Lecture 7: SPICE Simulation

Slides courtesy of Deming Chen

Slides based on the initial set from David Harris
Outline

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

- Readings 8.1-8.2
Introduction to SPICE

- Simulation Program with Integrated Circuit Emphasis
  - Developed in 1970’s at Berkeley
  - Many commercial versions are available
  - HSPICE is a robust 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 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!*
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    pw1    0ps 0 100ps 0 150ps 1.0 1ns 1.0
R1     in    out   2k
C1     out   gnd   100f

* Stimulus
*---------------------------------------------------
.tran 20ps 1ns
.plot v(in) v(out)
.end
Result (Graphical)
Sources

- **DC Source**
  
  \[
  \text{Vdd vdd gnd 2.5}
  \]

- **Piecewise Linear Source**
  
  \[
  \text{Vin in gnd pw1 0ps 0 100ps 0 150ps 1.0 1ns 1.0}
  \]

- **Pulsed Source**
  
  \[
  \text{Vck clk gnd PULSE 0 1.0 0ps 100ps 100ps 300ps 800ps}
  \]

PULSE v1 v2 td tr tf pw per

---

*SPICE Simulation*  
*CMOS VLSI Design*  
4th Ed.  
7
## SPICE Elements

<table>
<thead>
<tr>
<th>Letter</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Resistor</td>
</tr>
<tr>
<td>C</td>
<td>Capacitor</td>
</tr>
<tr>
<td>L</td>
<td>Inductor</td>
</tr>
<tr>
<td>K</td>
<td>Mutual Inductor</td>
</tr>
<tr>
<td>V</td>
<td>Independent voltage source</td>
</tr>
<tr>
<td>I</td>
<td>Independent current source</td>
</tr>
<tr>
<td>M</td>
<td>MOSFET</td>
</tr>
<tr>
<td>D</td>
<td>Diode</td>
</tr>
<tr>
<td>Q</td>
<td>Bipolar transistor</td>
</tr>
<tr>
<td>W</td>
<td>Lossy transmission line</td>
</tr>
<tr>
<td>X</td>
<td>Subcircuit</td>
</tr>
<tr>
<td>E</td>
<td>Voltage-controlled voltage source</td>
</tr>
<tr>
<td>G</td>
<td>Voltage-controlled current source</td>
</tr>
<tr>
<td>H</td>
<td>Current-controlled voltage source</td>
</tr>
<tr>
<td>F</td>
<td>Current-controlled current source</td>
</tr>
</tbody>
</table>
## Units

<table>
<thead>
<tr>
<th>Letter</th>
<th>Unit</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>atto</td>
<td>$10^{-18}$</td>
</tr>
<tr>
<td>f</td>
<td>fempto</td>
<td>$10^{-15}$</td>
</tr>
<tr>
<td>p</td>
<td>pico</td>
<td>$10^{-12}$</td>
</tr>
<tr>
<td>n</td>
<td>nano</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>u</td>
<td>micro</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>m</td>
<td>milli</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>k</td>
<td>kilo</td>
<td>$10^{3}$</td>
</tr>
<tr>
<td>x</td>
<td>mega</td>
<td>$10^{6}$</td>
</tr>
<tr>
<td>g</td>
<td>giga</td>
<td>$10^{9}$</td>
</tr>
</tbody>
</table>

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

* mosiv.sp

* Parameters and models
.include './models/ibm065/models.sp'
.temp 70
.option post

* Simulation netlist
*nmos
Vgs  g  gnd  0
Vds  d  gnd  0
M1   d  g  gnd  gnd  NMOS  W=100n  L=50n

* Stimulus
* .dc Vds 0 1.0 0.05 SWEEP Vgs 0 1.0 0.2
.end
I-V Characteristics

- nMOS I-V
  - $V_{gs}$ dependence
  - Saturation

![Graph showing I-V characteristics for nMOS with different $V_{gs}$ values]
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=1.0
.option scale=25n
.include '../models/ibm065/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 0.1ps 80ps
.end
Transient Results

- Unloaded inverter
  - Overshoot
  - Very fast edges

![Graph showing transients](image)

- Transient times:
  - $t_f = 2.5\, \text{ps}$
  - $t_{pdf} = 3.1\, \text{ps}$
  - $t_{pdr} = 3.6\, \text{ps}$
  - $t_r = 3.5\, \text{ps}$
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=1.0
.param H=4
.option scale=25n
.include '../models/ibm065/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 20ps 20ps 120ps 280ps
X1 a b inv * shape input waveform
X2 b c inv M='H' * reshape input waveform
FO4 Inverter Delay Cont.

<table>
<thead>
<tr>
<th>X3</th>
<th>c</th>
<th>d</th>
<th>inv</th>
<th>M='H**2' * device under test</th>
</tr>
</thead>
<tbody>
<tr>
<td>X4</td>
<td>d</td>
<td>e</td>
<td>inv</td>
<td>M='H**3' * load</td>
</tr>
<tr>
<td>x5</td>
<td>e</td>
<td>f</td>
<td>inv</td>
<td>M='H**4' * load on load</td>
</tr>
</tbody>
</table>

* Stimulus

```
.tran 0.1ps 280ps
.measure tpdr
+ TRIG v(c) VAL='SUPPLY/2' FALL=1
+ TARG v(d) VAL='SUPPLY/2' RISE=1
.measure tpdf
+ TRIG v(c) VAL='SUPPLY/2' RISE=1
+ TARG v(d) VAL='SUPPLY/2' FALL=1
.measure tpd param='(tpdr+tpdf)/2'
.measure trise
+ TRIG v(d) VAL='0.2*SUPPLY' RISE=1
+ TARG v(d) VAL='0.8*SUPPLY' RISE=1
.measure tfall
+ TRIG v(d) VAL='0.8*SUPPLY' FALL=1
+ TARG v(d) VAL='0.2*SUPPLY' FALL=1
.end
```
FO4 Results
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
* fo4opt.sp

* Parameters and models
*---------------------------------------------------------------------
.param SUPPLY=1.0
.option scale=25n
.include './models/ibm065/models.sp'
.temp 70
.option post

* Subcircuits
*---------------------------------------------------------------------
.global vdd gnd
.include './lib/inv.sp'

* Simulation netlist
*---------------------------------------------------------------------
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vdd</td>
<td>vdd</td>
<td>gnd</td>
<td>'SUPPLY'</td>
<td></td>
</tr>
<tr>
<td>Vin</td>
<td>a</td>
<td>gnd</td>
<td>PULSE</td>
<td>0</td>
</tr>
<tr>
<td>X1</td>
<td>a</td>
<td>b</td>
<td>inv</td>
<td>P='P1'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* shape input waveform</td>
</tr>
<tr>
<td>X2</td>
<td>b</td>
<td>c</td>
<td>inv</td>
<td>P='P1'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* reshape input</td>
</tr>
<tr>
<td>X3</td>
<td>c</td>
<td>d</td>
<td>inv</td>
<td>P='P1'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>* device under test</td>
</tr>
</tbody>
</table>
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 0.1ps 280ps 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 2.9:1
  - $t_{pd} = t_{pdr} = t_{pdf} = 17.9$ 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=$t_{pd}$ MODEL=optmod

  - P/N ratio of 1.8:1
    - $t_{pdr} = 18.8$ ps, $t_{pdf} = 15.2$ ps, $t_{pd} = 17.0$ ps
- P/N ratios of 1.5:1 – 2.2:1 gives $t_{pd} < 17.2$ 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
Summary

- Various analysis with SPIC
- Optimization and measurement
- Useful tool for transistor-level delay, power, process variation evaluations
  - Can be the gold model for gate-level and higher
  - Can be useful for novel device and circuit studies

- Next lecture
  - Combinational circuit design
  - Reading 9.1-9.2.2