CSE460: VLSI Design

Lecture 17 + 18

Delay

Contents

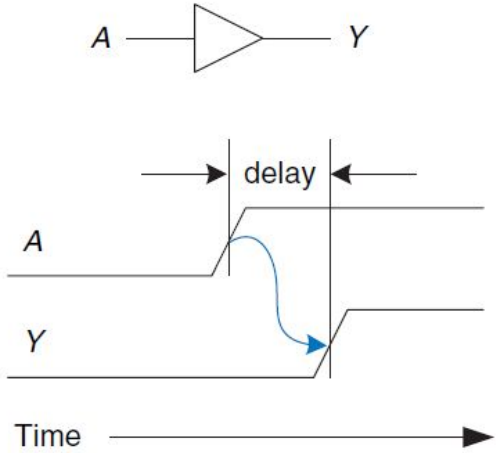
Introduction

In logic circuits, an output takes time to change in response to an input change, called ***delay***

For example, the figure shows the delay between an input change and the subsequent output change for a buffer. This figure is called a timing diagram

A timing diagram portrays the *transient response* of a logic circuit when inputs change

(The blue arrow indicates the input/s which cause/s the output to change. In this case, A causes Y to change.)



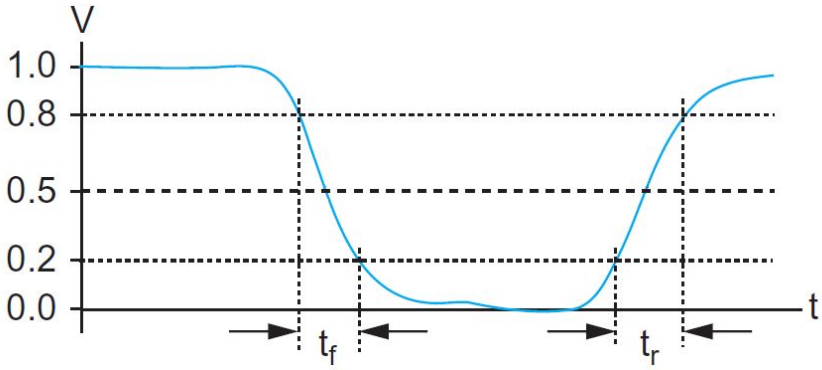
Rise and Fall Time

t_r: rise time

time for a waveform to rise from 20% to 80% of its steady-state value

t_f: fall time

time for a waveform to fall from 80% to 20% of its steady-state value



Propagation and Contamination Delay

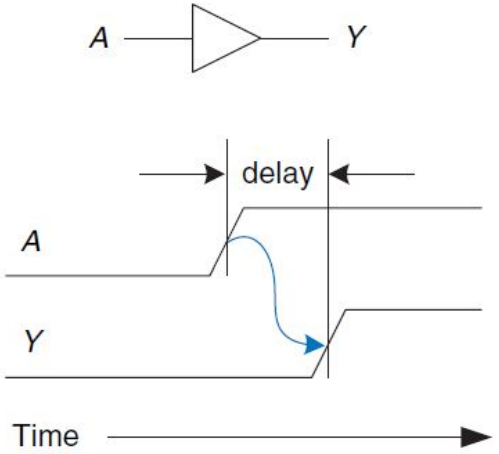
Combinational logic is characterized by its *propagation delay* and *contamination delay*

t_{pd}: propagation delay (worst-case delay)

the *maximum* time from when an input changes until the output or outputs reach their final value

t_{cd}: contamination delay (best-case delay)

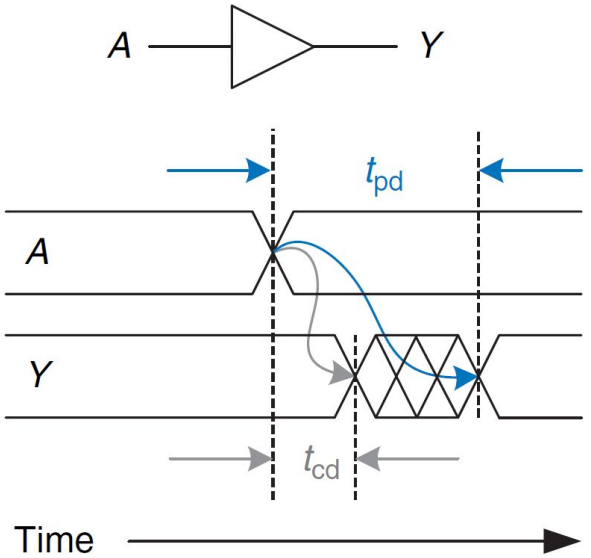
the *minimum* time from when an input changes until any output starts to change its value



Propagation and Contamination Delay

The figure illustrates a buffer's propagation delay and contamination delay

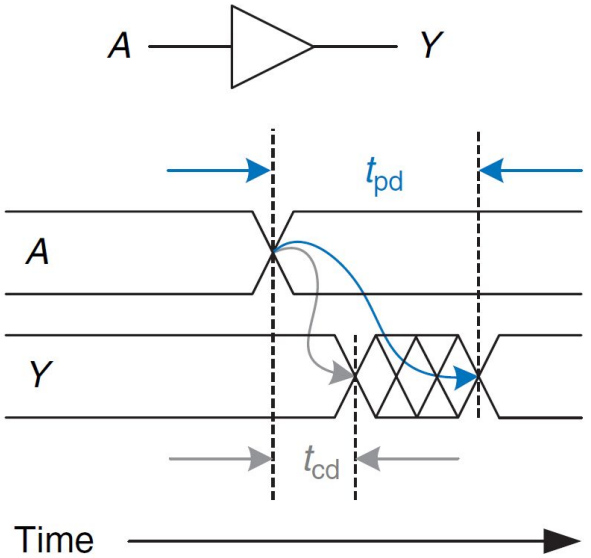
- **A** is initially either HIGH or LOW and changes to the other state at a particular time
  - we are interested only in the fact that it changes, not what value it has
- In response, **Y** changes some time later
- The arcs indicate that **Y** may start to change t_{cd} after input **A** transitions
- But **Y** definitely settles to its new value within t_{pd}



Propagation and Contamination Delay

Intuitively, we know that when an input changes, the output will:

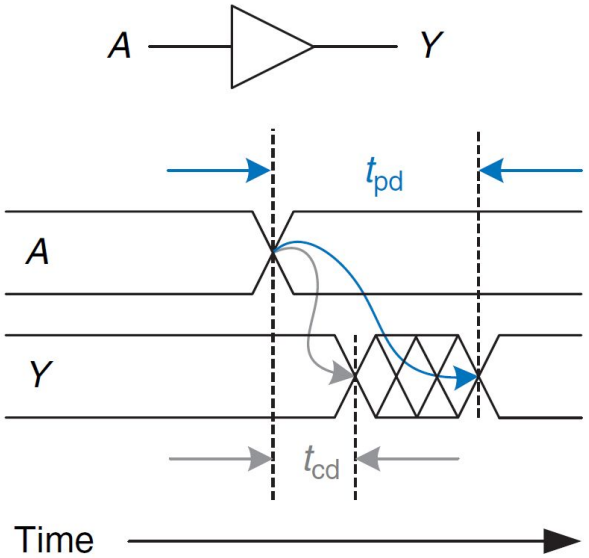
- retain its old value for at least the ***contamination delay***
- take on its new value in at most the ***propagation delay***



Propagation and Contamination Delay

The underlying causes of delay in circuits include the time required to charge the capacitance in a circuit. t_{pd} and t_{cd} may be different for many reasons:

- different rising and falling delays
- multiple inputs and outputs, some of which are faster than others
- circuits slowing down when hot and speeding up when cold



Delay Definitions

t_{pdr}: rising propagation delay

From input to rising output, crossing $V_{DD}/2$

t_{pdf}: falling propagation delay

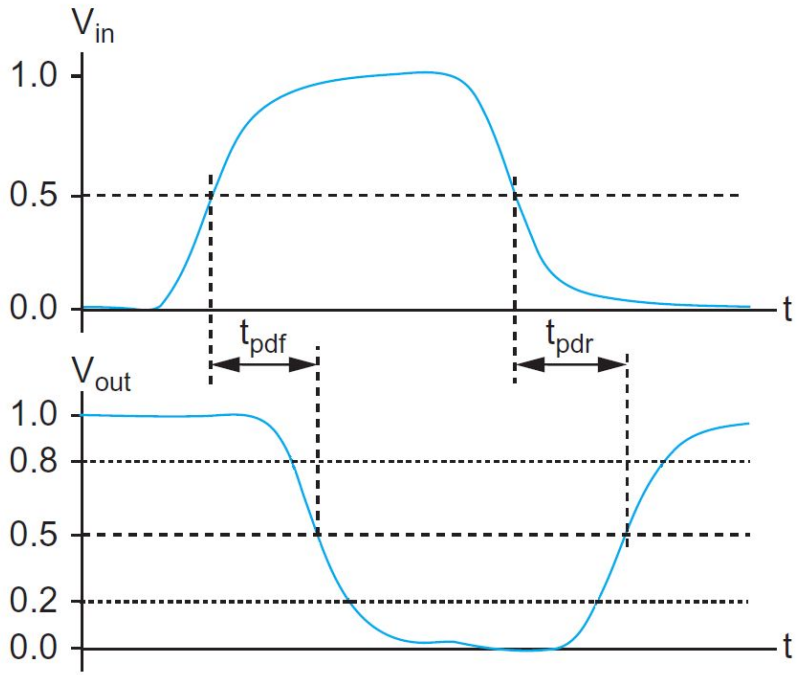
From input to falling output, crossing $V_{DD}/2$

t_{cdr}: rising contamination delay

From input to rising output, crossing $V_{DD}/2$

t_{cdf}: falling contamination delay

From input to falling output, crossing $V_{DD}/2$



Inverter Delay Simulation

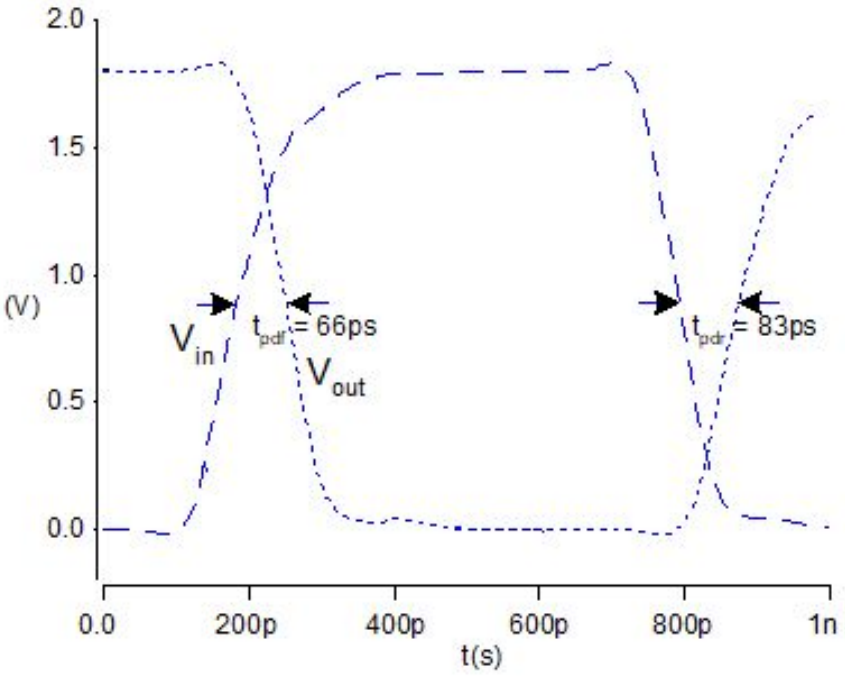
t_{pdr}: rising propagation delay

From input to rising output, crossing $V_{DD}/2$

t_{pdf}: falling propagation delay

From input to falling output, crossing $V_{DD}/2$

- Inverter Delay Simulation
- By solving differential equations numerically
- We would like to “estimate” this delay



Computing Delay

The most fundamental way to compute delay is to

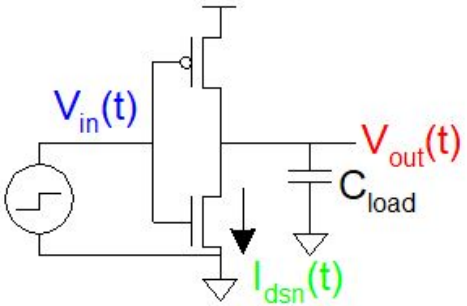
1. develop a physical model of the circuit of interest
2. write a differential equation describing the output voltage as a function of input voltage and time
3. solve the equation
4. ***delay*** is the time when the output reaches $V_{DD}/2$

The solution of the differential equation is called the ***transient response***

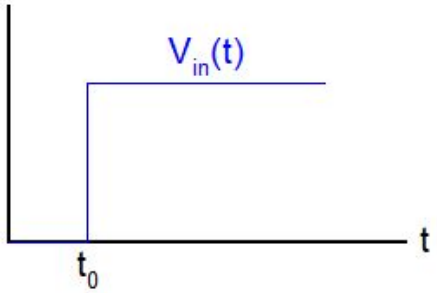
Inverter Transient Response

Example: Transient response of a CMOS inverter

- 1. develop a physical model of the circuit of interest



Circuit

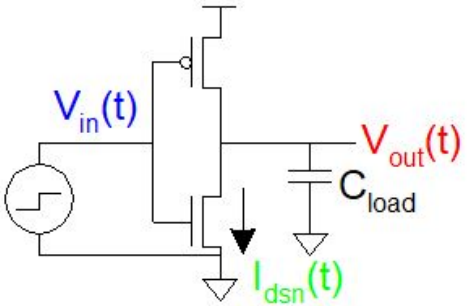


Input voltage

Inverter Transient Response

Example: Transient response of a CMOS inverter

- 2. write a differential equation describing the output voltage as a function of input voltage and time



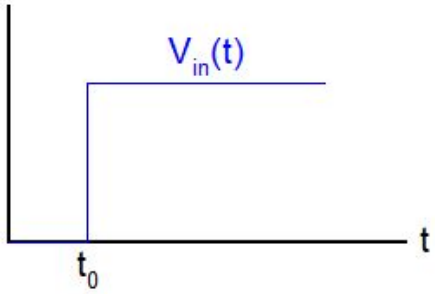
Circuit

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

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$

$$I_{dsn}(t) = \begin{cases} 0 & t \le t_0 \\ \frac{\beta}{2}(V_{DD} - V_t)^2 & V_{out} > V_{DD} - V_t \\ \beta\left(V_{DD} - V_t - V_{out}(t)/2\right)V_{out}(t) & V_{out} < V_{DD} - V_t \end{cases}$$

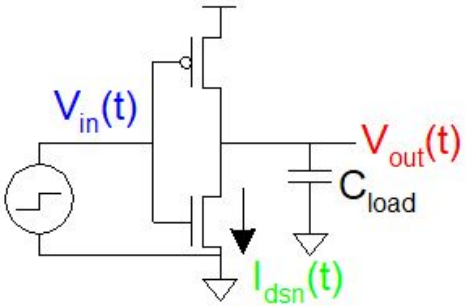


Input voltage

Inverter Transient Response

Example: Transient response of a CMOS inverter

- 3. solve the equation

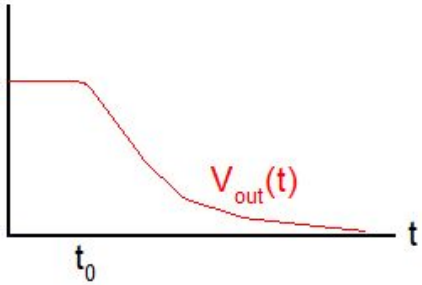


$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$

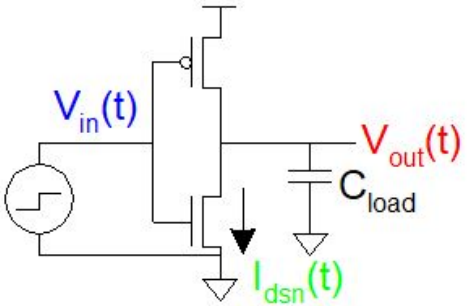
$$I_{dsn}(t) = \begin{cases} 0 & t \le t_0 \\ \frac{\beta}{2}(V_{DD} - V_t)^2 & V_{out} > V_{DD} - V_t \\ \beta\left(V_{DD} - V_t - V_{out}(t)/2\right)V_{out}(t) & V_{out} < V_{DD} - V_t \end{cases}$$



Inverter Transient Response

Example: Transient response of a CMOS inverter

- 4. ***delay*** is the time when the output reaches $V_{DD}/2$ ($t_0 = 0$)

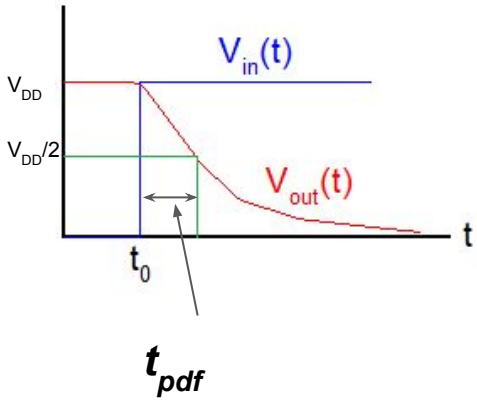


$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$

$$I_{dsn}(t) = \begin{cases} 0 & t \le t_0 \\ \frac{\beta}{2}(V_{DD} - V_t)^2 & V_{out} > V_{DD} - V_t \\ \beta\left(V_{DD} - V_t - V_{out}(t)/2\right)V_{out}(t) & V_{out} < V_{DD} - V_t \end{cases}$$



Transient Response

DC analysis tells us V_{out} if V_{in} is constant

Transient analysis tells us $V_{out}(t)$ if $V_{in}(t)$ changes

➔ Requires solving differential equations (which requires a lot of time and resources)

We would like to be able to easily estimate delay

➔ Not as accurate as simulation or solving differential equations

Use RC delay model!

➔ C = total capacitance on output node

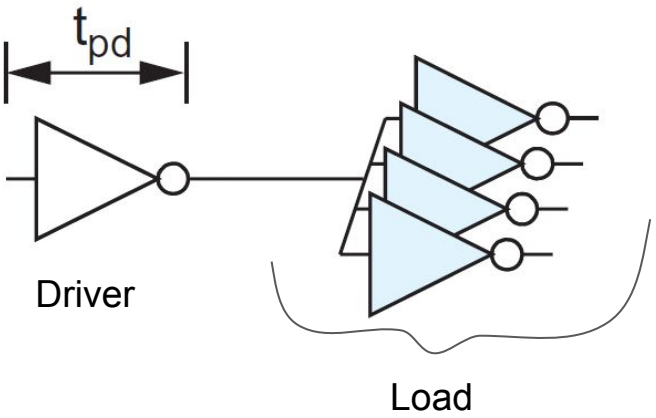
➔ R = effective resistance of the MOS

➔ So that $t_{pd} = RC$

Driver and Load

The gate that charges or discharges a node is called the ***driver***

The gates and wire being driven are called the ***load***



RC Delay Model

In RC Delay Model we need 2 important parameters:

1. Effective Resistance, ***R***
   a. Treat each transistor as a switch in series with a resistor
   b. The effective resistance ***R*** is the ratio of V_{ds} to average I_{ds}
2. Capacitance, ***C***
   a. Each transistor has gate and diffusion capacitance
   b. Approximate the gate and diffusion capacitances as the gate capacitance

RC Delay Model

Effective Resistance ***R*** ∝ 1 / transistor width ***k***

1. **nMOS** Effective Resistance

- A unit nMOS transistor (4/2 λ) is defined to have effective resistance ***R***
- A k times wide nMOS transistor (4k/2 λ) then has effective resistance ***R/k***

2. **pMOS** Effective Resistance

- A unit pMOS transistor (4/2 λ) is defined to have effective resistance ***2R***
- A k times wide pMOS transistor (4k/2 λ) then has effective resistance ***2R/k***

RC Delay Model

Capacitance ***C*** ∝ transistor width ***k***

1. **nMOS** Capacitance
- A unit nMOS transistor (4/2 λ) is defined to have capacitance ***C***
- A k times wide nMOS transistor (4k/2 λ) then has capacitance ***kC***

2. **pMOS** Capacitance
- A unit pMOS transistor (4/2 λ) is defined to have capacitance ***C***
- A k times wide pMOS transistor (4k/2 λ) then has capacitance ***kC***

Typical RC Values

Unit Transistor

- refer to minimum contacted device; W = 4λ, L = 2λ, **W/L = 4λ / 2λ = 4/2 λ**

Resistance

- **R ≈ 10 kΩ.µm** in *0.6 µm* process
- Improves with shorter channel lengths
- **R ≈ 1.25 kΩ.µm** in *65 nm* process

Capacitance

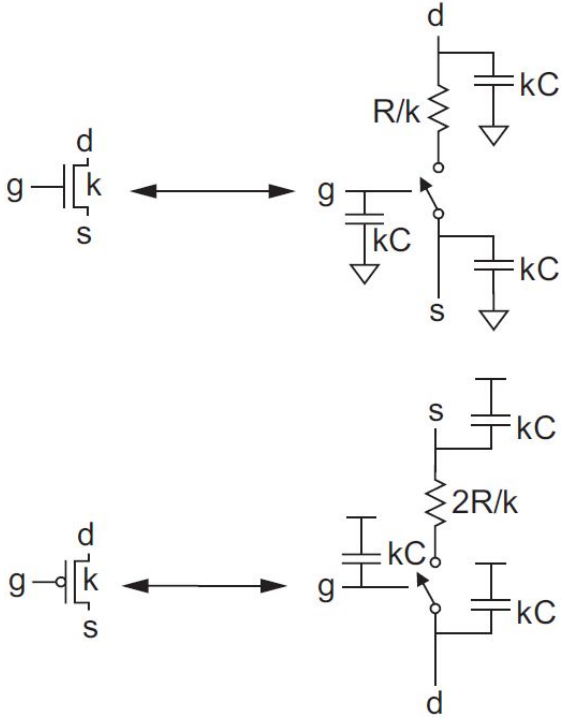
- **$C = C_g = C_s = C_d = 2$ fF/µm** in *0.6 µm* process
- Gradually decline to **1 fF/µm** in *65 nm* process

Equivalent RC Circuits

Equivalent RC circuit for MOSFET

1. **nMOS** of width “k”
- ideal switch + effective resistance of ***R/k*** between source and drain
- capacitance ***kC*** at source, drain and gate to ground

2. **pMOS** of width “k”
- ideal switch + effective resistance of ***2R/k*** between source and drain
- capacitance ***kC*** at source, drain and gate to supply

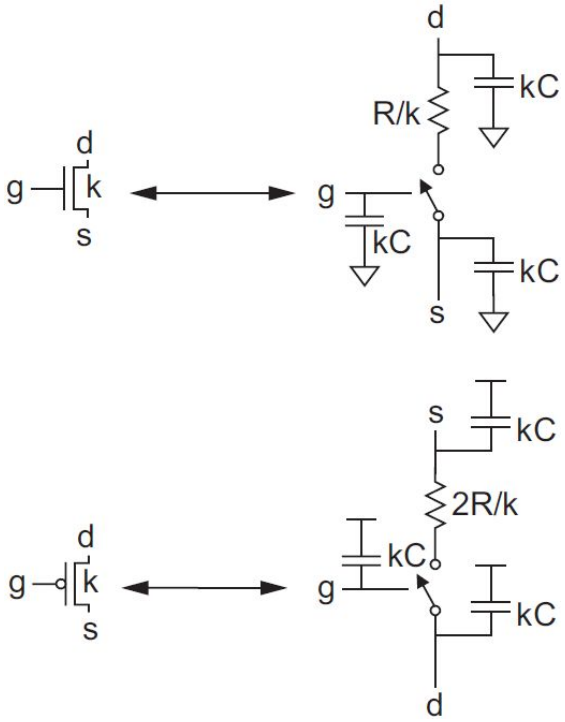


Equivalent RC Circuits

Second terminal of the capacitance, C

1. **nMOS** of width “k”
- capacitance ***kC*** at source, drain and gate to ground
- Capacitance is placed across ground because nMOS body is usually tied to ground (body tapping)

2. **pMOS** of width “k”
- capacitance ***kC*** at source, drain and gate to supply
- Capacitance is placed across supply V_{DD} because pMOS body is usually tied to V_{DD} (well tapping)

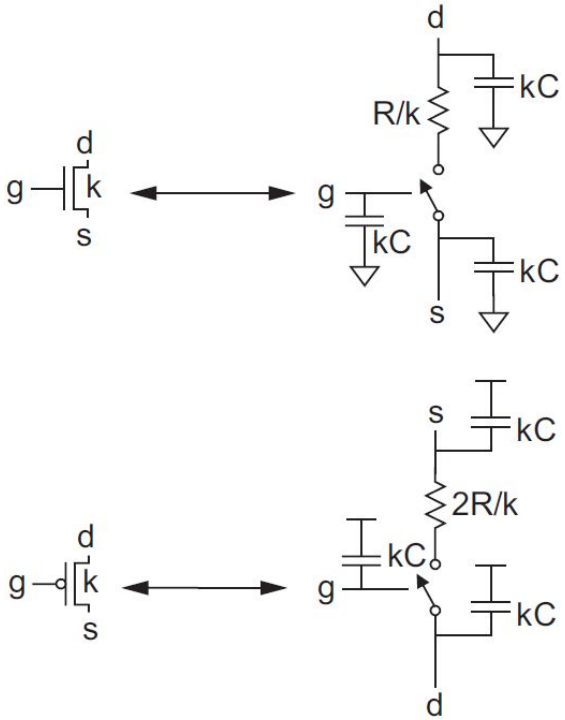


Equivalent RC Circuits

Second terminal of the capacitance, C

However, the behavior of the capacitor from a delay perspective is independent of the second terminal voltage so long as it is constant (V_{DD}/ground)

Hence, we sometimes draw the second terminal as ground for convenience.



Equivalent RC Circuits

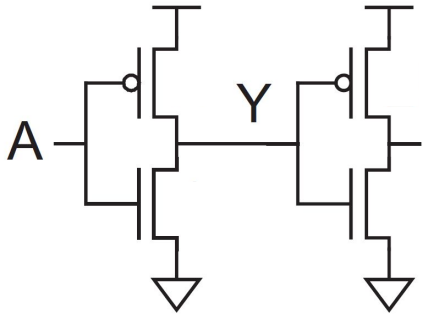
Example 1: Calculate the fanout delay of 1 CMOS unit inverter (FO1 delay)

- Step 1: draw the CMOS circuit
- Step 2: choose the width “k” values for nMOS and pMOS transistors individually from the given condition
- Step 3: place the capacitances using the calculated “k” values
- Step 4: simplify the circuit by combining/rejecting appropriate capacitances
- Step 5: replace the transistors with effective resistances
- Optional: calculate the delay! (if the circuit is simple enough)

Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

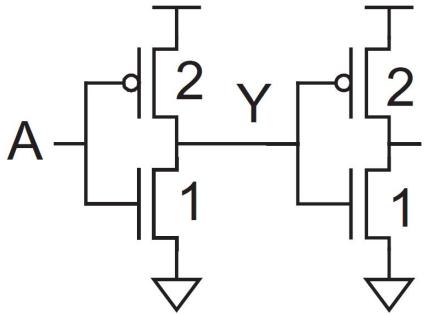
- Step 1: draw the CMOS circuit



Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

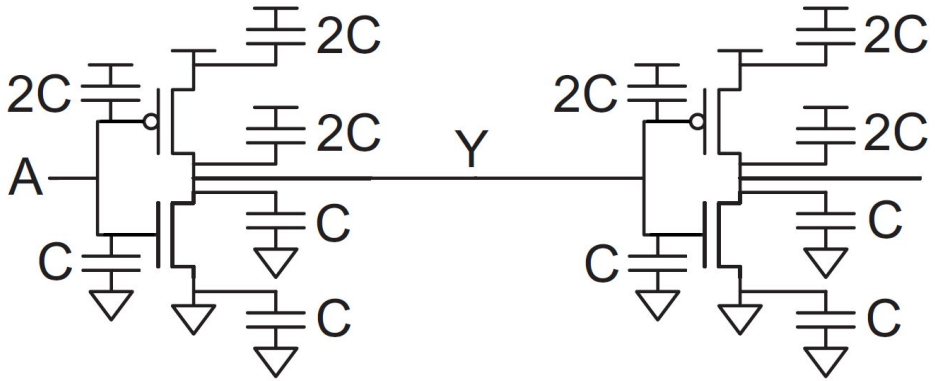
- Step 2: choose the width “k” values for nMOS and pMOS transistors



Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

- Step 3: place the capacitances using the calculated “k” values



Equivalent RC Circuits

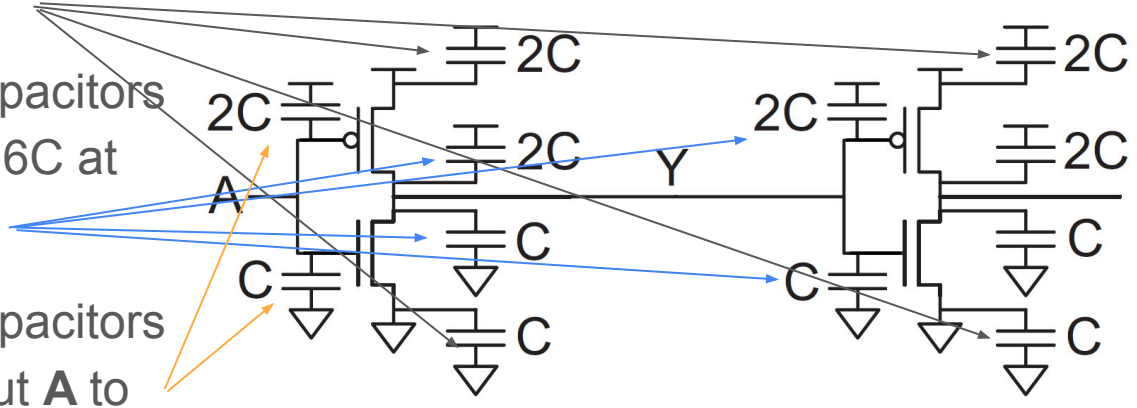
Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

- Step 4: simplify the circuit by combining/rejecting appropriate capacitances

  shorted capacitances

  combine 4 parallel capacitors (2C + C + 2C + C) -> 6C at output **Y** to *ground*

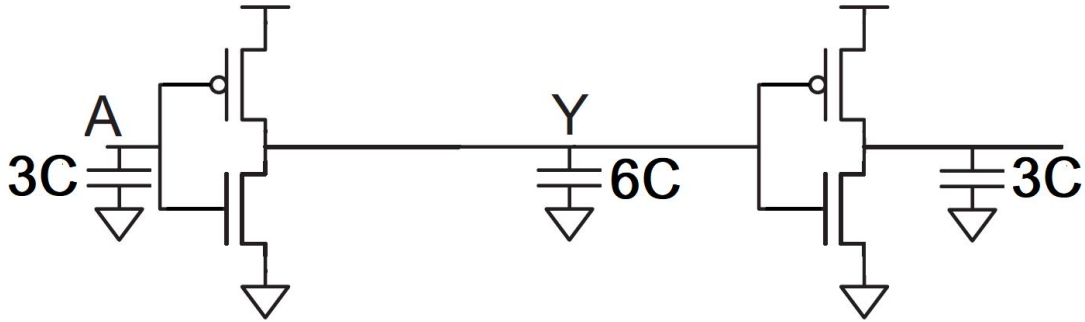
  combine 2 parallel capacitors (2C + C) -> 3C at input **A** to *ground*



Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

- Step 4: simplify the circuit by combining/rejecting appropriate capacitances



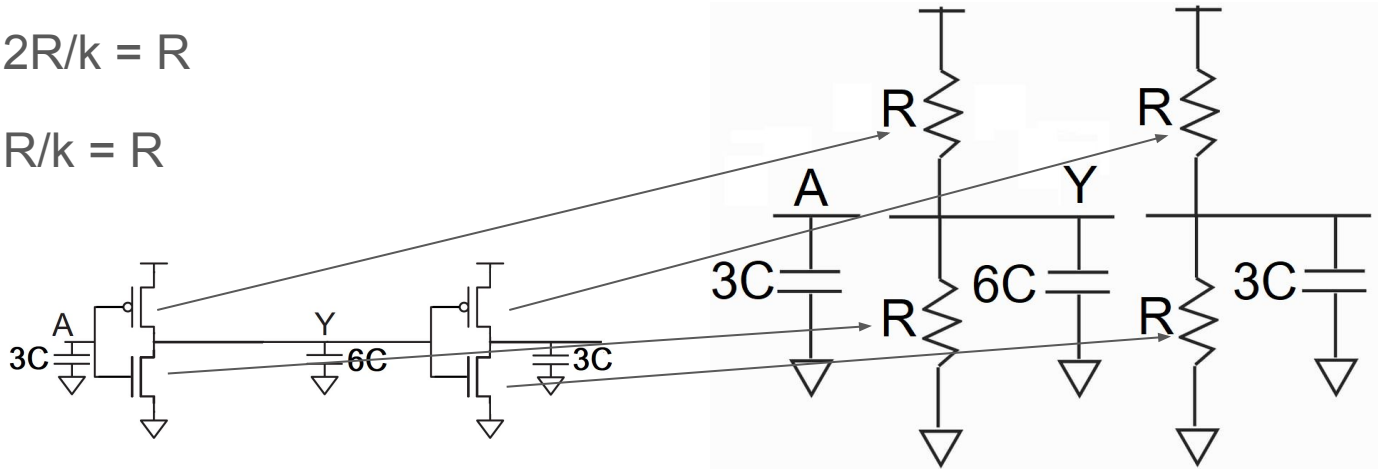
Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

- Step 5: replace the transistors with effective resistances

  pMOS: k = 2; 2R/k = R

  nMOS: k = 1; R/k = R



Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

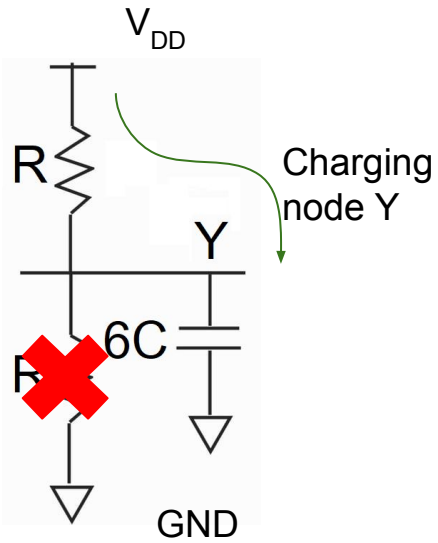
- Optional: calculate the delay!

  t_{pdr} ⇒ Delay when Y going 0 to 1

  ⇒ output capacitance charging through PUN

  ⇒ R* 6C = **6RC**

  t_{cdr} = t_{pdr} because only 1 combination



Equivalent RC Circuits

Calculate the RC propagation delay for an unit inverter driving another unit inverter, each having equal fall and rise resistances (fanout-of-1 delay)

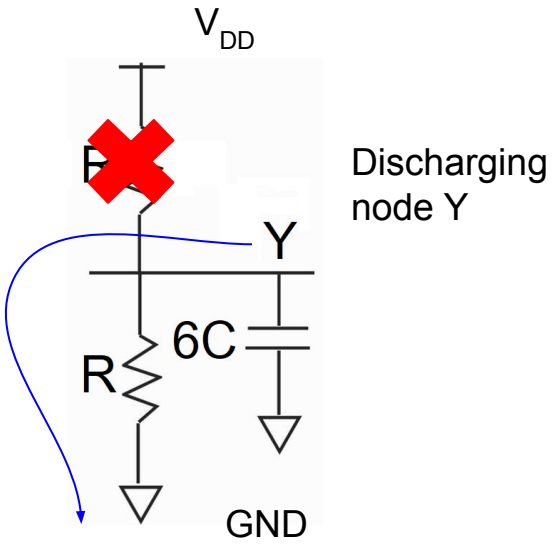
- Optional: calculate the delay!

  t_{pdf} ⇒ Delay when Y going 1 to 0

  ⇒ output capacitance discharging through PDN

  ⇒ R* 6C = **6RC**

  t_{cdf} = t_{pdf} because only 1 combination



Equivalent RC Circuits

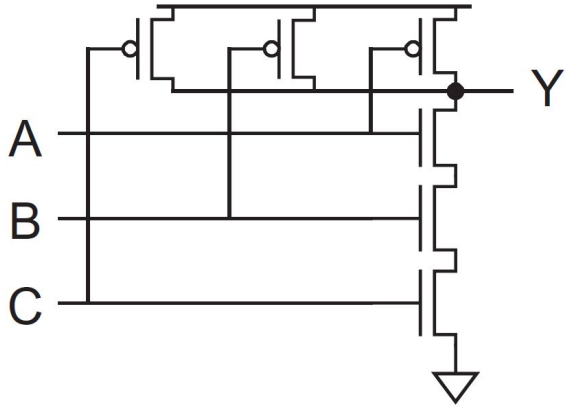
Example 2: Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (***R***)

- Step 1: draw the NAND3 CMOS gate
- Step 2: choose the width “k” values for nMOS and pMOS transistors individually from the given condition
- Step 3: place the capacitances using the calculated “k” values
- Step 4: simplify the circuit by combining/rejecting appropriate capacitances
- Step 5: replace the transistors with effective resistance

Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

- Step 1: draw the NAND3 CMOS gate



Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

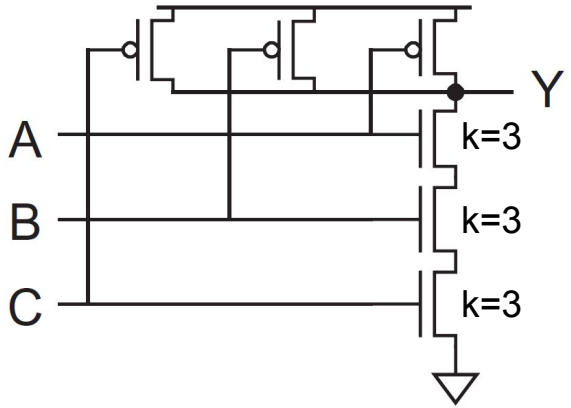
- Step 2: choose the width “k” values for nMOS and pMOS individually

  Eq. resistance of a single nMOS = ***R/k***

  ∴ Eq. resistance of 3 nMOS in series = ***3R/k***

  But we require ***3R/k = R***

  ⇒ ***k = 3***



Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

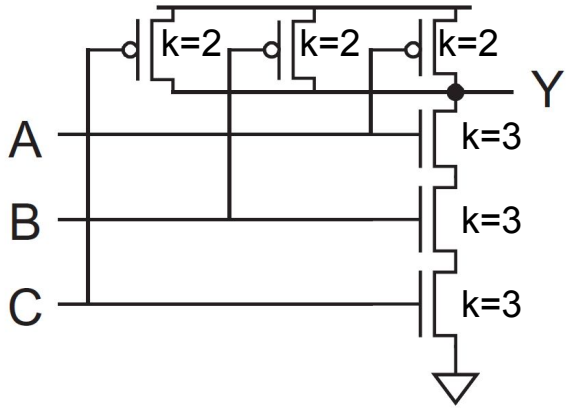
- Step 2: choose the width “k” values for nMOS and pMOS individually

  Eq. resistance of a single pMOS = ***2R/k***

  3 pMOS are in parallel

  In the worst case only 1 is ON, other 2 are OFF

  Then, we require ***2R/k = R ⇒ k = 2***



Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

- Step 3: place the capacitances using the calculated “k” values

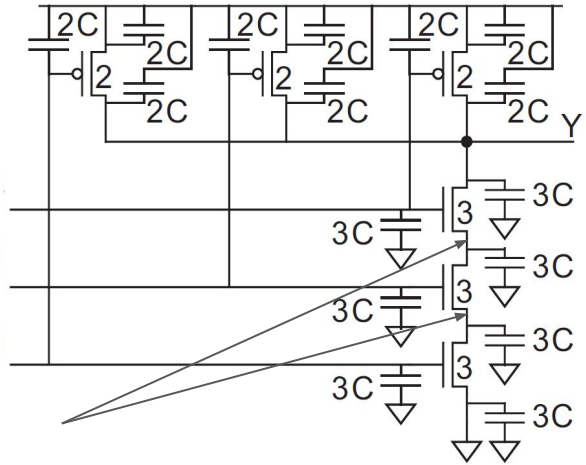
  Each **pMOS has k = 2**

  Place a **kC** or **2C** capacitor at each terminal to V_{DD}

  Each **nMOS has k = 3**

  Place a **kC** or **3C** capacitor at each terminal to gnd

  Shared terminal for series transistors have 1 capacitor instead of 2



Equivalent RC Circuits

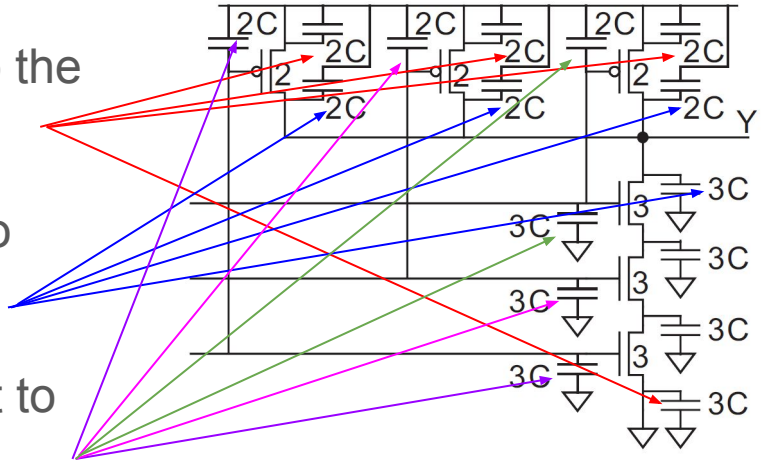
*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

- Step 4: simplify the capacitances

  capacitors on source diffusions attached to the rails have both terminals shorted together

  combine 4 parallel capacitors at output Y to *ground* (2C + 2C + 2C + 3C)

  combine 2 parallel capacitors at each input to *ground* (2C + 3C)



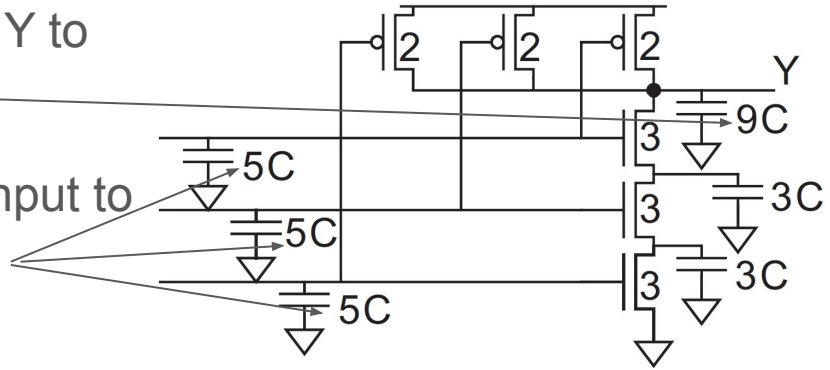
Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

- Step 4: simplify the capacitances

  combine 4 parallel capacitors at output Y to *ground* (2C + 2C + 2C + 3C) -> 9C

  combine 2 parallel capacitors at each input to *ground* (2C + 3C) -> 5C at each input



Equivalent RC Circuits

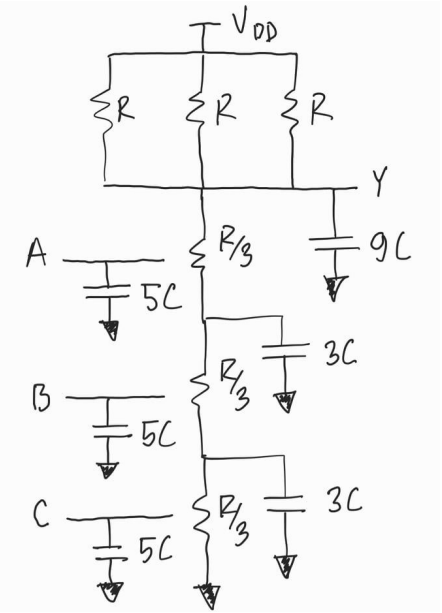
*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*

- Step 5: replace the transistors with effective resistance

  pMOS: k = 2; 2R/k = 2R/2 = R

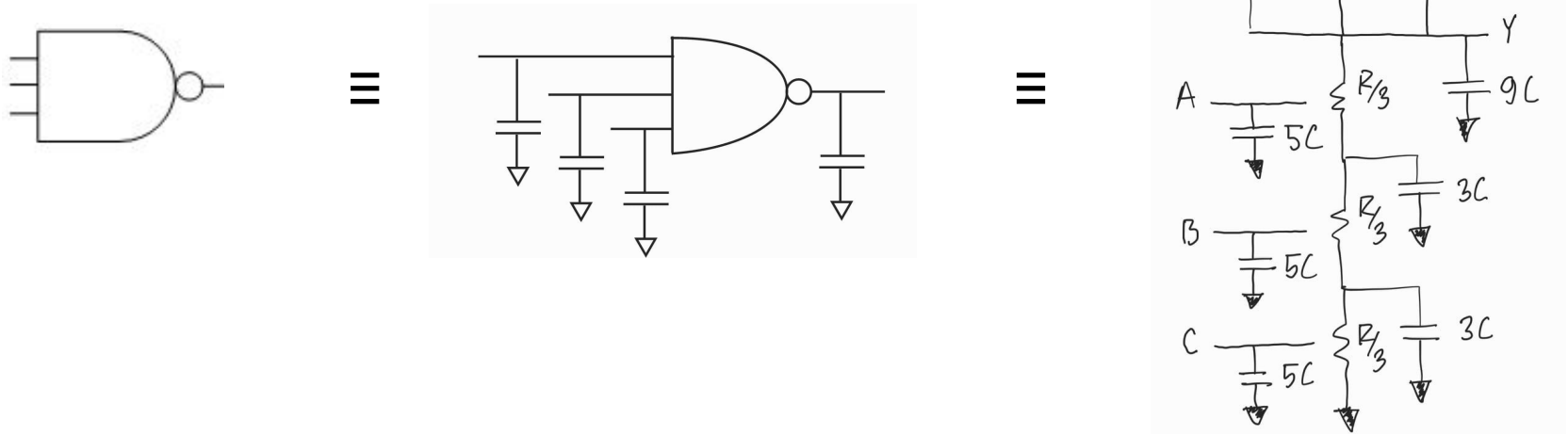
  nMOS: k = 3; R/k = R/3

  Note that the input terminals are not connected to the main circuit



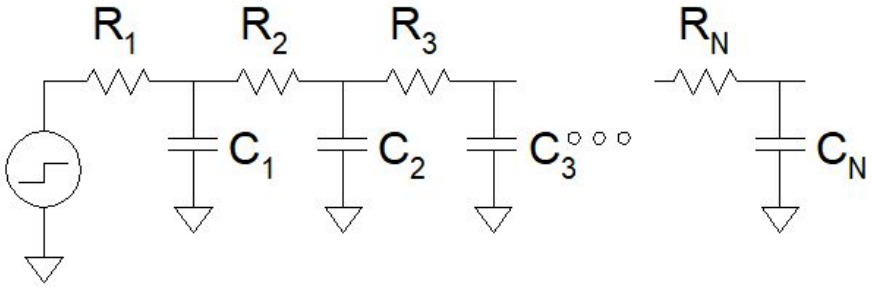
Equivalent RC Circuits

*Sketch a NAND3 RC equivalent circuit with widths chosen to achieve effective rise and fall resistance equal to that of a unit inverter (**R**)*



RC Tree

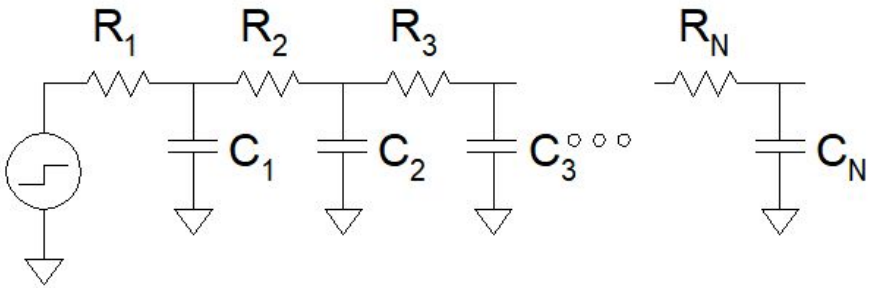
- In general, most circuits of interest can be represented as an RC tree: an RC circuit with no loops, as opposed to the simple single RC branch (compare the previous two examples)
- The root of the tree is the voltage source/ground and the leaves are the capacitors at the ends of the branches
- Pullup or pulldown network modeled as RC tree



Elmore Delay

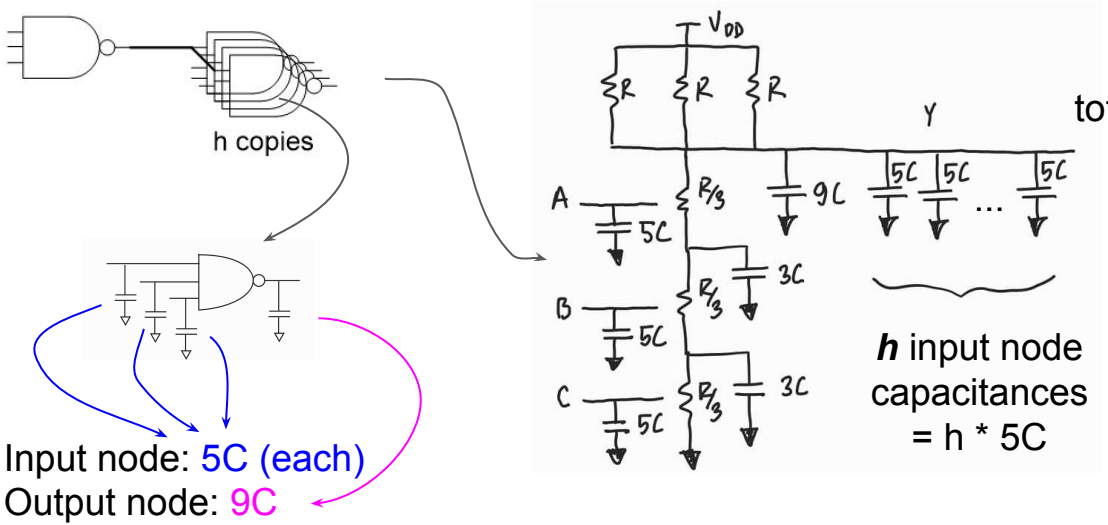
- Elmore delay of the pullup or pulldown RC tree can be calculated as:

$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$
$$= R_1 C_1 + (R_1 + R_2) C_2 + \ldots + (R_1 + R_2 + \ldots + R_N) C_N$$



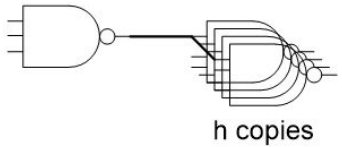
Elmore Delay

Example 3: Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with ***h*** identical NAND gates (FOh delay)

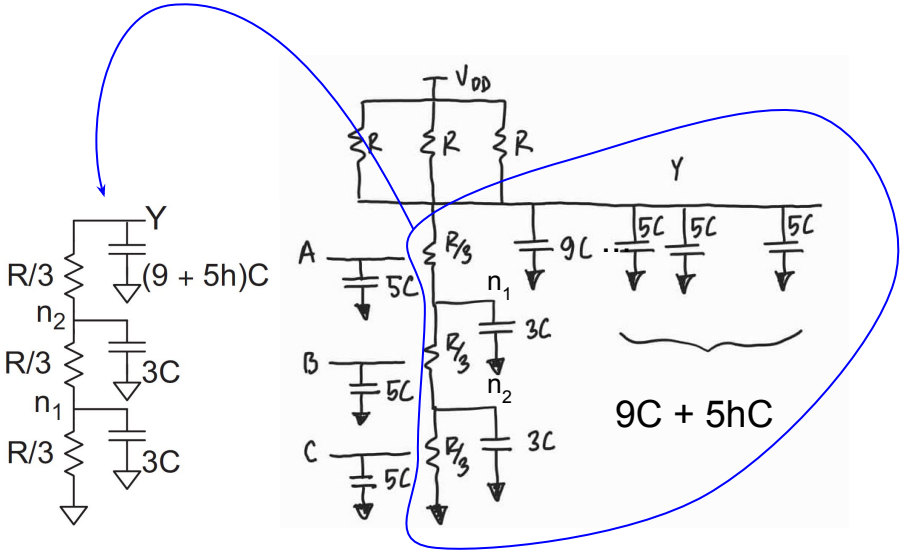


Elmore Delay

*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*

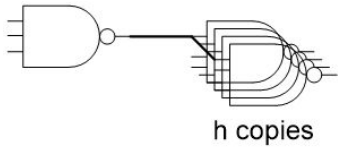


- t_{pdf} is the worst case falling delay
  - All inputs 1 (A=1, B=1, C=1)
  - Nodes Y, n2 & n1 *all* have to discharge
- n1 has capacitance 3C and resistance of R/3 to ground
- n2 has capacitance 3C and resistance (R/3 + R/3) to ground
- Y has capacitance (9 + 5h)C and resistance (R/3 + R/3 + R/3) to ground
- The Elmore delay for the falling output is the sum of these RC products



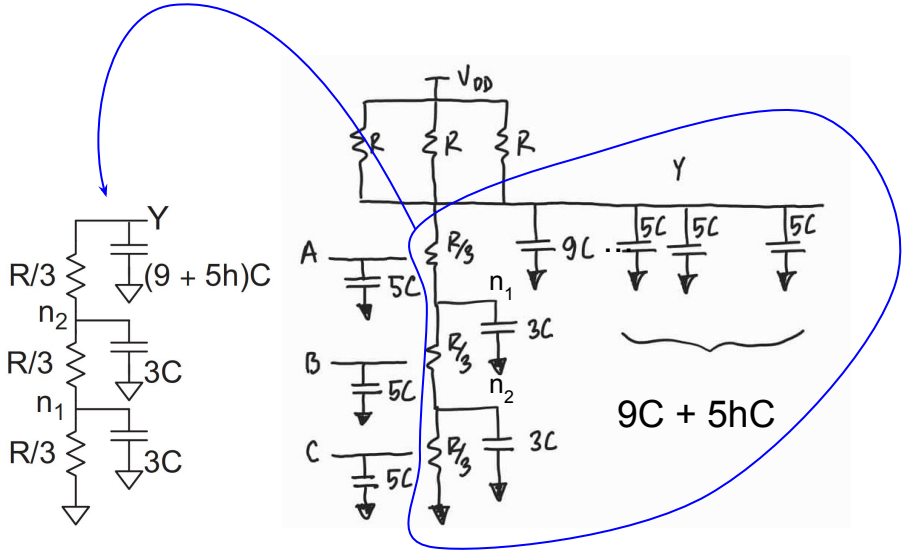
Elmore Delay

*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*



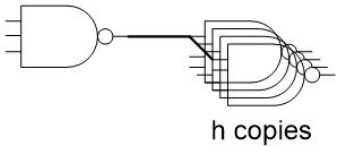
- t_{pdf} is the worst case falling delay
  - All inputs 1 (A=1, B=1, C=1)
  - Nodes Y, n2 & n1 *all* have to discharge

- The Elmore delay for the falling output is the sum of these RC products
- t_{pdf} = (3C)(R/3) +
  (3C)(R/3 + R/3) +
  ((9 + 5h)C)(R/3 + R/3 + R/3)
  = (12 + 5h)RC

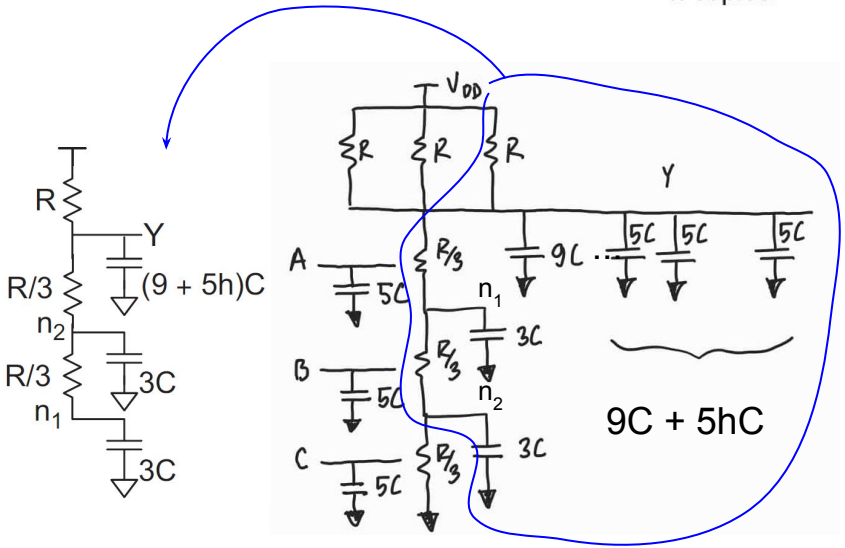


Elmore Delay

*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*

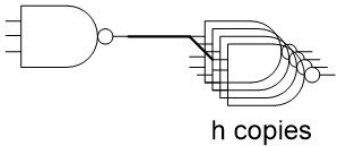


- t_{pdr} is the worst case rising delay
  - 2 inputs 1, 1 input 0 (A=1, B=1, C=0)
  - Nodes Y, n2 & n1 *all* have to charge
- Y has capacitance (9 + 5h)C and resistance R to the VDD
- n2 has capacitance 3C. The relevant resistance is only R, not (R + R/3), because the output is being charged only through R
- n1 has capacitance 3C and resistance R
- The Elmore delay for the rising output is the sum of these RC products



Elmore Delay

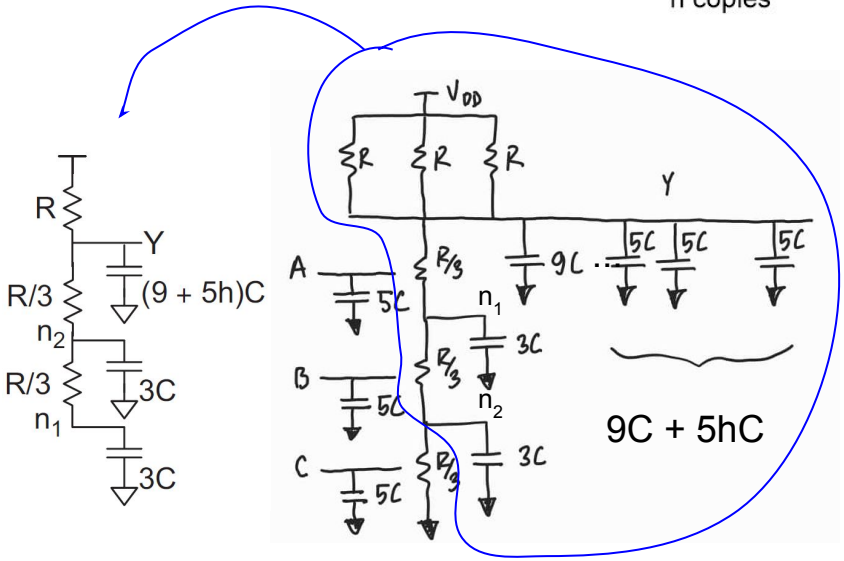
*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*



- t_{pdr} is the worst case rising delay
  - 2 inputs 1, 1 input 0 (A=1, B=1, C=0)
  - Nodes Y, n2 & n1 *all* have to charge

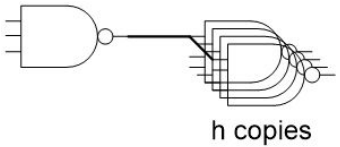
- The Elmore delay for the rising output is the sum of these RC products
- t_{pdr} = ((9 + 5h)C)(R) +
  (3C)(R) +
  (3C)(R)
  = (15 + 5h)RC

Here, the calculated Elmore delay is conservative and the actual delay is somewhat faster

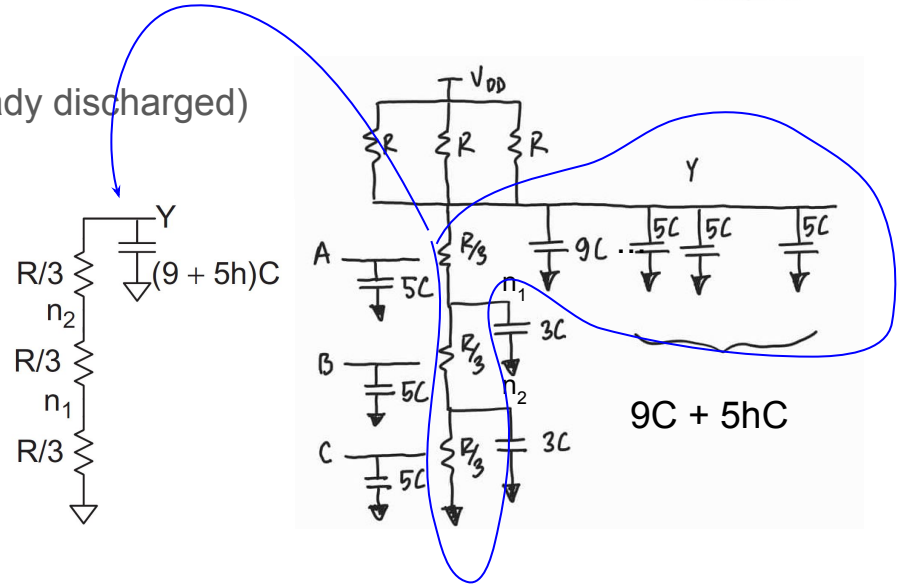


Elmore Delay

*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*

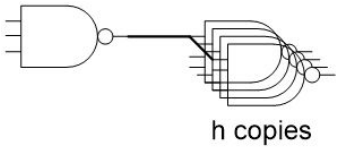


- t_{cdf} is the best case falling delay
  - All inputs 1 (A=1, B=1, C=1)
  - Only nodeY have to discharge (n2, n1 already discharged)
- Y has capacitance (9 + 5h)C and resistance (R/3 + R/3 + R/3) to ground
- The Elmore delay for the falling output is the RC product
- t_{cdf} = ((9 + 5h)C)(R/3 + R/3 + R/3)
  = (9 + 5h)RC

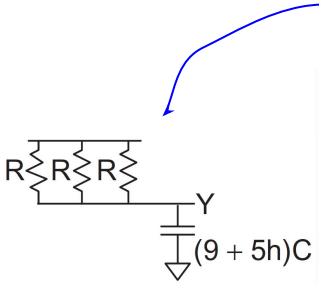


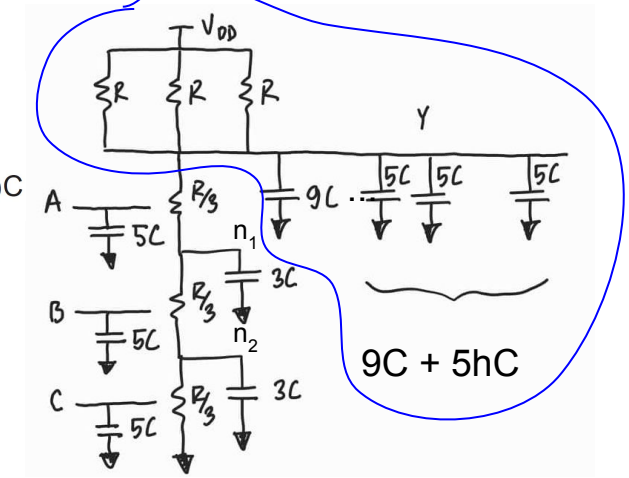
Elmore Delay

*Estimate t_{pdf}, t_{pdr}, t_{cdf}, and t_{cdr} for the 3-input NAND gate from Example 2 if the output is loaded with **h** identical NAND gates*



- t_{cdr} is the best case rising delay
  - All inputs 0 (A=0, B=0, C=0)
  - *Only* node Y have to charge
- Y has capacitance (9 + 5h)C and resistance (R || R || R) or R/3 to the VDD
- The Elmore delay for the rising output is the sum of these RC products
- t_{cdr} = ((9 + 5h)C)(R/3)
  = (3 + (5/3)h)RC





Delay Components

Usually the delay consists of two major components

1. The ***parasitic delay*** is the time for a gate to drive its own internal diffusion capacitance
2. The ***effort delay*** depends on h and is the time for a gate to drive the load capacitance

In the last example, t_{pdf} was found to be (12 + 5h)RC or 12*RC + 5h*RC, where:

- 12*RC is the *parasitic delay* component
- 5h*RC is the *effort delay* component

Thank You!