EE4800 CMOS Digital IC Design & Analysis

Lecture 7 Spice Simulation
Zhuo Feng
Outline

- Introduction to SPICE
- DC Analysis
- Transient Analysis
- Subcircuits
- Optimization
- Power Measurement
- Logical Effort Characterization
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

* Simulation netlist

Vin  in   gnd  pw1  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
### Legend

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

### Time vs. v(in)

<table>
<thead>
<tr>
<th>Time (p)</th>
<th>v(in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000p</td>
<td>0.0000m</td>
</tr>
<tr>
<td>20.0000p</td>
<td>0.2000p</td>
</tr>
<tr>
<td>40.0000p</td>
<td>0.4000p</td>
</tr>
<tr>
<td>60.0000p</td>
<td>0.6000p</td>
</tr>
<tr>
<td>80.0000p</td>
<td>0.8000p</td>
</tr>
<tr>
<td>100.0000p</td>
<td>1.0000p</td>
</tr>
<tr>
<td>120.0000p</td>
<td>1.2000p</td>
</tr>
<tr>
<td>140.0000p</td>
<td>1.4000p</td>
</tr>
<tr>
<td>160.0000p</td>
<td>1.6000p</td>
</tr>
<tr>
<td>180.0000p</td>
<td>1.8000p</td>
</tr>
<tr>
<td>200.0000p</td>
<td>2.0000p</td>
</tr>
<tr>
<td>220.0000p</td>
<td>2.2000p</td>
</tr>
<tr>
<td>240.0000p</td>
<td>2.4000p</td>
</tr>
<tr>
<td>260.0000p</td>
<td>2.6000p</td>
</tr>
<tr>
<td>280.0000p</td>
<td>2.8000p</td>
</tr>
<tr>
<td>300.0000p</td>
<td>3.0000p</td>
</tr>
<tr>
<td>320.0000p</td>
<td>3.2000p</td>
</tr>
<tr>
<td>340.0000p</td>
<td>3.4000p</td>
</tr>
<tr>
<td>360.0000p</td>
<td>3.6000p</td>
</tr>
<tr>
<td>380.0000p</td>
<td>3.8000p</td>
</tr>
<tr>
<td>400.0000p</td>
<td>4.0000p</td>
</tr>
<tr>
<td>420.0000p</td>
<td>4.2000p</td>
</tr>
<tr>
<td>440.0000p</td>
<td>4.4000p</td>
</tr>
<tr>
<td>460.0000p</td>
<td>4.6000p</td>
</tr>
<tr>
<td>480.0000p</td>
<td>4.8000p</td>
</tr>
<tr>
<td>500.0000p</td>
<td>5.0000p</td>
</tr>
<tr>
<td>520.0000p</td>
<td>5.2000p</td>
</tr>
<tr>
<td>540.0000p</td>
<td>5.4000p</td>
</tr>
<tr>
<td>560.0000p</td>
<td>5.6000p</td>
</tr>
<tr>
<td>580.0000p</td>
<td>5.8000p</td>
</tr>
<tr>
<td>600.0000p</td>
<td>6.0000p</td>
</tr>
<tr>
<td>620.0000p</td>
<td>6.2000p</td>
</tr>
<tr>
<td>640.0000p</td>
<td>6.4000p</td>
</tr>
<tr>
<td>660.0000p</td>
<td>6.6000p</td>
</tr>
<tr>
<td>680.0000p</td>
<td>6.8000p</td>
</tr>
<tr>
<td>700.0000p</td>
<td>7.0000p</td>
</tr>
<tr>
<td>720.0000p</td>
<td>7.2000p</td>
</tr>
<tr>
<td>740.0000p</td>
<td>7.4000p</td>
</tr>
<tr>
<td>760.0000p</td>
<td>7.6000p</td>
</tr>
<tr>
<td>780.0000p</td>
<td>7.8000p</td>
</tr>
<tr>
<td>800.0000p</td>
<td>8.0000p</td>
</tr>
</tbody>
</table>
Result (Graphical)
Sources

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

- **Piecewise Linear Source**
  
  \[ V_{in} \text{ in gnd pw1 0ps 0 100ps 0 150ps 1.8 800ps 1.8} \]

- **Pulsed Source**
  
  \[ V_{ck} \text{ clk gnd PULSE 0 1.8 0ps 100ps 100ps 300ps 800ps} \]
## 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>mili</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 femtofarad capacitor = 100fF, 100f, 100e-15
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**
  - $V_{gs}$ dependence
  - Saturation

![Graph showing NMOS I-V characteristics with different $V_{gs}$ values.](image)
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

** Parameter and models definitions **
.param SUPPLY=1.8
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post

* Simulation netlist

** Netlist Details **
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

** Simulation Settings **
.tran 1ps 200ps
.end
Transient Results

- **Unloaded inverter**
  - Overshoot
  - Very fast edges

![Diagram showing transient results with key parameters: t_f = 10ps, t_{pdr} = 12ps, t_{pdf} = 15ps, t_r = 16ps.](image)
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

![Diagram showing the fanout-of-4 inverter delay with loading and test points labeled as X1, X2, X3, X4, X5, and X with corresponding loads 1, 4, 16, 64, 256, and 1024.](image-url)
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.

\[
\begin{align*}
X3 & \quad c & \quad d & \quad \text{inv} & \quad M='H^{*2}' & \quad \text{device under test} \\
X4 & \quad d & \quad e & \quad \text{inv} & \quad M='H^{*3}' & \quad \text{load} \\
x5 & \quad e & \quad f & \quad \text{inv} & \quad M='H^{*4}' & \quad \text{load on load}
\end{align*}
\]

* Stimulus

*-----------------------------------------------------------------------------------------------*

.tran 1ps 1000ps
.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'  
* average prop delay

.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

\[ t_{pd} = 66\text{ps} \]

\[ t_{pd} = 83\text{ps} \]
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
Logical Effort

- Logical effort can be measured from simulation
  - As with FO4 inverter, shape input, load output

\[
\begin{array}{cccc}
X_1 & X_2 & X_3 & X_4 \\
M=1 & M=h & M=h^2 & M=h^3 \\
\end{array}
\]

Load on Load

Load

Device Under Test

Shape input
Logical Effort Plots

- Plot $t_{pd}$ vs. $h$
  - Normalize by $\tau$
  - $y$-intercept is parasitic delay
  - Slope is logical effort
- Delay fits straight line very well in any process as long as input slope is consistent

$\tau = 15$ ps

\begin{figure}
\centering
\includegraphics[width=0.7\textwidth]{plot}
\end{figure}
Logical Effort Data

- For NAND gates in TSMC 180 nm process:

<table>
<thead>
<tr>
<th># of inputs</th>
<th>Input</th>
<th>Rising Logical Effort $g_u$</th>
<th>Falling Logical Effort $g_d$</th>
<th>Average Logical Effort $g$</th>
<th>Rising Parasitic Delay $p_u$</th>
<th>Falling Parasitic Delay $p_d$</th>
<th>Average Parasitic Delay $p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$A$</td>
<td>1.45</td>
<td>0.82</td>
<td>1.14</td>
<td>2.37</td>
<td>1.42</td>
<td>1.90</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>1.36</td>
<td>0.97</td>
<td>1.17</td>
<td>1.60</td>
<td>1.22</td>
<td>1.41</td>
</tr>
<tr>
<td>3</td>
<td>$A$</td>
<td>1.82</td>
<td>0.80</td>
<td>1.31</td>
<td>4.90</td>
<td>2.11</td>
<td>3.51</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>1.73</td>
<td>0.95</td>
<td>1.34</td>
<td>3.94</td>
<td>1.86</td>
<td>2.90</td>
</tr>
<tr>
<td></td>
<td>$C$</td>
<td>1.63</td>
<td>1.10</td>
<td>1.36</td>
<td>2.60</td>
<td>1.54</td>
<td>2.07</td>
</tr>
<tr>
<td>4</td>
<td>$A$</td>
<td>1.96</td>
<td>0.89</td>
<td>1.42</td>
<td>6.54</td>
<td>3.08</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>1.86</td>
<td>1.02</td>
<td>1.44</td>
<td>5.71</td>
<td>2.84</td>
<td>4.28</td>
</tr>
<tr>
<td></td>
<td>$C$</td>
<td>1.80</td>
<td>1.14</td>
<td>1.47</td>
<td>4.69</td>
<td>2.42</td>
<td>3.56</td>
</tr>
<tr>
<td></td>
<td>$D$</td>
<td>1.71</td>
<td>1.28</td>
<td>1.49</td>
<td>3.26</td>
<td>1.97</td>
<td>2.62</td>
</tr>
</tbody>
</table>

- Notes:
  - Parasitic delay is greater for outer input
  - Average logical effort is better than estimated
# Comparison

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Orbit</th>
<th>AMI</th>
<th>HP</th>
<th>HP</th>
<th>AMI</th>
<th>AMI</th>
<th>TSMC</th>
<th>TSMC</th>
<th>TSMC</th>
<th>TSMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>MOSIS</td>
<td>TSMC</td>
</tr>
<tr>
<td>Feature Size $f$</td>
<td>2000</td>
<td>1600</td>
<td>800</td>
<td>800</td>
<td>600</td>
<td>600</td>
<td>350</td>
<td>250</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>$V_{DD}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3.3</td>
<td>5</td>
<td>3.3</td>
<td>3.3</td>
<td>2.5</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>FO4 inv delay</td>
<td>856</td>
<td>717</td>
<td>297</td>
<td>427</td>
<td>230</td>
<td>312</td>
<td>210</td>
<td>153</td>
<td>99.4</td>
<td>75.6</td>
</tr>
<tr>
<td>$\tau$</td>
<td>170</td>
<td>143</td>
<td>59</td>
<td>84</td>
<td>45</td>
<td>60</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>15</td>
</tr>
</tbody>
</table>

## Logical Effort

<table>
<thead>
<tr>
<th>Function</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>NAND2</td>
<td>1.13</td>
<td>1.07</td>
<td>1.07</td>
<td>1.09</td>
<td>1.05</td>
<td>1.08</td>
<td>1.12</td>
<td>1.12</td>
<td>1.17</td>
<td>1.14</td>
</tr>
<tr>
<td>NAND3</td>
<td>1.32</td>
<td>1.22</td>
<td>1.21</td>
<td>1.25</td>
<td>1.19</td>
<td>1.24</td>
<td>1.29</td>
<td>1.29</td>
<td>1.36</td>
<td>1.31</td>
</tr>
<tr>
<td>NAND4</td>
<td>1.48</td>
<td>1.35</td>
<td>1.33</td>
<td>1.37</td>
<td>1.33</td>
<td>1.38</td>
<td>1.44</td>
<td>1.43</td>
<td>1.49</td>
<td>1.42</td>
</tr>
<tr>
<td>NOR2</td>
<td>1.57</td>
<td>1.62</td>
<td>1.59</td>
<td>1.58</td>
<td>1.58</td>
<td>1.60</td>
<td>1.52</td>
<td>1.50</td>
<td>1.46</td>
<td>1.50</td>
</tr>
<tr>
<td>NOR3</td>
<td>2.16</td>
<td>2.32</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.30</td>
<td>2.07</td>
<td>2.02</td>
<td>1.94</td>
<td>2.00</td>
</tr>
<tr>
<td>NOR4</td>
<td>2.55</td>
<td>2.7</td>
<td>2.61</td>
<td>2.64</td>
<td>2.57</td>
<td>2.68</td>
<td>2.46</td>
<td>2.37</td>
<td>2.27</td>
<td>2.38</td>
</tr>
</tbody>
</table>

## Parasitic Delay

<table>
<thead>
<tr>
<th>Function</th>
<th>1.00</th>
<th>1.05</th>
<th>1.05</th>
<th>1.13</th>
<th>1.18</th>
<th>1.25</th>
<th>1.33</th>
<th>1.18</th>
<th>0.95</th>
<th>1.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1.00</td>
<td>1.05</td>
<td>1.05</td>
<td>1.13</td>
<td>1.18</td>
<td>1.25</td>
<td>1.33</td>
<td>1.18</td>
<td>0.95</td>
<td>1.03</td>
</tr>
<tr>
<td>NAND2</td>
<td>1.87</td>
<td>1.79</td>
<td>1.85</td>
<td>2.05</td>
<td>1.92</td>
<td>2.10</td>
<td>2.28</td>
<td>2.07</td>
<td>1.74</td>
<td>1.90</td>
</tr>
<tr>
<td>NAND3</td>
<td>3.34</td>
<td>3.22</td>
<td>3.30</td>
<td>3.75</td>
<td>3.40</td>
<td>3.79</td>
<td>4.15</td>
<td>3.65</td>
<td>3.08</td>
<td>3.51</td>
</tr>
<tr>
<td>NAND4</td>
<td>4.64</td>
<td>4.42</td>
<td>4.54</td>
<td>5.12</td>
<td>4.70</td>
<td>5.23</td>
<td>5.75</td>
<td>5.01</td>
<td>4.26</td>
<td>4.81</td>
</tr>
<tr>
<td>NOR2</td>
<td>2.86</td>
<td>2.97</td>
<td>2.91</td>
<td>3.13</td>
<td>3.29</td>
<td>3.56</td>
<td>3.52</td>
<td>2.95</td>
<td>2.41</td>
<td>2.85</td>
</tr>
<tr>
<td>NOR3</td>
<td>5.65</td>
<td>6.22</td>
<td>6.05</td>
<td>6.47</td>
<td>7.02</td>
<td>7.70</td>
<td>6.89</td>
<td>5.61</td>
<td>4.49</td>
<td>5.57</td>
</tr>
</tbody>
</table>
SPICE overview

- N equations in terms of N unknown Node voltages
- More generally using modified nodal analysis

\[ i = G(v) \]
Time Domain Equations at node 1:

\[
C \frac{d (v_1 - v_3)}{dt} + \frac{(v_1 - v_4)}{R} - G(v_2 - v_1) = 0
\]

- If we do this for all N nodes:

\[
F(\vec{x}(t), \vec{x}(t), \vec{u}(t)) = 0 \quad \vec{x}(0) = \vec{X}
\]

\[
\vec{x}(t) = \text{N dimensional vector of unknown node voltages}
\]

\[
\vec{u}(t) = \text{vector of independent sources}
\]

\[
F = \text{nonlinear operator}
\]
Closed form solution is not possible for arbitrary order of differential equations

We must approximate the solution of:

\[ F(\ddot{x}(t), \dot{x}(t), u(t)) = 0 \quad \ddot{x}(0) = \ddot{X} \]

This is facilitated in SPICE via numerical solutions
Basic circuit analyses

► (Nonlinear) DC analysis
  ▼ Finds the DC operating point of the circuit
  ▼ Solves a set of nonlinear algebraic eqns

► AC analysis
  ▼ Performs frequency-domain small-signal analysis
  ▼ Require a preceding DC analysis
  ▼ Solves a set of complex linear eqns

► (Nonlinear) transient analysis
  ▼ Computes the time-domain circuit transient response
  ▼ Solves a set of nonlinear different eqns
  ▼ Converts to a set nonlinear algebraic of eqns using numerical integration
SPICE offers practical techniques to solve circuit problems in time & freq. domains

- Interface to device models
  - Transistors, diodes, nonlinear caps etc

- Sparse linear solver

- Nonlinear solver – Newton-Raphson method

- Numerical integration

- Convergence & time-step control
Circuit equations are usually formulated using

- Nodal analysis
  - $N$ equations in $N$ nodal voltages

- Modified analysis
  - Circuit unknowns are nodal voltages & some branch currents
  - Branch current variables are added to handle
    - Voltages sources
    - Inductors
    - Current controlled voltage source etc

Formulations can be done in both time and frequency
How do we set up a matrix problem given a list of linear(ized) circuit elements?

Similar to reading a netlist for a linear circuit:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>From</th>
<th>To</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>0</td>
<td>1</td>
<td>$1mA$</td>
</tr>
<tr>
<td>$R_1$</td>
<td>1</td>
<td>0</td>
<td>$10\Omega$</td>
</tr>
<tr>
<td>$R_2$</td>
<td>1</td>
<td>2</td>
<td>$5\Omega$</td>
</tr>
<tr>
<td>$R_3$</td>
<td>2</td>
<td>0</td>
<td>$100\Omega$</td>
</tr>
</tbody>
</table>
The nodal analysis matrix equations are easily constructed via KCL at each node:

\[ Y \bar{v} = \bar{J} \]
Naïve approach
  a) Write down the KCL eqn for each node
  b) Combine all of them to a get N eqns in N node voltages

Intuitive for hand analysis

Computer programs use a more convenient “element” centric approach
  Element stamps
Instead of converting the netlist into a graph and writing KCL eqns, *stamp* in elements one at a time:

**Stamps: add to existing matrix entries**

\[
Y = \begin{bmatrix}
\frac{1}{R} & \frac{1}{R} \\
\frac{1}{R} & \frac{1}{R} \\
\end{bmatrix}
\]

From row \(i\) to row \(j\)

From col. \(i\) to col. \(j\)
RHS $\vec{J}$ of equations are stamped in a similar way:

\[
\begin{bmatrix}
-I \\
I
\end{bmatrix}
\]

From row $i$

To row $j$
Stamping our simple example one element at a time:

\[
\begin{array}{ccc}
I_1 & 0 & 1 & 1 \text{mA} \\
R_1 & 1 & 0 & 10 \ \Omega \\
R_2 & 1 & 2 & 5 \ \Omega \\
R_3 & 2 & 0 & 100 \ \Omega \\
\end{array}
\]

\[
\begin{bmatrix}
G_1 + G_2 & -G_2 \\
-G_2 & G_2 + G_3
\end{bmatrix}
\begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}
= 
\begin{bmatrix}
I_1 \\
0
\end{bmatrix}
\]
We know that nonlinear elements are first converted to linear components, then stamped.

\[ I_{EQ} \quad G_{EQ} \]
For 3 & 4 terminal elements we know that the linearized models have linear controlled sources.

We can stamp in MOSFETs in terms of a complete stamp, or in terms of simpler element stamps.
Voltage controlled current source

\[ I = 0 \]

\[ i = g_m v_{k\ell} \]

Voltmeter

\[
\begin{bmatrix}
g_m & -g_m \\
-g_m & g_m
\end{bmatrix}
\]

Large value that does not fall on diagonal of Y!
All other types of controlled sources include voltage sources.

Voltage sources are inherently incompatible with nodal analysis.

Grounded voltages sources are easily accommodated.

\[
\begin{bmatrix}
1 & \cdots & \vdots & v_1 \\
\vdots & \ddots & \vdots & v_2 \\
\vdots & \vdots & \ddots & \vdots \\
\end{bmatrix} = \begin{bmatrix} 2 \end{bmatrix}
\]
But a voltage source in between nodes is more difficult

Node voltages $k$ and $l$ are not independent
We no longer have $N$ independent node voltage variables.

So we can potentially eliminate one equation and one variable (section 2.3 of reference [1]).

But the more popular solution is **modified** nodal analysis (MNA).

Create one extra variable and one extra equation.
Extra variable: voltage source current

Allows us to write KCL at nodes $k$ and $\ell$

Extra equation

$$v_k - v_\ell = V$$

Advantage: now have an easy way of printing current results - - ammeter
Voltage source stamp:

\[
\begin{bmatrix}
\text{row} & k & \begin{bmatrix}
1 \\
-1
\end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\text{row} & \ell & \begin{bmatrix}
1 & -1 & 0
\end{bmatrix}
\end{bmatrix}
= \begin{bmatrix}
\text{row} & N+1
\end{bmatrix}
\begin{bmatrix}
i \\
V
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{col} & \text{col} & \text{col}
\end{bmatrix}
\begin{bmatrix}
k \\
\ell \\
N+1
\end{bmatrix}
\]
Current-controlled current source (e.g. BJT) has to stamp in an ammeter and a controlled current source

\[ i_2 = \alpha i_1 \]
In general, we would not blindly build the matrix from an input netlist and then attempt to solve it.

Various illegal ckts are possible:

**Cutsets of current sources**
Loops of voltage sources

Dangling nodes
Once we efficiently formulate MNA equations, an efficient solution to \( Y\hat{\mathbf{V}} = \hat{\mathbf{J}} \) is even more important.

For large circuits the matrix is really sparse:
- Number of entries in \( Y \) is a function of number of elements connected to the corresponding node.

Inverting a sparse matrix is never a good idea since the inverse is not sparse!

Instead direct solution methods employ Gaussian Elimination or LU factorization.