Advanced Computational Methods for VLSI Systems

Lecture 1 Introduction
Zhuo Feng
■ **Prof. Zhuo Feng**
  ► Office: EERC 513
  ► Phone: 487-3116
  ► Email: zhuofeng@mtu.edu

■ **Class Website**
  ► [http://www.ece.mtu.edu/~zhuofeng/EE5900Spring2013.html](http://www.ece.mtu.edu/~zhuofeng/EE5900Spring2013.html)
  ► Check the class website for lecture materials, and announcements

■ **Schedule**
  ► TR 9:35am-10:50am EERC 218
Textbook


References

Topics (tentative)

- Brief review of linear algebra and numerical methods
- Basic circuit analysis methods
- RF circuit simulation methods
- Statistical timing analysis methods
- Fast algorithms for IC modeling and simulations
- Parallel CAD algorithms
Project

- This is a project/research oriented course.
- Research explorations are strongly encouraged.
- Students are expected to read research papers assigned before lectures and actively participate in class discussions.

Grading

- Course project 1: 40% (a SPICE simulator)
- Course project 2: 60% (flexible research projects)
A plethora of VLSI CAD problems

Devices, interconnects, circuits, systems, signal, power, analog, digital .....
Where are we in the design flow?

- Specifications
- System-level Design
- RTL-level Design
- Gate-level Design
- Transistor/Circuit Level
- Layout
- Final Verification

Top-down Design

Bottom-up Verification

Electrical & Thermal Properties, Delays, Waveforms, Parasitics Effects, Coupling Noise ...
Why circuit analysis?

► Performance verification

▼ A critical step for evaluating expected performance prior to manufacturing

▼ Simulation is always cheaper and more efficient than actually making the chip

▼ True more than ever for today’s high manufacturing costs
Why circuit simulation (cont’d)?

Design optimization/synthesis

- Need to evaluate circuit performances *many times* in an optimization loop before meeting all the specs
- Can only be practically achieved via simulation (models)

---

**Diagram:**
- **Circuit Optimizer**
  - Update design parameters
- **Simulation Engine**
  - Performance Evaluation
  - Meet all specs?
    - No: Update design parameters
    - Yes: Convergence
Why models?

- Models are integral parts of system simulation and/or optimization

---

- Abstract Executable Models
  - High
  - Low

- Cycle Accurate Models
  - High
  - Low

- VHDL/Verilog Models

- Interconnect Gate Models
  - Low
  - High

- Device Models
Assessment of simulators

Accuracy  
Runtime  
Memory

Accuracy  
Runtime  
Memory

Accuracy  
Runtime  
Memory

Robustness/Applicability
Selected Topics

- Classical circuit simulation methods (SPICE)
  - LU factorization, Newton’s method
  - (Modified) nodal formulation (MNA)
  - Nonlinear DC analysis
  - AC analysis
  - Linear/nonlinear transient analyses
  - SPICE device models
Many problems are modeled by some form of coupled (nonlinear) first-order differential equations.

For circuit problems this is usually done using MNA formulation.
Write KCL (Kirchoff’s Current Law) at node 1:

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

If we do this for all N nodes:

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

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

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

\[ F = \text{nonlinear operator} \]
How can we solve this set of nonlinear differential equations?

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

- Closed-form formula/hand analysis easily becomes infeasible
- Need to develop computer programs (circuit simulators) to solve for the solution numerically
Simple example: nonlinear LC ckt with sinusoidal input current

\[ \frac{d^2v}{dt^2} + \frac{v}{LC(v)} = A_1 \cos(\omega_0 t) \]

\[ C(v) = \frac{1}{\lambda^2 + \mu v^2}; L = 1 \]

2nd order nonlinear diff. equation:

\[ \ddot{v} + \lambda^2 v + \mu v^3 = A_1 \cos(\omega_0 t) \]
Selected Topics

► Statistical timing analysis & optimization

Variability

- Process Characterization
- Interconnect Gate Models
- Delay Calculators

[Diagram of circuitry with bell curves and labels for various processes and models]
Selected topics

► Parallel CAD
GPU Computing for CAD

- Dedicated for graphics rendering

GPU is connected to MCH via 16x PCI-Express

Video Out

CPU (host)
Reading assignments
- Textbook. Chapters 1 and 2

History of SPICE
- CANCER at UCB: Computer Analysis of Nonlinear Circuits Excluding Radiation
- Government project SCEPTRE: System for Circuit Evaluation and Prediction of Transient Radiation Effects
- CANCER evolved to SPICE: Simulation Program with Integrated Circuit Emphasis
- SPICE became the industry standard
- **SPICE overview**
  - N equations in terms of N unknown Node voltages
  - More generally using modified nodal analysis

\[ i = G(v) \]

\[ v_2 + v = v_1 - v_3 \]
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:

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

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

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

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

- We must approximate the solution of:

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

- 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>10Ω</td>
</tr>
<tr>
<td>$R_2$</td>
<td>1</td>
<td>2</td>
<td>5Ω</td>
</tr>
<tr>
<td>$R_3$</td>
<td>2</td>
<td>0</td>
<td>100Ω</td>
</tr>
</tbody>
</table>
The nodal analysis matrix equations are easily constructed via KCL at each node:

\[ Y\vec{v} = \vec{J} \]
- **Naïve approach**
  - a) Write down the KCL eqn for each node
  - b) Combine all of them to 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. \(j\)

\(R\)
RHS \( \vec{j} \) of equations are stamped in a similar way:

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

From row \( i \)

To row \( j \)

\[
\begin{pmatrix}
i \\ I \\ j
\end{pmatrix}
\]
Stamping our simple example one element at a time:

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

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

\[ v_{k\ell} \quad I = 0 \]

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

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 voltage sources are easily accommodated.

\[
\begin{bmatrix}
1 & \cdots \\
\vdots & \ddots & \ddots \\
\vdots & & \ddots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
v_1 \\
v_2 \\
\vdots \\
\end{bmatrix}
= 
\begin{bmatrix}
2 \\
\vdots \\
\end{bmatrix}
\]
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
► 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:

\[
\begin{bmatrix}
    1 & 0 \\
    -1 & 0 \\
    1 & -1 \\
\end{bmatrix}
\begin{bmatrix}
i_k \\
i_\ell \\
i_{N+1}
\end{bmatrix}
= \begin{bmatrix}
V_k \\
V_\ell \\
V_{N+1}
\end{bmatrix}
\]
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 ckt's are possible:

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

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

For large circuits, the matrix is really sparse:
- Number of entries in \( Y \) is a function of the 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.
Reading assignment:
► Textbook, sections 3.1-3.4, 7.1-7.5

We need to solve a linear matrix problem for the simple DC circuit

Linear system solutions are also basic routines for many other analyses
► Transient analysis, nonlinear iterations etc

Sounds quite easy but can become nontrivial for large analysis problems
► $O(n^3)$ – general dense matrices
► $O(n^{1.1} \sim n^{1.5})$ -- sparse circuit problems
► $O(n^{1.5})$ – 2D mesh  $O(n^2)$ – 3D mesh