A brief introduction on HSPICE

Siavash Kananian
Sharif University of Technology
Electronics III
What is Spice?

• Simulation Program with Integrated Circuit Emphasis
• General purpose analog circuit simulator
• Used in IC and board-level design for check of integrity of circuit designs and prediction of circuit behavior
• Developed at Electronics Research Laboratory of the University of California, Berkeley
• SPICE simulation is industry-standard for verification of circuit operation at transistor level before manufacturing
• Description of circuit elements (transistors, resistors, capacitors, etc.) and connections by netlists
• Netlists translated into nonlinear differential algebraic equations
• Solving by implicit integration methods, Newton's method and sparse matrix techniques
HSpice features

- Superior convergence
- Accurate modeling, including many foundry models
- Hierarchical node naming and reference
- Circuit optimization for models and cells, with incremental or simultaneous Multiparameter optimizations in AC, DC, and transient simulations
- Monte Carlo and worst-case design support
- Input, output, and behavioral algebraics for cells with parameters
- Cell characterization tools to characterize standard cell libraries
- Geometric lossy-coupled transmission lines for PCB, multi-chip, package, and IC technologies
Examples of Multipoint Experiments

• **Process variation** – Monte Carlo or worst-case model parameter variation

• **Element variation** – Monte Carlo or element parameter sweeps

• **Voltage variation** – VCC, VDD, or substrate supply variation

• **Temperature variation** – design temperature sensitivity.

• **Timing analysis** – basic timing, jitter, and signal integrity analysis

• **Parameter optimization** – balancing complex constraints, such as speed versus power, or frequency versus slew rate versus offset (analog circuits)

Source: Synopsys, 2007
Circuit Analysis Types

HSPICE

- Parametric
  - Monte Carlo
  - Optimization
  - Data Driven

- Operating Point
  - Pole-Zero
  - Monte Carlo

- Frequency
  - S-parameter
  - Optimization
  - Monte Carlo
  - Data Driven

- Transient
  - Monte Carlo
  - Optimization
  - Mixed AC/Transient
  - Data Driven

Source: Synopsys, 2007
Modeling Technologies

Source: Synopsys, 2007
Input file

• Contains:
  – **Design netlist** (subcircuits, macros, power supplies, and so on).
  – Statement naming the library to use (optional).
  – Specifies the **type of analysis** to run (optional).
  – Specifies the **type of output desired** (optional).

• Can be from **texteditor** or **schematic tool** (Cadence Virtuoso, MMI, ...)

Source: Synopsys, 2007
Input format

• Input reader accept input token, such as:
  – a statement name
  – a node name
  – a parameter name or value
• No differences between upper and lower case (except in quoted filenames)
• Continuation of statement on next line by plus (+) sign as first non-numeric, non-blank character in the next line
• Indication of “to the power of” by two asterisks (**) 
  – E.g. 2**5 == two to the fifth power (2^5)
• All characters after the listed statement lines will be ignored:
  – .include 'filename'
  – .lib 'filename' corner
  – .enddata, .end, .endl, .ends and .eom
  – For example:
    • .include 'biasckt.inc'; $ semicolon ignored
    • .lib 'mos25l.l' tt, $ comma ignored

Source: Synopsys, 2007
First Character

• First character in every line specifies how HSPICE interprets the remaining line

• First line of a netlist:
  – Any character
  – Title or comment line

• Subsequent lines of netlist, and all lines of included files:
  – .(XXXX): Netlist keyword (e.g.: .TRAN 0.5ns 20ns)
  – * (asterisk): Comment line (HSPICE)
  – + (plus): Continues previous line
Numbers

- Numbers can be
  - Integer
  - Floating point
  - Floating point with integer exponent
  - Integer or floating point with one scale factor

- Numbers can use:
  - Exponential format
  - Engineering key letter format
  - Not both (1e-12 or 1p, but not 1e-6u)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Scale Factor</th>
<th>Multiplying Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tera</td>
<td>T</td>
<td>1e+12</td>
</tr>
<tr>
<td>Giga</td>
<td>G</td>
<td>1e+9</td>
</tr>
<tr>
<td>Mega</td>
<td>MEG or X</td>
<td>1e+6</td>
</tr>
<tr>
<td>Kilo</td>
<td>K</td>
<td>1e+3</td>
</tr>
<tr>
<td>Milli</td>
<td>M</td>
<td>1e-3</td>
</tr>
<tr>
<td>Micro</td>
<td>U</td>
<td>1e-6</td>
</tr>
<tr>
<td>Nano</td>
<td>N</td>
<td>1e-9</td>
</tr>
<tr>
<td>Pico</td>
<td>P</td>
<td>1e-12</td>
</tr>
<tr>
<td>Femto</td>
<td>F</td>
<td>1e-15</td>
</tr>
<tr>
<td>Atto</td>
<td>A</td>
<td>1e-18</td>
</tr>
</tbody>
</table>

Source: Synopsys, 2007
Simulation Program Structure

Simulation Experiment

- Single point Analysis
- Optimization
- Sweep
- Statistical Worst Case
- Timing Violations

Analysis

- Initial Conditions
- Circuit
- Results
- Library
- Stimuli

- Transient
- DC
- AC

Options

Source: Synopsys, 2007
Comments

* **** Parameters *****

• Comments:
  – First letter of line is asterisk (*) → whole line is comment
  – Dollar sign ($) anywhere on the line → text after is comment

• For example:
  – * <comment_on_a_line_by_itself>
    -or-
      – <HSPICE_statement> $ <comment_following_HSPICE_input>

• Comment statements can be placed anywhere in circuit description

9/28/2011
Parameters and Expressions

\[ \text{.param } Wn=2u \ L=0.6u \]
\[ \text{.param } Wp='2*Wn' \]

- Definition of netlist parameters
- Parameter can be defined with expressions
- Definition can occur after use in elements
- Parameter names must begin with alphabetic character
- At redefinition last parameter’s definition is used
- Expressions cannot exceed 1024 characters
Sources and Stimuli

* ***** Define power supplies and sources *****

V1 VDD 0 5
VPULSE VIN 0 PULSE 0 5 2N 2N 2N 98N 200N

• Source element statements to specify DC, AC, transient, and mixed voltage and current sources
• Grounding of voltage sources not necessary
  Hspice assumes: positive current flows from positive node, through the source, to negative node
• Independent and dependent voltage/current sources
Simple Sources: Syntax

\[ V_{xx} \ n^+ \ n^- \ DC=dcval \ tranfun \ AC=acmag \ acphase \]
\[ I_{xx} \ n^+ \ n^- \ DC=dcval \ tranfun \ AC=acmag \ acphase \ M=val \]

- **Vxx**: Voltage source element name, **must begin** with V
- **Ixx**: Current source element name, **must begin** with I
- **n+, n-**: Positive and negative node
- **DC=dcval**: DC source keyword and value (in volts)
- **tranfun**: Transient source function
  - One or more of: AM, DC, EXP, PAT, PE, PL, PU, PULSE, PWL, SFFM, SIN
  - Specification of characteristics of a time-varying source
- **AC**: AC source keyword for use in AC small-signal analysis
- **acmag**: Magnitude (RMS) of the AC source (in volts)
- **acphase**: Phase of the AC source (in degrees)
- **M**: Multiplier:
  - Multiplies all values with \textit{val}
  - For simulation of parallel current sources
Simple Sources: Examples

- **VX 1 0 5V**
  - Voltage source VX has 5-volt DC bias
  - Positive terminal connects to node 1
  - Negative terminal is grounded

- **VH 3 6 DC=2 AC=1,90**
  - Voltage source VH has 2-volt DC bias, 1-volt RMS AC bias, with 90 degree phase offset
  - Positive terminal connects to node 3
  - Negative terminal connects to node 6.

- **IG 8 7 PL(1mA 0s 5mA 25ms)**
  - Current source IG
  - Piecewise-linear relationship, which is 1 mA at time=0, and 5 mA at 25 ms
  - Positive terminal connects to node 8
  - Negative terminal connects to node 7

- **VMEAS 12 9**
  - Voltage source VMEAS has 0-volt DC bias
  - Positive terminal connects to node 12
  - Negative terminal connects to node 9
Source Functions

• For transient analysis

• Types:
  – Trapezoidal pulse (PULSE)
  – Sinusoidal (SIN)
  – Exponential (EXP)
  – Piecewise linear (PWL)
  – Single-frequency frequency-modeled (SFFM)
  – Single-frequency amplitude-modeled (AM)
  – Pattern (PAT)
  – Pseudo Random-Bit Generator Source (PRBS)
Trapezoidal Pulse

- \( V_{xx}/I_{xx} n+ n- \) PULSE \( v1 \) \( v2 \) \( td \) \( tr \) \( tf \) \( pw \) \( per \)
  - PULSE: Keyword
  - \( v1 \): Initial value of the voltage or current
  - \( v2 \): Pulse plateau value
  - \( td \): Delay to the first ramp
  - \( tr \): Duration of the rising ramp
  - \( tf \): Duration of the falling ramp
  - \( pw \): Pulse width
  - \( per \): Pulse repetition period

9/28/2011
Sinusoidal Pulse

- Vxx/lxx n+ n- SIN vo va freq td q j
  - SIN: Keyword
  - vo: Voltage or current offset
  - va: Voltage or current peak value
  - freq: Source frequency
  - td: Delay to the first sinus
  - q: Damping factor (in Hz)
  - j: Phase delay (in degrees)

0 to td: \[ v(t) = vo + va \cdot \sin \left( \frac{2\pi \cdot j}{360} \right) \]

from td: \[ v(t) = vo + va \cdot \exp \left( [t - td] \cdot q \right) \cdot \sin \left( 2\pi f [t - td] + \frac{2\pi \cdot j}{360} \right) \]
Element Names

• Names **begin** with the **element key letter** (exception: subcircuits)

• Maximum name length: 1024 characters

• Some element key letters:
  – C: Capacitor
  – D: Diode
  – J: JFET or MESFET
  – L: Linear inductor
  – M: MOS transistor
  – Q: Bipolar transistor
  – R: Resistor
  – T,U,W: Transmission Line
  – X: Subcircuit call
Elements examples

- **R1 n1 n2 20k M=2**
  - Type: Resistor
  - Name: R1
  - Connected nodes: n1, n2
  - Value: 20kΩ * 2 = 40kΩ

- **M1 ADDR SIG1 GND SBS nch ‘w1+w’ ‘l1+l’**
  - Type: MOSFET
  - Name: M1
  - Drain node: ADDR
  - Gate node: SIG1
  - Source node: GND
  - Substrate nodes: SBS
  - Model: nch
  - MOSFET dimensions: algebraic expressions (width=w1+w, length=l1+l)
Node Names

- Nodes **connect elements**
- **Maximum node name length**: 1024 characters
- Can be only numbers
  - Range of 0 to $10^{16}-1$
  - Leading zeros are ignored
  - Characters are ignored if 1. character is number (e.g.: 1 == 1A)
- **.GLOBAL** statement to make node names **global across all subcircuits**
- **0, GND, GND!, GROUND**: refer to the **global ground**
Subcircuit node names

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

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

* ***** Analysis statement *****

.TRAN 1n 300n

• Definition of analysis type (DC, transient, AC, ...)
• At begin of analysis: Determination of DC operating point values for all nodes and sources:
  1. Calculation of all values
  2. Setting values specified in .NODESET and .IC statements
  3. Setting of values stored in an initial conditions file
• Then: Iteratively searching of exact solution
• At transient analysis: resulting DC operating point is initial estimate to solve the next timepoint
• Initial estimates close to exact solution increase likelihood of convergent solution and lower simulation time
Transient Analysis

Simulation Experiment

DC

Transient

AC

UIC

.FOUR

.FFT

Time-sweep simulation

.OPTION:

Method

Tolerance

Limit

Source: Synopsys, 2007
Transient Analysis Cont’d

- **Transient** analysis simulates circuit in a **specific time**
- Simple syntax: `.TRAN <Tstep> <Tstop>`
  - `<Tstep>`: time step
  - `<Tstop>`: End time (duration) of simulation
- Also more complex commands possible
- E.g.: `.TRAN 200P 20N SWEEP TEMP -55 75 10`
  - Time step: 200 ps, Duration: 20 ns
  - Multipoint simulation: temperature is swept from -55 to 70°C by 10°C steps
AC Simulations

Simulation Experiment

DC

Transient

AC

Other AC analysis statements

AC small-signal simulation

.OPTION:

Method

DC options, to solve operating-point

Source: Synopsys, 2007
Output Files

- ***.st# Output Status File**
  - # is 0-9999
  - Start and end times for each CPU phase
  - Options
  - Status of preprocessing checks for licensing
  - Input syntax
  - Models
  - Circuit topology
  - Convergence strategies that for difficult circuits

- ***.mt# Transient Analysis Measurement Results File**
  - If .MEASURE TRAN statement

- ***.tr# Transient Analysis Results File**
  - Numerical results of transient analysis
  - If .TRAN and .OPTION POST statements
Output Files cont’d

- **.lis** Output Listing File
  - Name and version of the simulator
  - Synopsys message block and License details
  - Input filename
  - Copy of the input netlist file and node count
  - Operating point parameters
  - Details of the volt drop, current, and power for each source and subcircuit
  - Low-resolution ASCII plots, originating from a .PLOT statement
- **.ac#** AC Analysis Results File
- **.ma#** AC Analysis Measurement Results File
  - If .MEASURE AC statement
Output Files cont’d

- ***.sw#** DC Analysis Results File
  - If .DC statement
  - Results of applied stepped or swept DC parameters
  - Results can include noise, distortion, or network analysis

- ***.ms#** DC Analysis Measurement Results File
  - If .MEASURE DC statement

- ***.ft#** FFT Analysis Graph Data File
  - Graphical data needed to display the FFT analysis waveforms

- ***.ic#** Operating Point Node Voltages File
  - If .SAVE statement
  - DC operating point initial conditions
Noise Analysis Example

* A Common Source NMOS amplifier

```
.options list post
.model n_tran nmos level=49 version=3.22
+AF=.826 KF=4e-29
vdd vdd 0 DC=5

p1 in 0 port=1 ac=0.1 dc=2.1 z0=50
p2 out vdd port=2 z0=20k
rs in g1 50
m1 out g1 0 0 n_tran l=1.5u w=40u

.ac dec 10 10Meg 10G
.lin noisecalc=1
.print ac v(out) onoise
.end
```

Source: Synopsys, 2007
Noise Analysis Example cont’d

- First step: all the signal voltage and current sources set to 0

Source: Synopsys, 2007
Noise Analysis Example cont’d

- Next step: each resistor, diode, and transistor modeled with its noise model
- Then: calculation of output voltage resulting from the noise signal (one element at a time)
- Here:

  1. Replacement of Rs with its noise model
  2. Calculation of PSD of the noise voltage ($PSD_{Rs}$) as seen at output port for one frequency

PSD: Power Spectral Density

Source: Synopsys, 2007
Noise Analysis Example cont’d

• Here:
  3. Replacement of M1 with its noise model
  4. Calculation of PSD of the noise voltage ($PSD_{M1}$) as seen at output port for same frequency

• Total PSD ($PSD_{total}$) at observed frequency is sum of all PSD [$V^2$/Hz]
  \[ PSD_{total} = PSD_{Rs} + PSD_{M1} \]

Source: Synopsys, 2007
Example: Operating Point & Gain

\[ V_o = V_O + v_o \]

\[ V_i = V_I + v_i \]

\[ 100 \mu A \]

\[ Q_1, Q_2, Q_3, Q_4 \]
spice test
vcc 100 0 2.5
vee 200 0 -2.5
q1 2 3 4 n
q2 6 0 4 n
q3 2 2 100 p
q4 6 2 100 p
vi 3 0 ac
i1 4 200 100u
r1 4 200 1meg
.model n npn bf=200
va=130 rb=200 is=5f
.model p pnp bf=50
va=50 rb=300 is=2f
.op
.tf v(6) vi
.end

9/28/2011
Examples: Find Freq. Response

\[ r_b = 300 \, \Omega, \quad I_C = 0.5 \, mA, \quad \beta = 200, \quad f_T = 500 \, MHz \, (at \, I_C = 0.5 \, mA), \quad C_\mu = 0.3 \, pF, \quad C_{cs} = 0, \, and \, V_A = \infty. \]
• test 2 - CE Freq. Response
• vcc 5 0 5v
• rs 4 2 5k
• rl 1 5 3k
• q1 1 2 0 npn
• vi 4 7 ac
• vdc 7 0 0.8

• .tf v(1) vi

• .model npn npn is=1e-16a bf=200 rb=300 cjc=0.3pf cjs=0 tf=302pf
• .ac dec 10 100k 1000meg
• .plot ac vdb(1)
• .plot ac vp(1)
• .pz v(1) vi
• .op
• .end
DC Characteristic of an NPN

- DC
- vce 1 0 0
- q1 1 2 0 n
- vbe 2 0 0
- .dc vce 0 50 0.5 vbe 0.5 1 0.05
- .model n npn  is=1.26e-015  bf=290  rb=670  rc=300  tf=1.15n  br=2.5  cje=0.65p  cjc=0.36p  cjs=3.2p  vaf=172.5
- .op
- .end