1 RC Modeling 3

2 Delay 5

   2.1. RC Delay of Inverter ......................... 5
   2.2. RC Delay of 2-Input NAND Gate .............. 6
   2.3. Equal Rise/Fall Times ...................... 9
   2.4. Equal Drive Strength ...................... 10
   2.5. Larger Gates ................................ 11
   2.6. Larger Loads ................................ 12
   2.7. Comparison of Inverter, NAND, NOR Gates ... 13
   2.8. Logical Effort: Single Stage ............... 16
   2.9. Logical Effort: Multiple Stages ............. 22

3 Energy 37

4 Area 44
1. RC Modeling

- $C_{sb}$ capacitors do not actually switch, so ignore
- Lump $C_{dbp} + C_{dbn}$ since both tied to constant nodes
- Lump $C_{gsp} + C_{gsn}$ since both tied to constant nodes
- Assume PMOS mobility is $2 \times$ worse than NMOS mobility
• Let $C$ be the gate capacitance of minimum sized NMOS
• Let $R$ be the effective resistance of a minimum sized NMOS
• Let $k$ be width of a transistor relative to minimum sized NMOS

2-Input NAND Gate

Draw and label the parasitic capacitances.
2. Delay

- We will initially use RC modeling to estimate delay
- We will then use RC modeling to derive logical effort (LE)
- LE is a fast way to estimate delay for simple static CMOS circuits
- Often need to use a mix of RC modeling and LE

2.1. RC Delay of Inverter

- Let \( t_{pd} \) be the propagation delay, time until \( V_{out} = V_{dd}/2 \)
2. Delay

2.2. RC Delay of 2-Input NAND Gate

\[
V_{out} = V_{dd} e^{-t/\tau}
\]

\[
\frac{V_{dd}}{2} = V_{dd} e^{-t/\tau}
\]

\[
\frac{1}{V_{dd}} \frac{V_{dd}}{2} = e^{-t/\tau}
\]

\[
\ln \left( \frac{1}{2} \right) = -\frac{t}{\tau}
\]

\[
-\tau \ln \left( \frac{1}{2} \right) = t
\]

\[
t = \tau \ln(2)
\]

- So \( t_{pd} = \ln(2) \cdot R C_1 \)
- Let \( R' = \ln(2) \cdot R \), so \( t_{pd} = R'C_1 \)
- For inverter on previous page, \( t_{pd} = 2R'C \)
- We usually just assume effective resistance is scaled by \( \ln(2) \)
- So propagation delay of inverter on previous page:

\[
t_{pd} = 2RC
\]

2.2. RC Delay of 2-Input NAND Gate

- Requires complicated 2nd order model
- We can use a simple approximation

\[
\tau = \tau_1 + \tau_2 = RC_1 + (R + R)C_2
\]

\[
= RC + (2R)(3C)
\]

\[
= RC + 6RC = 7RC \quad (3.5 \times \text{slower than inverter})
\]

- Best when one \( \tau \) much larger than the other \( \tau \)
- Even if \( \tau_1 = \tau_2 \), error is < 15%
2. Delay

2.2. RC Delay of 2-Input NAND Gate

Generalized Elmore Delay

\[ t_{pd} = \sum_{i} R_{ij} C_{i} \]

Assume all resistances are R and all capacitances are C

• Delay of path from x to y is impacted by branch to z
• Delay of path from x to z is impacted by branch to y
• For path x to y, lump \( C_2 + C_3 \) and use shared resistance \( R_0 + R_1 \)
• For path x to z, lump \( C_1 \) and use shared resistance \( R_0 + R_1 \)
• This extra term estimates impact of delay due to “branch”

\[
T_{pd,xy} = R_0 C_0 + (R_0 + R_1 + R_2) C_1 + (R_0 + R_1)(C_2 + C_3) \\
= RC + 3RC + 4RC = 8RC
\]

\[
T_{pd,xz} = R_0 C_0 + (R_0 + R_1 + R_3) C_2 + (R_0 + R_1 + R_3 + R_4) C_3 + (R_0 + R_1) C_1 \\
= RC + 3RC + 4RC + 2RC = 10RC
\]
Use Elmore Delay to Estimate Rise/Fall Times for 2-Input NAND Gate

\[ t_{pd,1 \rightarrow 0} \]
\[ A = 1 \]
\[ B = 0 \rightarrow 1 \]

\[ t_{pd,1 \rightarrow 0} \]
\[ A = 0 \rightarrow 1 \]
\[ B = 1 \]

\[ t_{pd,0 \rightarrow 1} \]
\[ A = 1 \rightarrow 0 \]
\[ B = 1 \rightarrow 0 \]

\[ t_{pd,0 \rightarrow 1} \]
\[ A = 1 \]
\[ B = 1 \rightarrow 0 \]

\[ t_{pd,0 \rightarrow 1} \]
\[ A = 1 \rightarrow 0 \]
\[ B = 1 \]
2.3. Equal Rise/Fall Times

For equal rise/fall times, the effective resistance of pullup must equal effective resistance of pulldown.

If we assume PMOS mobility $2 \times$ worse than NMOS, then PMOS must be $2 \times$ size of NMOS in an inverter for equal rise/fall times.
2.4. Equal Drive Strength

- Size transistors so worst case effective resistance is equal in both the pullup and pulldown networks.

\[
\begin{array}{c}
\text{inverter} \\
\text{2-input NAND w/o internal cap} \\
\text{2-input NAND w/ internal cap}
\end{array}
\]

- Is this a fair comparison? No, we are not normalizing anything across these gates. We need to either normalize:
  - Input gate gap (i.e., load on previous gate)
  - Drive strength (i.e., effective resistance)
• All three gates with equal rise/fall times and equal drive strengths

\[ t_{pd,1 \rightarrow 0} = \]

\[ t_{pd,0 \rightarrow 1} = \]

• This is the parasitic delay, independent of size \((k)\)
2.6. Larger Loads

\[ t_{pd,1} \rightarrow 0 = \]
2.7. Comparison of Inverter, NAND, NOR Gates

- Complete a fair comparison assuming equal rise/fall times, equal drive strength, only parasitic delay

<table>
<thead>
<tr>
<th></th>
<th>$t_{pd,1 \rightarrow 0}$</th>
<th>$t_{pd,0 \rightarrow 1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>worst</td>
<td>best</td>
</tr>
<tr>
<td>inverter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-input NAND</td>
<td>w/o internal cap</td>
<td></td>
</tr>
<tr>
<td>2-input NAND</td>
<td>w/ internal cap</td>
<td></td>
</tr>
<tr>
<td>2-input NOR</td>
<td>w/o internal cap</td>
<td></td>
</tr>
<tr>
<td>2-input NOR</td>
<td>w/ internal cap</td>
<td></td>
</tr>
</tbody>
</table>
Use Elmore Delay to Estimate Rise/Fall Times for 2-Input NOR Gate

$t_{pd,1 \rightarrow 0}$

A = 0 → 1  
B = 0 → 1

$t_{pd,1 \rightarrow 0}$

A = 0       
B = 0 → 1

$t_{pd,1 \rightarrow 0}$

A = 0 → 1    
B = 0

$t_{pd,0 \rightarrow 1}$

A = 0       
B = 1 → 0

$t_{pd,0 \rightarrow 1}$

A = 1 → 0   
B = 0
Use RC Modeling to Estimate Delay of 2-Input NAND and NOR Gates

- Ignore internal capacitance
- Assume worst case delay
- Assume an output load of 15C
2.8. Logical Effort: Single Stage

- Logic effort (LE) is just an abstraction over RC modeling
- Logic effort (LE) is a *linear delay model*
- Useful for building intuition for static CMOS modeling
- Keep in mind often need to use a mix of RC modeling and LE

\[
C_{in} = \alpha C_t \\
R_i = R_{ui} = R_{di} = R_t / \alpha \\
C_{pi} = \alpha C_{pt}
\]
• We know the propagation delay of the gate instance is:

\[ t_{pd} = R_i (C_{out} + C_{pi}) \]

• Let’s rewrite this in terms of the template

\[
C_{in} = \alpha C_t \\
R_i = R_{ui} = R_{di} = R_t / \alpha \\
C_{pi} = \alpha C_{pt}
\]

\[
t_{pd} = R_i (C_{out} + C_{pi}) \\
= R_i C_{out} + R_i C_{pi} \\
= \frac{R_t}{\alpha} C_{out} + \frac{R_t}{\alpha} C_{pi} \\
= \frac{R_t}{\alpha} C_{out} + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
= \frac{R_t}{\alpha} \left( \frac{C_{in}}{C_{in}} \right) C_{out} + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
= \frac{R_t}{\alpha} \alpha C_t \left( \frac{C_{out}}{C_{in}} \right) + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
= R_t C_t \left( \frac{C_{out}}{C_{in}} \right) + R_t C_{pt}
\]
• We don’t want to deal with absolute delay
• Let’s rewrite our propagation delay equation to be a “relative” delay
• Relative to the delay of a single unloaded minimal inverter
• Let’s start by defining some new parameters

\[ \tau = R_{inv}C_{inv} \]  
\[ g = \frac{R_t}{R_{inv}C_{inv}} \]  
\[ h = \frac{C_{out}}{C_{in}} \]  
\[ p = \frac{R_t C_{pt}}{R_{inv}C_{inv}} \] 

“relative delay units”
logical effort
electrical effort
parasitic delay
• Let’s rewrite our propagation delay equation in terms of $\tau$

$$
\tau = R_{inv}C_{inv} \quad \text{“relative delay units”}
$$

$$
g = \frac{R_tC_t}{R_{inv}C_{inv}} \quad \text{logical effort}
$$

$$
h = \frac{C_{out}}{C_{in}} \quad \text{electrical effort}
$$

$$
p = \frac{R_tC_{pt}}{R_{inv}C_{inv}} \quad \text{parasitic delay}
$$

$$
t_{pd} = d_{abs} = R_tC_t\left(\frac{C_{out}}{C_{in}}\right) + R_tC_{pt}
$$

$$
= \left(\frac{R_{inv}C_{inv}}{R_{inv}C_{inv}}\right)R_tC_t\left(\frac{C_{out}}{C_{in}}\right) + \left(\frac{R_{inv}C_{inv}}{R_{inv}C_{inv}}\right)R_tC_{pt}
$$

$$
= R_{inv}C_{inv}\left(\frac{R_tC_t}{R_{inv}C_{inv}}\right)\left(\frac{C_{out}}{C_{in}}\right) + R_{inv}C_{inv}\left(\frac{R_tC_{pt}}{R_{inv}C_{inv}}\right)
$$

$$
= \tau gh + \tau p
$$

$$
= \tau(gh + p)
$$

\[
\boxed{d_{abs} = \tau(gh + p)}
\]

• Let $d$ be the delay in units of $\tau$ (i.e., $d = gh + p$)
Templates for Inverter, NAND, NOR Gates

\[ \begin{align*}
R_{\text{Eff}} & = R \\
C_{\text{in}} & = 3C \\
\frac{R_t C_t}{R_{\text{in}} C_{\text{in}}} & = 1 \\
\frac{C_{\text{out}}}{C_{\text{in}}} & = \frac{R_{\text{out}}}{R_{\text{in}}} \\
\frac{g}{g} & = 1 \\
\frac{g}{g} & = 2
\end{align*} \]

Recall \( d = gh + p \) (linear delay model)

Plot \( d \) as a function of \( h \)

\[ \begin{align*}
\text{INV} & : d = h + 1 \\
\text{NAND} & : d = \frac{5}{3} h + 2 \\
\text{NOR} & : d = \frac{4}{3} h + 2
\end{align*} \]

Why we prefer NAND vs. NOR? NOR was higher delay for same electrical effort.

\[ h = \frac{C_{\text{out}}}{C_{\text{in}}} \]
Use LE to Estimate Delay of 2-Input NAND and NOR Gates

- Assume an output load of 15C

Let’s list the many approximations we have made
2.9. Logical Effort: Multiple Stages

- **Path delay** is the sum of the delay of each stage

\[
D = \sum d_i = \sum (g_i h_i + p_i)
\]

- Calculate path delay assuming canonical sized gates

| \(g_i\) | 1 | 5/3 | 4/3 | 1 |
| \(h_i\) | 5/3 | 4/5 | 3/4 | 40/3 |
| \(g_i \cdot h_i\) | 5/3 | 4/3 | 1 | 40/3 |
| \(p_i\) | 1 | 2 | 2 | 1 |
| \(d_i\) | 2.67 | 3.33 | 2 | 13.33 |
| \(D\) | 21.33 |

- Calculate path delay assuming final gate is X16

| \(g_i\) | 1 | 5/3 | 4/3 | 1 |
| \(h_i\) | 5/3 | 4/5 | 48/4 | 40/48 |
| \(g_i \cdot h_i\) | 5/3 | 4/3 | 16 | 40/48 |
| \(p_i\) | 1 | 2 | 2 | 1 |
| \(d_i\) | 2.67 | 3.33 | 18 | 0.83 |
| \(D\) | 24.83 |
Q1: How should we size gates to minimize total delay?

- Independent variables are $h_i$ (i.e., internal gate sizing)
- We want to choose $h_i$ to minimize $D$
- Take the partial derivative of $D$ with respect to $h_i$, set to zero, and solve for optimum $h_i$

\[
D = (g_1 h_1 + p_1) + (g_2 h_2 + p_2)
\]

- Note that $h_1$ and $h_2$ are constrained since $C_1$ and $C_3$ are given and input cap of gate 2 is output cap for gate 1

\[
h_1 = \frac{C_2}{C_1} \quad h_2 = \frac{C_3}{C_2} \quad h_1 h_2 = \frac{C_2 C_3}{C_1 C_2} = \frac{C_3}{C_1}
\]

- Let $H = h_1 h_2 = C_3 / C_1$, $H$ is a constant since $C_1$ and $C_3$ are given
- Let’s rework $D$ to get it in terms of just one variable

\[
D = (g_1 h_1 + p_1) + (g_2 h_2 + p_2)
\]

\[
D = g_1 h_1 + g_2 h_2 + (p_1 + p_2)
\]

\[
= g_1 h_1 + g_2 H h_1^{-1} + (p_1 + p_2)
\]
• Take partial derivative with respect to the only variable $h_1$

$$D = g_1h_1 + g_2Hh_1^{-1} + (p_1 + p_2)$$

$$\frac{\partial D}{\partial h_1} = g_1 - g_2Hh_1^{-2} + 0$$

$$= g_1 - \frac{g_2H}{h_1^2}$$

• Set partial derivative to zero and solve for $h_1$

$$\frac{\partial D}{\partial h_1} = g_1 - \frac{g_2H}{h_1^2} = 0$$

$$g_1 = \frac{g_2H}{h_1^2}$$

$$g_1h_1^2 = g_2H$$

$$h_1^2 = \frac{g_2}{g_1}H$$

$$h_1 = \sqrt{\frac{g_2}{g_1}H}$$

• Can use similar approach to find optimal $h_i$ for more than 2 stages
• However, there is actually a much more interesting result!

$$g_1h_1^2 = g_2H$$

$$g_1h_1^2 = g_2h_1h_2$$

$$g_1h_1 = g_2h_2$$

$$f_1 = f_2$$
• Delay is minimized when stage effort \( (f_i) \) is the same in both stages!
• Let \( \hat{f} \) be the optimal stage effort (i.e., \( \hat{f} = f_1 = f_2 \))
• We can use a trick to quickly calculate \( \hat{f} \)

\[
\hat{f} = \sqrt{\hat{f}^2} = \sqrt{\hat{f} \hat{f}} = \sqrt{f_1 f_2} \\
= \sqrt{(g_1 h_1)(g_2 h_2)} \\
= \sqrt{(g_1 g_2)(h_1 h_2)}
\]

• Let \( G = g_1 g_2 \), this is the path logical effort
• Let \( H = h_1 h_2 = C_{out}/C_{in} \), this is the path electrical effort
• Let \( F = GH \), this is the path effort

\[
\hat{f} = \sqrt{(g_1 g_1)(h_1 h_2)} \\
= \sqrt{GH} \\
= \sqrt{F}
\]

• We can calculate \( \hat{f} \) without finding the optimal size of each gate!
• Minimal delay with optimal sizing can be quickly calculated using:

\[
\hat{D} = 2\hat{f} + (p_1 + p_2)
\]
• This generalizes to paths with any number of stages

\[ G = \prod g_i \quad \text{path logical effort} \]

\[ H = \prod h_i = \frac{C_{out}}{C_{in}} \quad \text{path electrical effort} \]

\[ F = GH \quad \text{path effort} \]

\[ \hat{f} = F^{1/N} \quad \text{optimal stage effort} \]

\[ P = \sum p_i \quad \text{path parasitic delay} \]

\[ \hat{D} = N\hat{f} + P \quad \text{min delay with opt sizing} \]

**Method for optimal sizing**

1. Calculate path effort \((F = GH)\)
2. Calculate effort for each stage \((\hat{f} = F^{1/N})\)
3. Estimate minimum delay with optimal sizing \((\hat{D} = N\hat{f} + P)\)
4. Starting with last stage, work backwards sizing each gate

\[ \hat{f} = gh = g \frac{C_{out}}{C_{in}} \quad C_{in} = \frac{g}{\hat{f}} C_{out} \]
Revisit earlier example

\[
\begin{array}{ccccc}
g_i & 1 & 5/3 & 4/3 & 1 \\
p_i & 1 & 2 & 2 & 1 \\
\end{array}
\]
Optimal sizing with standard cells

- This assumes we can size gates arbitrarily using full custom design
- What about if we are using a standard cell library?
- Assume we have a standard cell library with the following cells
  - INVX1, INVX2, INVX4, INVX8
  - NANDX1, NANDX2, NANDX4
  - NORX1, NORX2, NORX4

- Assume we have determined optimal sizing in $C_{in}$
- How do we figure out which standard cell to use?
• Given optimum $C_{in}$ from before, what is $\alpha$?

<table>
<thead>
<tr>
<th>$C_{in}$</th>
<th>$g$</th>
<th>$C_{in}/(g \times 3C) = \alpha$</th>
<th>gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.17C</td>
<td>1</td>
<td>17.17C/(1 \times 3C) = 5.72</td>
<td>INVX4</td>
</tr>
<tr>
<td>9.83C</td>
<td>4/3</td>
<td>9.83C/((4/3) \times 3C) = 2.45</td>
<td>NANDX2</td>
</tr>
<tr>
<td>7.03C</td>
<td>5/3</td>
<td>7.03C/((5/3) \times 3C) = 1.41</td>
<td>NORX1</td>
</tr>
<tr>
<td>3.02C</td>
<td>1</td>
<td>3.02C/(1 \times 3C) = 1.00</td>
<td>INVX1</td>
</tr>
</tbody>
</table>

• Recalculate actual delay given these gates
• First calculate actual $C_{in}$ for each standard cell gate

<table>
<thead>
<tr>
<th>gate</th>
<th>$\alpha$</th>
<th>$g$</th>
<th>$\alpha \times g \times 3C = C_{in}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVX4</td>
<td>4</td>
<td>1</td>
<td>$4 \times 1 \times 3C = 12C$</td>
</tr>
<tr>
<td>NANDX2</td>
<td>2</td>
<td>4/3</td>
<td>$2 \times 4/3 \times 3C = 8C$</td>
</tr>
<tr>
<td>NORX1</td>
<td>1</td>
<td>5/3</td>
<td>$1 \times 5/3 \times 3C = 5C$</td>
</tr>
<tr>
<td>INVX1</td>
<td>1</td>
<td>1</td>
<td>$1 \times 1 \times 3C = 3C$</td>
</tr>
</tbody>
</table>

• Now use path delay equation

$$D = \sum gh + \sum p$$

$$= (1 \times \frac{40}{12}) + (\frac{4}{3} \times \frac{12}{8}) + (\frac{5}{3} \times \frac{8}{5}) + (1 \times \frac{5}{3}) + (1 + 2 + 2 + 1)$$

$$= 3.33 + 2 + 2.67 + 1.67 + 6 = 9.67 + 6 = 15.67$$

• Compare with optimal delay which is 15.32, off by 2.3%
What about branching?

- Consider the following simple example

\[
G = 1 \times 1 = 1 \\
H = \frac{90C}{5C} = 18 \\
F = GH = 18 \\
F = \prod g_i h_i \\
= (1 \times 6) \times (1 \times 6) = 36
\]

- So in this example \( F = 2GH \)
- The factor of two is called the branching effort
- Key Idea: some drive current is directed off path we are analyzing
- Similar to Elmore delay for trees

\[
b = \frac{C_{\text{onpath}} + C_{\text{offpath}}}{C_{\text{onpath}}} \quad \text{stage branching effort}
\]

\[
B = \prod b_i \quad \text{path branching effort}
\]

- So our new path effort equation is now:

\[
F = \prod f_i = GBH
\]

- Note that path effort depends on circuit topology and loading of entire path, but not size of transistors in network
- Note that path effort does not change if we add or remove inverters!
Q2: How should we change topology to minimize delay?

- Assume we want to implement an eight input AND gate
- Calculate min delay assuming optimal sizing for three topologies
- First assume $H = 1$, then assume $H = 12$

<table>
<thead>
<tr>
<th>Topology</th>
<th>$NF^{1/N}$</th>
<th>$P$</th>
<th>$\hat{D}$</th>
<th>$NF^{1/N}$</th>
<th>$P$</th>
<th>$\hat{D}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAND4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAND2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Delay

2.9. Logical Effort: Multiple Stages

Determine optimal number of stages for chain of inverters

\[ \hat{D} = NF^{1/N} + NP_{inv} \]

\[ \frac{\partial \hat{D}}{\partial N} = F^{1/N} - F^{1/N} \ln(F^{1/N}) + P_{inv} = 0 \]

- If \( P_{inv} = 0 \)

\[ \frac{\partial \hat{D}}{\partial N} = F^{1/N} - F^{1/N} \ln(F^{1/N}) = 0 \]

\[ \ln(F^{1/N}) = 1 \]

\[ F^{1/N} = e \]

\[ \hat{f} = e \]

- So if we assume \( P_{inv} = 0 \), then the optimal number of stages results in a stage effort of \( e \) (i.e., 2.718) for every stage

- Since \( G = 1 \) for an inverter, this means \( h = 2.718 \) for every stage
• If $P_{inv} = 1$, then we need to solve this nonlinear equation:

$$F^{1/N} - F^{1/N} \ln(F^{1/N}) + 1 = 0$$

• Let $\rho = F^{1/\hat{N}}$ where $\hat{N}$ is optimal number of stages

$$1 + \rho(1 - \ln(\rho)) = 0$$

• We can solve this numerically to find that $\rho \approx 3.59$
• So if we assume $P_{inv} = 1$, then the optimal number of stages results in a stage effort of 3.59 for every stage
• Since $G = 1$ for an inverter, this means $h = 3.59$ for every stage
• We can roughly approximate 3.59 to be 4
• Let’s solve for $\hat{N}$ as a function of $F$

$$F^{1/\hat{N}} = 4$$

$$\log(F^{1/\hat{N}}) = \log(4)$$

$$\frac{1}{\hat{N}} \log(F) = \log(4)$$

$$\hat{N} = \frac{\log(F)}{\log(4)} = \log_4(F)$$

• This is actually a pretty good estimate even for a path of gates which are not inverters!
logical effort can help give us intuition on how to size gates and choose a topology to minimize delay but it may have limitations.

To deal with more complicated scenarios, we can also write the delay equations for each gate in a system and minimize the latest arrival time.

**Example**

![Diagram showing gate delays and arrival times]

Let's write our linear delay equations as a function of $\alpha$.

\[
\begin{align*}
  d &= gh + p \\
  g &= \frac{12 \cdot C_T}{q \cdot C_{\text{inv}}} \quad C_{\text{inv}} = \alpha \cdot C_T \\
  C_T &= \frac{C_{\text{inv}}}{\alpha}
\end{align*}
\]

\[
\begin{align*}
  d &= \frac{C_{\text{inv}} \cdot C_T}{p} + p \\
  g &= \frac{C_{\text{inv}}}{d \cdot C_{\text{inv}}}
\end{align*}
\]

\[
\begin{align*}
  d &= \frac{C_{\text{inv}}}{3d} + p \\
  \text{delay as function of } \alpha \text{ (size cell drive)}
\end{align*}
\]

Recall $C_{\text{inv}} = 3g \alpha$
2.9. Logical Effort: Multiple Stages

Now unit delay can now be found for each stage

\[
\begin{align*}
    d_0 & = \frac{(y_0 + 3d_2)}{3d_1} + 1 = \frac{y_0}{3} d_1 + \frac{1}{3} d_2 + 1 \\
    d_1 & = \frac{(y_1 + 3d_2)}{3d_1} + 2 = \frac{y_1}{3} d_2 + 2 \\
    d_2 & = \frac{(y_2 + 3d_3)}{3d_2} + 3 = \frac{y_2}{3} d_3 + 3 \\
    d_3 & = \frac{(y_3)}{3d_3} + 1 = \frac{1}{3} d_3 + 1
\end{align*}
\]

Arrival Times

\[
\begin{align*}
    t_0 & = d_0 \\
    t_1 & = \max(t_0, d_{w}) + d_1 \\
    t_2 & = \max(t_0, t_1) + d_2 \\
    t_3 & = t_2 + d_3 \\
    t_4 & = \max(t_0, t_1) + d_2 + d_3 \\
        & = \max(d_0, \max(d_0, d_{w}) + d_1) + d_2 + d_3
\end{align*}
\]
\[ t_j = \max (d_0, \max (d_0, d_{in}) + d_1) + d_2 + d_3 \]

Minimize \( t_j \) subject to above constraints with \( d_0, d_2, d_j \) as the independent variables.

Actually in synthesis we really want to minimize area (or weight) subject to constraint on \( t_j \).

So we could craft optimization problem to be minimize sum of \( d_1, d_2, d_j \) (prox for area) subject to constraint:

\[ t_{\text{clk period}} > \max (d_0, \max (d_0, d_{in}) + d_{in}) + d_2 + d_3 \]

\( \Rightarrow \) clock period constraint
3. Energy

- Energy is a measure of work
- Power is the rate at which work is done

<table>
<thead>
<tr>
<th>Electric Potential</th>
<th>Capacity for doing work which arises from position of a charge in an electric field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential Energy</td>
<td>Electric potential energy of a position per unit charge</td>
</tr>
<tr>
<td>Electric Potential</td>
<td>Volts</td>
</tr>
<tr>
<td>Current</td>
<td>Rate at which charge flows past position</td>
</tr>
<tr>
<td>Power</td>
<td>Rate at which electric energy is supplied or consumed</td>
</tr>
</tbody>
</table>

- $1V = 1J/C$
- $\Delta V = \Delta E/Q$
- $1A = 1C/S$
- $I = Q/\Delta t$
- $1W = 1J/S$
- $P = \Delta E/\Delta t = \frac{\Delta V \cdot Q}{Q/I} = VI$

\[ E = \int_{0}^{T} P(t) \, dt \]
3. Energy

Energy Stored on a Capacitor

\[ E_C = \int_0^\infty P(t)dt = \int_0^\infty V(t)I(t)dt \]
\[ = \int_0^\infty V(t)\frac{dQ}{dt}dt = \int_0^\infty V(t)\frac{C}{dt}dvdt \]
\[ = C\int_0^{V_{DD}} V(t)dv = \frac{1}{2}CV_{DD}^2 \]

- So on 1 → 0 input transition, \( \frac{1}{2}CV_{DD}^2 \) is stored on capacitor
- This energy is released on 0 → 1 input transition

Energy Delivered From Power Supply

\[ E_{\text{supply}} = \int_0^\infty P(t)dt = \int_0^\infty V_{DD}I(t)dt \]
\[ = V_{DD} \int_0^\infty \frac{dQ}{dt}dt = V_{DD} \int_0^\infty \frac{C}{dt}dvdt \]
\[ = CV_{DD} \int_0^{V_{DD}} dV = CV_{DD}^2 \]

- 0 → 1 output transition
  - \( CV_{DD}^2 \) energy is delivered from power supply
  - half this energy dissipated as heat in PMOS
  - half this energy is stored on the capacitor

- 1 → 0 output transition
  - no energy is delivered from power supply
  - remaining energy on capacitor dissipated as heat in NMOS
3. Energy

- On average, each bit transition requires $\frac{1}{2}CV_{DD}^2$
- Let $\alpha$ be the activity factor, probability of a bit transitions per cycle

$$E_{\text{node}} = \alpha \frac{1}{2}CV_{DD}^2$$

**Power Consumption**

$$P_{\text{tot}} = P_{\text{switching}} + P_{\text{static}}$$

$$= \alpha f \frac{1}{2}CV_{DD}^2 + V_{DD}I_{\text{off}}$$

- Sometimes engineers will assume $\alpha$ is the probability of just a $0 \rightarrow 1$ output transition instead of the probability of any transition
  - $\alpha = \text{probability of any transition}$
  - $\alpha' = \text{probability of a } 0 \rightarrow 1 \text{ transition only}$
- If you use $\alpha'$ then do not include the factor of $1/2$
- Note that book uses $\alpha$ but it is really $\alpha'$ in our notation!
Comparing Energy

- Calculate the total switched cap in worst case

\[
\begin{align*}
  g &= \frac{10}{3} \\
  p &= 8 \\
  G &= \frac{10}{3} \\
  H &= 1 \\
  B &= 1 \\
  F &= \frac{10}{3} \\
  \hat{f} &= 1.8
\end{align*}
\]

\[
\begin{align*}
  C_{\text{inv},g} &= \frac{1}{1.8} \times 10 = 5.6 \\
  C_{\text{nand},g} &= \frac{10/3}{1.8} \times 5.6 = 10.4 \\
  C_{\text{tot},g} &= C_{\text{inv},g} + 8C_{\text{nand},g} = 88.8C
\end{align*}
\]

\[
\begin{align*}
  g &= 2 \\
  p &= 4 \\
  G &= \frac{10}{3} \\
  H &= 1 \\
  B &= 1 \\
  F &= \frac{10}{3} \\
  \hat{f} &= 1.8
\end{align*}
\]

\[
\begin{align*}
  C_{\text{nor},g} &= \frac{5/3}{1.8} \times 10 = 9.3 \\
  C_{\text{nand},g} &= \frac{2}{1.8} \times 9.3 = 10.3 \\
  C_{\text{tot},g} &= 2C_{\text{nor},g} + 8C_{\text{nand},g} = 101C
\end{align*}
\]

- To determine parasitic cap need to understand how gate cap is distributed across transistors
3. Energy

**4-Wire NAND Gate**

\[ C_{\text{inv.out}} = 5.6 \]

\[ C_{\text{inv.in},p} = 5.6 \]

**2-Wire NAND Gate**

\[ C_{\text{nan.in}},q = 10.4 \]

\[ C_{\text{nan.in},p} = 24.96 \]

**2-Wire NOR Gate**

\[ C_{\text{nor.in}},q = 9.3 \]

\[ C_{\text{nor.in},p} = 11.16 \]

**4-Wire NAND Gate**

\[ C_{\text{nan.in}},q = 10.3 \]

\[ C_{\text{nan.in},p} = 20.58 \]
$E_{\text{node}} = \alpha \frac{1}{2} CV_{\text{DD}}^2$

- Assume $\alpha = 0.1$ and $V_{\text{DD}} = 1\text{V}$ for both
- Only difference is amount of switched cap

- For 8-input NAND topology

$$C_{\text{tot}} = C_{\text{tot},g} + C_{\text{tot},p} = 88.8 + (5.6 + 24.96) = 119.36\text{C}$$

- For 4-input NAND topology

$$C_{\text{tot}} = C_{\text{tot},g} + C_{\text{tot},p} = 101 + (11.16 + 20.58) = 132.74\text{C}$$

- So second topology requires 10% more energy in the worst case
- Worst case is when all capacitance is switched
- This ignores the energy for switching the output load
- Let’s assume $C = 0.5\text{ fF}$ (see extra notes)
- Assume clock frequency is 500 MHz

$$E = \alpha \frac{1}{2} CV_{\text{DD}}^2 = 0.1 \times \frac{1}{2} \times 120\text{C} \times \frac{0.5fF}{C} \times (1\text{V})^2 = 3\text{fJ}$$

$$P = \alpha f \frac{1}{2} CV_{\text{DD}}^2 = (0.5 \times 10^9)(30 \times 10^{-15}) = 1.5\mu\text{W}$$
Activity Factors

- Previous example used fixed $\alpha = 0.1$ for all nodes
- Can improve accuracy by:
  - Propagate activity factor of inputs to internal nodes
  - Use RTL to calculate activity of inputs, then propagate
  - Use gate-level simulation to find activity of each node

\[ P_i = \text{probability node is one on cycle } i \]
\[ \bar{P}_i = 1 - P_i = \text{probability node is zero on cycle } i \]
\[ \alpha = P_{i-1}\bar{P}_i + \bar{P}_{i-1}P_i \]
\[ \alpha' = \bar{P}_{i-1}P_i \]

- Assuming inputs have uncorrelated random data
- Each of these is equally likely: 0→0, 0→1, 1→0, 1→1

\[ \alpha = P_{i-1}\bar{P}_i + \bar{P}_{i-1}P_i = 0.5 \]
\[ \alpha' = \bar{P}_{i-1}P_i = 0.25 \]
\[ \alpha' = \frac{1}{2}\alpha \]
4. Area

Output Activity Factor of NAND2

- Calculate output activity factor of a NAND2 gate
- Assume inputs are uncorrelated random data
- Output of NAND2 is zero if both inputs one, otherwise output is one

\[
\alpha'_{out} = \bar{P}_{out, i-1} P_{out, i} \\
= (P_A P_B)(1 - P_A P_B) \\
= (0.5 \times 0.5)(1 - 0.5 \times 0.5) \\
= (0.25)(1 - 0.25) \\
= 0.1875
\]

Output Activity Factor of NAND8

4. Area

- Sum the transistor widths across all transistors in design
- Use standard cell footprints