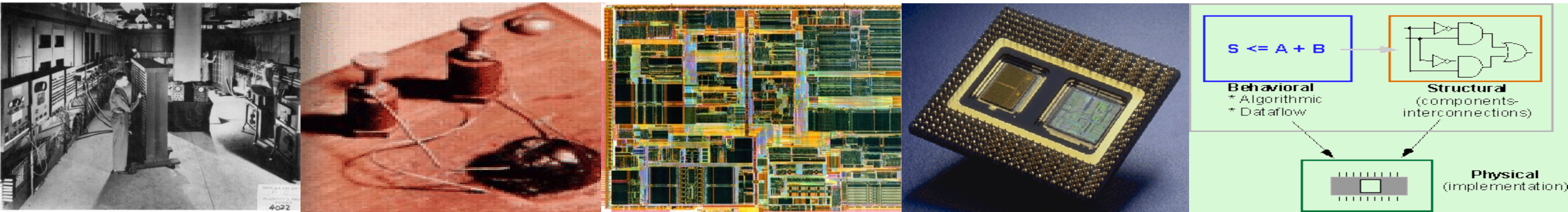
CpE 690: Introduction to VLSI Design

Lecture 8
Transient Response and Delay

Bryan Ackland
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

Adapted from Lecture Notes, David Mahoney Harris CMOS VLSI Design



Activity

1) If the width of a transistor increases, the current will

increase decrease not change

2) If the length of a transistor increases, the current will

increase decrease not change

3) If the supply voltage of a chip increases, the maximum transistor current will

increase decrease not change

4) If the width of a transistor increases, its gate capacitance will

increase decrease not change

5) If the length of a transistor increases, its gate capacitance will

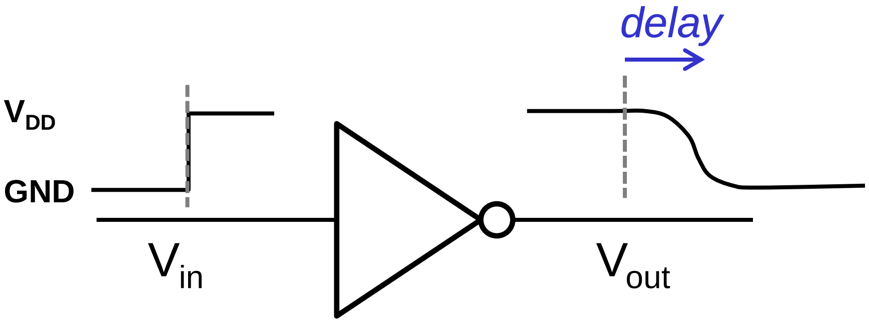
increase decrease not change

6) If the supply voltage of a chip increases, the gate capacitance of each transistor will

increase decrease not change

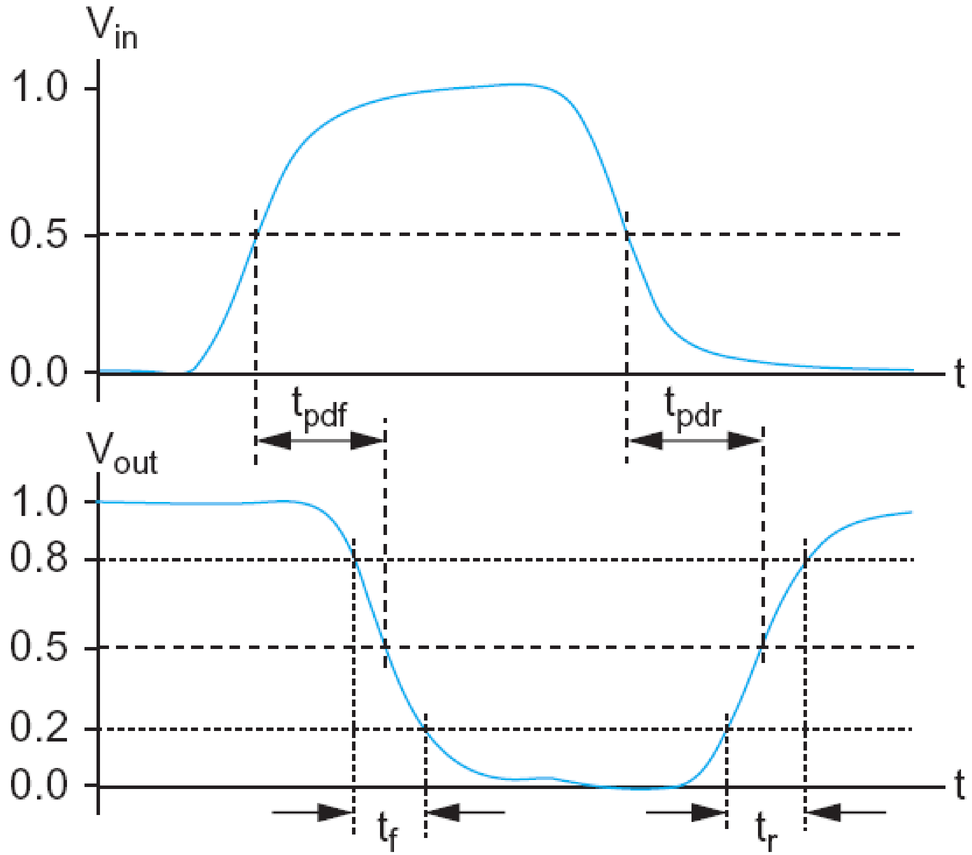
Transient Response

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



Delay Definitions

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

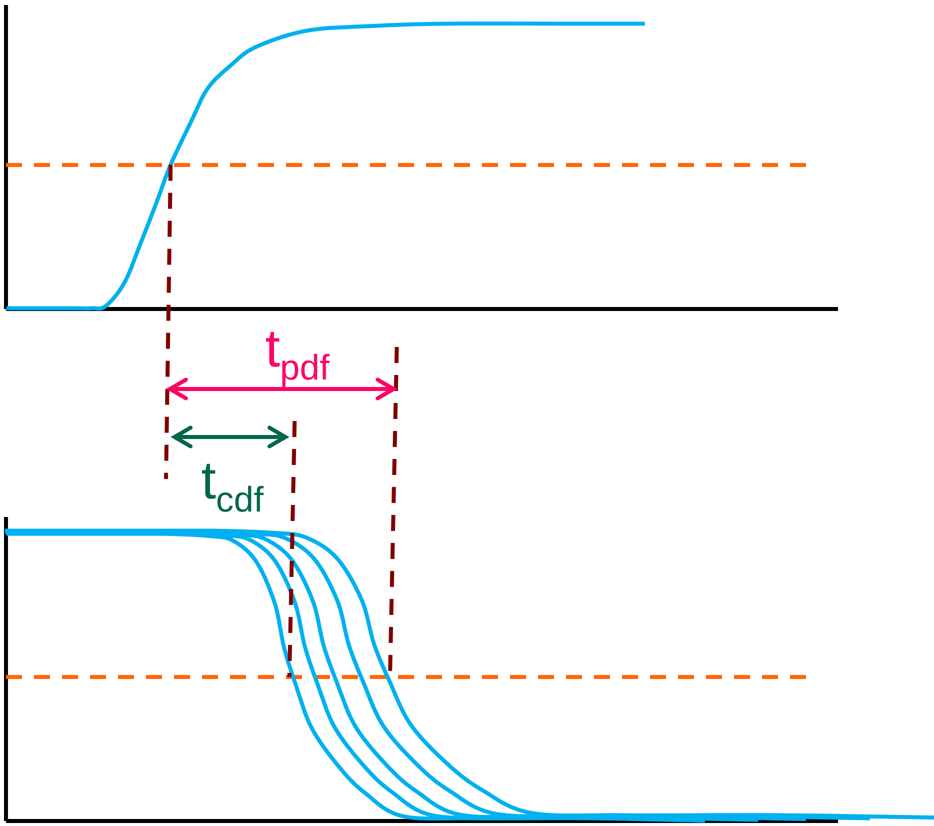


Delay Definitions (cont.)

- **t_{cdf}** : *falling contamination delay*
  - minimum time from input crossing $V_{DD}/2$ to falling output crossing $V_{DD}/2$

- **t_{cdr}** : *rising contamination delay*
  - minimum time from input crossing $V_{DD}/2$ to rising output crossing $V_{DD}/2$

- **t_{cd}** : *avg. contamination delay*
  - $t_{pd} = (t_{cdr} + t_{cdf})/2$

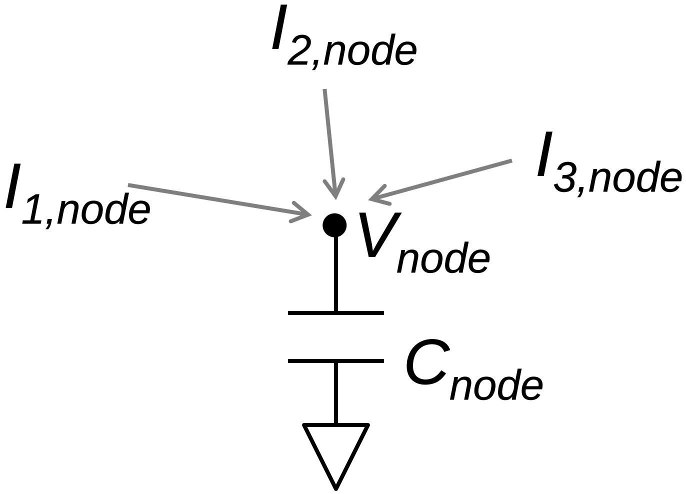


Delay in CMOS Circuits

- Switching CMOS gate generates output current in response to changing input voltages
- All nodes have some finite capacitance (to ground)
  - gate capacitance
  - parasitic source/drain (diode) capacitance
  - parasitic wiring capacitance
- Transient waveforms found by solving:

$$C_{node} \bullet (dV_{node}/dt) = \sum_k I_{k,node}$$

for each node in circuit



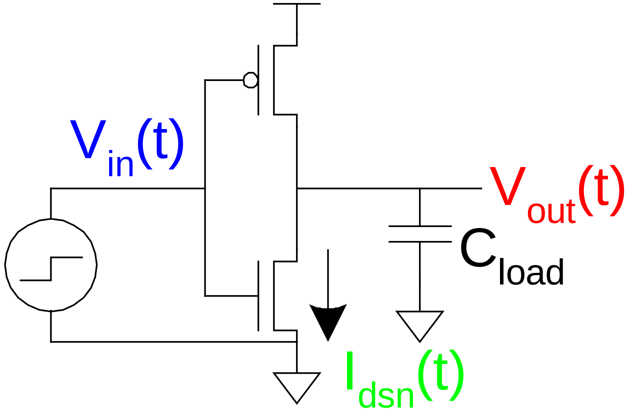
Inverter Step Response

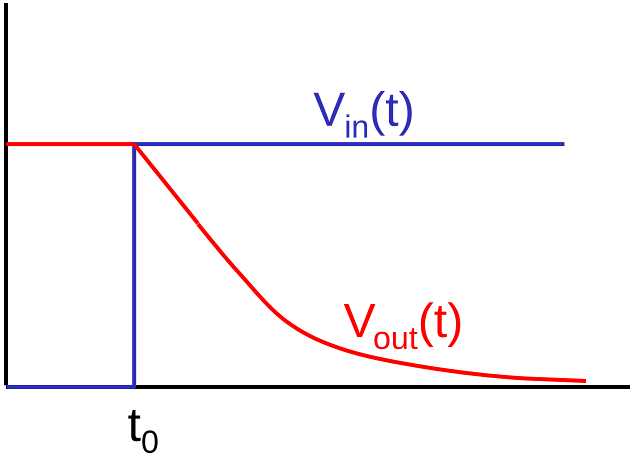
- Find step response of inverter driving C_{load}

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

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

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

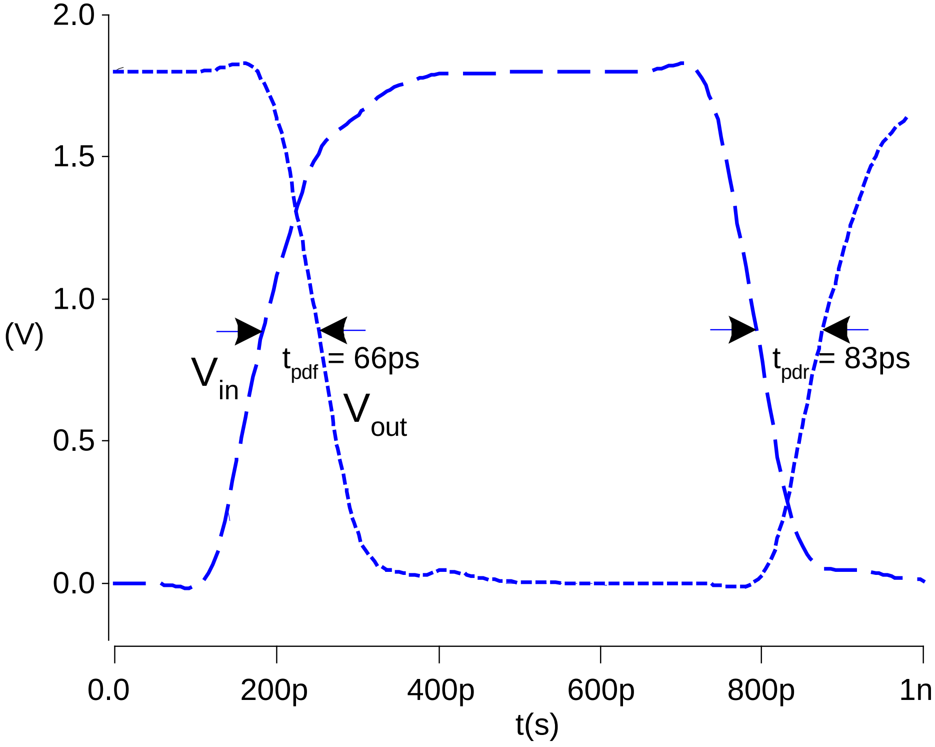




$$I_{dsn}(t) = \begin{cases} 0 & \text{for } t < t_0 \\ (\beta/2)\bullet(V_{DD}-V_t)^2 & \text{for } V_{out} > V_{DD} - V_t \\ \beta\bullet(V_{DD}-V_t-V_{out}(t)/2)\bullet V_{out}(t) & \text{for } Vout < V_{DD} - V_t \end{cases}$$

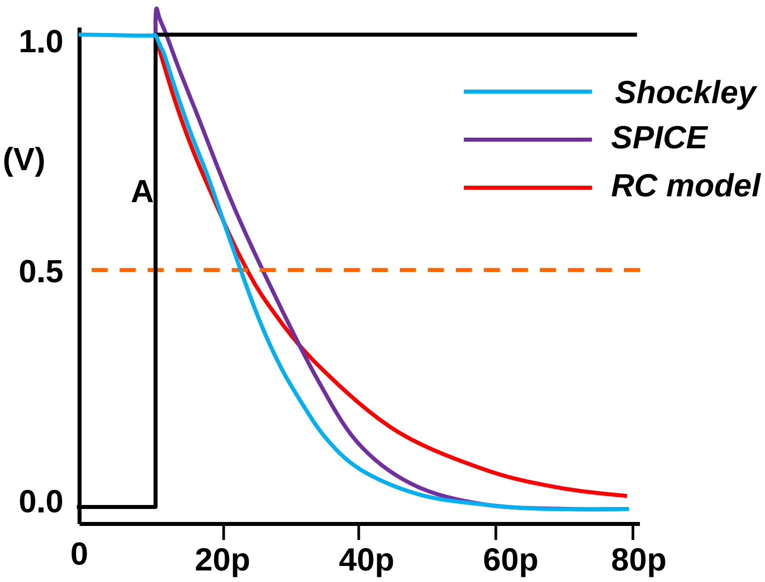
Simulated Inverter Delay

- Solving differential equations by hand is too hard
- SPICE simulator solves the equations numerically
- Uses more accurate I-V models too!
- But simulations take time to write!



Delay Estimation

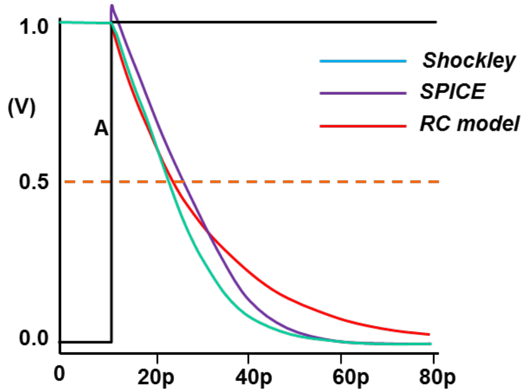
- We would like to be able to easily estimate delay
  - For exploration of design space, don’t need to be as accurate as simulation
  - Want a technique where its easier to ask “What if?”
- The step response usually looks like a 1st order RC response with a decaying exponential.
- Can we model conducting transistor as effective resistance?



Effective Resistance

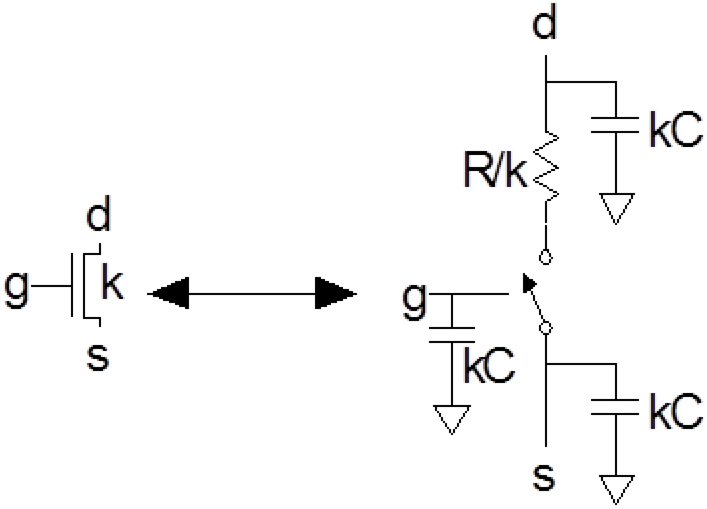
- Simplification: treat transistor as resistor
  - – Replace $I_{ds}(V_{ds}, V_{gs})$ with effective resistance R
  - – $I_{ds} = V_{ds}/R$ or 0 depending on gate voltage

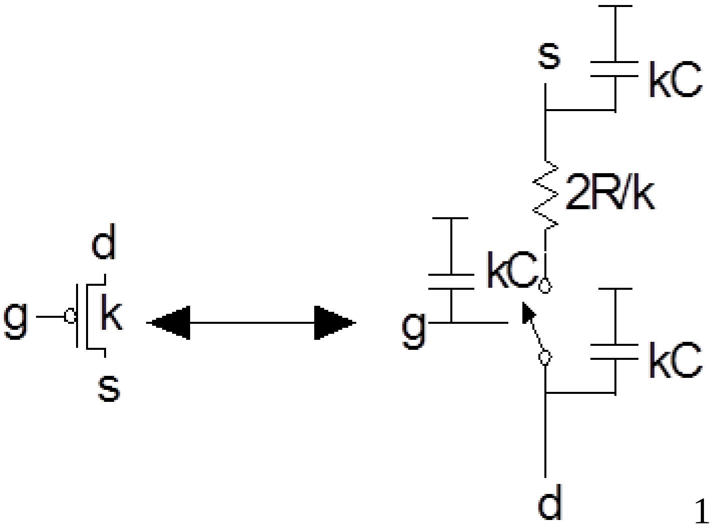
- R averaged across switching of digital gate
  - – Too inaccurate to predict current at any given time
  - – But good enough to predict gate delay



RC Delay Model

- Use equivalent circuits for MOS transistors
  - Ideal switch + capacitance and ON resistance
  - Unit nMOS has resistance R, capacitance C
  - Unit pMOS has resistance 2R, capacitance C
- Capacitance (gate & diffusion) proportional to width
- Resistance inversely proportional to width





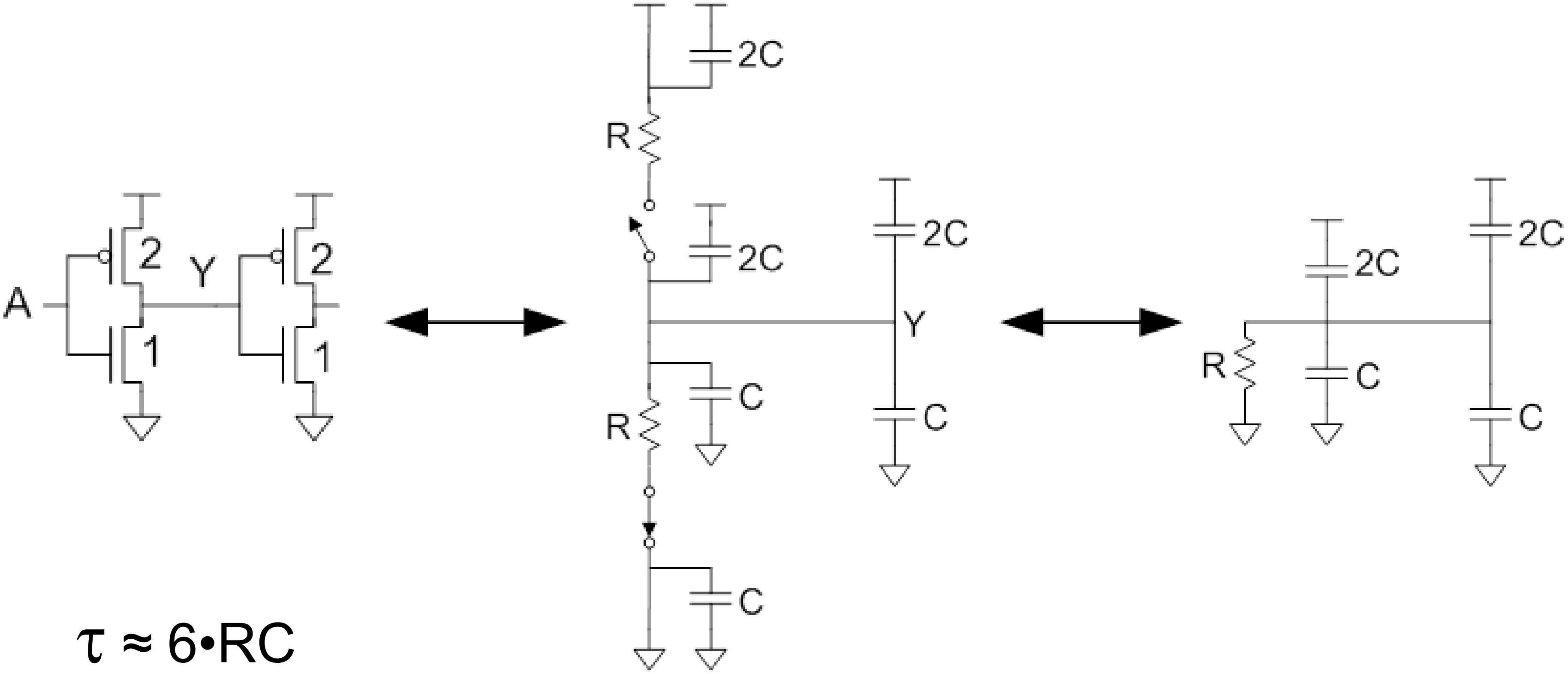
RC Values

- Capacitance
  - $C = C_g = C_s = C_d$ = 2 fF/μm of gate width in 0.6 μm
  - Gradually decline to 1 fF/μm in nanometer techs.
- Resistance
  - R ≈ 5-10 KΩ•μm in 0.6 μm process
  - Improves with shorter channel lengths
- Unit transistors
  - May refer to minimum contacted device (4/2 λ)
  - Or maybe W=1 μm device (doesn't matter as long as you are consistent)

	AMI 0.6μm	TSMC 250nm	TSMC 180nm	IBM 130nm	IBM 65nm
R_n (kΩ.μm)	9.2	4.0	2.7	2.5	1.3
R_n (kΩ.4λ)	7.7	8.0	7.5	9.6	10
R_p (kΩ.μm)	19.9	8.9	6.5	6.4	2.9
R_p (kΩ.4λ)	16.6	17.8	18.1	24.7	22.3

RC Values

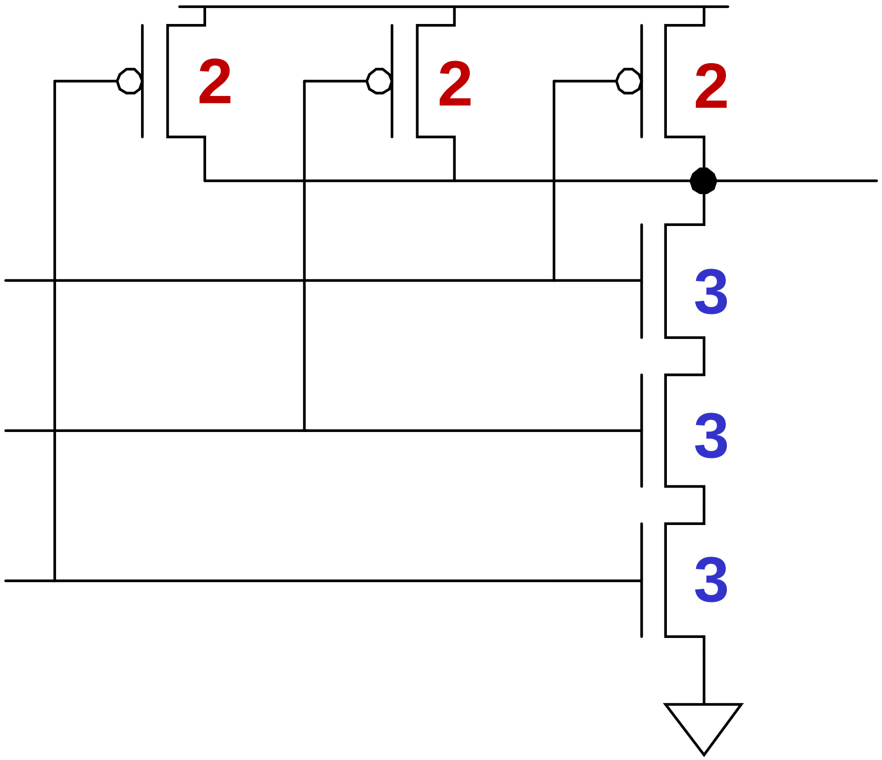
- Estimate the delay of a fanout-of-1 inverter



$\tau \approx 6 \cdot RC$

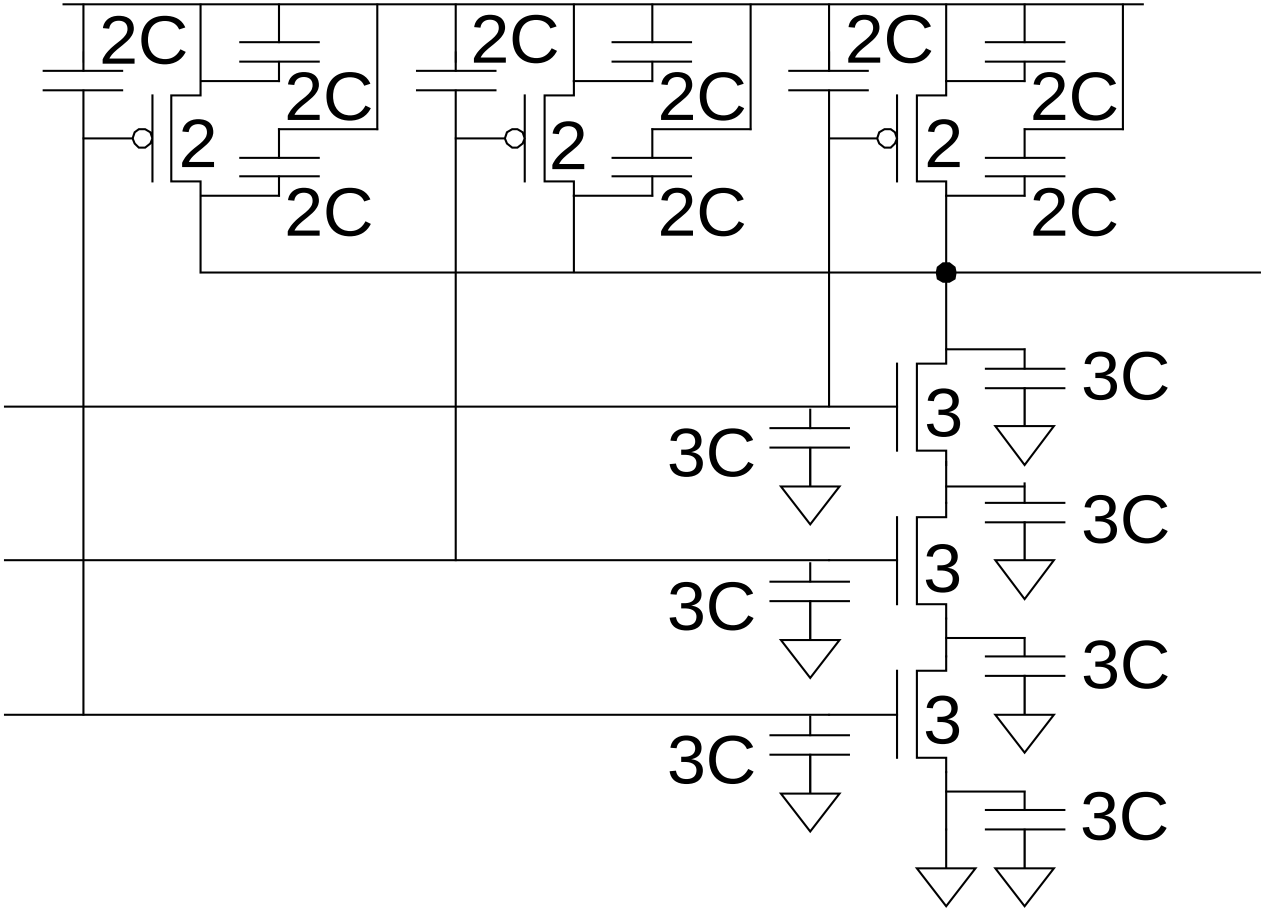
Example: 3-input NAND

- Sketch a 3-input NAND with transistor widths chosen to achieve (worst case) effective rise and fall resistances equal to a unit inverter (R).



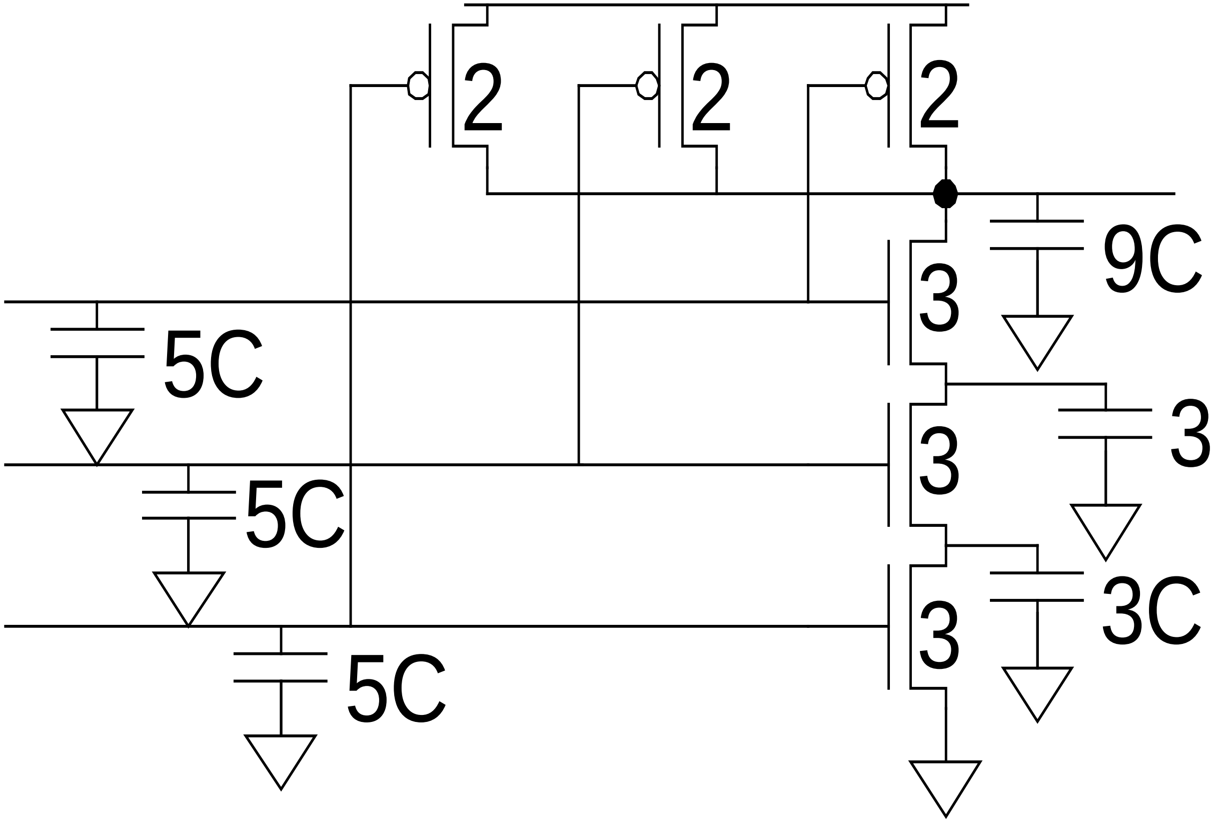
3-input NAND Capacitors

- Annotate the 3-input NAND gate with gate and diffusion capacitance.



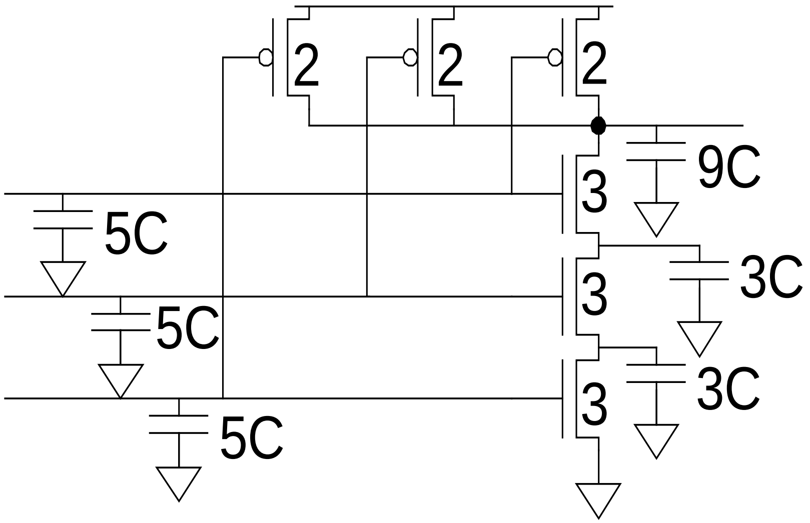
3-input NAND Capacitors

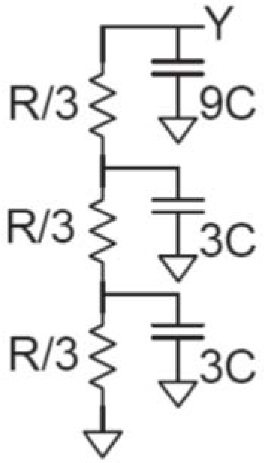
- Annotate the 3-input NAND gate with gate and diffusion capacitance.



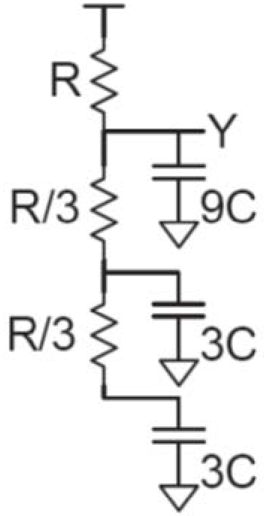
Rise & Fall Delay

- What are worst-case rise and fall delays?





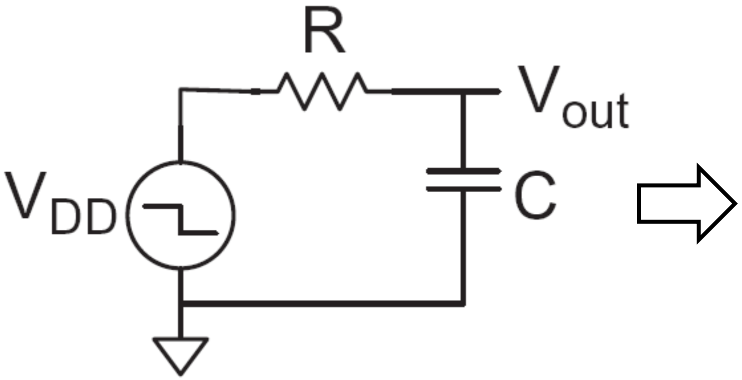
Falling

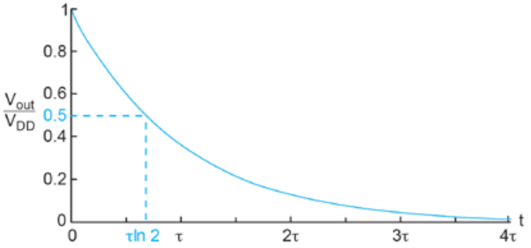


Rising

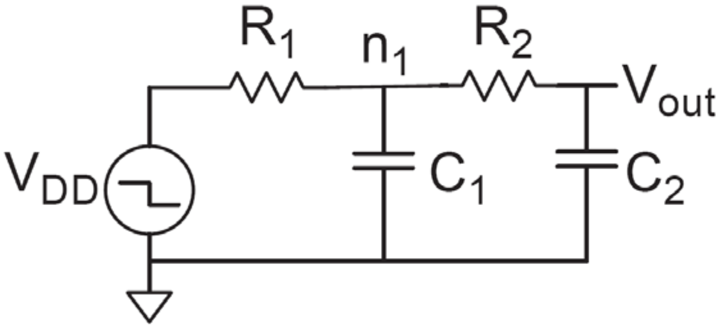
- How can we estimate delay of these networks?

τ with multiple RC components





τ = RC



$$H(s) = \frac{1}{1 + s[R_1C_1 + (R_1 + R_2).C_2] + s^2R_1C_1R_2C_2}$$

- Second order response is too complicated
  - defeats whole purpose of simplifying to an RC network
- Can approximate to:

$$\tau = \tau_1 + \tau_2 = R_1C_1 + (R_1+R_2).C_2$$

Elmore Delay

- ON transistors modeled as resistors
- Pullup or pulldown network represented as an RC tree
  - root of tree is driving voltage source (often VDD or GND)
  - leaves are capacitors at end of branches
- Elmore delay to any target (node j) in the branch:

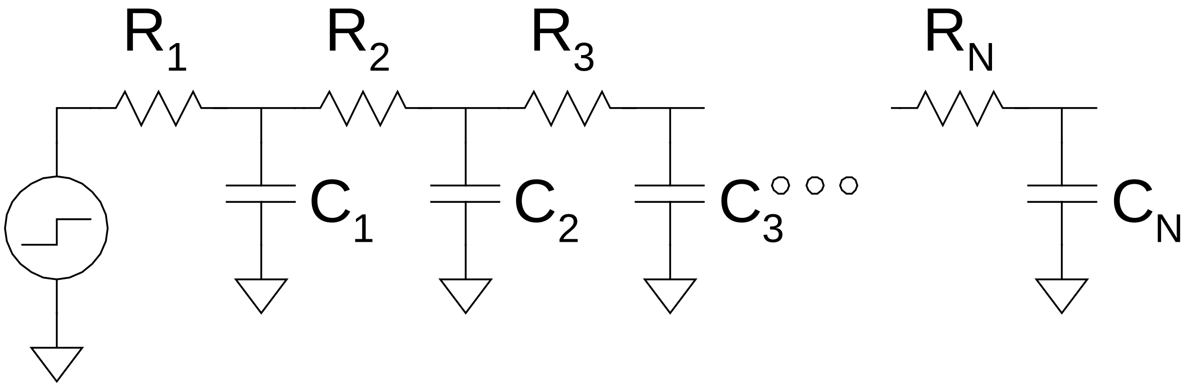
$$t_{pdj} = \sum_i R_{sij}. C_i$$

where:

- i represents all the nodes in the branch
- C_i is the capacitance at node i
- R_{sij} is the resistance of the shared path from the source to $node_i$ and from the source to the target $node_j$

- Elmore delay is conservative
  - over-estimates the delay

Shared Path



- delay to node N is:

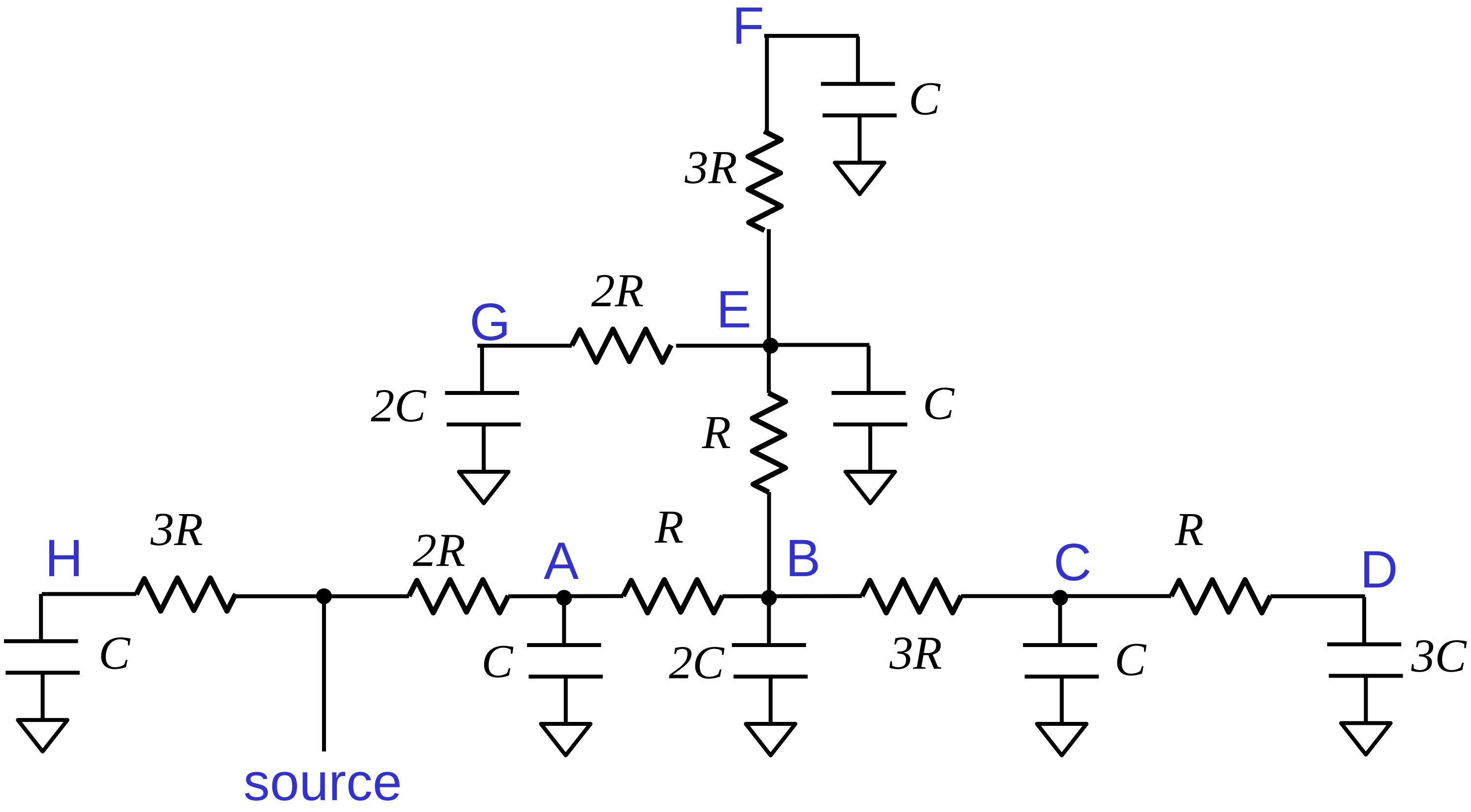
$$R_1C_1 + (R_1+R_2).C_2 + \ldots + (R_1+R_2+\ldots+R_n).C_N$$

- delay to node 2 is:

$$R_1C_1 + (R_1+R_2).C_2 + (R_1+R_2).(C_3+C_4+\ldots+C_N)$$

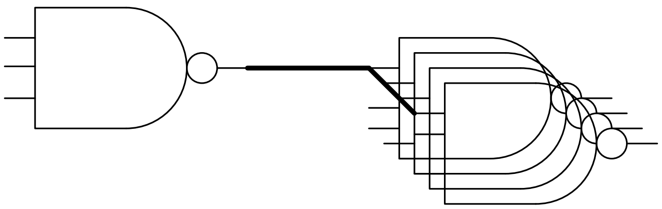
Example: Elmore Delay

- Calculate delay from source to all nodes in circuit:

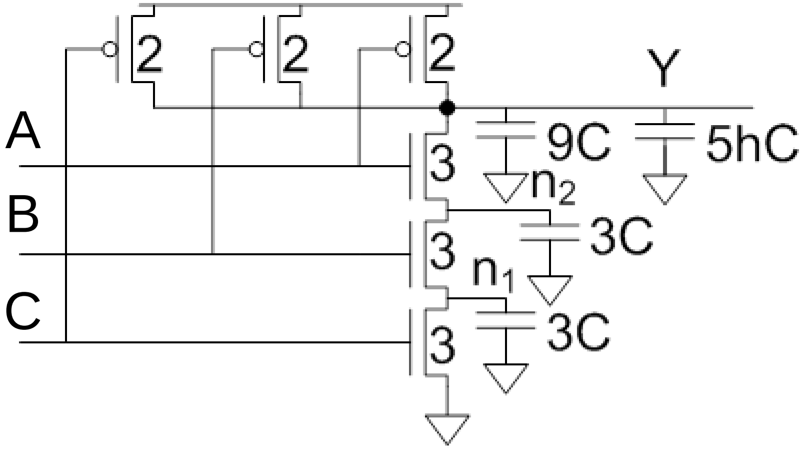


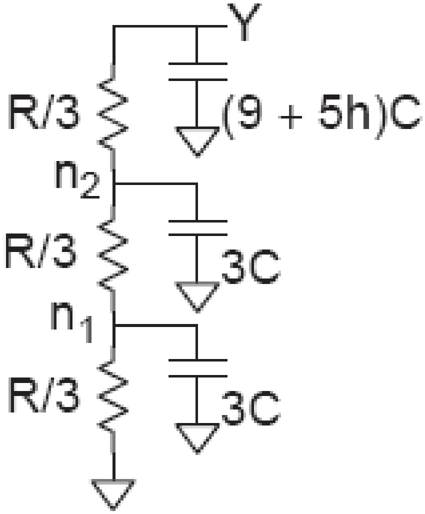
3-input NAND: pull-down delay

- Estimate worst-case rising and falling delay of 3-input NAND driving *h* identical gates.



h copies





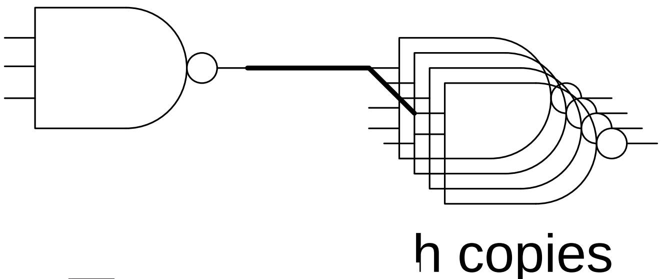
Worst case pull-down delay occurs when ABC goes from (110) to (111)

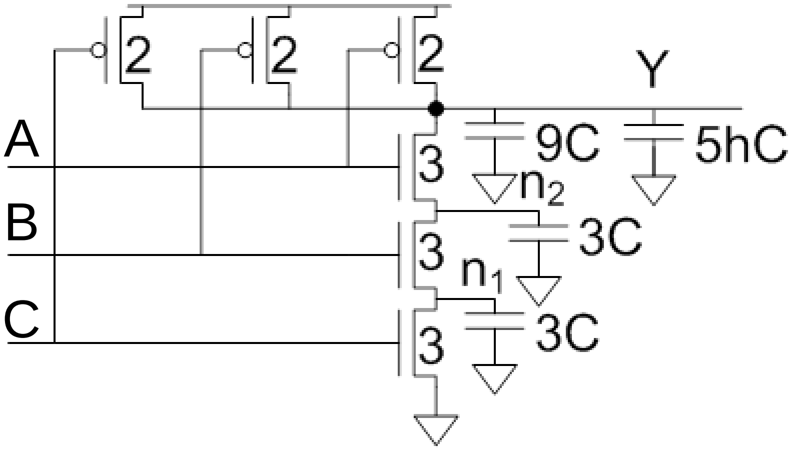
$$t_{pdf} = (3C)\left(\frac{R}{3}\right) + (3\text{C})\left(\frac{R}{3} + \frac{R}{3}\right) + [(9 + 5h)C]\left(\frac{R}{3} + \frac{R}{3} + \frac{R}{3}\right)$$

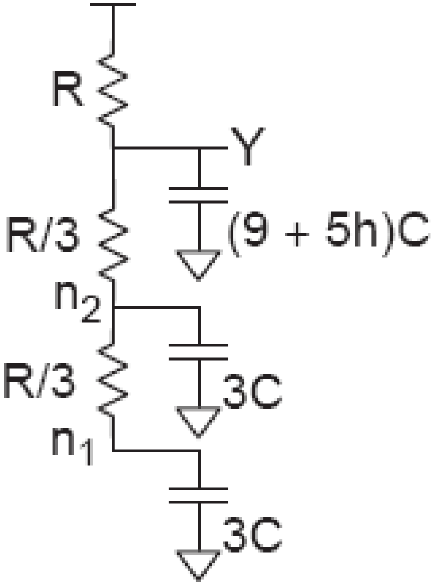
$$t_{pdf} = (12 + 5h)RC$$

3-input NAND: pull-up delay

- Estimate worst-case rising and falling delay of 3-input NAND driving *h* identical gates.







Worst case pull-up delay occurs when ABC goes from (111) to (110)

$$t_{pdr} = [(9+5h)\mathrm{C}](\mathrm{R}) + (3\mathrm{C})(\mathrm{R}) + (3\mathrm{C})(\mathrm{R})$$

$$t_{pdr} = (15+5h)RC$$

Delay Components

$$t_{pdf} = (12 + 5h)RC \qquad t_{pdr} = (15 + 5h)RC$$

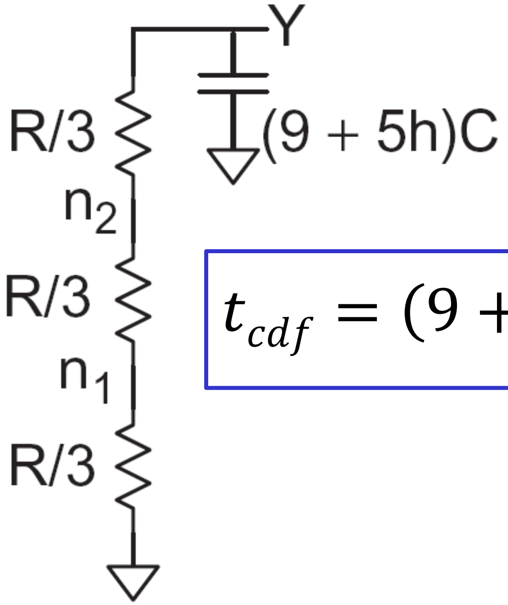
- Delay has two parts
  - *Parasitic delay*
    - 15 or 12 RC
    - Independent of load
  - *Effort delay*
    - 5h RC
    - Proportional to load capacitance

Contamination Delay

- Best-case (contamination) delay can be substantially less than propagation delay:

if top nMOS is last to turn on:

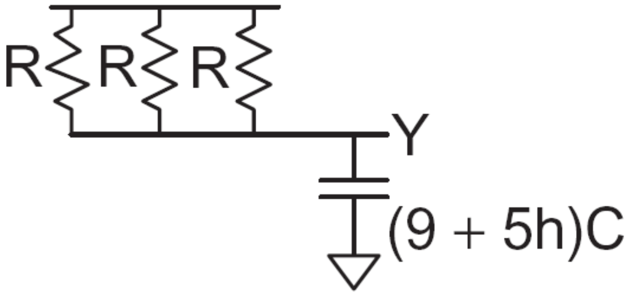
i.e. ABC goes from (011) to (111)



$$\boxed{t_{cdf} = (9 + 5h)RC}$$

if all pMOS turn on simultaneously:

i.e. ABC goes from (111) to (000)



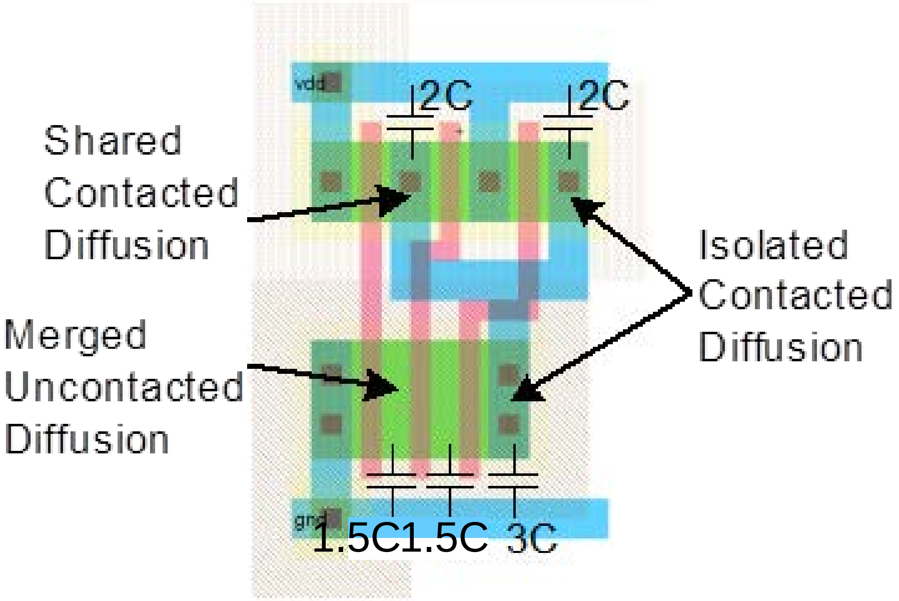
$$\boxed{t_{cdr} = \left(3 + \tfrac{5}{3}h\right)RC}$$

compare to: $t_{pdf} = (12 + 5h)RC$

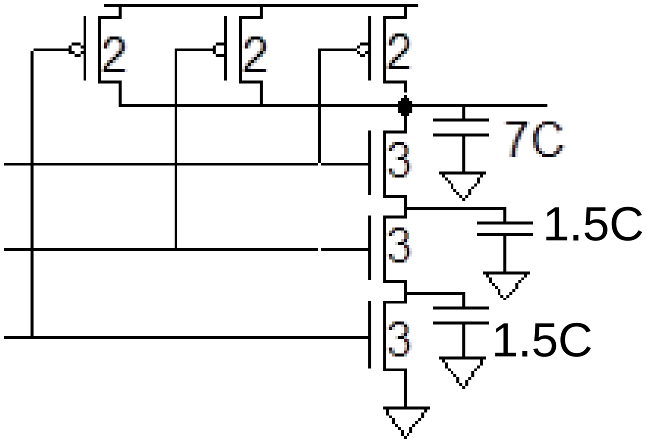
$t_{pdr} = (15 + 5h)RC$

Diffusion Capacitance

- We assumed contacted diffusion on every s / d.
  - but shared on series nMOS chain
- Good layout minimizes diffusion area
- Good NAND3 layout shares one diffusion contact
  - Reduces output capacitance by 2C
- Merged un-contacted diffusion also helps

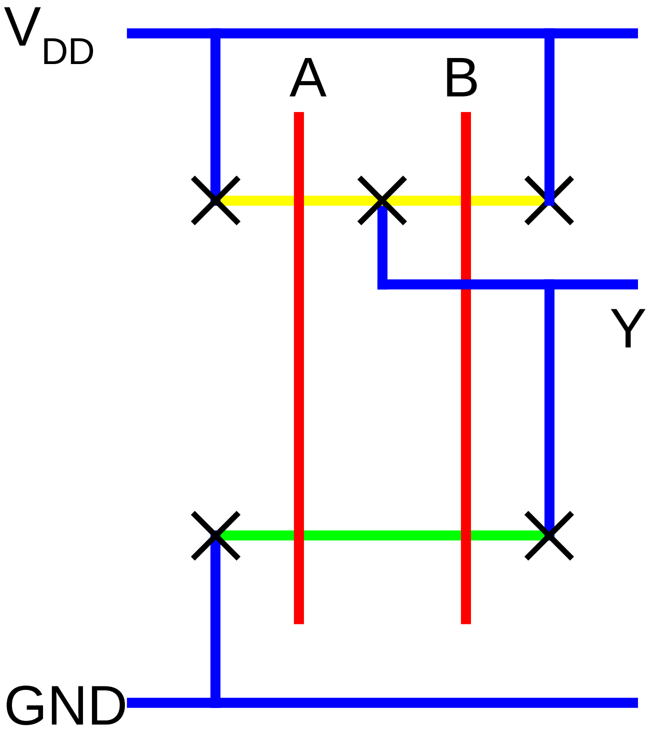


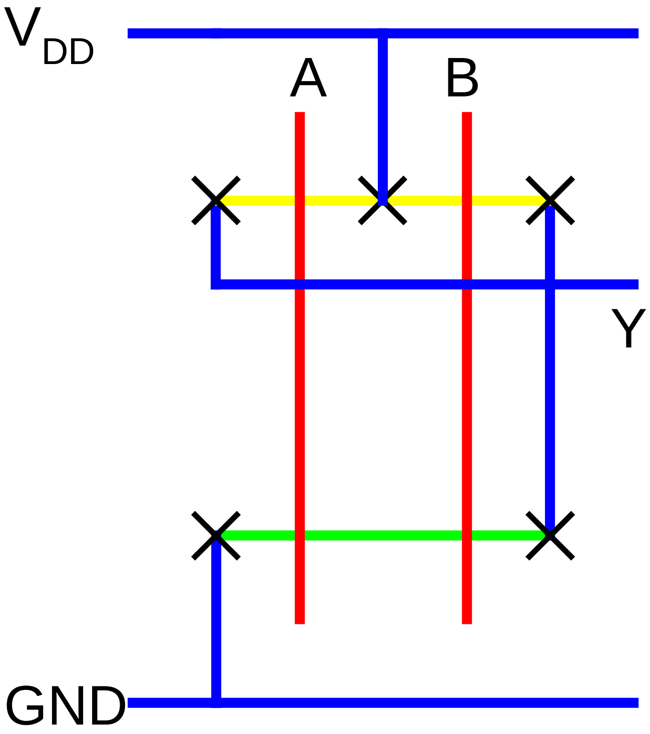
vdd
2C
2C
Shared Contacted Diffusion
Isolated Contacted Diffusion
Merged Uncontacted Diffusion
gnd
1.5C
1.5C
3C



Layout Comparison

- Which layout is better?





Example: Gate delays

For the gate $Y = \overline{A + B.C}$

a) Draw the schematic

b) Size the transistors to give pullup and pulldown strength equal to unit size inverter

c) Annotate with effective R of each transistor and C of each node

d) Calculate worst case rising & falling propagation delay while driving *h* similar gates (via input B)

e) Calculate best case rising & falling contamination delay while driving *h* similar gates (via input B)