Lecture 6: Gate Delay

Based on material prepared by prof. Visvesh S. Sathe

ELECTRICAL & COMPUTER ENGINEERING

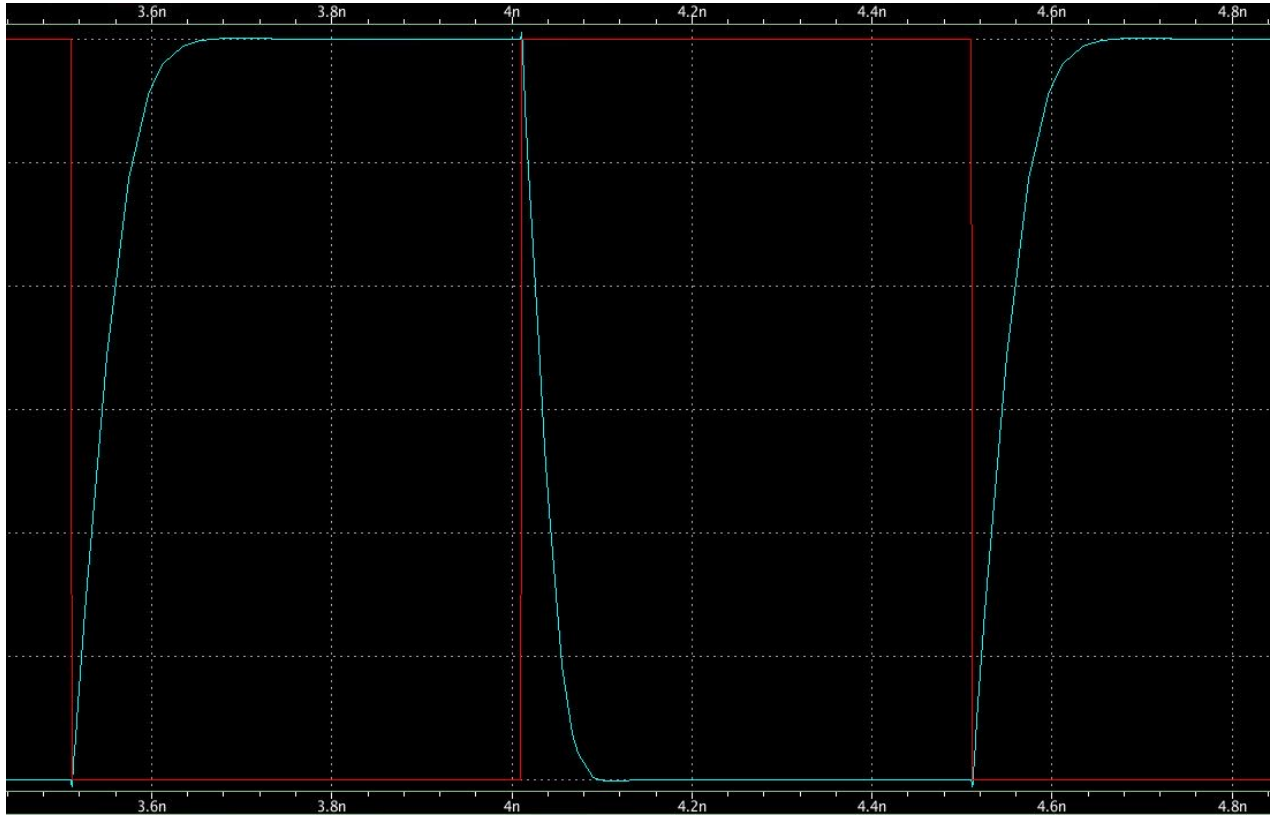
Acknowledgements

All class materials (lectures, assignments, etc.) based on material prepared by Prof. Visvesh S. Sathe, and reproduced with his permission

Visvesh S. Sathe
Associate Professor
Georgia Institute of Technology
https://psylab.ece.gatech.edu

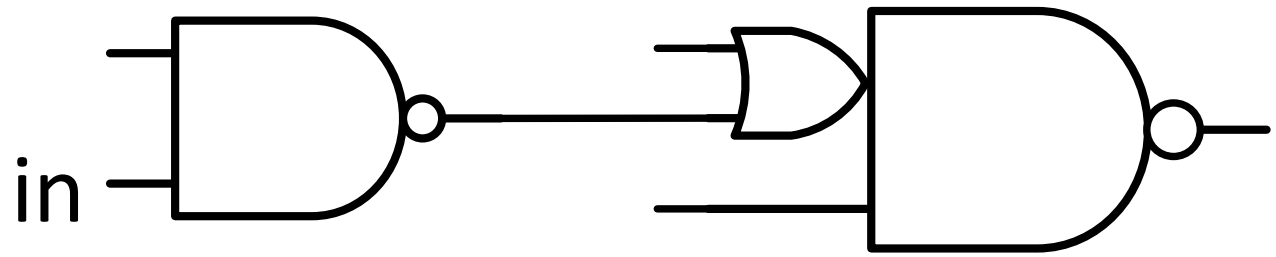
UW (2013-2022)
GaTech (2022-present)

CMOS Inverter Delay



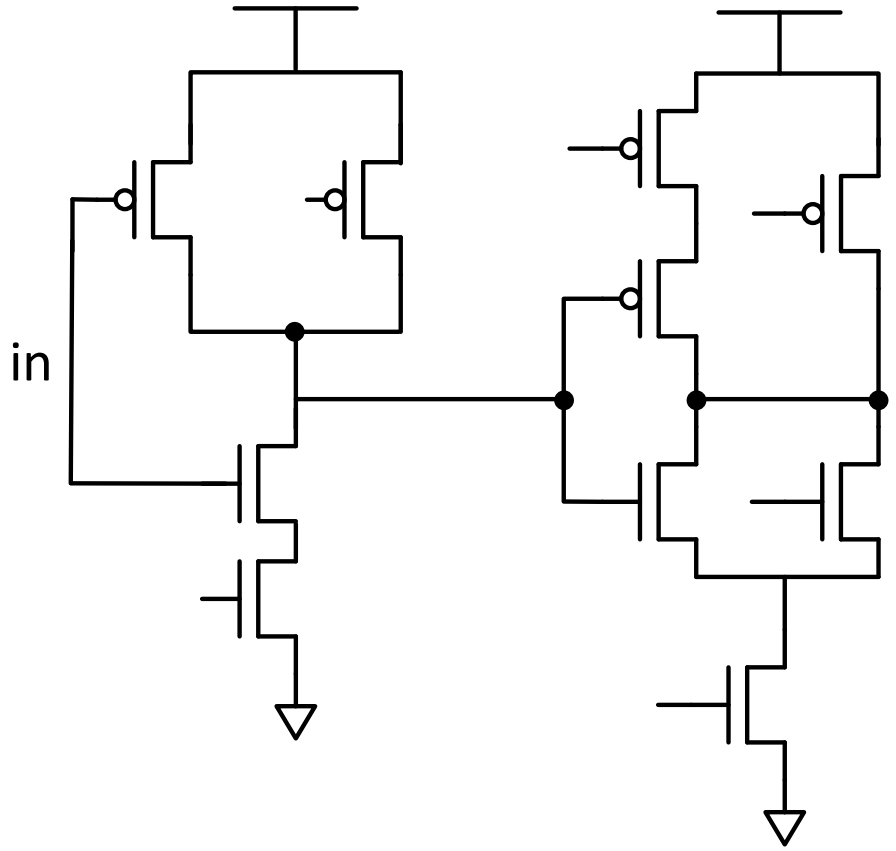
- Definitions:
  - Rise delay - Input 50% fall to output 50% rise
  - Fall delay - Input 50% rise to output 50% fall
  - Rise time – Output 20% rise to 80% rise
  - Fall time – Output 80% fall to 20% fall

Simple Delay Model



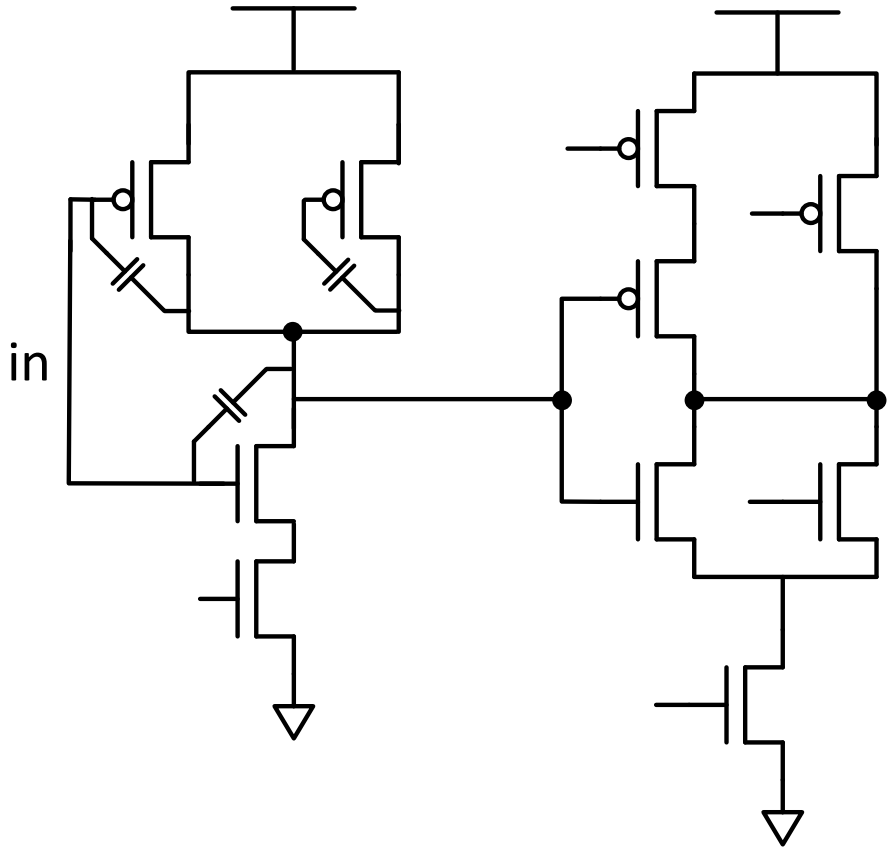
- Gate delays determine the time taken to compute a boolean function

Simple Delay Model



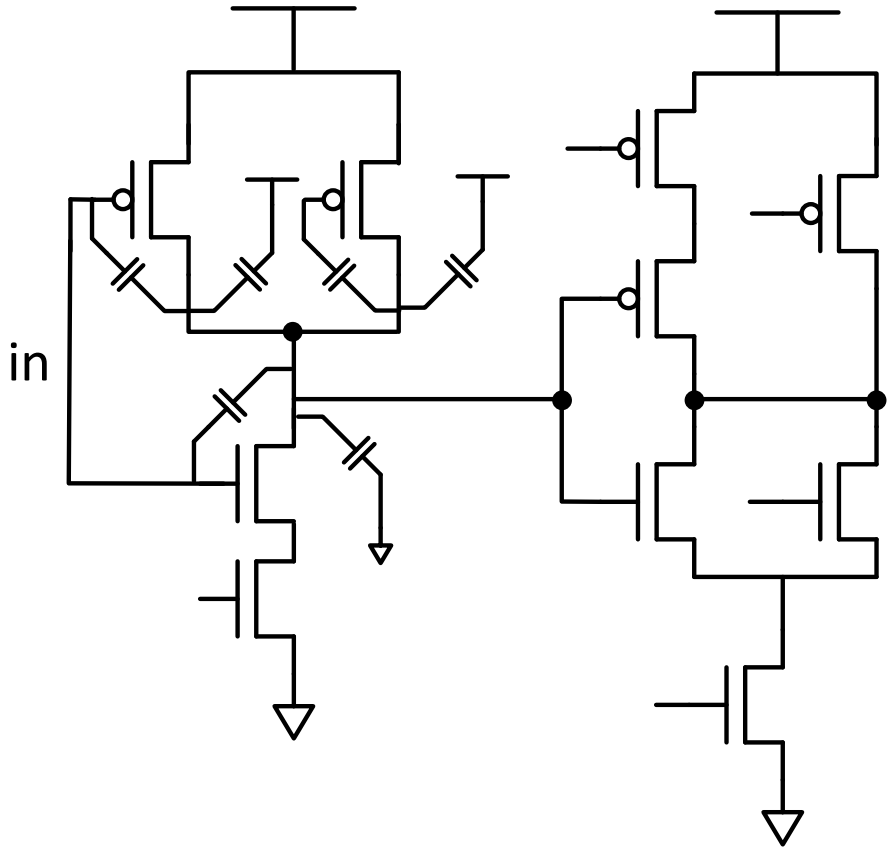
- Pmos or Nmos device drive capacitive load to charge (discharge) output to logic 1 (0)

Simple Delay Model



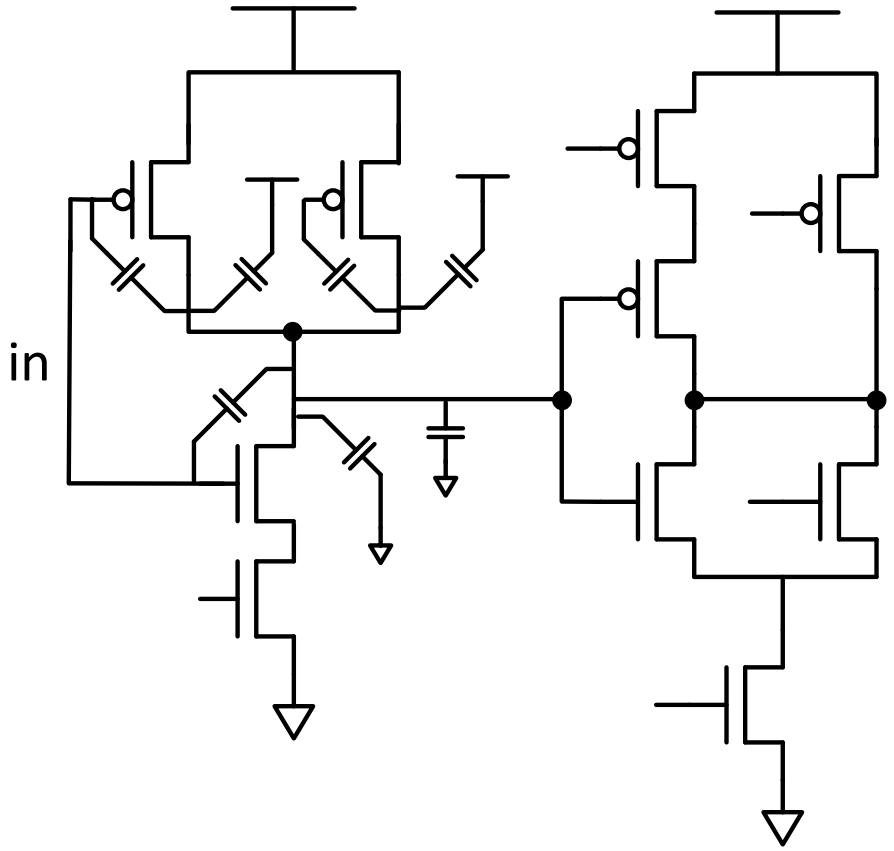
- Pmos or Nmos device drive capacitive load to charge (discharge) output to logic 1 (0)

Simple Delay Model



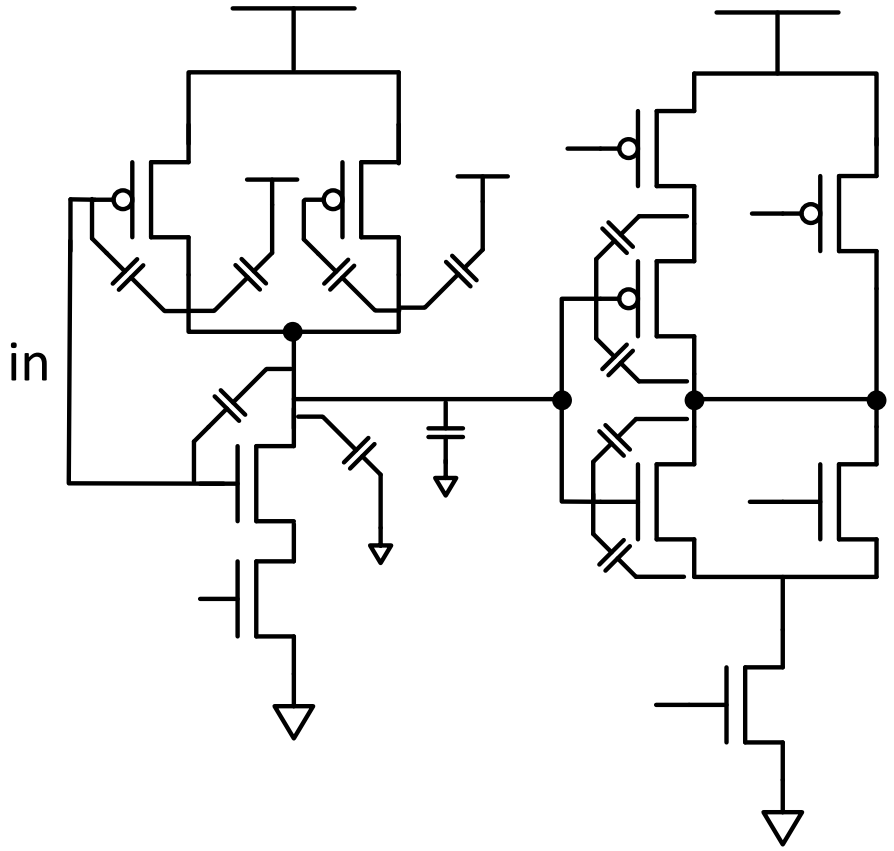
- Pmos or Nmos device drive capacitive load to charge (discharge) output to logic 1 (0)

Simple Delay Model



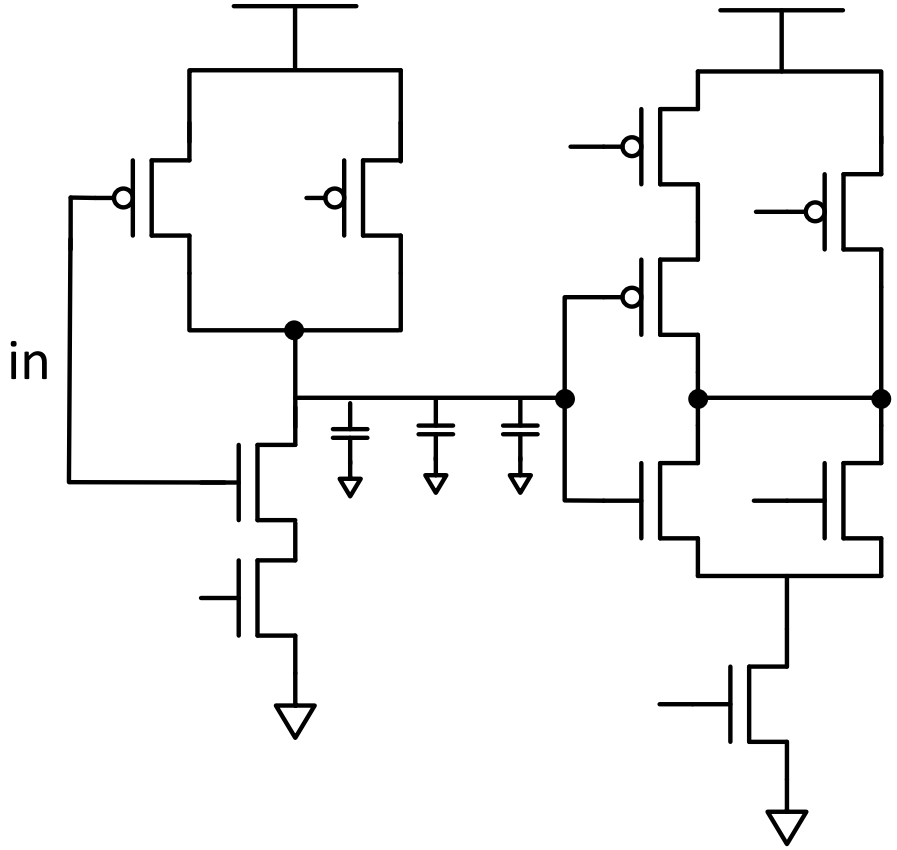
- Pmos or Nmos device drive capacitive load to charge (discharge) output to logic 1 (0)

Simple Delay Model



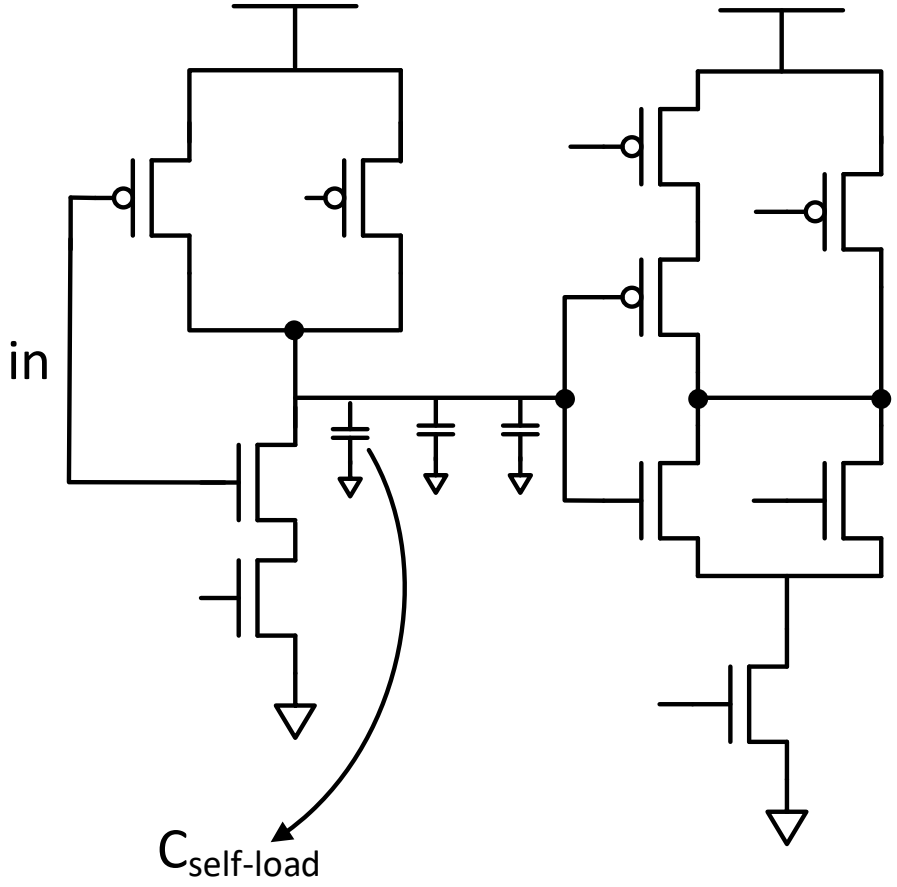
- Pmos or Nmos device drive capacitive load to charge (discharge) output to logic 1 (0)

Simple Delay Model



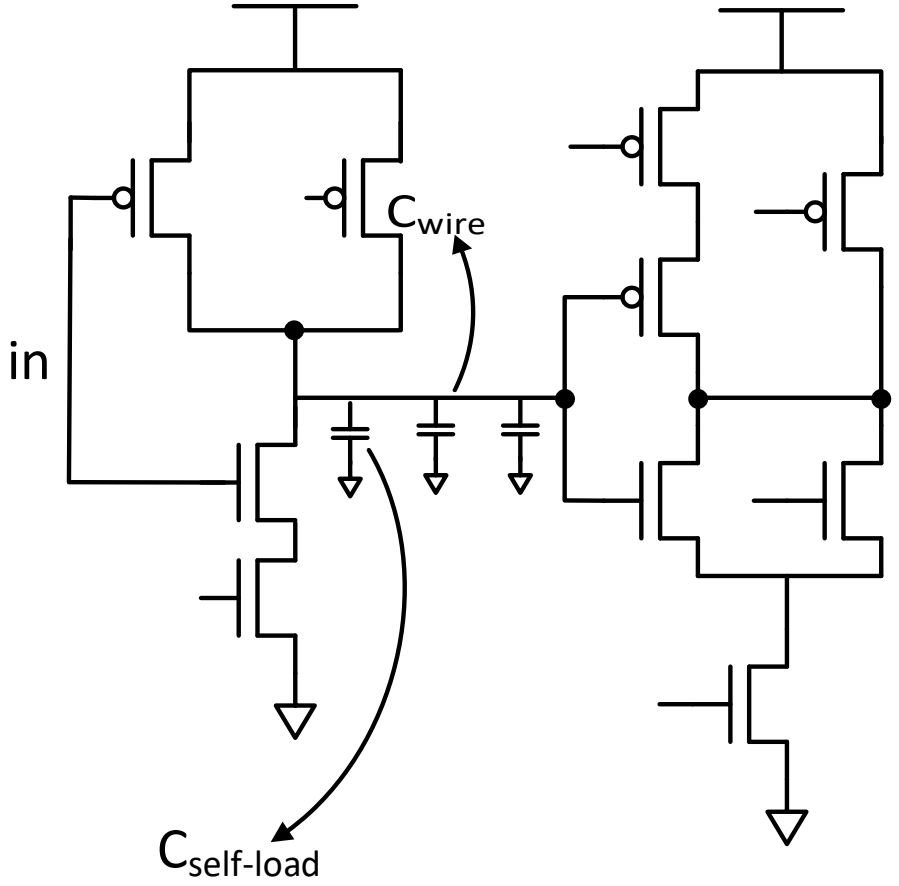
- Effectively 3 components

Simple Delay Model



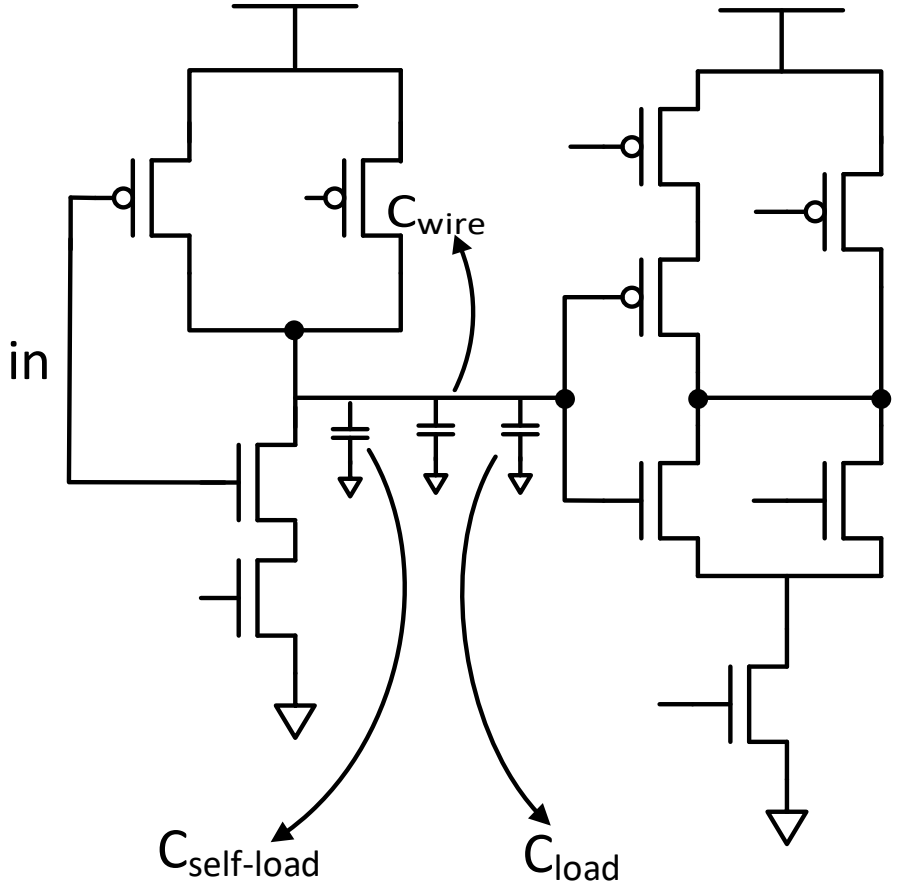
- Effectively 3 components

Simple Delay Model



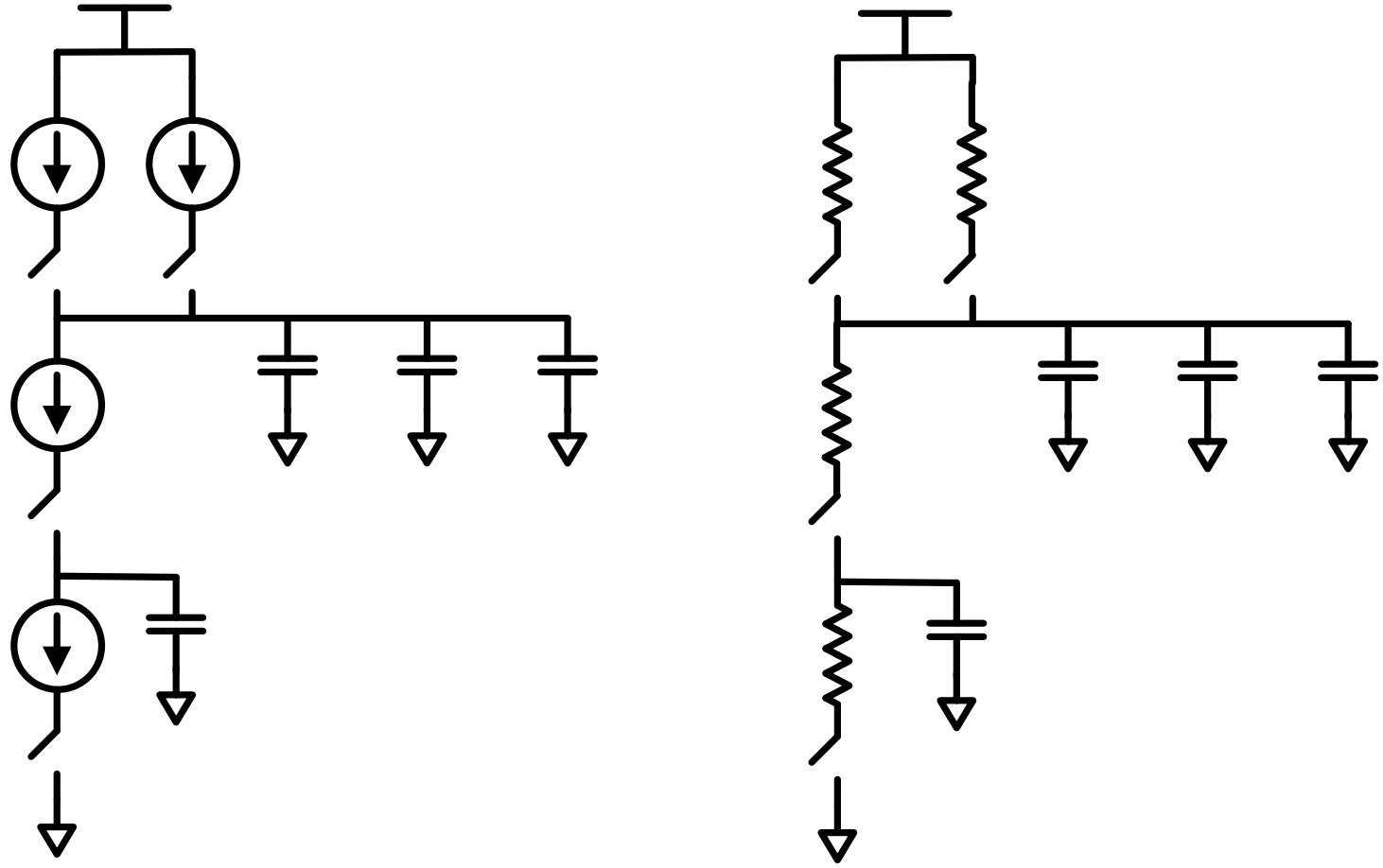
- Effectively 3 components

Simple Delay Model



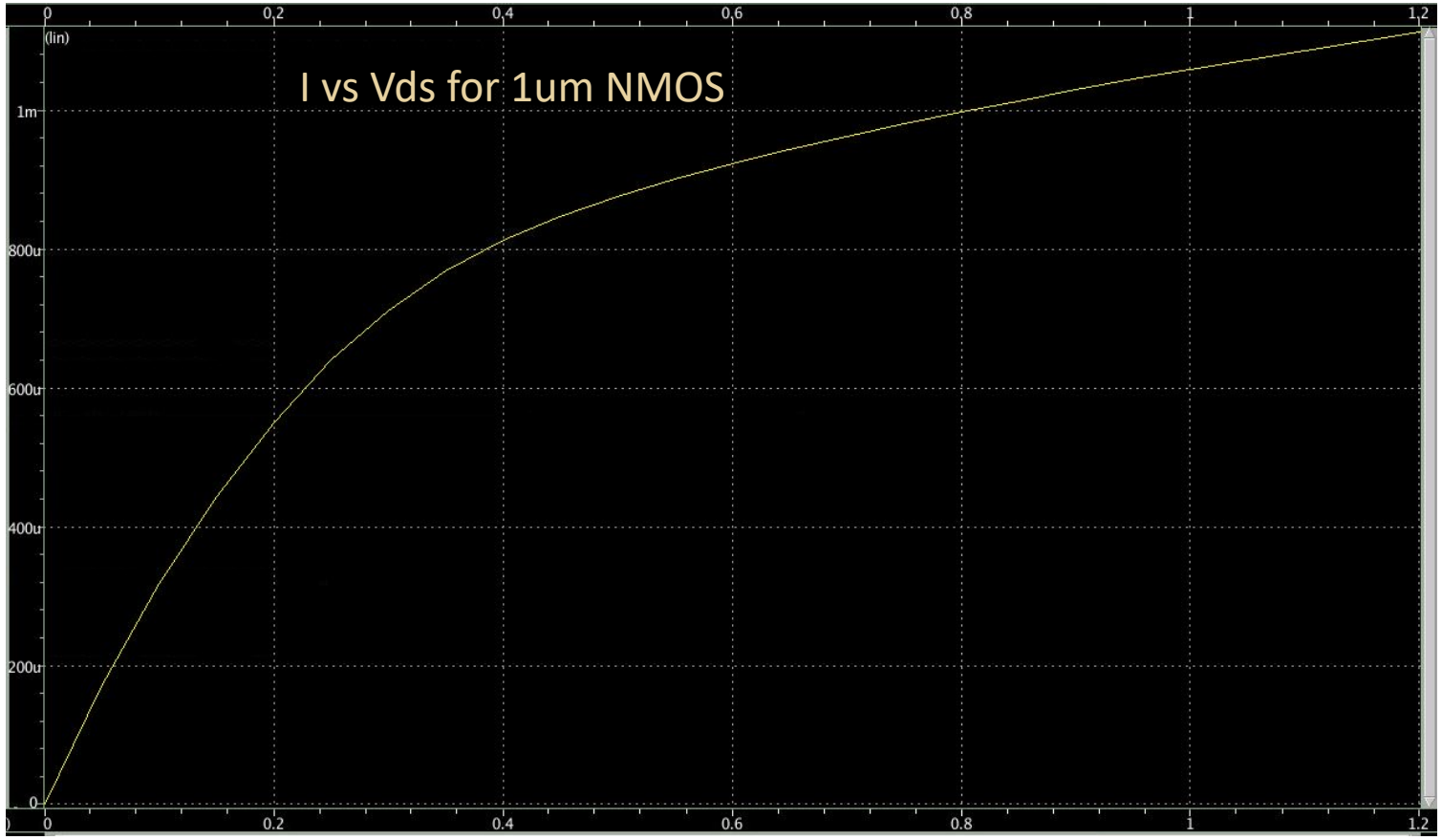
- Effectively 3 components

Simple Delay Model



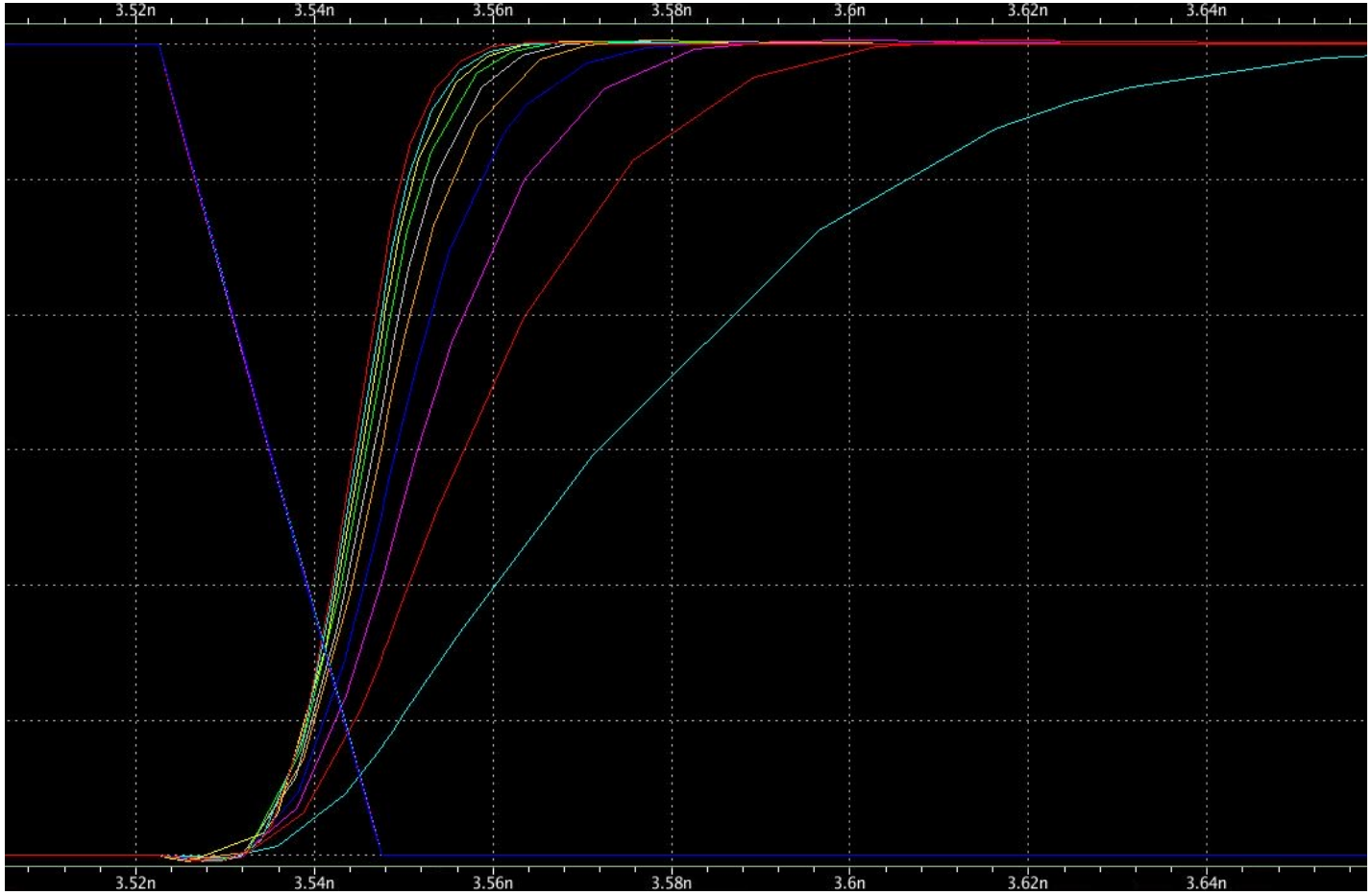
- Circuit delay viewed as
  - Voltage-dependent current source charging/discharging capacitance
  - Voltage-dependent resistance charging/discharging capacitance

Charging/Discharging Current



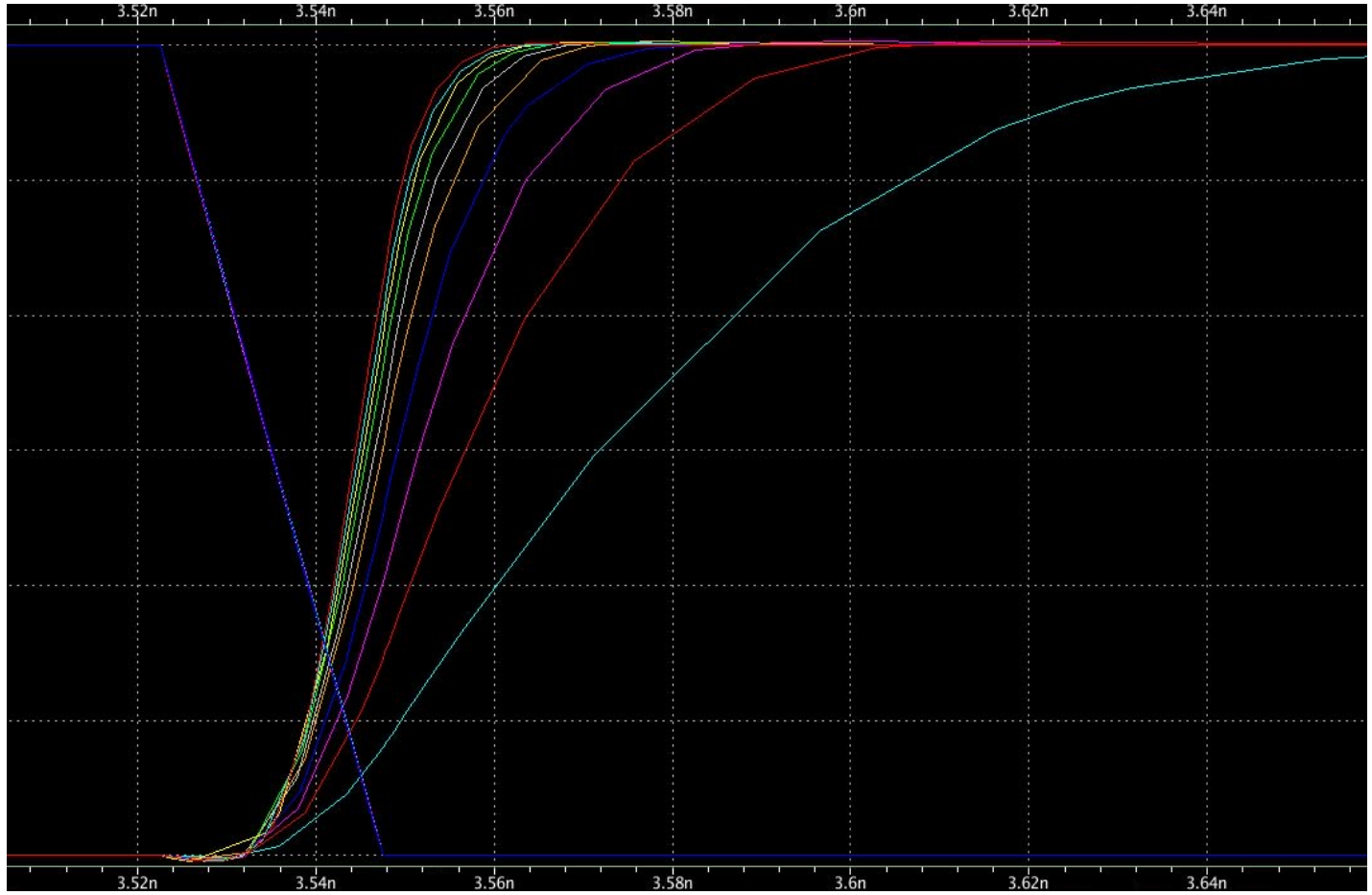
- Gate evaluating to Vdd
  - PMOS operating in different regions
  - Working out exact equations is not productive
  - Design perspective: Linear or Saturation, current drive scaled by W, 1/L

Delay Dependence on W



- Does delay scale as 1/W?

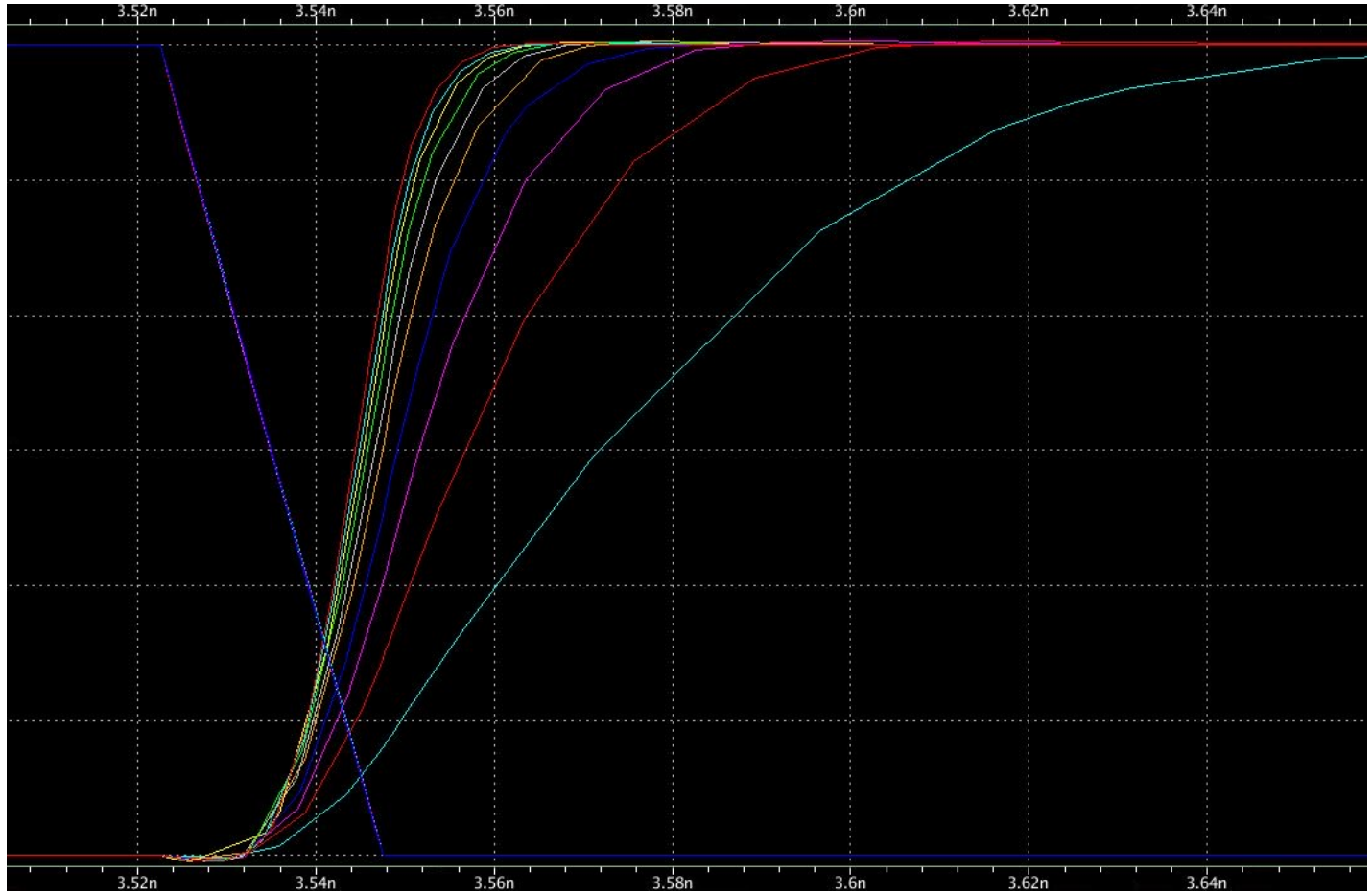
Delay Dependence on W



If I have 2 cascaded inverters and the scale, x of the driving inverter is variable, how does the delay vary with x

- Does delay scale as 1/W?

Delay Dependence on W

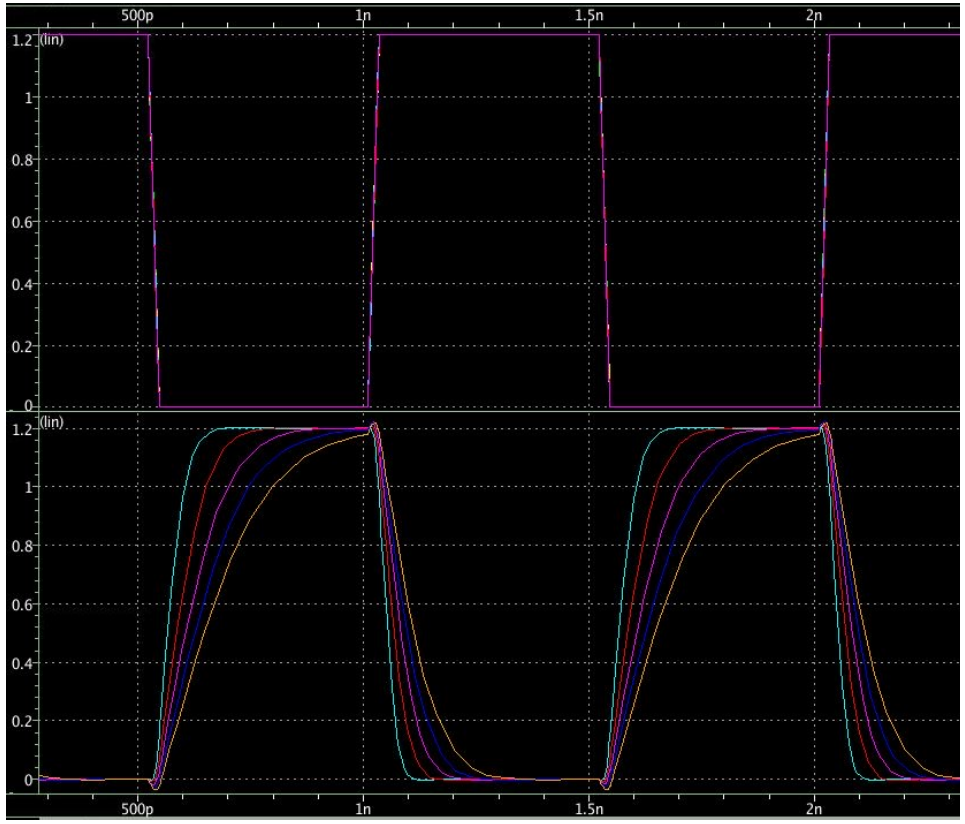


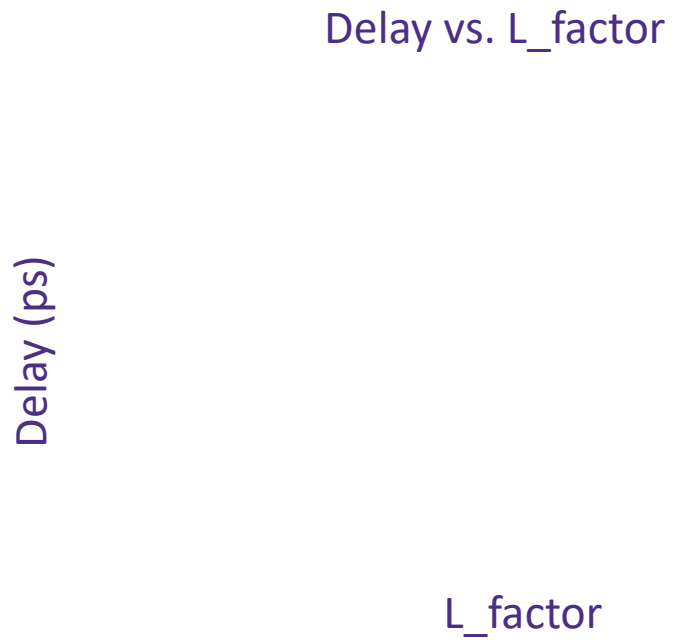
If I have 2 cascaded inverters and the scale, x of the driving inverter is variable, how does the delay vary with x

Self-loading!

- Does delay scale as 1/W?

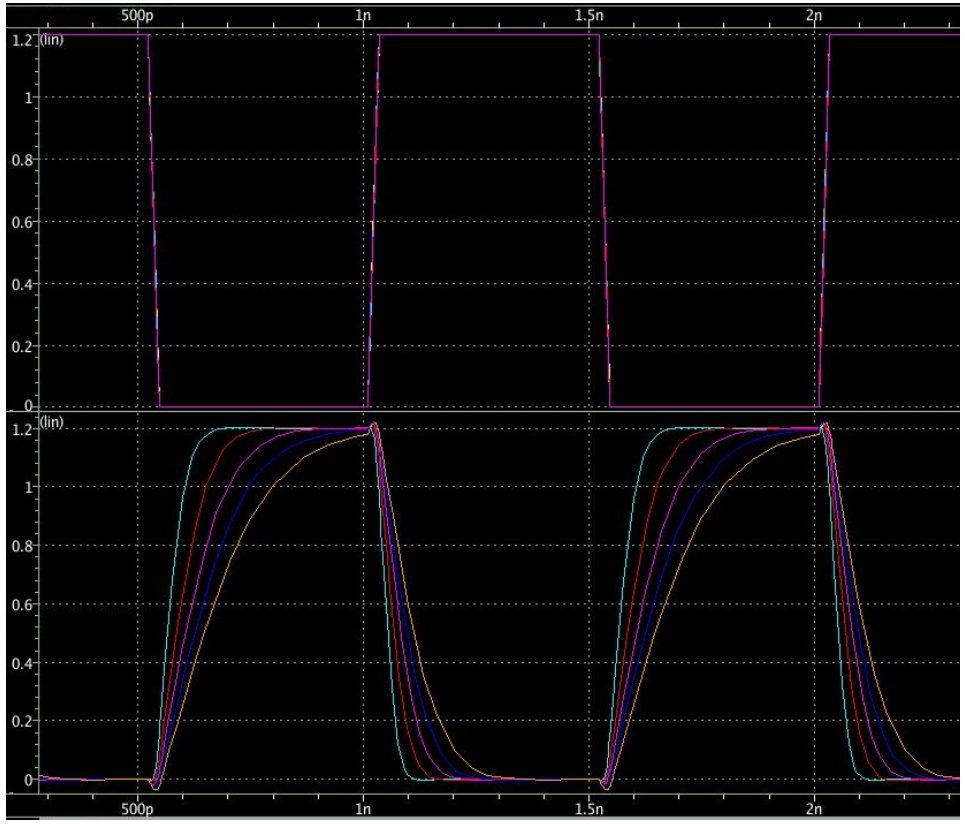
Delay Dependence on L

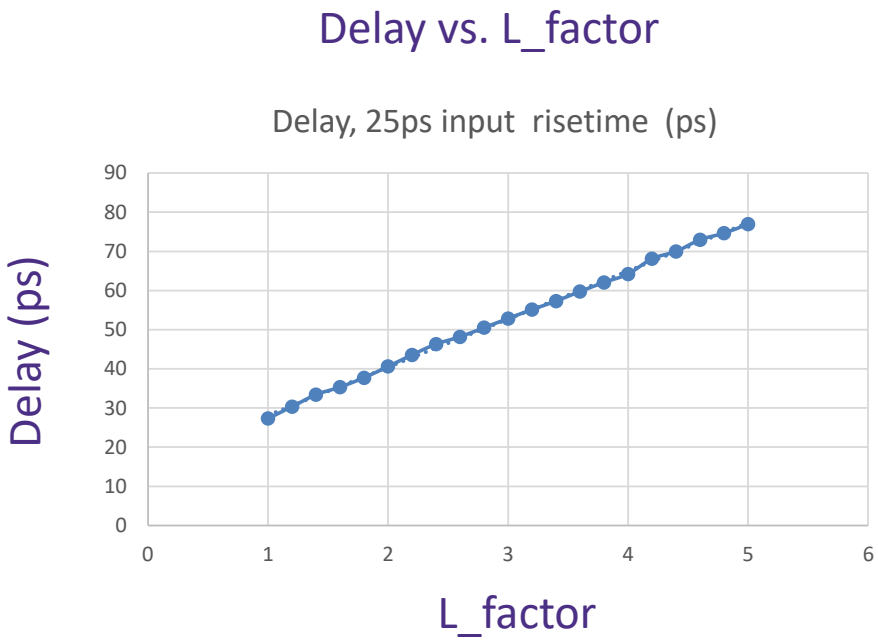




- If W and L are both designer tunable, why do most digital designs tweak only W?

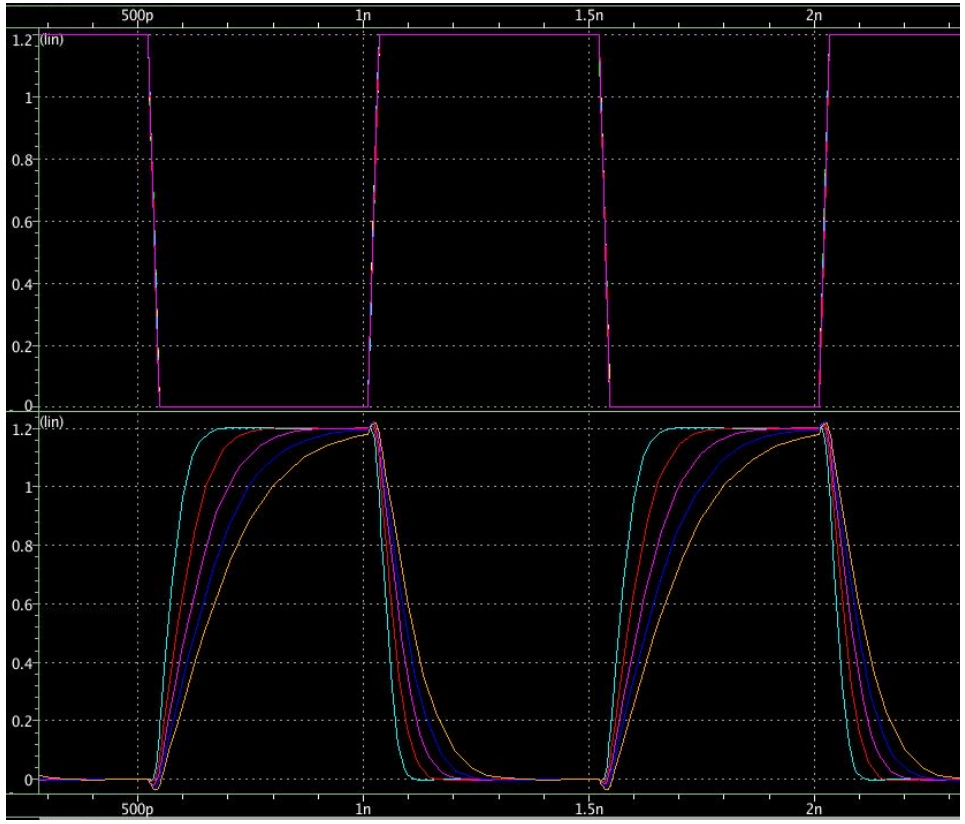
Delay Dependence on L

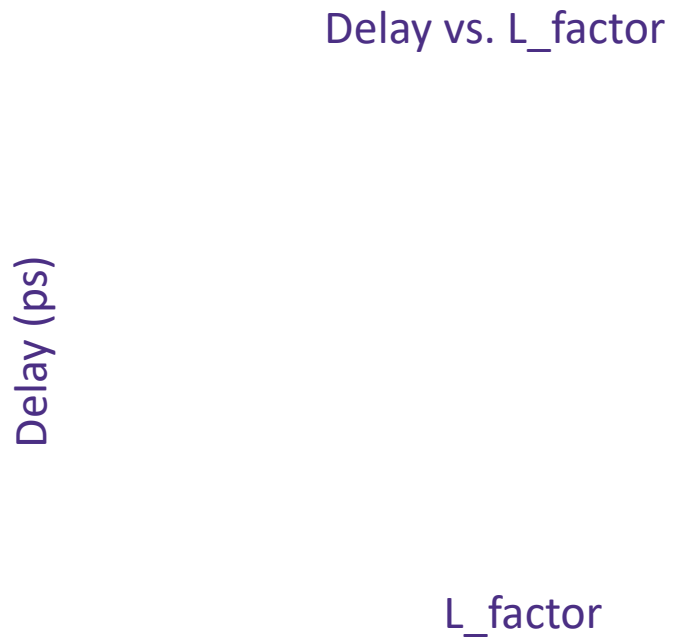




- If W and L are both designer tunable, why do most digital designs tweak only W?

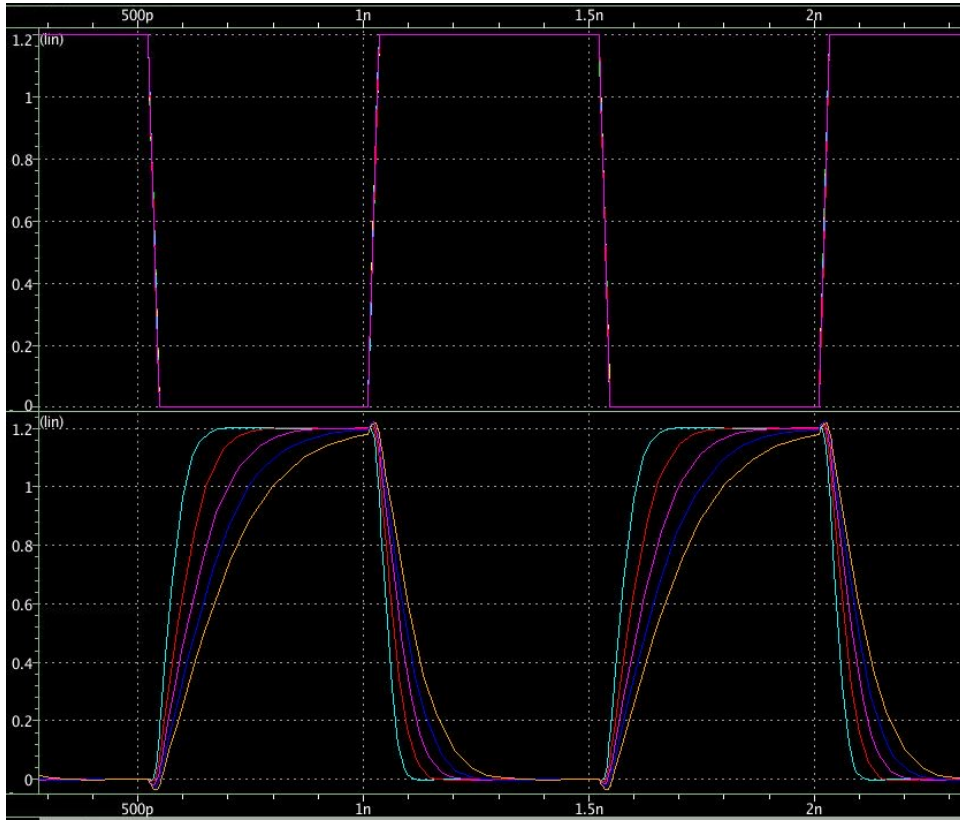
Delay Dependence on L





- If W and L are both designer tunable, why do most digital designs tweak only W?

Delay Dependence on L



Delay vs. L_factor

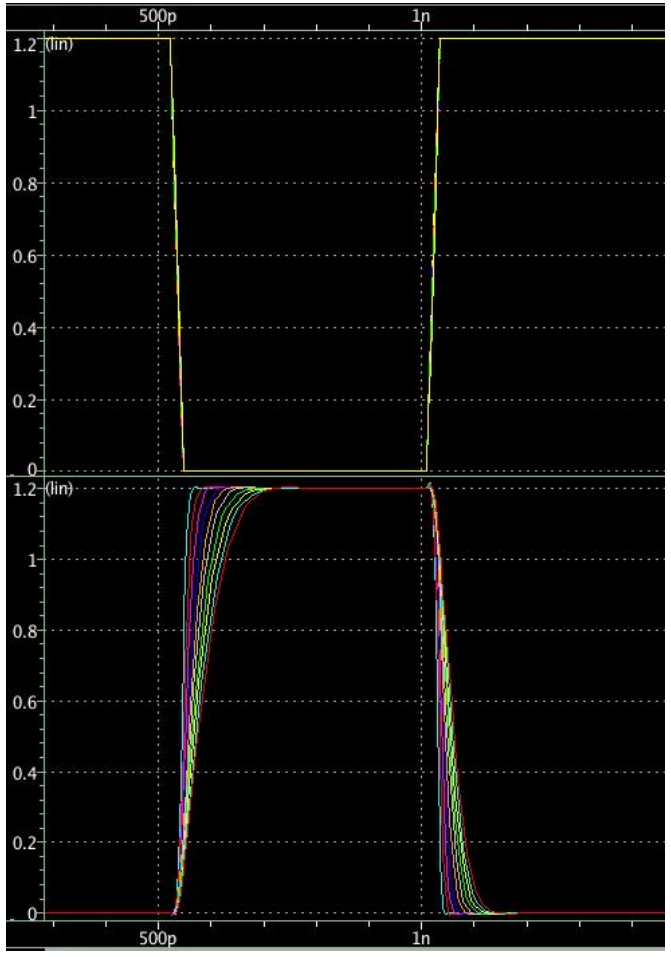
Delay (ps)

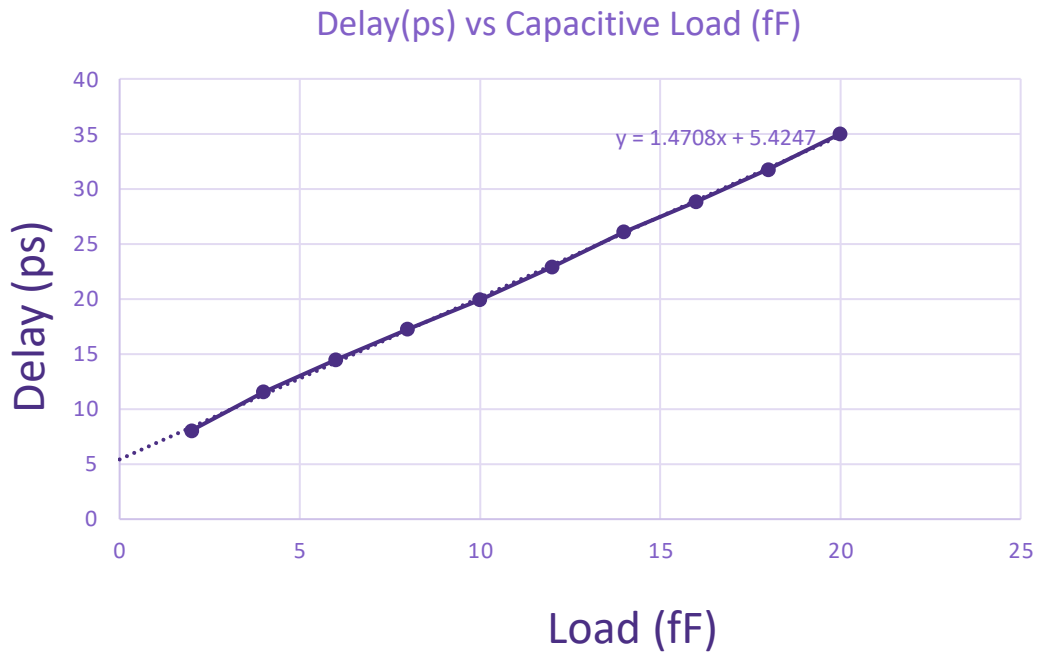
L_factor

Extra:
Relationship with
L is Linear (Affine, y-intercept)

- If W and L are both designer tunable, why do most digital designs tweak only W?

Delay Dependence on C





- Self loading causes non-zero intercept

Delay Dependence on Vdd

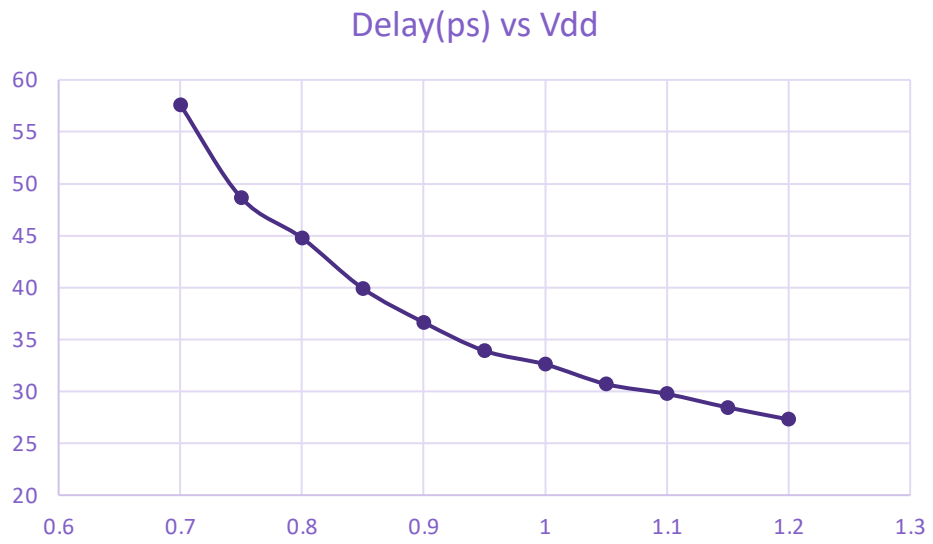
Delay Dependence on Vdd

- $\uparrow V_{dd} \rightarrow \downarrow$Delay

Delay Dependence on Vdd

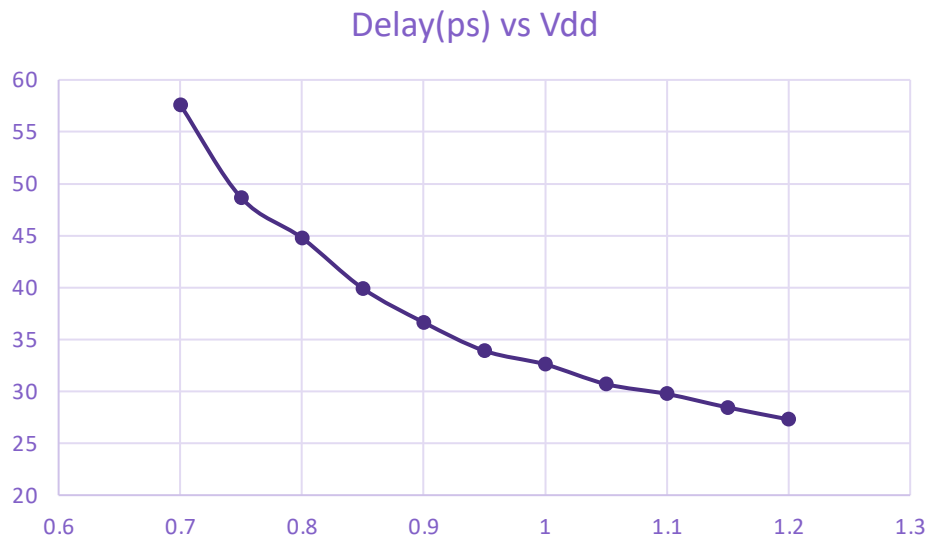
- ↑V_{dd} → ↓Delay
- $\tau = \frac{Q_{load}}{I} = \frac{C_{load}V_{dd}}{I} = \frac{kC_{load}V_{dd}}{\beta\,(V_{dd}-V_{th})^{\alpha}}$

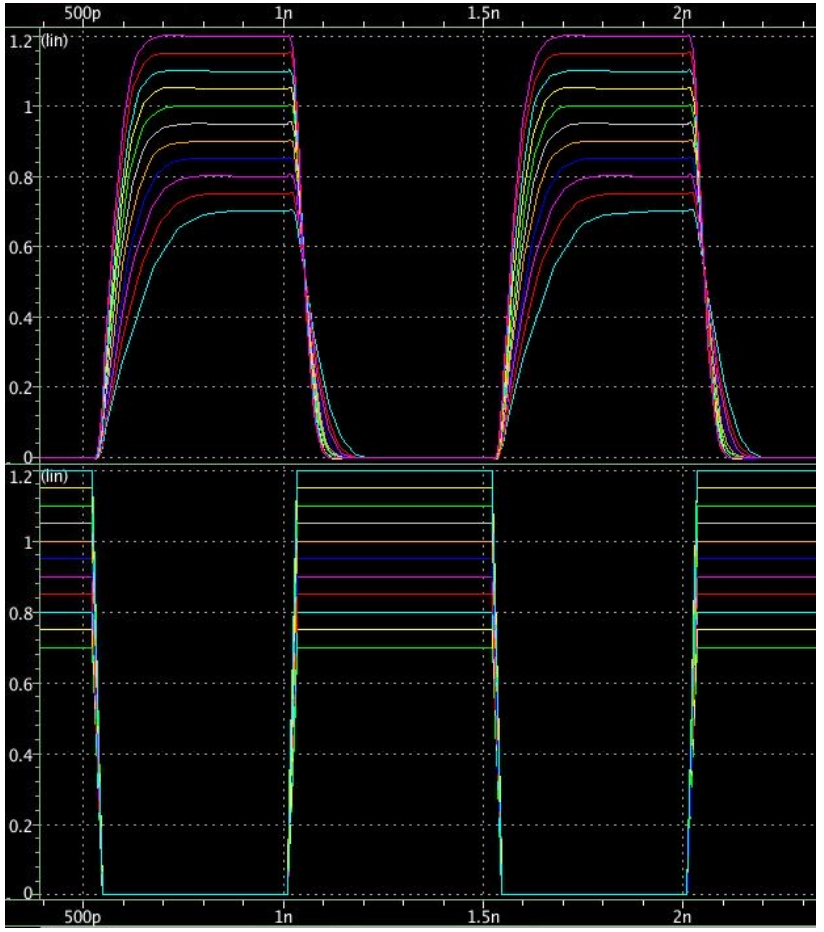
Delay Dependence on Vdd



- ↑V_{dd} → ↓Delay
- $\tau = \frac{Q_{load}}{I} = \frac{C_{load}V_{dd}}{I} = \frac{kC_{load}V_{dd}}{\beta\,(V_{dd}-V_{th})^{\alpha}}$

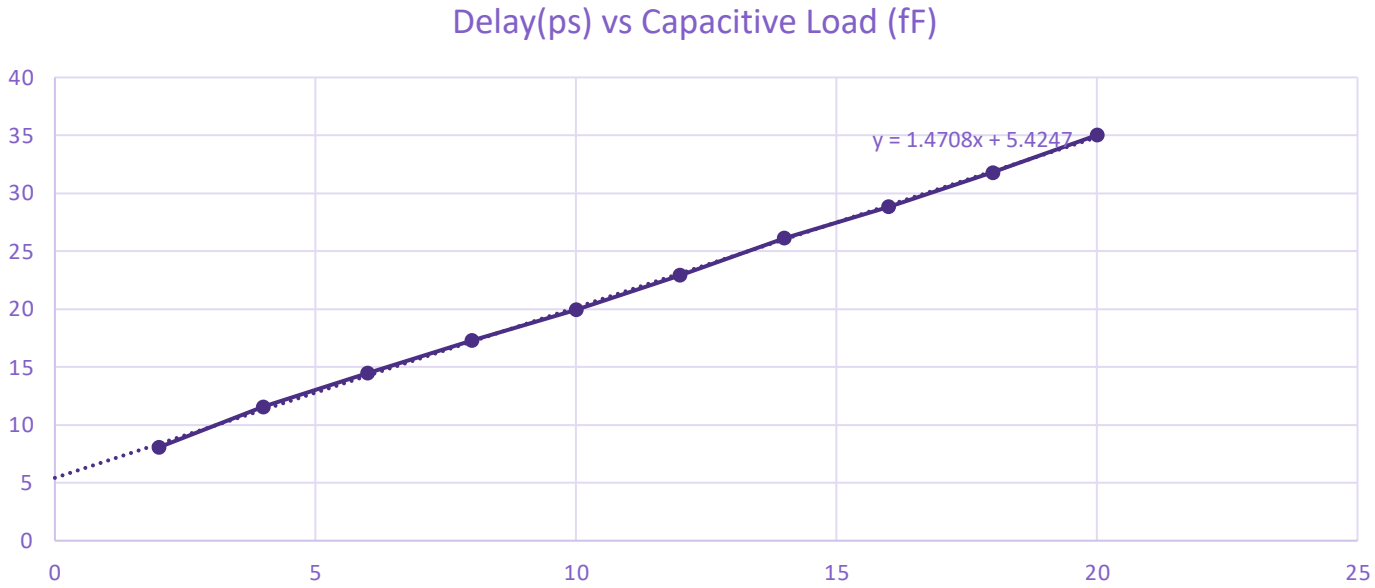
Delay Dependence on Vdd





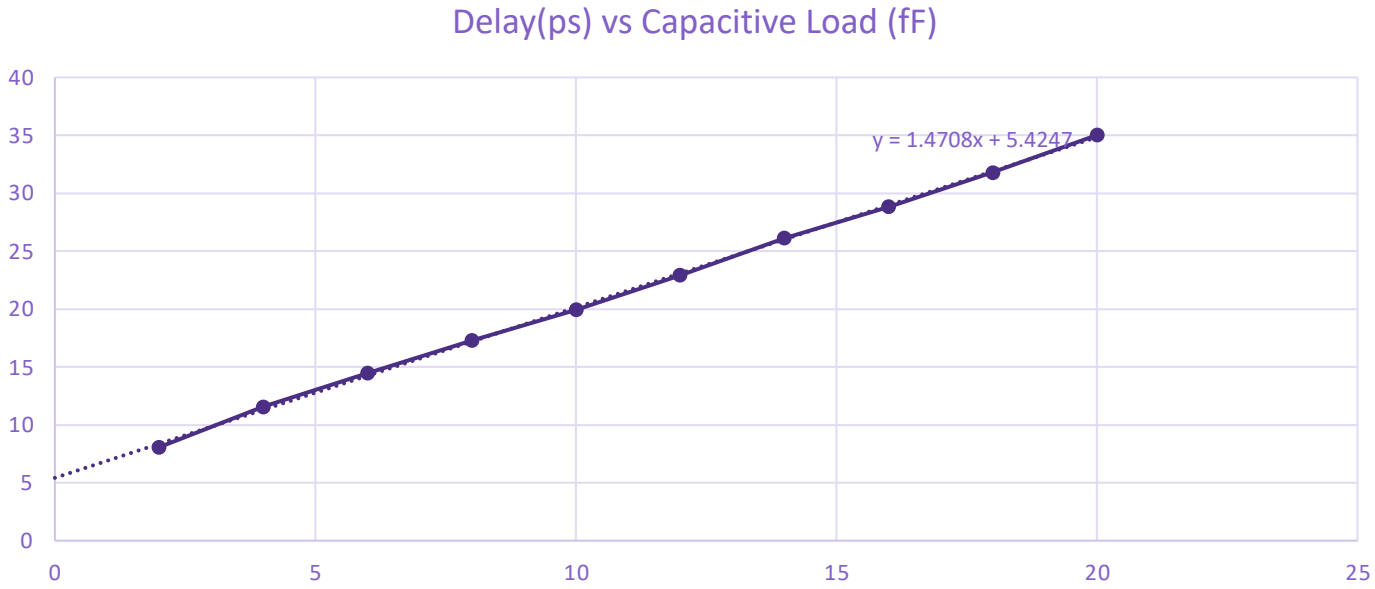
- $\uparrow V_{dd} \rightarrow \downarrow$Delay
- $\tau = \frac{Q_{load}}{I} = \frac{C_{load}V_{dd}}{I} = \frac{kC_{load}V_{dd}}{\beta\,(V_{dd}-V_{th})^{\alpha}}$

CMOS Inverter Delay : Inverter model



- Simplified views:
  - Effective resistance
  - Notice 2W allows twice the current (half the resistance)
  - 2L allows half the current
  - Limitations (small signal resistance is diff.)
- Can also think of equivalent current source?
  - Current source a function of V_{gs}, V_{th} W, L (No control over V_{th}*, μ, C_{ox})
  - Norton resistance for λ
  - Typically device operates in saturation while bringing output to 50%

CMOS Inverter Delay : Inverter model



Estimate R(V)
I_{eff} (V)

- Simplified views:
  - Effective resistance
  - Notice 2W allows twice the current (half the resistance)
  - 2L allows half the current
  - Limitations (small signal resistance is diff.)
- Can also think of equivalent current source?
  - Current source a function of V_{gs}, V_{th} W, L (No control over V_{th}*, μ, C_{ox})
  - Norton resistance for λ
  - Typically device operates in saturation while bringing output to 50%

Delay analysis : Some Thoughts

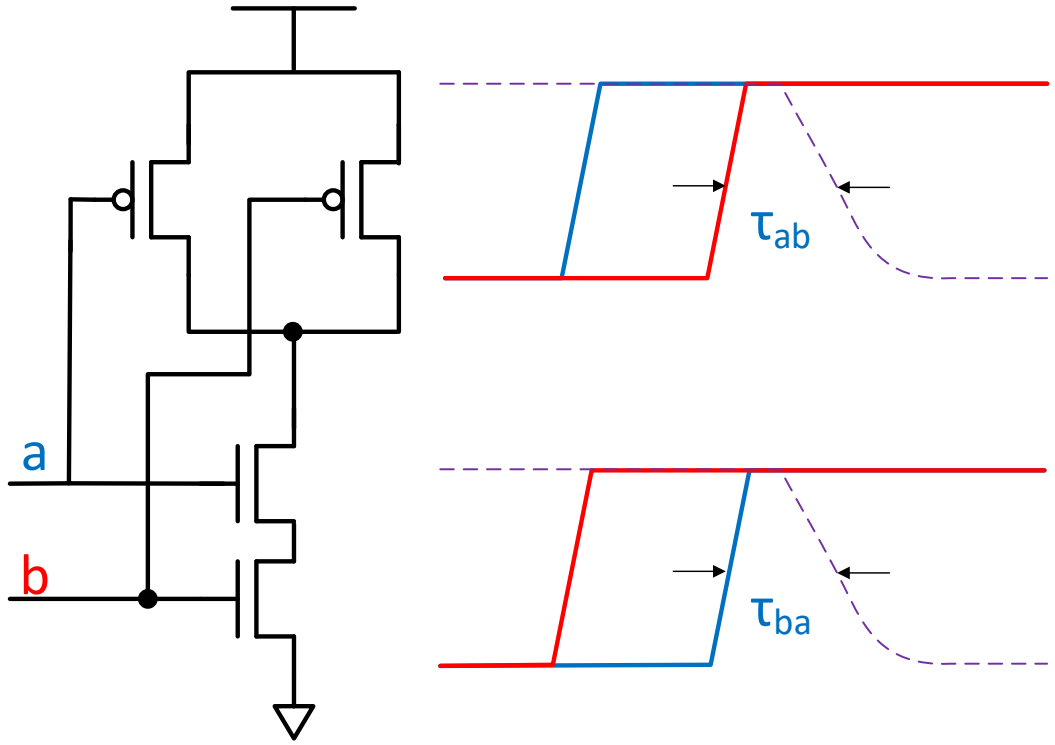
- Designers casually refer to the “resistance of the inverter”
  - If this resistance is a constant, what will be the impact of voltage on delay with the RC model?

- Scaled technologies rely on ↑ C_{ox} for improved current drive…
  - But what’s the point if that increases the load of the transistor?!

Delay analysis : Some Thoughts

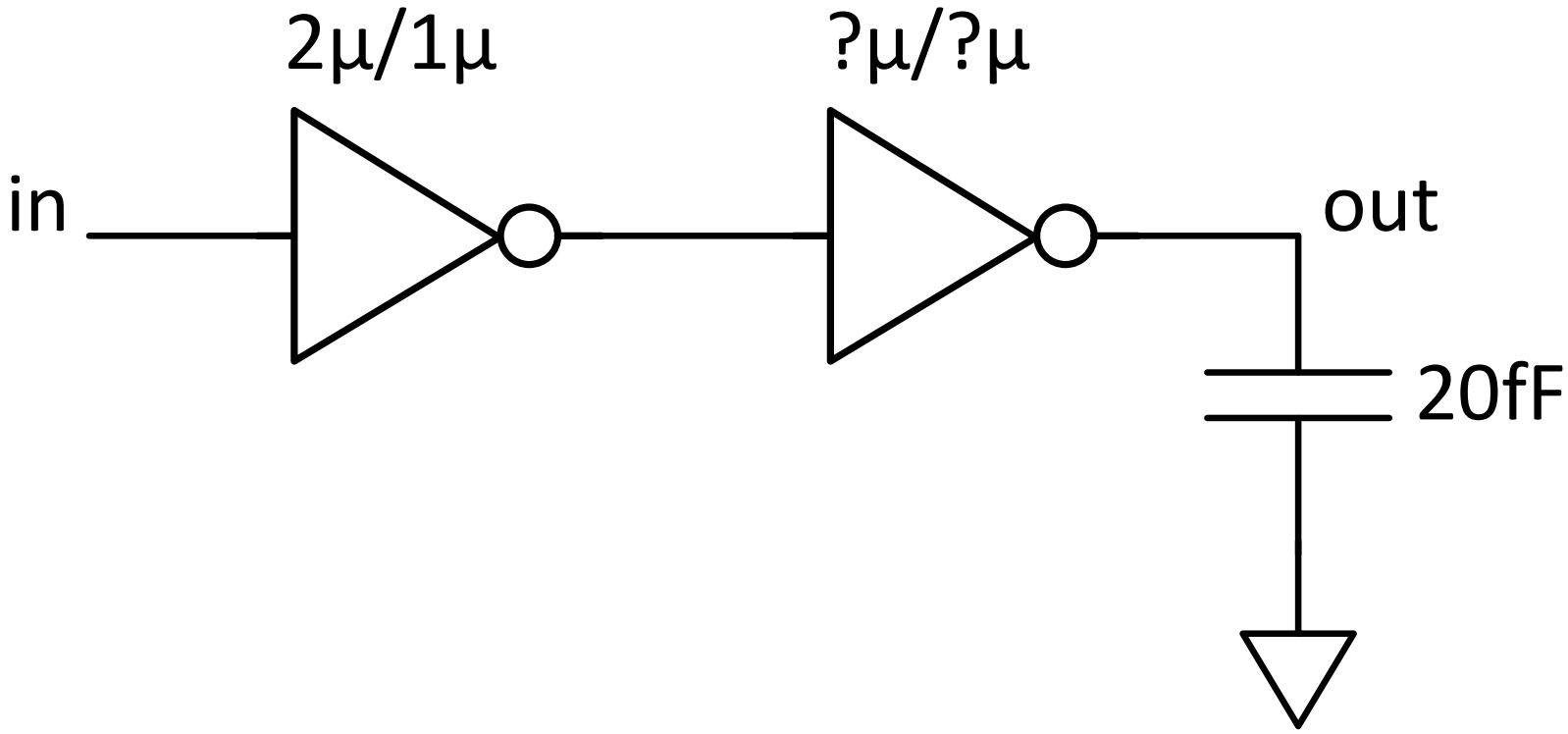
- Designers casually refer to the “resistance of the inverter”
  - If this resistance is a constant, what will be the impact of voltage on delay with the RC model?
  - Resistance is Voltage Dependent!
- Scaled technologies rely on ↑ C_{ox} for improved current drive…
  - But what’s the point if that increases the load of the transistor?!

CMOS Gate Delay : Input Ordering



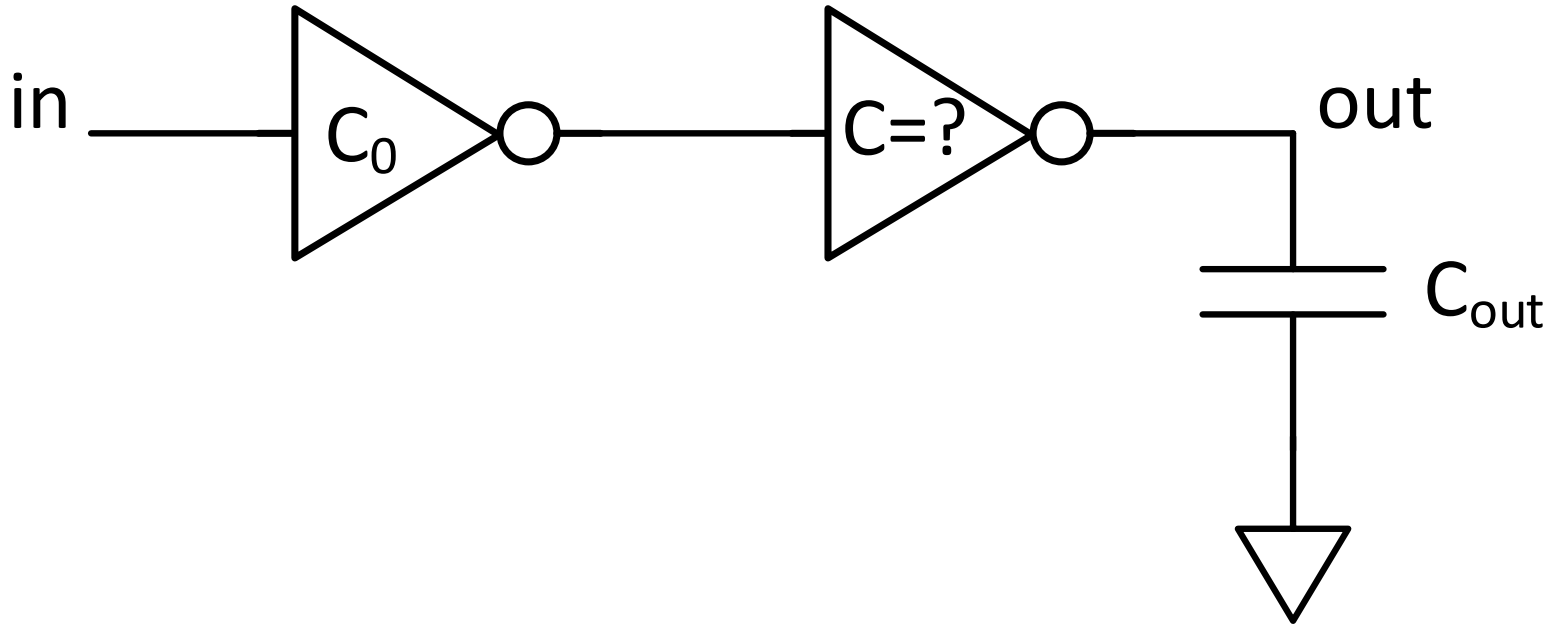
- Which is faster? τ_{ba} or τ_{ab} ??

Inverter Chain



- Total delay = $\sum \tau_i$
- How can we achieve minimum delay?

Inverter Chain



- In general, given a chain with:
  - Fixed input cap
  - Output load

Inverter Chain (contd)

- Assume balanced rise/fall delay
- $\tau = \sum \tau_i = \sum R_i C_{i+1}$
- $R = k/C$
- $\tau = R_0 C_1 + R_1 C_{out}$

$$= \frac{kC_1}{C_0} + \frac{kC_{out}}{C_1}$$

$$= k\left(\frac{C_1}{C_0} + \frac{C_{out}}{C_1}\right)$$

- Setting $\frac{\partial \tau}{\partial C_1} = 0$ gives …

Inverter Chain (contd)

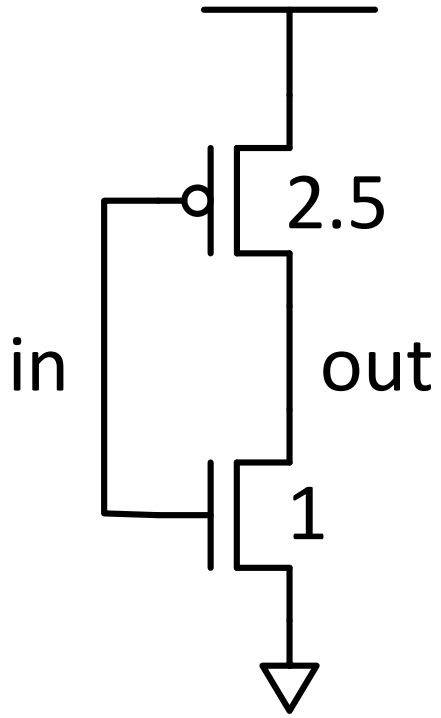
- Assume balanced rise/fall delay
- $\tau = \sum \tau_i = \sum R_i C_{i+1}$
- $R = k/C$
- $\tau = R_0 C_1 + R_1 C_{out}$

$$= \frac{kC_1}{C_0} + \frac{kC_{out}}{C_1}$$

$$= k\left(\frac{C_1}{C_0} + \frac{C_{out}}{C_1}\right)$$

- Setting $\frac{\partial \tau}{\partial C_1} = 0$ gives …
- $C_1 = k\sqrt{C_0 C_{out}}$
- $\tau_{min} = 2k\sqrt{C_{out}/C_0}$

Intrinsic RC delay



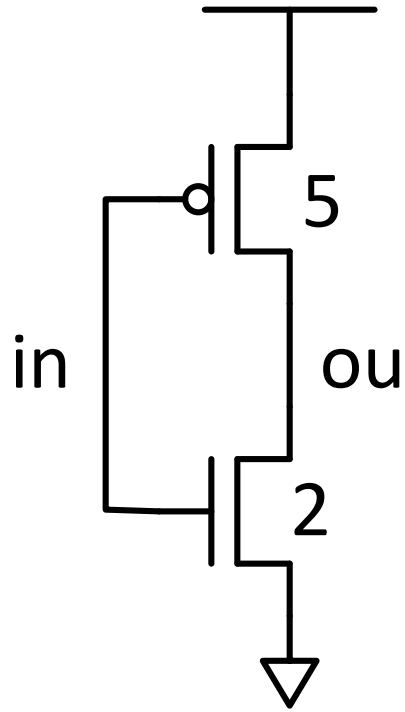
$R = k_r/W_n$

$C = k'_c(W_n + W_p)$
$= k'_c W_n(1+\beta)$
$= k_c W_n$

$R_n =$

$C_{in} =$

$R_n C_{in} =$



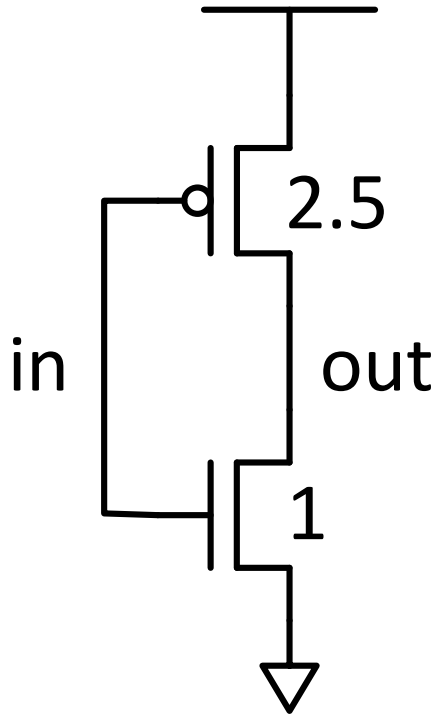
$R_n =$

$C_{in} =$

$R_n C_{in} =$

- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$R = k_r/W_n$

$C = k_c'(W_n + W_p)$

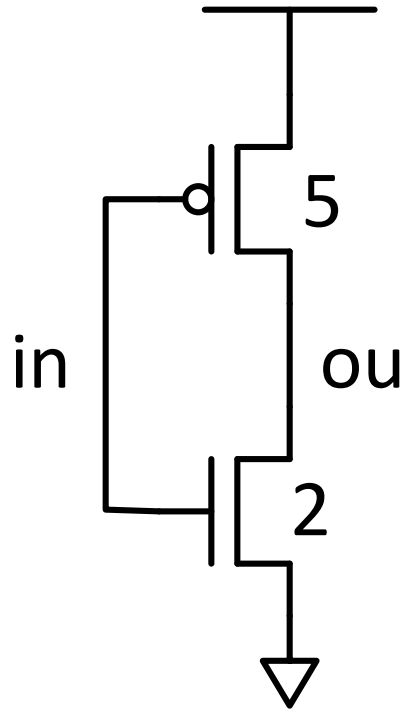
$= k_c'W_n(1+\beta)$

$= k_c W_n$

$R_n = \frac{k_r}{1.0}$

$C_{in} =$

$R_n C_{in} =$



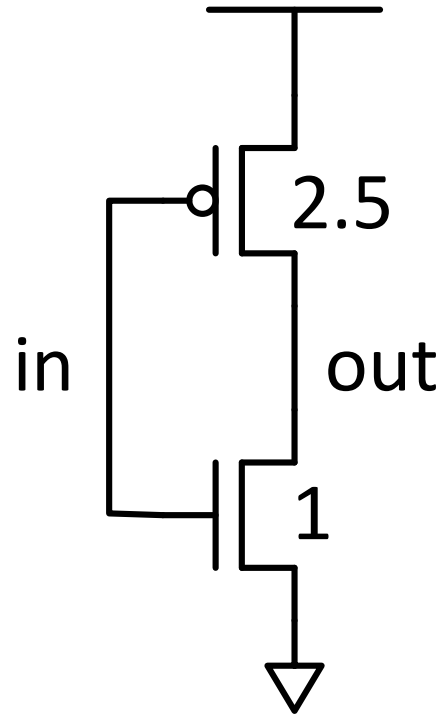
$R_n =$

$C_{in} =$

$R_n C_{in} =$

- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$R = k_r/W_n$

$C = k'_c(W_n + W_p)$

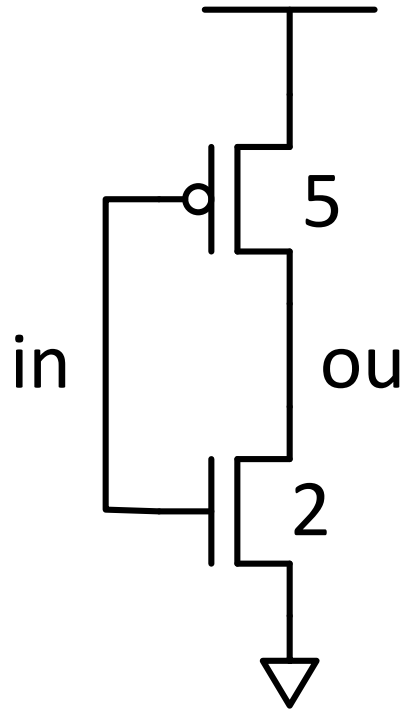
$= k'_c W_n(1 + \beta)$

$= k_c W_n$

$R_n = \frac{k_r}{1.0}$

$C_{in} = 3.5k'_c$

$R_n C_{in} =$



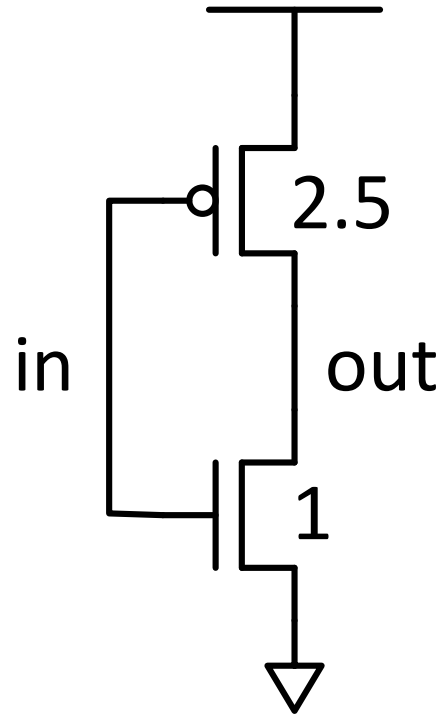
$R_n =$

$C_{in} =$

$R_n C_{in} =$

- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$R = k_r/W_n$

$C = k'_c(W_n + W_p)$

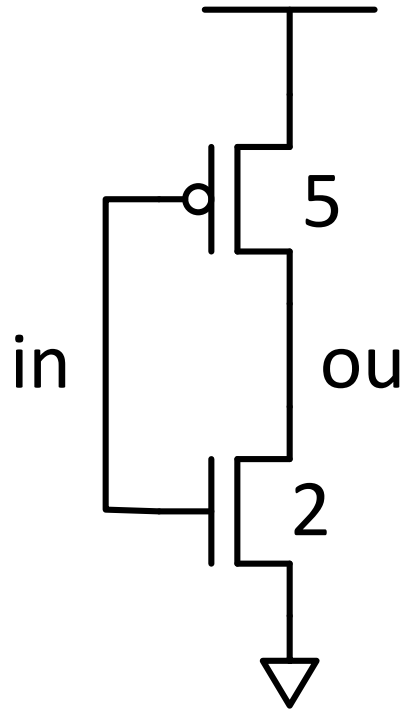
$= k'_c W_n(1 + \beta)$

$= k_c W_n$

$R_n = \frac{k_r}{1.0}$

$C_{in} = 3.5k'_c$

$R_n C_{in} = 3.5k_r k'_c$



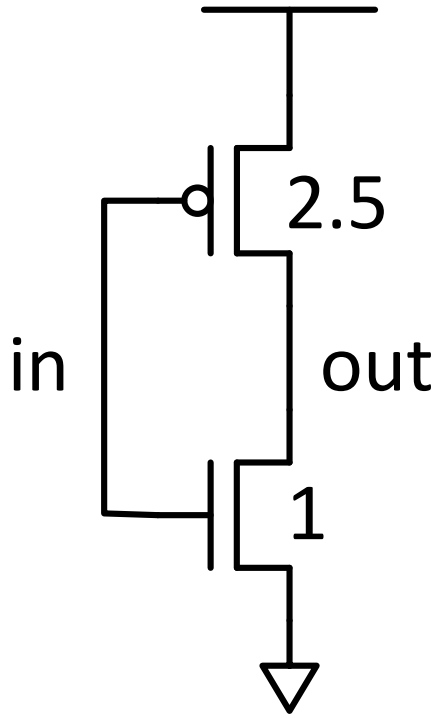
$R_n =$

$C_{in} =$

$R_n C_{in} =$

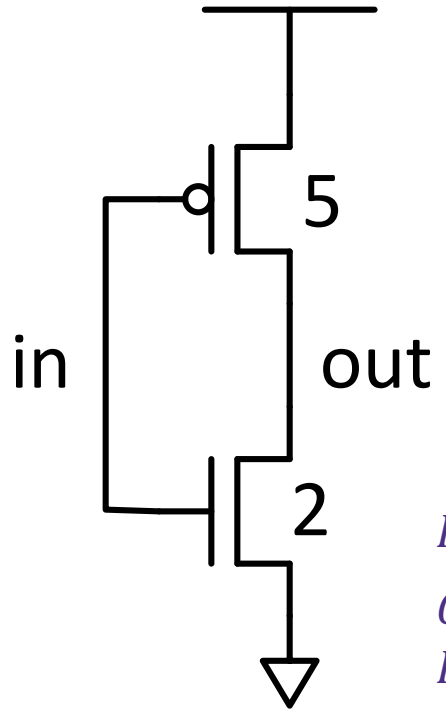
- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$$R = k_r/W_n$$
$$C = k'_c(W_n + W_p)$$
$$= k'_c W_n(1+\beta)$$
$$= k_c W_n$$

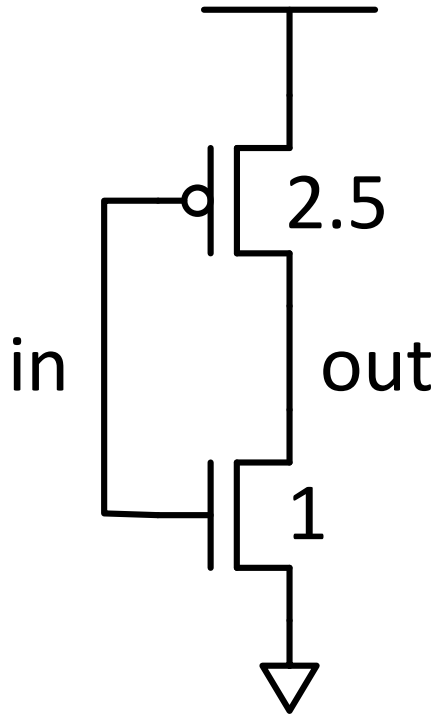
$$R_n = \frac{k_r}{1.0}$$
$$C_{in} = 3.5k'_c$$
$$R_n C_{in} = 3.5k_r k'_c$$



$$R_n = \frac{k_r}{2.0}$$
$$C_{in} =$$
$$R_n C_{in} =$$

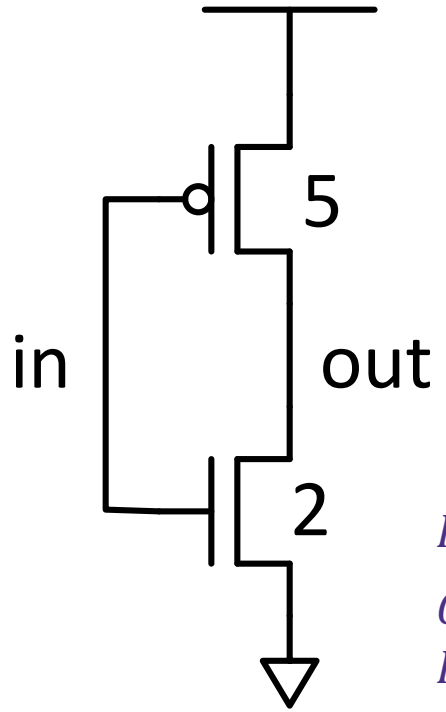
- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$$R = k_r/W_n$$
$$C = k'_c(W_n + W_p)$$
$$= k'_c W_n(1+\beta)$$
$$= k_c W_n$$

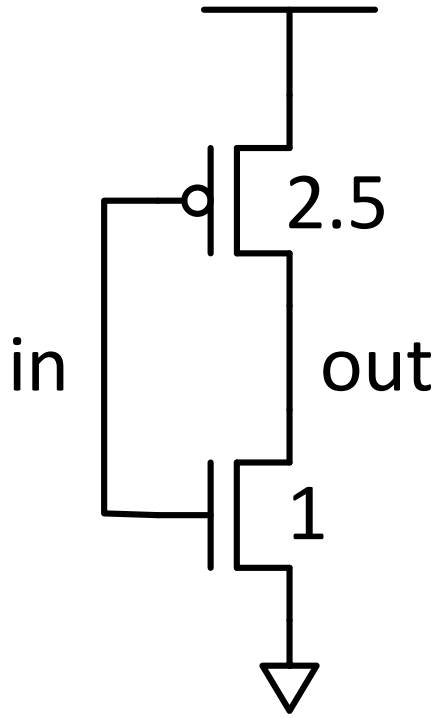
$$R_n = \frac{k_r}{1.0}$$
$$C_{in} = 3.5k'_c$$
$$R_n C_{in} = 3.5k_r k'_c$$



$$R_n = \frac{k_r}{2.0}$$
$$C_{in} = k'_c \times 7.0$$
$$R_n C_{in} =$$

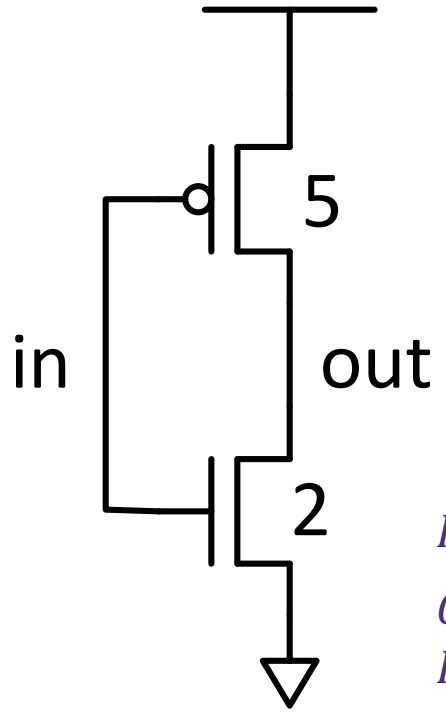
- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

Intrinsic RC delay



$$R = k_r/W_n$$
$$C = k'_c(W_n + W_p)$$
$$= k'_c W_n(1+\beta)$$
$$= k_c W_n$$

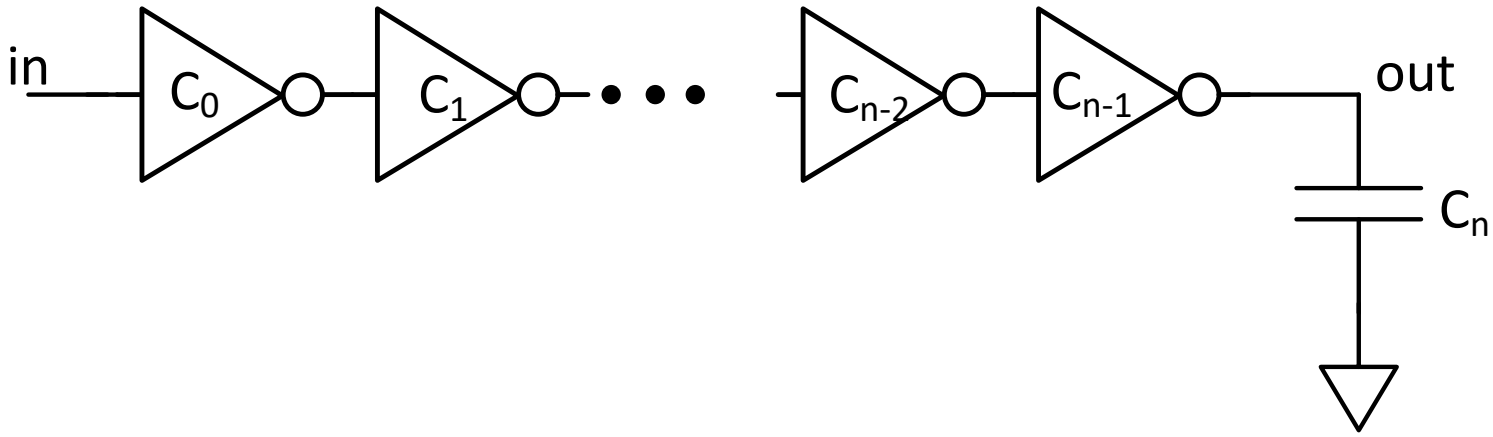
$$R_n = \frac{k_r}{1.0}$$
$$C_{in} = 3.5k'_c$$
$$R_n C_{in} = 3.5k_r k'_c$$



$$R_n = \frac{k_r}{2.0}$$
$$C_{in} = k'_c \times 7.0$$
$$R_n C_{in} = 3.5k_r k'_c$$

- Not to be confused with self loading
- $R_{eff}C_{in}$ product of an inverter (assume balanced inverter)
  - Independent of sizing
  - Topology driven
- Quick exercise: Compute R_nC_n for a 2 input NOR
  - Repeat R_pC_{in} calculation assuming β=3.5

A Slightly More General Problem

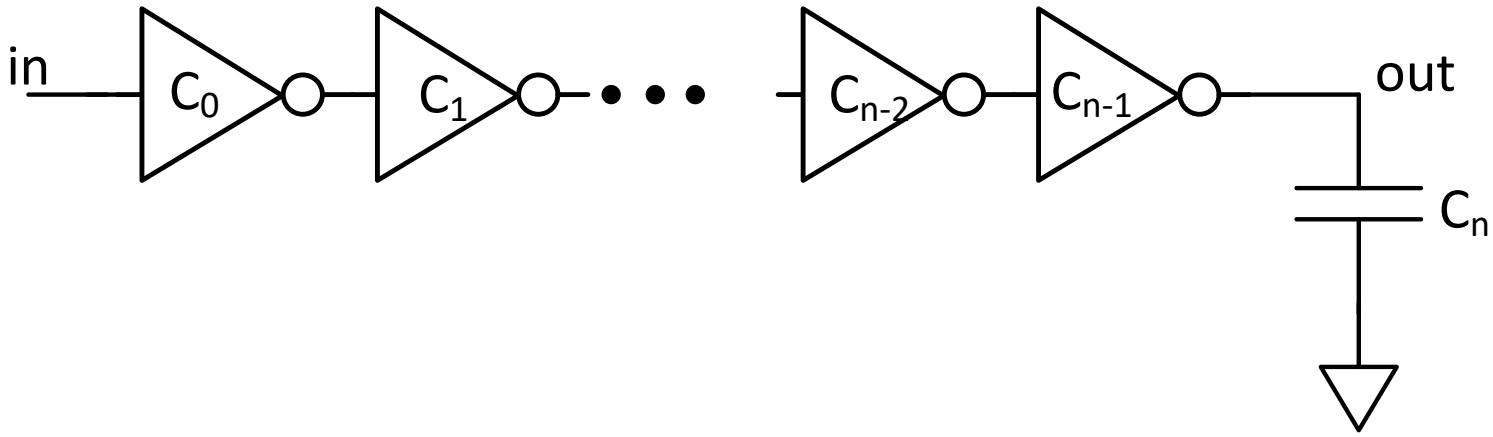


- Solve for optimal C_0, C_1 … C_{n-1} given C_n and C_0
- Recall $\tau = \sum \tau_i$

$$\tau = \sum R_i C_{i+1} = \sum \frac{kC_{i+1}}{C_i}$$

- Minimizing multivariate functions

A Slightly More General Problem

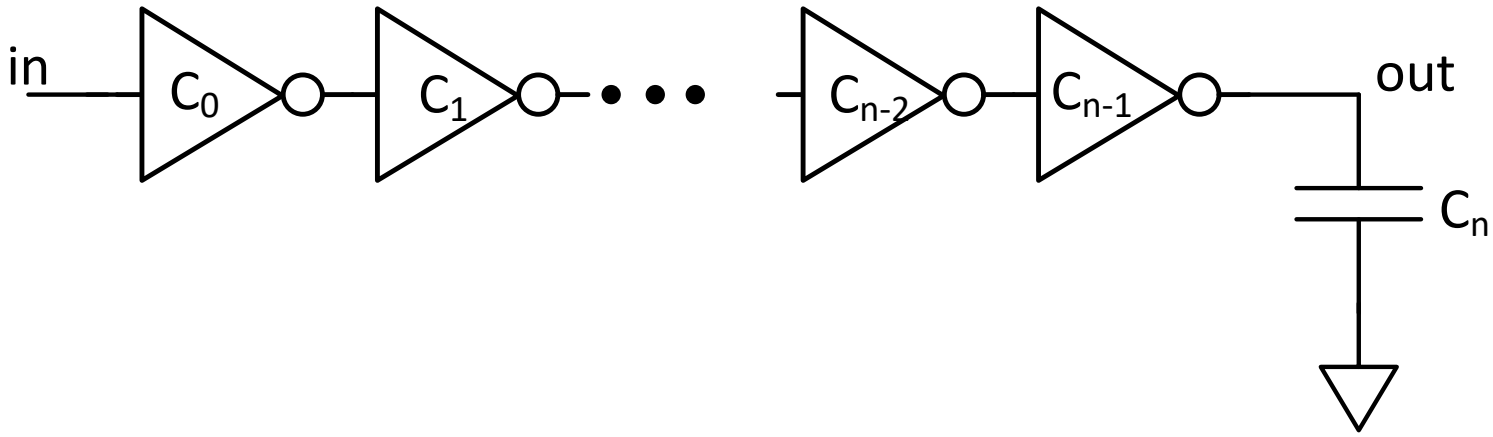


- Solve for optimal C_0, C_1 … C_{n-1} given C_n and C_0
- Recall $\tau = \sum \tau_i$

$$\tau = \sum R_i C_{i+1} = \sum \frac{k C_{i+1}}{C_i}$$

- Minimizing multivariate functions

A Slightly More General Problem



- Solve for optimal C_0, C_1 … C_{n-1} given C_n and C_0
- Recall $\tau = \sum \tau_i$

$$\tau = \sum R_i C_{i+1} = \sum \frac{kC_{i+1}}{C_i}$$

- Minimizing multivariate functions …$\nabla \tau = 0$

Minimization of τ

- Solving gives $\frac{C_1}{C_0} = \frac{C_2}{C_1} = \cdots \frac{C_{n-1}}{C_{n-2}} = \frac{C_n}{C_{n-1}} = h$
- However, $\frac{C_n}{C_0} = constant = \prod \frac{C_{i+1}}{C_i} = \prod h_i$ (Telescoping…)

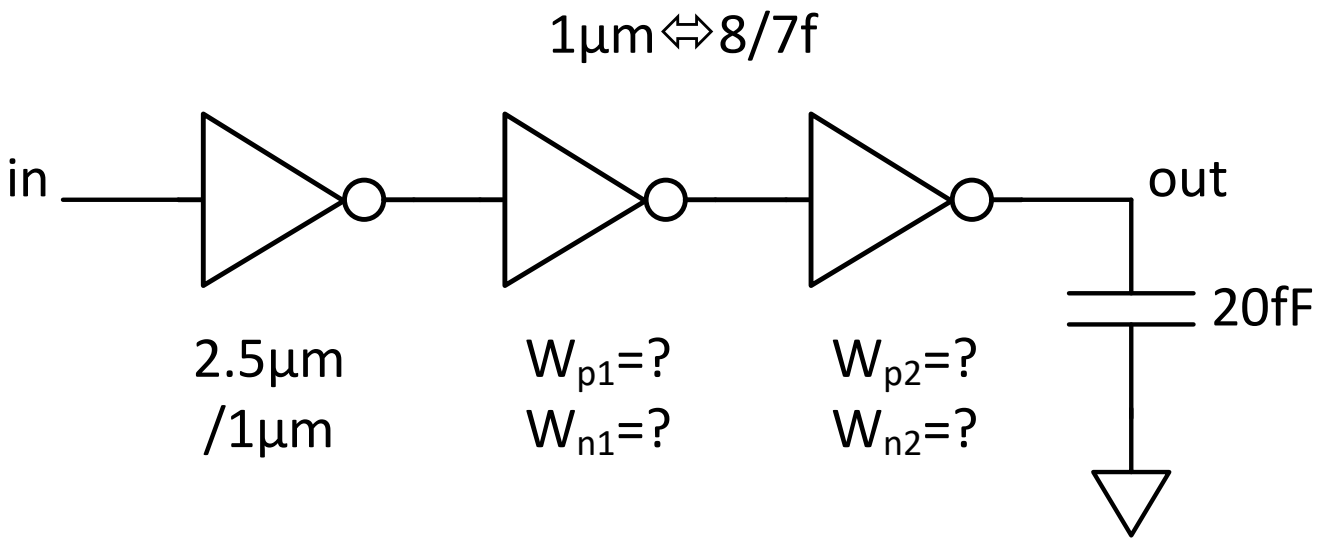
- Important assumptions
  - All of the input capacitance of the inverter scales with W
    - Wire cap ignored
    - Poly-contact cap ignored
  - Mosfet serves as the only source of resistance

- Question: How does self-loading impact the optimal solution?

Minimization of τ

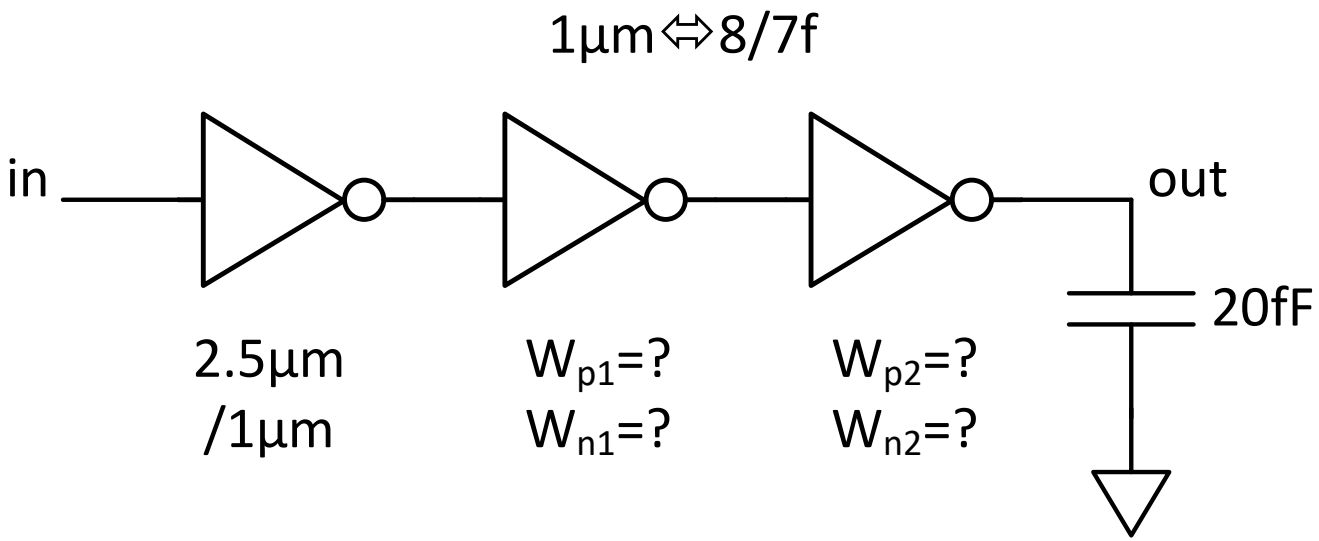
- Solving gives $\frac{C_1}{C_0} = \frac{C_2}{C_1} = \cdots \frac{C_{n-1}}{C_{n-2}} = \frac{C_n}{C_{n-1}} = h$
- However, $\frac{C_n}{C_0} = constant = \prod \frac{C_{i+1}}{C_i} = \prod h_i$ (Telescoping…)
- $\frac{C_n}{C_0} = h^n \Rightarrow h = \sqrt[n]{\frac{C_n}{C_0}}$
- Important assumptions
  - All of the input capacitance of the inverter scales with W
    - Wire cap ignored
    - Poly-contact cap ignored
  - Mosfet serves as the only source of resistance
- Question: How does self-loading impact the optimal solution?

Sizing Example



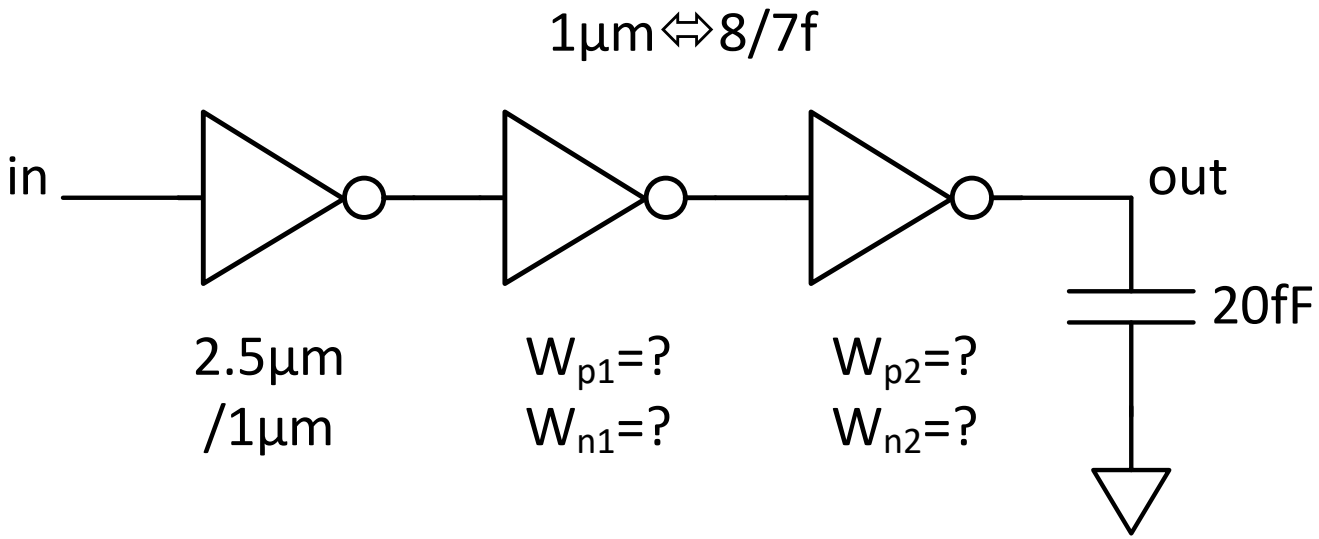
- Note: C_{in} and C_{out} units are not consistent

Sizing Example



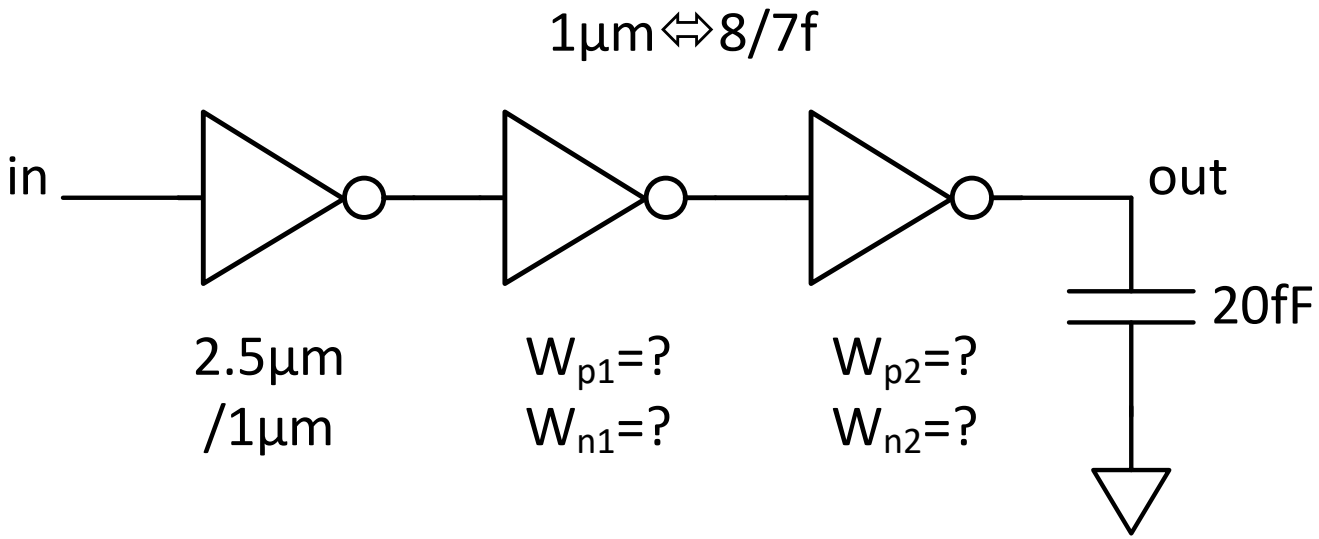
- Note: C_{in} and C_{out} units are not consistent
- C_{in} = 3.5*8/7 = 4f

Sizing Example



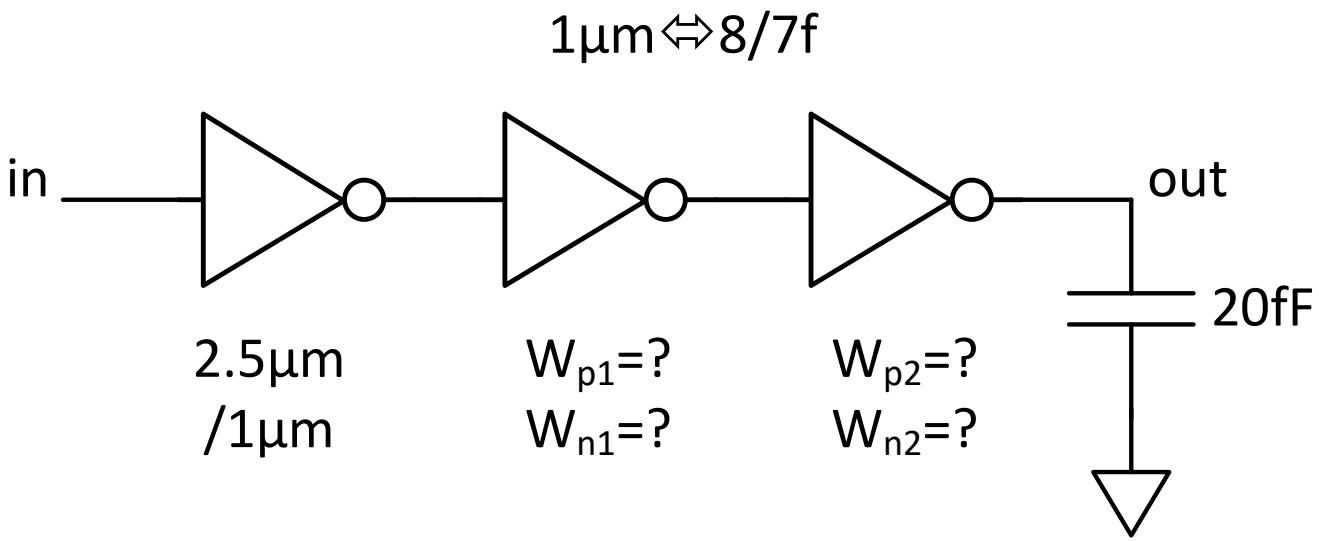
- Note: C_{in} and C_{out} units are not consistent
- C_{in} = 3.5*8/7 = 4f
- $h^3 = 20/4 \Rightarrow h = \sqrt[3]{5} = 1.71$

Sizing Example



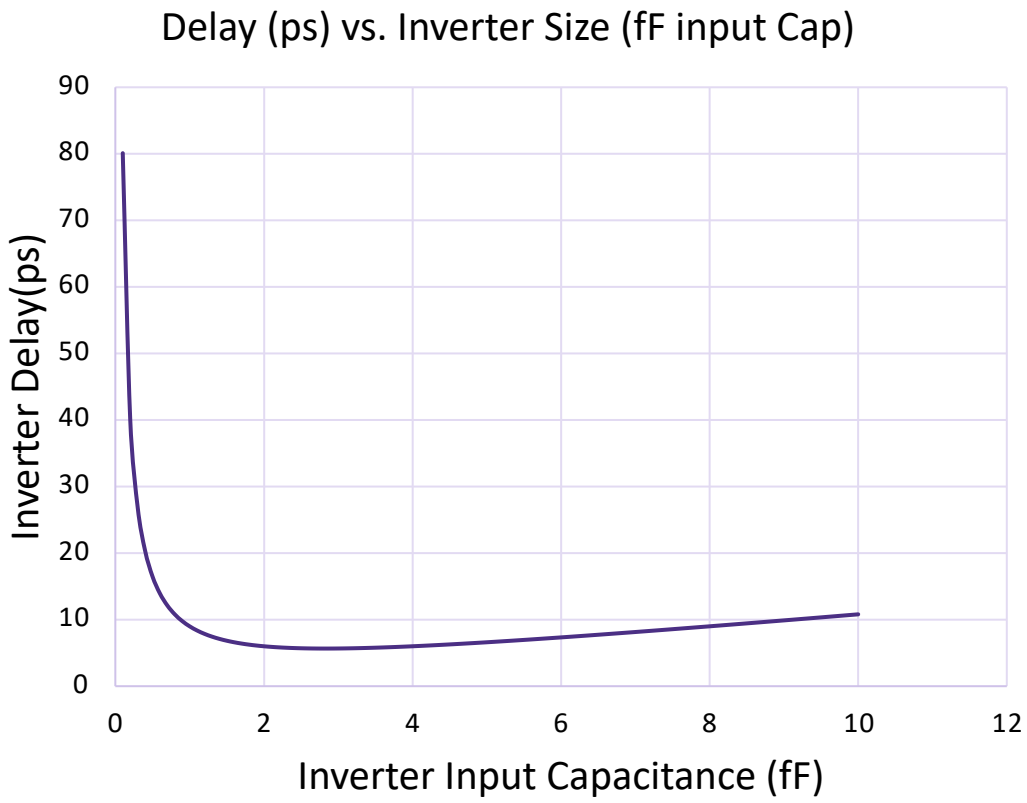
- Note: C_{in} and C_{out} units are not consistent
- C_{in} = 3.5*8/7 = 4f
- $h^3 = 20/4 \Rightarrow h = \sqrt[3]{5} = 1.71$
- C_1 = 4f * 1.71 = 6.84 ⇒ W_{p1} = 4.25 µm, W_{n1} = 1.71µm

Sizing Example



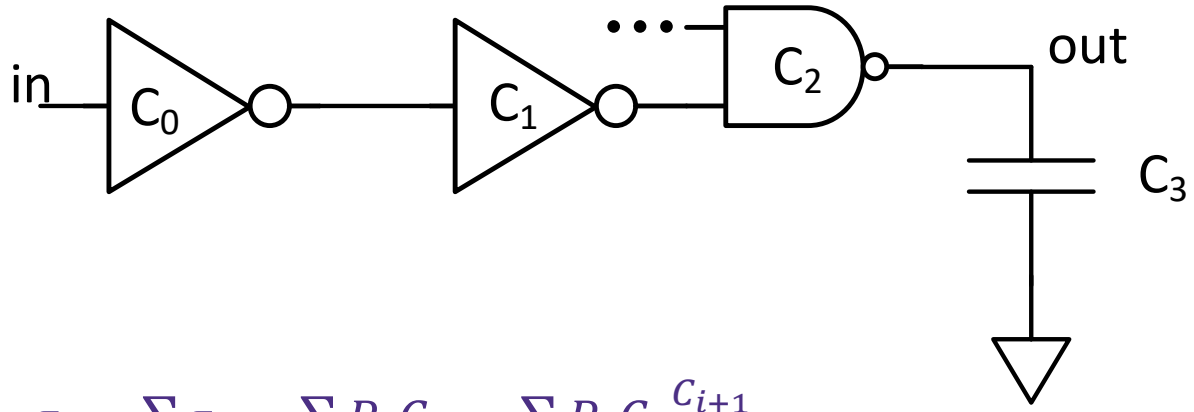
- Note: C_{in} and C_{out} units are not consistent
- C_{in} = 3.5*8/7 = 4f
- $h^3 = 20/4 \Rightarrow h = \sqrt[3]{5} = 1.71$
- C_1 = 4f * 1.71 = 6.84 ⇒ W_{p1} = 4.25 µm, W_{n1} = 1.71µm
- $C_2 = C_1$ * 1.71 = 11.69 ⇒ W_{p1} = 7.30 µm, W_{n1} = 2.92µm

Practical Sizing Considerations



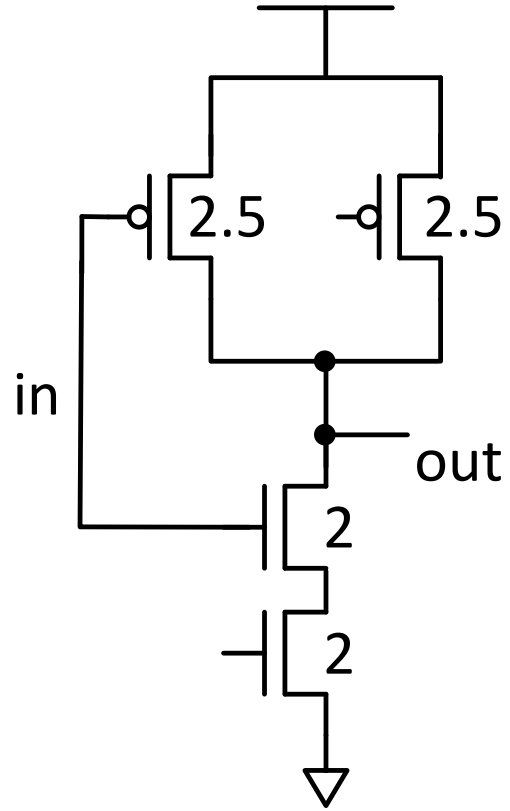
- Better to err on the side of larger drivers
- Min width issues (Allowed/Recommended for technology)
- Technique is great as an **initial starting point** for design

Quick Note on General Sizing (Extra…)

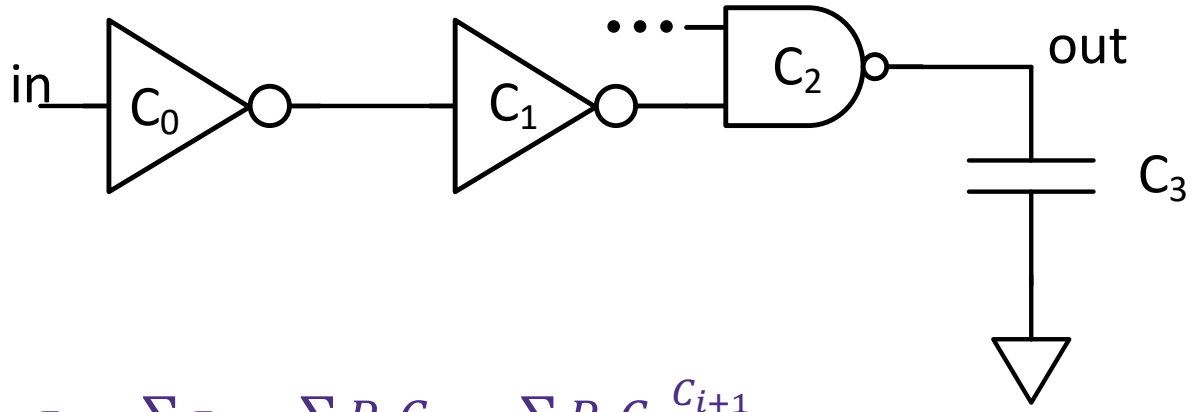


- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$



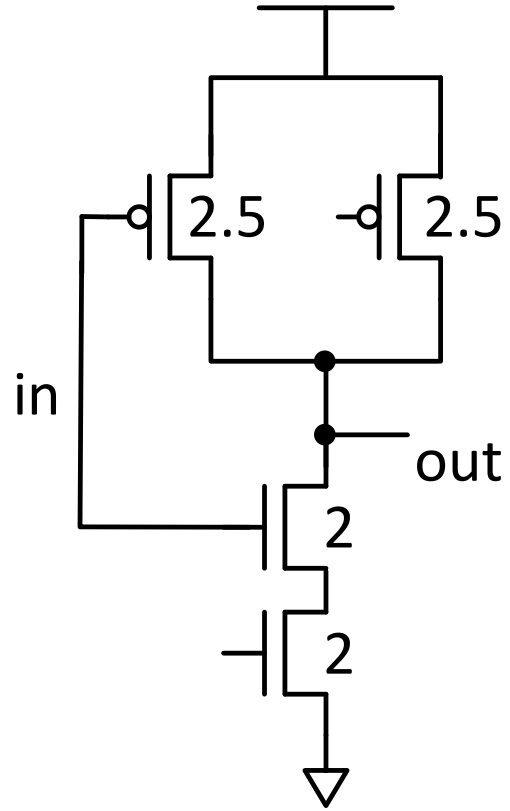
Quick Note on General Sizing (Extra…)



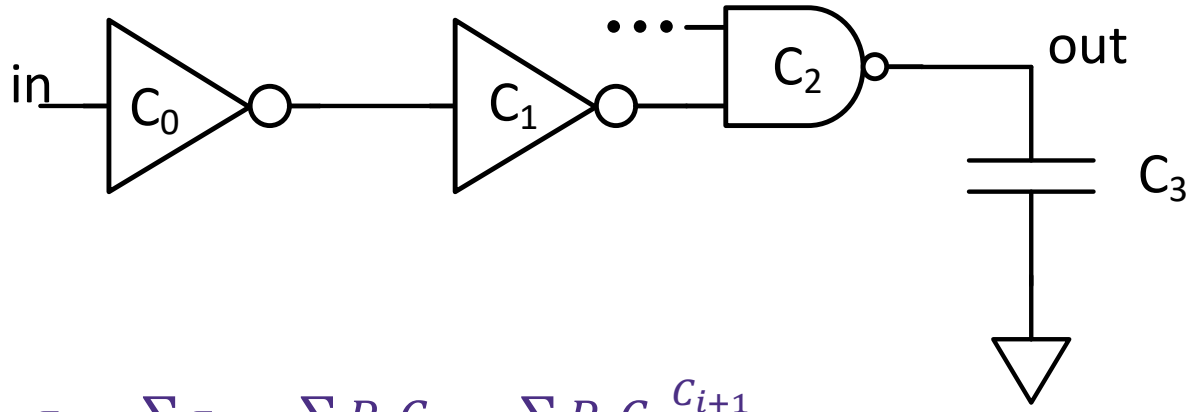
- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$

- $RC_{intrinsic,inv} = 3.5 k_r k_c$



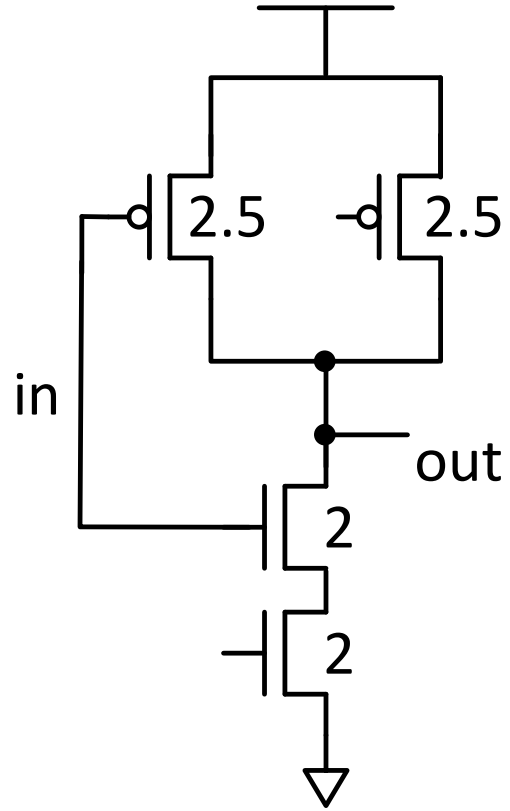
Quick Note on General Sizing (Extra…)



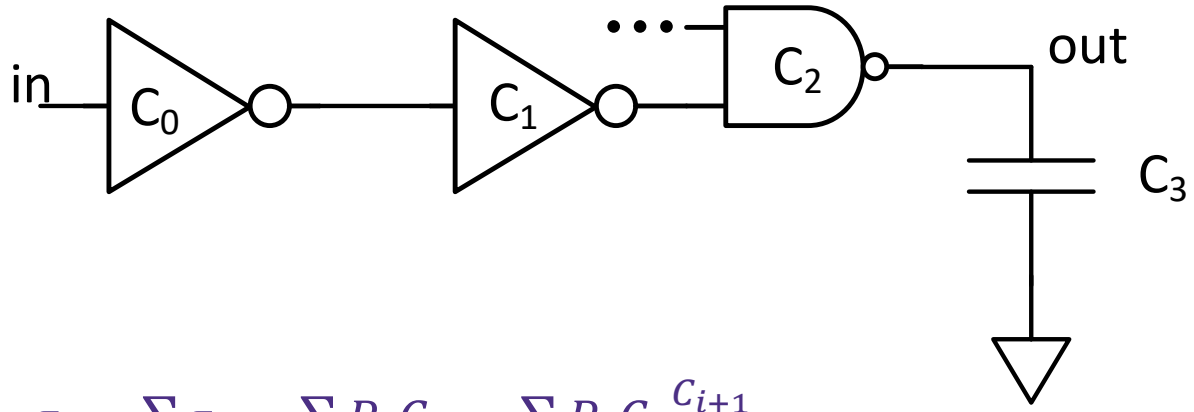
- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$

- $RC_{intrinsic,inv} = 3.5 k_r k_c$
- $\tau_{intrinsic,nand} = 4.5 k_r k_c$



Quick Note on General Sizing (Extra…)

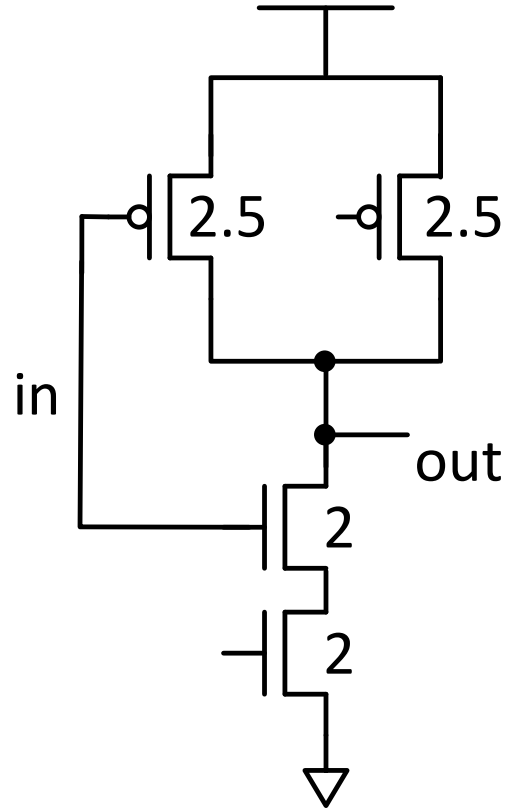


- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

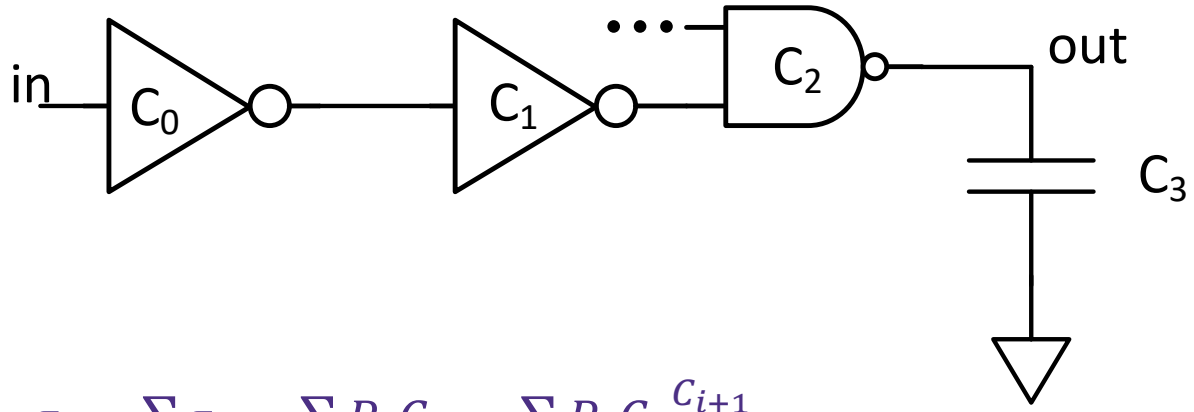
$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$

- $RC_{intrinsic,inv} = 3.5 k_r k_c$
- $\tau_{intrinsic,nand} = 4.5 k_r k_c$
- $\tau = 3.5 k_r k_c \frac{C_1}{C_0} + 3.5 k_r k_c \frac{C_2}{C_1} + 4.5 k_r k_c \frac{C_3}{C_2}$

  $\tau_{inv\_units} = \frac{C_1}{C_0} + \frac{C_2}{C_1} + \frac{4.5}{3.5}\frac{C_3}{C_2}$
- Rule-of-thumb: Maintain a stage-effort of ≈ 3.4



Quick Note on General Sizing (Extra…)

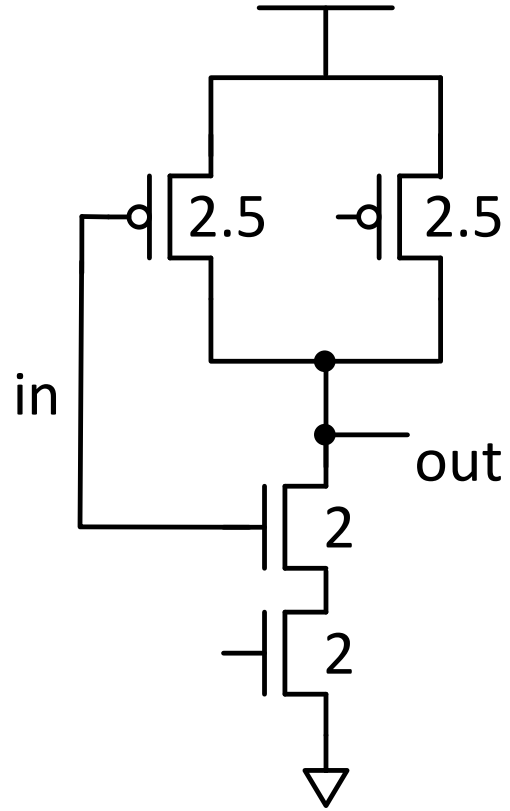


- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

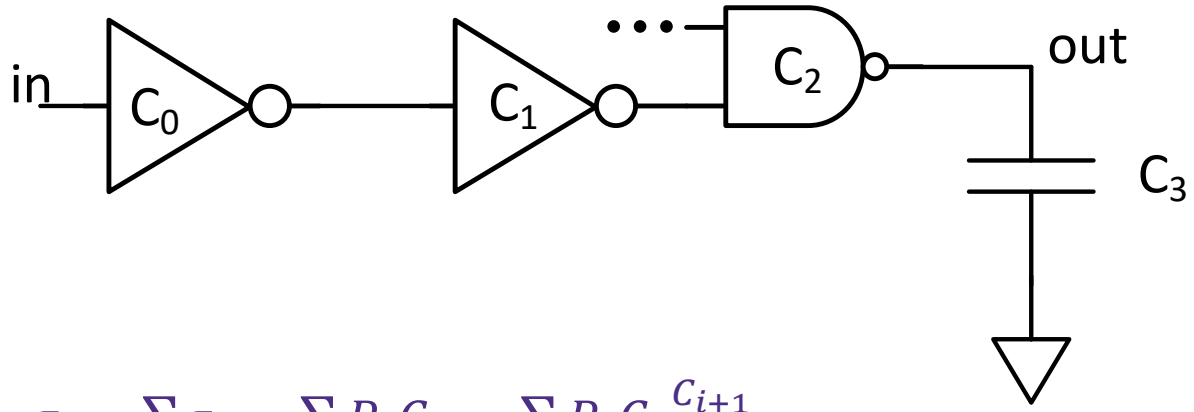
$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$

- $RC_{intrinsic,inv} = 3.5 k_r k_c$
- $\tau_{intrinsic,nand} = 4.5 k_r k_c$
- $\tau = 3.5 k_r k_c \frac{C_1}{C_0} + 3.5 k_r k_c \frac{C_2}{C_1} + 4.5 k_r k_c \frac{C_3}{C_2}$

  $\tau_{inv\_units} = \frac{C_1}{C_0} + \frac{C_2}{C_1} + \frac{4.5}{3.5}\frac{C_3}{C_2}$ ← Logical Effort
- Rule-of-thumb: Maintain a stage-effort of ≈ 3.4



Quick Note on General Sizing (Extra…)

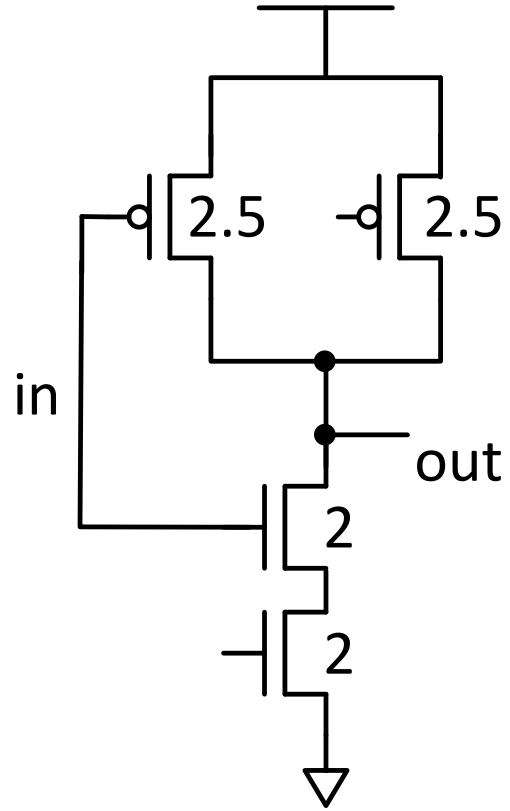


- $\tau = \sum \tau_i = \sum R_i C_{i+1} = \sum R_i C_i \frac{C_{i+1}}{C_i}$

$$= \sum \tau_{intrinsic} \frac{C_{i+1}}{C_i}$$

- $RC_{intrinsic,inv} = 3.5 k_r k_c$
- $\tau_{intrinsic,nand} = 4.5 k_r k_c$
- $\tau = 3.5 k_r k_c \frac{C_1}{C_0} + 3.5 k_r k_c \frac{C_2}{C_1} + 4.5 k_r k_c \frac{C_3}{C_2}$

  $\tau_{inv\_units} = \frac{C_1}{C_0} + \frac{C_2}{C_1} + \boxed{\frac{4.5}{3.5}\frac{C_3}{C_2}}$ ← Stage Effort
- Rule-of-thumb: Maintain a stage-effort of ≈ 3.4



STATIC TIMING ANALYSIS