

University of Southern California

Viterbi School of Engineering

EE577A
VLSI System Design

Logical Effort

**References: syllabus textbooks, Slides and notes from
Professors Gupta and Pedram, online resources**

Shahin Nazarian

Spring 2013

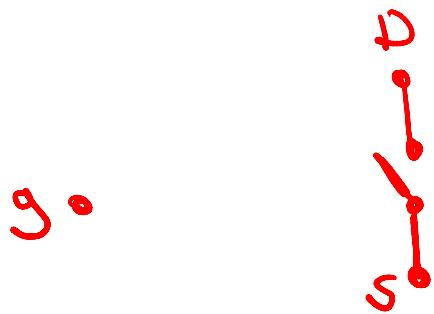
Delay Optimization Objectives

- Design to be faster
- Many ways to design
- Properties of a delay model for optimization purposes
 - Accuracy in design optimization
- Design selection and sizing

Building a Model

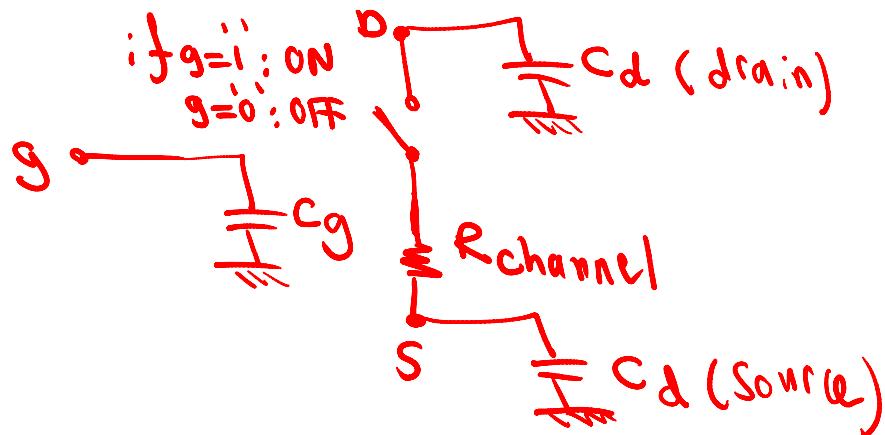
- Building a model for:
 - A transistor
 - A gate
 - A circuit
- Optimization

Transistor Level Model



Transistor Level Model (Cont.)

nMOS:



C_g

C_d

R

μ_n

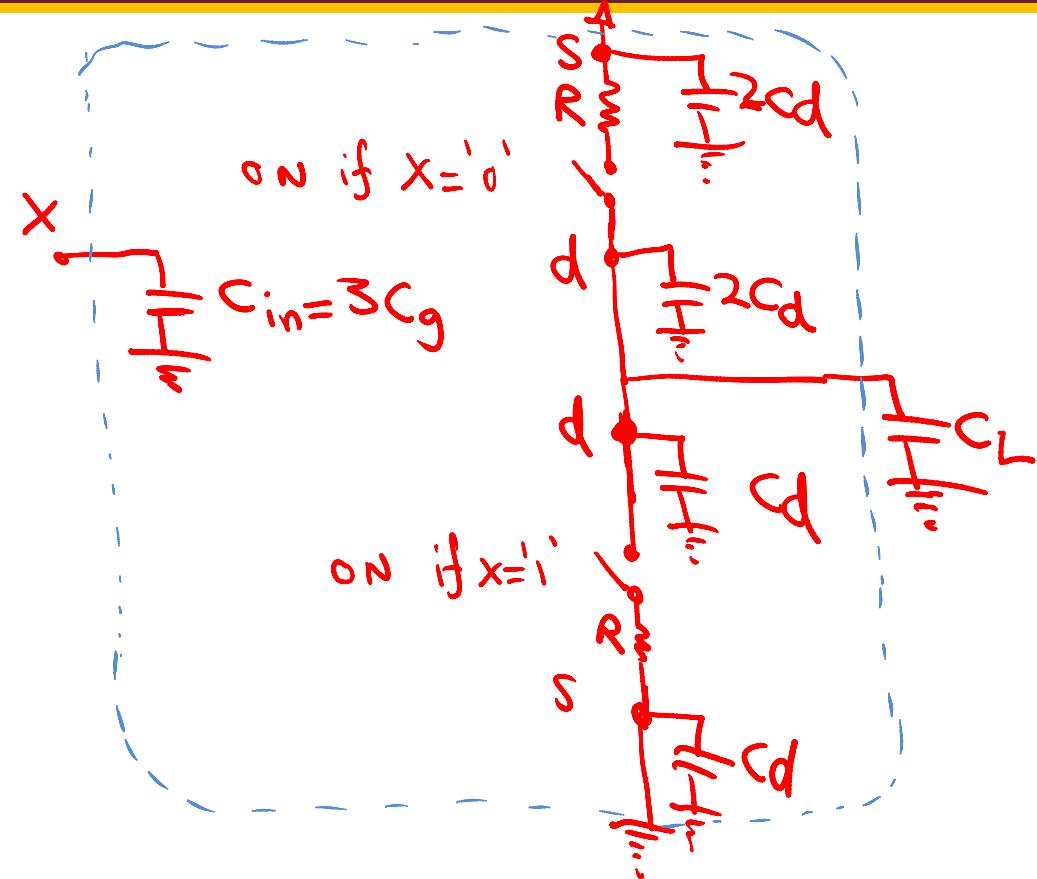
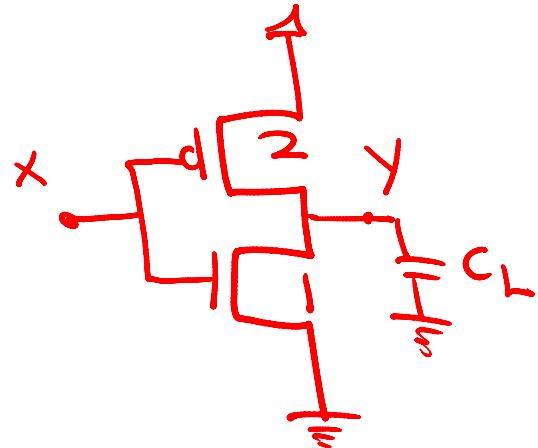
C_g

C_d

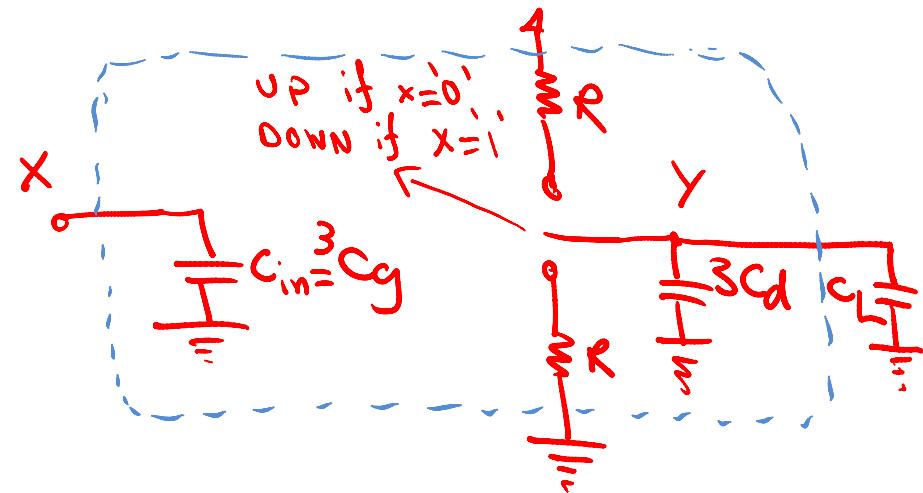
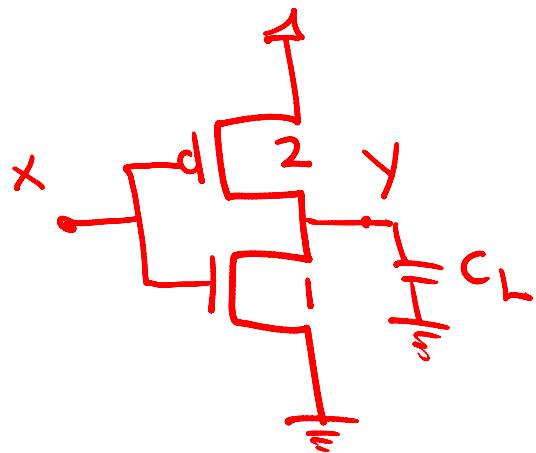
$\mu_R R$

μ_p

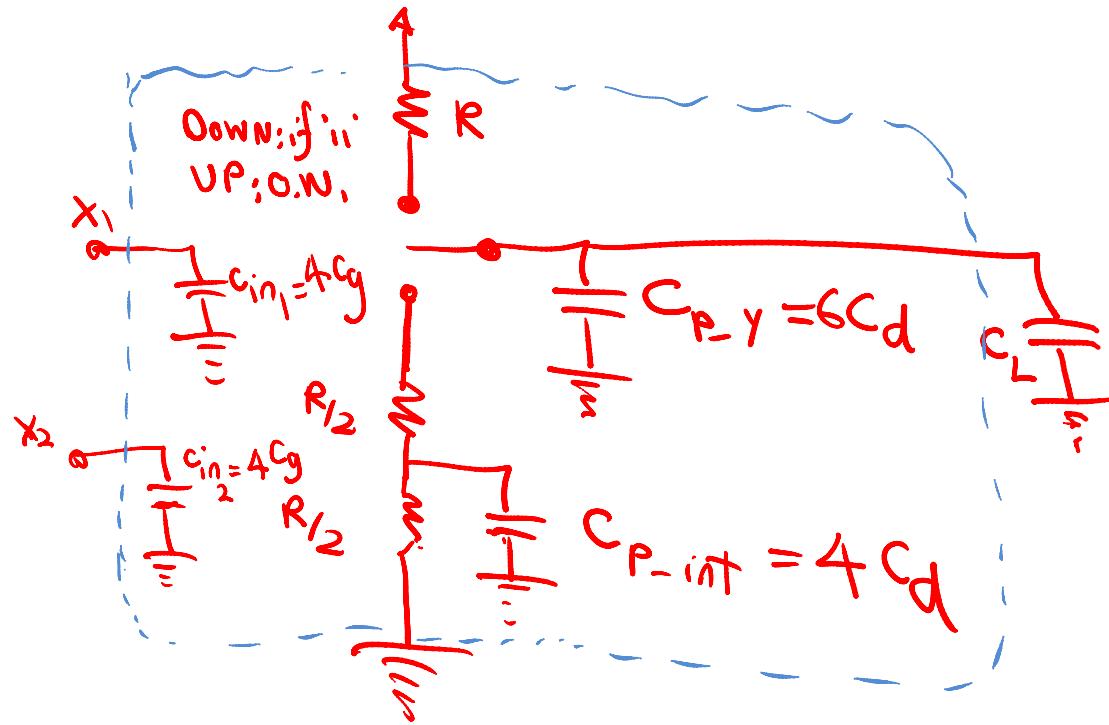
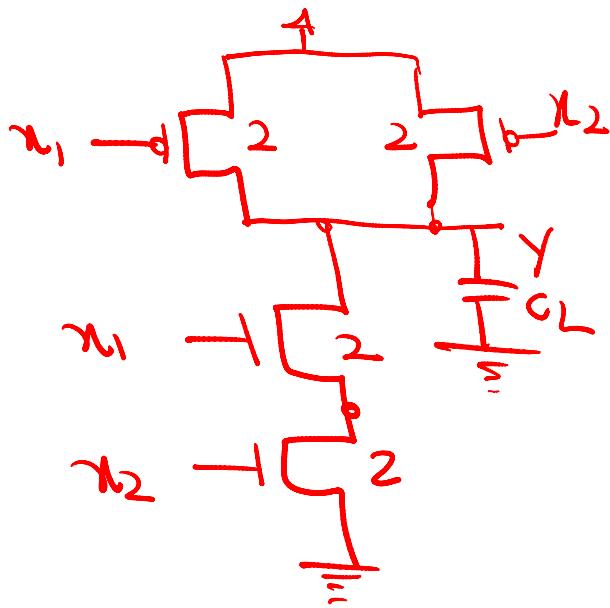
Gate Level Model



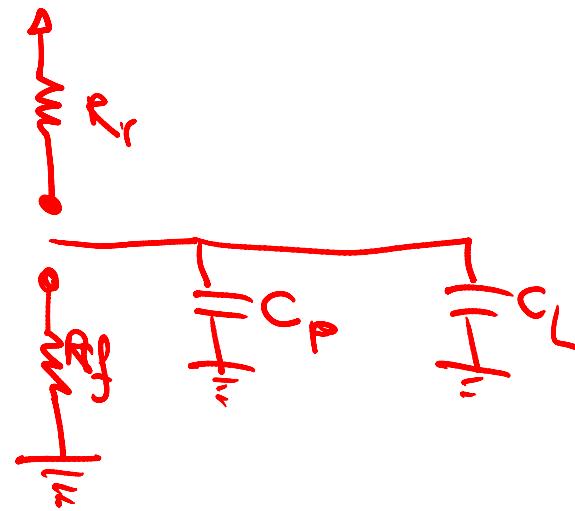
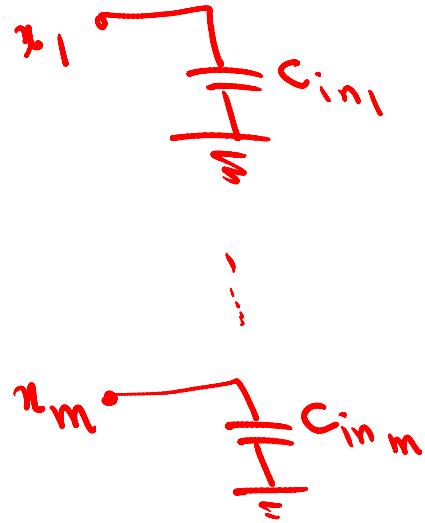
Gate Level Model (Cont.)



Gate Level Model (Cont.)

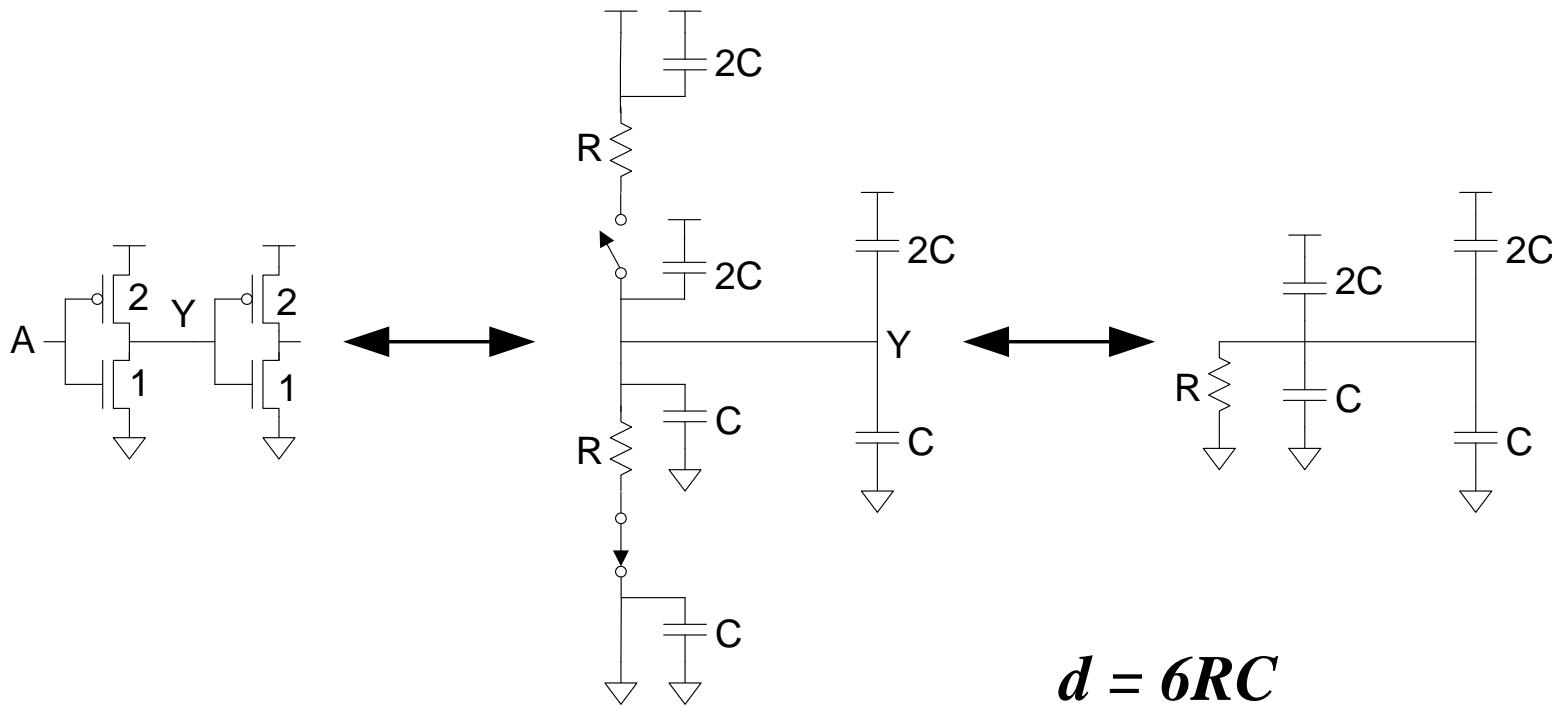


Gate Level Model (Cont.)



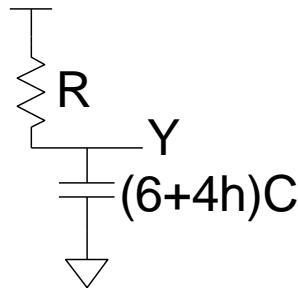
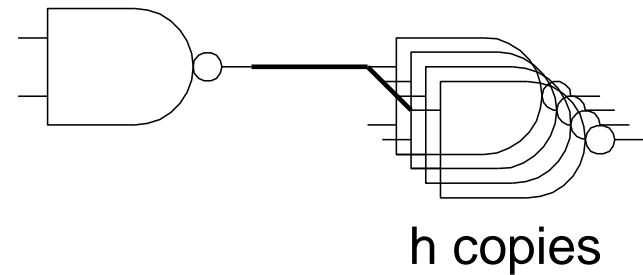
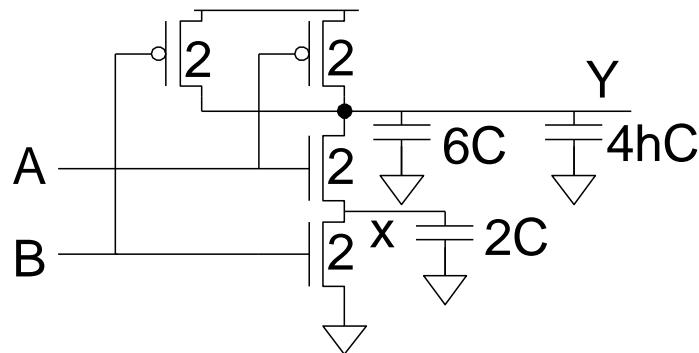
Example: Inverter Delay Estimate

- Estimate the delay of a fanout-of-1 inverter



Example: 2-input NAND

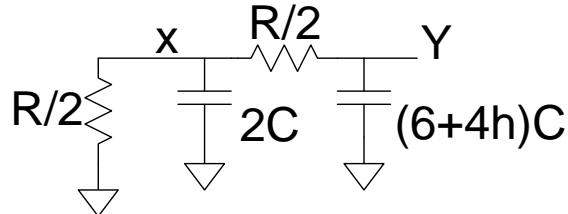
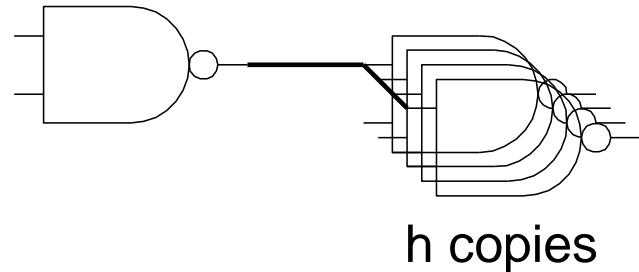
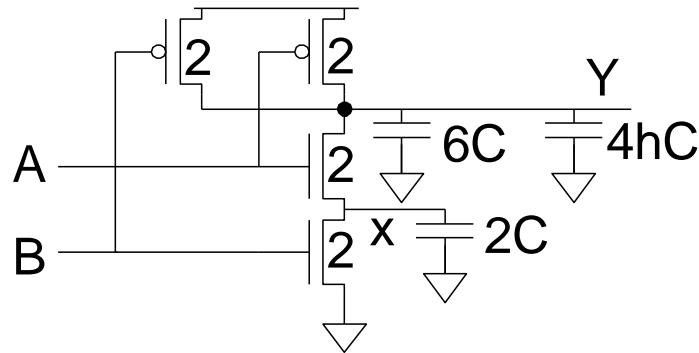
- Estimate **rising** and falling propagation delays of a 2-input NAND driving h identical gates



$$t_{pLH} = (6 + 4h)RC$$

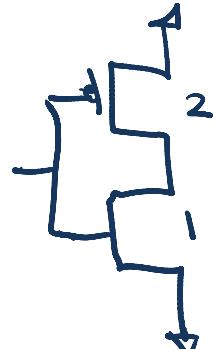
Example: 2-input NAND (Cont.)

- Estimate rising and **falling** propagation delays of a 2-input NAND driving h identical gates

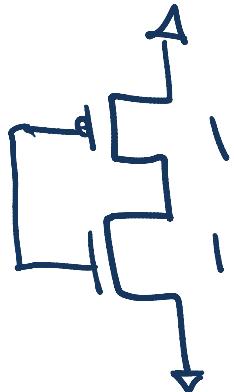


$$\begin{aligned} t_{pHL} &= \left(2C\right)\left(\frac{R}{2}\right) + \left[(6+4h)C\right]\left(\frac{R}{2} + \frac{R}{2}\right) \\ &= (7+4h)RC \end{aligned}$$

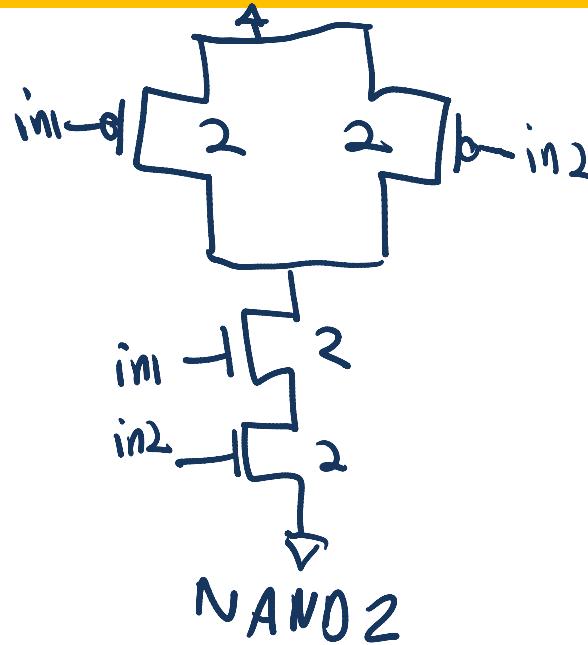
Template Example



INV



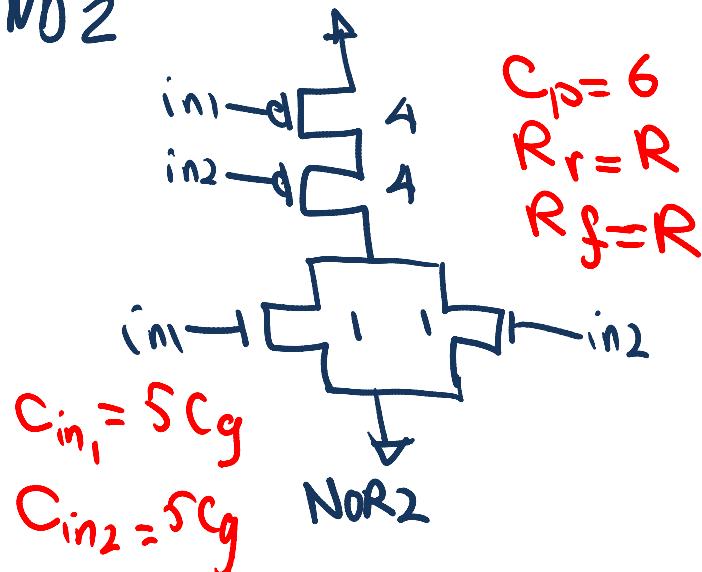
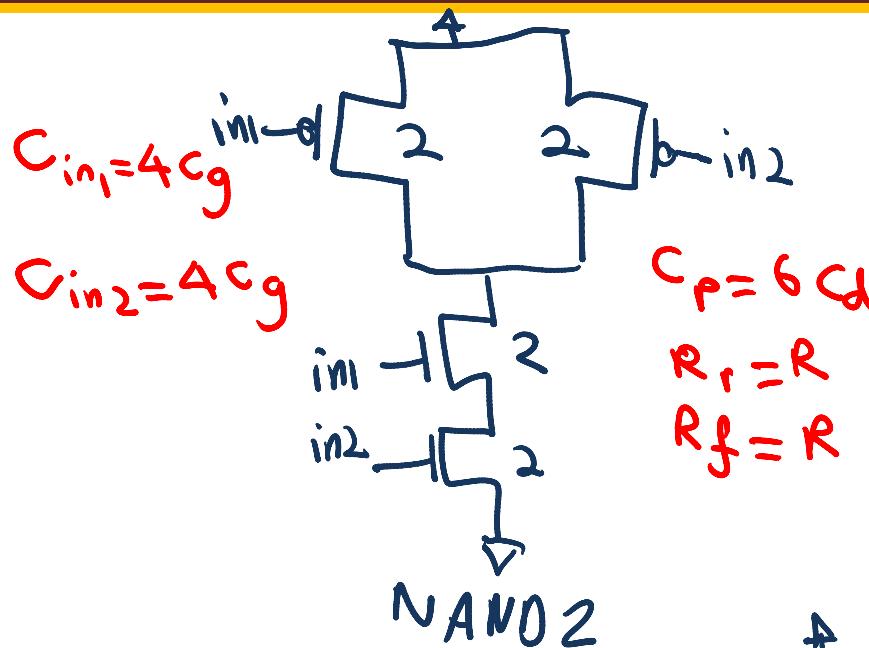
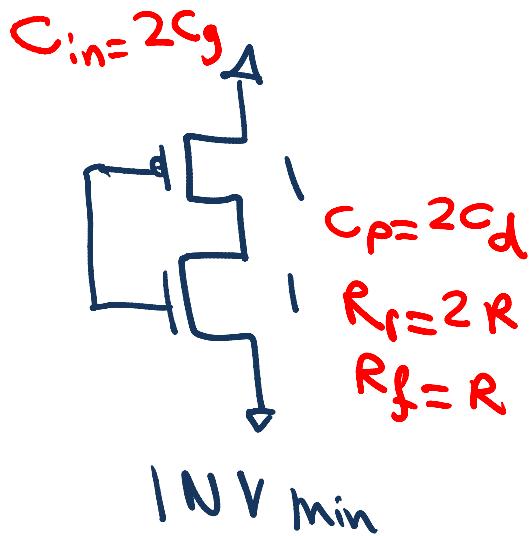
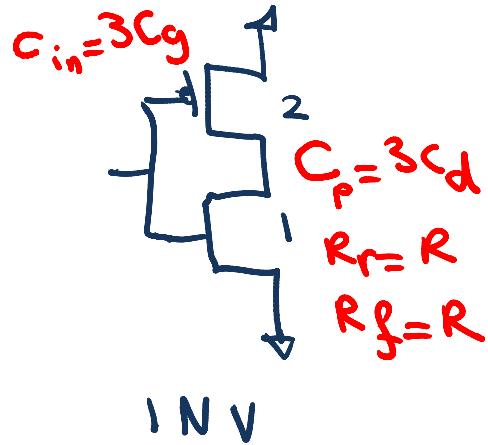
INV_{min}



Gate Delay and Templates (Cont.)

$$\begin{aligned} & \kappa R_f (C_L + C_P) \\ &= \kappa \frac{R_f^{rt} C_{in}^{rt}}{R_f^{rt} C_{in}^{rt}} (R_f C_L + R_f C_P) \\ &= \kappa \frac{R_f^{rt} C_{in}^{rt}}{R_f^{rt} C_{in}^{rt}} (R_f C_{in_1} \frac{C_L}{C_{in_1}} + R_f C_P) \\ &= \kappa R_f^{rt} C_{in}^{rt} \left(\frac{R_f C_{in_1}}{R_f^{rt} C_{in}^{rt}} \left(\frac{C_L}{C_{in_1}} \right) + \frac{R_f C_P}{R_f^{rt} C_{in}^{rt}} \right) \end{aligned}$$

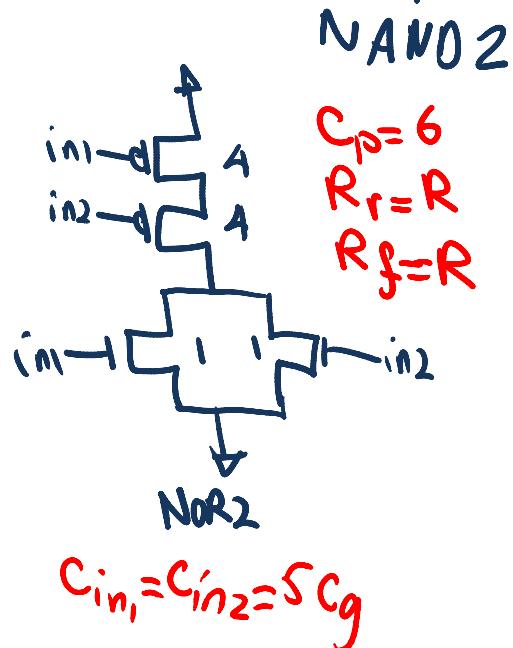
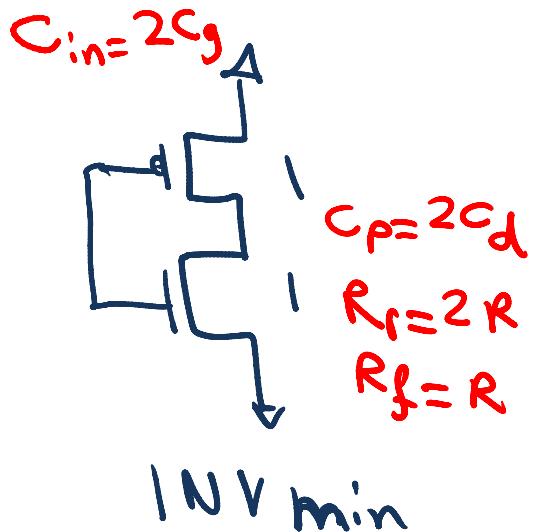
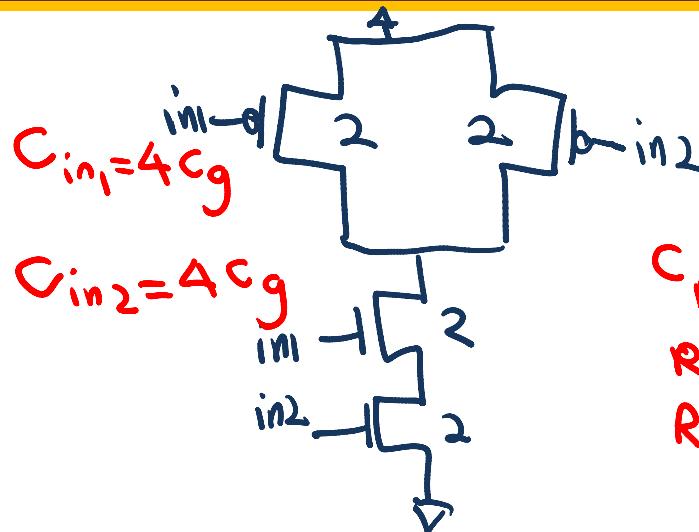
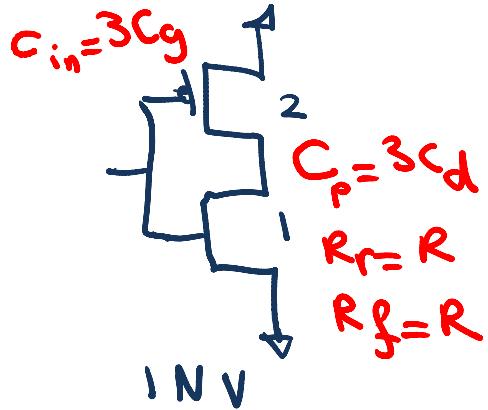
Template Example



Gate Delay and Templates (Cont.)

$$\begin{aligned} & \kappa R_f (C_L + C_P) \\ &= \kappa \frac{R_f^{rt} C_{in}}{R_f^{rt} C_{in}} (R_f C_L + R_f C_P) \\ &= \kappa \frac{R_f^{rt} C_{in}}{R_f^{rt} C_{in}} (R_f C_{in_1} \frac{C_L}{C_{in_1}} + R_f C_P) \\ &= \kappa R_f^{rt} C_{in} \left(\frac{R_f C_{in_1}}{R_f^{rt} C_{in}} \left(\frac{C_L}{C_{in_1}} \right) + \frac{R_f C_P}{R_f^{rt} C_{in}} \right) \\ &\triangleq \tau(g_{f-in_1}, h_{in_1}, p_{f-in_1}) \end{aligned}$$

Template Example

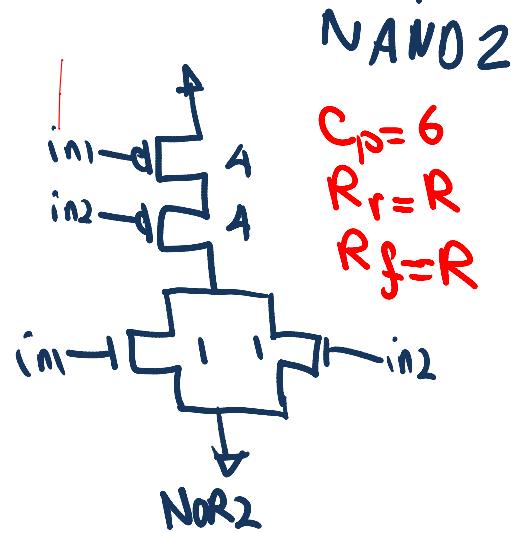
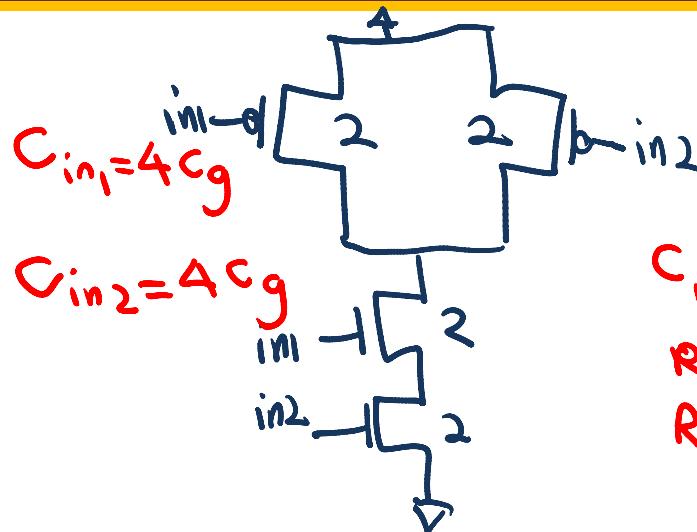
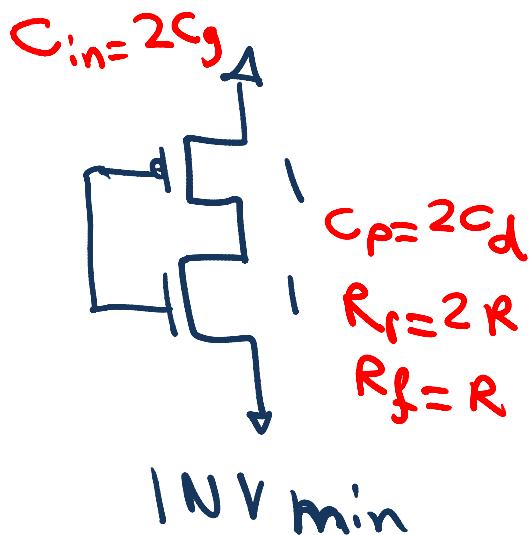
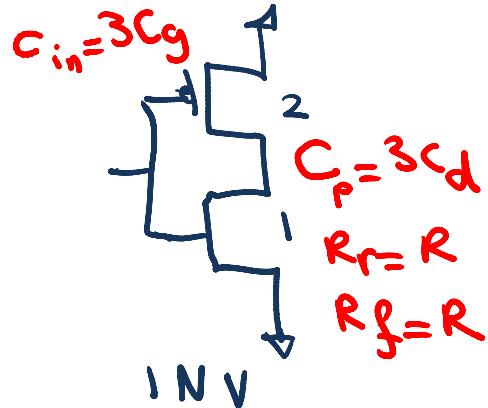


$$g_s = \frac{R}{R_s^{it}} C_{in}^{it}$$

$$P_f = \frac{R_f C_p}{R_s^{it} C_{in}^{it}}$$

	g_f	g_r	P_f	P_r
INV	1	1	$\frac{P_f}{g_f g_r}$	$\frac{P_r}{C_p C_d}$
INV _{min}	$2/3$	$4/3$	$\frac{2C_d}{3C_g}$	$\frac{4C_d}{3C_g}$
NAND2	$4/3$	$4/3$	$\frac{2C_d}{C_g}$	$\frac{2C_d}{C_g}$
NOR2	$5/3$	$5/3$	$\frac{2C_d}{C_g}$	$\frac{2C_d}{C_g}$

Template Example (Cont.)



$$C_{in1} = C_{in2} = 5C_g$$

$$g_s = \frac{R_s}{R_s^{it} C_{in}^{it}}$$

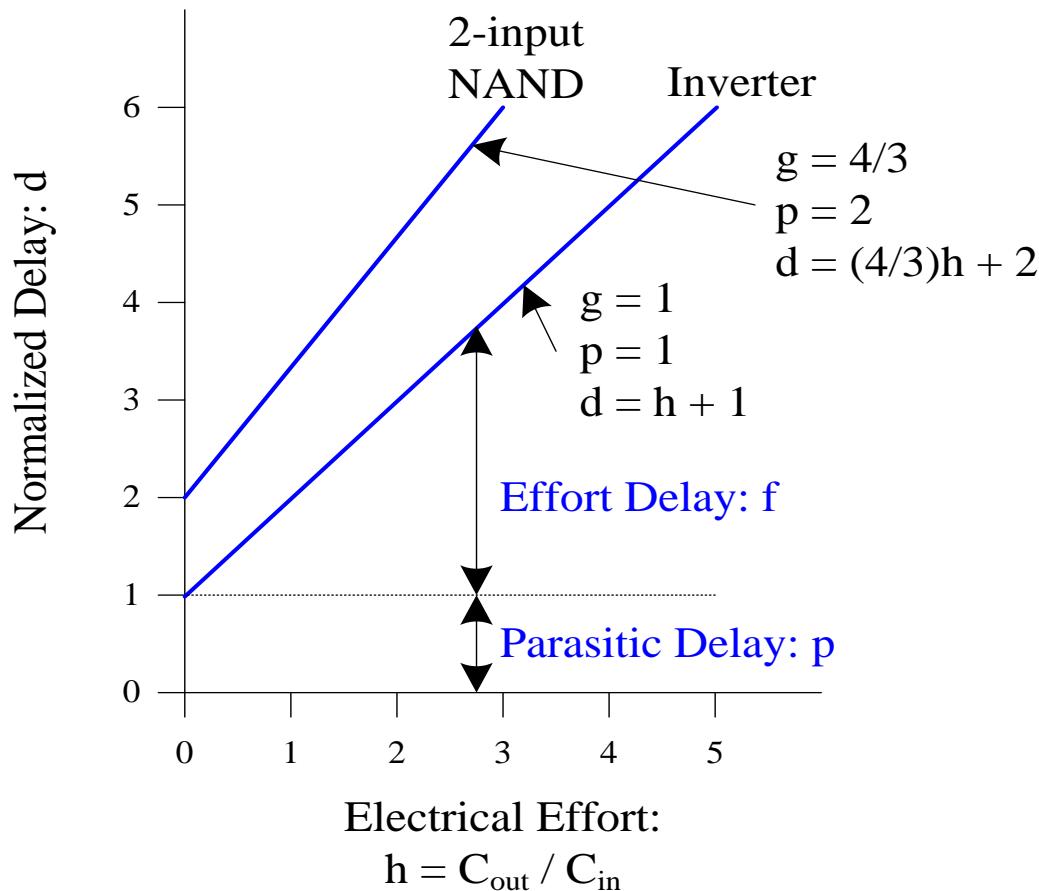
$$P_f = \frac{R_f C_p}{R_s^{it} C_{in}^{it}}$$

	g_s	g_r	P_f	P_r
INV	1	1	P_{inv}	P_{inv}
INV _{min}	$2/3$	$4/3$	$\frac{3}{3} P_{inv}$	$\frac{4}{3} P_{inv}$
NAND2	$4/3$	$4/3$	$2 P_{inv}$	$2 P_{inv}$
NOR2	$5/3$	$5/3$	$2 P_{inv}$	$2 P_{inv}$

Delay in a Logic Gate

- Delay has two components: $d = f + p$
 - f : *stage effort* (a.k.a. effort delay)
 - Again has two components i.e., $f = g h$
 - 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 the output to input pin capacitance
 - p : *parasitic delay*
 - It represents delay of a gate driving no load

Delay Plots



$$d = g h + p$$

Logical Effort of Common Logic Gates

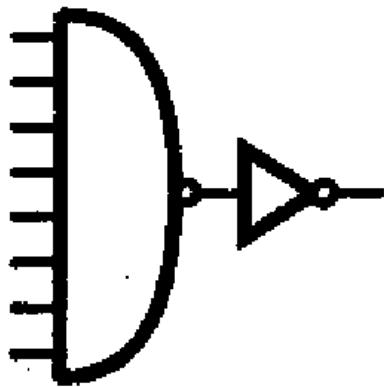
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	$(n+2)/3$
NOR		5/3	7/3	9/3	$(2n+1)/3$
Tristate Buffer / Mux	2	2	2	2	2
XOR, XNOR per bundle		4	12	32	$n2^{n-1}$

Parasitic Delay of Common Logic Gates

- Parasitic delay given in multiples of p_{inv} (≈ 1)

Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate Buffer / Mux	2	4	6	8	$2n$
XOR, XNOR		4	12	32	$n2^{n-1}$

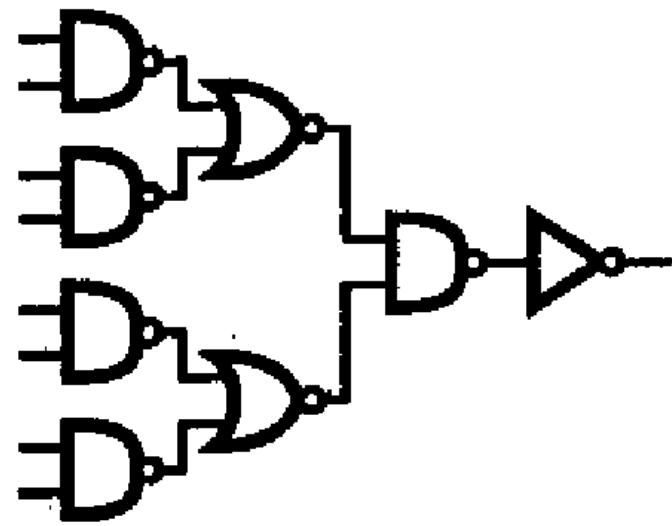
Example: AND8



$g=10/3$
 $p=8$

$g=1$
 $p=1$

$g=6/3=2$ $g=5/3$
 $p=4$ $p=2$



$g=4/3$ $g=5/3$ $g=4/3$ $g=1$
 $p=2$ $p=2$ $p=2$ $p=1$

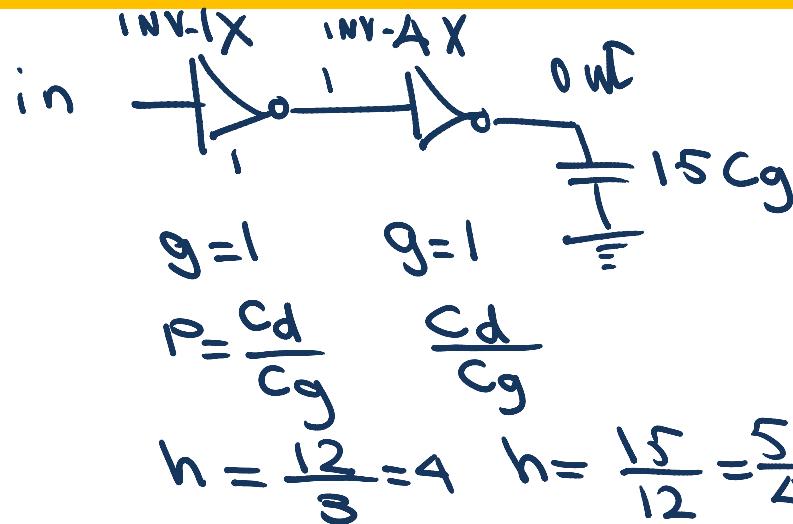
Effect of Sizing on the Logical Effort

- Sizing does not change the logical effort of a gate
 - Let the size of a gate be increased by a factor α , then we have:

$$\frac{R}{\alpha} \alpha C = RC$$

which proves that the logical effort is independent of gate sizing

Example: Delay Calculation



$$d_{r-out} = d_f + d_i$$

$$\begin{aligned} d_{r-out} &= d_{f-out} = \gamma \left(4 + \frac{C_d}{C_g} + \frac{5}{4} + \frac{C_d}{C_g} \right) = \gamma \left(\frac{21}{4} + \frac{2C_d}{C_g} \right) \\ &= \gamma (5.25 + 2P_{inv}) \end{aligned}$$

Example: Delay Calculation (Cont.)

i_{in1} i_{in2} $NAND2ZX$ INV_SX

$g = \frac{4}{3}$ $g = 1$ $\frac{1}{100 C_g}$

$P = \frac{2 C_d}{C_g}$ $P = \frac{C_d}{C_g}$ $P_{inv} \triangleq \frac{C_d}{C_g}$

$\Rightarrow P = 2 P_{inv}$ $P = P_{inv}$

$h = \frac{15}{8}$ $h = \frac{100}{15}$

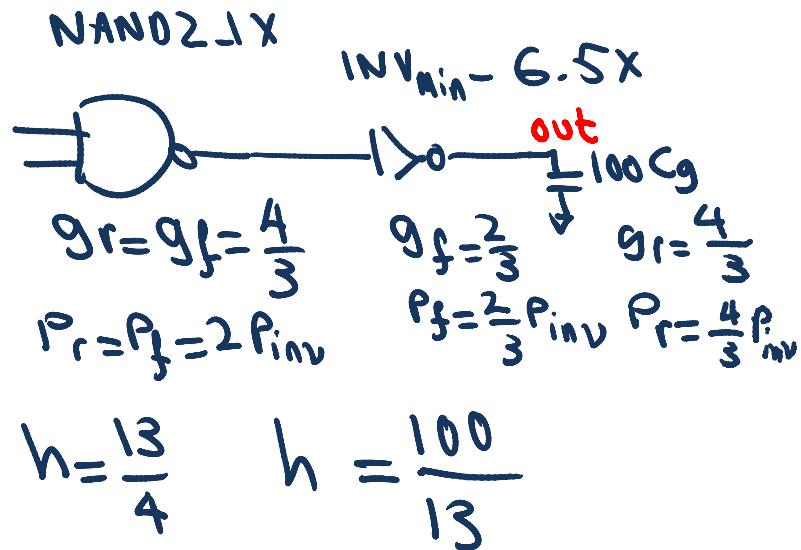
normalized to \sim :

$$d_{f-out} = d_{f-out} = \frac{5}{2} + 2P_{inv} + \frac{20}{3} + P_{inv} = \frac{55}{6} + 3P_{inv}$$

assume: $P_{inv} \approx 1 \Rightarrow$

$$\approx \frac{73}{6} = 12.2$$

Example: Delay Calculation (Cont.)



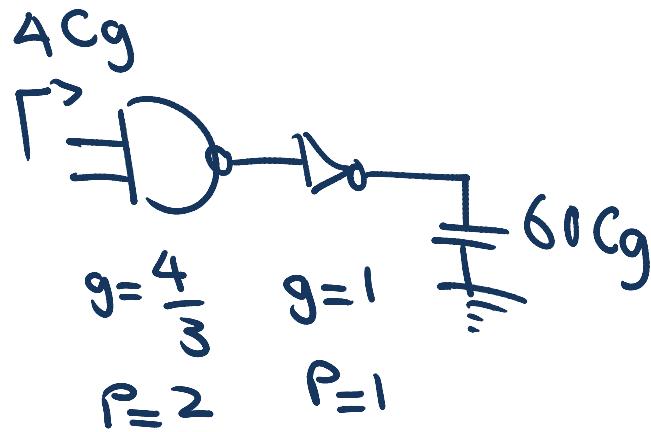
$$d_{r-out} = \frac{13}{3} + 2P_{inv} + \frac{400}{39} + \frac{4}{3}P_{inv}$$
$$= \frac{569}{39} + \frac{10}{3}P_{inv} \approx \frac{569}{39} + \frac{10}{3} \approx 17.9$$

Delay Calculation of a circuit

$$\text{delay} = \sum_{i=1}^N g_i h_i + p_i \quad \text{for the } N \text{ gates on the path}$$

Example: Delay Optimization

- Find the fastest AND2 given its input and output caps are $4C_g$ and $60C_g$, respectively

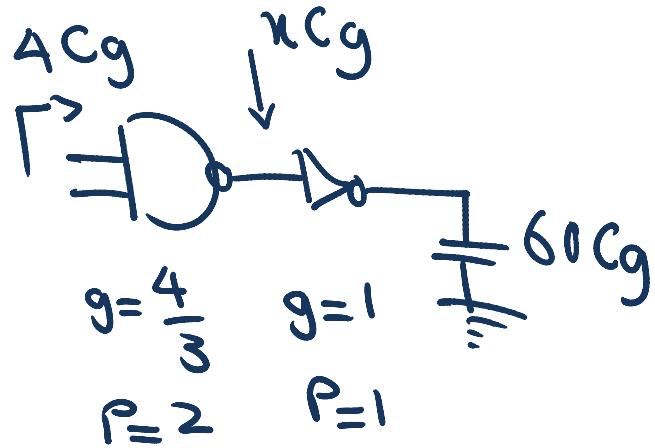


$$h = \frac{3}{4} \quad h = \frac{60}{3} = 20$$

$$\text{delay} = \frac{4}{3} \times \frac{3}{4} + 2 + 20 + 1 = 24$$

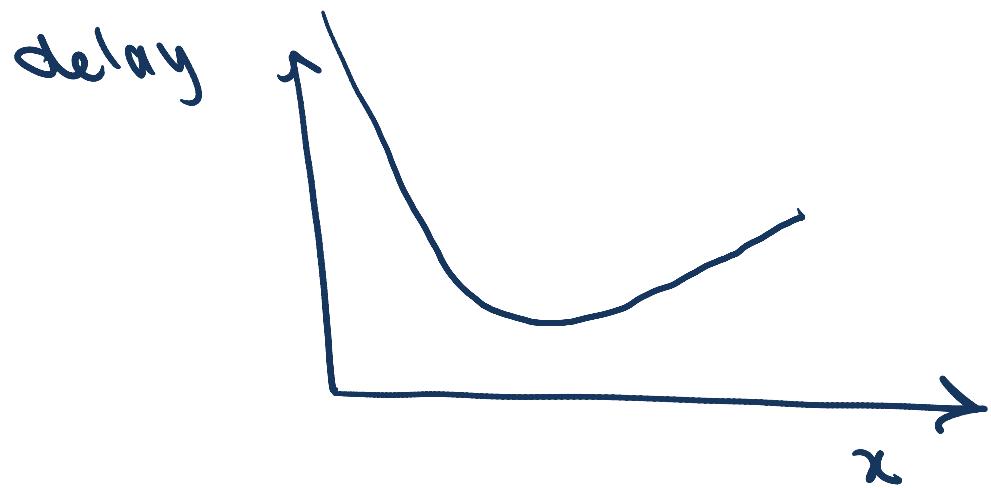
Example: Delay Optimization (Cont.)

- Find the fastest AND2 given its input and output caps are $4C_g$ and $60C_g$, respectively



$$\begin{aligned}\text{delay}(\pi) &= \frac{4}{3} \times \frac{\pi}{4} + 2 + \frac{60}{\pi} + 1 \\ &= \frac{\pi}{3} + \frac{60}{\pi} + 3 \quad \approx 12\end{aligned}$$

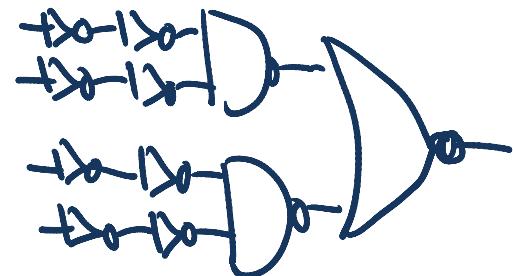
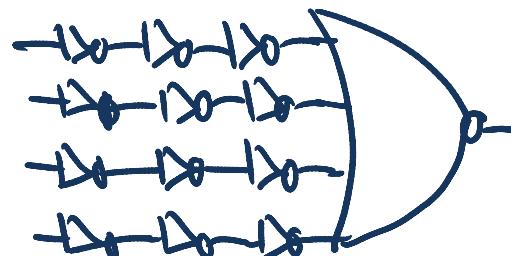
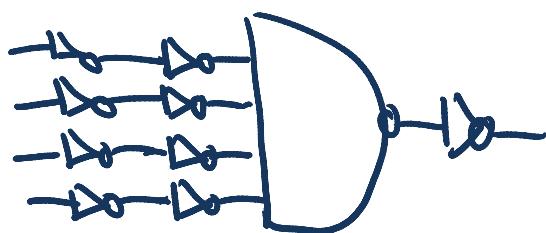
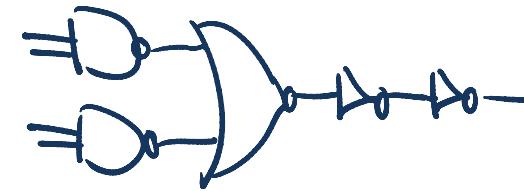
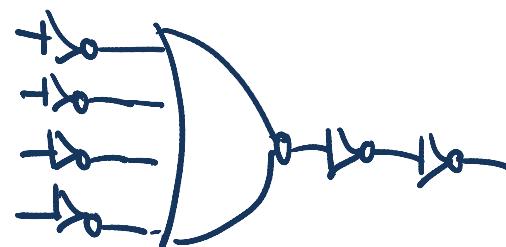
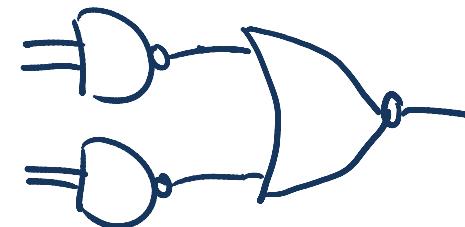
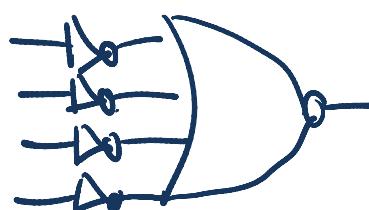
Sizing Properties



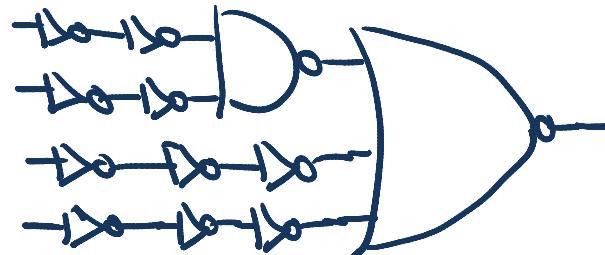
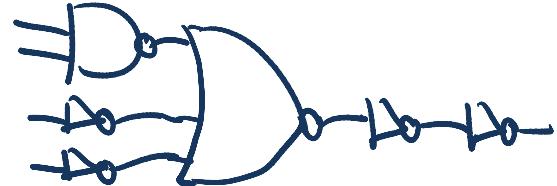
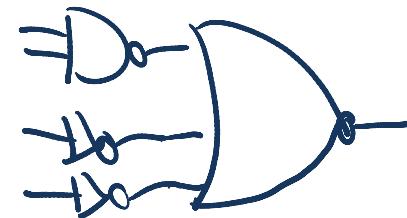
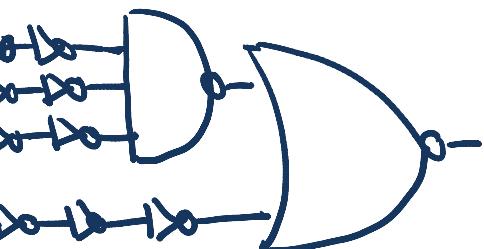
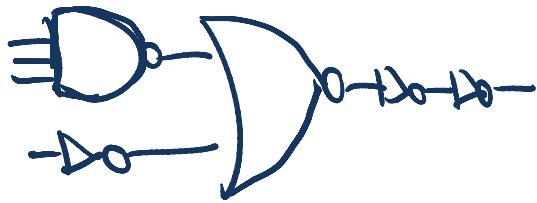
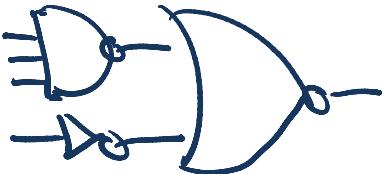
example:

$$x/3 + \frac{60}{x} + 3$$

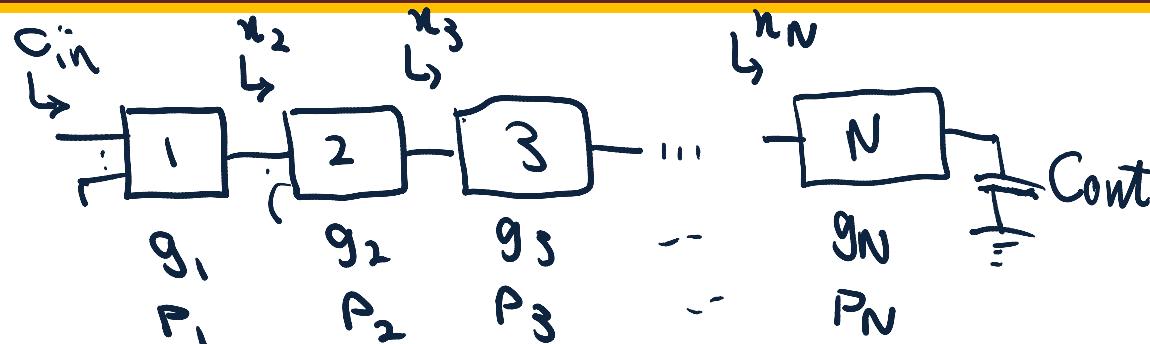
Design Alternatives



Design Alternatives (Cont.)



Delay Optimization formulation



$$\text{Path Delay} = g_1 \frac{x_2}{C_{in}} + P_1 + g_2 \frac{x_3}{x_2} + P_2 + \dots + g_N \frac{C_{out}}{x_N} + P_N$$

$$\frac{\partial \text{Delay}}{\partial x_2} = \frac{g_1}{C_{in}} - \frac{g_2 x_3}{x_2^2} = 0 \Rightarrow \frac{g_1 x_2}{C_{in}} = g_2 \frac{x_3}{x_2}$$

$$\frac{\partial \text{Delay}}{\partial x_3} = \frac{g_2}{x_2} - \frac{g_3 x_4}{x_3^2} = 0 \Rightarrow \frac{g_2 x_3}{x_2} = g_3 \frac{x_4}{x_3}$$

$$f \triangleq g_i h_i \Rightarrow \text{Path delay} = N f + \sum P_i \quad F \triangleq f^N = g_1 g_2 \dots g_N \frac{C_{out}}{C_{in}}$$

$$\Rightarrow = N \left[g_1 g_2 \dots g_N \frac{C_{out}}{C_{in}} \right]^N + \sum P_i = N [GH]^N + P$$

Note that: $g_1 \hat{h}_1 = g_2 \hat{h}_2 = g_3 \hat{h}_3 = \dots = g_N \hat{h}_N$

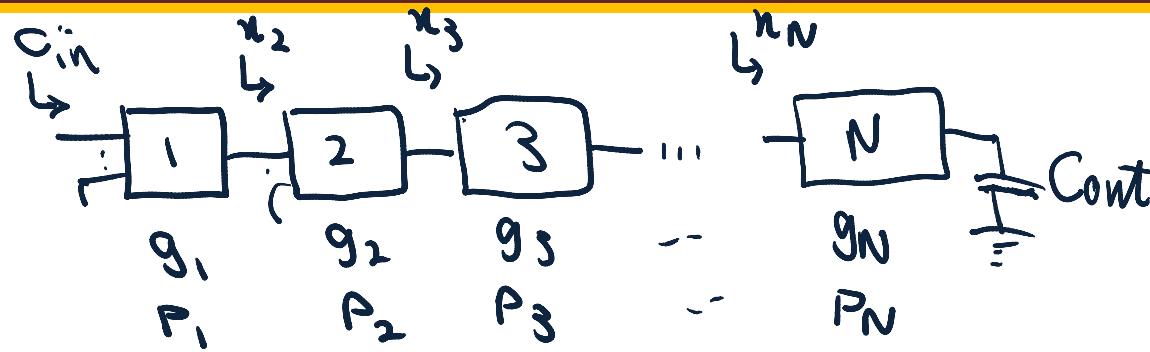
$$G \triangleq g_1 g_2 \dots g_N$$

$$H \triangleq C_{out}/C_{in}$$

$$P \triangleq P_1 + P_2 + \dots + P_N$$

$$F \triangleq GH$$

Delay Optimization formulation (Cont.)



$$\hat{D} = N (GH)^{N-1} + P$$

$$G = \prod_{i=1}^N g_i, \quad H = \frac{C_{out}}{C_{in}}$$

$$P = \sum_{i=1}^N p_i$$

Comparison

- Compare many alternatives with a spreadsheet

Design	N	G	P	D
NAND4-INV	2	2	5	29.8
NAND2-NOR2	2	20/9	4	30.1
INV-NAND4-INV	3	2	6	22.1
NAND4-INV-INV-INV	4	2	7	21.1
NAND2-NOR2-INV-INV	4	20/9	6	20.5
NAND2-INV-NAND2-INV	4	16/9	6	19.7
INV-NAND2-INV-NAND2-INV	5	16/9	7	20.4
NAND2-INV-NAND2-INV-INV-INV	6	16/9	8	21.6

Example: Delay Optimization of an OR8

- Library of templates

	g	p
INV	1	1
NAND2	4/3	2
NAND4	6/3	4
NAND8	10/3	8
NOR2	5/3	2
NOR4	9/3	4
NOR8	17/3	8

$\frac{g}{p}$

$\frac{17}{3} \times \frac{x}{10} + 8 + \frac{100}{x} + 1$

For optimality: $\frac{17}{3} \times \frac{x}{10} = \frac{100}{x}$

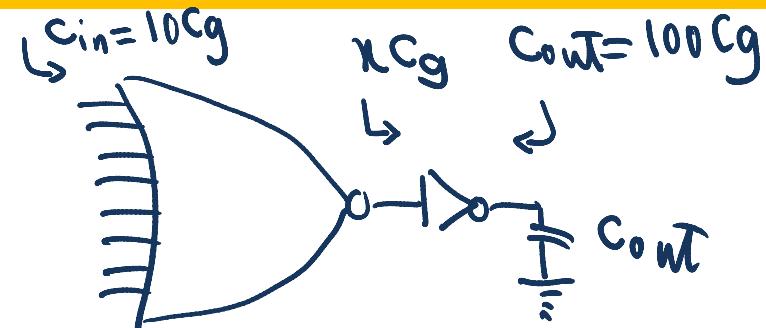
Now instead of calculating x ,

$$\hat{D} = N(GH)^{1/N} + P$$

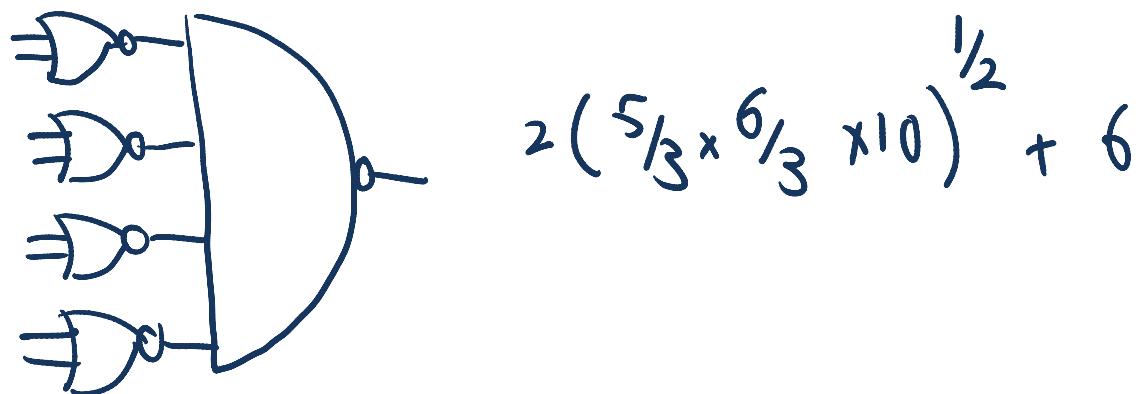
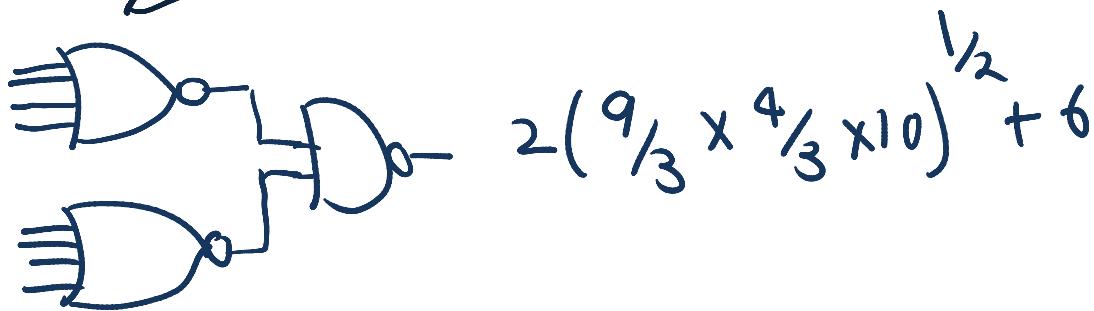
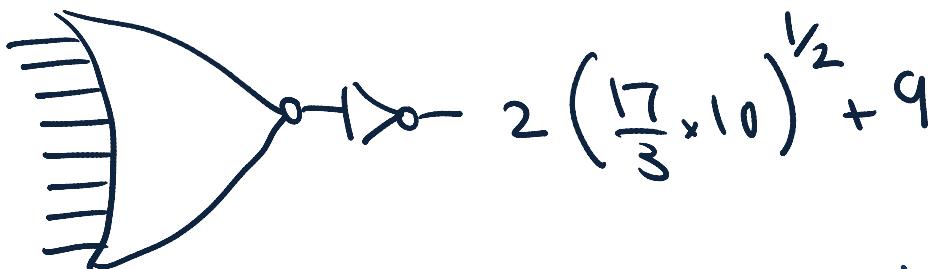
$$G = g_1 g_2 \dots g_N$$

$$H = \frac{C_{out}}{C_{in}}$$

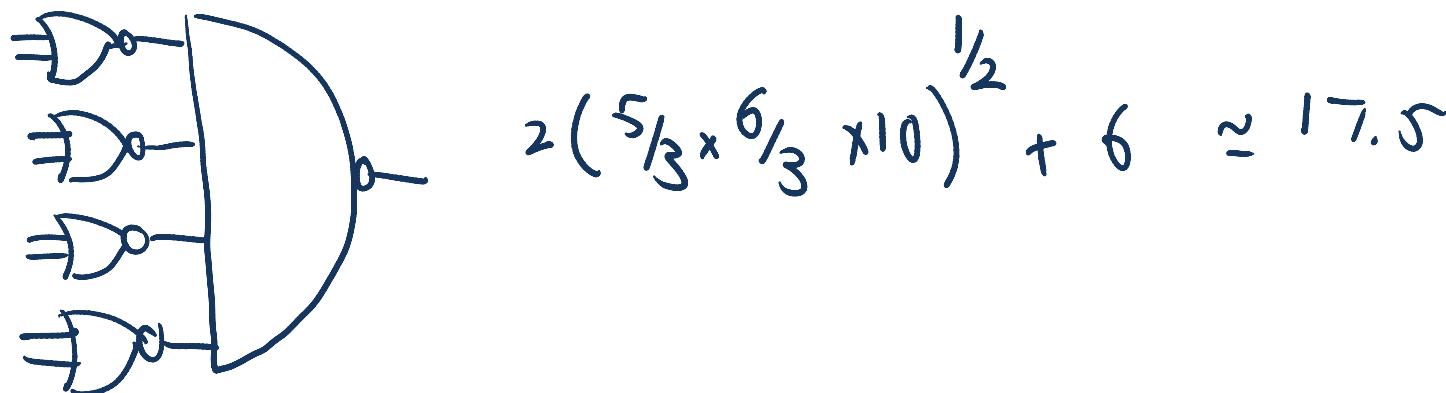
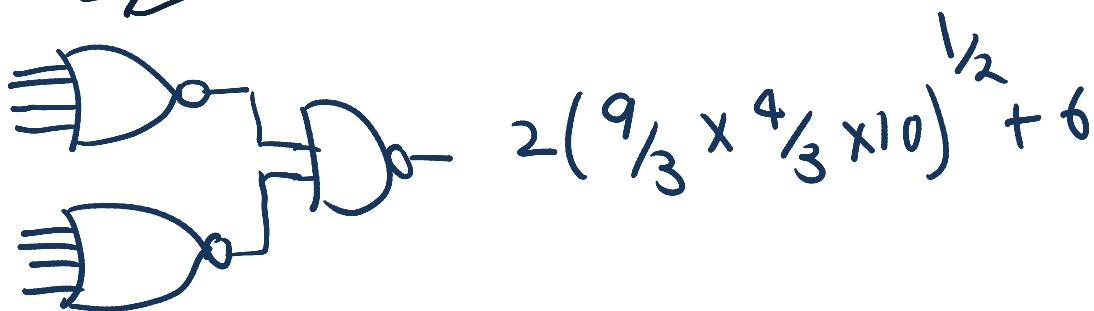
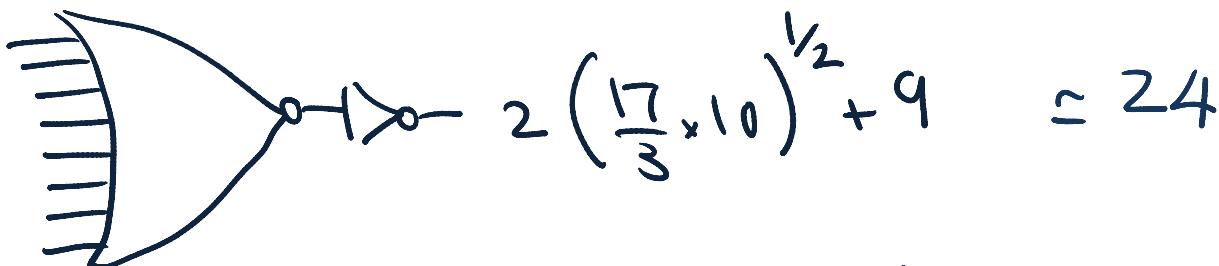
$$P = P_1 + P_2 + \dots + P_N$$



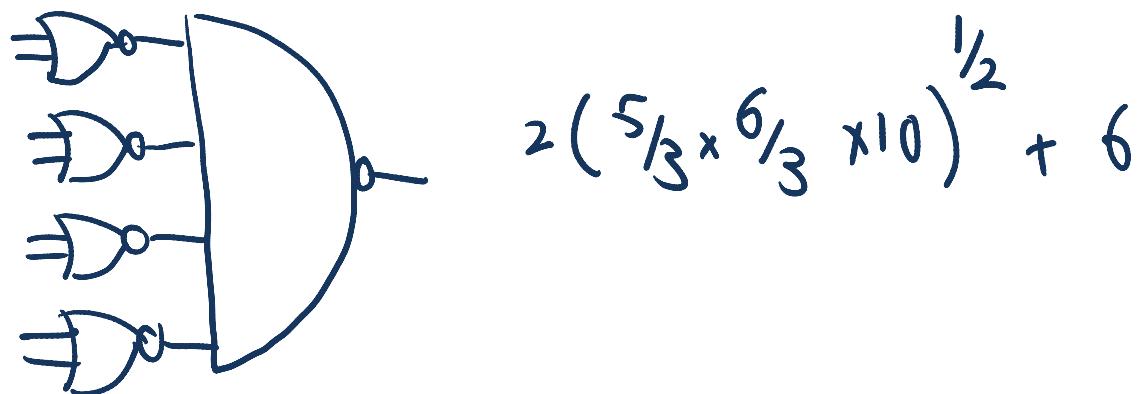
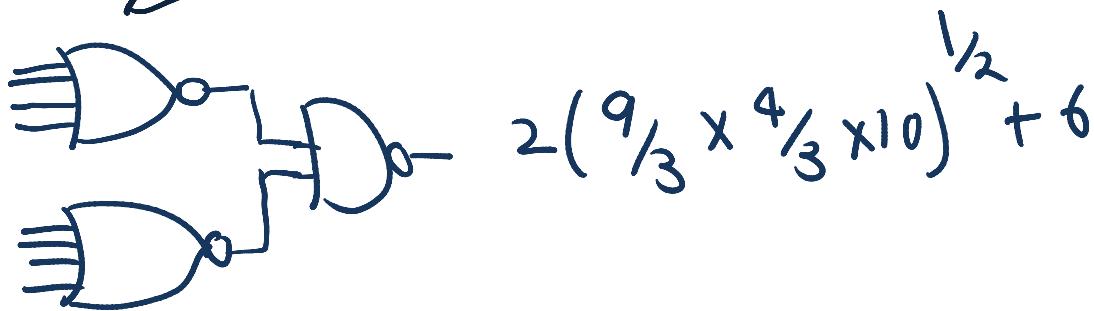
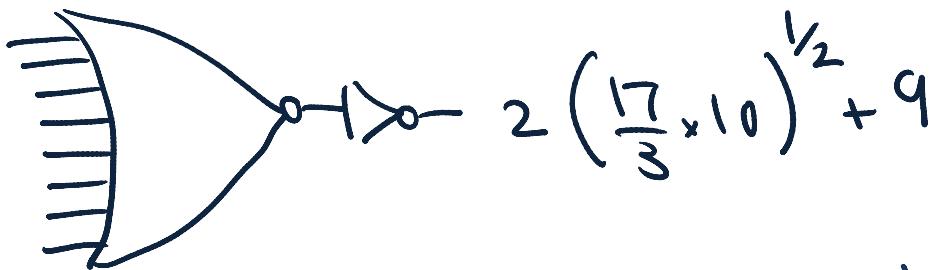
Example: Optimal OR8 (Cont.)



Example: Optimal OR8 (Cont.)



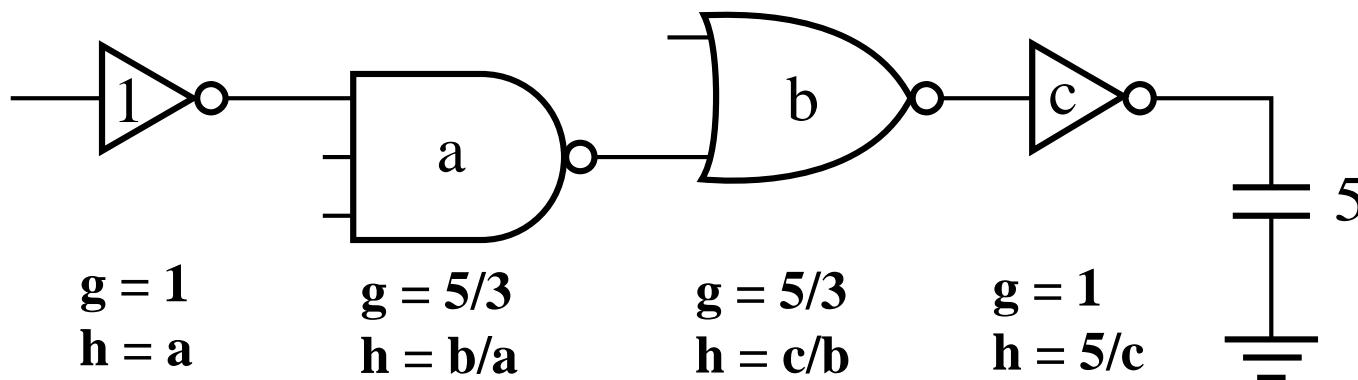
Example: Optimal OR8 (Cont.)



," ... \rightarrow \rightarrow \rightarrow

$$4 \left(\frac{5}{3} \times \frac{6}{3} \times 10 \right)^{\frac{1}{4}} + 8$$

Example: Optimal Delay for a Path



$$H = 5/1 = 5$$

$$G = 25/9$$

$$F = GH = 125/9 = 13.9$$

$$f = \sqrt[4]{F} = 1.93 \text{ (this is the optimal stage effort)}$$

$$\text{1}^{\text{st}} \text{ stage: } a = 1.93$$

$$\text{2}^{\text{nd}} \text{ stage: } (5/3)(b/1.93) = 1.93 \Rightarrow b = 2.23$$

$$\text{3}^{\text{rd}} \text{ stage: } (5/3)(c/2.23) = 1.93 \Rightarrow c = 2.58$$

Confirming that for the 4th stage: $gh=5/c=1.93$

Method of Logical Effort (so far!)

- Compute the path effort: $F = GBH$
- Compute the stage effort $f = F^{1/N}$
- Work from either end, find sizes:
 $C_{in} = C_{out} * g/f$

Branching Effort - Symmetric Branches

- Can we write $F = GH$?
 - No! Consider paths that branch

$$G = 1$$

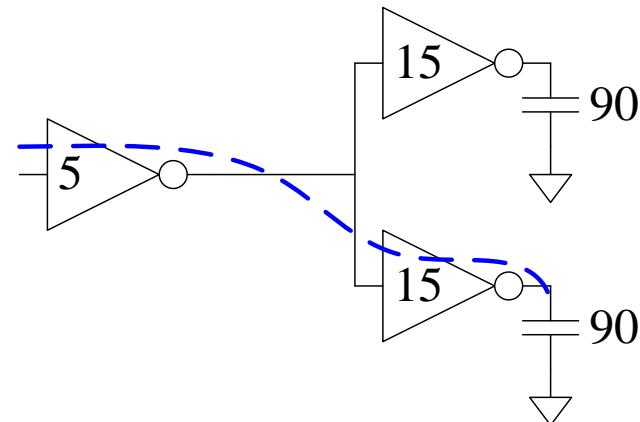
$$H = 90 / 5 = 18$$

$$GH = 18 ?$$

$$h_1 = (15 + 15) / 5 = 6$$

$$h_2 = 90 / 15 = 6$$

$$F = g_1 g_2 h_1 h_2 = 36 = 2GH !$$



Branching Effort

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

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

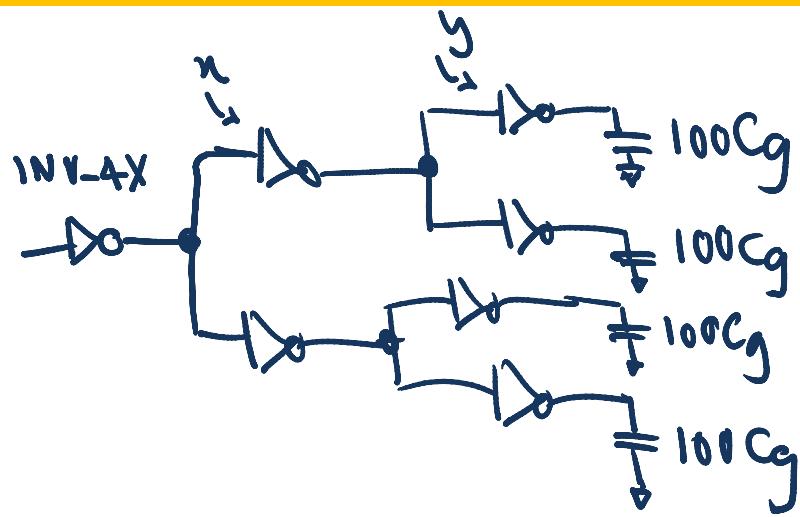
$$B = \prod_i b_i$$

$$\text{Note: } \prod_i h_i = BH$$

- Now we compute the path effort

$$F = GBH$$

Example: Symmetric paths with Branches



Example: 4-to-16 Decoder

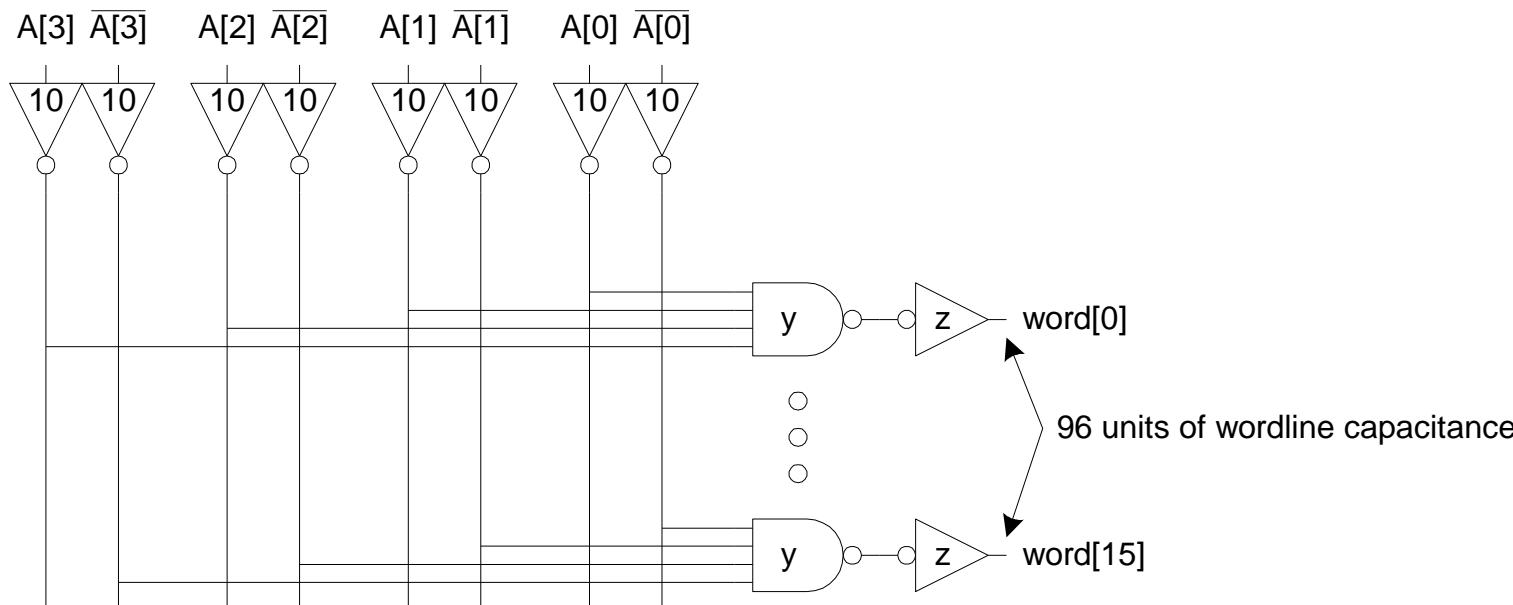
Optimize delay from $A[i]$ or $\sim A[i]$

$$G = 1*6/3*1 = 2, \quad B = 8*1*1=8, \quad H = 96/10=9.6$$

Path Effort: $F = GBH = 153.6$

Stage Effort: $\hat{f} = F^{1/3} = 5.36$ Path Delay: $D = 3\hat{f} + 1 + 4 + 1 = 22.08$

Gate sizes: $z = 96*1/5.36=18$ $y = 18*2/5.36= 6.7$



FO4 Inverter

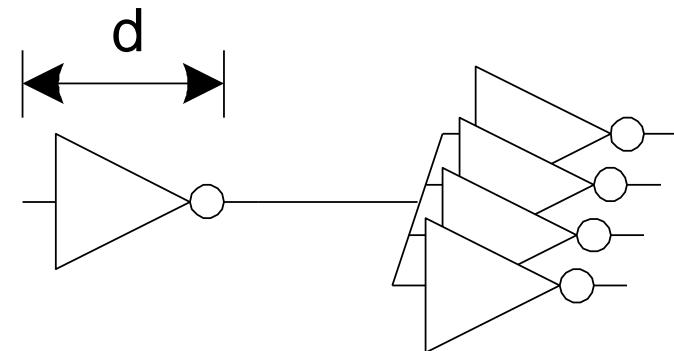
- Estimate the delay of a fanout-of-4 (FO4) inverter

Logical Effort: $g = 1$

Electrical Effort: $h = 4$

Parasitic Delay: $p = 1$

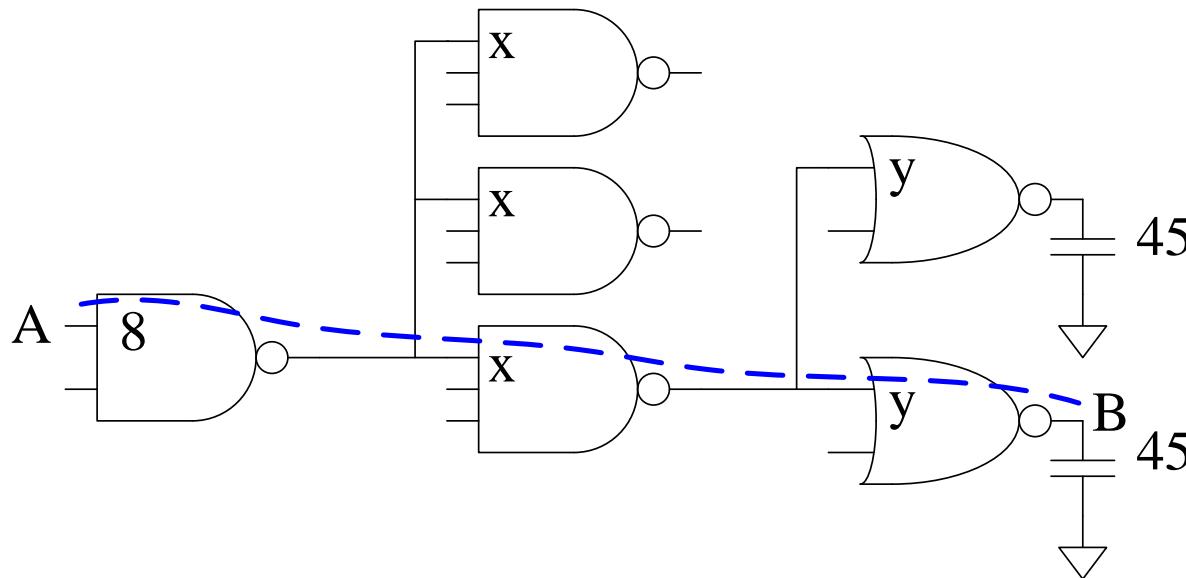
Stage Delay: $d = 5$



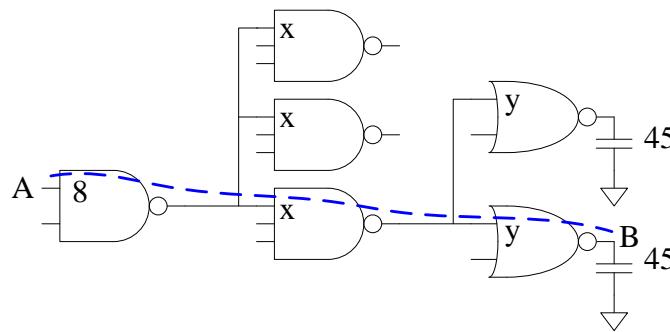
- The FO4 delay is about:
 - 200ps in 0.6 μ m process, 60ps in a 0.18 μ m process
 - $q/3$ ns in a q μ m process

Example: A 3-Stage Path with Branch

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



Example: A 3-Stage Path (Cont.)



Logical Effort $G = (4/3) * (5/3) * (5/3) = 100/27$

Electrical Effort $H = 45/8$

Branching Effort $B = 3 * 2 = 6$

Path Effort $F = GBH = 125$

Best Stage Effort $\hat{f} = \sqrt[3]{F} = 5$

Parasitic Delay $P = 2 + 3 + 2 = 7$

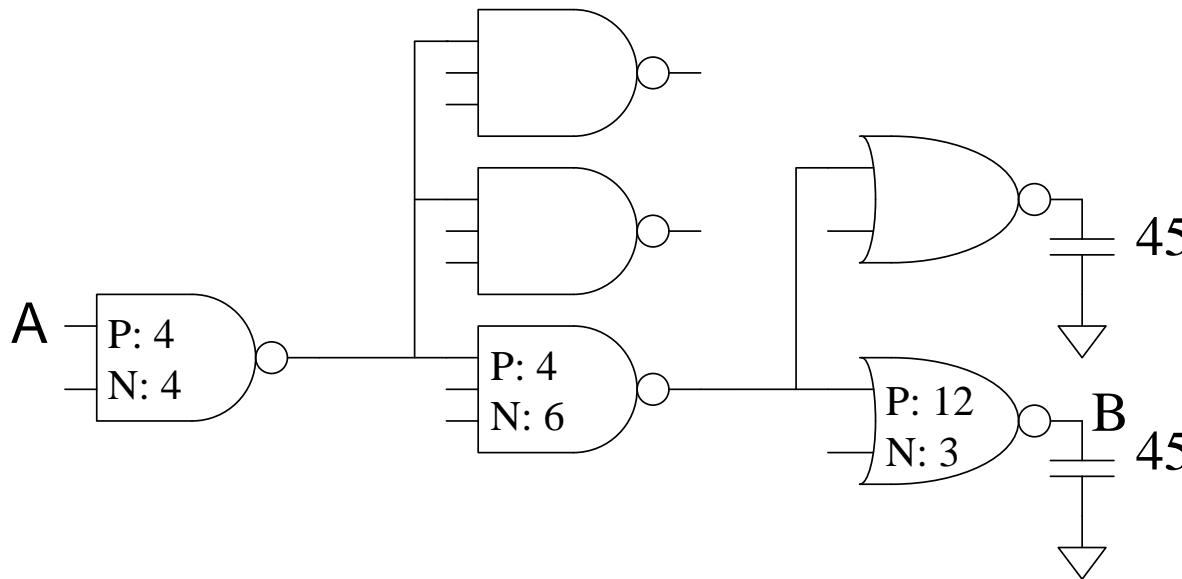
Delay $D = 3*5 + 7 = 22 = 4.4 \text{ FO4}$

Example: A 3-Stage Path with (Cont.)

- Working backward for sizes:

$$y = 45 * (5/3) / 5 = 15$$

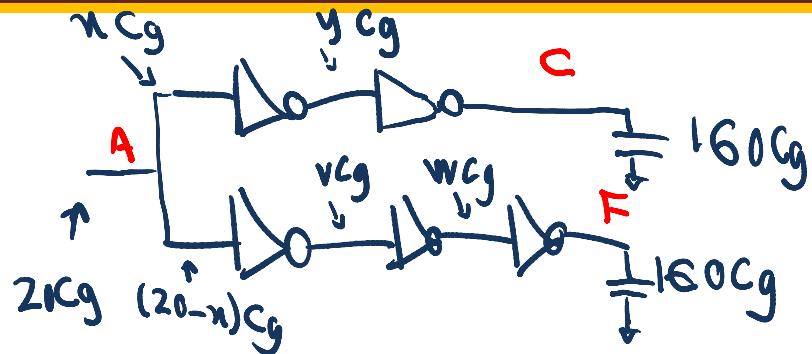
$$x = (15*2) * (5/3) / 5 = 10$$



Review of Definitions

Term	Stage	Path
number of stages	1	N
logical effort	g	$G = \prod g_i$
electrical effort	$h = \frac{C_{\text{out}}}{C_{\text{in}}}$	$H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$
branching effort	$b = \frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}}$	$B = \prod b_i$
effort	$f = gh$	$F = GBH$
effort delay	f	$D_F = \sum f_i$
parasitic delay	p	$P = \sum p_i$
delay	$d = f + p$	$D = \sum d_i = D_F + P$

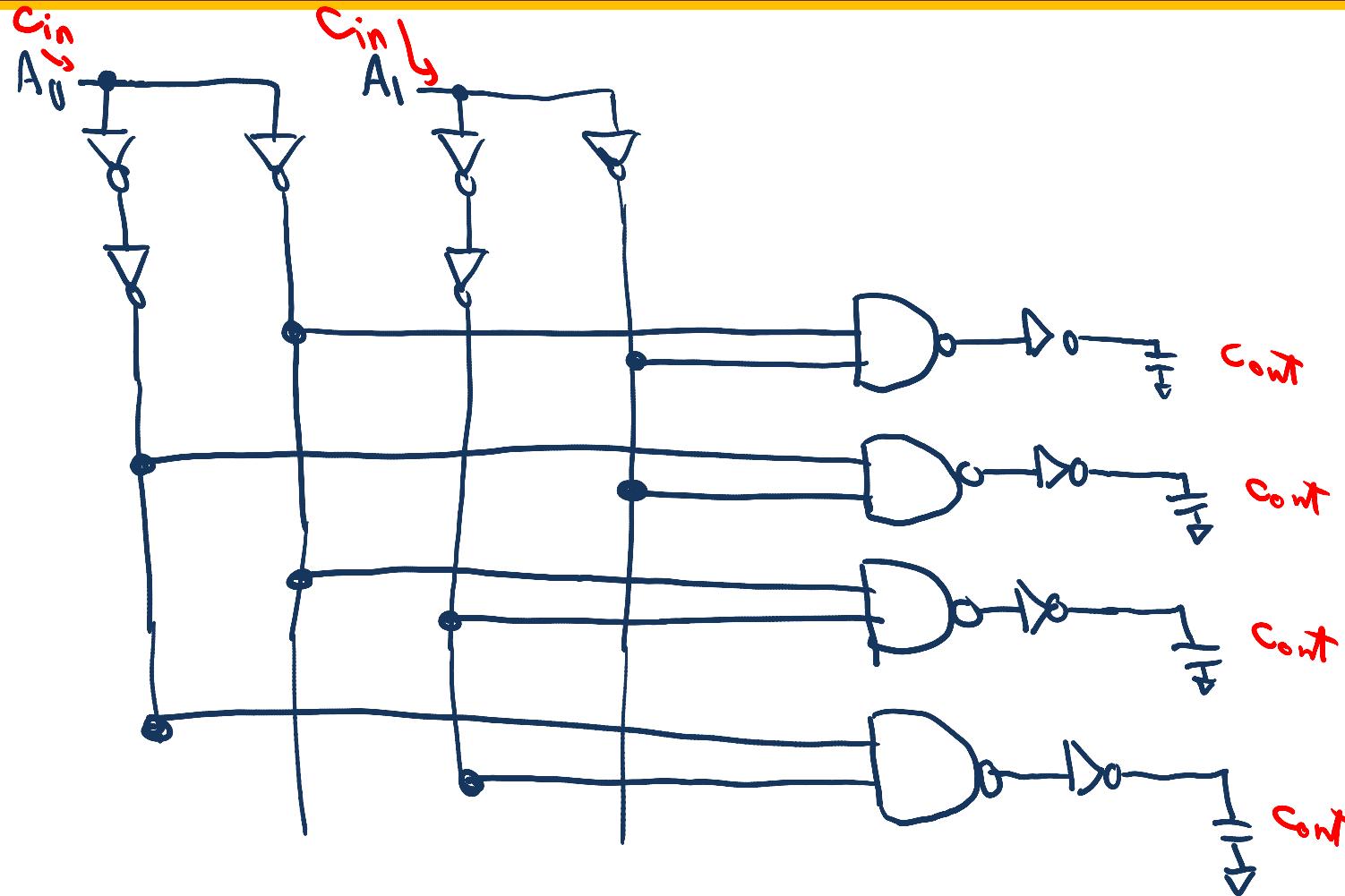
Example: Asymmetric Paths (Fork)



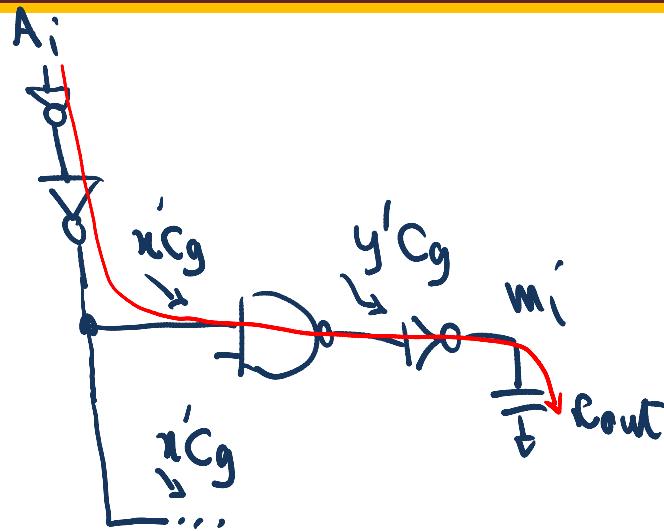
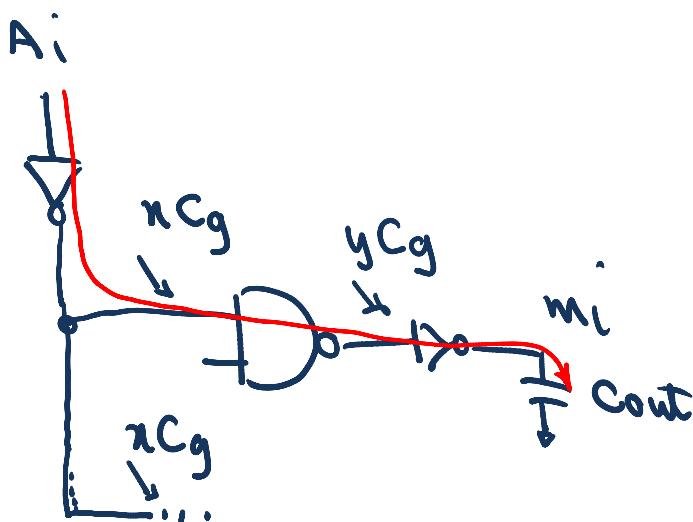
$$\text{Delay}_{A-C} = \frac{y}{x} + \frac{160}{y} + 2$$

$$\text{Delay}_{A-F} = \frac{v}{20-x} + \frac{w}{v} + \frac{160}{w} + 3$$

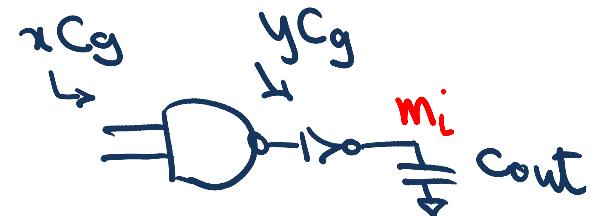
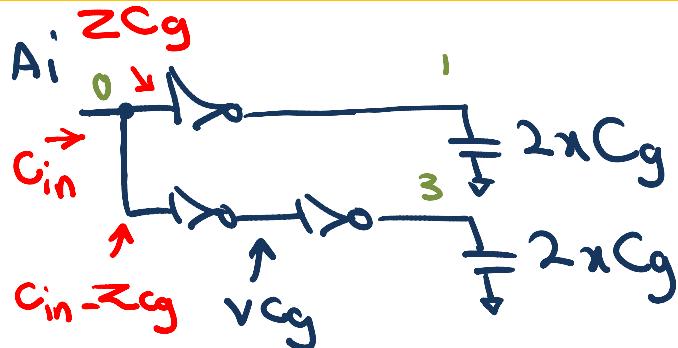
Example: Decoder



Example: Decoder (Cont.)



Example: Decoder (Cont.)



$$\text{Delay}_{01} = \frac{2x C_g}{z C_g} + 1$$

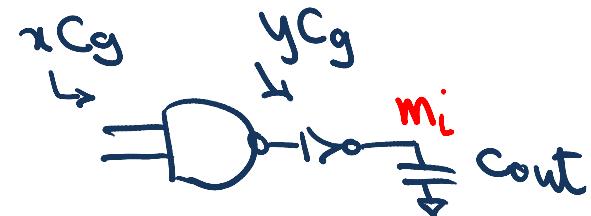
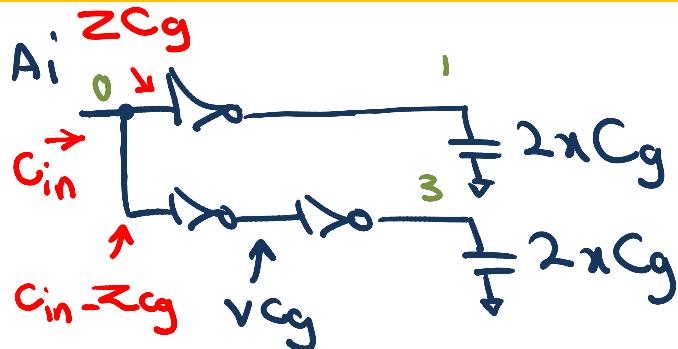
$$\text{Delay}_{03} = \frac{V_{cg}}{C_{in} - Z_{cg}} + \frac{2x C_g}{V_{cg}} + 2$$

$$\text{Delay}_{\text{fork}} = \max \{ \text{Delay}_{01}, \text{Delay}_{03} \}$$

$$\text{Delay}_{\text{decoder}} = \text{Delay}_{\text{fork}} + \text{Delay}_{\text{AND}}$$

$$\text{Delay}_{\text{AND}} = \frac{4}{3} \cdot \frac{V_{cg}}{x C_g} + \frac{C_{out}}{V_{cg}} + 3$$

Example: Decoder (Cont.)



$$\hat{\text{Delay}}_{\text{fork}} = \max \left\{ \hat{\text{Delay}}_{01}, \hat{\text{Delay}}_{03} \right\}$$

$$= \max \left\{ \left(\frac{2 \times C_g}{Z_{cg}} + 1 \right), 2 \left(\frac{2 \times C_g}{C_{in} - Z_{cg}} \right)^{1/2} + 2 \right\}$$

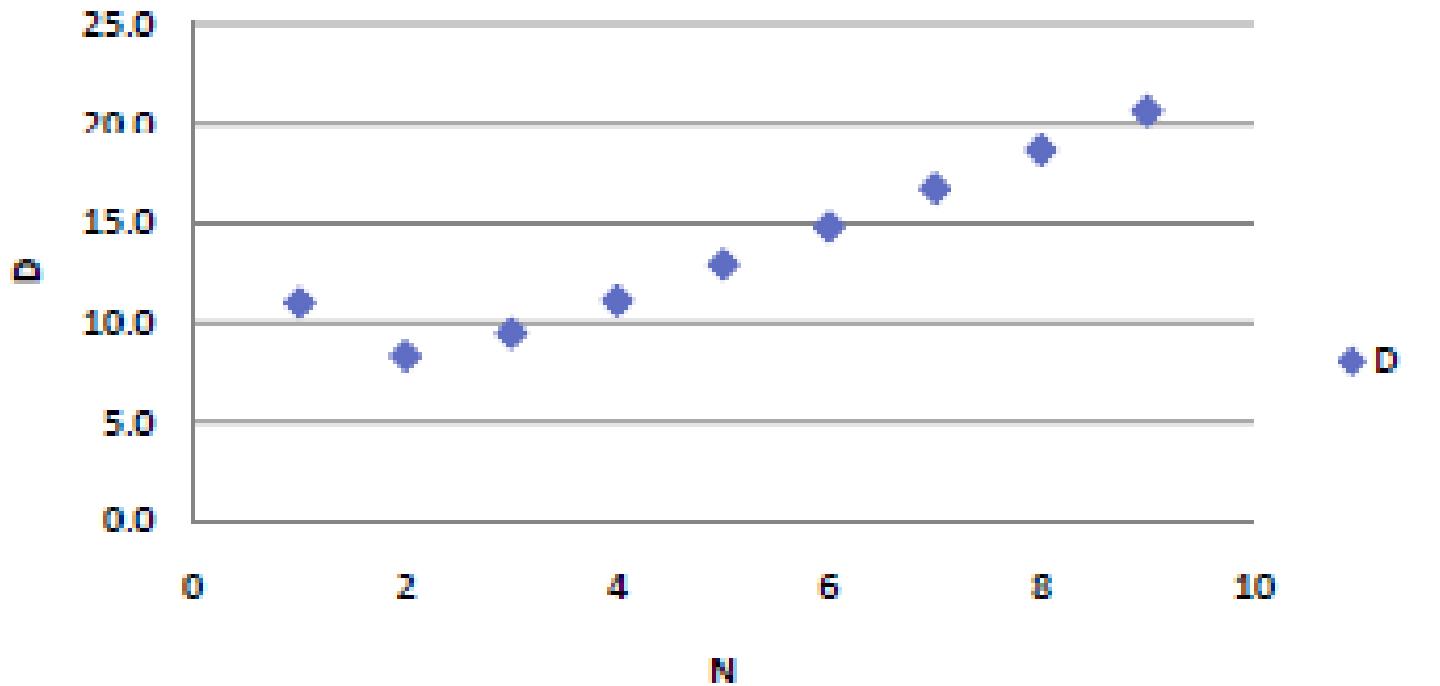
$$\hat{\text{Delay}}_{\text{AND}} = 2 \left(\frac{4}{3} \cdot \frac{C_{out}}{x_{cg}} \right)^{1/2} + 3$$

$$\hat{\text{Delay}}_{\text{decoder}} = \hat{\text{Delay}}_{\text{fork}} + \hat{\text{Delay}}_{\text{AND}}$$

Circuit Delay Optimization Formulation

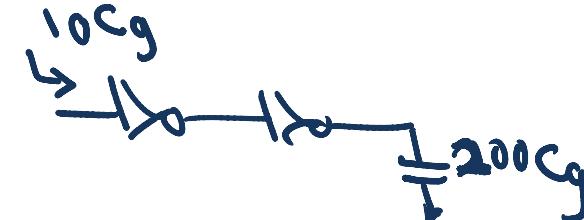
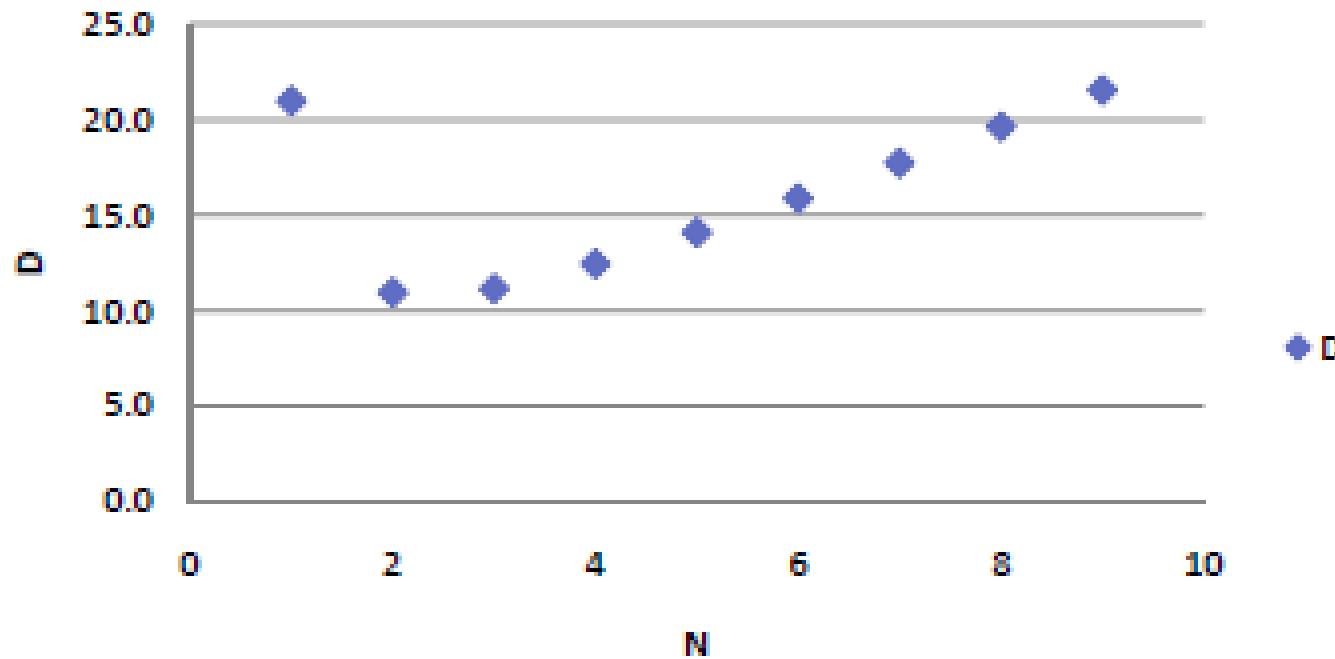
Delay vs the Number of Stages

D vs N for H=10

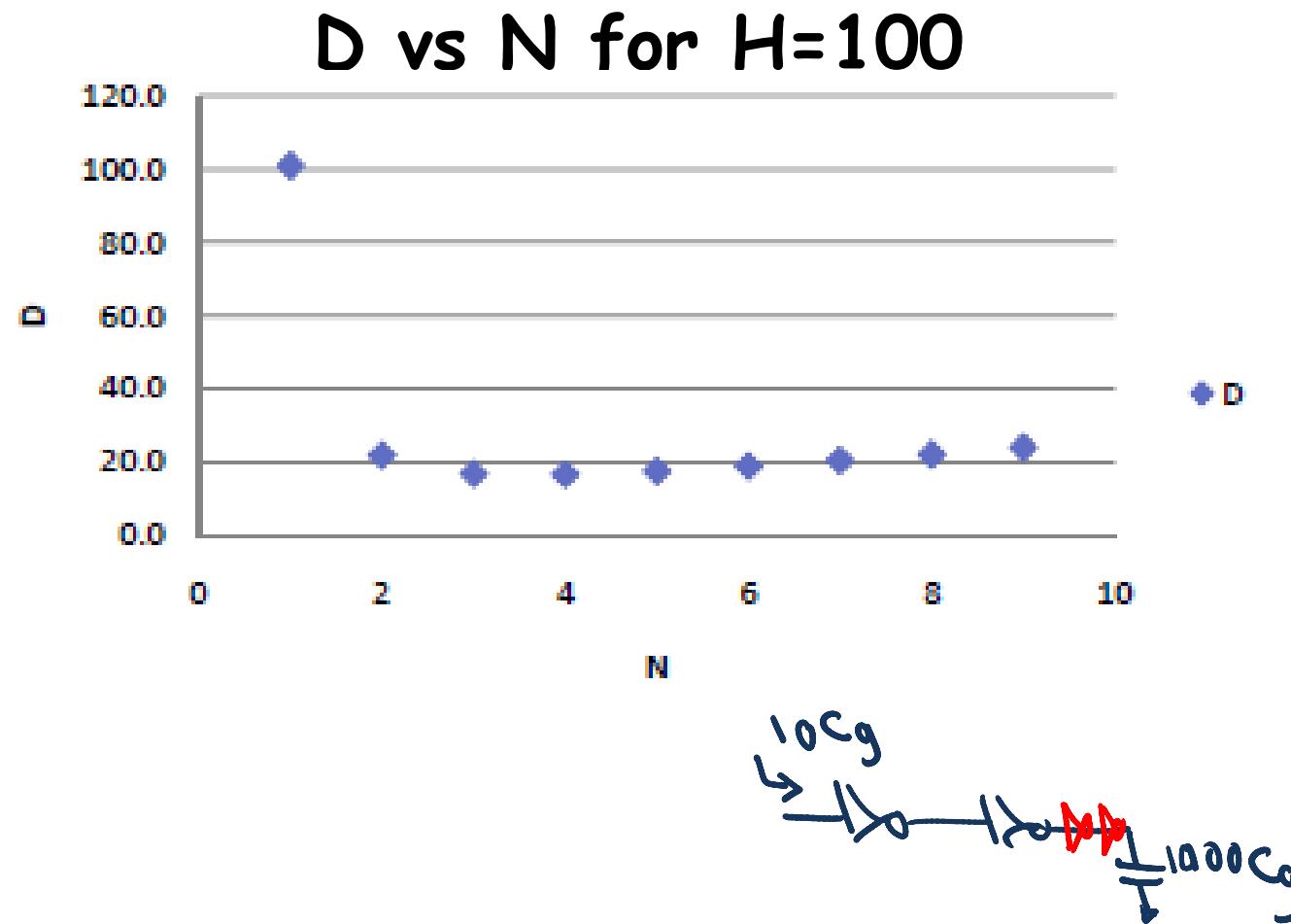


Delay vs the Number of Stages (Cont.)

D vs N for H=20



Delay vs the Number of Stages (Cont.)

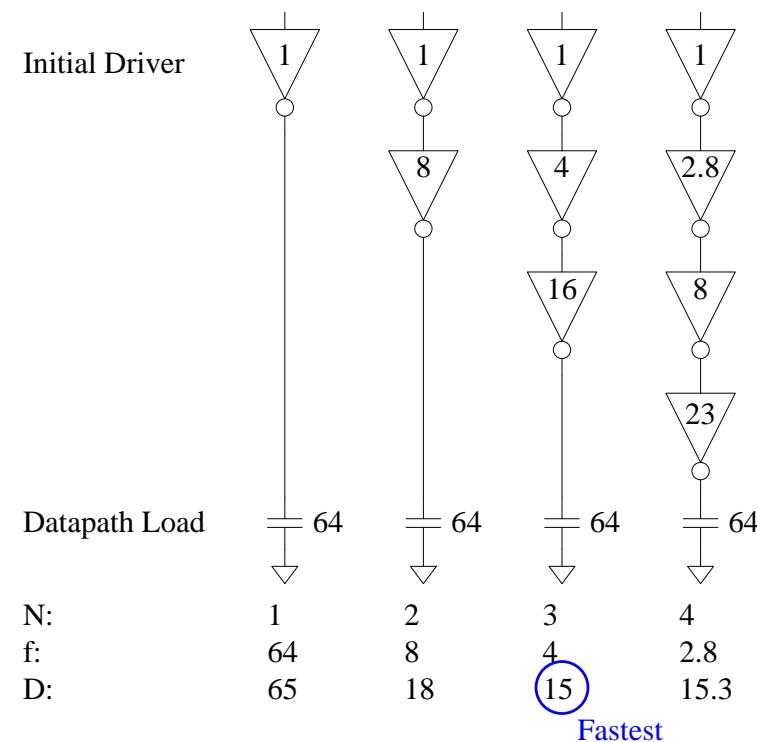


Example: Super Buffer Design

- How many stages should a path use?
 - Minimizing the number of stages would not always result in the fastest
- Example: drive 64-bit data path with unit inverter

$$F = GBH = 1 \times 1 \times 64$$

$$\begin{aligned}D &= NF^{1/N} + P \\&= N(64)^{1/N} + N\end{aligned}$$



Example: Super Buffer Design (Cont.)

- **1 Stage**

$$\hat{f} = F = 64$$

$$d = 1 * 64 + 1 * 1 = 65$$

- **2 Stages**

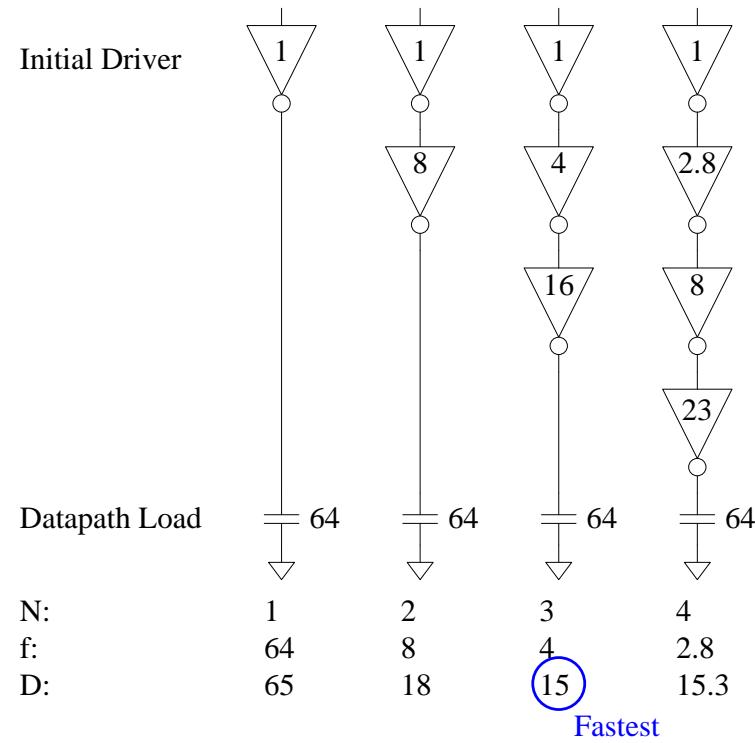
$$\hat{f} = \sqrt[2]{F} = \sqrt[2]{64} = 8$$

$$d = 2 * 8 + 2 * 1 = 18$$

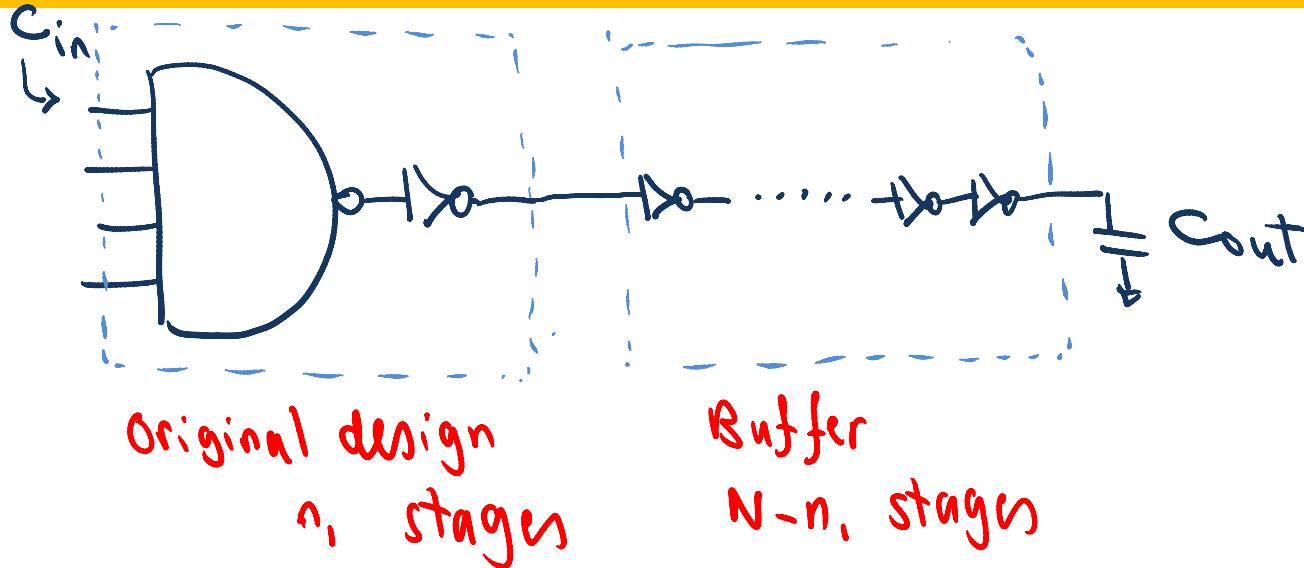
- **3 Stages**

$$\hat{f} = \sqrt[3]{F} = \sqrt[3]{64} = 4$$

$$d = 3 * 4 + 3 * 1 = 15$$



Optimal Number of Stages



$$G_{orig}$$

$$P_{orig}$$

$$H_{orig} = \frac{C_{out}}{C_{in}}$$

$$G_{buf} = 1$$

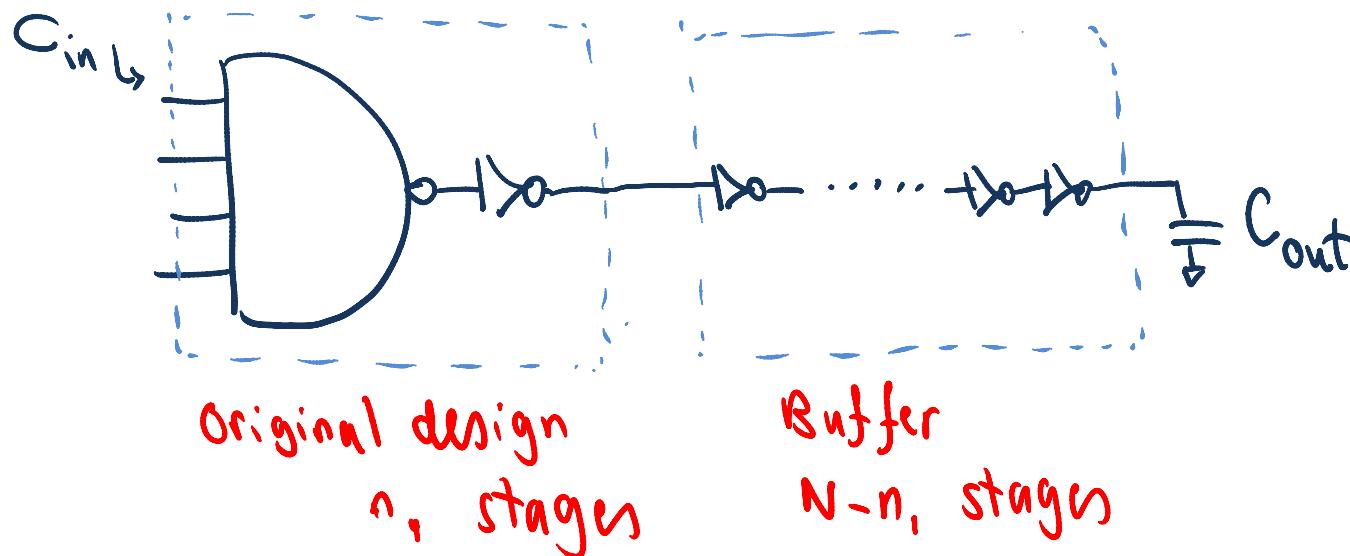
$$P_{buf} = (N - \eta_1) P_{inv}$$

$$G_{tot} = G_{orig}$$

$$P_{tot} = P_{orig} + (N - \eta_1) P_{inv}$$

$$H_{tot} = H_{orig}$$

Optimal Number of Stages (Cont.)



$$\hat{D} = N(G_{orig} H_{orig})^{\frac{1}{N}} + P_{orig} + (N-n)P_{inv}$$

$$\frac{\partial \hat{D}}{\partial N} = 0 \Rightarrow f(1-\ln f) + P_{inv} = 0$$

where $P = (G_{orig} H_{orig})^{\frac{1}{N}}$

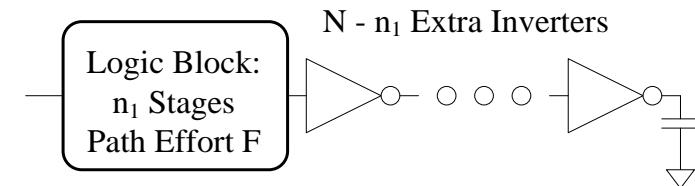
solving numerically $\Rightarrow (G_{orig} H_{orig})^{\frac{1}{N}} \approx 4$

$$\Rightarrow \hat{N} \approx \frac{\ln(G_{orig} H_{orig})}{\ln 4}$$

Optimal Number of Stages (Cont.)

- Consider adding inverters to the end of a path with n_1 stages
 - How many inverters give the least delay?

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



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

- Define *best stage effort*:
- Then $p_{inv} + \rho(1 - \ln \rho) = 0$

Recall differentiation rules:

$$\rho = F^{\frac{1}{N}}$$

$$(fg)' = f'g + fg'$$

$$(f^g)' = (e^{g \ln f})' = f^g \left(f' \frac{g}{f} + g' \ln f \right)$$

Optimal Number of Stages (Cont.)

- $p_{inv} + \rho(1 - \ln \rho) = 0$ has no closed-form solution
- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = 2.718$ (e)
- For $p_{inv} = 1$, solve numerically to get $\rho = 3.59$
- Should we add additional inverters at the beginning or at the end ?

Method of Logical Effort (Complete Version)

- Compute the path effort: $F = GBH$
- Find the best number of stages $N \sim \log_4 F$
- Compute the stage effort $f = F^{1/N}$
- Sketch the path with this number of stages
- Work from either end, find sizes:
 $C_{in} = C_{out}^* g/f$

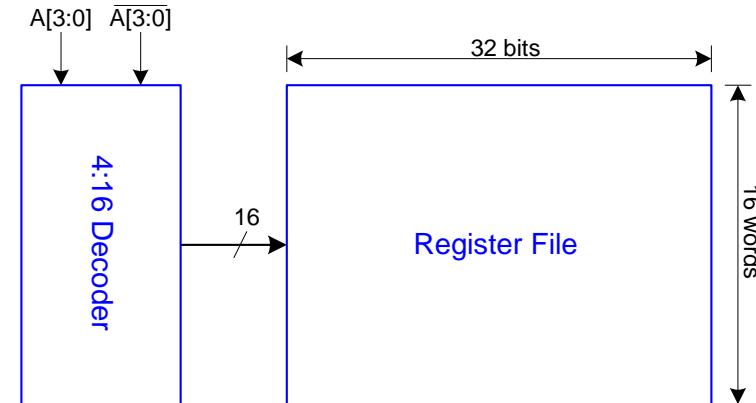
Example: 4-to-16 Decoder Design

- 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

- Use LE to decide:

- How many stages to use?
- How large should each gate be?
- How fast can decoder operate?



Example: 4-to-16 Decoder Design

- Decoder effort is mainly electrical and branching

Electrical Effort: $H = (32 * 3) / 10 = 9.6$

Branching Effort: $B = 8$

- If we neglect logical effort (assume $G = 1$)

Path Effort: $F = GBH = 76.8$

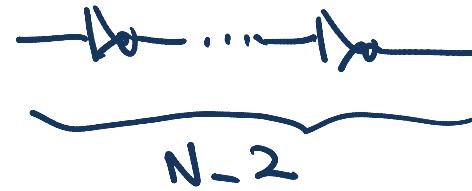
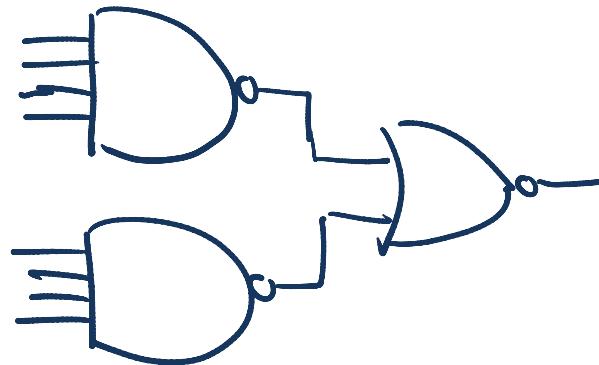
Number of Stages: $N = \log_4 F = 3.1$

Try a 3-stage design

$$\begin{aligned}\hat{f} &= F^{\frac{1}{N}} \\ &= \sqrt[3]{76.8} = 4.3\end{aligned}$$

Example: AND8 - Optimal Stage Number

- $H=1024$



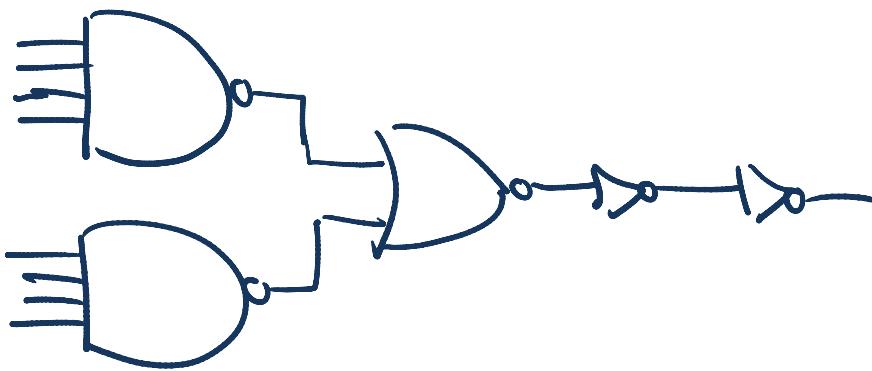
$$G = \frac{6}{3} \times \frac{5}{3} \times 1 \times \dots \times 1$$

$$B = 1, H = 1024$$

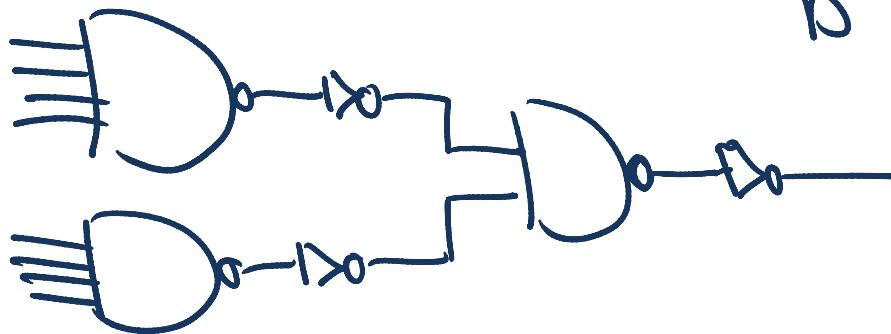
$$\hat{O} = 2\sqrt{\frac{10}{3} \times 1024} + 6 \approx 122.8$$

$$\hat{O} = N \sqrt[N]{\frac{10}{3} \times 1024 + 6 + (N-2)}$$

Example: AND8 (Cont.)

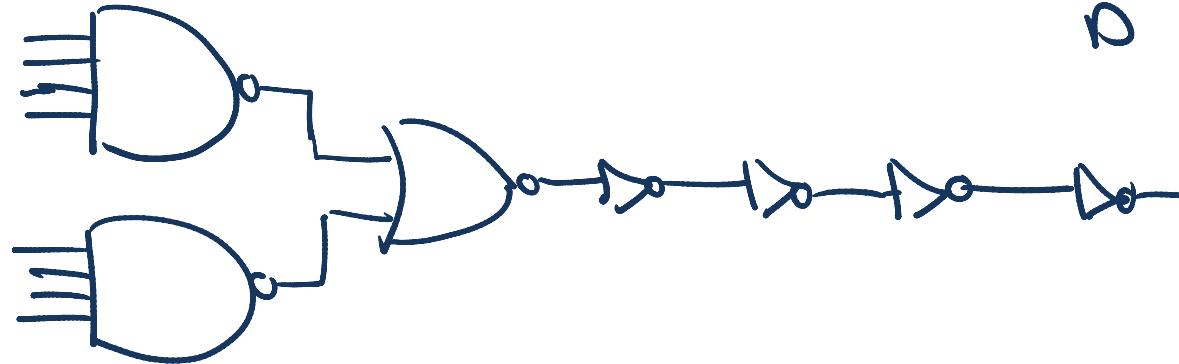


$$\hat{D} = 4 \sqrt[4]{\frac{1}{3} \cdot 1024} + 8 \approx 38.6$$

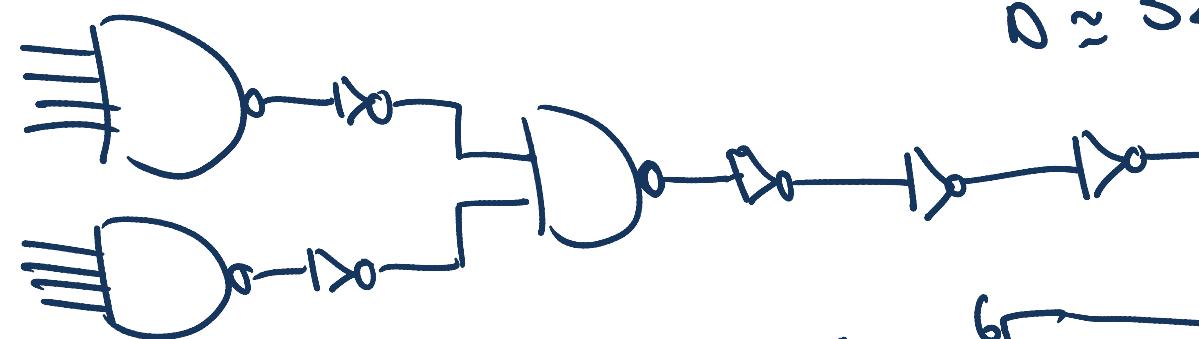


$$\hat{D} = 4 \sqrt[4]{\frac{8}{3} \times 1024} + 8 \approx 36.9$$

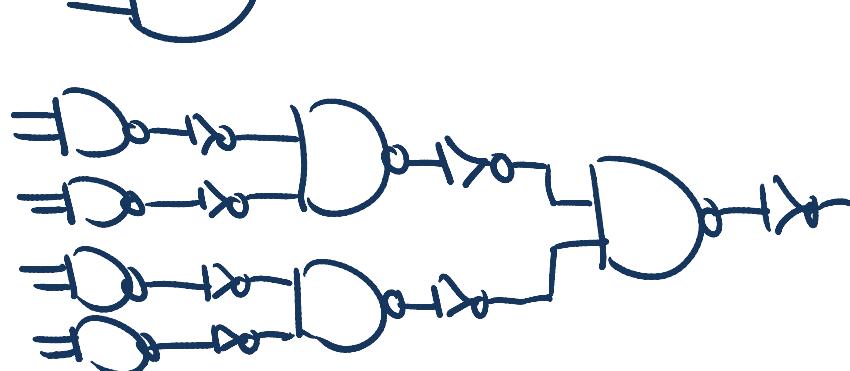
Example: AND8 (Cont.)



$$\hat{D} \approx 33.3$$



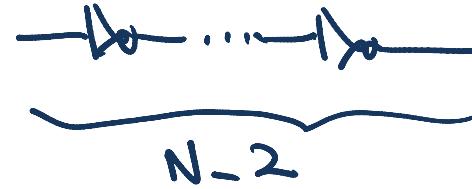
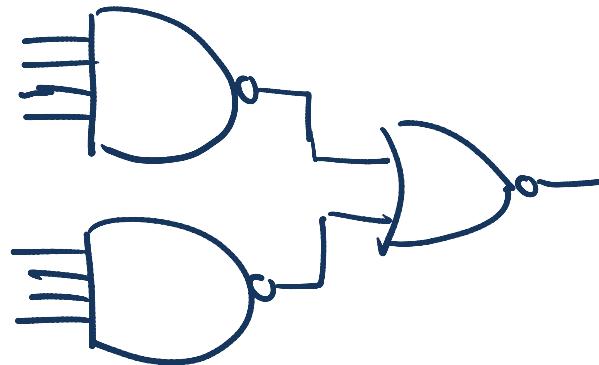
$$\hat{D} \approx 32.4$$



$$\hat{D} = 6 \sqrt[6]{\frac{4}{3} \times \frac{4}{3} \times \frac{4}{3} \times 1024 + 9} \approx 31$$

Example: AND8 - Optimal Stage Number

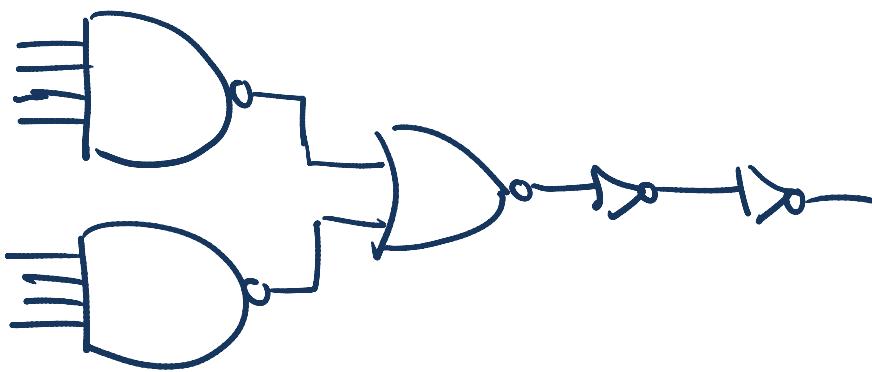
- $H=512$



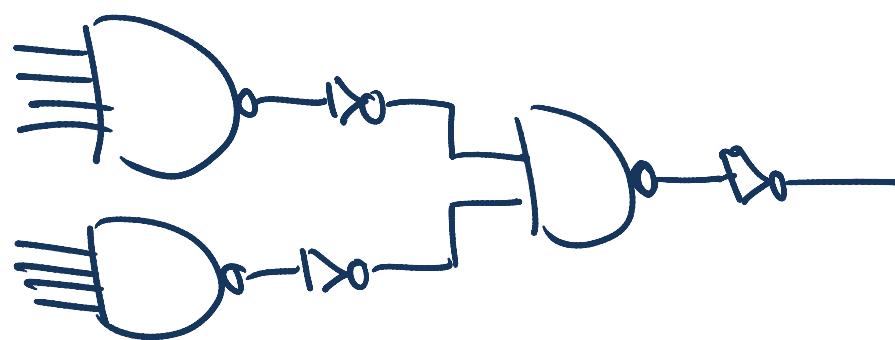
$$G = \frac{6}{3} \times \frac{5}{3} \times 1 \times \dots \times 1$$
$$B = 1, H = 512$$

$$\hat{D} \approx 88.6$$

Example: AND8 (Cont.)

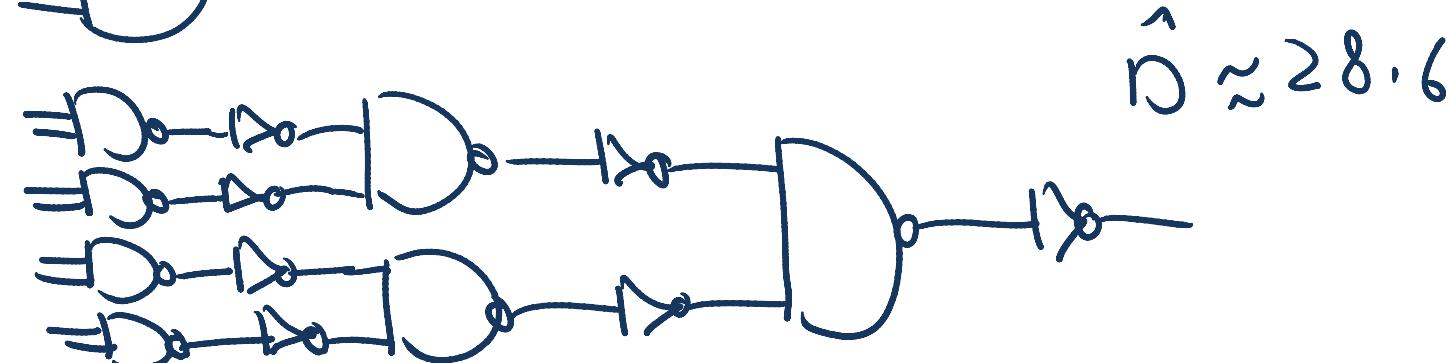
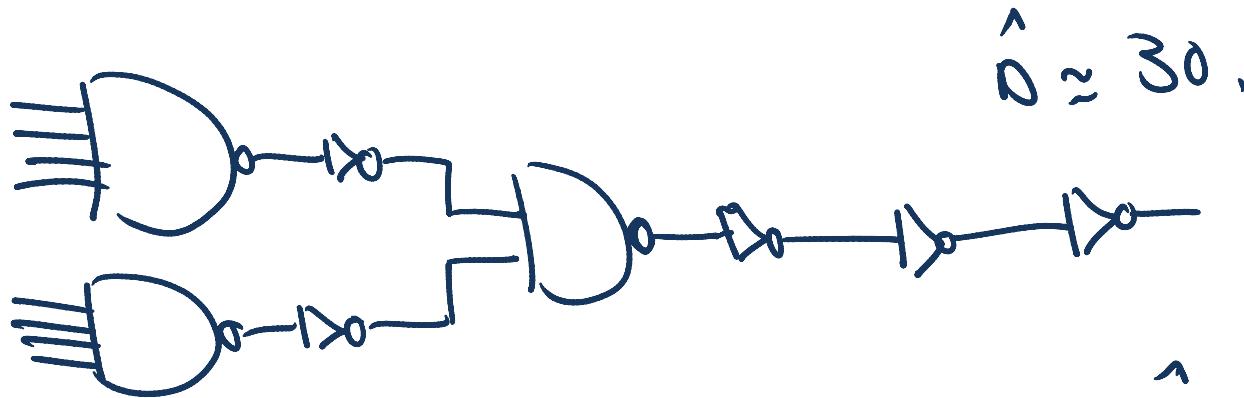
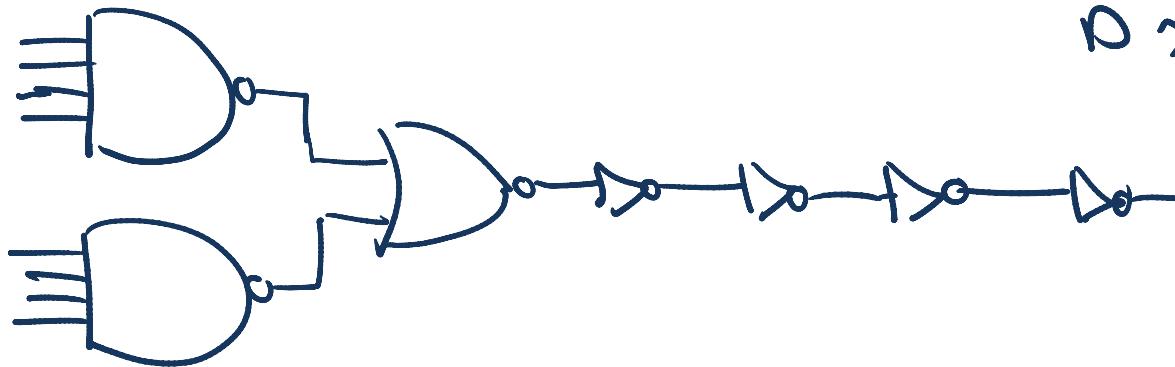


$$\hat{D} \approx 33.7$$

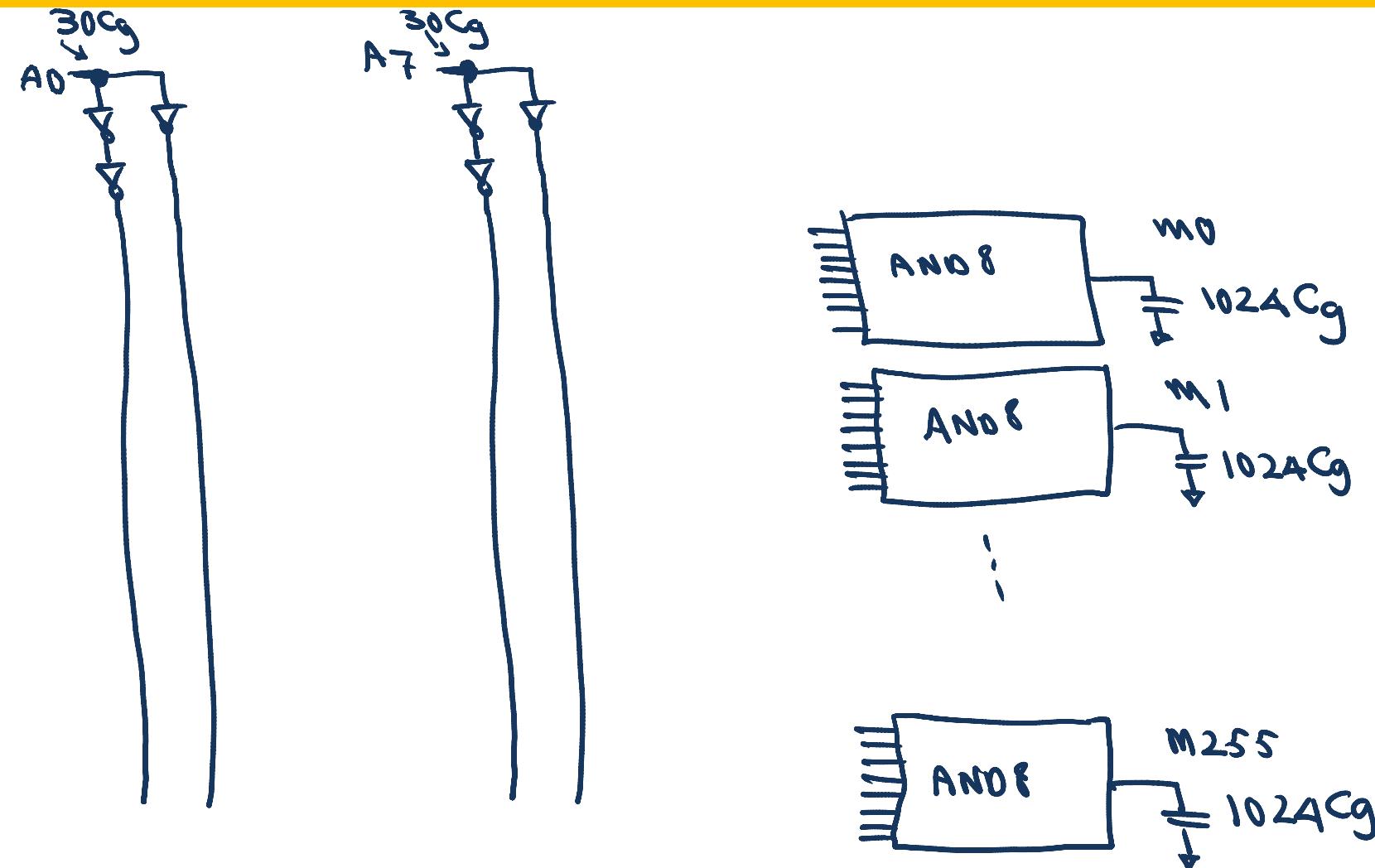


$$\hat{D} \approx 32.3$$

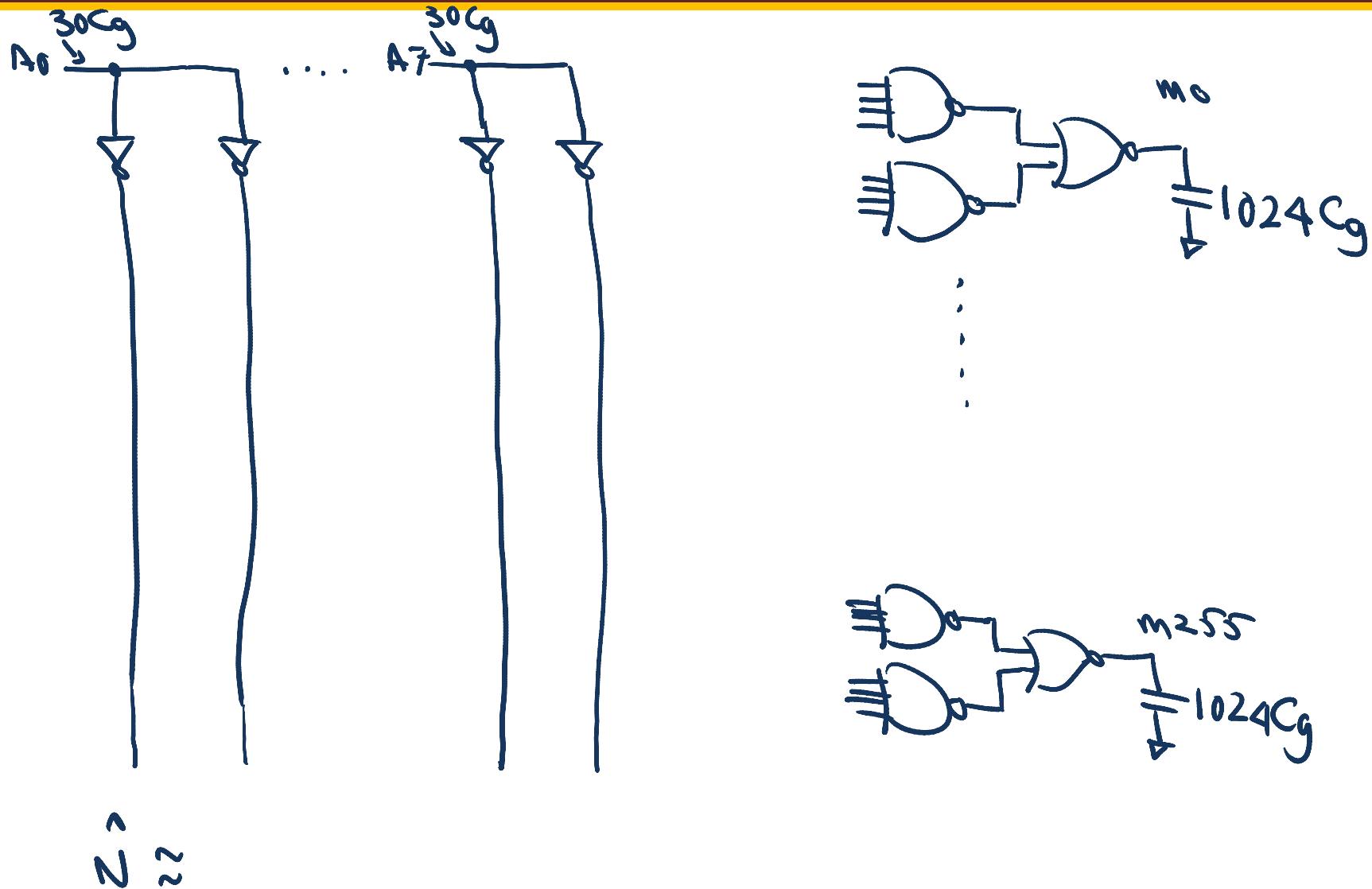
Example: AND8 (Cont.)



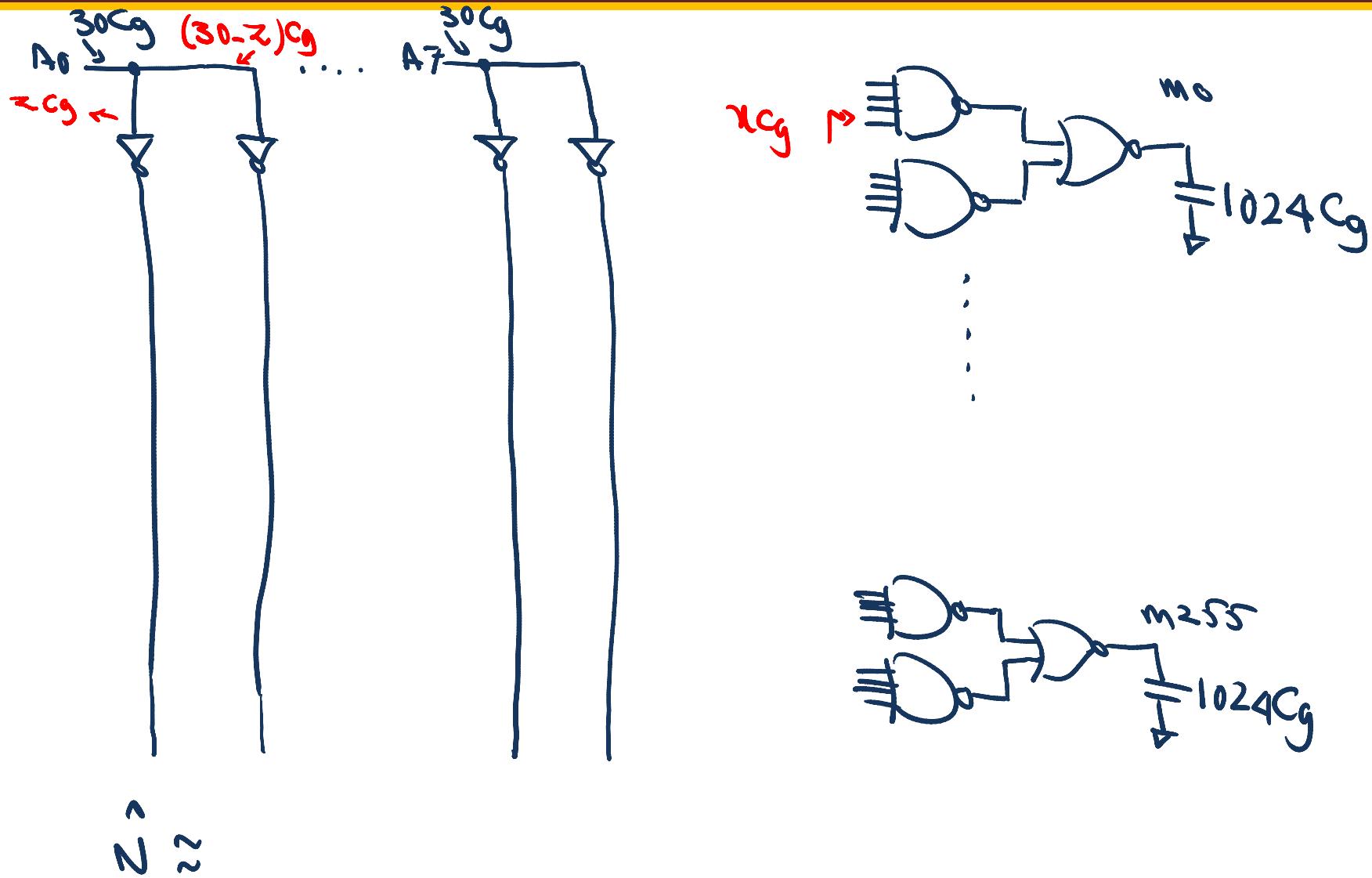
Example: 8-to-256 Decoder



Example: 8-to-256 Decoder (Cont.)

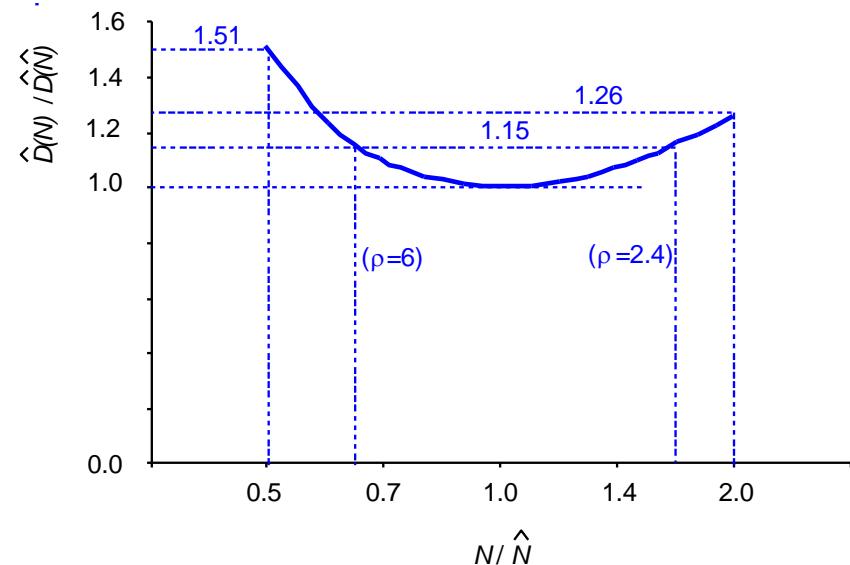


Example: 8-to-256 Decoder (Cont.)



Review of Delay Sensitivity Analysis

- How sensitive is delay to using exactly the best number of stages?

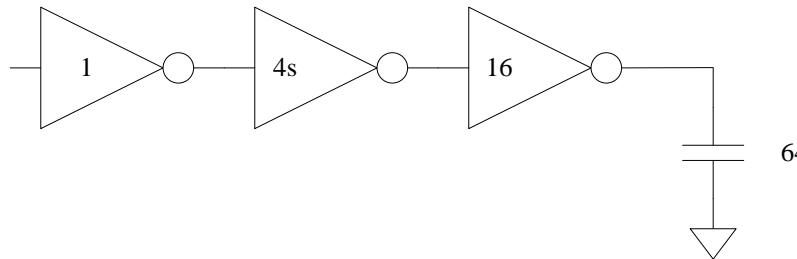


- $2.4 < \rho < 6$ in super buffer design gives delay within 15% of the optimal
 - We can be sloppy!
 - Let's use $\rho = 4$ $\rightarrow N = \log_4 F$

Using the Wrong Stage Effort and Number of Stages

- If effort is between 2 and 8, the design will be within 35% of the best delay
- If effort is between 2.4 and 6, the design will be within 15% of the best delay
 - Stage effort of 4 produces within 2% of minimum
- Avoid using excessively large stage effort since it results in slow rise and fall times and “hot electron” problems occur; and greatest damage occurs to NMOS in saturation
 - Better use too many stages than too few

Using the Wrong Gate Size



- For s ranging from 0.5 to 2, the actual delay is within 15% of the minimum
- For s in the range of $2/3$ to 1.5, the actual delay is within 5% of the minimum
 - The designer has great deal of freedom to select gate sizes
- Standard cell libraries with limited repertoire of gate sizes can achieve acceptable performance

