Micro transductors ’08

Introduction to HSpice

Dr.-Ing. Frank Sill

Department of Electrical Engineering, Federal University of Minas Gerais,
Av. Antônio Carlos 6627, CEP: 31270-010, Belo Horizonte (MG), Brazil

franksill@ufmg.br
http://www.cpdee.ufmg.br/~frank/
Recap: Soft Errors

- In 70’s observed: DRAMs occasionally flip bits for no apparent reason
- Ultimately linked to alpha particles and cosmic rays
- Collisions with particles create electron-hole pairs in substrate
- These carriers are collected on dynamic nodes, disturbing the voltage
Recap: Electromigration

Electromigration:
- Transport of material caused by the gradual movement of ions in a conductor
- One of the major failure mechanisms in interconnects
- Inversely proportional to the width and thickness of the metal lines
- Proportional to the current density

Source: Plusquellic, UMBC
Recap: Bathtube Failure Model

**Infant mortality**
- Declining failure rate
- Based on latent reliability defects

**Normal lifetime**
- Constant failure rate
- Based on TDDB, EM, hot-electrons…

**Wearout period**
- Increasing failure rate
- Based on TDDB, EM, etc.

1-40 weeks 7-15 years Time

Failure rate

Copyright Sill, 2008
Recap: Classification

Error
- Permanent
  - Defects, wearout, out of range parameters, EM, TDDB ...
- Temporary
  - Defects, wearout, out of range parameters, EM, TDDB ...
  - Process variations, infant mortality, random dopant fluctuation, ...
  - Radiation
    - Soft errors
  - Non-Radiation
    - Power supply, coupling, operation peaks

Source: Mitra, 2007
Triple Module Redundancy (TMR)

![Diagram of TMR]

- Input
- Logic L
- Copy of Logic L
- Copy of Logic L
- Voter
- Output
Statistical Static Timing Analysis (SSTA)

Deterministic STA: (Worst Case Analysis)

Statistical STA:
Self Adaptive Design

- Extend idea of clock domains to Adaptive Power Domains
- Tackle static process and slowly varying timing variations
- Control VDD, $V_{th}$ (indirectly by body bias), $f_{clk}$ by calibration at Power On

![Diagram of test module and module with VDD, f_{clk}, V_{BB}, and test inputs and responses connections]
Razor Flip-Flop cont’d

Source: Austin, 2004

Copyright Sill, 2008
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
Circuit Analysis Types

- 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
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 ($2^5$)
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

Source: Synopsys, 2007
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>Mikro</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
Basic Netlist structure

Simple inverter circuit
* **** Parameters *****
.param Wn=2u L=0.6u
.param Wp='2*Wn'
* ***** Define power supplies and sources *****
V1 VDD 0 5
VPULSE VIN 0 PULSE 0 5 2N 2N 2N 98N 200N
* ***** Actual circuit topology *****
M1 VOUT VIN VDD VDD pch Wp L M=1
M2 VOUT VIN GND GND nch Wn L
* ***** Analysis statement *****
.TRAN 1n 300n
* ***** Output control statements *****
.OPTION POST
.PRINT V(VIN) V(VOUT)
* **** Library *****
.LIB ‘AMS.lib’ nominal
.END
Title of Simulation

Sample inverter circuit

- First line is title of simulation → statements are ignored
- Included files: same rule
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
Parameters and Expressions

\[ \text{.param } W_n = 2u \quad \text{L} = 0.6u \]
\[ \text{.param } W_p = '2\times W_n' \]

- 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 *****

\[ V1 \text{ VDD 0 5} \]
\[ \text{VPULSE VIN 0 PULSE 0 5 2N 2N 2N 98N 200N} \]

- 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

Vxx n+ n- DC=dcval tranfun AC=acmag acphase
lxx 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 val
  - For simulation of parallel current sources
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

- $V_{xx/lxx\ n+\ n-}$ 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
Circuit topology

* ***** Actual circuit topology *****
M1 VOUT VIN VDD VDD pch Wp L M=1
M2 VOUT VIN GND GND nch Wn L M=1

- Netlist of applied elements
- Connection of elements by nodes
- Element statements specify:
  - Type of device
  - Nodes to which the device is connected
  - Operating electrical characteristics of the device
- Passive elements (resistors, capacitors, inductors, …) need no model type
- Active elements (transistors, diodes, …) need model type
- Element multiplier M replicates all values (not negative, zero)
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
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
Example

M1 VOUT VIN VDD VDD pmos_AMS Wp L
M2 VOUT VIN GND GND nmos_AMS Wn L
Subcircuits

- Subcircuits for commonly-used circuit
- Definition with `.SUBCKT` and `.ENDS`

**Use** $X<\text{subcircuit\_name}>$ to call a subcircuit
- $<\text{subcircuit\_name}>$: element name of the subcircuit
- Up to 15 characters

**.INCLUDE** statement includes other netlist as subcircuit into current netlist (e.g.: `.INLCUDE <path>/nand.sp`)

Subcircuit example:
- `.SUBCKT Inv A Y Wid=0`
- `mp1 Y A VDD VDD pch L=1u W='Wid*2'`
- `mn1 Y A 0 0 nch L=1u W=Wid`
- `.ENDS`

$Xinv1$ in out Inv Wid=1u
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

.OPTION:
- UIC: Method
- FOUR: Tolerance
- FFT: 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
.PRINT Statement

```plaintext
.print <ana_type> ov1 [ov2 ... ovN]
```

- Output from the .PRINT statement saved in *.print file
  - Header line: column labels.
  - First column: time
  - Remaining columns: output variables specified with .PRINT
  - Rows after header line: data values for simulated time points

- `<ana_typ>`: type of analysis (tran, dc, ac, ..)

- `oVx` can be:
  - `V(n)`: voltage at node $n$.
  - `V(n1<n2>`: voltage between the $n1$ and $n2$ nodes.
  - `Vn(d1)`: voltage at $n$th terminal of the $d1$ device.
  - `In(d1)`: current into $n$th terminal of the $d1$ device.
  - `expression`: expression, involving the plot variables above
.MEASURE Statement

- .MEASURE statement produces a measurement parameter
- .MEASURE <ana_type> <param_name> <meas_mode>
  - <param_name>: Parameter name
  - <Meas_mode> Measurement mode, e.g.:
    - Rise, fall, and delay
    - Find-when
    - Average, RMS, min, max, and peak-to-peak
    - Integral evaluation
    - Derivative evaluation
- E.g.: .MEASURE tran vin AVG V(nt1) from=0 to=1n
  - Parameter name: vin
  - Measurement type: Average
  - Value: Voltage of net n1
Libraries

* **** Library *****

.LIB ‘AMS.lib’ nominal

- Libraries include model files
- Model files contain information about behavior of applied elements (.MODEL statement)
- .MODEL statement can be also placed in netlist
- Applied Model file for simulation chosen by option
- Syntax: .LIB <library> <option>
- Libraries can also contain commonly-used commands, subcircuit analysis, and parameters
Practical issues

- How to start a simulation?
  - user@ws:> cd hspice
  - user@ws:> hspice trans.sp

- How to measure delay?
  - Delay from 50% of input slope to 50% of output slope
  - `.meas tran tdelay trig v(in) VAL = 2.5 RISE = 1 +TARG v(out) VAL = 2.5 FALL = 1`

- How to analyze data?
  - CosmosScope
  - user@ws:> cscope
CosmosScope

- New Results: File > Open Plotfiles > *.trxxx

- Zoom

- Trace

- Signal Manager
  - select simulation
  - select signals

- Refresh signals after simulation

- Calculator
Exercises

- **Inverter** (*trans_inv.sp, inv.inc*)
  - Simulate the circuit and look at the output signals.
  - Draw the circuit (elements, names, node names)
  - What is the maximum delay?
  - How can I determine the dynamic power dissipation?
  - What is the dynamic power dissipation?

- **Chain of Inverter** (*trans_inv-chain.sp, inv.inc*)
  - What is the delay?
  - What can you do to reduce the delay?
  - What is the minimum delay?
Exercises cont’d

- **Dynamic Power dissipation** (*trans_inv_pdyn.sp, inv.sp*)
  - What is the dynamic power dissipation?
  - What happens if you modify the load? Simulate.
  - What else can you do to change the dynamic power dissipation? Simulate.
  - What happens with the delay?

- **Leakage** (*trans_circuit.sp, nand.sp, nand_hvt.sp*)
  - Draw the circuit.
  - How can you determine the leakage? Estimate.
  - How can you reduce the leakage at constant performance? Design and Simulate.
  - How differ leakage, delay and threshold voltage of the design or the transistor, respectively?