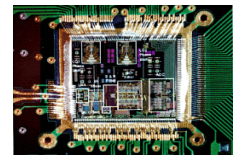


# Lecture 8: Wire and Delay

## CSCE 5730 Digital CMOS VLSI Design

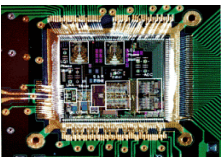
**Instructor:** Saraju P. Mohanty, Ph. D.

**NOTE:** The figures, text etc included in slides are borrowed from various books, websites, authors pages, and other sources for academic purpose only. The instructor does not claim any originality.

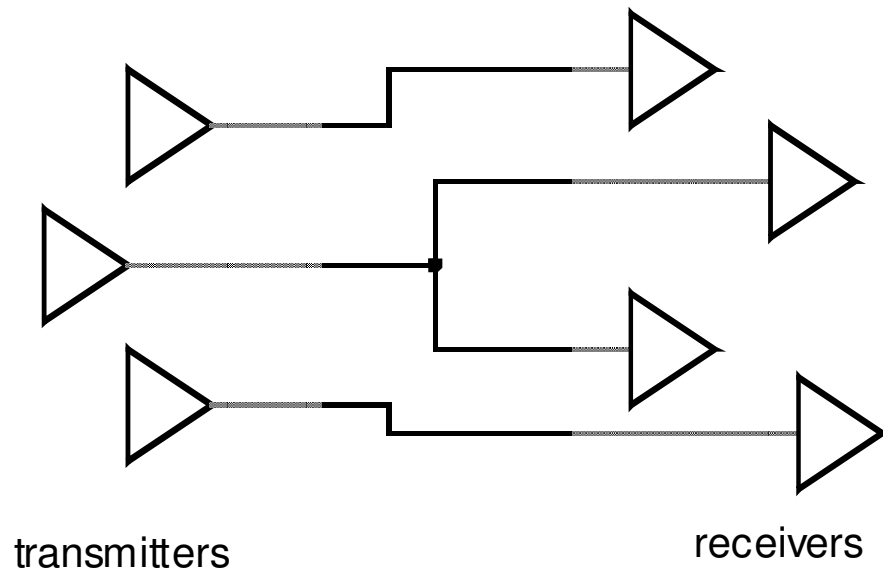


# Outline of the Lecture

- Capacitance of wire
- Resistance of wire
- Inductance of wire
- Elmore Delay Model
- Delay Definitions



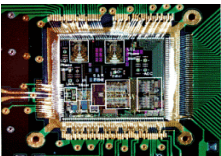
# The Wire



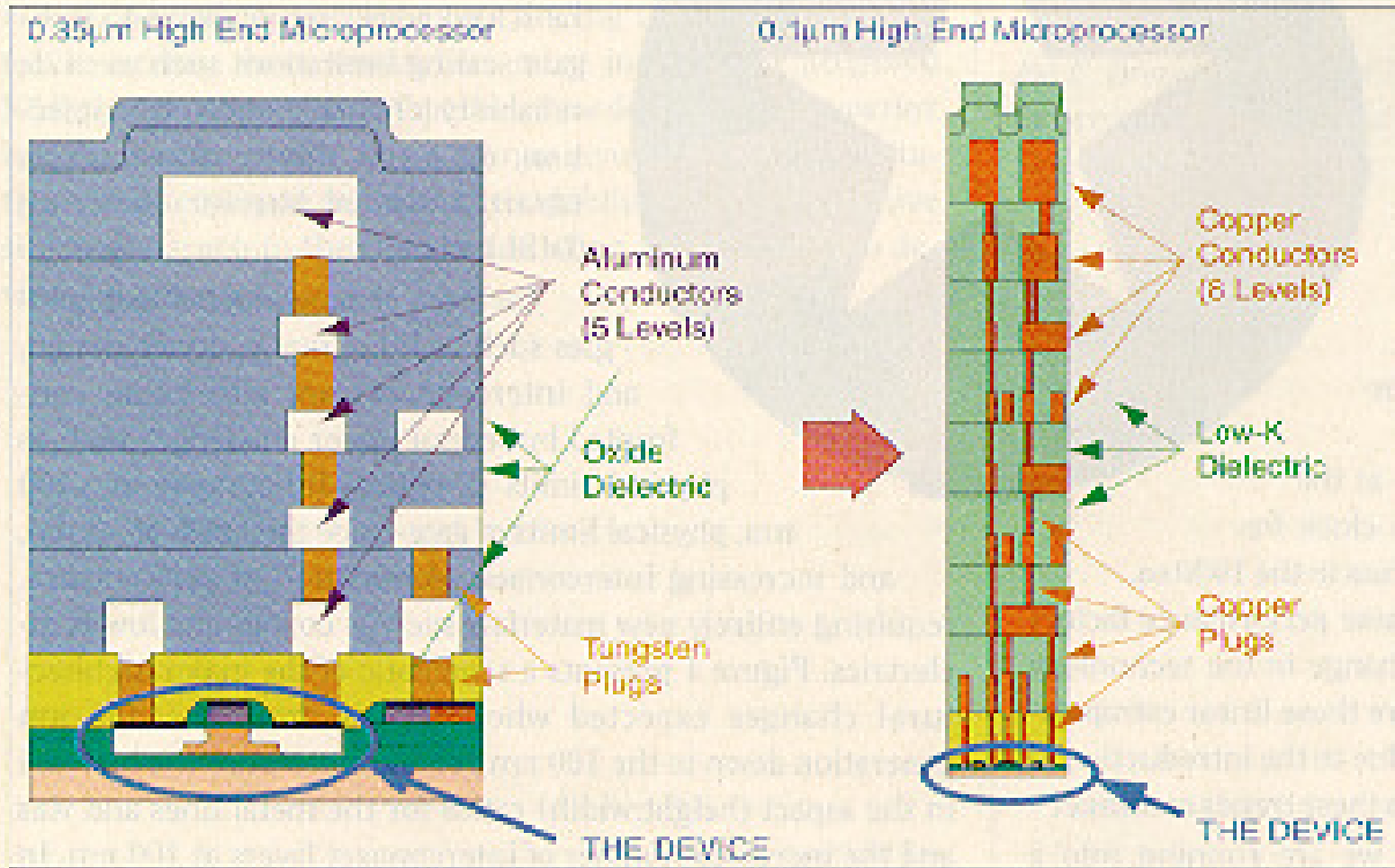
**schematics**



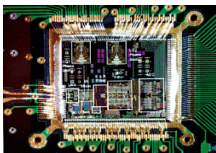
**physical**



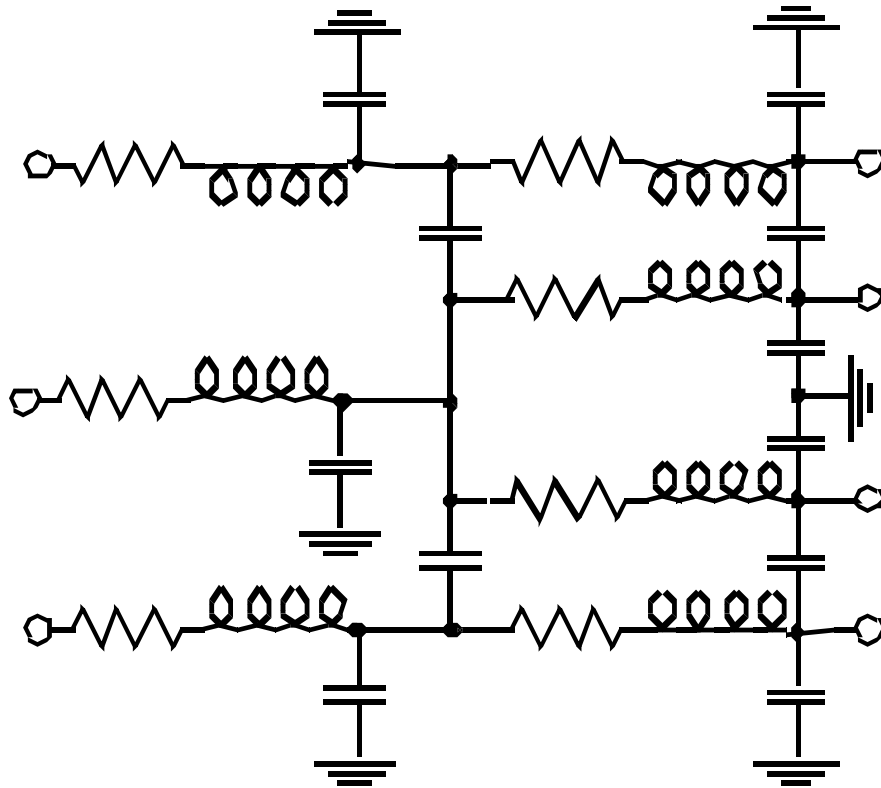
# Interconnect Impact on Chip



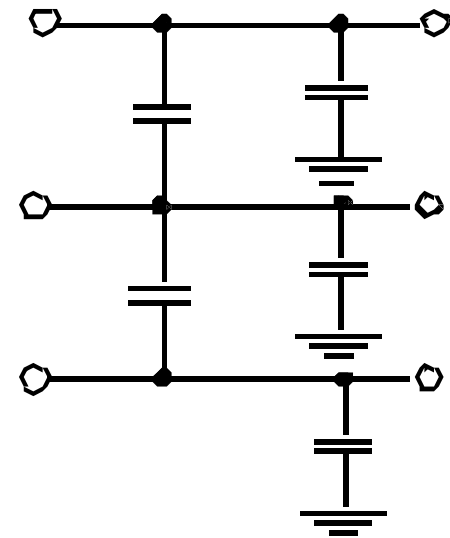
*1. Process architecture challenge.*



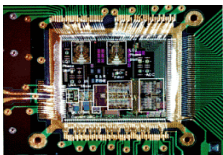
# Wire Models



All-inclusive model

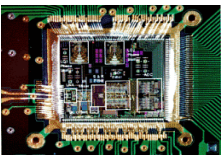
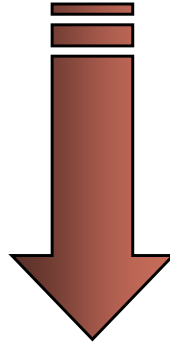


Capacitance-only

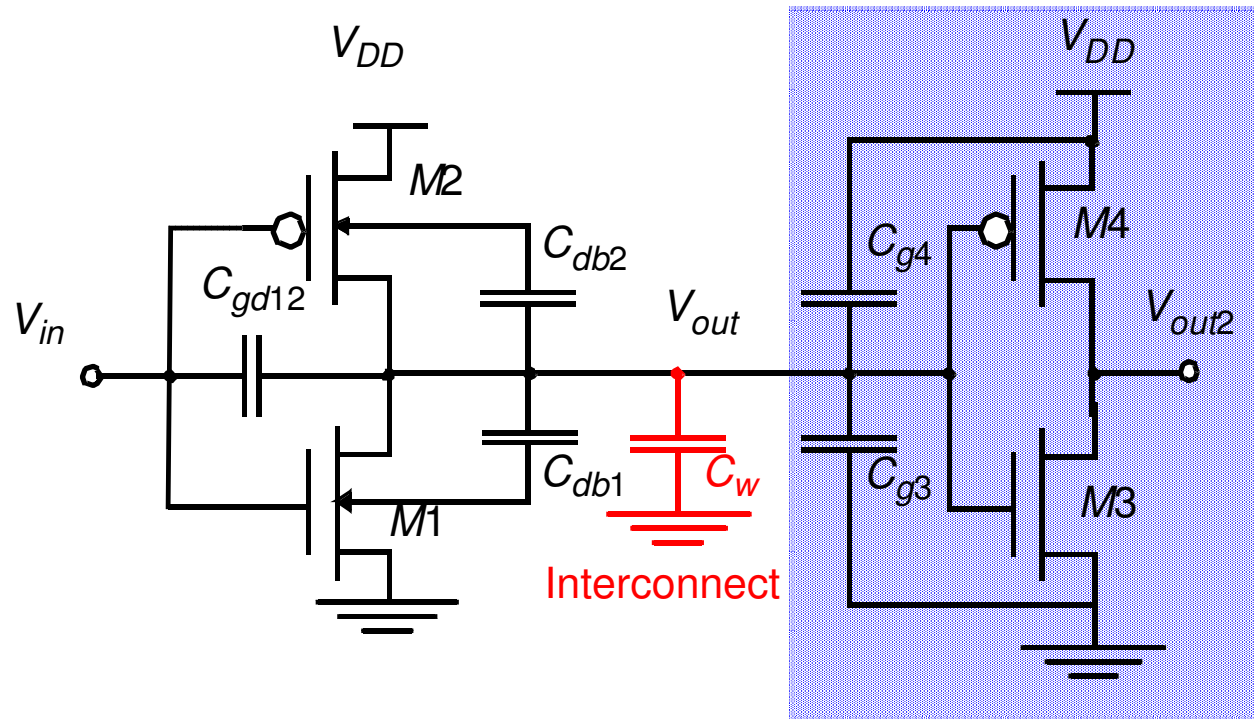


# Impact of Interconnect Parasitics

- Interconnect parasitics
  - reduce reliability
  - affect performance and power consumption
- Classes of parasitics
  - Capacitive
  - Resistive
  - Inductive

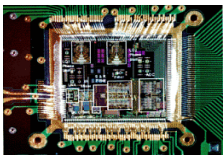
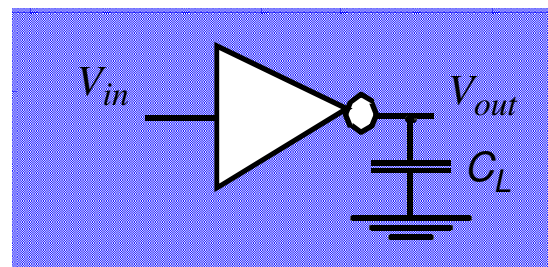


# Capacitance of Wire Interconnect

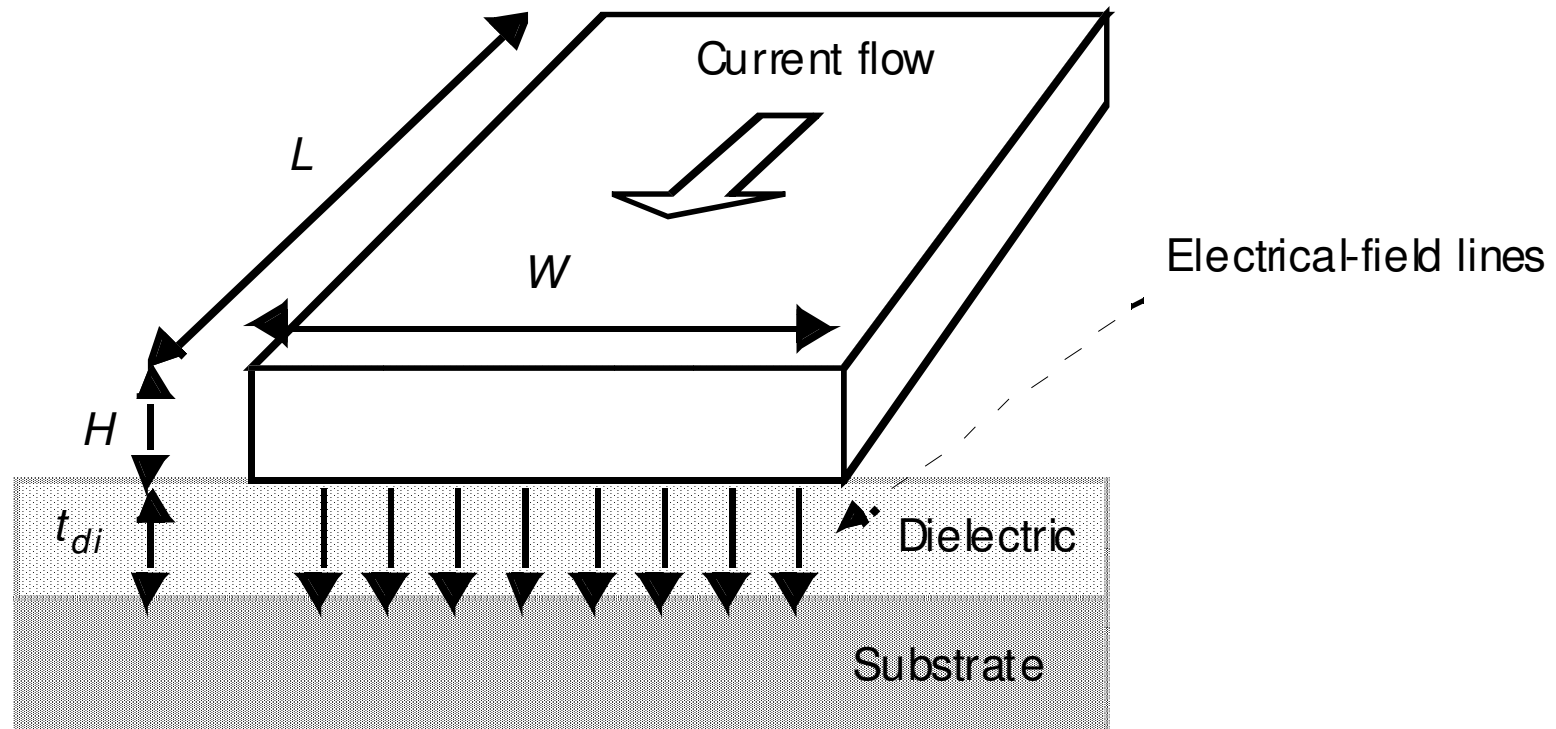


Fanout

**Simplified Model**

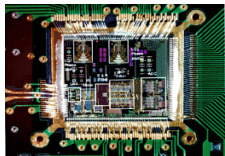


# Capacitance: The Parallel Plate Model



$$C_{int} = \frac{\epsilon_{di}}{t_{di}} WL$$

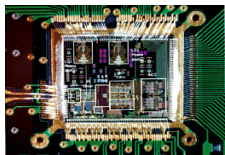
$$S_{Cwire} = \frac{S}{S \cdot S_L} = \frac{1}{S_L}$$



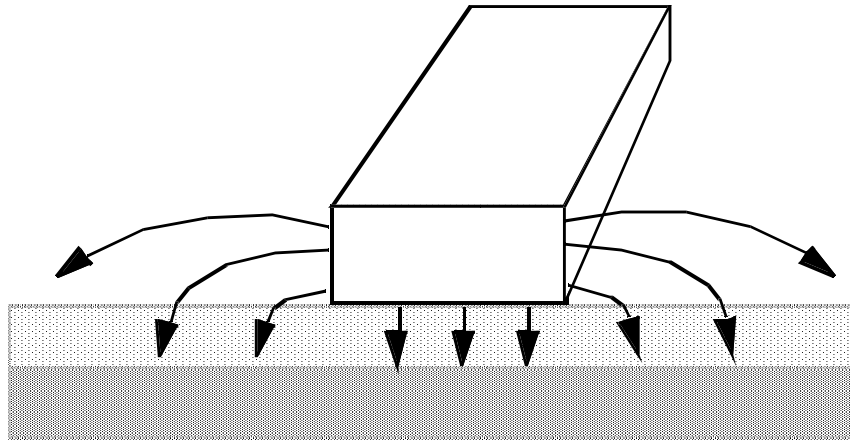


# Permittivity

Material	$\epsilon_r$
Free space	1
Aerogels	$\sim 1.5$
Polyimides (organic)	3-4
Silicon dioxide	3.9
Glass-epoxy (PC board)	5
Silicon Nitride ( $\text{Si}_3\text{N}_4$ )	7.5
Alumina (package)	9.5
Silicon	11.7

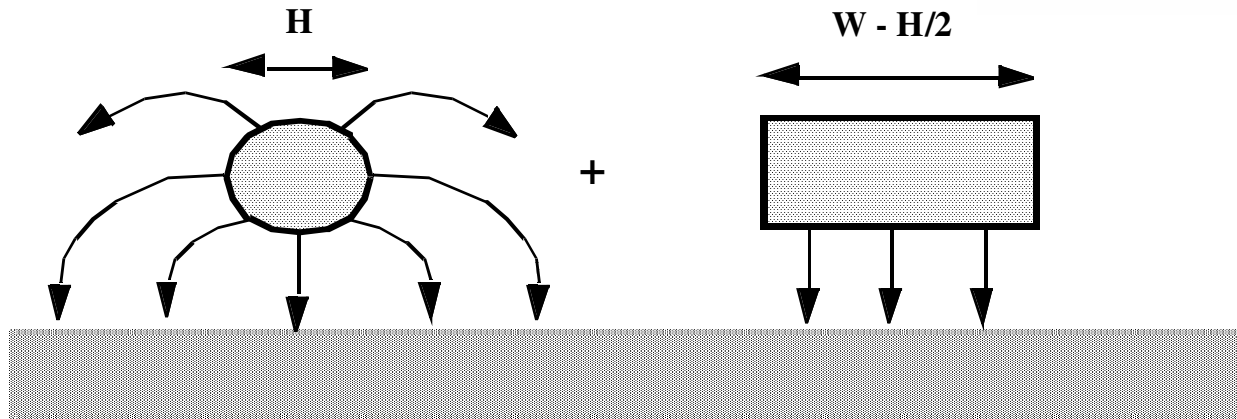


# Fringing Capacitance

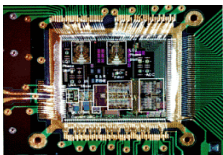


(a)

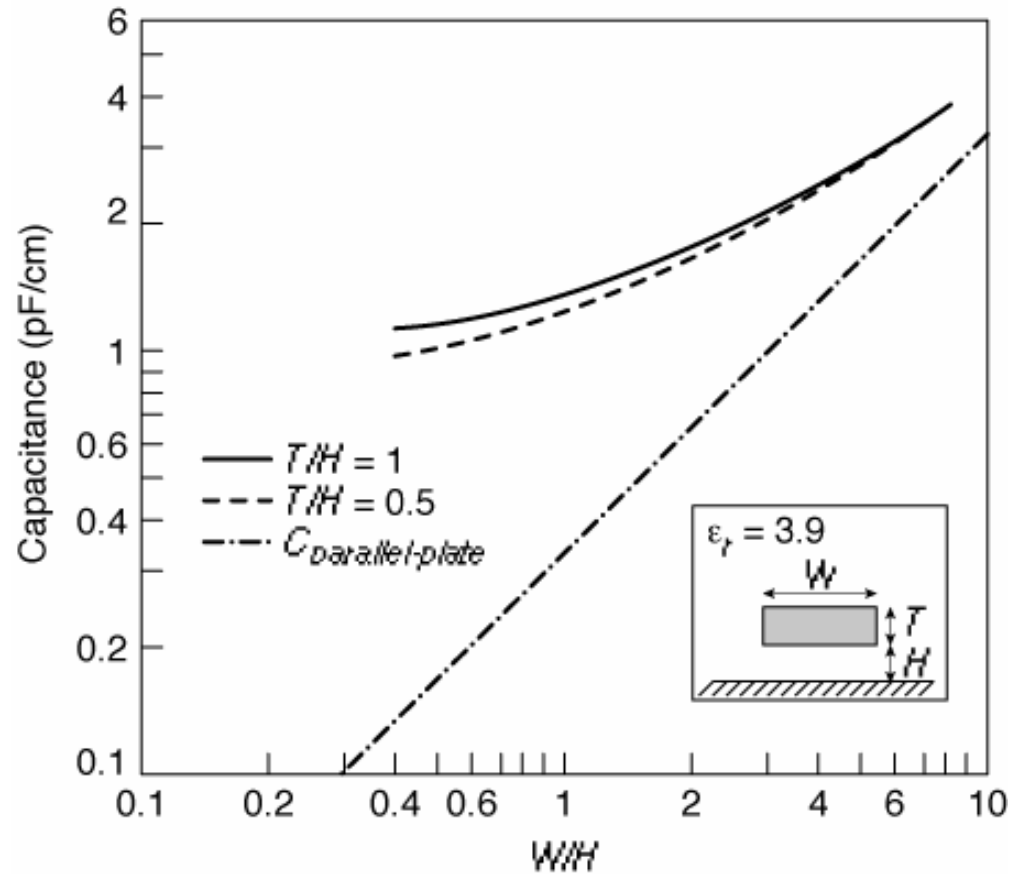
$$C_{wire} = C_{pp} + C_{fringe} = \frac{w\epsilon_{di}}{t_{di}} + \frac{2\pi\epsilon_{di}}{\log(t_{di}/H)}$$



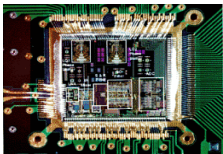
(b)



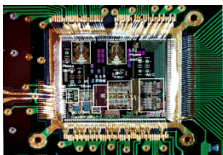
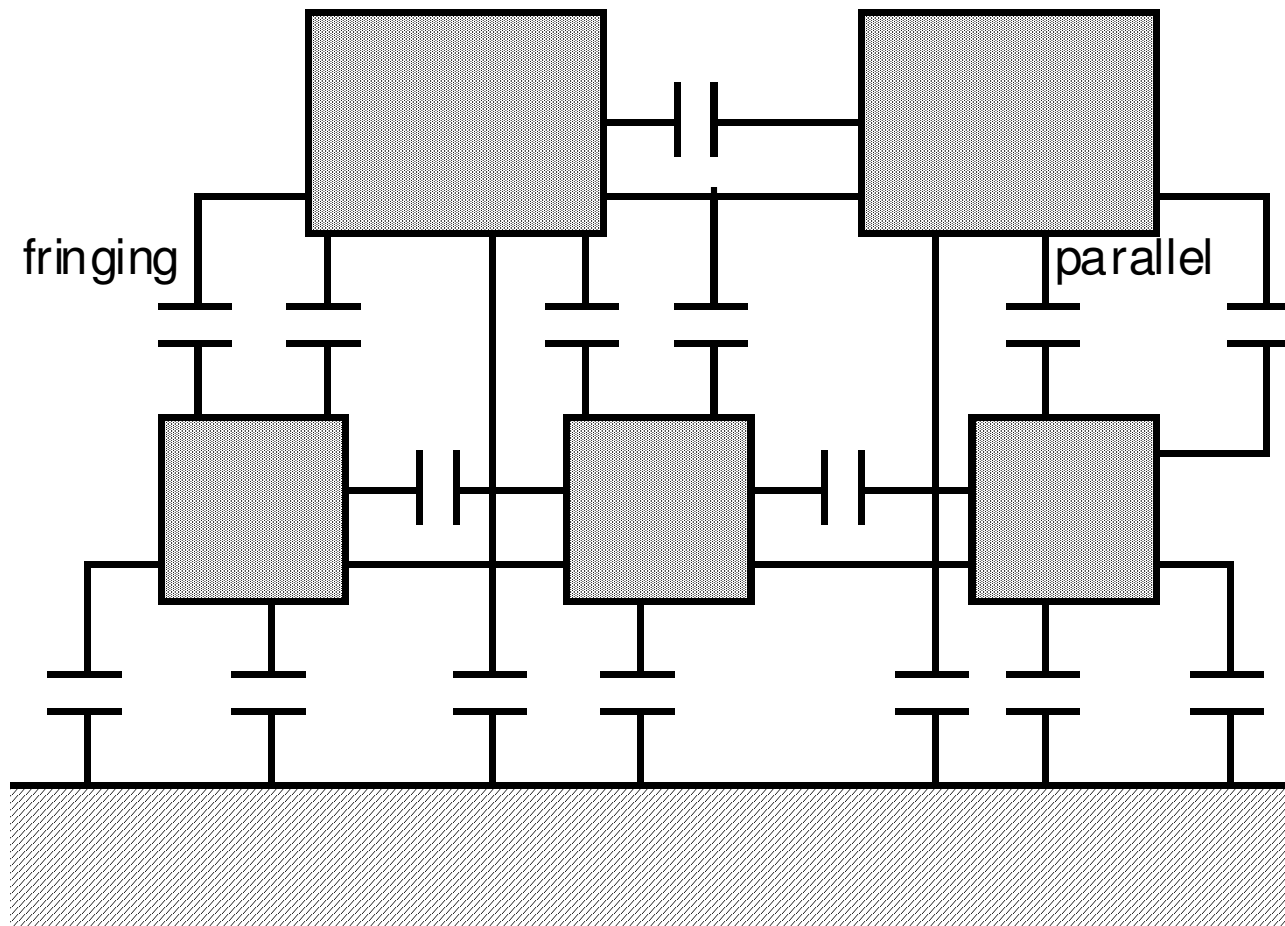
# Fringing versus Parallel Plate



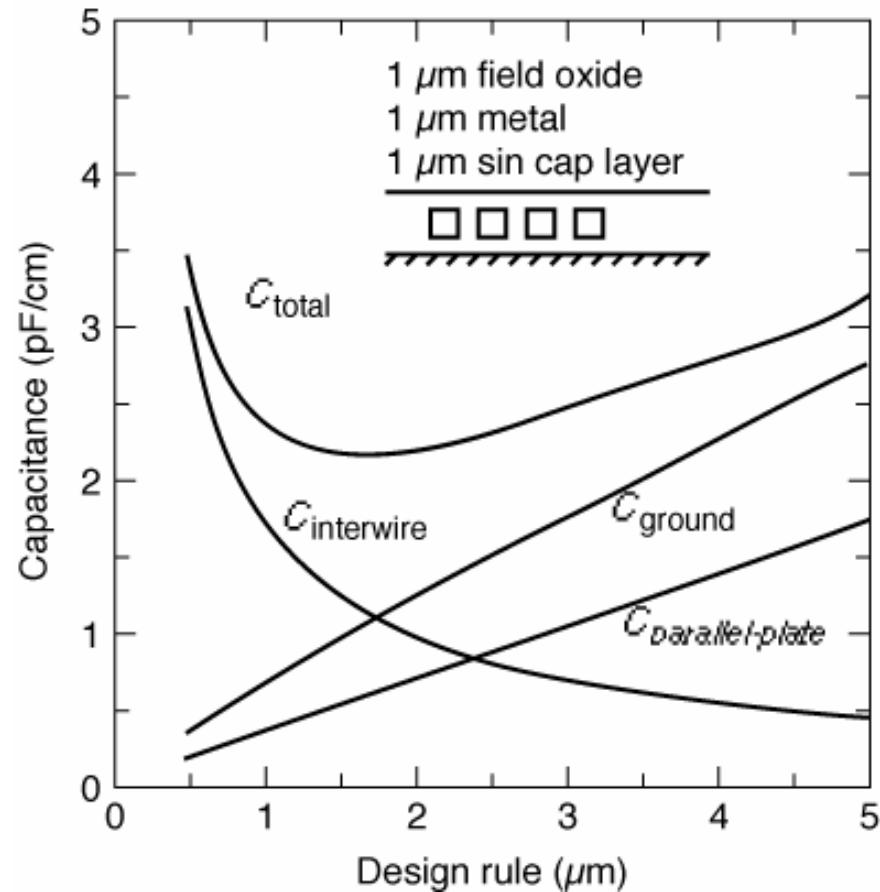
(from [Bakoglu89])



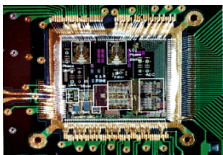
# Interwire Capacitance



# Impact of Interwire Capacitance

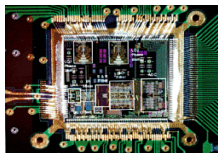


(from [Bakoglu89])

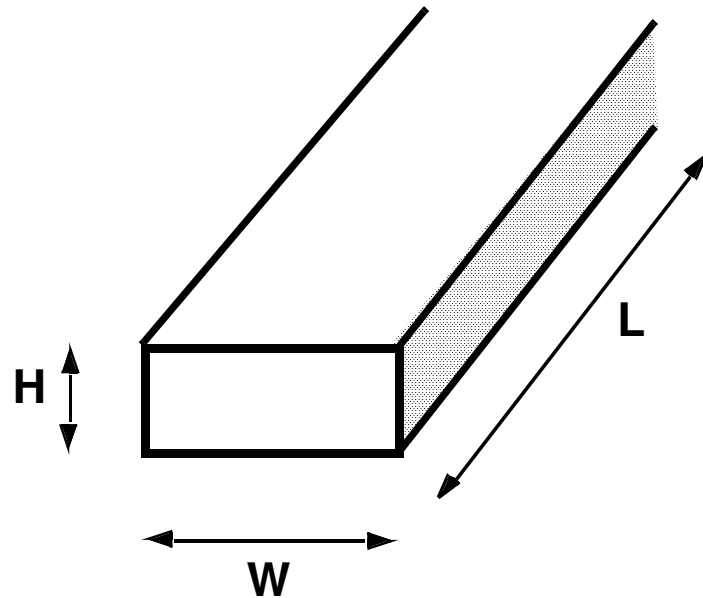


# Wiring Capacitances (0.25 $\mu\text{m}$ CMOS)

	Field	Active	Poly	Al1	Al2	Al3	Al4
Poly	88						
	54						
Al1	30	41	57				
	40	47	54				
Al2	13	15	17	36			
	25	27	29	45			
Al3	8.9	9.4	10	15	41		
	18	19	20	27	49		
Al4	6.5	6.8	7	8.9	15	35	
	14	15	15	18	27	45	
Al5	5.2	5.4	5.4	6.6	9.1	14	38
	12	12	12	14	19	27	52



# Wire Resistance

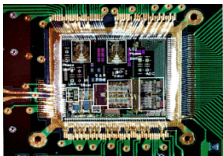
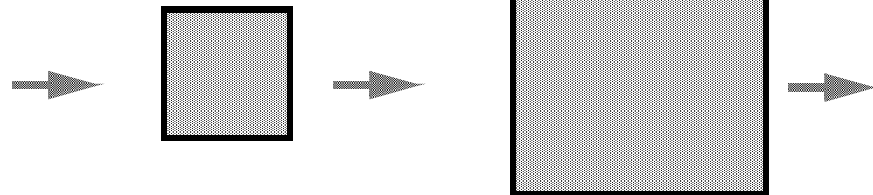


$$R = \frac{\rho L}{HW}$$

Sheet Resistance  
 $R_0$

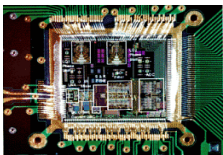
$$R_1 \equiv R_2$$

$R_2$



# Interconnect Resistance

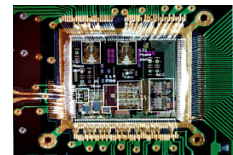
Material	$\rho$ ( $\Omega\text{-m}$ )
Silver (Ag)	$1.6 \times 10^{-8}$
Copper (Cu)	$1.7 \times 10^{-8}$
Gold (Au)	$2.2 \times 10^{-8}$
Aluminum (Al)	$2.7 \times 10^{-8}$
Tungsten (W)	$5.5 \times 10^{-8}$



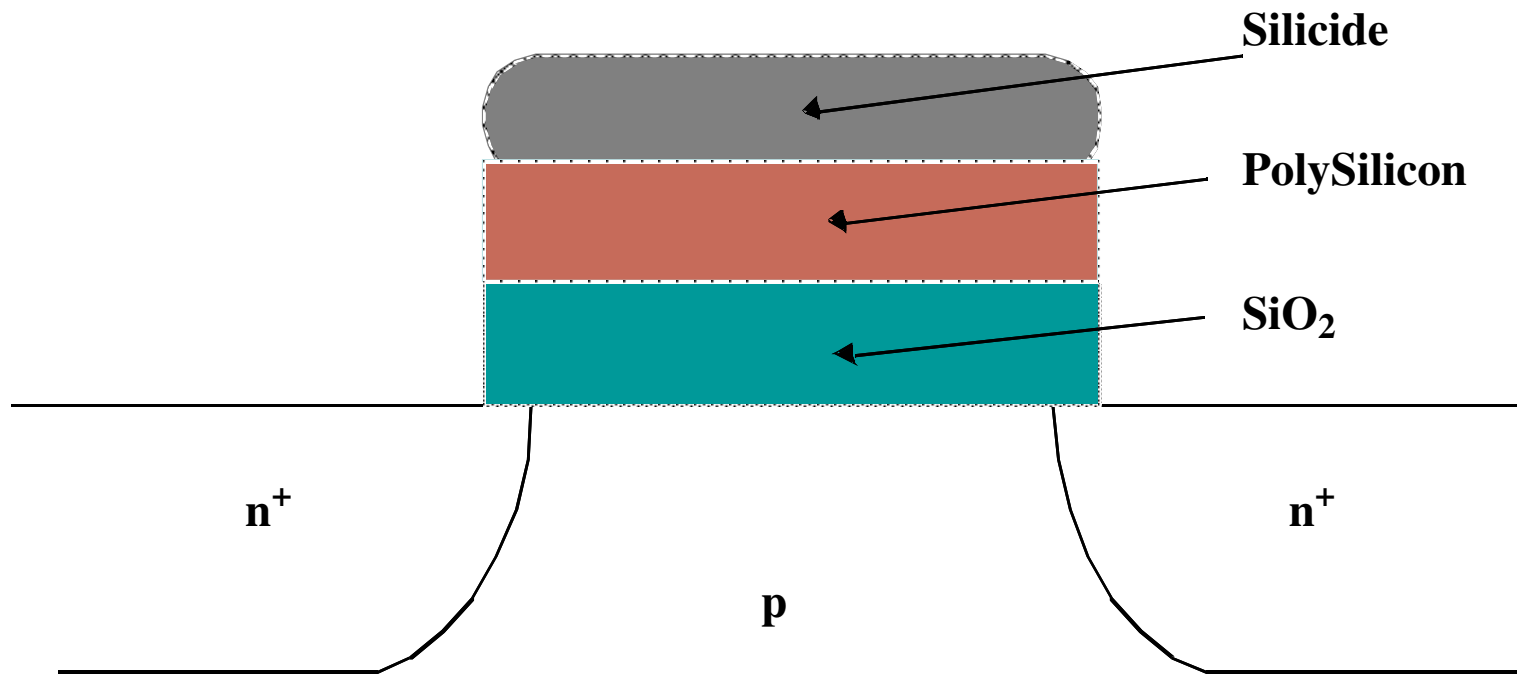


# Dealing with Resistance

- **Selective Technology Scaling**
- **Use Better Interconnect Materials**
  - reduce average wire-length
  - e.g. copper, silicides
- **More Interconnect Layers**
  - reduce average wire-length

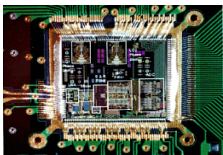


# Polycide Gate MOSFET



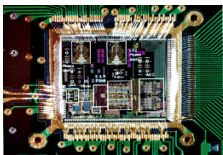
Silicides: WSi<sub>2</sub>, TiSi<sub>2</sub>, PtSi<sub>2</sub> and TaSi

Conductivity: 8-10 times better than Poly

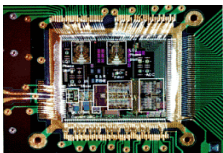
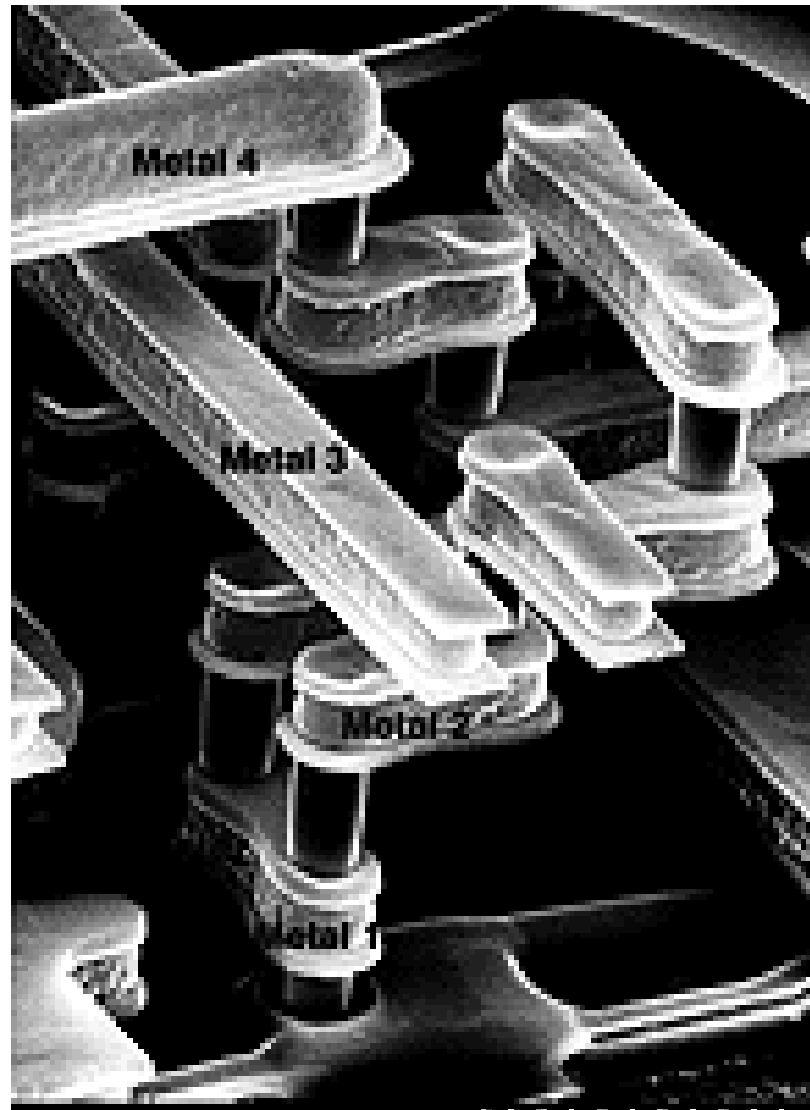


# Sheet Resistance

Material	Sheet Resistance ( $\Omega/\square$ )
n- or p-well diffusion	1000 – 1500
$n^+$ , $p^+$ diffusion	50 – 150
$n^+$ , $p^+$ diffusion with silicide	3 – 5
$n^+$ , $p^+$ polysilicon	150 – 200
$n^+$ , $p^+$ polysilicon with silicide	4 – 5
Aluminum	0.05 – 0.1



# Modern Interconnect



# Example: Intel 0.25 micron Process

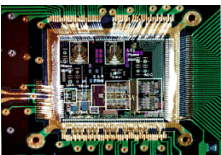
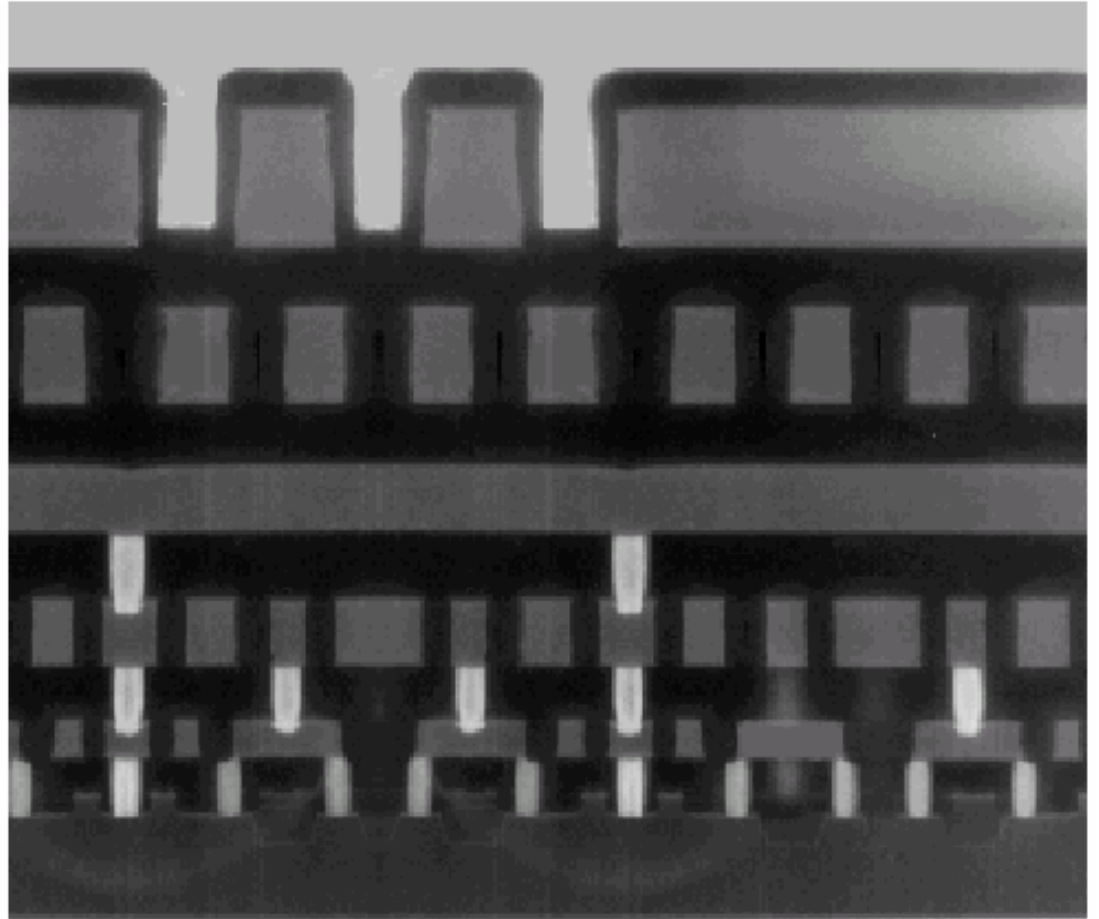
5 metal layers

Ti/Al - Cu/Ti/TiN

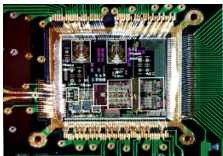
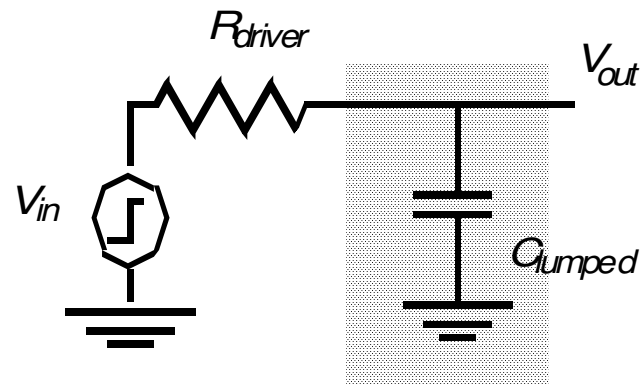
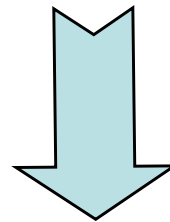
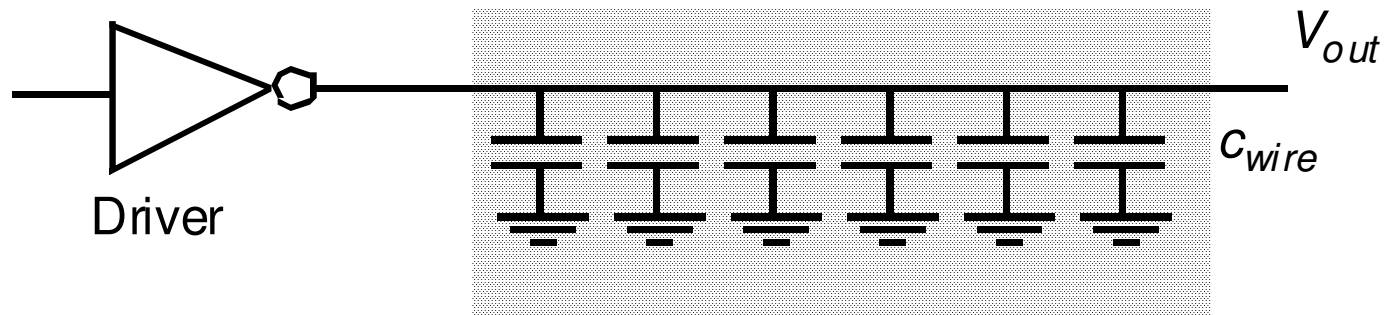
Polysilicon dielectric

<u>LAYER</u>	<u>PITCH</u>	<u>THICK</u>	<u>A.R.</u>
Isolation	0.67	0.40	-
Polysilicon	0.64	0.25	-
Metal 1	0.64	0.48	1.5
Metal 2	0.93	0.90	1.9
Metal 3	0.93	0.90	1.9
Metal 4	1.60	1.33	1.7
Metal 5	2.56	1.90	1.5
	$\mu\text{m}$	$\mu\text{m}$	

Layer pitch, thickness and aspect ratio

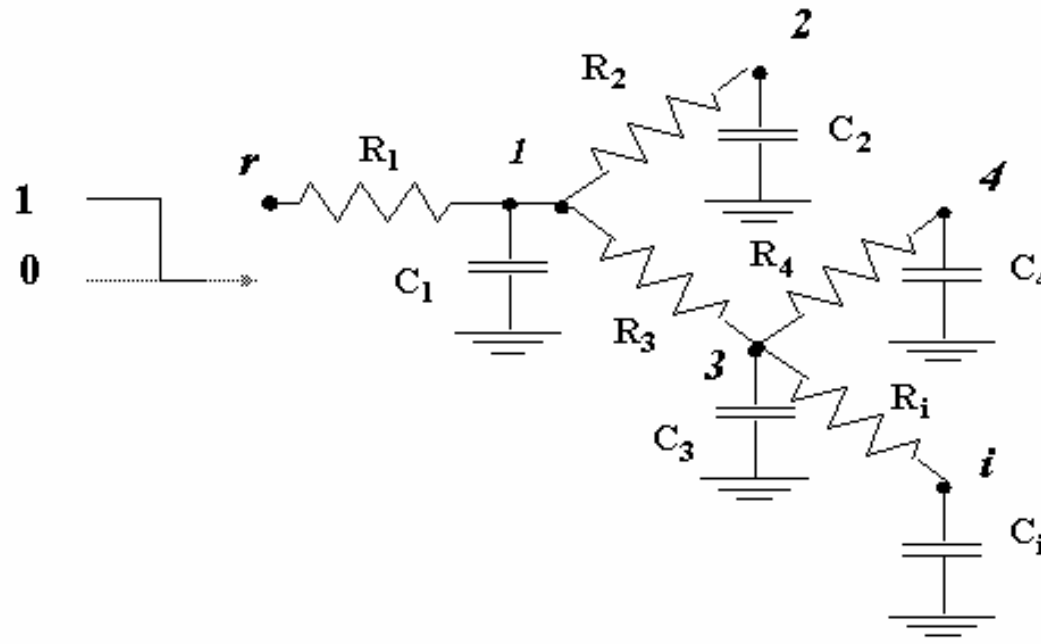


# The Lumped Model



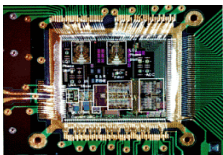
# The Lumped RC-Model

## The Elmore Delay

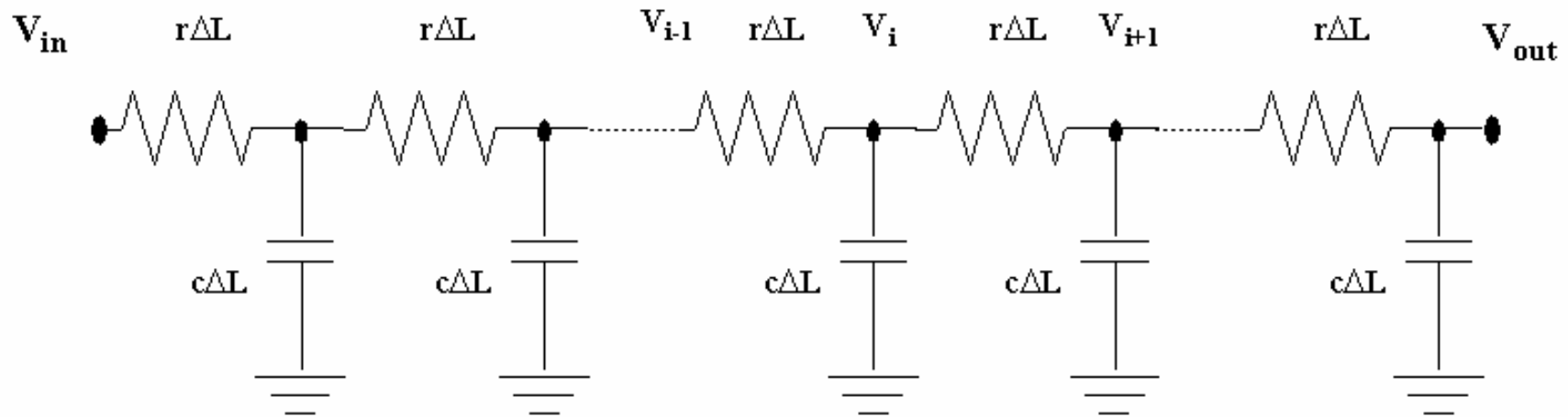


$$R_{ik} = \sum_N R_j \Rightarrow (R_j \in [path(s \rightarrow i) \cap path(s \rightarrow k)])$$

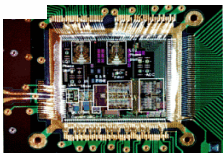
$$\tau_{Di} = \sum_{k=1} C_k R_{ik}$$



# The Elmore Delay RC Chain



$$\tau_N = \sum_{i=1}^N R_i \sum_{j=i}^N C_j = \sum_{i=1}^N C_i \sum_{j=1}^i R_j$$





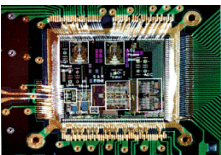
# Wire Model

Assume: Wire modeled by N equal-length segments

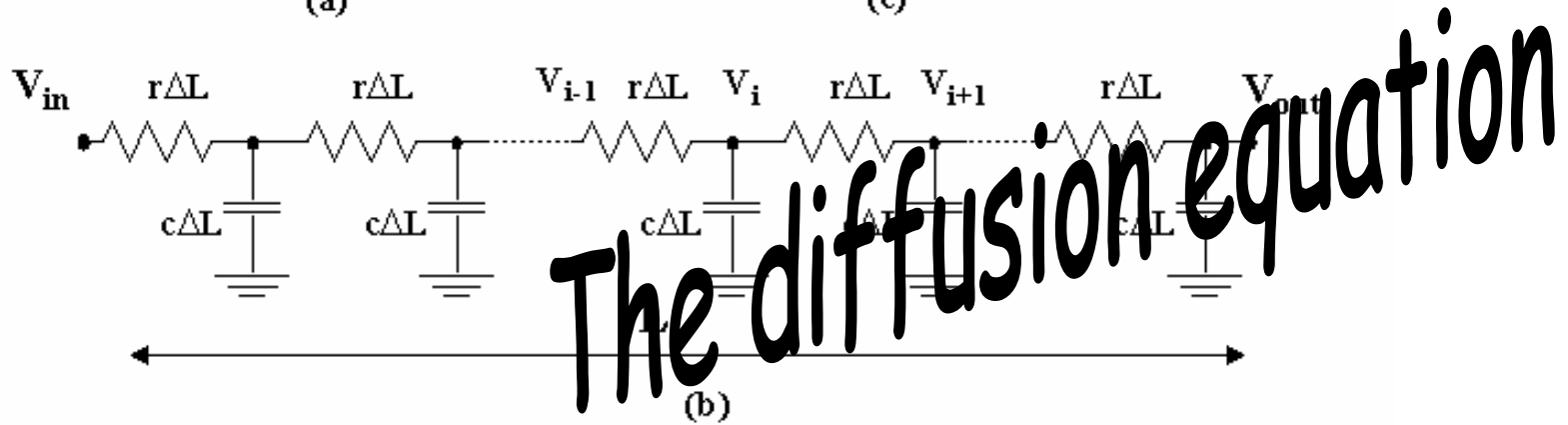
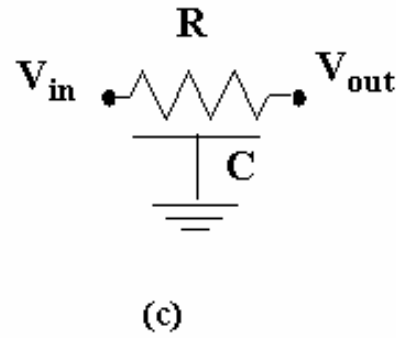
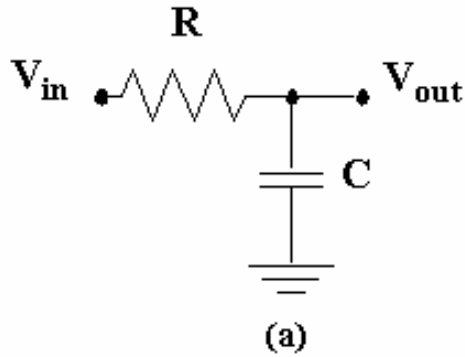
$$\tau_{DN} = \left(\frac{L}{N}\right)^2 (rc + 2rc + \dots + Nrc) = (rcL^2) \frac{N(N+1)}{2N^2} = RC \frac{N+1}{2N}$$

For large values of N:

$$\tau_{DN} = \frac{RC}{2} = \frac{rcL^2}{2}$$

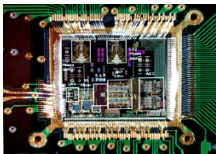


# The Distributed RC-line

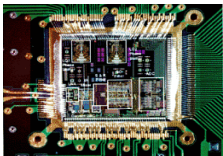
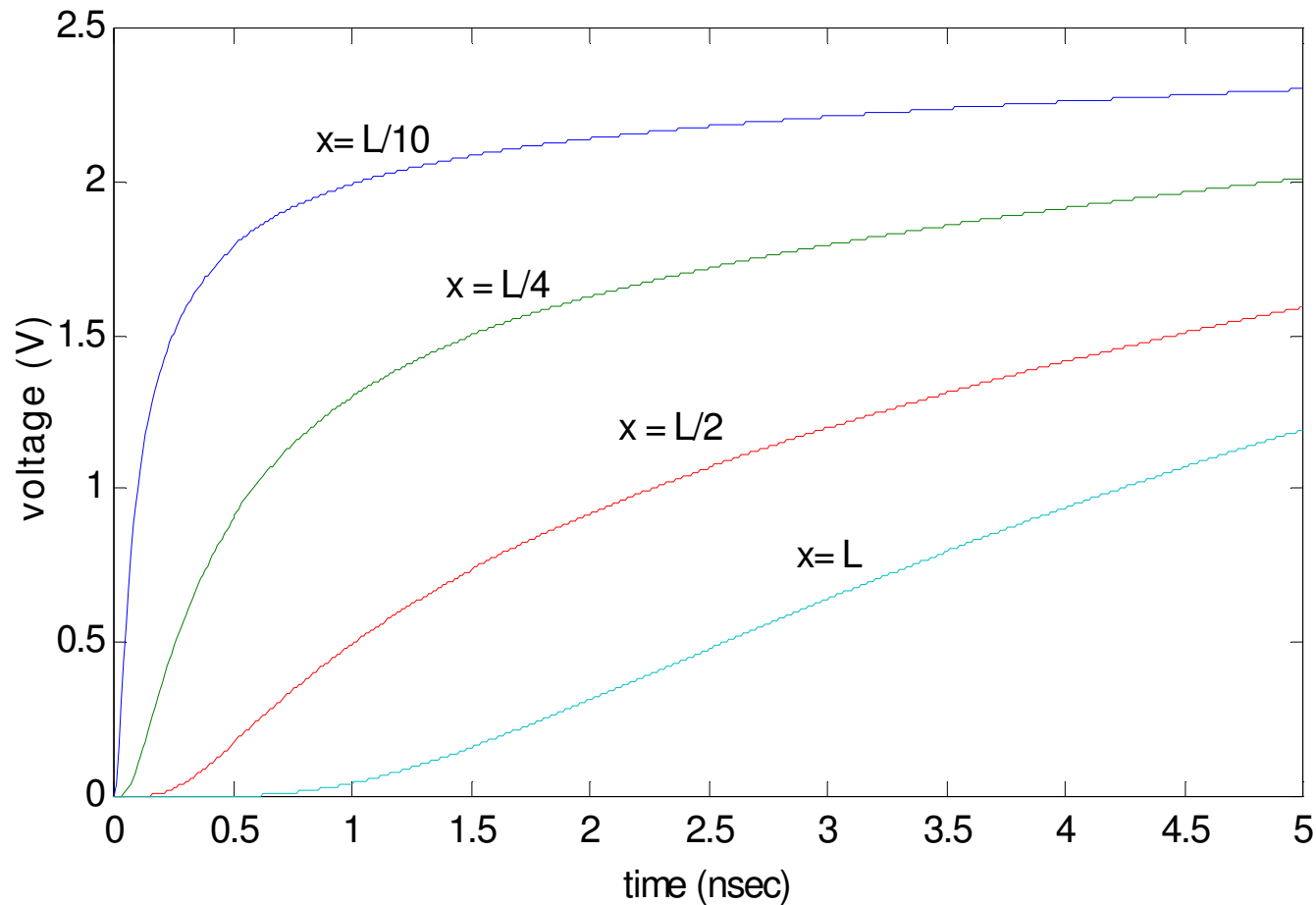


$$rc \frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2}$$

$$\tau(V_{out}) = \frac{rc L^2}{2}$$



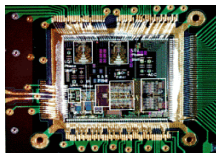
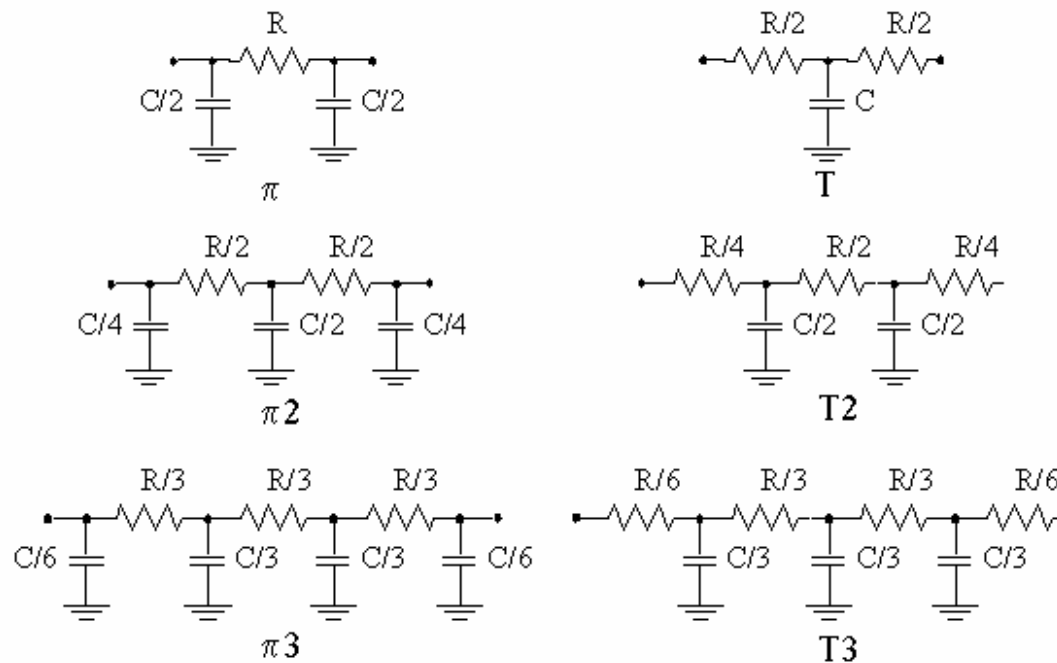
# Step-response of RC wire as a function of time and space



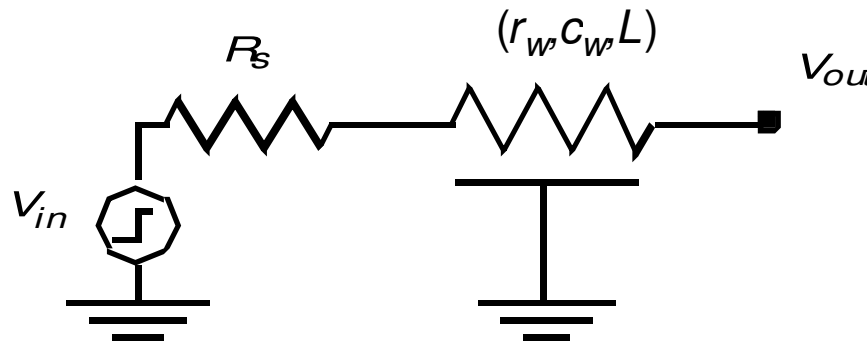
# RC-Models

Voltage Range	Lumped RC-network	Distributed RC-network
0→50% ( $t_p$ )	0.69 RC	0.38 RC
0→63% ( $\tau$ )	RC	0.5 RC
10%→90% ( $t_T$ )	2.2 RC	0.9 RC

Step Response of Lumped and Distributed RC Networks:  
Points of Interest.

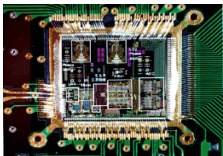


# Driving an RC-line



$$\tau_D = R_s C_w + \frac{R_w C_w}{2} = R_s C_w + 0.5 r_w c_w L^2$$

$$t_p = 0.69 R_s C_w + 0.38 R_w C_w$$



# Design Rules of Thumb

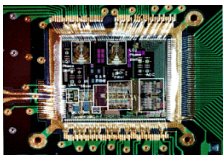
- RC delays should only be considered when  $t_{pRC} \gg t_{pgate}$  of the driving gate

$$L_{crit} \gg \sqrt{t_{pgate}/0.38RC}$$

- RC delays should only be considered when the rise (fall) time at the line input is smaller than RC, the rise (fall) time of the line

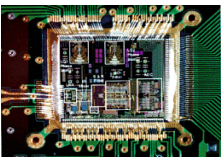
$$t_{rise} < RC$$

- when not met, the change in the signal is slower than the propagation delay of the wire

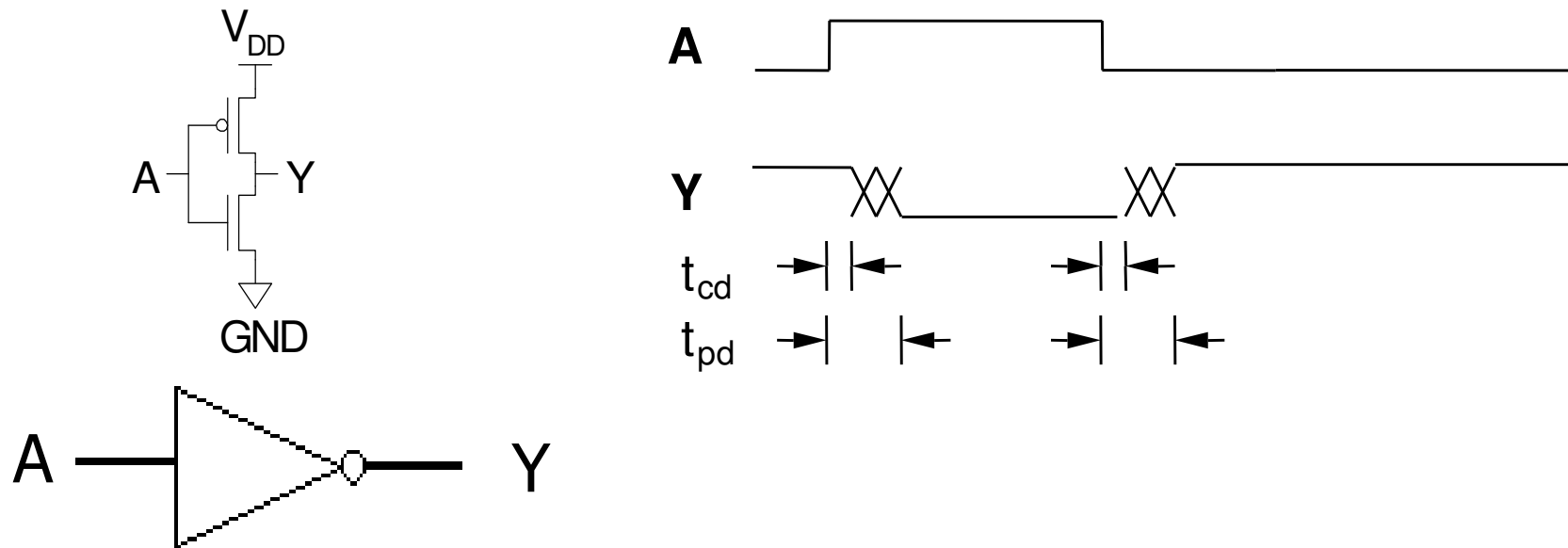


# Delay Definitions

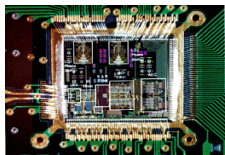
- Combinational logic has two types of delay:
  - Propagation
  - Contamination
- When the input changes, output retains its old value for **at least** the **contamination** delay and takes on its new value in **at most** the **propagation** delay.
- The gate that charges or discharges a node is called **driver**, and the gates and wires that are being driven are called **load**.



# Delay Definitions : Prop. Vs Contamination

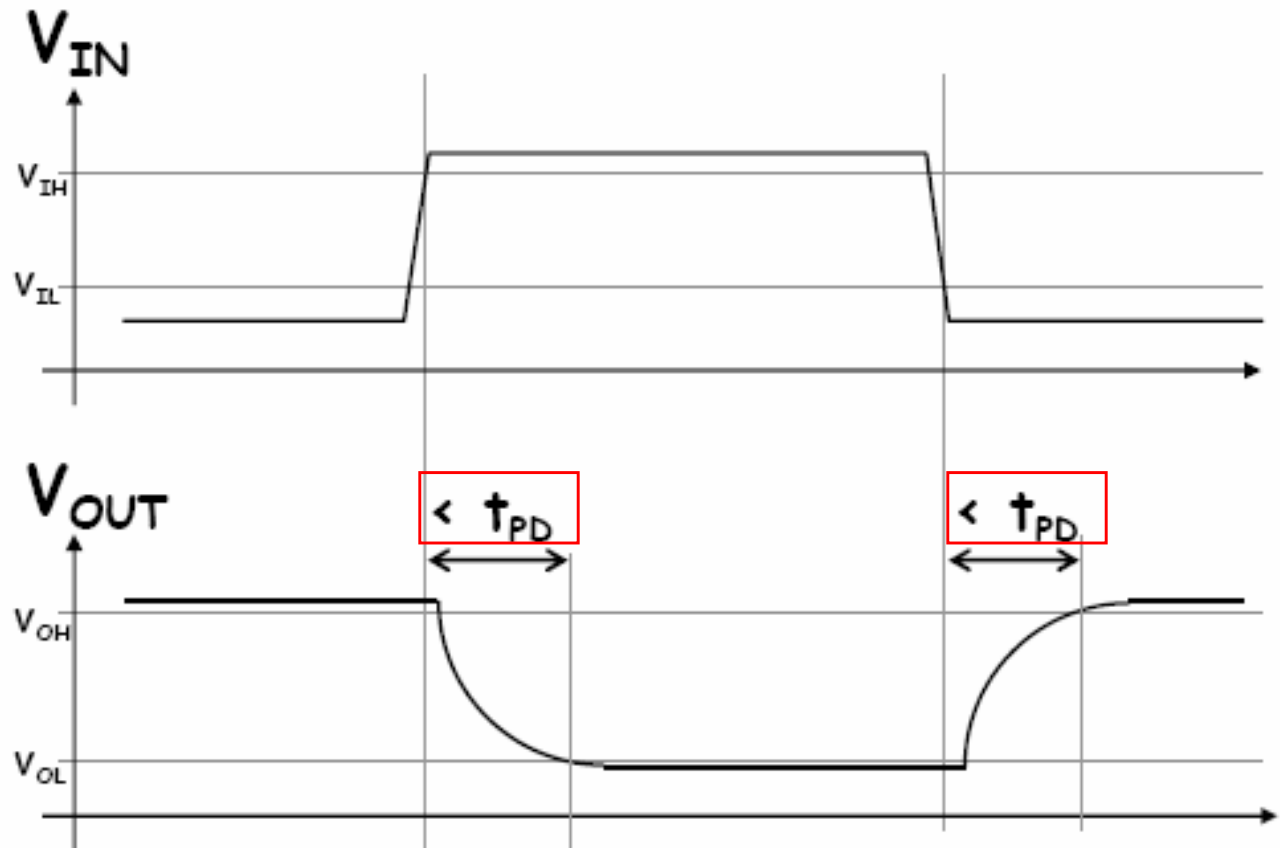
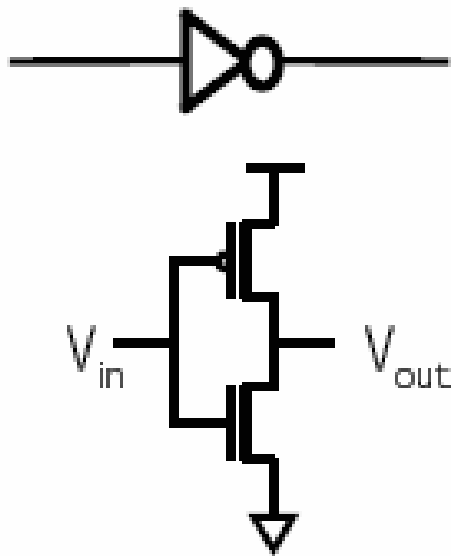


- The output remains unchanged for a time period equal to the **contamination delay**,  $t_{cd}$
- The new output value is guaranteed to valid after a time period equal to the **propagation delay**,  $t_{pd}$

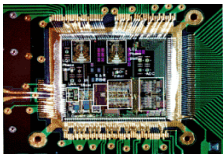




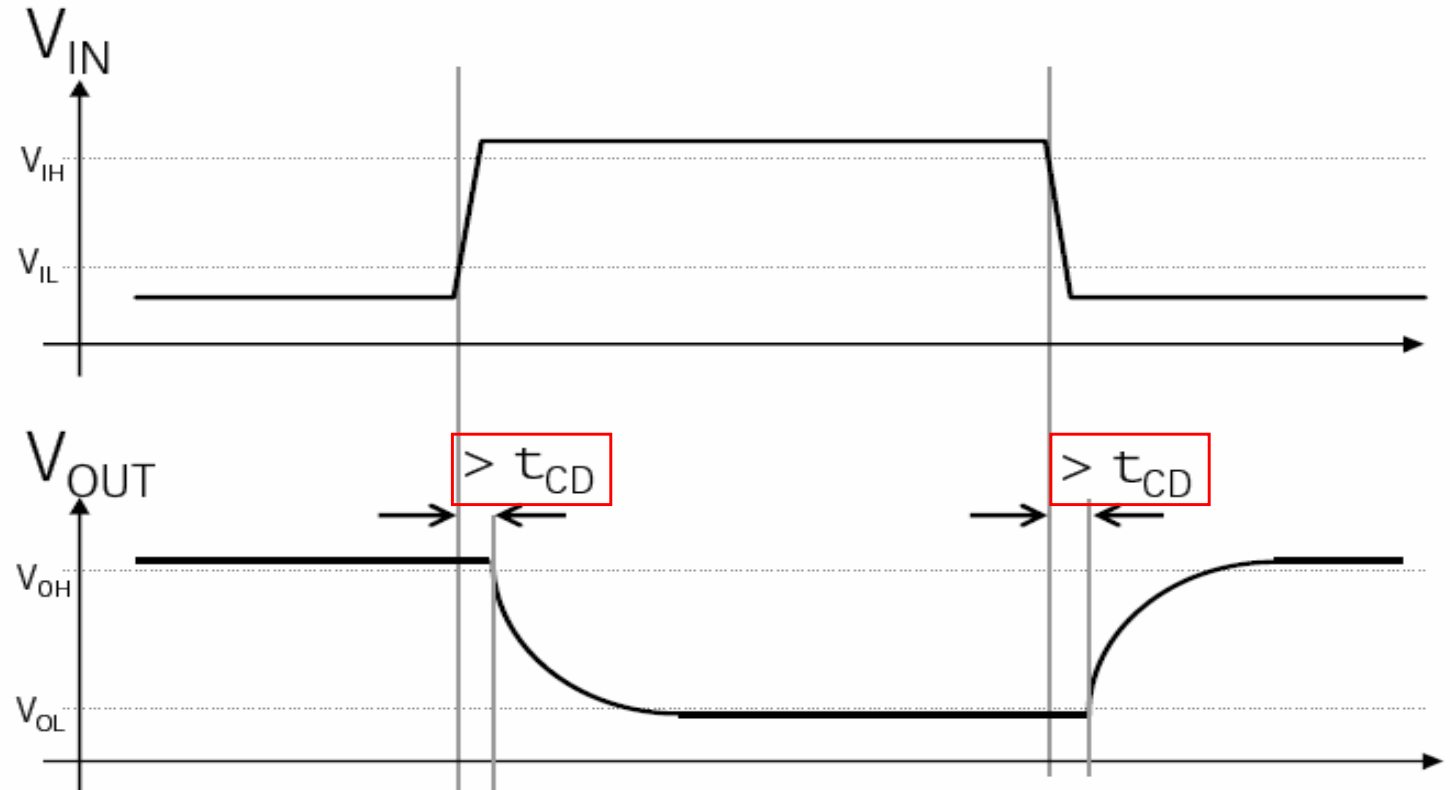
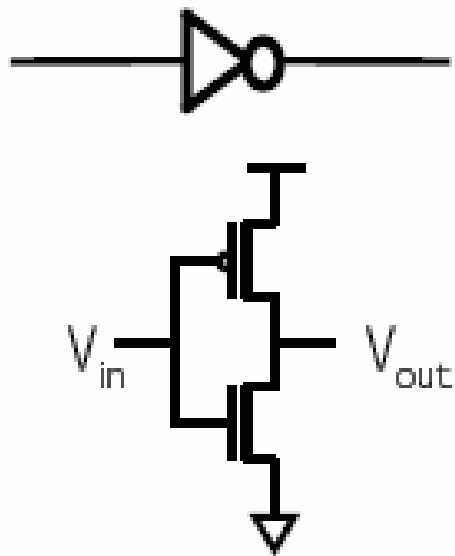
# Delay Definitions : Propagation



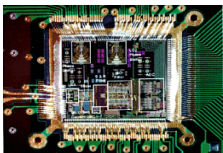
Source: <http://www.unc.edu/courses/2003fall/comp/120/001/handouts/Lecture04.pdf>



# Delay Definitions : Contamination

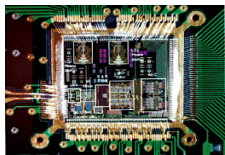


Source: <http://www.unc.edu/courses/2003fall/comp/120/001/handouts/Lecture04.pdf>



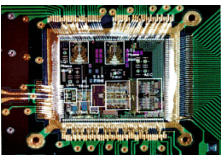
# Delay Definitions : Propagation

- $t_{pdr}$ : rising propagation delay
  - Time delay from the reference voltage ( $V_{DD}/2$ ) at the input to the reference voltage at the output, when output voltage is going from **low-to-high**.
- $t_{pdf}$ : falling propagation delay
  - Time delay from the reference voltage ( $V_{DD}/2$ ) at the input to the reference voltage at the output, when output voltage is going from **high-to-low**.
- $t_{pd}$ : (average) **propagation** delay (also **max-time**)
  - defined in two ways: (maximum or average of two)
    - maximum ( $t_{pdr}$ ,  $t_{pdf}$ )
    - $t_{pd} = (t_{pdr} + t_{pdf})/2$

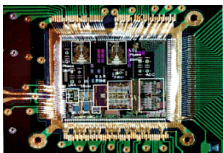
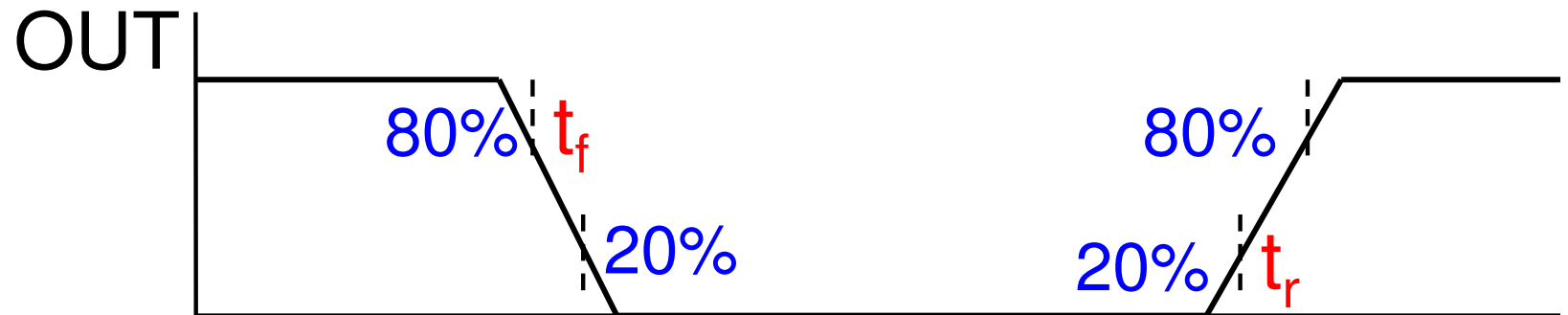
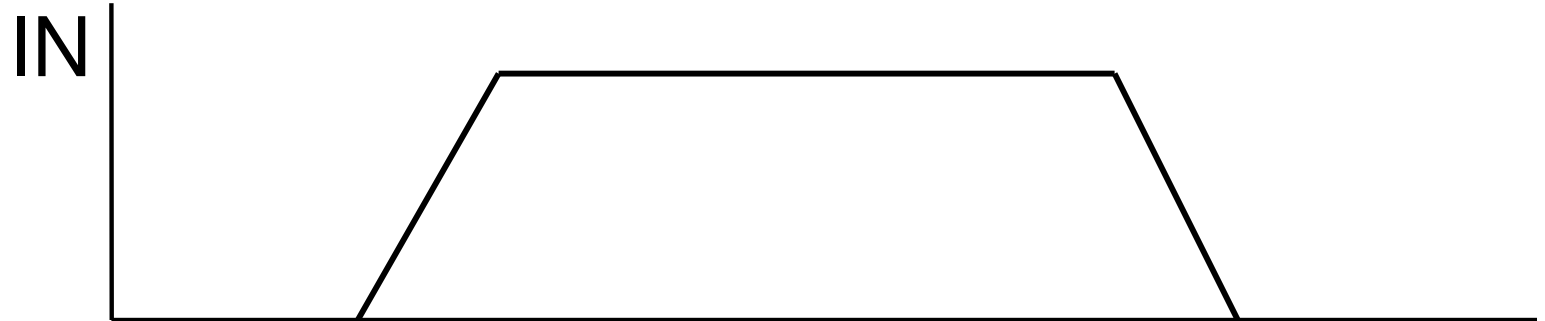
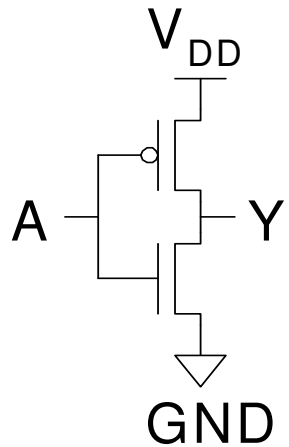


# Delay Definitions : Rise and Fall

- $t_r$ : 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  $0.2 V_{DD}$
- Rise / Fall times are also called slope or edge rates.
- **Edge Rate**:  $t_{rf} = (t_r + t_f)/2$

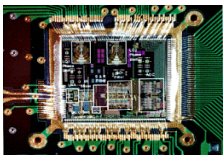
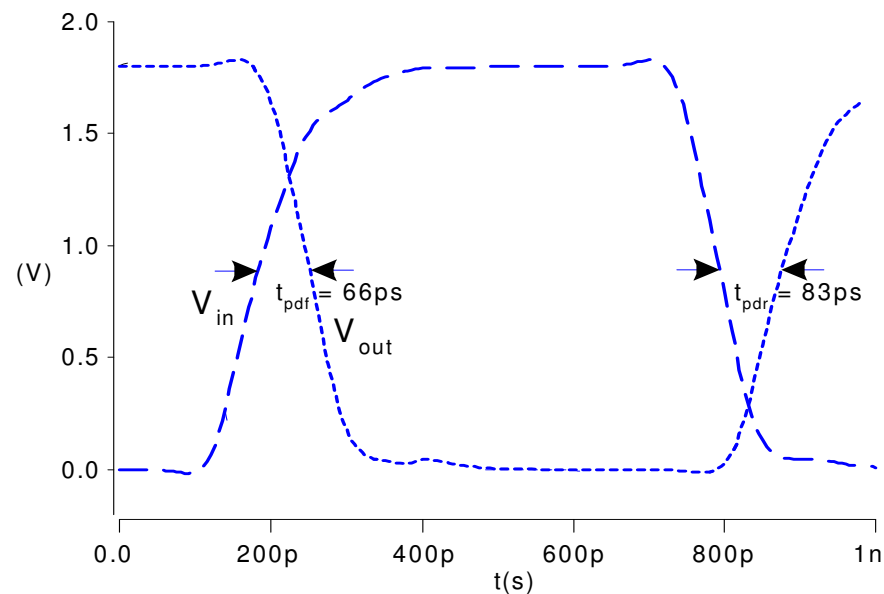


# Delay Definitions : Rise and Fall ....



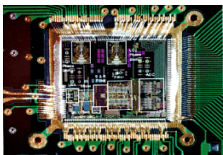
# Simulated Inverter Delay

- 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
  - Not as accurate as simulation
- The step response usually looks like a 1<sup>st</sup> order RC response with a decaying exponential.
- Use RC delay models to estimate delay
  - $C$  = total capacitance on output node
  - Use effective resistance  $R$
  - So that  $t_{pd} = RC$
- Characterize transistors by finding their effective  $R$ 
  - Depends on average current as gate switches



# Switch-level RC Delay Models

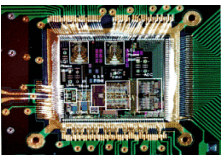
- RC models treat MOSFETs as switches in series with resistors.
- Unit effective resistance  $R$  can be obtained from any operating point of I-V characteristics as:

$$R = 1 / (\partial I_{ds} / \partial V_{ds})$$

- When  $\partial V_{ds}$  is small the resistance  $R$  can be obtained by differentiating the  $I_{ds}$  equation:

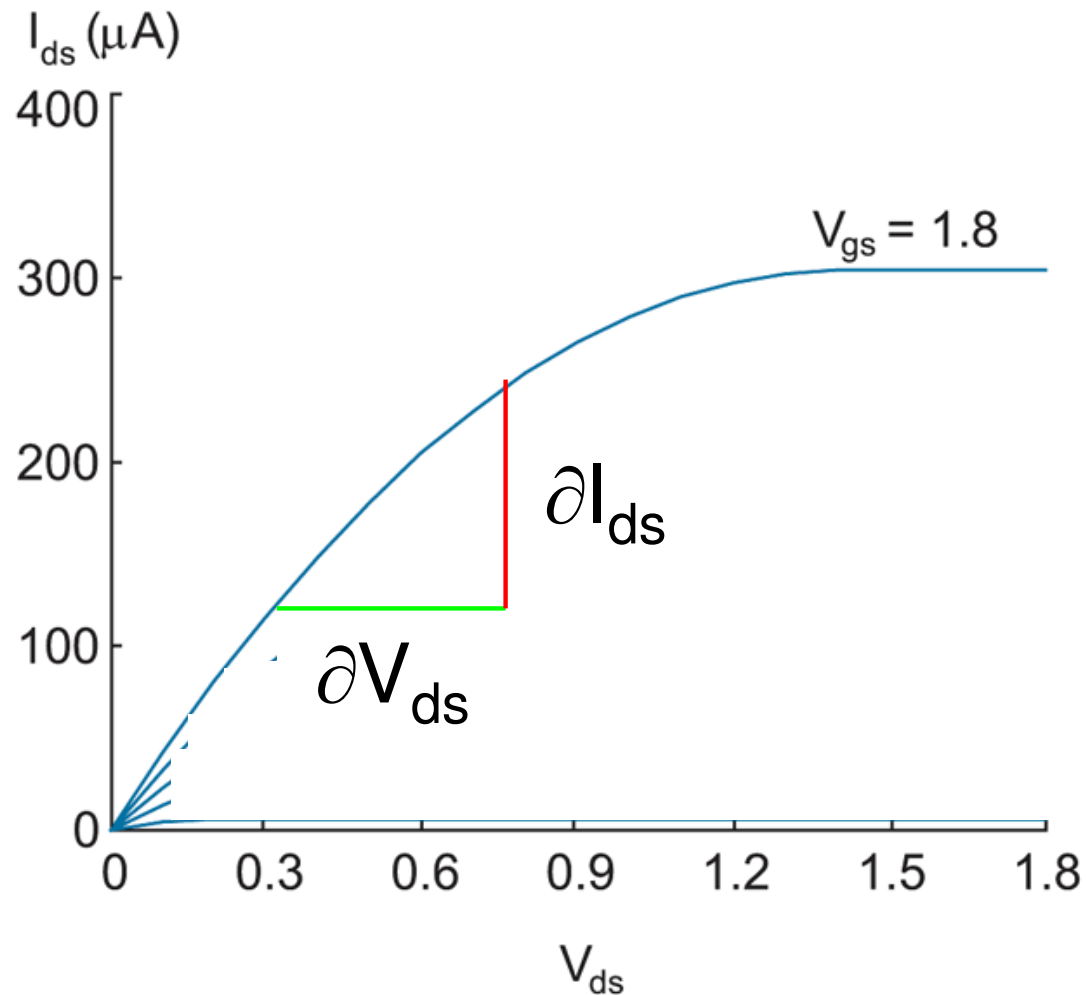
$$R = 1 / [\beta (V_{gs} - V_t)]$$

- **NOTE:** The above way of calculating resistance is not practically accurate as the non-ideal effects (velocity saturation) have strong impact on it.

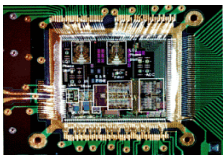




# Switch-level RC Delay Models ...

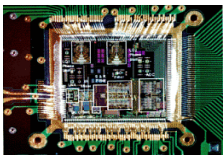


Slope of a curve gives conductance, inverse of which is resistance.



# Switch-level RC Delay Models ...

- 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 proportional to width: If unit effective resistance is  $R$ , then the transistor of width  $k$  units has resistance  $R/k$ .
- Resistance inversely proportional to width: If  $C$  is the capacitance of a unit transistor, then the transistor of width  $k$  units has capacitance  $kC$ .



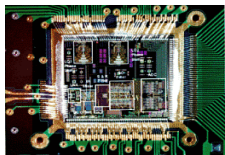
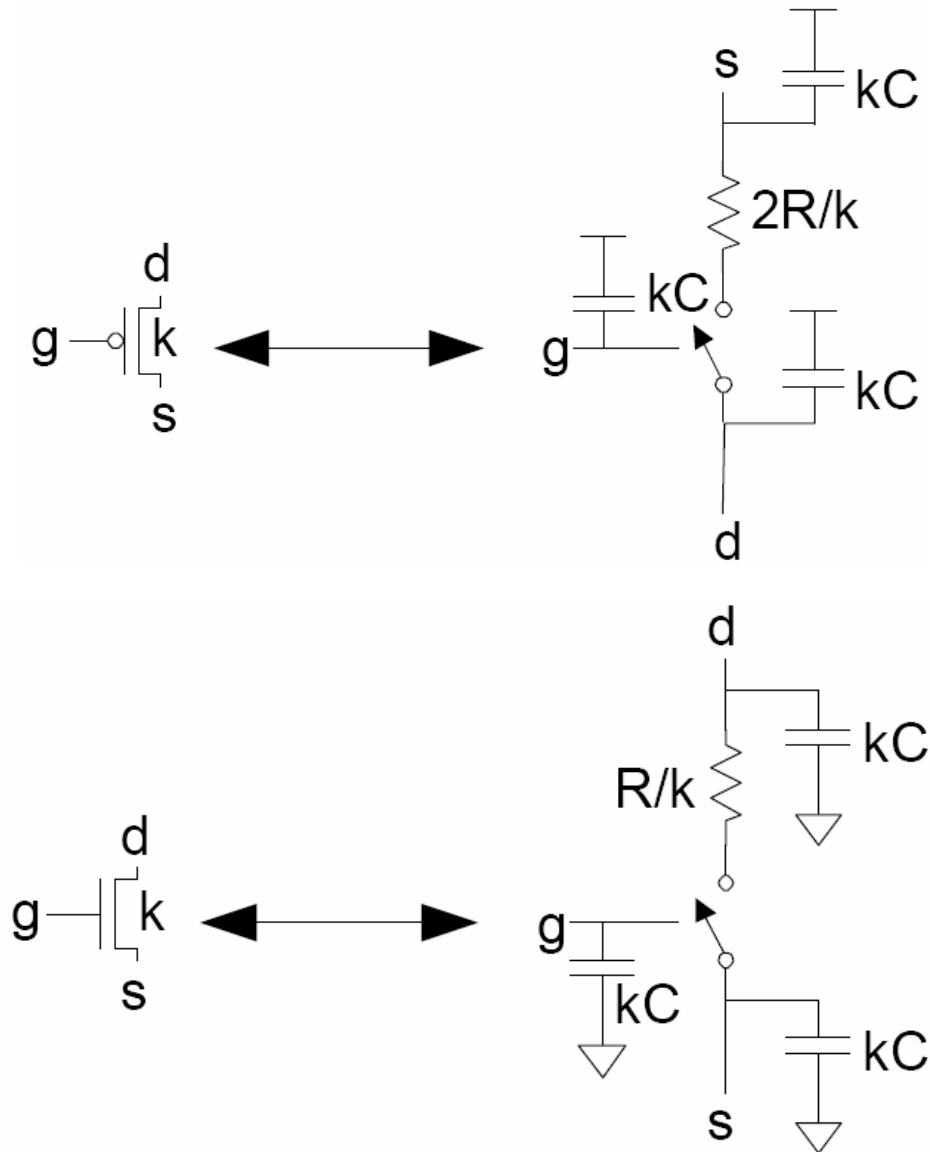
# RC Delay Models: Inverter

## PMOS equivalent RC model:

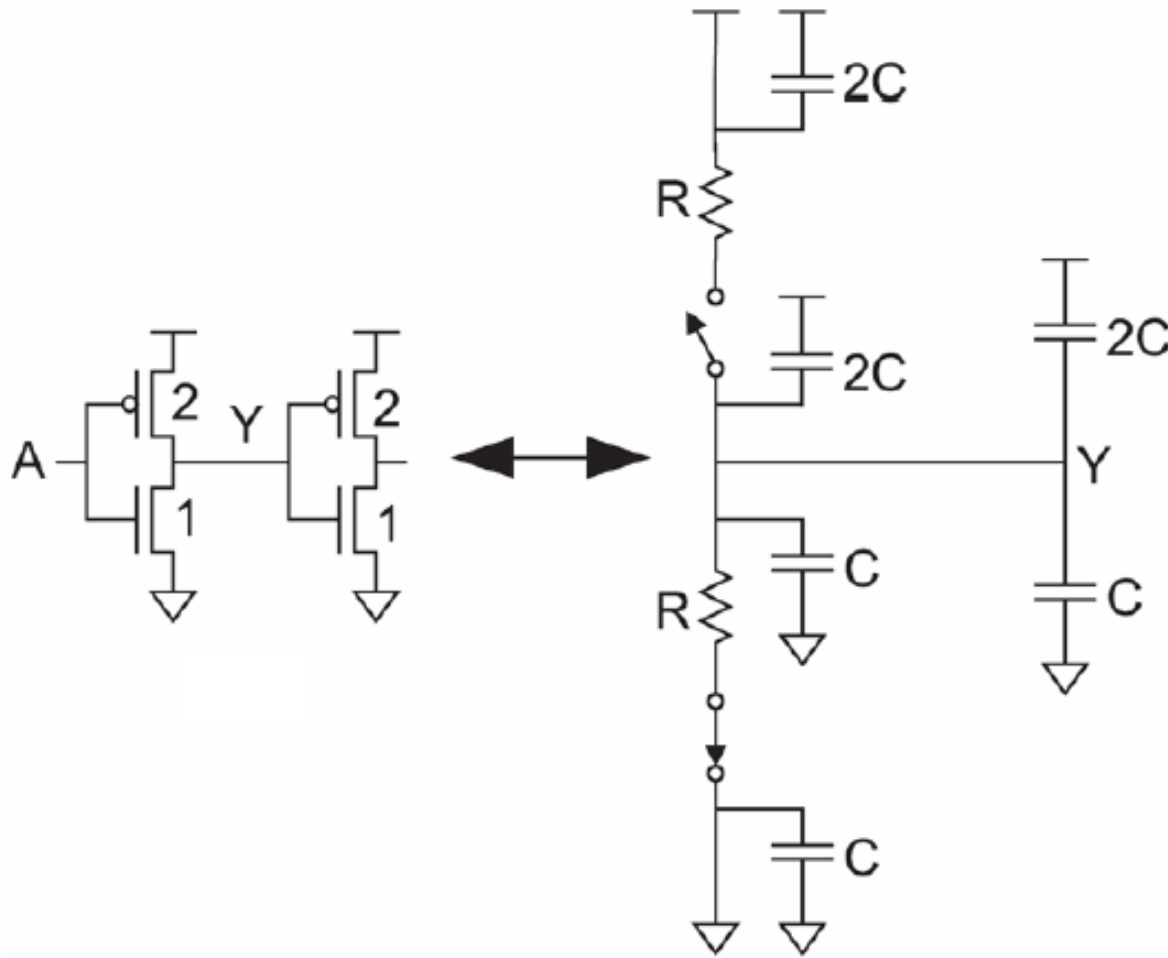
- Width of transistor is  $k$  units
- Both gate and diffusion capacitances shown
- One terminal is shown connected to  $V_{dd}$  (n-well)

## NMOS equivalent RC model:

- Width of transistor is  $k$  units
- Both gate and diffusion capacitances shown
- One terminal is shown connected to GND (substrate)



# RC Delay Models: Inverter ...

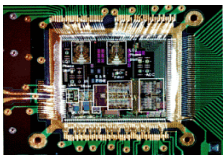


Estimation of delay of a fanout-of-1 inverter.

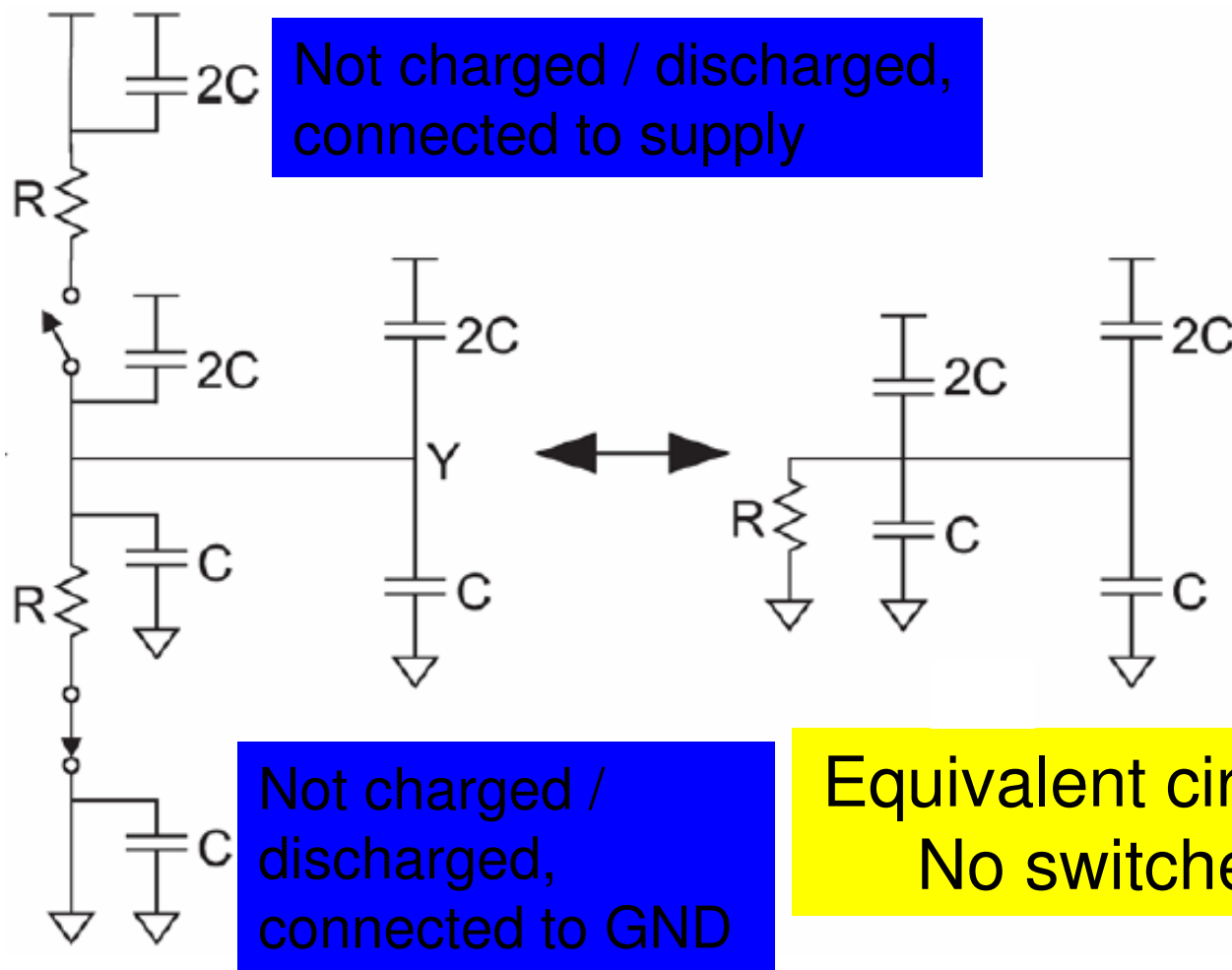
NMOS is of 1-unit width and PMOS is of 2-unit width to achieve equal fall / rise resistance.

Inverter  
fanout-of-1

Equivalent circuit :  
1<sup>st</sup> inverter driving 2<sup>nd</sup>



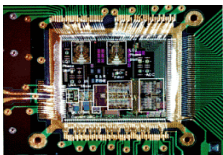
# RC Delay Models: Inverter ...



- $T_{pd} = R * (6C) = 6RC$

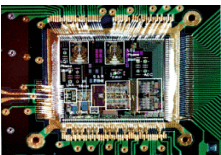
- Time constant  
 $\tau = RC$

Equivalent circuit :  
1<sup>st</sup> inverter driving 2<sup>nd</sup>



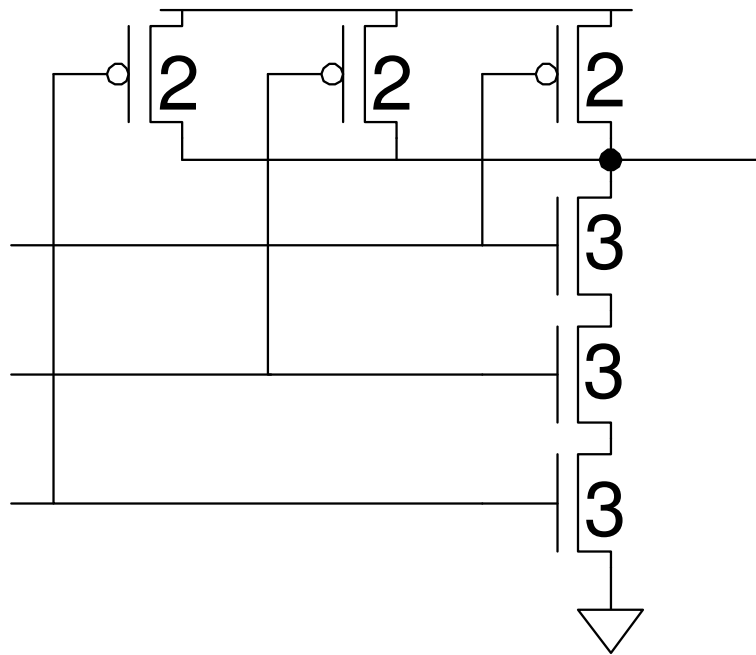
# Effective Resistance and Capacitance

- Parallel and series transistors combine like conventional resistors.
- **When in series**: Total resistance is the sum of all
- **When in parallel**: Total conductance is the sum of conductance, inverse of which is the total resistance.
- Resistance is low if they are in parallel.
- Worst case delay  $\rightarrow$  when only one of several parallel transistors is ON.



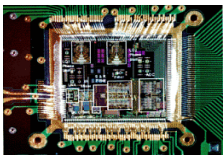
# Effective R and C : 3-input NAND Example

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



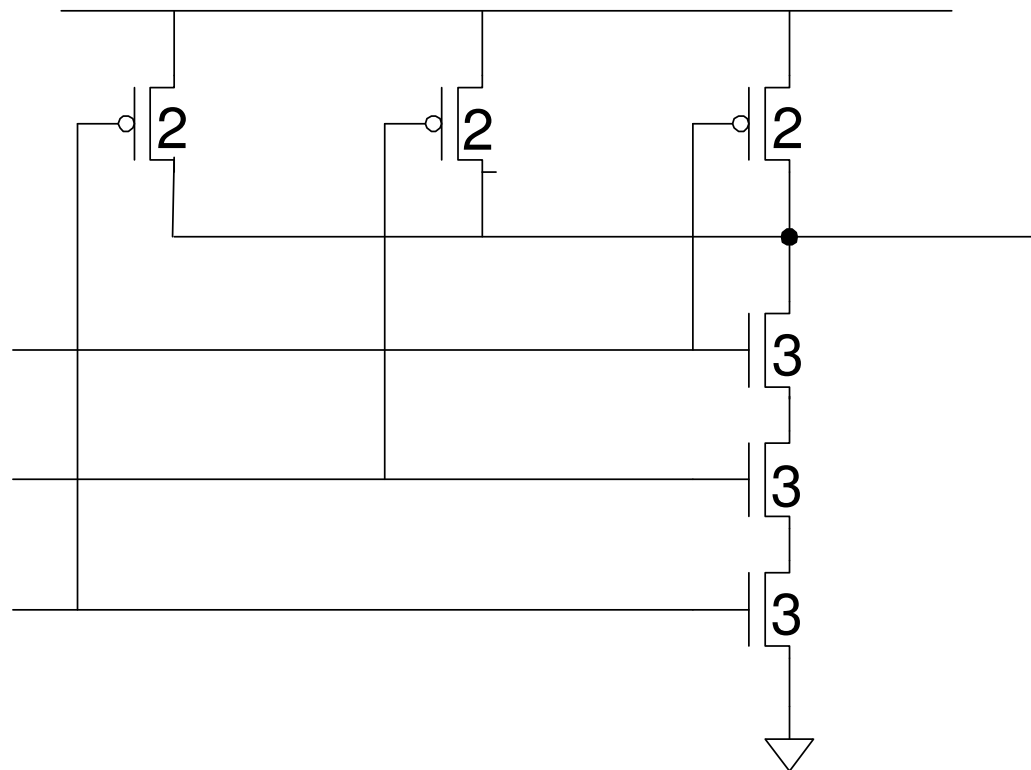
3-input NAND

- Each NMOS should have  $R/3$  resistance
- Each PMOS should have  $R$  resistance (worst case one even one ON should provide  $R$  resistance).
- Since 1-unit NMOS has  $R$  resistance, so its  $W/L$  is 3.
- Also 1-unit PMOS has  $2R$  resistance, so its  $W/L$  is 2.

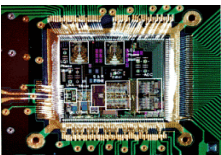


# Effective R and C : 3-input NAND Capacitance

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

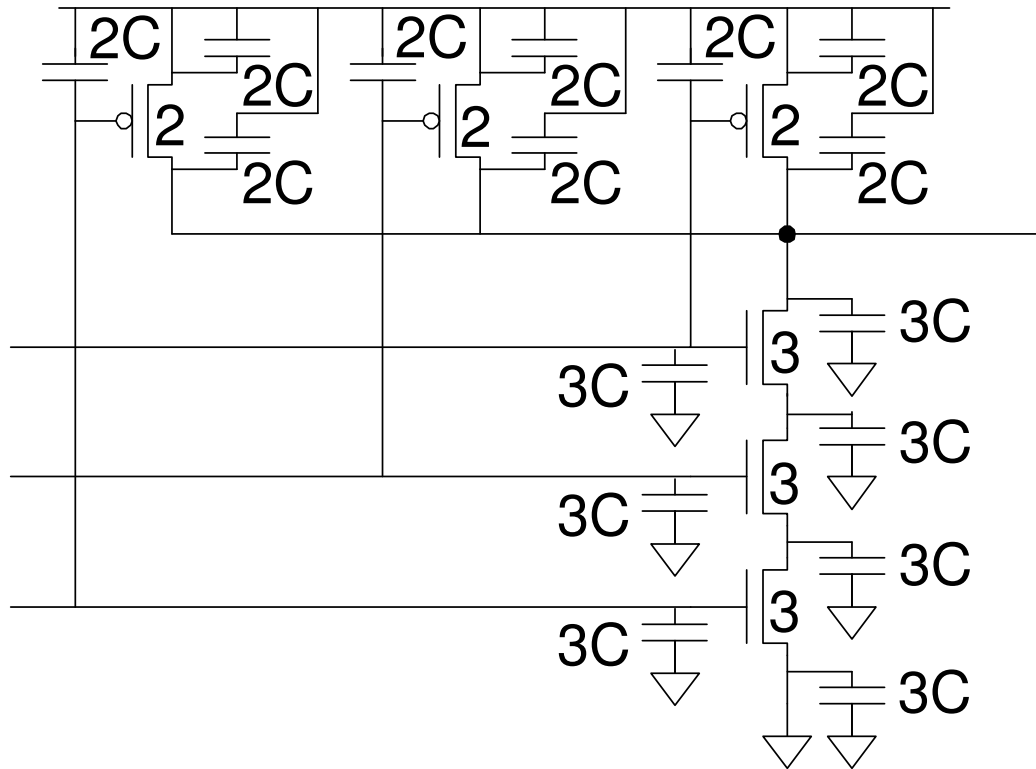


3-input NAND





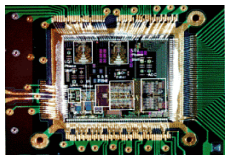
# Effective R and C : 3-input NAND Capacitance



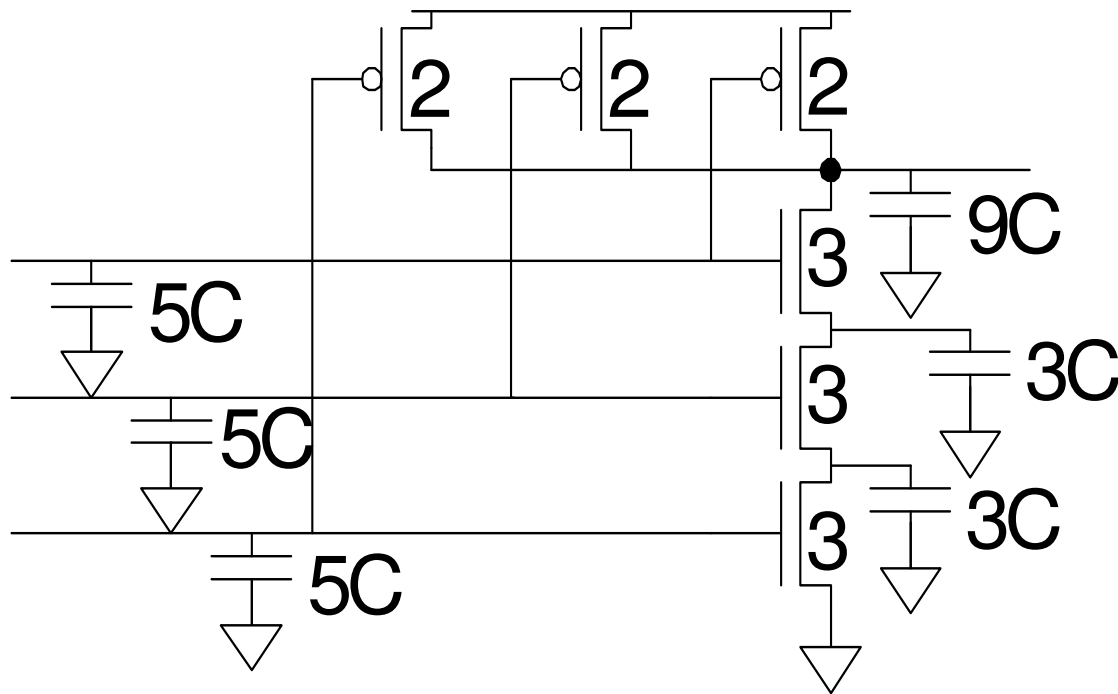
3-input NAND

## Recall

- Unit NMOS has resistance  $R$ , capacitance  $C$
- Unit PMOS has resistance  $2R$ , capacitance  $C$
- $k$  units has capacitance  $kC$ .



# Effective R and C : 3-input NAND Capacitance



Shorted capacitances deleted and remaining capacitances lumped.

3-input NAND

