Delay Minimization
<table>
<thead>
<tr>
<th></th>
<th>INV X1</th>
<th>NAND2 X1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cin</td>
<td>3.6fF</td>
<td>4.2fF</td>
</tr>
<tr>
<td>Intrinsic rise</td>
<td>25.3ps</td>
<td>31.3ps</td>
</tr>
<tr>
<td>Intrinsic fall</td>
<td>14.6ps</td>
<td>19.5ps</td>
</tr>
<tr>
<td>Intrinsic delay (avg)</td>
<td>20ps</td>
<td>25.4ps</td>
</tr>
<tr>
<td>Rise delay slope</td>
<td>4.52ps/fF</td>
<td>4.53ps/fF</td>
</tr>
<tr>
<td>Fall delay slope</td>
<td>2.37ps/fF</td>
<td>2.84ps/fF</td>
</tr>
<tr>
<td>Delay slope (avg)</td>
<td>3.45ps/fF</td>
<td>3.68ps/fF</td>
</tr>
<tr>
<td>τ</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>p</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>g</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>Delay eqn</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Logical Effort Calculation

\[ t^{inv} = \tau(p_{inv} + Cout/3.6\text{ff}) \]
\[ t^{n_2} = \tau(p_{n_2} + g_{N_2} Cout/4.2\text{ff}) \]

Inverter's average slope gives
\[ \tau \times 1\text{ff}/3.6\text{ff} = 3.45\text{pS} \]
\[ \tau = 12.42\text{pS} \]

Inverter's average intrinsic delay gives
\[ \tau \times p_{inv} = 20\text{pS} \]
\[ p_{inv} = 1.61 \]

Nand2's average slope gives
\[ \tau \times g_{N_2} \times 1\text{ff}/4.2\text{fF} = 3.68 \]
\[ g_{n_2} = 1.25 \]

Logical effort of NAND2 is less than 4/3 (1.33), why?
Current is independent of length for completely velocity saturated

\[ I_{dsat} = W C_{ox} v_{sat} (V_{GS} - V_T) \]

So a series stack of 'n' is equivalent to longer channel Length of 'nL' has same pull down current as minimum Channel length.

So logical effort is 1.0 for this extreme case.

Actual logical efforts are somewhere in between.
Sizing to minimize delay

\[
\begin{align*}
  g_1 &= 1 \\
  h_1 &= x/10 \\
  g_2 &= 5/3 \\
  h_2 &= y/x \\
  g_3 &= 4/3 \\
  h_3 &= z/y \\
  g_4 &= 1 \\
  h_4 &= 20/z
\end{align*}
\]

Capacitance unit = size of unit transistor
Find scale of each gate: Relative transistor sizes is fixed and hence logical effort is fixed.
Sizing to minimize delay

\[ g_1 = 1 \quad h_1 = x/10 \]
\[ g_2 = 5/3 \quad h_2 = y/x \]
\[ g_3 = 4/3 \quad h_3 = z/y \]
\[ g_4 = 1 \quad h_4 = 20/z \]

Note:

\[
\min_{h_1, \ldots, h_4} D = \sum_{i=1}^{4} D_i
\]

\[
D = \sum_{i=1}^{4} p_i + f_i
\]

where: \( f_i = g_i \cdot h_i \)

\[
\prod_{i=1}^{4} f_i = \prod_{i=1}^{4} g \cdot h = \prod_{i=1}^{4} g \prod_{i=1}^{4} h = G \cdot H
\]

\[
G = \prod_{i=1}^{4} g
\]

\[
H = \prod_{i=1}^{4} h = \frac{C_2}{C_{in}} \frac{C_3}{C_2} \frac{C_4}{C_3} \frac{C_{out}}{C_4} = \frac{C_{out}}{C_{in}}
\]

Capacitance unit = size of unit

Transistor

Independent of hi
Multistage Logic Networks

Note:

Capacitance unit
= size of unit
Transistor

Problem: \( \min_{f_1 \ldots f_4} D = \sum_{i=1}^{4} D_i \)

\[
\text{minimize } D = \sum_{i=1}^{4} f_i + p_i
\]

With constraint \( \prod_{i=1}^{4} f_i = G H \)

Solution: \( f_{\text{opt}} = f_1 = \ldots = f_4 \)

\[
f_{\text{opt}} = (GH)^{1/4}
\]

\[
g_1 \frac{C_2}{C_{\text{in}}} = g_2 \frac{C_3}{C_2} = g_3 \frac{C_4}{C_3} = g_4 \frac{C_{\text{out}}}{C_4} = f_{\text{opt}}
\]
Multistage Logic Networks

\[ g_1 = 1 \]
\[ h_1 = \frac{x}{10} \]
\[ g_2 = \frac{5}{3} \]
\[ h_2 = \frac{y}{x} \]
\[ g_3 = \frac{4}{3} \]
\[ h_3 = \frac{z}{y} \]
\[ g_4 = 1 \]
\[ h_4 = \frac{20}{z} \]

Problem: \( \min_{f_1, \ldots, f_4} D = \sum_{i=1}^{4} D_i \)

\[ D = \sum_{i=1}^{4} f_i + p_i \]

\[ \prod_{i=1}^{4} f_i = GH = 1 \frac{5}{3} \frac{4}{3} \frac{20}{10} = \frac{40}{9} \]

\[ f_{opt} = f_1 = \ldots = f_4 = \left(\frac{40}{9}\right)^{1/4} = 1.45 \]

\[ x = 1 \frac{y}{10} = \frac{5}{3} \]
\[ \frac{z}{y} = \frac{4}{3} \]
\[ 1 \frac{20}{z} = 1.45 \]

\[ z = 13.8 \]
\[ y = 12.7 \]
\[ x = 14.6 \]

Note:

Capacitance unit
= size of unit
Transistor
Transistor sizing within the gate

(a) 
\[ C_{in} = 3 \]
\[ g = \frac{3}{3} \]

(b) 
\[ C_{in} = 4 \]
\[ g = \frac{4}{3} \]

(c) 
\[ C_{in} = 5 \]
\[ g = \frac{5}{3} \]

\[ z = 13.8 \]
\[ y = 12.7 \]
\[ x = 14.6 \]

\[ Z_{pmos} = \frac{2}{3} \times 13.8 = 9.2 \]
\[ Z_{nmos} = \frac{1}{3} \times 13.8 = 4.6 \]

\[ y_{pmos} = \frac{2}{4} \times 12.7 = 6.35 = y_{nmos} \]

\[ x_{pmos} = \frac{4}{5} \times 14.6 = 11.7 \]
\[ x_{nmos} = \frac{1}{5} \times 14.6 = 2.9 \]
Treat a path as a large compound gate

- Logical effort generalizes to multistage paths

- **Path Logical Effort**
  \[ G = \prod g_i \]

- **Path Electrical Effort**
  \[ H = \frac{C_{out\text{-}path}}{C_{in\text{-}path}} \]

- **Path Effort**
  \[ F = \sum f_i = \sum g_i h_i \]

- However we cannot write \( F = GH \)
Paths that Branch

\[ D_1 = \frac{15 + 15}{5} + 1 \]

\[ D_2 = \frac{90}{15} + 1 \]

\[ h_1 = \frac{C_{i+1} + C_{\text{offpath}}}{C_i} \]

\[ h_2 = \left( 1 + \frac{C_{i+1}}{C_{i+1}} \right) \frac{C_{i+1}}{C_i} b_i \]

\[ \prod h_i = \frac{C_2}{C_{\text{in}}} b_1 * \frac{C_3}{C_2} b_2 * \ldots * \frac{C_{\text{out}}}{C_n} b_n = \prod b_i * \frac{C_{\text{out}}}{C_{\text{in}}} = BH \]
Optimum stage effort

\[
\min D = \sum_{1}^{N} (f_i + p_i)
\]

\[f_i = g_i \ast h_i\]

\[h_i = \frac{C_{i+1}}{C_i}\]

\[\prod_{1}^{N} f_i = GBH\]

optimum at \(f_1 = f_2 = \ldots = f_N = f_{opt}\)

\[f_{opt}^{N} = GBH\]

\[D = Nf_{opt} + \sum_{1}^{N} p_i\]

\[f_{opt} = (GBH)^{1/N}\]

\[C_{in} = C_{out} \times \frac{g_i}{f_{opt}}\]

Minimum delay occurs when all stage efforts in the path are equal
Finding the gate sizes for the optimized path

- Work backwards, apply capacitance transformation to find input capacitance of each gate given load it drives.
- Check work by verifying input cap spec is met.
Find $x$ and $y$ to minimize delay
Example: 3-stage path

Logical Effort \[ G = \frac{4}{3} \times \frac{5}{3} \times \frac{5}{3} = \frac{100}{27} \]

Electrical Effort \[ H = \frac{45}{8} \]

Branching Effort \[ B = 3 \times 2 = 6 \]

Path Effort \[ F = GBH = 125 \]

Best Stage Effort \[ f_{opt} = F^{(1/3)} = 5 \]

Parasitic Delay \[ P = 2 + 3 + 2 = 7 \]

Delay \[ D = 3 \times 5 + 7 = 22 = 4.4 \text{ FO4} \]
Example: 3-stage path

- Work backward for sizes
  
  \[ y = \]
  
  \[ x = \]
Example: 3-stage path

- Work backward for sizes

\[ y = 45 \times \left(\frac{5}{3}\right) / 5 = 15 \]
\[ x = (15 \times 2) \times \left(\frac{5}{3}\right) / 5 = 10 \]
How many stages should a path use?

- Minimizing number of stages is not always fastest

Example: drive 64-bit datapath with unit inverter

\[ D = NF^{1/N} + P \]

\[ = N(64)^{1/N} + N \]
Optimum Stage effort with freedom to choose number of stages

Logic Block: \( n_1 \) Stages Path Effort \( F \)

\[
D(N) = NF^N + \sum_{i=1}^{n_1} p_i + (N - n_1) p_{\text{inv}}
\]

\[
\frac{dD}{dN} = F^N + NF^N \ln(F) \cdot \left( -\frac{1}{N^2} \right) + p_{\text{inv}}
\]

\[ f = F^N = \text{stage effort} \]

\[
\frac{dD}{dN} = f \cdot (1 - \ln(f)) + p_{\text{inv}}
\]

For minima, \( \frac{dD}{dN} = 0 \): hence,

\[ f_{\text{opt}} \cdot (1 - \ln(f_{\text{opt}})) + p_{\text{inv}} = 0 \]

\[ f_{\text{opt}} = e \text{ if } p_{\text{inv}} = 0 \]

\[ f_{\text{opt}} = 3.5 \text{ assuming } p_{\text{inv}} = 1 \]

\[ N_{\text{opt}} = \log_f(F) = \frac{\ln(F)}{\ln(f)} \]
How sensitive is delay to using exactly the best number of stages?

- $2.4 < f < 6$ gives delay within 15% of optimal
  - $f = 4$ is convenient