Delay

Pravin Zode

Outline

- Important Definitions
- Timing Optimizations
- Dynamic Behavior of Inverter
- RC Delay Model
- Delays
  - Parasitic Delay
  - Effort delay (Logical, Electrical effort)

Introduction

- Reducing delay enhances performance and efficiency
- Delay directly impacts the timing behavior, setup and hold requirements, and overall speed of circuits
- Higher performance often leads to increased power consumption

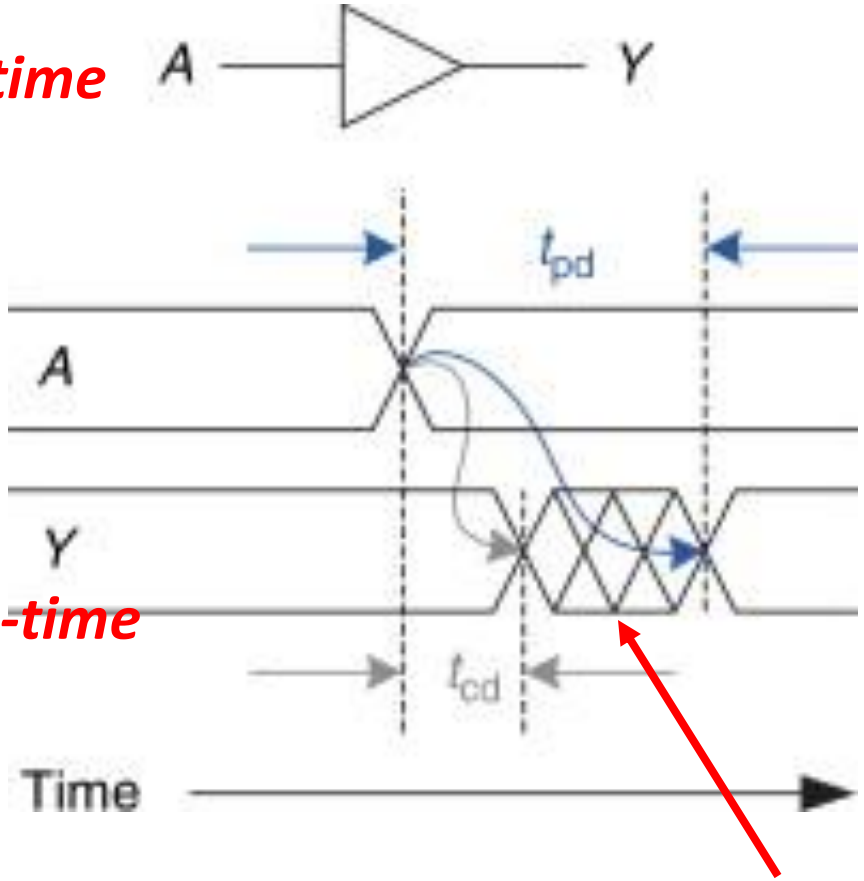
Important Definitions

Propagation delay time ***max-time***

t_{pd} = **maximum** time from the input crossing 50% to the output crossing 50%

Contamination delay time ***min-time***

tcd = minimum time from the input crossing 50% to the output crossing 50%



$t_{cd} <= t_{pd}$

Important Definitions

Rise time

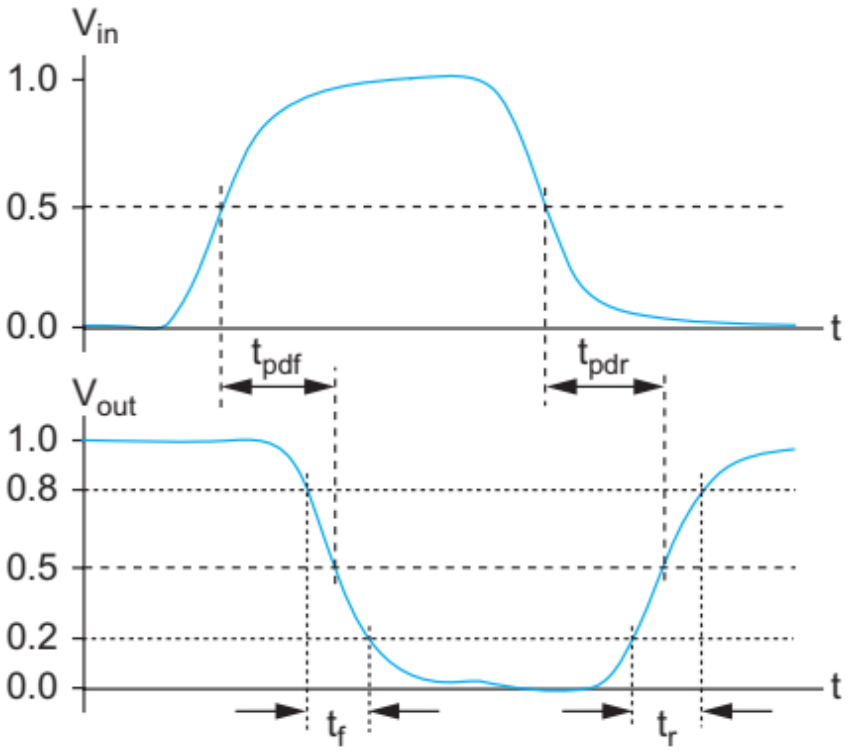
tr = time for a waveform to rise from 20% to 80% of its steady-state value

Fall time

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

Edge rate

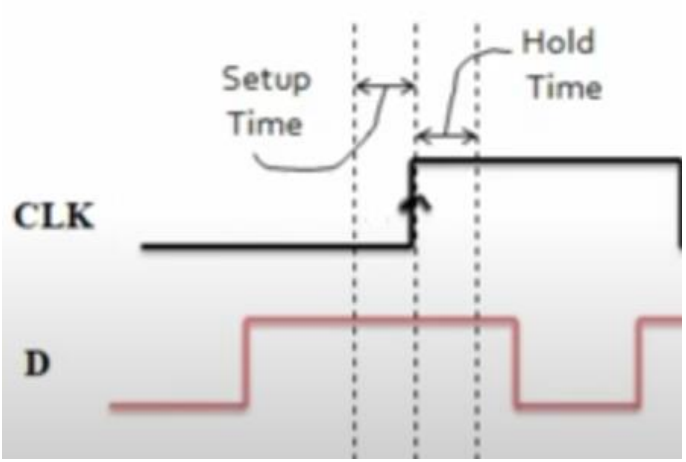
trf = (tr + tf)/2



Important Definitions

Setup time is the minimum time for which data should be stable at the input before the active edge of clock arrives

Hold time is the minimum time for which the data should be stable at the input after the active edge of clock has arrived.

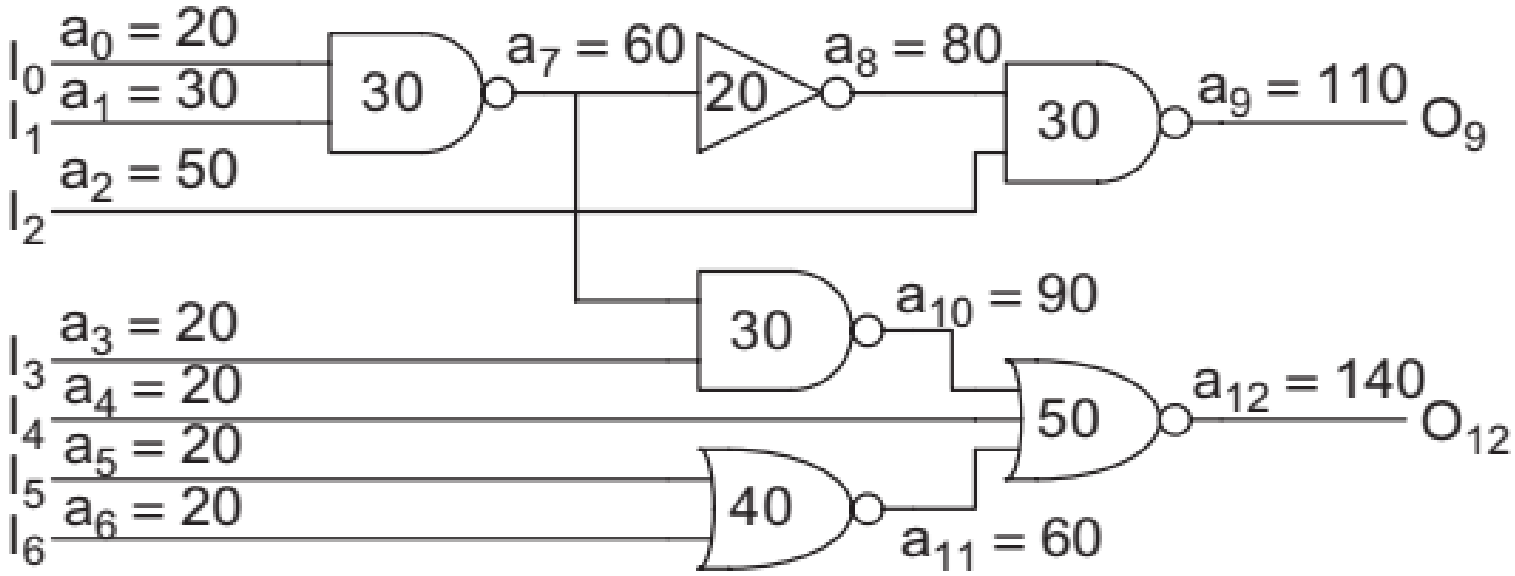


Important Definitions

The slack is the difference between the required and arrival times

Positive slack means that the circuit meets timing

Negative slack means that the circuit is not fast enough



Timing Optimizations

The *critical paths* can be affected at four main levels:

- The architectural/microarchitectural level
- The logic level
- The circuit level
- The layout level

Microarchitectural Level

- This requires a broad knowledge of both the algorithms that implement the function and the technology being targeted
- How many gate delays fit in a clock cycle
- How quickly addition occurs
- How fast memories are accessed
- How long signals take to propagate along a wire

Trade-offs at the microarchitectural level include

- number of pipeline stages,
- number of execution units (parallelism)
- size of memories

Logic Level

- ***Experience-Based:*** Leverage design expertise for selecting gates and structures
- ***Experimentation:*** Test different configurations to achieve minimal delay
- ***Logic Synthesis Tools:*** Automated tools to translate function into gates and registers for optimized performance.

Trade-offs at the logic level include

- Types of functional blocks (e.g., ripple carry vs. lookahead adders)
- Number of stages of gates in the clock cycle
- fan-in and fan-out of the gates

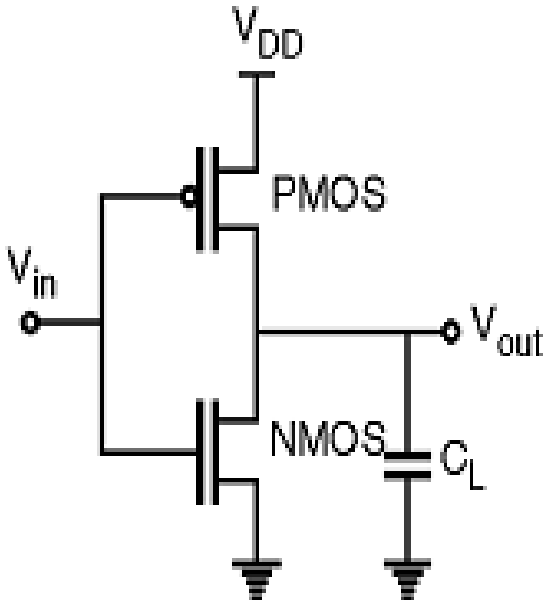
Circuit Level

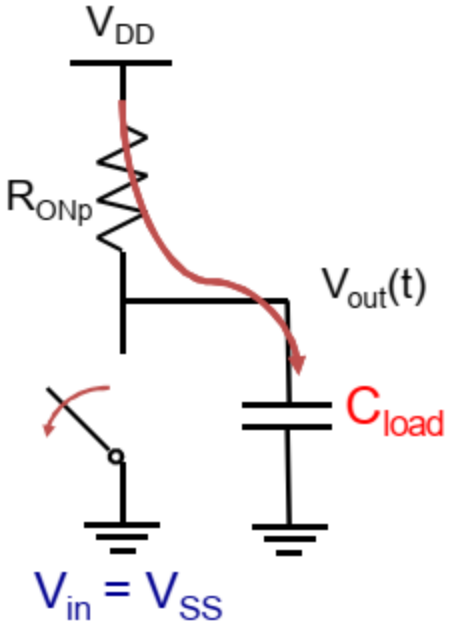
- ***Transistor Sizing :*** Adjusting Widths of transistor balances drive strength and speed
- Larger transistors reduce delay but increase area and power consumption.
- ***Alternative CMOS Logic Styles:*** Dynamic CMOS, Pass-Transistor Logic, Complementary Pass-Transistor Logic (CPL)
- Minimizes the number of transistors and interconnections, potentially lowering delay
- Increases speed by reducing load capacitance on critical paths

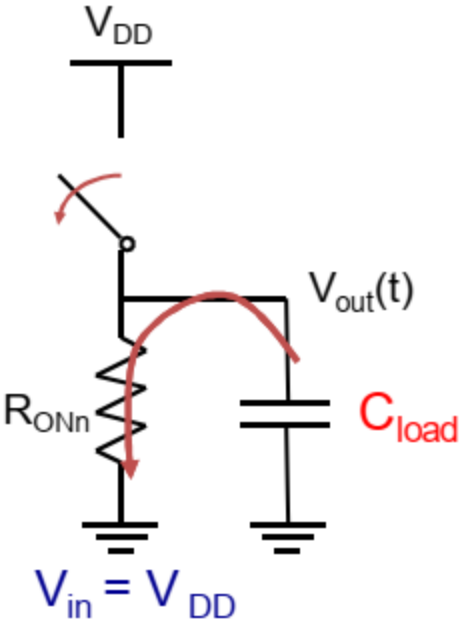
Layout Level

- ***Wire Length Reduction***: Place cells optimally to shorten interconnects, reducing signal propagation delay
- ***Parasitic Capacitance Reduction:*** Optimize cell geometry to minimize capacitance on signal lines
- ***Compact Layouts:*** Reduce spacing within cells to lower resistance and improve performance

Dynamic Behavior of Inverter





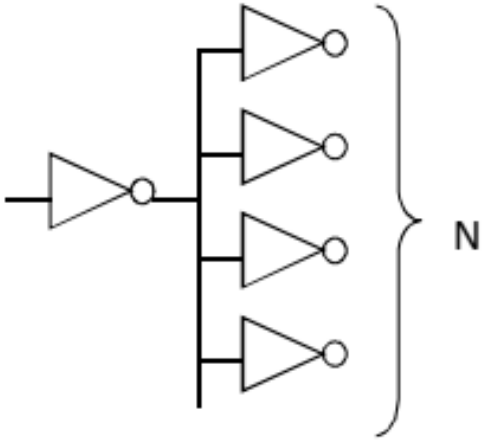


Vout completely depends on $\tau = R\,C_{load}$

Dynamic Behavior of Inverter

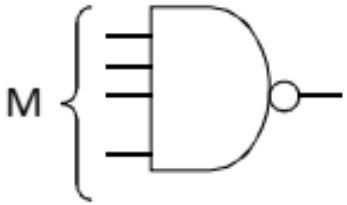
C_{load} depends on fan-in and fan-out

Fan-out: *number of load gates connected to output of driving gate*



All wiring and total input capacitance determine C_{load}

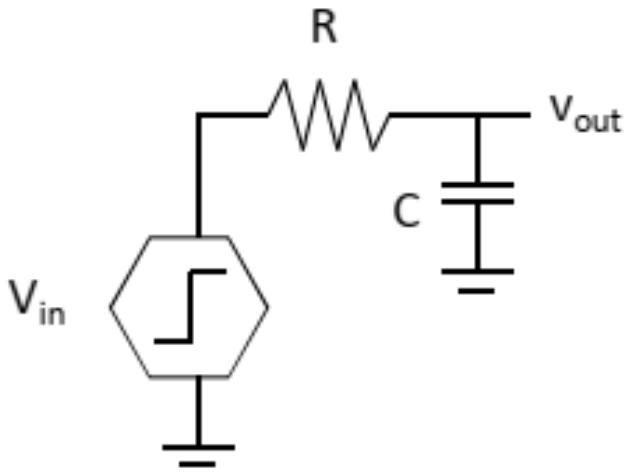
Fan-in: *number of inputs to gate*



RC Model for Dynamic Behavior

- ***First-order RC model***
- ***Step function VSS -> VDD***

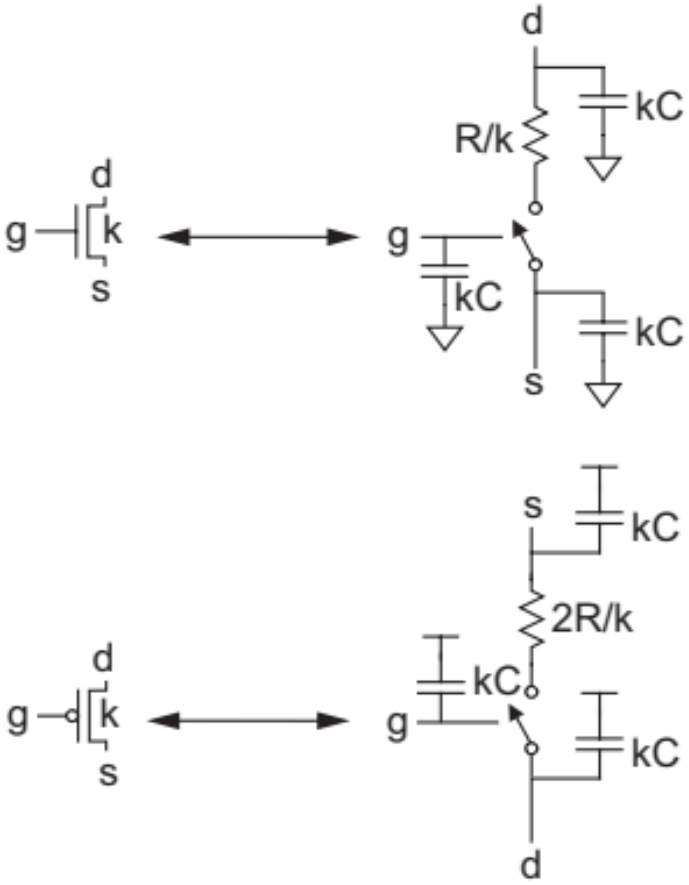
$$I = C\frac{dV}{dt}$$



$$V_{out}(t) = V_{dd}(1 - e^{-t/\tau}) \qquad \tau = RC$$

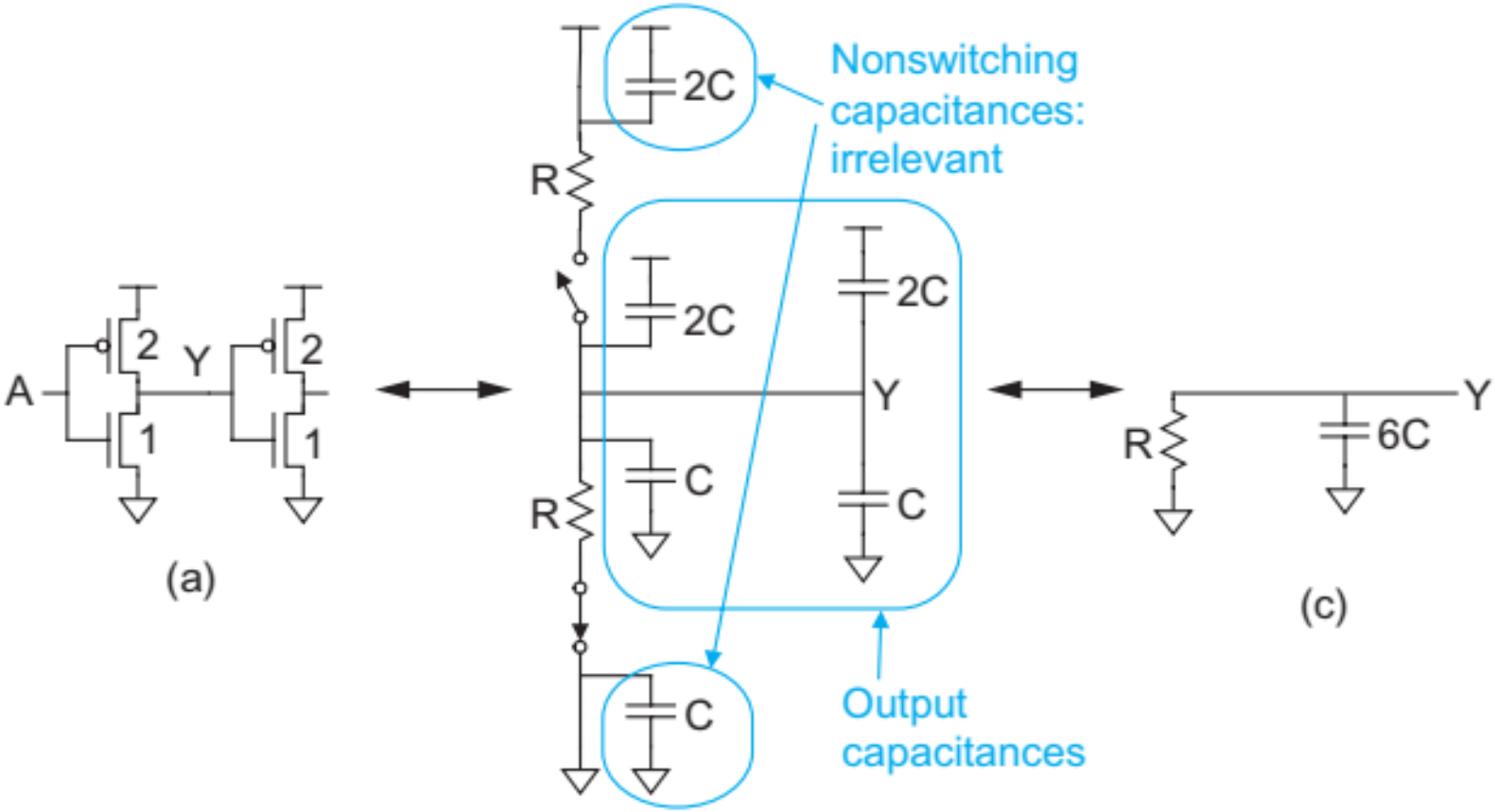
RC Delay Model

- 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 proportional to width
- Resistance inversely proportional to width



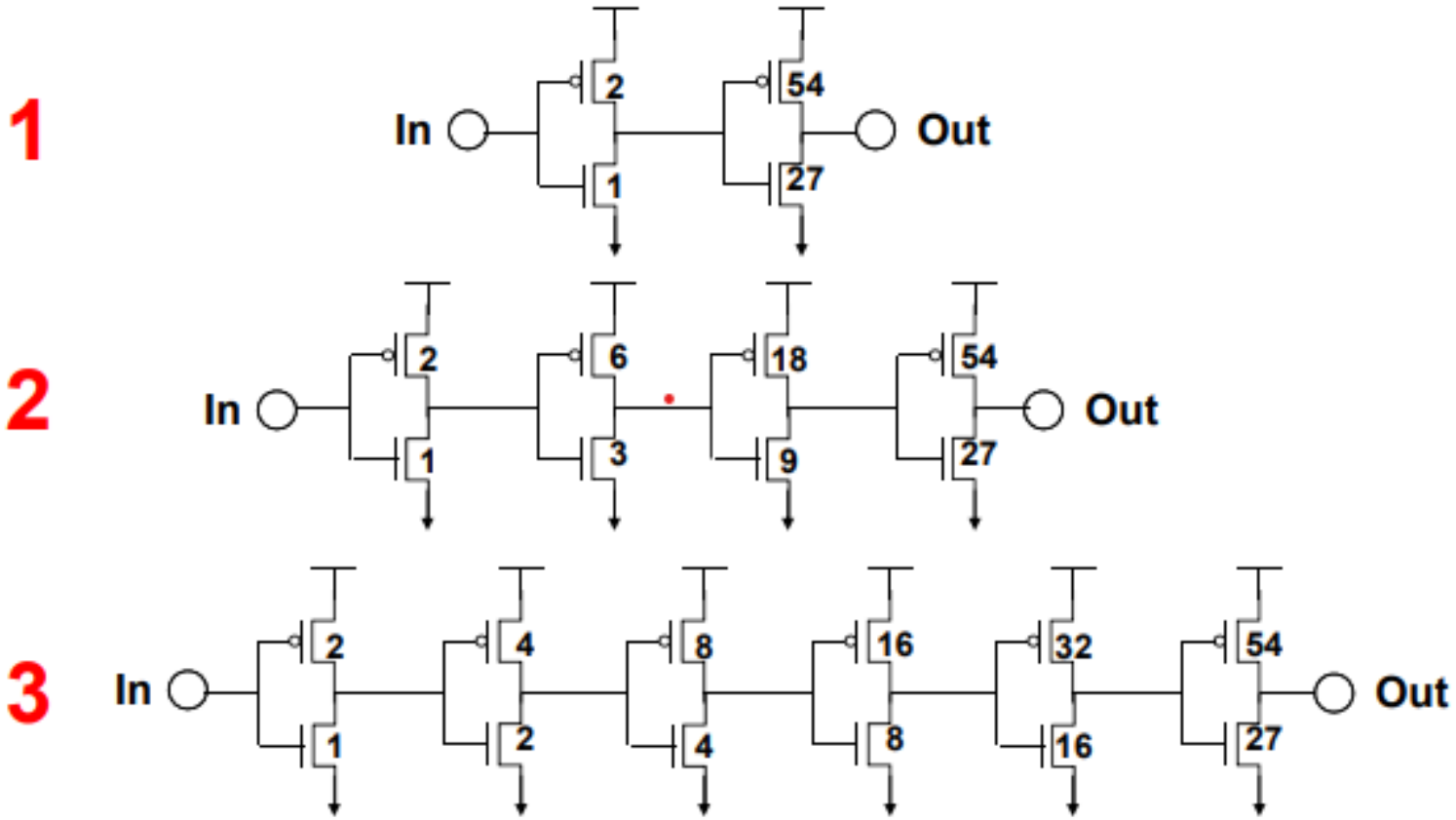
RC Delay Model

Estimate the delay of a fanout-of-1 inverter



Exercise : RC Delay Model

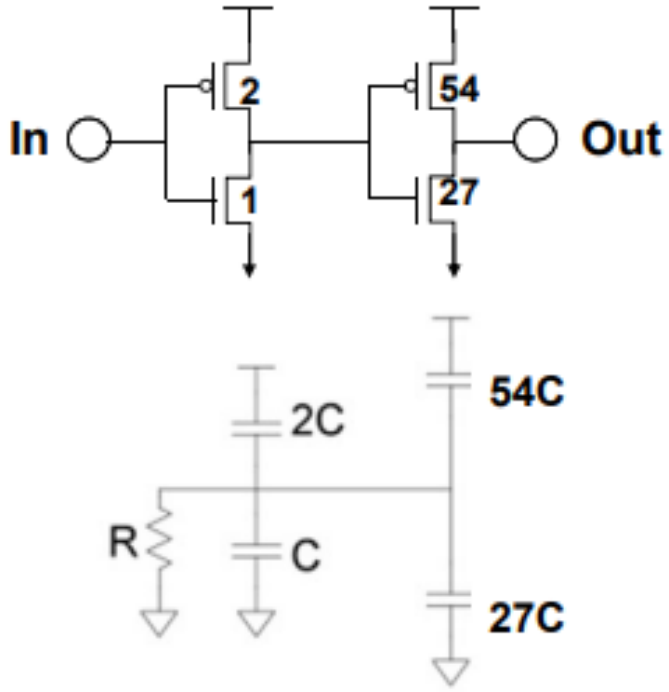
Compare three delay cases



Is Case 1, Case 2 or Case 3 Faster?

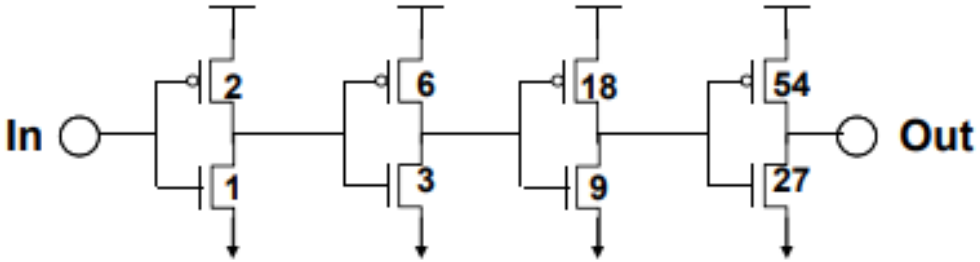
Case 1 :

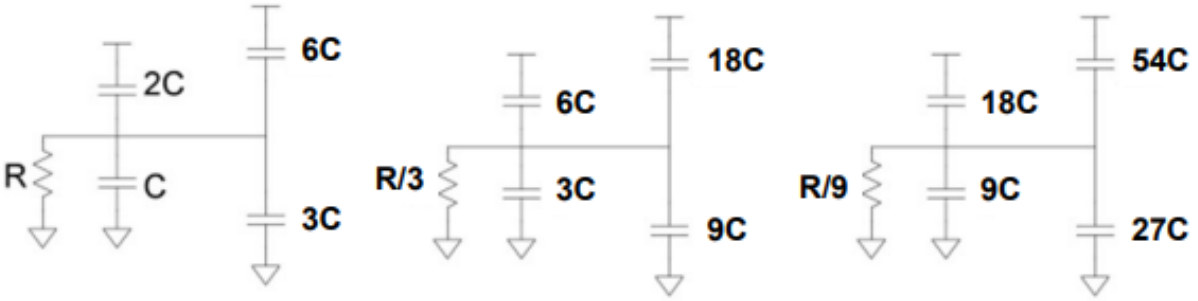
Compare three delay cases



d = 84 RC

Case 2 :



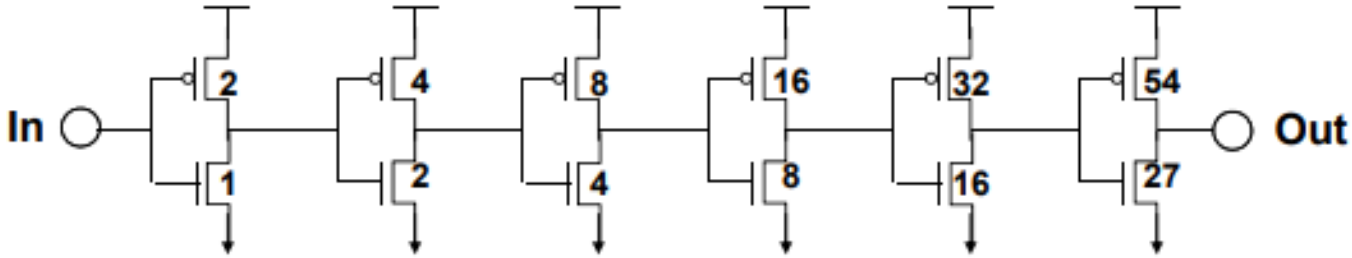


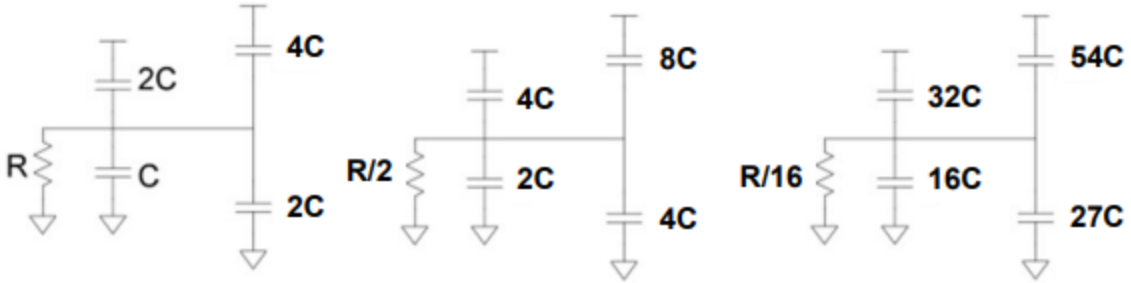
d1 = 12 RC **d2 = 36/3 RC = 12 RC** **d2 = 108/9 RC = 12 RC**

d = d1 + d2 + d3 = 36 RC << 84 RC

- **Note the geometric progression in size!**
  - **3X per stage.**
- **The delay for each stage is the same**

Case 3 :





d1 = 9 RC

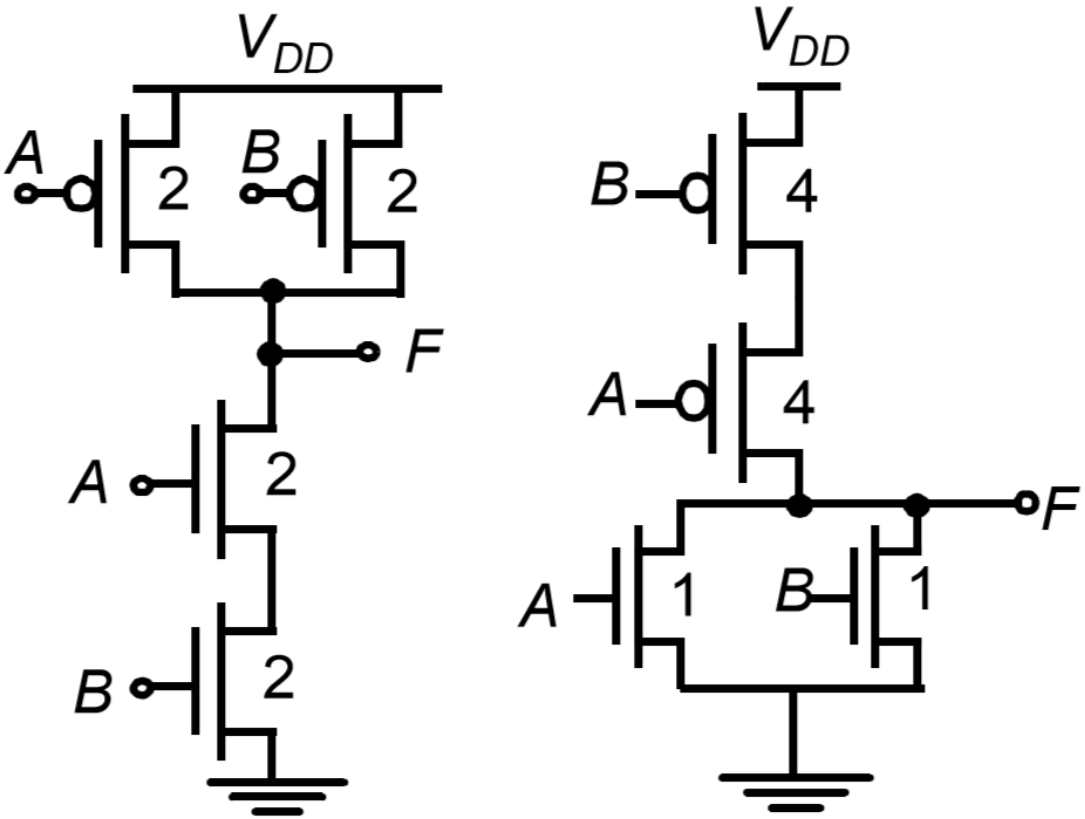
d2 = 9RC
d3 = 9RC
d4 = 9RC

d5 = 129/16 RC = 8.1 RC

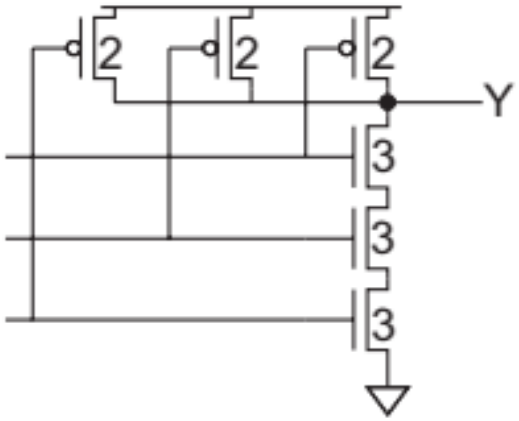
d = d1 + d2 + d3 + d4 + d5 = 44.1 RC

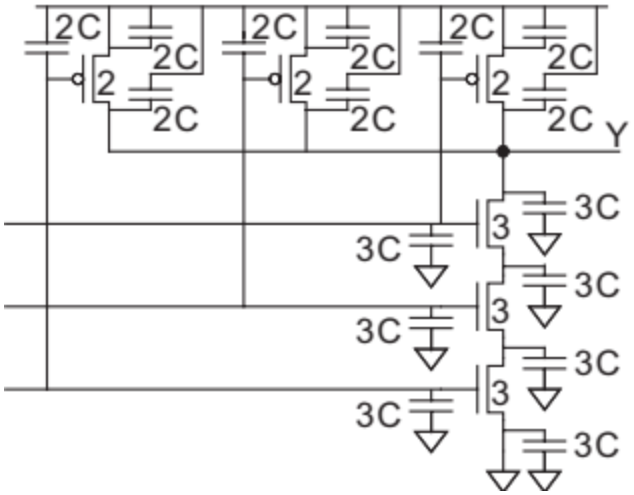
- **Case 2 = 36 RC < Case 3 = 44 RC < Case 1 = 84 RC**
- **You can have too much of a good thing!**

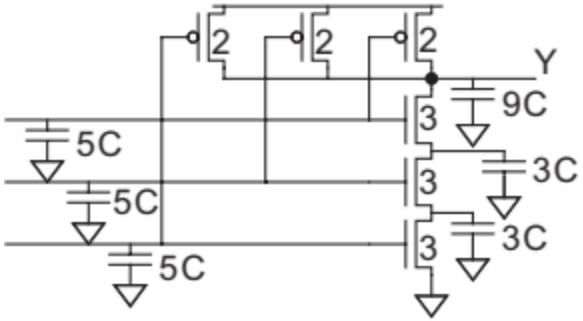
Transistor Sizing

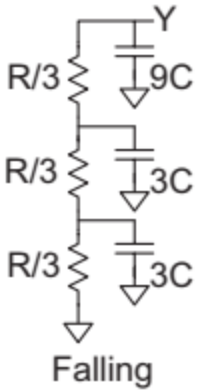


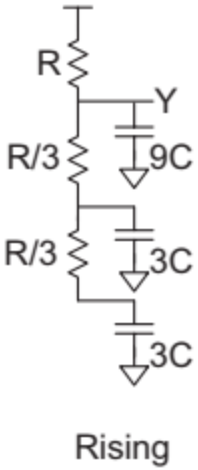
RC Delay Model 3-Input NAND Gate











Elmore Delay Model

- Most circuits can be represented as an RC tree (with no loops)
- The root of the tree is the voltage source and the leaves are the capacitors at the ends of the branches.
- The Elmore delay model estimates the delay from a source switching to one of the leaf nodes changing as the sum over each node of the capacitance and multiplied by the effective resistance R

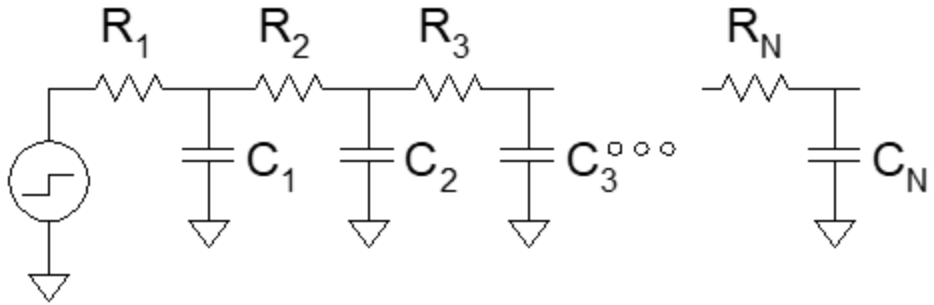
$$t_{pd} = \sum_i R_{is} C_i$$

Elmore Delay Model

- ON transistors look like resistors
- Pullup or pulldown network modeled as RC ladder
- Elmore delay of RC ladder

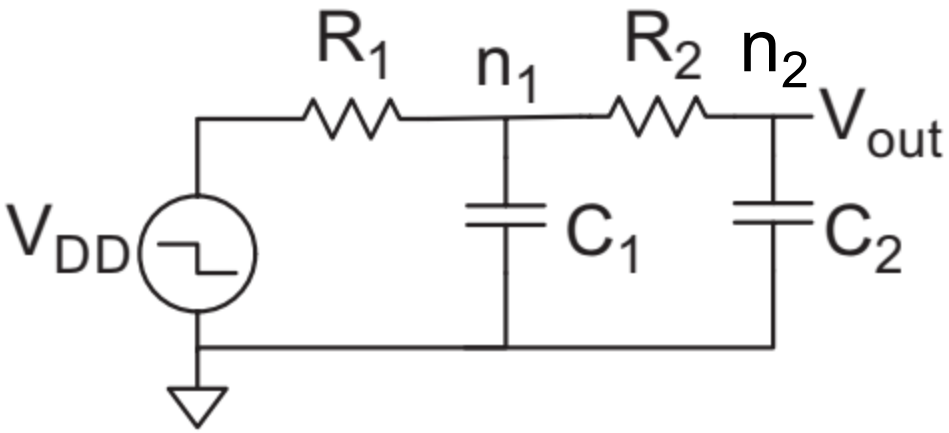
$$t_{pd} \approx \sum_{\text{nodes } i} R_{i\text{-to-source}} C_i$$

$$= R_1 C_1 + (R_1 + R_2) C_2 + \ldots + (R_1 + R_2 + \ldots + R_N) C_N$$



Example -01

- Compute the Elmore delay for Vout in 2nd order system

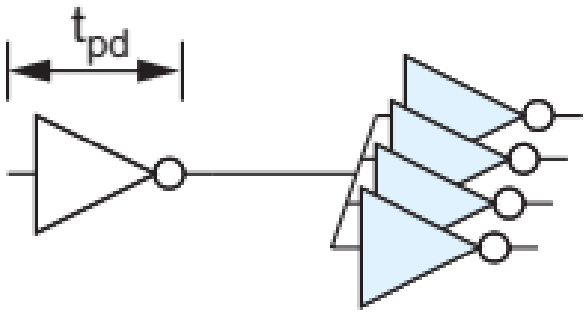


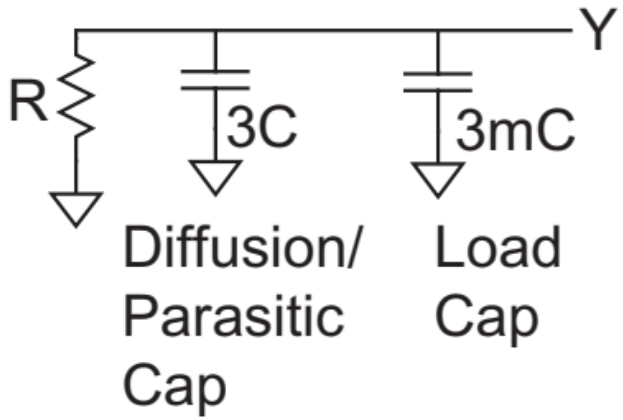
$$tpd = R_{1s}C_1 + R_{2s}C_2$$

$$= R_1C_1 + (R_1+R_2)\,C_2$$

Example -02

- Estimate tpd for a unit inverter driving m identical unit inverters



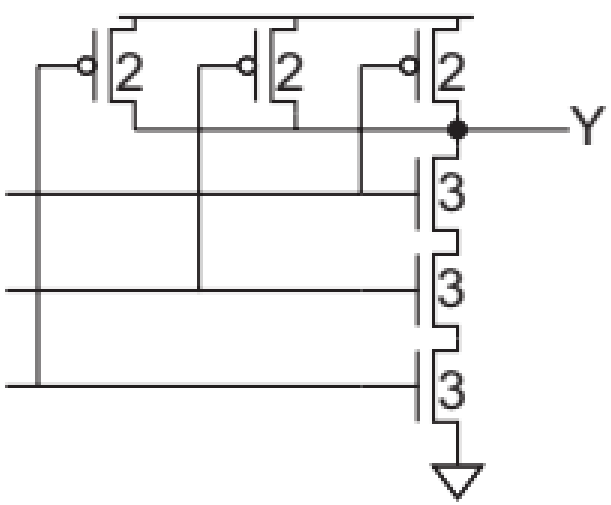


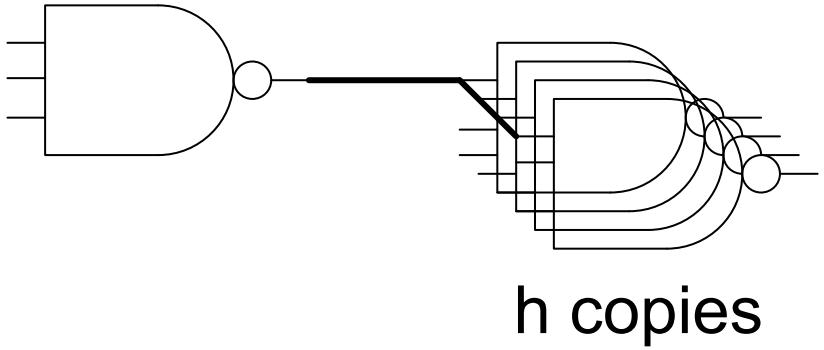
$$tpd = \text{R*3C + R*3mC}$$

$$= (3+3m)RC$$

Example -03

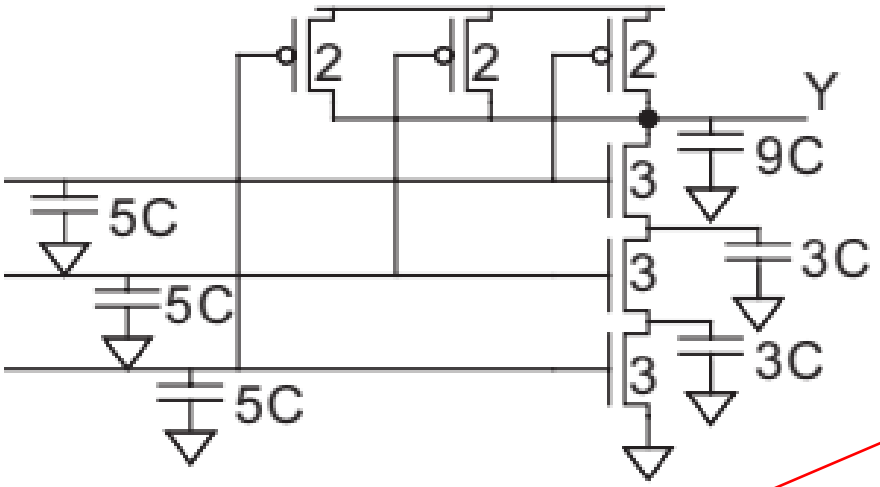
- Estimate t_{pdf} and t_{pdr} for the 3-input NAND gate, if the output is loaded with *h* identical NAND gates

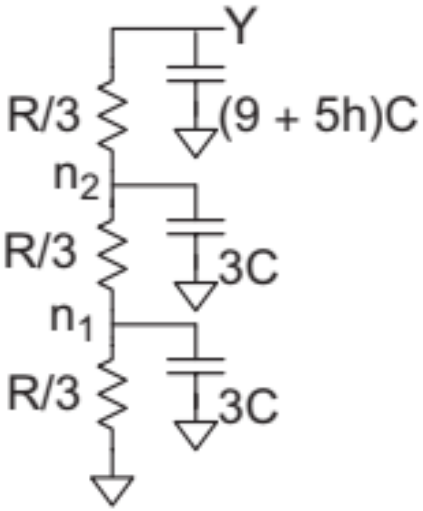


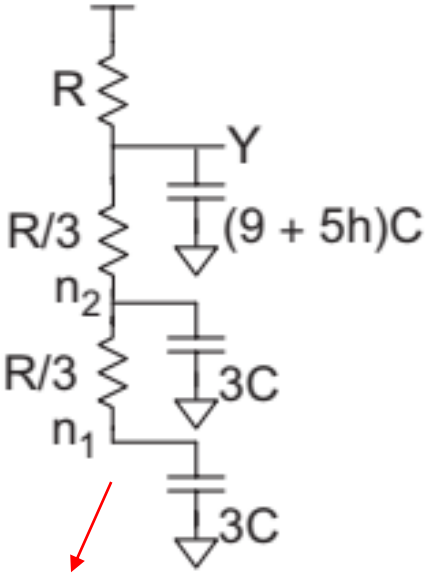


Example -03

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







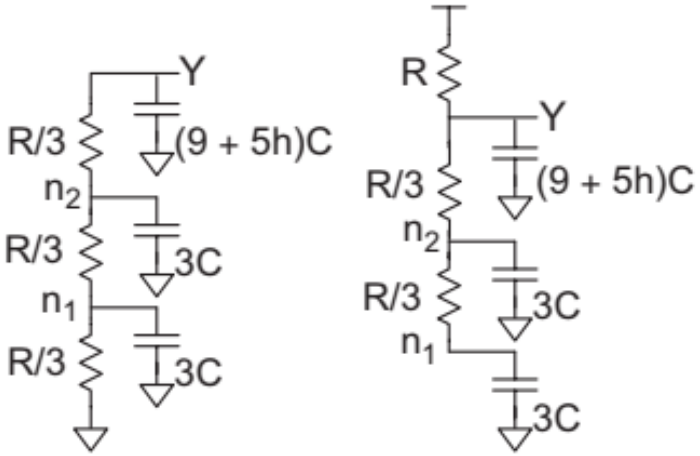
$$t_{pdr} = (9+5h)RC$$

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

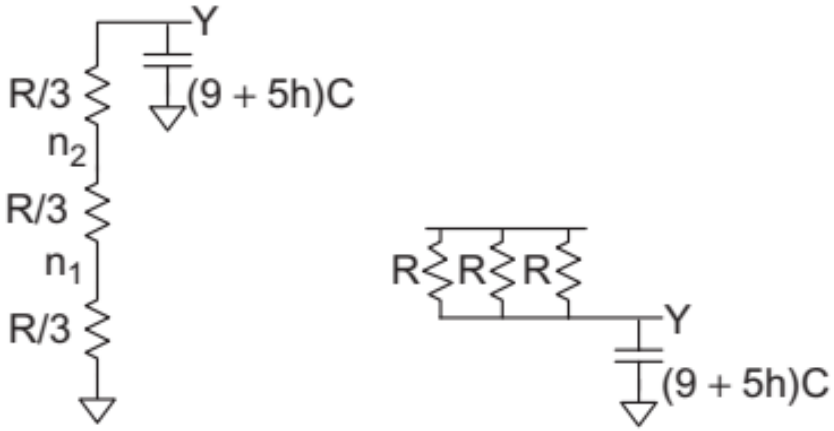
$$= (12+5h)RC$$

Example -04 Contamination Delay

- Estimate t_{cdf} and t_{cdr} (Contamination Delay) for the 3-input NAND gate, if the output is loaded with *h* identical NAND gates



Propagation Delay



Contamination Delay

Delay Components

$$t_{pd} = t_p + t_f$$
$$= t_p + (gh)$$

Delay has two parts

- *Parasitic delay*
  - Independent of load
  - Intrinsic delay due to transistor and wire capacitances.
- *Effort delay*
  - Proportional to load capacitance
  - Load-dependent delay, also known as stage effort or fan-out delay

Parasitic Delay

- Parasitic Delay (tp) is the intrinsic delay of a gate when driving no external load
- Sources of Parasitic Delay
  - **Internal Capacitance:** Capacitance at the drain and source of transistors in the gate.
  - **Wire Capacitance:** Capacitance of the interconnects or wires connected to the output.
  - **Gate Capacitance:** Internal to the gate, adding to the delay regardless of load.

Independent of the load, but depends on the technology, layout, and intrinsic properties of the transistors

Calculation of Parasitic Delay

- Formula for Parasitic Delay:
  - $tp = R \times Cparasitic$

  Where,

  R is the resistance of the transistor's output path.

  $Cparasitic$ parasitic includes internal capacitances of the gate

- Example:

For a CMOS inverter with a drain capacitance of $3C$ and a resistance R

the parasitic delay is: $tp = R \times 3C$

Parasitic Delay

Parasitic delay of common gate

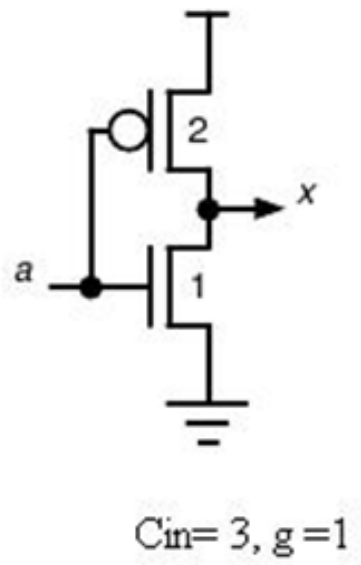
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate / mux	2	4	6	8	2n
XOR, XNOR		4	6	8	

Effort (Stage) Delay

- Effort Delay (t_f), or stage effort, is the load-dependent delay of a gate
- Components of Effort Delay:
  - Logical Effort (g): Measures the gate's drive strength relative to an inverter.
  - Electrical Effort (Fan-out, h): Ratio of the load capacitance to the input capacitance of the gate.
- Formula for Effort Delay:

$$t_f = g \cdot h$$

Logical Effort

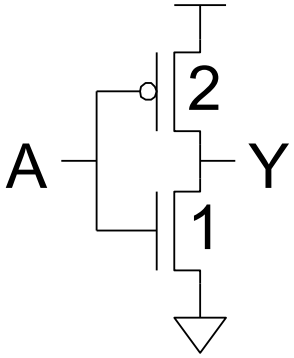


Cin= 3, g =1

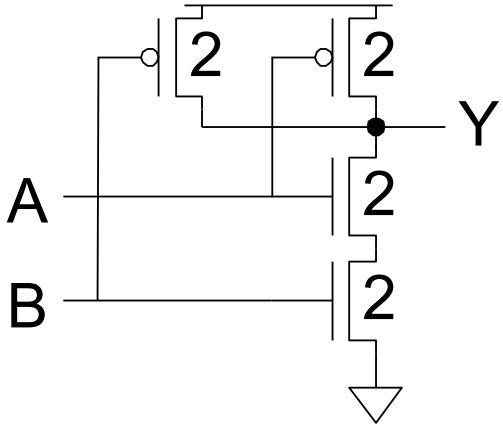
- A metric to compare the speed of CMOS gates relative to an inverter
- Helps estimate delay and optimize gate sizing
- Logical effort (g) quantifies how much slower a gate is compared to an ideal inverter when driving the same load
- Inverters have a logical effort of g=1, which is considered the baseline
- Other gates have g>1, meaning they are slower than inverters.

$$g = \frac{Input\ Capacitance\ of\ Gate}{Input\ Capacitance\ of\ Inverter}$$

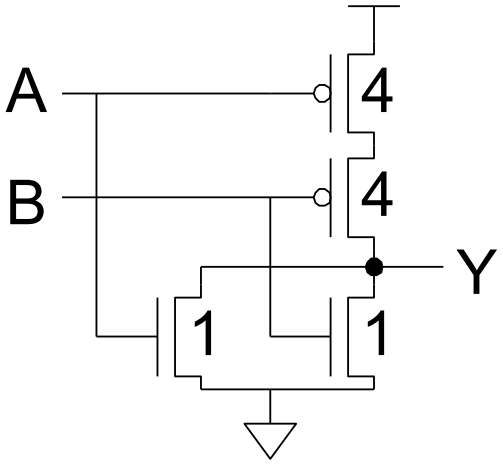
Logical Effort



C_{in} = 3
g = 3/3



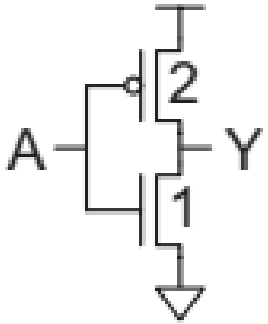
C_{in} = 4
g = 4/3



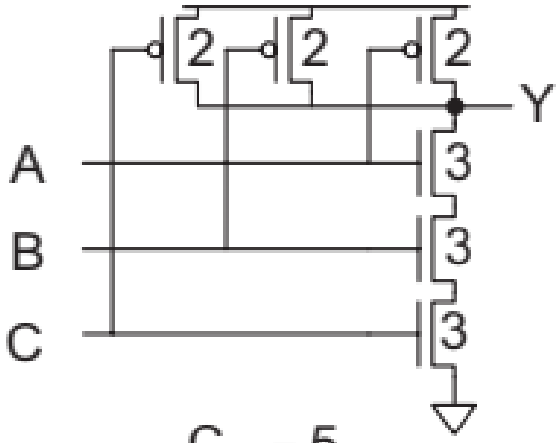
C_{in} = 5
g = 5/3

$$g = \frac{Input\ Capacitance\ of\ Gate}{Input\ Capacitance\ of\ Inverter}$$

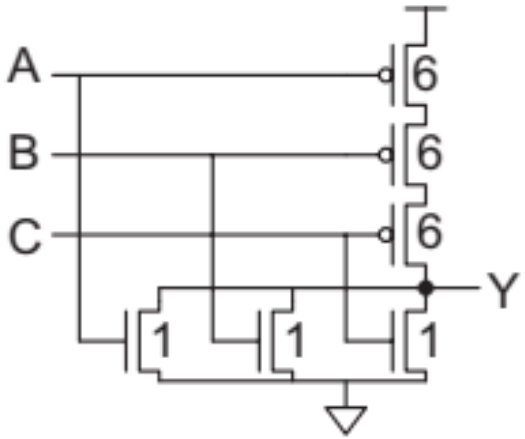
Logical Effort



$C_{in} = 3$
$g = 3/3$



$C_{in} = 5$
$g = 5/3$



$C_{in} = 7$
$g = 7/3$

$$g = \frac{Input\ Capacitance\ of\ Gate}{Input\ Capacitance\ of\ Inverter}$$

Logical Effort of Common Gates

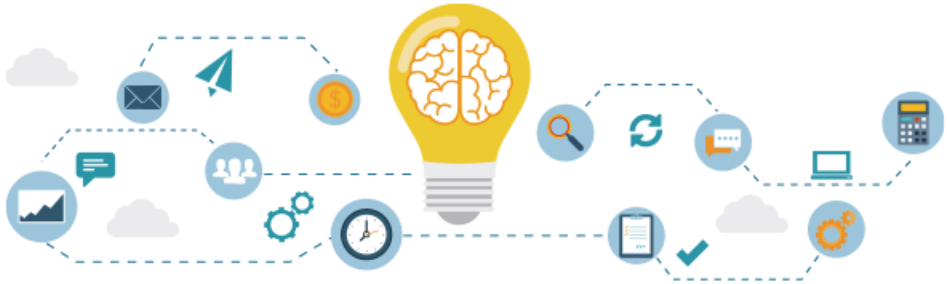
Logical effort of common gates

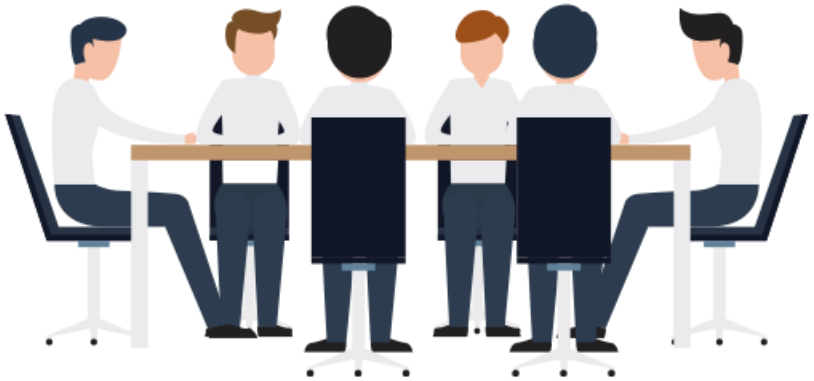
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
Tristate / mux	2	2	2	2	2
XOR, XNOR		4, 4	6, 12, 6	8, 16, 16, 8	

Electrical Effort

- Electrical effort (often denoted as h) is a concept used to quantify the load a gate must drive relative to its own input capacitance
- It helps designers understand how to size transistors and balance delays in multi-stage logic paths
- It is used to optimize the delay of CMOS circuits

$$h = \frac{Load\ Capacitance\ of\ Gate(Cload)}{Input\ Capacitance\ of\ Gate(Cin)}$$





Thank you !

Happy Learning