Lecture 8: Logic Effort and Combinational Circuit Design

Deming Chen

Slides based on the initial set from David Harris
Outline

- Logical Effort
- Delay in a Logic Gate
- Multistage Logic Networks
- Bubble Pushing
- Compound Gates
- Logical Effort Example
- More advanced topics

Readings: 4.4.1-4.4.4; 4.5.1; 9.1-9.2.2
Delay in a Logic Gate

- Express delays in process-independent unit
- Delay has two components: \( d = f + p \)
- \( f \): effort delay = \( gh \) (a.k.a. stage effort)
  - Again has two components
- \( g \): logical effort
  - Measures relative ability of gate to deliver current
  - \( g \equiv 1 \) for inverter
- \( h \): electrical effort = \( C_{out} / C_{in} \)
  - Ratio of output to input capacitance
  - Sometimes called fanout
- \( p \): parasitic delay
  - Represents delay of gate driving no load
  - Set by internal parasitic capacitance

\[
\tau = \frac{d_{abs}}{\tau} \\
\approx 3 \text{ ps in 65 nm process} \\
60 \text{ ps in 0.6 } \mu\text{m process}
\]
Computing Logical Effort

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

- Measure from delay vs. fanout plots
- Or estimate by counting transistor widths

<table>
<thead>
<tr>
<th>Circuit</th>
<th>$C_{in}$</th>
<th>$g$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Circuit 1" /></td>
<td>3</td>
<td>3/3</td>
</tr>
<tr>
<td><img src="image2.png" alt="Circuit 2" /></td>
<td>4</td>
<td>4/3</td>
</tr>
<tr>
<td><img src="image3.png" alt="Circuit 3" /></td>
<td>5</td>
<td>5/3</td>
</tr>
</tbody>
</table>
$d = f + p$
$= gh + p$

What about NOR2?

Delay Plots

$g = 4/3$
$p = 2$
$d = (4/3)h + 2$

Effort Delay: $f$

Parasitic Delay: $p$

$h = C_{out} / C_{in}$

Normalized Delay: $d$
## Catalog of Gates

- Logical effort of common 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>Tristate / mux</td>
<td>2</td>
</tr>
<tr>
<td>XOR, XNOR</td>
<td>4, 4</td>
</tr>
</tbody>
</table>
Catalog of Gates

- Parasitic delay of common gates
  - In multiples of $p_{inv} \approx 1$

<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>2</td>
</tr>
<tr>
<td>NOR</td>
<td>2</td>
</tr>
<tr>
<td>Tristate / mux</td>
<td>2</td>
</tr>
<tr>
<td>XOR, XNOR</td>
<td>4</td>
</tr>
</tbody>
</table>

Parasitic delay of common gates

$\theta$ in multiples of $p_{inv} \approx 1$
Example: Ring Oscillator

- Estimate the frequency of an N-stage ring oscillator

Logical Effort: \( g = 1 \)
Electrical Effort: \( h = 1 \)
Parasitic Delay: \( p = 1 \)
Stage Delay: \( d = 2 \)
Frequency: \( f_{\text{osc}} = \frac{1}{2N \cdot d} = \frac{1}{4N} \)

31 stage ring oscillator in 0.6 μm process has frequency of ~ 200 MHz
Example: FO4 Inverter

Estimate the delay of a fanout-of-4 (FO4) inverter

Logical Effort: \( g = 1 \)

Electrical Effort: \( h = 4 \)

Parasitic Delay: \( p = 1 \)

Stage Delay: \( d = 5 \)

The FO4 delay is about 300 ps in 0.6 µm process

15 ps in a 65 nm process
Multistage Logic Networks

- Logical effort generalizes to multistage networks

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

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

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

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

- Logical effort generalizes to multistage networks

- Path Logical Effort \( G = \prod g_i \)

- Path Electrical Effort \( H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}} \)

- Path Effort \( F = \prod f_i = \prod g_i h_i \)

- Can we write \( F = GH \)?
No! Consider paths that branch:

\[ G = 1 \]
\[ H = \frac{90}{5} = 18 \]
\[ GH = 18 \]
\[ h_1 = \frac{(15 + 15)}{5} = 6 \]
\[ h_2 = \frac{90}{15} = 6 \]
\[ F = g_1 g_2 h_1 h_2 = 36 = 2GH \]
Branching Effort

- Introduce *branching effort*
  - Accounts for branching between stages in path

\[ b = \frac{C_{\text{on path}} + C_{\text{off path}}}{C_{\text{on path}}} \]

\[ B = \prod b_i \]

Note: \[ \prod h_i = BH \]

- Now we compute the path effort
  - \( F = GBH \)
Multistage Delays

- Path Effort Delay
  \[ D_F = \sum f_i \]

- Path Parasitic Delay
  \[ P = \sum p_i \]

- Path Delay
  \[ D = \sum d_i = D_F + P \]
Designing Fast Circuits

\[ D = \sum d_i = D_F + P \]

- Delay is smallest when each stage bears same effort

\[ \hat{f} = g_i h_i = F^{\frac{1}{N}} \]

- Thus minimum delay of N stage path is

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

- This is a key result of logical effort
  - Find fastest possible delay
  - Doesn’t require calculating gate sizes
Gate Sizes

- How wide should the gates be for least delay?

\[ \hat{f} = gh = g \frac{C_{out}}{C_{in}} \]

\[ \Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}} \]

- Working backward, apply capacitance transformation to find input capacitance of each gate given load it drives.
- Check work by verifying input cap spec is met.
Example: 3-stage path

- Select gate sizes x and y for least delay from A to B
Example: 3-stage path

\[
\text{Logical Effort} \quad G = \left(\frac{4}{3}\right) \times \left(\frac{5}{3}\right) \times \left(\frac{5}{3}\right) = \frac{100}{27} \\
\text{Electrical Effort} \quad H = \frac{45}{8} \\
\text{Branching Effort} \quad B = 3 \times 2 = 6 \\
\text{Path Effort} \quad F = GBH = 125 \\
\text{Best Stage Effort} \quad \hat{f} = \sqrt[3]{F} = 5 \\
\text{Parasitic Delay} \quad P = 2 + 3 + 2 = 7 \\
\text{Delay} \quad D = 3 \times 5 + 7 = 22 = 4.4 \text{ FO4}
\]
Example: 3-stage path

- Work backward for sizes
  \[ y = 45 \times \left(\frac{5}{3}\right) / 5 = 15 \]
  \[ x = (15 \times 2) \times \left(\frac{5}{3}\right) / 5 = 10 \]
Work backward for sizes

\[ y = 45 \times \frac{5}{3} \div 5 = 15 \]
\[ x = (15 \times 2) \times \frac{5}{3} \div 5 = 10 \]
Combinational Design - Example 1

module mux(input  s, d0, d1,
            output y);

    assign y = s ? d1 : d0;
endmodule

1) Sketch a design using AND, OR, and NOT gates.

\begin{center}
\begin{tikzpicture}
    \node (D0) at (0,0) {D0};
    \node (S) at (0,-0.5) {S};
    \node (D1) at (1,0) {D1};
    \node (S_bar) at (1,-0.5) {
        \overline{S}
    };
    \draw (D0) -- (S);\draw (D0) -- (D1);
    \draw (S) -- (S_bar);
    \draw (S) -- (D1);
\end{tikzpicture}
\end{center}
Example 2

2) Sketch a design using NAND, NOR, and NOT gates. Assume \( \sim S \) is available.
Bubble Pushing

- Start with network of AND / OR gates
- Convert to NAND / NOR + inverters
- Push bubbles around to simplify logic
  - Remember DeMorgan’s Law

(a)  

(b)  

(c)  

(d)
Example 3

3) Sketch a design using one compound gate and one NOT gate. Assume \( \sim S \) is available.
Compound Gates

Logical Effort of compound gates

- Unit inverter
  \[ Y = \overline{A} \]

- AOI21
  \[ Y = A \cdot B + C \]

- AOI22
  \[ Y = A \cdot B + C \cdot D \]

- Complex AOI
  \[ Y = A \cdot (B + C) + D \cdot E \]

\[ g_A = 6/3 \]
\[ g_B = 6/3 \]
\[ g_C = 5/3 \]
\[ g_D = 6/3 \]
\[ g_E = 8/3 \]
\[ p = 12/3 \]
\[ p = 16/3 \]
Example 4

- The multiplexer has a maximum input capacitance of 16 units on each input. It must drive a load of 160 units. Estimate the delay of the two designs.

\[ H = \frac{160}{16} = 10 \quad B = 1 \quad N = 2 \]

\[
\begin{align*}
P &= 2 + 2 = 4 \\
G &= (4/3) \cdot (4/3) = 16/9 \\
F &= GBH = 160/9 \\
\hat{f} &= \sqrt[3]{F} = 4.2 \\
D &= N\hat{f} + P = 12.4\tau \\
\end{align*}
\]

\[
\begin{align*}
P &= 4 + 1 = 5 \\
G &= (6/3) \cdot (1) = 2 \\
F &= GBH = 20 \\
\hat{f} &= \sqrt[3]{F} = 4.5 \\
D &= N\hat{f} + P = 14\tau \\
\end{align*}
\]
Example 5

- Annotate your designs with transistor sizes that achieve this delay.

\[ 160 \times \left( \frac{4}{3} \right) \div 4.2 = 50 \]

\[ 16 \times \frac{1}{4.5} = 36 \]
Our parasitic delay model was too simple
- Calculate parasitic delay for $Y$ falling
  - If $A$ arrives latest? $2\tau$
  - If $B$ arrives latest? $2.33\tau$
Inner & Outer Inputs

- *Inner* input is closest to output (A)
- *Outer* input is closest to rail (B)

- If input arrival time is known
  - Connect latest input to inner terminal
Asymmetric Gates

- Asymmetric gates favor one input over another.
- Ex: suppose input A of a NAND gate is most critical:
  - Use smaller transistor on A (less capacitance).
  - Boost size of noncritical input.
  - So total resistance is same.

- \( g_A = \frac{10}{9} \)
- \( g_B = 2 \)
- \( g_{\text{total}} = g_A + g_B = \frac{28}{9} \)
- Asymmetric gate approaches g = 1 on critical input.
- But total logical effort goes up.
Symmetric Gates

- Inputs can be made perfectly symmetric

```
A
2
1
B
1
1
1
2
1
Y
```
Skewed Gates

- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical nMOS transistor

![HI-skew inverter](image)

![Unskewed inverter (equal rise resistance)](image)

![Unskewed inverter (equal fall resistance)](image)

- Calculate logical effort by comparing to unskewed inverter with same effective resistance on that edge.
  - $g_u = \frac{2.5}{3} = \frac{5}{6}$
  - $g_d = \frac{2.5}{1.5} = \frac{5}{3}$
HI- and LO-Skew

- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.

- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)

- Logical effort is smaller for favored direction
- But larger for the other direction
Catalog of Skewed Gates

Inverter

unskewed

HI-skew

LO-skew

NAND2

NOR2

Combination Circuits

CMOS VLSI Design 4th Ed.
Asymmetric Skew

- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input

![Diagram showing asymmetric skew in a combinational circuit](attachment:diagram.png)
Best P/N Ratio

- We have selected P/N ratio for unit rise and fall resistance ($\mu = 2-3$ for an inverter).
- Alternative: choose ratio for least average delay
- Ex: inverter
  - Delay driving identical inverter
  - $t_{pdf} = (P+1)$
  - $t_{pdr} = (P+1)(\mu/P)$
  - $t_{pd} = (P+1)(1+\mu/P)/2 = (P + 1 + \mu + \mu/P)/2$
  - $dt_{pd} / dP = (1 - \mu/P^2)/2 = 0$
  - Least delay for $P = \sqrt{\mu}$
P/N Ratios

- In general, best P/N ratio is sqrt of equal delay ratio.
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power

Inverter

- fastest P/N ratio
- $g_u = 1.15$
- $g_d = 0.81$
- $g_{avg} = 0.98$

NAND2

- $g_u = 4/3$
- $g_d = 4/3$
- $g_{avg} = 4/3$

NOR2

- $g_u = 2$
- $g_d = 1$
- $g_{avg} = 3/2$
Observations

- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages
Summary

- Logic effort is an effective way for hands-on delay estimations for simple gates and paths
  - In reality, people use timing analysis and logic synthesis for delay estimation and optimization
- Combinational circuit design is the foundation for modern computer design
  - Lead to sequential design in the next lecture

- Next lecture
  - Sequential circuit design
  - Readings: 10.1-10.3.4