CPE/EE 427, CPE 527  
VLSI Design I  
L13: Wires, Design for Speed  

Department of Electrical and Computer Engineering  
University of Alabama in Huntsville  

Aleksandar Milenkovic ( www.ece.uah.edu/~milenka )  
www.ece.uah.edu/~milenka/cpe527-05F  

Course Administration  

• Instructor: Aleksandar Milenkovic  
milenka@ece.uah.edu  
www.ece.uah.edu/~milenka  
EB 217-L  
Mon. 5:30 PM – 6:30 PM,  
Wen. 12:30 – 13:30 PM  
• URL:  
http://www.ece.uah.edu/~milenka/cpe527-05F  
• TA: Joel Wilder  
• Labs:  
Lab#4: due 10/14/05; Lab#5: 10/21/05  
• Hws: Solutions in secure directory /scr (cpe427fall05, ?)  
• Project:  
Proposals due was on 10/10/05  
• Test I:  
10/17/05  
• Text:  
CMOS VLSI Design, 3rd ed., Weste, Harris  
• Review:  
Chapters 1, 2, 3, 4;  
• Today:  
Wires, Design for Speed (meet AM in the Lab tonight)
Outline

- Introduction
- Wire Resistance
- Wire Capacitance
- Wire RC Delay
- Crosstalk
- Wire Engineering
- Repeaters

Introduction

- Chips are mostly made of wires called *interconnect*
  - In stick diagram, wires set size
  - Transistors are little things under the wires
  - Many layers of wires
- Wires are as important as transistors
  - Speed
  - Power
  - Noise
- Alternating layers run orthogonally
Wire Geometry

- Pitch = w + s
- Aspect ratio: AR = t/w
  - Old processes had AR ≪ 1
  - Modern processes have AR ≈ 2
    - Pack in many skinny wires

Layer Stack

- AMI 0.6 µm process has 3 metal layers
- Modern processes use 6-10+ metal layers
- Example:
  - Intel 180 nm process
  - M1: thin, narrow (< 3λ)
    - High density cells
  - M2-M4: thicker
    - For longer wires
  - M5-M6: thickest
    - For V_{DD}, GND, clk

<table>
<thead>
<tr>
<th>Layer</th>
<th>T (nm)</th>
<th>W (nm)</th>
<th>S (nm)</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1720</td>
<td>860</td>
<td>860</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1800</td>
<td>800</td>
<td>800</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1080</td>
<td>540</td>
<td>540</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>700</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>700</td>
<td>320</td>
<td>320</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>700</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>700</td>
<td>320</td>
<td>320</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>700</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>480</td>
<td>250</td>
<td>250</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Wire Resistance

- \( \rho \) = resistivity \((\Omega \cdot \text{m})\)

\[
R = \frac{\rho l}{t w} = R_{\square} \frac{l}{w}
\]

- \( R_{\square} \) = sheet resistance \((\Omega / \square)\)
  - \( \square \) is a dimensionless unit (!)

- Count number of squares
  - \( R = R_{\square} \times \) (# of squares)

Choice of Metals

- Until 180 nm generation, most wires were aluminum
- Modern processes often use copper
  - Cu atoms diffuse into silicon and damage FETs
  - Must be surrounded by a diffusion barrier

<table>
<thead>
<tr>
<th>Metal</th>
<th>Bulk resistivity ((\mu \Omega \cdot \text{cm}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silver (Ag)</td>
<td>1.6</td>
</tr>
<tr>
<td>Copper (Cu)</td>
<td>1.7</td>
</tr>
<tr>
<td>Gold (Au)</td>
<td>2.2</td>
</tr>
<tr>
<td>Aluminum (Al)</td>
<td>2.8</td>
</tr>
<tr>
<td>Tungsten (W)</td>
<td>5.3</td>
</tr>
<tr>
<td>Molybdenum (Mo)</td>
<td>5.3</td>
</tr>
</tbody>
</table>
Sheet Resistance

• Typical sheet resistances in 180 nm process

<table>
<thead>
<tr>
<th>Layer</th>
<th>Sheet Resistance (Ω/□)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion (silicided)</td>
<td>3-10</td>
</tr>
<tr>
<td>Diffusion (no silicide)</td>
<td>50-200</td>
</tr>
<tr>
<td>Polysilicon (silicided)</td>
<td>3-10</td>
</tr>
<tr>
<td>Polysilicon (no silicide)</td>
<td>50-400</td>
</tr>
<tr>
<td>Metal1</td>
<td>0.08</td>
</tr>
<tr>
<td>Metal2</td>
<td>0.05</td>
</tr>
<tr>
<td>Metal3</td>
<td>0.05</td>
</tr>
<tr>
<td>Metal4</td>
<td>0.03</td>
</tr>
<tr>
<td>Metal5</td>
<td>0.02</td>
</tr>
<tr>
<td>Metal6</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Contacts Resistance

• Contacts and vias also have 2-20 Ω
• Use many contacts for lower R
  – Many small contacts for current crowding around periphery
Wire Capacitance

- Wire has capacitance per unit length
  - To neighbors
  - To layers above and below
- \( C_{\text{total}} = C_{\text{top}} + C_{\text{bot}} + 2C_{\text{adj}} \)

Capacitance Trends

- Parallel plate equation: \( C = \varepsilon A/d \)
  - Wires are not parallel plates, but obey trends
  - Increasing area (W, t) increases capacitance
  - Increasing distance (s, h) decreases capacitance
- Dielectric constant
  - \( \varepsilon = k\varepsilon_0 \)
  - \( \varepsilon_0 = 8.85 \times 10^{-14} \text{ F/cm} \)
  - \( k = 3.9 \) for SiO\(_2\)
- Processes are starting to use low-k dielectrics
  - \( k \approx 3 \) (or less) as dielectrics use air pockets
M2 Capacitance Data

- Typical wires have ~ 0.2 fF/µm
  - Compare to 2 fF/µm for gate capacitance

![M2 Capacitance Data Graph]

Diffusion & Polysilicon

- Diffusion capacitance is very high (about 2 fF/µm)
  - Comparable to gate capacitance
  - Diffusion also has high resistance
  - Avoid using diffusion runners for wires!
- Polysilicon has lower C but high R
  - Use for transistor gates
  - Occasionally for very short wires between gates
**Lumped Element Models**

- Wires are a distributed system
  - Approximate with lumped element models

![Diagram of lumped element models](image)

- 3-segment $\pi$-model is accurate to 3% in simulation
- L-model needs 100 segments for same accuracy!
- Use single segment $\pi$-model for Elmore delay

---

**Example**

- Metal2 wire in 180 nm process
  - 5 mm long
  - 0.32 µm wide
- Construct a 3-segment $\pi$-model
  - $R_{\square} =$
  - $C_{\text{permicron}} =$
Example

• Metal2 wire in 180 nm process
  – 5 mm long
  – 0.32 \( \mu \)m wide
• Construct a 3-segment \( \pi \)-model
  – \( R_{\square} = 0.05 \, \Omega/\square \) \( \Rightarrow R = 781 \, \Omega \)
  – \( C_{\text{permicron}} = 0.2 \, fF/\mu m \) \( \Rightarrow C = 1 \, \text{pF} \)

\[ \begin{array}{ccc}
260 \, \Omega & & 260 \, \Omega \\
\downarrow & | & \downarrow \\
167 \, fF & | & 167 \, fF \\
\downarrow & & \downarrow \\
260 \, \Omega & & 260 \, \Omega \\
\end{array} \]

Wire RC Delay

• Estimate the delay of a 10x inverter driving a 2x inverter at the end of the 5mm wire from the previous example.
  – \( R = 2.5 \, k\Omega \times \mu m \) for gates
  – Unit inverter: 0.36 \( \mu m \) nMOS, 0.72 \( \mu m \) pMOS

\[ t_{pd} = \]
Wire RC Delay

• Estimate the delay of a 10x inverter driving a 2x inverter at the end of the 5mm wire from the previous example.
  – $R = 2.5 \, \text{k}\Omega \cdot \mu\text{m}$ for gates
  – Unit inverter: 0.36 $\mu$m nMOS, 0.72 $\mu$m pMOS

\[ \begin{align*}
\text{Driver} & \quad 781 \, \Omega \\
\text{Wire} & \quad 690 \, \Omega \quad 500 \, \text{fF} \quad 500 \, \text{fF} \\
\text{Load} & \quad 4 \, \text{fF}
\end{align*} \]

– $t_{pd} = 1.1 \, \text{ns}$

Simulated Wire Delays

\[ \begin{align*}
&V_{in} \quad \overline{L/10} \quad L/4 \quad L/2 \quad L \\
&V_{out}
\end{align*} \]

\[ \begin{align*}
\text{Voltage (V)} & \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \\
\text{Time (nsec)} & \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \quad 3 \quad 3.5 \quad 4 \quad 4.5 \quad 5
\end{align*} \]
Wire Delay Models

- Ideal wire
  - same voltage is present at every segment of the wire at every point in time - at equi-potential
  - only holds for very short wires, i.e., interconnects between very nearest neighbor gates
- Lumped C model
  - when only a single parasitic component (C, R, or L) is dominant the different fractions are lumped into a single circuit element
    - When the resistive component is small and the switching frequency is low to medium, we can consider only C; the wire itself does not introduce any delay; the only impact on performance comes from wire capacitance
  - good for short wires; pessimistic and inaccurate for long wires

Wire Delay Models, con’t

- Lumped RC model
  - total wire resistance is lumped into a single R and total capacitance into a single C
  - good for short wires; pessimistic and inaccurate for long wires
- Distributed RC model
  - circuit parasitics are distributed along the length, L, of the wire
    - c and r are the capacitance and resistance per unit length
  - Delay is determined using the Elmore delay equation
    \[ \tau_{Di} = \sum_{k=1}^{N} C_k r_{ik} \]
Chain Network Elmore Delay

Elmore delay equation

\[ \tau_{DN} = \sum c_i r_{ij} = \sum c_i \sum r_j \]

\[ \tau_{D1} = c_1 r_1 \]
\[ \tau_{D2} = c_1 r_1 + c_2 (r_1 + r_2) \]
\[ \tau_{Di} = c_1 r_{eq} + 2c_2 r_{eq} + 3c_3 r_{eq} + \ldots + ic_i r_{eq} \]
Distributed RC Model for Simple Wires

• A length L RC wire can be modeled by N segments of length L/N
  – The resistance and capacitance of each segment are given by \( r \frac{L}{N} \) and \( c \frac{L}{N} \)

\[
\tau_{DN} = \left(\frac{L}{N}\right)^2 (cr + 2cr + \ldots + Ncr) = (crL^2) \frac{N(N+1)}{2N^2} = CR\frac{(N+1)}{2N})
\]

where \( R (= rL) \) and \( C (= cL) \) are the total lumped resistance and capacitance of the wire

• For large \( N \)

\[
\tau_{DN} = \frac{RC}{2} = \frac{rcL^2}{2}
\]

• Delay of a wire is a quadratic function of its length, \( L \)
• The delay is \( 1/2 \) of that predicted (by the lumped model)

Putting It All Together

- Total propagation delay consider driver and wire

\[
\tau_D = R_{\text{Driver}} C_w + \frac{(R_w C_w)}{2} = R_{\text{Driver}} C_w + 0.5 r_w c_w L^2
\]

and \( t_p = 0.69 R_{\text{Driver}} C_w + 0.38 R_w C_w \)

where \( R_w = r_w L \) and \( C_w = c_w L \)

- The delay introduced by wire resistance becomes dominant when \( (R_w C_w)/2 \geq R_{\text{Driver}} C_w \) (when \( L \geq 2 R_{\text{Driver}}/R_w \))
  – For an \( R_{\text{Driver}} = 1 \, \text{kΩ} \) driving an 1 \( \mu \text{m} \) wide Al1 wire, \( L_{\text{crit}} \) is 2.67 cm
Design Rules of Thumb

• RC delays should be considered when \( t_{pRC} > t_{pgate} \) of the driving gate

\[
L_{crit} > \sqrt{\left( \frac{t_{pgate}}{0.38rc} \right)}
\]

– actual \( L_{crit} \) depends upon the size of the driving gate and the interconnect material

• RC delays should be considered when the rise (fall) time at the line input is smaller than RC, the rise (fall) time of the line

\[ t_{rise} < RC \]

– when not met, the change in the signal is slower than the propagation delay of the wire so a lumped C model suffices

Delay with Long Interconnects

• When gates are farther apart, wire capacitance and resistance can no longer be ignored.

\[
t_p = 0.69R_{dr}C_{int} + (0.69R_{dr} + 0.38R_w)C_w + 0.69(R_{dr} + R_w)C_{fan}
\]

where \( R_{dr} = (R_{eqn} + R_{eqp})/2 \)

\[
= 0.69R_{dr}(C_{int} + C_{fan}) + 0.69(R_{dr}C_w + r_wC_{fan})L + 0.38r_wC_wL^2
\]

• Wire delay rapidly becomes the dominate factor (due to the quadratic term) in the delay budget for longer wires.
Crosstalk

- A capacitor does not like to change its voltage instantaneously.
- A wire has high capacitance to its neighbor.
  - When the neighbor switches from 1→0 or 0→1, the wire tends to switch too.
  - Called capacitive coupling or crosstalk.
- Crosstalk effects
  - Noise on nonswitching wires
  - Increased delay on switching wires

Crosstalk Delay

- Assume layers above and below on average are quiet
  - Second terminal of capacitor can be ignored
  - Model as Cgnd = Ctop + Cbot
- Effective Cadj depends on behavior of neighbors
  - Miller effect

<table>
<thead>
<tr>
<th>B</th>
<th>$\Delta V$</th>
<th>$C_{eff(A)}$</th>
<th>MCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switching with A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switching opposite A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Crosstalk Delay

- Assume layers above and below on average are quiet
  - Second terminal of capacitor can be ignored
  - Model as C_{gnd} = C_{top} + C_{bot}
- Effective C_{adj} depends on behavior of neighbors
  - Miller effect

\[
\begin{array}{|c|c|c|}
\hline
B & \Delta V & C_{\text{eff}(A)} & MCF \\
\hline
\text{Constant} & V_{\text{DD}} & C_{\text{gnd}} + C_{\text{adj}} & 1 \\
\text{Switching with A} & 0 & C_{\text{gnd}} & 0 \\
\text{Switching opposite A} & 2V_{\text{DD}} & C_{\text{gnd}} + 2C_{\text{adj}} & 2 \\
\hline
\end{array}
\]

Crosstalk Noise

- Crosstalk causes noise on nonswitching wires
- If victim is floating:
  - model as capacitive voltage divider

\[
\Delta V_{\text{victim}} = \frac{C_{\text{adj}}}{C_{\text{gnd-v}} + C_{\text{adj}}} \Delta V_{\text{aggressor}}
\]

\[
\begin{array}{c}
\Delta V_{\text{aggressor}} \\
\hline
\text{Aggressor} \\
\hline
\Delta V_{\text{victim}} \\
\text{Victim} \\
\hline
C_{\text{adj}} \\
C_{\text{gnd-v}} \\
\hline
\end{array}
\]
Driven Victims

- Usually victim is driven by a gate that fights noise
  - Noise depends on relative resistances
  - Victim driver is in linear region, agg. in saturation
  - If sizes are same, $R_{\text{aggressor}} = 2-4 \times R_{\text{victim}}$

$$
\Delta V_{\text{victim}} = \frac{C_{\text{adj}}}{C_{\text{gnd-v}} + C_{\text{adj}}} \frac{1}{1+k} \Delta V_{\text{aggressor}}
$$

$$
k = \frac{\tau_{\text{aggressor}}}{\tau_{\text{victim}}} = \frac{R_{\text{aggressor}}}{R_{\text{victim}}} \frac{C_{\text{gnd-a}} + C_{\text{adj}}}{C_{\text{gnd-v}} + C_{\text{adj}}}
$$

Coupling Waveforms

- Simulated coupling for $C_{\text{adj}} = C_{\text{victim}}$
Noise Implications

- *So what* if we have noise?
- If the noise is less than the noise margin, nothing happens
- Static CMOS logic will eventually settle to correct output even if disturbed by large noise spikes
  - But glitches cause extra delay
  - Also cause extra power from false transitions
- Dynamic logic never recovers from glitches
- Memories and other sensitive circuits also can produce the wrong answer

Wire Engineering

- Goal: achieve delay, area, power goals with acceptable noise
- Degrees of freedom:
Wire Engineering

- Goal: achieve delay, area, power goals with acceptable noise
- Degrees of freedom:
  - Width
  - Spacing
  - Layer

[Graphs showing delay and coupling vs. pitch and spacing]
Wire Engineering

• Goal: achieve delay, area, power goals with acceptable noise
• Degrees of freedom:
  – Width
  – Spacing
  – Layer
  – Shielding

Repeaters

• R and C are proportional to $l$
• RC delay is proportional to $l^2$
  – Unacceptably great for long wires
Repeater Design

• How many repeaters should we use?
• How large should each one be?
• Equivalent Circuit
  – Wire length \( l/N \)
    • Wire Capacitance \( C_w^{*}l/N \), Resistance \( R_w^{*}l/N \)
    – Inverter width \( W \) (nMOS = \( W \), pMOS = \( 2W \))
      • Gate Capacitance \( C^{*}W \), Resistance \( R/W \)
Repeater Design

• How many repeaters should we use?
• How large should each one be?
• Equivalent Circuit
  – Wire length $l$
    • Wire Capacitance $C_w l$, Resistance $R_w l$
  – Inverter width $W$ (nMOS = $W$, pMOS = 2$W$)
    • Gate Capacitance $C'_W$, Resistance $R/W$

\[
\frac{R_w}{N} \quad \frac{R/W}{C_w/2N} \quad \frac{R/W}{C_w/2N} \quad C'_W
\]

Repeater Results

• Write equation for Elmore Delay
  – Differentiate with respect to $W$ and $N$
  – Set equal to 0, solve

\[
\frac{l}{N} = \sqrt{\frac{2RC'}{R_wC_w}}
\]

\[
\frac{t_{pd}}{l} = \left(2 + \sqrt{2}\right) \sqrt{\frac{RC'W}{R'_wC_w}} \quad \text{~60-80 ps/mm in 180 nm process}
\]

\[
W = \sqrt{\frac{RC_w}{R'_wC'}}
\]
Designing for Speed

Department of Electrical and Computer Engineering
University of Alabama in Huntsville

Review: CMOS Inverter: Dynamic

\[ t_{pHL} = f(R_n, C_L) \]
\[ t_{pHL} = 0.69 \, R_{eqn} \, C_L \]
\[ t_{pHL} = 0.69 \left( \frac{3}{4} \left( C_L \, V_{DD} / I_{DSATn} \right) \right) \]
\[ = 0.52 \, C_L \, / \left( W/L_n \, k'_n \, V_{DSATn} \right) \]
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\,\mu m/0.25 \,\mu m
PMOS = 0.75\,\mu m/0.25 \,\mu m
$C_L = 10 \,\text{fF}$

<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

Fan-In Considerations

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

Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.
**Fast Complex Gates: Design Technique 1**

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

- Progressive sizing
  - Distributed RC line
  - \( \text{M1} > \text{M2} > \text{M3} > \ldots > \text{MN} \)
  - 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

  \[
  \begin{align*}
  &\text{critical path} \\
  &\text{charged} \\
  &\text{charged} \\
  &\text{charged} \\
  &\text{discharged} \\
  &\text{discharged}
  \end{align*}
  \]

  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.

Logical Effort: Design Technique 5

- 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
  \[ F = \prod f_i = \prod g_i h_i \]
Branching Effort

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

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

\[
B = \prod h_i
\]

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

Note:
\[
\prod h_i = BH
\]

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.
Best Number of Stages

• How many stages should a path use?
  – Minimizing number of stages is not always fastest
• Example: drive 64-bit datapath with unit inverter

\[
D = \frac{N}{f} + P = \frac{N}{64} + N
\]
Derivation

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

\[
D = NF^{1/2} + \sum_{i=1}^{n_1} p_i + (N - n_1)p_{inv}
\]

\[
\frac{\partial D}{\partial N} = -F^{1/2} \ln F^{1/2} + F^{1/2} + p_{inv} = 0
\]

- Define best stage effort \( \rho = F^{1/2} \)

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

Best Stage Effort

- 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 $\rho = 4$