Week 4:

Logical Effort Theory (for Gate Sizing)

Prof. Dejan Markovic
UCLA

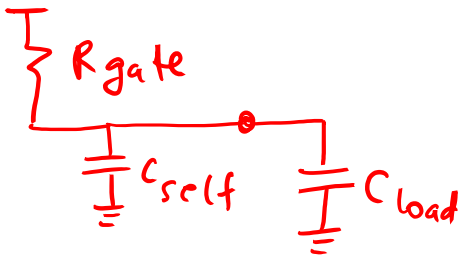
Week 4 Agenda

- <u>The Concept of Logical Effort</u>
- Buffer Chain Example
- Multi-Stage Logic Networks

The Concept of Logical Effort

- **Instead of running lots of simulations**
  - Simplified: (almost) back-of-envelope calculations for speed performance

- **Basic concept:**
  - Delay = $R_{gate}(C_{load} + C_{self}) = \underbrace{R_{gate}C_{load}}_{\text{load}} + \underbrace{R_{gate}C_{self}}_{\text{internal (parasitic)}}$
  - Logical Effort basic equation: $\boxed{d = f + p}$ (load, int)
    - d is the delay (normalized)
    - f is known as the effort delay.
    - p is known as the parasitic delay.



  - $d = \text{Delay}/\tau = (R_{gate}C_{load} + R_{gate}C_{self})/R_0C_0$
    - Normalized to the delay of a FO-1 inverter (no self load) $R_{INV} \cdot C_{in,INV}$
    - With $R_0 = R_{gate}$, d = fanout + normalized parasitic.
    - So f is essentially equivalent to fanout.
    - d is a measure that is independent of process, voltage, temp.

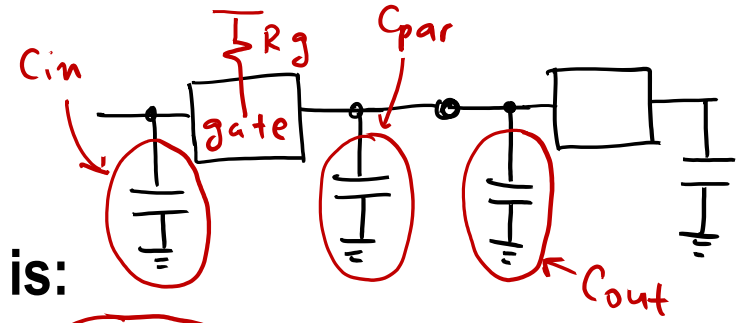
The Logical Effort Way of Thinking…

- **Gate delay we used up to now:**

$$t_p = 0.69 \cdot R_g \cdot (C_{par} + C_{out})$$

$C_{par} = C_{int}$

$R_g = R_{on}$



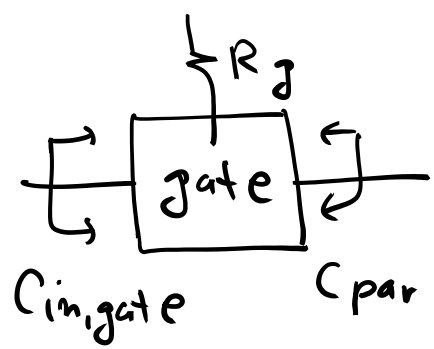
- **Another way to write this formula is:**

$$t_p = \underbrace{0.69\, R_g \cdot C_{in,gate}}_{\tau_{gate}} \left(\frac{C_{par}}{C_{in,gate}} + \frac{C_{out}}{C_{in,gate}} \right)$$

$$= \tau_{gate} \cdot \left(\gamma_{gate} + \frac{C_{out}}{C_{in,gate}} \right)$$

parasitic

fanout



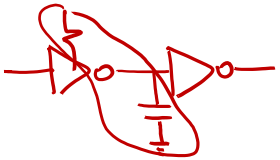
Now Normalize the Delay ($\tau = ?$)

Strategy: normalize to a time constant of an inverter

- **Approach 1: normalize to fictitious "technology time constant"**

$$d = \boxed{\frac{Delay}{\tau_{INV}}} = \frac{\tau_{gate}}{\tau_{INV}}\left(\gamma_{gate} + \frac{C_{out}}{C_{in,gate}}\right)$$

$\tau_{INV} = R_{INV} \cdot C_{in,INV}$



- **Approach 2: normalize to intrinsic delay of inverter**

$$\frac{Delay}{t_{p0,INV}} = \frac{\tau_{gate}}{t_{p0,INV}}\left(\gamma_{gate} + \frac{C_{out}}{C_{in,gate}}\right)$$

$t_{p0} = R_{INV} \cdot C_{int}$

Both formulations exist in the literature

<u>We use approach 1</u> from the original logical effort theory

(Doesn't really matter – it's just a normalization constant)

Normalized Delay

Strategy: normalize to a time constant of an inverter

- **Approach 1: normalize to fictitious "technology time constant"**

$$\frac{Delay}{\tau_{INV}} = \frac{\tau_{gate}}{\tau_{INV}}\left(\gamma_{gate} + \frac{C_{out}}{C_{in,gate}}\right)$$

- **Normalized delay:** $\boldsymbol{d = g \cdot (\gamma_{gate} + h)}$

- **Even simpler:** $\boxed{\boldsymbol{d = g \cdot h + p \cdot \gamma}}$

***gh* – load** (size dependent)
***pγ* – parasitic** (size independent)

Logical effort terms

- **Logical effort (g)** **Electrical fanout (h)** **Parasitic effort (p)**

$$\boldsymbol{g = \frac{R_{gate} \cdot C_{in,gate}}{R_{INV} \cdot C_{in,INV}}} \qquad \boldsymbol{h = \frac{C_{out}}{C_{in,gate}}} \qquad \boldsymbol{p = \frac{C_{par,gate}}{C_{par,INV}}}$$

The Meaning of Logical Effort Terms: Summary

Logical effort terms

- **Logical effort (g)** **Electrical fanout (h)** **Parasitic effort (p)**

$$g = \frac{R_{gate} \cdot C_{in,gate}}{R_{INV} \cdot C_{in,INV}} \qquad h = \frac{C_{out}}{C_{in,gate}} \qquad p = \frac{C_{par,gate}}{C_{par,INV}}$$

Intuition

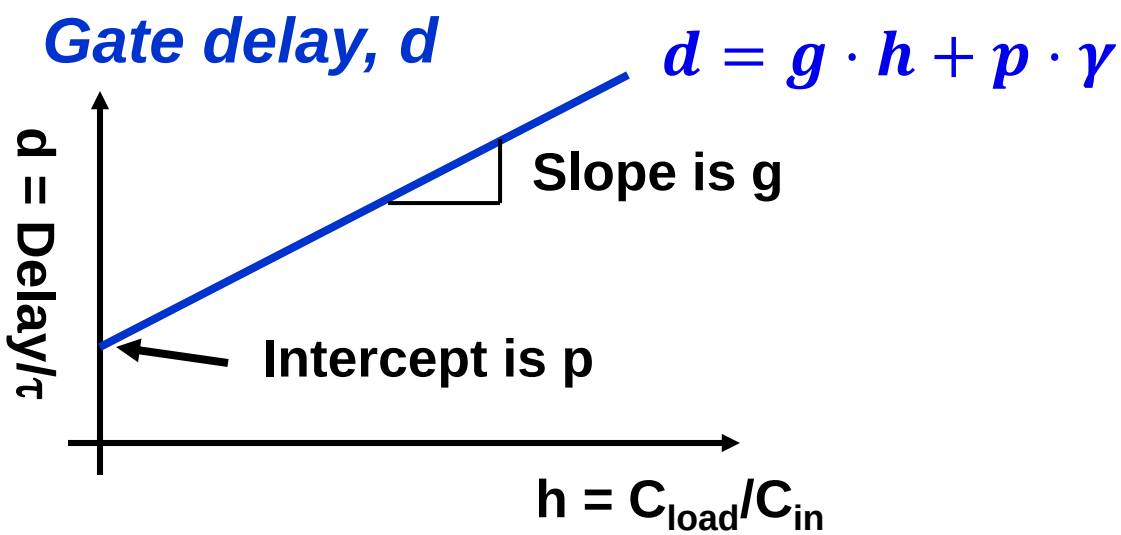
- **Logical effort (g)**
  - R_{on} ratio for equal C_{in} (or, equivalently, C_{in} ratio for **equal R_{on}**)

$$g = \frac{C_{in,gate}}{C_{in,INV}}$$

- **Electrical fanout (h)**
  - C_{out} / C_{in} ratio (gate cap only, diffusion counts in the *p* term)
- **Parasitic effort (p)**
  - Ratio of parasitic capacitances for **equal R_{on}**

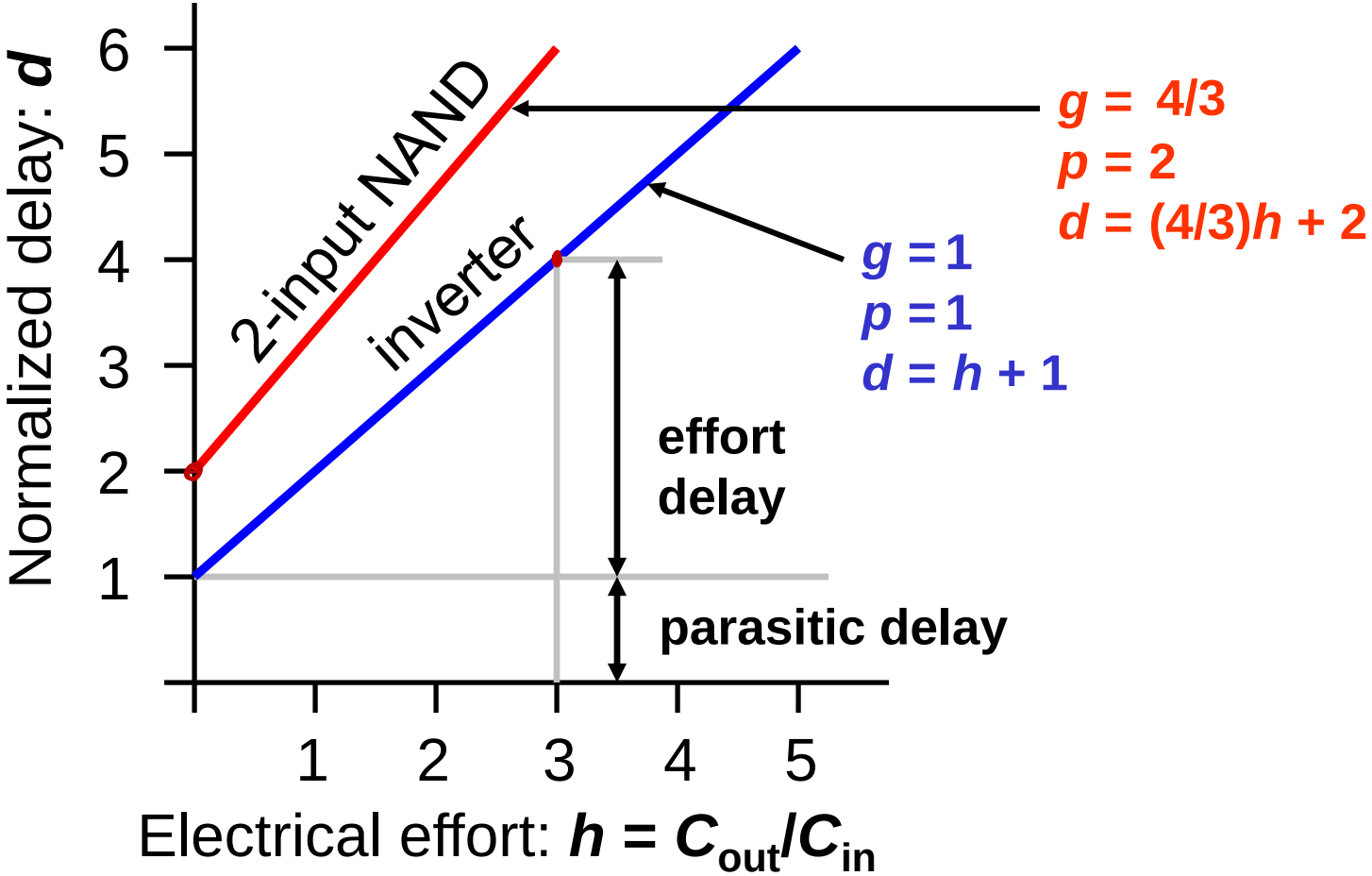
Calibrating the Model

- **The values for g and p can be extracted from simulation**
  - Because, d = g*h+p*γ
  - Simulating the delay of the gate for different loads
    - Drive itself with different multiplication factor
  - Extract τ using inverter with no self-loading (AS, AD, PS, PD = 0)
  - Vary the inputs (and rise/fall) for different g and p



Typical Simulation Data (*)

(*) assumes $\gamma = 1$



$d_{gate} = g \cdot h + p \cdot \gamma$ = effort delay + parasitic delay

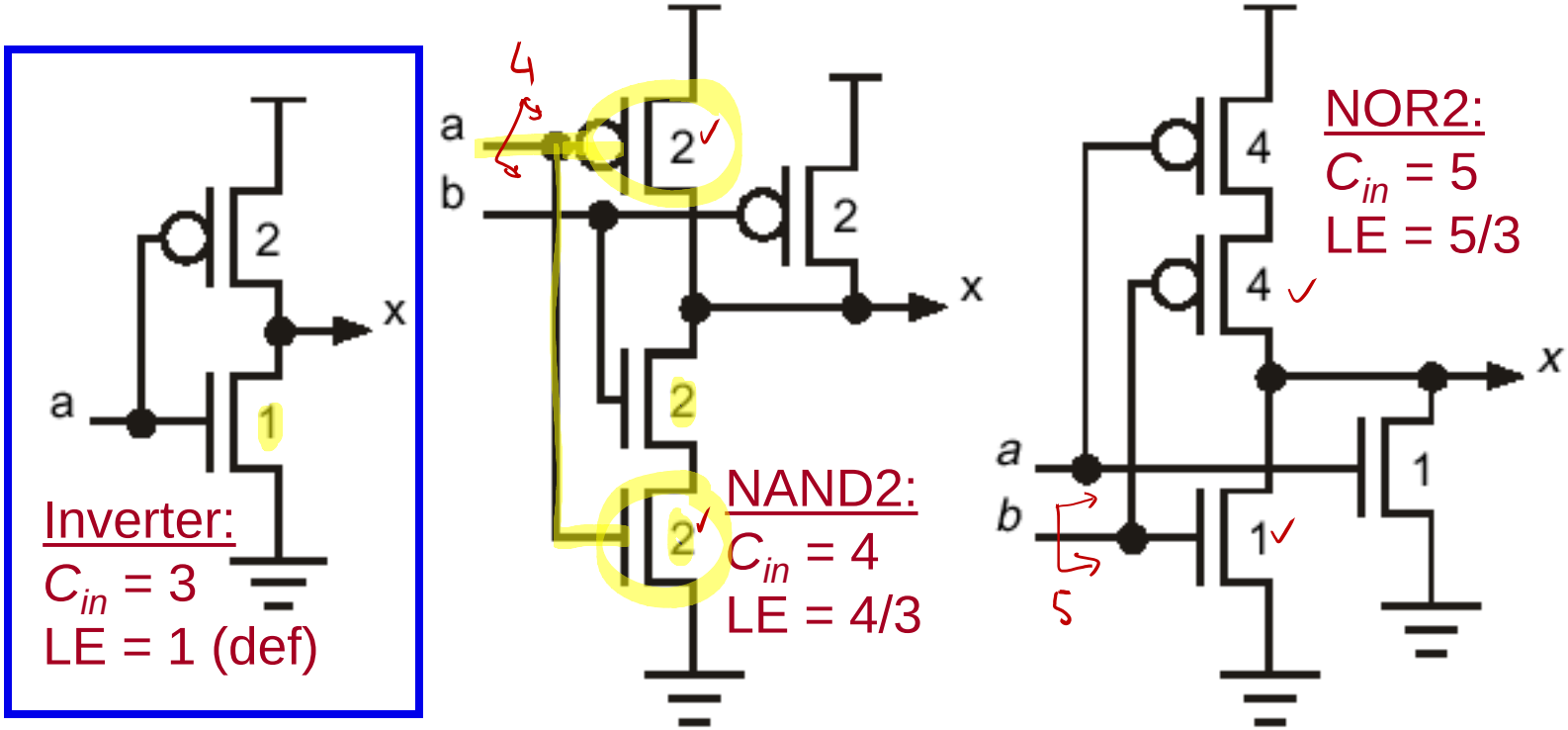
Computing Logical Effort: g

- ***g* is an unitless inherent characteristic of the gate**
  - Not a function of size of the gate
  - It is a function of the construction of the gate (connection and relative size between transistors)
  - An indication of the "cost" of implementing the function.

- **Procedure:**
  1. Choose an input.
  2. Find total device width driven by that input.
  3. Find W_P, the pull-up device width of a single device that has equivalent drive strength as a gate's pull-up of that input.
  4. For a reference inverter with <u>Equal Rise/Fall</u>, $\beta=\mu$, with W_P from Step 3, determine the total gate widths of the inverter devices.
  5. Divide Step 2 by Step 4 to determine g_{up}.
  6. Repeat Steps 3-5 for pull-down device for g_{down}.
     - The two g's would only be different if β of gate is not μ.

Calculating Logical Effort – Simple Example

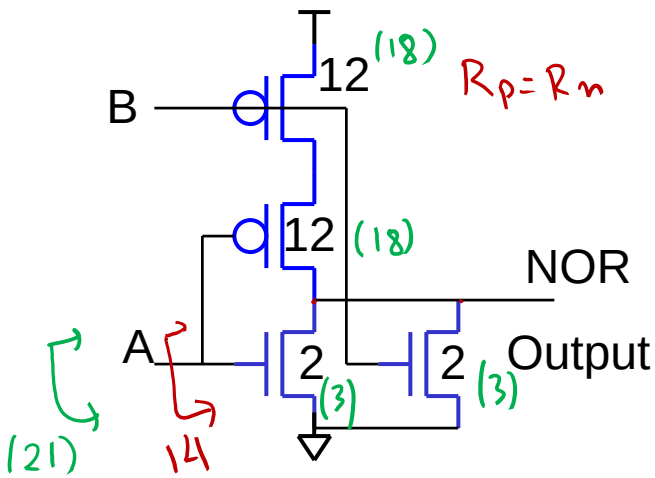
DEF: **Logical effort is the ratio of the <u>input capacitance</u> to the input capacitance of an inverter delivering the same output current**

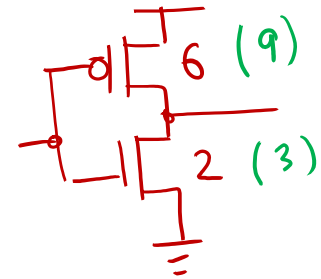


Reference

Another Example

- **Common assumptions**
  - C_{gate} proportional to Device Width
  - R_{gate} inversely proportional to Device Width.
- **For a NOR gate**
  - $\beta = \mu = 3$
  - Units are not so important



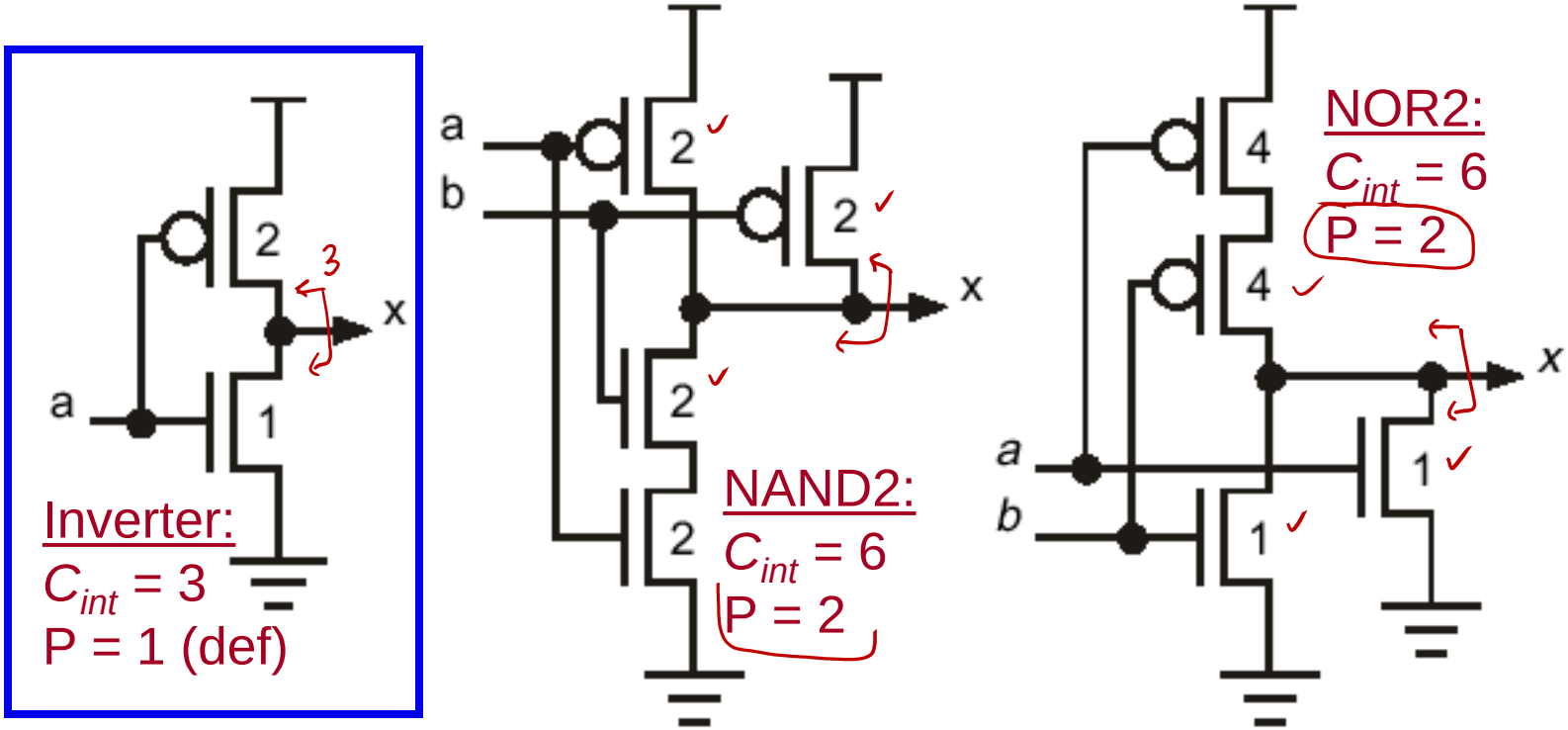


- **Equivalent inverter**
  - $W_P:W_N = 6:2$
  - $C_{G\_INV} = 8$
- **NOR gate input capacitance**
  - $C_{G\_NOR} = 14$
  - Logical Effort = 7/4 $\quad \frac{21}{12} = \frac{7}{4}$

Caveat: don't get confused with absolute transistor sizing!

Calculating Parasitic Effort – Simple Example

DEF: **Parasitic delay is the ratio of <u>intrinsic capacitance</u> at the output and intrinsic capacitance at the output of an equivalent inverter**

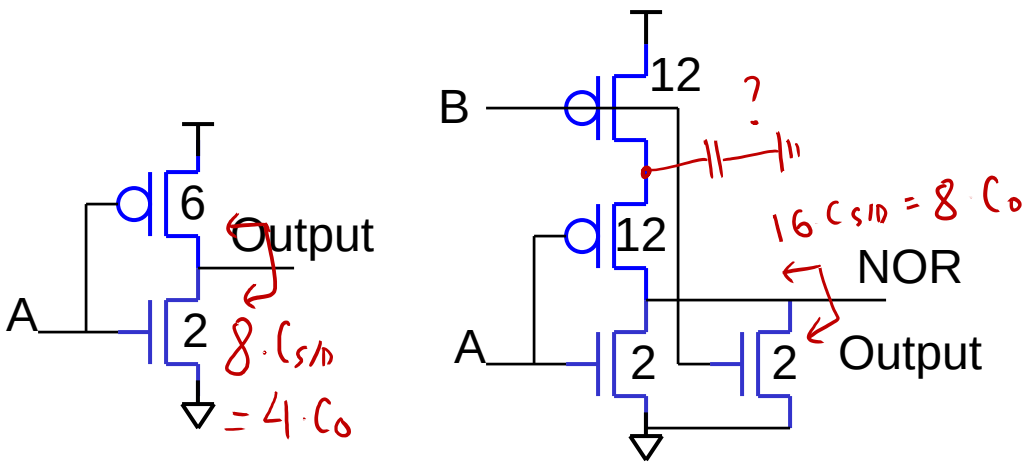


Reference

Calculating Parasitic Effort – Another Example

- **Typically given since it depends on $C_{diffusion}$ of a gate**

- **Example: assume $C_{S/D}=0.5C_G=0.5C_o$** $(\gamma = 0.5)$
  - For an inverter $C_{self}/C_{inv} = p_{INV} = 0.5$ $= 1 \cdot \gamma$
    - Higher $C_{S/D}/C_G$ results in larger p (penalizing delay more).
    - $C_{S/D}/C_G$ is often close to 1



$p_{NOR} = \frac{8}{4} = 2$

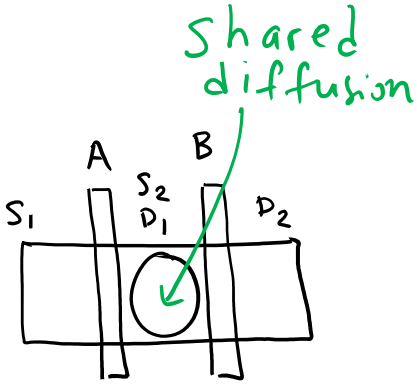
$p_{INV} = 1 \cdot \gamma$

Caveat: γ is part of p_{INV} $(\gamma \neq 1)$

Calculating p Including Series Stacking

- **What about the intermediate nodes?**
  - One way to account for them is to use an "effective" *p*.
  - For example: NOR pull up of B input
    - $R_{NOR} = 2*R_{PMOS}$
    - Delay = $(R_{NOR}/2)*C_1 + R_{NOR}*C_2 + R_{NOR}*C_{load}$
    - $p_{BUP} = [(R_{NOR}/2)*C_1 + R_{NOR}*C_2]/(R_{inv}*C_{inv})$ (where $R_{inv}=R_{gate}$)
    - $p_{BUP} = (C_1/2 + C_2)/C_{inv}$

Self-loading

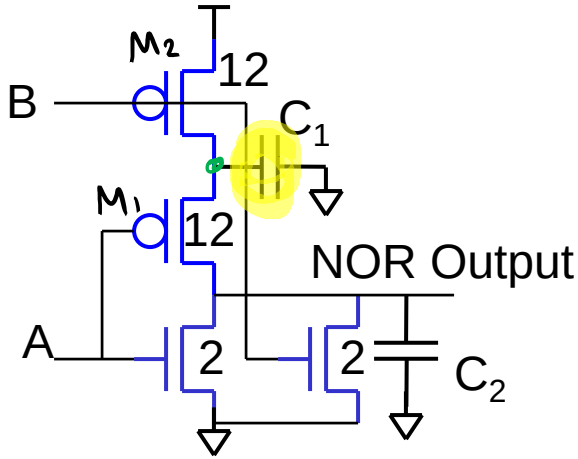


Using $C_{S/D} = 0.5C_G$

$C_1 = 6C_o$ (shared)

$C_2 = 8C_o$

$p_{BUP} = 11/4$



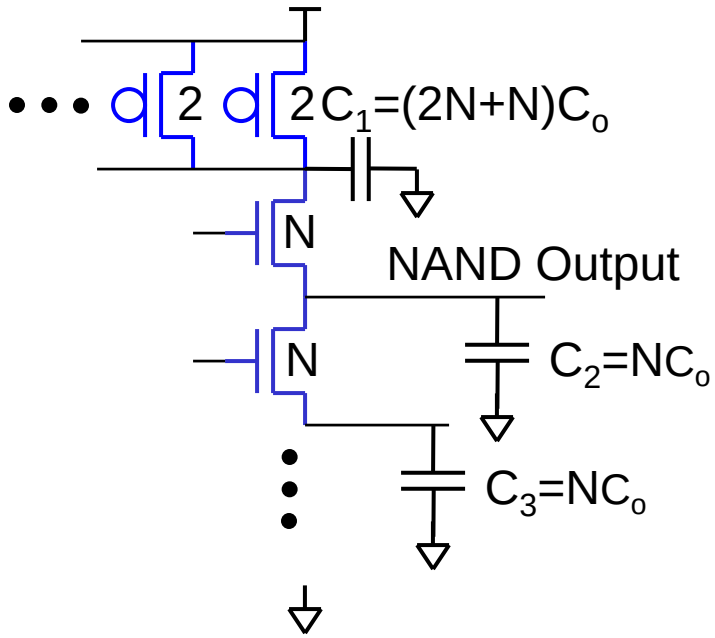
Note: *this increased accuracy requires different p's for different input AND pull up/down.*

EE115C model: *simplify by ignoring these nodes*

Generalize N-input NAND

N-NOR: $C_{in} \sim 2N + 1$

N-NAND: $C_{in} \sim 2 + N$



N-NOR: $C_{par} \sim 3N$

N-NAND: $C_{par} \sim 3N$

- **Output load = 3N**
  - N size-2 PMOS=2N
  - 1 size-N NMOS = N
- **Intermediate load = N (shared)**
- **Total pull down delay**
  - $T = R(3NC_o) + \text{sum}(i=1\text{-}N\text{-}1)\{(iR/N)*NC_o\}$
  - $d\ (\text{norm}) = 3N + (N^2/2 - N/2)$
  - $p = (N^2/2 - N/2)$
  - Proportional to N^2!!!
- **This is bad news for long stacks**
  - Even worse for PMOS (NOR)
  - Reality is even worse since C_{GS} makes each intermediate node capacitance > NC_o

A Catalog of Gates

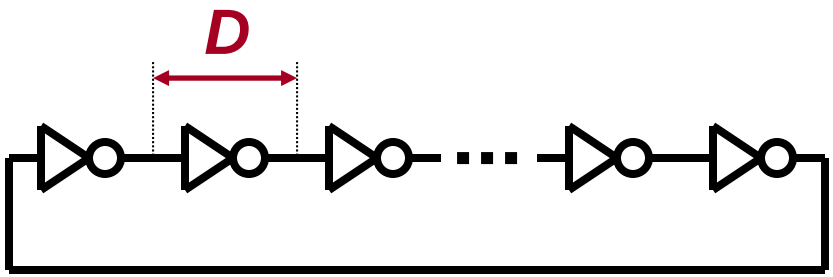
Gate Type	g for Different number of inputs					
	1	2	3	4	5	n
Inverter	1					
NAND		4/3	5/3	6/3	7/3	(n+2)/3
NOR		5/3	7/3	9/3	11/3	(2n+1)/3
Multiplexer		2	2	2	2	2
XOR,XNOR		4	12	32		

Gate Type	Parasitic delay
Inverter	p_{inv}
n-input NAND	np_{inv}
n-input NOR	np_{inv}
n-way Multiplexer	$2np_{inv}$
2-input XOR,XNOR (sym)	$n2^{n-1}p_{inv}$

- $\beta = \mu = 2$
- Mux is tri-state inverters shorted together.
- XOR assumes that input is bundled (a,a')
- $p_{INV} \sim 1$
- p_{GATE} in this table does not include intermediate nodes.

Example #1: Ring Oscillator

Estimate the frequency of an *N*-stage ring oscillator:



Logical Effort: $g = 1$

Electrical Effort: $h = C_{out}/C_{in} = 1$

Parasitic Delay: $p = p_{inv} = 1$

Stage Delay: $d = g \cdot h + p = 2$

OSC Frequency: $f_{osc} = \dfrac{1}{2N\underbrace{d\tau}} = \dfrac{1}{4N\tau}$

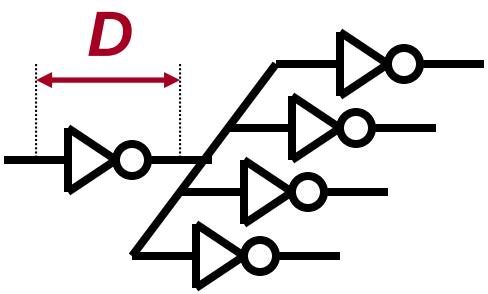
gpdk090:

$t_{stage} = 13ps\ (TT)$

$d\tau$

Example #2: Fanout-of-4 (FO4) Inverter

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



Logical Effort: $g = 1$

Electrical Effort: $h = C_{out}/C_{in} = 4$

Parasitic Delay: $p = p_{inv} = 1$

Stage Delay: $d = g \cdot h + p = 5$

gpdk090:

$t_{FO4} = 33ps\ (TT)$

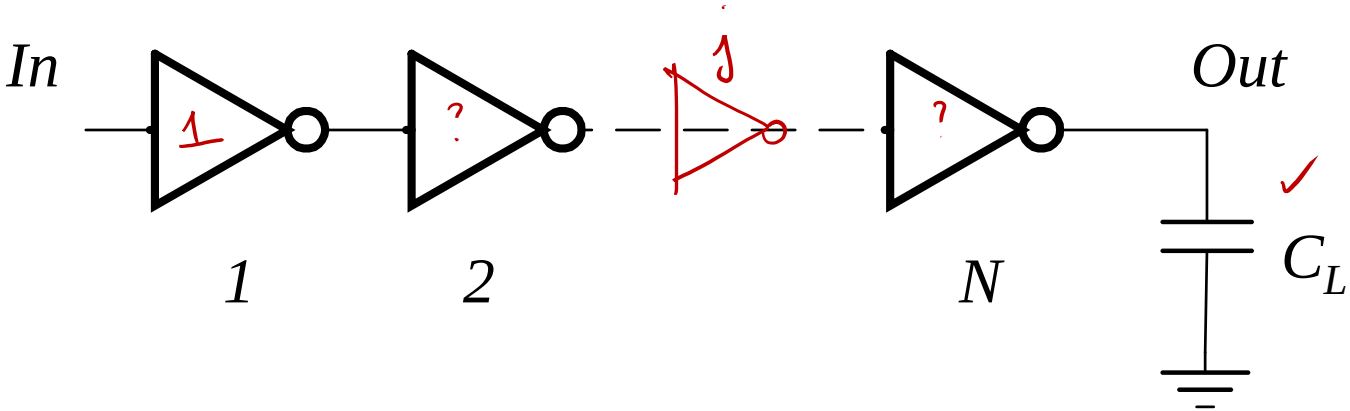
Summary

- **Delay and/or power of a logic network depend significantly on the relative sizes of logic gates (not transistors within a gate)**
- **Inverter buffering is a simple example of the analysis**
  - The analysis leads to ~FO4 as being optimal fanout for driving larger **C** loads
- **To generalize analysis of delay, we introduce logical effort**
  - Delay normalized by inverter delay, $\mathbf{d = gh + p\gamma}$
  - g and p are characteristics of a logic gate that depends on its structure and does not depend on gate size.
    - **May have different g's and p's** for different inputs and pull-up / pull-down
    - Simplify by using g_{AVG} and ignoring C's of intermediate nodes
  - Once a table of g's and p's are created for the catalog of gates, delay can be calculated quickly and easily
- **Next, we will look at how to size a network instead of just analyzing it**

Week 4 Agenda

- The Concept of Logical Effort
- <u>Buffer Chain Example</u>
- Multi-Stage Logic Networks

Logical Effort Applied to Inverter Chain



$$t_p = t_{p1} + t_{p2} + \ldots + t_{pN}$$

$$t_{pj} \sim R_{gate,unit} C_{in,gate,unit} \left(\gamma + \frac{C_{gin,j+1}}{C_{gin,j}} \right)$$

h_j

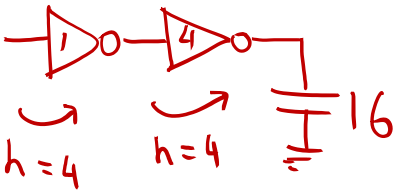
$$t_p = \sum_{j=1}^{N} t_{p,j} = \tau \sum_{i=1}^{N} \left(\gamma + \frac{C_{gin,j+1}}{C_{gin,j}} \right), \quad C_{gin,N+1} = C_L$$

Optimal Tapering for Given *N*

- **Delay equation has N-1 unknowns, $C_{gin,2} \ldots C_{gin,N}$**
- Minimize the delay, **find N-1 partial derivatives**
- **Result:** $\boxed{C_{gin,j+1} / C_{gin,j} = C_{gin,j} / C_{gin,j-1}}$
- **Size of each stage is the geometric mean of two neighbors:**

$$\boxed{C_{gin,j} = \sqrt{C_{gin,j-1} \cdot C_{gin,j+1}}}$$

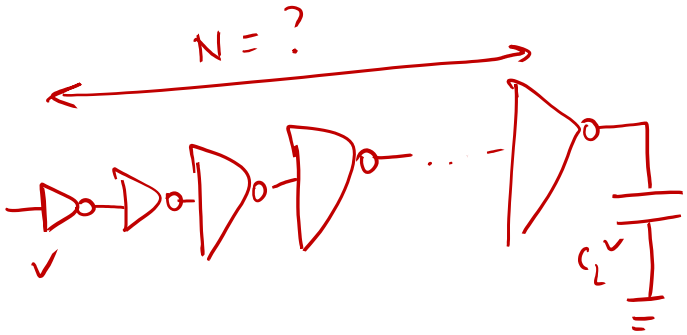
  - Each stage has the same fanout, C_{out} / C_{in}
  - Each stage has the same delay



Optimum Delay and Number of Stages

- When each stage is sized by f and has same fanout f:

$$f^N = F = C_L / C_{gin,1}$$



- Effective fanout of each stage:

$$\boxed{f = \sqrt[N]{F}}$$

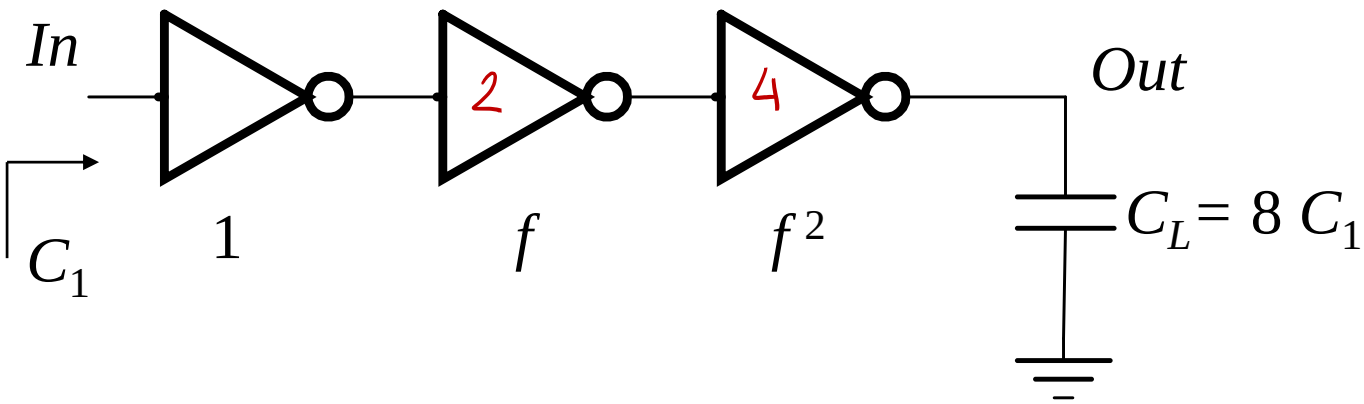
$gh = f$ (stage effort)

- Minimum path delay:

$$\boxed{t_p = N\tau\left(\gamma + \sqrt[N]{F}\right)}$$

Toy Example: Gate Sizing

given $C_1 = 1$ & $C_L = 8 \cdot C_1$



C_L/C_1 has to be evenly distributed across $N = 3$ stages:

$$f = \sqrt[3]{8} = 2$$

$D = 3 \times (2 + 8)$

Next Question: Optimum Number of Stages?

- **For a given C_L and given C_{in} find optimal sizing f**

$$C_L = F \cdot C_{in} = f^N C_{in} \quad \text{with} \quad \boxed{N = \frac{\ln F}{\ln f}}$$

$$t_p = N\tau\left(F^{1/N} + \gamma\right) = \frac{t_{p0} \ln F}{\gamma}\left(\frac{f}{\ln f} + \frac{\gamma}{\ln f}\right)$$

$$\frac{\partial t_p}{\partial f} = \frac{t_{p0} \ln F}{\gamma} \cdot \frac{\ln f - 1 - \gamma / f}{\ln^2 f} = 0$$

$$\boxed{\text{For } \gamma = 0,\ f = \mathrm{e},\ N = \ln F}$$

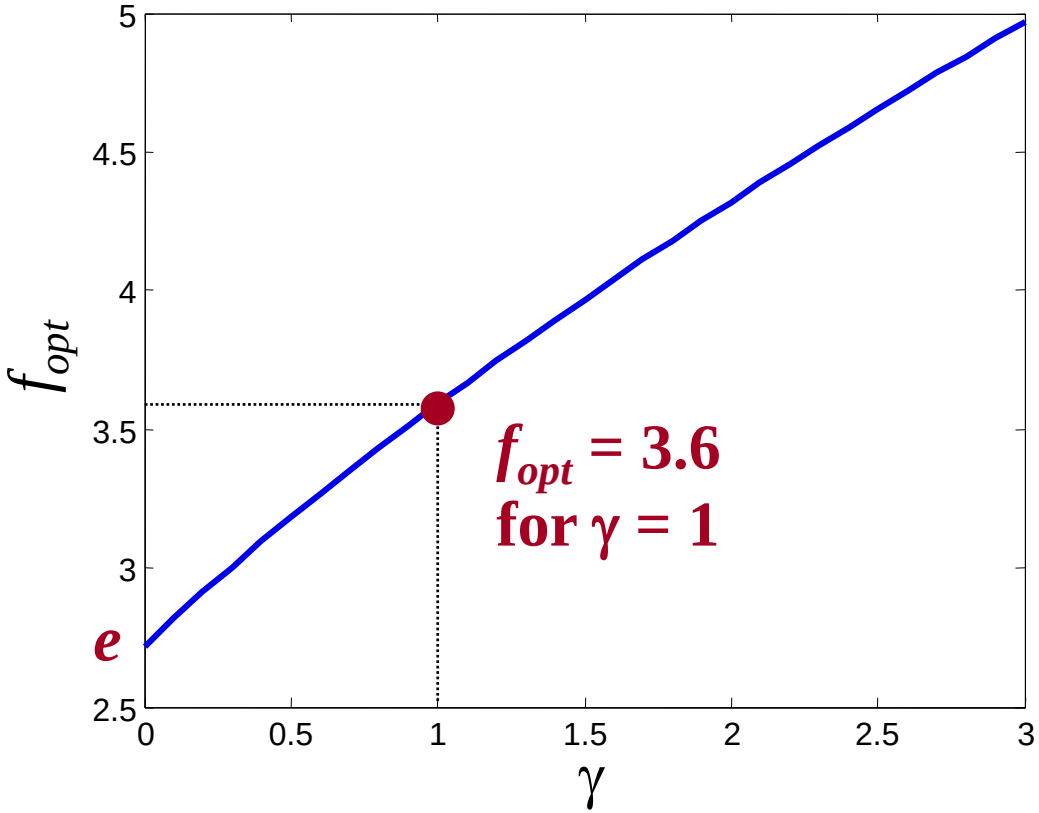
2.7 ≈ 3

$$\boxed{f = \exp\left(1 + \gamma / f\right)}$$

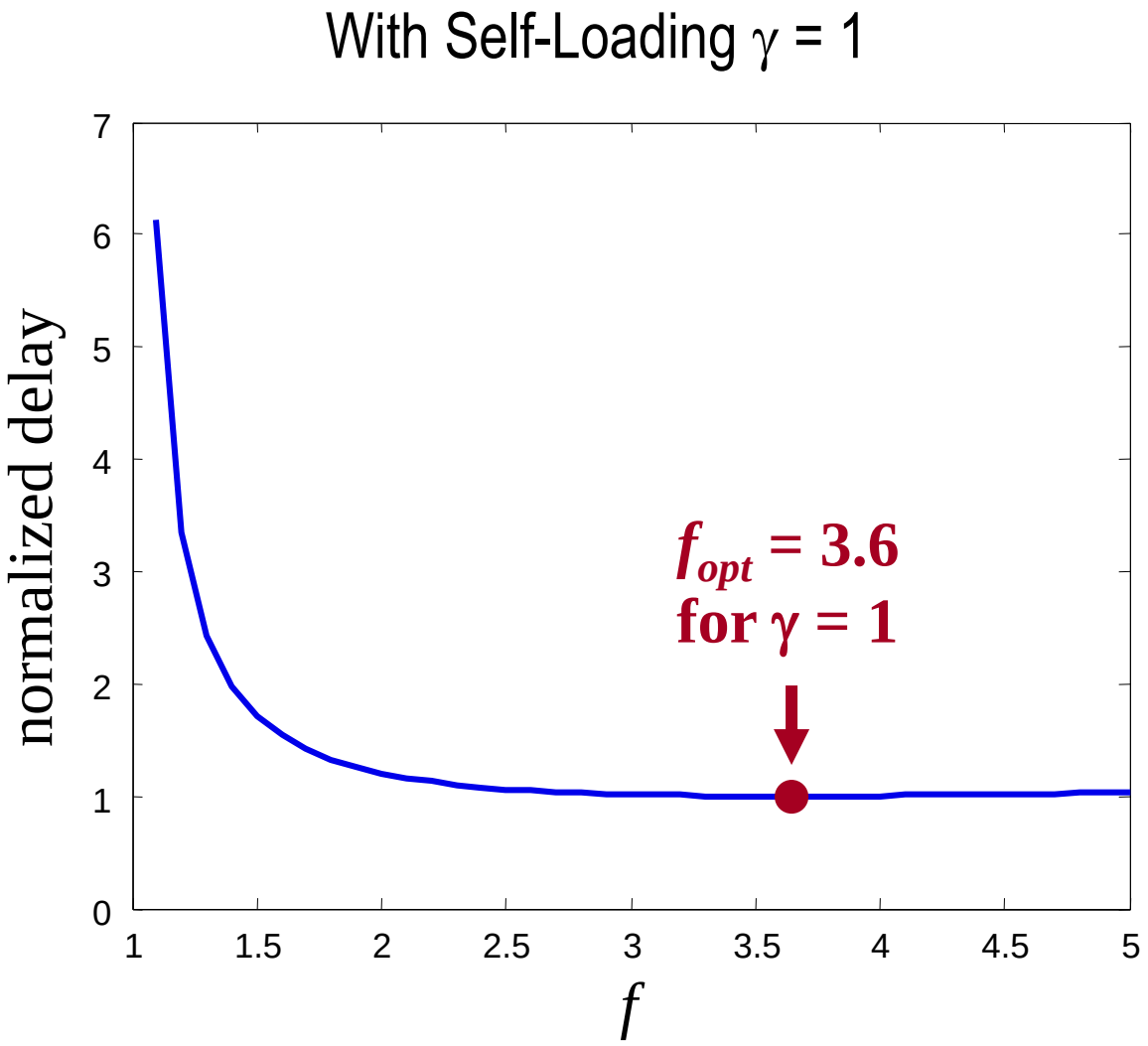
Optimum Effective Fanout *f*

- **Optimum *f* for given process defined by γ**

$$f = \exp(1 + \gamma / f)$$



Impact of Self-Loading on t_p



Normalized Delay Function of *F*

$$t_p = N t_{p0}\left(1 + \sqrt[N]{F} / \gamma\right) \qquad (\gamma = 1)$$

F	Unbuffered	Two Stage	Inverter Chain
10	11	8.3	8.3
100	101	22	16.5
1000	1001	65	24.8
10,000	10,001	202	33.1

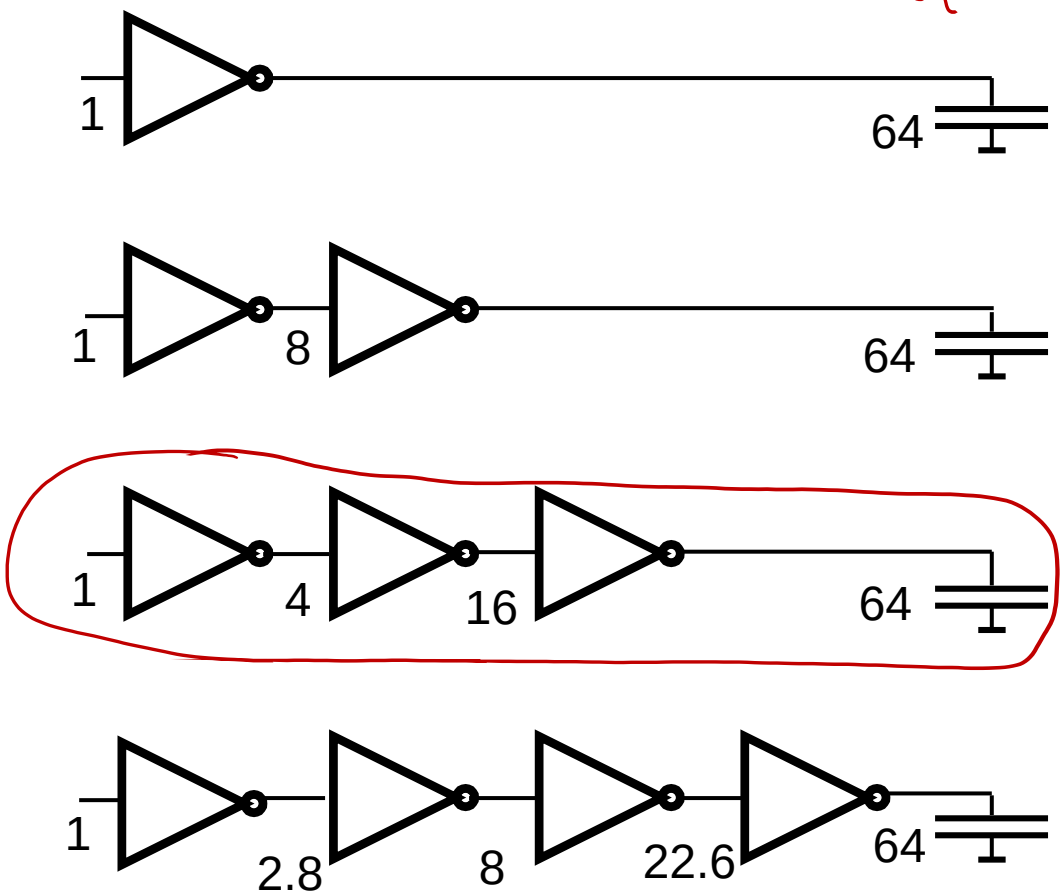
Textbook: page 210

$f^* \approx 4$

Example: Buffer Design

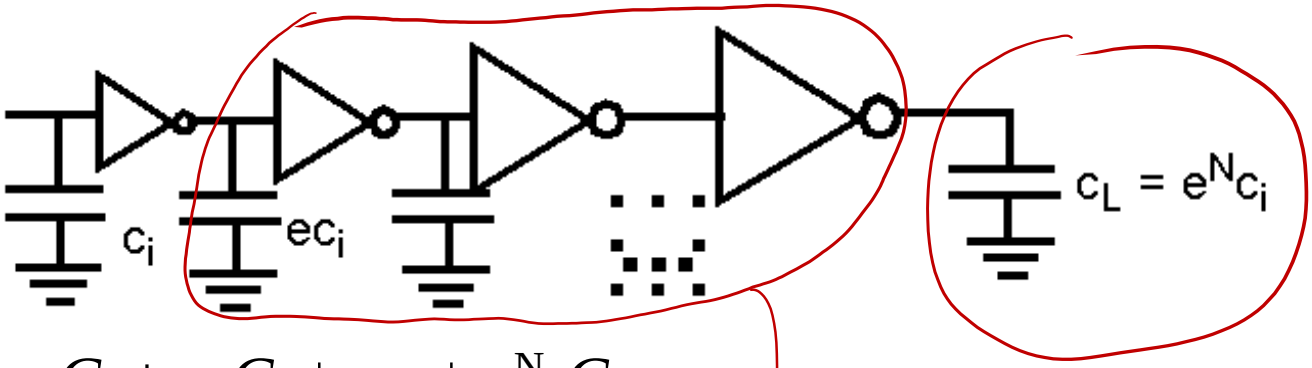
$C_{in,1} = 1$

$C_L = 64$



N	f	t_p
1	64	65
2	8	18
3	4	15
4	2.8	15.3

What About Power Consumption (and Area?)



$$C_{tot} = C_i + e \cdot C_i + \ldots + e^N \cdot C_i$$
$$= C_i \cdot (1 + e + \ldots + e^N)$$
$$= C_i + C_i \cdot e^N + C_i \cdot e \cdot (1 + e + \ldots + e^{N-2})$$

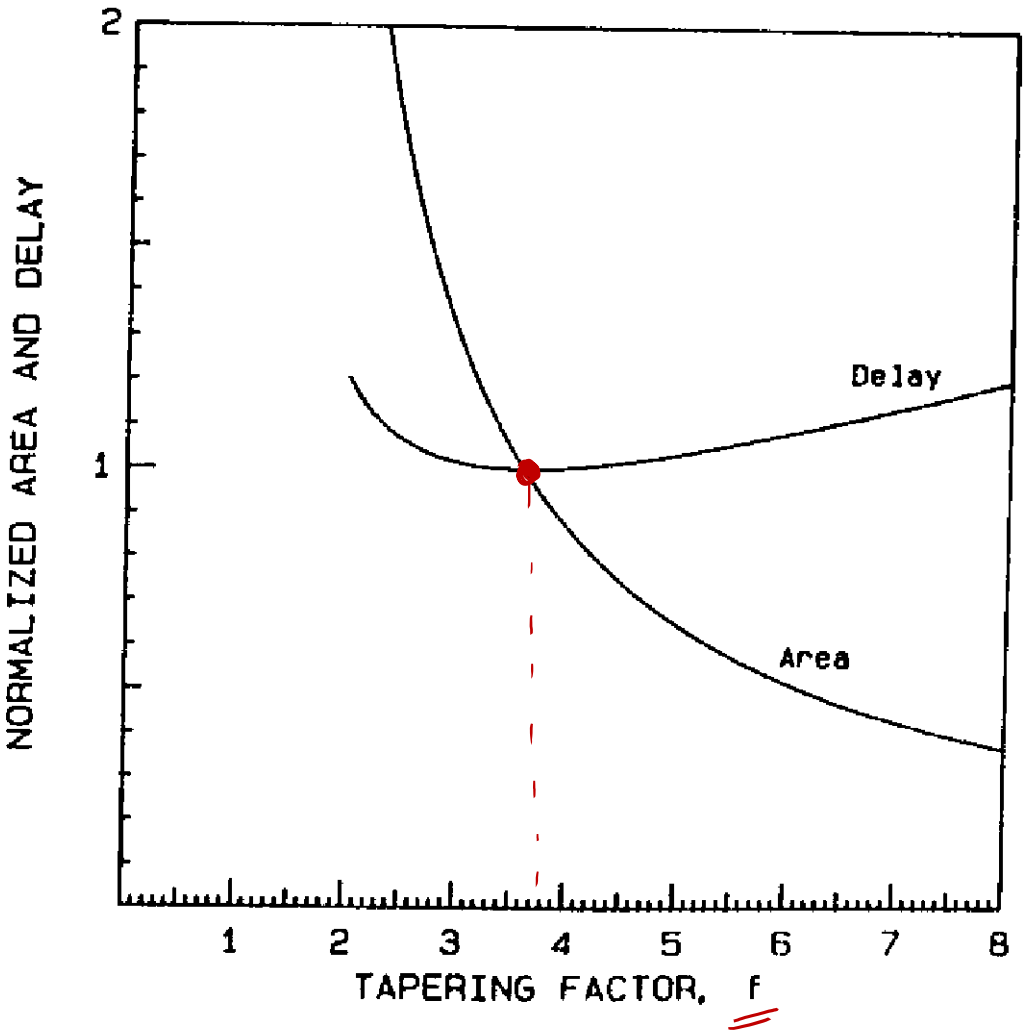
Overhead !!! $(e^{N-1}-1) / (e-1)$

Example: C_L = 20pF; C_i = 50fF $\rightarrow$ N = 6

Fixed: 20pF

Overhead: 11.66pF !!!

Delay vs. Area and Power

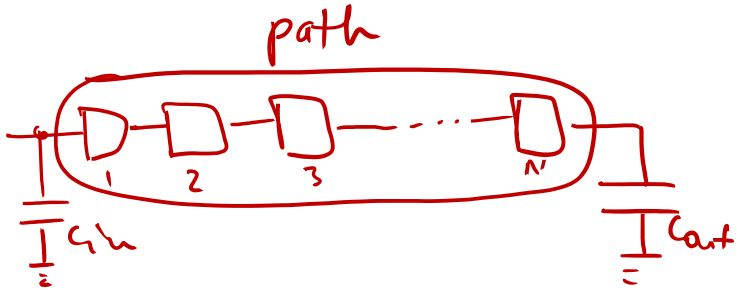


Week 4 Agenda

- The Concept of Logical Effort
- Buffer Chain Example
- <u>Multi-Stage Logic Networks</u>

Generalization for Multistage Networks

$$Delay = \sum_{i=1}^{N} (p_i + g_i \cdot h_i)$$



Stage effort: $f_i = g_i \cdot h_i$

Path electrical effort: $H_{path} = C_{out}/C_{in}$

Path logical effort: $G_{path} = g_1 g_2 \ldots g_N$

Branching effort: $B_{path} = b_1 b_2 \ldots b_N$

Path effort = $G_{path} \cdot H_{path} \cdot B_{path}$

Forget this for now

Path delay $D = \Sigma D_i = \Sigma P_i + \Sigma G_i \cdot H_i$

Optimum Effort per Stage

When each stage bears the same effort:

$$f^N = \prod g \cdot h = PathEffort$$

$$f = \sqrt[N]{PathEffort}$$

Fanout of each stage: $h_i = f/g_i$

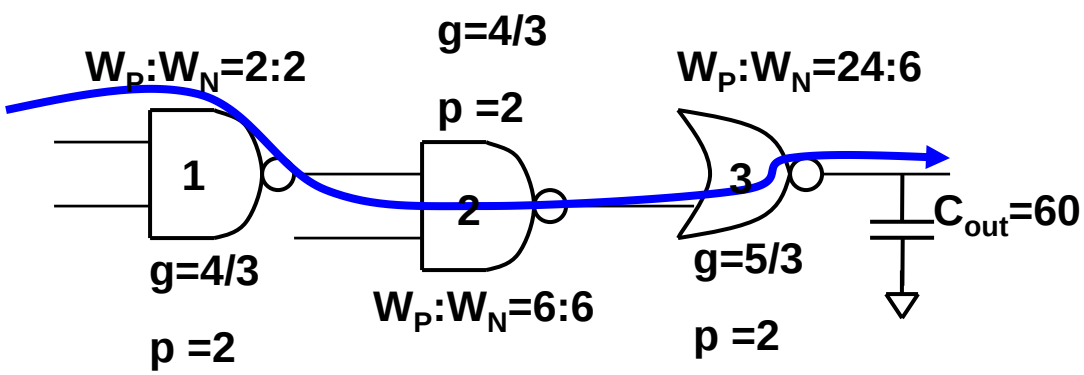
Complex gates should drive smaller load!!!

Minimum path delay

$$D_{min} = \sum (g_i \cdot h_i + p_i) = \underbrace{N \cdot f + P}$$

Example: A Random Multi-stage Network

- **Delay of a multi-stage network = sum of stage delays**
  - Path Effort Delay $D_F = \sum_i f_i$
  - Path Parasitic Delay $P = \sum_i p_i$
  - Total Path Delay $\boxed{D = \sum_i d_i = D_F + P}$



$f_1 = 4$

$f_2 = 3.33$

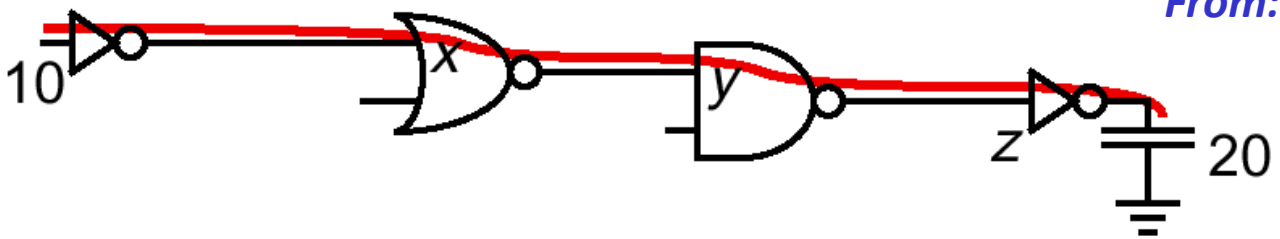
$f_3 = 3.33$

$D_F = 10.66$, $P = 6$

$D = 16.66$ (τ delays)

Gate Sizing Example

From: *David Harris*



$g_1 = 1$ $\quad$ $g_2 = 5/3$ $\quad$ $g_3 = 4/3$ $\quad$ $g_4 = 1$

$h_1 = x/10$ $\quad$ $h_2 = y/x$ $\quad$ $h_3 = z/y$ $\quad$ $h_4 = 20/z$

- **First Compute Path Effort**

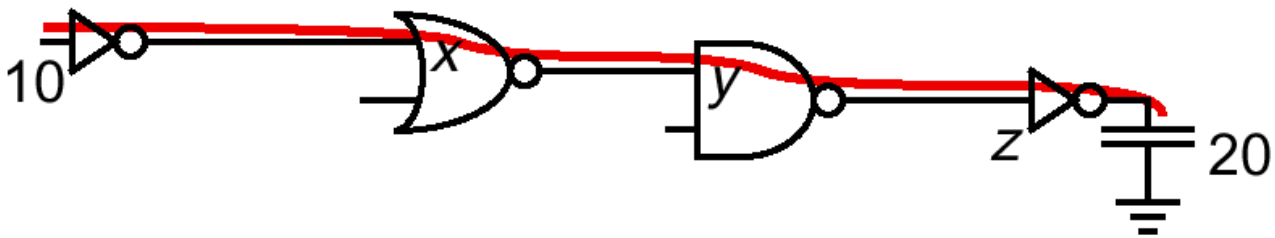
Path Effort = $F = \Pi g \cdot h$

$$= 1(\frac{x}{10}) \times \frac{5}{3}(\frac{y}{x}) \times \frac{4}{3}(\frac{z}{y}) \times 1(\frac{20}{z}) = \frac{40}{9}$$

- **The optimal stage effort is:**

$$f^* = g \cdot h = (\frac{40}{9})^{1/4} = 1.45$$

Gate Sizing Example (Cont.)



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

$$C_{in} = g \cdot \frac{C_{out}}{f^*}$$

- **We can now size the gates:**

$$z = 1 \cdot \frac{20}{1.45} = 13.8 \qquad x = \frac{5}{3} \cdot \frac{y}{1.45} = 14.5$$

$$y = \frac{4}{3} \cdot \frac{z}{1.45} = 12.7 \qquad C_{in} = 1 \cdot \frac{x}{1.45} = 10$$

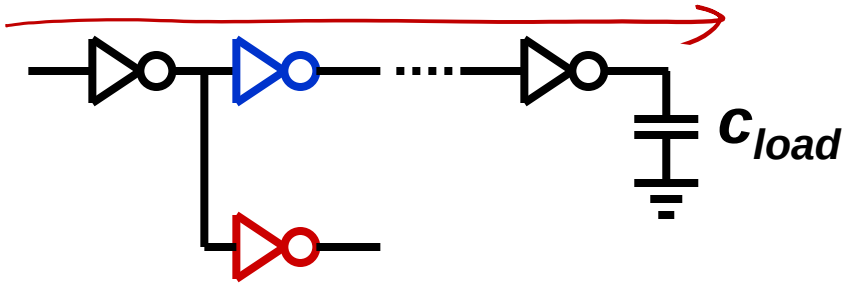
- **The total normalized delay is (assuming ~~P_{inv}=1~~ $\gamma = 1$):**

$$D = 4SE^* + \sum P = \underbrace{4 \cdot 1.45}_{} + \underbrace{(1+2+2+1)}_{\times \gamma} = 11.8$$

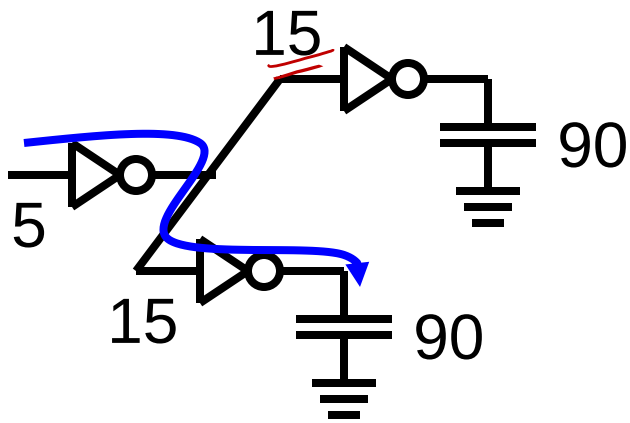
Add Branching Effort

Branching effort:

$$b = \frac{C_{on\text{-}path} + C_{off\text{-}path}}{C_{on\text{-}path}}$$



Branching Example #1



g	=	1
h	=	90/5 = 18
F	=	18 (wrong!)
f_1	=	(15+15)/5 = 6
f_2	=	90/15 = 6
F	=	36, not 18!

Introduce new kind of effort to account for branching:

- **Branching Effort**: $b = \dfrac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}} = \dfrac{15+15}{15} = 2$
- **Path Branching Effort**: $B = \prod b_i$

Now we can compute the path effort:

- **Path Effort**: $H = \prod g \cdot h \cdot b$

Multistage Networks with Branching

General Logical Effort formulation

$$Delay = \sum_{i=1}^{N} (p_i + g_i \cdot h_i)$$

Stage effort: $f_i = g_i \cdot b_i \cdot h_i$

Path electrical effort: $H_{path} = C_{out}/C_{in}$

Path logical effort: $G_{path} = g_1 g_2 \ldots g_N$

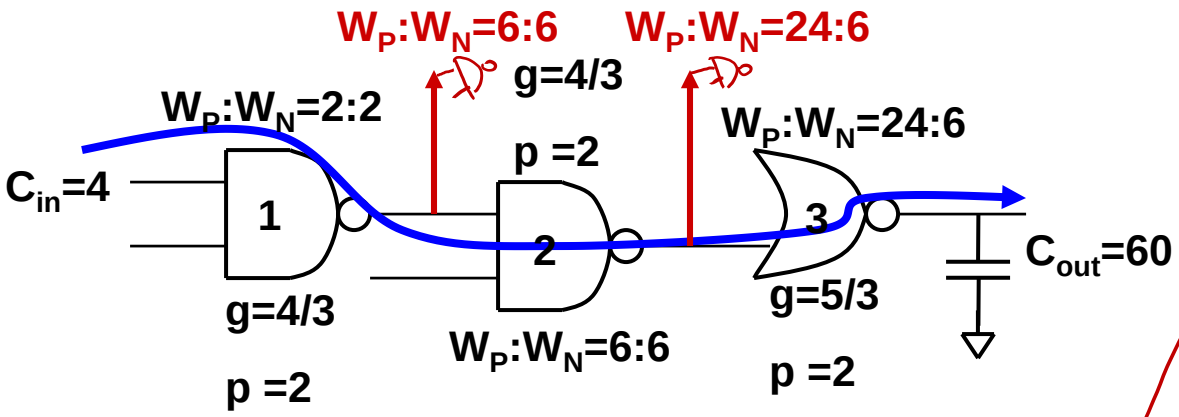
Branching effort: $B_{path} = b_1 b_2 \ldots b_N$

Path effort = $F_{path} = G_{path} \cdot H_{path} \cdot B_{path}$

Branching

Path delay $D = \Sigma D_i = \Sigma P_i + \Sigma G_i \cdot H_i$

Branching Example #2



branching	w/o bran.
f_1 = 8	f_1 = 4
f_2 = 6.66	f_2 = 3.33
f_3 = 3.33	f_3 = 3.33

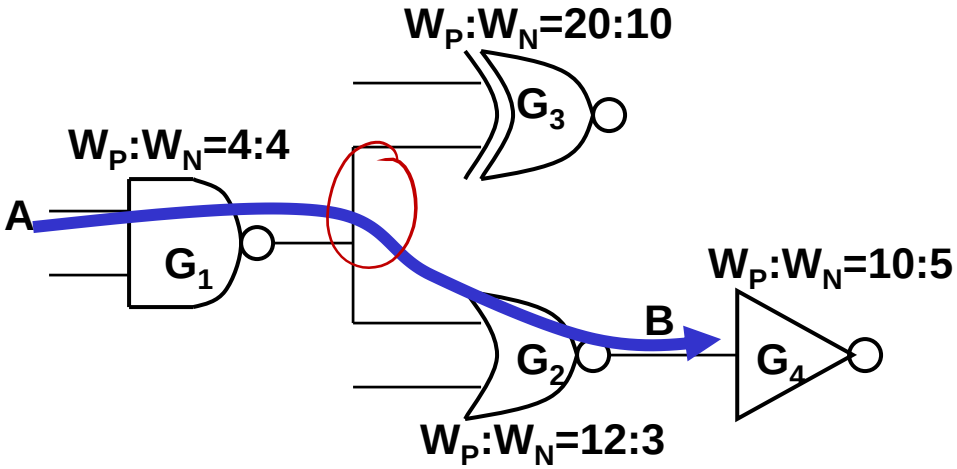
$b_1 = 2$
$b_2 = 2$

$B = 4$

- **For circuits with branching:**
  - G_{PATH} is the same = 3
  - H_{PATH} is the same = 15
  - F_{PATH} differs
    - h_1 = 24/4= 6, h_2 = 60/12=5, h_3=2
    - F_{PATH} = 4/3 * 6 * 4/3 * 5 * 5/3 * 2 = 177.8
- **F is no longer GH**
  - New F_{PATH} with branching: F_{PATH} = GBH

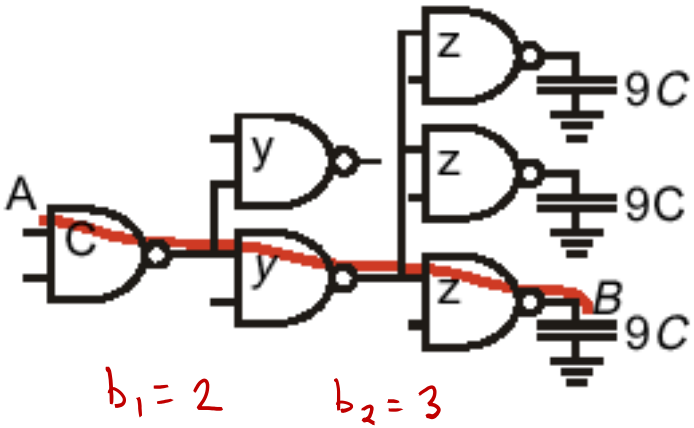
Branching Example #3: Gate Delays

- **Delay of the path from A to B where $\beta = \mu = 2$ and $p_{INV}=1$**
  - $g_{G1} = 4/3$, $p_{G1}=2$, $C_{IN\_G1}=8$
  - $g_{G2} = 5/3$, $p_{G2}=2$, $C_{IN\_G2}=15$
  - $g_{G3} = 4$, $p_{G3}=4$, $C_{IN\_G3}=30$
  - $C_{IN\_G4}=15$
  - $h_{G1} = (C_{IN\_G2}+C_{IN\_G3})/C_{IN\_G1} = 5.625$, $h_{G2} = C_{IN\_G4}/C_{IN\_G2} = 1$
  - $d_{G1} = g_{G1}h_{G1}+p_{G1} = 9.5$
  - $d_{G2} = g_{G2}h_{G2}+p_{G2} = 3.66$
  - Delay = 13.16
    - Normalized



Branching Example #4

Select gate sizes ***y*** and ***z*** to minimize delay from ***A*** to ***B***



Logical Effort: $G = (4/3)^3$

Electrical Effort: $H = C_{out}/C_{in} = 9$

Branching Effort: $B = 2 \cdot 3 = 6$

Path Effort: $PE = G \cdot H \cdot B = 128$

Best Stage Effort: $f = PE^{1/3} \approx 5$

Delay: $D = 3 \cdot 5 + 3 \cdot 2 = 21$

$\times \gamma$

Work backward for sizes:

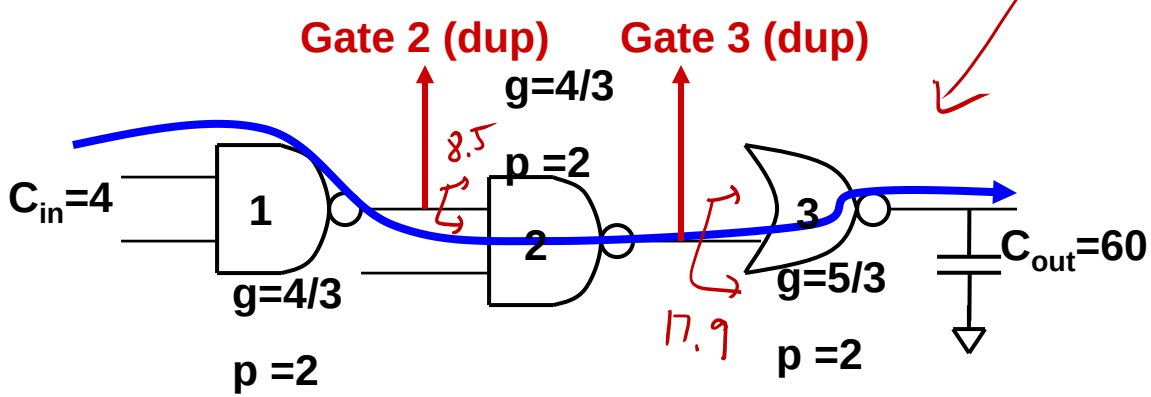
$$z = \frac{9C \cdot (4/3)}{5} = 2.4C$$

$$y = \frac{3z \cdot (4/3)}{5} = 1.9C$$

Sizing Example with Branching

- **Size the gates for optimum delay**
- **Once the path effort is determined, it is quite easy to determine the appropriate gate sizes**
  - Start from the output of a path
  - Work backwards to the input
    - Check your work if the input is the same as the specification
    - Assuming each unit W has capacitance of unit C

$$C_{in\_i} = \frac{C_{out\_i} g_i}{f_{opt}}$$



$B=b_1b_2=4$

$F = 177.8$, $f_{opt} = 5.62$

$C_{in3} = 60*5/3*1/5.6=17.9$

$C_{in2} = b_2*17.9*4/3*1/5.6=8.5$

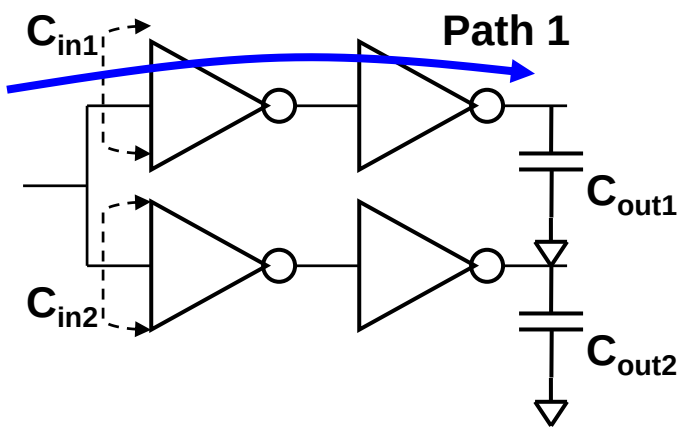
$C_{in1} = b_1*8.5*4/3*1/5.6=4$

$W_{P3} = 4/5*17.9 = 14.3$

$W_{P2} = ½*8.5 = 4.25$

Branches: Same # Stages, Different Loads

- **Example: two paths with the same number of stages but different loads**
  - Optimal system has all paths with equal delay
  - Branching for path 1, b = 1+α
    - Assumption is that $p_1 \sim p_2$



$$D_1 = D_2$$

$$N\sqrt[N]{F_1} + p_1 = N\sqrt[N]{F_2} + p_2$$

$$F_1 = \frac{B_1 G_1 C_{out1}}{C_{in1}} = F_2 = \frac{B_2 G_2 C_{out2}}{C_{in2}}$$

$$\boxed{\frac{C_{in2}}{C_{in1}} = \alpha = \frac{B_2 G_2 C_{out2}}{B_1 G_1 C_{out1}}}$$

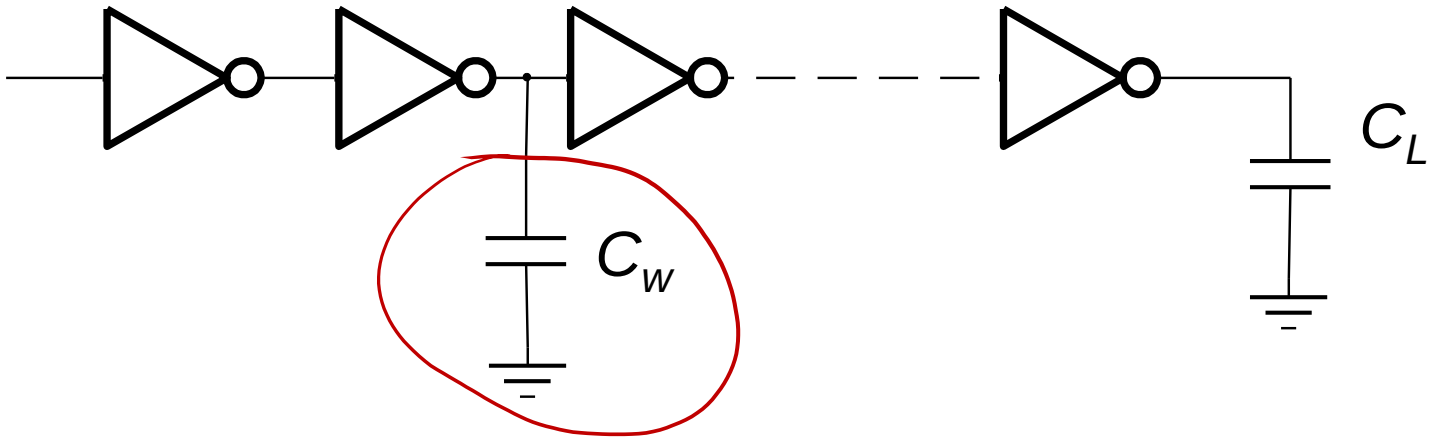
Summary: Logical Effort "Design Flow"

- **Compute the path effort: *Path Effort* = $\prod$ *LE*· *FO*· *B***
- **Find the best number of stages: $N^* \sim \log_4(PathEffort)$**
- **Compute the stage effort: $f^* = (PathEffort)^{1/N}$** ≈ 4
- **Working from either end, determine gate sizes:**

$$\boxed{C_{in} = g \cdot b \cdot \frac{C_{out}}{f^*}}$$

Reference: Sutherland, Sproull, Harris, "Logical Effort," (Morgan-Kaufmann 1999)

Handling Wires & Fixed Loads



$$Delay = \sum_{i=1}^{N} \left(p_i + g_i \cdot (h_i + \frac{C_{Wi+1}}{C_i}) \right)$$

Multi-level Logic: What is Best?

Assumes first stage has same Ron as unit inverter

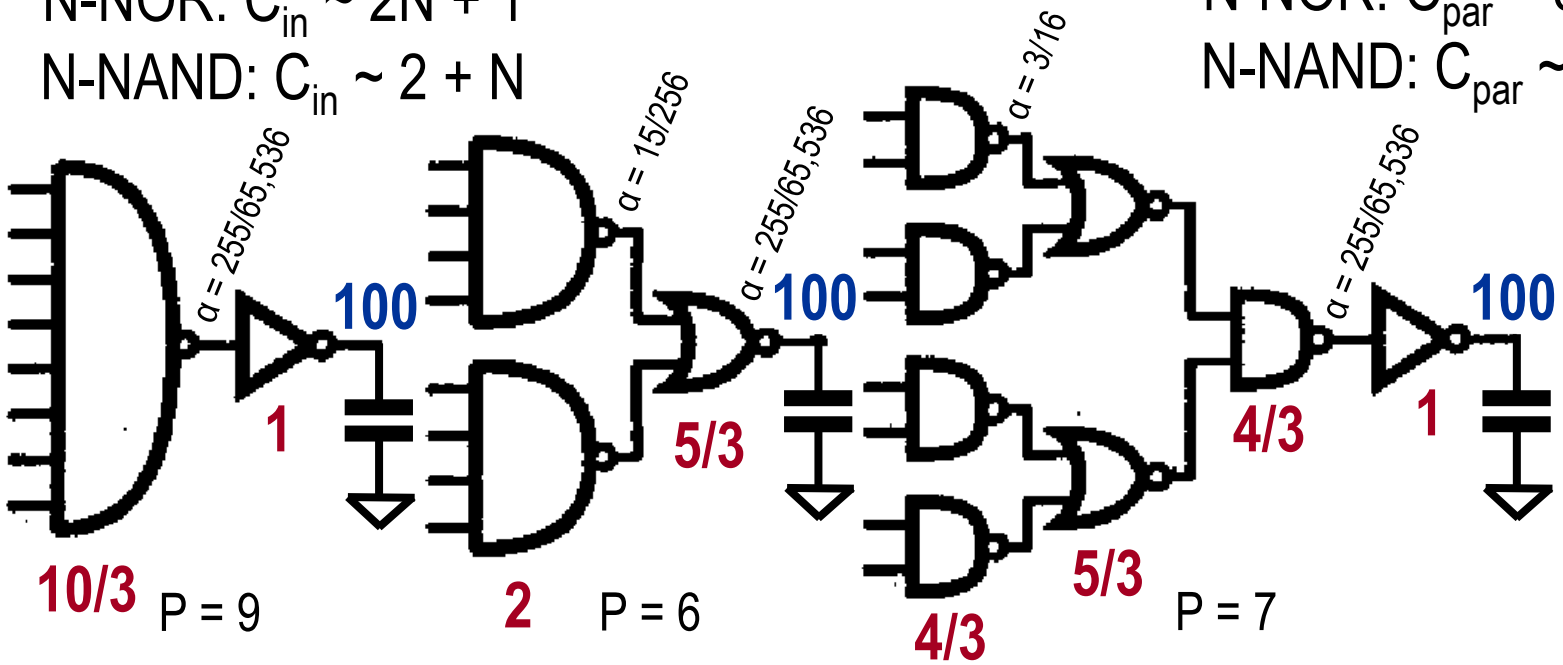
$\gamma = 0.5$

N-NOR: $C_{in} \sim 2N + 1$

N-NAND: $C_{in} \sim 2 + N$

N-NOR: $C_{par} \sim 3N$

N-NAND: $C_{par} \sim 3N$



P = 9

H = 100/10
G = 10/3 (3.33)
F = 100/3
$f^* = F^{1/2} = 5.77$

$D = 2f^* + P = 16.04$

P = 6

H = 100/6
G = 10/3 (3.33)
F = 500/9
$f^* = F^{1/2} = 7.45$

$D = 2f^* + P = 17.9$

P = 7

H = 100/4
G = 80/27 (2.96)
F = 2,000/27
$f^* = F^{1/4} = 2.93$

$D = 4f^* + P = 15.22$