Lecture 12: Logical Effort

PMOS/NMOS Ratio Effects

\[ \beta = \frac{W/L_p}{W/L_n} \]

\( \beta \) of 2.4 (= 31 k\( \Omega \)/13 k\( \Omega \)) gives symmetrical response

\( \beta \) of 1.6 to 1.9 gives optimal performance
Example of Inverter Chain Sizing

- $C_L/C_{g,1}$ has to be evenly distributed over $N = 3$ inverters
  
  $$F = \frac{C_L}{C_{g,1}} = 8/1$$

  $f = \phantom{=}$
Heads up

☐ This lecture
  ● Logical Effort
    - Reading assignment – textbook pp251-257, and handout

☐ Next lecture
  ● Designing energy efficient logic
    - Reading assignment – Rabaey, et al, 5.5 & 6.2.1
History

- First proposed by Ivan Sutherland and Bob Sproull in 1991
  - Both authors are vice president and fellow at Sun Microsystems

- Gain-based synthesis based on Logical effort
  - Implemented in IBM’s logic synthesis tool BooleDozer
  - Also adopted by Magma’s logic synthesis tool
Inverter Delay

- Divide capacitive load, \( C_L \), into
  - \( C_{\text{int}} \): intrinsic - diffusion and Miller effect (\( C_g \))
  - \( C_{\text{ext}} \): extrinsic - wiring and fanout

\[
 t_p = 0.69 \, R_{\text{eq}} \, C_{\text{int}} \left(1 + \frac{C_{\text{ext}}}{C_{\text{int}}}ight) = t_{p0} \left(1 + \frac{C_{\text{ext}}}{C_{\text{int}}}ight)
\]
\[
= 0.69 (R_{\text{eq}} C_{\text{int}} + R_{\text{eq}} C_{\text{ext}})
\]

- where \( t_{p0} = 0.69 \, R_{\text{eq}} \, C_{\text{int}} \) is the intrinsic (unloaded) delay of the gate
Logical Effort Delay Model

- Delay of logic gate has two components
  - \( d = f + p \)
  - \( f \): effort delay
  - \( p \): parasitic delay

- Effort delay \( fg \) has two components:
  - \( f = gh \)
  - \( g \): logical effort
  - \( h \): electrical effort = \( \frac{C_{out}}{C_{in}} \) (the ratio of output capacitance to input capacitance)
Gate Delay Components

- Split delay of logic gate into three components

  \[ \text{Delay} = \text{Logical Effort} \times \text{Electrical Effort} + \text{Parasitic Delay} \]

- Logical Effort
  - Complexity of logic function (Invert, NAND, NOR, etc)
  - Define inverter has logical effort = 1
  - Depends only on topology not transistor sizing

- Electrical Effort
  - Ratio of output capacitance to input capacitance \( \frac{C_{\text{out}}}{C_{\text{in}}} \)

- Parasitic Delay
  - Intrinsic delay
  - Independent of transistor sizes and output load
Computing Logical Effort

- g represents the fact that, for a given load, complex gates have to work harder than an inverter to produce a similar (speed) response.
  - The logical effort of a gate tells how much worse it is at producing an output current than an inverter (how much more input capacitance a gate presents to deliver the same output current).

- Logical effort is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.

- Defined to be 1 for an inverter.

Inverter: $C_{in} = 3$
$g = 1$ (def)
Computing Logical Effort

\[ d = g \cdot h + p \]

Inverter:
\( V_{in} = 3 \)
\( g = 1 \) (def)

NAND2:
\( V_{in} = 4 \)
\( g = 4/3 \)

NOR2:
\( V_{in} = 5 \)
\( g = 5/3 \)
Logic Gate Delay

Electrical effort: \( h = \frac{C_{out}}{C_{in}} \)

\[
g = \frac{4}{3}
\]
\[
p = \frac{2}{3} \cdot h + 2
\]
\[
d = 5 \cdot h + p = h + 1
\]
## Logic Gate Delay

### Table 1: Logical effort of static CMOS gates

<table>
<thead>
<tr>
<th>Gate type</th>
<th>Number of inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>inverter</td>
<td>1</td>
</tr>
<tr>
<td>NAND</td>
<td>4/3</td>
</tr>
<tr>
<td>NOR</td>
<td>5/3</td>
</tr>
<tr>
<td>multiplexer</td>
<td>2</td>
</tr>
<tr>
<td>XOR, XNOR</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table 2: Parasitic delay of static CMOS gates

<table>
<thead>
<tr>
<th>Gate type</th>
<th>Parasitic delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>inverter</td>
<td>( p_{inv} )</td>
</tr>
<tr>
<td>( n )-input NAND</td>
<td>( np_{inv} )</td>
</tr>
<tr>
<td>( n )-input NOR</td>
<td>( np_{inv} )</td>
</tr>
<tr>
<td>( n )-way multiplexer</td>
<td>( 2np_{inv} )</td>
</tr>
<tr>
<td>2-input XOR, XNOR</td>
<td>( 4np_{inv} )</td>
</tr>
</tbody>
</table>

\( p_{inv} \approx 1 \)

Parasitic delays depend on diffusion capacitance.
Example

- Estimate the delay of an inverter driving 4 identical inverters: (FO4)

\[ d = g \cdot h + p \]

\[ d = 1 \cdot 4 + 1 = 5 \]
Example

\[ s = \frac{4}{3} \quad h = 3 \quad p = 2 \]
\[ d = \frac{4}{3} \times 3 + 2 = 6 \]
Path Delay of Complex Logic Gate Network

- Total path delay through a combinational logic block
  \[ t_p = \sum d_j = \sum p_j + \sum h_j g_j \]
- The minimum delay through the path determines that each stage should bear the same gate effort
  \[ h_1 g_1 = h_2 g_2 = \ldots = h_N g_N \]

\[ g_1 = 1 \quad g_2 = \frac{5}{3} \quad g_3 = \frac{4}{3} \quad g_4 = 1 \]

\[ h_1 = \frac{x}{10} \quad h_2 = \frac{y}{x} \quad h_3 = \frac{z}{y} \quad h_4 = \frac{20}{z} \]
Application of Logical Effort

- Alternative logic structures, which is the fastest?

F = ABCDEFGH
Application of Logical Effort

Alternative logic structures, which is the fastest?

\[ F = ABCDEFGH \]

\[ g_1 = \frac{10}{3} \quad g_2 = 1 \]

\[ g_1 = \frac{4}{3} \quad g_2 = \frac{5}{3} \quad g_3 = \frac{4}{3} \quad g_4 = 1 \]

\[ g_1 = \frac{6}{3} \quad g_2 = \frac{5}{3} \]
Isolating fan-in from fan-out using buffer insertion
Questions

- $d = gh + p$
  \[ d = \frac{d_{abs}}{\tau} \]

- How to derive the model from Elmore delay model?

- Why logical effort $g$ is independent of transistor sizing?

- How to calculate parasitic delay $p$? Why it is independent of transistor sizing?

- How to calculate single delay parameter: $\tau$

- What if the ratio of p-type to n-type transistor widths changes?
From Elmore model to Logical Effort Model

\[ \text{Elmore Delay} = R(C_p + C_{out}) \]
\[ = R\cdot C_{out} + R\cdot C_p \]
\[ = R\cdot C_{in}\cdot (C_{out}/C_{in}) + R\cdot C_p \]
Parasitic Delay

- Main cause is drain capacitances
- These scale with transistor width so it is independent of transistor sizes
- For inverter:

Parasitic Delay $\approx 1.0 \tau$
How to calculate single delay parameter: $\tau$

- Characterize process speed with single delay parameter: $\tau$

\[ \tau \approx 15 \text{ ps for } 0.18\text{um} \quad \approx 20 \text{ ps for } 0.25 \text{um} \]

- How to estimate it for a new process? (such as 0.13 or 0.09 um)
For each stage:

Delay = Logical Effort $\times$ Electrical Effort + Parasitic Delay

= 1.0 \((definition)\) $\times$ 1.0 \((in = out)\) + 1.0 \((drain caps)\)

= 2.0 units
Multistage Logic Network

\[ G = \prod g_i \quad (g_i = \text{L.E. stage } i) \]
\[ H = \frac{C_{\text{out}}}{C_{\text{in}}} \quad (h_i = \text{E.E. stage } i) \]
\[ P = \sum p_i \quad (p_i = \text{P.D. stage } i) \]
\[ F = \prod f_i = \prod g_i h_i \]

\[ D = \sum g_i h_i + P \]
\[ D_{\text{min}} = \sqrt[n]{F} + P \]
Paths that Branch

- Consider paths that branch:

\[ G = \]
\[ H = \]
\[ GH = \]
\[ h_1 = \]
\[ h_2 = \]
\[ F = GH? \]
No! Consider paths that branch:

\[
\begin{align*}
G &= 1 \\
H &= 90 / 5 = 18 \\
GH &= 18 \\
h_1 &= (15 + 15) / 5 = 6 \\
h_2 &= 90 / 15 = 6 \\
F &= g_1g_2h_1h_2 = 36 = 2GH
\end{align*}
\]
Add Branching Effort

Branching effort:

\[ b = \frac{C_{on\text{-}path} + C_{off\text{-}path}}{C_{on\text{-}path}} \]
Path electrical effort: \( H = \frac{C_{out}}{C_{in}} \)

Path logical effort: \( G = g_1 g_2 \ldots g_N \)

Branching effort: \( B = b_1 b_2 \ldots b_N \)

Path effort: \( F = GBH \)

Path delay \( D = F + B = GBH + P \)
Optimal Number of Stages

- Minimum delay when:
  
  \[
  \text{stage effort} = \text{logical effort} \times \text{electrical effort} = 3.4 - 3.8 \sim 4
  \]

- Fan-out-of-four (FO4) is convenient design size (~5τ)

**FO4 delay:** Delay of inverter driving four copies of itself
Method of Logical Effort

- Compute the path effort: \( F = GBH \)
- Find the best number of stages \( N \sim \log_4 F \)
- Compute the stage effort \( f = F^{1/N} \)
- Sketch the path with this number of stages
- Work either from either end, find sizes:
  \( C_{in} = C_{out} \cdot g/f \)
Example of Inverter (Buffer) Staging

<table>
<thead>
<tr>
<th>N</th>
<th>f</th>
<th>t_p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>2.8</td>
<td>15.3</td>
</tr>
</tbody>
</table>
### Table 4: Key Definitions of Logical Effort

<table>
<thead>
<tr>
<th>Term</th>
<th>Stage expression</th>
<th>Path expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical effort</td>
<td>$g$ (see Table 1)</td>
<td>$G = \prod g_i$</td>
</tr>
<tr>
<td>Electrical effort</td>
<td>$h = \frac{C_{out}}{C_{in}}$</td>
<td>$H = \frac{C_{out\ (path)}}{C_{in\ (path)}}$</td>
</tr>
<tr>
<td>Branching effort</td>
<td>n/a</td>
<td>$B = \prod b_i$</td>
</tr>
<tr>
<td>Effort</td>
<td>$f = gh$</td>
<td>$F = GBH$</td>
</tr>
<tr>
<td>Effort delay</td>
<td>$f$</td>
<td>$D_F = \sum f_i$</td>
</tr>
<tr>
<td>Number of stages</td>
<td>1</td>
<td>$N$</td>
</tr>
<tr>
<td>Parasitic delay</td>
<td>$\rho$ (see Table 2)</td>
<td>$P = \sum \rho_i$</td>
</tr>
<tr>
<td>Delay</td>
<td>$d = f + \rho$</td>
<td>$D = D_F + P$</td>
</tr>
</tbody>
</table>
Next Lecture and Reminders

- Next lecture
  - Designing energy efficient logic
    - Reading assignment – Rabaey, et al, 5.5 & 6.2.1