EECS151 : Introduction to Digital Design and ICs

Lecture 13 – Delay

Bora Nikolić and Sophia Shao

Reminiscences of the VLSI Revolution

I introduced the new chip design methods in a special VLSI design course at MIT in 1978, following the 'script' Charles Steinmetz used to propagate his revolutionary AC electricity methods at Union College back in 1912.

THE M.I.T. 1978
VLSI SYSTEM DESIGN COURSE

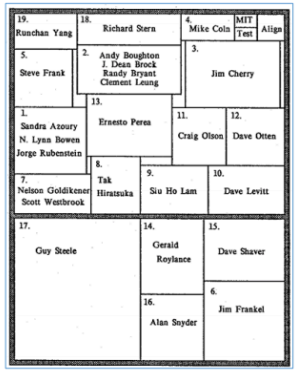
by Lynn Conway

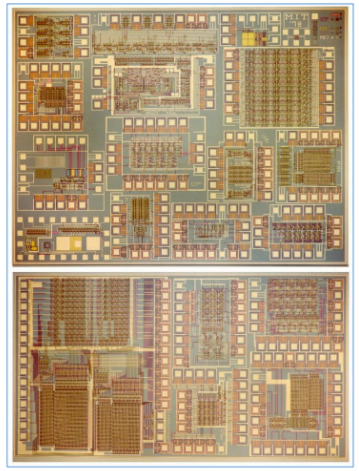
Link
Link
Link

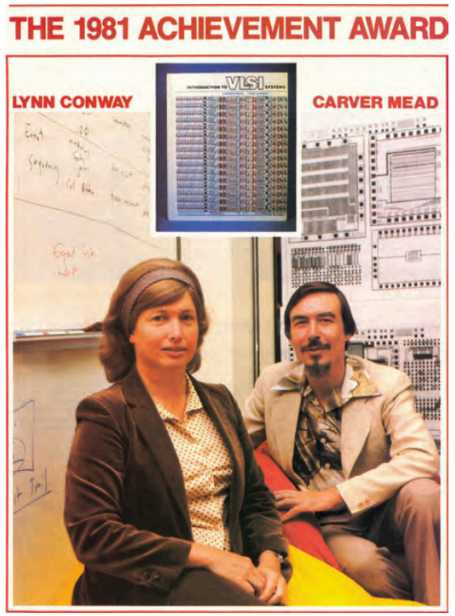
[Update: 11-14-07]

This course was an important milestone in the development, demonstration and evaluation of the Mead-Conway structured VLSI design methods. Lynn Conway conceptualized and planned the course during the late spring and summer of '78, and taught the course while serving as Visiting Associate Professor of EECS at MIT in the fall of '78 and early '79.

Map and photomicrograph of the 19 student projects on the MIT'78 'MultiProject' Chip







http://ai.eecs.umich.edu/people/conway/Memoirs/VLSI/Lynn_Conway_VLSI_Reminiscences.pdf

Review

- CMOS allows for convenient switch level abstraction
- CMOS pull-up and pull-down networks are complementary
- Transistor sizing affects gate performance

CMOS Delay

Overview
Transistor Size
RC Delay
Logical Effort

CMOS Delay

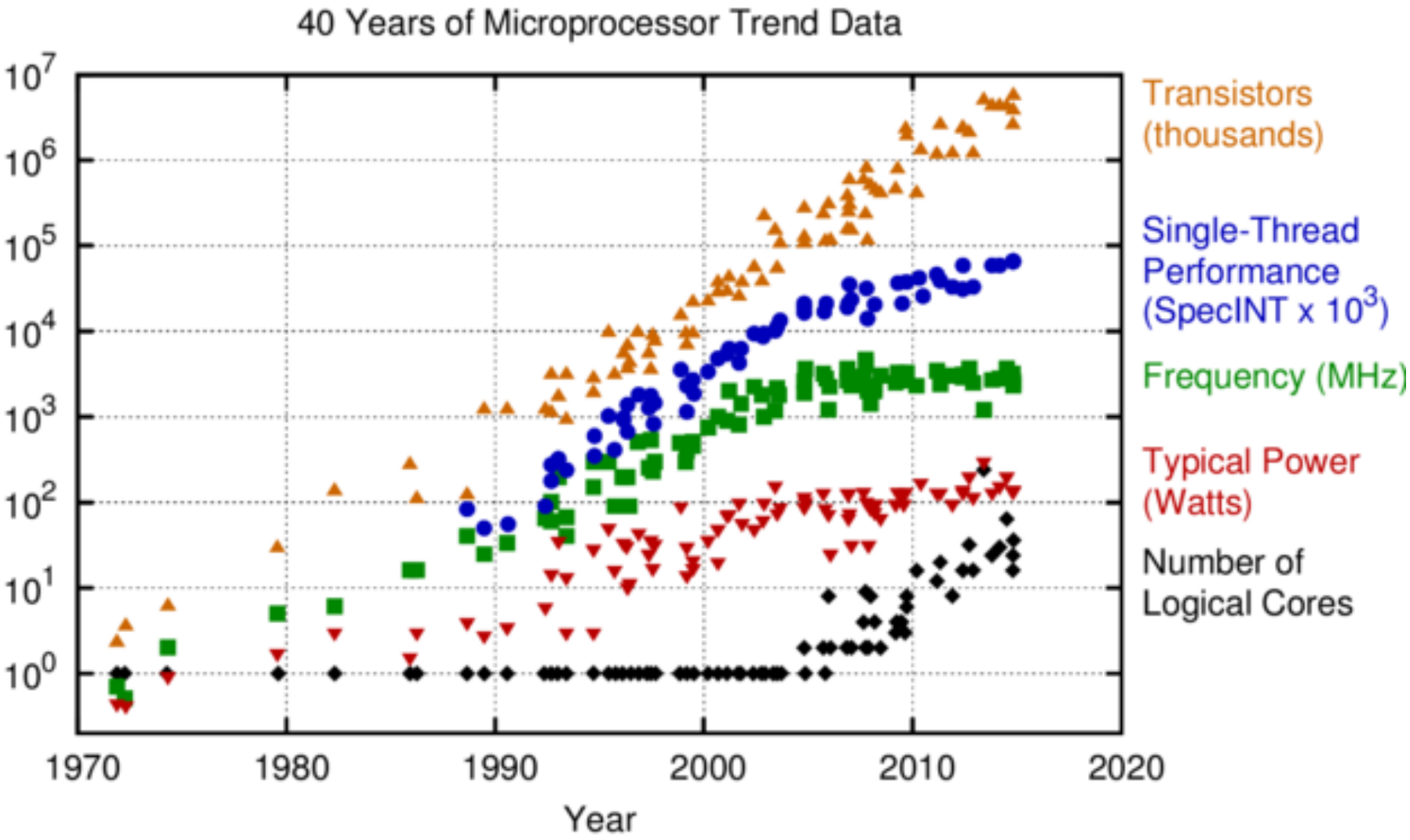
Overview

Transistor Size

RC Delay

Logical Effort

Processor Frequency Scaling



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Delay Optimization

- Critical paths limit the operating speed of the system.
- Four main levels:
  - Architectural/Microarchitectural Level, e.g., # of pipeline stages
  - Logic Level, e.g., types of functional blocks
  - Circuit Level, e.g., transistor sizings
  - Layout Level, e.g., floorplanning
- This lecture: using simple models that offer designers intuitions on logic and circuit optimizations.
  - RC delay model: transistor -> resistor + capacitor
  - Logical effort: further simplifies into a linear model

CMOS Delay

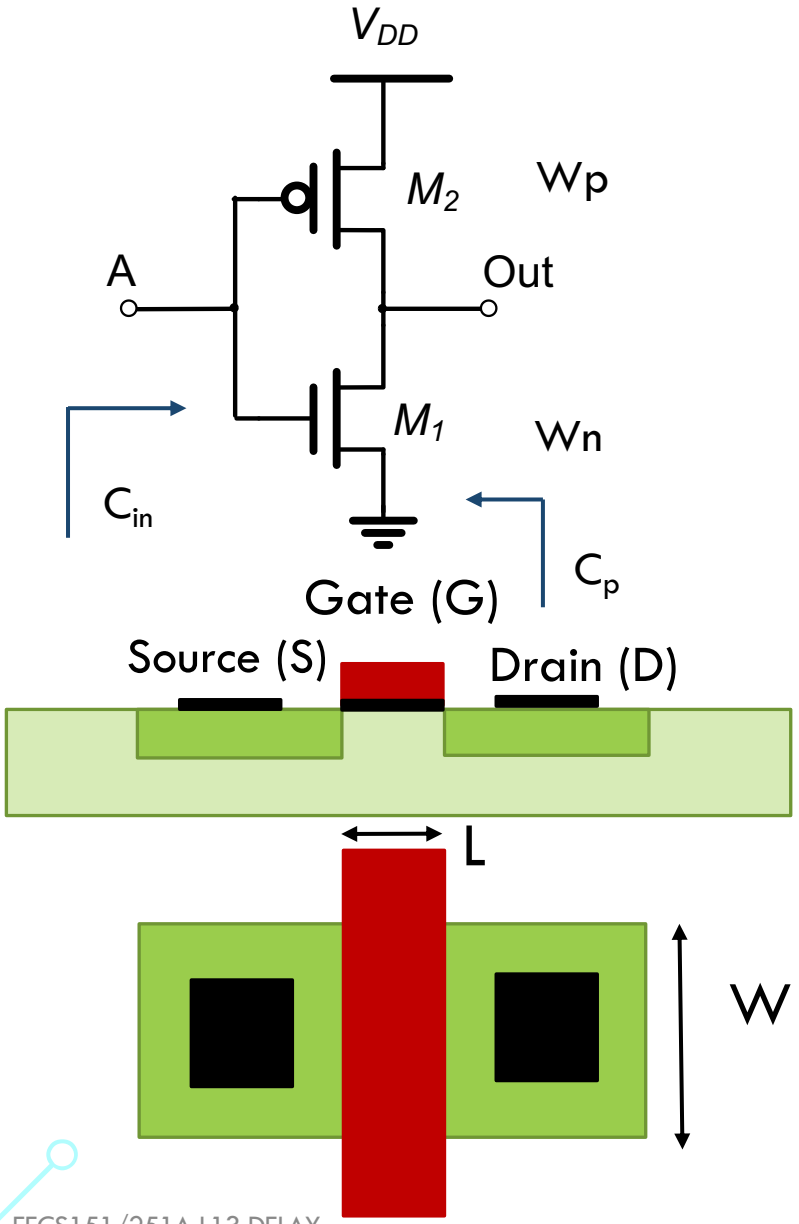
Overview

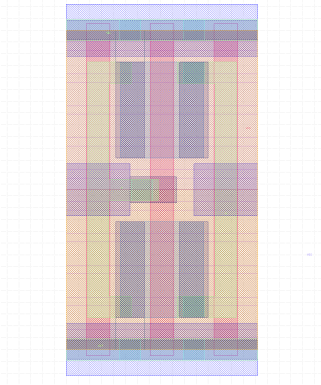
Transistor Size

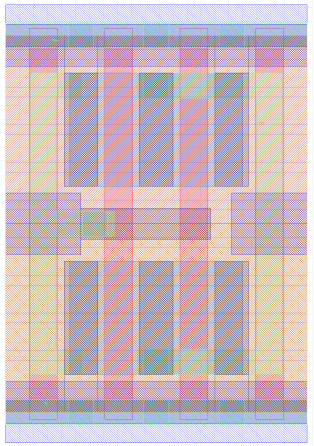
RC Delay

Logical Effort

Capacitances



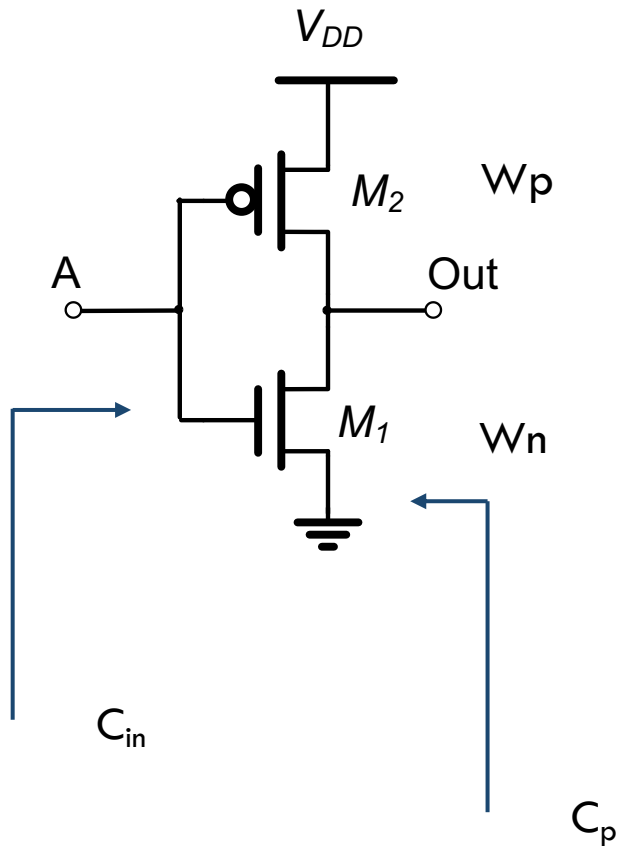




- C_{in} is largely set by the gate cap
  - ~WL
  - 2xW = 2xC_{in}
  - It is non-linear, but we will ignore that
- C_p is largely set by the drain cap
  - ~W (drain area/perimeter)
  - 2xW = 2xC_p
- $C_p = \gamma C_{in}$

Transistor Sizing

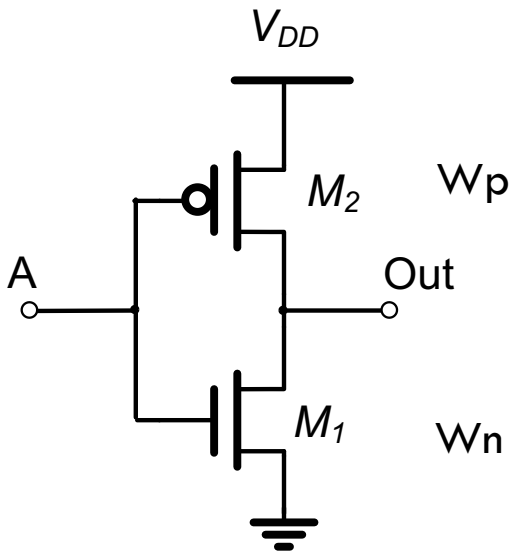
- Doubling the gate size (by doubling Ws):

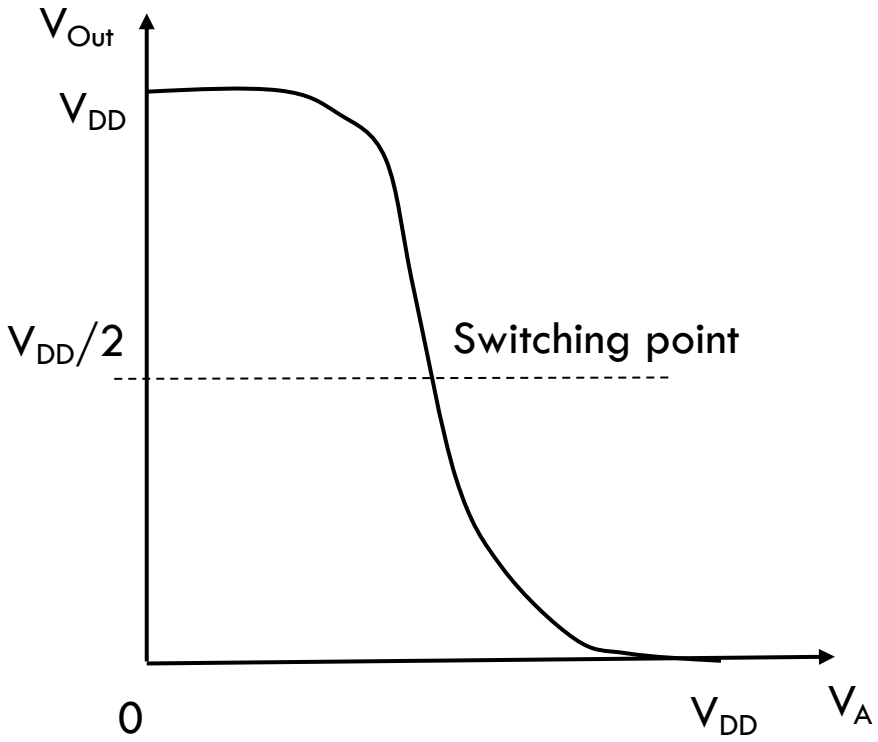


- Doubles C_{in}
- Doubles C_p
- Halves equivalent gate resistance
  - Allows more current to flow

Transistor Sizing

- Optimal Wp/Wn





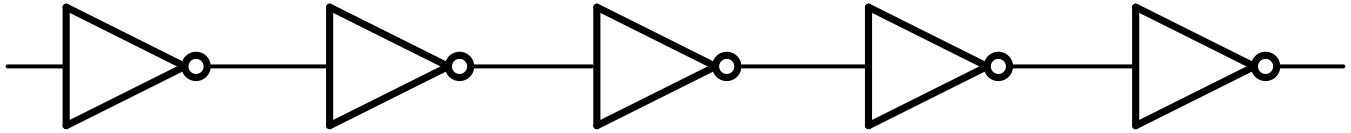
- In the past, Wp > Wn (see Rabaey, 2nd ed)
- In modern processes (finFET), Wp = Wn

CMOS Delay

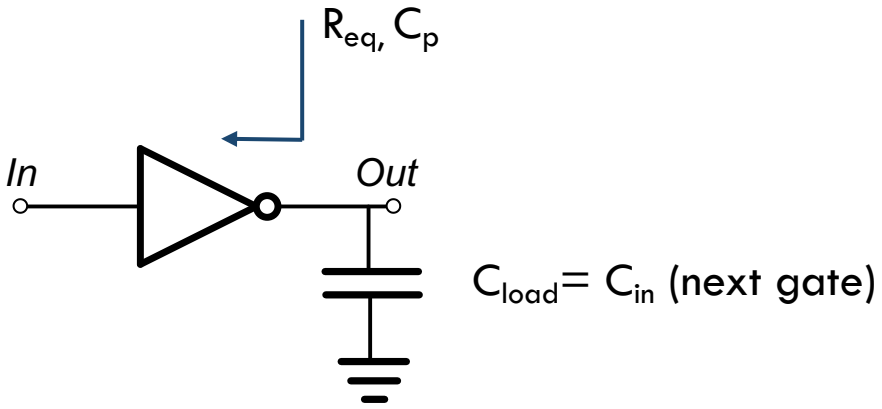
Overview
Transistor Size
RC Delay
Logical Effort

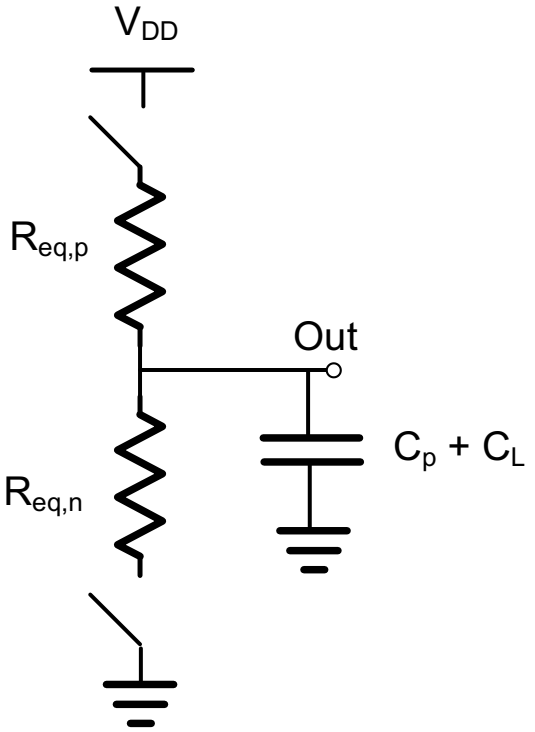
RC Delay

- How to time this?

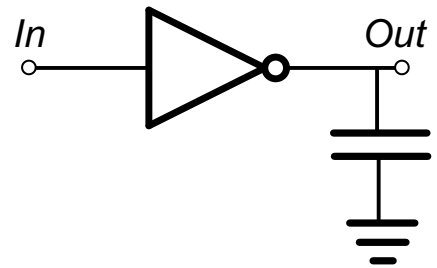


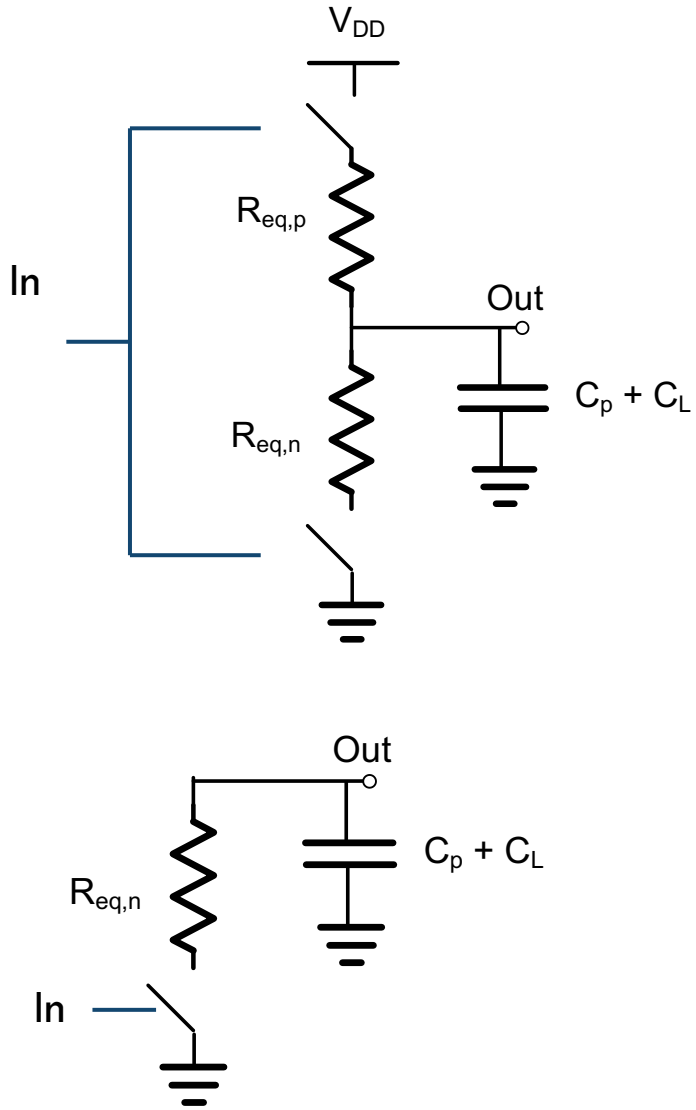
- Each gate has an R_{eq} and drives C_{in} of the next gate

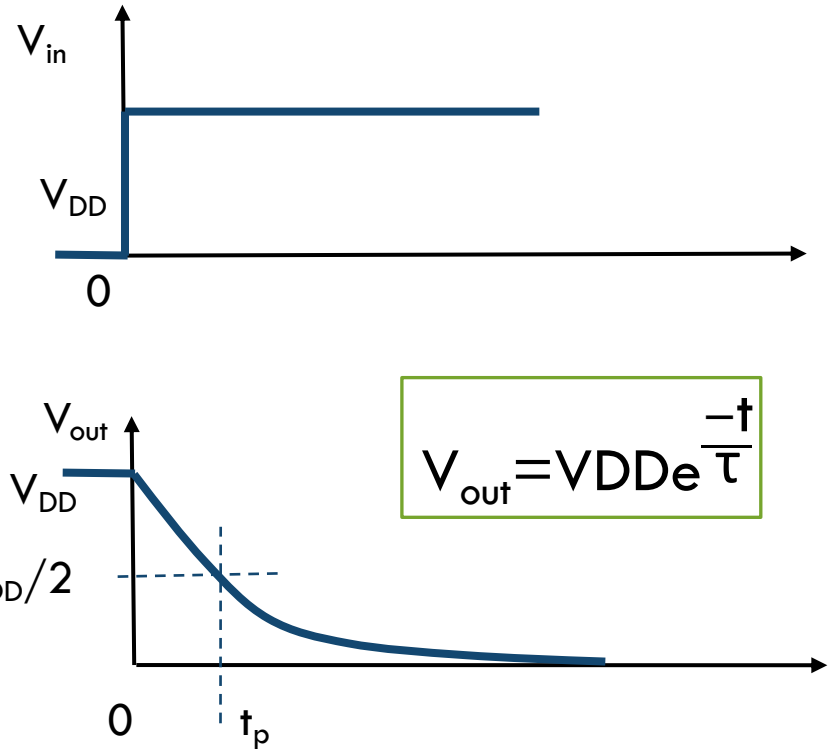




Inverter Delay: High-to-low





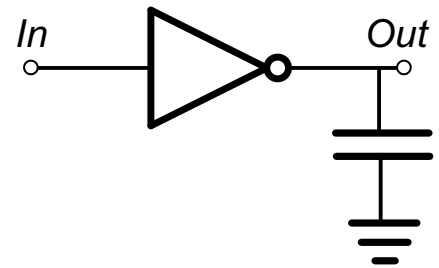


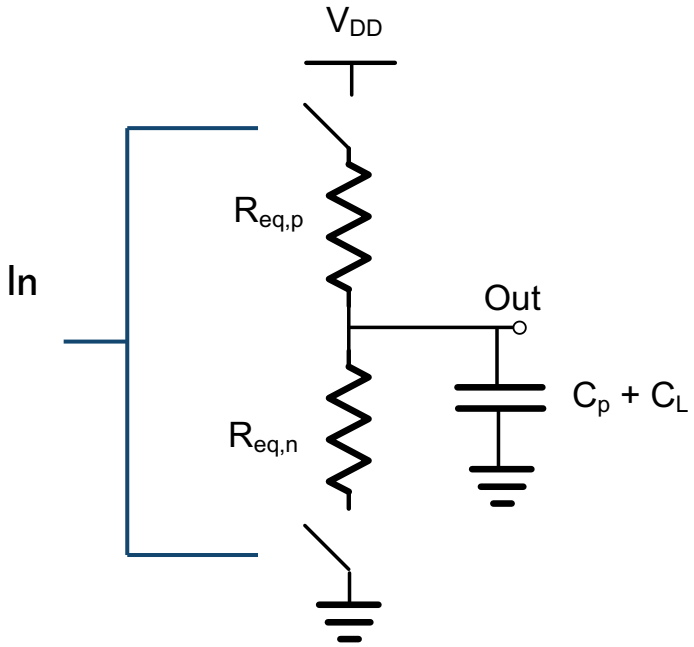
$$V_{out} = VDDe^{\frac{-t}{\tau}}$$

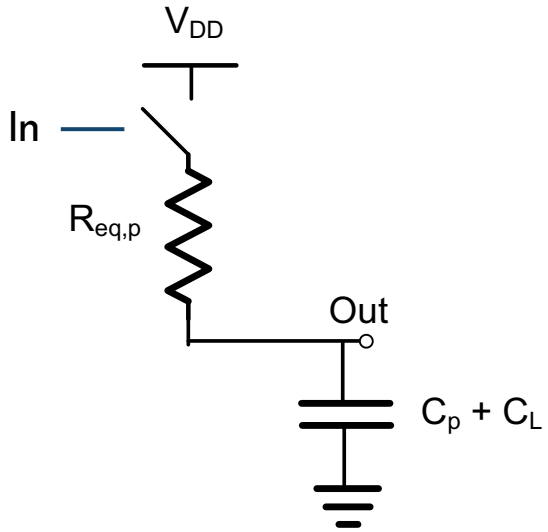
$$t_{p,HL} = (\ln 2)\tau = 0.7\ Req_{,n}(C_p + C_L)$$

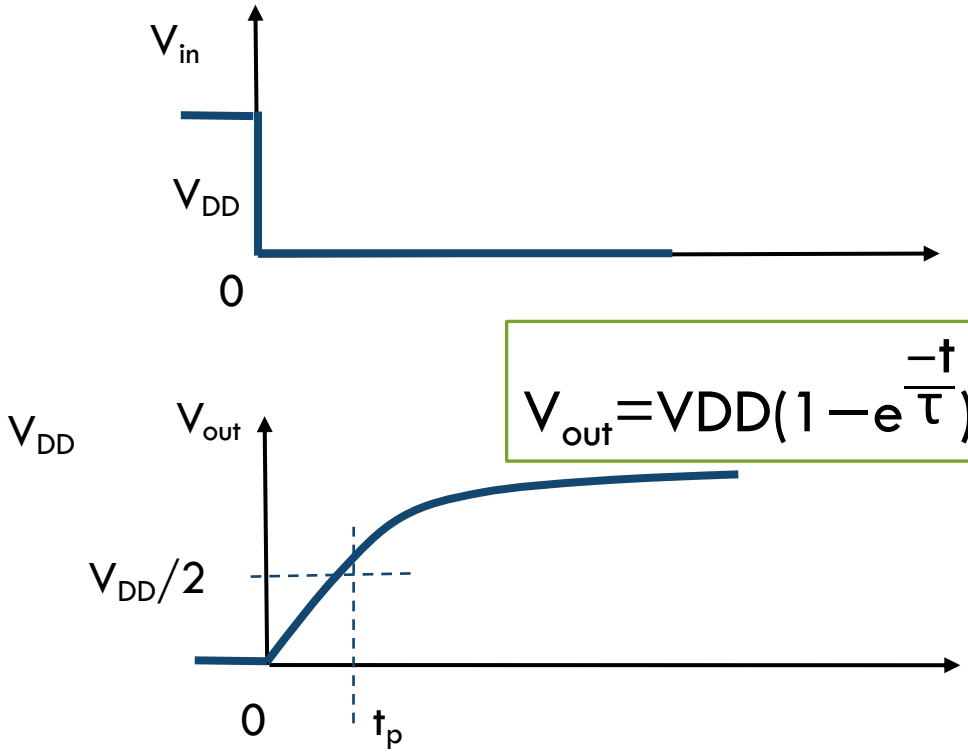
$$\tau = Req_{,n}(C_p + C_L)$$

Inverter Delay: Low-to-high





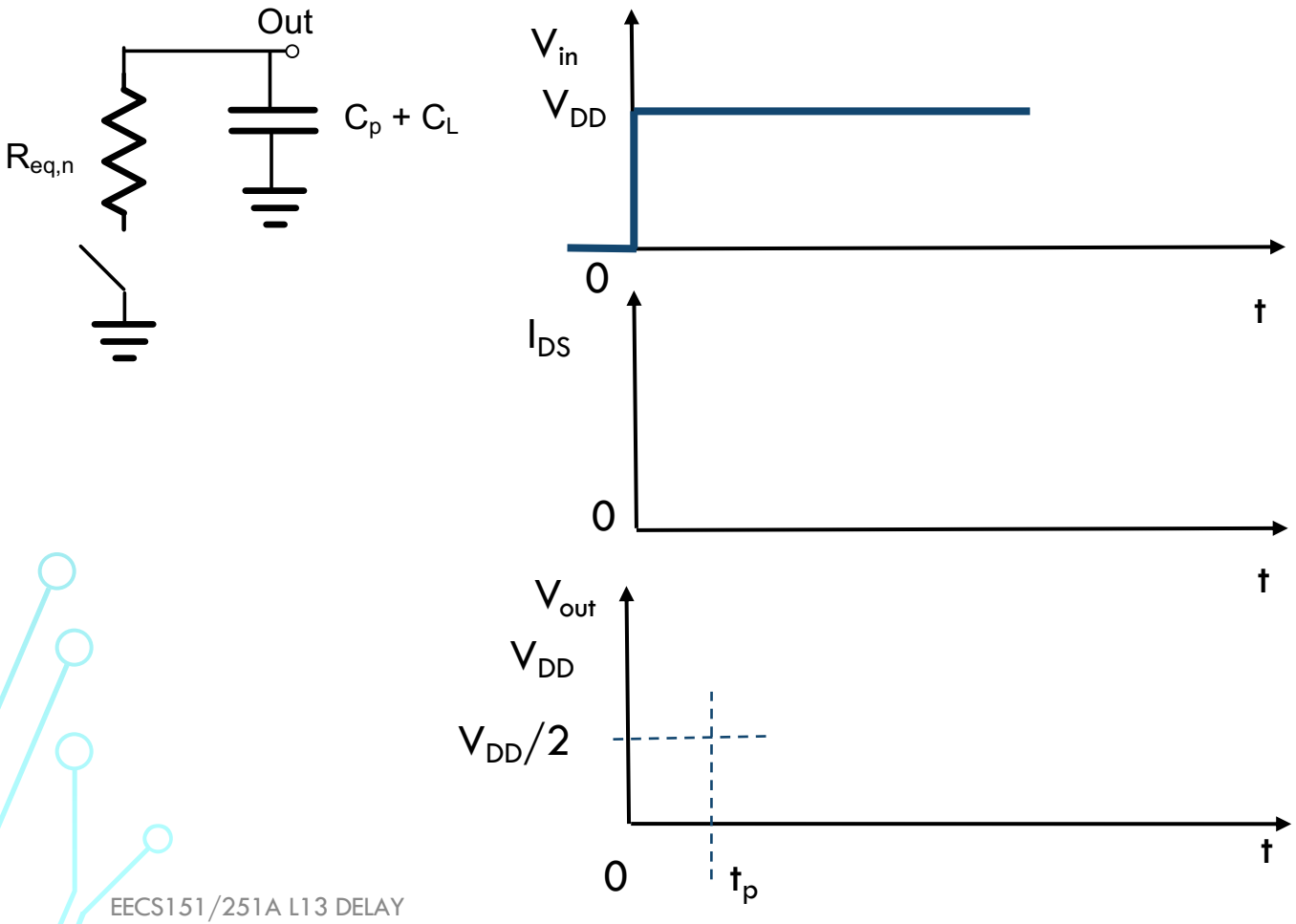


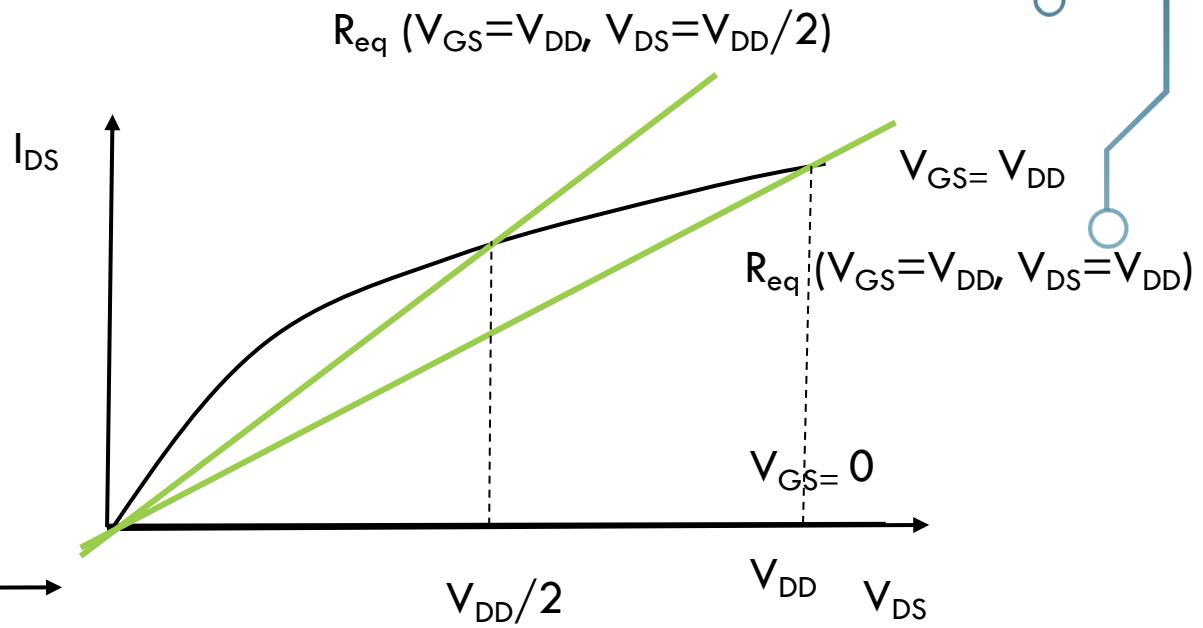


$$t_{p,LH} = (\ln 2)\tau = 0.7\ Req_{,p}(C_p + C_L)$$

Equivalent Resistances

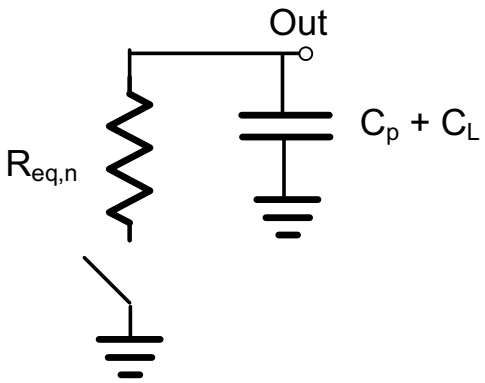
- Transistor I_{DS}-V_{DS} trajectory
- Averaging produces R_{eq}

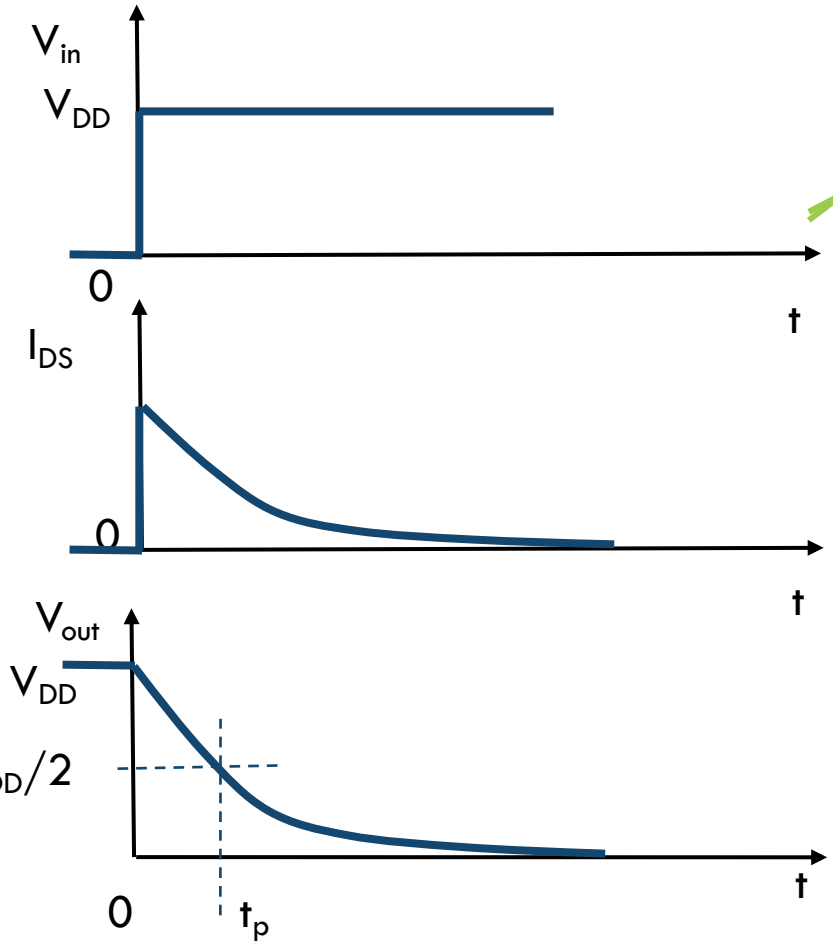


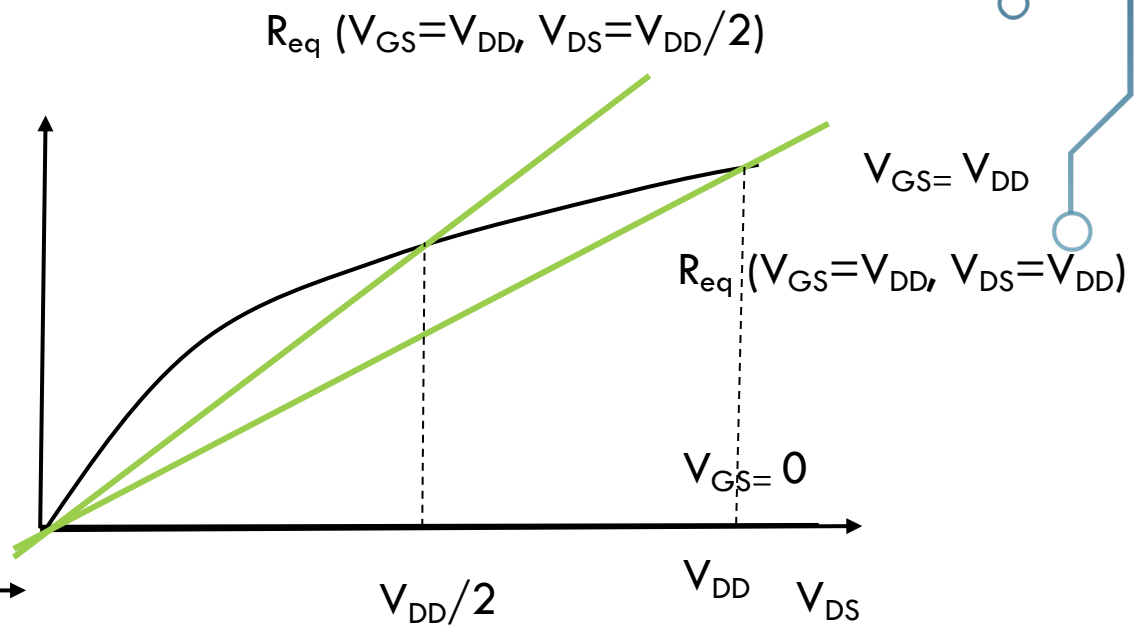


Equivalent Resistances

- Transistor I_{DS}-V_{DS} trajectory
- Averaging produces R_{eq}



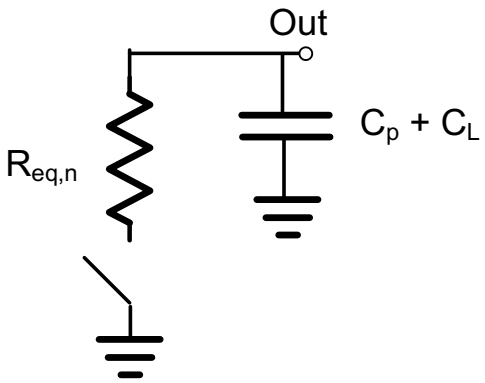


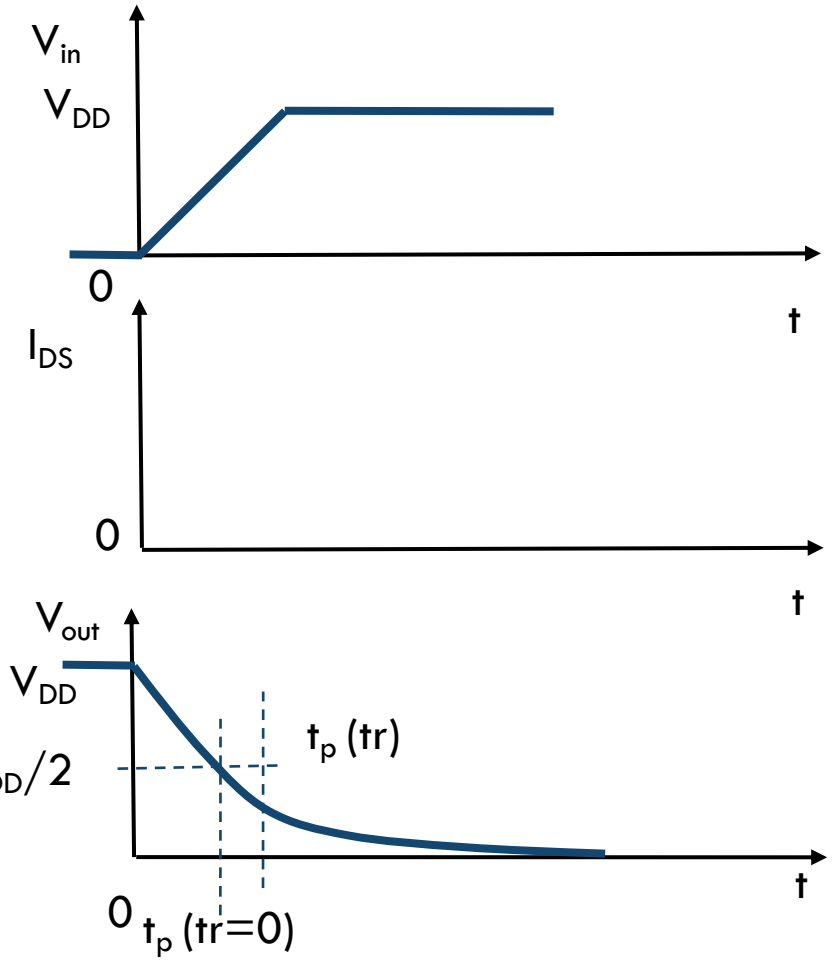


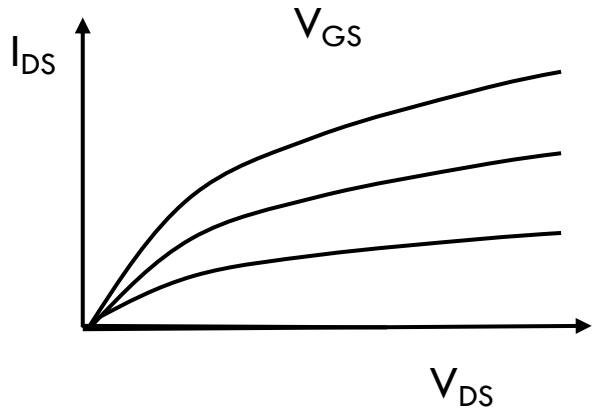
$$R_{eq} = (R_{eq,start} + R_{eq,mid})/2$$

Impact of Rise/Fall times

- Impacts the I_{DS}-V_{DS} trajectory

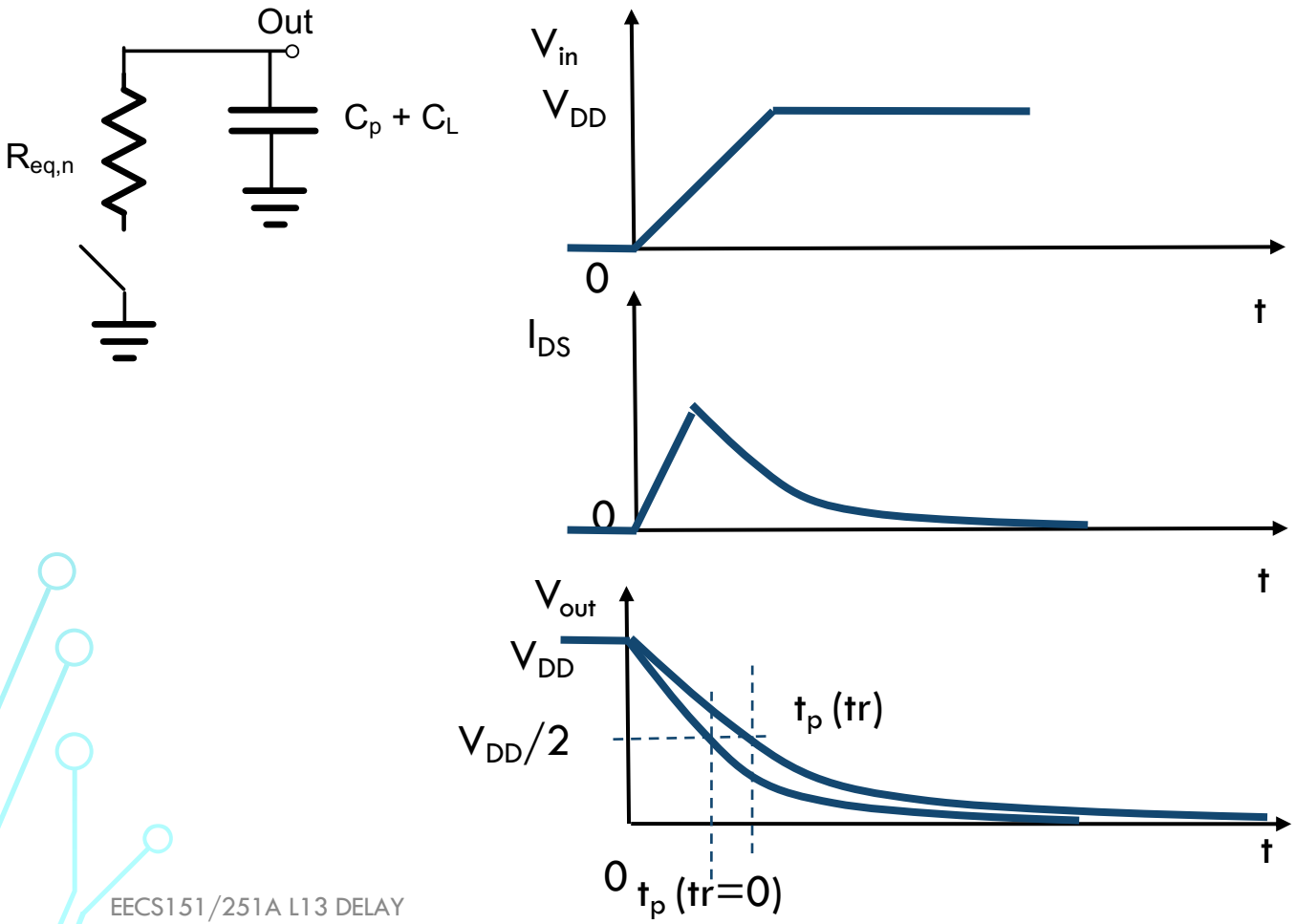


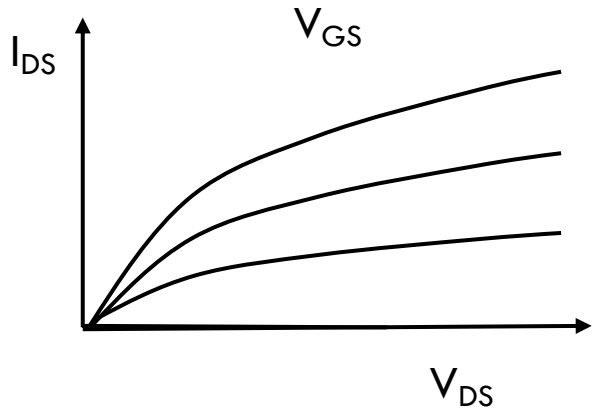




Impact of Rise/Fall times

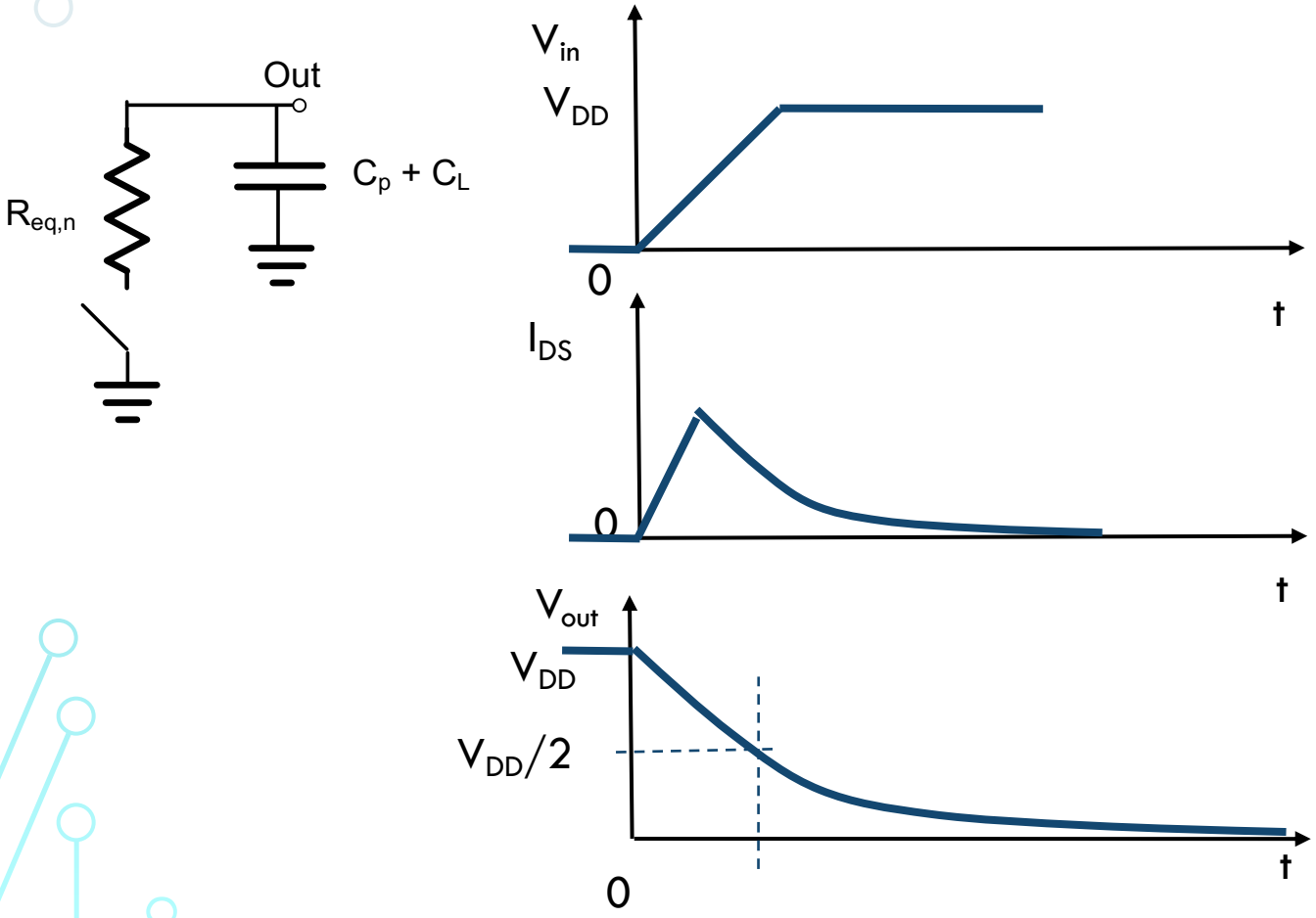
- Impacts the I_{DS}-V_{DS} trajectory

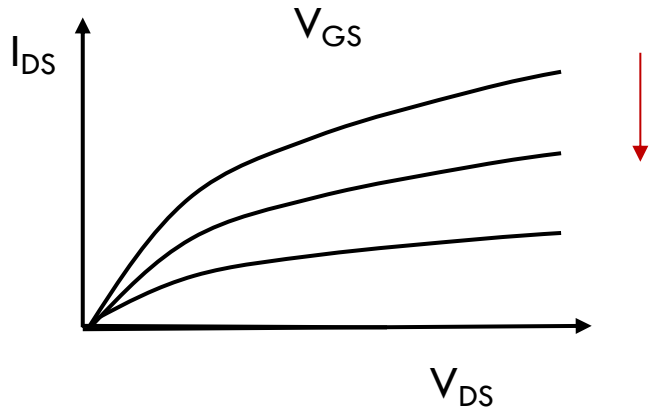




Impact of Supply Voltage

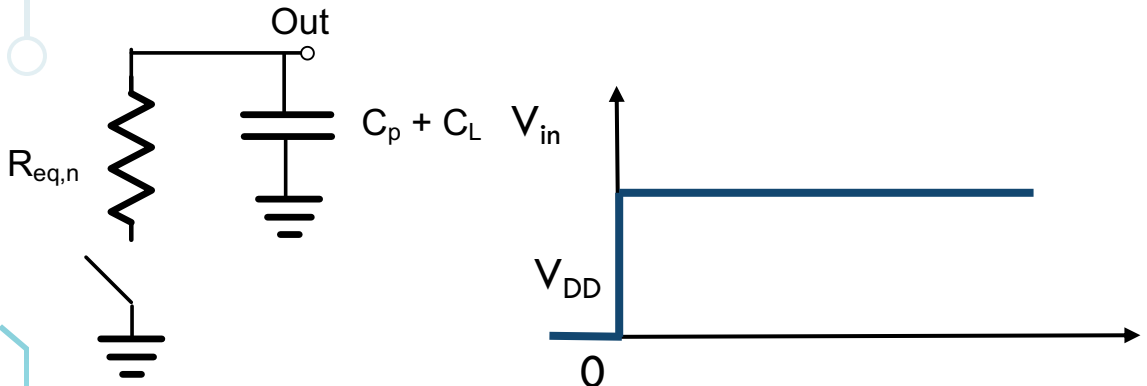
- Lowering VDD, slows down the circuit

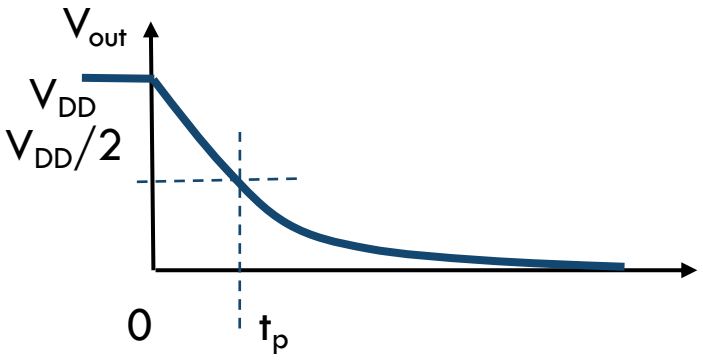




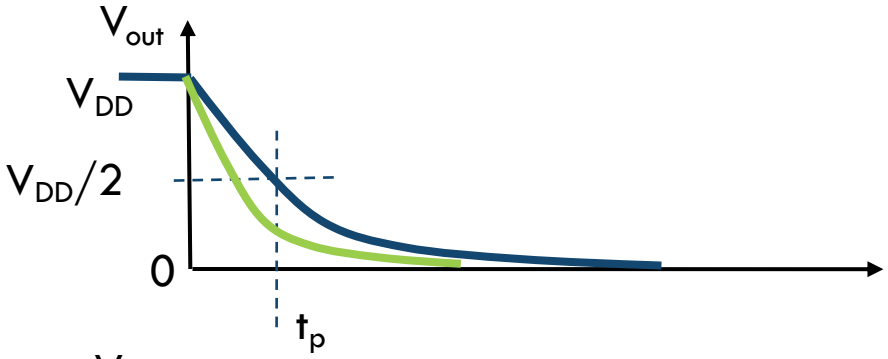
Quiz: Inverter Delay

- If we double the load capacitance, assuming the default Vout shown in blue, which of the following waveforms shows the new Vout?

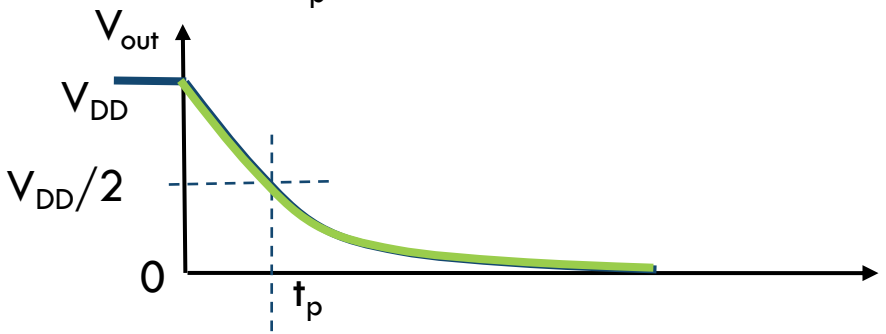




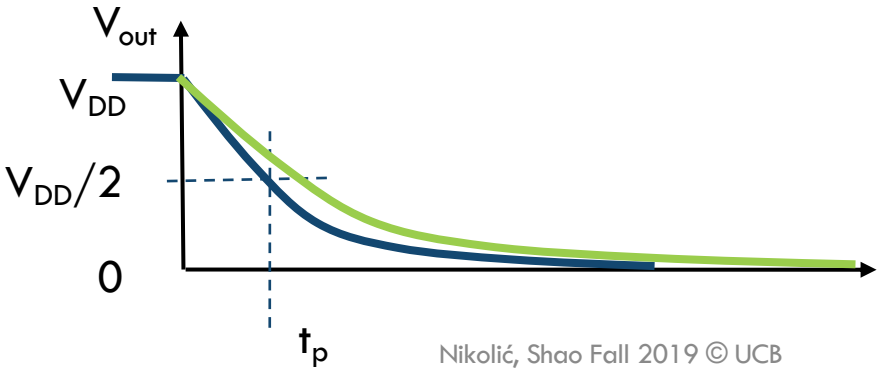
A



B



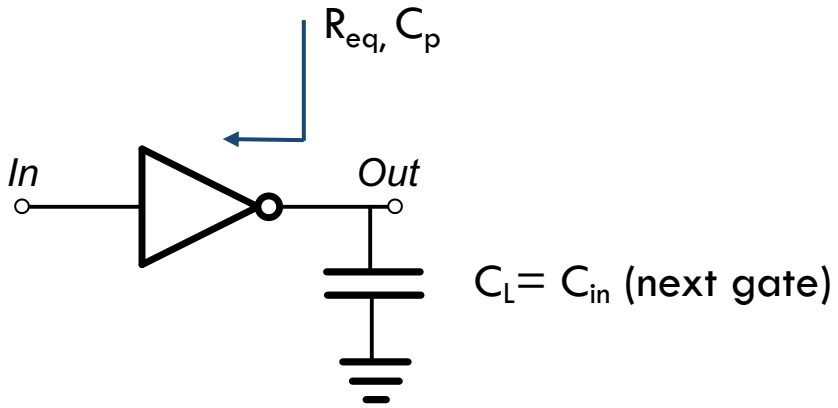
C



CMOS Delay

Overview
Transistor Size
RC Delay
Logical Effort

Inverter RC Delay



- $t_p = R_{eq}(C_p + C_L) = Req(Cin/\gamma + C_L)$
  - $\gamma = 1$ (closer to 1.2 in recent processes)
- $t_p = R_{eq}C_{in}(1+C_L/C_{in}) = \tau_{INV}(1+f)$
  - Propagation delay is proportional to fanout
- Normalized Delay = 1 + f

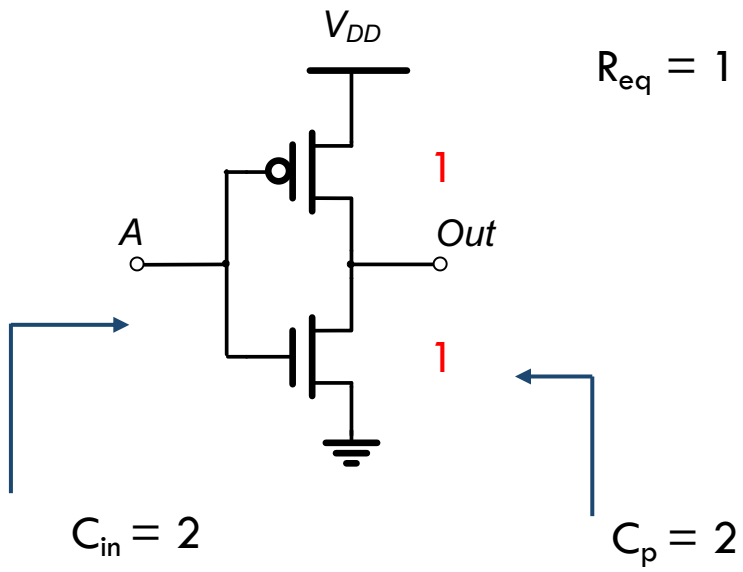
Fanout = $f = C_L/C_{in}$

$t_p = \tau_{INV}(1+f)$

Generalizing to Arbitrary Gates

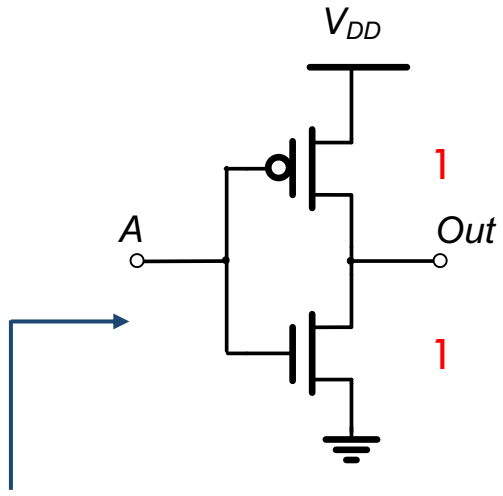
- Delay has two components: $d = f + p$
- f: *effort delay* = gh (a.k.a. stage effort)
  - Again has two components
- g: *logical effort*
  - Measures relative ability of gate to deliver current
  - $g = 1$ for inverter
- f: *electrical fanout* = C_{out} / C_{in}
  - Ratio of output to input capacitance
  - Sometimes called electrical effort
- p: parasitic delay
  - Represents delay of gate driving no load
  - Set by internal parasitic capacitance

Inverter Delay



- Parasitic p is the ratio of intrinsic capacitance to an inverter
  - p(inverter) =
- Logical Effort g is the ratio of input capacitance to an inverter
  - g(inverter) =
- Electrical Effort h is the ratio of the load capacitance to the input capacitance
  - h(inverter) =
- Delay = p + f = p + g * h = 1 + f

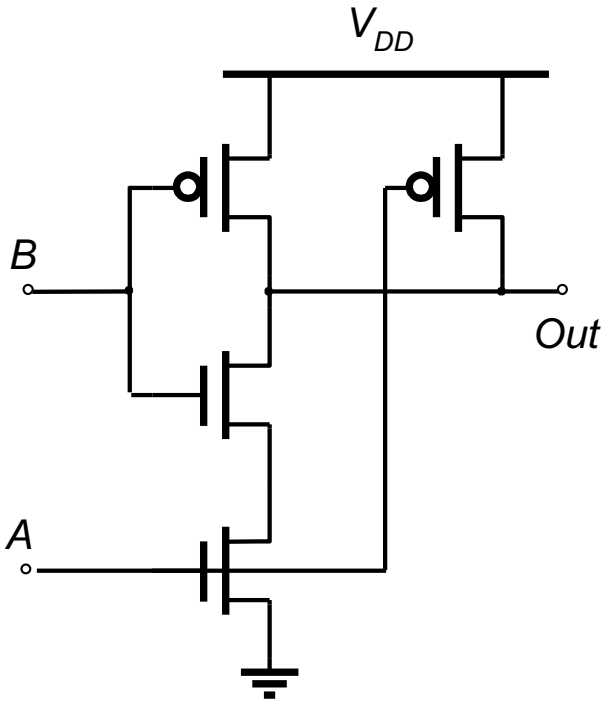
NAND2 Gate



$R_{eq} = 1$

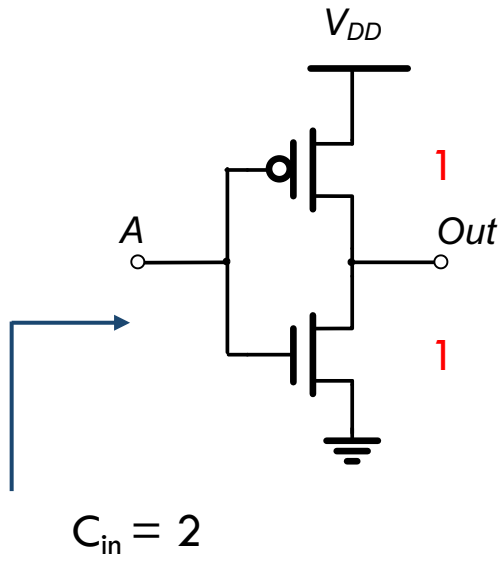
$C_{in} = 2$

$C_{in} =$

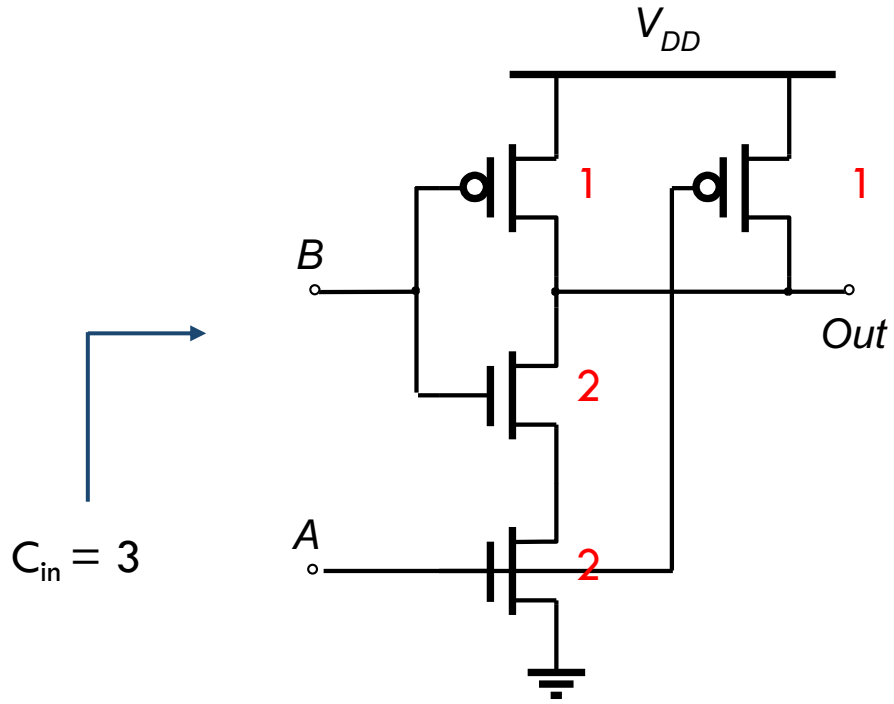


$R_{eq} = 1$

Logical Effort of NAND2 Gate



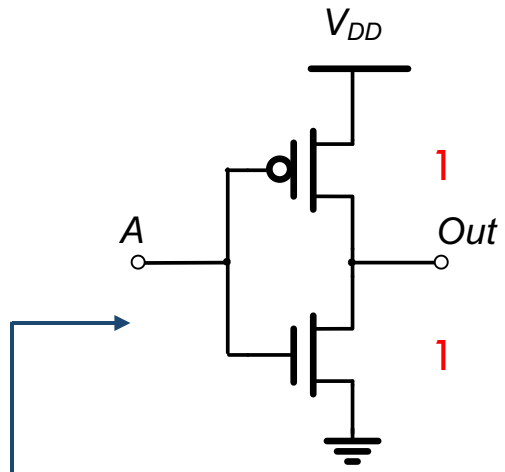
$R_{eq} = 1$



$R_{eq} = 1$

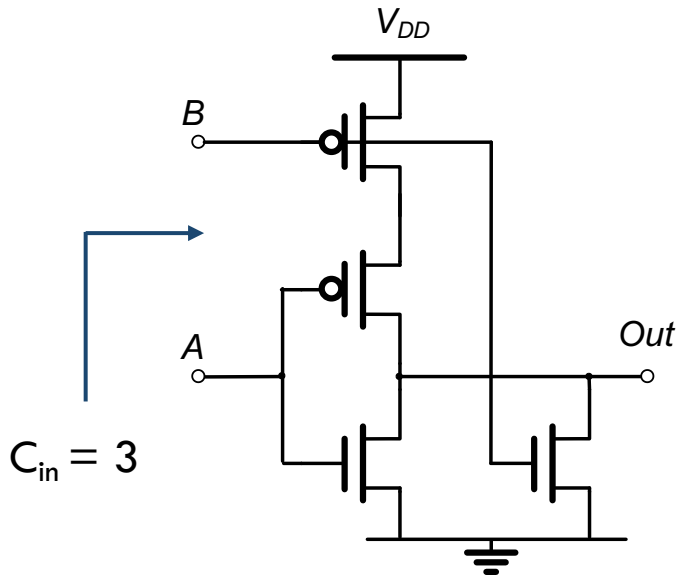
- In velocity-saturated devices Ion of a stack is 2/3 (not a half) of two devices
  - So the correct upsizing factor is 1.5 (not 2)
- We will use 2, as it makes calculations easier

NOR2 Gate



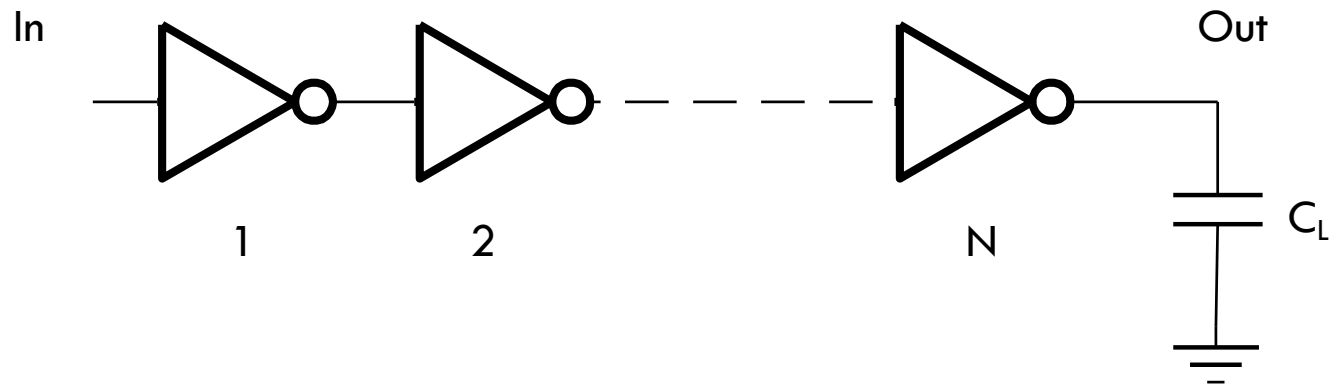
$C_{in} = 2$

$R_{eq} = 1$



$C_{in} = 3$

Example: Inverter Chain



Logical Effort: g =

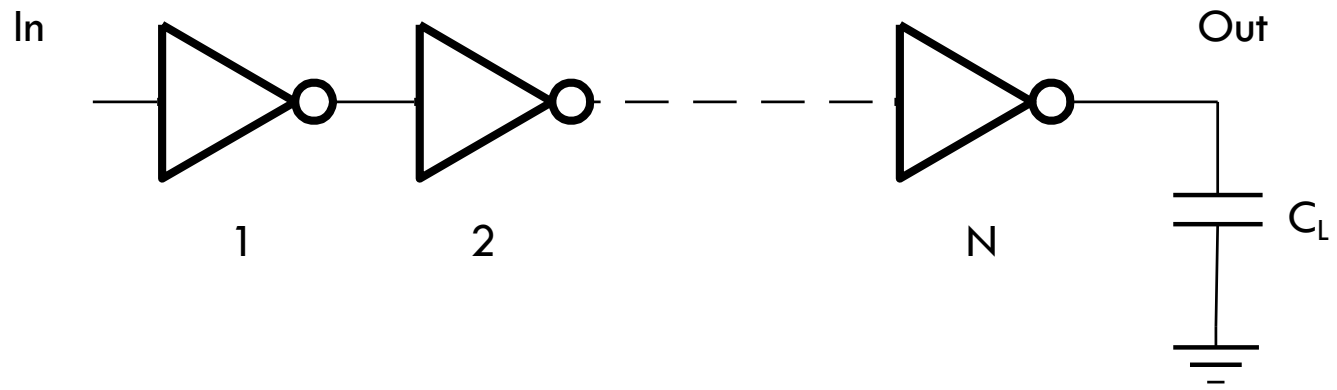
Electrical Effort: h =

Parasitic Delay: p =

Stage Delay: d =

Total Delay: d_total =

Example: Inverter Chain



Logical Effort: g = 1

Electrical Effort: h = 1

Parasitic Delay: p = 1

Stage Delay: d = 2

Total Delay: d_total = 2*N

Administrivia

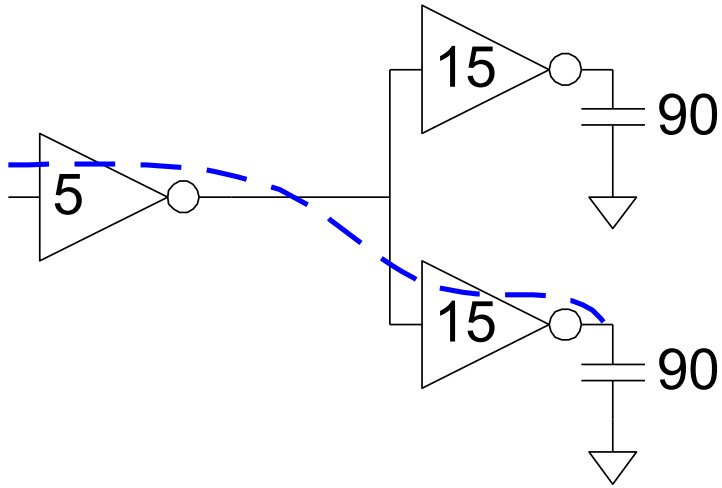
- Finish labs soon.
- Project start this week.
- Explore Internships and research opportunities.
- EE290: Hardware for Machine Learning
  - Spring 2020

Multi-stage Logic Networks

- Logical effort generalizes to multistage networks
- *Path Logical Effort* $G = \prod g_i$
- *Path Electrical Effort* $H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$
- *Path Effort* $F = \prod f_i = \prod g_i h_i$

Branching Effect

$$b = \frac{C_{\text{on path}} + C_{\text{off path}}}{C_{\text{on path}}} \qquad B = \prod b_i$$



G = 1

H = 90 / 5 = 18

GH = 18

h_1 = (15 +15) / 5 = 6

h_2 = 90 / 15 = 6

B = 2

F = $g_1 g_2 h_1 h_2$ = 36 = BGH

Designing Fast Circuits

$$D = \sum d_i = D_F + P$$

- Delay is smallest when each stage bears same effort

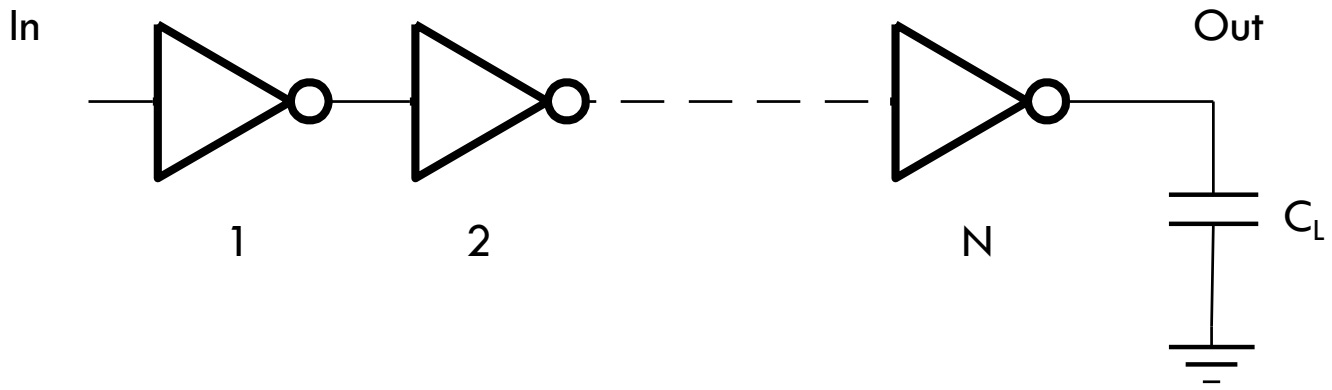
$$\hat{f} = g_i h_i = F^{\frac{1}{N}}$$

- Thus minimum delay of N stage path is

$$D = NF^{\frac{1}{N}} + P$$

- This is a key result of logical effort
  - Find fastest possible delay
  - Doesn't require calculating gate sizes

Example: Minimizing the delay of an inverter chain



Delay = $t_{p1} + t_{p2} + \ldots + t_{pN}$

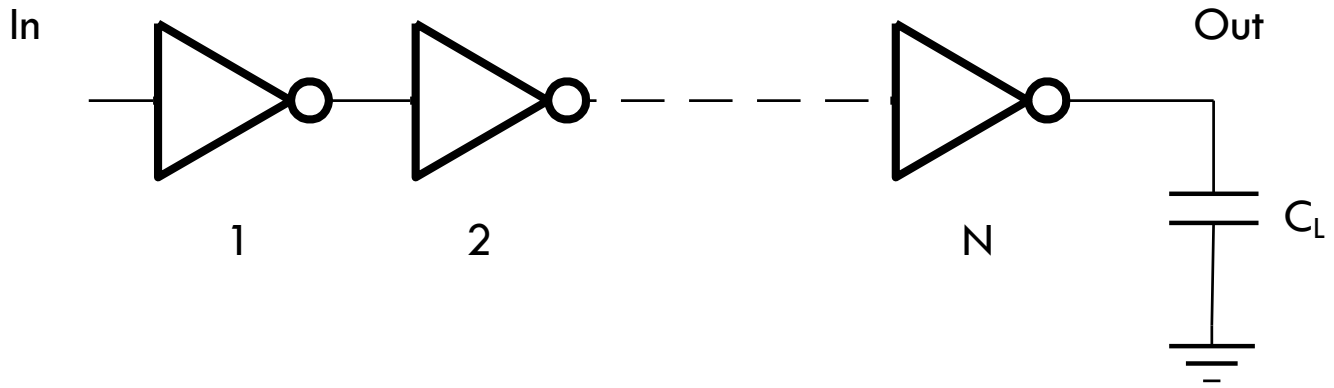
Delay = $(1+f_1) + (1+f_2) + \ldots + (1+f_N)$;

$f_1 = C_2/C_1$, $f2 = C_3/C_2$

Minimum -> $f_1 = f_2 = \ldots f_N =$

Example: Best Number of Stages

- How many stages should a path use?
  - Minimizing number of stages is not always fastest



$$D = NF^{1/N} + Np_{inv}$$

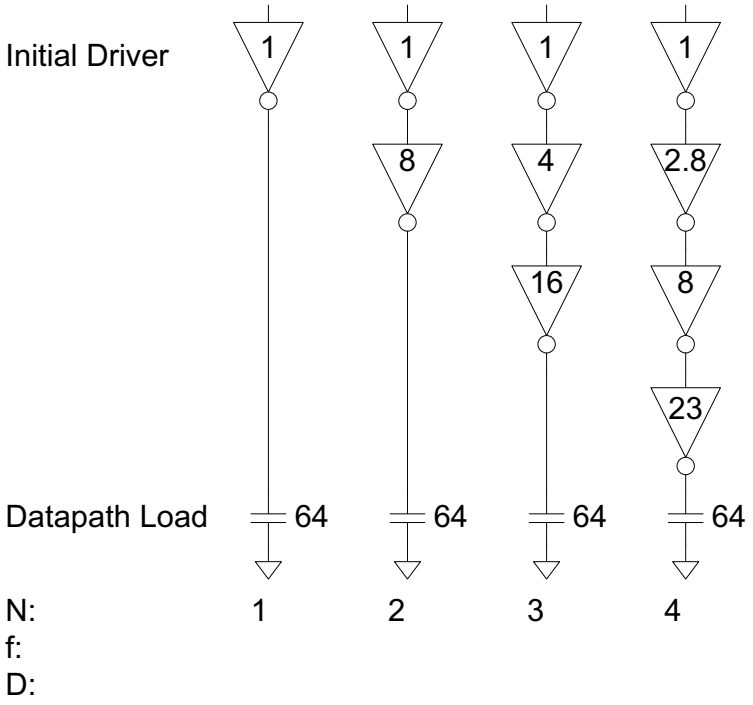
- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = 2.718$ (e)
- For $p_{inv} = 1$, solve numerically for $\rho = 3.59$

Example: Best Number of Stages

- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

$$D = NF^{1/N} + P$$

$$= N(64)^{1/N} + N$$

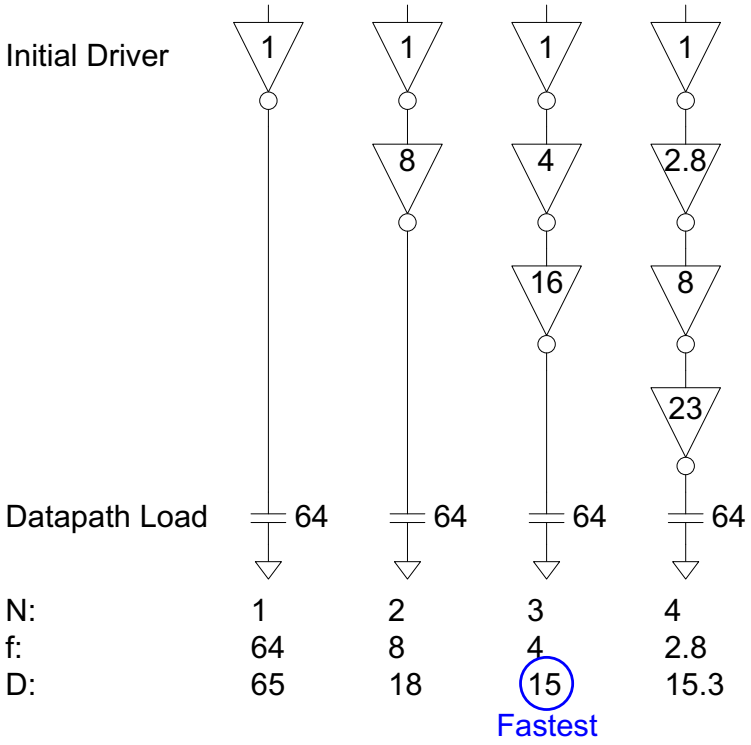


Example: Best Number of Stages

- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

$$D = NF^{1/N} + P$$

$$= N(64)^{1/N} + N$$



Summary

- Delay optimization is critical to improve the frequency of the circuit.
- The dimensions of a transistor affect its capacitance and resistance.
- We use RC delay model to describe the delay of a circuit.
- Two delay components:
  - Parasitic delay (p)
  - Effort delay (F)
    - Logical effort (g): intrinsic complexity of the gate
    - Electrical effort (h): load capacitance dependent