Digital Integrated Circuits Workshop

Week 5:

Introduction to Adders, Delay and Power Models

Prof. Dejan Markovic

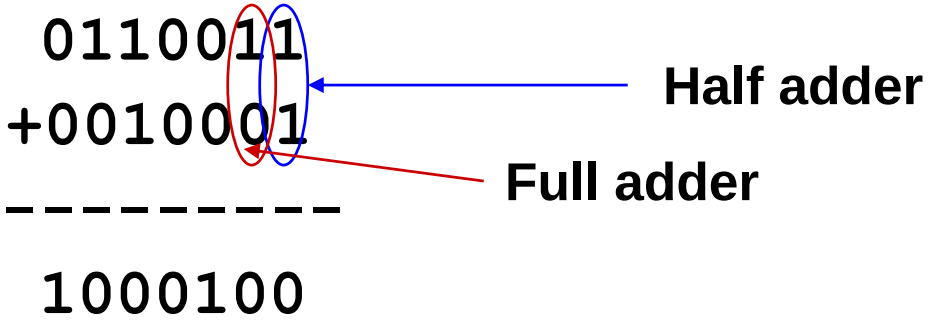
UCLA

Week 5 Agenda

- Introduction to Adders
- Delay Model
- Power Model

Simple Addition

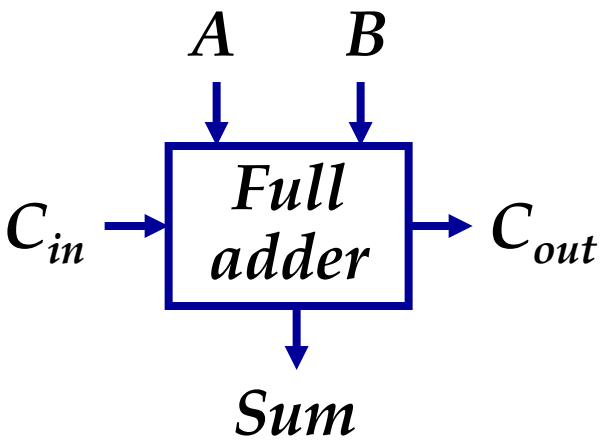
- **Grade school addition**



- **Use an adder cell per bit position**
- **Half adder**
  - A+B = Sum, Carry_out
- **Full adder**
  - A+B+Carry_in = Sum, Carry_out
  - a.k.a 3-2 adder

Full Adder – Preliminary Considerations

- **Sum function**
  - S_o=1 when Odd input 1's
  - S_o = XOR(A,B,C_i)
  - S_o = A x B x C_i
- **Carry function**
  - C_o=1 when 2 or more input 1's
  - C_o = Majority(A,B,C_i)
  - C_o = AB + BC_i + AC_i



Binary Adder

A	B	C_i	S_o	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

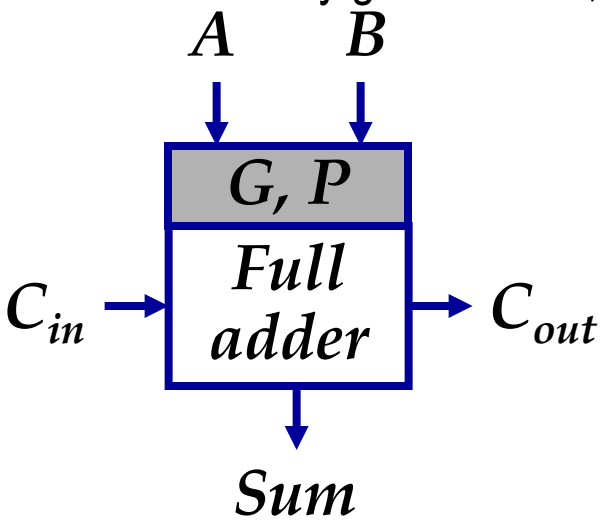
$$S = A \oplus B \oplus C_i$$

$$= A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + \bar{A}\bar{B}C_i + ABC_i$$

$$C_o = AB + BC_i + AC_i$$

Full Adder – Another Look…

- It's all about carry, so redefine terms for more efficient design
- Carry status
  - Delete (D): $D = \bar{A} \cdot \bar{B}$
  - Propagate (P): $P = A \oplus B$
  - Generate (G): $G = A \cdot B$
- Full adder
  - Internally generates P, G, (D)



Binary Adder

C_o status	A	B	C_i	C_o
Delete	0	0	0	0
Delete	0	0	1	0
Propagate	0	1	0	0
Propagate	0	1	1	1
Propagate	1	0	0	0
Propagate	1	0	1	1
Generate	1	1	0	1
Generate	1	1	1	1

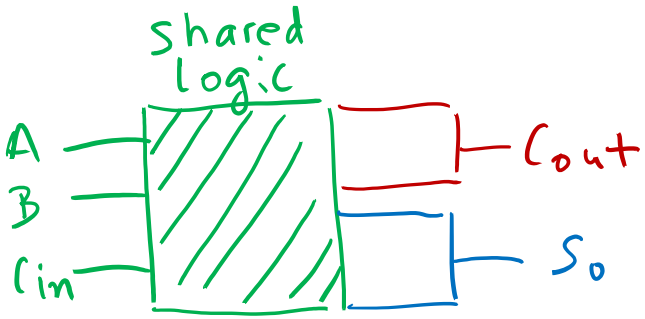
$$C_o(G,P) = G + P \cdot C_i$$

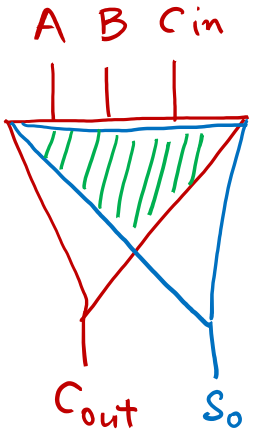
$$S(G,P) = P \oplus C_i$$

Gate Realization of the Full Adder Cell

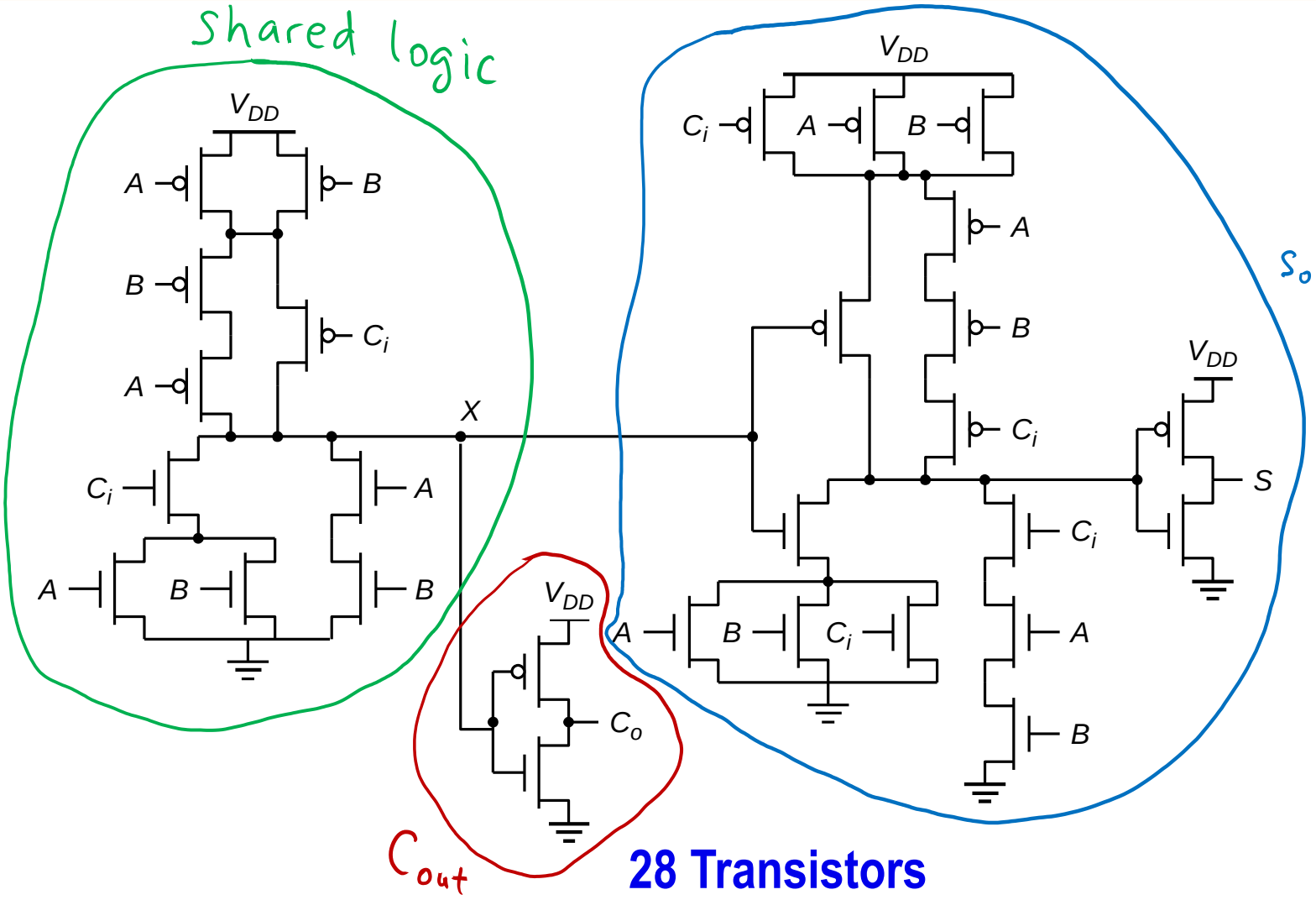
- **Strategy: share logic between Sum and Carry sub-circuits if this doesn't slow Carry generation**

- **Some possible implementations**
  - Complementary CMOS
  - Mirror Adder
  - Manchester-Carry Chain (lecture 14)





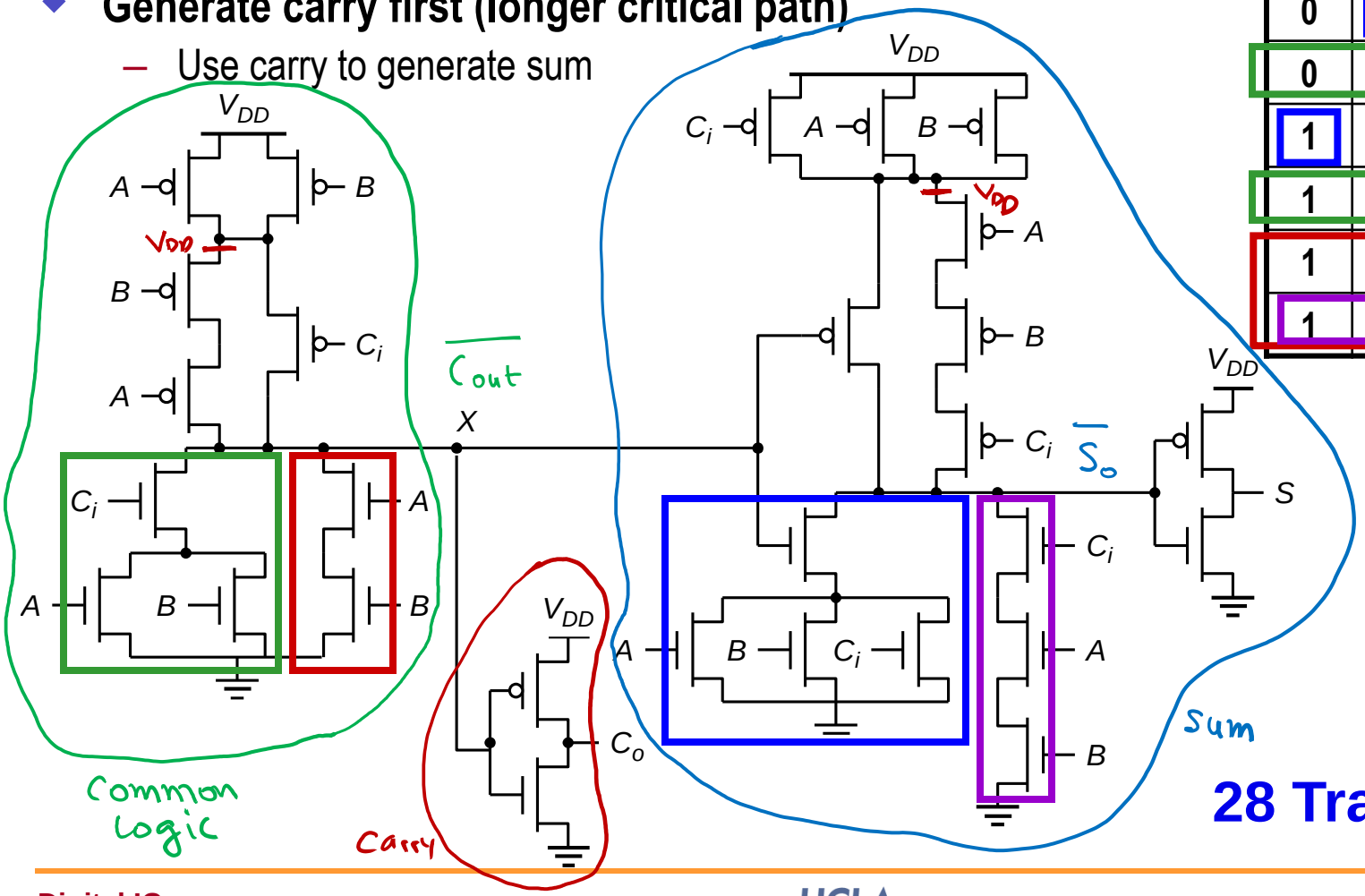
Complementary Static CMOS Full Adder



Implementation Strategy

- **Implement as separate gates…**
  - Not as efficient (especially XOR)
- **Generate carry first (longer critical path)**
  - Use carry to generate sum

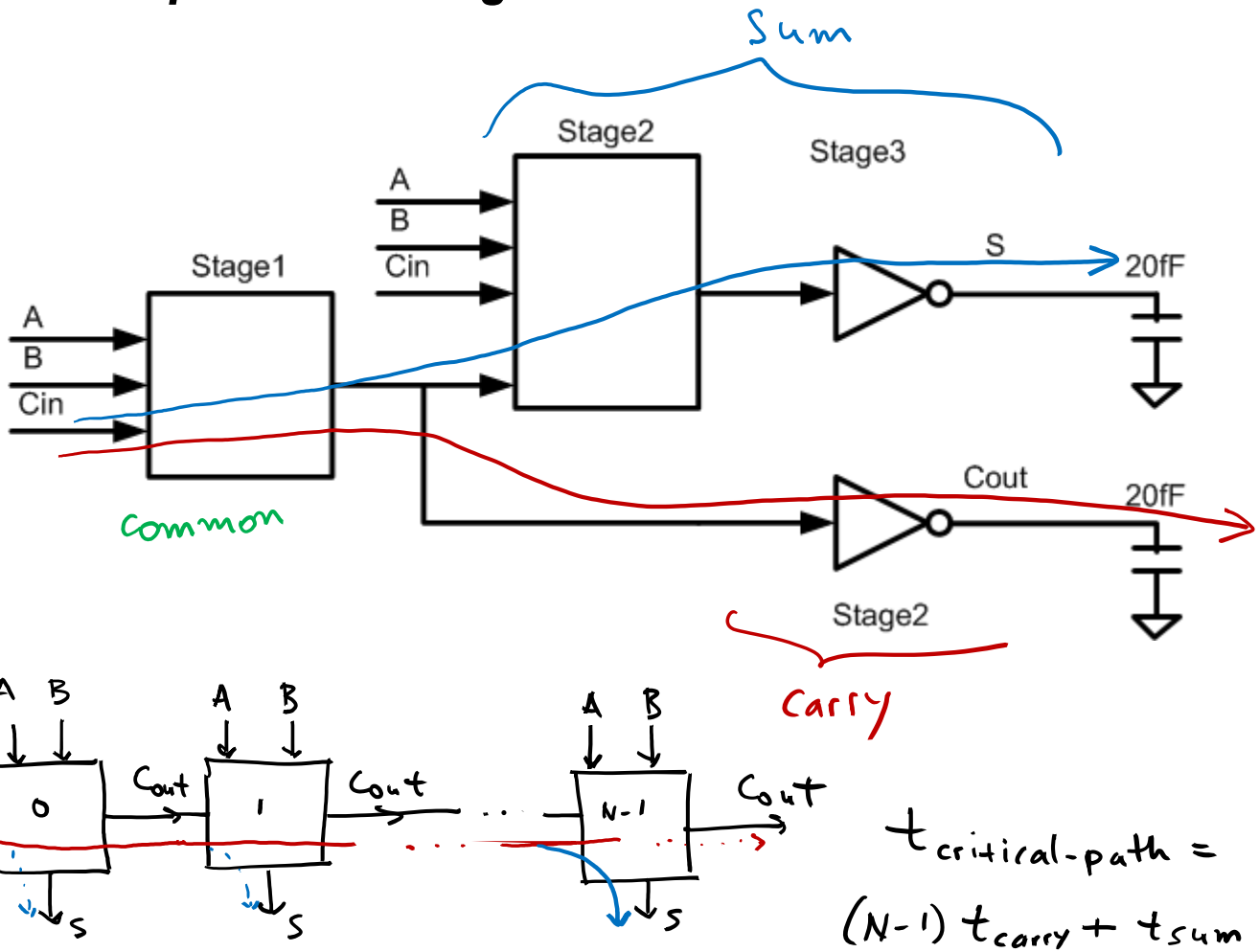
A	B	C_i	S_o	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



28 Transistors

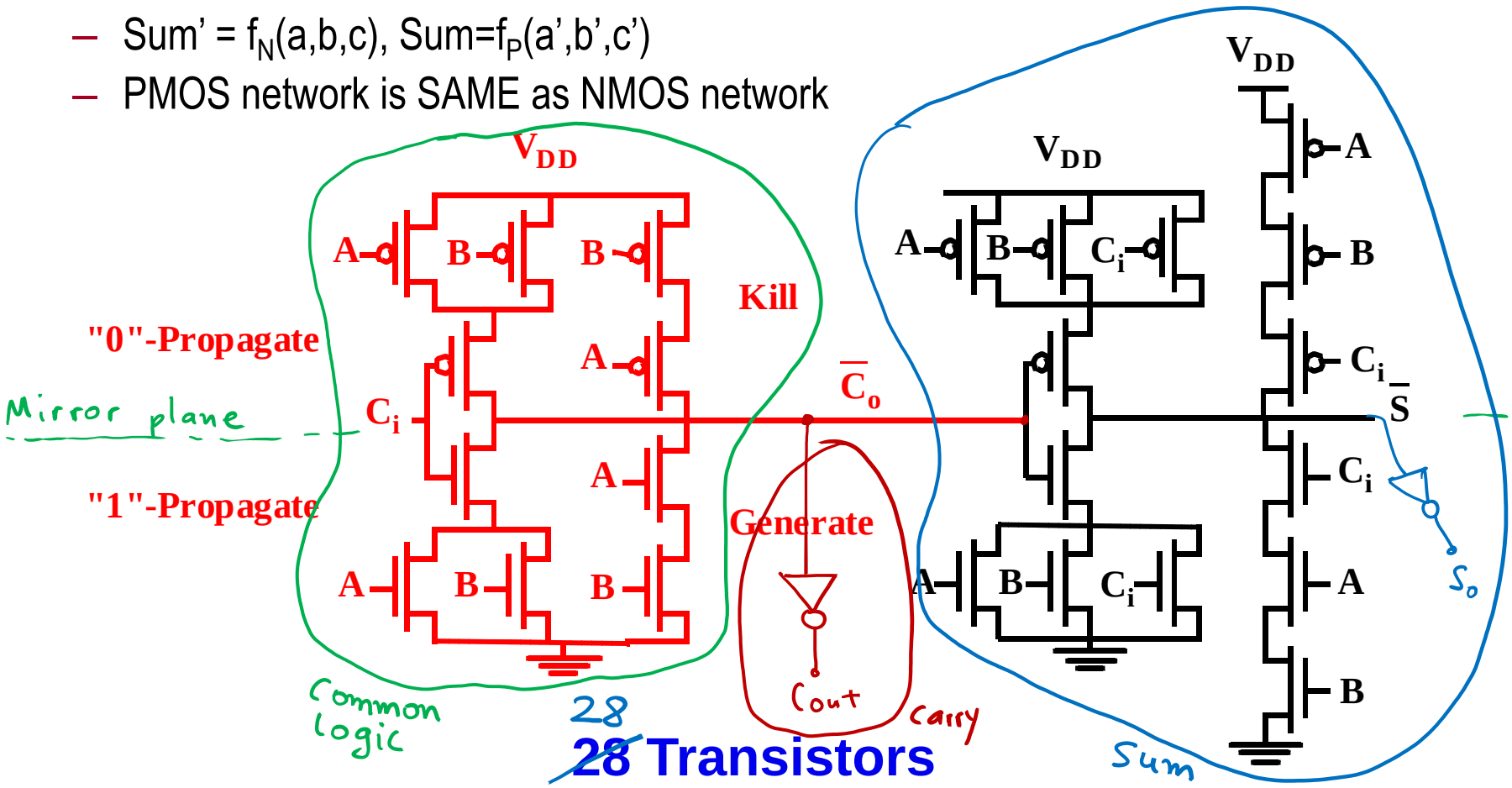
Critical Path Analysis…

Assume multiple adder stages:



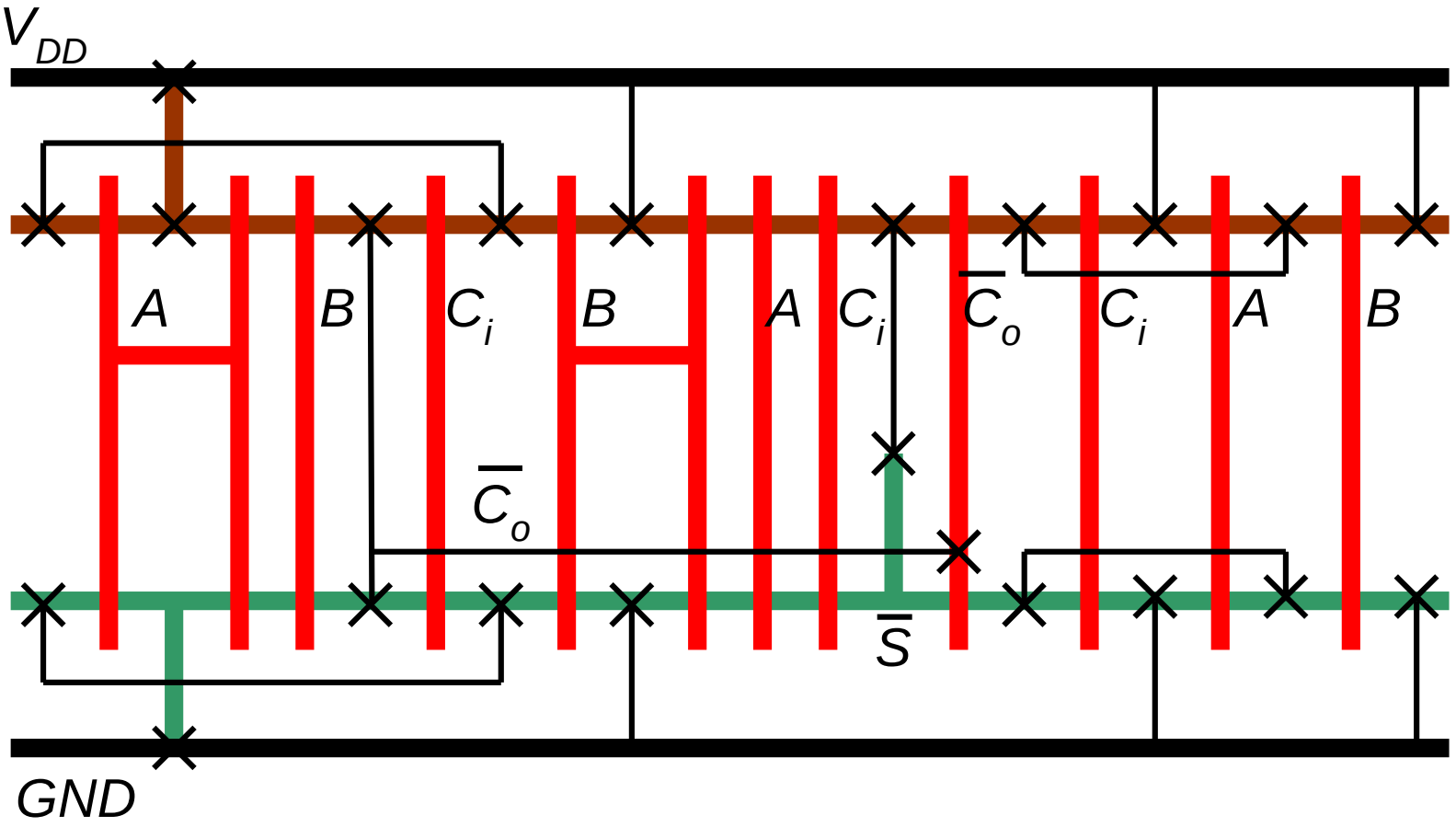
A Better Structure: The Mirror Adder

- **Both Sum and Carry are symmetric functions**
  - Inverted inputs results in inverted outputs
  - Sum' = $f_N(a,b,c)$, Sum=$f_P(a',b',c')$
  - PMOS network is SAME as NMOS network



Mirror Adder

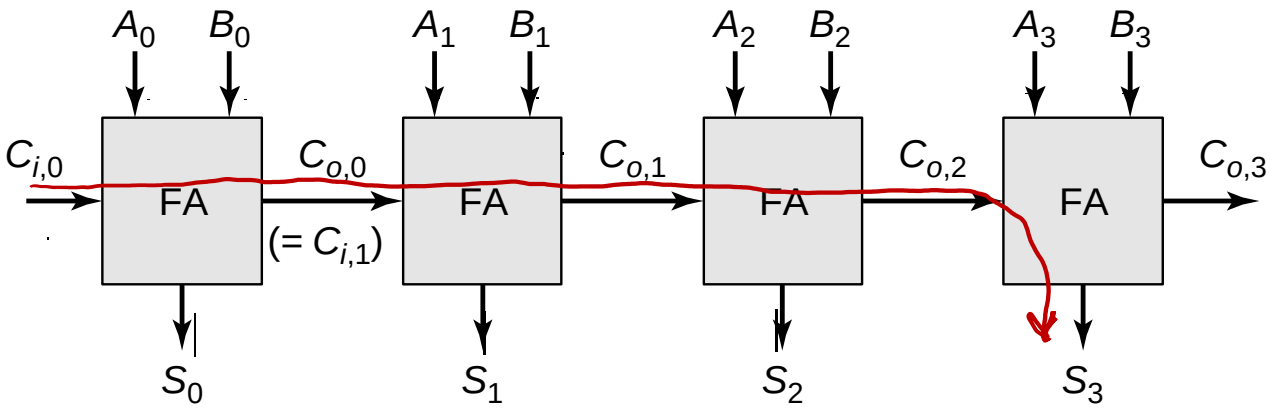
Stick Diagram



Basic Adder Topologies

- **Ripple Carry**
- **Carry Bypass**
- **Carry Select**
  - Linear
  - Square Root
- **Carry Look-Ahead**

The Ripple-Carry Adder



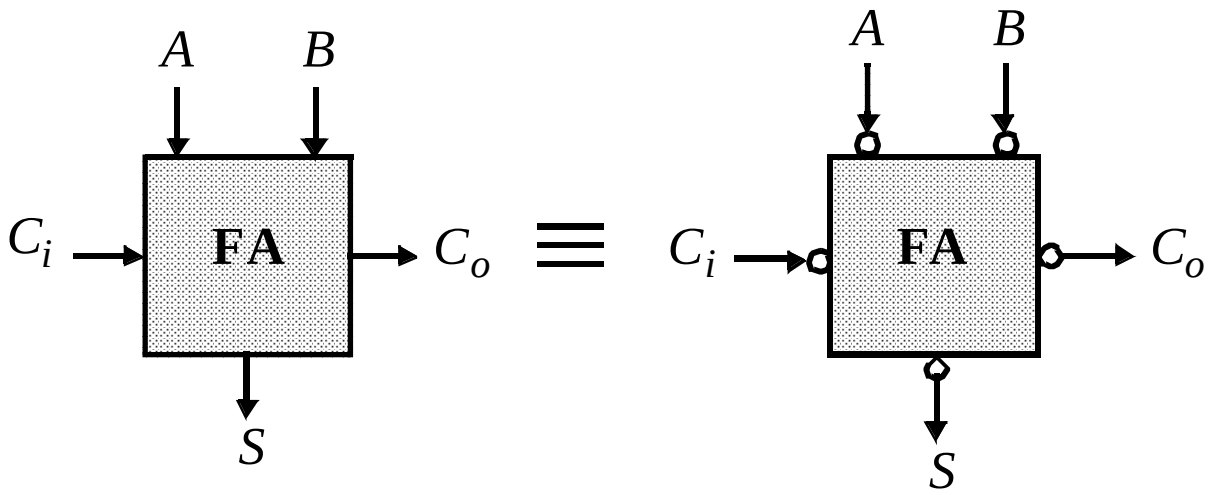
Worst case delay linear with the number of bits

$$t_d = O(N)$$

$$t_{adder} = (N-1)t_{carry} + t_{sum}$$

Goal: Make the fastest possible carry path circuit

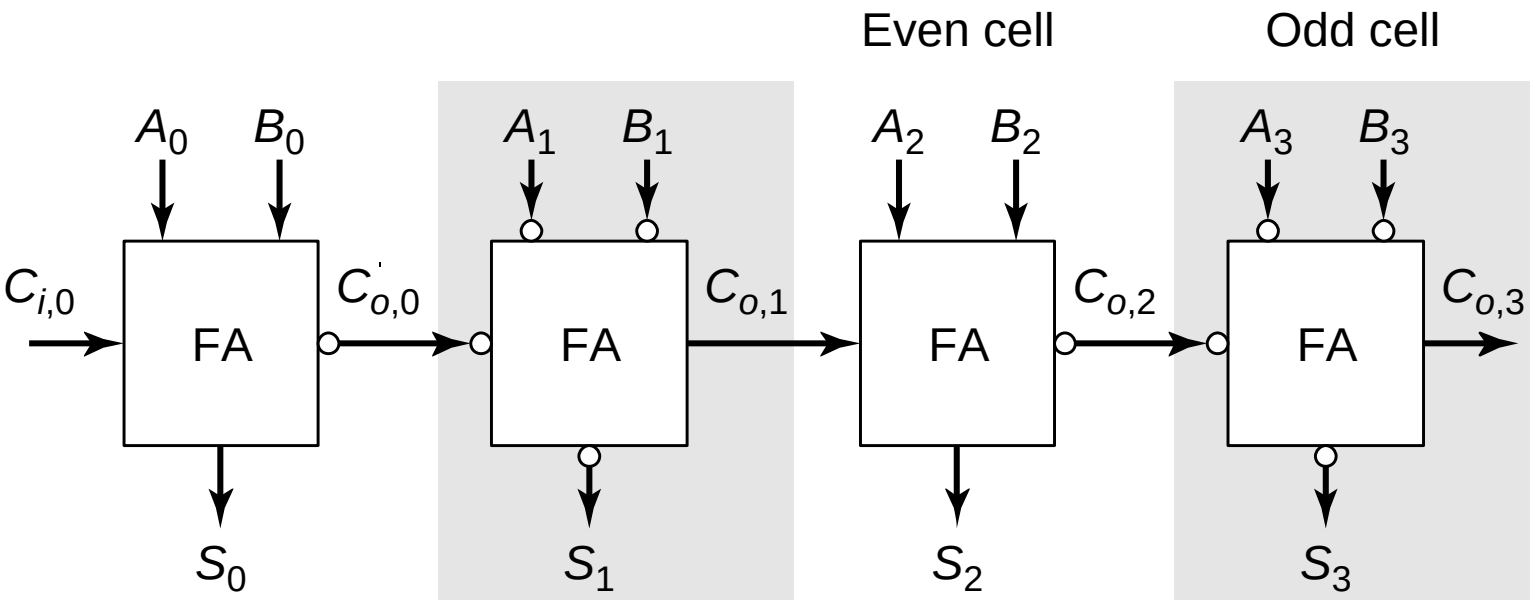
Inversion Property



$$\bar{S}(A, B, C_i) = S(\bar{A}, \bar{B}, \overline{C_i})$$

$$\overline{C_o}(A, B, C_i) = C_o(\bar{A}, \bar{B}, \overline{C_i})$$

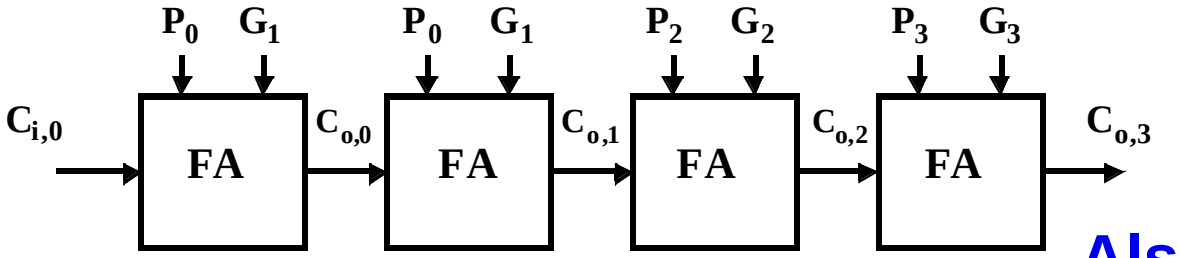
Minimize Critical Path by Reducing Inverting Stages



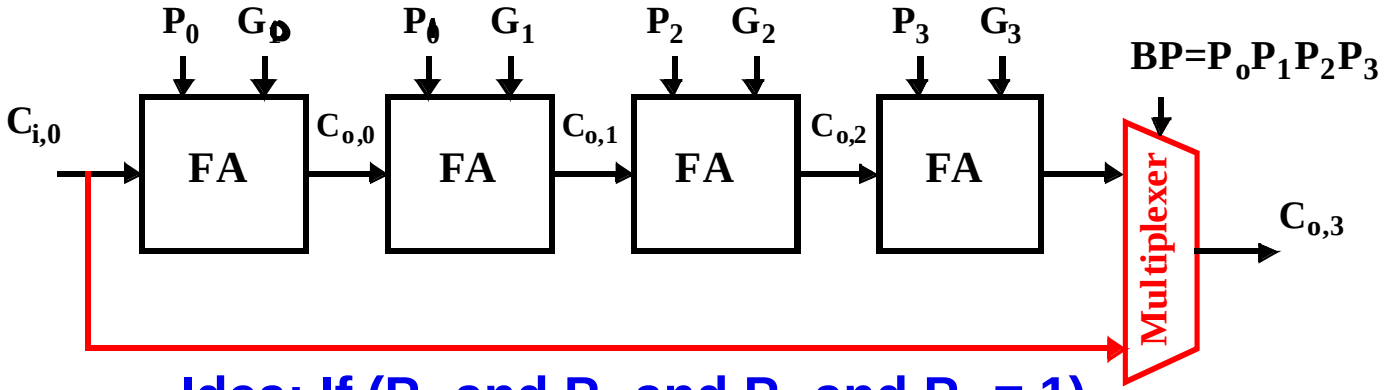
Exploit Inversion Property

Carry-Bypass Adder

- **Simple adders ripple the carry, faster ones bypass it**
  - Calculate the carry several bits at a time
  - Good for small adders (n < 16)

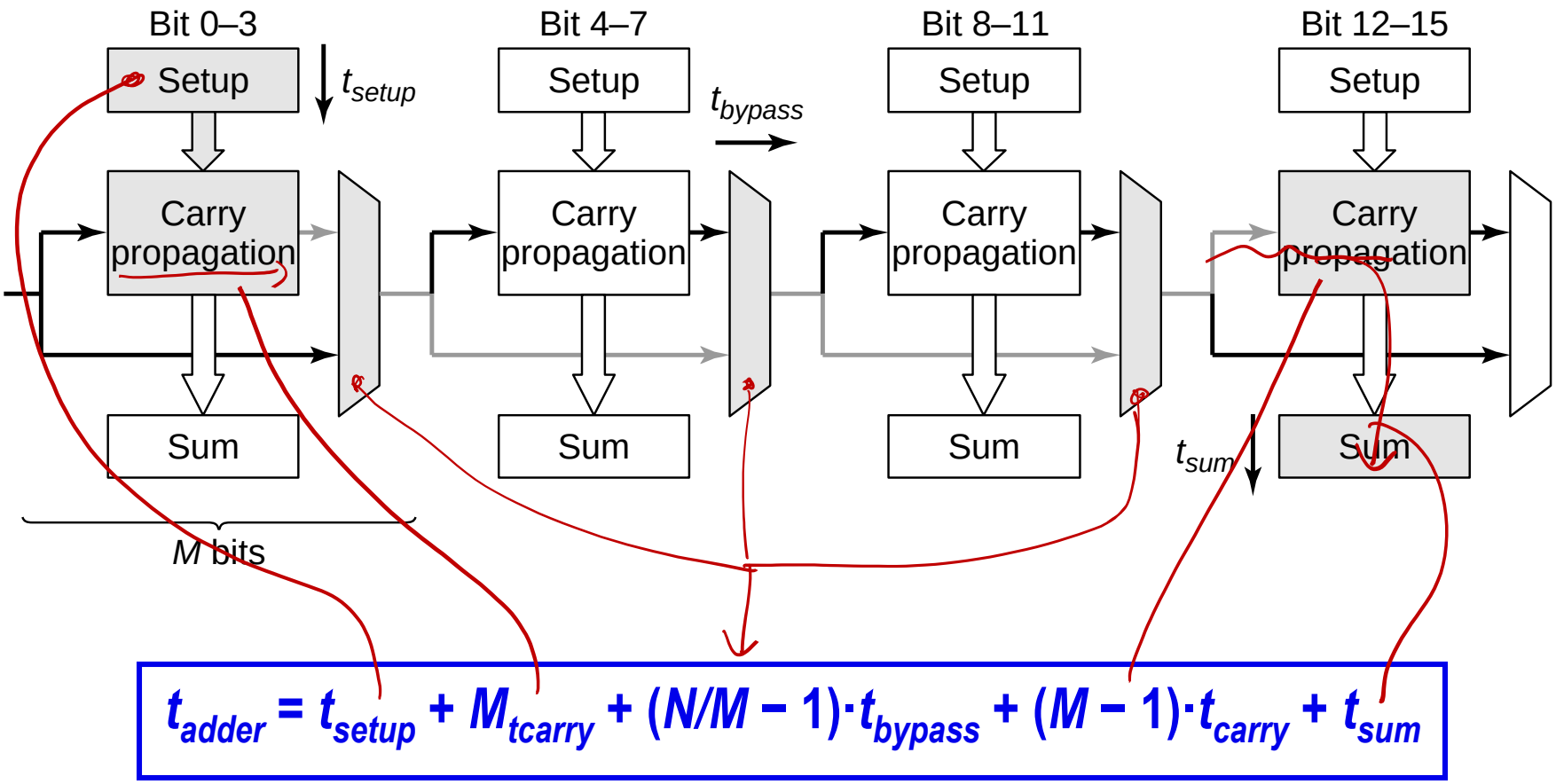


Also called Carry-Skip



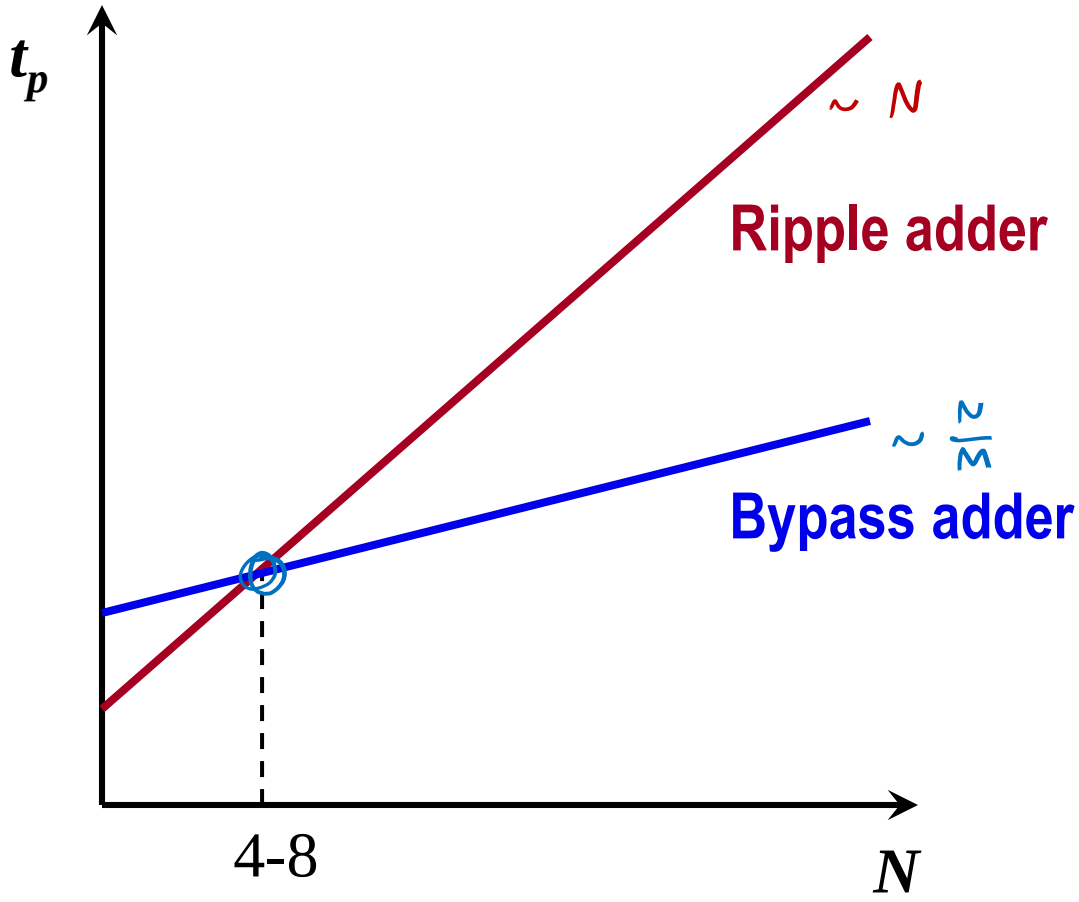
Idea: If (P_0 and P_1 and P_2 and P_3 = 1), then $C_{o,3} = C_o$, else "kill" or "generate"

Critical Path in Carry-Bypass Adder



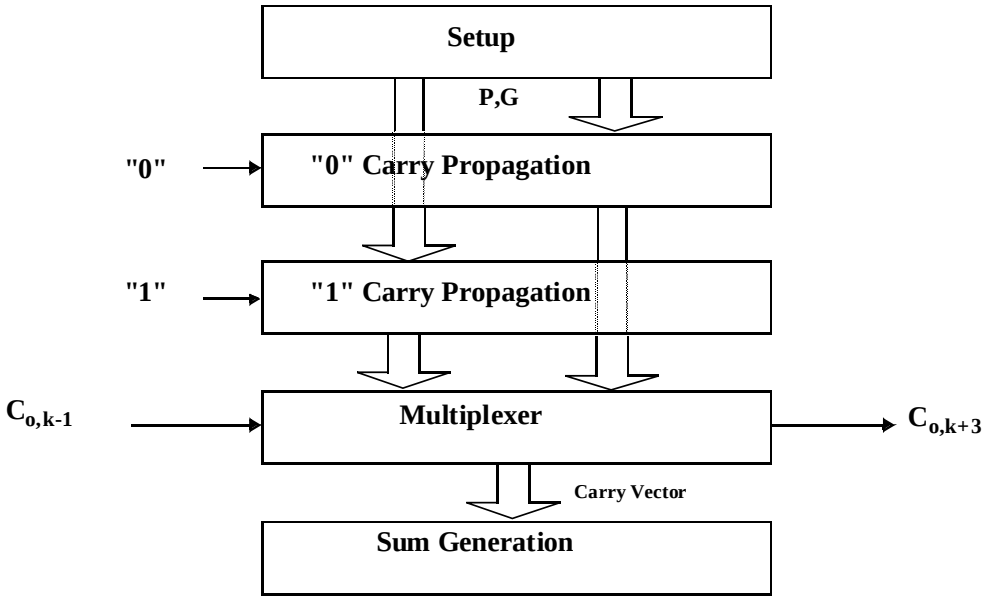
$$t_{adder} = t_{setup} + M_{tcarry} + (N/M - 1)\cdot t_{bypass} + (M - 1)\cdot t_{carry} + t_{sum}$$

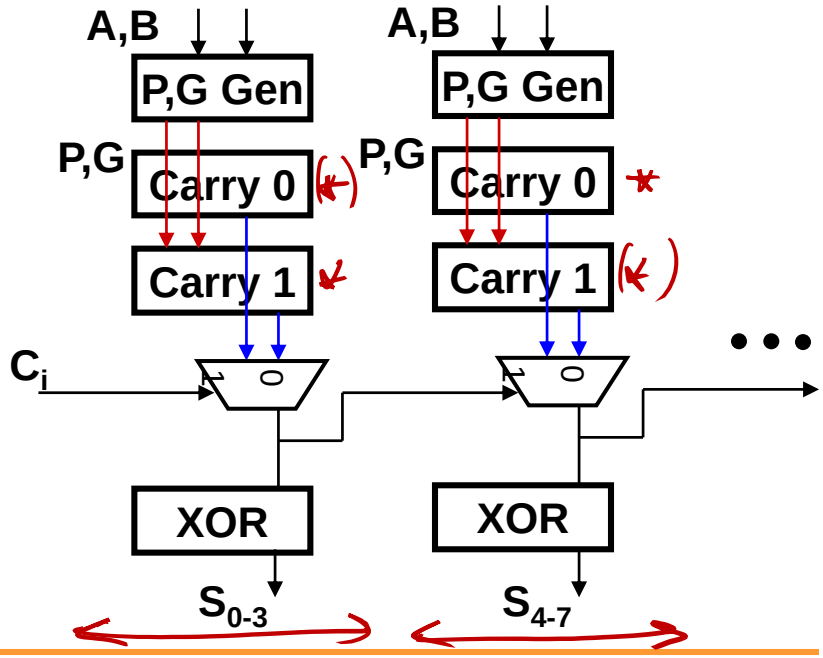
Carry Ripple vs. Carry Bypass



Carry-Select Adder

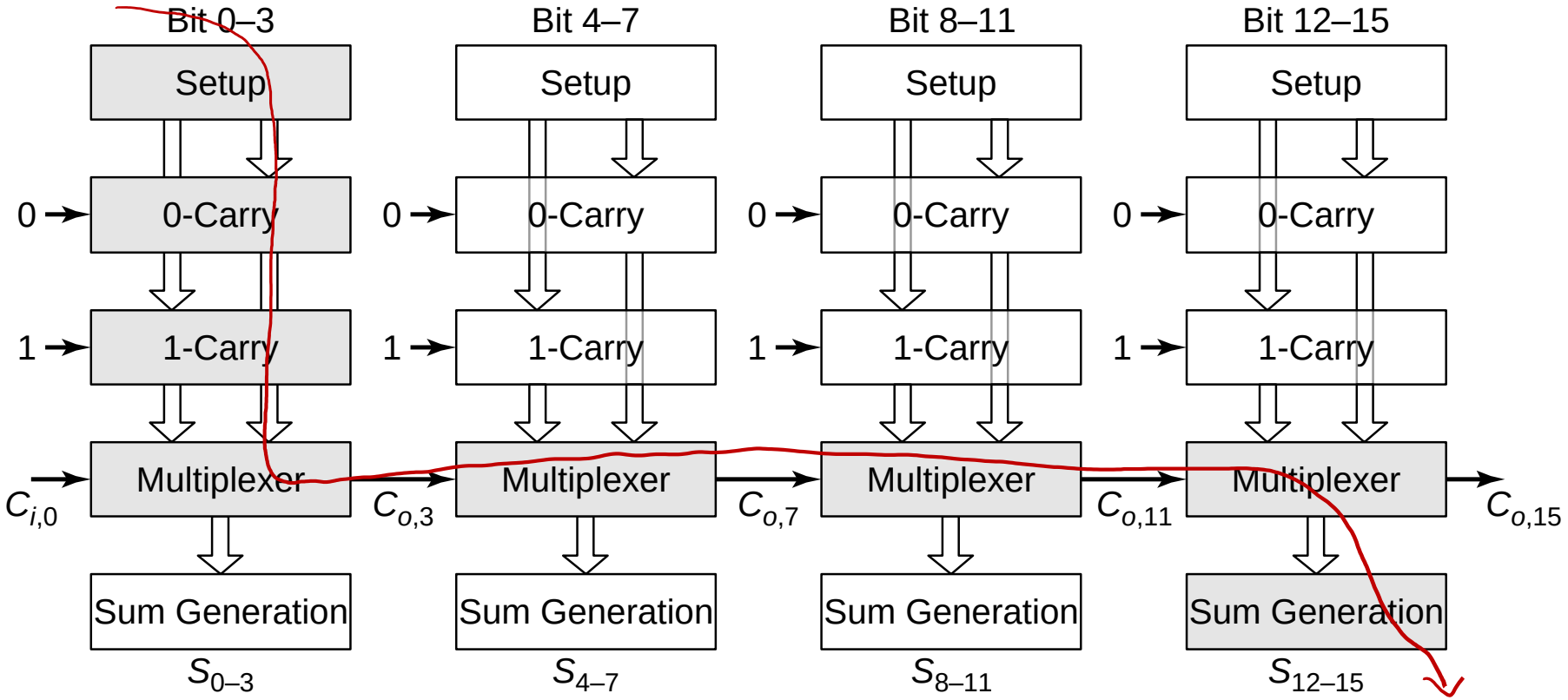
- **Calculate answers for a group assuming C_i=1 and C_i=0**
  - Select the appropriate output based on carry calculated from prev. stage (A.K.A Conditional Sum)
- **Similar result as Carry Bypass**
  - Break into segments, delay is a function of the # segments
  - Linear but flatter delay vs. number of bits



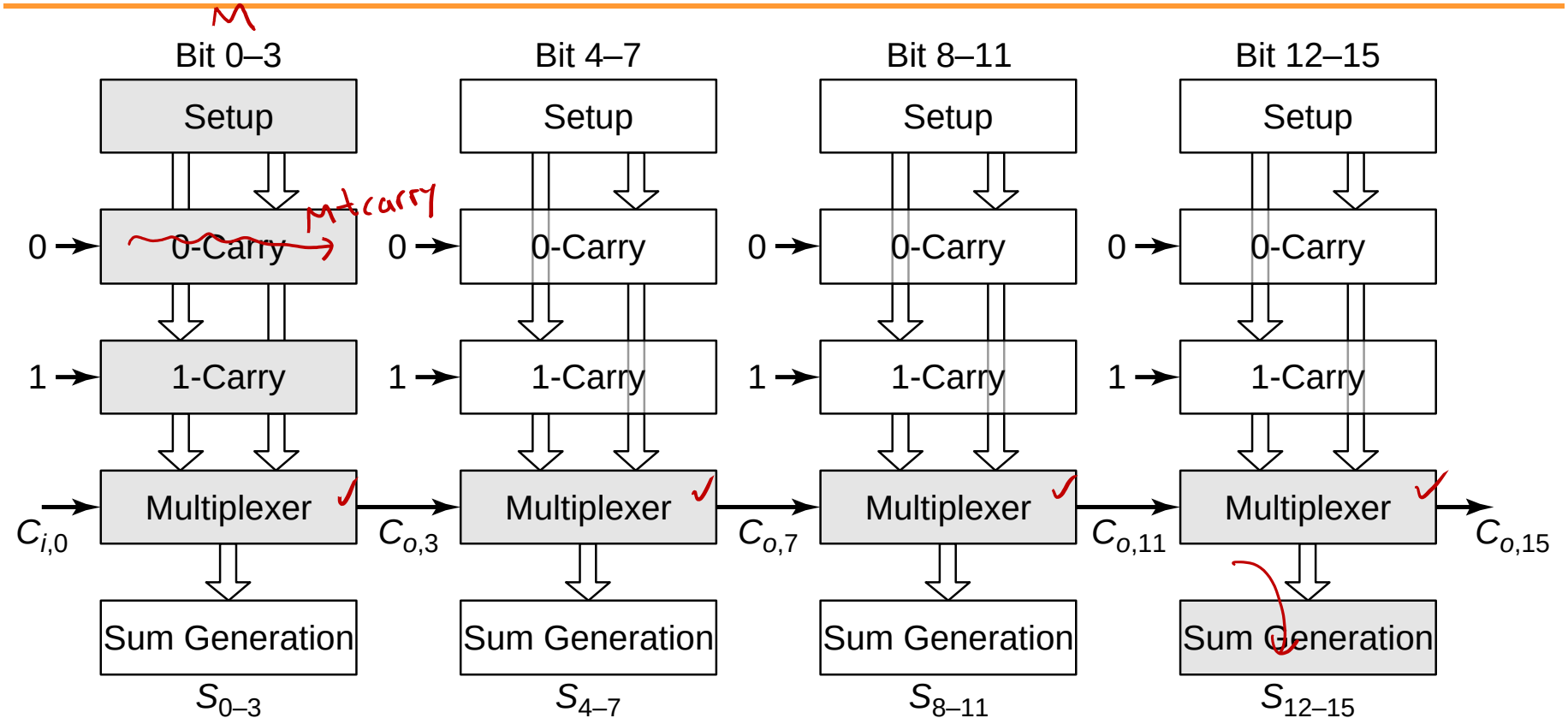


Critical Path in Linear Carry Select Adder

- **Uniform group size is not good from a timing perspective**
  - Increasing timing slack toward MSB bits

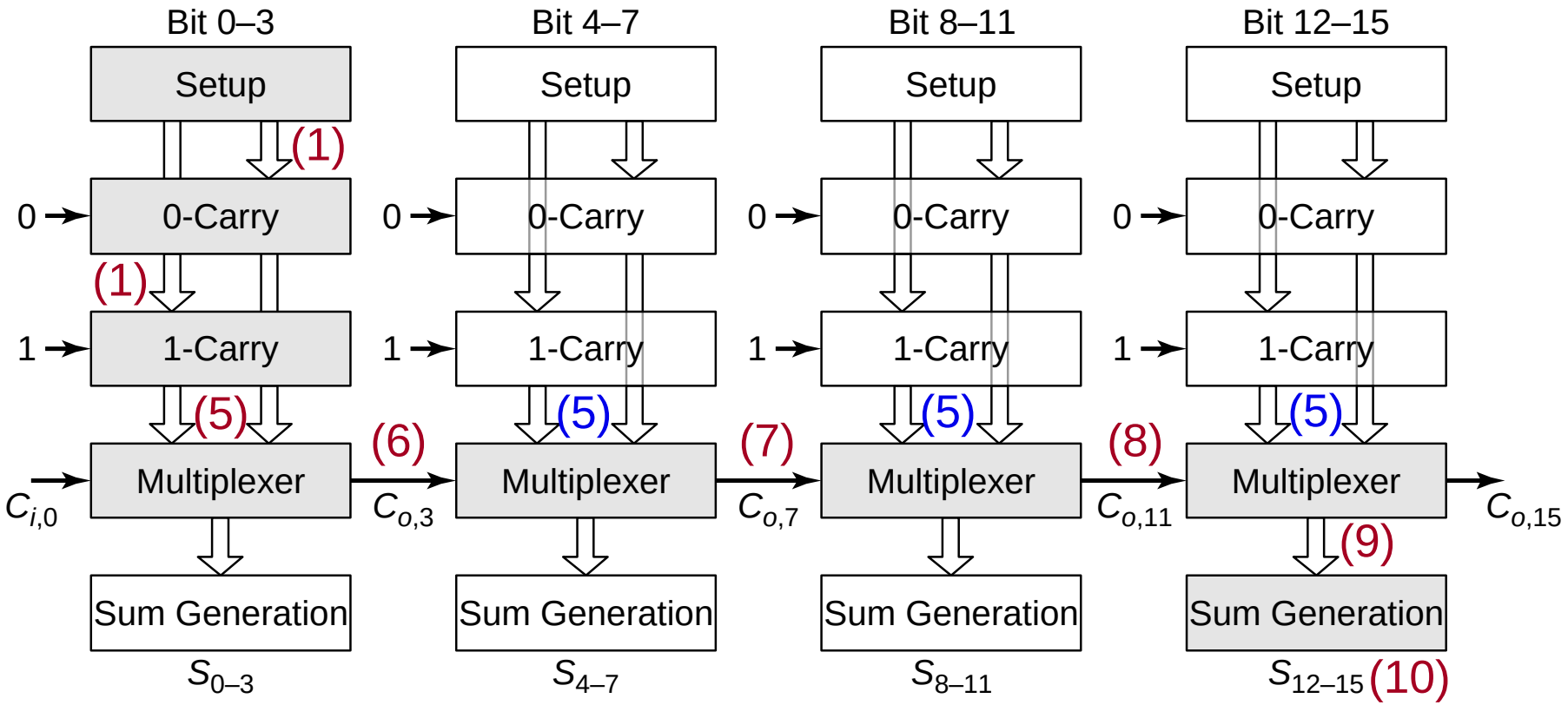


Carry Select Adder: Critical Path



$$t_{add} = t_{setup} + Mt_{carry} + \frac{N}{M}t_{mux} + t_{sum}$$

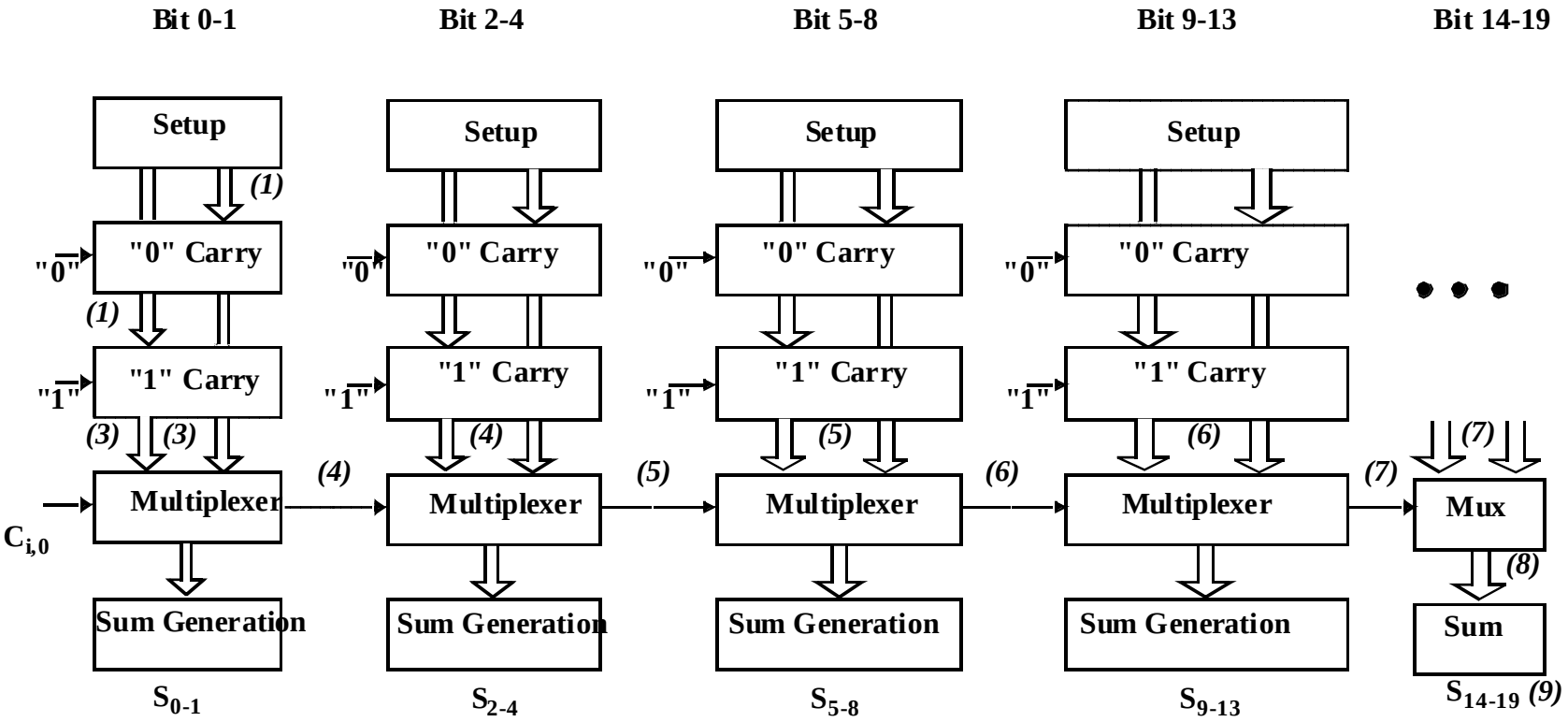
Linear Carry Select



$$t_{adder} = t_{setup} + Mt_{carry} + (N/M)t_{MUX} + t_{sum}$$

Square Root Carry Select

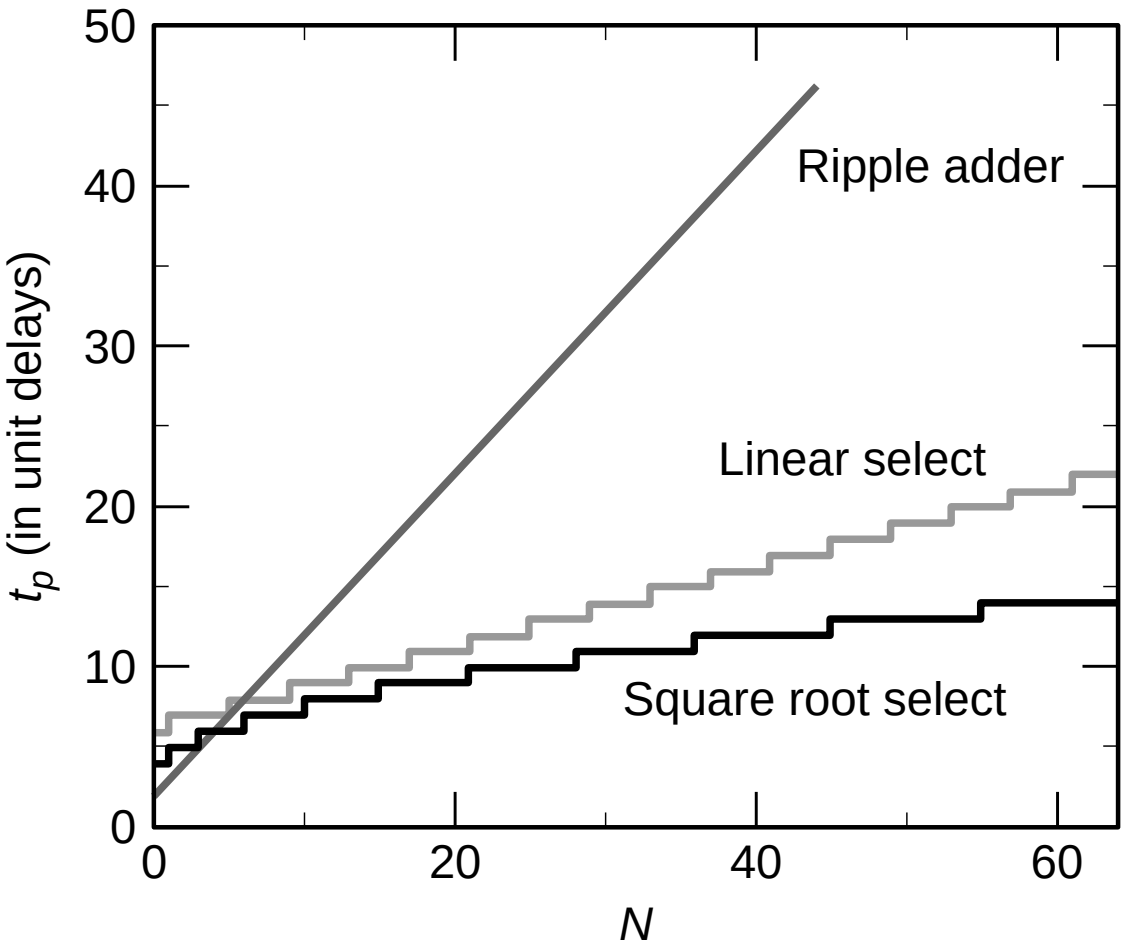
- Increase group size toward MSBs to fix the slack



$$t_{add} = t_{setup} + M \cdot t_{carry} + \sqrt{2N} \cdot t_{mux} + t_{sum}$$

(N/M) in the linear adder

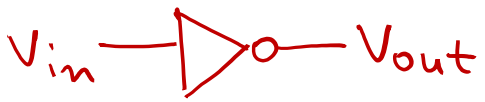
Adder Delays – Comparison

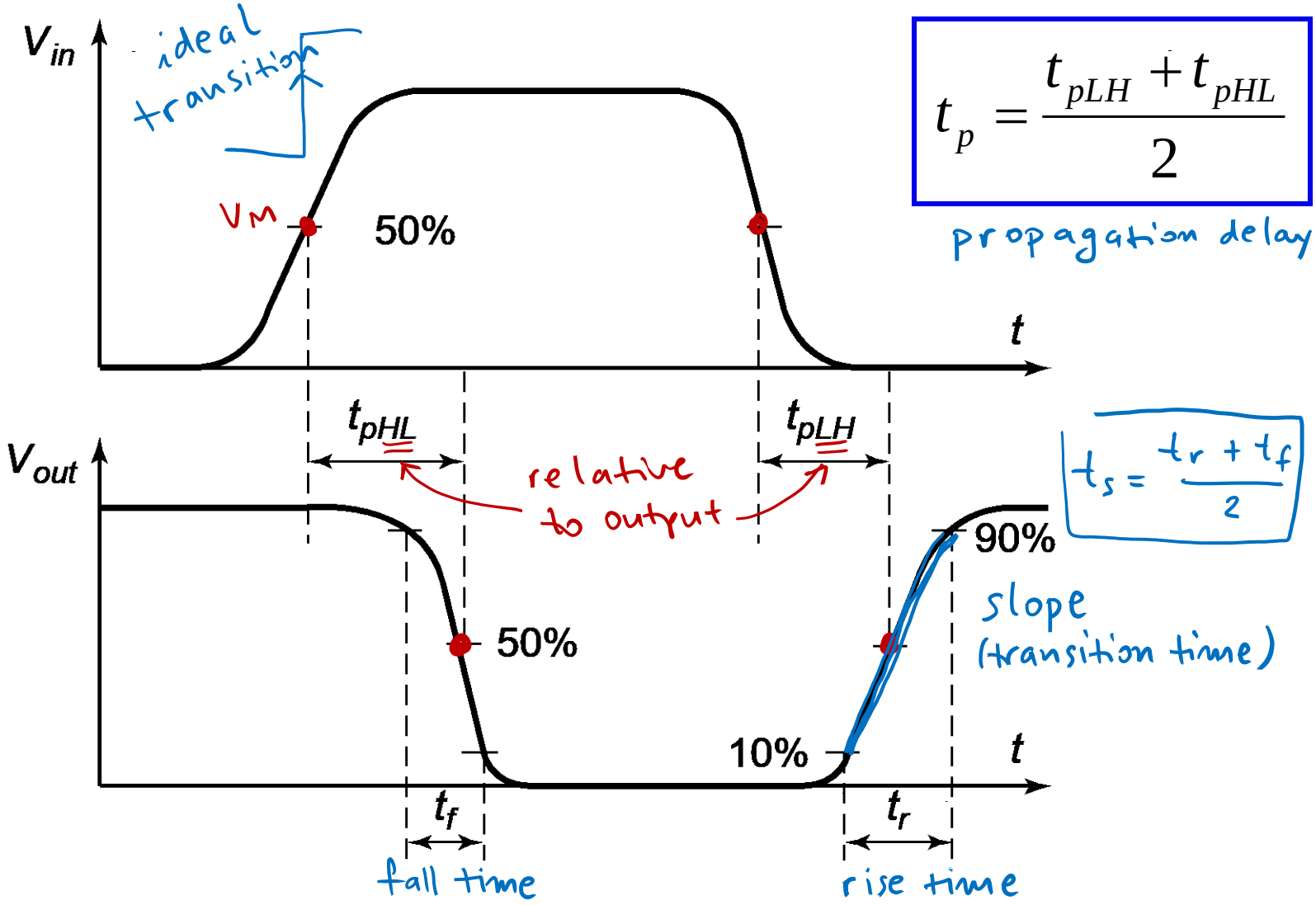


Week 5 Agenda

- Introduction to Adders
- Delay Model
- Power Model

Performance: Delay Definitions



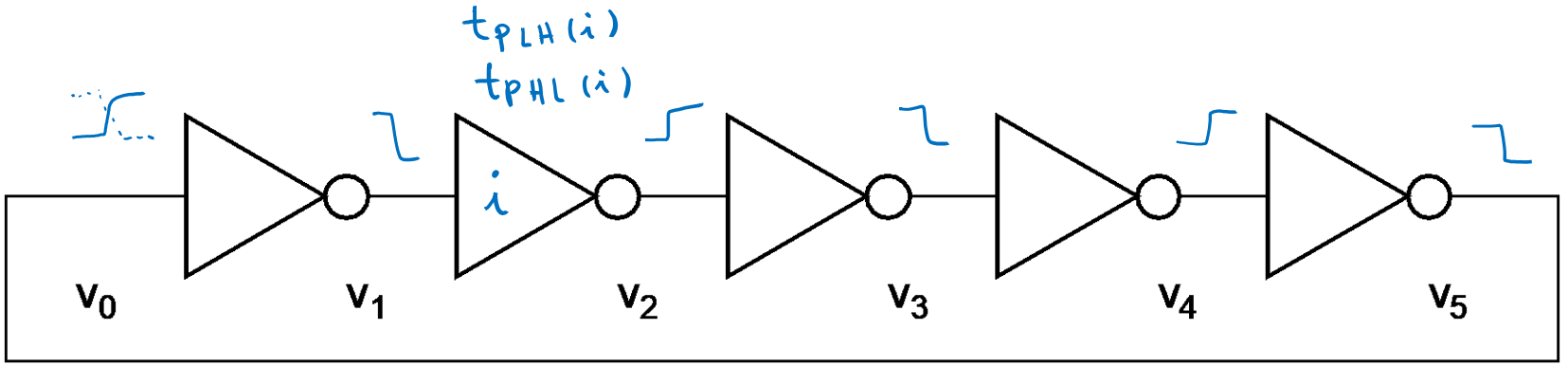


$$t_p = \frac{t_{pLH} + t_{pHL}}{2}$$

propagation delay

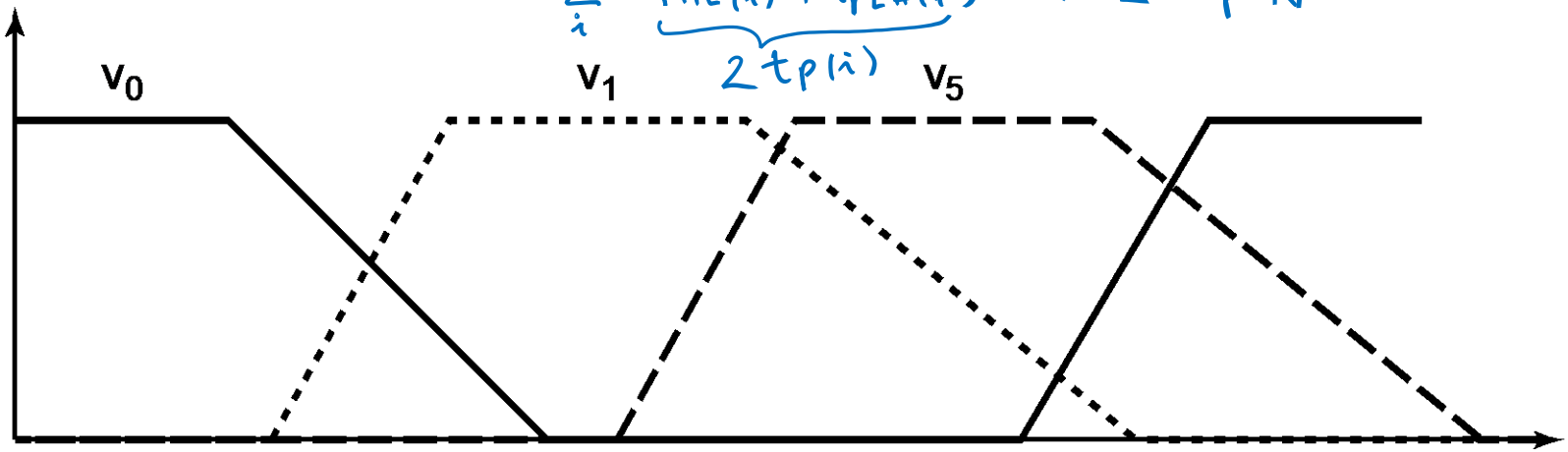
$$t_s = \frac{t_r + t_f}{2}$$

Technology Characterization: Ring Oscillator for t_p



$$T = \sum_i t_{pHL(i)} + t_{pLH(i)} \rightarrow 2\, t_p\, N$$

$2\,t_{p(i)}$



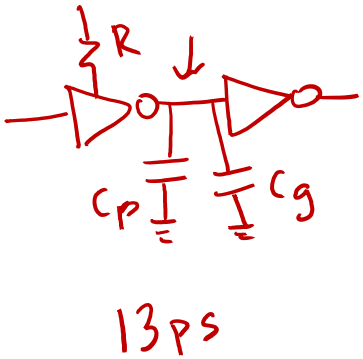
measure T by simulation

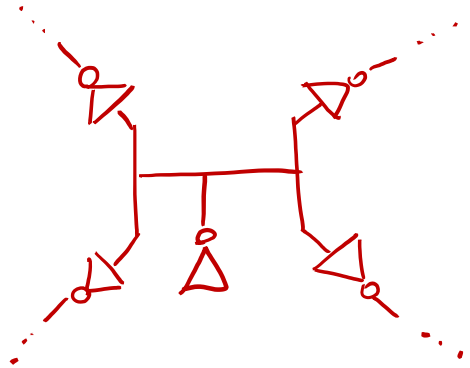
$$T = 2 \times t_p \times N$$

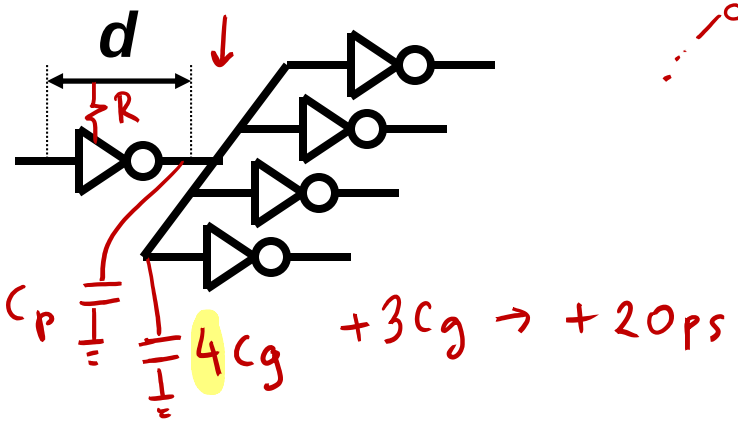
Tutorial 2: $t_p = 13ps$

Performance: FO4 Inverter

- Measures quality of design across different technology generations



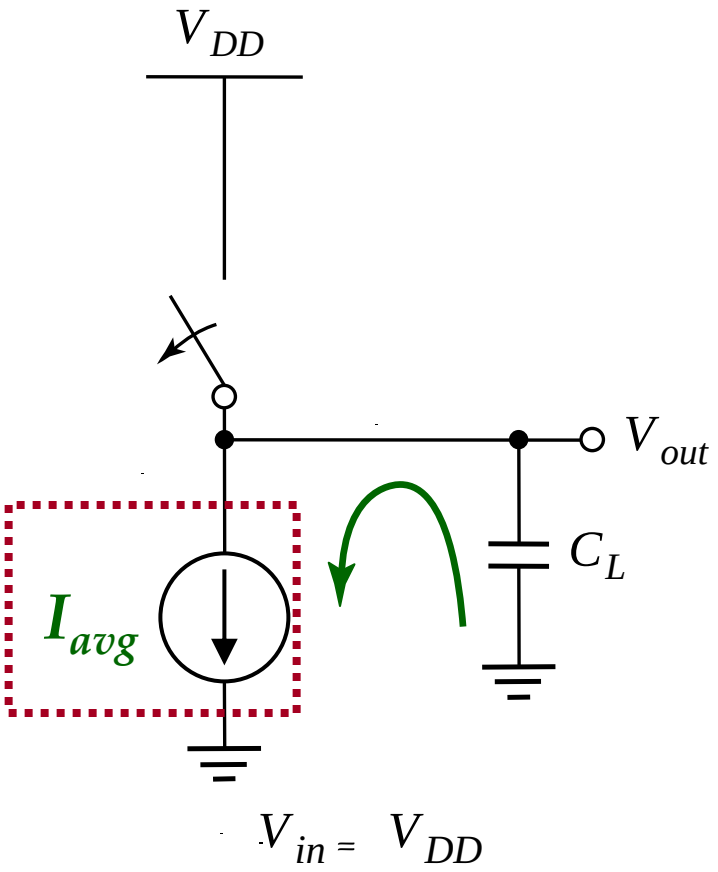




Tutorial 2: ***FO4 = 33ps***

CMOS Inverter Propagation Delay

MOS Current Model



$$t_{pHL} = \frac{C_L \cdot V_{swing}/2}{I_{avg}}$$

out: "1" → "0"

reduce delay by:

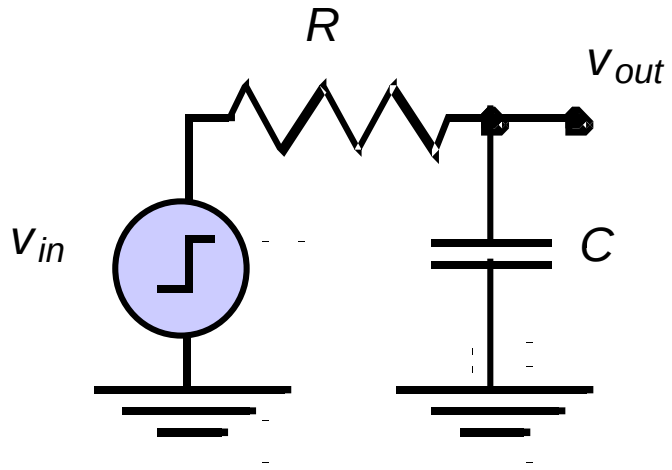
$$t_{pHL} \sim \frac{C_L}{k_n \cdot V_{DD}}$$

#1: reduce cap

#2: increase $\frac{W}{L}$ (factors into C_L)

#3: increase voltage

A First-Order RC Network: Step Response



Step response:

$$v_{out}(t) = \left(1 - e^{-t/\tau}\right) \cdot v_{in}$$

① $v_{out}(0) = V_0$

② $v_{out}(\infty) = V_\infty$

$V_0, V_\infty \in \{V_{OL}, V_{OH}\}$

Switching pt: $v_{out}(t_{pHL}^{pLH}) = V_M$

$\tau = RC$

$t=\infty: e^{-t/\tau} = 0$

$$v_{out}(t) = V_\infty + (V_0 - V_\infty)\, e^{-t/\tau}$$

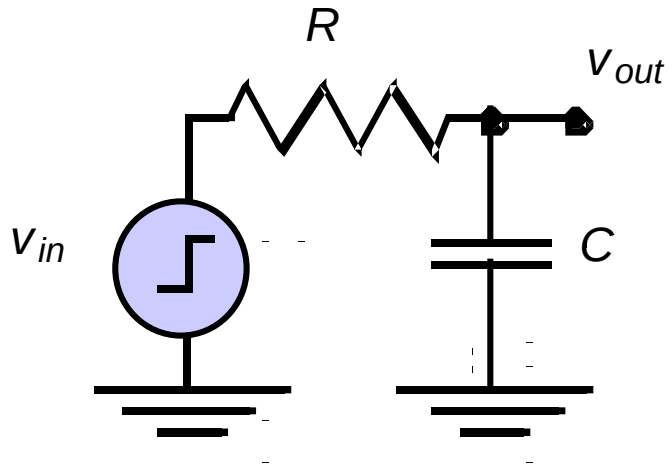
GENERAL FORMULA

① $t=0: e^{-t/\tau} = 1$

special case: $V_\infty = V_{DD}$ & $V_0 = 0$

↳ $v_{out}(t) = V_{DD}\left(1 - e^{-t/\tau}\right)$

A First-Order RC Network: Propagation Delay



Propagation delay:

$$t_p = \tau \cdot ln2 = 0.69RC$$ ✓

$t_s \sim 2.2\ RC$ slope 90% point

$v_{out}(t) = V_\infty + (V_0 - V_\infty)\, e^{-t/\tau}$

$v_{out}(t_p) = V_M$

$V_M = V_\infty + (V_0 - V_\infty)\, e^{-t_p/\tau}$

$$t_p = \tau \ln \frac{V_0 - V_\infty}{V_M - V_\infty}$$ GENERAL

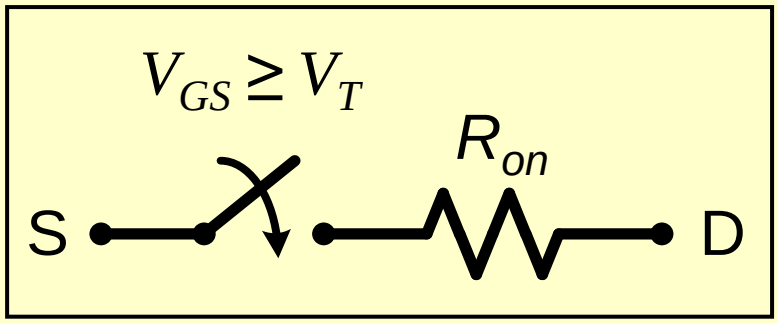
special case:

$V_M = \frac{V_{DD}}{2}$

$V_0 = 0,\ V_\infty = V_{DD}$

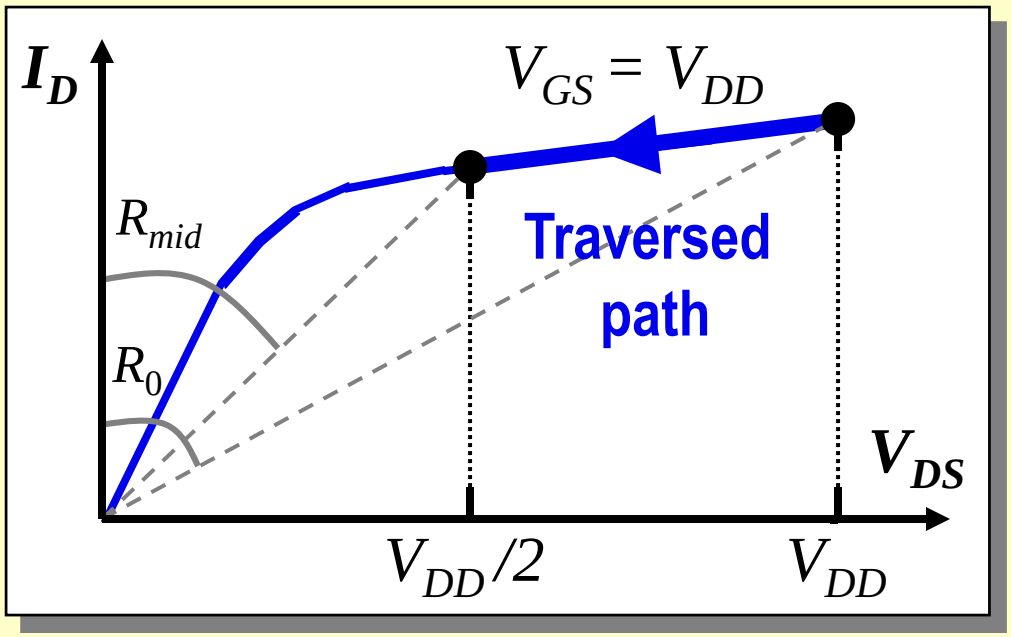
$t_p = \tau \ln \frac{-V_{DD}}{-V_{DD}/2} = \tau \ln 2 = 0.69\ RC$

Review: Transistor as a Switch



$$R_{on} \approx \frac{1}{2}(R_0 + R_{mid})$$

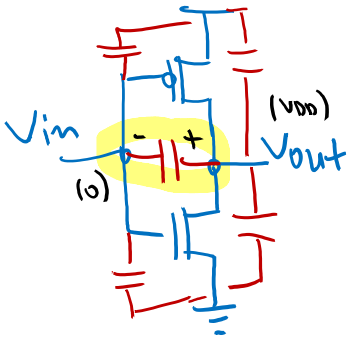
good approximation (I-V ≈ linear)

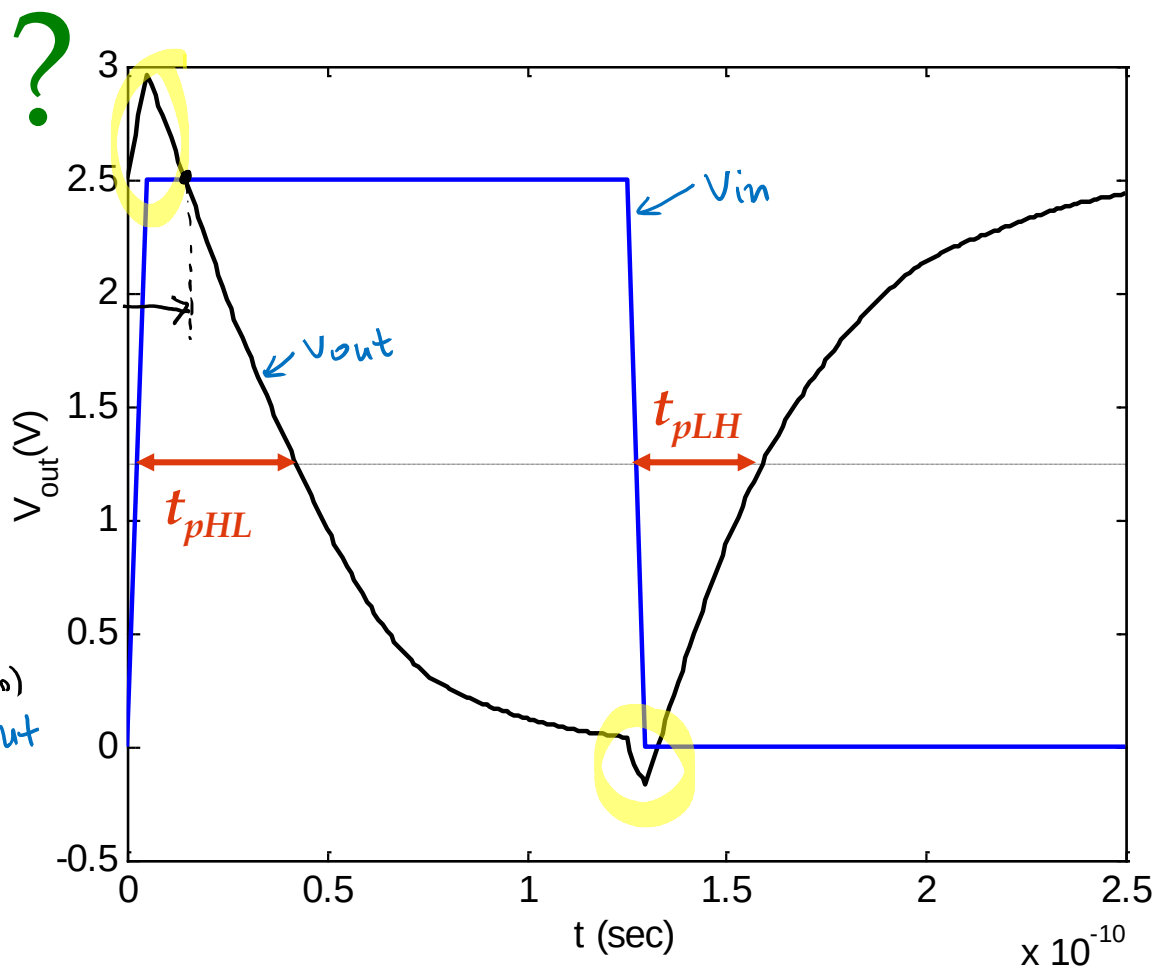


$$R_{on} = \frac{1}{2}\left(\frac{V_{DD}}{I_{DSAT} \cdot (1 + \lambda V_{DD})} + \frac{V_{DD}/2}{I_{DSAT} \cdot (1 + \lambda V_{DD}/2)}\right)$$

$$R_{on} \approx \frac{3}{4}\frac{V_{DD}}{I_{DSAT}}\left(1 - \frac{5}{6}\lambda V_{DD}\right)$$

Transient Response

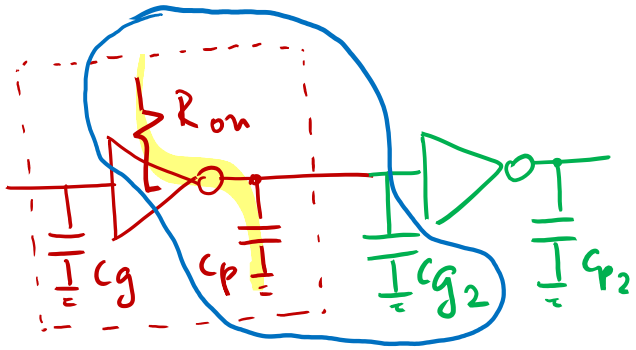




Design for Performance

- **Keep capacitances small**
- **Increase transistor sizes**
  - watch out for self-loading!
- **Increase V_{DD} (?)**

$$t_{pHL} \sim \frac{C_L}{k_n \cdot V_{DD}}$$

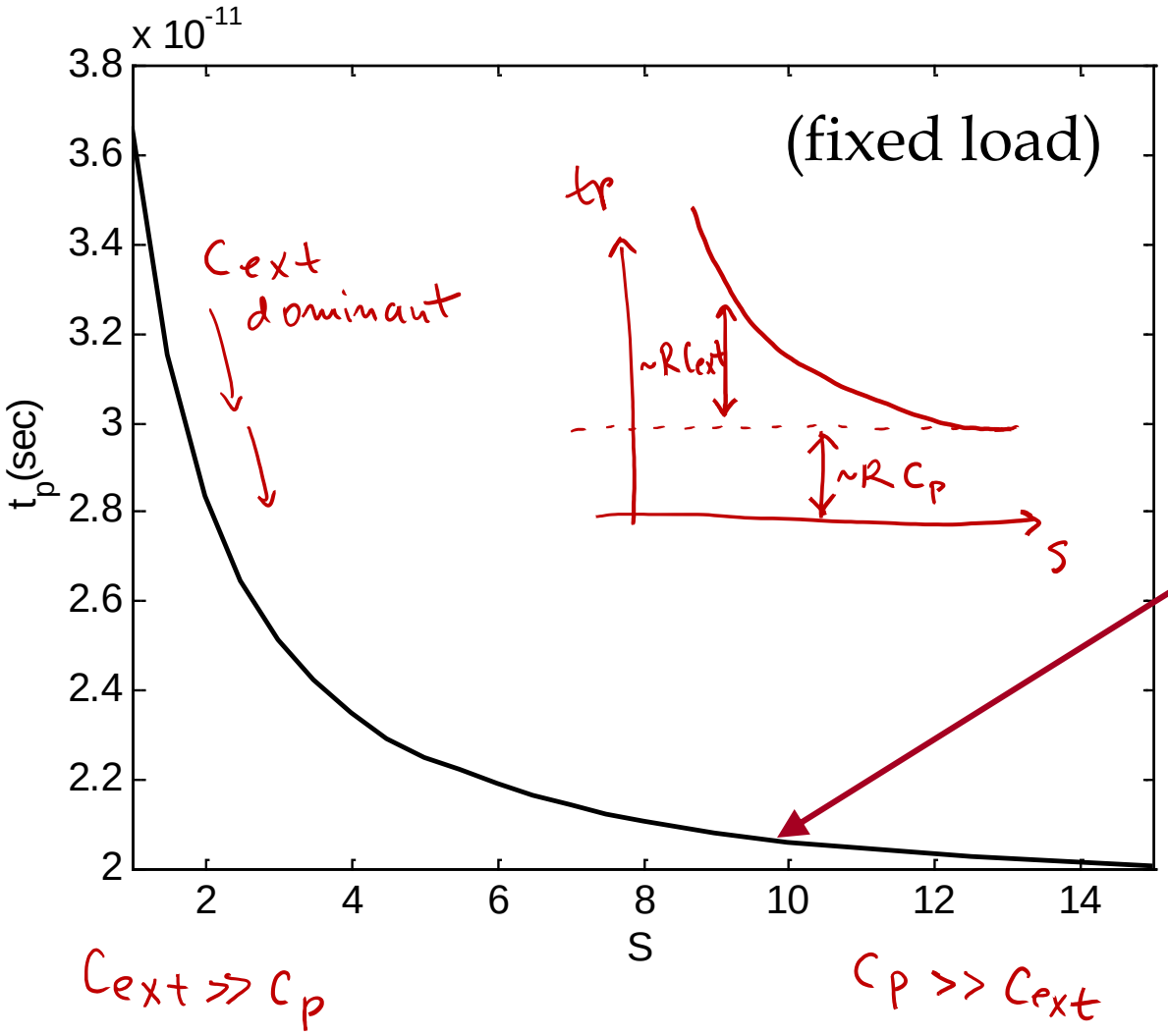


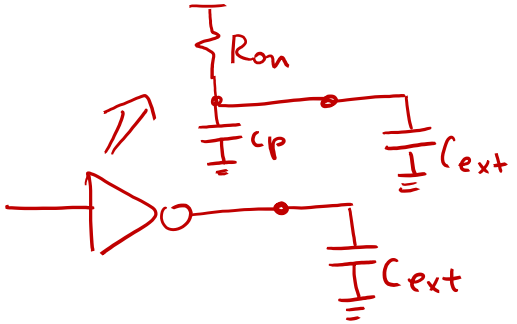
$R_{on} \sim \frac{1}{W}$

$C_p, C_g \sim W$

$R_{on} \cdot C_p \sim \frac{1}{W} \cdot W =$ constant

Device Sizing





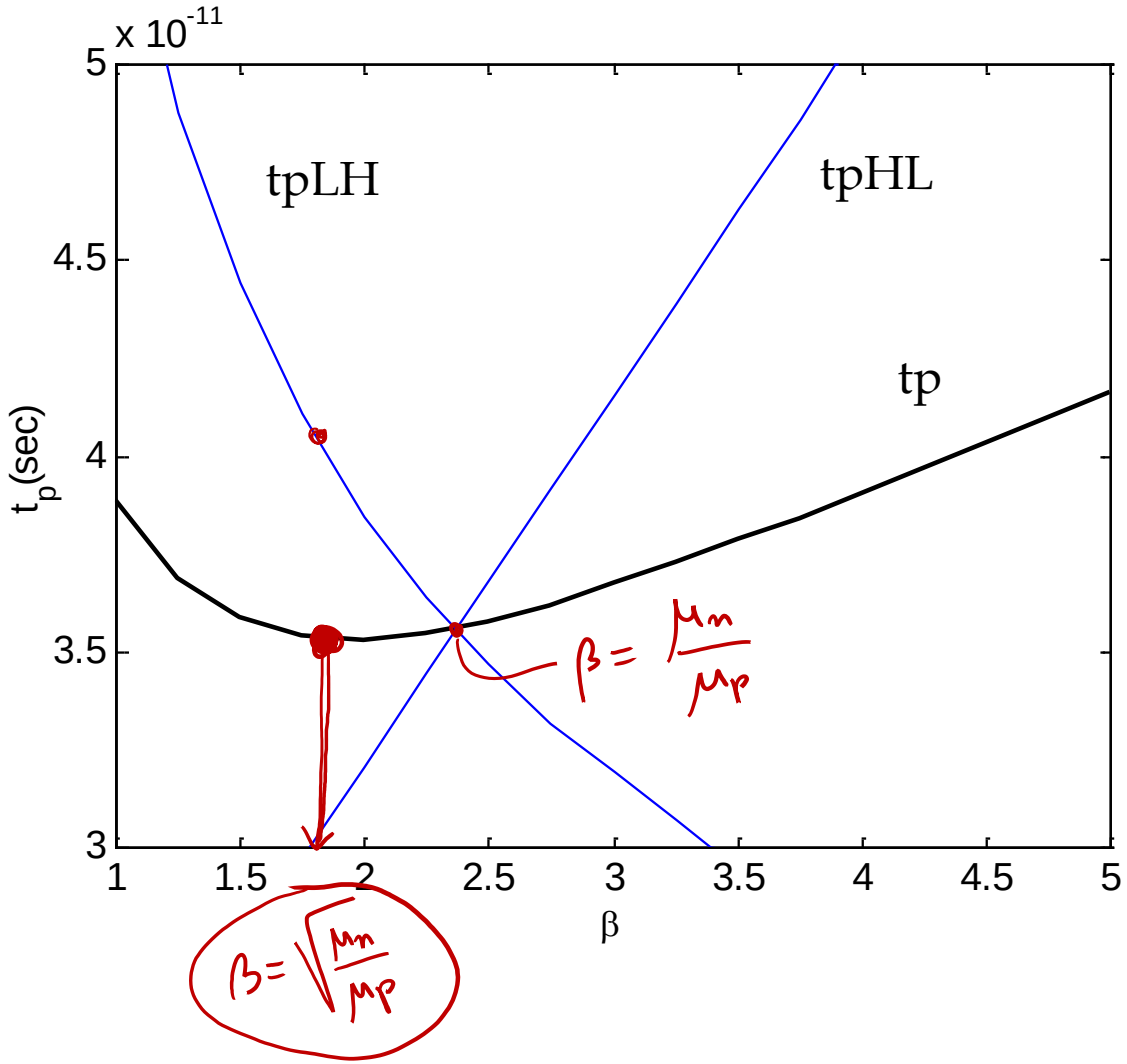
Self-loading effect: Intrinsic capacitances dominate

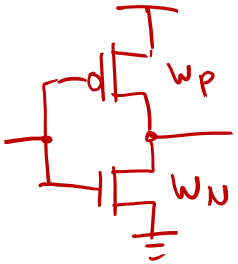
$W \rightarrow S \cdot W \uparrow$

$R \sim \frac{1}{S} \downarrow$

$C_p \sim S \uparrow$

NMOS/PMOS Ratio



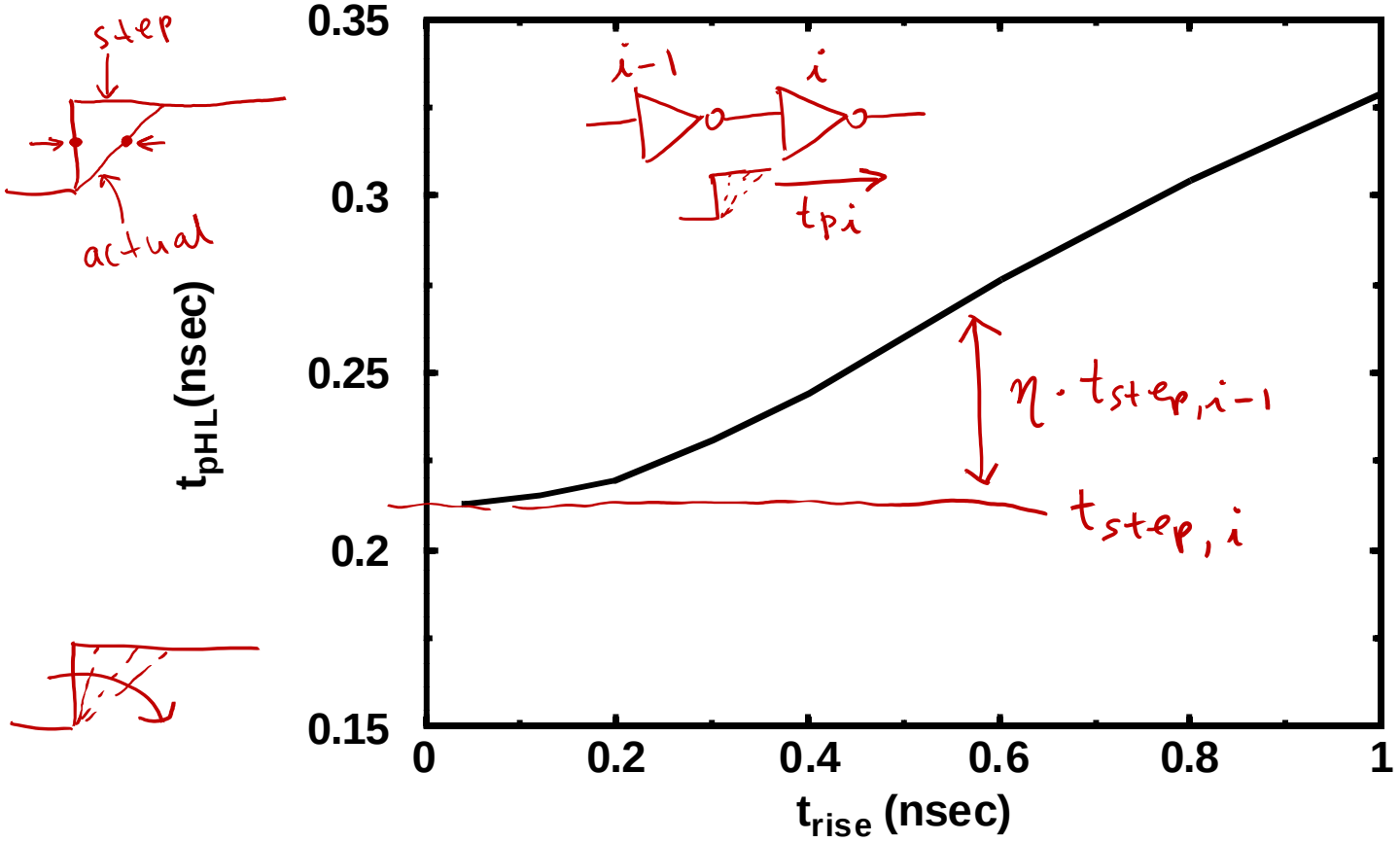


$$\beta = W_p / W_n$$

typical std-cell lib:

$\frac{W_p}{W_N} \approx 1.3 - 1.7$

Impact of Rise Time on Delay



$$t_p = t_{step(i)} + \eta \cdot t_{step(i-1)}$$

$t_p \sim 0.69RC$

$t_s \sim 2.2RC$

Simplified Macro Model

- **Consider two macro capacitances**
  - Input gate capacitance, C_{in} (or C_{gate})
  - Output parasitic (self-loading capacitance), C_{par}

- **Assume that both capacitances are linearized**
  - C_{in} and C_{par} are proportional to W
    (remember, we keep L at L_{min}, so it is lumped into constant)
  - In our 90nm technology, C_{par} / C_{in} is about 0.6

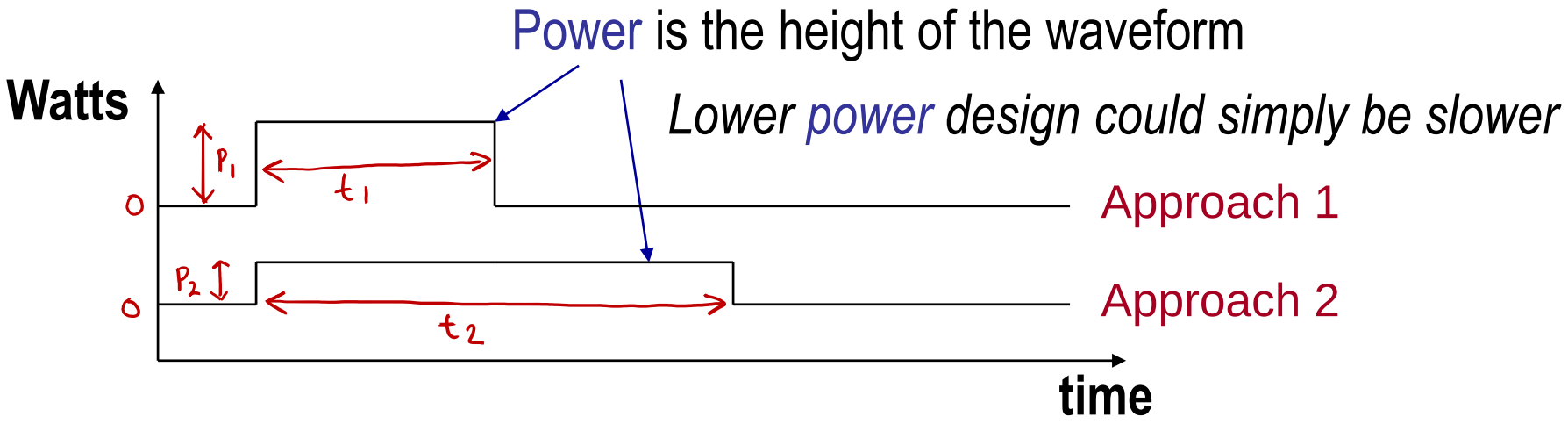
- **For gate delay analysis, we will use:**

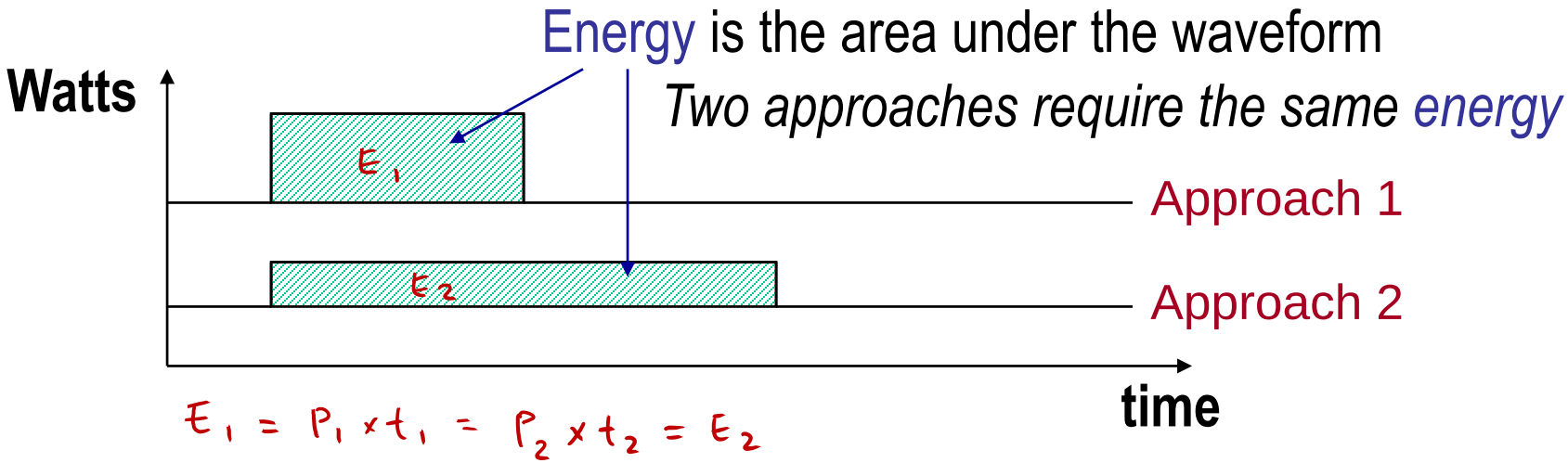
$$C_{in} = 2fF/\mu m \qquad C_{par}/C_{in} = 0.61$$

Week 5 Agenda

- Introduction to Adders
- Delay Model
- Power Model

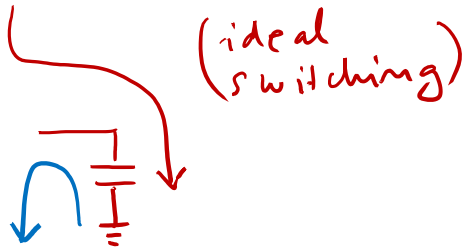
Power versus Energy



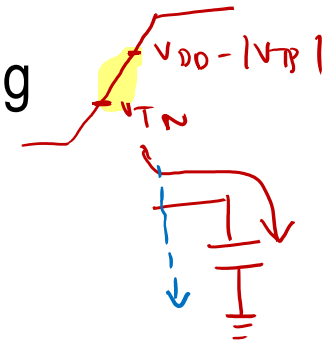


Where Does Power Go in CMOS?

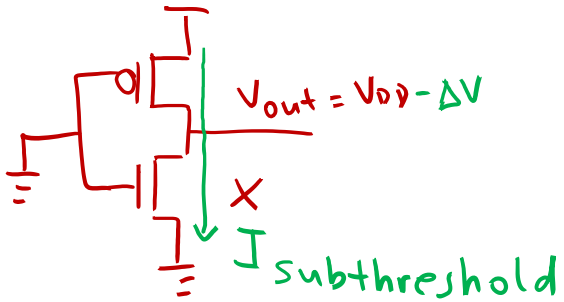
- **#1: Dynamic Power Consumption** ~75%
  - Charging and discharging capacitors



- **#2: Short Circuit Currents** ~5%-10%
  - Short-circuit path between supply rails during switching



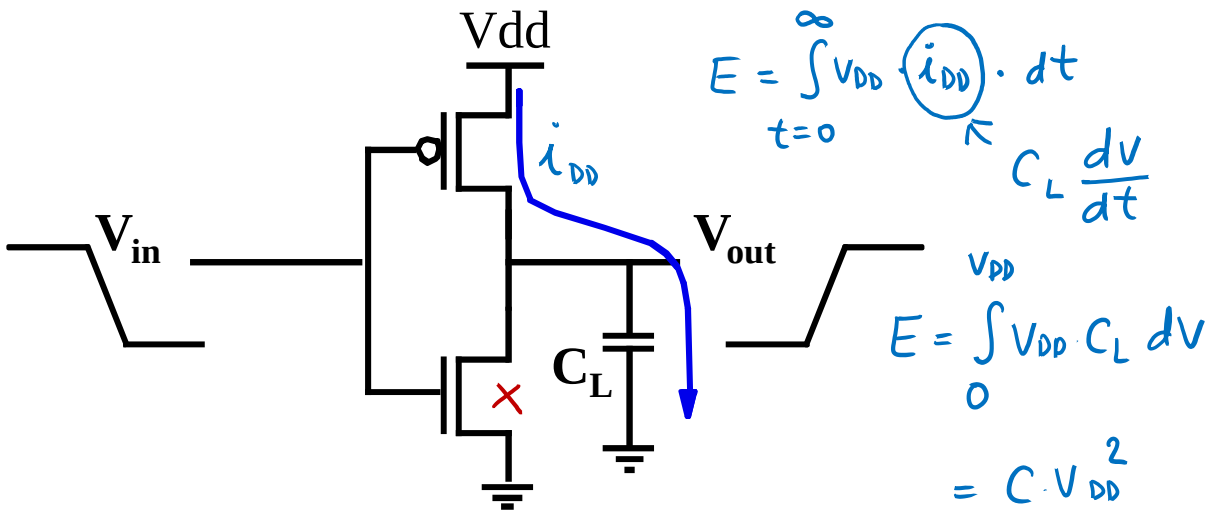
- **#3: Leakage Currents** ~15%-20%
  - Leaking diodes and transistors



switching related

static

#1: Dynamic Power Dissipation



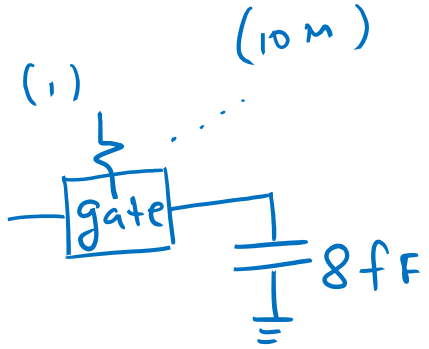
$$\text{Energy/transition} = C_L \cdot V_{dd}^2$$

$$\text{Power} = \text{Energy/transition} \cdot f = f_{0\to1} \cdot C_L \cdot V_{dd}^2$$

- **Dynamic power: observations**
  - Not a function of transistor sizes!
  - Need to reduce C_L, V_{dd}, and f to reduce power

Example

- **Parameters** $V_{DD} = 1\,V$
  - Switched capacitance: 2fF / gate
  - Fanout 4 gates $\Rightarrow C_L = 8\,fF$
  - Clock frequency: 2.5 GHz



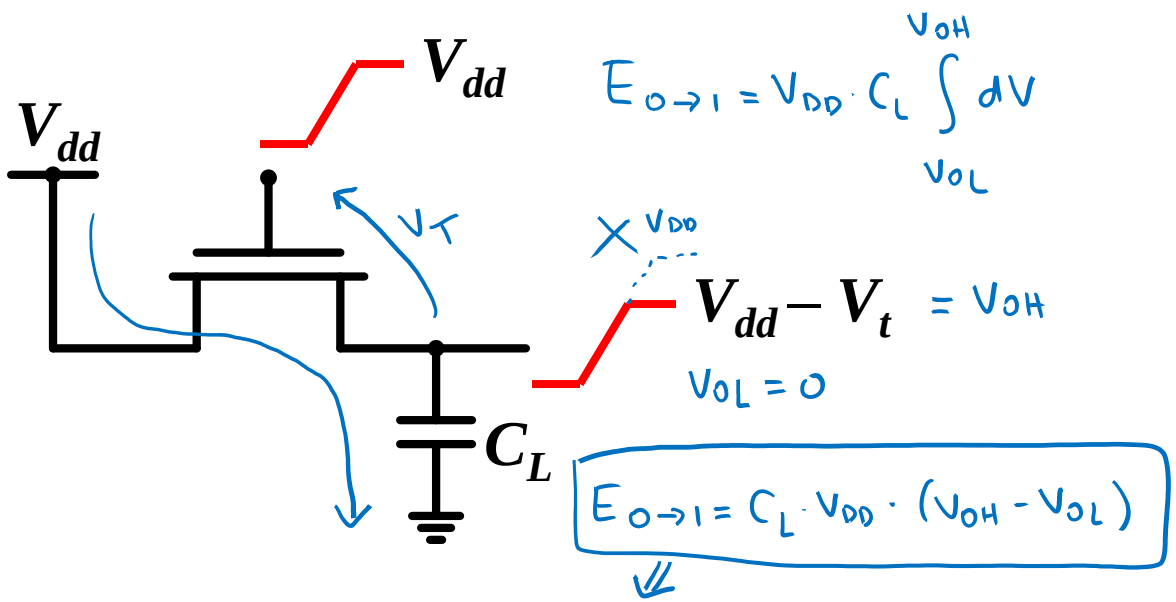
- **Power per gate**

$$P = f_{CLK} \cdot C_L \cdot V_{DD}^2 = 2.5 \times 10^9 \cdot 8 \times 10^{-15} = 20\,\mu W$$

- **Now, with many gates**
  - Activity: 0.1 $= \alpha_{0\to1}$
  - 10 M gates $= N$

$$P_{tot} = \underbrace{\alpha_{0\to1} \cdot f_{CLK}}_{f_{0\to1}} \cdot (C_L \cdot N) \cdot V_{DD}^2 = 20\,W$$

Modification for Circuits With Reduced Swing

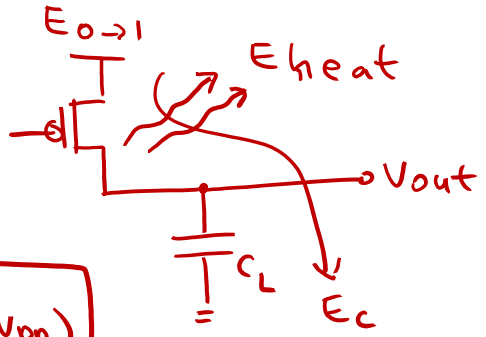


$$E_{0\to1} = C_L \cdot V_{DD} \cdot (V_{DD} - V_T) < C_L \cdot V_{DD}^2$$

- Can exploit reduced swing for lower power (e.g., reduced bit-line swing in memory)

General Formulas

- **Basic formula**
  - Energy(V_{DD}) = Energy(heat) + Energy(C_L)
- **Components:**



$$E_{0\to1}(V_{DD}) = V_{DD} \cdot C_L \int_{V_{OL}}^{V_{OH}} dV = C_L \cdot V_{DD} \cdot (V_{OH} - V_{OL}) = E_{0\to1}(V_{DD})$$

$$E_C = C_L \int_{V_{OL}}^{V_{OH}} V\, dV = \frac{1}{2} C_L \left(V_{OH}^2 - V_{OL}^2\right) = E_C$$

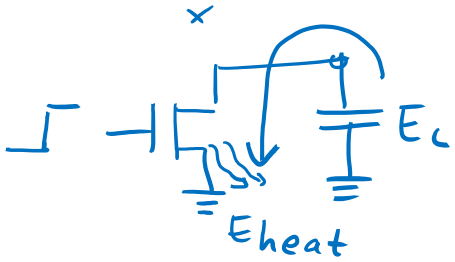
$$E_{heat} = E_{0\to1}(V_{DD}) - E_C$$

special case (CMOS): $V_{OH} = V_{DD}$, $V_{OL} = 0$

$$E_{0\to1}(V_{DD}) = C_L \cdot V_{DD}^2$$

$$E_C = E_{heat} = \frac{1}{2} C_L V_{DD}^2$$

Falling transition:



$$E_{1\to0} = 0$$

$$E_{heat} = E_C$$

Node Transition Activity and Power

- **Consider switching a CMOS gate for N clock cycles**

$$E_N = C_L \cdot V_{DD}^2 \cdot n(N)$$

E_N: the energy consumed for N clock cycles
$n(N)$: the number of 0→1 transitions in N clock cycles

$$P_{avg} = \lim_{N\to\infty} \frac{E_N}{N} \cdot f_{clk} = \left(\lim_{N\to\infty} \frac{n(N)}{N} \right) \cdot C_L \cdot V_{DD}^2 \cdot f_{clk}$$

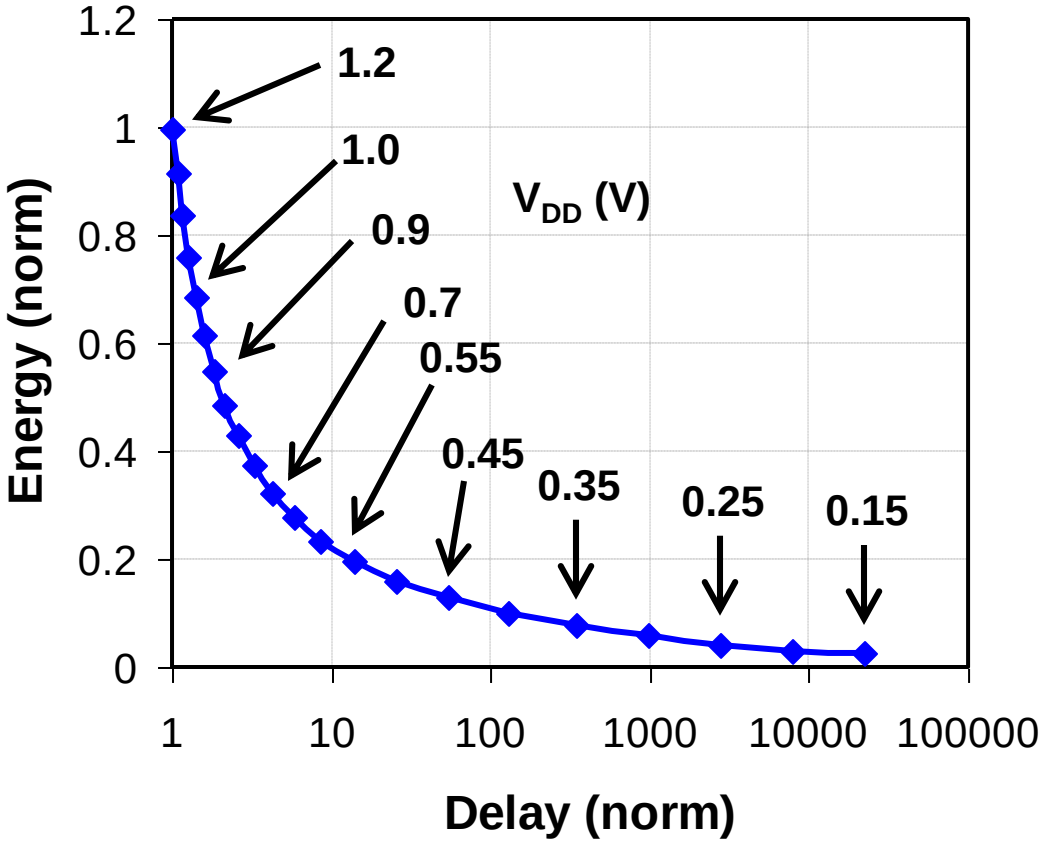
$$\alpha_{0\to1} = \lim_{N\to\infty} \frac{n(N)}{N}$$

$\alpha_{0\to1} \cdot C_L = C_{sw}$
$\alpha_{0\to1} \cdot f_{clk} = f_{0\to1}$

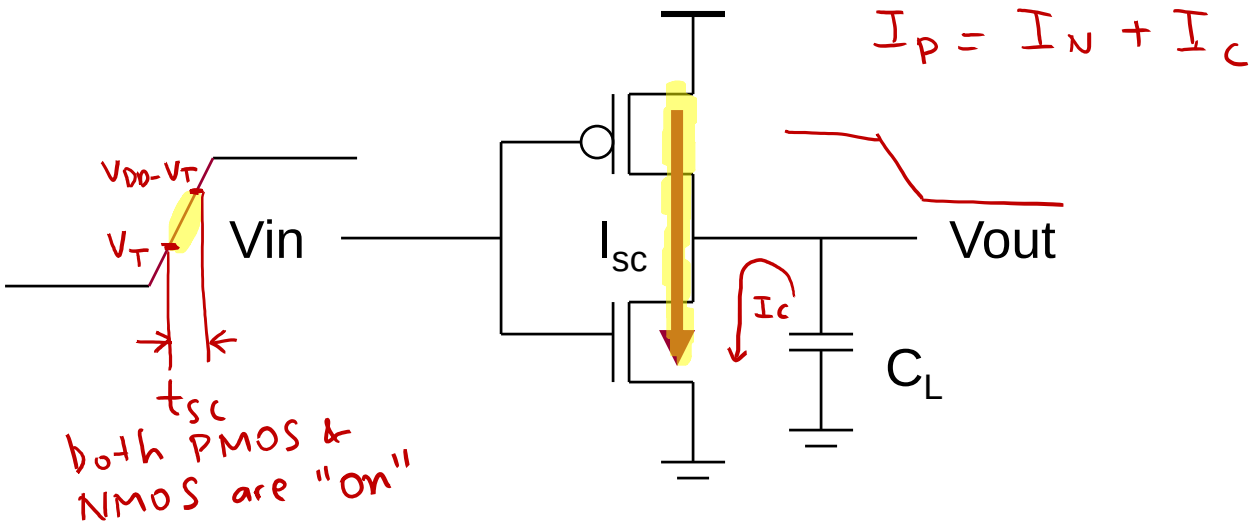
$$\boxed{P_{avg} = \alpha_{0\to1} \cdot C_L \cdot V_{DD}^2 \cdot f_{clk}}$$

Dynamic Power as a Function of V_{DD}

- Decreasing V_{DD} decreases dynamic energy consumption (quadratically)
- But, increases gate delay (decreases performance)
- Scaling into the sub-threshold regime results in very large delays (100-1000x)



#2: Short-Circuit Power Consumption

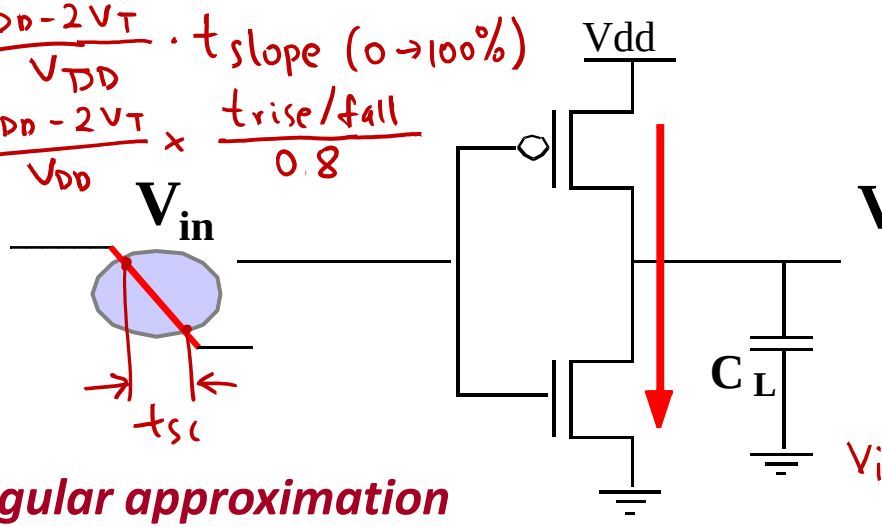


- **Short-circuit current**
  - Finite slope of the input signal causes a direct current path between V_{DD} and GND for a short period of time during switching when both the NMOS and PMOS transistors are conducting
  - **Both LH and HL transitions have short-circuit current**

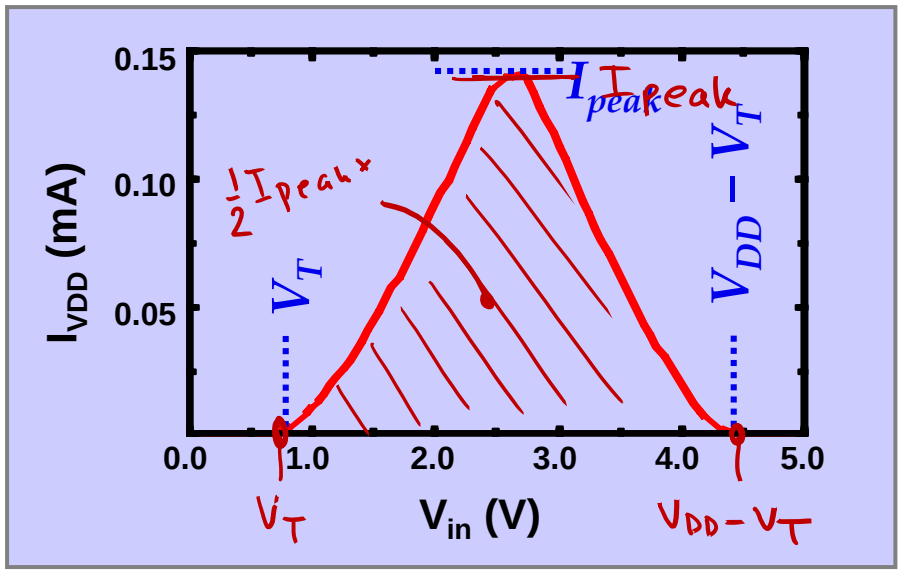
Calculating Short-Circuit Power

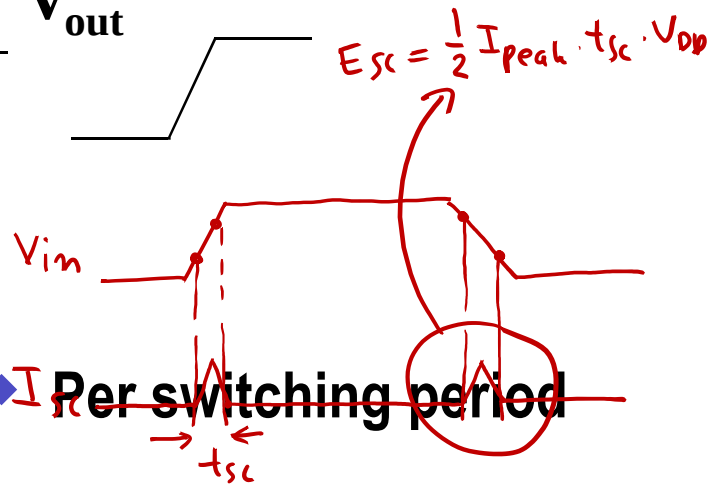
$t_{sc} = \frac{V_{DD} - 2V_T}{V_{DD}} \cdot t_{slope\ (0 \to 100\%)}$

$\approx \frac{V_{DD} - 2V_T}{V_{DD}} \times \frac{t_{rise/fall}}{0.8}$



Triangular approximation





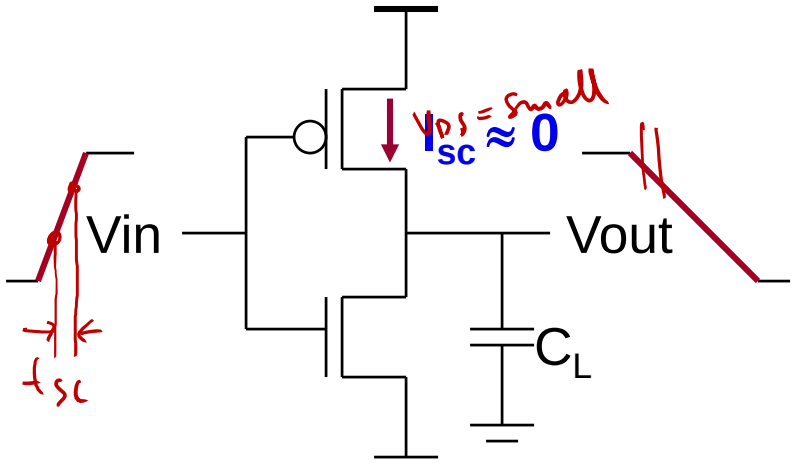
- Per switching period

$$E_{sc} = t_{sc} \cdot V_{DD} \cdot I_{peak}$$

$$P_{sc} = t_{sc} \cdot V_{DD} \cdot I_{peak} \cdot f_{0 \to 1}$$

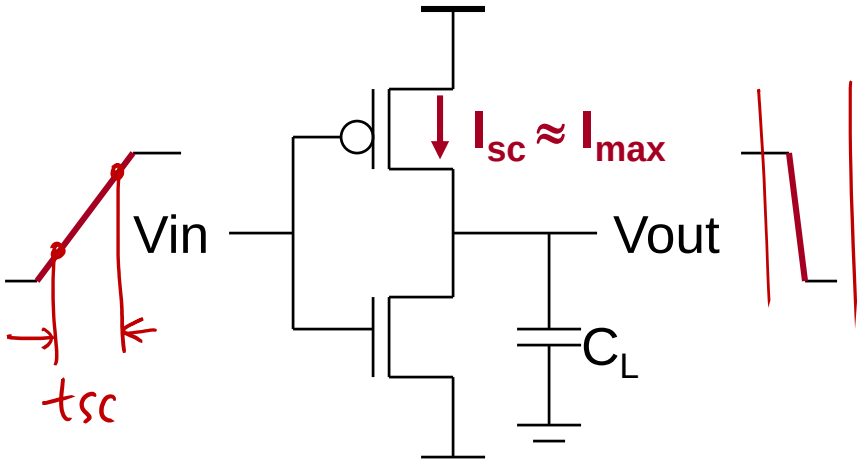
- I_{peak} depends on C_L

Impact of C_L on P_{sc}



Large capacitive load

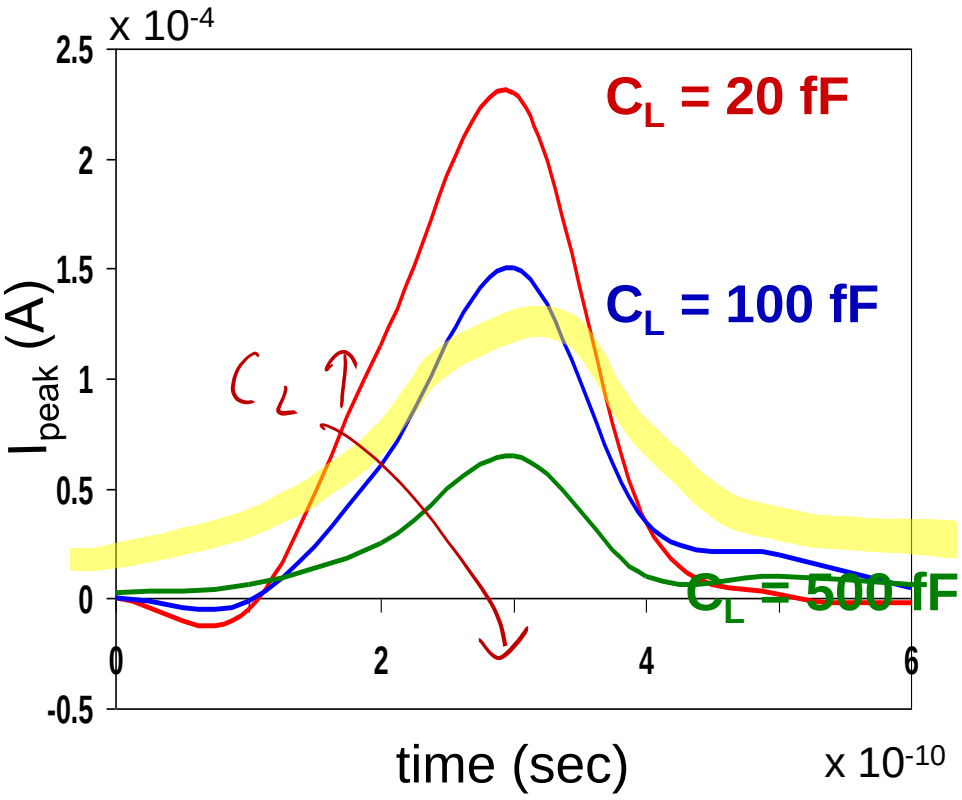
Output fall time significantly larger than input rise time.



Small capacitive load

Output fall time substantially smaller than the input rise time.

I_{peak} as a Function of C_L



500 ps input slope

Small C_L → I_{peak} is large.

⇒ Use small C_{in} and large C_L

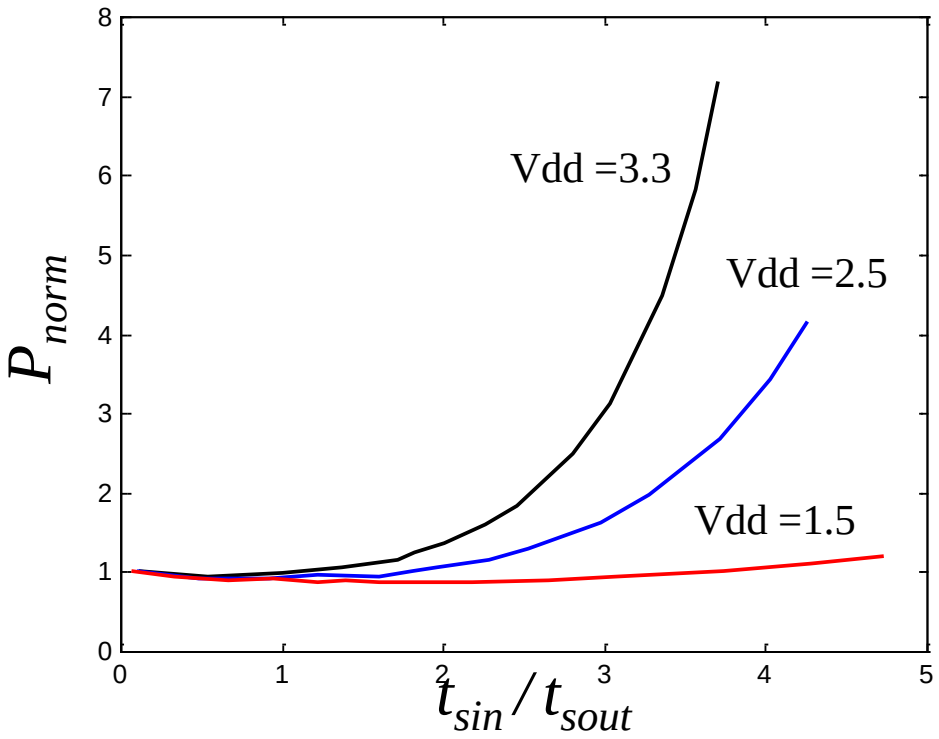
⇒ Gate delay is higher and the input rise time of the next gate (fanout gate) is higher

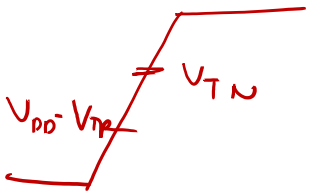
⇒ Higher I_{sc} in fanout gate!!

⇒ Local optim. is not good

Short circuit dissipation is minimized by matching the rise/fall times of the input and output signals - slope engineering.

Minimizing Short-Circuit Power



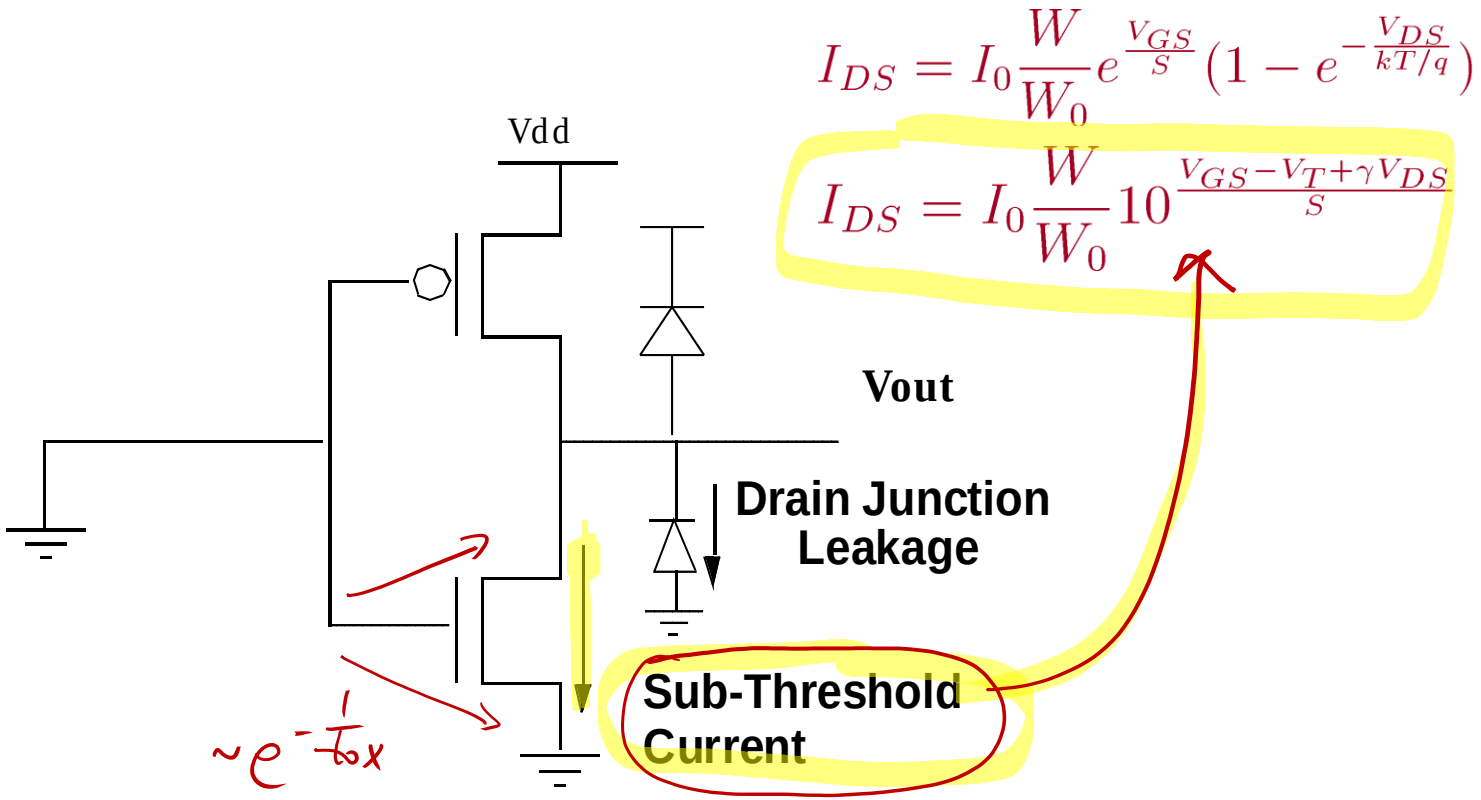


- **Keep the input and output rise/fall times the same (<10% of total consumption)**

  From: Veendrick, IEEE Journal of Solid-State Circuits, Aug'84

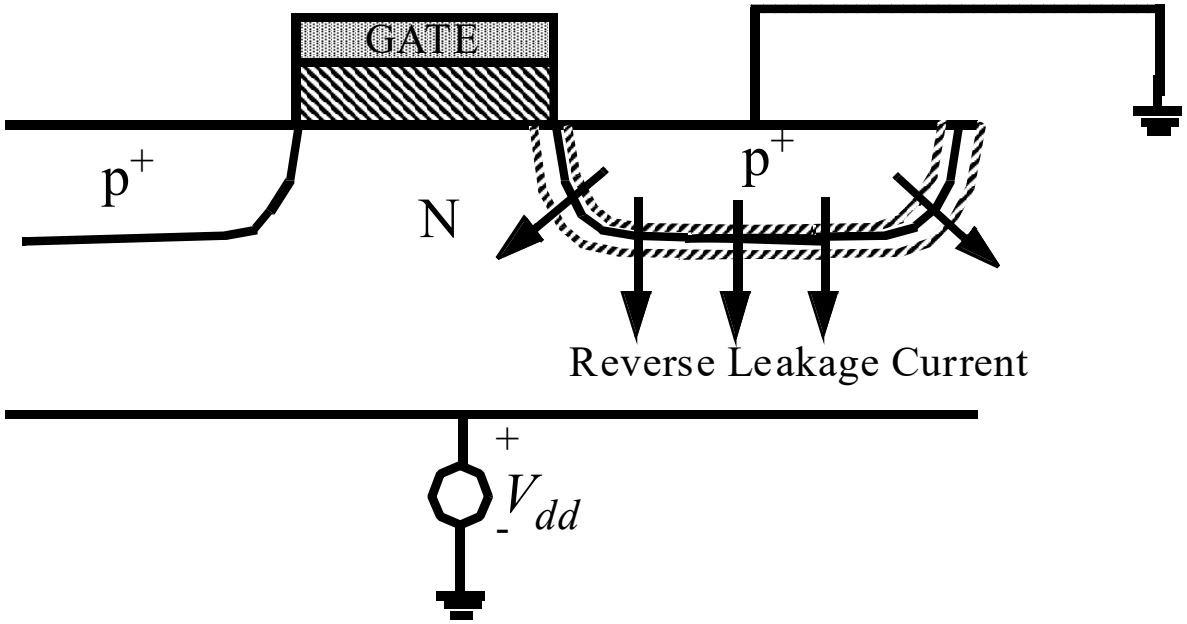
- **If $V_{dd} < V_{Tn} + |V_{Tp}|$ then short-circuit power can be eliminated!**

#3: Leakage Current



Sub-threshold current is one of the most compelling issues in low-energy circuit design!

Reverse-Biased Diode Leakage



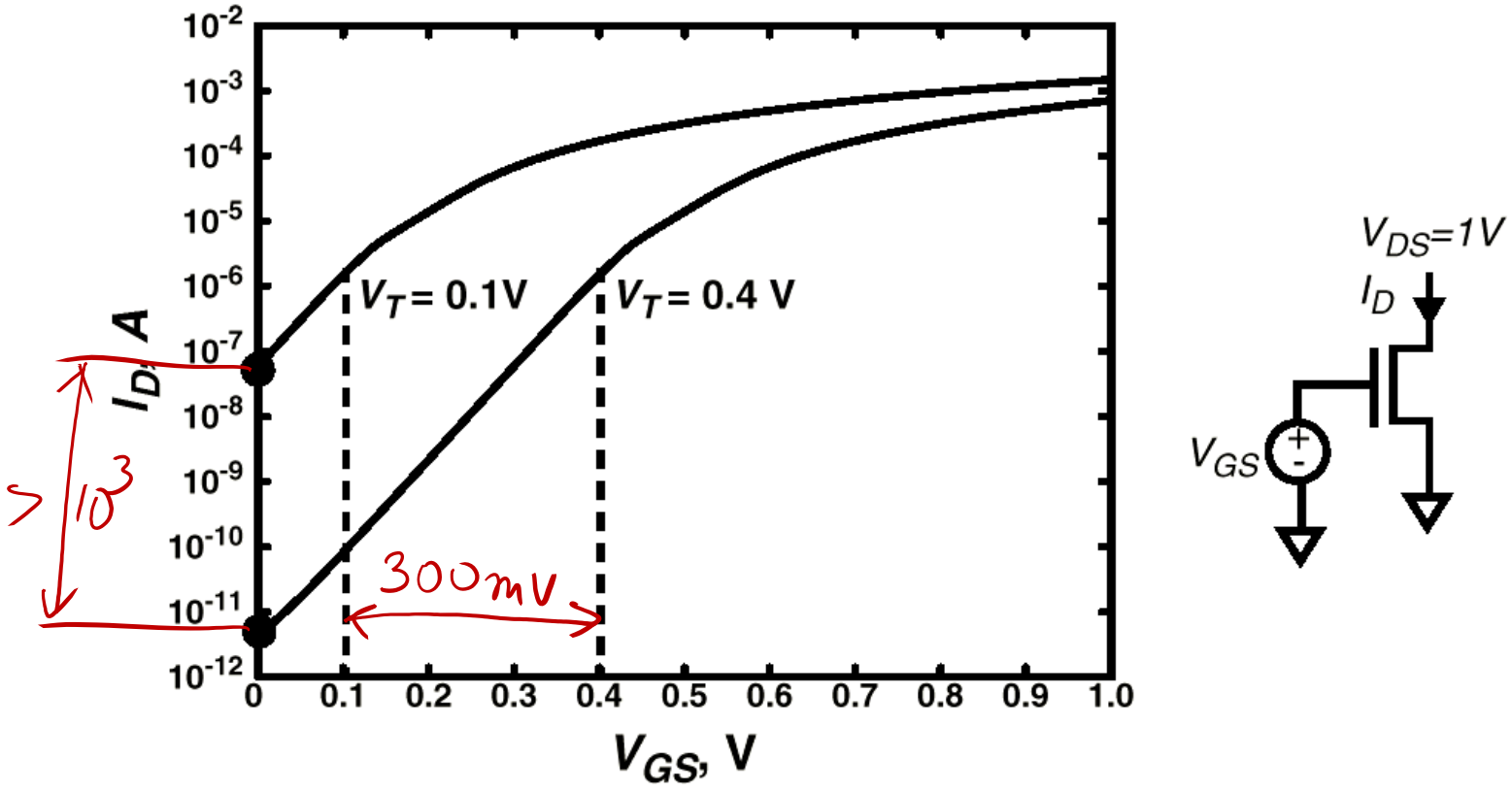
$$I_{DL} = J_S \times A \ll I_{DS}\,(\text{subthreshold})$$

J_S = 10-100 pA/μm^2 at 25 deg C for 0.25μm CMOS

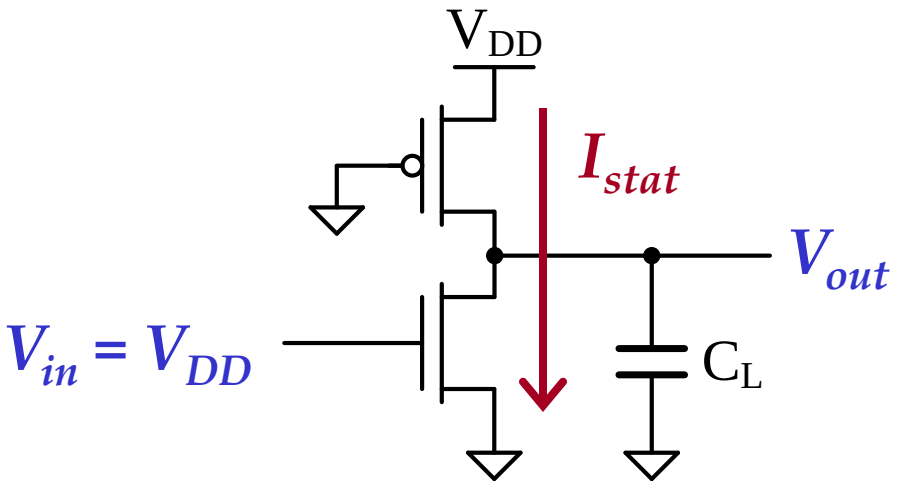
J_S doubles for every 9 deg C!

Sub-Threshold Leakage Component

- Leakage control is critical for low-voltage operation



#4: Static Power Consumption



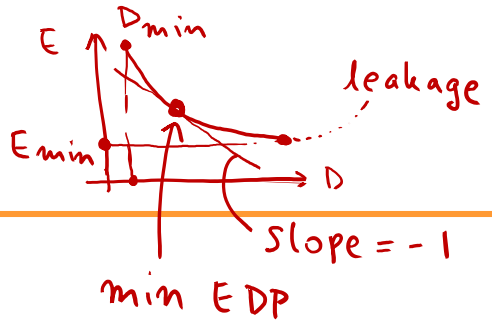
$$P_{stat} = P(in = 1) \cdot V_{DD} \cdot I_{stat}$$

Wasted energy …
Should be avoided in most cases,
but could help reducing energy in others (e.g. sense amps)

Power and Energy Figures of Merit

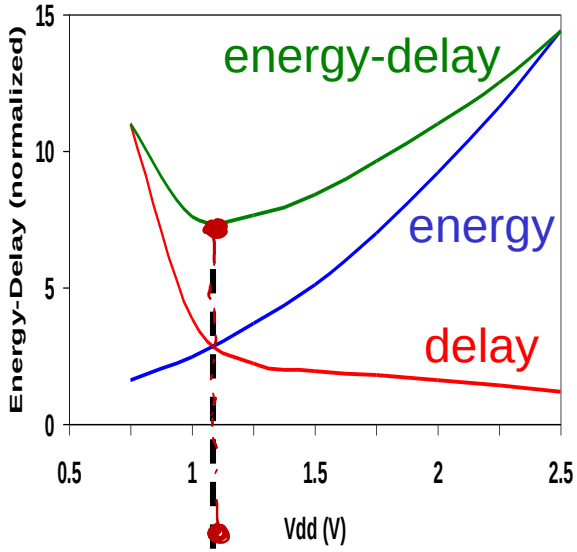
- **Power consumption in Watts**
  - determines battery life in hours
- **Peak power**
  - determines power ground wiring designs
  - sets packaging limits
  - impacts signal noise margin and reliability analysis
- **Energy efficiency in Joules**
  - rate at which power is consumed over time
- **Energy = power * delay**
  - Joules = Watts * seconds
  - lower energy number means less power to perform a computation at the same frequency

PDP and EDP



- **Power-delay product (PDP) = $P_{av} * t_p = (C_L V_{DD}^2)/2$**
  - PDP is the average energy consumed per switching event (Watts * sec = Joule)
  - **Lower power design could simply be a slower design**

- **Energy-delay product (EDP)**
  - EDP = PDP * t_p = $P_{av} * t_p^2$
  - EDP is the average energy consumed multiplied by the computation time required
  - **Takes into account that one can trade increased delay for lower energy/op (e.g. via V_{DD} scaling)**



CMOS Energy & Power Equations (Summary)

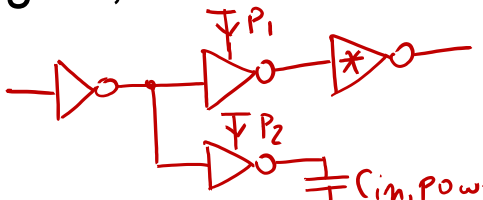
	Dynamic	Short-circuit	Leakage
Energy	$C_L \cdot V_{DD}^2$	$t_{sc} \cdot V_{DD} \cdot I_{peak}$	$V_{DD} \cdot I_{leakage} / f_{clock}$
Power	$C_L \cdot V_{DD}^2 \cdot f_{0\rightarrow1}$	$t_{sc} \cdot V_{DD} \cdot I_{peak} \cdot f_{0\rightarrow1}$	$V_{DD} \cdot I_{leakage}$
	~75% today and decreasing relatively	~5% today and decreasing absolutely	~20% today and slowly increasing

- Switching frequency / activity
  - $f_{0\rightarrow1} = \alpha_{0\rightarrow1} \cdot f_{clock}$

Simplified Model for Circuit Analysis

- **Often we assume that switching energy is dominant**
- **Similarly to delay analysis, we can find "equivalent" capacitance for power analysis**
  - It is to expect that this capacitance will be higher, because it includes short-circuit power and leakage

$P_1 = P_2 \Rightarrow C_{in,power} = C_{g,power}^{*}$



- **In our process, C_{in} (power) = 2.45fF**
  - Including output parasitic $C_{in} + C_{par}$ = 3.95fF (1.61 * 2.45fF)

- **Simplified model for hand analysis:**

power:

$$C_{in} = 2.45fF/\mu m$$

$$C_{par}/C_{in} = 0.61$$

"power" cap ≠ "delay" cap! ↑ delay: 2fF/μm