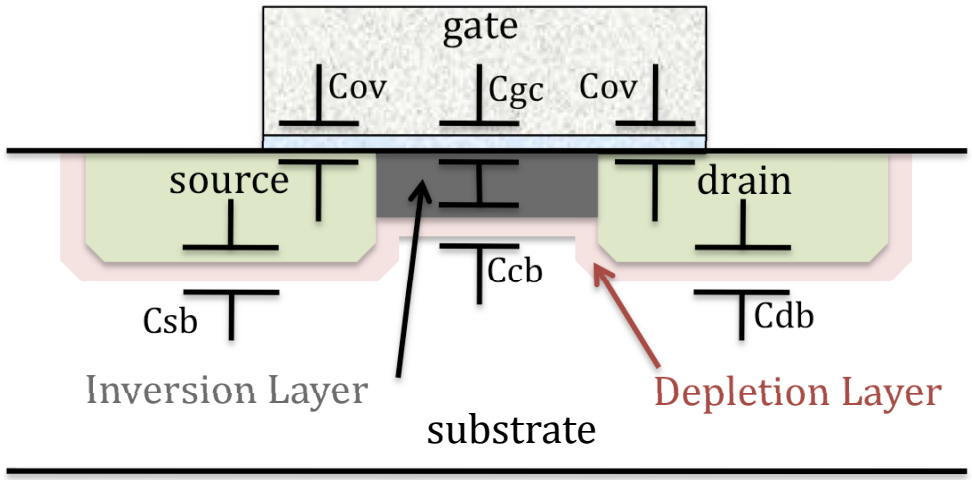
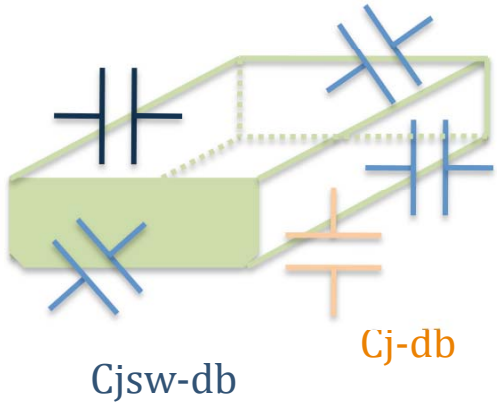



**FIGURE 2.22** Substrate to diffusion diodes in CMOS circuits



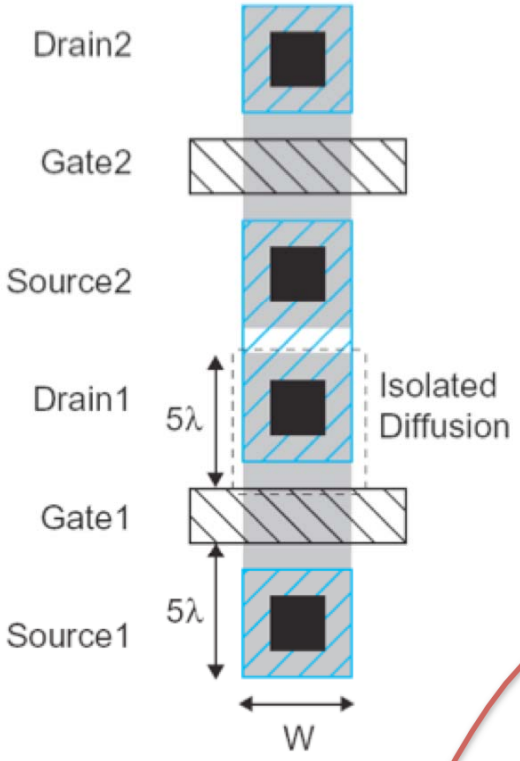




$$Cdb = AD \times Cj\text{-}db + PD \times Cjsw\text{-}db$$

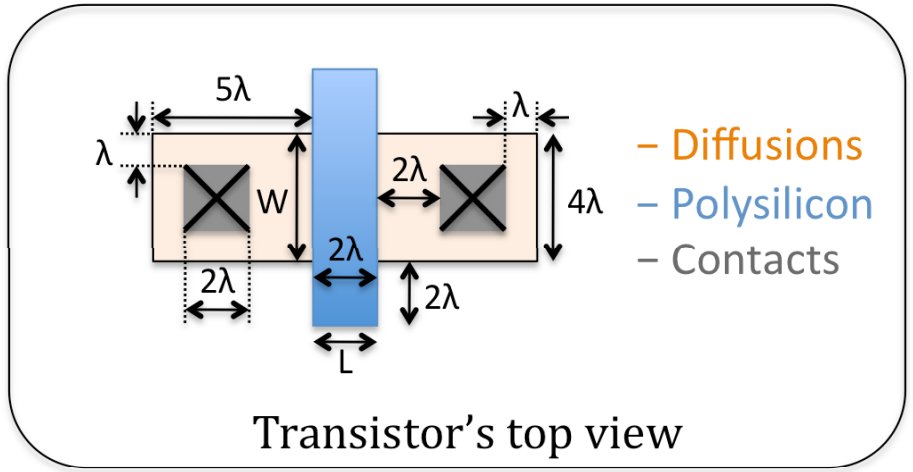
bottom junction cap. per area

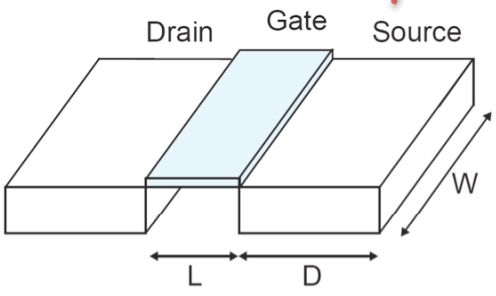
Side wall junction cap. per perimeter

# Area and Side Wall Perimeter of the diffusion regions




```
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.subckt inv a y N=4 P=8
```



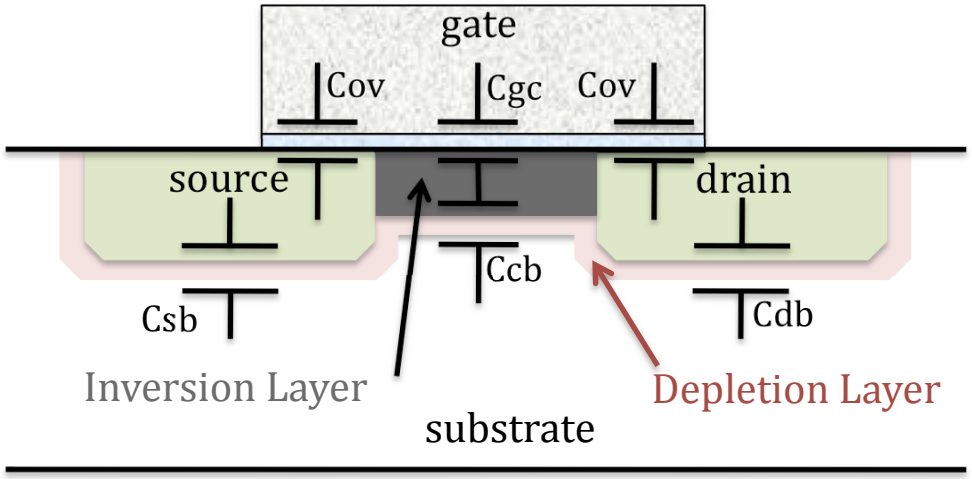

Make sure to check MOS model LEVEL and ACM

AD=AS = W×D; PD=PS = 2W+2D

AD=AS=5$W\lambda^2$; PD=PS=(2$W$+10)λ

Don't forget we are using .SCALE

# MOSFET Capacitances




Cgc modeled through Cgs and Cgd

Ccb modedel through Cgb

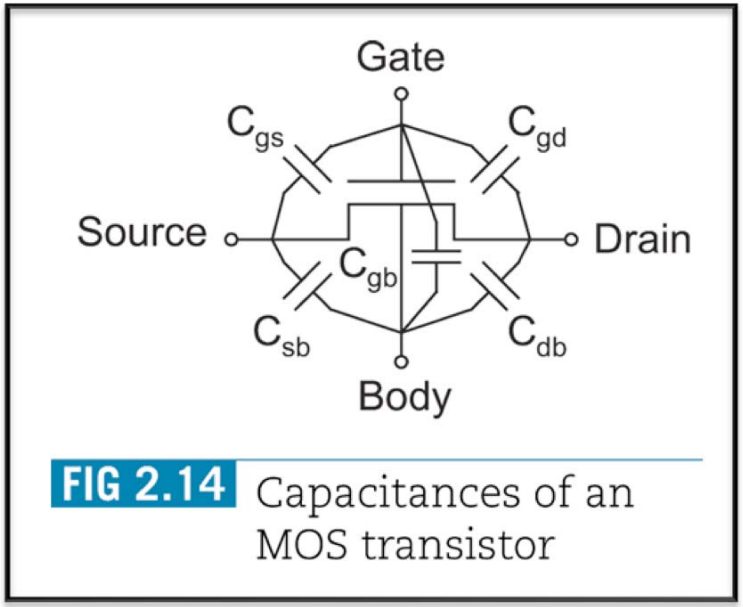
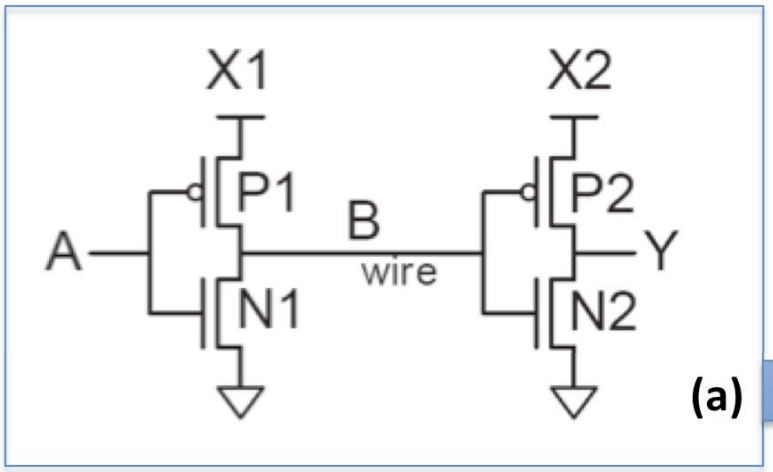
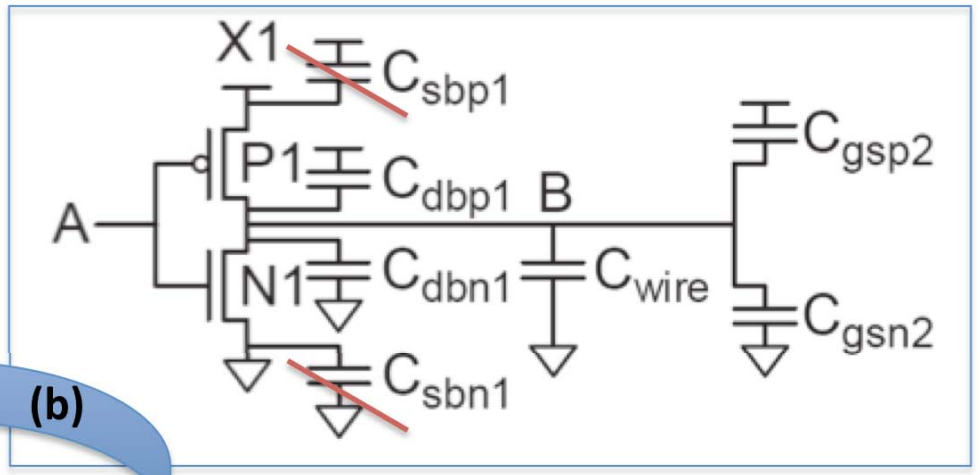
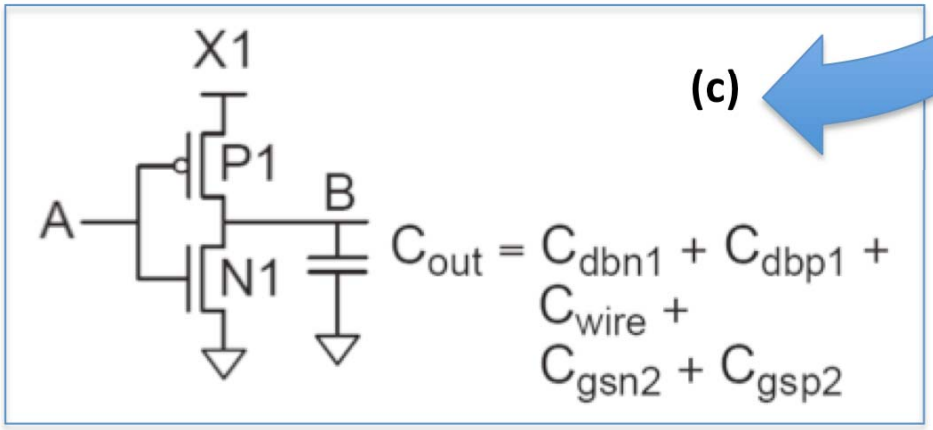



**FIG 2.14** Capacitances of an MOS transistor

# Inverter Delay Calculation (1/3)




**Csb shorted, Cgb negligible**
**What about Cgd ?**

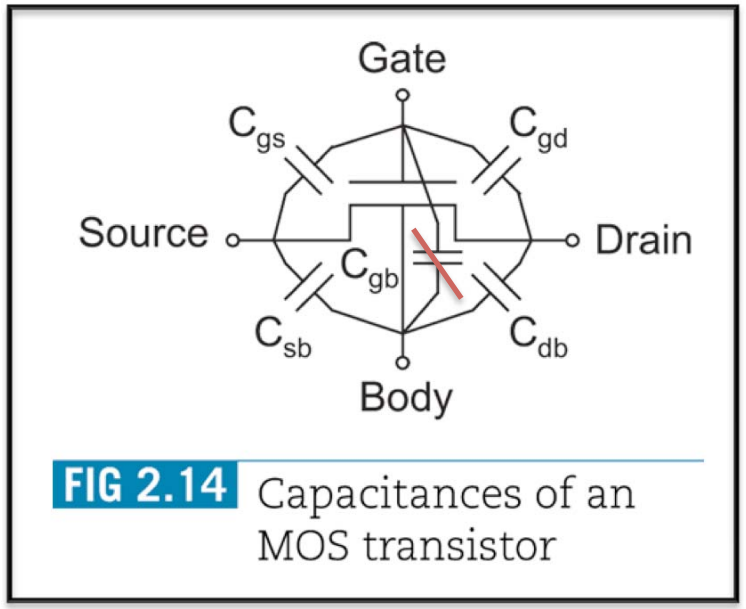
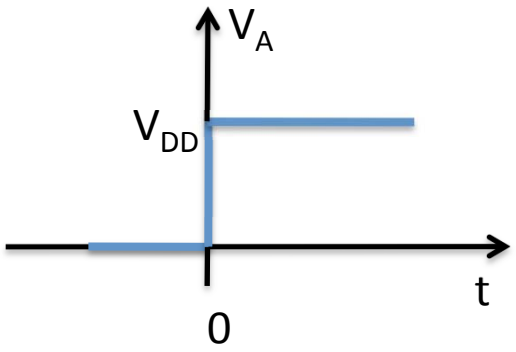
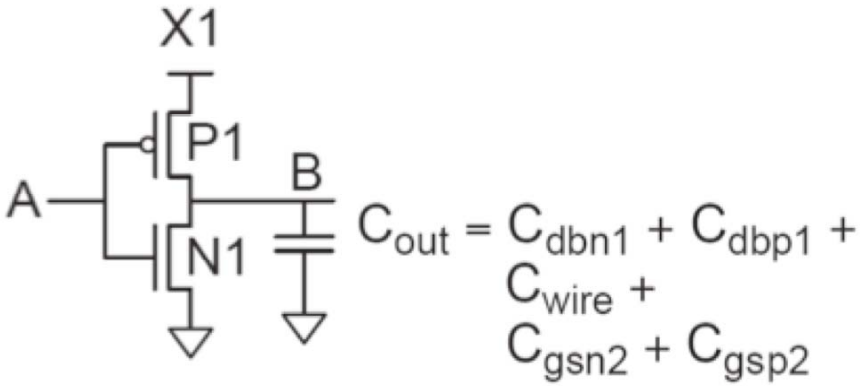



**FIG 2.14** Capacitances of an MOS transistor

# Inverter Delay Calculation (2/3)

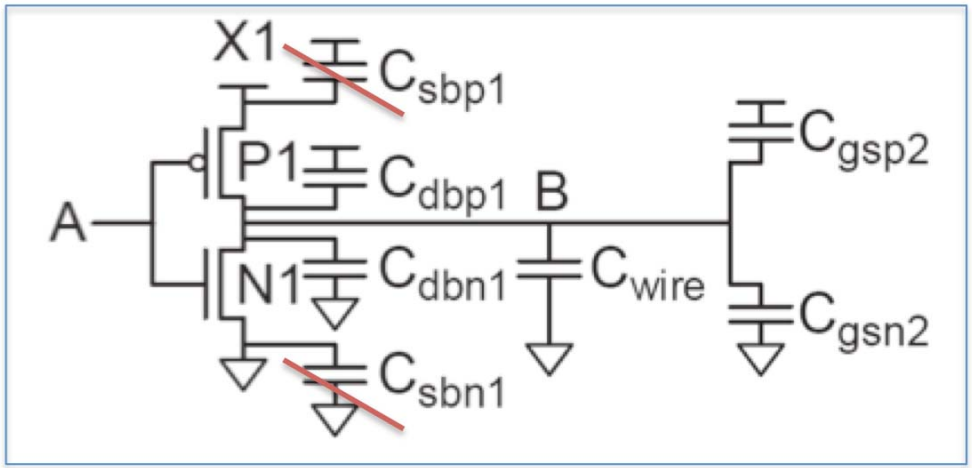



Before the step voltage is applied, $V_A$=0, $N_1$ is OFF, $P_1$ is ON and $V_B$=$V_{DD}$.
After the step, $N_1$ turns ON and it is initially in saturation ($V_{ds1}$ =$V_{DD}$ > $V_{gs1}$-$V_t$=$V_{DD}$-$V_t$) and later as $V_B$ drops below $V_{DD}$-$V_t$ it enters linear region.

$$-I_{dn1} = C_{out}\frac{dV_B}{dt}$$

# Delay Calculations (3/3)



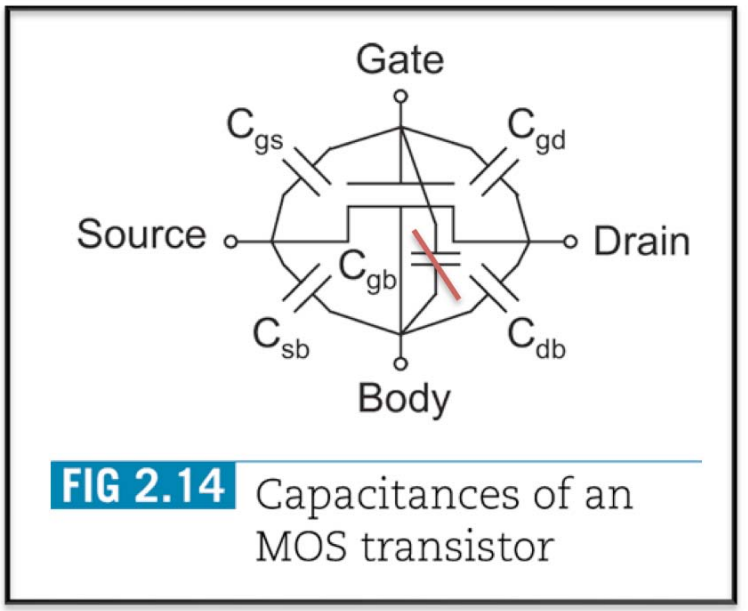
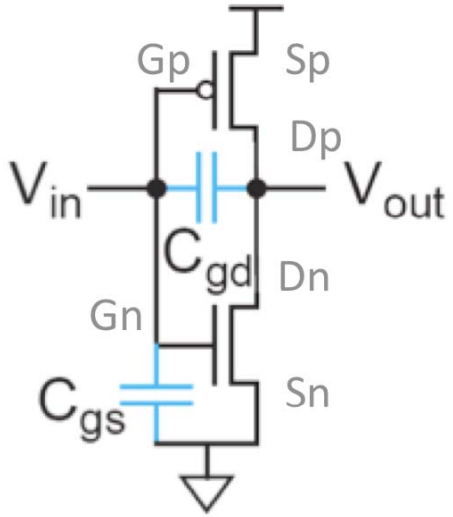
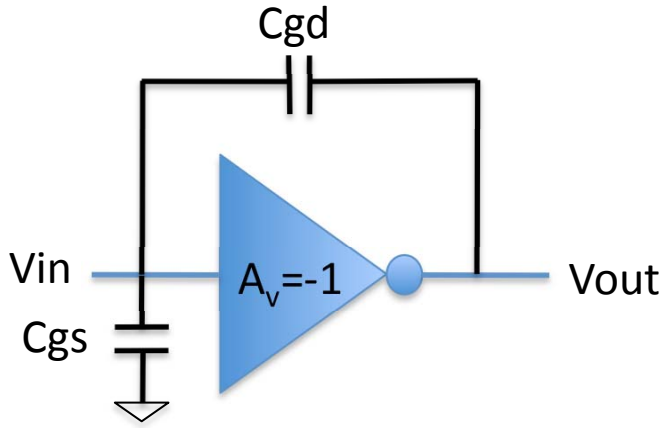




FIG 2.14 Capacitances of an MOS transistor

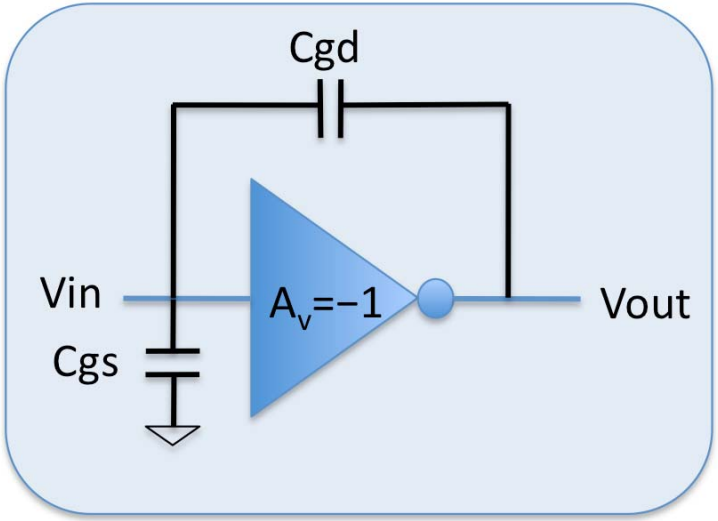
**Csb shorted, Cgb negligible. What about Cgd ?**

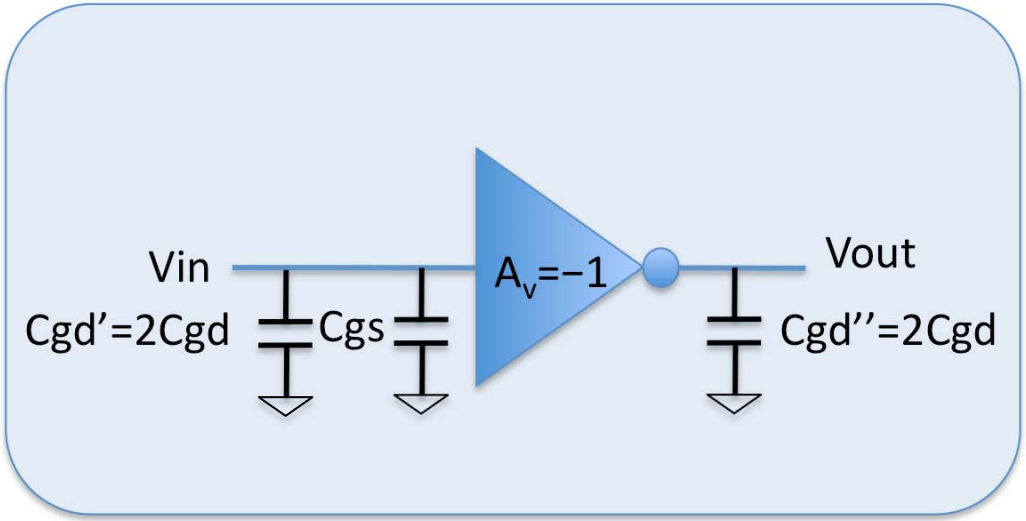
Cgd = Cgdp1+Cgdn1
Cgs= Cgsp1+Cgsn1



# Bootstrapping (Miller Effect)



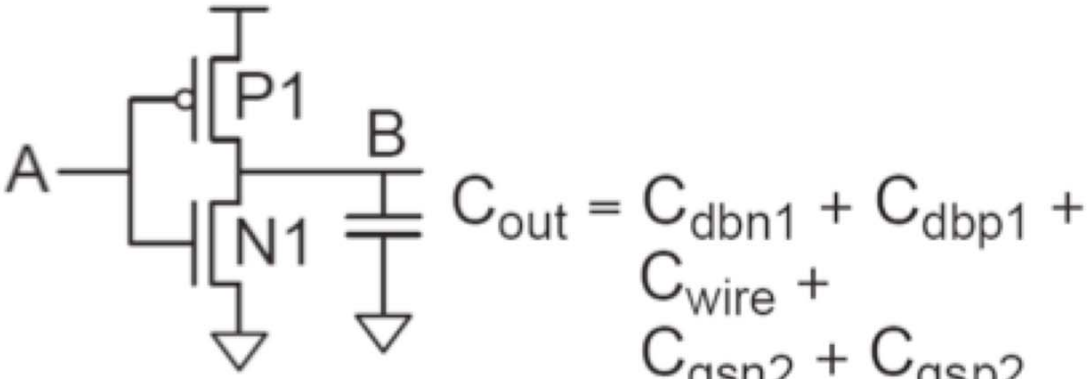

$$A_V = \frac{Vout}{Vin}$$

$$C_{gd}^{'} = C_{gd}\left(1 - A_V\right)$$

$$C_{gd}^{''} = C_{gd}\left(1 - \frac{1}{A_V}\right)$$

# Factors affecting delay ? (1/3)

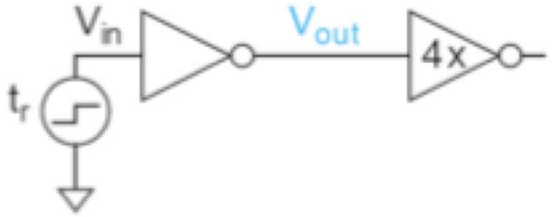
- Output capacitance

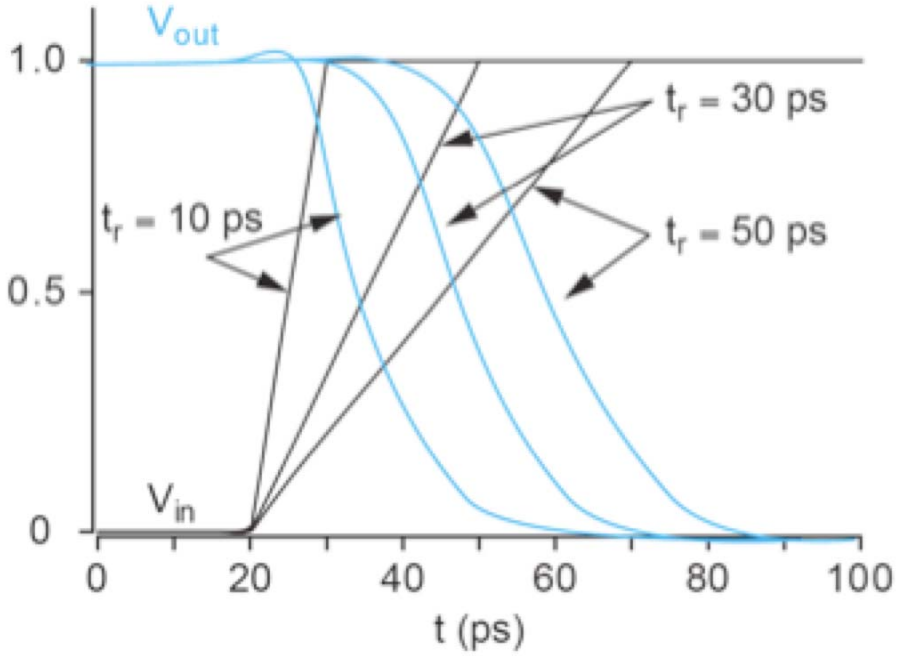


# Factors affecting delay ? (2/3)

- Slope of the input waveform (a.k.a. input slew rate)

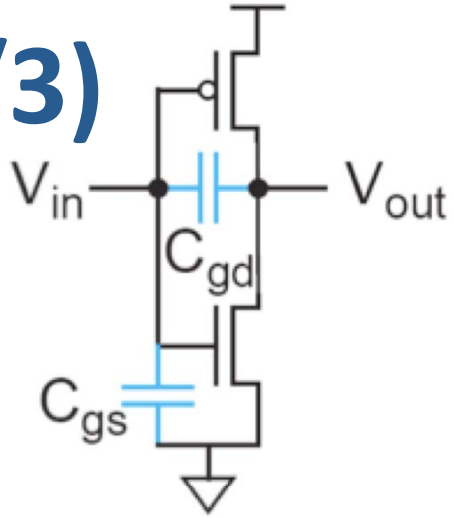
  As the gate voltage changes so does the capacitance (the transistors go through different operating modes, before turning fully ON or OFF)

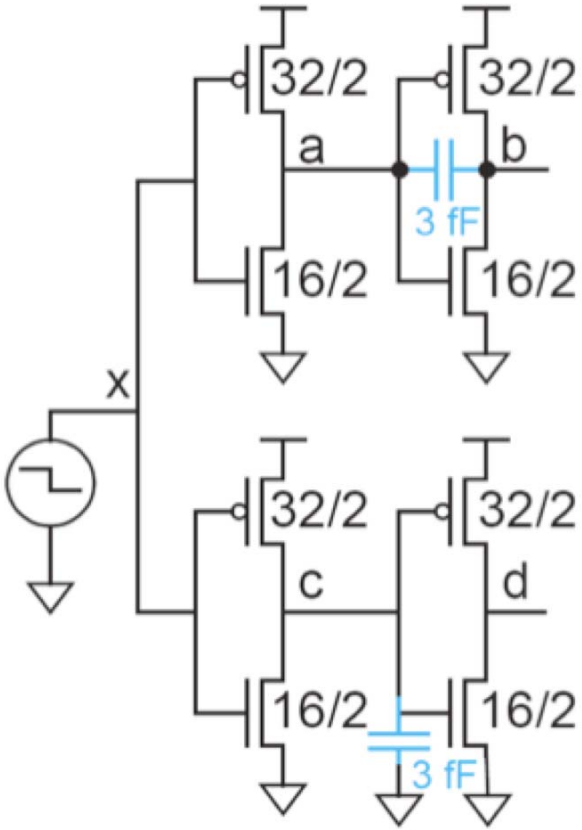


# Factors affecting delay ? (3/3)

- Bootstrapping
  the effect of Cgd on gate capacitance is effectively doubled



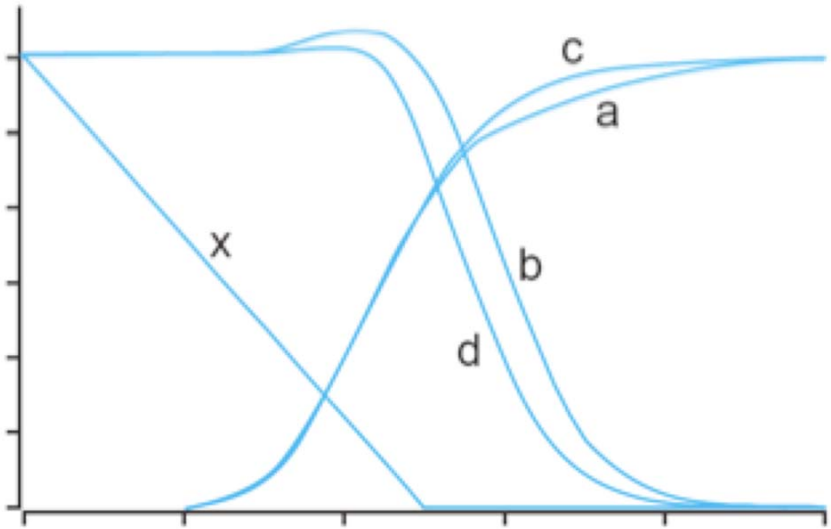




**FIGURE 4.28** The effect of bootstrapping on inverter delay and waveform shape

# more on Bootstrapping …

- The extra capacitance has greater effect when connected between input and output (feedback) as compared to when  is connected between input and ground.
- Because Cgd is fairly small, bootstrapping is only a mild annoyance in digital circuits.
- However is the inverter is biased in its linear region near VDD/2 the Cgd is multiplied by the large gain of the inverter (this is of major importance in analog circuits)

# Other factors affecting delay …

- Arrival time at multiple-input gates
- Velocity saturation
- Supply voltage dependence
- Gate-to-source (rather than gate to gnd) capacitance

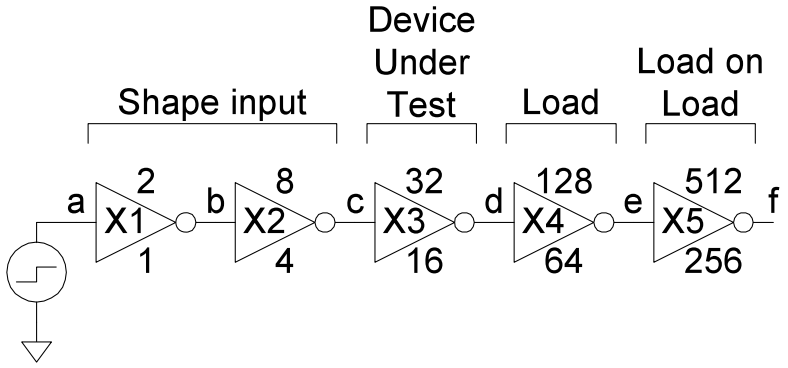
- We'll see the details later …

# Subcircuits

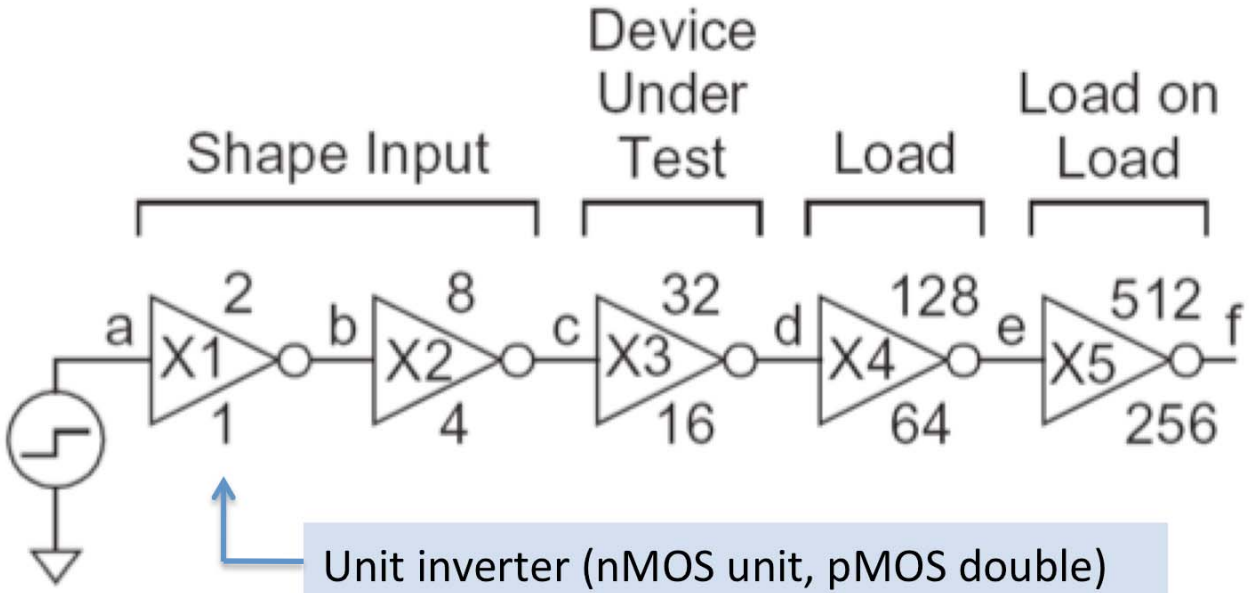
- Declare common elements as subcircuits

```
.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
```

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



# FO4 inverter delay



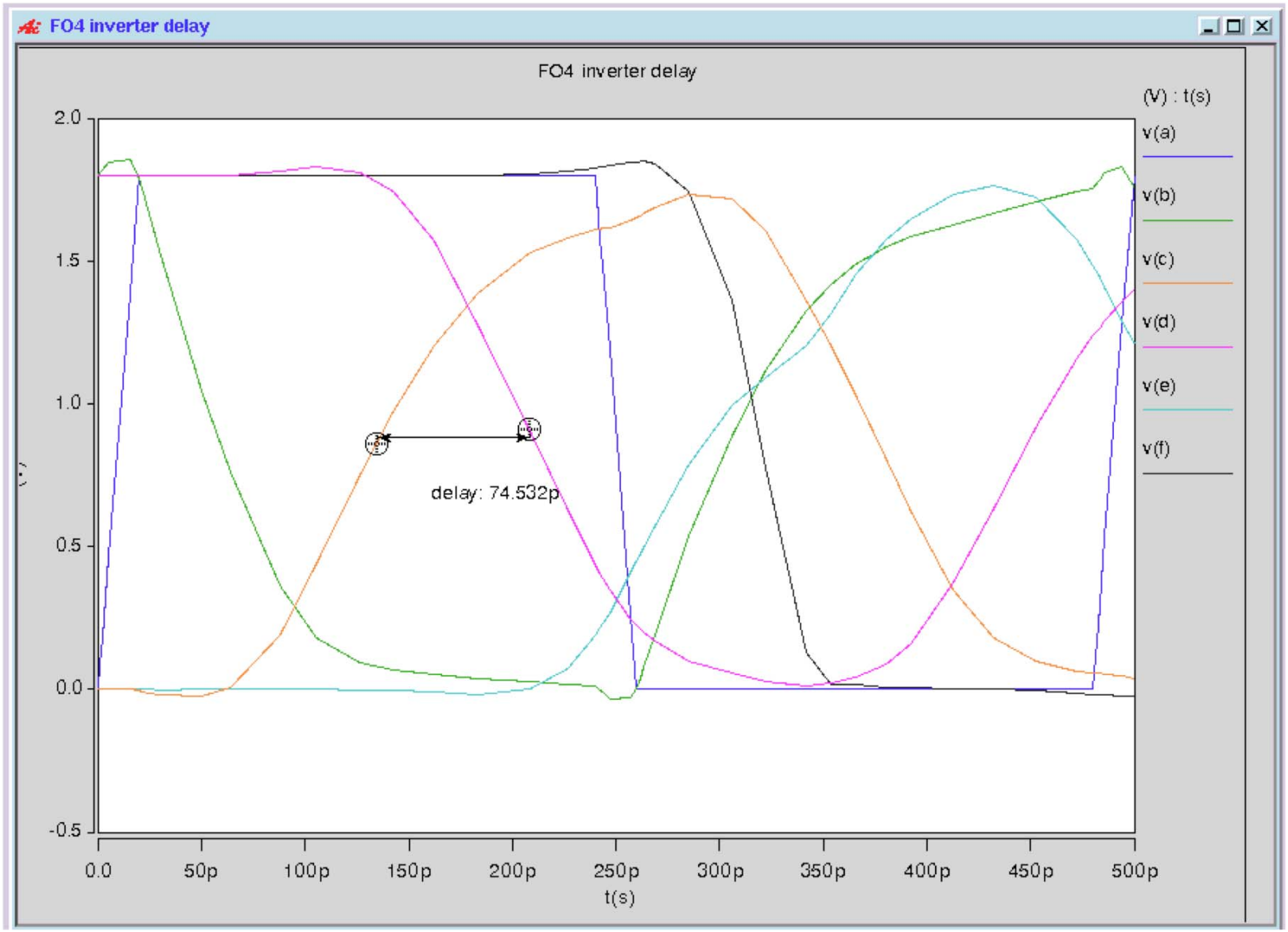

```
X1 a b inv N=4    P=8     *shape input waveform
X2 b c inv N=16   P=32    *reshape input waveform
X3 c d inv N=64   P=128   *DUT
X4 d e inv N=256  P=512   *load
X5 e f inv N=1024 P=2048  *slow down switching rate of e
```

# FO4 inverter delay

```
* fo4.sp
* David_Harris@hmc.edu 2/16/05
* Modified C. Talarico 9/30/11
• Delay (typ) of FO4 inverter in TSMC 180nm process

*************** Parameters and models
.param SUPPLY=1.8
.param H=4
.option scale=90n
.include '../mosistsmc180/mosistsmc180.sp'
.option post nomod brief

*************** Subcircuits
.global vdd gnd

.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

************** Simulation netlist
Vdd     vdd     gnd     'SUPPLY'
Vin     a       gnd     PULSE   0 'SUPPLY' 0ps 20ps 20ps 220ps 480ps
X1      a       b       inv             * shape input waveform
X2      b       c       inv     M='H'   * reshape input waveform
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

************ Analysis and Measurements
•.tran 1ps 500ps
.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'      * average propagation 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
```

**FIGURE 8.10** FO4 SPICE deck

# FO4 inverter delay



# 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 simulations with different P size
  - (2) let HSPICE optimizer do it for us

# Optimization P/N ratio

```
* fo4opt.sp
* David_Harris@hmc.edu 2/2/03
* Modified C. Talarico 9/30/11
* P/N ratio for equal rise/fall delay (typ)

•*************** Parameters and models
•.option accurate
.param SUPPLY=1.8
.param H=4
.option scale=90n nomod brief
.include '../mosistsmc180/mosistsmc180.sp'
*************** Subcircuits
.global vdd gnd
.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
***************  Simulation netlist
Vdd     vdd     gnd     'SUPPLY'
Vin     a       gnd     PULSE   0 'SUPPLY' 0ps 20ps 20ps 220ps 480ps
X1      a       b       inv     P='P1'          * shape input waveform
X2      b       c       inv     P='P1'  M=4     * reshape input waveform
X3      c       d       inv     P='P1'  M=16    * device under test
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 0.1ps 480ps SWEEP OPTIMIZE=optrange RESULTS=diff MODEL=optmod
.tran 0.1ps 280ps SWEEP OPTIMIZE=optrange RESULTS=tpd 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
```

# Optimization Results

## Best P/N ratio for lowest average delay is 1:1

.TITLE '* fo4opt.sp'

| Index | p1 | bestratio | tpdr | tpdf | tpd | diff | temper | alter# |
|---|---|---|---|---|---|---|---|---|
| 1.0000 | 4.0000 | 1.0000 | 9.163e-11 | 5.860e-11 | 7.511e-11 | 3.303e-11 | 25.0000 | 1 |

## Best P/N ratio for equal rise and fall delay is ≈ 2.84:1

.TITLE '* fo4opt.sp'

| index | p1 | bestratio | tpdr | tpdf | tpd | diff | temper | alter# |
|---|---|---|---|---|---|---|---|---|
| 6.0000 | 11.3522 | 2.8381 | 8.271e-11 | 8.270e-11 | 8.270e-11 | 2.192e-15 | 25.0000 | 1 |

# Design Corners

- It is important to simulate circuits in multiple design corners to verify the correct operation of the circuit across variations in device characteristics (process variations) and environmental conditions (supply an temperature variations)

# Design Corners

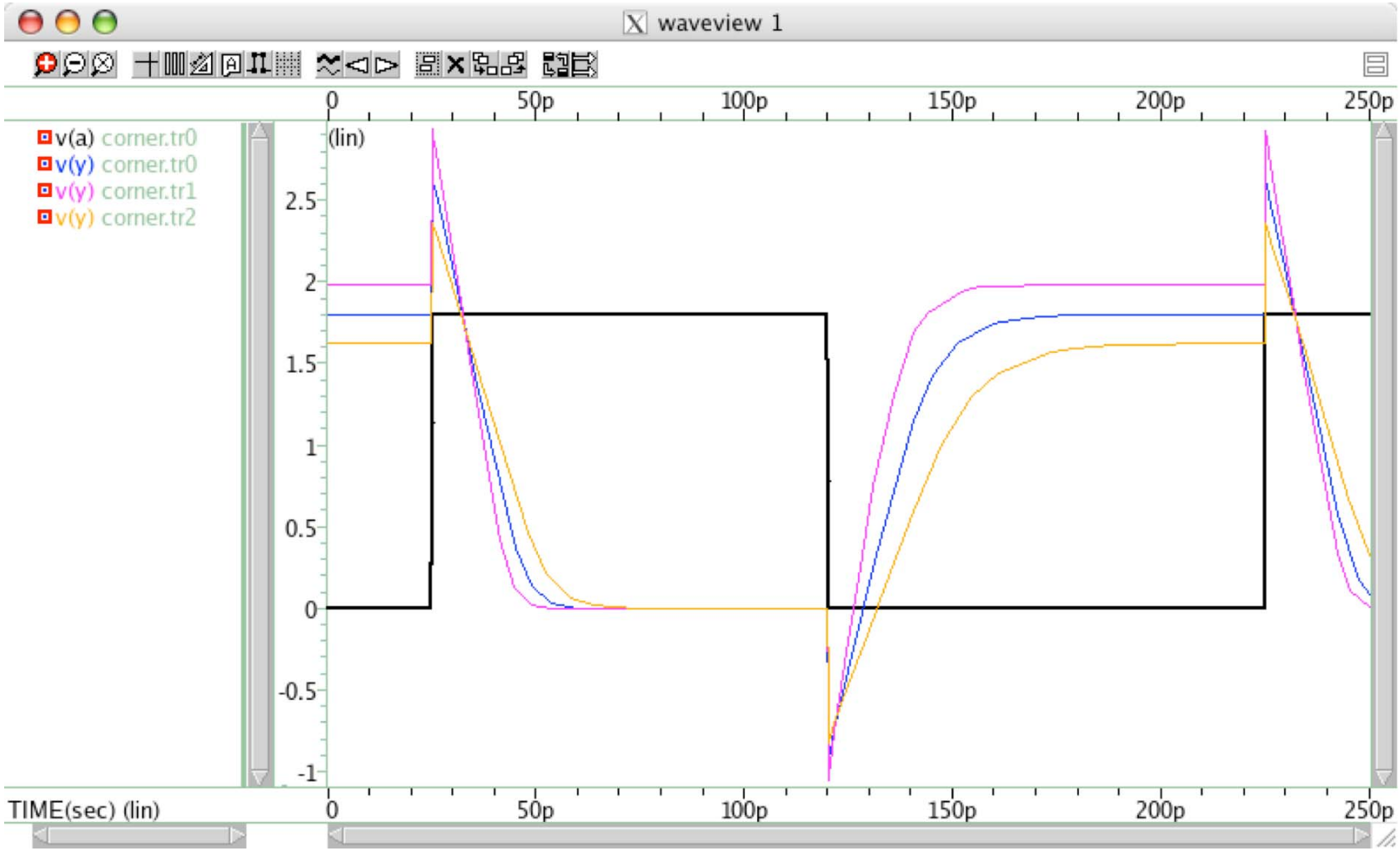
```
* Corner.sp
* Step response of uloaded unit inverter across
* design corners

******** Parameters and models
.options scale=90nm
.param SUP=1.8 *must be set before calling lib
.lib '../mosistsmc180/opConditions.lib' TT
.option post nomod brief

******** Simulation Netlist
Vdd vdd gnd 'SUPPLY'
Vin a gnd PULSE 0 'SUPPLY' 25ps 0ps 0ps 95ps 200ps
Mn y a gnd gnd NMOS W=4 L=2
+ AS=20 PS=18 AD=20 PD=18
Mp y a vdd vdd PMOS W=8 L=2
+ AS=40 PS=26 AD=40 PD=26

******** Analysis
.tran 0.1ps 250ps
.alter
.lib '../mosistsmc180/opConditions.lib' FF
.alter
.lib '../mosistsmc180/opConditions.lib' SS
.end
```

# Design Corners

```
* opConditions.lib
* Example of OPCONDITIONS library

* TT: Typical nMOS, pMOS, voltage, temperature
.lib TT
.temp 70
.param SUPPLY='SUP'
.include 'modelsTT.sp'
.endl

* SS: Slow nMOS, pMOS, low voltage, high temperature
.lib SS
.temp 125
.param SUPPLY='0.9 * SUP'
.include 'modelsSS.sp'
.endl

* FF: Fast nMOS, pMOS, high voltage, low temperature
.lib FF
.temp 0
.param SUPPLY='1.1 * SUP'
.include 'modelsFF.sp'
.endl

* and so on ...
```

# Design Corners

TT
FF
SS



# Power Measurements

- 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
.measure charge INTEGRAL I(Vdd) FROM=0ns TO=10ns
.measure energy param='charge*SUPPLY'
```

# Power Measurements

- Power in single gate
  - Connect to separate $V_{DD}$ supply
  - When the input of a “logic gate” switches, it delivers power to the supply through the gate-to-source capacitances. Be careful to differentiate this input power from the power drawn by the “logic gate” discharging its internal and load capacitance