Effective Capacitance for the RC Interconnect in VDSM Technologies

Soroush Abbaspour and Massoud Pedram
Department of Electrical Engineering-Systems
University of Southern California

Outline

• Background
• Prior Work
• A New Algorithm for Calculating the Effective Capacitance
• Experimental Results
• Conclusion
Circuit Delay

The circuit delay in VLSI circuits consists of two components:

1. the 50% propagation delay of the driving gates (known as the gate propagation delay)
2. the delay of electrical signals through the wires (known as the interconnect propagation delay)

\[ \text{Delay}_{AC} = \text{Delay}_{AB} + \text{Delay}_{BC} \]

Gate Delay

The gate load delay is a function of both input slew and the output load.
Library-based Delay Model

A pair of two-dimensional delay tables, one for providing the gate delay, the other for the output rise/fall time as a function of the effective load and the input transition time.

Second RC-p Model for Load

Using Taylor Expansion around $s = 0$

$$Y_{in}(s) = A_1 s + A_2 s^2 + A_3 s^3 + ....$$

$$\hat{Y}_{in}(s) = (C_1 + C_2)s - R\pi C_2^2 s^2 + R\pi^2 C_2^3 s^3 + ....$$

$$C_1 = A_1 - \frac{A_2^2}{A_3}$$

$$R\pi = -\frac{A_1^2}{A_2^3}$$

$$C_2 = \frac{A_2^2}{A_3}$$
Second RC-p Model (Cont’d)

Therefore, it is required to create a four-dimensional table to achieve high accuracy

This is however costly in terms of memory space and computational requirements

Effective Capacitance Approach

The “Effective Capacitance” approach attempts to find a single capacitance value that can be replaced instead of the RC-π load such that both circuits behave similar during transition
Effective Capacitance (Cont’d)

Because of the shielding effect of the interconnect resistance, the driver will only “see” a portion of the far-end capacitance $C_2$.

\[
R_\pi \rightarrow 0 \quad k = 1
\]

\[
R_\pi \rightarrow \infty \quad k = 0
\]

Prior Work - Macys’s Approach

Assumption: If two circuits have the same loads and output transition times, then their effective capacitance are the same. In other words, the effective capacitance is only a function of the output transition time and the load.
Macys’s Approach (Cont’d)

1. Compute $\alpha$ from $C_1$ and $C_2$
2. Choose an initial value for $C_{eff}$
3. Compute $t_{out}$ for the given $C_{eff}$ and $T_{in}$
4. Compute $\beta$
5. Compute $\gamma$ from $\alpha$ and $\beta$
6. Find new $C_{eff}$
7. Go to step 3 until $C_{eff}$ converges
Prior Work - Qian’s Approach

Calculate the effective capacitance by equating the currents at the gate output by using:

(a) the driving-point admittance as the load
(b) using a single effective capacitance as the load

Average currents for both loads models are equated until the gate output voltage reaches the 50% threshold

Outline

- Background
- Prior Work
- A New Algorithm for Calculating the Effective Capacitance
- Experimental Results
- Conclusion
A New Effective Capacitance Algorithm

\[ V_M(t) = \begin{cases} \frac{V_{dd}}{T_R}(t - B + A\text{e}^{-at}\text{cosh}(a\phi + \phi)) & 0 \leq t \leq T_R \\ \frac{V_{dd}}{T_R}(T_R + A\text{e}^{-at}\text{cosh}(a\phi + \phi)) & T_R < t \end{cases} \]

New Algorithm (Cont’d)

\[ V_T(t) = \begin{cases} \frac{V_{dd}}{T_R}(1 - R_\text{d}C_{\text{eff}} + R_\text{d}C_{\text{eff}}e^{\frac{t}{R_\text{d}C_{\text{eff}}}}) & 0 \leq t \leq T_R \\ \frac{V_{dd}}{T_R}(t - \frac{R_\text{d}C_{\text{eff}}e^{\frac{t}{R_\text{d}C_{\text{eff}}}}}{R_\text{d}C_{\text{eff}}}) & T_R < t \end{cases} \]
**Eff_Cap Equation**

\[
C_{\text{eff}} = (C_1 + C_2) \frac{1 - e^{-\alpha t} \cosh(\omega t + \phi)}{\cosh(\phi) \frac{t}{R_d C_{\text{eff}}}} (1 - e^{-\frac{t}{R_d C_{\text{eff}}}})
\]

This is an Non-Linear Iterative Equation

A good initial value for \(C_{\text{eff}}\) can speed up the procedure to find the answer

---

**Initial Guess**

\((C_1=15\text{fF}, C_2=20\text{fF})\)

\(C_{\text{eff}}\) (fF)

\(R_\pi\) (KΩ)

\(C_{\text{eff}} = C_1 + \frac{R_d}{R_d + R_\pi} C_2\)

(a) \text{driver size}=500\lambda, T_R=100pS

(b) \text{driver size}=100\lambda, T_R=200pS
Iterative Procedure to Calculate $C_{eff}$

1. Start with the initial guess for $C_{eff}$
2. Obtain $t_{0-50\%}$ based on values of $C_{eff}$ and $T_R$
3. Obtain $R_d$ based on values of $C_{eff}$ and $T_R$
4. Compute a new value of $C_{eff}$ from the Eff_Cap equation
5. Find new $t_{0-50\%}$ based on the new $C_{eff}$ and given $T_R$
6. Compare the values of $t_{0-50\%}$ from step 5
7. If not within acceptable tolerance, then return to step 3 until $t_{0-50\%}$ converges
8. Report $t_{50\%}$ propagation delay and $t_{0-80\%}$ from the table

Extension to Complex Gates

To extend the previous algorithm to complex gates, we only need to compute the value of $R_d$

The gate output driver resistance changes as a function of the applied input waveforms
Due to the body effect, this value is over-estimated

Our approach (using body effect coefficients):

\[ W_{\text{eff}}^{neff} = K_n \frac{1}{K_{b1} + K_{b2} + \ldots + K_{b\text{inputs}}} \]

\[ W_{\text{eff}}^{peff} = K_p \frac{1}{K_{p1} + K_{p2} + \ldots + K_{p\text{inputs}}} \]

\( K_i \) is set to 1 if the corresponding input is not switching; Read \( K_i \)'s from a lookup table if the correspondent input is switching
### Experimental Results

#### Inverter Size (Wp/Wn) mm

<table>
<thead>
<tr>
<th>Inverter Size (Wp/Wn) mm</th>
<th>C1 (pF)/R1 (W)/C2 (pF)</th>
<th>HSPICE 50% delay (pS)</th>
<th>Estimated 50% delay (pS)</th>
<th>Error</th>
<th>HSPICE 80% delay (pS)</th>
<th>Estimated 80% delay (pS)</th>
<th>Error</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/5</td>
<td>0.05/410/0.15</td>
<td>66.1</td>
<td>69.0</td>
<td>4.5%</td>
<td>142.3</td>
<td>140.6</td>
<td>1.2%</td>
<td>3</td>
</tr>
<tr>
<td>40/20</td>
<td>0.1/290/0.25</td>
<td>39.3</td>
<td>41.0</td>
<td>4.3%</td>
<td>95.3</td>
<td>97.4</td>
<td>2.2%</td>
<td>3</td>
</tr>
<tr>
<td>40/20</td>
<td>0.5/810/0.7</td>
<td>74.0</td>
<td>76.4</td>
<td>3.2%</td>
<td>136.2</td>
<td>134.5</td>
<td>1.5%</td>
<td>2</td>
</tr>
<tr>
<td>30/15</td>
<td>0.4/1000/0.8</td>
<td>76.3</td>
<td>79.4</td>
<td>4.1%</td>
<td>142.5</td>
<td>138.5</td>
<td>2.8%</td>
<td>1</td>
</tr>
<tr>
<td>100/50</td>
<td>0.9/300/1.4</td>
<td>62.7</td>
<td>65.1</td>
<td>3.8%</td>
<td>121.0</td>
<td>123.1</td>
<td>1.7%</td>
<td>2</td>
</tr>
<tr>
<td>Avg. Error</td>
<td></td>
<td></td>
<td></td>
<td>4.0%</td>
<td></td>
<td></td>
<td>1.8%</td>
<td></td>
</tr>
</tbody>
</table>

#### 3-input NAND (Wp/Wn)

<table>
<thead>
<tr>
<th>3-input NAND (Wp/Wn)</th>
<th>C1 (pF)/R1 (W)/C2 (pF)</th>
<th>HSPICE 50% delay (pS)</th>
<th>Estimated 50% delay (pS)</th>
<th>Error</th>
<th>HSPICE 80% delay (pS)</th>
<th>Estimated 80% delay (pS)</th>
<th>Error</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>20/60</td>
<td>0.4/1000/0.8</td>
<td>34.7p</td>
<td>36.4p</td>
<td>4.9%</td>
<td>42.1p</td>
<td>43.2p</td>
<td>2.6%</td>
<td>2</td>
</tr>
<tr>
<td>40/120</td>
<td>0.5/510/1.2</td>
<td>26.4p</td>
<td>27.1p</td>
<td>2.7%</td>
<td>78.1p</td>
<td>79.5p</td>
<td>1.8%</td>
<td>2</td>
</tr>
<tr>
<td>Avg. Error</td>
<td></td>
<td></td>
<td></td>
<td>3.6%</td>
<td></td>
<td></td>
<td>2.2%</td>
<td></td>
</tr>
</tbody>
</table>

The topmost transistor in the stack is switching

<table>
<thead>
<tr>
<th>3-input NAND (Wp/Wn)</th>
<th>C1 (pF)/R1 (W)/C2 (pF)</th>
<th>HSPICE 50% delay (pS)</th>
<th>Estimated 50% delay (pS)</th>
<th>Error</th>
<th>HSPICE 80% delay (pS)</th>
<th>Estimated 80% delay (pS)</th>
<th>Error</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>20/60</td>
<td>0.4/1000/0.8</td>
<td>64.7p</td>
<td>67p</td>
<td>3.6%</td>
<td>64.4p</td>
<td>64.8p</td>
<td>0.6%</td>
<td>3</td>
</tr>
<tr>
<td>40/120</td>
<td>0.5/510/1.2</td>
<td>54.1p</td>
<td>55.5p</td>
<td>2.6%</td>
<td>83.5p</td>
<td>84.5p</td>
<td>1.2%</td>
<td>2</td>
</tr>
<tr>
<td>Avg. Error</td>
<td></td>
<td></td>
<td></td>
<td>3.1%</td>
<td></td>
<td></td>
<td>0.9%</td>
<td></td>
</tr>
</tbody>
</table>

All three transistors in the stack are switching.