
EE3230 Lecture 4: Circuit Characterization and Performance Estimation I

Ping-Hsuan Hsieh (謝秉璇)

Delta Building R908

EXT 42590

phsieh@ee.nthu.edu.tw

Outline

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

Transient Response

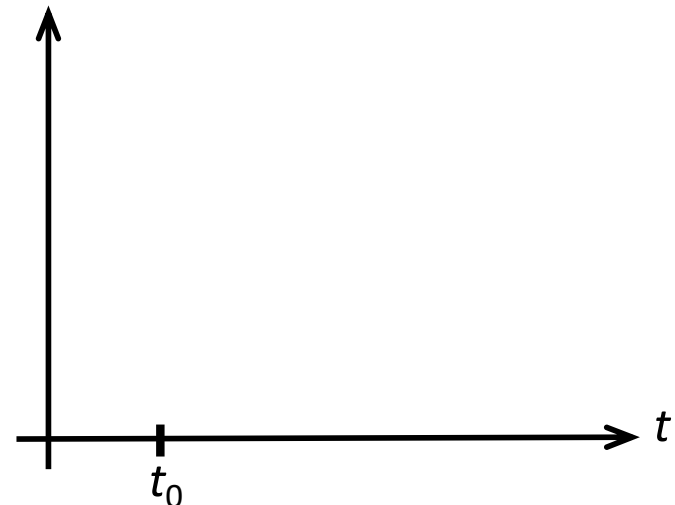
- **DC** analysis tells us V_{out} if V_{in} is constant
- **Transient** analysis tells us $V_{out}(t)$ with certain $V_{in}(t)$
 - Requires solving differential equations
- Input is usually considered to be a step or ramp
 - From 0 to V_{DD} or vice versa

Inverter Step Response

- With load capacitance of C_{load}

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

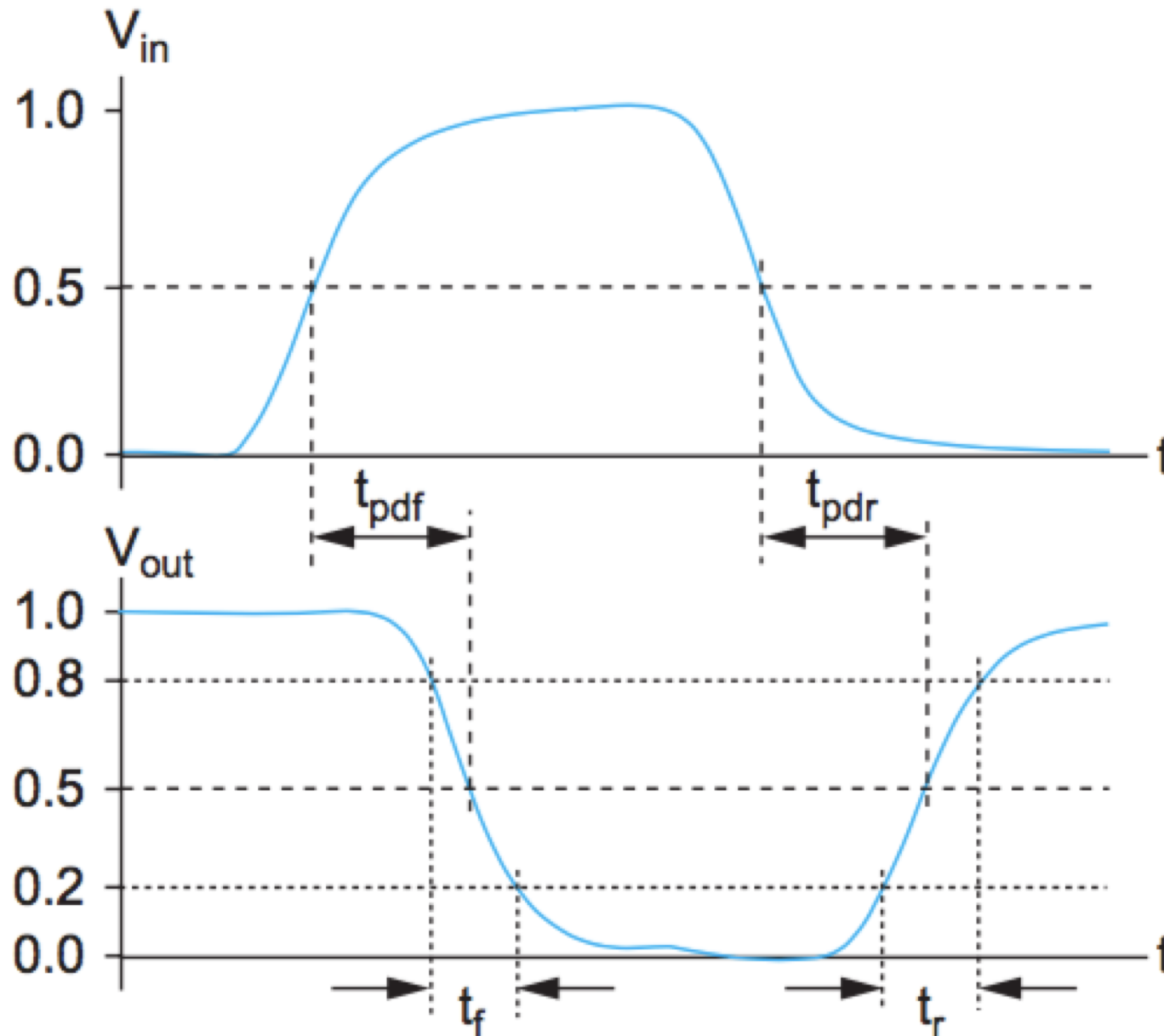
- Current discharging the cap



Delay Definitions (I)

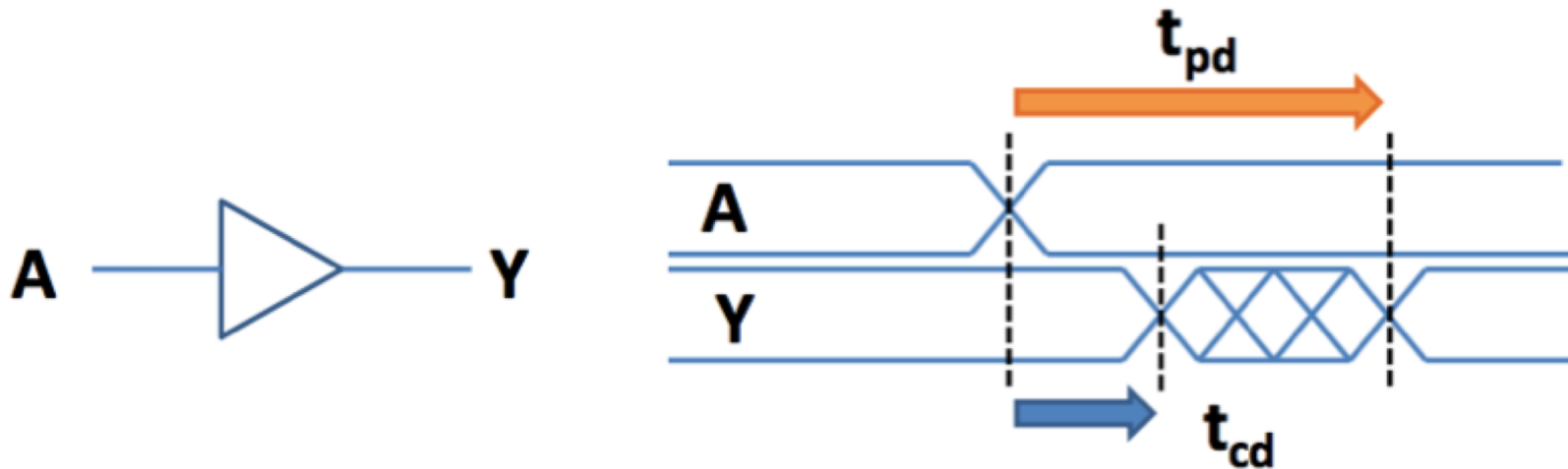
- t_{pdr} maximum rising propagation delay
 - From input to rising output crossing $V_{DD}/2$
- t_{pdf} maximum falling propagation delay
 - From input to falling output crossing $V_{DD}/2$
- t_{pd} average propagation delay
 - $t_{pd} = (t_{pdr} + t_{pdf})/2$
- t_r rise time
 - For output to go from $0.2 V_{DD}$ 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
 - $t_{cd} = (t_{cdr} + t_{cdf})/2$



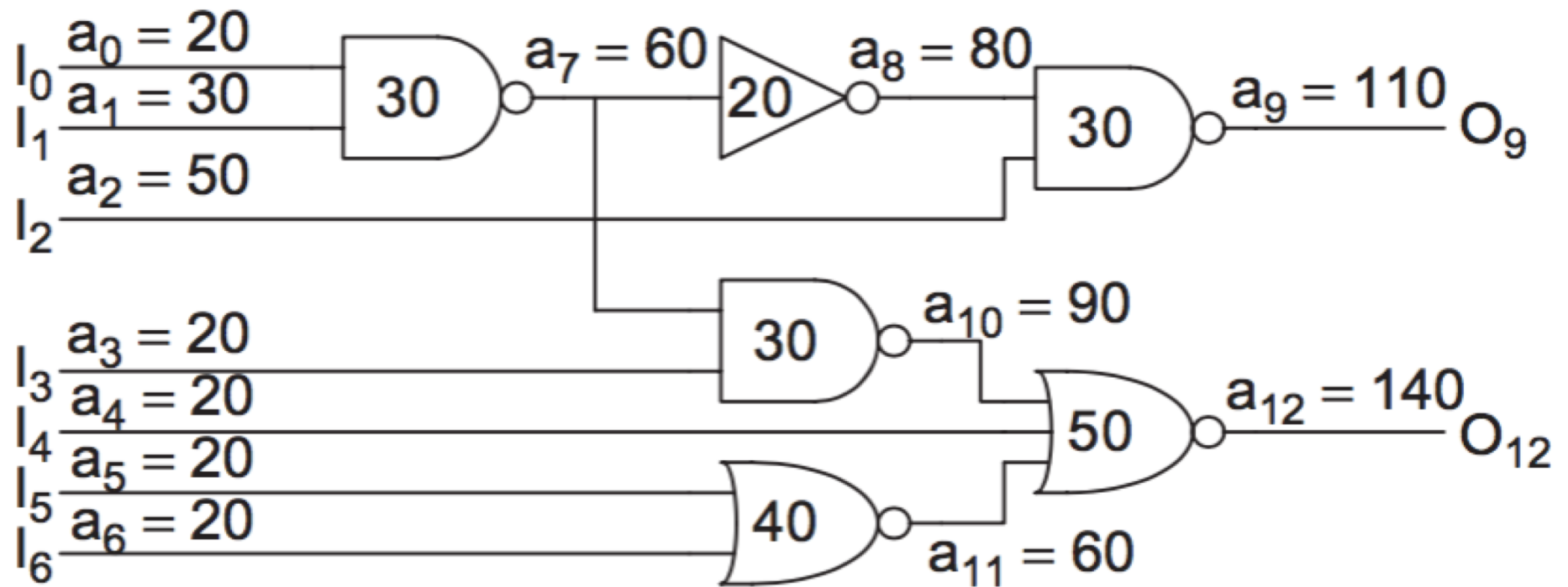
Delay Estimation (I)

- 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)

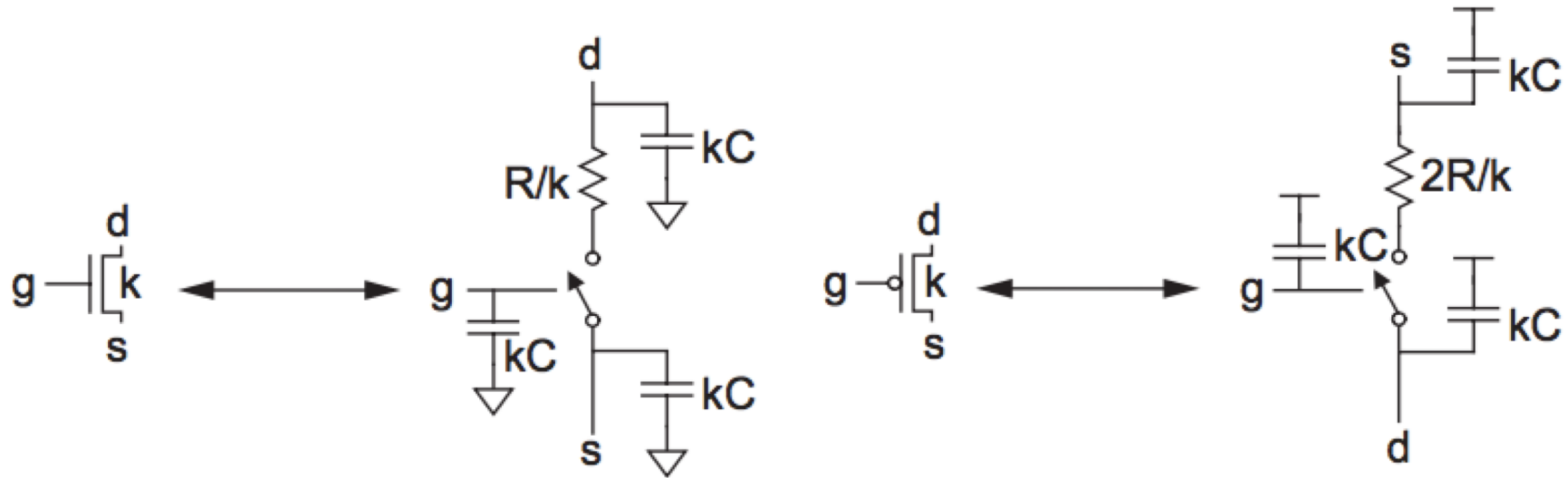
- **Critical path**
 - The signal path with the slowest (most critical) timing
 - Affected at 4 different levels
- Architecture/micro-architecture levels
 - Tradeoff of pipeline stages, number of execution units, and size of memory. It's the level that impacts the most.
- Logic level
 - Tradeoff of functional block types, number of gate in the cycle, fan-in and fan-out number
- Circuit level
 - Transistor size and logic styles/families
- Layout level
 - Floor-plan, wire length, and parasitics

Critical Path

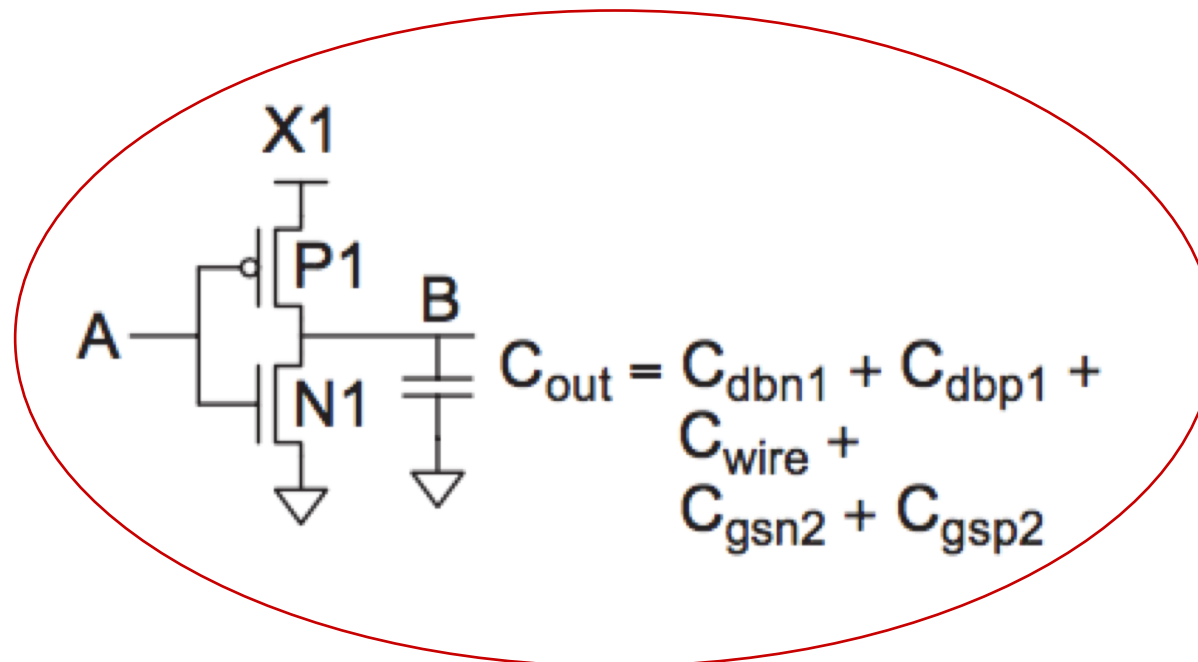
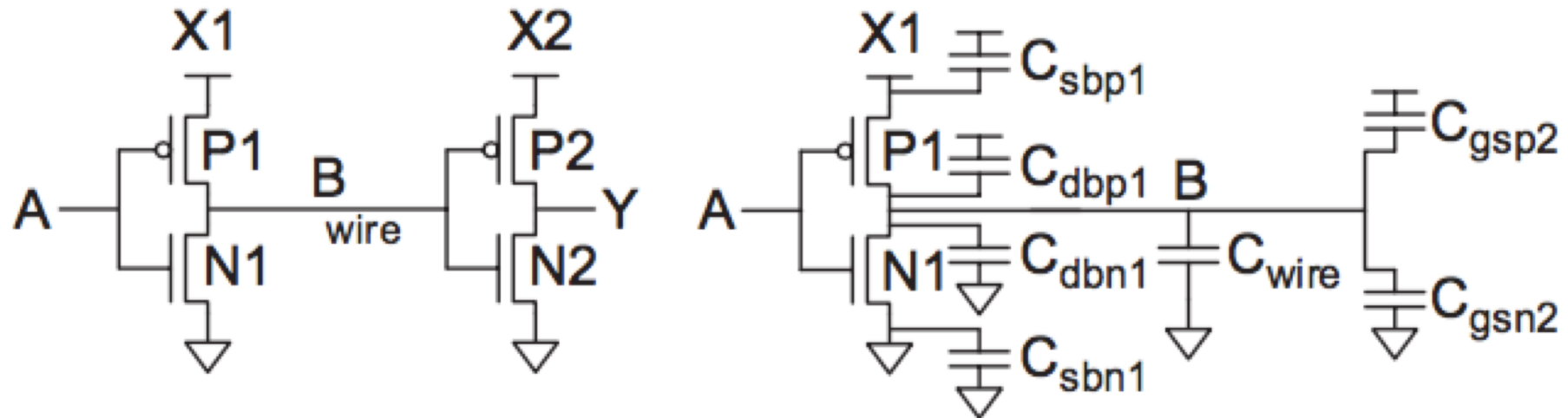


RC Delay Models

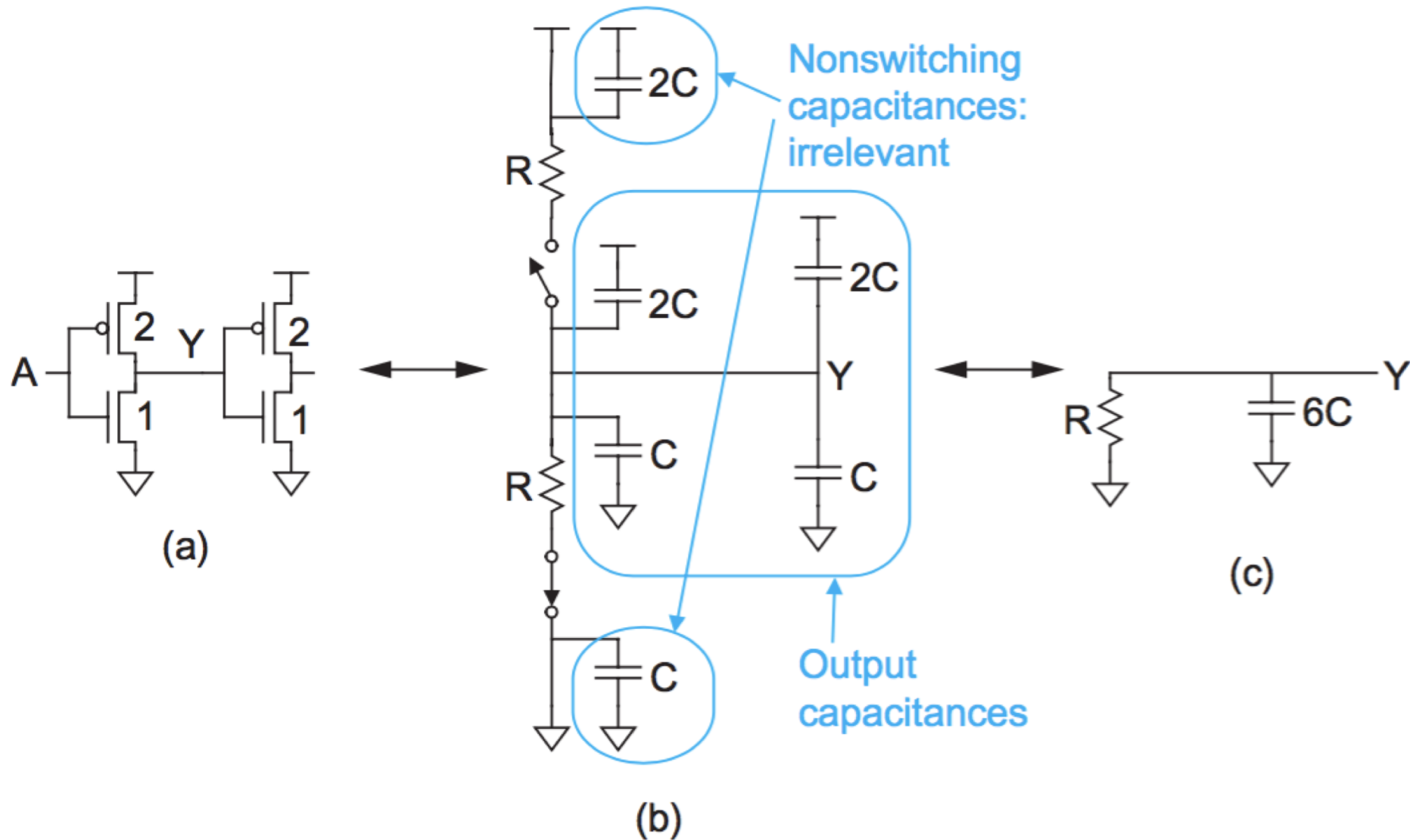
- Equivalent circuit for MOS transistors
 - Ideal switch + capacitance and ON resistance
 - Unit NMOS has resistance R and capacitance C
 - Unit PMOS has resistance $2R$ and capacitance C
- Capacitance proportional to width
- Resistance inversely proportional to width



Example: Inverter



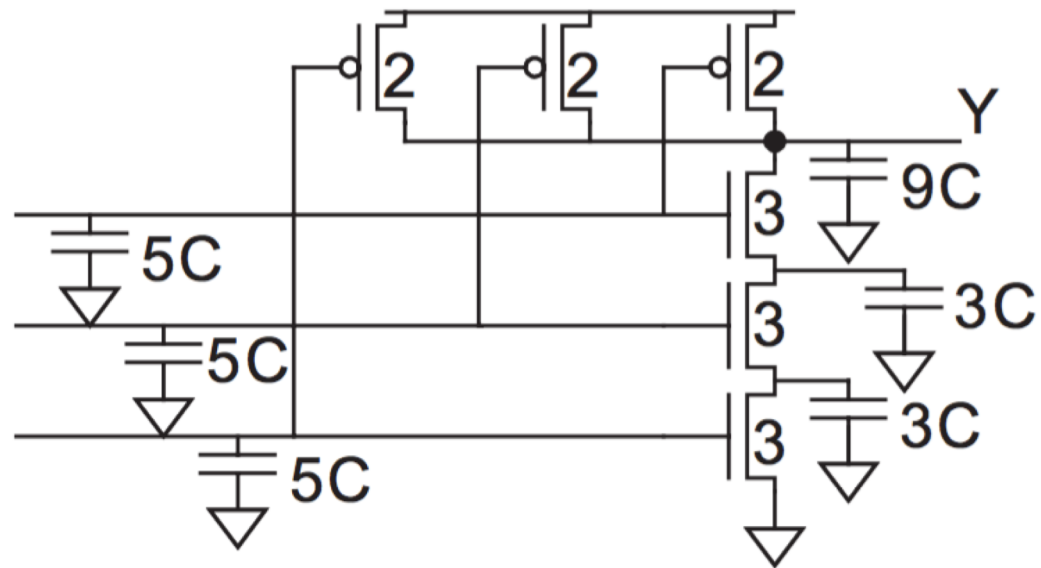
Example: Inverter



Example: NAND3

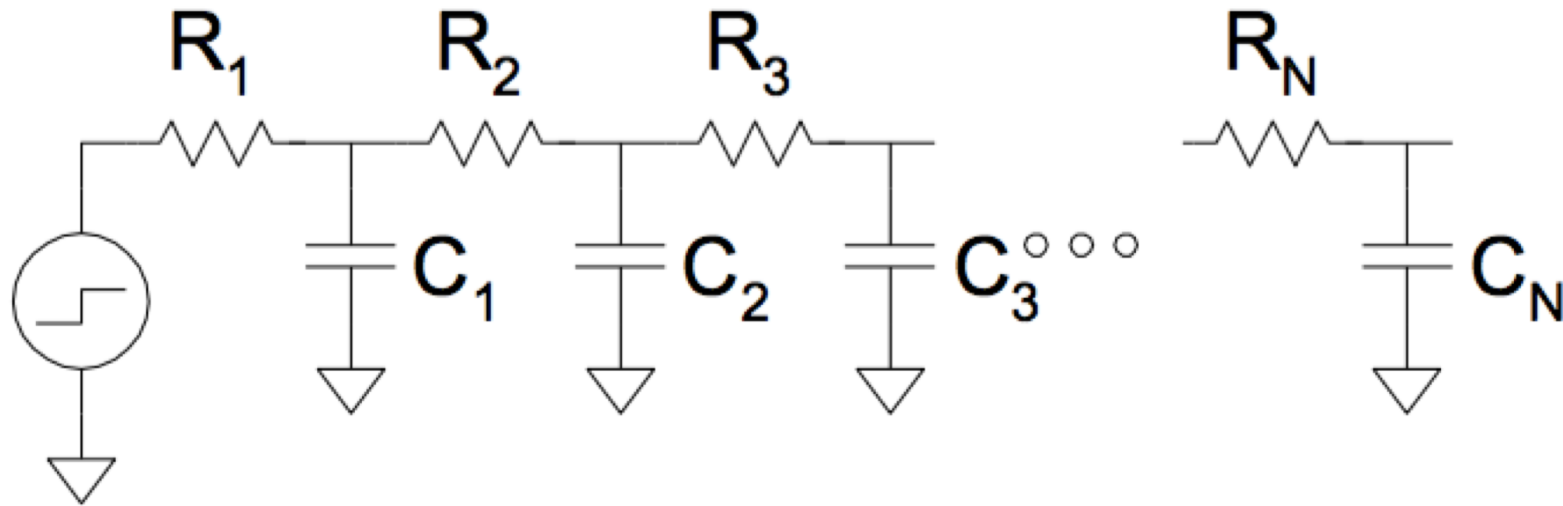
- Sketch a 3-input NAND with transistor widths chosen to achieve effective rise and fall resistances equal to a unit inverter (R)
- Annotate the 3-input NAND gate with gate and diffusion capacitance

Delay of NAND3



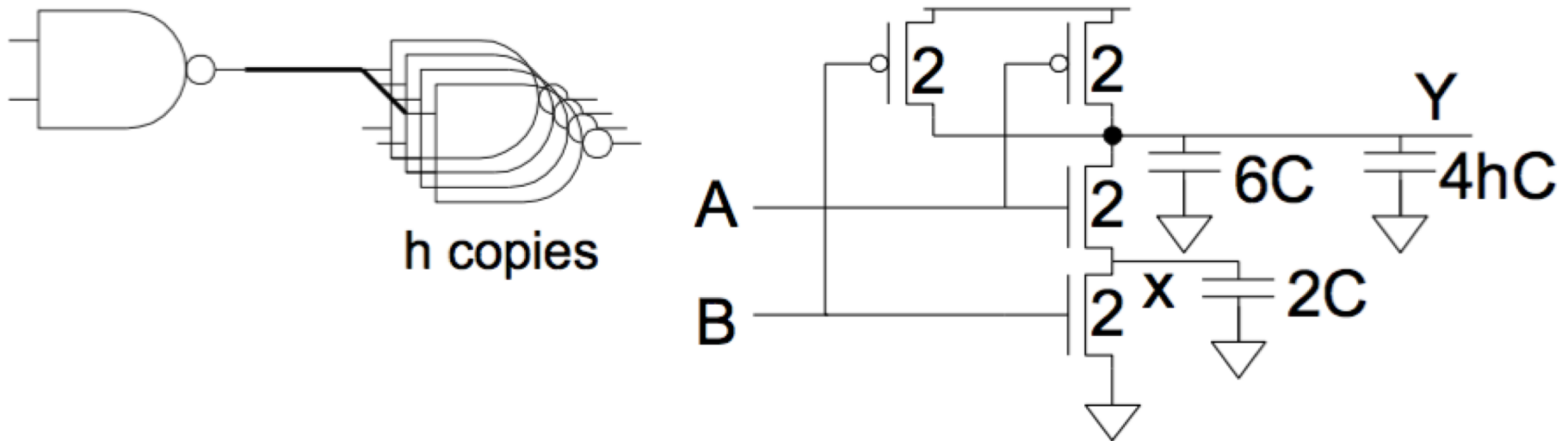
Elmore Delay Model

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



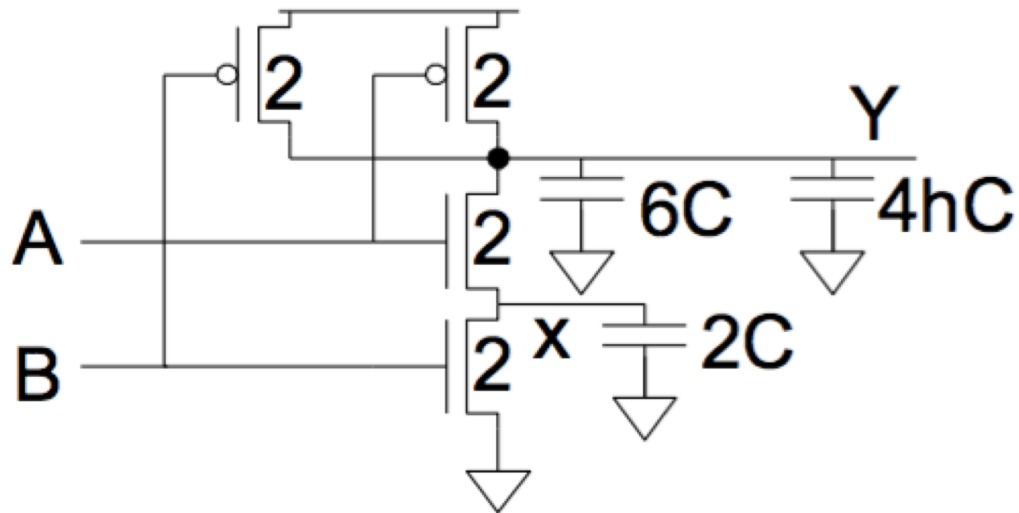
Example: 2-Input NAND

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



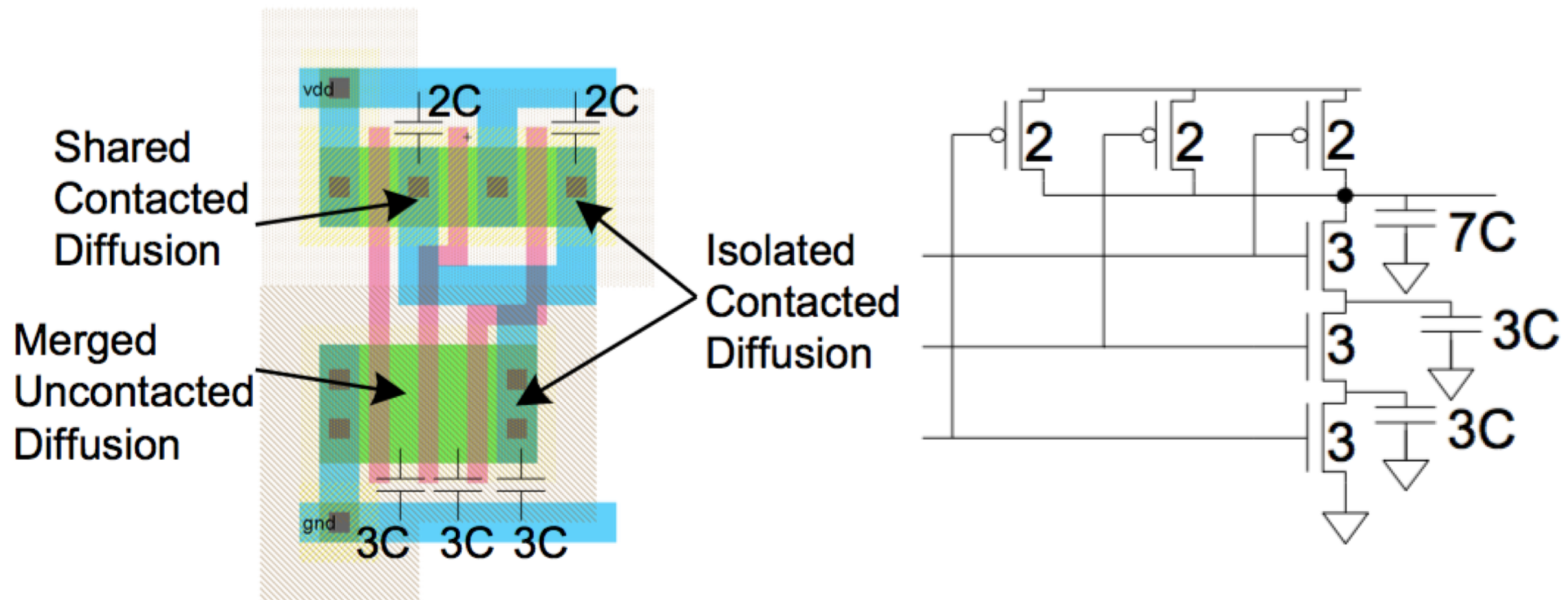
Contamination Delay

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

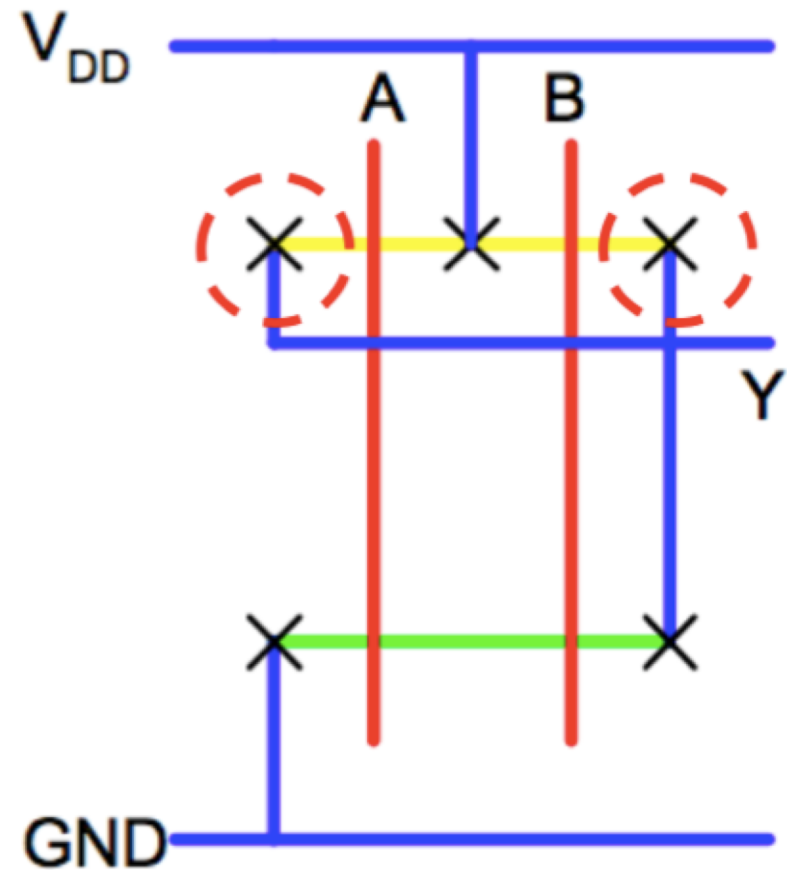
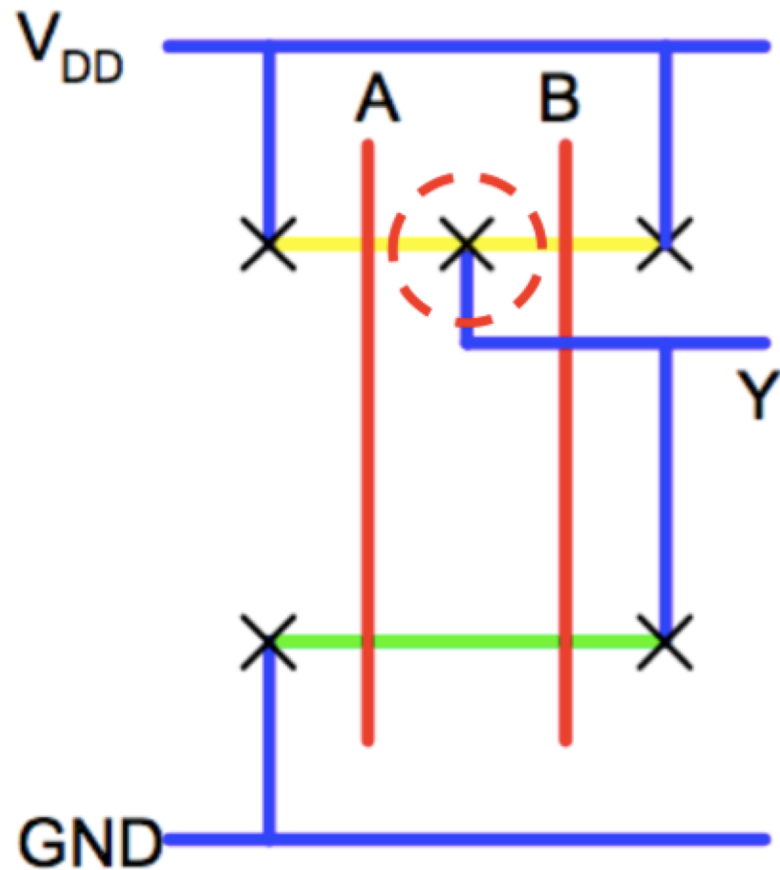


Diffusion Capacitance

- 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

- Parasitic delay
 - Independent of load
- Effort delay
 - Proportional to load capacitance

Outline

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

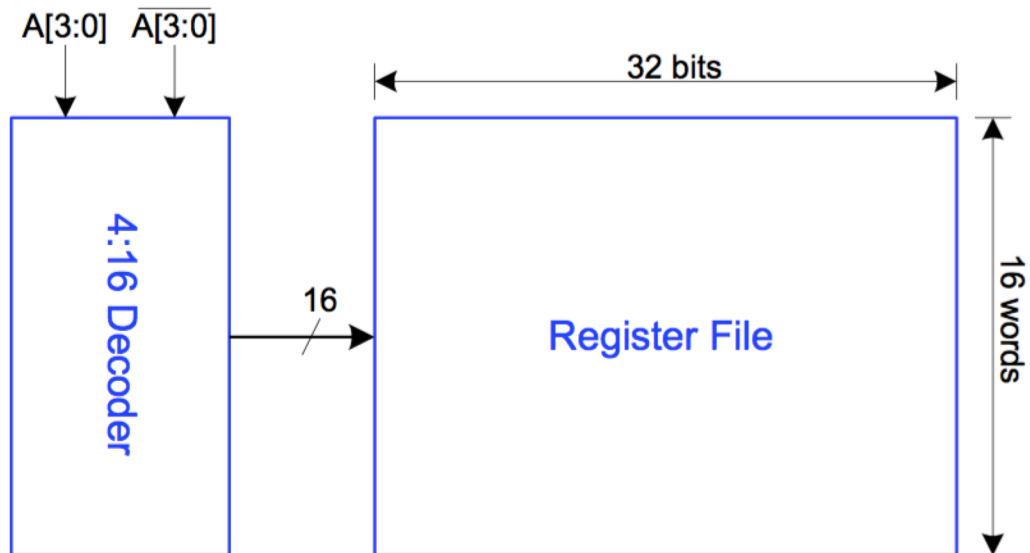
Introduction

- 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

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

- Need to decide:
 - How many stages?
 - How large should each gate be?
 - How fast can the decoder operate?



Delay of a Logic Gate

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

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

$$d = f + p$$

- Effort delay (or stage effort) has two components

$$f = gh$$

- 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

$$d = f + p$$

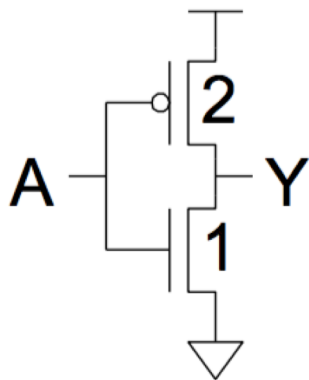
- Parasitic delay ***p***
 - Delay of gate driving no load
 - Due to internal parasitic capacitance

Computing Logical Effort

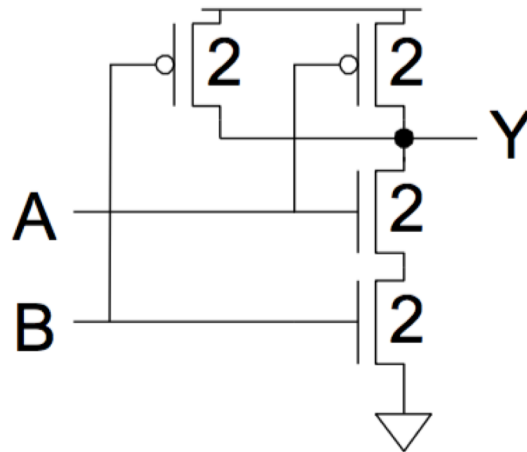
- Ratio of input capacitance of a gate to that of an inverter delivering the same output current

Method #1: Measure from delay vs. fanout plots

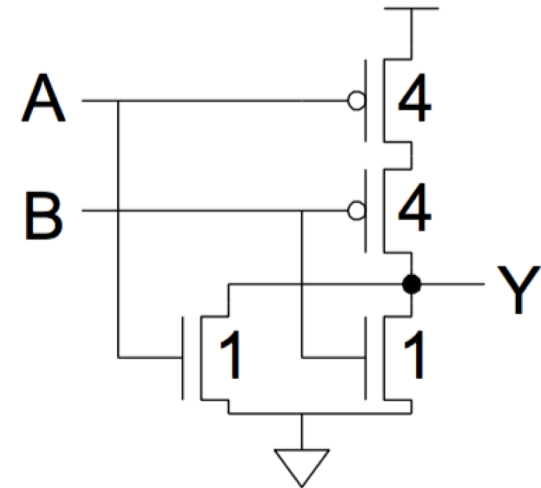
Method #2: Estimate by counting transistor widths



$$C_{in} = 3$$
$$g = 3/3$$

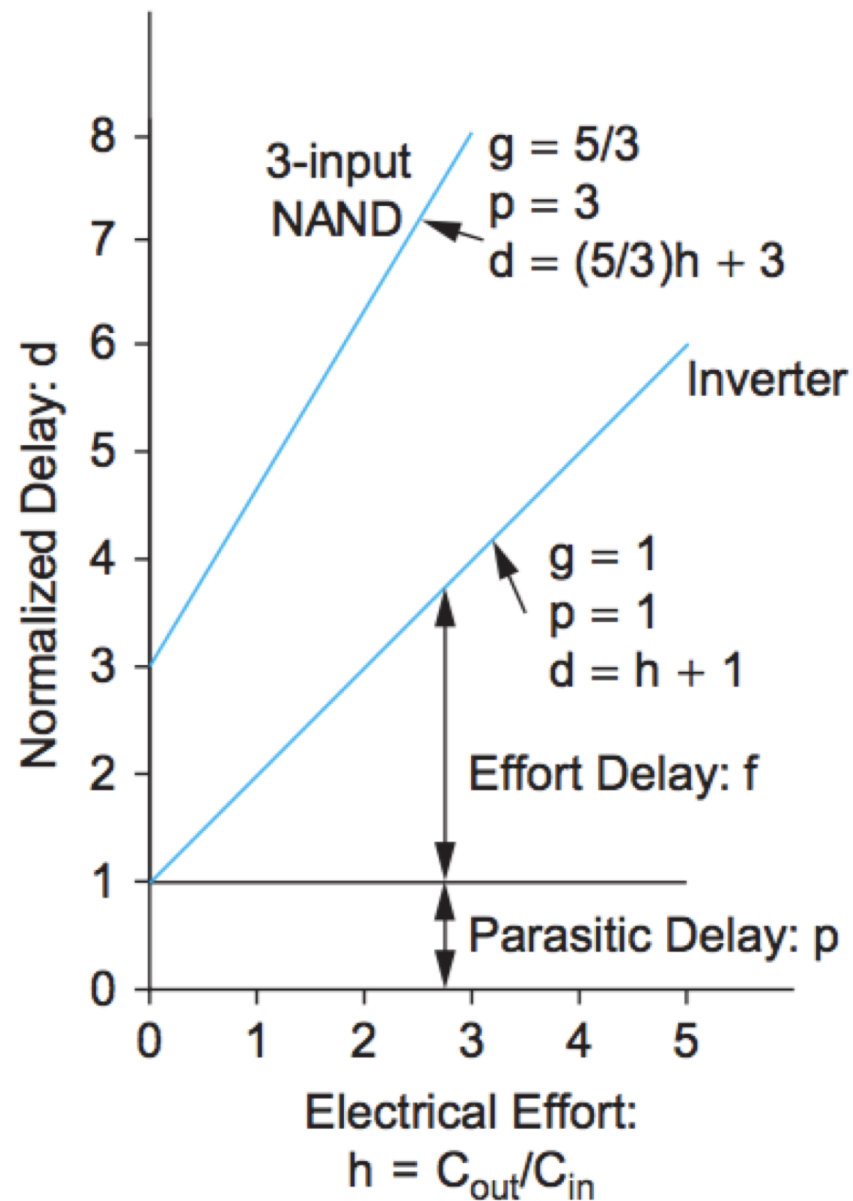


$$C_{in} = 4$$
$$g = 4/3$$



$$C_{in} = 5$$
$$g = 5/3$$

Delay vs. Fanout Plots



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	

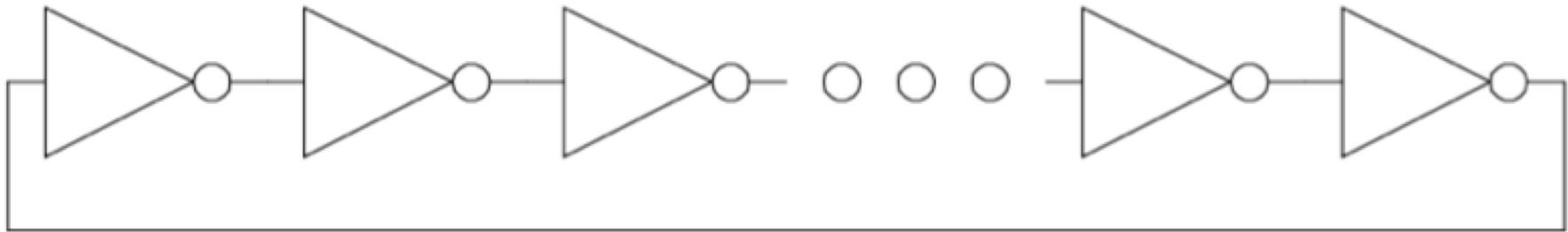
Catalog of Gates

- 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

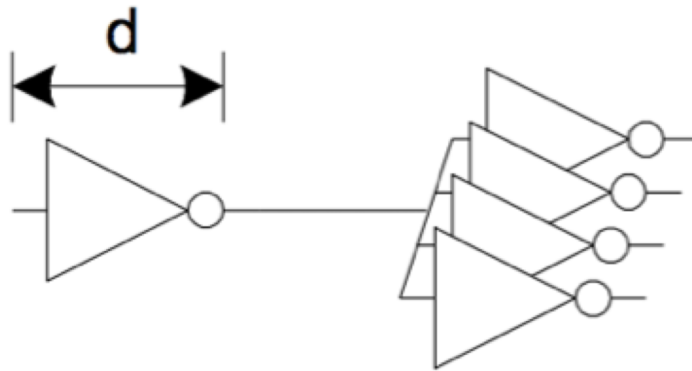
- Estimate the frequency



- Logic effort:
- Electrical effort:
- Parasitic delay:
- Stage delay:
- Frequency:

Example: FO-4 Inverter

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

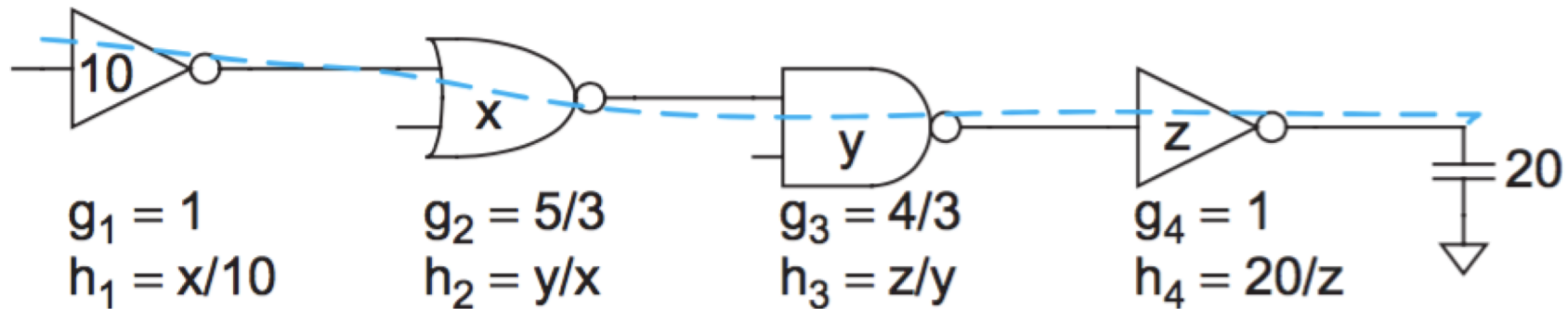


- Logic effort:
- Electrical effort:
- Parasitic delay:
- Stage delay:

- 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 $G = \prod g_i$
- Path electrical effort $H = C_{out-path} / C_{in-path}$
- Path effort $F = \prod f_i = \prod g_i h_i$



Paths with Branches

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

$$G = 1$$

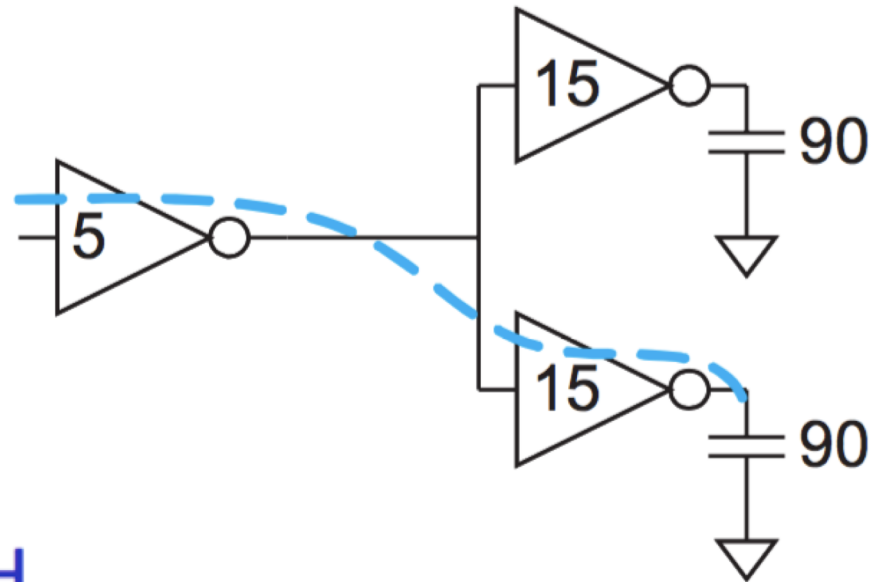
$$H = 90/5 = 18$$

$$GH = 18$$

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

$$h_2 = 90/15 = 6$$

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



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

$$F = GBH$$

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

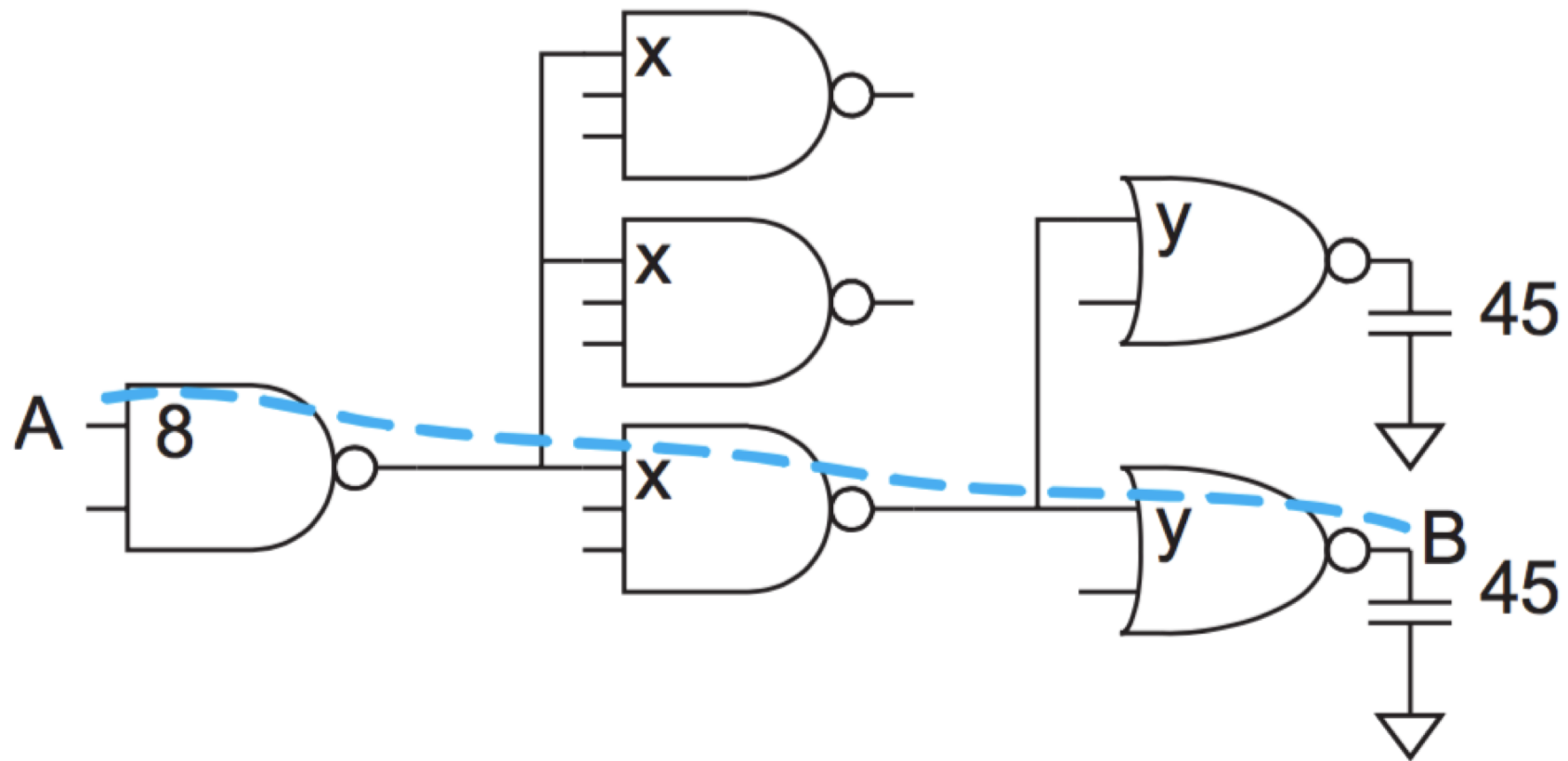
- How wide should the gates be for the least delay?

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

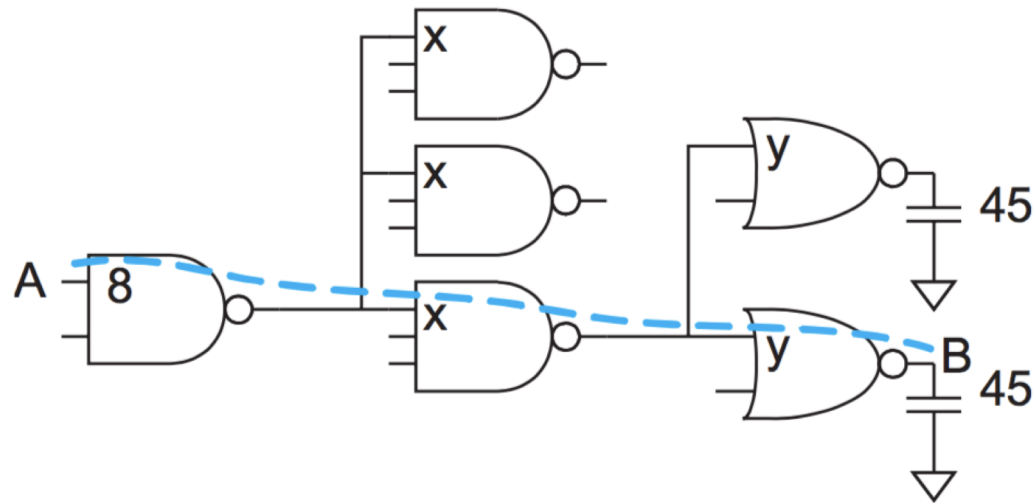
- Working backwards, apply capacitance transformation to find input capacitance of each gate with given load it drives
- Check work by verifying input cap spec is met

Example: 3-Stage Path

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



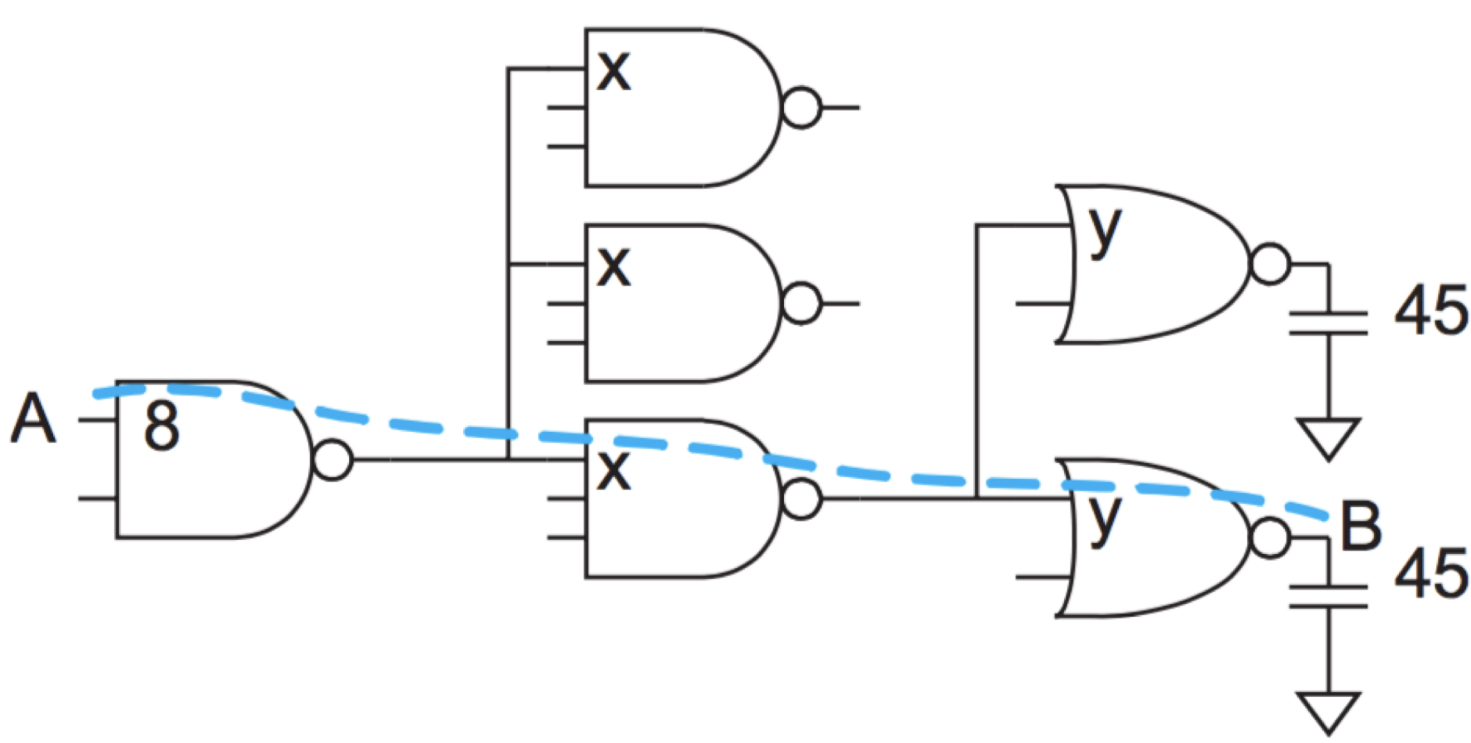
Example: 3-Stage Path



- Logical effort $G = (4/3)(5/3)(5/3) = 100/27$
- Electrical effort
- Branching effort
- Path effort $F = GBH$
- Best stage effort
- Parasitic delay $P = 2 + 3 + 2 = 7$
- Delay

Example: 3-Stage Path

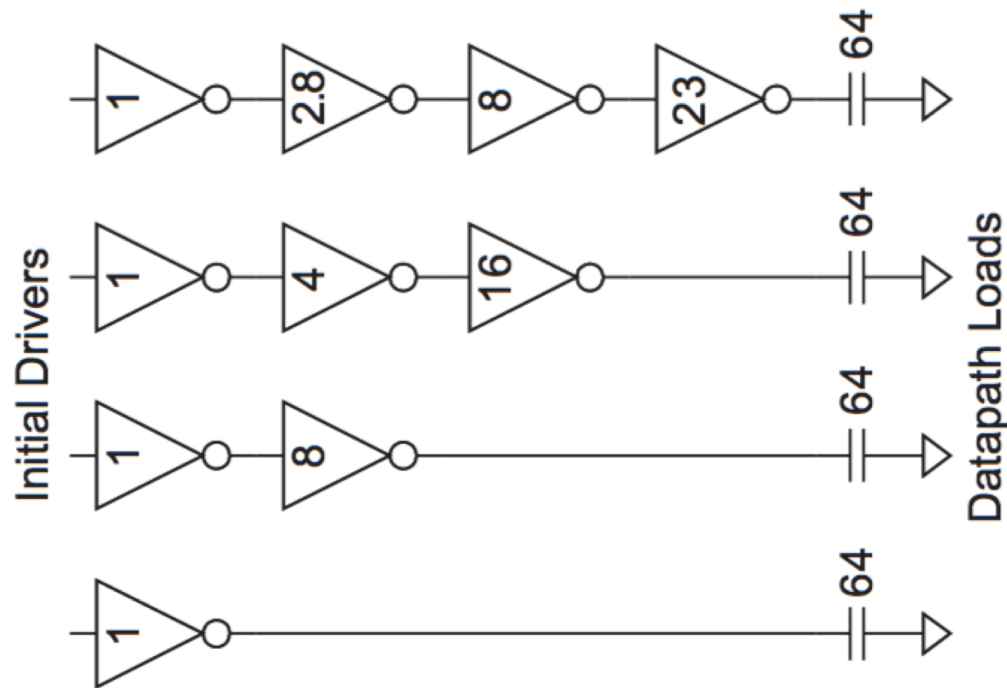
- Work backwards for sizes
 - $y = 45 * (5/3) / 5 = 15$
 - $x = (15 * 2) * (5/3) / 5 = 10$



- What about NMOS and PMOS sizes in each gate?

Best Number of Stages

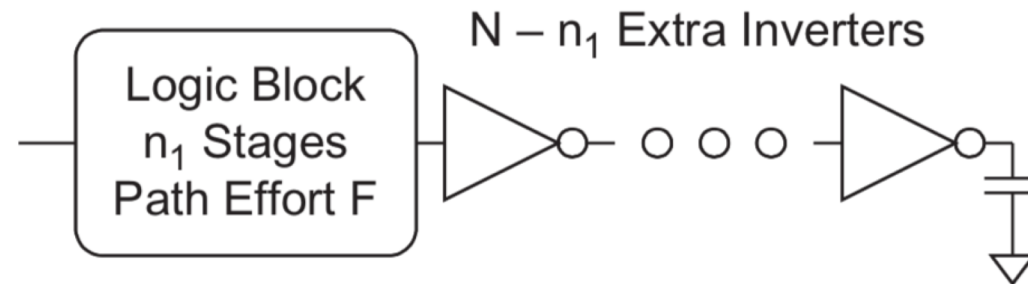
- How many stages should a path use
 - Minimizing number of stages is not always the fastest
- Example: Drive 64-bit datapath with unit inverter



N	t	D

Derivation

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



$$D = NF^{\frac{1}{N}} + \sum_{i=1}^{n_1} p_i + (N - n_1) p_{inv}$$
$$\frac{\partial D}{\partial N} = -F^{\frac{1}{N}} \ln F^{\frac{1}{N}} + F^{\frac{1}{N}} + p_{inv} = 0$$

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

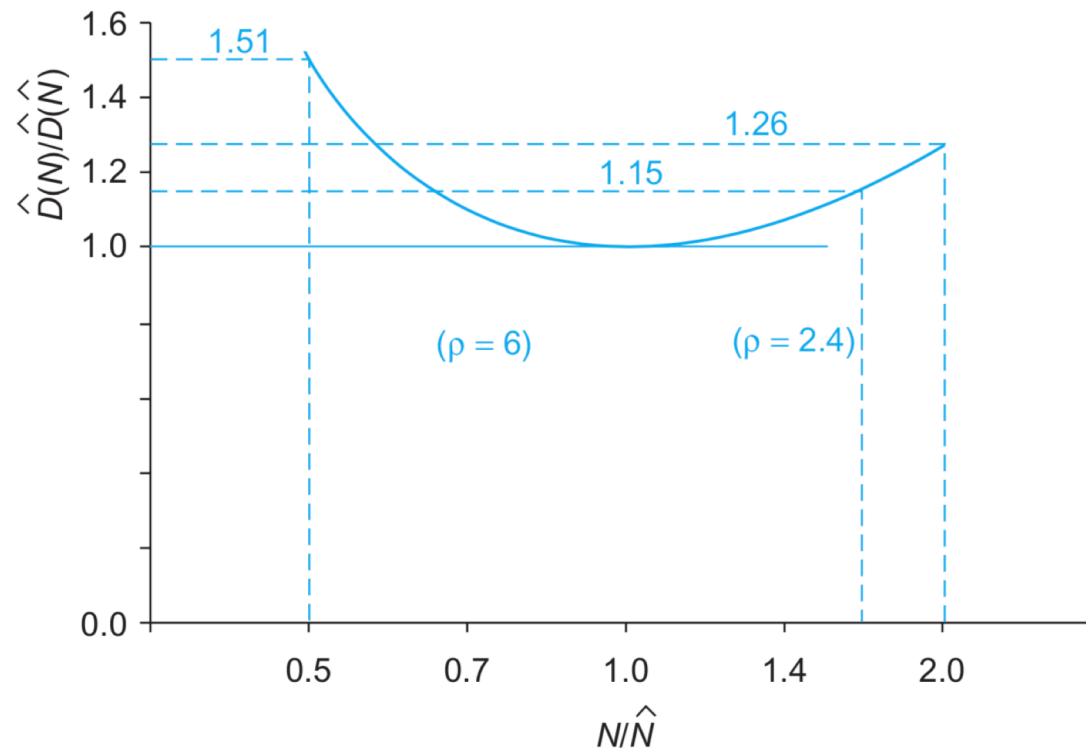
$$p_{inv} + \rho(1 - \ln \rho) = 0$$

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?



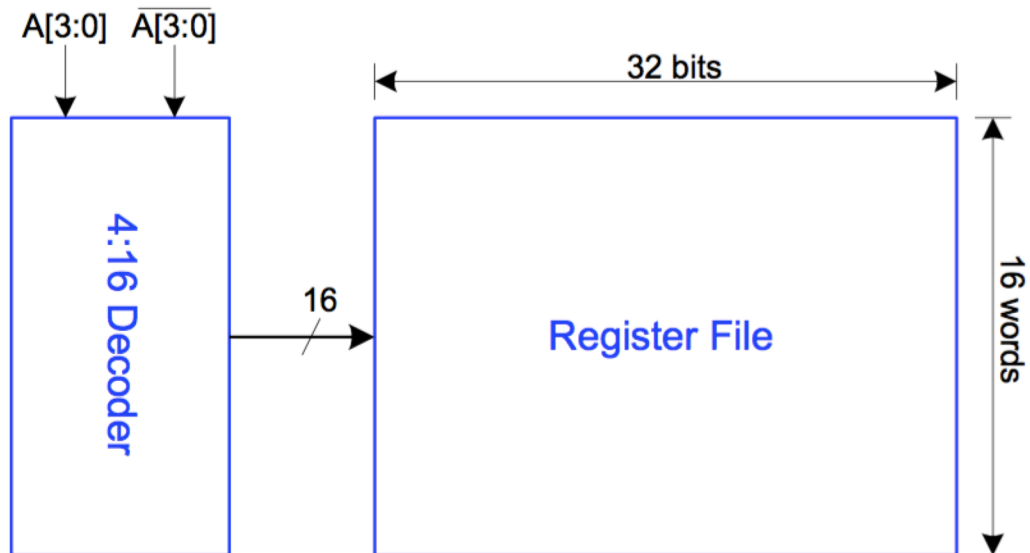
- $2.4 < \rho < 6$ gives delay with 15% variations
 - 4 is a common choice

Example: Decoder for a Register File

(revisited)

- 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

- Need to decide:
 - How many stages?
 - How large should each gate be?
 - How fast can the decoder operate?



Number of Stages

- Decoder effort is mainly electrical and branching

Electrical effort $H =$

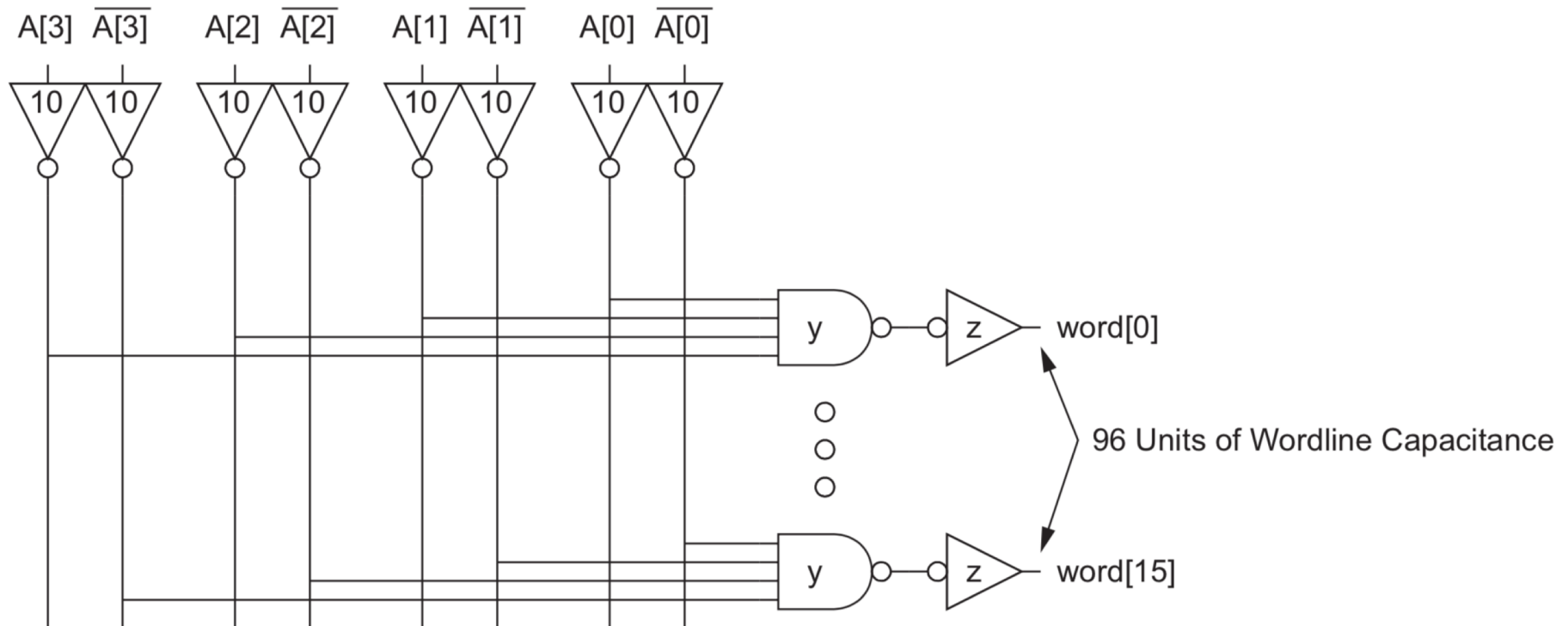
Branching effort $B =$

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

Path effort $F =$

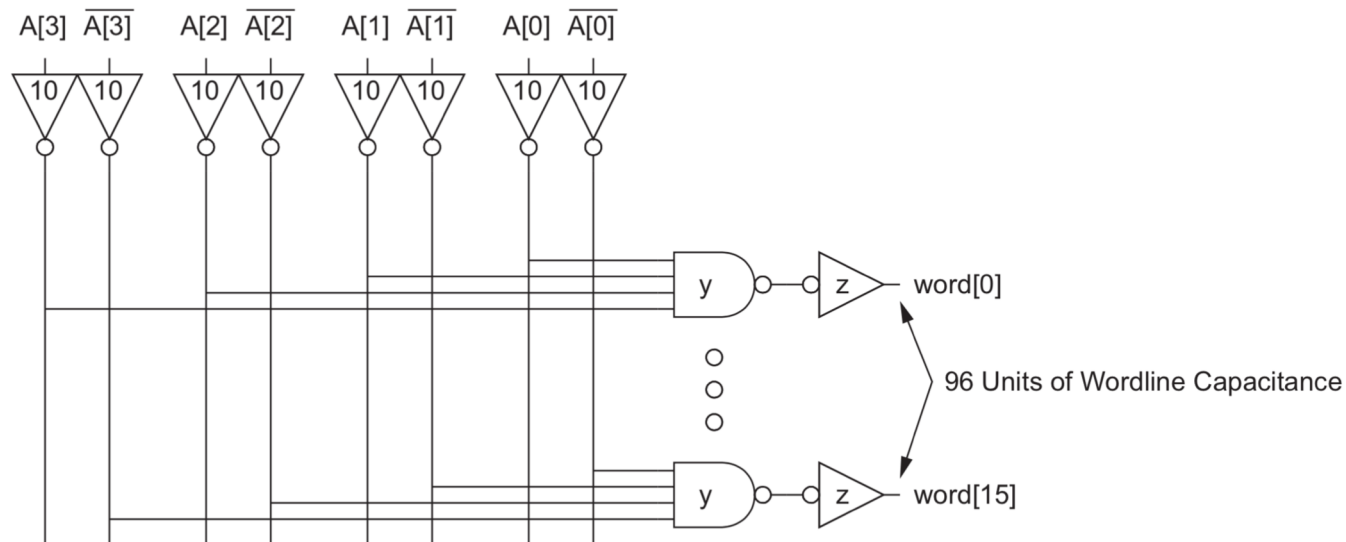
Number of stages $N =$

3 Stage 4:16 Decoder



Gate Sizes and Delay

- Logical effort $G =$
- Path effort $F =$
- Stage effort $\hat{f} =$
- Path delay $D =$
- Gate sizes $z =$ $y =$



Comparison

- Different alternatives

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

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

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