EE 471: Transport Phenomena in Solid State Devices

Spring 2018

Lecture 12

CMOS Delay & Transient Response

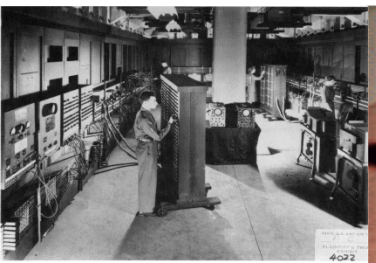
Bryan Ackland

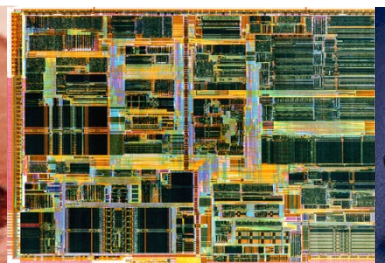
Department of Electrical and Computer Engineering

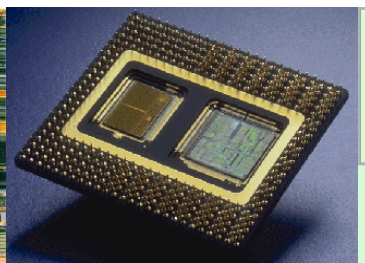
Stevens Institute of Technology

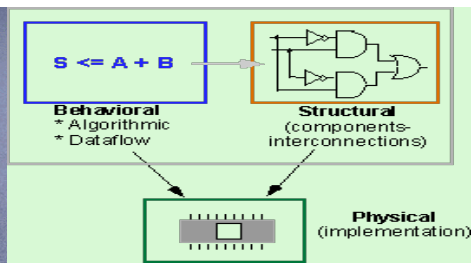
Hoboken, NJ 07030

Adapted from Lecture Notes, David Mahoney Harris CMOS VLSI Design



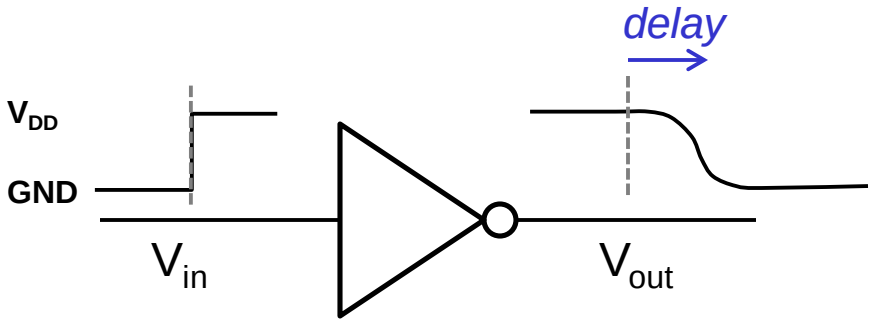






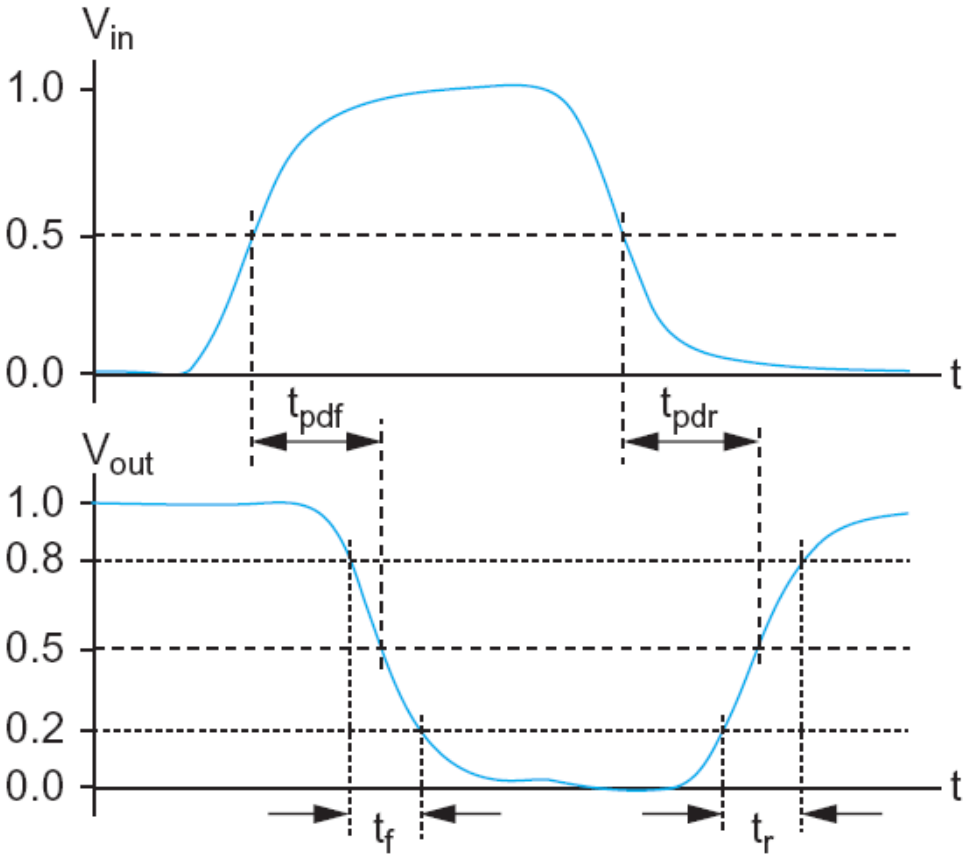
Transient Response

- 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



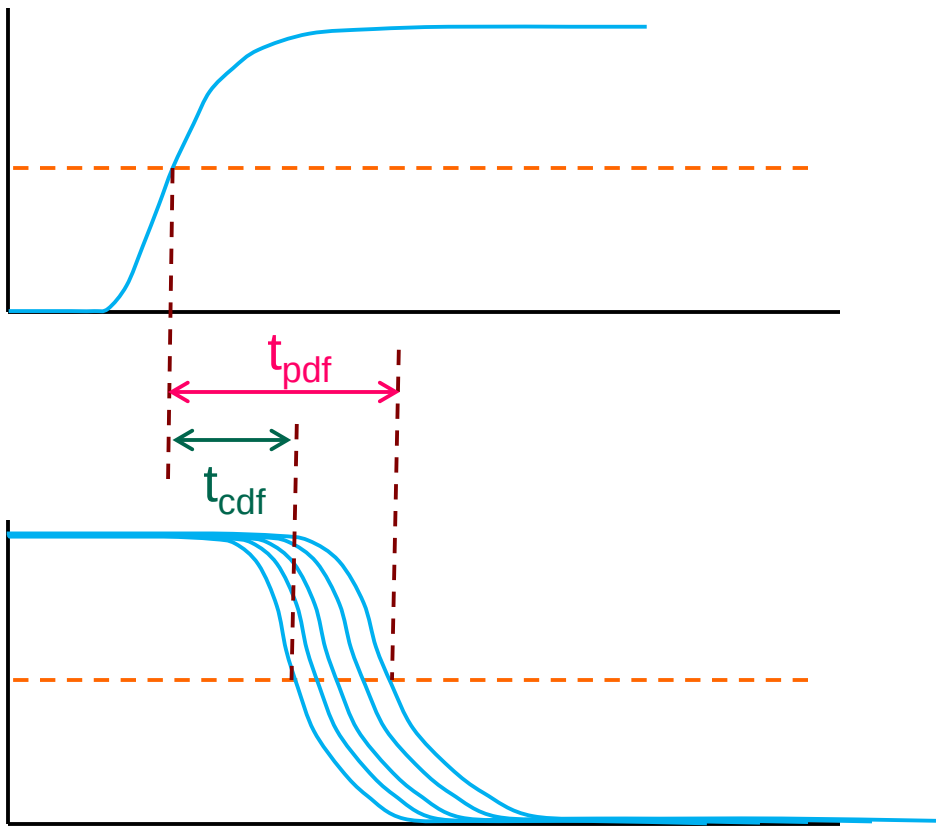
Delay Definitions

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



Delay Definitions (cont.)

- $\mathbf{t_{cdf}}$: *falling contamination delay*
  - minimum time from input crossing $V_{DD}/2$ to falling output crossing $V_{DD}/2$
- $\mathbf{t_{cdr}}$: *rising contamination delay*
  - minimum time from input crossing $V_{DD}/2$ to rising output crossing $V_{DD}/2$
- $\mathbf{t_{cd}}$: *avg. contamination delay*
  - $t_{pd} = (t_{cdr} + t_{cdf})/2$

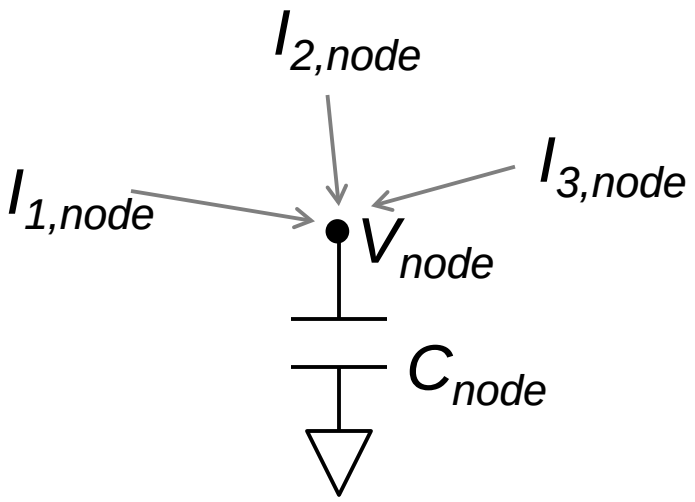


Delay in CMOS Circuits

- A 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



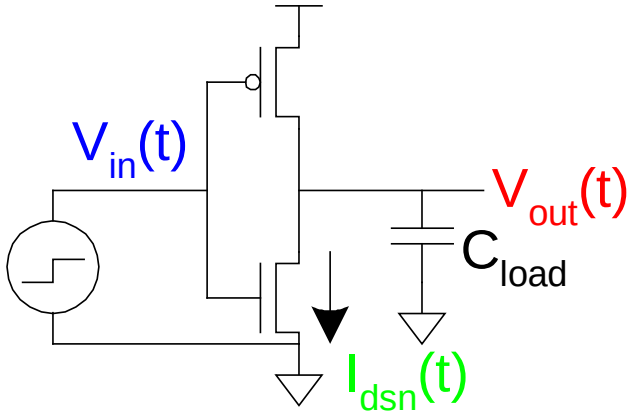
Inverter Step Response

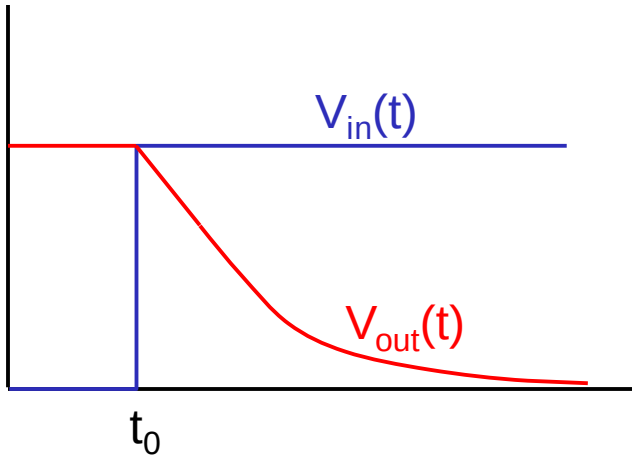
- Find step response of inverter driving C_{load}

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

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

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

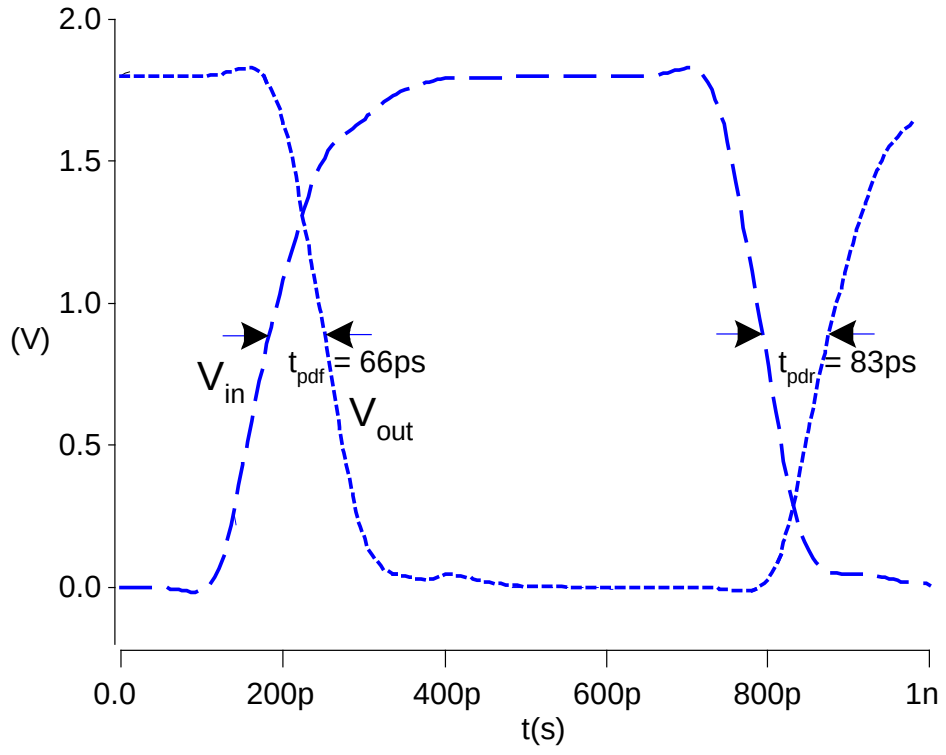




$$I_{dsn}(t) = \begin{cases} 0 & \text{for } t < t_0 \\ (\beta/2m) \bullet (V_{DD} - V_t)^2 & \text{for } V_{out} > V_{DD} - V_t \\ \beta \bullet (V_{DD} - V_t - V_{out}(t)/2m) \bullet V_{out}(t) & \text{for } Vout < V_{DD} - V_t \end{cases}$$

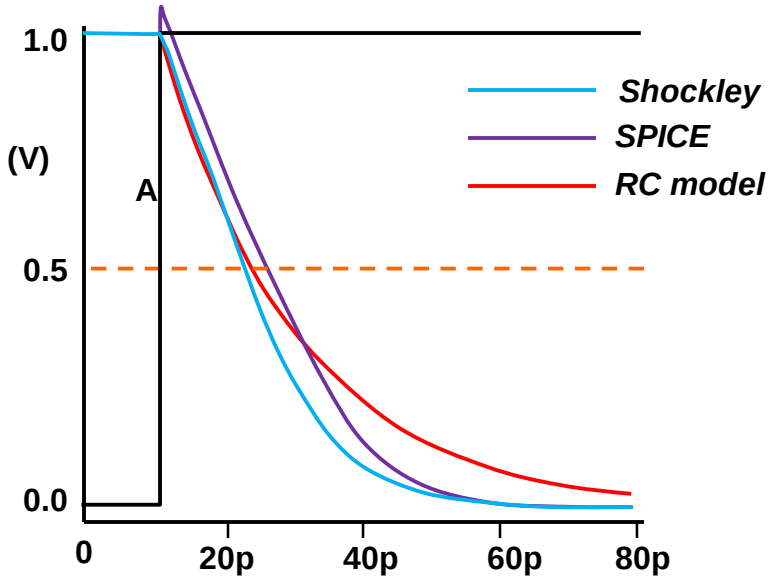
Simulated Inverter Delay

- 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

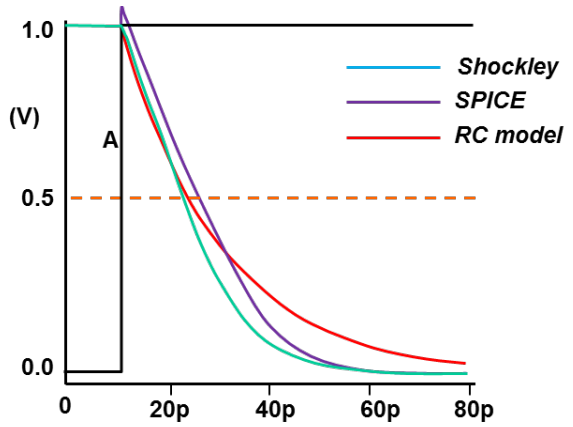
- We would like to be able to easily estimate delay
  - For exploration of design space, don't need to be as accurate as simulation
  - Want a technique where its easier to ask "What if?"
- The step response usually looks like a 1st order RC response with a decaying exponential.
- Can we model conducting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

- Pick R to best model dynamic response of gate
  - Too inaccurate to predict current at any given time
  - But good enough to predict gate delay

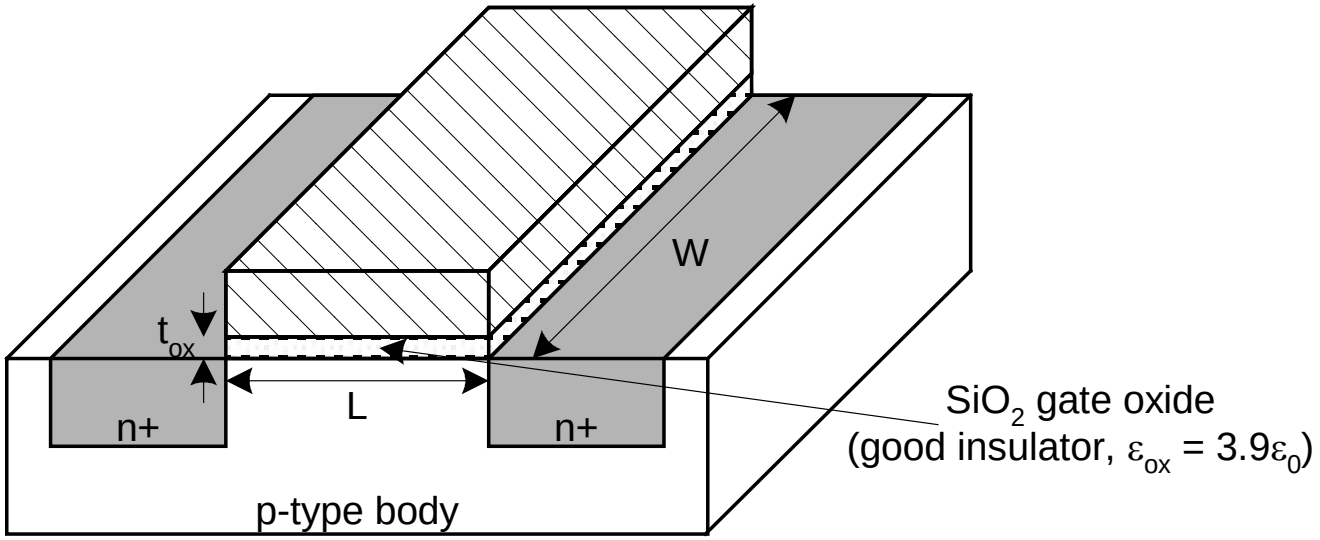


Capacitance

- Input to CMOS gate presents effectively infinite input resistance
- The dominant load in CMOS circuits is capacitance
- Capacitance exists wherever there are two conductors separated by a thin insulator
- Gate to channel capacitor is very important
  - Creates channel charge necessary for operation
- Source and drain have capacitance to body
  - Parasitic capacitance across reverse-biased diode depletion region
  - Called diffusion capacitance because it is associated with source/drain diffusion
- Long interconnect wires also have parasitic capacitance to the substrate

Gate Capacitance

- Gate is top plate of capacitor
- Assume bottom plate is source
  - In cut-off, bottom plate is actually the body
  - In linear mode, bottom plate is channel which is connected to source and drain
  - In saturation, bottom plate is channel connected to source
- $C_g \approx \varepsilon_{ox}.W.L/t_{oxe} = C_{oxe}.W.L = C_{permicron}.W$ *(for $L = L_{min}$)*
- $C_{permicron}$ is typically about 1-2 fF/μm of width

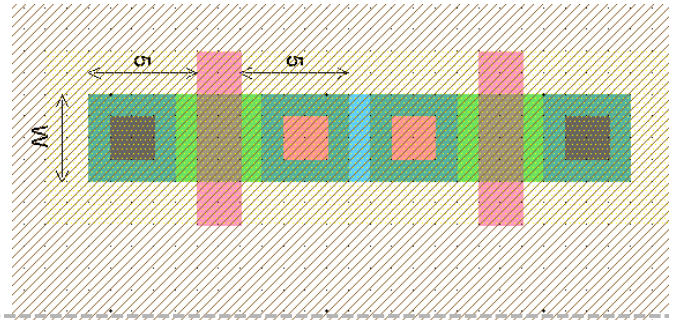


Diffusion Capacitance

- C_{sb}, C_{db}
- Diffusion (source/drain) region is resistive and capacitive (to body)
- Capacitance depends on area and perimeter
- Use small as possible diffusion nodes
- Comparable to C_g for min. contacted diffusion
- Use $C_g/2$ for merged
- Varies with process

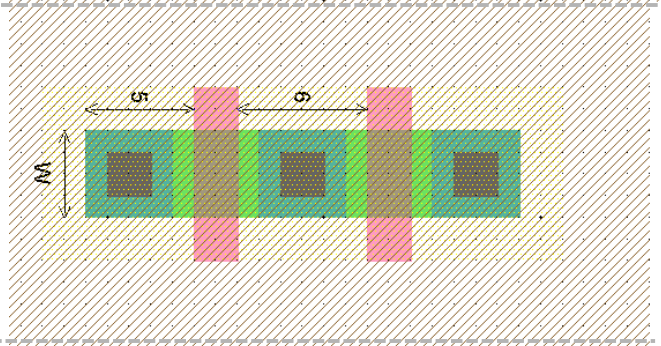
Isolated Diffusion
≈ C_g

$C_{node} = 2.C_g$



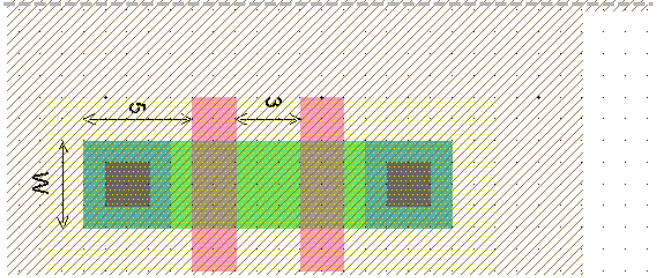
Shared Diffusion
≈ C_g

$C_{node} = C_g$



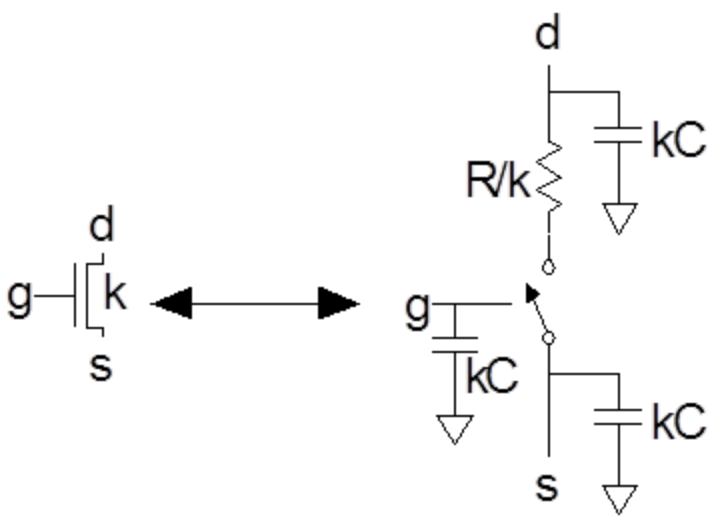
Merged Diffusion
≈ $C_g/2$

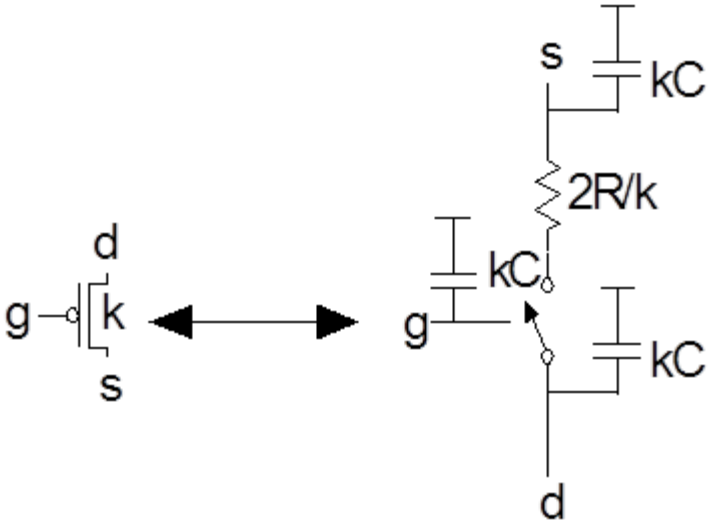
$C_{node} = C_g/2$



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





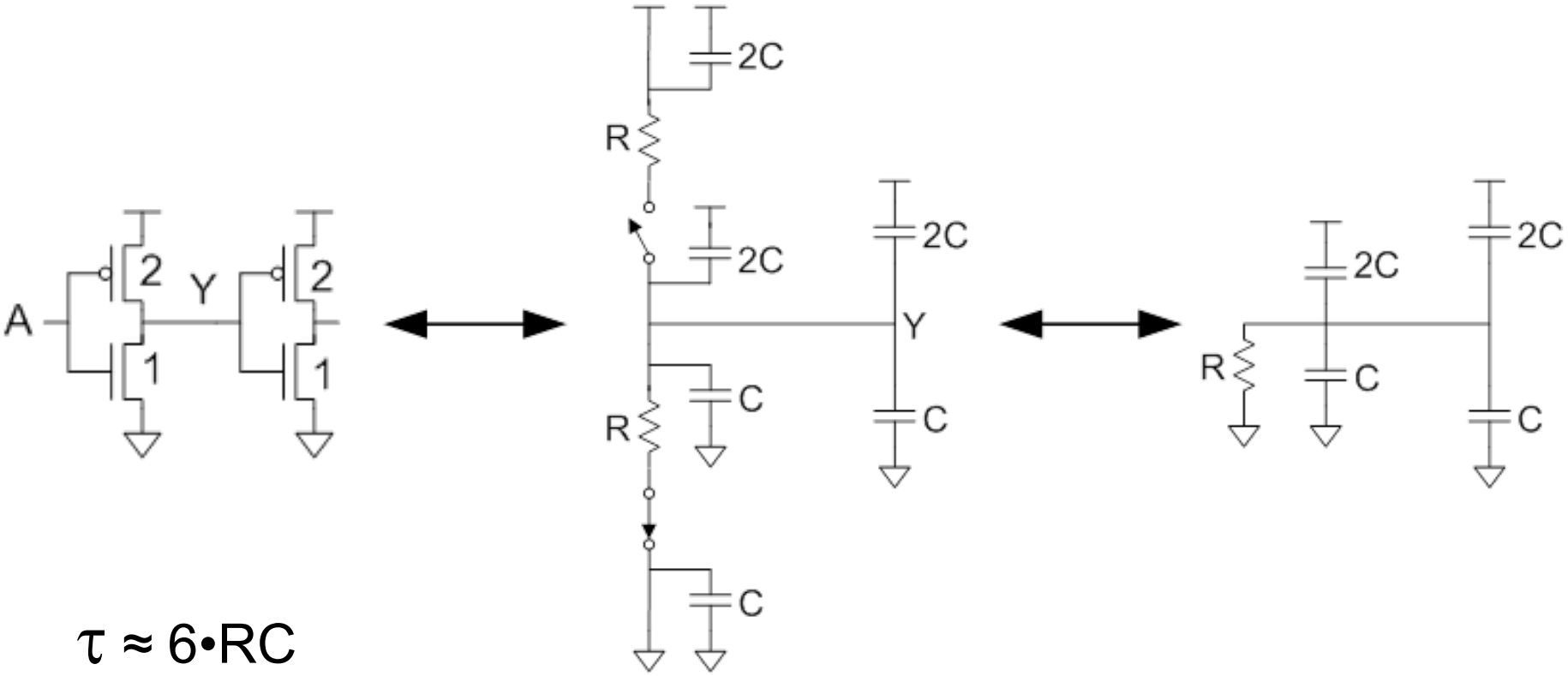
RC Values

- 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
  - $R \approx$ 5-10 KΩ•μm in 0.6 μm process
  - Improves with shorter channel lengths
- Unit transistors
  - May refer to minimum contacted device (4 λ / 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

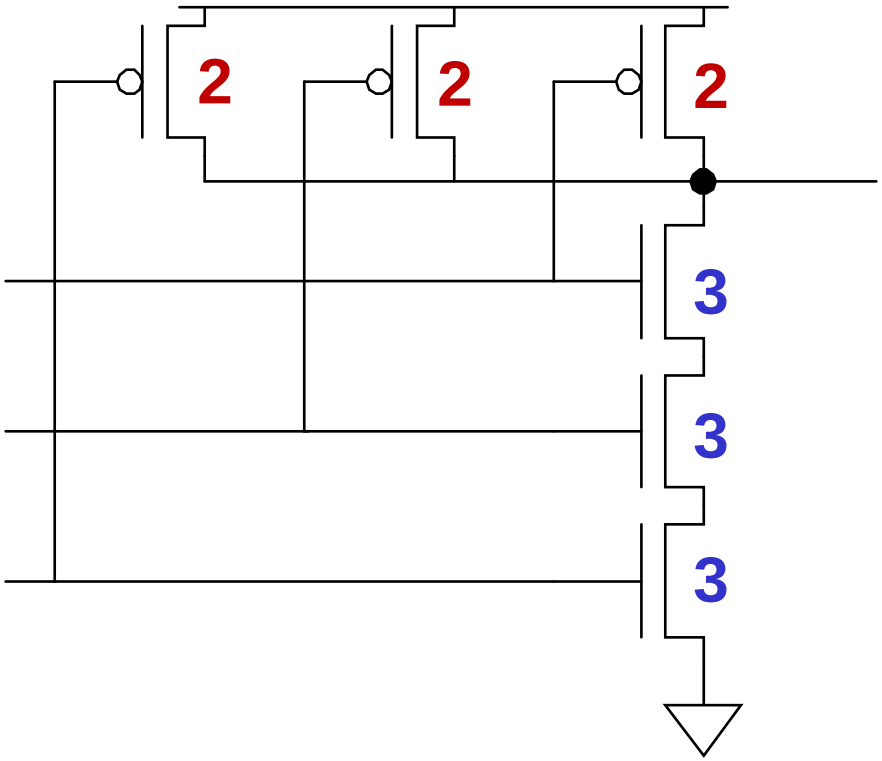
- Estimate the delay of a fanout-of-1 inverter
- Set size (width) of PMOS to 2 x unit size to have equal pull-up (rising) and pull-down (falling) drive resistance



$\tau \approx 6 \cdot RC$

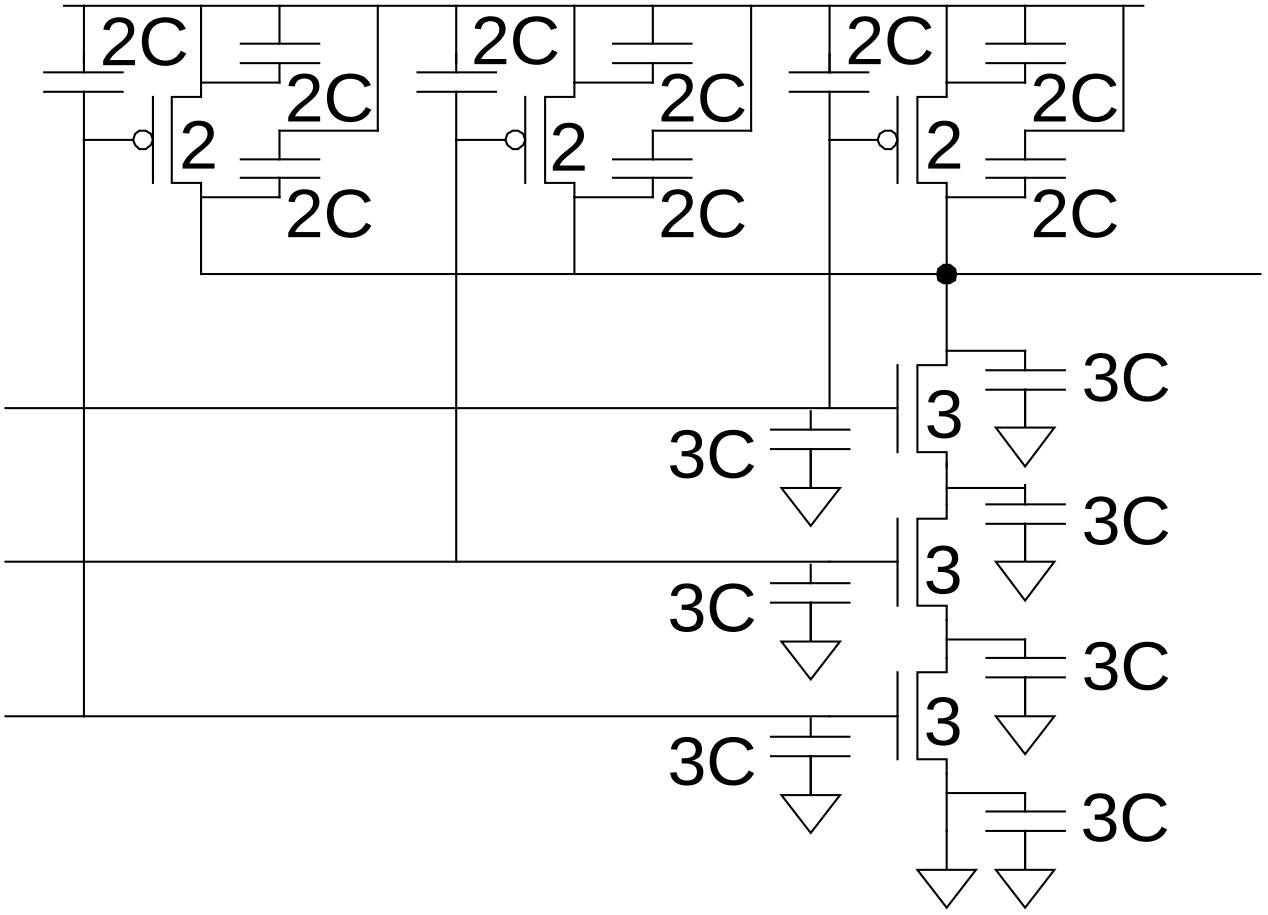
Example: 3-input NAND

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



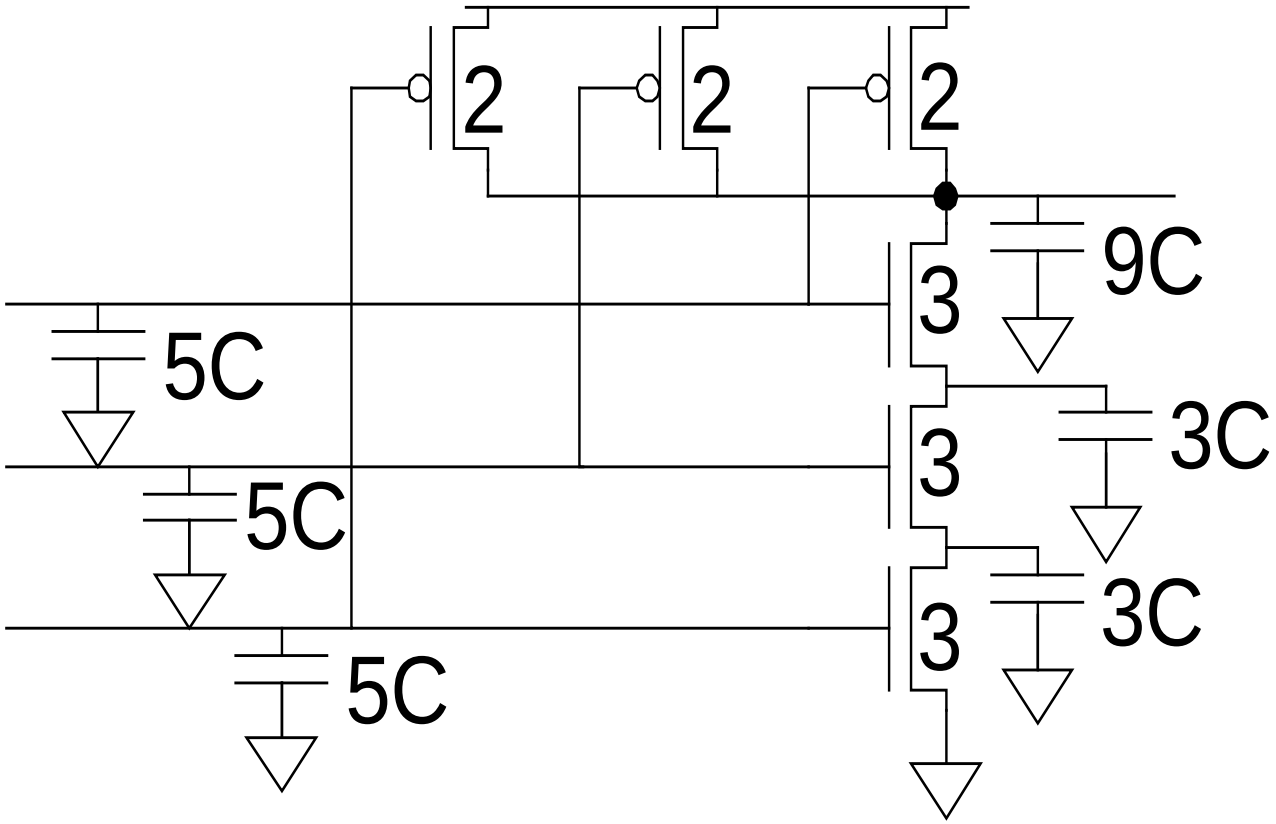
3-input NAND Capacitors

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



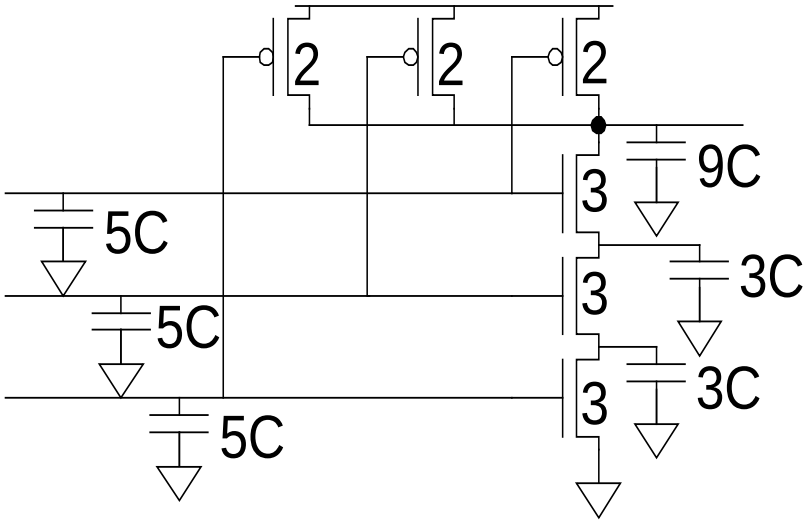
3-input NAND Capacitors

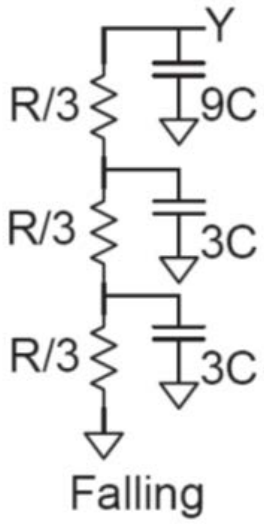
- Annotate the 3-input NAND gate with gate and diffusion capacitance.

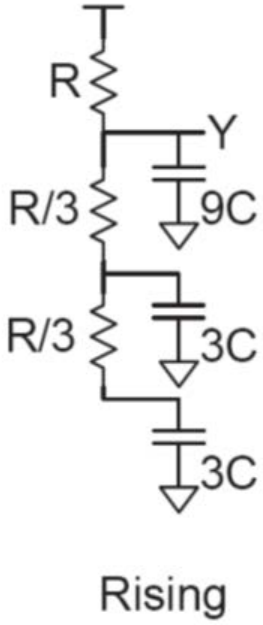


Rise & Fall Delay

- What are worst-case rise and fall delays?

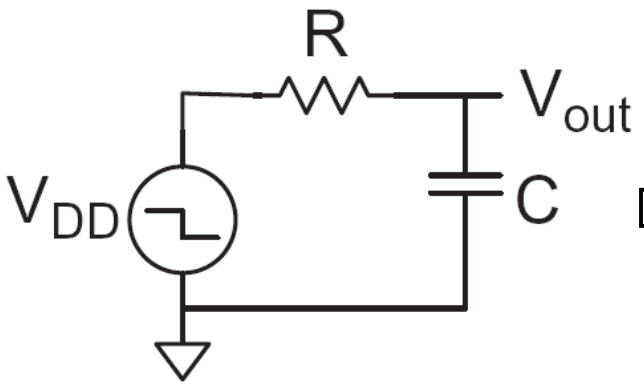




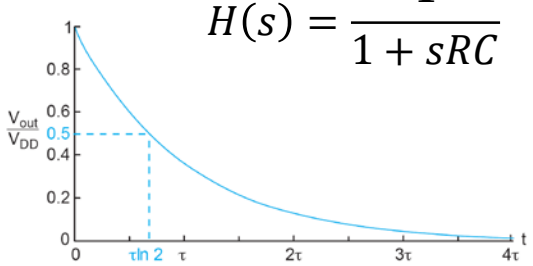


- How can we estimate delay of these networks?

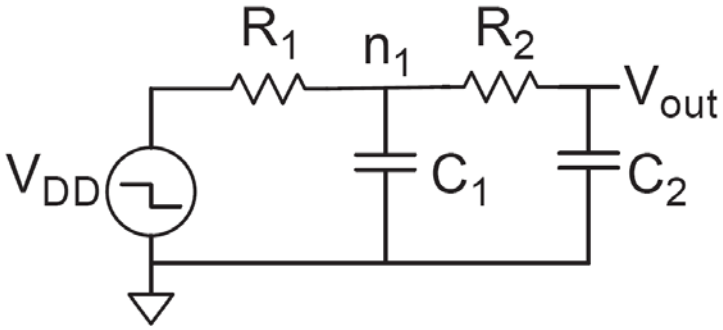
τ with multiple RC components



$$H(s) = \frac{1}{1 + sRC}$$



$\tau = RC$



$\tau = ?$

$$H(s) = \frac{1}{1 + s[R_1C_1 + (R_1 + R_2).C2] + s^2R_1C_1R_2C_2}$$

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

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

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)
  - resistors are branches
  - leaves are capacitors at ends of branches
- Elmore delay to any target (node j) in the branch:

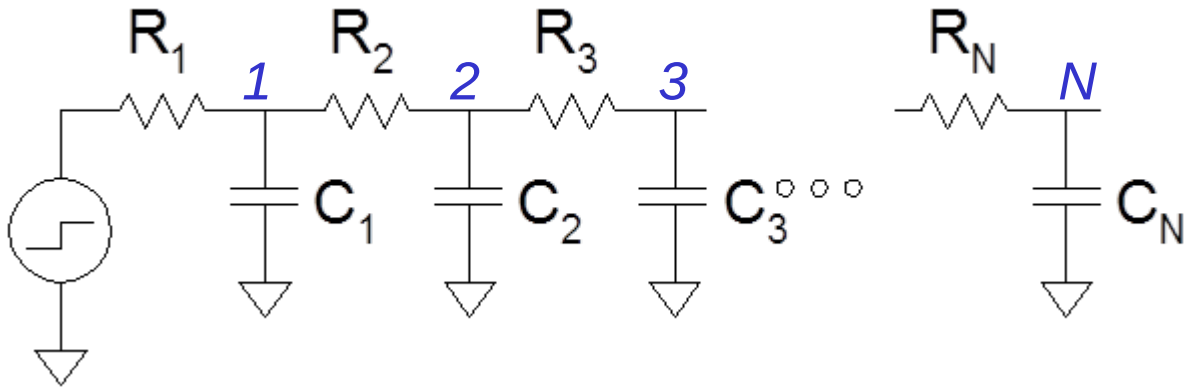
$$t_{pdj} = \sum_{i} R_{sij}.C_i$$

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:

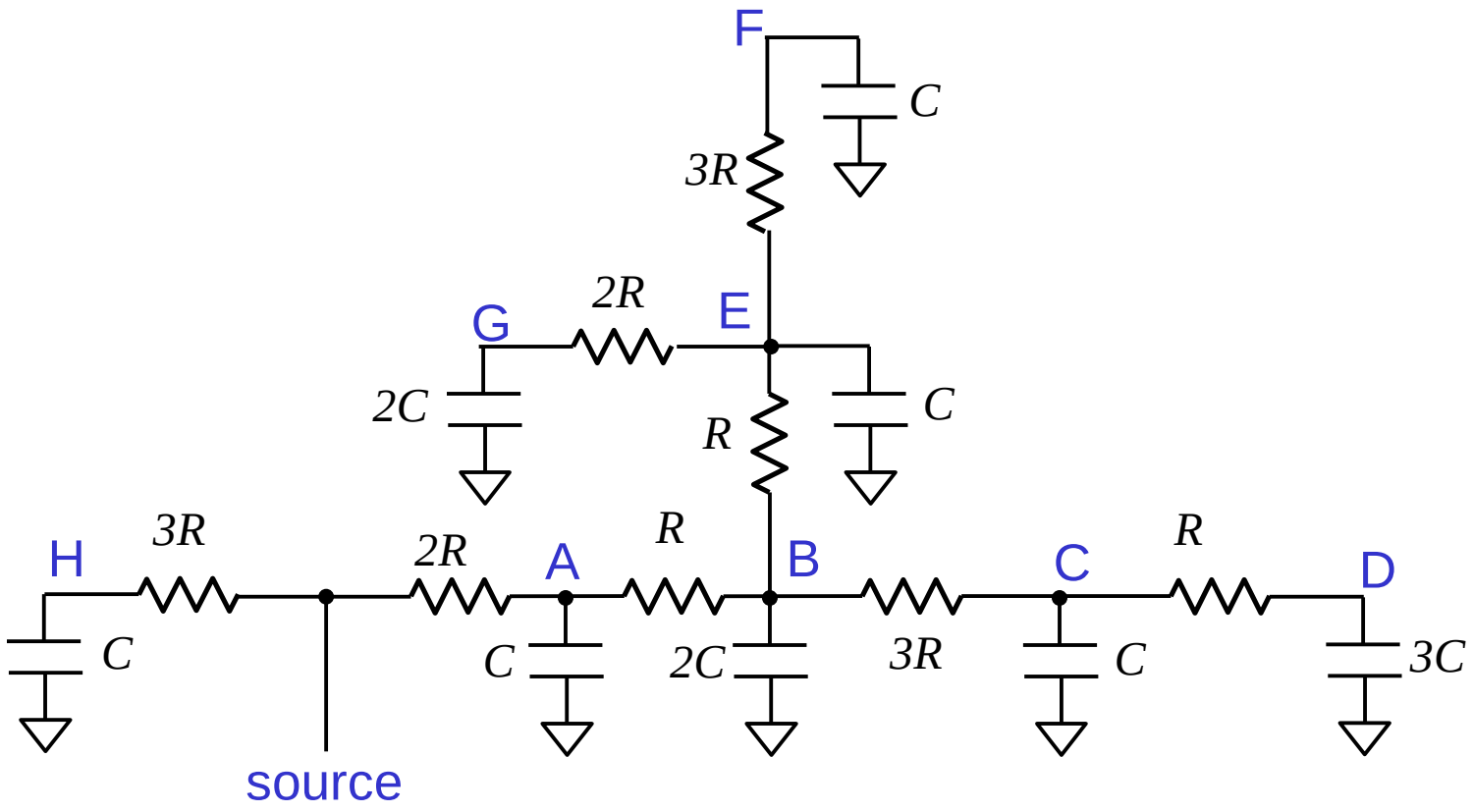
$$R_1C_1 + (R_1+R_2).C_2 + \ldots + (R_1+R_2+\ldots+R_n).C_N$$

- delay to node 2 is:

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

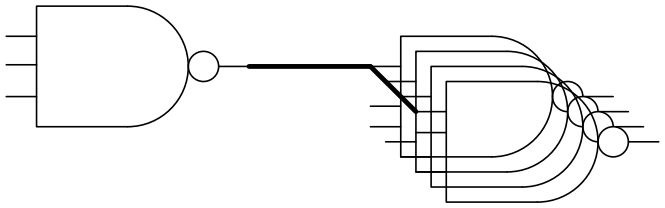
Example: Elmore Delay

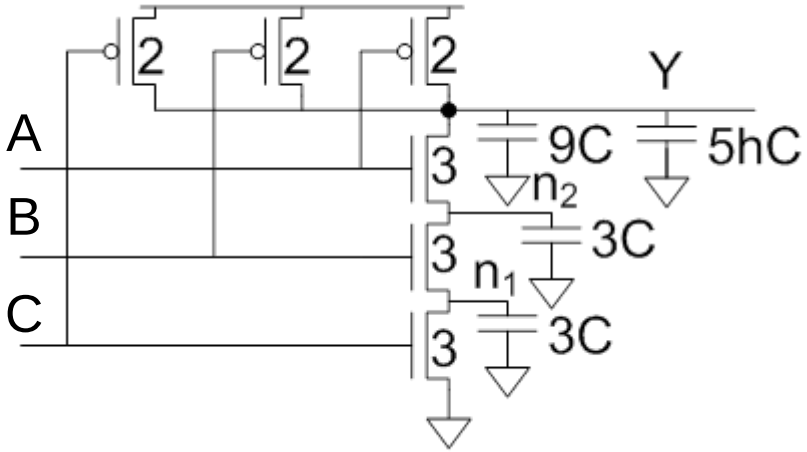
- Calculate delay from source to all nodes in circuit:

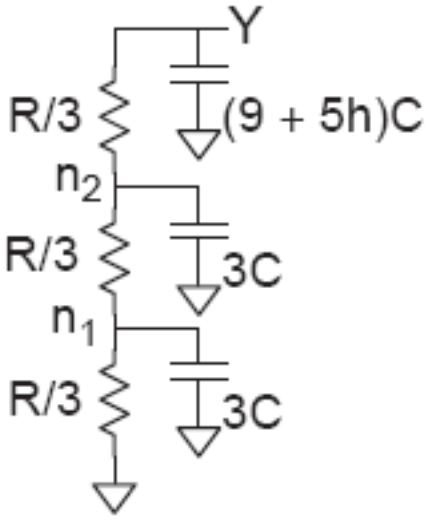


3-input NAND: pull-down delay

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







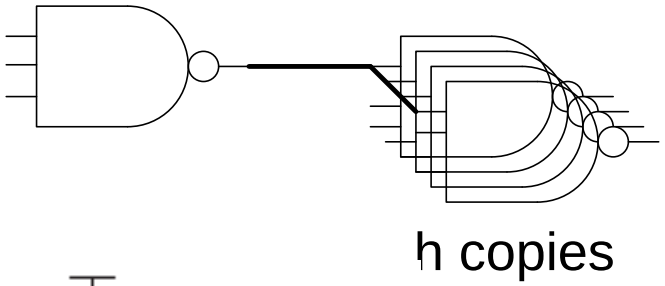
Worst case pull-down delay occurs when ABC goes from (110) to (111)

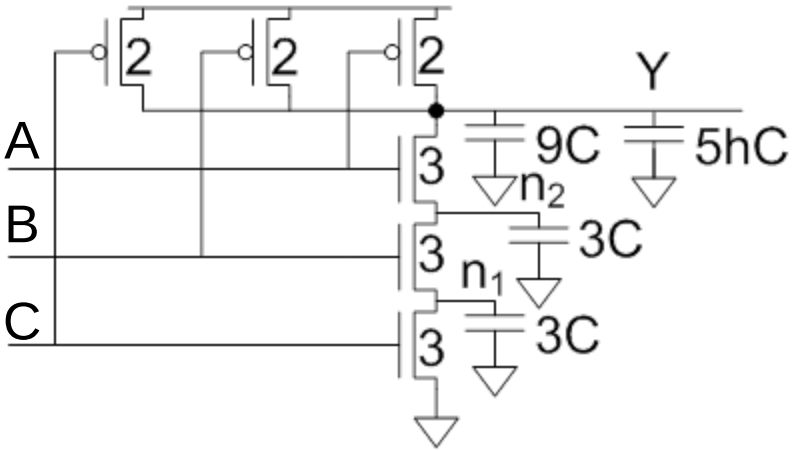
$$t_{pdf} = (3C)\left(\frac{R}{3}\right) + (3C)\left(\frac{R}{3}+\frac{R}{3}\right) + [(9+5h)C]\left(\frac{R}{3}+\frac{R}{3}+\frac{R}{3}\right)$$

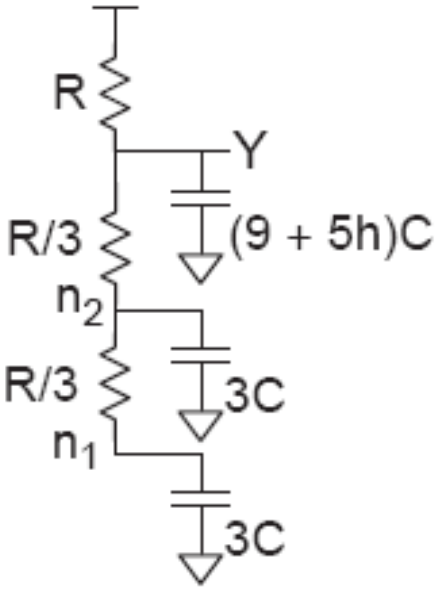
$$t_{pdf} = (12+5h)RC$$

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

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

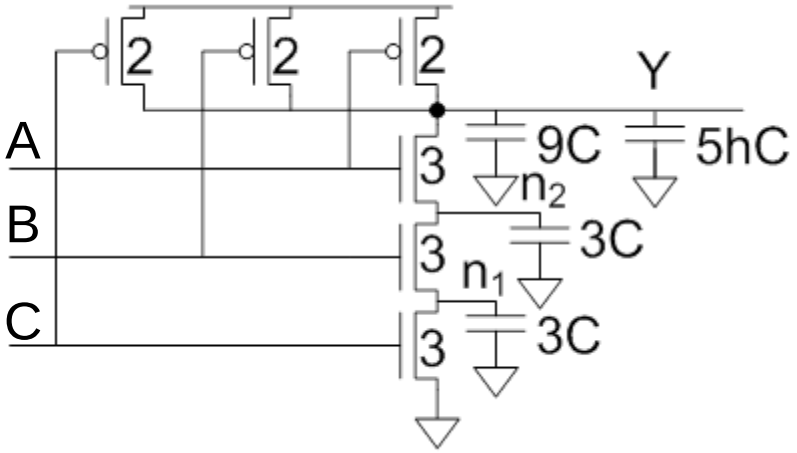
- Delay has two parts
  - *Parasitic delay*
    - 15 or 12 RC
    - Independent of load
  - *Effort delay*
    - 5h RC
    - Proportional to load capacitance

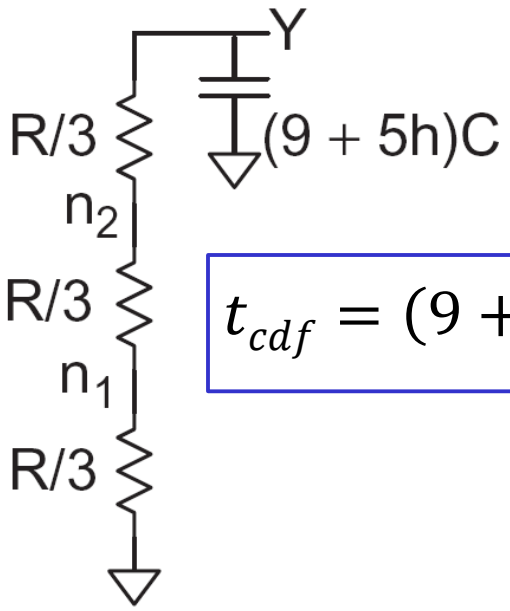
Falling Contamination Delay

- 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)





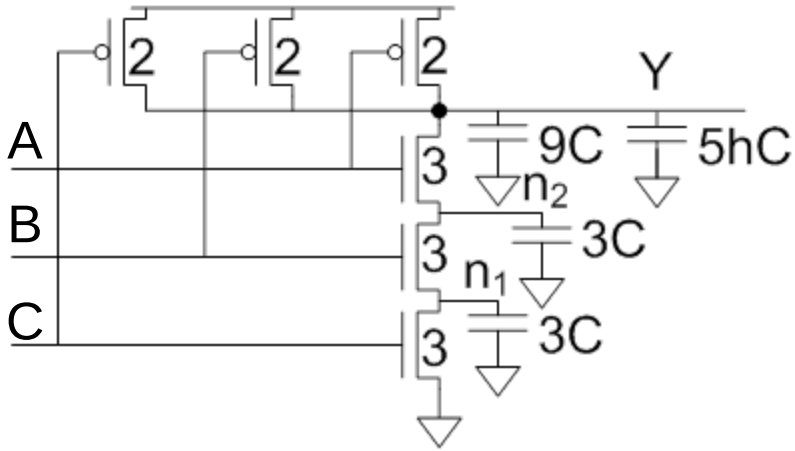
$$t_{cdf} = (9 + 5h)RC$$

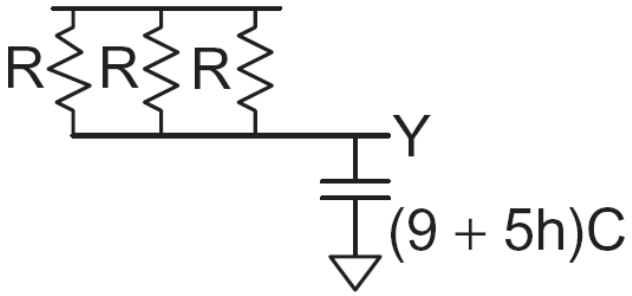
compare to: $t_{pdf} = (12 + 5h)RC$

Rising Contamination Delay

Fastest response if all pMOS turn on simultaneously:

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



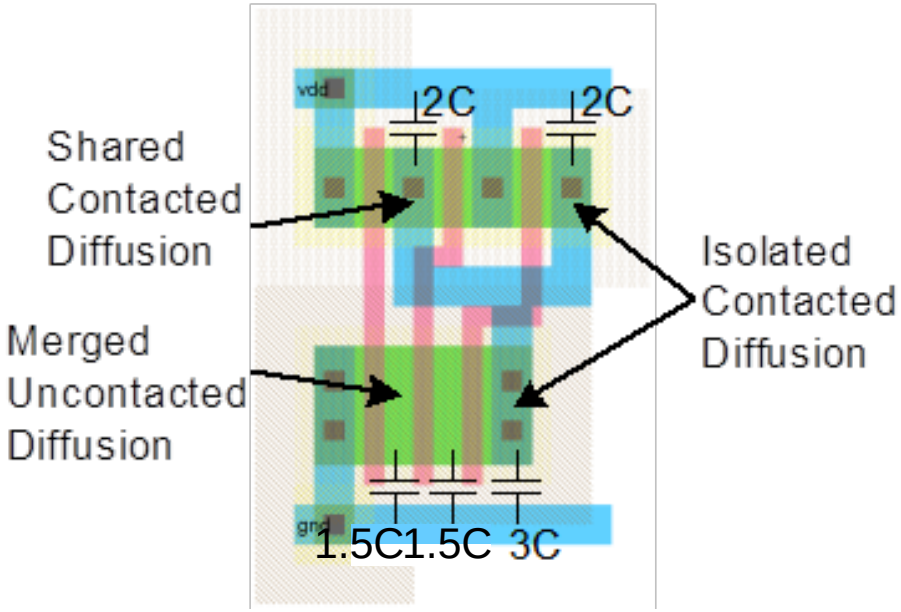


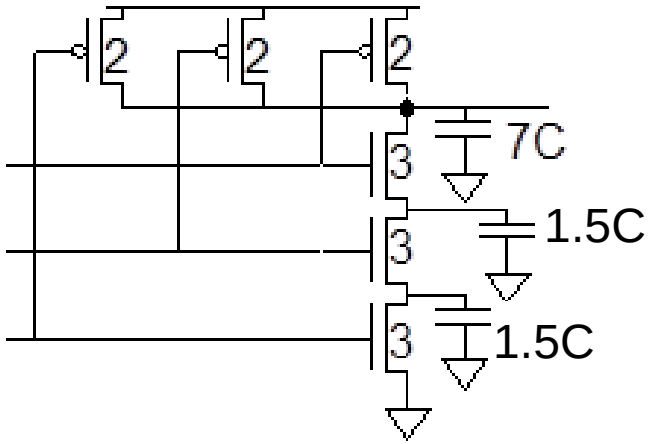
$$t_{cdr} = \left(3 + \frac{5}{3}h\right)RC$$

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

Diffusion Capacitance

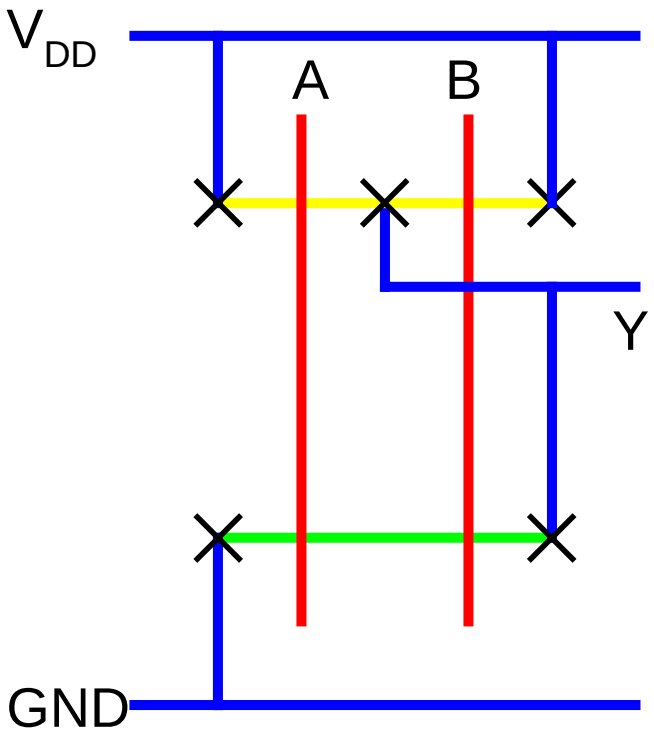
- We assumed contacted diffusion on every s / d.
  - but shared on series nMOS chain
- Good layout minimizes diffusion area
- Good NAND3 layout shares one diffusion contact
  - Reduces output capacitance by 2C
- Merged un-contacted diffusion also helps

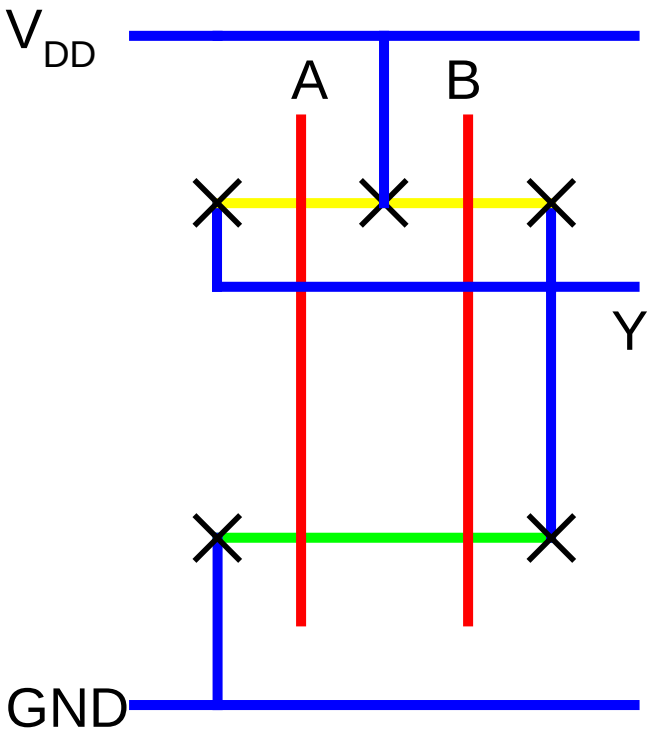




Layout Comparison

- Which layout is better?





Example: Gate delays

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

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

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