Lecture 2: Modeling interconnect delay

Modeling and Optimization of VLSI Interconnect

1) "Ideal" Interconnect (R = 0, C = 0, L = 0)
2) Capacitive interconnect (C ≠ 0)
3) Resistive interconnect (R ≠ 0, C ≠ 0)
4) Inductive interconnect (R ≠ 0, C ≠ 0, L ≠ 0)

The Typical Structure

Interconnect Tree:

Driver

Receivers
The modeling tradeoff

Accuracy

Complexity

Circuit simulation vs. Timing Analysis
Waveform abstraction for timing analysis

Definitions: Don't take them too seriously

Figure 3.1: Pictorial depiction of the 50% delay and transition time.
Distributed RC Line

Response of Single RC

\[ V_{out}(t) = V_{in}(t) \times e^{-\frac{t}{RC}} \]

\[ h(t) = g(t) = \omega RC \]

\[ V_{out} = 0.69RC \times e^{-0.5RC} \]

\[ V_{out} = 0.69RC \times e^{-0.63RC} \]

\[ 1/RC \]

\[ 0.5 \]

\[ 0.63 \]

\[ 0.9RC \]

\[ 1 \]

\[ V_{in}(t) \]

\[ V_{out}(t) \]
<table>
<thead>
<tr>
<th>RC Network</th>
<th>Time Elapsed</th>
<th>Output Potential Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lumped</td>
<td>0.1 RC</td>
<td>0% to 10%</td>
</tr>
<tr>
<td>Distributed</td>
<td>0.4 RC</td>
<td>0% to 50%</td>
</tr>
<tr>
<td></td>
<td>0.5 RC</td>
<td>0% to 63%</td>
</tr>
<tr>
<td></td>
<td>0.9 RC</td>
<td>10% to 90% (rise time)</td>
</tr>
<tr>
<td></td>
<td>1.0 RC</td>
<td>0% to 90%</td>
</tr>
</tbody>
</table>

**Lumped vs. Distributed RC**
The Transfer Function

\[
V_{\text{out}}(t) = (s) H(s) \times (s) A \Lambda V_{\text{in}}(t)
\]

\[
H(s) = \frac{V_{\text{out}}(s)}{V_{\text{in}}(s)} = \frac{H(s)}{V_{\text{out}}(s)} \Lambda V_{\text{in}}(t)
\]

\[
\boxed{\begin{array}{c}
S \\
V_{\text{in}}(t) \\
V_{\text{out}}(t)
\end{array}}
\]

\(\text{π – network interconnect model}\)
Dominant Poles

However, only a few of these poles greatly influence the system response. Such poles are referred to as dominant poles.
What are we trying to do?

\[ \text{Compute the delay of a wire driven by a step-function voltage:} \]

Two cases we know

1. Why is it difficult in other (general) cases?
Moments of the Transfer Function

\[ m_k = \int_0^\infty K_k \left( \int_0^\infty (I - \gamma) s^k \int_0^\infty e^{-\gamma s} \right) dt \]

The Elmore Delay Metric

Source: "Timing", S. Sapatnekar 2004
Elmore delay expressions for RC tree

Elmore delay

\[
\begin{align*}
\text{Downstream formula:} & & \sum_{k} R_k C_k \cdot C_k \cdot (R_k + 1) \\
\text{Upstream formula:} & & \sum_{k} R_k C_k \cdot C_k \cdot (R_k + 1)
\end{align*}
\]

**Figure 1:** RC tree network with discrete elements

Elmore delay of 2-stage RC

\[
(1) \quad V_{\text{out}}(t) = V_{\text{in}}(t) - \sum R_k C_k \cdot C_k \cdot (R_k + 1)
\]

\[
H(s) = \frac{1 - 3RCs + 8(RC)^2 s^2 + 21(RC)^3 s^3 + 55(RC)^4 s^4}{s^4 + 55(RC)^4 s^4 + 21(RC)^3 s^3 + 8(RC)^2 s^2 + 3RCs - 1}
\]
Let's try Elmore's formulas on wire models:

\[ V_{in}(t) \]

Let's try it with \( n \) segments:
Relation of Elmore to dominant pole

RC tree: Examples of upstream resistance

\[ I_{\text{Elmore}} (t) = \sum R C \]

[Diagram of an RC tree with various resistors and capacitors labeled.]
Example of Elmore delay calculation

Algorithm to compute Elmore in a tree

Downstream formula:

\[ D_i = \sum_{k \in \Pi(i)} R_k L_k \]

where:
- \( D_i \) is the total downstream capacitive load driven by \( R_k \).
- \( L_k \) is the total downstream capacitive load driven by \( R_k \).

Figure 1. RC tree network with discrete elements.
Elmore delay as upper bound of 50% delay

Advantages/Disadvantages of Elmore
Asymptotic Waveform Evaluation (AWE)

The essential idea:
Derive an approximation to \( H(s) \), with more than one pole.

Example:

\[
\frac{\frac{1}{s^2}q + s^1q + 0}{s^2q + 0} = \cdots + e^{-sRC}q - 21(1RC)q - 3RC + 8(1RC)q - \cdots
\]

\[
V_{\text{out}}^*(t) = C \int \frac{V_{\text{in}}(t)}{R} dt
\]

Resistive Shielding
Step 1: Find circuit moments efficiently (A trick based on DC circuit solutions)

Step 2: Find a Pade approximation (watch for stability!)

Step 3: Find poles and residues

Step 4: Inverse Laplace transform to sum of exponentials:

\[
\frac{b_d - s}{b} + \cdots + \frac{z_d - s}{z} + \frac{1_d - s}{1} = (s) H
\]

Delay metrics based on probability functions

\[
\frac{1 + s^1 d + \cdots + b s^b d}{0q + s^1 q + \cdots + d s^d q} = (s)^{b,d} H
\]

\[
(s)^{O} + s^{1-\mu} + \cdots + s^{1-\mu} + 0 \mu = (s) H
\]
For capacitive load:

\[ V_{\text{in}}(t) \]

\[ V_{\text{out}} \]

Modeling the Driver

Circuit Reduction
Switch-level RC Model

Driver’s effective resistance

Driver's effective resistance
Switch level for purely capacitive loads

The big idea: Keep driver characterization unchanged – pretend that the load is just a capacitance.

**Effective capacitance for delay under RC loads**

Step 1: Find the effective capacitance

Step 2: Using C_{eff} and a nonlinear model for the driver,

find the waveform at the driving point

Step 3: Use the waveform to drive a linear RC model of the interconnect.
Again: Stage / gate / wire delay

\[ T \approx 0.7R_C + 0.4R_i \]

\[ T \approx 2.3R_C + 0.1R_i \]

\[ T = R_C + 2.3R_C + 0.1R_i \]

Effective Capacitance Modeling
A second look at the bottom of deep submicron

How to separate gate delay from wire delay?

How strong should the driving gate be?

Conclusion:

50K-100K gate blocks are OK for traditional design flow.