Technology and trends for static timing analysis in deep submicron digital integrated circuits

Ken Shepard
Department of Electrical Engineering
Columbia University
shepard@ee.columbia.edu

© Kenneth L. Shepard. All rights reserved
Outline

• Technology trends
• Interconnect
  – RC delays
  – Capacitive coupling
  – Inductance
• Transistor-level timing
• Hierarchical techniques
Design Complexity

- Tens of millions of transistors
- Static analysis techniques essential
- Hierarchical analysis essential
Performance Trends

Year

1980  1990  2000

Linewidth

2.5  1.2  0.6  0.35  0.1

Frequency

10 MHz  30 MHz  100 MHz  300 MHz  1 GHz

Cu wires

© Kenneth L. Shepard. All rights reserved
Technology Trends in Interconnect

- Lateral coupling capacitance is increasing
- RC delay continues to grow
- Increasing faster
- Total capacitance is decreasing but resistance is scaling
Technology Trends in Interconnect

• Materials changes
  – low resistivity metals (e.g. Cu)
  – low & dielectrics
Technology Trends in Interconnect

• Materials changes
  – low resistivity metals (e.g. Cu)
  – low $\varepsilon$ dielectrics

• Consequences
  – inductance problem growing

\[ R_{\text{line}}, R_{\text{driver}} < Z_0 = \sqrt{\frac{L}{C}} \]

© Kenneth L. Shepard. All rights reserved
Division of interconnect analysis

ground

supply

© Kenneth L. Shepard. All rights reserved
Division of interconnect analysis

- Separate power supply integrity from signal propagation.
- Power and ground treated as equipotentials in signal extraction.
RC Delays in Static Timing
RC Delays in Static Timing

Partition at gate with cap to ground
RC Delays in Static Timing

- Estimated routes
- Detailed extraction
Load Model

\[ y(s) = m_1 s + m_2 s^2 + m_3 s^3 + \cdots \]

\[ C_{\text{near}} = \frac{m_2}{m_3} \]
\[ C_{\text{far}} = m_1 - \frac{m_2}{m_3} \]
\[ R = -\frac{m_2^2}{m_3^2} \]

© Kenneth L. Shepard. All rights reserved
Load Model

\[ y(s) = m_1 s + m_2 s^2 + m_3 s^3 + \cdots \]

\[ C_{\text{near}} = \frac{m_2}{m_3} \]

\[ C_{\text{far}} = m_1 - \frac{m_2}{m_3} \]

\[ R = -\frac{m_3}{m_2} \]

© Kenneth L. Shepard. All rights reserved
Load Model

\[ y(s) = m_1 s + m_2 s^2 + m_3 s^3 + \cdots \]

\[ C_{\text{near}} = \frac{m_2}{m_3} \]
\[ C_{\text{far}} = \frac{m_1 - m_2}{m_3} \]
\[ R = -\frac{m_3^2}{m_2^3} \]

Saturate ramp abstraction

© Kenneth L. Shepard. All rights reserved
Net transfer function
Net transfer function

\[
\frac{V_{\text{receiver}}}{V_{\text{input}}} = \frac{C_1}{R_1} + \frac{C_2}{R_2} + \frac{C_3}{R_3}
\]
Net transfer function
What about coupling?
Simple, simple approach

Miller Effect

© Kenneth L. Shepard. All rights reserved
Simple, simple approach

Miller Effect

Nominal

© Kenneth L. Shepard. All rights reserved
Simple, simple approach

Miller Effect

Worst-case

© Kenneth L. Shepard. All rights reserved
Simple, simple approach

Miller Effect

Best-case
Net Complexes

- For each (primary) net, build a complex of (secondary) nets with significant coupling to the primary net.
- Introduces some “redundancy”
Net Complexes

- For each (primary) net, build a complex of (secondary) nets with significant coupling to the primary net.
- Introduces some “redundancy”
Net Complexes

- For each (primary) net, build a complex of (secondary) nets with significant coupling to the primary net.
- Introduces some “redundancy”
Net Complexes

- For each (primary) net, build a complex of (secondary) nets with significant coupling to the primary net.
- Introduces some “redundancy”
Multiport Macromodels

With the increased importance of coupling in timing and delay analysis, multiple drivers must frequently be considered in the analysis of a piece of the interconnect network.

Multiport models can take the form of admittance, impedance, hybrid, or scattering parameters.

Admittance has the advantage of being compatible with SPICE-like simulators.
Hybrid transfer/admittance multiports

- For fixed, “linearizable” loads, as when driving MOS gates, it is convenient to reduce the number of driving ports.
- At these “tap points”, we instead calculate a transfer function from each of the ports.

\[
\begin{bmatrix}
  i_{\text{port}} \\
  v_{\text{tap}}
\end{bmatrix} = \begin{bmatrix} Y & H \end{bmatrix} v_{\text{port}}
\]
## Summary of Multiport Techniques

<table>
<thead>
<tr>
<th>Moments</th>
<th>Passive RC?</th>
<th>Passive RLC?</th>
<th>Number of state matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block PVL</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Block syPVL</td>
<td>Yes</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Block Arnoldi</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Prima</td>
<td>Yes</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>

**Conclusion:**
Block syPVL better for RC only. Prima better for RLC.
Recursive Convolution

This is an efficient technique for convolving an input waveform with a transfer function. **Constant cost at each time point.**

\[ v_{receiver}(t) = \int_{0}^{t} H(t - \tau) v_{driver}(\tau) d\tau \]

\[ = \sum \left[ + + - \right] \]

\[ = - + \left( - + - \right) - \]

\[ = - + \left( - \right) - \]

\[ = \left( - - + - \right) \]
Recursive Convolution and Circuit Simulation

© Kenneth L. Shepard. All rights reserved
Recursive Convolution and Circuit Simulation

When applied to multiport driving point admittances, the recursive convolution approach can provide a convenient stencil to incorporate reduced-order models into the MNA formulation of SPICE-like circuit simulators.

© Kenneth L. Shepard. All rights reserved
Multiport models and coupling in static timing analysis

- Extracted netlist
- Nonlinear circuit elements
- Partitioning
- Net complexes
Multiport models and coupling in static timing analysis

- Extracted netlist
- Nonlinear circuit elements
- Net complexes
- Reduction engine
- Gate capacitances

© Kenneth L. Shepard. All rights reserved
Multiport models and coupling in static timing analysis

- Extracted netlist
- Partitioning
- Nonlinear circuit elements
- Net complexes
- Gate capacitances
- Reduction engine
- Simulation with recursive convolution
- Hybrid multiport models
Multiport models and coupling in static timing analysis

- Extracted netlist
- Nonlinear circuit elements
- Simulation with recursive convolution
- Gate capacitances
- Net complexes
- Reduction engine
- Hybrid multiport models
- Calculated waveshapes
- Static timing analysis engine
Multiport models and coupling in static timing analysis

- Extracted netlist
- Partitioning
- Net complexes
- Nonlinear circuit elements
- Gate capacitances
- Simulation with recursive convolution
- Reduction engine
- Hybrid multiport models
- Calculated waveshapes
- Static timing analysis engine

© Kenneth L. Shepard. All rights reserved
General approach

Perform an initial timing with all secondary nets grounded
General approach

- Perform an initial timing with all secondary nets grounded
- Freeze the arrival time windows and slews at each driver

© Kenneth L. Shepard. All rights reserved
General approach

- Perform an initial timing with all secondary nets grounded
- Freeze the arrival time windows and slews at each driver
- Reset the timing model and recalculate all delays including the effects of coupling
General approach

1. Perform an initial timing with all secondary nets grounded.
2. Freeze the arrival time windows and slews at each driver.
3. Reset the timing model and recalculate all delays including the effects of coupling.
General approach

- Tends to converge well for well-tuned designs
- Poorly tuned designs can have convergence problems
Determining secondary net arrival times
Determining secondary net arrival times
Transistor-level timing
Transistor-level timing

More smarts in preconditioning and sensitization

- Honoring input logic constraints for “side-branch limit” and path pruning
- Simultaneous switching

© Kenneth L. Shepard. All rights reserved
Hierarchy?
Hierarchical Techniques

Global

Macro

© Kenneth L. Shepard. All rights reserved
Hierarchical Techniques

Global

Abstracts

Macro

Assertions

© Kenneth L. Shepard. All rights reserved
Hierarchical Abstraction
Hierarchical Abstraction

Internal timing points
- transparent latches
- dynamic gates

\[ t_{\text{delay}} = k_0 + k_1 C_L + k_2 C_L^2 \]
\[ t_{\text{slew}} = k_0' + k_1' C_L + k_2' C_L^2 \]

© Kenneth L. Shepard. All rights reserved
Hierarchical Abstraction

Internal timing points
- transparent latches
- dynamic gates

DCL as a standard for abstraction

\[ t_{\text{delay}} = k_0 + k_1 C_L + k_2 C_L^2 \]
\[ t_{\text{slew}} = k_0' + k_1' C_L + k_2' C_L^2 \]

Put in the whole last CCC!
Timing Assertions
Timing Assertions

Primary inputs

- Input capacitance limit or input drive strength
- Input arrival times with clock reference edges
- Phase assertions

© Kenneth L. Shepard. All rights reserved
Timing Assertions

Primary inputs
- Input capacitance limit or input drive strength
- Input arrival times with clock reference edges
- Phase assertions

Primary outputs
- Output load model
- Output required arrival times with clock reference edges
- Phase assertions

© Kenneth L. Shepard. All rights reserved
Conclusions

• Static timing analysis critical
• Transistor-level analysis an essential core competency for all static timing analyzers
• Coupled multiport interconnect models will become an essential part of static timing methodologies (late stage final verification)