EE 330
Lecture 43

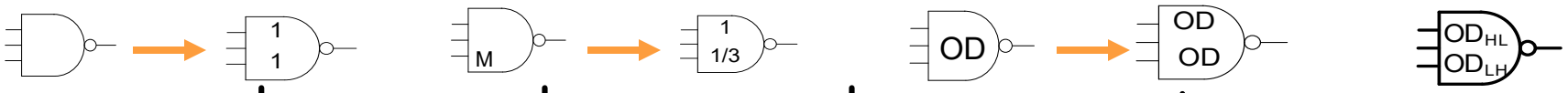
Digital Circuits

- Elmore Delay
- Power Dissipation

Fall 2025 Exam Schedule

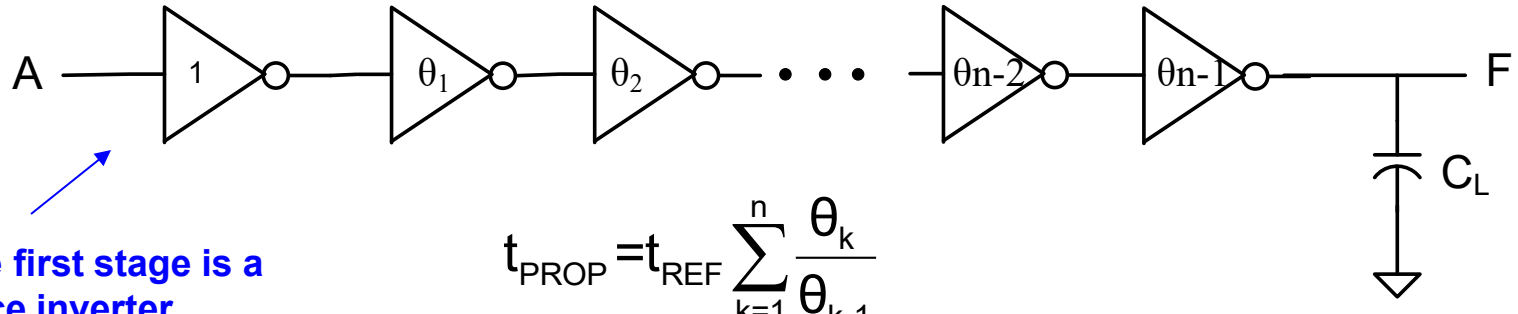
Exam 1	**Friday**	**Sept 26**
Exam 2	**Friday**	**October 24**
Exam 3	**Friday**	**Nov 21**
Final Exam	**Monday**	**Dec 15 12:00 - 2:00 PM**

Summary: Propagation Delay in Multiple-Levels of Logic with Stage Loading



	Equal Rise/Fall	Equal Rise/Fall (with OD)	Minimum Sized	Asymmetric OD (OD_{HL}, OD_{LH})
C_{IN}/C_{REF}				
Inverter	1	OD	1/2	$\frac{OD_{HL}+3 \bullet OD_{LH}}{4}$
NOR	$\frac{3k+1}{4}$	$\frac{3k+1}{4} \bullet OD$	1/2	$\frac{OD_{HL}+3k \bullet OD_{LH}}{4}$
NAND	$\frac{3+k}{4}$	$\frac{3+k}{4} \bullet OD$	1/2	$\frac{k \bullet OD_{HL}+3 \bullet OD_{LH}}{4}$
Overdrive				
Inverter				
HL	1	OD	1	OD_{HL}
LH	1	OD	1/3	OD_{LH}
NOR				
HL	1	OD	1	OD_{HL}
LH	1	OD	1/(3k)	OD_{LH}
NAND				
HL	1	OD	1/k	OD_{HL}
LH	1	OD	1/3	OD_{LH}
t_{PROP}/t_{REF}	$\sum_{k=1}^{n} F_{I(k+1)}$	$\sum_{k=1}^{n} \frac{F_{I(k+1)}}{OD_k}$	$\frac{1}{2}\sum_{k=1}^{n} F_{I(k+1)}\left(\frac{1}{OD_{HLk}}+\frac{1}{OD_{LHk}}\right)$	$\frac{1}{2}\sum_{k=1}^{n} F_{I(k+1)}\left(\frac{1}{OD_{HLk}}+\frac{1}{OD_{LHk}}\right)$

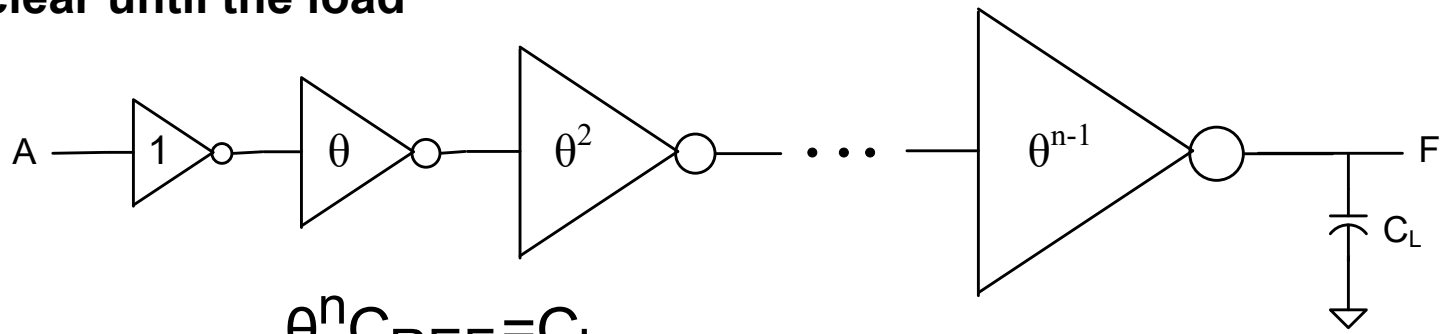
Optimal Driving of Capacitive Loads



Assume first stage is a reference inverter

$$t_{PROP} = t_{REF} \sum_{k=1}^{n} \frac{\theta_k}{\theta_{k-1}}$$

Order reduction strategy : Assume overdrive of stages increases by the same factor clear until the load



$$\theta^n C_{REF} = C_L$$

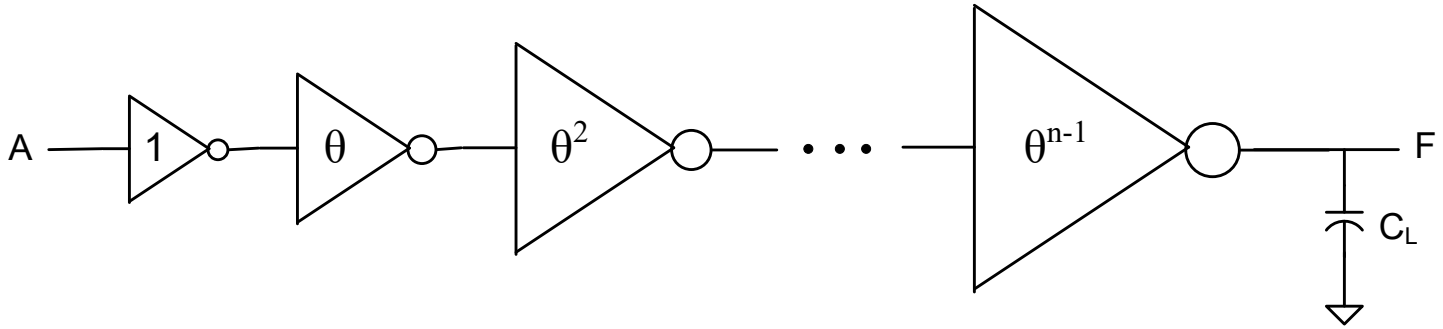
This becomes a 2-parameter optimization (minimization) problem !

Unknown parameters: $\{\theta, n\}$

One degree of freedom

One constraint : $\theta^n C_{REF} = C_L$

Optimal Driving of Capacitive Loads



$$\theta_{OPT} = e$$

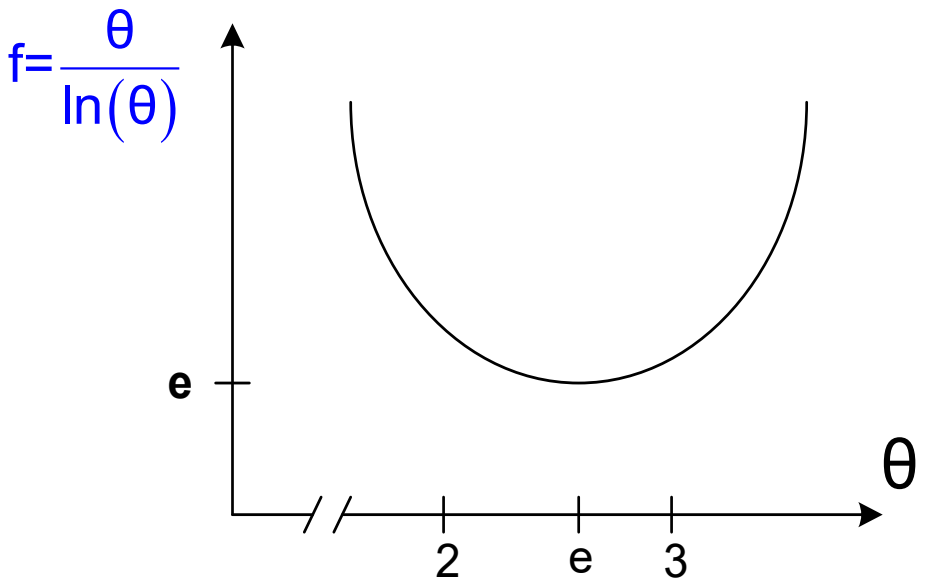
$$\theta^n C_{REF} = C_L$$

$$n_{OPT} = \ln\left(\frac{C_L}{C_{REF}}\right) = \ln(FI_L)$$

$$t_{PROP} = t_{REF}\frac{\theta}{\ln(\theta)}\left[\ln\frac{C_L}{C_{REF}}\right] \qquad t_{PROP} = t_{REF} e\left[\ln\frac{C_L}{C_{REF}}\right] = n\theta t_{REF}$$

Optimal Driving of Capacitive Loads

A practical solution



- **minimum at θ=e but shallow inflection point for 2<θ<3**
- **practically pick θ=2, θ=2.5, or θ=3**
- **since optimization may provide non-integer for n, must pick close integer**

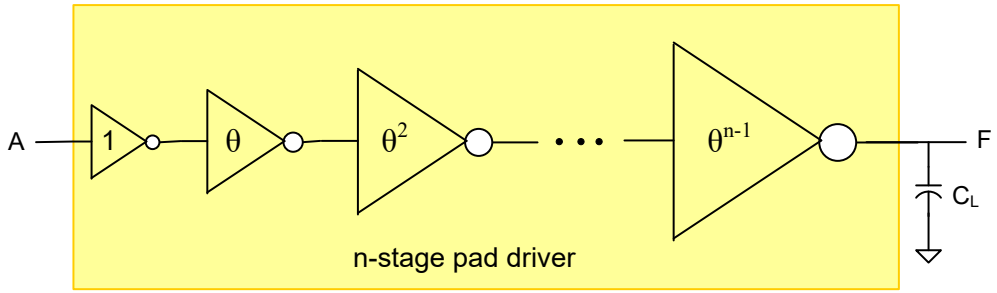
Propagation Delay in "Logic Effort" approach

$$t_{PROP} = t_{REF} \sum_{k=1}^{n} f_k = t_{REF} \sum_{k=1}^{n} g_k h_k = t_{REF} \sum_{k=1}^{n} \frac{F_{I(k+1)}}{OD_k}$$

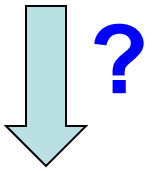
- **Note this expression is identical to what we have derived previously**
  (t_{REF} scaling factor not included in W_H text)

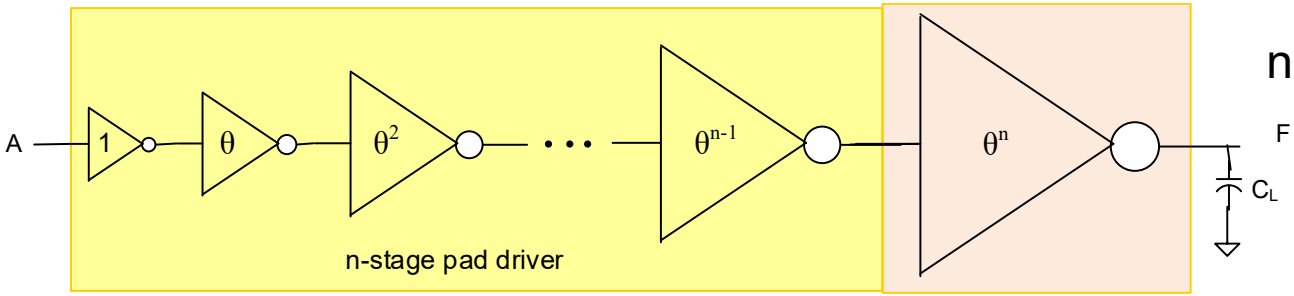
- **Probably more tedious to use the "Logical Effort" approach**
- **Extensions to asymmetric overdrive factors may not be trivial**
- **Extensions to include parasitics may be tedious as well**
- **Logical Effort is widely used throughout the industry**

Will the circuit operate even faster if we increase the number of stages beyond n_{opt}?



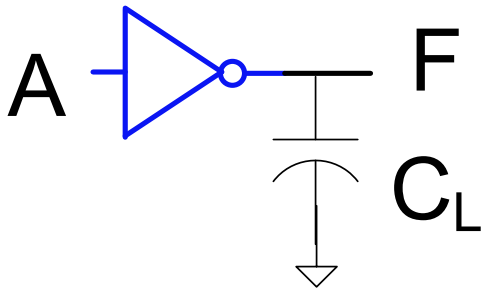
$n=n_{OPT}$





$n=n_{OPT}+1$

Fundamental Limit on Size of Load that Can be Driven at given Clock Rate



Can C_L=10pF be clocked at 100 MHz with a reference inverter?

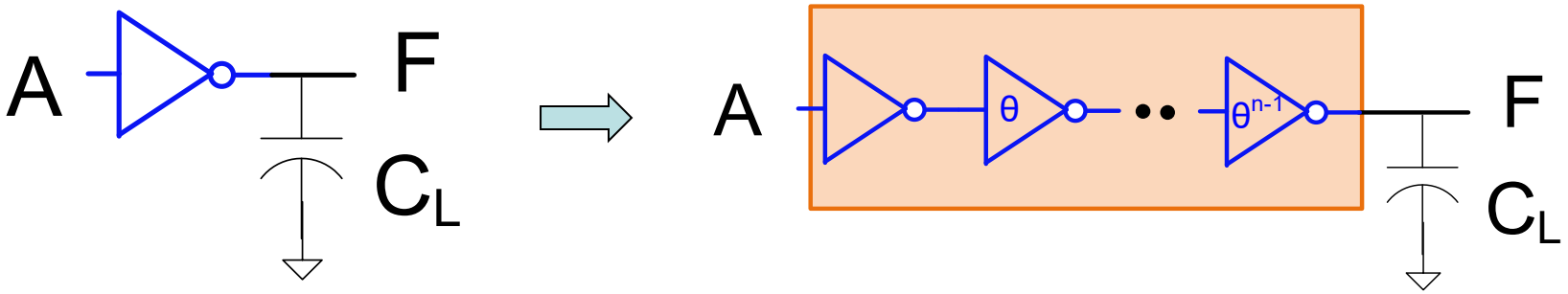
Assume C_{REF}=4fF, t_{REF}=20ps $f_{IN-MAX}=1/t_{PROP}$

$$t_{PROP} = t_{REF} \bullet FI_{LOAD} = 20p\,\text{sec} \bullet \frac{10pF}{4fF} = 20p\,\text{sec} \bullet 2500 = 50n\,\text{sec}$$

$$f_{IN-MAX} = \frac{1}{50n\,\text{sec}} = 20MHz$$

No ! f_{IN-MAX}<100MHz

Fundamental Limit on Size of Load that Can be Driven at given Clock Rate



Can C_L=10pF be clocked at 100 MHz if a pad driver (sized for equal rise/fall) is used?

Assume C_{REF}=4fF, t_{REF}=20ps $f_{IN\text{-}MAX}$=1/t_{PROP} $FI_{LOAD} = \frac{10pF}{4fF} = 2500$

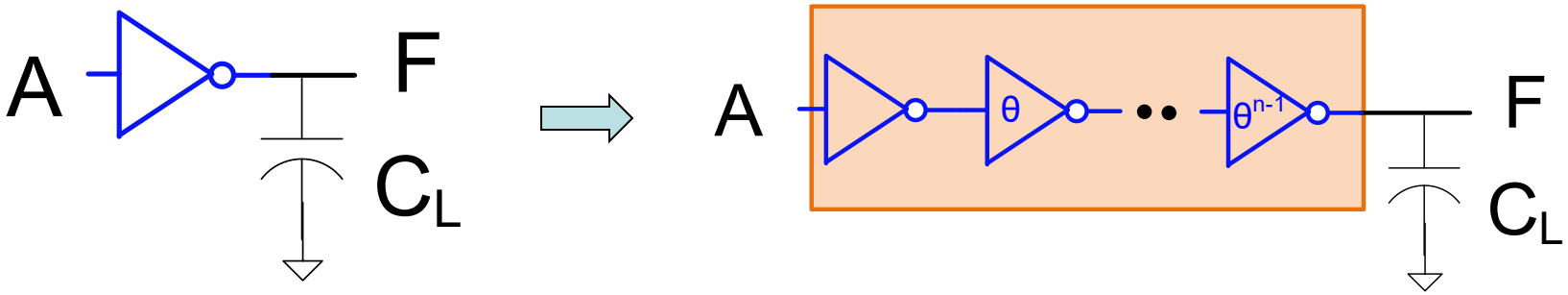
$$n_{OPT} = \ln\left(\frac{C_L}{C_{REF}}\right) = \ln(FI_L) = 7.8 \simeq 8$$

$$t_{PROP} = n\theta t_{REF} = 8 \bullet e \bullet t_{REF} = 434 \text{ psec}$$

$$f_{IN\text{-}MAX} = \frac{1}{n\theta t_{REF}} = \frac{1}{434 \text{ psec}} = 2.30 \text{ GHz}$$

Yes !

Fundamental Limit on Size of Load that Can be Driven at given Clock Rate



Can C_L=100pF be clocked at 500 MHz if a pad driver is used?

Assume C_{REF}=4fF, t_{REF}=20ps $f_{IN\text{-}MAX}$=1/t_{PROP} $\quad FI_{LOAD} = \frac{100pF}{4fF} = 25{,}000$

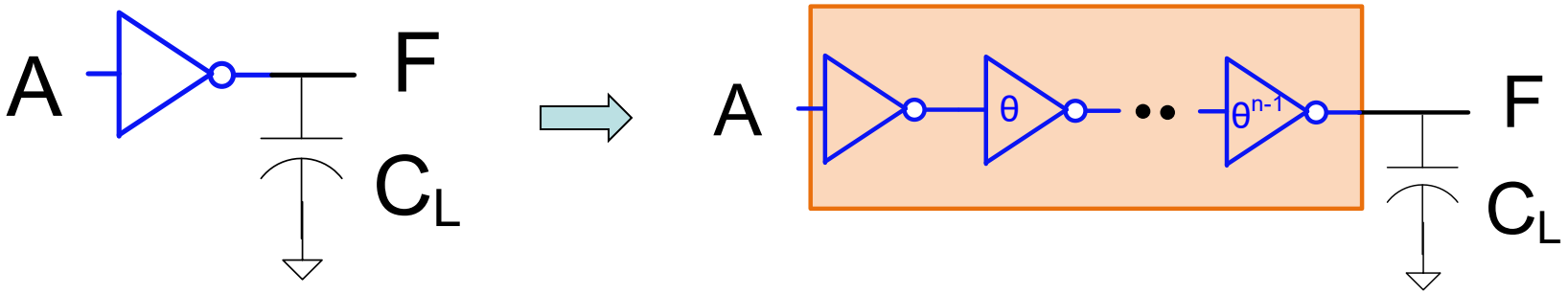
$$n_{OPT} = \ln\left(\frac{C_L}{C_{REF}}\right) = \ln(FI_L) = 10.1 \approx 10$$

$$t_{PROP} = n\theta t_{REF} = 10 \bullet e \bullet t_{REF} = 542 \text{ psec}$$

$$f_{IN\text{-}MAX} = \frac{1}{n\theta t_{REF}} = \frac{1}{542 \text{ psec}} = 1.85 \text{ GHz}$$

Yes !

Fundamental Limit on Size of Load that Can be Driven at given Clock Rate



Can C_L=500pF be clocked at 2GHz if a pad driver is used?

Assume C_{REF}=4fF, t_{REF}=20ps $f_{IN\text{-}MAX}$=1/t_{PROP} $FI_{LOAD} = \frac{500pF}{4fF} = 125,000$

$$n_{OPT} = \ln\left(\frac{C_L}{C_{REF}}\right) = \ln(FI_L) = 11.7 \approx 12$$

$$t_{PROP} = n\theta t_{REF} = 12 \bullet e \bullet t_{REF} = 652 \text{ psec}$$

$$f_{IN\text{-}MAX} = \frac{1}{n\theta t_{REF}} = \frac{1}{652 \text{ psec}} = 1.54 \text{ GHz}$$

No !

Digital Circuit Design

- Hierarchical Design
- Basic Logic Gates
- Properties of Logic Families
- Characterization of CMOS Inverter
- Static CMOS Logic Gates
  - Ratio Logic
- Propagation Delay
  - Simple analytical models
    - FI/OD
    - Logical Effort
  - Elmore Delay
- Sizing of Gates
  - The Reference Inverter

- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
- Other Logic Styles
- Array Logic
- Ring Oscillators

---

done

partial

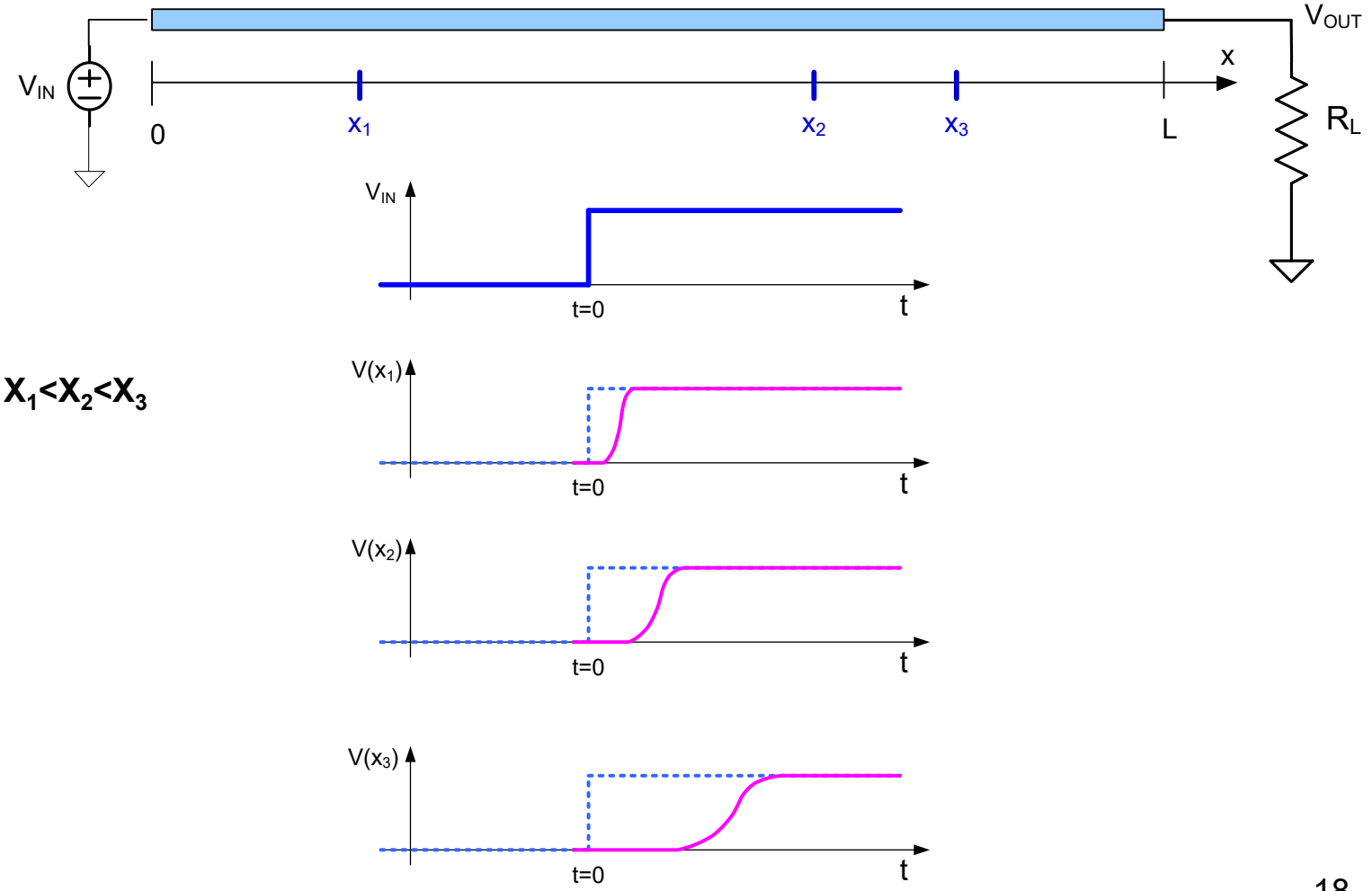
Elmore Delay Calculations

- **Interconnects have a distributed resistance and a distributed capacitance**
  - **Often modeled as resistance/unit length and capacitance per unit length**
- **These delay the propagation of the signal**
- **Effectively a transmission line**
  - **analysis is really complicated**
- **Can have much more complicated geometries**

Elmore Delay Calculations

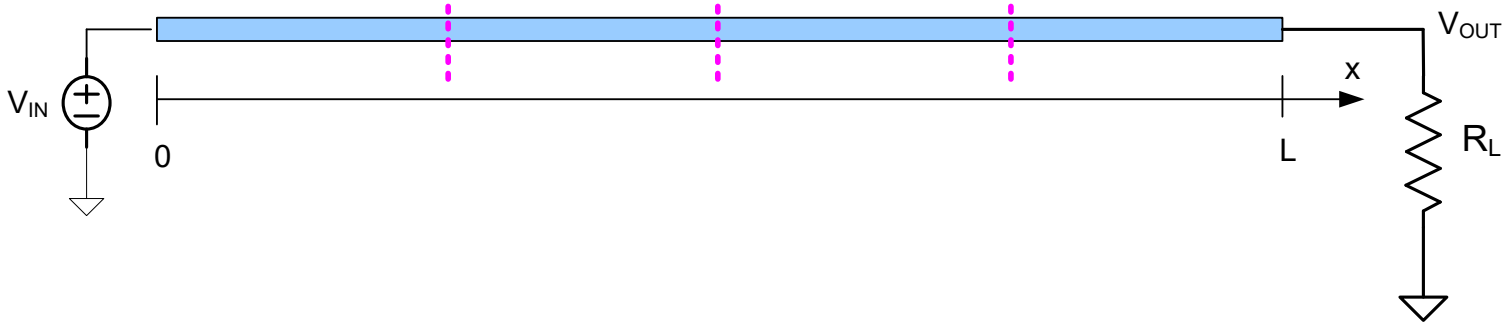
Can have much more complicated geometries

Elmore Delay Calculations

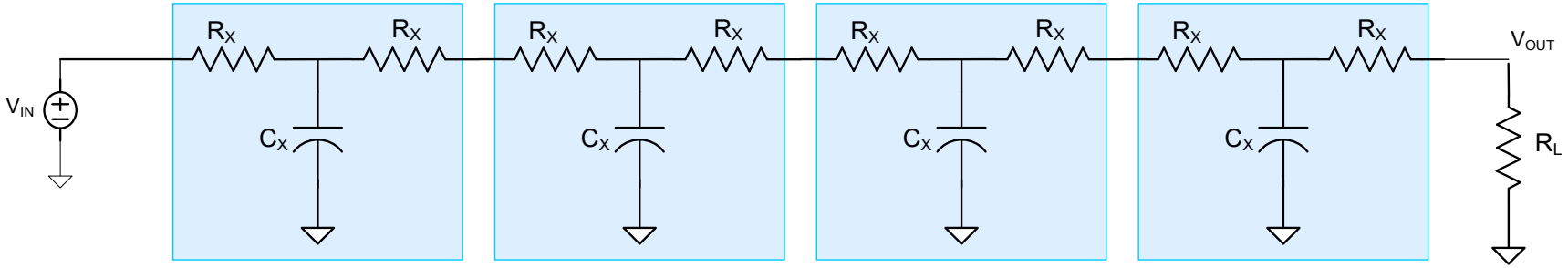


For $X_1<X_2<X_3$

Elmore Delay Calculations



A lumped element model of transmission line (with "T" elements)



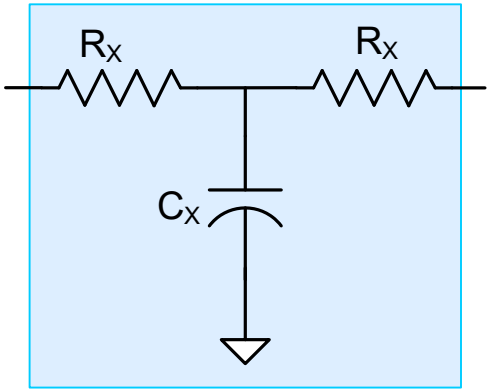
Even this lumped model is 4-th order and a closed-form solution is very tedious !

Can use "L" or other lumped segments as well (with small number some perform better than others)

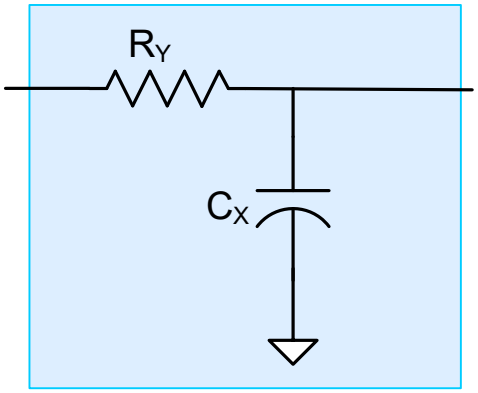
Need a quick (and reasonably good) approximation to the delay of a delay line !!

Did anyone actually analyze a circuit like this in EE 201?

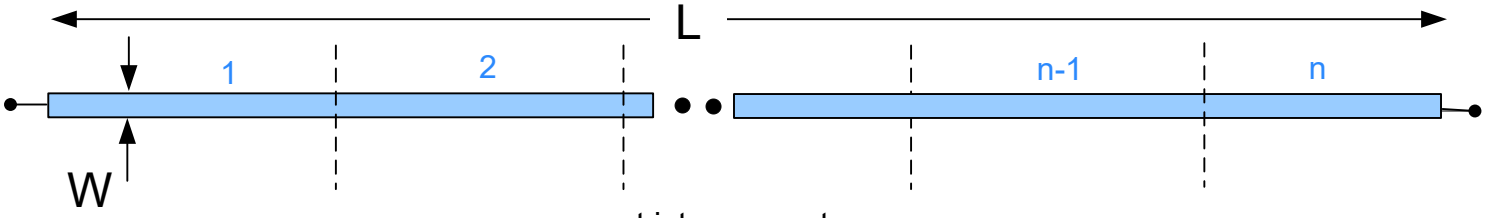
Elmore Delay Calculations



T-Model



L-Model



n-segment interconnect

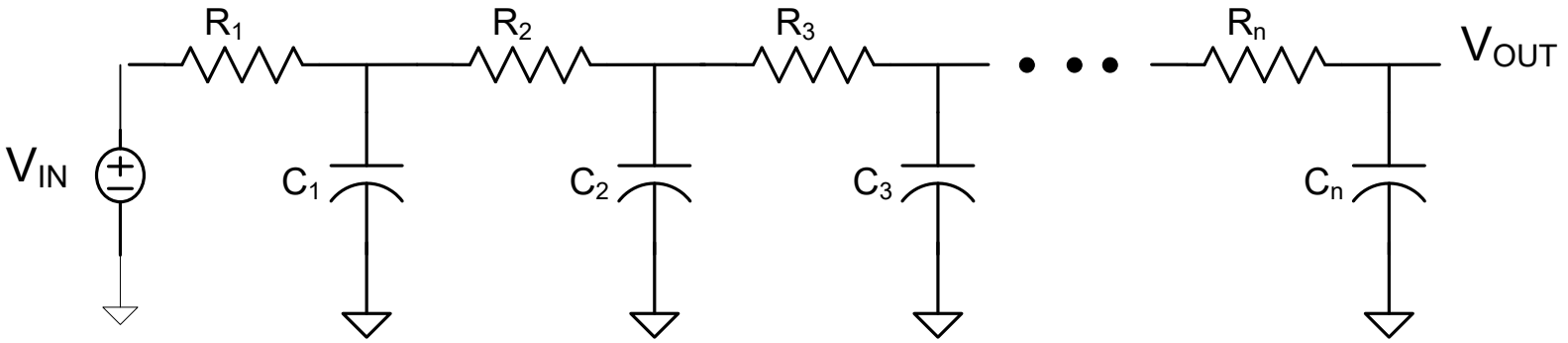
$$R_X = \frac{R_\square \frac{L}{W}}{2n}$$

$$C_X = \frac{C_{Density} WL}{n}$$

$$R_Y = \frac{R_\square \frac{L}{W}}{n}$$

$$C_X = \frac{C_{Density} WL}{n}$$

Elmore Delay Calculations



Elmore delay: $$t_{ED} = \sum_{i=1}^{n} \left(C_i \sum_{j=1}^{i} R_j \right)$$

- **It can be shown that this is a reasonably good approximation to the actual delay**
  - **provided sufficient number of stages are used**
  - **number does not need to be very large**
- **Numbering is critical (resistors and capacitors numbered from input to output)**
- **As stated, only applies to this specific structure**
- **HL and LH Elmore Delays are the same**
- **Since $t_{EHL}=t_{ELH}$, $t_{PROP} = 2\ t_{ED}$**

Elmore Delay Calculations

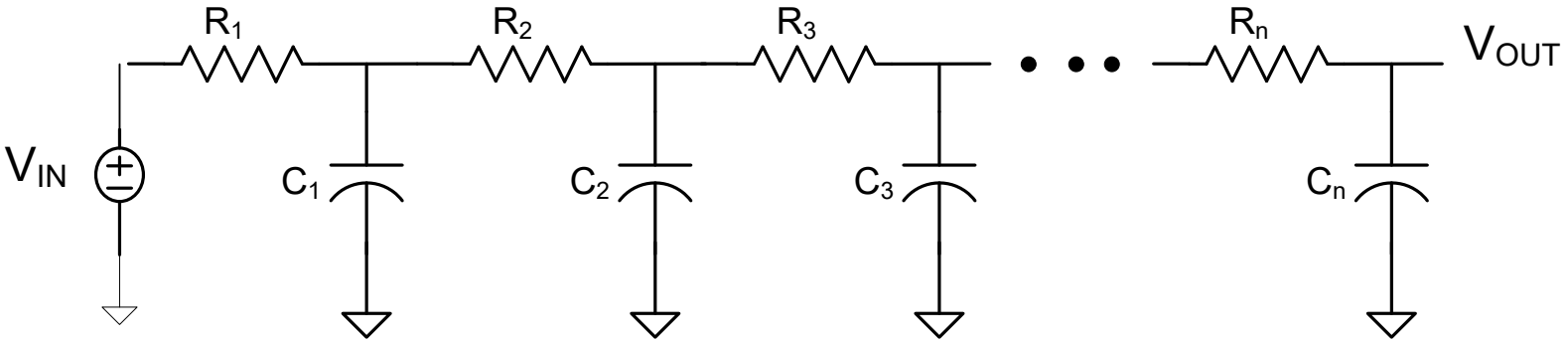
Elmore delay: $$t_{PD} = \sum_{i=1}^{n} \left(C_i \sum_{j=1}^{i} R_j \right)$$

- **Note error in text on Page 161 of first edition of WH**

$$t_{pd} = \sum_{i} R_{n-i} C_i = \sum_{i=1}^{N} C_i \sum_{j=i}^{i} R_j$$

- **Not detailed definition on Page 150 of second edition of WH**

Elmore Delay Calculations



$$t_{PD} = \sum_{i=1}^{n} \left(C_i \sum_{j=1}^{i} R_j \right)$$

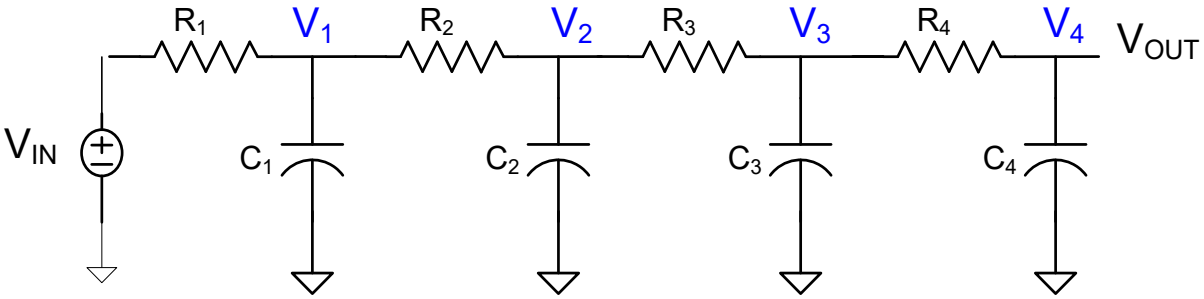
From Wikipedia (Dec 8 2021):

Elmore delay[1] is a simple approximation to the delay through an RC network in an electronic system. It is often used in applications such as logic synthesis, delay calculation, static timing analysis, placement and routing, since it is simple to compute (especially in tree structured networks, which are the vast majority of signal nets within ICs) and is reasonably accurate. Even where it is not accurate, it is usually faithful, in the sense that reducing the Elmore delay will almost always reduce the true delay, so it is still useful in optimization.

[1] W.C. Elmore. *The Transient Analysis of Damped Linear Networks with Particular Regard to Wideband Amplifiers*. J. Applied Physics, vol. 19(1), 1948.

Elmore Delay Calculations

Example:



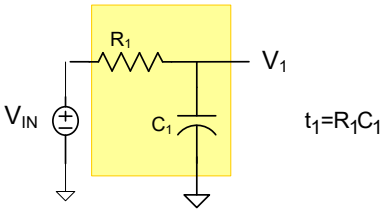
Elmore delay:

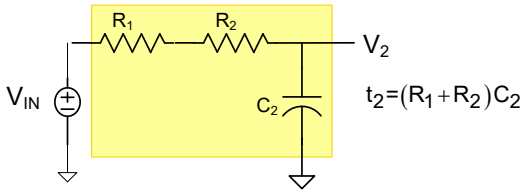
$$t_{ED} = \sum_{i=1}^{4} \left(C_i \sum_{j=1}^{i} R_j \right)$$

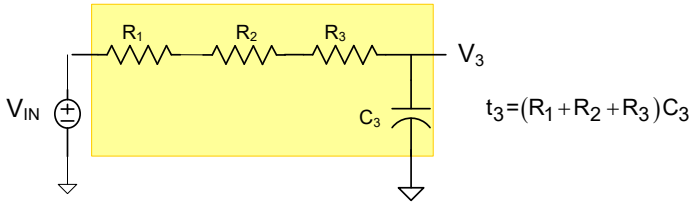
$$t_{ED} = \sum_{i=1}^{4} (t_i)$$

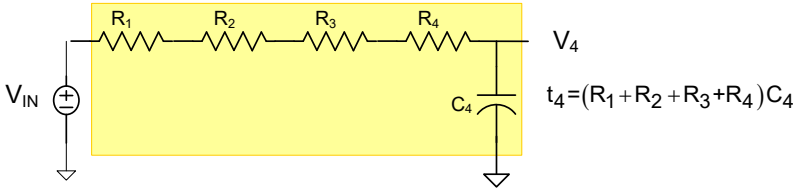
where

$$t_i = C_i \sum_{j=1}^{i} R_j \qquad j = 1,2,3,4$$







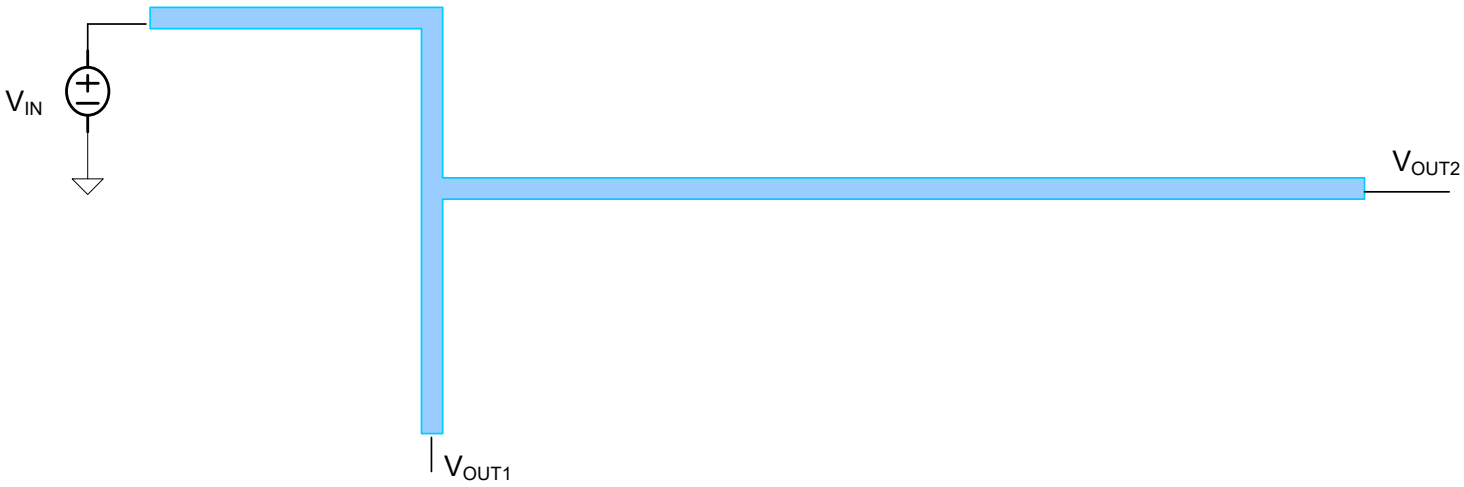


What is really happening?

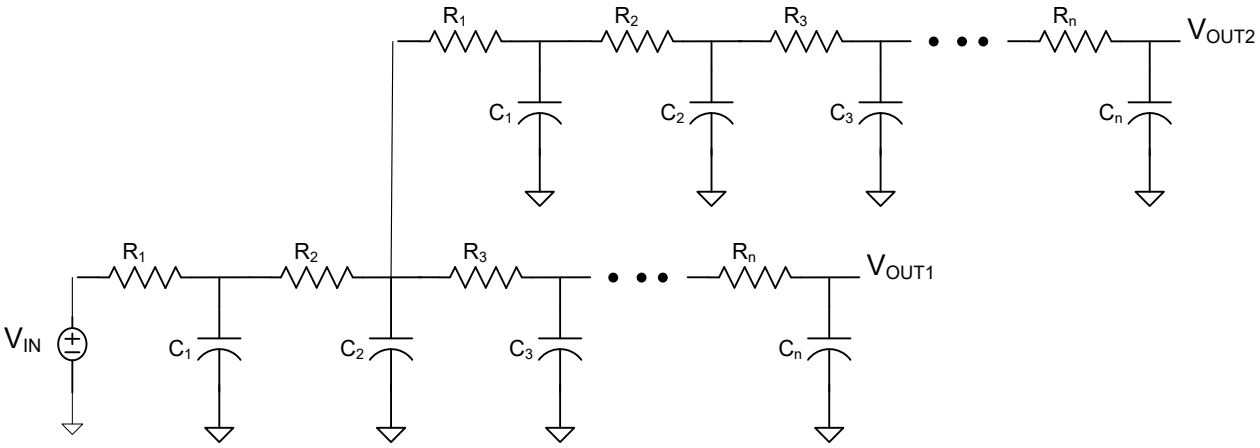
- **Creating 4 first-order circuits**
- **Delay to V_1, V_2, V_3 and V_4 calculated separately by considering capacitors one at a time and assuming others are 0**

Elmore Delay Calculations

Extensions:



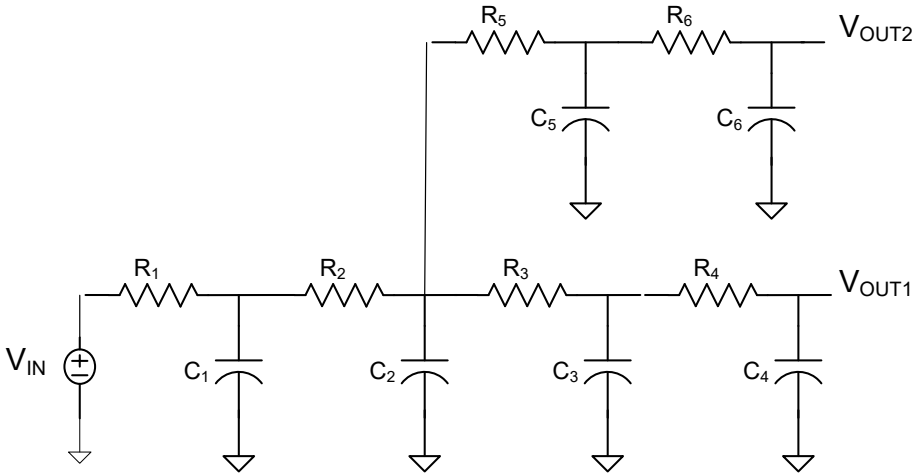
Lumped Network Model:



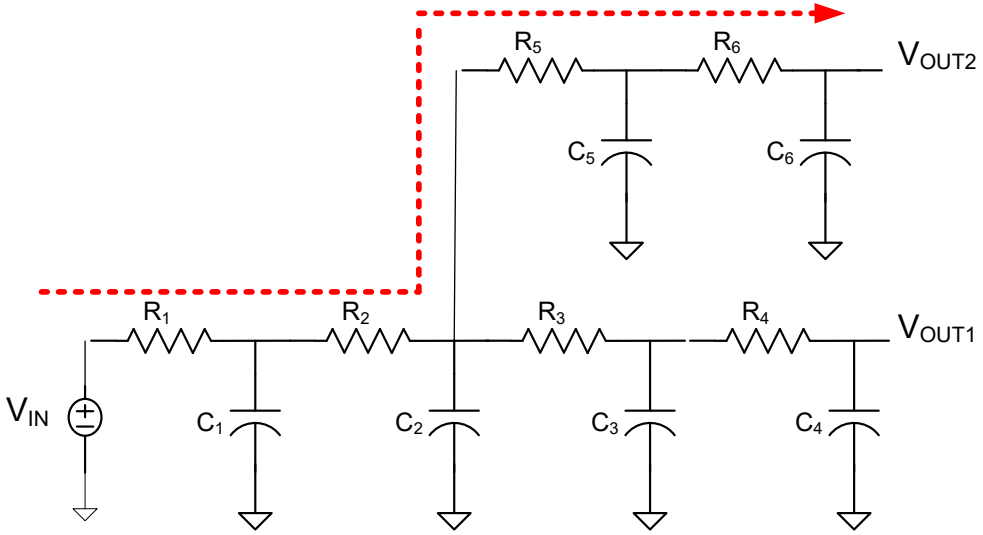
Elmore Delay Calculations

Extensions:

1. **Create a lumped element model**



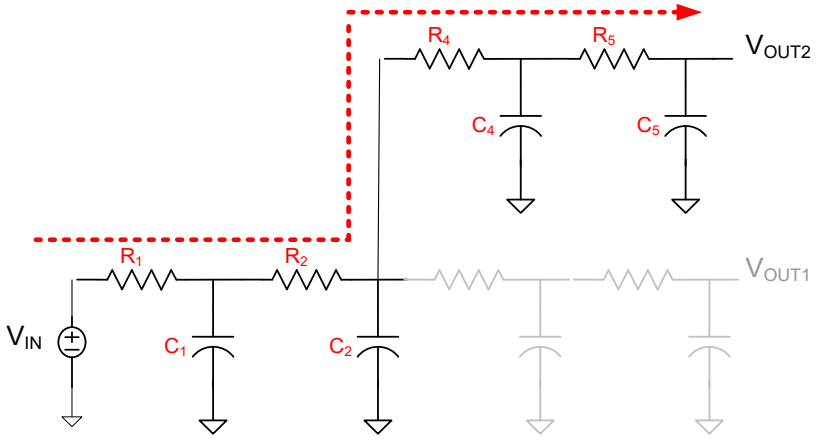
2. **Identify the path from input to output**



Elmore Delay Calculations

Extensions:

3. Renumber elements along path from input to output and neglect off-path elem



4. Use Elmore Delay equation for elements on this RC network

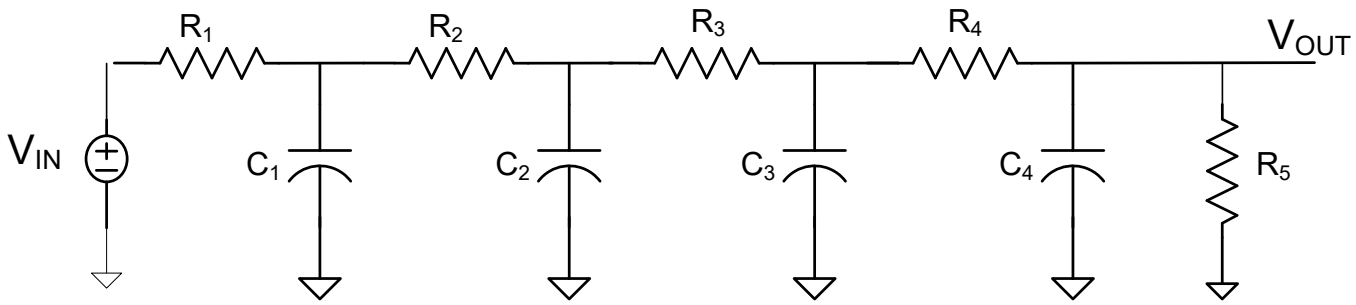
$$t_{ED} = \sum_{i=1}^{4} \left(C_i \sum_{j=1}^{i} R_j \right)$$

Elmore Delay Calculations

How is a resistive load handled?

Elmore Delay Calculations

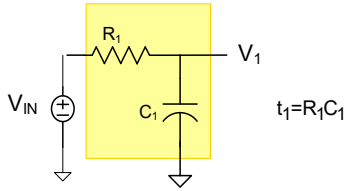
Example with resistive load:

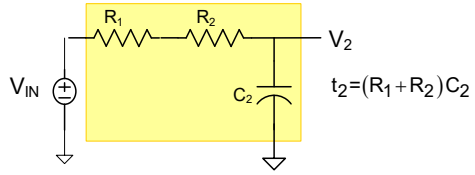


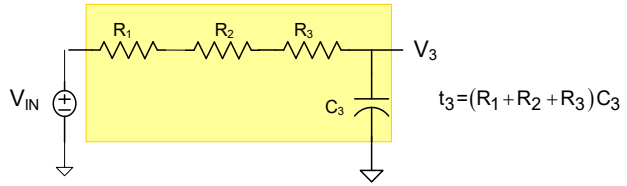
Elmore delay:

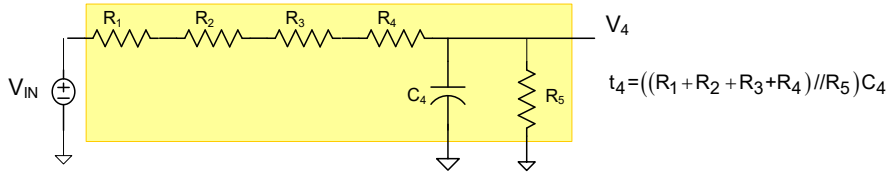
$$t_{ED} = \sum_{i=1}^{4} \left(C_i \sum_{j=1}^{i} R_j \right)$$

where









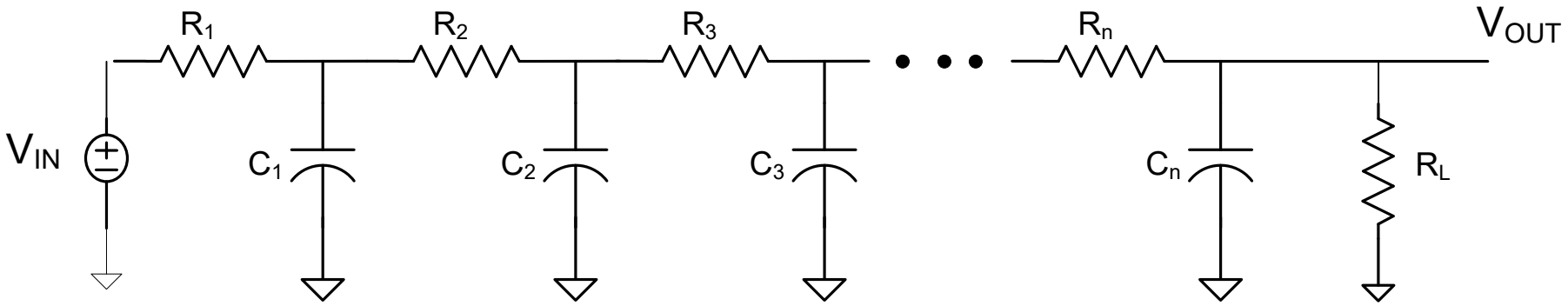
$$t_{ED} = \sum_{i=1}^{4} (t_i)$$

$$t_i = C_i \sum_{j=1}^{i} R_j \qquad j = 1,2,3$$

$$t_4 = C_4 \left(\left[\sum_{j=1}^{4} R_j \right] // R_5 \right)$$

Elmore Delay Calculations

With resistive load:

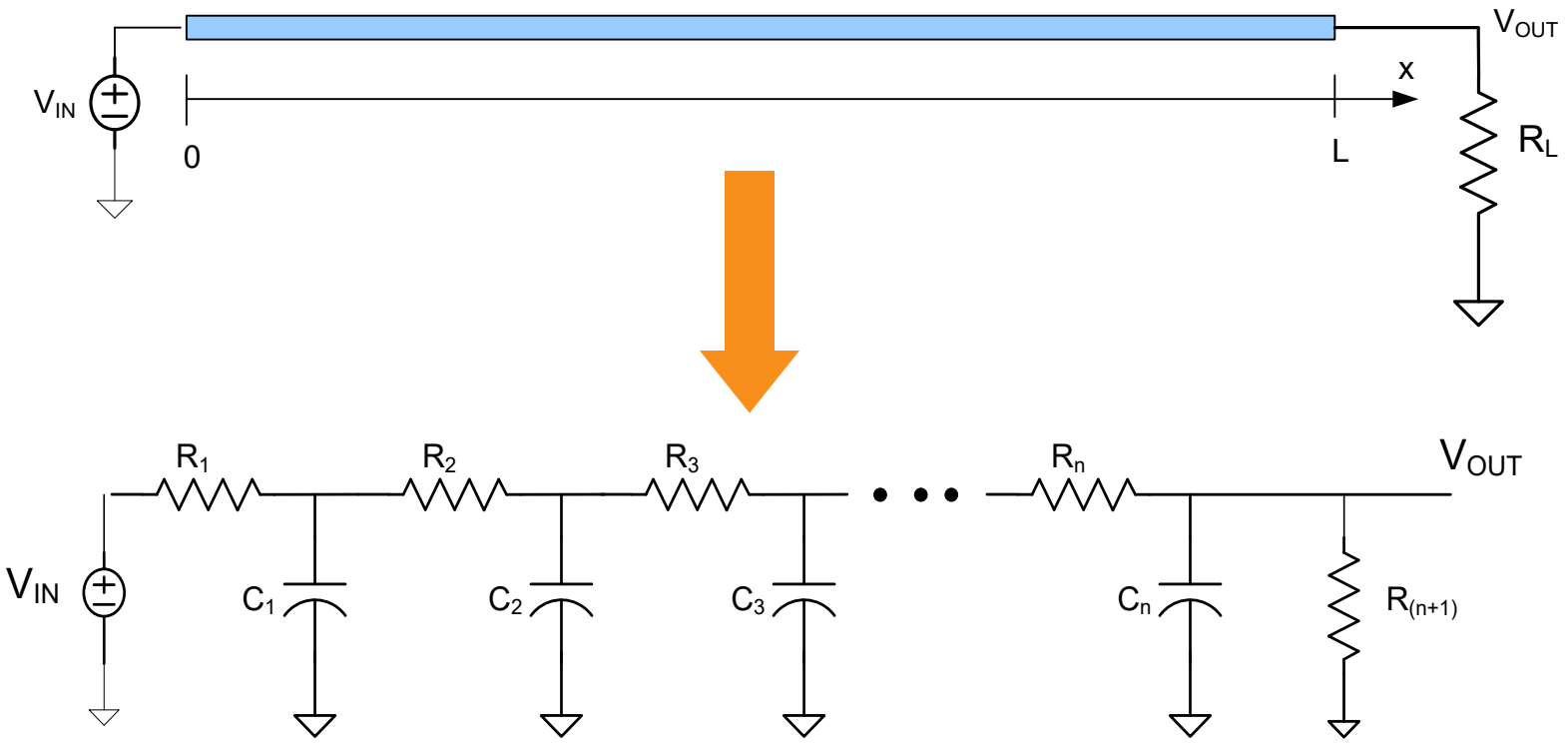


Simple Elmore delay:

$$t_{ED} = \sum_{i=1}^{n-1} \left(C_i \sum_{j=1}^{i} R_j \right) + C_n \left(\left(\sum_{j=1}^{n} R_j \right) / / R_L \right)$$

Actually, R_L affects all of the delays and a modestly better but modestly more complicated delay model is often used

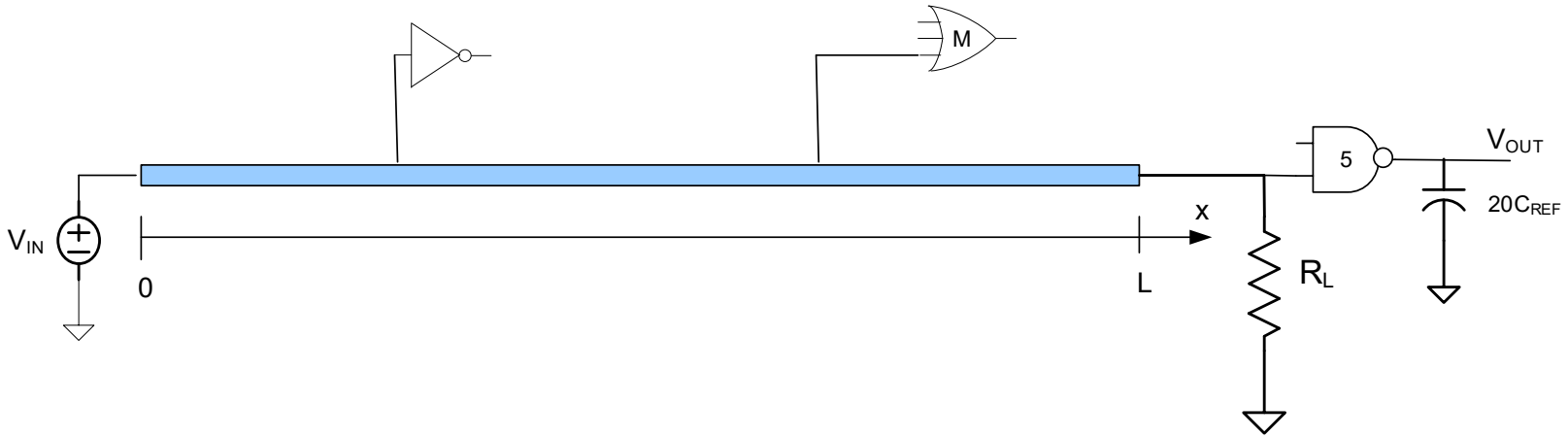
Elmore Delay Calculations



How are the number of stages chosen?

- **For hand analysis, keep number of stages small (maybe 3 or 4 for simple delay line) if possible)**
- **If "faithfulness" is important, should keep the number of stages per unit length constant**

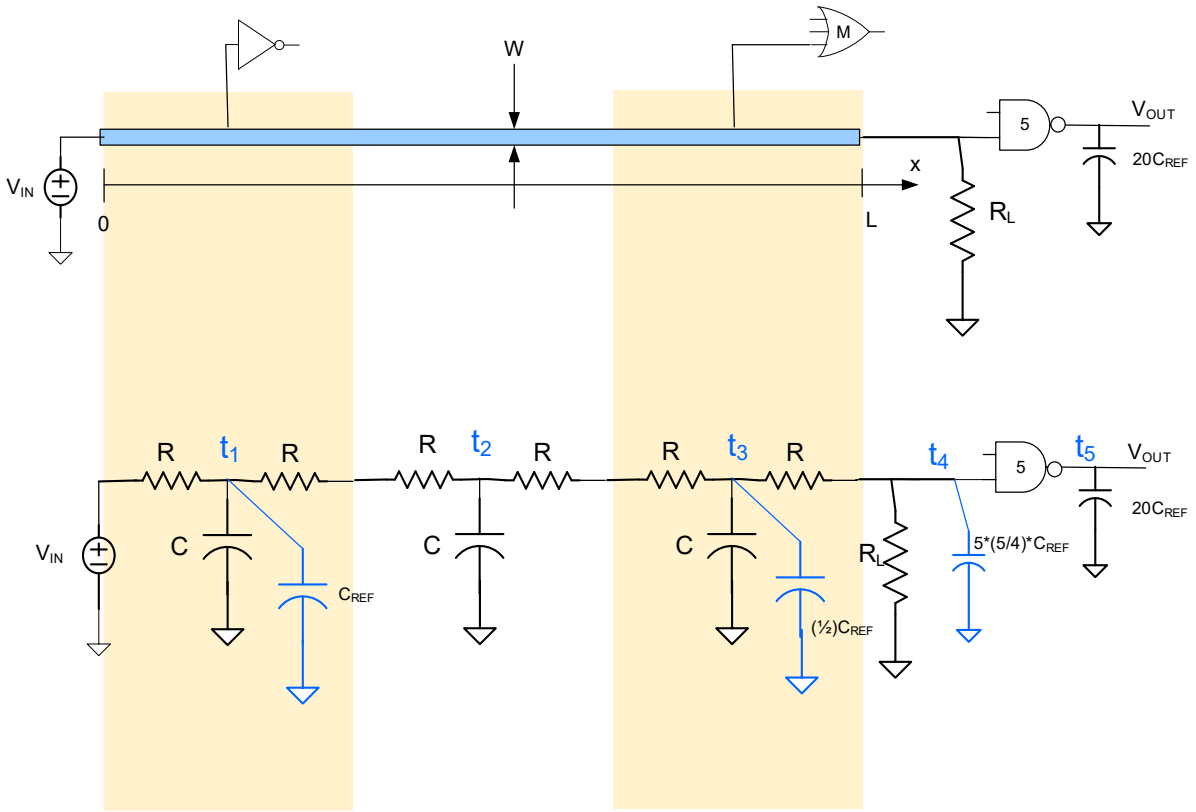
Elmore Delay Calculations



?

Elmore Delay Calculations

Determine propagation delay



$$R = \frac{1}{6} R_{\square} \frac{L}{W}$$

$$C = \frac{1}{3} C_D$$

$$t_1 = R\left(C + C_{REF}\right) \qquad t_2 = 3RC \qquad t_3 = 5R\left(C + \frac{1}{2}C_{REF}\right) \qquad t_4 = \left[6R \,/\!/\!/\, R_L\right]\frac{25}{4}C_{REF} \qquad t_{PROP5} = t_{REF}\frac{20}{5}$$

$$t_{PROP} = 2\sum_{i=1}^{4} t_i + t_{PROP5}$$

Digital Circuit Design

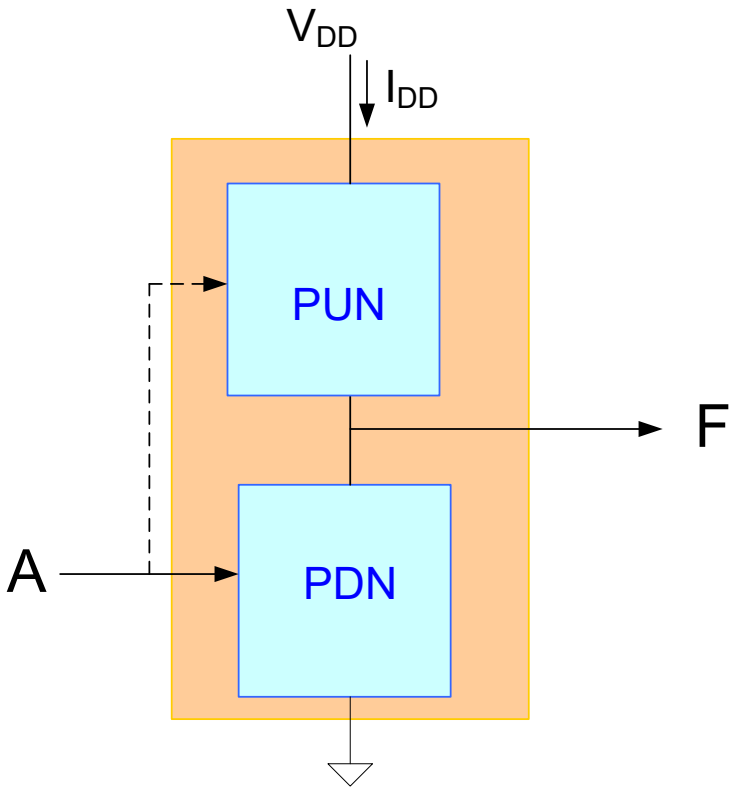
- Hierarchical Design
- Basic Logic Gates
- Properties of Logic Families
- Characterization of CMOS Inverter
- Static CMOS Logic Gates
  - Ratio Logic
- Propagation Delay
  - Simple analytical models
    - FI/OD
    - Logical Effort
  - Elmore Delay
- Sizing of Gates
  - The Reference Inverter
- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
- Other Logic Styles
- Array Logic
- Ring Oscillators

---

done

partial

Power Dissipation in Logic Circuits



Assume current periodic with period T_{CL}

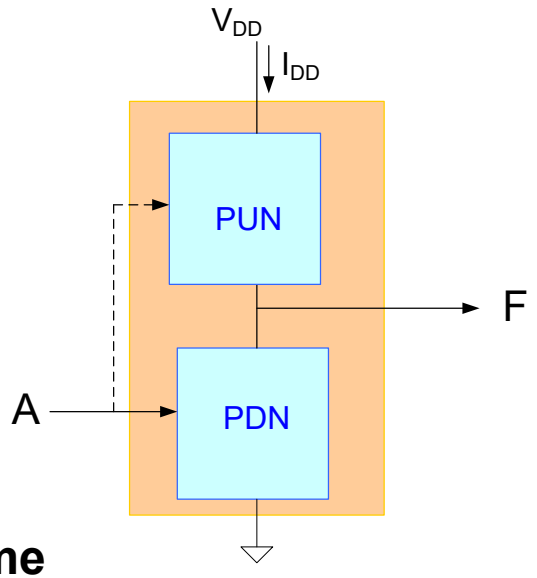
$$P_{AVG,T} = \frac{1}{T_{CL}} \int_{t_1}^{t_1+T_{CL}} V_{DD} I_{DD}(t)\,dt$$

Power Dissipation in Logic Circuits

Types of Power Dissipation

- **Static**
- **Pipe**
- **Dynamic**
- **Leakage**
  - **Gate**
  - **Diffusion**
  - **Drain**

Static Power Dissipation



If Boolean output averages H and L 50% of the time

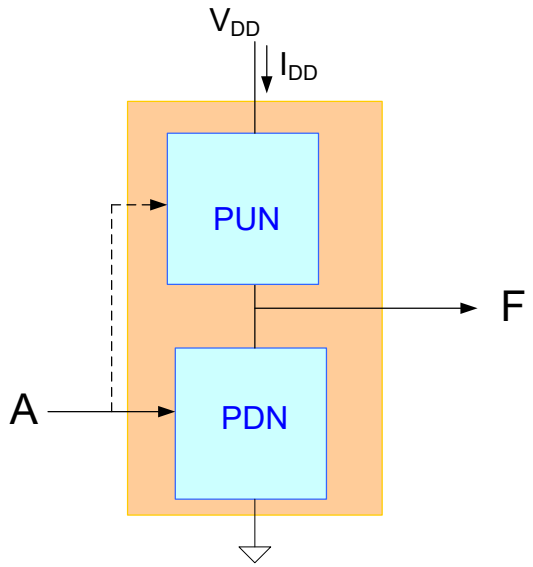
$$P_{STAT,AVG} = \frac{P_H + P_L}{2}$$

$$P_{STAT,AVG} = \frac{V_{DD}(I_{DDH} + I_{DDL})}{2}$$

- **Generally decreases with V_{DD}**
- **$I_{DDH}=I_{DDL}=0$ for static CMOS gates so $P_{STAT}=0$**

- **A major source of power dissipation in ratio logic circuits and the major reason CMOS is so widely used**

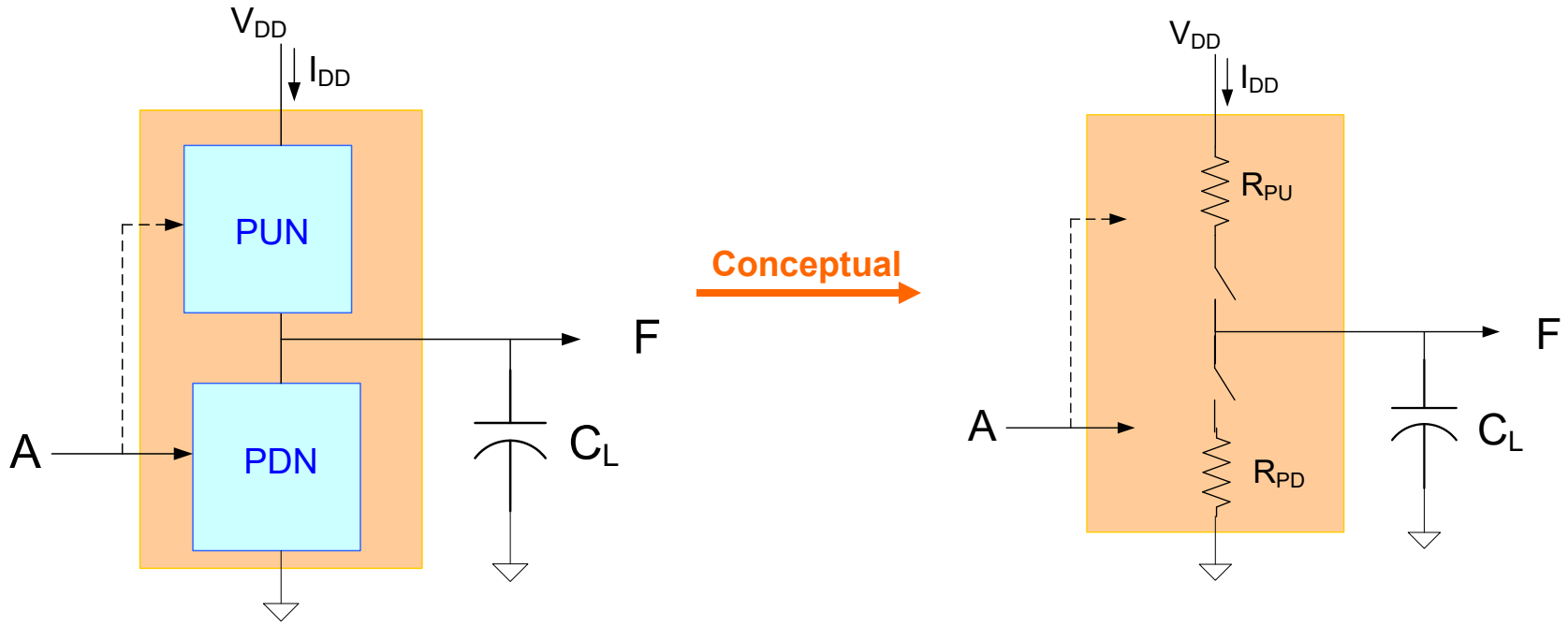
Pipe Power Dissipation



Due to conduction of both PUN and PDN during transitions

- **Can be made small if transitions are fast**
- **Usually negligible in Static CMOS circuits**

Dynamic Power Dissipation



Due to charging and discharging C_L on logic transitions

C_L dissipates no power but PUN and PDN dissipate power during charge and discharge of C_L

C_L includes all gate input capacitances of loads and interconnect capacitances

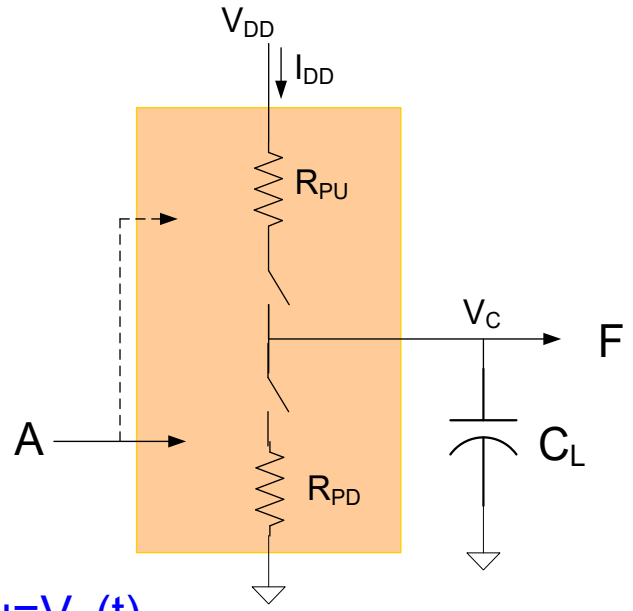
Dynamic Power Dissipation

Energy supplied by V_{DD} when C_L charges

Assume a HL input transition starts at t = t_1

$$E = \int_{t_1}^{\infty} V_{DD} I_{DD}(t)\,dt$$

$$I_{DD} = C_L \frac{dV_C}{dt}$$



$$E = \int_{t_1}^{\infty} V_{DD} C_L \frac{dV_C}{dt}\,dt$$

change variable $u = V_C(t)$

$$E = \int_{V_C=0}^{V_{DD}} V_{DD} C_L\,dV_C = V_{DD} C_L \int_{V_C=0}^{V_{DD}} dV_C = V_{DD} C_L\, V_C\Big|_{V_C=0}^{V_{DD}} = V_{DD}^2 C_L$$

Energy stored in C_L after C_L is charged to V_{DD}:

$$E = \frac{1}{2} C_L V_{DD}^2$$

Dynamic Power Dissipation

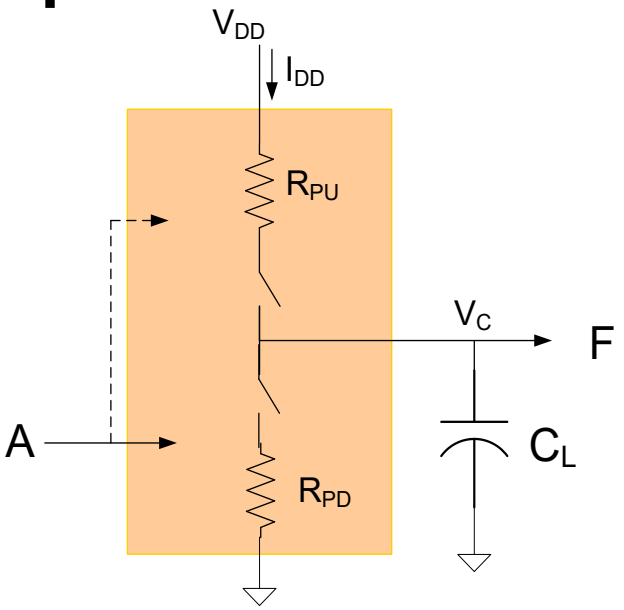
Energy supplied by V_{DD} and dissipated in R_{PU} when C_L charges

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

Energy stored on C_L after L-H transition

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

$$E = E_{DIS} + E_{STORE} = C_L V_{DD}^2$$



When the output transitions from H to L, energy stored on C_L is dissipated in PDN

Thus, energy from V_{DD} for one L-H: H-L output transition sequence is

$$E = C_L V_{DD}^2$$

If f is the average transition rate of the output, determine P_{AVG}

Dynamic Power Dissipation

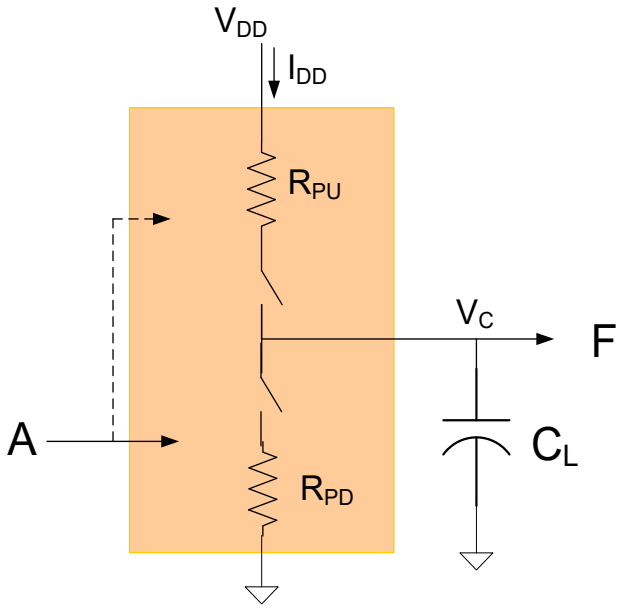
Energy from V_{DD} for one L-H: H-L output transition sequence is

$$E = C_L V_{DD}^2$$

If f is the average transition rate of the output, determine P_{AVG}

$$P_{AVG} = \frac{E}{T} = Ef$$

$$P_{DYN} = fC_L V_{DD}^2$$



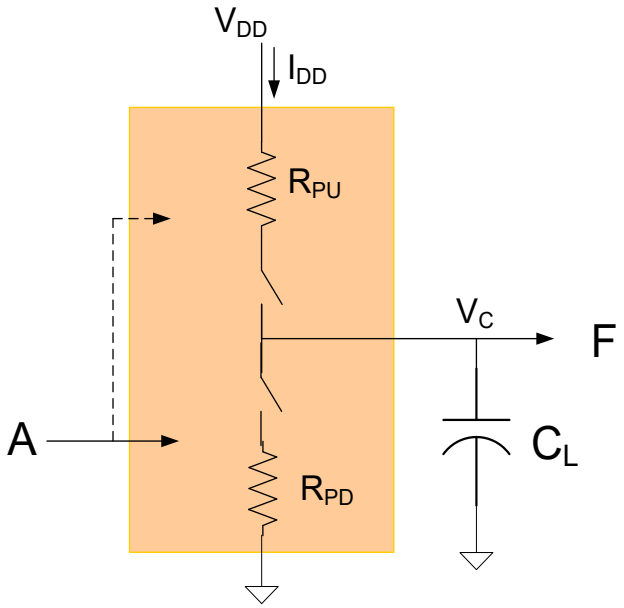
If a gate has a transition duty cycle of 50% with a clock frequency of f_{CL}

$$P_{DYN} = \frac{f_{CL}}{2} C_L V_{DD}^2$$

Note dependent on the square of V_{DD} ! …. Want to make V_{DD} small !!!

Major source of power dissipation in many static CMOS circuits for L_{min}>0.1u

Dynamic Power Dissipation



Energy dissipated with clock signal itself

$$P_{DYN} = f_{CL} C_L V_{DD}^2$$

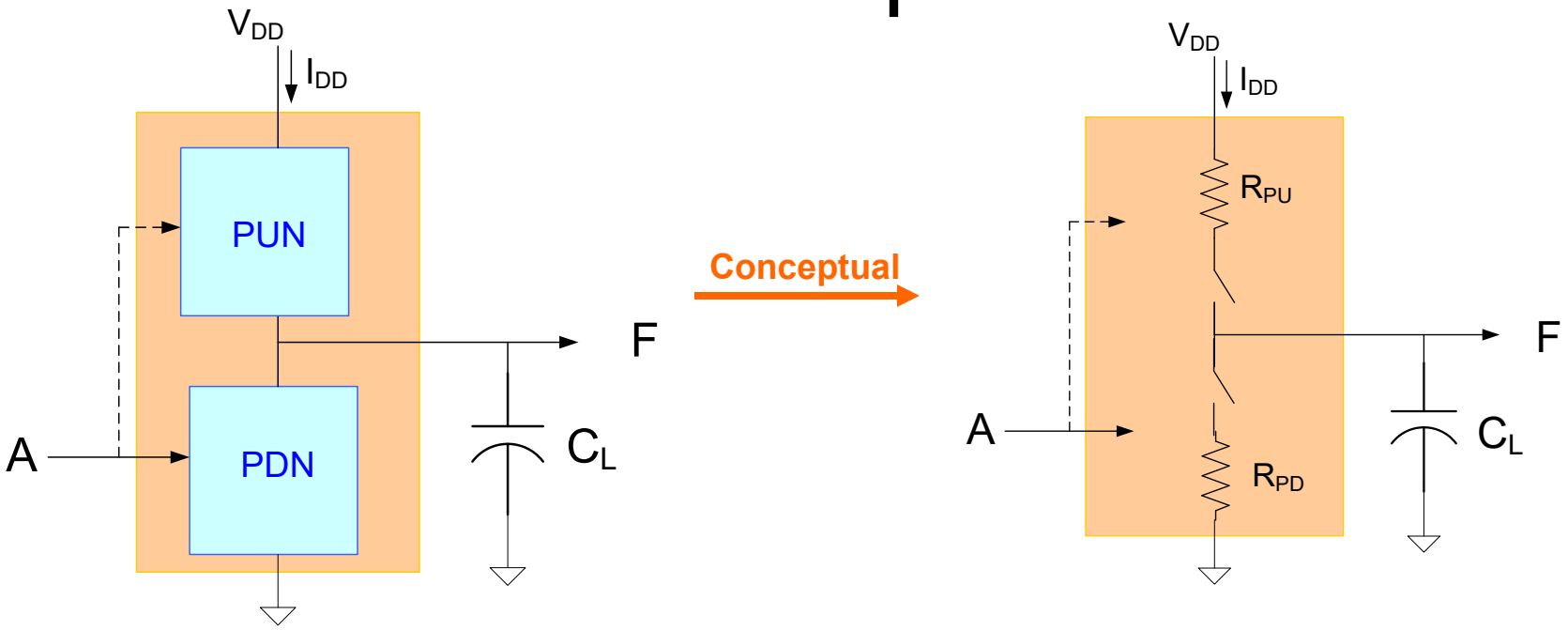
The clock transitions on every clock cycle (i.e. it has a transition duty cycle of 100%)

Clock distribution can cause significant power dissipation

But if a gate has a transition duty cycle of 50% with a clock frequency of f_{CL}

$$P_{DYN} = \frac{f_{CL}}{2} C_L V_{DD}^2$$

Power Dissipation

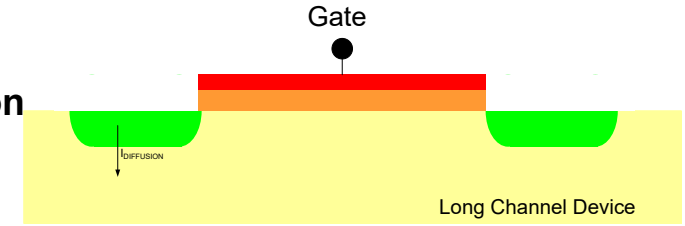


- **All power is dissipated in pull-up and pull-down devices**
- **C_L dissipates no power but PUN and PDN dissipate power when charging and discharging C_L**
- **Dynamic power dissipation reduced by more (often much more) than a factor of 2 if minimum sizing strategy is used**
- **NAND logic more attractive than NOR logic when multiple inputs required**

Leakage Power Dissipation

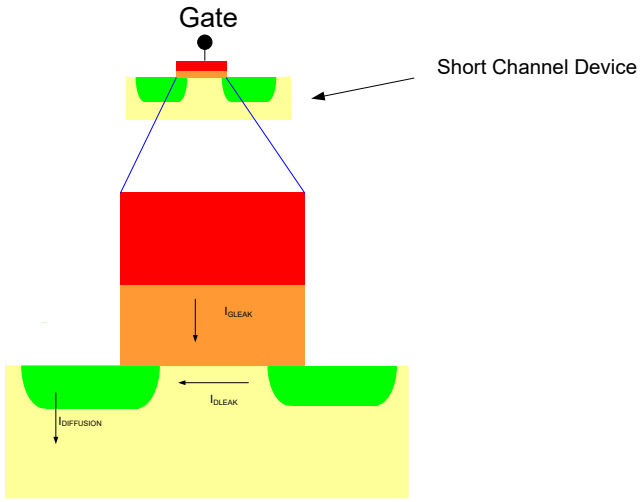
- Gate

- **with very thin gate oxides, some gate leakage current flows**
- **major concern in 60nm and smaller processes**
- **actually a type of static power dissipation**



-Diffusion

- **Leakage across a reverse-biased pn junction**
- **Dependent upon total diffusion area**
- **May actually be dominant power loss on longer-channel devices**
- **Actually a type of static power dissipation**

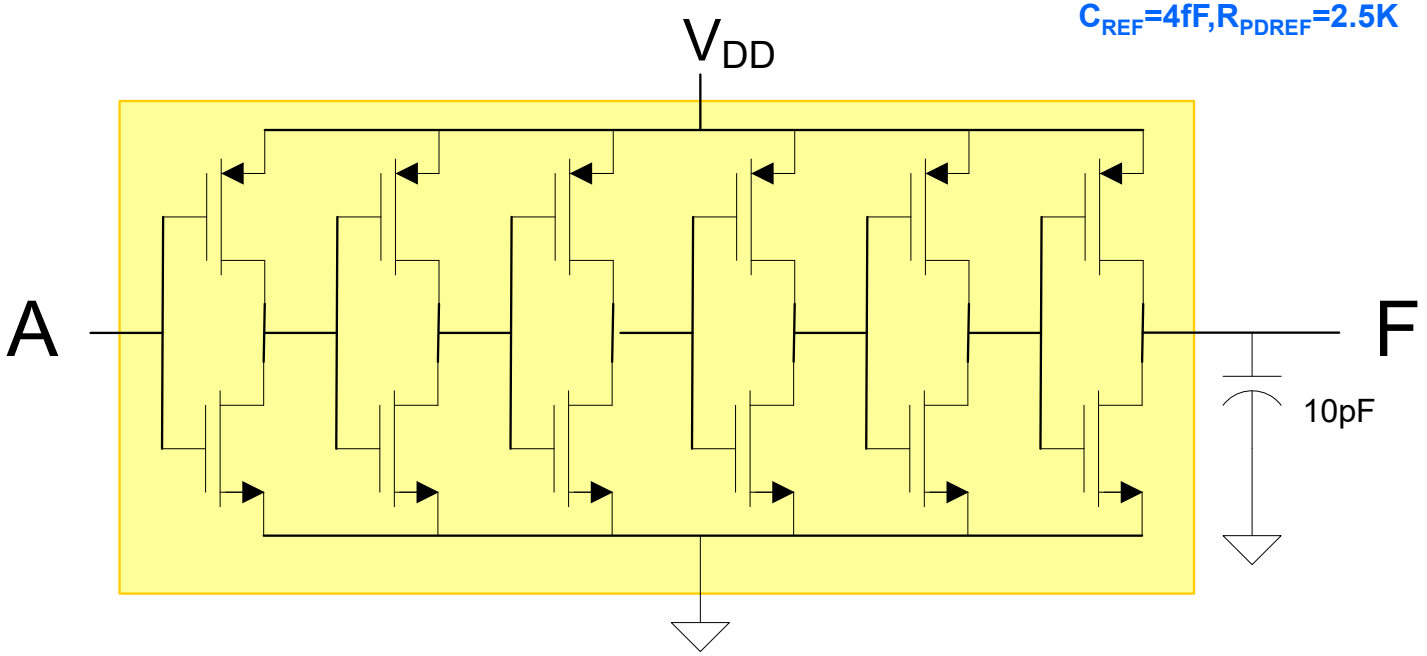


-Drain

- **channel current due to small V_{GS}-V_T**
- **of significant concern only with low V_{DD} processes**
- **actually a type of static power dissipation**

Example: Determine the dynamic power dissipation in the last stage of a 6-stage CMOS pad driver if used to drive a 10pF capacitive load if the system clock is 500MHz and the output changes with 50% of the clock transitions. Assume pad driver with OD of θ=2.5 and V_{DD}=3.5V

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K

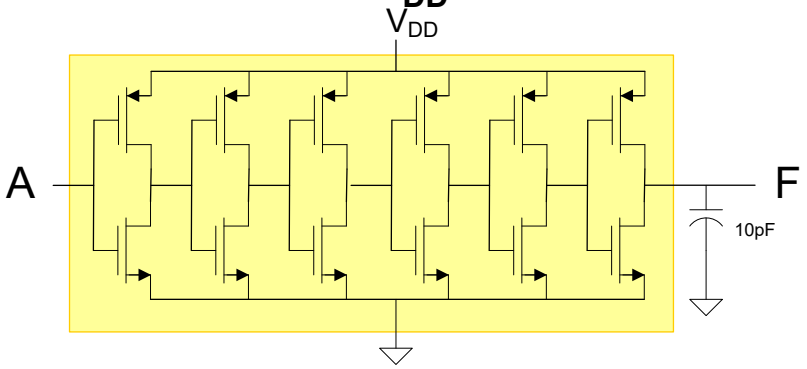


Solution: (assume output changes with 50% of clock transitions)

$$P_{DYN}=\frac{f_{CL}}{2}C_L V_{DD}^2=\frac{5E8}{2}\bullet 10pF\bullet 3.5^2=30.5\ mW$$

Note this solution is independent of the OD and the process

Example: Determine the power that would be required in the last stage of a CMOS pad driver to drive a 32-bit data bus off-chip if the capacitive load on each line is 10pF. Assume the clock speed is 500MHz and that each bit has an average 50% toggle rate. Assume a pad driver with OD of θ=2.5 and V_{DD}=3.5V



In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K

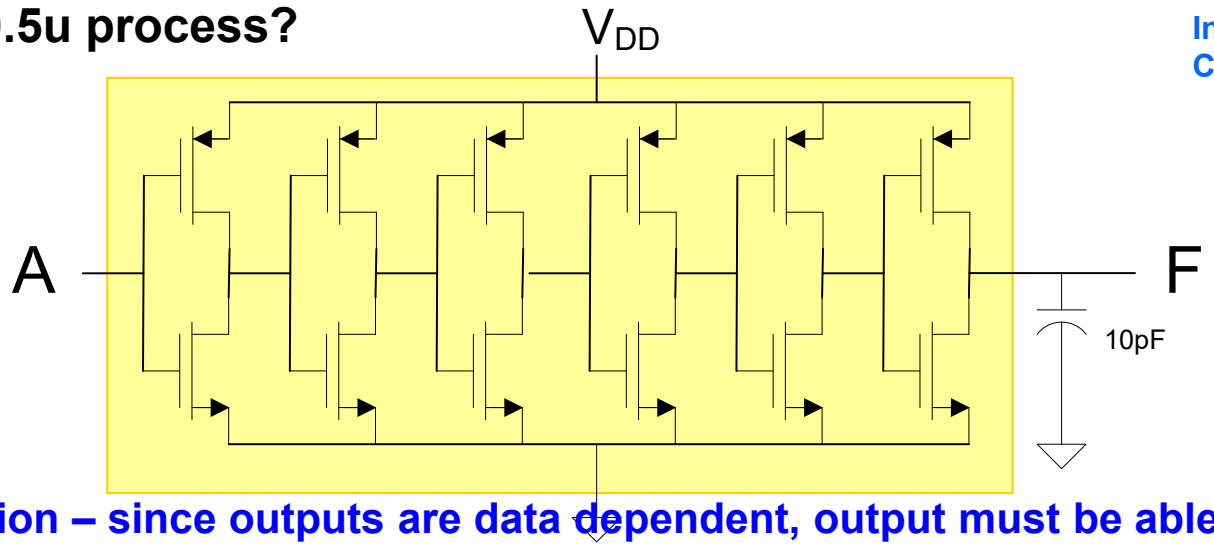
Solution:

$$P_{DYN}=32 \bullet \frac{f_{CL}}{2} C_L V_{DD}^2 = 32 \bullet \frac{5E8}{2} \bullet 10\text{pF} \bullet 3.5^2 = 980\text{mW}$$

Note: A very large amount of power is required to take a large bus off-chip if bus has a high rate of activity.

Example: Will the CMOS pad driver actually be able to drive the 10pF load with a system clock of 500MHz as in the previous example in the 0.5u process?

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K



Solution – since outputs are data dependent, output must be able to operate 500Mz:

$$t_{CLK} = \frac{1}{500MHz} = 2nsec$$

$$Fl_{load} = \frac{10\,pF}{4\,fF} = 2500 \qquad OD_6 = \theta^5 = 98$$

$$t_{PROP} = 5\theta \bullet t_{REF} + \frac{Fl_{load}}{OD_6} t_{REF} \qquad \frac{Fl_{load}}{OD_6} = \frac{2500}{98} \cong 25$$

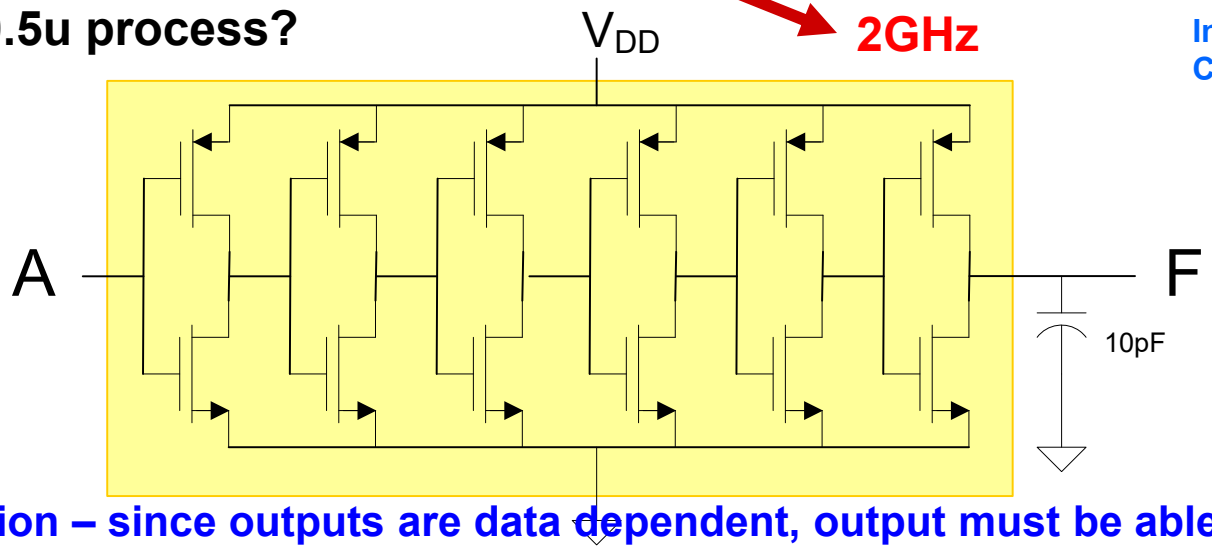
$$t_{prop} = 5 \bullet 2.5 \bullet 20psec + 25 \bullet 20psec = (12.5 + 25)20psec = 0.75nsec$$

since t_{CLK}>t_{PROP}, this pad driver can drive the 10pF load at 500MHz

Example: Will the CMOS pad driver actually be able to drive the 10pF load with a system clock of 500MHz as in the previous example in the 0.5u process?

2GHz

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K



Solution – since outputs are data dependent, output must be able to operate 500Mz:

$$t_{CLK} = \frac{1}{500MHz} = 2nsec$$

2GHz

0.5nsec

$$t_{PROP} = 5 \bullet \theta \bullet t_{REF} + \frac{Fl_{load}}{OD_6} t_{REF}$$

$$Fl_{load} = \frac{10\,pF}{4\,fF} = 2500 \qquad OD_6 = \theta^5 = 98$$

$$\frac{Fl_{load}}{OD_6} = \frac{2500}{98} \cong 25$$

$$t_{prop} = 5 \bullet 2.5 \bullet 20psec + 25 \bullet 20psec = (12.5 + 25)20psec = 0.75nsec$$

since $t_{CLK} < t_{PROP}$, this pad driver can drive the 10pF load at 500MHz

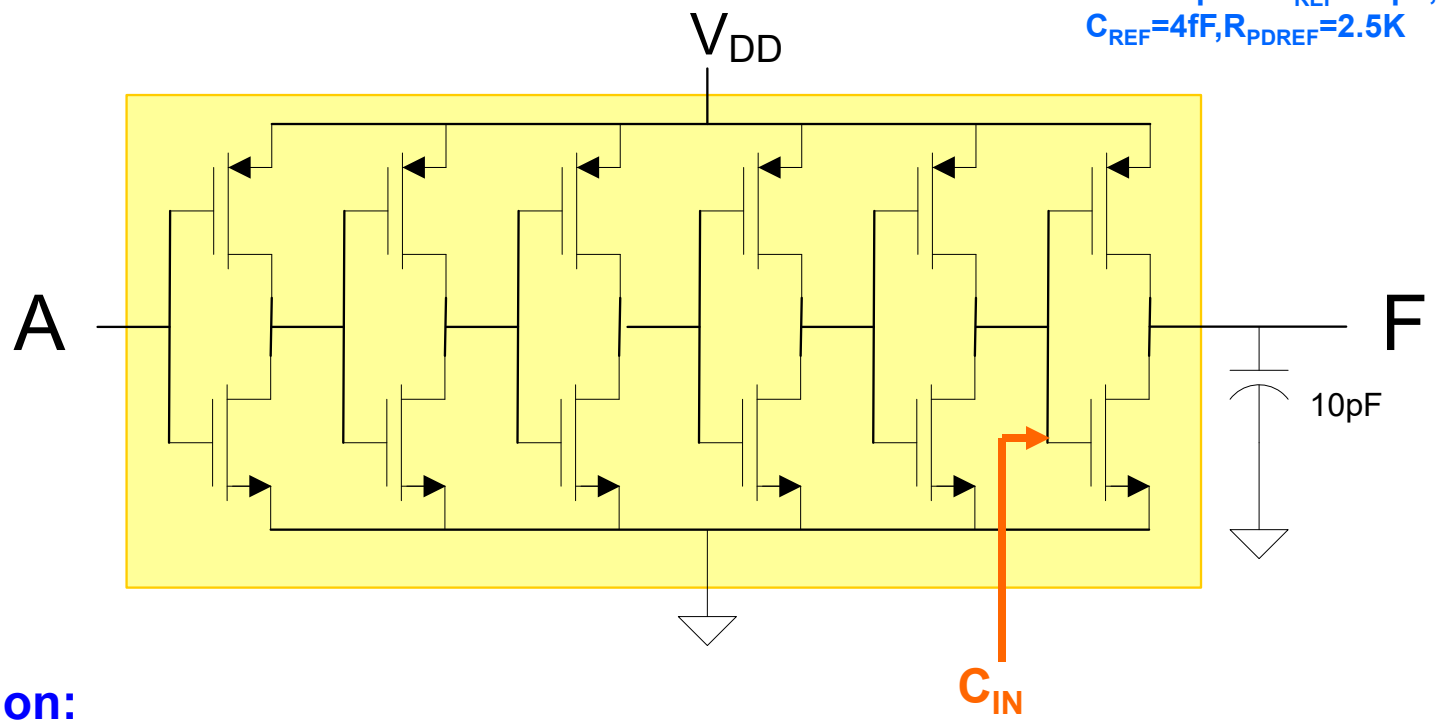
$t_{CLK} < t_{PROP}$

can not

2GHz

Example: Determine the dynamic power dissipation in the next to the last stage of a 6-stage CMOS pad driver if used to drive a 10pF capacitive load if clocked at 500MHz. Assume pad driver with OD of θ=2.5 and V_{DD}=3.5V

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K



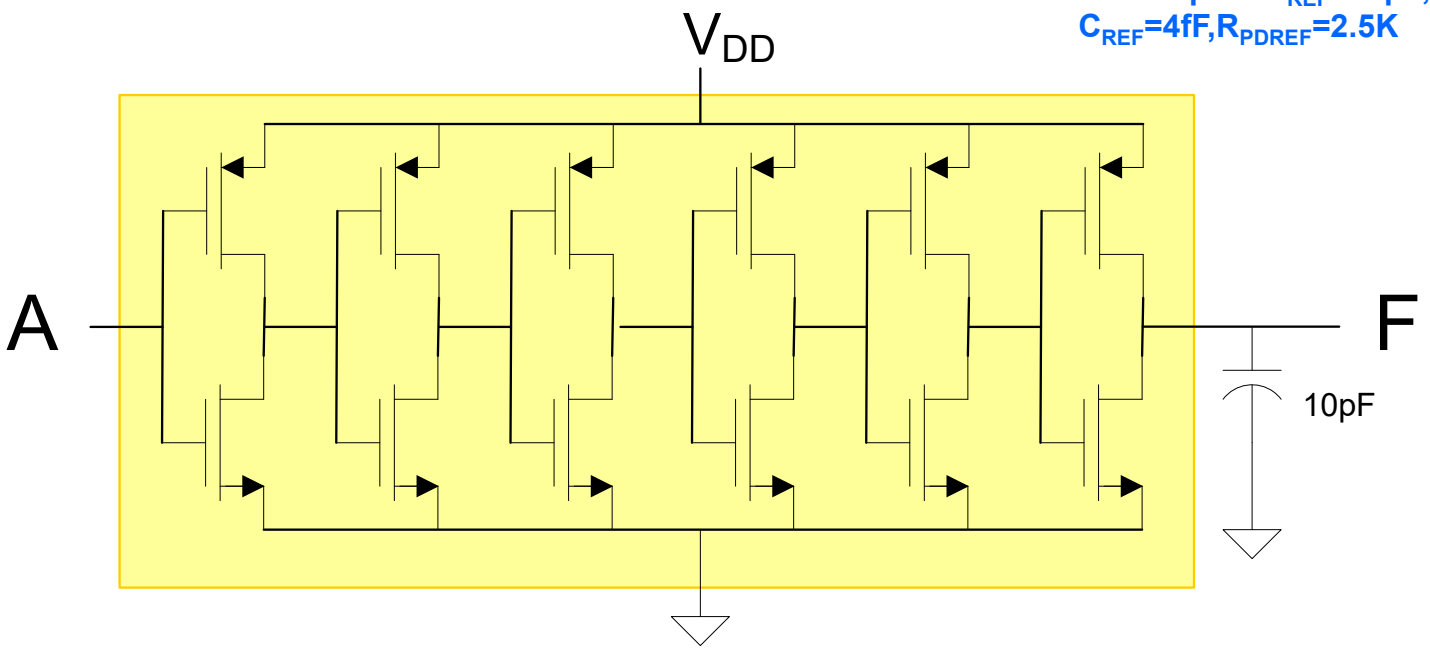
Solution:

$$C_{IN}=\theta^5 C_{REF}=2.5^5 \bullet 4fF=390fF$$

$$P_{DYN}=f_{CL} C_L V_{DD}^2 \quad =5E8 \bullet 390fF \bullet 3.5^2 = 2.4mW$$

Example: Is the 6-stage CMOS pad driver adequate to drive the 10pF capacitive load as fast as possible? Assume pad driver with OD of θ=2.5 and V_{DD}=3.5V

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K



Solution:

$$n_{OPT} = \ln\left(\frac{C_L}{C_{REF}}\right) = \ln\left(\frac{10pF}{4fF}\right) = 7.8$$

No – an 8-stage pad driver would drive the load much faster (but is not needed If clocked at only 500MHz)

Digital Circuit Design

- Hierarchical Design
- Basic Logic Gates
- Properties of Logic Families
- Characterization of CMOS Inverter
- Static CMOS Logic Gates
  - Ratio Logic
- Propagation Delay
  - Simple analytical models
    - FI/OD
    - Logical Effort
  - Elmore Delay
- Sizing of Gates
  - The Reference Inverter
- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
- Other Logic Styles
  - Array Logic
  - Ring Oscillators

---

done

partial

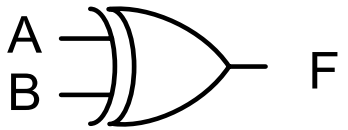
Logic Styles

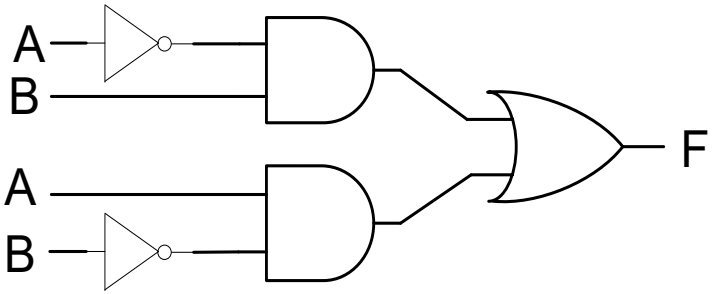
- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

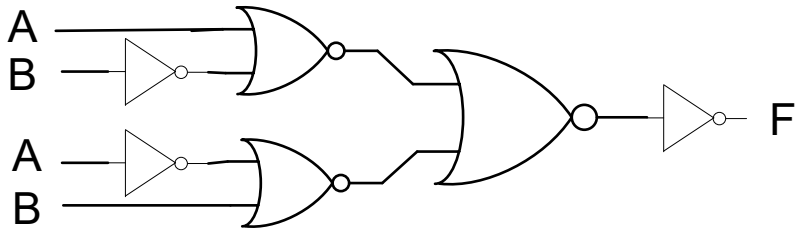
Static CMOS

Example: $F=A \oplus B$

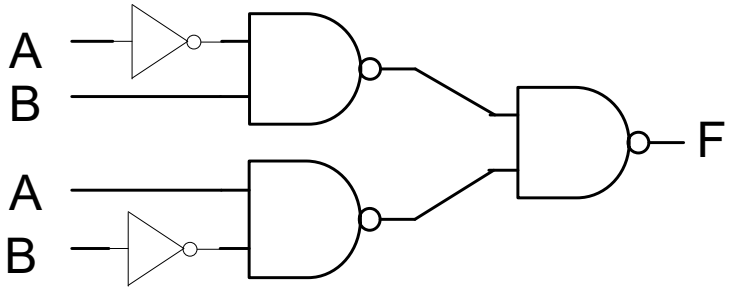
$F=A\bar{B}+\bar{A}B$







18 transistors, 4 levels of logic



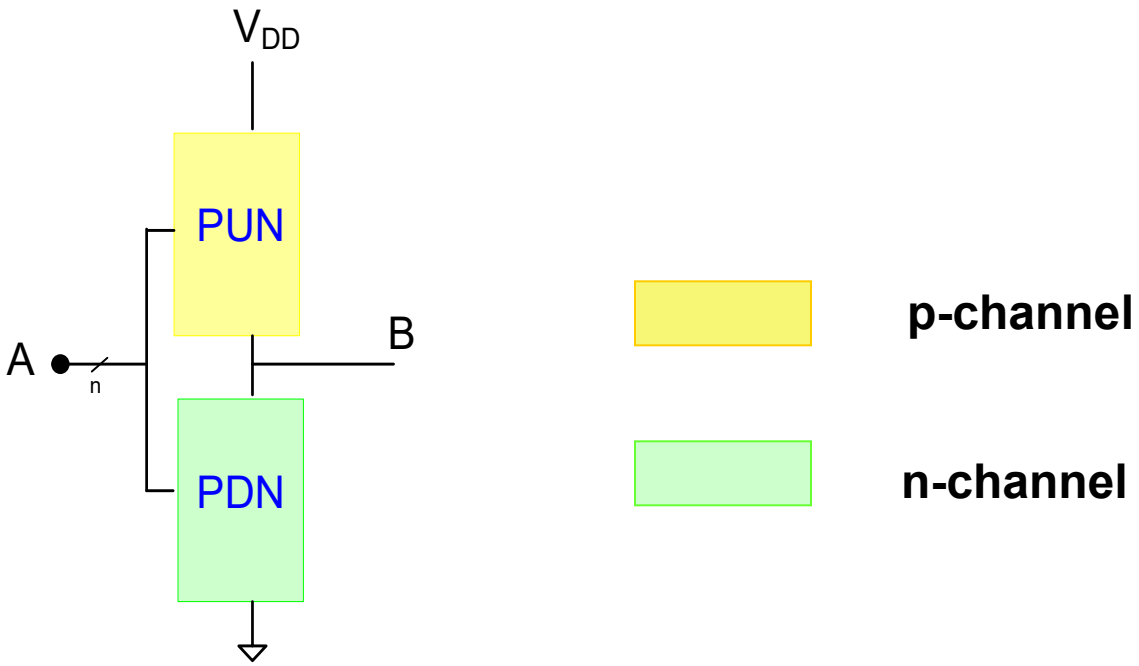
16 transistors, 3 levels of logic

Number of devices is unacceptably large in some applications

Dynamic Power Dissipation can be large, in particular for multiple-input NOR gates because of their large Fan In

Static CMOS Logic Gates

Consider any multiple-input NAND or NOR gate . They can be represented as:



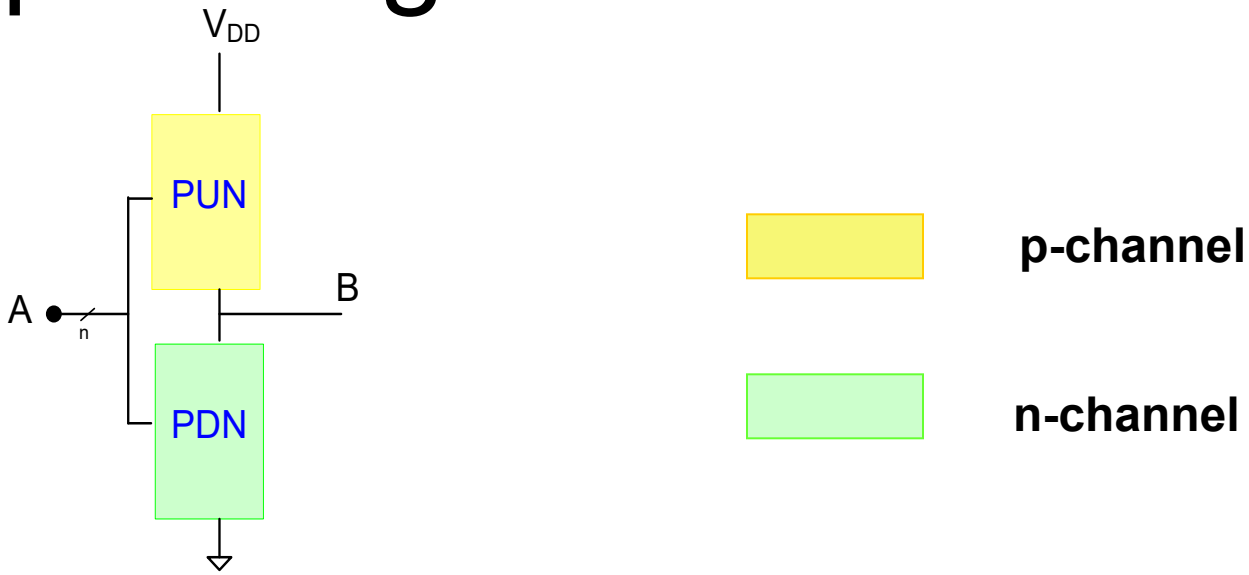
- **Implement $\overline{B}$ in PDN**
- **Implement B in PUN with complimented input variables**
- **Zero static power dissipation**
- **$V_H=V_{DD}$, $V_L=0V$ (or V_{SS})**
- **Complimented input variables often required**

Have implemented the logical function twice (once in PU, again in PD) and this is a major contributor to increased area and dynamic power dissipation

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Complex Logic Gates



- Implement $\overline{B}$ in PDN
- Implement B in PUN with complimented input variables
- Zero static power dissipation
- $V_H=V_{DD}$, $V_L=0V$ (or V_{SS})
- Complimented input variables often required

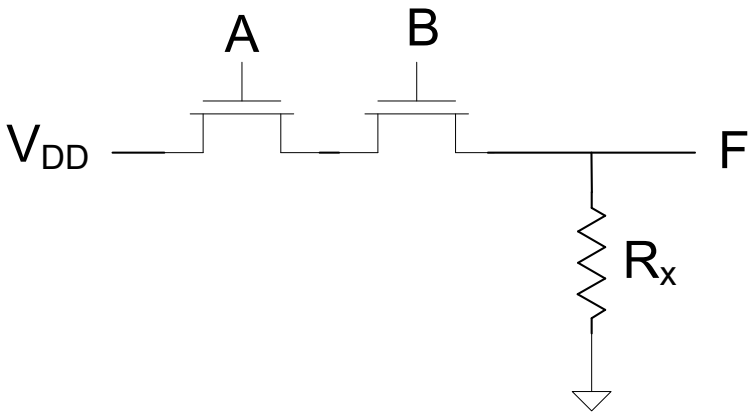
Can reduce the number of levels of logic and the total device count for some functions

Have implemented the logical function twice (once in PU, again in PD) and this is a major contributor to increased area and dynamic power dissipation

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

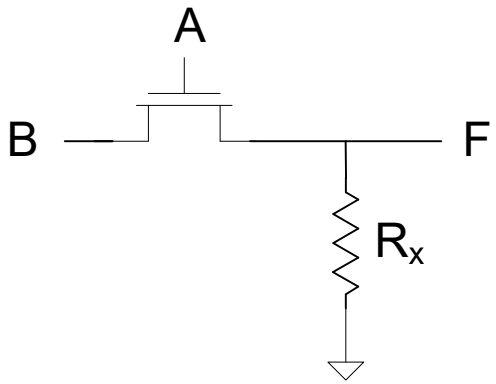
Pass Transistor Logic



F=AB

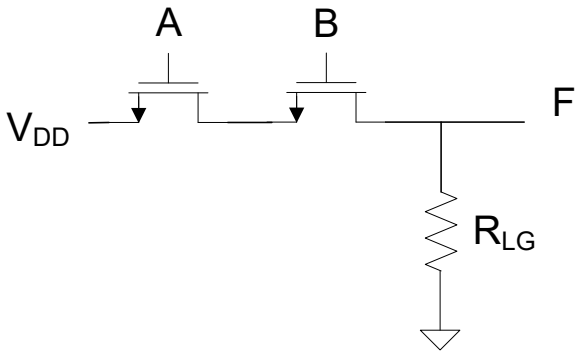
AND Gate

Requires only 3 components



Even simpler AND gate, requires only 2 components

Pass Transistor Logic



F=A•B

Observations about PTL

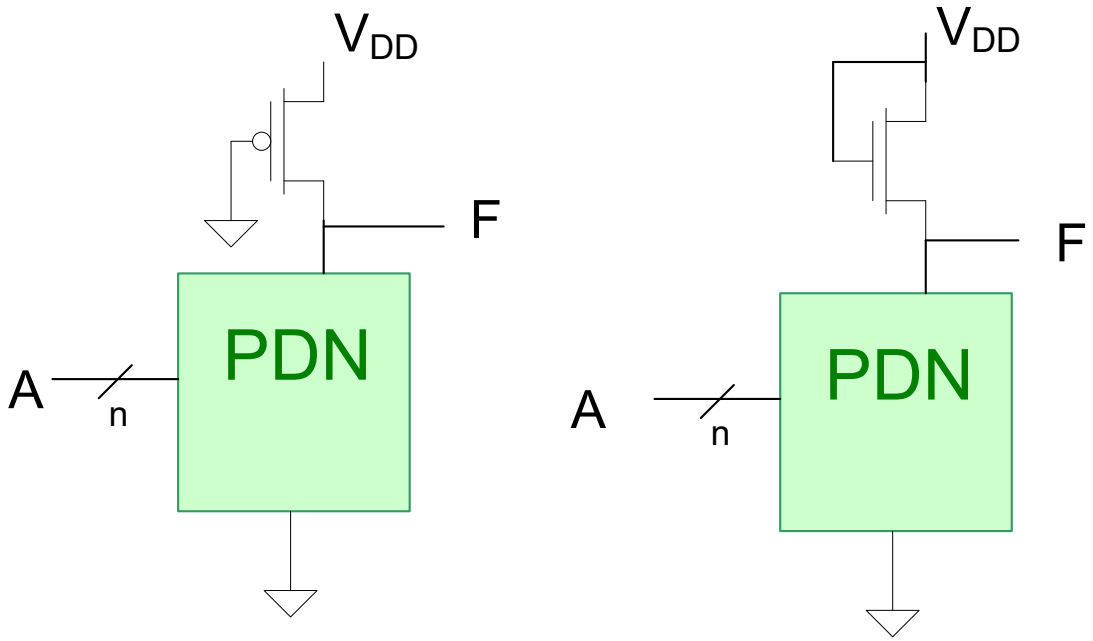
- Low device count implementation of non inverting function (can be dramatic)
- Logic Swing not rail to rail
- Static power dissipation not 0 when F high
- R_{LG} may be unacceptably large
- Slow t_{LH}
- Signal degradation can occur when multiple levels of logic are used
- Widely used in some applications
- Implements basic logic function only once!
- Fan In can be very small – so low dynamic power dissipation !

Is there a way to take advantage of the dynamic power dissipation advantages of a small fan-in without the dramatic energy penalty of a large static power dissipation?

Logic Styles

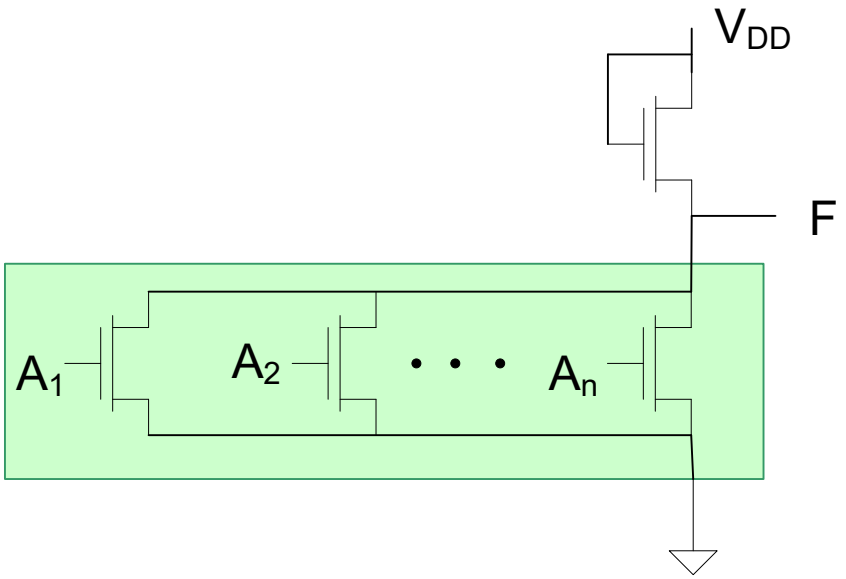
- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Pseudo NMOS Logic



- **May be viewed as a special case of PTL**
- **Ratioed Logic**
- **Static power dissipation not 0 (in PD state)**
- **Often used for really large number of inputs – e.g. NOR**
- **Only one additional transistor for each additional Boolean input**
- **Would be particularly useful for identifying one (or more) of many events that occur very infrequently**

Pseudo NMOS Logic



n could be several hundred or even several thousand

Static power dissipation independent of the number of inputs

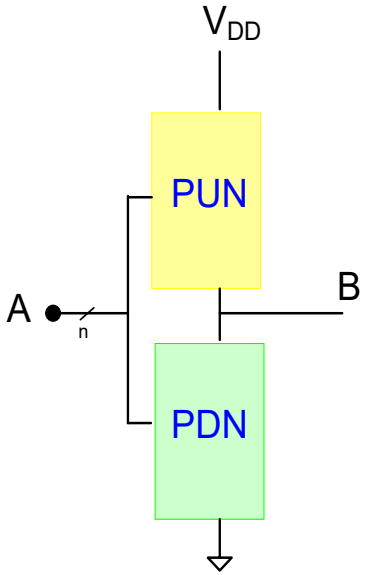
May justify paying the static power dissipation penalty if a large number of inputs are needed, particularly if the conditions to trigger the HL transition occur very rarely

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Dynamic Logic

- **PTL reduced complexity of either PUN or PDN to single "resistor"**
- **PTL relaxed requirement of all n-channel or all p-channel devices in PUN/PDN**



What is the biggest contributor to area?

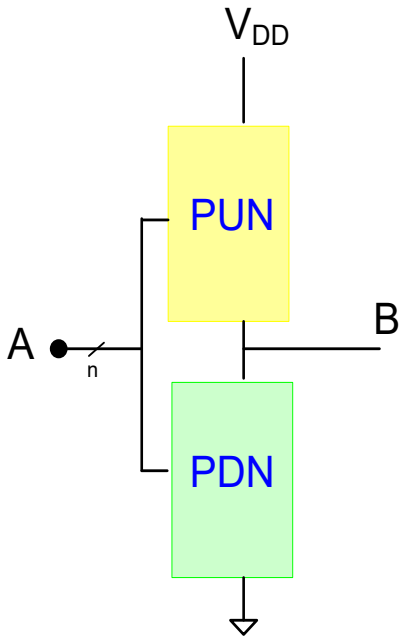
PUN (3X active area for inverter, more for NOR gates, and Well)

What is biggest contributor to dynamic power dissipation ?

PUN and is responsible for approximately 75% of the dynamic power dissipation in equal rise/fall inverter, and much more in NOR gates !

Can the PUN be <u>eliminated</u> W/O compromising signal levels and power dissipation?

Dynamic Logic

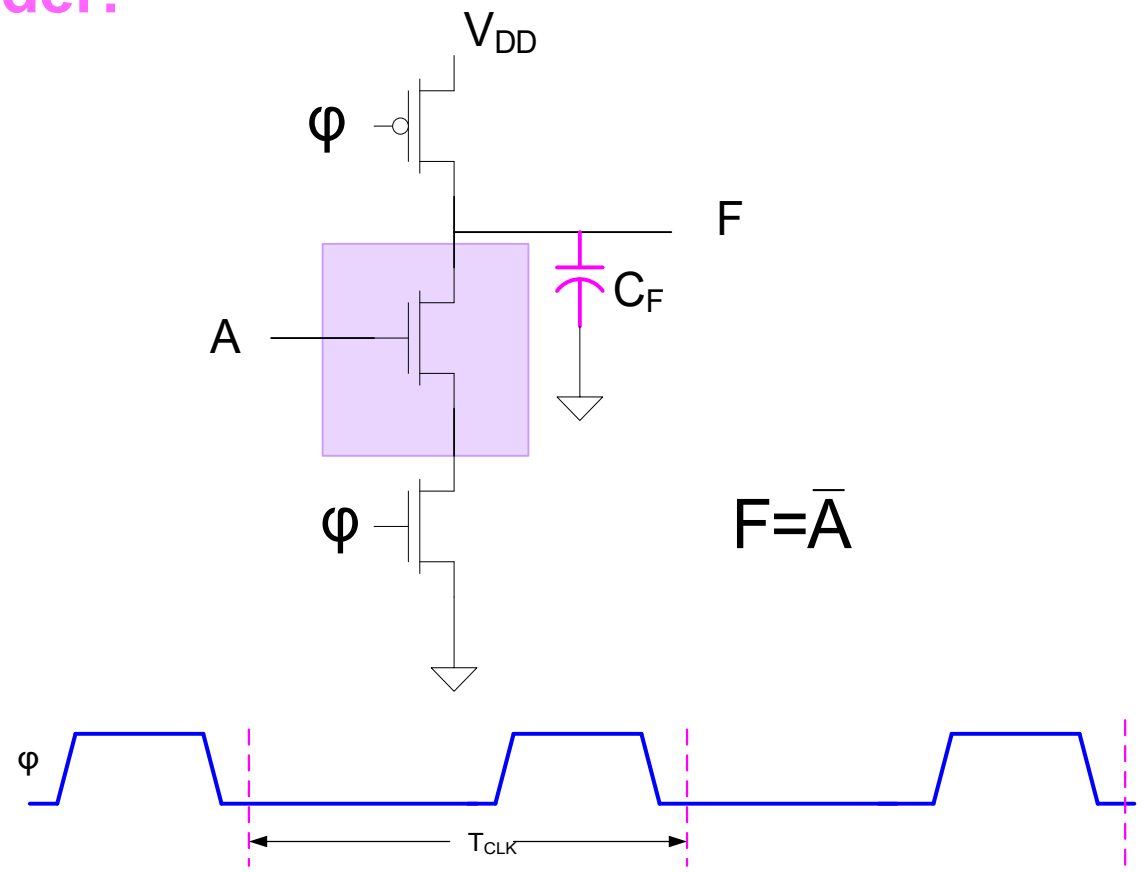


Can the PUN be eliminated W/O compromising signal levels and power dissipation?

Benefits could be most significant !

Dynamic Logic

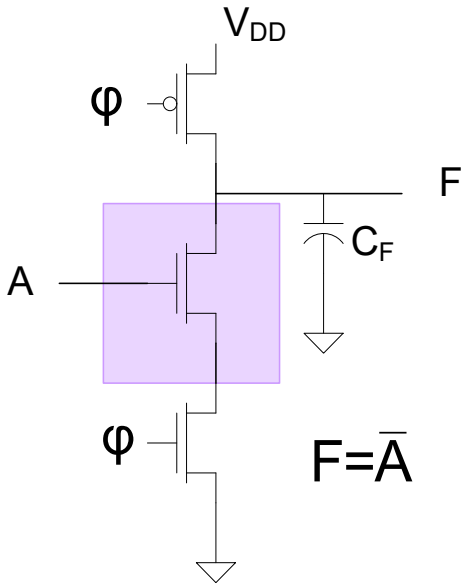
Consider:

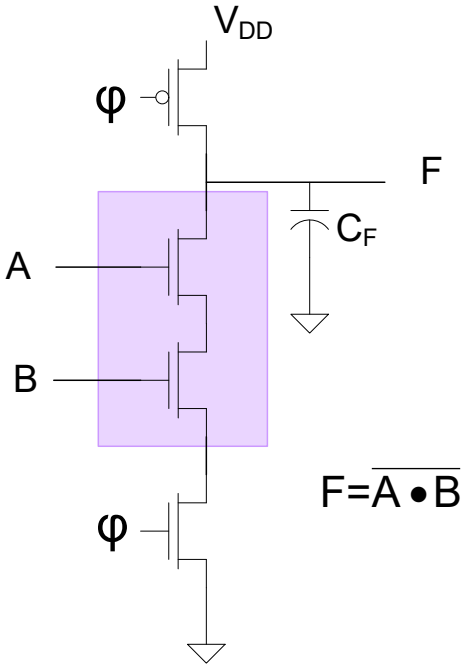


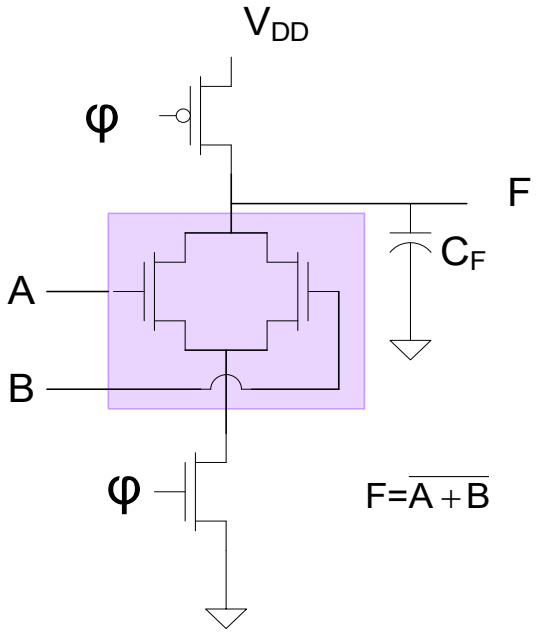
Precharges C_F to "1" when φ is low
F either stays high if output is to be high or changes to low on evaluation
C_F is usually the parasitic capacitances on the node (drain diffusion and gates)

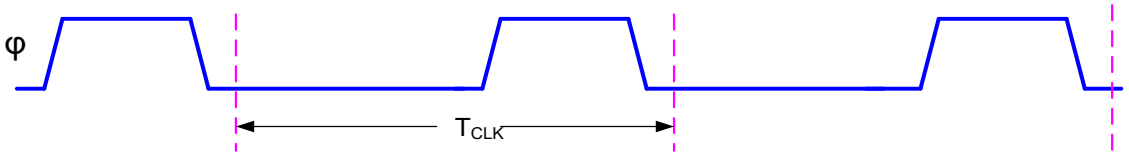
Dynamic Logic

Consider:





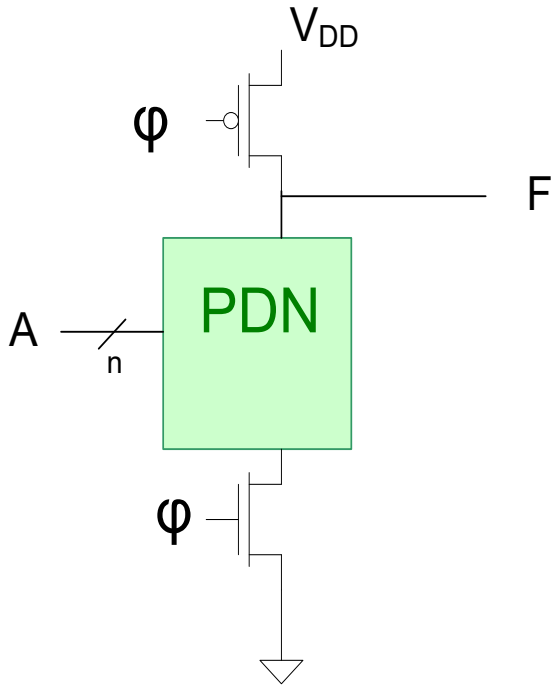




- **Termed Dynamic Logic Gates**
- **Parasitic capacitors actually replace C_F**
- **If Logic Block is n-channel, will have rail to rail swings**
- **Logic Block is simply a PDN that implements $\overline{F}$**

Dynamic Logic

Basic Dynamic Logic Gate

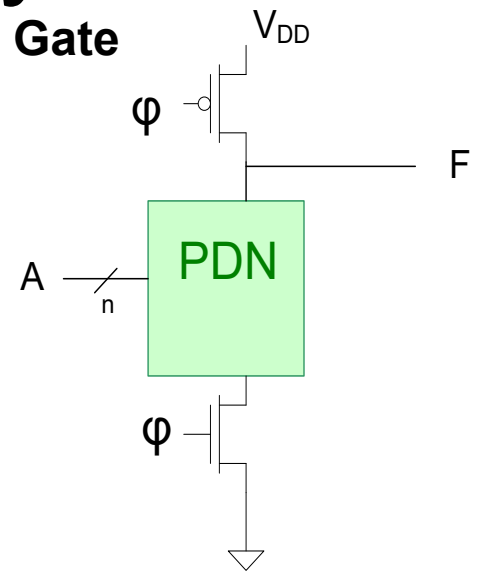


Any of the PDNs used in complex logic gates would work here !

- **Have eliminate the PUN !**
- **Ideally will have a factor of 4 or more reduction in C_{IN}**
- **Ideally will have a factor of 4 or more reduction in dynamic power dissipation relative to that of equal rise/fall !**
- **Ideally will have a factor of 2 reduction in dynamic power dissipation relative to that of minimum size!**

Dynamic Logic

Basic Dynamic Logic Gate



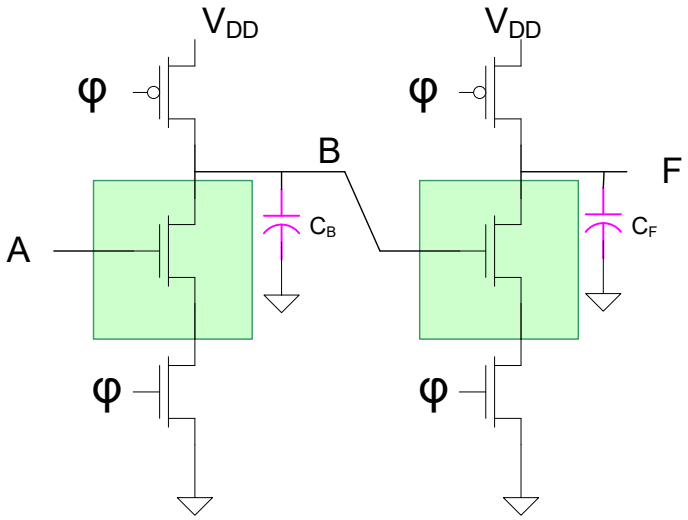
Advantages:

- **Lower dynamic power dissipation (Ideally 4X)**
- **Improved speed (ideally 4X)**

Limitations:

- **Output only valid during evaluate state**
- **Need to route a clock**
  (and this dissipates some power)
- **Premature Discharge !**
- **More complicated**
- **Charge storage on internal nodes of PDN**
- **No Static hold if output H**
- **Dynamic power dissipation in pre-charge circuit**

Dynamic Logic



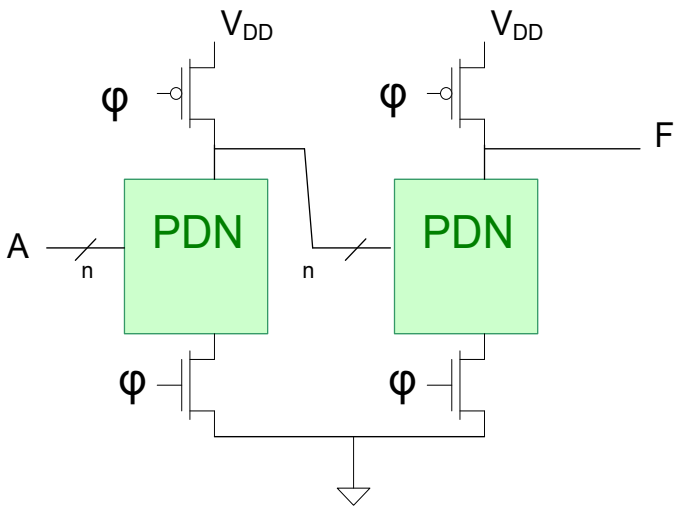
Premature Discharge Problem

B will be pulled high during the pre-charge state and try to discharge C_F thus pulling F low

If input A is high, then if F goes low at the start of the evaluate cycle, there is no way to recover a high output later in the evaluate phase - i.e. there may be a boolean error!.

Can not reliably cascade dynamic logic gates !

Dynamic Logic



Premature Discharge Problem

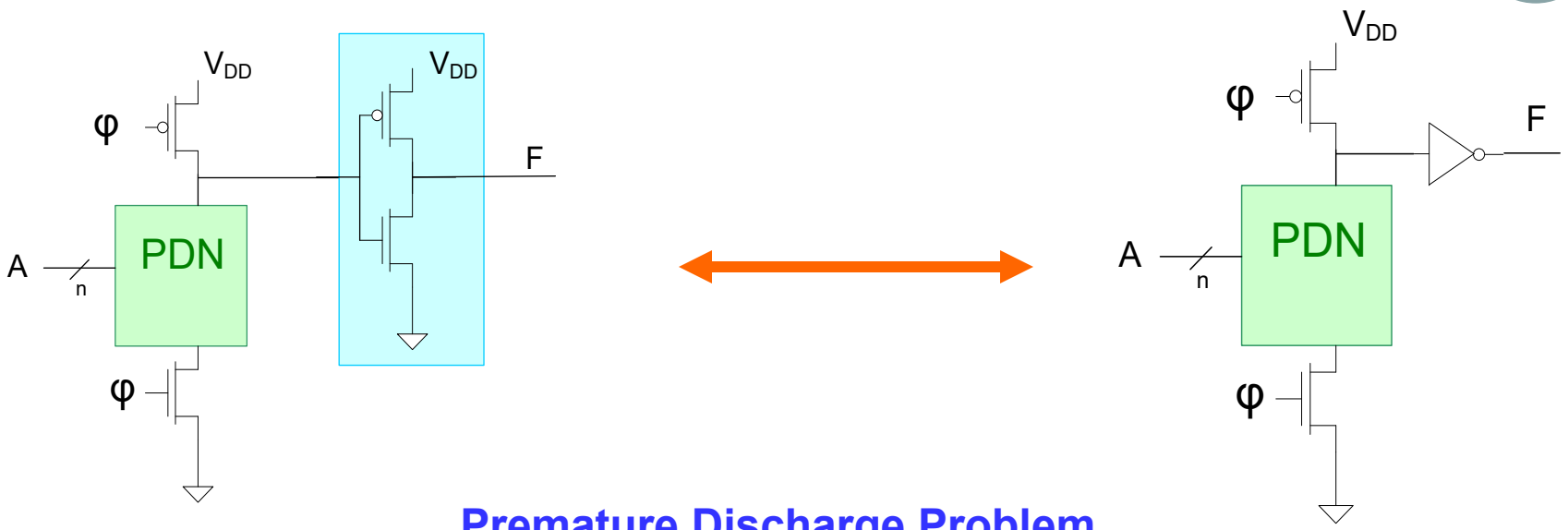
This problem occurs when any inputs to an arbitrary dynamic logic gate create an R_{PD} path in the PDN during at the start of the evaluate phase that is not to pull down later in that evaluate phase

How can this problem be fixed?

Precharging to the low level all inputs to a PDN that may change to the high state later in the evaluate cycle (called domino)

Alternating gates with n-channel and p-channel pull networks (Zipper Logic)

Dynamic Logic



Premature Discharge Problem

Adding an inverter at the output will cause F to precharge low so it can serve as input to subsequent gate w/o causing premature discharge

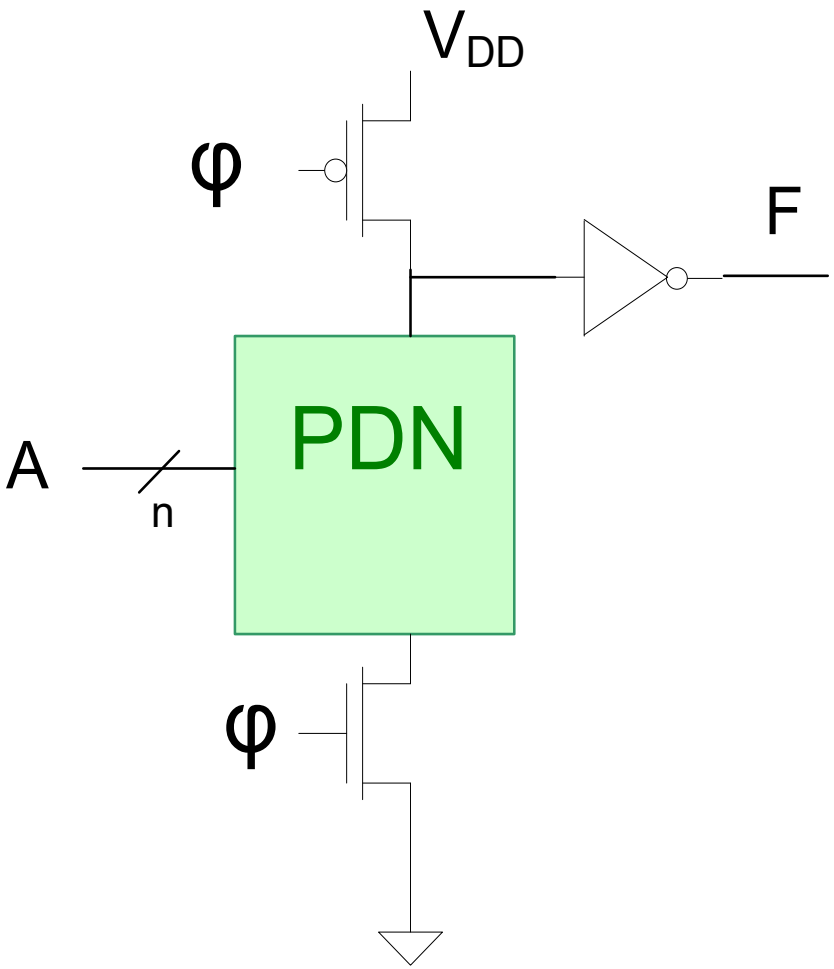
Implement F instead of $\overline{F}$ in the PDN

Termed Domino Logic

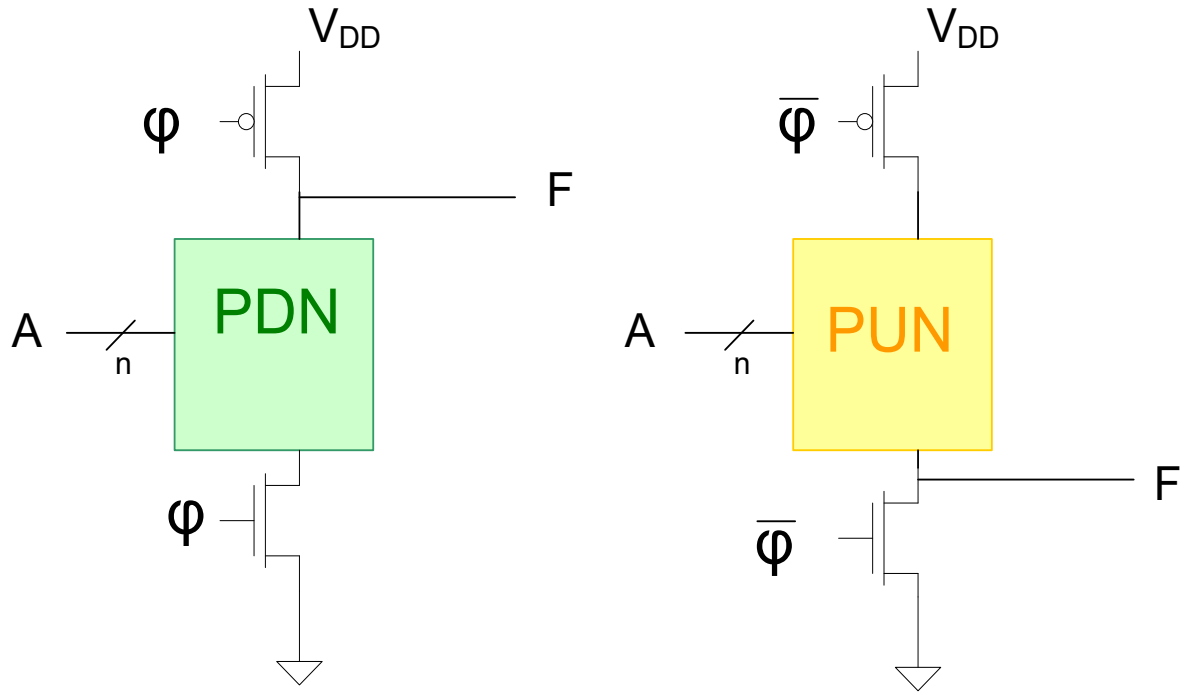
Some additional dynamic power dissipation in the inverter

Some additional delay during the evaluate state in inverter

Domino Logic

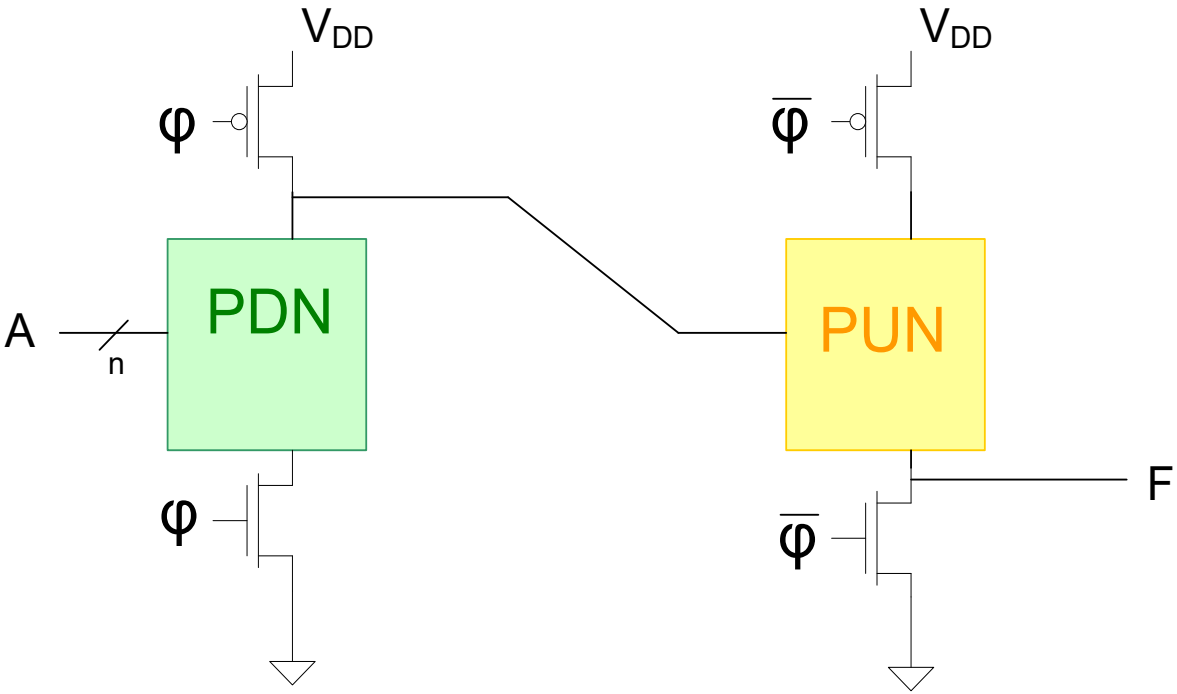


Dynamic Logic



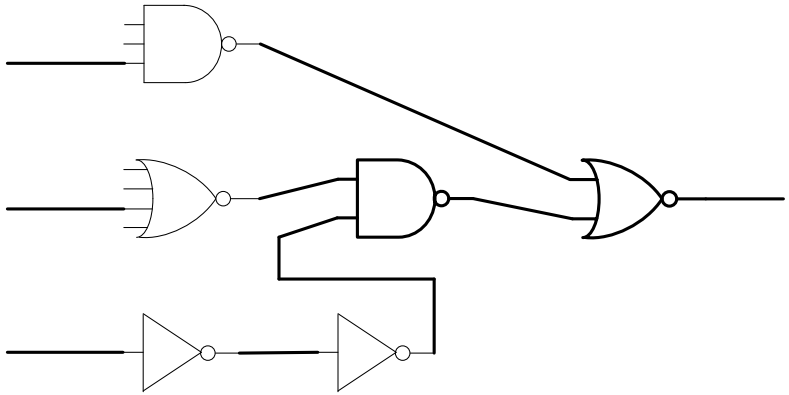
- **p-channel logic gate will pre-charge low**
- **Phasing of PUN and PDN networks is reversed**
- **Some performance loss with p-channel logic devices**
- **Direct coupling between alternate type dynamic gates is possible without causing a premature discharge problem**

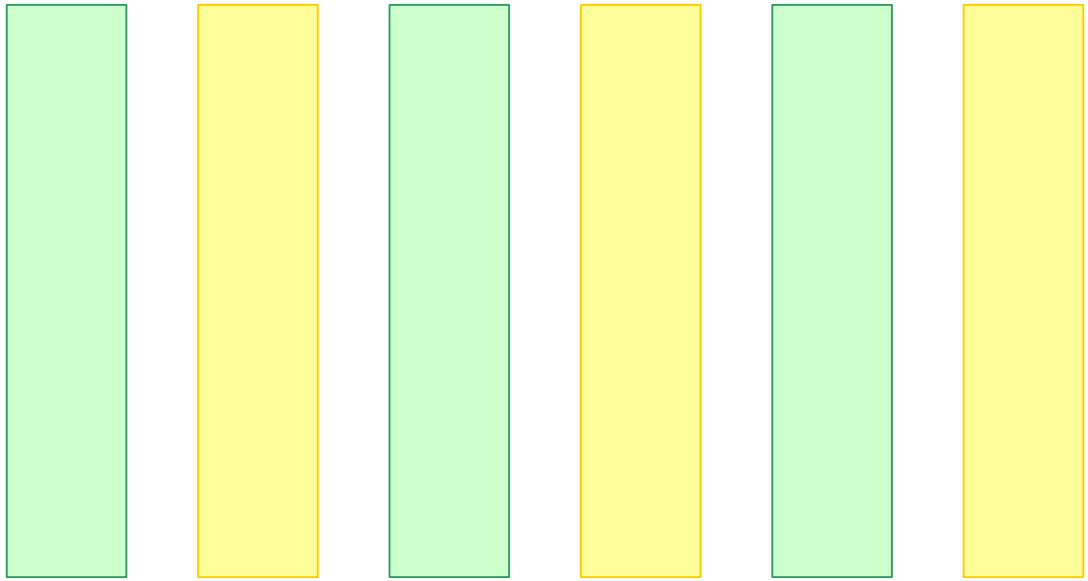
Dynamic Logic



Direct coupling between alternate type dynamic gates

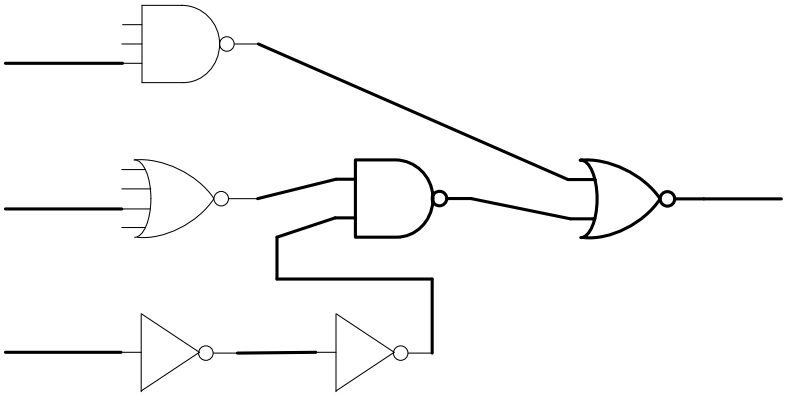
Zipper Logic

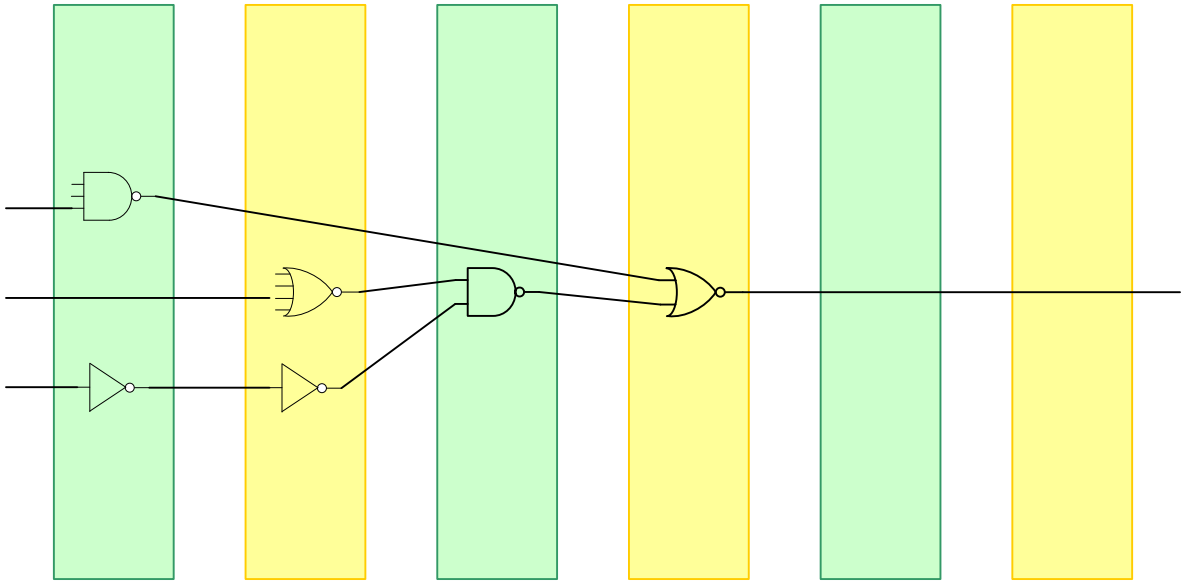




Map gates to appropriate precharge type

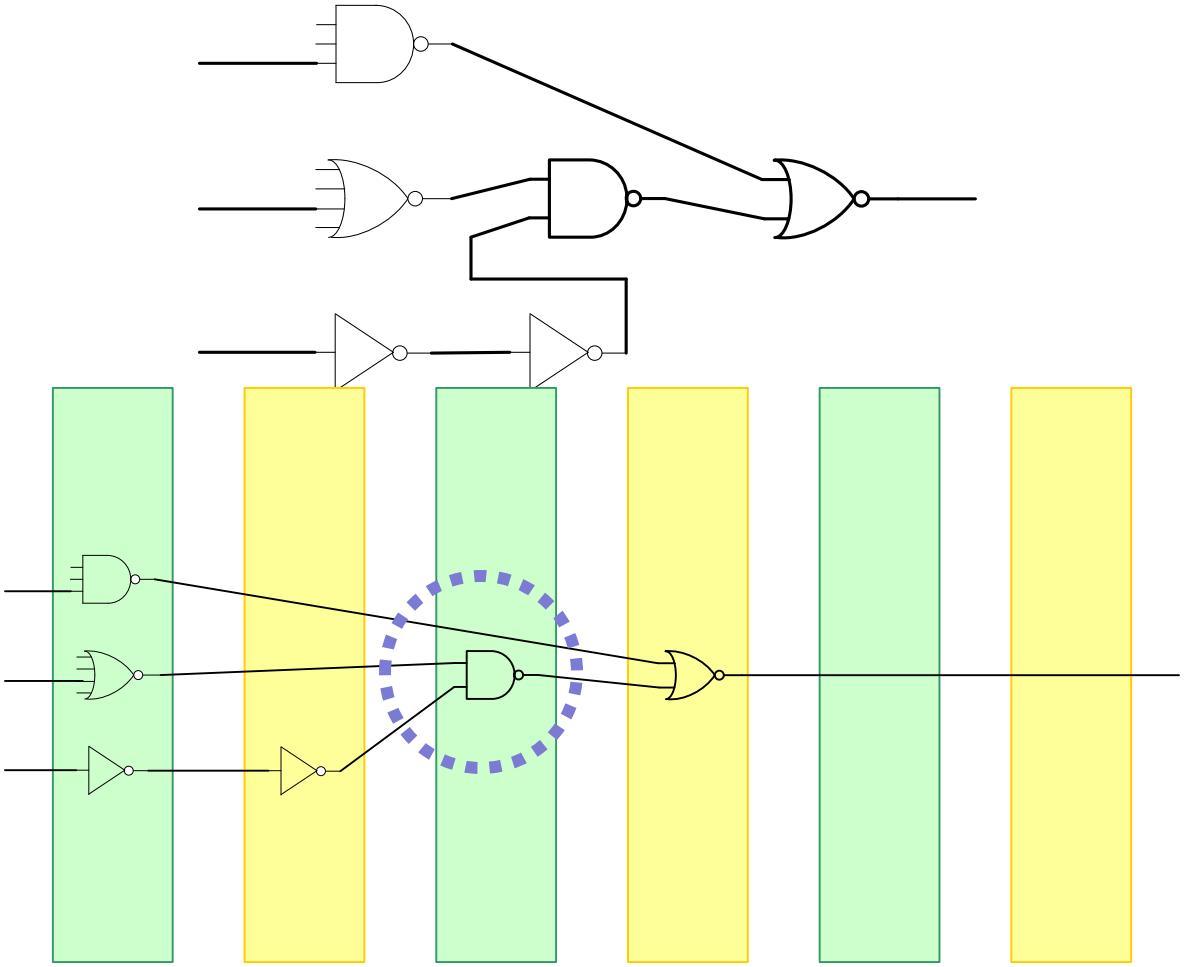
Zipper Logic





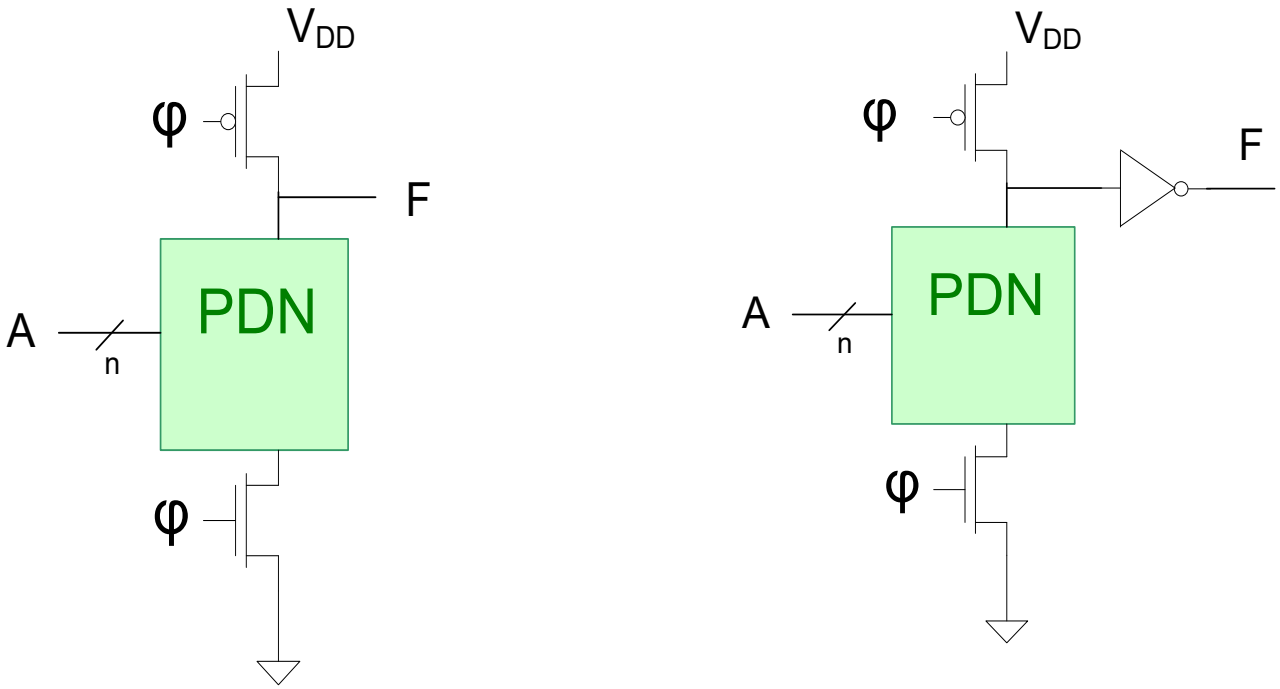
Acceptable Implementation in Zipper

Zipper Logic



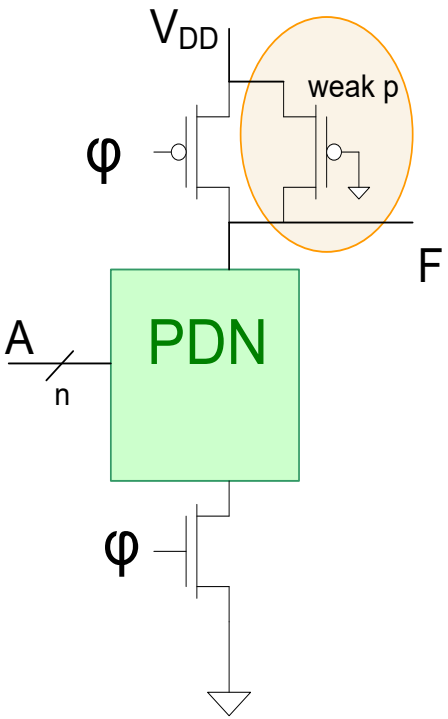
Unacceptable Implementation in Zipper
- Premature discharge at output of 2-input NAND

Static Hold Option

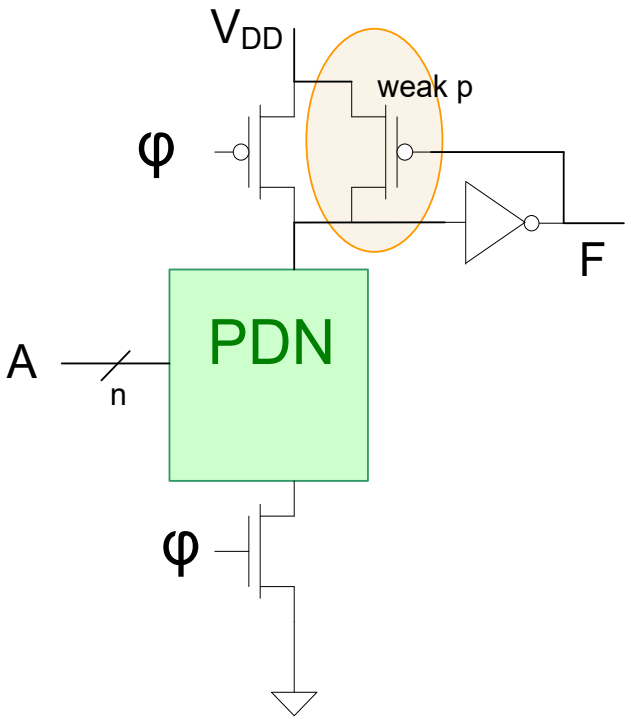


If not clocked, charge on upper node of PDN will drain off causing H output to degrade

Static Hold Option

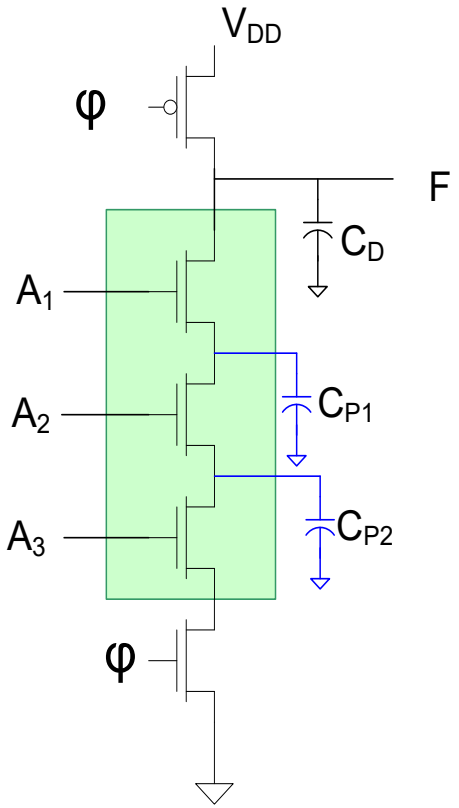


- **weak p will hold charge**
- **size may be big (long L)**
- **some static power dissipation**
- **can use small current source**
- **sometimes termed “keeper”**



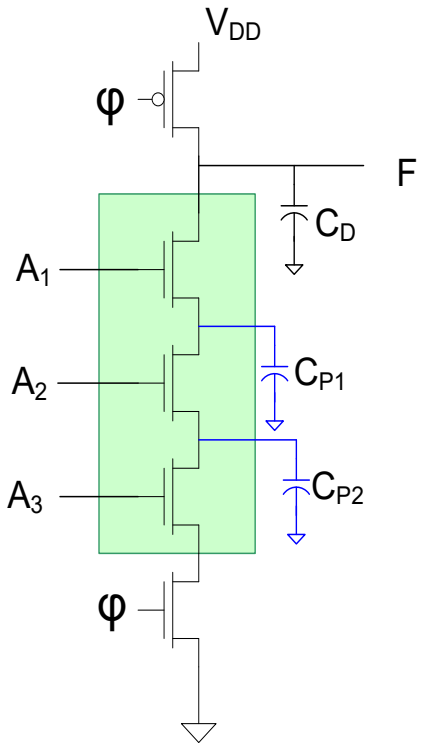
- **weak p will hold charge**
- **size may be big (long L)**
- **can eliminate static power with domino**
- **sometimes termed “keeper”**

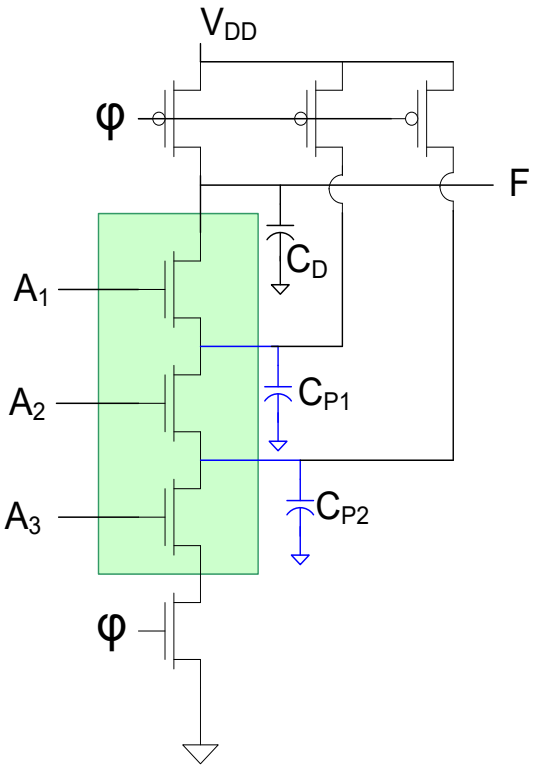
Charge stored on internal nodes of PDN



If voltage on C_{P1} and C_{P2} was 0V on last evaluation, these may drain charge (charge redistribution) on C_P if output is to evaluate high (e.g. On last evaluation $A_1=A_2=A_3=H$, on next evaluation $A_3=L$, $A_1=A_2=H$.)

Charge stored on internal nodes of PDN





Can precahrge internal nodes to eliminate undesired charge redistribution

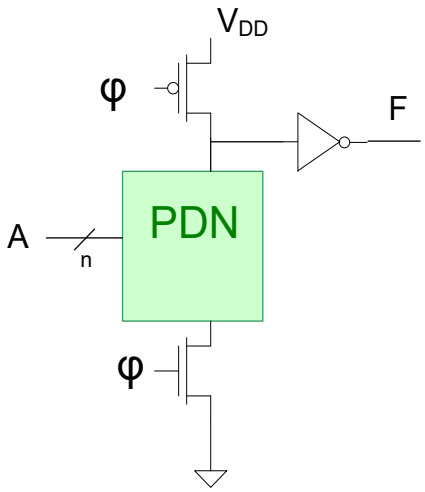
Dynamic Logic

Many variants of dynamic logic are around

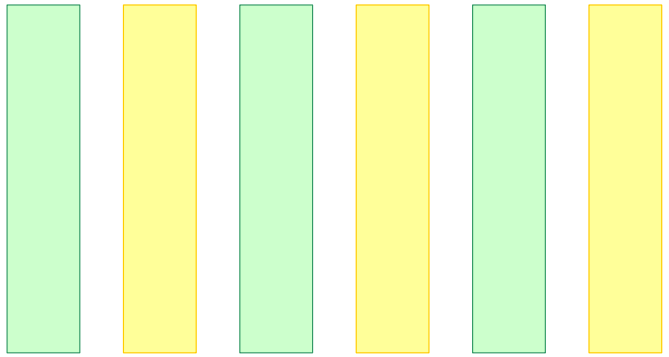
- **Domino**
- **Zipper**
- **Ratio-less 2-phase**
- **Ratio-less 4-phase**
- **Output Prediction Logic**
- **Fully differential**
- **…**

Benefits disappear, however, when interconnect (and diffusion) capacitances dominate gate capacitances

Future of Dynamic Logic



Domino



Zipper

Dynamic logic will likely disappear in deep sub-micron processes because interconnect parasitics will dominate gate parasitics

Other types of Logic (list is not complete and some have many sub-types)

From Wikipedia:

B
BiCMOS
C
CMOS
Major emphasis in this course
Cascode Voltage Switch Logic
Clocked logic
Complementary Pass-transistor Logic
Current mode logic
Current steering logic
D
Differential TTL
Diode logic
Diode–transistor logic
Domino logic
Dynamic logic (digital logic)
E
Emitter-coupled logic
F
Four-phase logic
G
Gunning Transceiver Logic
H
HMOS
HVDS
High-voltage differential signaling
I
Integrated injection logic
L
LVDS
Low-voltage differential signaling
Low-voltage positive emitter-coupled logic
M
Multi-threshold CMOS
N
NMOS logic
P
PMOS logic
Philips NORbits
Positive emitter-coupled logic
R
Resistor-transistor logic
S
Static logic (digital logic)
T
Transistor–transistor logic

Stay Safe and Stay Healthy !

End of Lecture 43