
EE3230 Lecture 4: Circuit Characterization and Performance Estimation I

Ping-Hsuan Hsieh (謝秉璇)

Delta Building R908

EXT 42590

pshieh@ee.nthu.edu.tw

Outline

- **Delay estimation**
- Logical effort and transistor sizing
- Power dissipation
- Interconnect
- Wire engineering
- Design margin
- Reliability
- Scaling

Transient Response

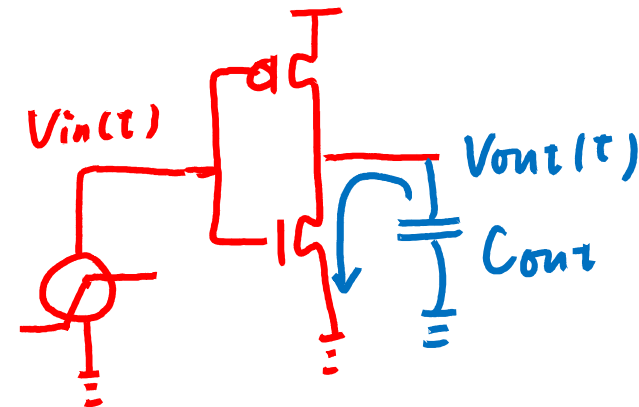
- **DC** analysis tells us V_{out} if V_{in} is constant
- **Transient** analysis tells us $V_{\text{out}}(t)$ with certain $V_{\text{in}}(t)$
 - Requires solving differential equations
- Input is usually considered to be a step or ramp
 - From 0 to V_{DD} or vice versa

Inverter Step Response $V_{in} \rightarrow V_{out}$

- With load capacitance of C_{load}

$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t_0^-) = V_{DD}$$



- Current discharging the cap

@ t_0^+ PMOS turned OFF

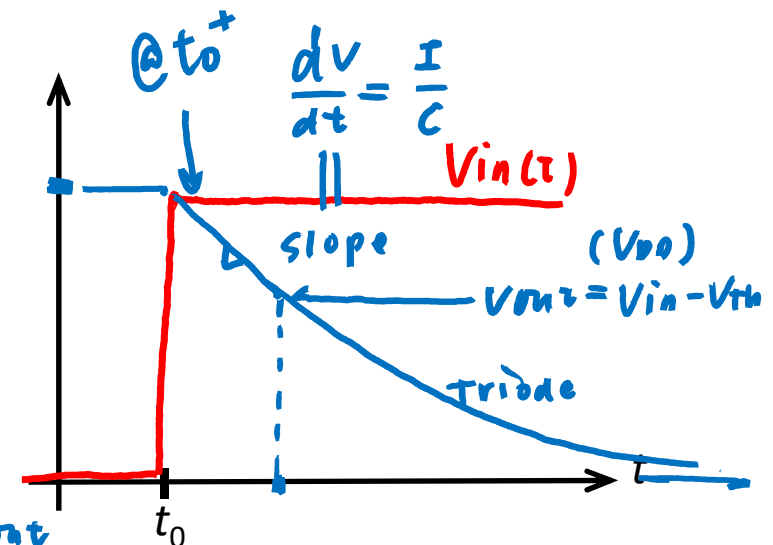
NMOS turned ON

initially in saturation

$$I_{dsn}(t) = \frac{1}{2} \mu_n \frac{W}{L} (V_{DD} - V_{th})^2$$

$\approx V_{out} < V_{DD} - V_{th}$, NMOS in triode

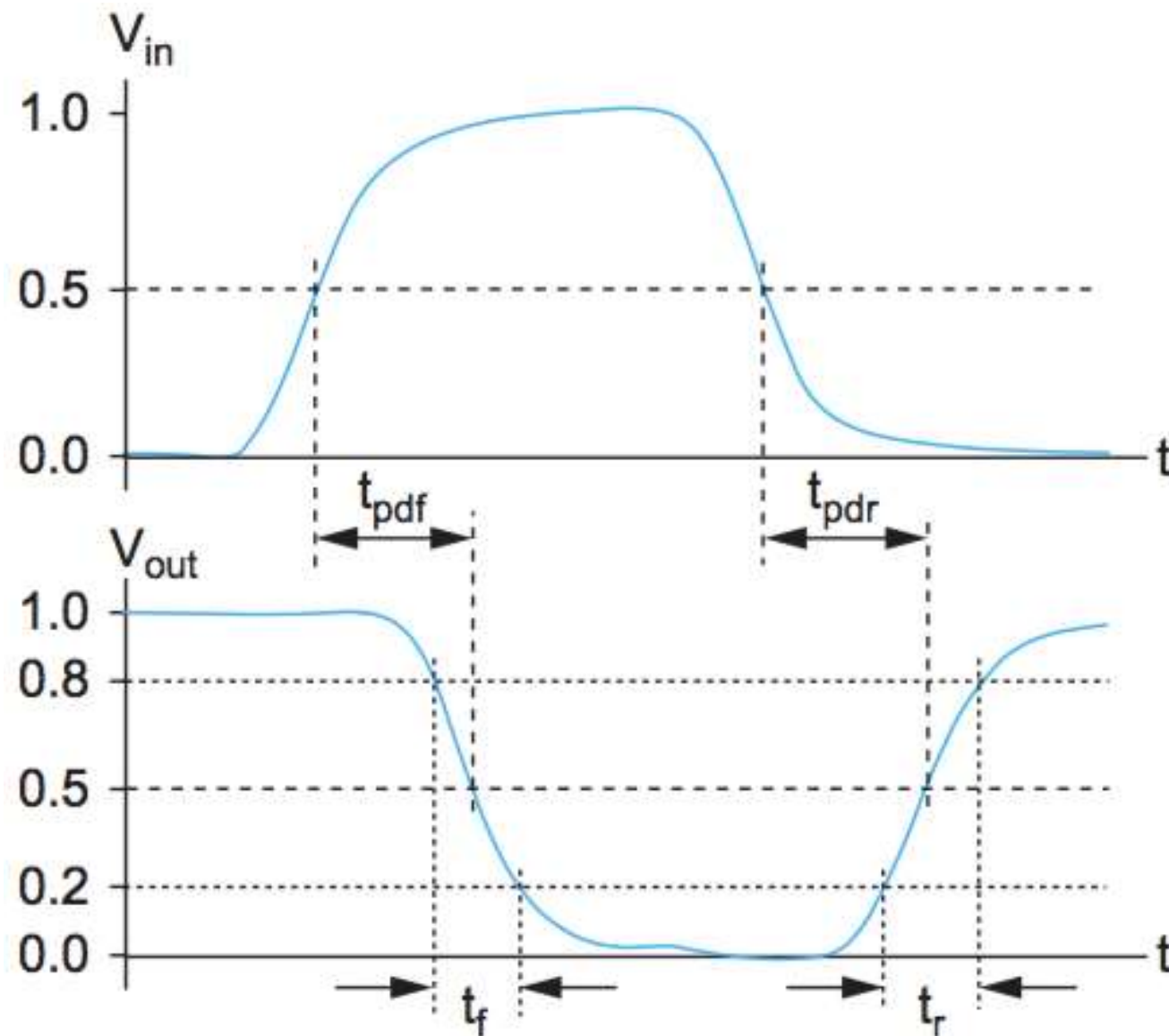
$$I_{dsn}(t) = \mu_n \frac{W}{L} (V_{DD} - \frac{1}{2} V_{out}) \cdot V_{out}$$



Delay Definitions (I)

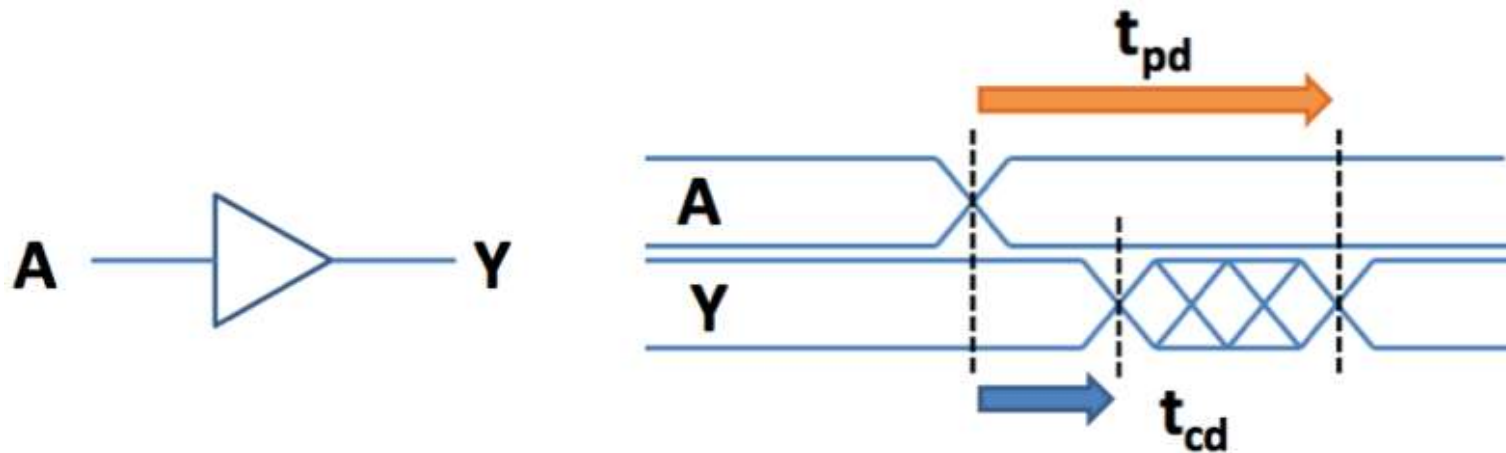
- t_{pdr} maximum ^{PMOS} rising propagation delay
 - From input to rising output crossing $V_{DD}/2$
- t_{pdf} maximum ^{NMOS} falling propagation delay
 - From input to falling output crossing $V_{DD}/2$
- t_{pd} average propagation delay
 - $t_{pd} = (t_{pdr} + t_{pdf})/2$
- t_r rise time
 - For output to go from $0.2 V_{DD}$ to $0.8 V_{DD}$
- t_f fall time
 - For output to go from $0.8 V_{DD}$ to $0.2 V_{DD}$

Delay Definitions (II)



Delay Definitions (III)

- t_{cdr} ^{minimum} rising contamination delay
 - From input to rising output crossing $V_{DD}/2$
- t_{cdf} maximum falling contamination delay
 - From input to falling output crossing $V_{DD}/2$
- t_{cd} average contamination delay
 - $t_{cd} = (t_{cdr} + t_{cdf})/2$



Delay Estimation (I)

- Estimate delay easily
 - Not as accurate as simulations
 - Easier to ask “what if?”
- Step response looks like a 1st order RC response (decaying exponential)
- Use RC delay models
 - C = total capacitance on output node
 - Use effective R
 - $t_{pd} = RC$
- Characterize transistors by finding their effective **R**
 - Depend on average current of gate switches

Delay Estimation (II)

- **Critical path**

- The signal path with the slowest (most critical) timing
- Affected at 4 different levels

- Architecture/micro-architecture levels

- Tradeoff of pipeline stages, number of execution units, and size of memory. It's the level that impacts the most.

- Logic level

- Tradeoff of functional block types, number of gate in the cycle, fan-in and fan-out number

- Circuit level

- Transistor size and logic styles/families

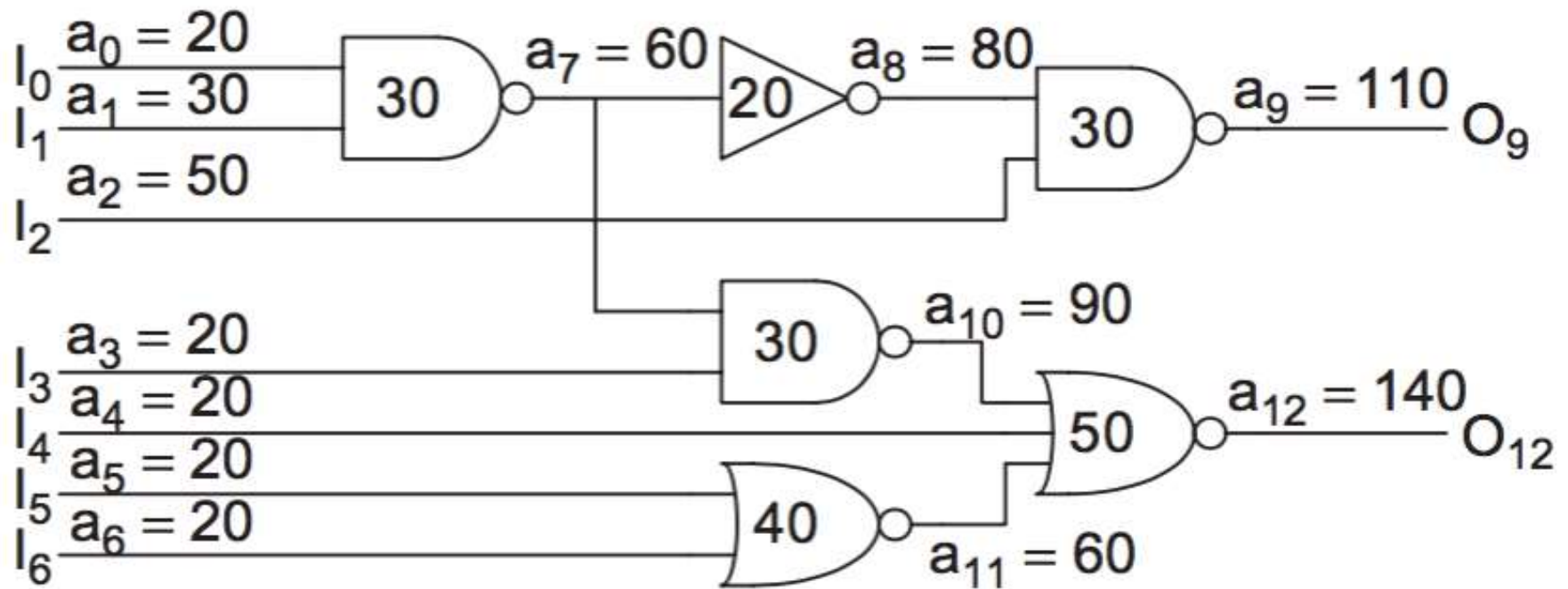
CMOS (PMOS NMOS)

小心不要卡在這裡
implementation

- Layout level

- Floor-plan, wire length, and parasitics

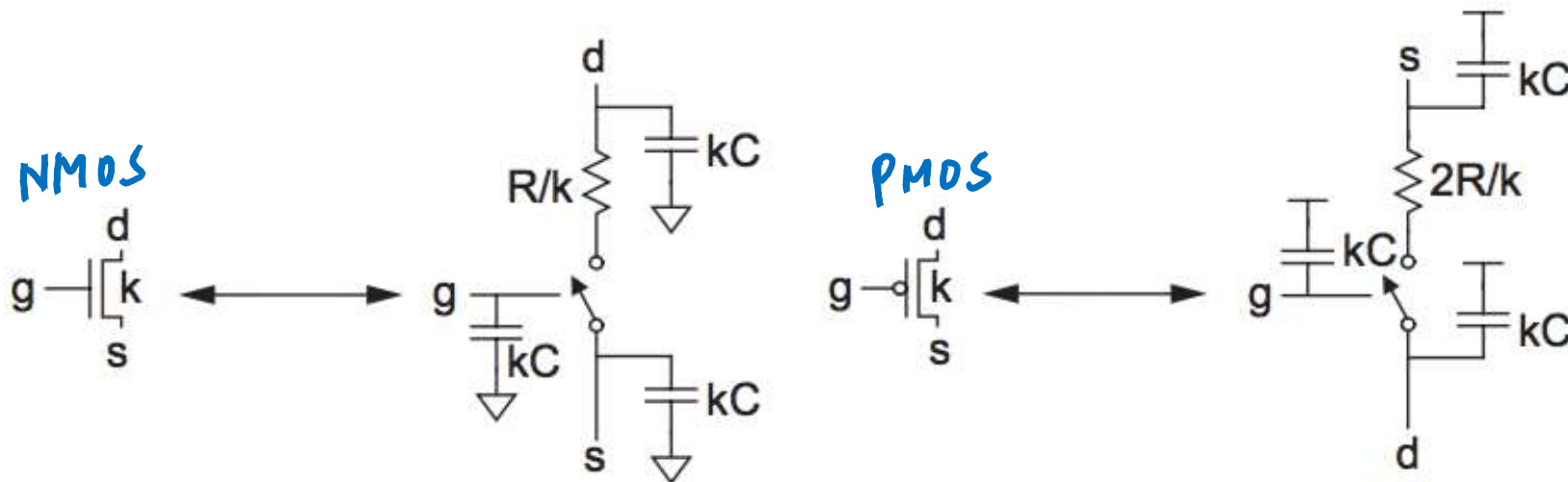
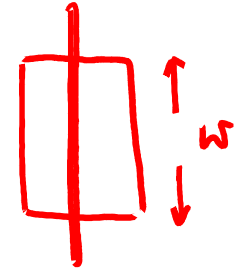
Critical Path



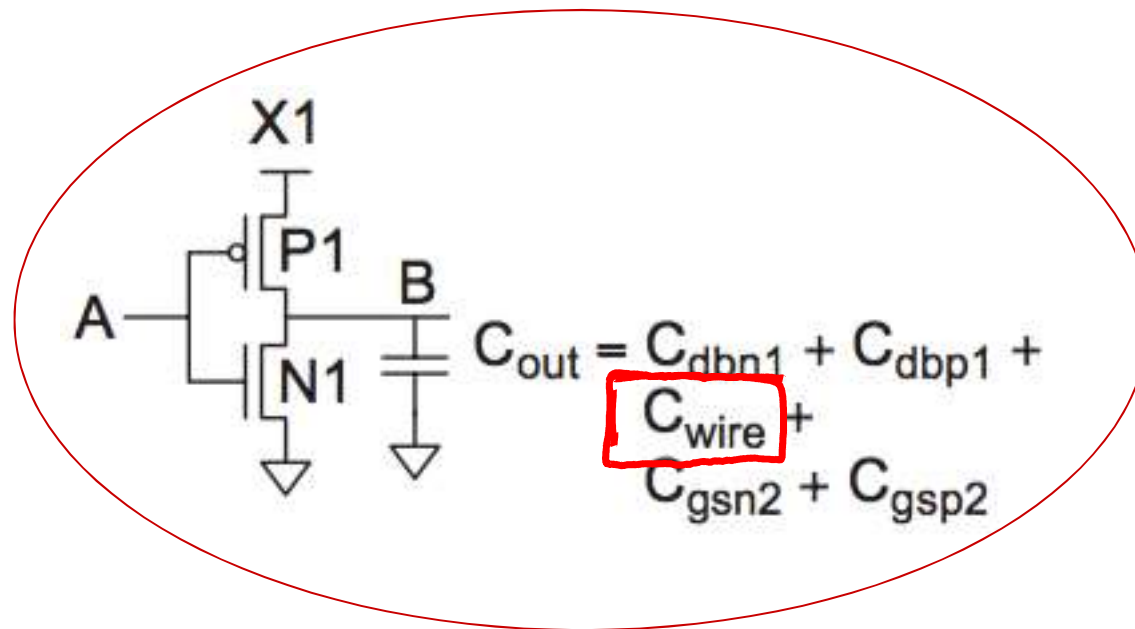
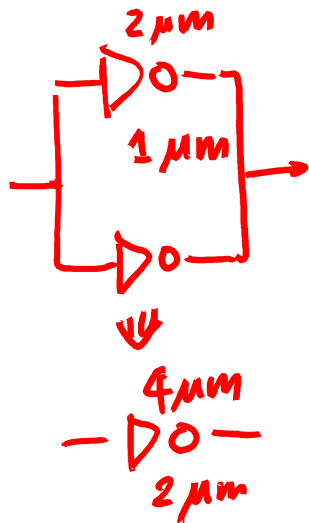
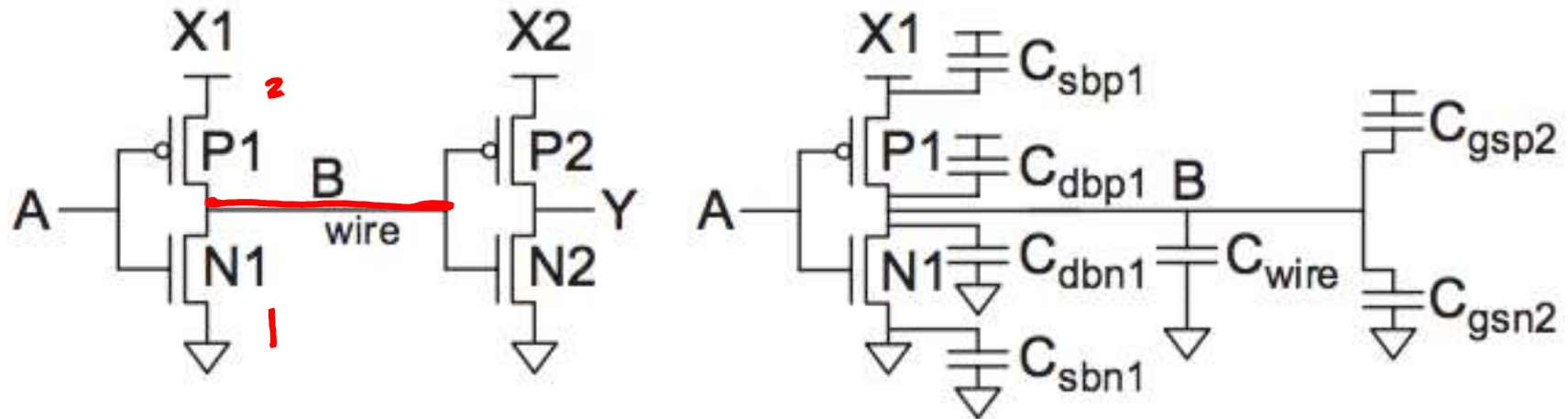
RC Delay Models

channel length = L_{min}

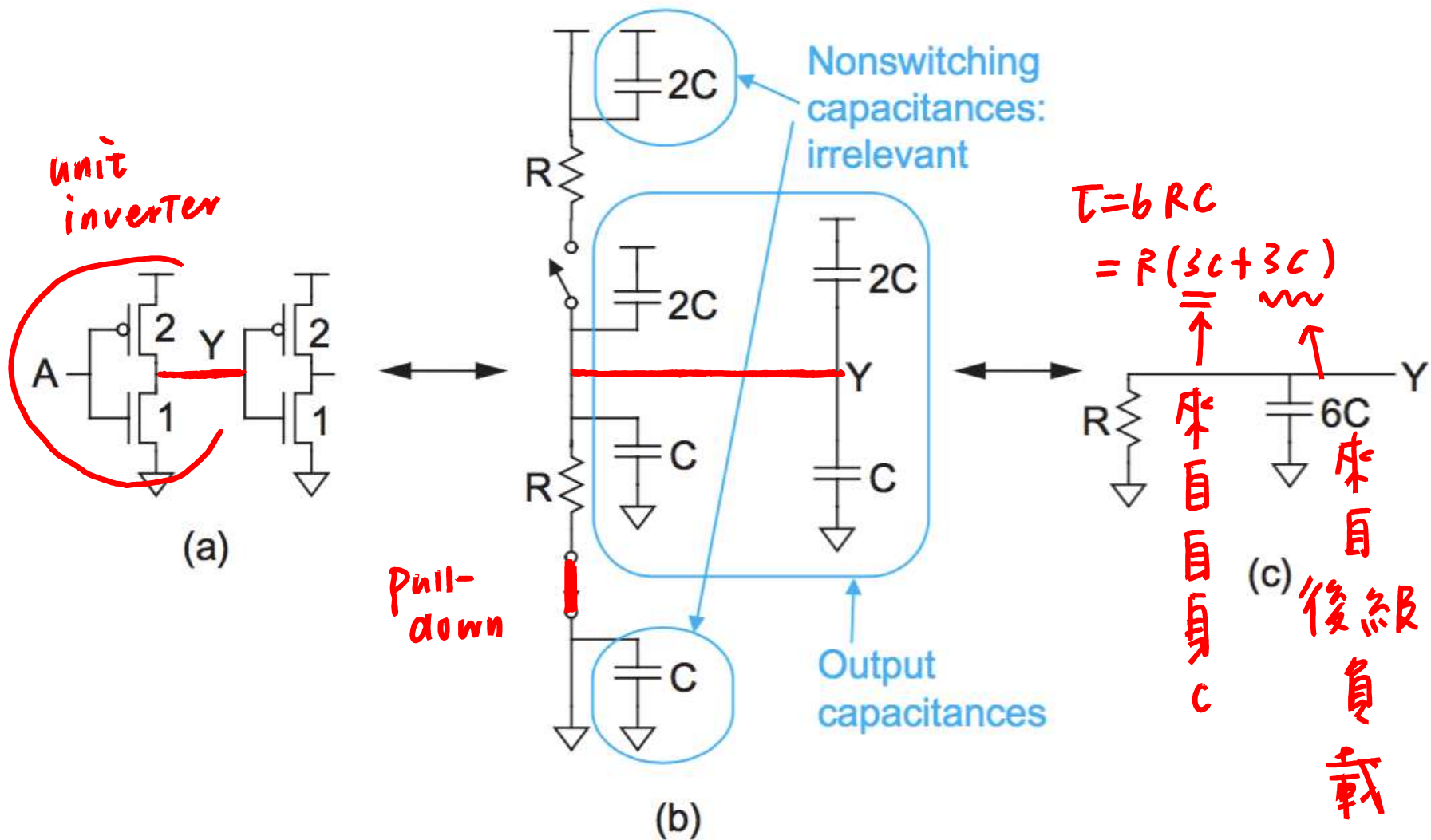
- Equivalent circuit for MOS transistors
 - Ideal switch + capacitance and ON resistance
 - Unit NMOS has resistance R and capacitance C
 - Unit PMOS has resistance $2R$ and capacitance C
- Capacitance proportional to width
- Resistance inversely proportional to width



Example: Inverter

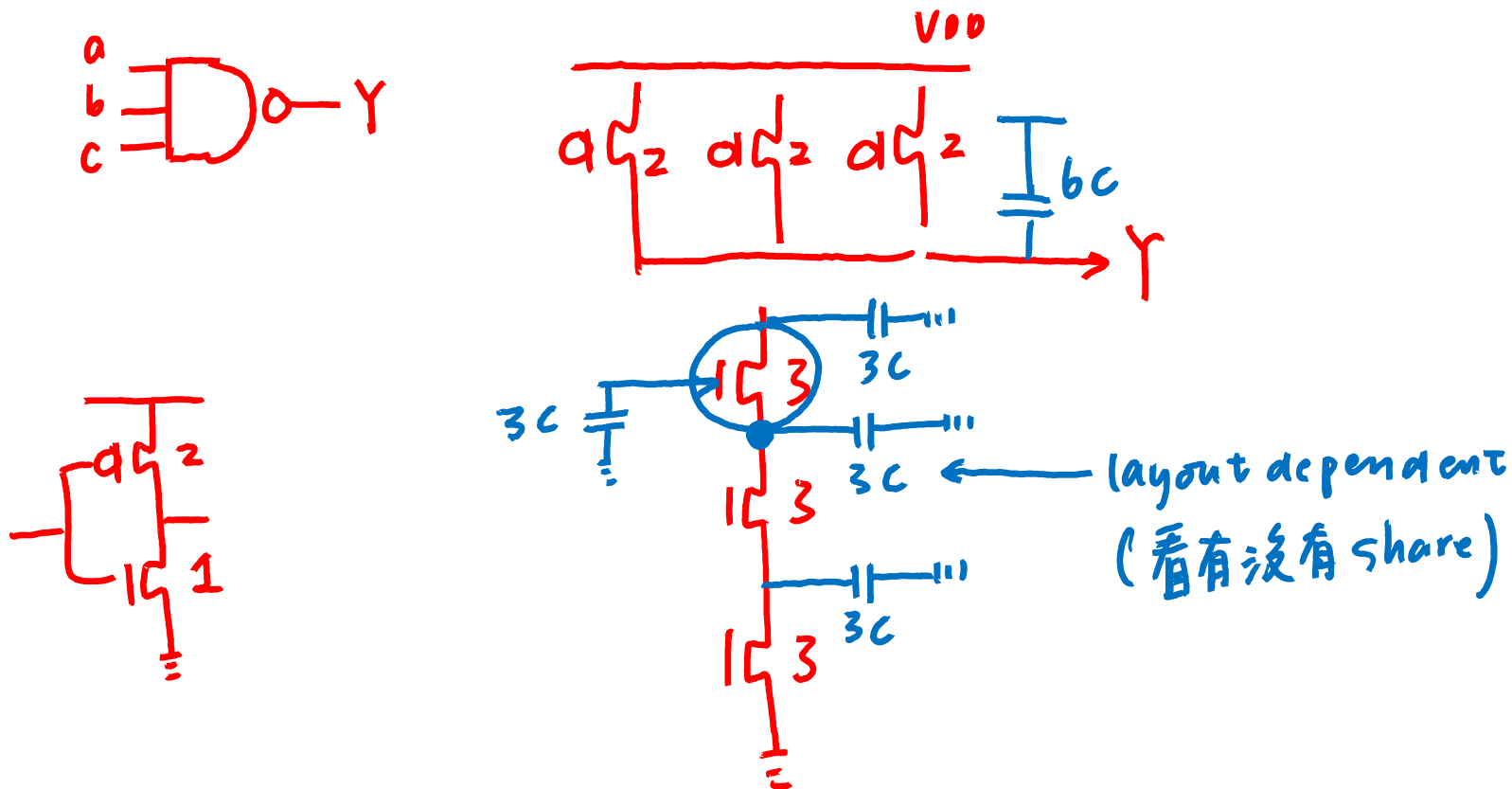


Example: Inverter



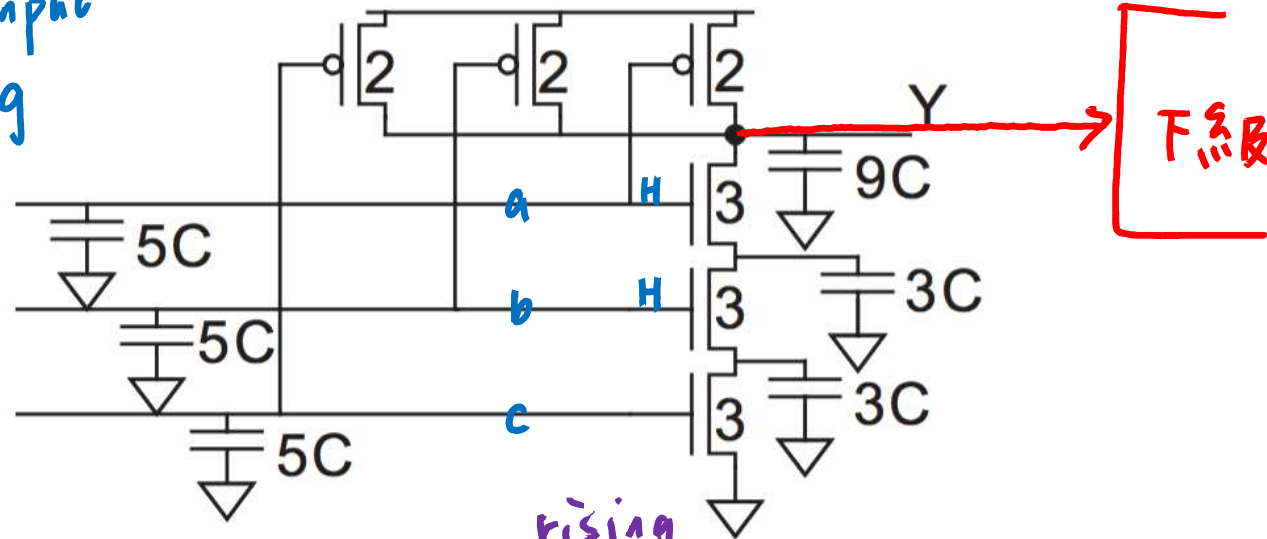
Example: NAND3 L_{min}

- Sketch a 3-input NAND with transistor widths chosen to achieve effective rise and fall (resistances) equal to a unit inverter (R)
- Annotate the 3-input NAND gate with gate and diffusion capacitance



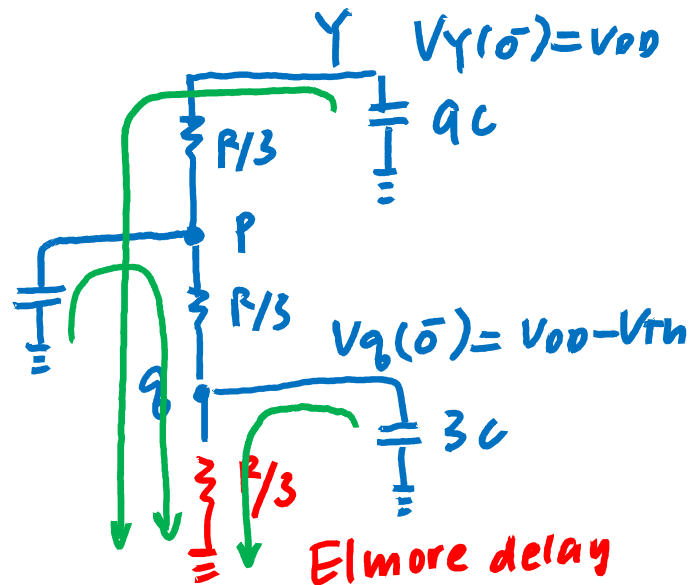
(Worst-Case) Delay of NAND3

only one of the inputs
is switching



falling: $a, b = H$

下面 c ↑

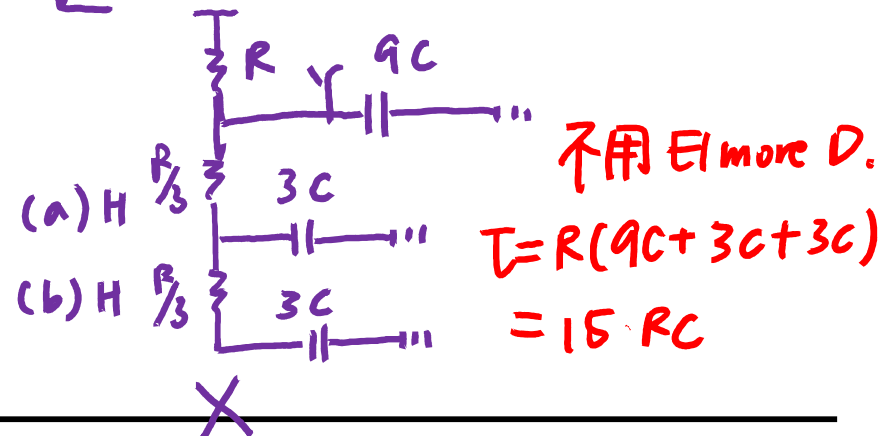


rising

originally all 3 inputs are H

output L

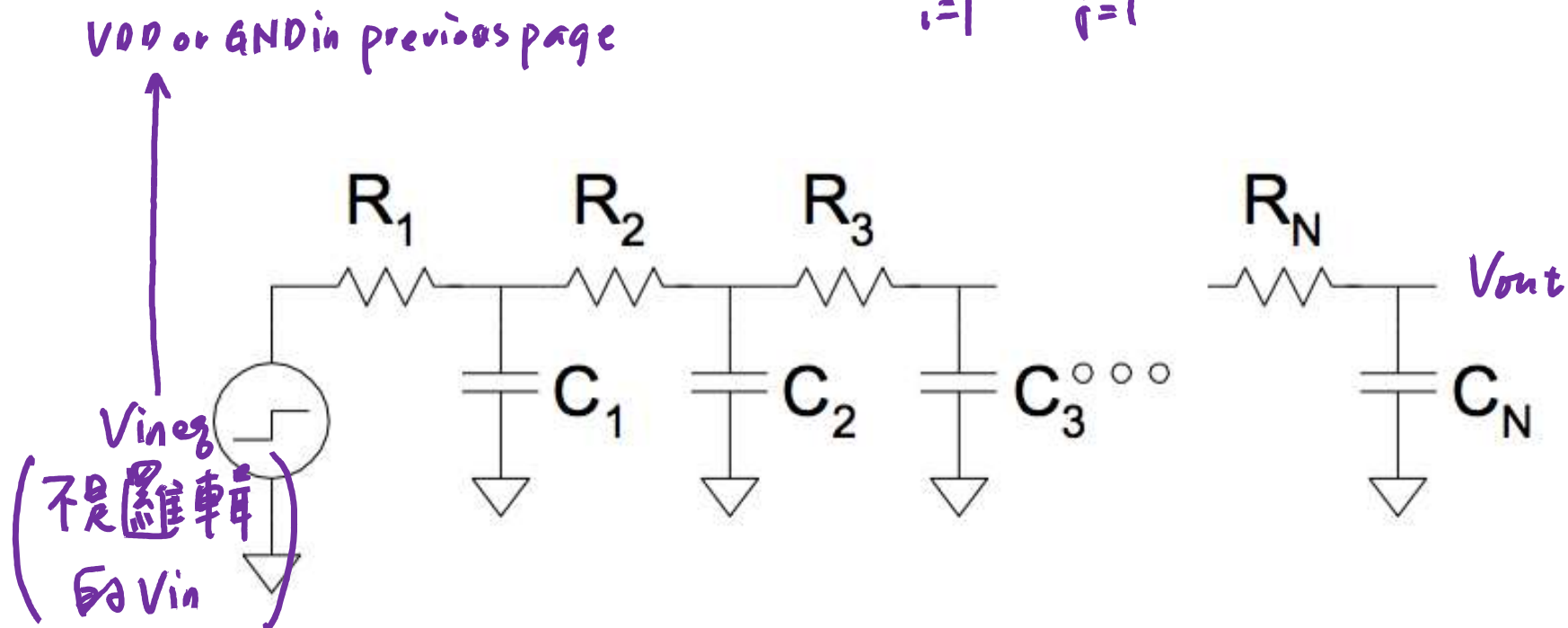
c ↓



Elmore Delay Model

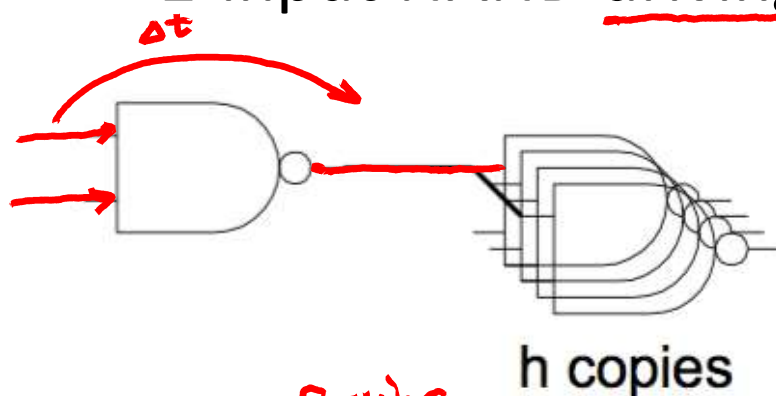
- Pull-up or pull-down network can be modeled as RC ladder
- Elmore delay model of an RC ladder

$$\tau_{pd} = \sum_{i=1}^N C_i \sum_{j=1}^i R_j$$

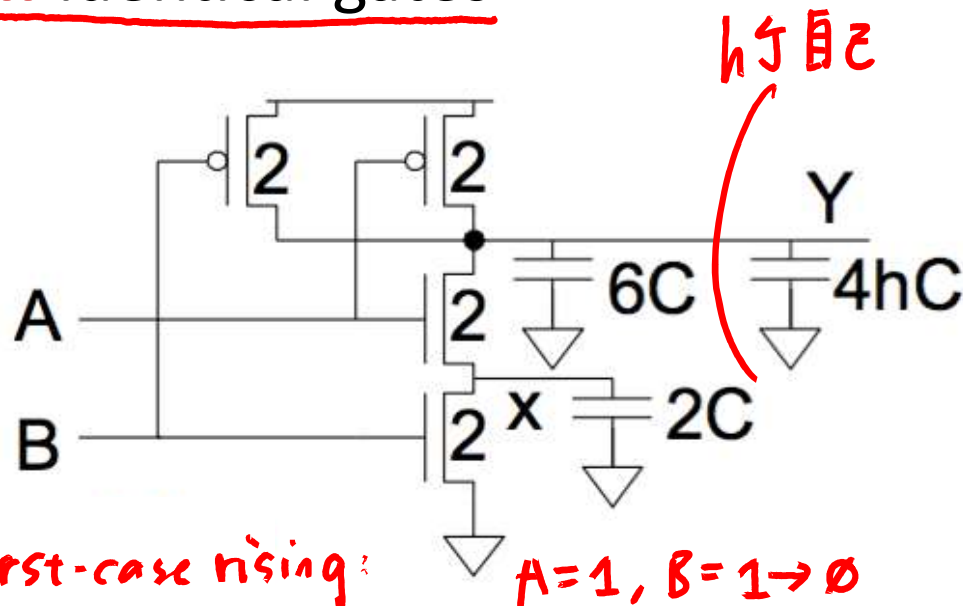


Example: 2-Input NAND Worst-case < B̄A1F >

- Estimate worst-case rising and falling delays of 2-input NAND driving h identical gates

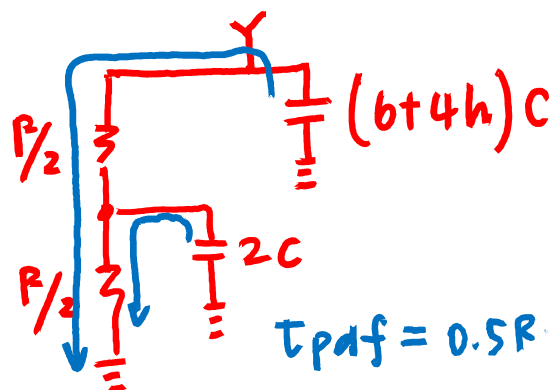


Worst-case falling



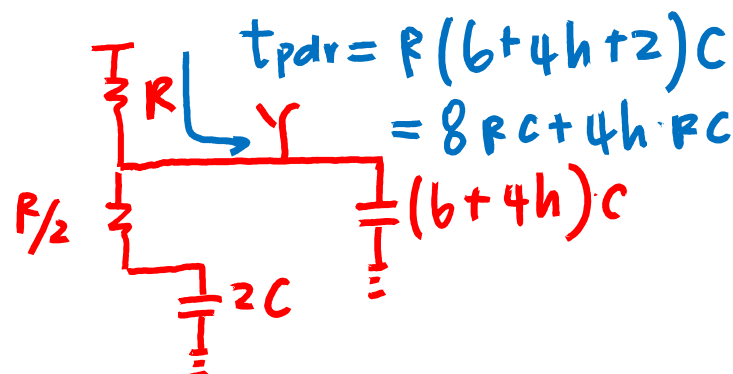
Worst-case rising:

$A=1, B=1 \rightarrow 0$



$$t_{pdf} = 0.5R \cdot 2C + R(b+4h)C$$

$$= 1RC + 4h \cdot RC$$

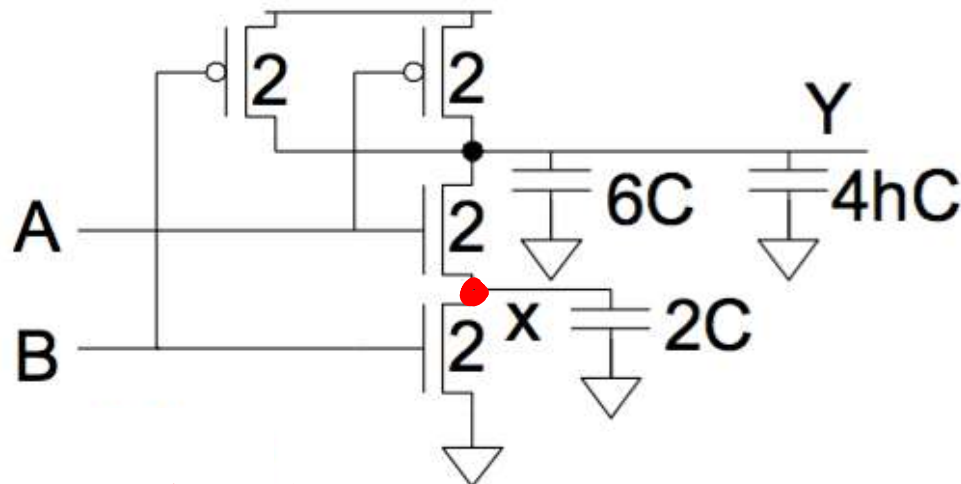


$$t_{pdr} = R(b+4h+2)C$$

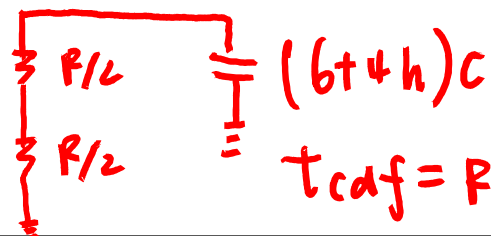
$$= 8RC + 4h \cdot RC$$

Contamination Delay Best-case < A and B >

- Best-case (contamination) delay can be substantially less than worst-case delay
- Example: If both inputs fall simultaneously

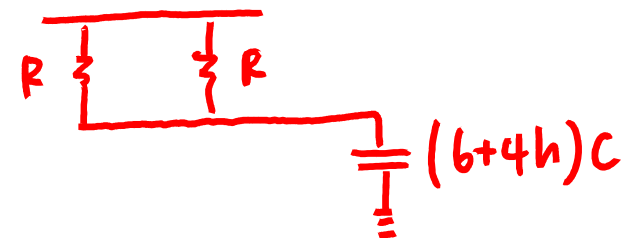


b.c. falling $B=1, A=0 \rightarrow 1$



$$t_{cdf} = R \cdot (6+4h)C = 6RC + 4hRC$$

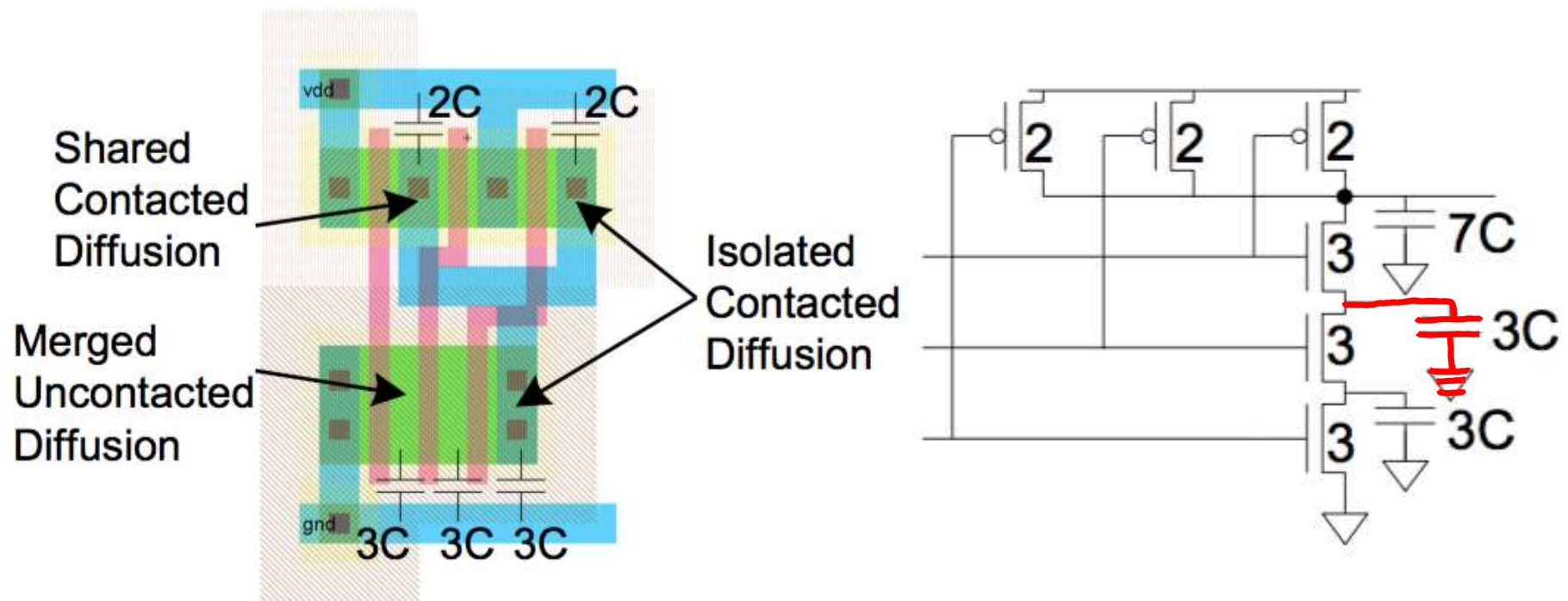
b.c. rising



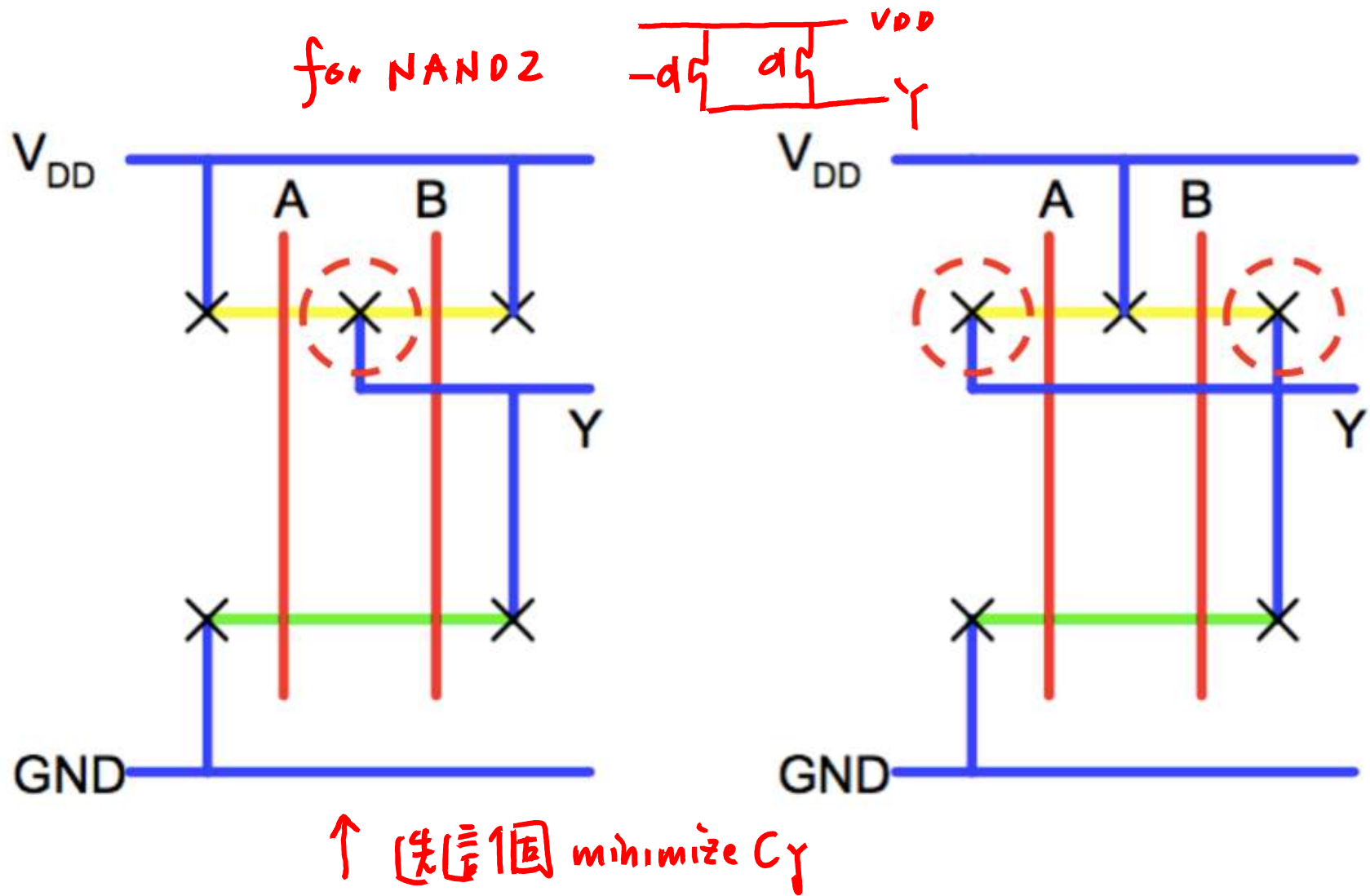
$$t_{cdr} = (3+2h)RC = 3RC + 2hRC$$

Diffusion Capacitance

- Good layout minimizes diffusion area
- Example: NAND3
 - Sharing diffusion contacts reduces output cap by $2C$
 - Merged un-contacted diffusion might help too



Layout Comparison



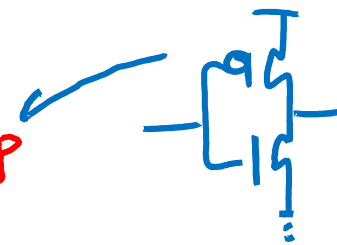
Delay Components


P • Parasitic delay

- Independent of load (和h無關)

f • Effort delay 和h有關, proportional to h (Cont)

- Proportional to load capacitance

$$d = f + p$$


$$P_{inv} = 3RC$$


不論 $P:N = 2:1, 4:2, 6:3, \dots$

Outline

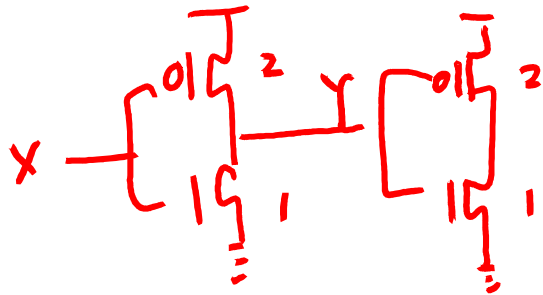
- Delay estimation
- **Logical effort and transistor sizing**
- Power dissipation
- Interconnect
- Wire engineering
- Design margin
- Reliability
- Scaling

Introduction

- Chip designers face a bewildering array of choices
 - What is the best circuit topology for a given function?
 - How many stages of logic gives the least delay?
 - How wide should the transistors be?
- **Logical effort** is a method to make these decisions
 - Uses a simple model of delay
 - Allows back-of-the-envelope calculations
 - Helps make rapid comparison between alternatives
 - Emphasizes remarkable symmetries

Delay of a Logic Gate

- Express delays in **process-independent** unit



1st inverter 推 h inverter

$$\text{delay} = 3RC + 3 \cdot h \cdot RC$$

effort delay

delay = $6RC = 3RC + 3RC$ (1st inverter 推 1st inverter 的 delay)

d = $\frac{\text{delay}}{\tau}$ where $\tau = 3RC$ ideally

normalized

$\leq 12\text{ps in } 180\text{nm}$
 $40\text{ps in } 0.6\mu\text{m}$

Delay of a Logic Gate

- Express delays in **process-independent** unit

$$d = d_{abs} / \tau$$

- Delay has two components

Delay of a Logic Gate

- Express delays in **process-independent** unit

$$d = d_{abs} / \tau$$

- Delay has two components

$$d = f + p$$

- Effort delay (or stage effort) has two components

Delay of a Logic Gate

- Express delays in **process-independent** unit

$$d = d_{abs} / \tau$$

- Delay has two components

$$\underline{d = f + p}$$

- Effort delay (or stage effort) has two components

$$f = \boxed{gh}$$

- g**: logical effort
 - Measure relative ability of gate to deliver current
 - $g = 1$ for inverter

Delay of a Logic Gate

- Express delays in **process-independent** unit

$$d = d_{abs} / \tau$$

- Delay has two components

$$d = f + p$$

- Effort delay (or stage effort) has two components

$$f = gh$$

- h***: electrical effort
 - Ratio of output to input capacitance
 - Sometimes called fanout

Delay of a Logic Gate

- Express delays in **process-independent** unit

$$d = d_{abs} / \tau$$

- Delay has two components

$$d = f + p$$

$$d = gh + p$$

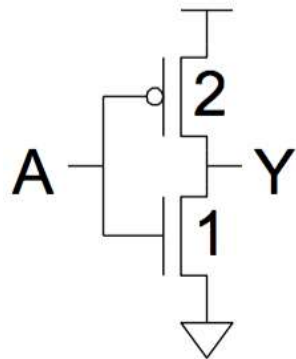
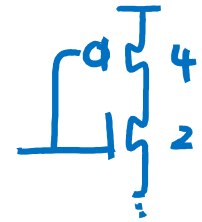
- Parasitic delay ***p***
 - Delay of gate driving no load
 - Due to internal parasitic capacitance

Computing Logical Effort

- Ratio of input capacitance of a gate to that of an inverter delivering the same output current

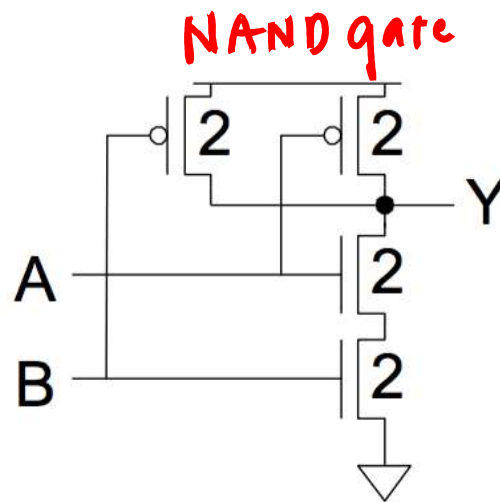
Method #1: Measure from delay vs. fanout plots

Method #2: Estimate by counting transistor widths



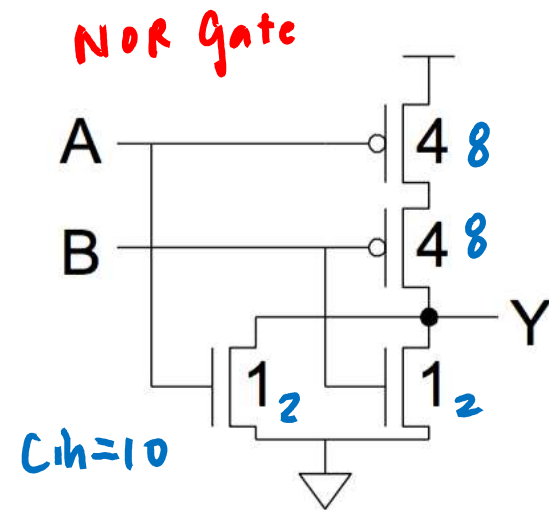
$$C_{in} = 3$$

$$\underline{g = 3/3}$$



$$C_{in} = 4$$

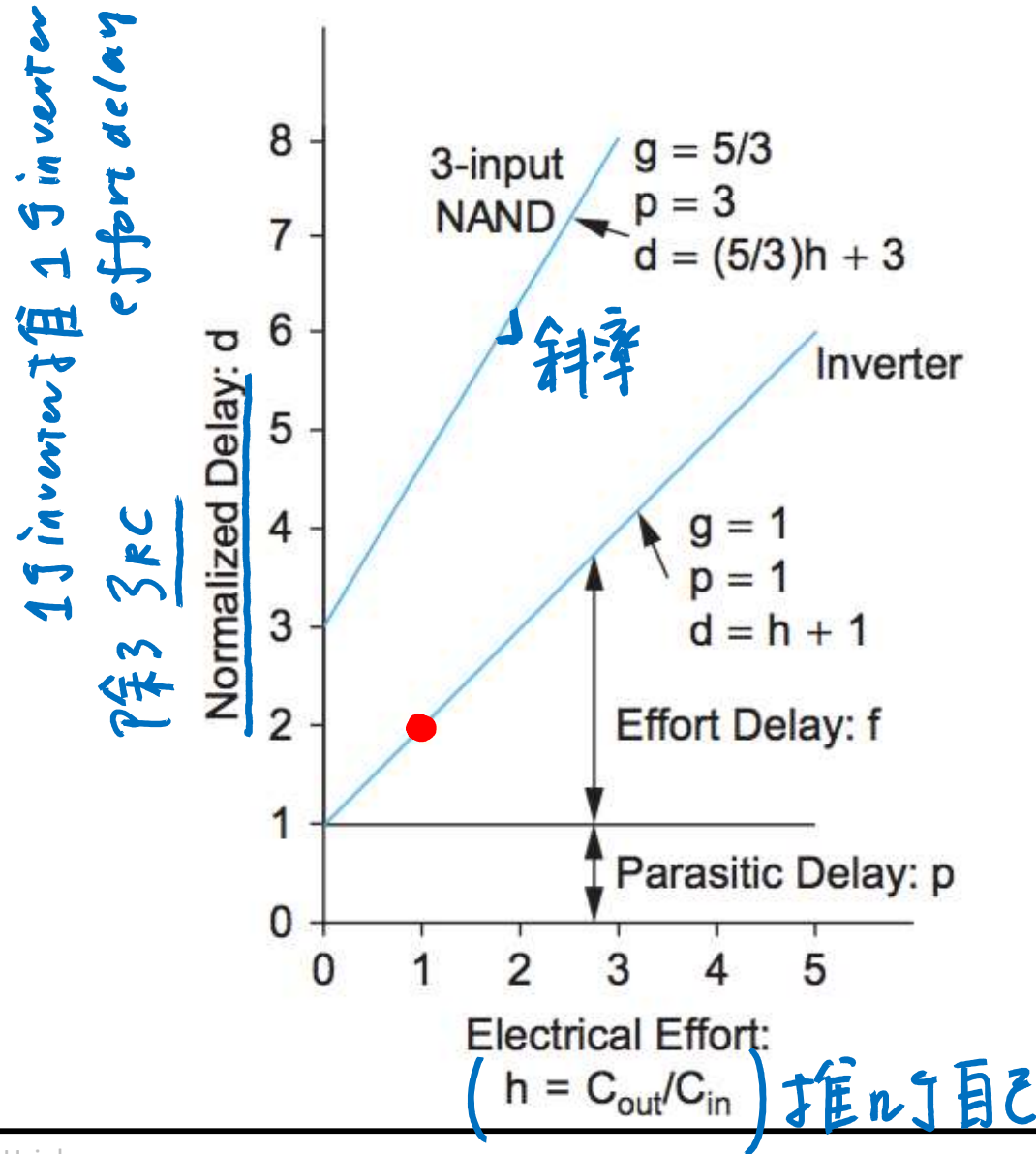
$$\underline{g = 4/3}$$



$$C_{in} = 5$$

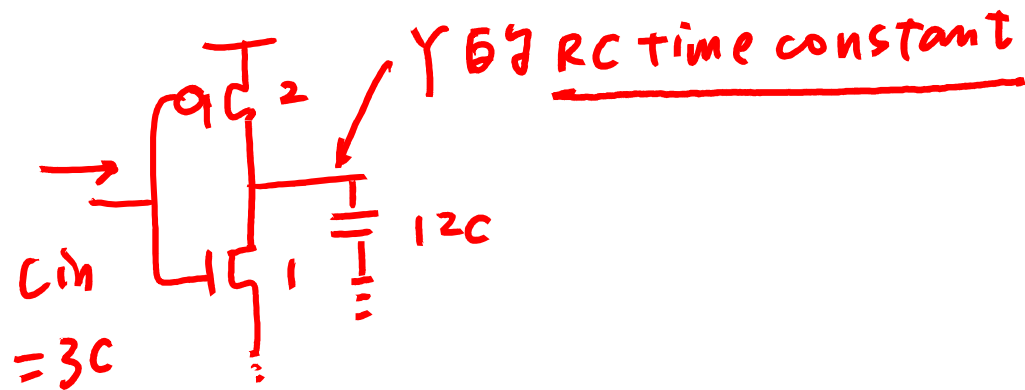
$$\underline{g = 5/3}$$

Delay vs. Fanout Plots

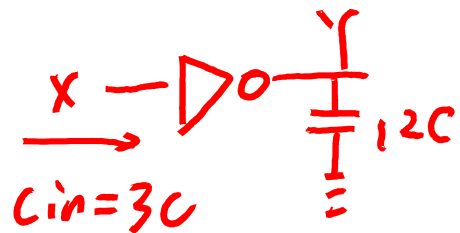


只看 effort delay

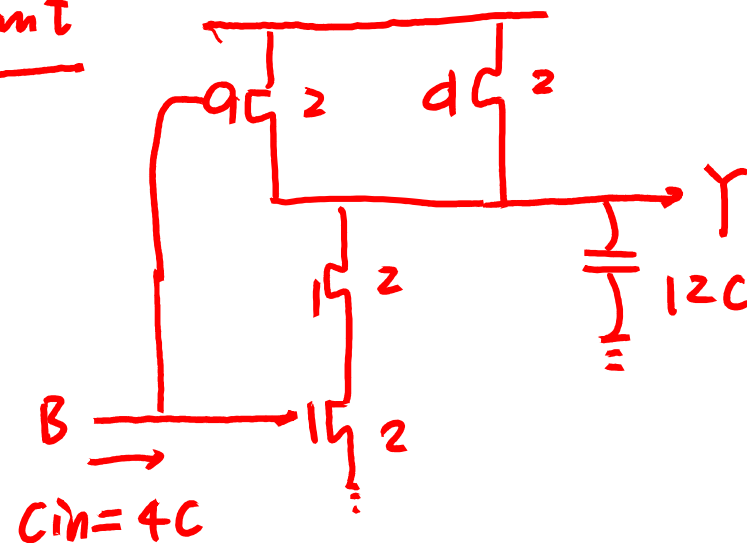
unit inv. 推 $C_L = 12C$



$$t_{paf} = R \cdot (12C)$$

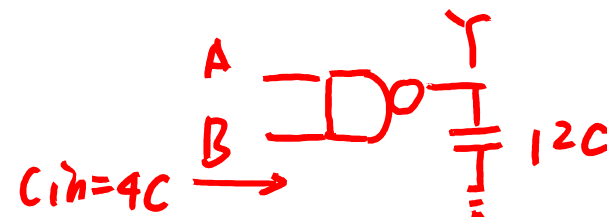


NAND2

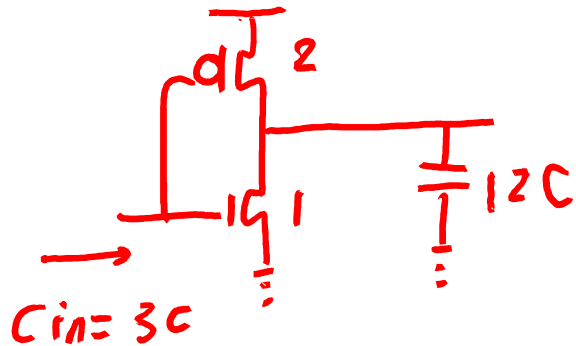


$$A = 1, B = \phi \rightarrow 1$$

$$t_{paf} = R(12C)$$



只看 effort delay

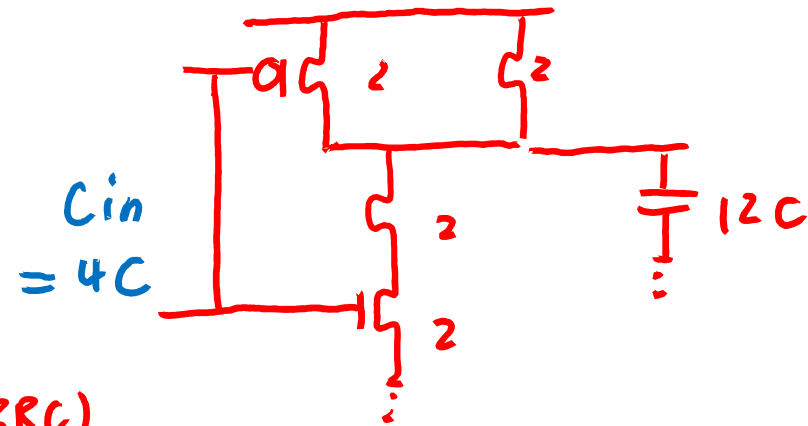


normalized wrt 1 inv 推 1 inv (3RC)

$$f = \frac{12RC}{3RC} = 4 \text{ (4倍 in time domain)}$$

$$= \frac{3C}{3C} \cdot \frac{R}{R} \cdot \frac{12C}{3C}$$

$$= 1 \cdot 4 = \text{推4个自己}$$



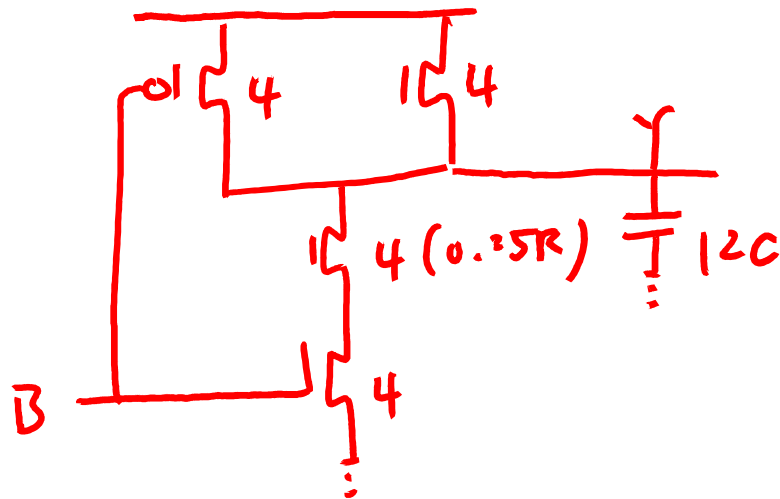
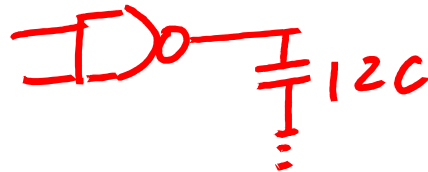
$$f = \frac{12RC}{3RC} = 4 \text{ (4倍)}$$

$$= \frac{4C}{3C} \cdot \frac{R}{R} \cdot \frac{12C}{4C} \quad h = \frac{C_{out}}{C_{in}}$$

$$= \frac{4}{3} \cdot 3 = \text{他其实是推3个自己}$$

One more example (Effort delay)

NAND2



$$t_{\text{par}} = (0.25 + 0.25) R \cdot 12C = 6RC$$

$$f = g \cdot h = \frac{4}{3} \frac{C_{\text{out}}}{C_{\text{in}}} = \frac{4}{3} \frac{12C}{8C} = 2 \frac{2}{3}$$

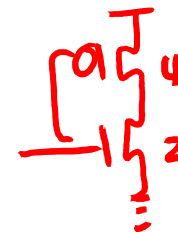
normalized effort delay

$$f = \frac{0.5 \cdot R \cdot 12C}{3RC}$$

$$= \frac{8C \cdot 0.5R \cdot 12C}{6C \cdot 0.5R \cdot 8C}$$

$$g = \frac{12C}{8C} = \frac{3}{2} \quad h = \frac{1}{2}$$

努力因子



Catalog of Gates

- Logic effort of common gates

Gate Type	Number of inputs				
	1	2	3	4	n
✓ Inverter	1				
✓ NAND		$4/3$	$5/3$	$6/3$	$(n+2)/3$
✓ NOR		$5/3$	$7/3$	$9/3$	$(2n+1)/3$
Tri-state, MUX	2	2	2	2	2
XOR, XNOR		4, 4	4, 12, 6	8, 16, 16, 8	

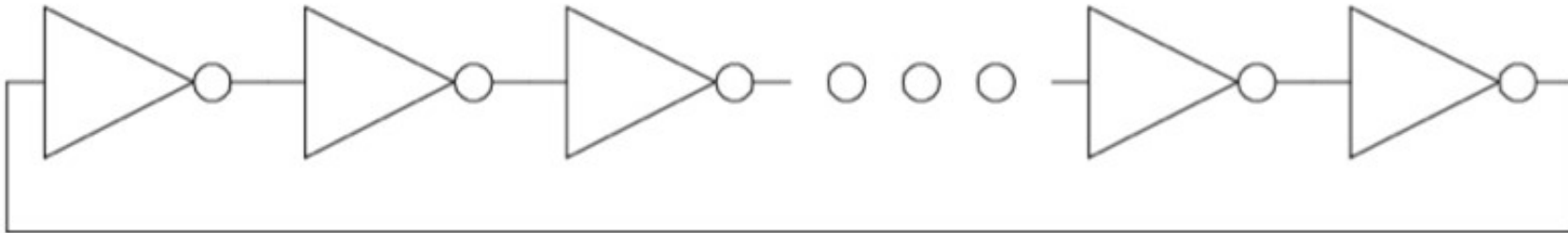
Catalog of Gates

- Parasitic delay of common gates

Gate Type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tri-state, MUX	2	4	6	8	2n
XOR, XNOR		4	6	8	

N-stage Ring Oscillator

- Estimate the frequency

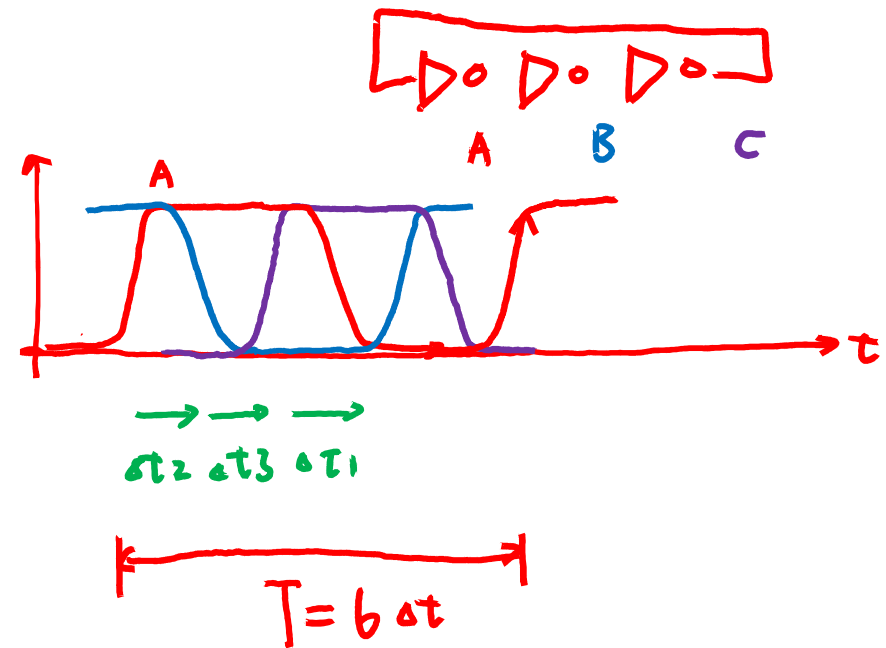


- Logic effort: $g = 1$
- Electrical effort: $h = 1$
- Parasitic delay: $p = 1$
- Stage delay: $\text{delay} = 6RC$
- Frequency: $3RC + 3RC$

$$\boxed{\frac{f}{2}} \text{ 其 } A = 2N\sigma t$$

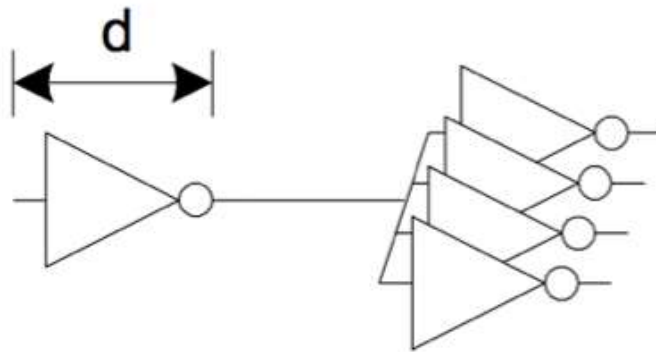
$$= 2N \cdot 2 \quad f \cdot g = 1/4N$$

$$d = gh + p = 2$$



Example: FO-4 Inverter

- Estimate the delay of an inverter with fanout of 4 (FO4)



- Logic effort: $g = 1$
- Electrical effort: $h = 4$
- Parasitic delay: $p = 1$
- Stage delay: $d = gh + p = 5$

- Rule of thumb: FO4 delay for a process is 1/3 to 1/2 of the minimum channel length. **EX 180 nm: FO4 = 60~90 ps**
- Highly sensitive to process, voltage, & temperature variations

Multistage Logic Networks

- Logic effort generalizes to multistage networks

- Path logical effort

$$G = \prod g_i$$

$$g_1, g_2, g_3, g_4$$

- Path electrical effort

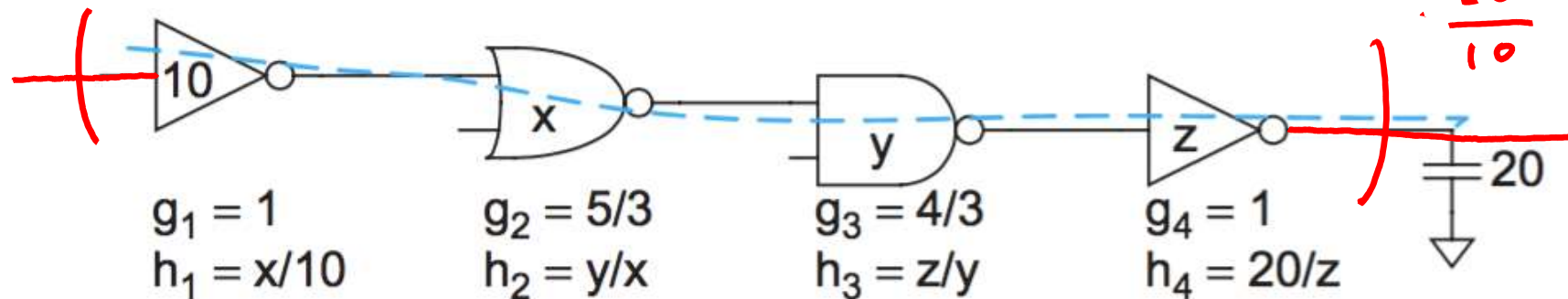
$$H = C_{out-path} / C_{in-path} \quad \frac{20}{10}$$

- Path effort

$$F = \prod f_i = \prod g_i h_i$$

$C_{in,max}$

$$G \cdot H = g_1 \cdot h_1 \cdot g_2 \cdot h_2 \cdot g_3 \cdot h_3 \cdot g_4 \cdot h_4 = g_1 \cdot g_2 \cdot g_3 \cdot g_4 \cdot \frac{20}{10}$$



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 the smallest when each stage bears the same effort

$$\hat{f} = g_i h_i = F^{\frac{1}{N}}$$

- Minimum delay of N-stage path is

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

- This is the **key** result of logic effort analysis
 - Find fastest possible delay
 - Doesn't require calculating gate size

Gate Size

- How wide should the gates be for the least delay?

$$\hat{f} = gh = g \frac{C_{out}}{C_{in}}$$
$$\Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$$

- Working backwards, apply capacitance transformation to find input capacitance of each gate with given load it drives
- Check work by verifying input cap spec is met

Paths with Branches

- $F = GH$?
- No! Consider paths with branches

$$G = 1$$

$$H = 90/5 = 18$$

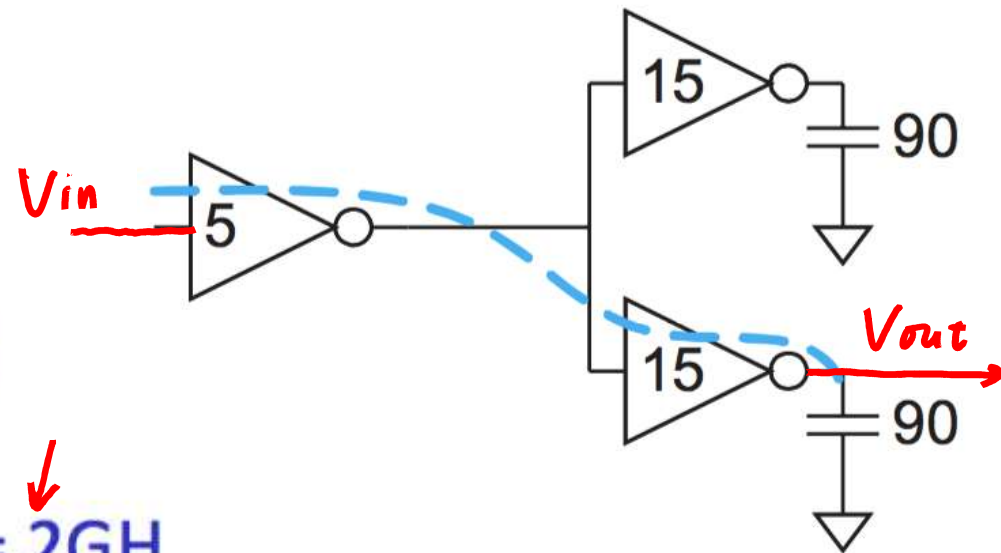
$$GH = 18$$

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

$$h_2 = 90/15 = 6$$

$$F = g_1 g_2 h_1 h_2 = 36 = \underline{2GH}$$

Branching factor



Branching Effort

- Account for branches in path

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

- Path branching effort $B = \prod b_i$

- Now we can compute path effort

$F = GBH$

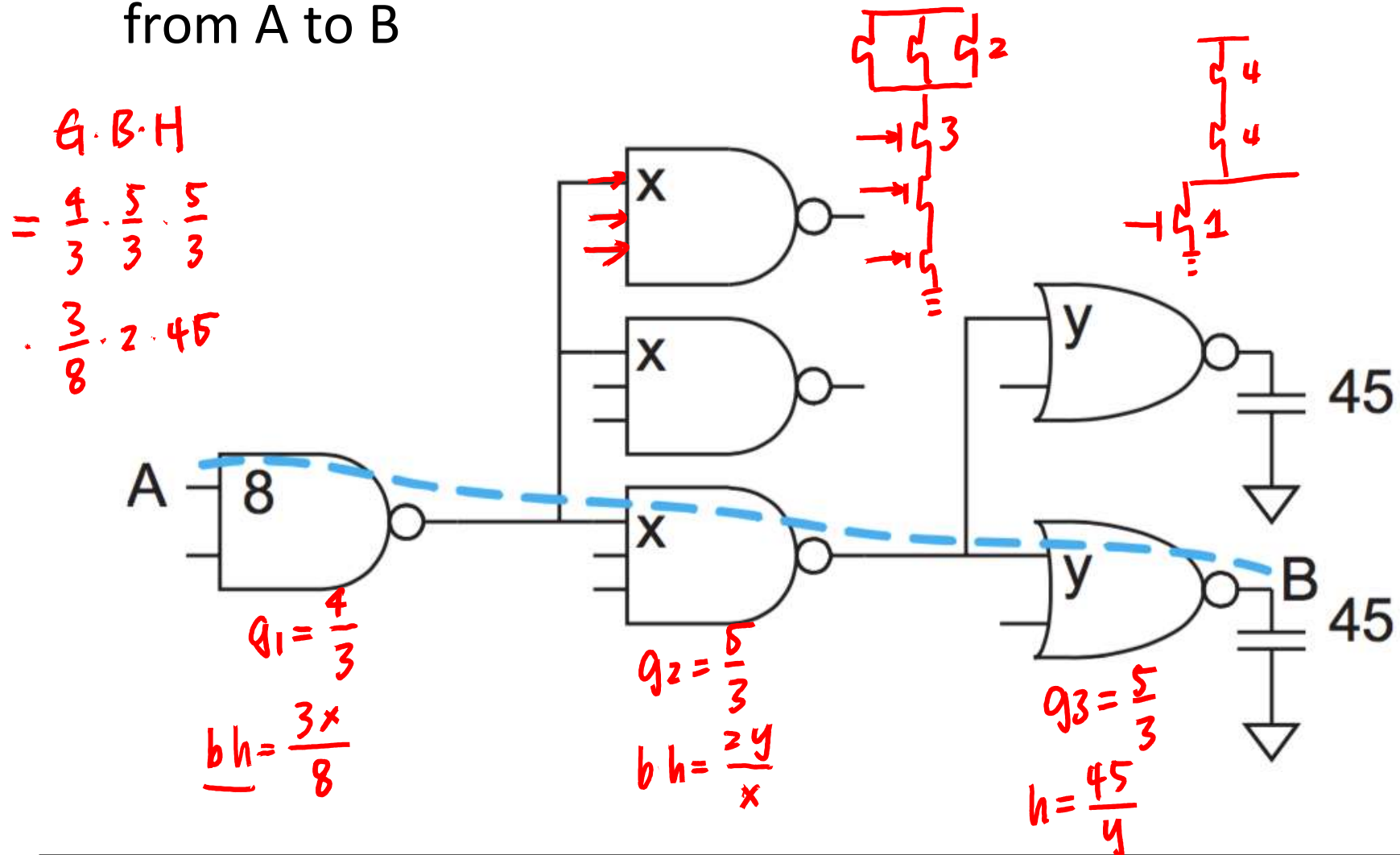
$q_1 \cdot q_2 \cdot q_3 \cdot q_4$

Branching factor

$\frac{C_{\text{out}}}{C_{\text{in}}}$

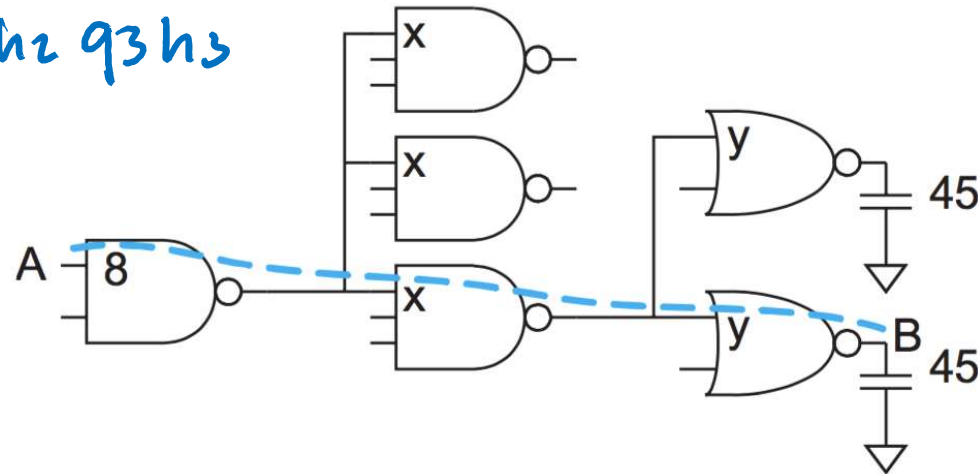
Example: 3-Stage Path

- Select gate size x and y that minimize the delay from A to B



Example: 3-Stage Path

已知 b_1 b_2
 $q_1 h_1, q_2 h_2, q_3 h_3$



- Logical effort $G = (4/3)(5/3)(5/3) = 100/27$

- Electrical effort

- Branching effort

已知

- Path effort

$$F = GBH = 125$$

$$\frac{1}{G} \min \sum q_i b_i h_i \quad (q_1 h_1 + q_2 h_2 + q_3 h_3)$$

- Best stage effort

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

- Parasitic delay

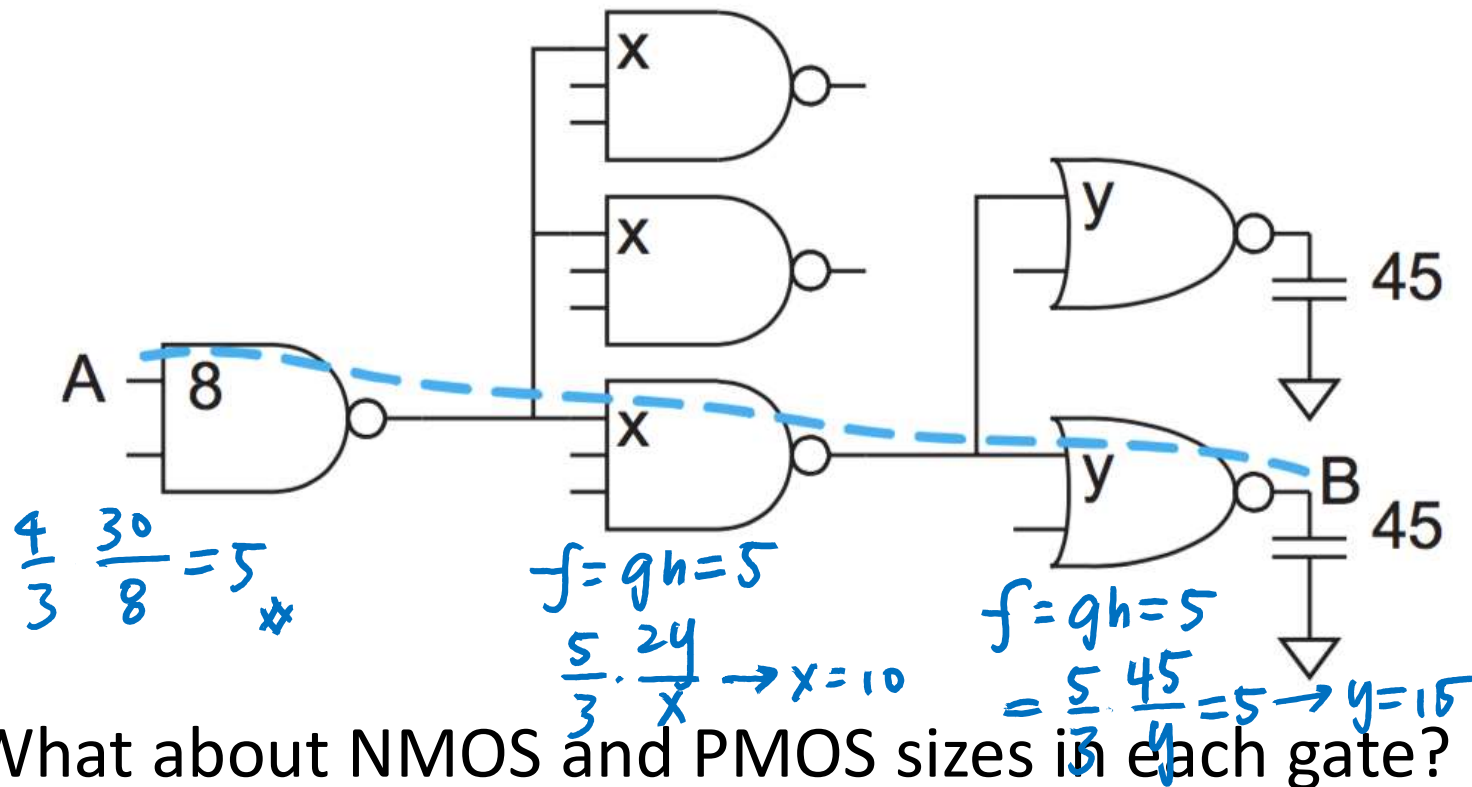
$$P = 2 + 3 + 2 = 7$$

- Delay

$$D = 5 + 5 + 5 + 2 + 3 + 2 = 22$$

Example: 3-Stage Path

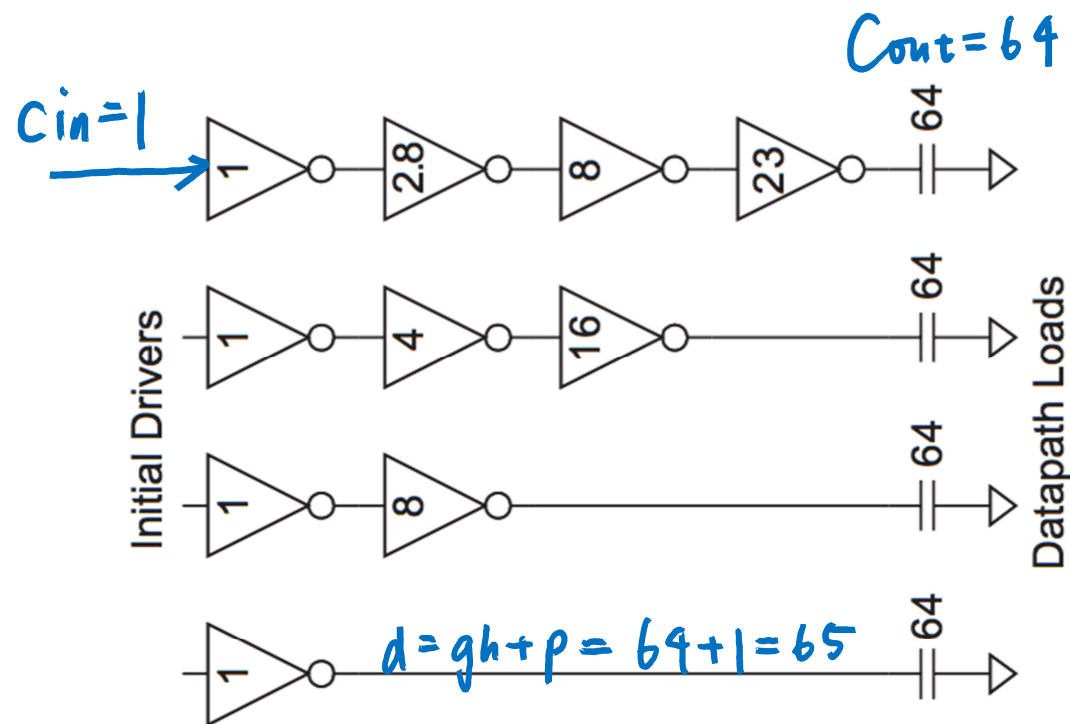
- Work backwards for sizes
 - $y = 45 \cdot (5/3) / 5 = 15$
 - $x = (15 \cdot 2) \cdot (5/3) / 5 = 10$



- What about NMOS and PMOS sizes in each gate?

Best Number of Stages *inv p=1*

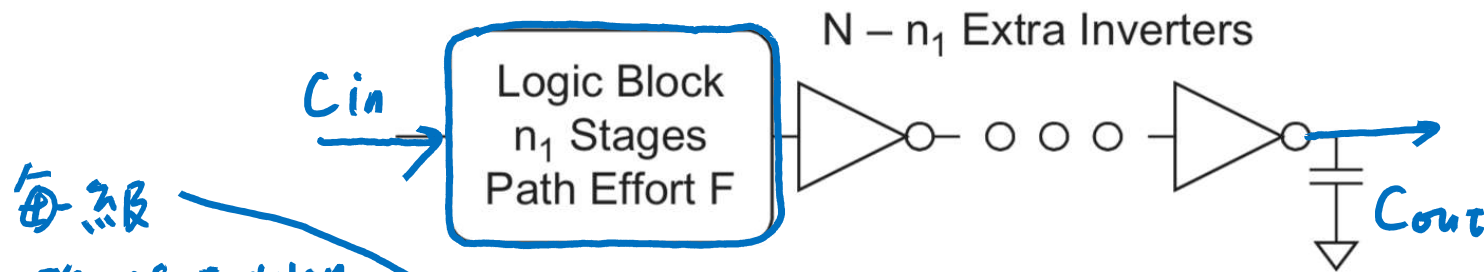
- How many stages should a path use
 - Minimizing number of stages is not always the fastest
- Example: Drive 64-bit datapath with unit inverter



N	<i>f</i>	D
4	2.8	15.3
3	4	15
2	8	18
1	64	65

Derivation

- Consider inserting inverters into the signal chain
 - How many stages give the least delay?



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

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

- Define the best stage effort $\rho = F^{\frac{1}{N}}$

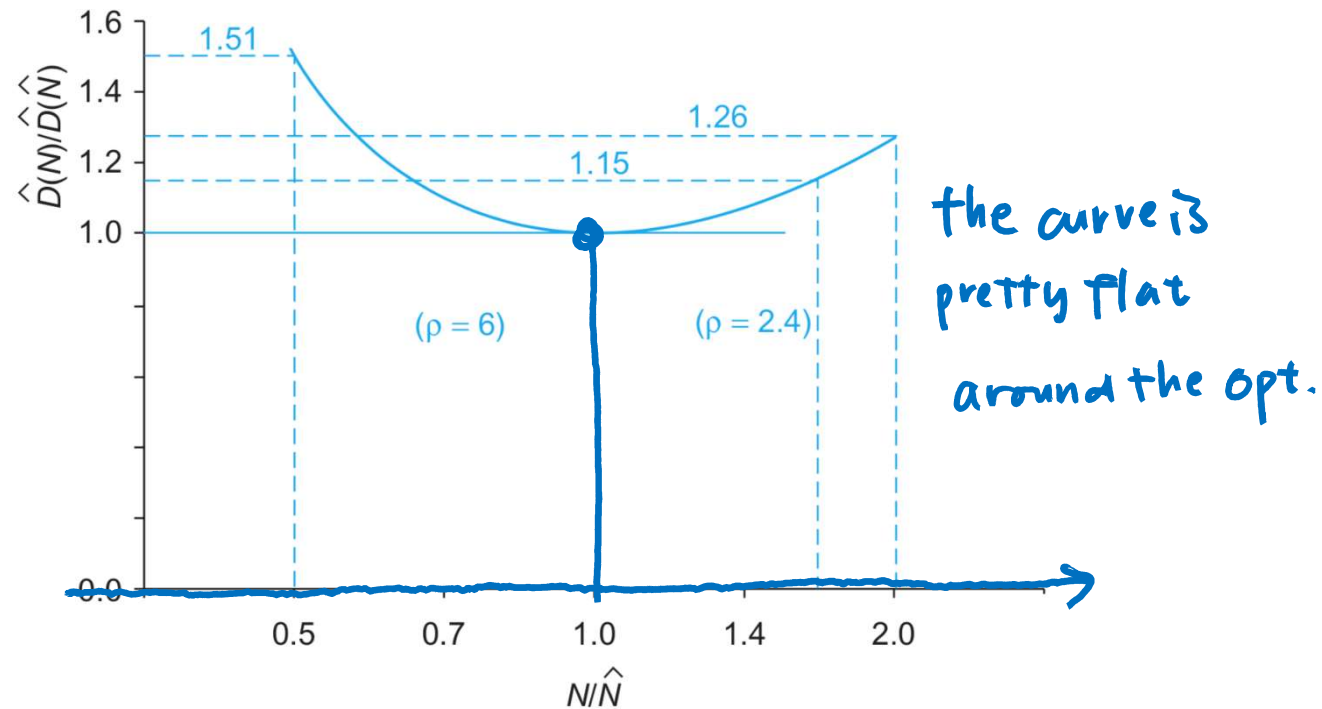
$$p_{inv} + \rho(1 - \ln \rho) = 0$$

Best Stage Effort

- $p_{\text{inv}} + \rho (1 - \ln \rho) = 0$ has no closed-form solution
- Neglecting parasitics ($p_{\text{inv}} = 0$), we define
$$\rho = 2.718 (e)$$
- For $p_{\text{inv}} = 1$, solve numerically for $\rho = 3.59$

Sensitivity Analysis

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



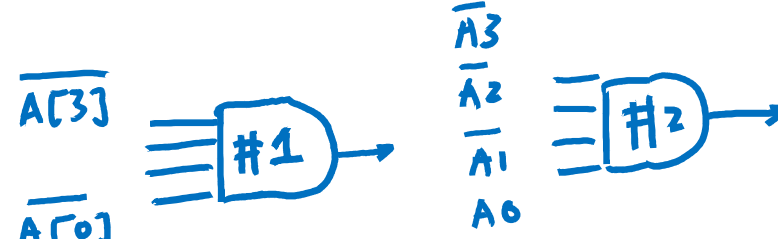
- $2.4 < \rho < 6$ gives delay with 15% variations
— 4 is a common choice

Example: Decoder for a Register File

- Specifications

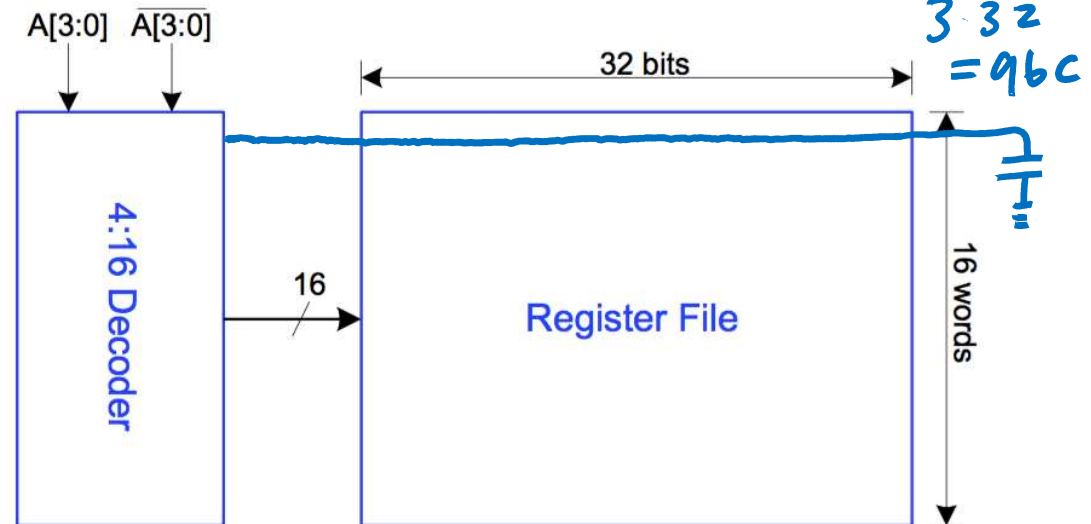
- 16-word register file
- Each word is 32-bit wide
- Each bit presents a load of 3 unit-sized transistors
- True and complementary address inputs A[3:0]
- Each input may drive 10 unit-sized transistors

(revisited)



- Need to decide:

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



Number of Stages plan

- Decoder effort is mainly electrical and branching

Electrical effort $H = 96/10 = 9.6$

Branching effort $B = 8$

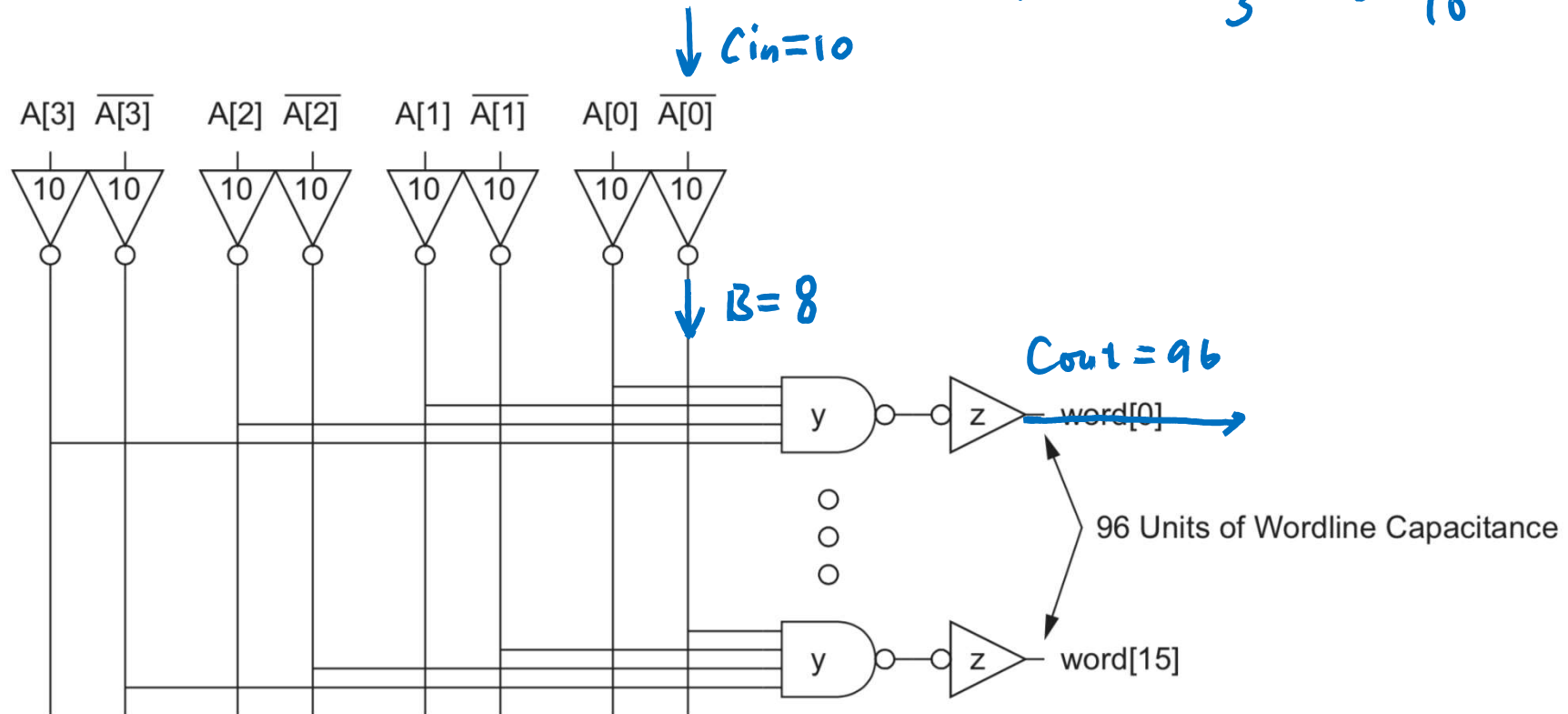
- If we neglect logical effort (by assuming $G = 1$)

Path effort $F = GBH = 76.8$

Number of stages $N = \log_4 F = 3.1$

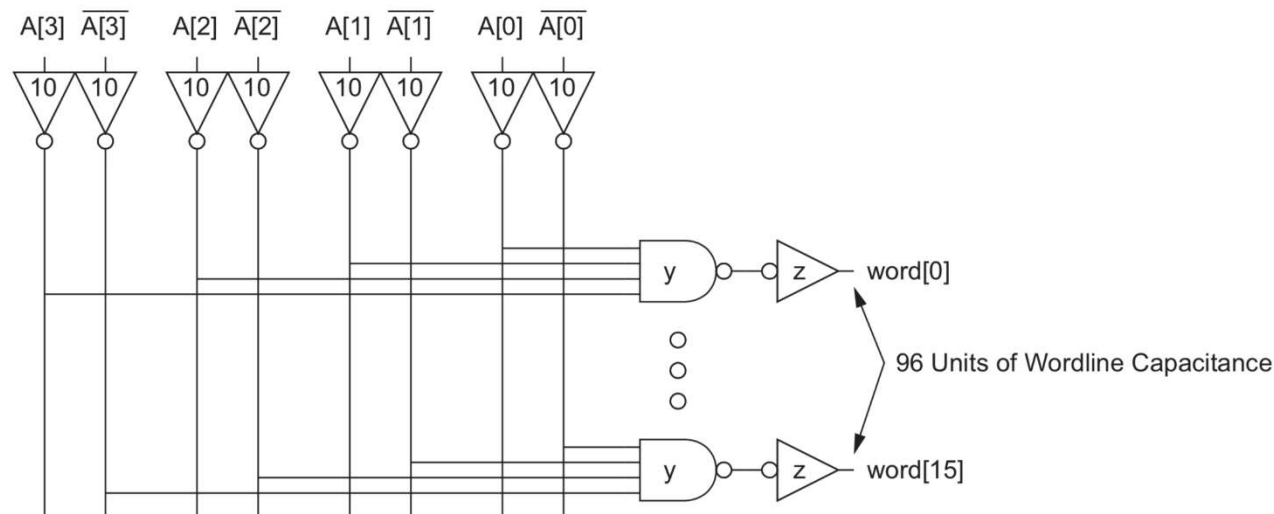
3 Stage 4:16 Decoder

$$GBH = 1 \cdot \frac{6}{3} \cdot 1 \cdot 8 \cdot \frac{96}{10} = 154$$



Gate Sizes and Delay

- Logical effort $G =$
- Path effort $F = 6BH = 154$
- Stage effort $\hat{f} = \sqrt[3]{F} = 5.36$ *parasitic delay*
- Path delay $D = 3 \cdot 5.36 + \underline{1+4+1} = 22.1$
- Gate sizes $z =$ $y =$



Comparison

- Different alternatives

Design	<i>f</i>	N	G	P	D
NAND4-INV	12.4	2	2	5	29.8
NAND2-NOR2	13.06	2	20/9	4	30.1
INV-NAND4-INV	5.36	3	2	6	22.1
NAND4-INV-INV-INV	3.52	4	2	7	21.1
NAND2-NOR2-INV-INV	3.61	4	20/9	6	20.5
NAND2-INV-NAND2-INV	3.41	4	16/9	6	19.7
INV-NAND2-INV-NAND2-INV	2.67	5	16/9	7	20.4
NAND2-INV-NAND2-INV-INV-INV	2.26	6	16/9	8	21.6

Key Insights from Logical Effort

- Logical Effort characterizes the complexity of a logic gate or path
 - Allow comparison of alternative circuit topologies
- NAND structures are faster than NOR structures in static CMOS circuits
- Paths are fastest when
 - Effort delays of each stage are about the same
 - These delays are close to 4 (FO4 inverter delays)
 - Each quadrupling of the load adds about one FO4 inverter delay to the path
- Path delay is **insensitive** to modest deviations from the optimum
 - Stage efforts of 2.4–6 give designs within 15% of minimum delay.
 - There is no need to make calculations to more than 1–2 significant figures, so many estimations can be made in your head. There is no need to choose transistor sizes exactly according to theory and there is little benefit in tweaking transistor sizes if the design is reasonable.
- Using fewer stages for “less # of gate delays” does not make a circuit faster. Making gates larger also does not make a circuit faster; it only increases the area and power consumption.
- Stage efforts somewhat greater than 4 reduces area and power consumption at a slight cost in speed. Using efforts greater than 6–8 comes at a significant cost in speed.
- Logical Effort of a gate increases as the number of inputs grows
 - Considering both logical effort and parasitic delay, we find a practical limit of about 4 series transistors in logic gates and about 4 inputs to multiplexers.
 - Beyond this fan-in, it is faster to split gates into multiple stages of skinnier gates.
- Inverters or 2-input NAND gates with low logical efforts are best for driving nodes with a large branching effort. Use small gates after the branches to minimize load on the driving gate.
- When a path forks and one leg is more critical than the others, buffer the noncritical legs to reduce the branching effort on the critical path

Review

	Stage	Path
Number of stages	1	N
Logical effort	g	$G = \prod g_i$
Electrical effort	$h = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{out-path}}{C_{in-path}}$
Branching effort	$b = \frac{(C_{on-path} + C_{off-path})}{C_{on-path}}$	$B = \prod b_i$
Effort	$f = gh$	$F = GBH$
Effort delay	f	$D_F = \sum f_i$
Parasitic delay	p	$P = \sum p_i$
Delay	$d = f + p$	$D = \sum d_i = D_F$

Method of Logical Effort

1. Compute path effort

$$F = GBH$$

2. Estimate the best number of stages

$$N = \log_4 F$$

3. Sketch path with N stages

4. Estimate the least delay

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

5. Determine the best stage effort

$$\hat{f} = F^{\frac{1}{N}}$$

6. Find gate sizes

$$C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$$

Limits of Logical Effort

- Chicken and egg problem
 - Need path to compute G
 - Don't know the number of stages with G
- ✓ • Simplified delay model
 - Neglect input rise time effects, velocity saturation, body effect, ...
- ✓ • Neglect interconnect effects
 - Require iterations to take wire capacitance into account
- ✓ • Maximum speed only
 - Not minimum area/power for constrained delay
- ✓ *Non-uniform branching factor*

*high-speed
dense layout
 C_w
 $C_w \uparrow$ dominate*

Summary

- Logical effort is useful when considering circuit delay
 - Numerical logical effort characterize gates
 - NANDs are faster than NORs in CMOS
 - Paths are fastest when effort delays are ~ 4
 - Path delay is not very sensitive to stages and sizes
 - Using fewer stages doesn't necessarily give faster result
- Language for discussing fast circuits
 - Practice required to master