Logical Effort

Sizing Transistors for Speed

Would be nice to have a “back of the envelope” method for sizing gates for speed

Logical Effort

- Book by Sutherland, Sproull, Harris
- Chapter 1 is on our web page
- Also Chapter 4 in our textbook

Estimating Delays
Gate Delay Model

- First, normalize a model of delay to dimensionless units to isolate fabrication effects
  - \( d_{\text{abs}} = d\tau \)
    - \( \tau \) is the delay of a minimum inverter driving another minimum inverter with no parasitics
    - In a 0.6\( \mu \) process, this is approx 40ps
    - Now we can think about delay in terms of \( d \) and scale it to whatever process we’re using

Gate Delay

- Delay of a gate \( d \) has two components
  - A fixed part called \textit{parasitic delay} \( p \)
  - A part proportional to the load on the output called the \textit{effort delay} or \textit{stage effort} \( f \)
  - \textit{Total delay} is measured in units of \( \tau \), and is sum of these delays
  - \( d = f + p \)
Effort Delay

- The effort delay (due to load) can be further broken down into two terms: \( f = g \times h \)

  - \( g \) = logical effort which captures properties of the gate’s structure
  - \( h \) = electrical effort which captures properties of load and transistor sizes
    - \( h = \frac{C_{out}}{C_{in}} \)
    - \( C_{out} \) is capacitance that loads the output
    - \( C_{in} \) is capacitance presented at the input
  - So, \( d = gh + p \)

Logical Effort

- Logical effort normalizes the output drive capability of a gate to match a unit inverter
  - How much more input capacitance does a gate need to present to offer the same drive as an inverter?

\[ g = 1 \]
\[ g = \frac{4}{3} \]
\[ g = \frac{5}{3} \]
**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

![Logic Gates Diagram]

- **Logical Effort of Other Gates**

  Logical effort of common gates assuming that P/N size ratio is 2

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inverter</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NAND</strong></td>
<td>4/3</td>
<td>5/3</td>
<td>6/3</td>
<td>7/3</td>
<td>(n+2)/3</td>
<td></td>
</tr>
<tr>
<td><strong>NOR</strong></td>
<td>5/3</td>
<td>7/3</td>
<td>9/3</td>
<td>11/3</td>
<td>(2n+1)/3</td>
<td></td>
</tr>
<tr>
<td><strong>MUX</strong></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>XOR</strong></td>
<td>4</td>
<td>12</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Electrical Effort

- Value of logical effort $g$ is independent of transistor size
  - It's related to the ratios and the topology

- Electrical effort $h$ captures the drive capability of the transistors via sizing
  - Electrical effort $h = \frac{C_{out}}{C_{in}}$
  - Note that as transistor sizes for a gate increase, $h$ decreases because $C_{in}$ goes up

Parasitic Delay

- Parasitic delay $p$ is caused by the internal capacitance of the gate
  - It's constant and independent of transistor size
  - As you increase the transistor size, you also increase the cap of the gate/source/drain areas which keeps it constant
  - For our purposes, normalize $p_{inv}$ to 1
    - N-input NAND = $n*p_{inv}$
    - N-input NOR = $n*p_{inv}$
    - N-way mux = $2n*p_{inv}$
    - XOR = $4*p_{inv}$
Plots of Gate Delay

Remember, $\tau$ in our process $\sim 40$ps

\[ A_{\text{delay}} = g \times h + p = 1 \times (\text{CinB}/\text{CinA}) + 1 = 1 \times (4 \times \text{CinA}/\text{CinA}) + 1 = 4 + 1 = 5 \text{ time units} \]

\[ A_{\text{delay}} = (4/3)(\text{CinB}/\text{CinA}) + 2 \times 1 \]
\[ \text{CinB = 4} \times 3 = 12, \quad \text{CinA} = 4 \]
\[ A_{\text{delay}} = (4/3)(12/4) + 2 = 4 + 2 = 6 \text{ units} \]

Nand2 worse because of higher parasitic delay than inverter.
Note that $g \times h$ term was same for both because NAND2 sized to provide same current drive.

Delay Estimation
Delay Estimation

Remember, \( \tau \) in our process \( \sim 40 \text{ps} \)

\( \tau \) in 180nm \( \sim 12 \text{ps} \)

FO4 Inverter delay = 60ps

FO4 NAND delay = 72ps

Example: Ring Oscillator

- Estimate the frequency of an N-stage ring oscillator

Logical Effort: \( g = \)

Electrical Effort: \( h = \)

Parasitic Delay: \( p = \)

Stage Delay: \( d = \)

Period of osc =
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: $d = 2$ so $d_{\text{abs}} = 80\text{ps}$
Period: $2Nd_{\text{abs}} = 4.96\text{ns}$, Freq $\approx 200\text{MHz}$
For $N = 31$

Example: FO4 Inverter

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

Logical Effort: $g =$
Electrical Effort: $h =$
Parasitic Delay: $p =$
Stage Delay: $d =$
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: $d = gh + p = 5$

The FO4 delay is about
- 200 ps in 0.6 μm process
- 60 ps in a 180 nm process
- $f/3$ ns in an $f$ μm process

Delay Estimation

- If $C_{in} = x$, $C_{out} = 10x$, thus $h = 10$
- $g = 9/3 = 3$
- $d = gh + p = 3*10 + 4*1 = 34$ (1360 ps)
Multi Stage Delay

MultiStage Delay

- Recall rule of thumb that said to balance the delay at each stage along a critical path
- Concepts of logical effort and electrical effort can be generalized to multistage paths

\[
g_1 g_2 g_3 g_4
\]

Path logical effort = \( g_1 g_2 g_3 g_4 \)

In general, Path logic effort \( G = \Pi g(i) \)

Path electrical effort \( H = \frac{C_{out}}{C_{in_{first\_gate}}} \)

Must remember that electrical effort only is concerned with effect of logic network on input drivers and output load.

Off-Path Load

Off path load will divert electrical effort from the main path, must account for this. Define a branching effort \( b \) as:

\[
b = \frac{(\text{Con\_path} + \text{Coff\_path})}{\text{Con\_path}} / \frac{\text{C_{total}}}{\text{C_{useful}}}\]

The branching effort will modify the electrical effort needed at that stage. The branch effort \( B \) of the path is:

\[
B = \Pi b(i)
\]
Summary – multistage networks

- Logical effort generalizes to multistage networks
- **Path Logical Effort**
  \[ G = \prod g_i \]
- **Path Electrical Effort**
  \[ H = \frac{C_{out\text{-}path}}{C_{in\text{-}path}} \]
- **Path Effort**
  \[ F = \prod f_i = \prod g_i h_i \]
- Can we write \( F = GH? \)

Branching Effort

- Remember *branching effort*
  - Accounts for branching between stages in path
    \[ b = \frac{C_{on\text{-}path}}{C_{off\text{-}path}} \]
    \[ B = \prod b_i \]
    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

- Delay is smallest when each stage bears same effort
  \[ \hat{f} = g_i h_i = F_i^{1/N} \]
- Thus minimum delay of N stage path is
  \[ D = \sqrt{NF_i^{1/N}} + P \]
- This is a key result of logical effort
  - Find fastest possible delay
  - Doesn’t require calculating gate sizes
Minimizing Path Delay

The absolute delay will have the parasitic delays of each stage summed together.

However, can focus on just Path effort $F$ for minimization purposes since parasitic delays are constant.

For an $N$-stage network, the path delay is least when each stage in the path bears the same stage effort.

$$f\ (\text{min}) = g(i) \times h(i) = F^{1/N}$$

Minimum achievable path delay

$$D(\text{min}) = N \times F^{1/N} + P$$

Note that if $N=1$, then $d = f + p$, the original single gate equation.

Choosing Transistor Sizes

Remember that the stage effort $h(i)$ is related to transistor sizes.

$$f\ (\text{min}) = g(i) \times h(i) = F^{1/N}$$

So

$$h(i)\ \text{min} = F^{1/N} / g(i)$$

To size transistors, start at end of path, and compute:

$$C_{in}(i) = g_{i} \times C_{out}(i) / f(\text{min})$$

Once $C_{in}(i)$ is known, can distribute this among transistors of that stage.
Size the transistors of the nand2 gates for the three stages shown.

Path logic effort \( G = g_0 * g_1 * g_2 = 4/3 * 4/3 * 4/3 = 2.37 \)

Branching effort \( B = 1.0 \) (no off-path load)

Electrical effort \( H = \frac{C_{out}}{C_{in}} = \frac{C}{C} = 1.0 \)

Min delay achievable \( = 3^* (G^*B^*H)^{1/3} + 3 (2^*pinv) \)

\( \min D = N^*F^{1/N} + P \)

\( = 3*(1.3333) + 6 = 10 \)

---

**Example, continued**

The effort of each stage will be:

\( f_{\text{min}} = (G^*B^*H)^{1/3} = (2.37^*1.0^*1.0)^{1/3} = 1.33 = 4/3 \)

Cin of last gate should equal:

\( f(\text{min}) = g_i * b_i * h_i \)

\( C_{\text{in \ last \ gate \ (min)}} = g_i * C_{\text{out \ (i)}} / f(\text{min}) \)

\( = 4/3 * C / (4/3) = C \)

Cin of middle gate should equal:

\( C_{\text{in \ middle \ gate}} = g_i * C_{\text{in \ last \ gate}} / f(\text{min}) \)

\( = 4/3 * C / (4/3) = C \)

All gates have same input capacitance, distribute it among transistors.
Transistor Sizes for Example

Where gate capacitance of
2 \* W \* L  Mosfet =  C/2

Choose W accordingly.

Another Example, Larger Load

Let Load = 8C, what changes?

Size the transistors of the nand2 gates for the three stages shown.

Path logic effort = G = g0 \* g1 \* g2 = 4/3 \* 4/3 \* 4/3 = 2.37
Branching effort B = 1.0 (no off-path load)
Electrical effort H = Cout/Cin = 8C/C = 8.0
Min delay achievable = 3 * (G*B*H)\frac{1}{3} + 3 * (2*pinv)
= 3 * (2.37*1*8)\frac{1}{3} + 3 * (2*1.0) = 14.0
8C Load Example Cont.

The effort of each stage will be:

\[ f_{\text{min}} = (G \times B \times H)^{1/3} = (2.37 \times 1.0 \times 8)^{1/3} = 2.67 = \frac{8}{3} \]

Cin of last gate should equal:

\[ \text{Cin last gate (min)} = g_i \times \text{Cout (i)} / f(\text{min}) = \frac{4/3 \times 8C}{(8/3)} = 4C \]

Cin of middle gate should equal:

\[ \text{Cin middle gate} = g_i \times \text{Cin last gate} / f(\text{min}) = \frac{4/3 \times 4C}{(8/3)} = 2C \]

Note that each stage gets progressively larger, as is typical with a multi-stage path driving a large load.

Example 1.6 from Chap 1

Size path from A to B

\[ \text{Cin} = C \]

\[ \text{Cin} = y \]

\[ \text{Cin} = z \]

\[ \text{Cin} = 0 \]

Path logic effort \( G = g_0 \times g_1 \times g_2 = 4/3 \times 4/3 \times 4/3 = 2.37 \)

Branch effort, 1st stage = \((y+y)/y = 2.\)

Branch effort, 2nd stage = \((z+z+z)/z = 3\)

Path Branch effort \( B = 2 \times 3 = 6.\)

Path electrical effort \( H = \text{Cout}/\text{Cin} = 4.5C/C = 4.5 \)

Path stage effort = \( F = G \times B \times H = 2.37 \times 6 \times 4.5 = 64. \)

Min delay = \( N(F)^{1/N} + P = 3 \times (64)^{1/3} + 3 \times (2\text{p inv}) = 18.0 \text{ units} \)
Example 1.6 Continued

Stage effort of each stage should be:
\[ f(\text{min}) = (F)^{1/N} = (GBH)^{1/N} = (64)^{1/3} = 4 \]

Determine Cin of last stage:
\[ f(\text{min}) = g_i \cdot b_i \cdot h_i \]
\[ \text{Cin}(z) = g \cdot C_{\text{out}} / f(\text{min}) = 4/3 \cdot 4.5 \cdot C / 4 = 1.5 \, \text{C} \]

Determine Cin of middle stage:
\[ \text{Cin}(y) = g \cdot (3 \cdot \text{Cin}(z)) / f(\text{min}) = 4/3 \cdot (3 \cdot 1.5 \cdot C) / 4 = 1.5 \, \text{C} \]

Is first stage correct?
\[ \text{Cin}(A) = g \cdot (2 \cdot \text{Cin}(y)) / f(\text{min}) = 4/3 \cdot (2 \cdot 1.5 \cdot C) / 4 = C. \]

Yes, self-consistent.

Example: 3-stage path

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

Logical Effort \( G = \frac{4}{3} \times \frac{5}{3} \times \frac{5}{3} = \frac{100}{27} \)

Electrical Effort \( H = 45/8 \)

Branching Effort \( B = 3 \times 2 = 6 \)

Path Effort \( F = GBH = 125 \)

Best Stage Effort \( \hat{f} = \sqrt[3]{F} = 5 \)

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

\[
y = \frac{45 \times (5/3)}{5} = 15
\]
\[
x = \frac{(15 \times 2) \times (5/3)}{5} = 10
\]

\[
f(\text{min}) = g_i b_i C_{\text{out}} / f_{\text{min}} = C_{\text{in}}
\]
Example 1.7 from Chap 1

Note: Don’t care about parasitics for gate sizing, only if you want to know absolute delay…

\[ \frac{g_i b_i c_{out}}{f_{min}} = c_{in} \]

Misc. Comments

- Note that you never size the first gate
  - This gate is assumed to be fixed
  - If you were allowed to size it, the algorithm would try to make it as large as possible
- This is an estimation algorithm
  - Authors claim that sizing a gate by 1.5x too big or small still results in a path delay within 15% of minimum
Sensitivity Analysis

- How sensitive is delay to using exactly the best number of stages?
- $2.4 < \rho < 6$ gives delay within 15% of optimal
  - We can be sloppy!
  - I like $\rho = 4$

Evaluating Different Options

- $Cin=C$
  - The problem
  - 8C

- $Cin=C$
  - Option #1
  - 8C

- $Cin=C$
  - Option #2
  - 8C
**Option #1**

Cin=C

Path logic effort $G = g_0 \cdot g_1 \cdot g_2 = 1 \cdot 6/3 \cdot 1 = 2$
Path Branch effort $B = 1$
Path electrical effort $H = Cout/Cin = 8C/C = 8$
Path stage effort $F = G \cdot B \cdot H = 2 \cdot 1 \cdot 8 = 16$
Min delay: $= N \cdot (F)^{1/N} + P$

$= 3 \cdot (16)^{1/3} + (p_{inv} + 4 \cdot p_{inv} + p_{inv})$

$= 3 \cdot (2.5) + 6 = 13.5$

---

**Option #2**

Path logic effort $G = g_0 \cdot g_1 \cdot g_2 = 1 \cdot 4/3 \cdot 5/3 = 20/9$
Path Branch effort $B = 1$
Path electrical effort $H = Cout/Cin = 8C/C = 8$
Path stage effort $F = G \cdot B \cdot H = 20/9 \cdot 1 \cdot 8 = 160/9$
Min delay: $= N \cdot (F)^{1/N} + P$

$= 3 \cdot (160/9)^{1/3} + (p_{inv} + 2 \cdot p_{inv} + 2 \cdot p_{inv})$

$= 3 \cdot 2.6 + 5 = 12.8$

Option #2 appears to be better than Option #1, by a slight margin.

What if we consider gate area and power?
What about a 4-input NOR?
How many stages?

- Consider three alternatives for driving a load 25 times the input capacitance
  - One inverter
  - Three inverters in series
  - Five inverters in series
- They all do the job, but which one is fastest?

In all cases: \( G = 1, B = 1, \) and \( H = 25 \)
- Path delay is \( N(25)^{1/N} + N P_{\text{inv}} \)
  - \( N = 1, D = 26 \) units
  - \( N = 3, D = 11.8 \) units
  - \( N = 5, D = 14.5 \) units
- Since \( N=3 \) is best, each stage will bear an effort of \( (25)^{1/3} \approx 2.9 \)
  - So, each stage is \(~3x\) larger than the last
  - In general, the best stage effort is between 3 and 4 (not \( e \) as often stated)
    - The \( e \) value doesn’t use parasitics…
Choosing the Best # of Stages

- You can solve the delay equations to determine the number of stages \( N \) that will achieve the minimum delay
  - Approximate by \( \log_4 F \)

<table>
<thead>
<tr>
<th>Path Effort ( F )</th>
<th>Best ( N )</th>
<th>Min Delay ( D )</th>
<th>Stage effort ( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5.83</td>
<td>1</td>
<td>1.0-6.8</td>
<td>0-5.8</td>
</tr>
<tr>
<td>5.83-22.3</td>
<td>2</td>
<td>6.8-11.4</td>
<td>2.4-4.7</td>
</tr>
<tr>
<td>22.3-82.2</td>
<td>3</td>
<td>11.4-16.0</td>
<td>2.8-4.4</td>
</tr>
<tr>
<td>82.2-300</td>
<td>4</td>
<td>16.0-20.7</td>
<td>3.0-4.2</td>
</tr>
<tr>
<td>300-1090</td>
<td>5</td>
<td>20.7-25.3</td>
<td>3.1-4.1</td>
</tr>
<tr>
<td>1090-3920</td>
<td>6</td>
<td>25.3-29.8</td>
<td>3.2-4.0</td>
</tr>
</tbody>
</table>

Example

- String of inverters driving an off-chip load
  - Pad cap and load = 40pf
  - Equivalent to 20,000 microns of gate cap
  - Assume first inverter in chain has 7.2u of input cap
  - How many stages in inv chain?
  - \( H = 20,000/7.2 = 2777 \)
- From the table, 6 stages is best
- Stage effort = \( f = (2777)^{1/6} = 3.75 \)
- Path delay \( D = 6 \times 3.75 + 6 \times P_{\text{inv}} = 28.5 \)
  - \( D = 1.14\text{ns if } \tau = 40\text{ps} \)
Other N’s?

- N=2: \( f = (2777)^{1/2} = 52.7 \)
  - delay = 2(52.7) + 2 = 158.1 = 6.324ns
- N=3: \( f = (2777)^{1/3} = 14 \)
  - delay = 3(14) + 3 = 45 = 1.8ns
- N=4: \( f = (2777)^{1/4} = 7.26 \)
  - delay = 4(7.26) + 4 = 33.04 = 1.32ns
- N=5: \( f = (2777)^{1/5} = 4.88 \)
  - delay = 5(4.88) + 5 = 29.4 = 1.18ns
- N=6: delay = 1.14ns
- N=7: \( f = (2777)^{1/7} = 3.105 \)
  - delay = 7(3.105) + 7 = 28.7 = 1.15ns

Summary

- Compute path effort \( F = GBH \)
- Use table, or estimate \( N = \log_4 F \) to decide on number of stages
- Estimate minimum possible delay \( D = NF^{1/N} + \sum p_i \)
- Add or remove stages in your logic to get close to \( N \)
- Compute effort at each stage \( f_{\text{min}} = F^{1/N} \)
- Starting at output, work backwards to compute transistor sizes \( C_{in} = (g_i * b_i * C_{out}) / f_{\text{min}} \)
Limits of Logical Effort

- Chicken and egg problem
  - Need path to compute G
  - But don’t know number of stages without G
- Simplistic delay model
  - Neglects input rise time effects
- Interconnect
  - Iteration required in designs with wire
- Maximum speed only
  - Not minimum area/power for constrained delay

Summary

- Logical effort is useful for thinking of delay in circuits
  - Numeric logical effort characterizes gates
  - NANDs are faster than NORs in CMOS
  - Paths are fastest when effort delays are ~4
  - Path delay is weakly sensitive to stages, sizes
  - But using fewer stages doesn’t mean faster paths
  - Delay of path is about log₂F FO4 inverter delays
  - Inverters and NAND2 best for driving large caps
- Provides language for discussing fast circuits
  - But requires practice to master