Transient Response: Delay Models

Dr. Bassam Jamil

Adopted from slides of the textbook
Topics

- Delay definitions
- Delay Calculations Using Differential Equations
- RC Delay Model
  - Elmore Delay
- Linear Delay Model
  - Logical Effort
Transient Response

- **DC analysis** tells us $V_{out}$ if $V_{in}$ is constant
- **Transient analysis** tells us $V_{out}(t)$ if $V_{in}(t)$ changes
  - Requires solving differential equations
  - Usually, we are calculate the delays, transition time, power, energy
- Input is usually considered to be a step or ramp
  - From 0 to $V_{DD}$ or vice versa
Delay Definitions

- $t_{\text{pdr}}$: rising propagation delay
  - From input to rising output crossing $V_{\text{DD}}/2$
- $t_{\text{pdf}}$: falling propagation delay
  - From input to falling output crossing $V_{\text{DD}}/2$
- $t_{\text{pd}}$: average propagation delay
  - $t_{\text{pd}} = (t_{\text{pdr}} + t_{\text{pdf}})/2$
- $t_{\text{r}}$: rise time
  - From output crossing 0.2 $V_{\text{DD}}$ to 0.8 $V_{\text{DD}}$
- $t_{\text{f}}$: fall time
  - From output crossing 0.8 $V_{\text{DD}}$ to 0.2 $V_{\text{DD}}$
Delay Definitions

- \( t_{cdr} \): rising contamination delay
  - From input to rising output crossing \( V_{DD}/2 \)
- \( t_{cdf} \): falling contamination delay
  - From input to falling output crossing \( V_{DD}/2 \)
- \( t_{cd} \): average contamination delay
  - \( t_{pd} = (t_{cdr} + t_{cdf})/2 \)
Simulated Inverter Delay

- Solving differential equations by hand is too hard
- SPICE simulator solves the equations numerically
  - Uses more accurate I-V models too!
- But simulations take time to write, may hide insight

\[ V_{\text{in}} \] and \[ V_{\text{out}} \] with

- \[ t_{\text{pdf}} = 66\text{ps} \]
- \[ t_{\text{pdr}} = 83\text{ps} \]
Inverter Step Response

- Ex: find step response of inverter driving load cap

\[ V_{in}(t) = u(t - t_0)V_{DD} \]

\[ V_{out}(t < t_0) = V_{DD} \]

\[ \frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}} \]

\[ I_{dsn}(t) = \begin{cases} 
0 & t \leq t_0 \\
\frac{\beta}{2}(V_{DD} - V_t)^2 & V_{out} > V_{DD} - V_t \\
\beta\left(V_{DD} - V_t - \frac{V_{out}(t)}{2}\right)V_{out}(t) & V_{out} < V_{DD} - V_t 
\end{cases} \]
Delay Estimation

- We would like to be able to easily estimate delay
  - Not as accurate as simulation
  - But easier to ask “What if?”

- The step response usually looks like a 1st order RC response with a decaying exponential.

- Use RC delay models to estimate delay
  - C = total capacitance on output node
  - Use effective resistance R
  - So that \( t_{pd} = RC \)

- Characterize transistors by finding their effective R
  - Depends on average current as gate switches
Effective Resistance

- Shockley models have limited value
  - Not accurate enough for modern transistors
  - Too complicated for much hand analysis

- Simplification: treat transistor as resistor
  - Replace $I_{ds}(V_{ds}, V_{gs})$ with effective resistance $R$
    - $I_{ds} = \frac{V_{ds}}{R}$
  - $R$ averaged across switching of digital gate

- Too inaccurate to predict current at any given time
  - But good enough to predict RC delay
RC Delay Model

- Use equivalent circuits for MOS transistors
  - Ideal switch + capacitance and ON resistance
  - Unit nMOS has resistance $R$, capacitance $C$
  - Unit pMOS has resistance $2R$, capacitance $C$
- Capacitance proportional to width
- Resistance inversely proportional to width
RC Values

- Capacitance
  - \( C = C_g = C_s = C_d = 2 \, \text{fF/}\mu\text{m} \) of gate width in 0.6 \( \mu\text{m} \)
  - Gradually decline to 1 fF/\( \mu\text{m} \) in 65 nm

- Resistance
  - \( R \approx 10 \, \text{K}\Omega \cdot \mu\text{m} \) in 0.6 \( \mu\text{m} \) process
  - Improves with shorter channel lengths
  - 1.25 K\( \Omega \cdot \mu\text{m} \) in 65 nm process

- Unit transistors
  - May refer to minimum contacted device (\( 4/2 \, \lambda \))
  - Or maybe 1 \( \mu\text{m} \) wide device
  - Doesn’t matter as long as you are consistent
Estimate the delay of a fanout-of-1 inverter

d = 6RC
Delay Model Comparison

![Graph showing delay model comparison with axes labeled (V) and t(s)]
Example: 3-input NAND

Sketch a 3-input NAND with transistor widths chosen to achieve effective rise and fall resistances equal to a unit inverter (R).

![3-input NAND circuit diagram]
Annotate the 3-input NAND gate with gate and diffusion capacitance.

For n-input NAND Gate
3-input NAND Equivalent Circuit
Transient Response Using RC Model

- For first order RC, the step response

\[ V_{out}(t) = V_{DD}e^{-t/\tau} \]

\[ t_{pd} = RC \ln 2 \]

The factor of \( \ln 2 = 0.69 \) is cumbersome. The effective resistance \( R \) is an empirical parameter anyway, so it is preferable to incorporate the factor of \( \ln 2 \) to define a new effective resistance \( R' = R \ln 2 \). Now the propagation delay is simply \( R'C \). For the sake of convenience, we usually drop the prime symbols and just write

\[ t_{pd} = RC \] (4.9)
For first order RC, the step response:

\[
V_{\text{out}}(t) = V_{DD} \frac{\tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}}{\tau_1 - \tau_2}
\]

The above solution is complex. Elmore model offers a mechanism to simplify the RC network.

\[
\tau_{1,2} = \frac{R_1 C_1 + (R_1 + R_2) C_2}{2} \left( 1 \pm \sqrt{1 - \frac{4 R^* C^*}{\left[1 + (1 + R^*) C^*\right]^2}} \right)
\]

\[
R^* = \frac{R_2}{R_1}; \quad C^* = \frac{C_2}{C_1}
\]
Elmore Delay

- ON transistors look like resistors
- Pullup or pulldown network modeled as RC ladder
- Elmore delay of RC ladder

\[ t_{pd} \approx \sum_{\text{nodes } i} R_{i\to\text{source}} C_i \]

\[ = R_1 C_1 + \left( R_1 + R_2 \right) C_2 + \ldots + \left( R_1 + R_2 + \ldots + R_N \right) C_N \]

Diagram:
- RC ladder network
- \( R_1, R_2, R_3, \ldots, R_N \) resistors
- \( C_1, C_2, C_3, \ldots, C_N \) capacitors
- Current flowing from source to output

5: DC and Transient Response  CMOS VLSI Design 4th Ed.
Example: INV

Estimate $t_{pd}$ for a unit inverter driving $m$ identical unit inverters.

$t_{pd} = (3 + 3m)RC$. 

Diagram showing a CMOS inverter circuit with annotations for estimating $t_{pd}$. 

3: CMOS Transistor Theory

CMOS VLSI Design 4th Ed.
Example: 3-input NAND

- Estimate worst-case rising and falling delay of 3-input NAND driving \( h \) identical gates.

\[ t_{pdr} = (9 + 5h)RC \]

\[ t_{pdf} = (3C)\left(\frac{R}{3}\right) + (3C)\left(\frac{R}{3} + \frac{R}{3}\right) + \left[(9 + 5h)C\right]\left(\frac{R}{3} + \frac{R}{3} + \frac{R}{3}\right) = (12 + 5h)RC \]
Delay Components

- Delay has two parts
  - *Parasitic delay*
    - 9 or 12 RC
    - Independent of load
  - *Effort delay*
    - 5h RC
    - Proportional to load capacitance
Contamination Delay

- Best-case (contamination) delay can be substantially less than propagation delay.
- Ex: If all three inputs fall simultaneously

\[
 t_{cdr} = \left[ (9 + 5h)C \right] \left( \frac{R}{3} \right) = \left( 3 + \frac{5}{3}h \right) RC
\]

\[
 t_{cdr} = \left[ (9 + 5h)C \right] \left( \frac{R}{3} \right) = \left( 3 + \frac{5}{3}h \right) RC
\]
Diffusion Capacitance

- We assumed contacted diffusion on every s / d.
- Good layout minimizes diffusion area
- Ex: NAND3 layout shares one diffusion contact
  - Reduces output capacitance by 2C
  - Merged uncontacted diffusion might help too
Linear Delay Model and Logical Effort
Outline

- Logical Effort
- Delay in a Logic Gate
- Multistage Logic Networks
- Choosing the Best Number of Stages
- Example
- Summary
Introduction

- Chip designers face a bewildering array of choices
  - What is the best circuit topology for a function?
  - How many stages of logic give least delay?
  - How wide should the transistors be?

- Logical effort is a method to make these decisions
  - Uses a simple model of delay
  - Allows rough and quick calculations
  - Helps make rapid comparisons between alternatives
  - Emphasizes remarkable symmetries
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

\[
d = \frac{d_{abs}}{\tau}
\]

\( \tau = 3RC \)

\[\approx 3 \text{ ps in 65 nm process} \]
\[60 \text{ ps in 0.6 \( \mu \)m process} \]
Electrical Effort:

\[ h = \frac{C_{out}}{C_{in}} \]

Normalized Delay:

\[ d = f + p = gh + p \]

Effort Delay:

\[ f \]

Parasitic Delay:

\[ p \]

Delay Plots: Inv, NAND

\[ d = f + p = gh + p \]

Inverter:

- \( g = 1 \)
- \( p = 1 \)
- \( d = h + 1 \)

2-input NAND:

- \( g = \frac{4}{3} \)
- \( p = 2 \)
- \( d = \left(\frac{4}{3}\right)h + 2 \)
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 and calculations]

- \( C_{in} = 3 \)
  \( g = 3/3 \)

- \( C_{in} = 4 \)
  \( g = 4/3 \)

- \( C_{in} = 5 \)
  \( g = 5/3 \)
## 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>

Not Our Focus
### Catalog of Gates

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

<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 in multiples of $p_{\text{inv}} \approx 1$
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: \( d = 2 \)
- Period: \( T = 2*N*d \)
- Frequency: \( f_{osc} = 1/T = 1/(4N* \tau) \)

Assuming that \( \tau = 3 \text{ps} \), \( N = 31 \) \( \Rightarrow \) \( f_{osc} = 2.7 \text{ GHz} \)

Notice how delay calculations are process independent.
Example: FO4 Inverter

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

\[ d \]

Logical Effort: \( g = 1 \)
Electrical Effort: \( h = 4 \)
Parasitic Delay: \( p = 1 \)
Stage Delay: \( d = 5 \)

In 65nm process, \( \tau = 3ps \) \( \Rightarrow \) delay = \( d \times \tau = 15 \) ps
In 0.6 \( \mu \)m process \( \tau = 60ps \) \( \Rightarrow \) delay = \( d \times \tau = 300 \) ps

Notice how delay calculations are process independent.
Logical Effort For Multi-Stage Paths

- 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** \( F = \prod f_i = \prod g_i h_i \)

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

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

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

\[ B = \prod b_i \quad \text{Note:} \quad \prod h_i = BH \]

- 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.
Path Example

- $N = 4$
- $H = 2$
- $G = 20/9$
- $B = 1$
- $F = HGB = 40/9$

- $g_1 = 1$
  $h_1 = x/10$
- $g_2 = 5/3$
  $h_2 = y/x$
- $g_3 = 4/3$
  $h_3 = z/y$
- $g_4 = 1$
  $h_4 = 20/z$

- $f = (F)^{1/N} = 1.45$
- $X = 14.52$
- $Y = 12.64$
- $Z = 13.77$
Example with Branches

No! Consider paths that branch:

\[
\begin{align*}
G & = 1 \\
H & = 90 / 5 = 18 \\
GH & = 18 \\
h_1 & = (15 + 15) / 5 = 6 \\
h_2 & = 90 / 15 = 6 \\
F & = g_1g_2h_1h_2 = 36 = 2GH \\
\end{align*}
\]

OR

\[
B=2 \Rightarrow F=GBH= 2(1)(18)=36
\]
Example: 3-stage path

- Select gate sizes x and y for least delay from A to B

![Diagram of a 3-stage path with gate sizes x and y.](image)
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 \( \hat{f} = \sqrt[3]{F} = 5 \)

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

- Minimizing number of stages is not always fastest

Example: drive 64-bit datapath with unit inverter

\[
D = NF^{1/N} + P = N(64)^{1/N} + N
\]
Best Number of Stages

- Consider adding inverters to end of path
  - How many give least delay?

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

\[ \frac{\partial D}{\partial N} = -F^{\frac{1}{N}} \ln F^{\frac{1}{N}} + F^{\frac{1}{N}} + p_{inv} = 0 \]

- Define best stage effort \( \rho = F^{\frac{1}{N}} \)

\[ p_{inv} + \rho (1 - \ln \rho) = 0 \]
Best Stage Effort

- $p_{inv} + \rho \left(1 - \ln \rho\right) = 0$ has no closed-form solution

- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = 2.718$ (e)
- For $p_{inv} = 1$, solve numerically for $\rho = 3.59$
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 $\rho = 4$
Example, Revisited

- Ben Bitdiddle is the memory designer for the Motoroil 68W86, an embedded automotive processor. Help Ben design the decoder for a register file.

- Decoder specifications:
  - 16 word register file
  - Each word is 32 bits wide
  - Each bit presents load of 3 unit-sized transistors
  - True and complementary address inputs $A[3:0]$
  - Each input may drive 10 unit-sized transistors

- Ben needs to decide:
  - How many stages to use?
  - How large should each gate be?
  - How fast can decoder operate?
Number of Stages

- Decoder effort is mainly electrical and branching
  
  Electrical Effort: \[ H = \frac{(32 \times 3)}{10} = 9.6 \]
  
  Branching Effort: \[ B = 8 \]

- If we neglect logical effort (assume \( G = 1 \))
  
  Path Effort: \[ F = GBH = 76.8 \]

  Number of Stages: \[ N = \log_4 F = 3.1 \]

- Try a 3-stage design
Gate Sizes & Delay

Logical Effort: \[ G = 1 \times \frac{6}{3} \times 1 = 2 \]

Path Effort: \[ F = GBH = 154 \]

Stage Effort: \[ \hat{f} = F^{1/3} = 5.36 \]

Path Delay: \[ D = 3 \hat{f} + 1 + 4 + 1 = 22.1 \]

Gate sizes: \[ z = \frac{96 \times 1}{5.36} = 18 \quad y = \frac{18 \times 2}{5.36} = 6.7 \]
Comparison

- Compare many alternatives with a spreadsheet
- \( D = N(76.8 \ G)^{1/N} + P \)

<table>
<thead>
<tr>
<th>Design</th>
<th>N</th>
<th>G</th>
<th>P</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>234</td>
</tr>
<tr>
<td>NAND4-INV</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>29.8</td>
</tr>
<tr>
<td>NAND2-NOR2</td>
<td>2</td>
<td>20/9</td>
<td>4</td>
<td>30.1</td>
</tr>
<tr>
<td>INV-NAND4-INV</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>22.1</td>
</tr>
<tr>
<td>NAND4-INV-INV-INV</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>21.1</td>
</tr>
<tr>
<td>NAND2-NOR2-INV-INV</td>
<td>4</td>
<td>20/9</td>
<td>6</td>
<td>20.5</td>
</tr>
<tr>
<td>NAND2-INV-NAND2-INV</td>
<td>4</td>
<td>16/9</td>
<td>6</td>
<td>19.7</td>
</tr>
<tr>
<td>INV-NAND2-INV-NAND2-INV</td>
<td>5</td>
<td>16/9</td>
<td>7</td>
<td>20.4</td>
</tr>
<tr>
<td>NAND2-INV-NAND2-INV-INV-INV</td>
<td>6</td>
<td>16/9</td>
<td>8</td>
<td>21.6</td>
</tr>
</tbody>
</table>
## Review of Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Stage</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of stages</td>
<td>1</td>
<td>$N$</td>
</tr>
<tr>
<td>logical effort</td>
<td>$g$</td>
<td>$G = \prod g_i$</td>
</tr>
<tr>
<td>electrical effort</td>
<td>$h = \frac{C_{\text{out}}}{C_{\text{in}}}$</td>
<td>$H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$</td>
</tr>
<tr>
<td>branching effort</td>
<td>$b = \frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}}$</td>
<td>$B = \prod b_i$</td>
</tr>
<tr>
<td>effort</td>
<td>$f = gh$</td>
<td>$F = GBH$</td>
</tr>
<tr>
<td>effort delay</td>
<td>$f$</td>
<td>$D_F = \sum f_i$</td>
</tr>
<tr>
<td>parasitic delay</td>
<td>$p$</td>
<td>$P = \sum p_i$</td>
</tr>
<tr>
<td>delay</td>
<td>$d = f + p$</td>
<td>$D = \sum d_i = D_F + P$</td>
</tr>
</tbody>
</table>
1) Compute path effort
2) Estimate best number of stages
3) Sketch path with N stages
4) Estimate least delay
5) Determine best stage effort
6) Find gate sizes

\[ F = GBH \]

\[ N = \log_4 F \]

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

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

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