Cray was a legend in computers … said that he liked to hire inexperienced engineers right out of school, because they do not usually know what’s supposed to be impossible.

_The Soul of a New Machine_, Kidder, pg. 77
Review: CMOS Inverter: Dynamic

\[ V_{in} = V_{DD} \]

\[ V_{out} \]

\[ R_n \]

\[ C_L \]

\[ t_{pHL} = f(R_n, C_L) \]

\[ t_{pHL} = 0.69 R_{eqn} C_L \]

\[ t_{pHL} = 0.69 \left( \frac{3}{4} (C_L V_{DD})/ I_{DSATn} \right) \]

\[ = 0.52 C_L / (W/L_n k_n V_{DSATn}) \]

Review: Designing Inverters for Performance

- **Reduce** \( C_L \)
  - internal diffusion capacitance of the gate itself
  - interconnect capacitance
  - fanout
- **Increase** \( W/L \) ratio of the transistor
  - the most powerful and effective performance optimization tool in the hands of the designer
  - watch out for self-loading!
- **Increase** \( V_{DD} \)
  - only minimal improvement in performance at the cost of increased energy dissipation
- **Slope engineering** - keeping signal rise and fall times smaller than or equal to the gate propagation delays and of approximately equal values
  - good for performance
  - good for power consumption
Switch Delay Model

Input Pattern Effects on Delay

- Delay is dependent on the **pattern** of inputs
- Low to high transition
  - both inputs go low
    - delay is $0.69 \frac{R_p}{2} C_L$ since two p-resistors are on in parallel
  - one input goes low
    - delay is $0.69 R_p C_L$
- High to low transition
  - both inputs go high
    - delay is $0.69 2R_n C_L$
- Adding transistors in series (without sizing) slows down the circuit
Delay Dependence on Input Patterns

2-input NAND with
NMOS = 0.5 µm/0.25 µm
PMOS = 0.75 µm/0.25 µm
CL = 10 fF

<table>
<thead>
<tr>
<th>Input Data Pattern</th>
<th>Delay (psec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=B=0 → 1</td>
<td>69</td>
</tr>
<tr>
<td>A=1, B=0 → 1</td>
<td>62</td>
</tr>
<tr>
<td>A= 0 → 1, B=1</td>
<td>50</td>
</tr>
<tr>
<td>A=B=1 → 0</td>
<td>35</td>
</tr>
<tr>
<td>A=1, B=1 → 0</td>
<td>76</td>
</tr>
<tr>
<td>A= 1 → 0, B=1</td>
<td>57</td>
</tr>
</tbody>
</table>

Transistor Sizing

- NMOS = 0.5 µm/0.25 µm
- PMOS = 0.75 µm/0.25 µm
- CL = 10 fF
Transistor Sizing a Complex CMOS Gate

\[ \text{OUT} = ! ( D + A \cdot (B + C)) \]
Fan-In Considerations

Distributed RC model (Elmore delay)

\[ t_{pHL} = 0.69 \, R_{eqn}(C_1 + 2C_2 + 3C_3 + 4C_L) \]

Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

\( t_p \) as a Function of Fan-In

Gates with a fan-in greater than 4 should be avoided.
Fast Complex Gates: Design Technique 1

- Transistor sizing
  - as long as fan-out capacitance dominates

- Progressive sizing
  - Distributed RC line
  - \( M_1 > M_2 > M_3 > \ldots > M_N \)
  - The fet closest to the output should be the smallest.
  - Can reduce delay by more than 20%; decreasing gains as technology shrinks

Fast Complex Gates: Design Technique 2

- Input re-ordering
  - when not all inputs arrive at the same time

Note: Diagrams showing the critical path with charged terminals.
Fast Complex Gates: Design Technique 2

- Input re-ordering
  - when not all inputs arrive at the same time

\[ \begin{array}{c}
\text{charged} \\
\text{charged} \\
\text{charged}
\end{array} \]

delay determined by time to discharge \( C_L, C_1 \) and \( C_2 \)

delay determined by time to discharge \( C_L \)

Sizing and Ordering Effects

Progressive sizing in pull-down chain gives up to a 23% improvement.

Input ordering saves 5% critical path A – 23%
critical path D – 17%
Fast Complex Gates: Design Technique 3

- Alternative logic structures

\[ F = ABCDEFGH \]

Fast Complex Gates: Design Technique 4

- Isolating fan-in from fan-out using buffer insertion

- Real lesson is that optimizing the propagation delay of a gate in isolation is misguided.
Fast Networks: Design Technique 5 - Logical Effort

- The optimum fan-out for a chain of N inverters driving a load \( C_L \) is

\[
f = \sqrt{\frac{C_L}{C_{in}}}
\]

- so, if we can, keep the fan-out per stage around 4.

- Can the same approach (logical effort) be used for any combinational circuit?
  - For a complex gate, we expand the inverter equation

\[
t_p = t_{p0} \left( 1 + \frac{C_{ext}}{\gamma C_g} \right) = t_{p0} \left( 1 + \frac{f}{\gamma} \right)
\]

\[
t_p = t_{p0} \left( p + g \frac{f}{\gamma} \right)
\]

- \( t_{p0} \) is the intrinsic delay of an inverter
- \( f \) is the effective fan-out \( \left( \frac{C_{ext}}{C_g} \right) \) – also called the electrical effort
- \( p \) is the ratio of the intrinsic (unloaded) delay of the complex gate and a simple inverter (a function of the gate topology and layout style)
- \( g \) is the logical effort

Intrinsic Delay Term, \( p \)

- The more involved the structure of the complex gate, the higher the intrinsic delay compared to an inverter

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>n-input NAND</td>
<td>( n )</td>
</tr>
<tr>
<td>n-input NOR</td>
<td>( n )</td>
</tr>
<tr>
<td>n-way mux</td>
<td>( 2n )</td>
</tr>
<tr>
<td>XOR, XNOR</td>
<td>( n 2^{n-1} )</td>
</tr>
</tbody>
</table>

Ignoring second order effects such as internal node capacitances
Logical Effort Term, $g$

- $g$ 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 it same output current)

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>$g$ (for 1 to 4 input gates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>NAND</td>
<td>4/3 5/3 (n+2)/3</td>
</tr>
<tr>
<td>NOR</td>
<td>5/3 7/3 (2n+1)/3</td>
</tr>
<tr>
<td>mux</td>
<td>2 2 2</td>
</tr>
<tr>
<td>XOR</td>
<td>4 12</td>
</tr>
</tbody>
</table>

Example of Logical Effort

- Assuming a pmos/nmos ratio of 2, the input capacitance of a minimum-sized inverter is three times the gate capacitance of a minimum-sized nmos ($C_{\text{unit}}$)
Example of Logical Effort

- Assuming a pmos/nmos ratio of 2, the input capacitance of a minimum-sized inverter is three times the gate capacitance of a minimum-sized nmos ($C_{\text{unit}}$).

\[
\begin{array}{c}
\text{A} \\
\text{B} \\
\text{A • B} \\
\end{array}
\]

\[
\begin{array}{c}
\text{A} \\
\text{B} \\
\text{A + B} \\
\end{array}
\]

$C_{\text{unit}} = 3$

$C_{\text{unit}} = 4$

$C_{\text{unit}} = 5$

Delay as a Function of Fan-Out

- The slope of the line is the logical effort of the gate.
- The y-axis intercept is the intrinsic delay.
- Can adjust the delay by adjusting the effective fan-out (by sizing) or by choosing a gate with a different logical effort.
- Gate effort: $h = fg$.
**Path Delay of Complex Logic Gate Network**

- Total path delay through a combinational logic block
  \[ t_p = \sum t_{p,j} = t_{p0} \sum (p_j + (f_j g_j)/\gamma) \]
- So, the minimum delay through the path determines that each stage should bear the same gate effort
  \[ f_1 g_1 = f_2 g_2 = \ldots = f_N g_N \]
- Consider optimizing the delay through the logic network

**Path Delay Equation Derivation**

- The path logical effort, \( G = \prod g_i \)
- And the path effective fan-out (path electrical effort) is \( F = C_L/C_{g1} \)
- The branching effort accounts for fan-out to other gates in the network
  \[ b = (C_{on-path} + C_{off-path})/C_{on-path} \]
- The path branching effort is then \( B = \prod b_j \)
- And the total path effort is then \( H = GFB \)
- So, the minimum delay through the path is
  \[ D = t_{p0} \left( \sum p_j + (N \sqrt{H}/\gamma) \right) \]
Path Delay of Complex Logic Gates, con’t

• For gate i in the chain, its size is determined by

\[ s_i = \left( \frac{g_1 s_1}{g_i} \right) \prod_{j=1}^{i-1} \left( \frac{f_j}{b_j} \right) \]

• For this network
  – \( F = \frac{C_L}{C_{g_1}} = 5 \)
  – \( G = 1 \times \frac{5}{3} \times \frac{5}{3} \times 1 = \frac{25}{9} \)
  – \( B = 1 \) (no branching)
  – \( H = GFB = 125/9, \) so the optimal stage effort is \( \sqrt{4H} = 1.93 \)
    • Fan-out factors are \( f_1 = 1.93, f_2 = 1.93 \times \frac{3}{5} = 1.16, f_3 = 1.16, f_4 = 1.93 \)
  – So the gate sizes are \( a = f_1 g_1/g_2 = 1.16, b = f_1 f_2 g_1/g_3 = 1.34 \) and \( c = f_1 f_2 f_3 g_1/g_4 = 2.60 \)

Fast Complex Gates: Design Technique 6

• Reducing the voltage swing

\[ t_{pHL} = 0.69 \left( \frac{3}{4} \left( C_L V_{DD} / I_{DSATn} \right) \right) \]

\[ = 0.69 \left( \frac{3}{4} \left( C_L V_{swing} / I_{DSATn} \right) \right) \]

– linear reduction in delay
– also reduces power consumption
– requires use of “sense amplifiers” on the receiving end to restore the signal level (will look at their design when covering memory design)
TG Logic Performance

- Effective resistance of the TG is modeled as a parallel connection of $R_p = (V_{DD} - V_{out})/(-I_{Dp})$ and $R_n = (V_{DD} - V_{out})/I_{Dn}$.

\[ R_{eq} = R_n || R_p \]

- So, the assumption that the TG switch has a constant resistive value, $R_{eq}$, is acceptable.

Delay of a TG Chain

- Delay of the RC chain (N TG’s in series) is

\[ t_p(V_n) = 0.69 \sum_{k=1}^{N} kC R_{eq} = 0.69 C R_{eq} (N(N+1))/2 \approx 0.35 C R_{eq} N^2 \]
TG Delay Optimization

- Can speed it up by inserting buffers every M switches

\[ t_p = 0.69 \left\lceil \frac{N}{M} \cdot C_{\text{req}} \left(\frac{M(M+1)}{2}\right) + \left(\frac{N}{M} - 1\right) t_{\text{tpbuf}} \right\rceil \]

\[ M_{\text{opt}} = 1.7 \sqrt{\frac{t_{\text{tpbuf}}}{C_{\text{req}}}} \approx 3 \text{ or } 4 \]