集成电路原理与设计

12. 组合逻辑门

Combinational Logic Gates

宋爽

shuangsonghz@zju.edu.cn

Syllabus

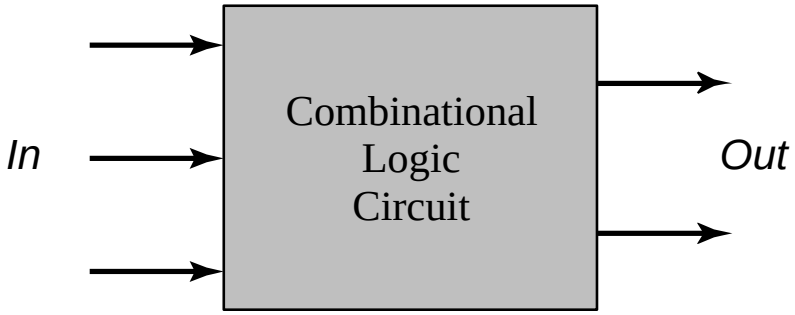
课数	内容	课数	内容
1	导论	9	差分放大器
2	器件模型	10	运算放大器
3	电容特性，小信号模型	11	导线
4	工艺流程	12	反相器
5	模拟基本单元	13	反相器
6	电流镜与基准	14	组合逻辑
7	单级放大器	15	集成电路专题讲座一
8	课堂测验	16	集成电路专题讲座二

Goals

- In-depth discussion of logic families in CMOS
  - Static and Dynamic
  - Non-retioed and ratioed logic
  - Pass-transistor
- Optimizing a logic gate for area, speed, energy, or robustness

Combinational vs. Sequential Logic

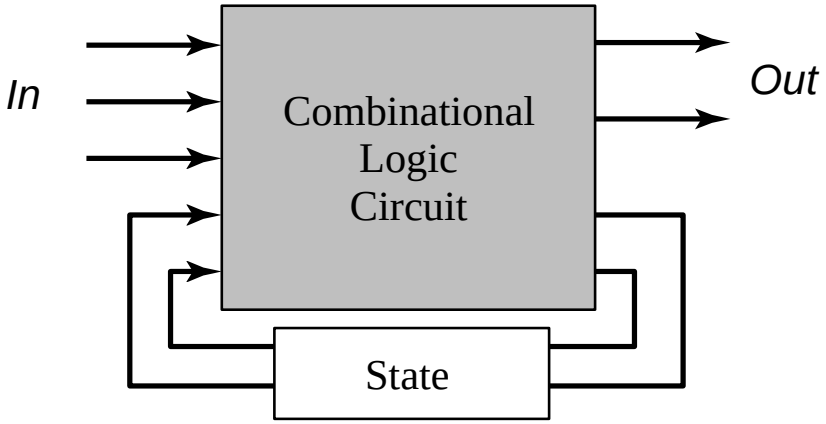
Combinational



Output=*f*(In)

组合逻辑：输出只与
当前状态有关

Sequential



Output=*f*(In, Previous In)

时序逻辑：输出与当前状态
及上一时钟周期状态有关

Combinational logic expressions

- Combinational logic: function value is a combination of **function arguments**. (布尔函数)
- A **logic gate** implements a particular logic function.
- Both specification (**logic equations**) and implementation (logic **gate networks**) are written in Boolean logic.

Why designing gates for logic functions is non-trivial:

- may not have logic gates in the library for all logic expressions;
- a logic expression may map into gates that consume a lot of **area, delay, or power**.

CMOS Circuit Styles

Static CMOS

output connected to either VDD or GND via a low-resistance path

- **high noise margins**
- **low output impedance**, high input impedance
- no steady state path between VDD and GND
- delay a function of load capacitance and transistor resistance

Dynamic CMOS

relies on temporary storage of signal values on the capacitance of ***high-impedance*** *circuit nodes*

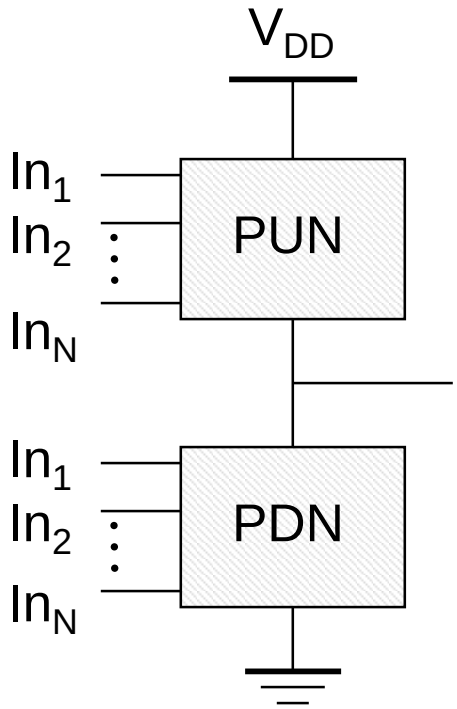
- simpler, faster gates
- increased **sensitivity to noise**

Outline

- **Static CMOS Design**
  - **Complementary CMOS**
  - **Ratioed Logic**
  - **Pass-Transistor Logic**
- **Dynamic CMOS Design**

Static Complementary CMOS

- **Complementary: complementary Pull-up Network (PUN, P-type) *vs.* Pull-down Network (PDN, N-type)**
- **Static: do not rely on stored charge**
- **Simple, effective, reliable; hence ubiquitous**



PMOS transistors only

pull-up: make a connection from V_{DD} to F when $F(In_1,In_2,\ldots In_N) = 1$

pull-down: make a connection from F to GND when $F(In_1,In_2,\ldots In_N) = 0$

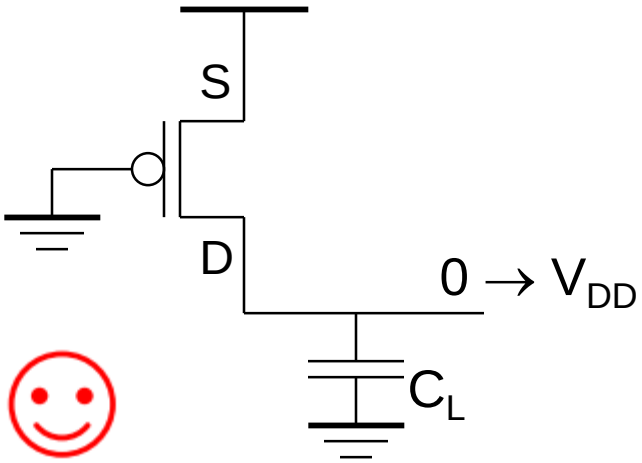
NMOS transistors only

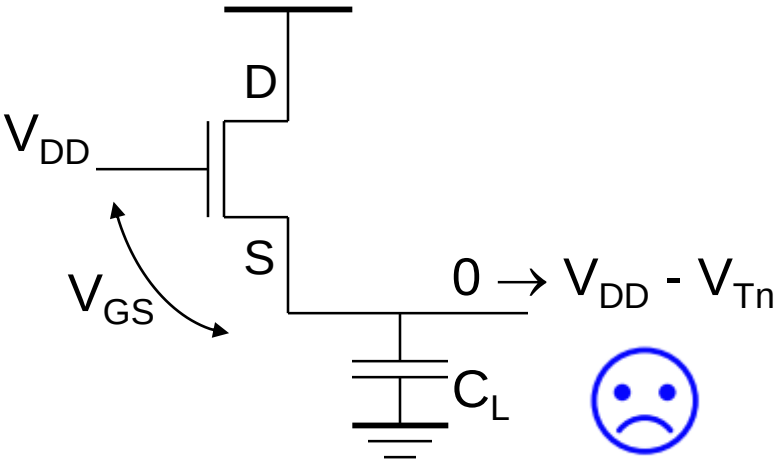
PUN and PDN are dual logic networks

Threshold Drops

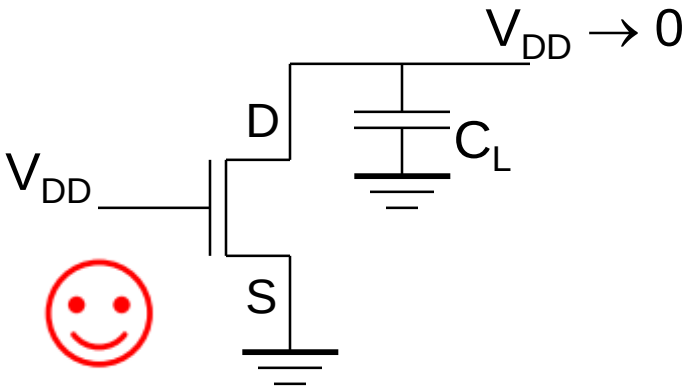
A. Why PMOS in PUN and NMOS in PDN?

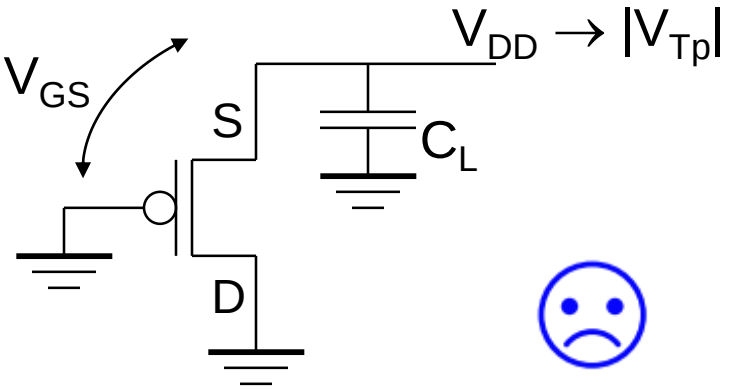
PUN





PDN

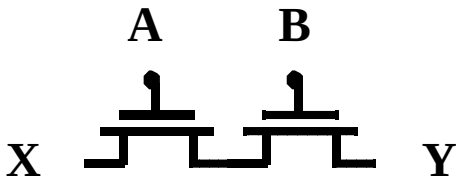




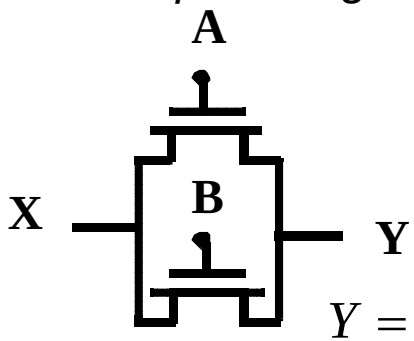
NMOS/PMOS in Series/Parallel Connection

B. Transistors can be thought as a switch controlled by its gate signal

- *NMOS switch closes when switch control input is high*

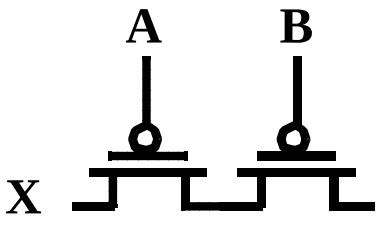


$Y = X$ if A *and* B

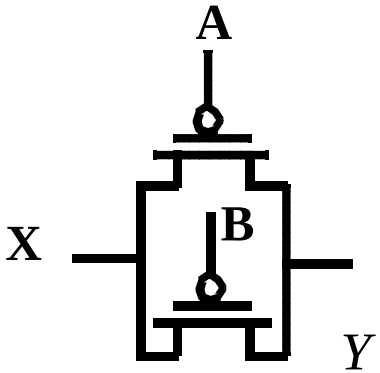


$Y = X$ if A *or* B

- *PMOS switch closes when switch control input is low*



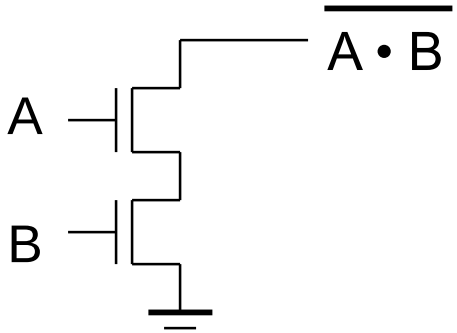
$Y = X$ if $\bar{A}$ *and* $\bar{B} = \overline{A+B}$



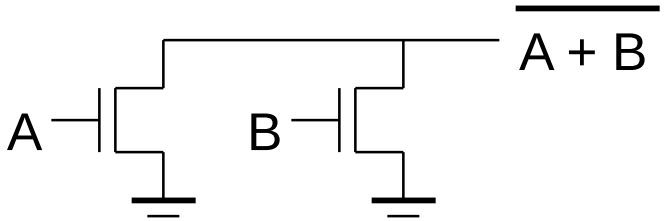
$Y = X$ if $\bar{A}$ or $\bar{B} = \overline{AB}$

Construction of PDN

- NMOS devices in series implement a NAND function



- NMOS devices in parallel implement a NOR function



Dual PUN and PDN

C. PUN and PDN are dual networks 对偶网络

- De Morgan's theorems

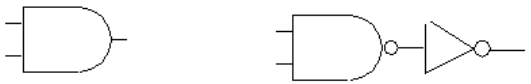
  $\overline{A + B} = \overline{A} \cdot \overline{B}$ [!(A + B) = !A • !B or !(A | B) = !A & !B]

  $\overline{A \cdot B} = \overline{A} + \overline{B}$ [!(A • B) = !A + !B or !(A & B) = !A | !B]

- a parallel connection of transistors in the PUN corresponds to a series connection of the PDN

D. Complementary gate is naturally inverting (NAND, NOR)

{NAND, NOR} is a complete sets; {AND, OR} is not complete.

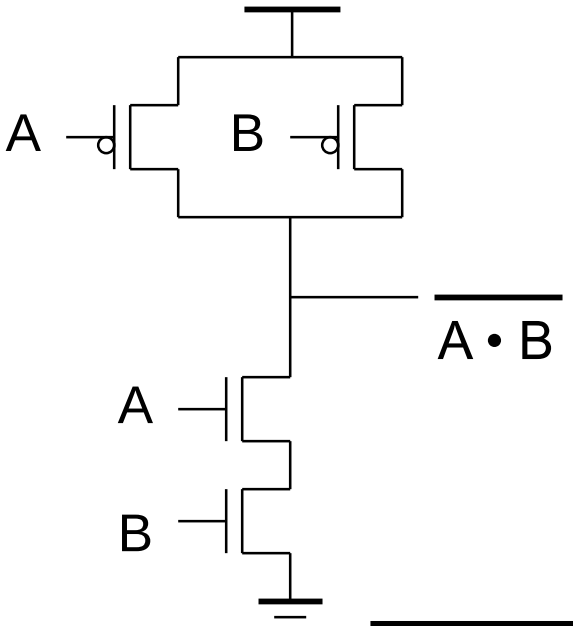


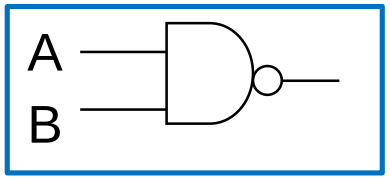
AND = NAND + INV

E. Number of transistors for N-input logic gate: 2N

Dual PUN and PDN: Example

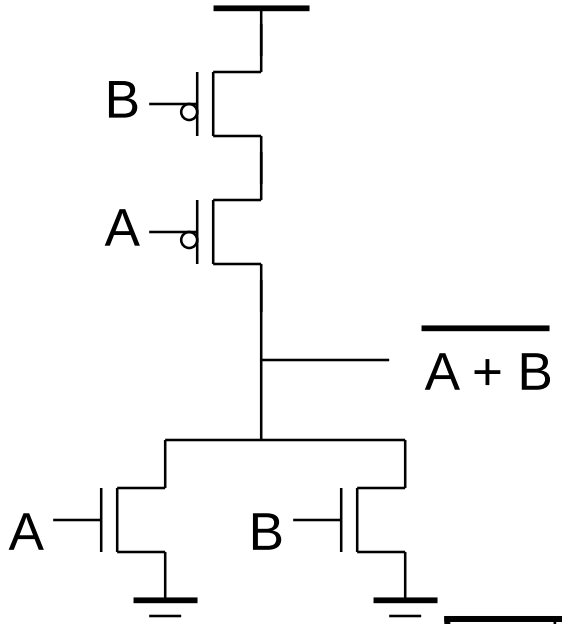
- **CMOS NAND**

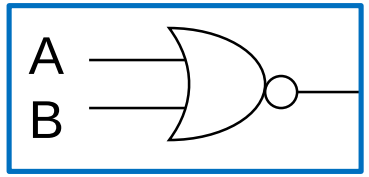




A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

- **CMOS NOR**

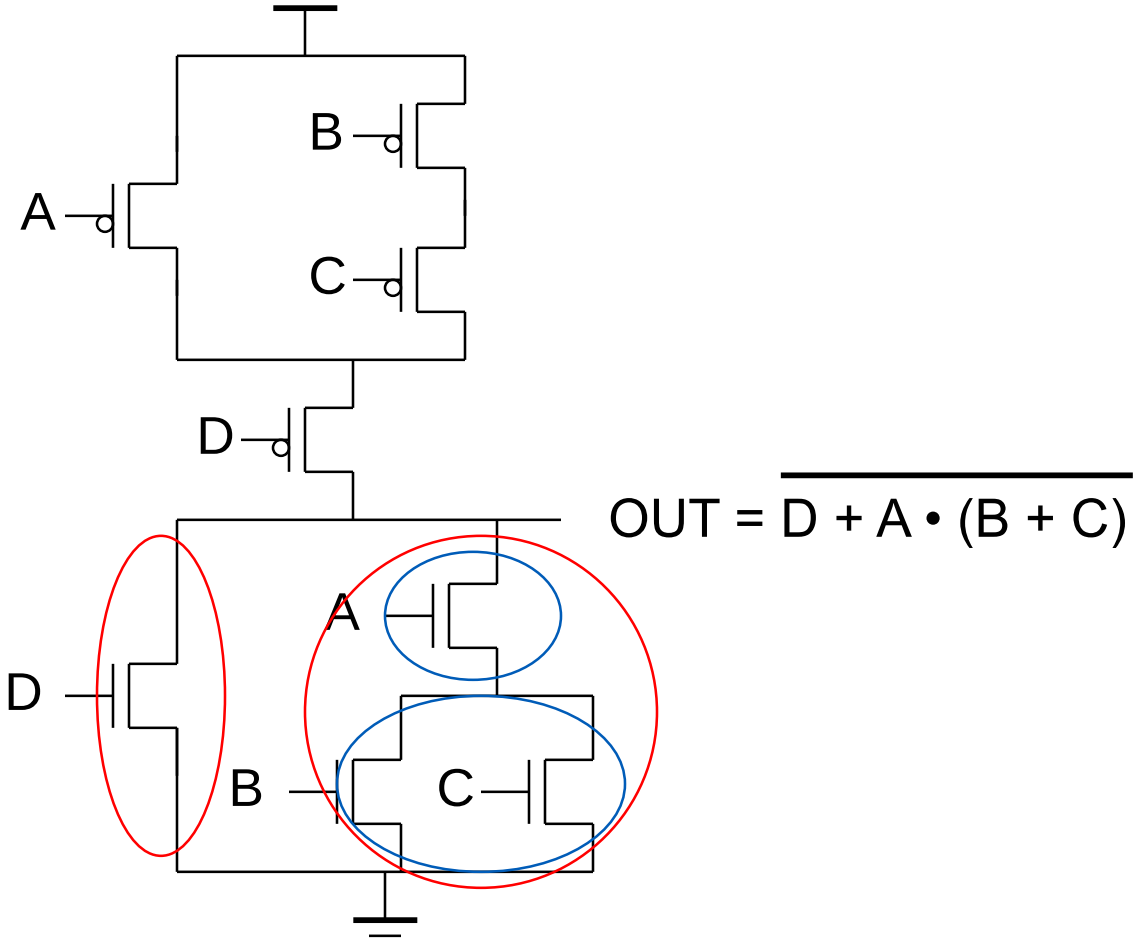




A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

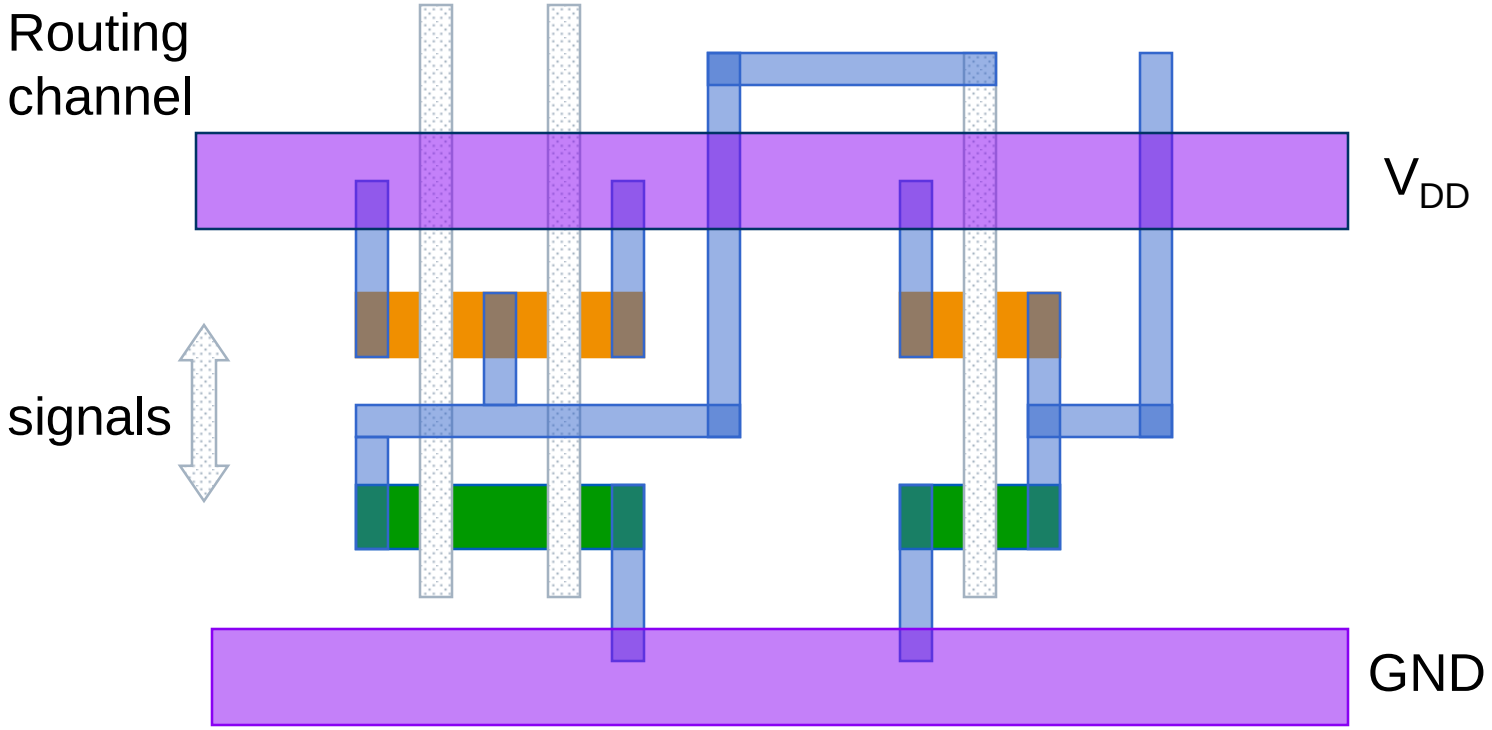
Dual PUN and PDN: Example

- **Complex CMOS Gate**



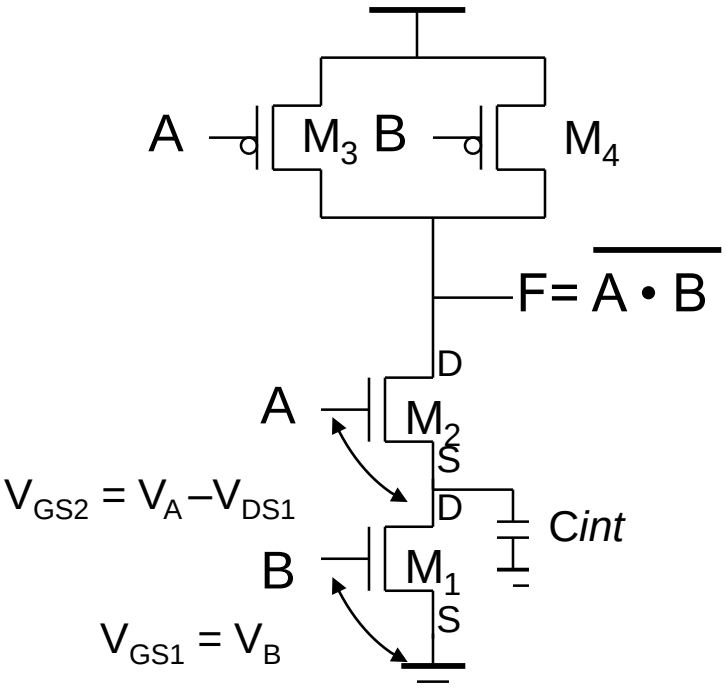
Dual PUN and PDN: Example

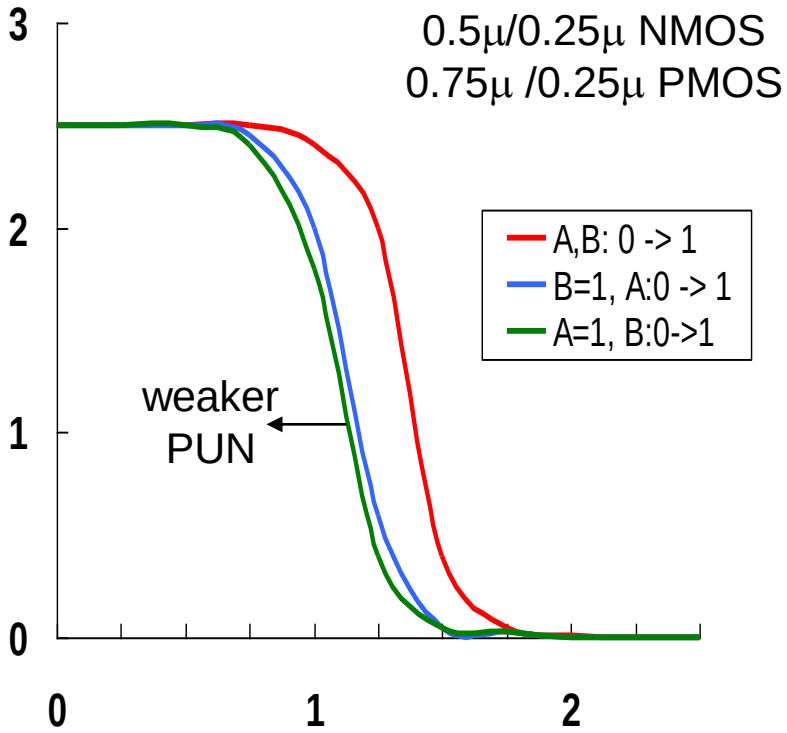
- **CMOS Gate Layout**



What logic function is this?

VTC: Data-Dependent



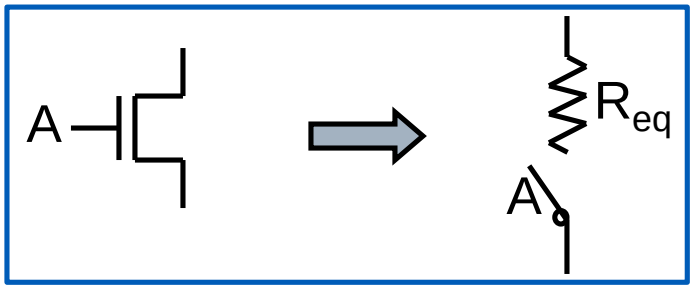


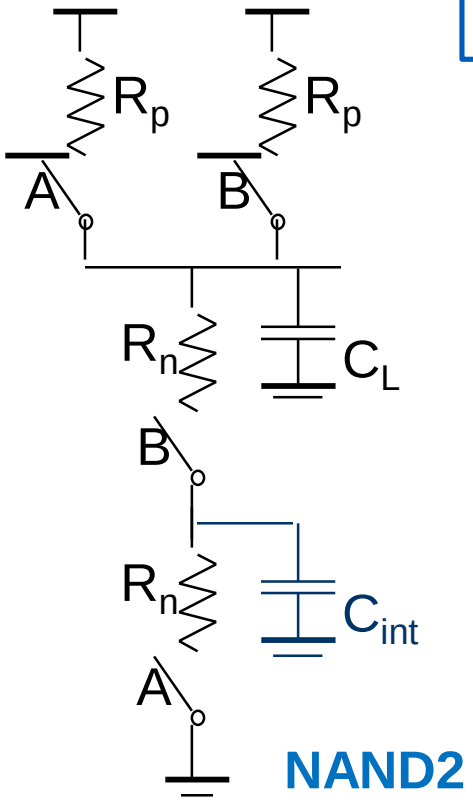
- V_{TH} of M_2 is higher than M_1: the body effect (γ)

$$V_{Tn1} = V_{Tn0} \qquad V_{Tn2} = V_{Tn0} + \gamma(\sqrt{(|2\phi_F| + V_{int})} - \sqrt{|2\phi_F|})$$

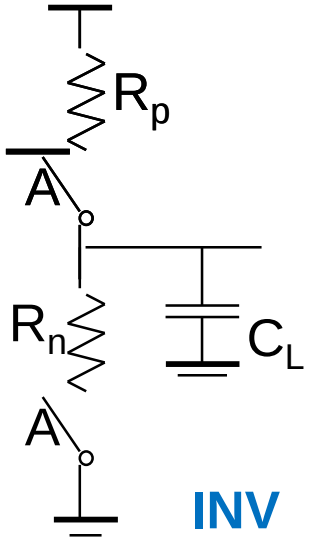
The noise margins are input-pattern dependent.

Switch Delay Model

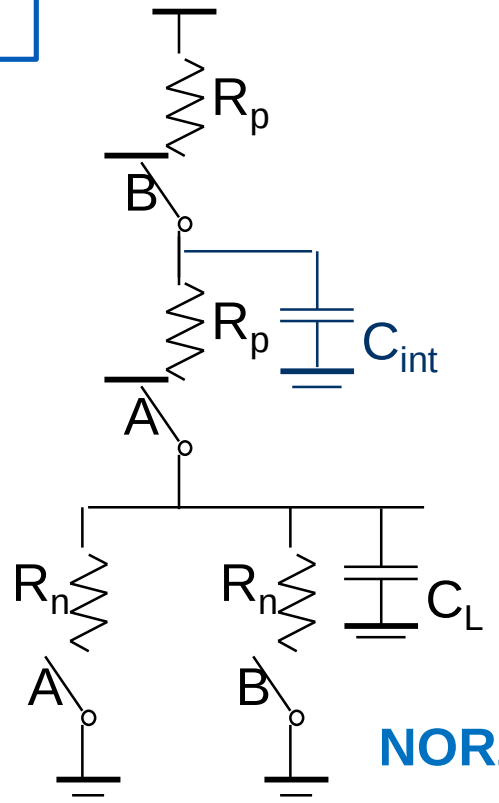




NAND2

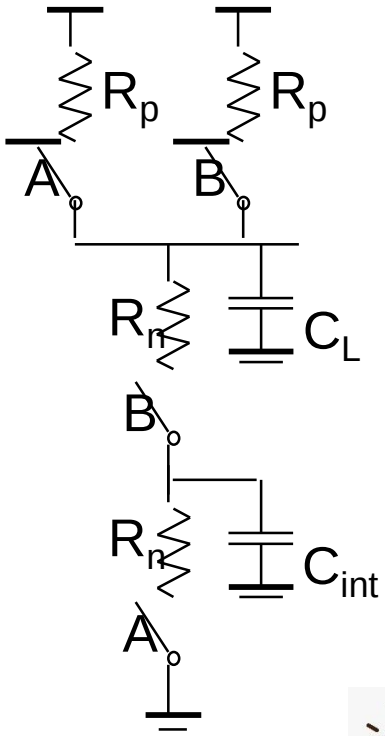


INV



NOR2

Propagation Delay: Input Patterns



NAND2

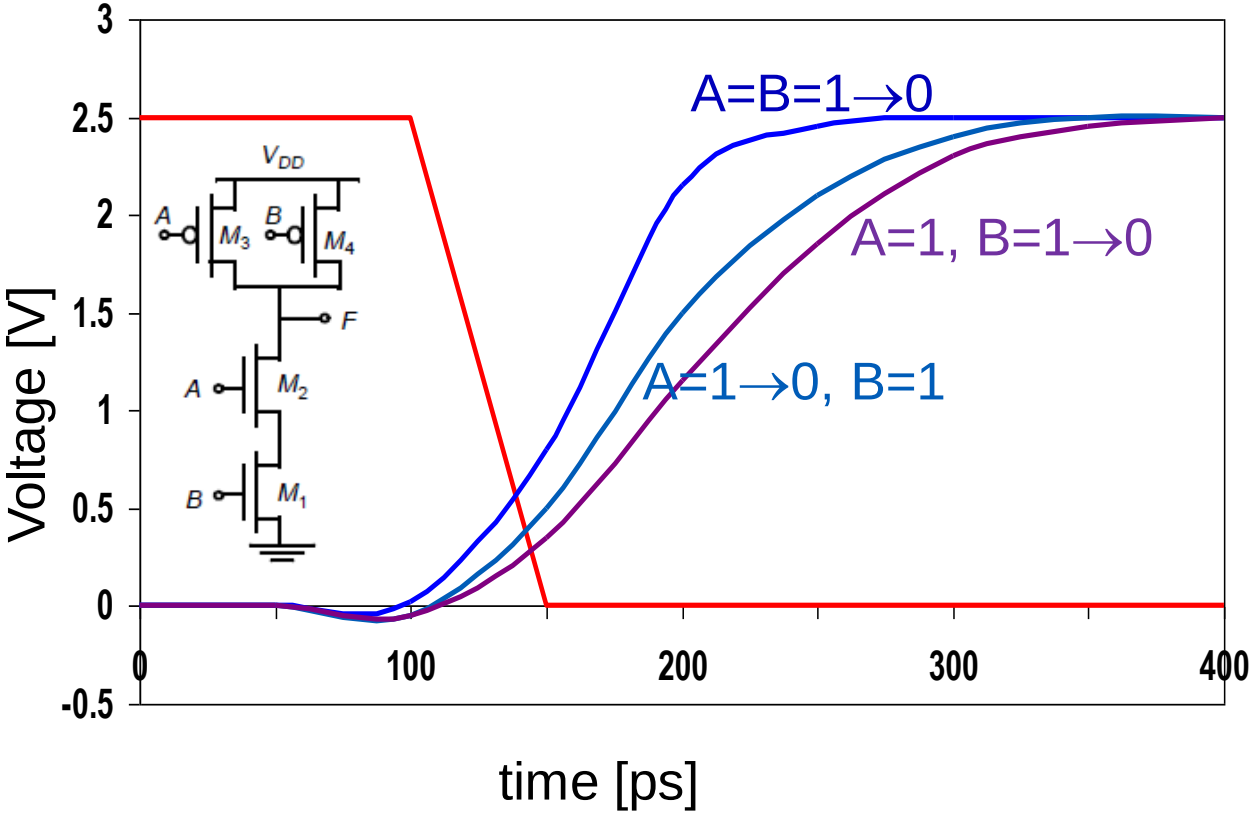
- Delay is dependent on **the input patterns**
- Input: high to low transition
  - both inputs go low: 0.69 $R_p/2$ C_L
  - one input goes low: 0.69 R_p C_L
- Input: low to high transition
  - both inputs go high: 0.69 $2R_n$ C_L

Adding devices in series: slow down

Wider device: avoid a performance penalty

A gate with multi-in's: the combination of input that triggers the worst-case conditions (*resistance in series*)

Delay Dependence on Input Patterns

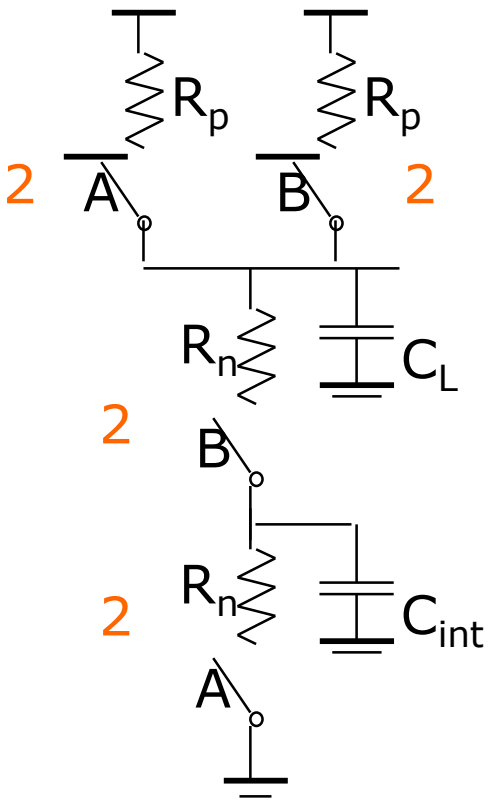


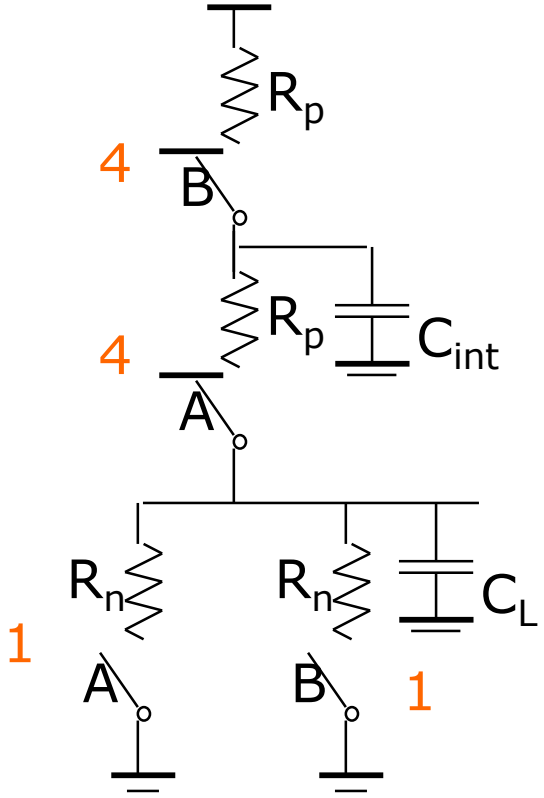
Input Data Pattern	Delay (psec)
A=B=0→1	69
A=1, B=0→1	62
A= 0→1, B=1	50
A=B=1→0	35
A=1, B=1→0	76
A= 1→0, B=1	57

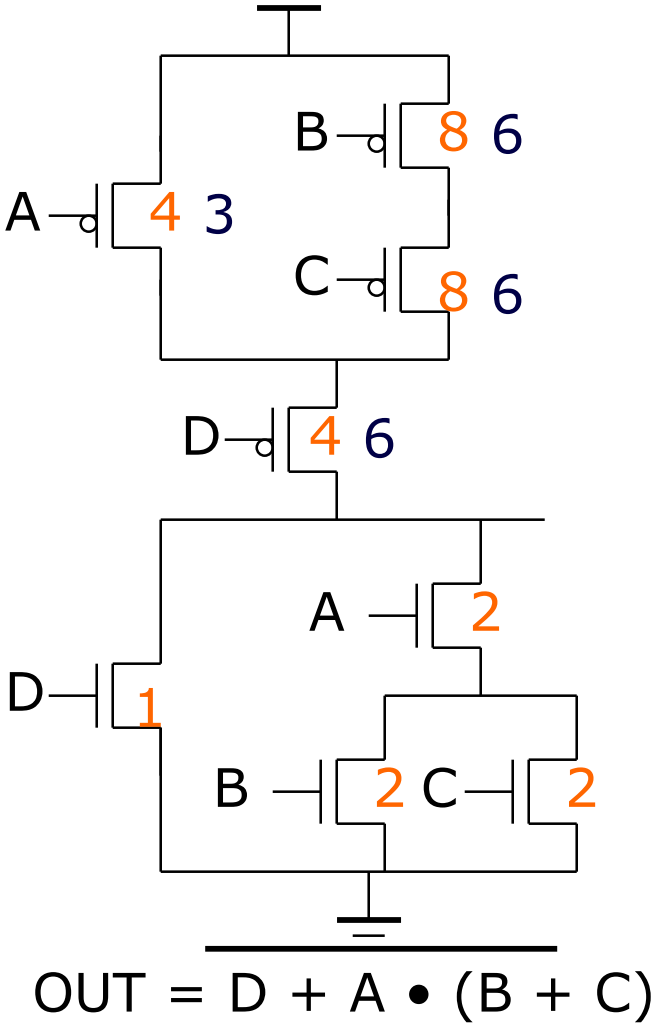
Inverter: NMOS = 0.5μm/0.25 μm
PMOS = 1.5μm/0.25 μm

⇒ NMOS = 1.0μm/0.25 μm
PMOS = 1.5μm/0.25 μm

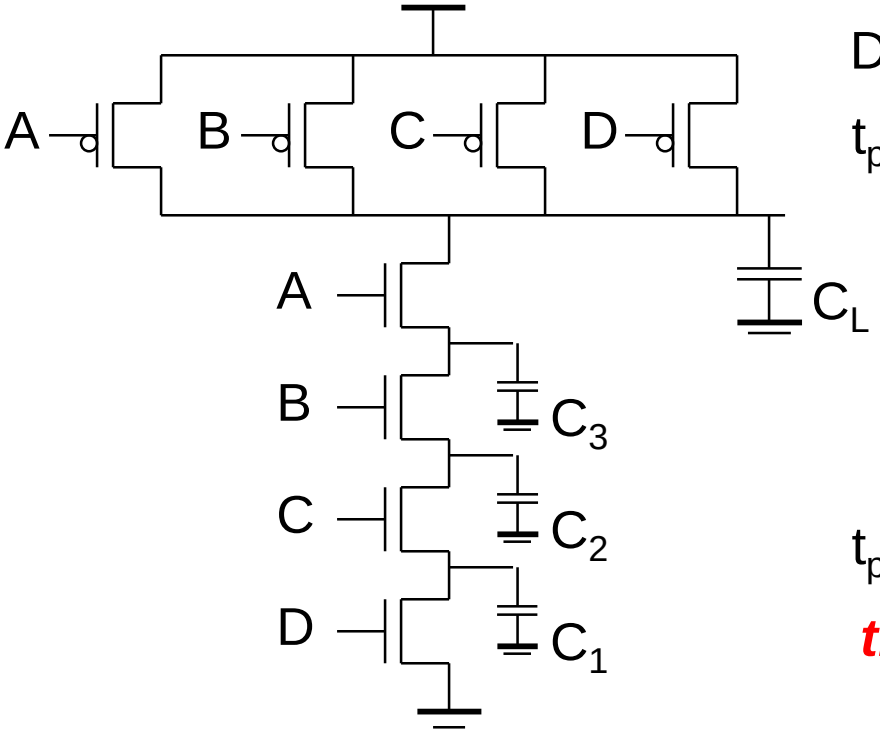
Transistor sizing







Fan-In Considerations



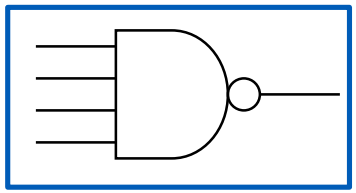
Distributed RC model (Elmore delay)

$$t_{pHL} = 0.69\,[R_1C_1+(R_1+R_2)C_2+(R_1+R_2+R_3)C_3+(R_1+R_2+R_3+R_4)C_L]$$

NMOS: same size R_{eqn}

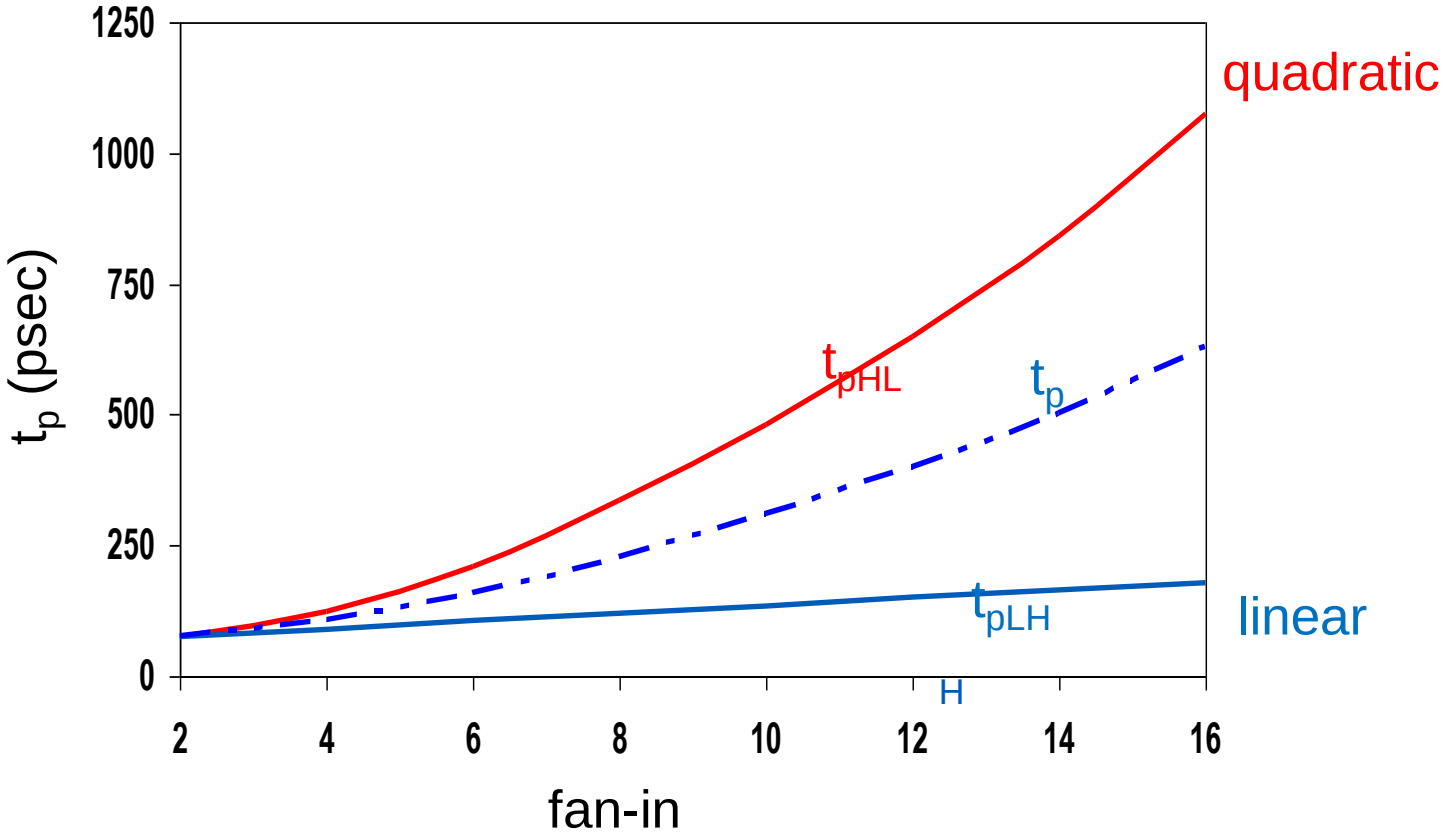
$$t_{pHL} = 0.69\,R_{eqn}(C_1+2C_2+3C_3+4C_L)$$

the worst-case



- The number of transistor required to implement an N fan-in is 2N =>area, capacitance
- Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case. -> 0.5*(1+N)*N

t_p as a Function of Fan-In



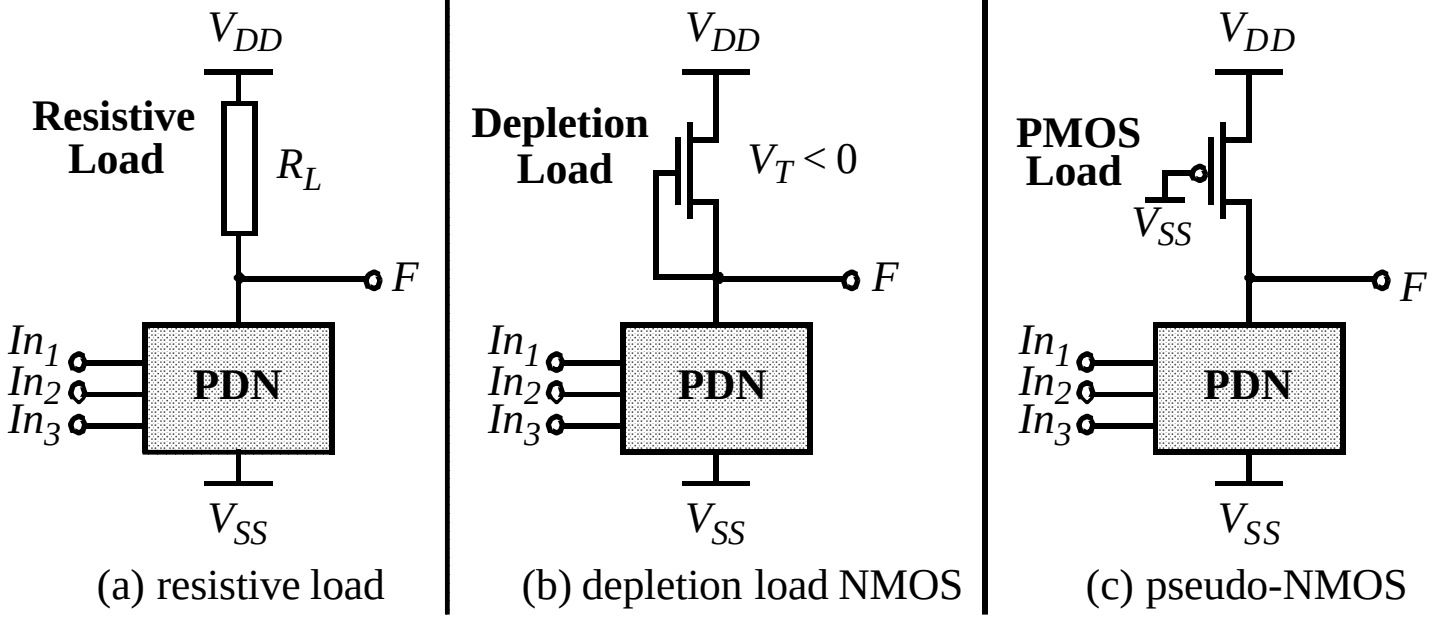
NOTE!: Gates with a fan-in greater than 4 should be avoided.

Outline

- **Static CMOS Design**
  - **Complementary CMOS**
  - **Ratioed Logic**
  - **Pass-Transistor Logic**
- **Dynamic CMOS Design**

Ratioed Logic

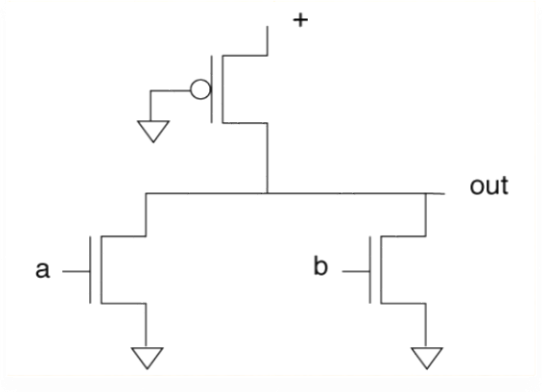
Goal: to reduce the number of devices over complementary CMOS



- *2N -> N+1*
- V_{OH}=*VDD,* V_{OL}*≠0* -》 noise margin ↓, static power ↑
- Tadeoff*: the size of the load relative to the PDN* vs. *NM, delay, power*
- **Ratioed Logic**
- **Pseudo-nMOS**

Pseudo-NMOS: 2-input NOR

- Uses a p-type as a resistive pullup, n-type network for PDN.



- Consumes static **power**.
- Much **smaller** PUN network than static gate.
- Pull-up time is longer.
- Wiring is simple.

- V_{OH}=*VDD*,
  V_{OL}≠>0 (Vss): V_{OL} = 0.2 (V_{DD} - V_{SS}) is one plausible choice.
- For logic 0 output, PUN and PDN form a **voltage divider**.
  Must choose N/P transistor sizes to create R_{eq} of the required ratio.
- Effective resistance of PDN must be computed in worst case:
  series n-types means larger transistors.

Transistor ratio calculation

- **In steady state logic 0 output:**

  PUN: *saturation* region $V_{ds}>V_{gs}-V_t$; PDN: *linear region* $V_{ds}<V_{gs}-V_t$.

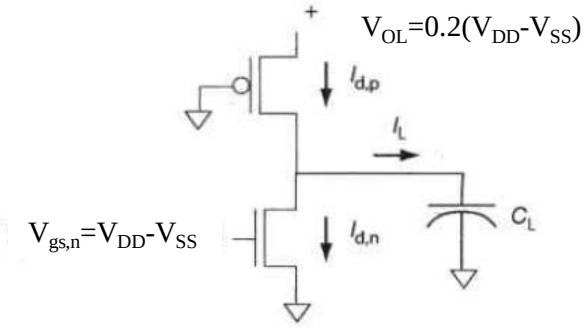
  PUN and PDN: $I_{dp} = I_{dn}$.

- **Equate two currents while generate V_{OL}:**

  $I_{dp}=1/2k_p'(W_p/L_p)(V_{DD}-V_t)^2$;

  $I_{dn}=1/2k_n'(W_n/L_n)[2(V_{DD}-V_t)V_{OL}-V_{OL}^2]$;

  $$\Rightarrow V_{OL} = (V_{DD} - V_t)(1 - \sqrt{1 - \frac{k_p'(W_p / L_p)}{k_n'(W_n / L_n)}})$$



V_{OL} is $0.2V_{DD}$, V_t, k_p' and k_n' is dependent on technology.

Using 0.18 um parameters, VDD=1.8V, $V_t \approx 0.4\sim0.5V$, $k_p'/k_n' \approx 0.18$

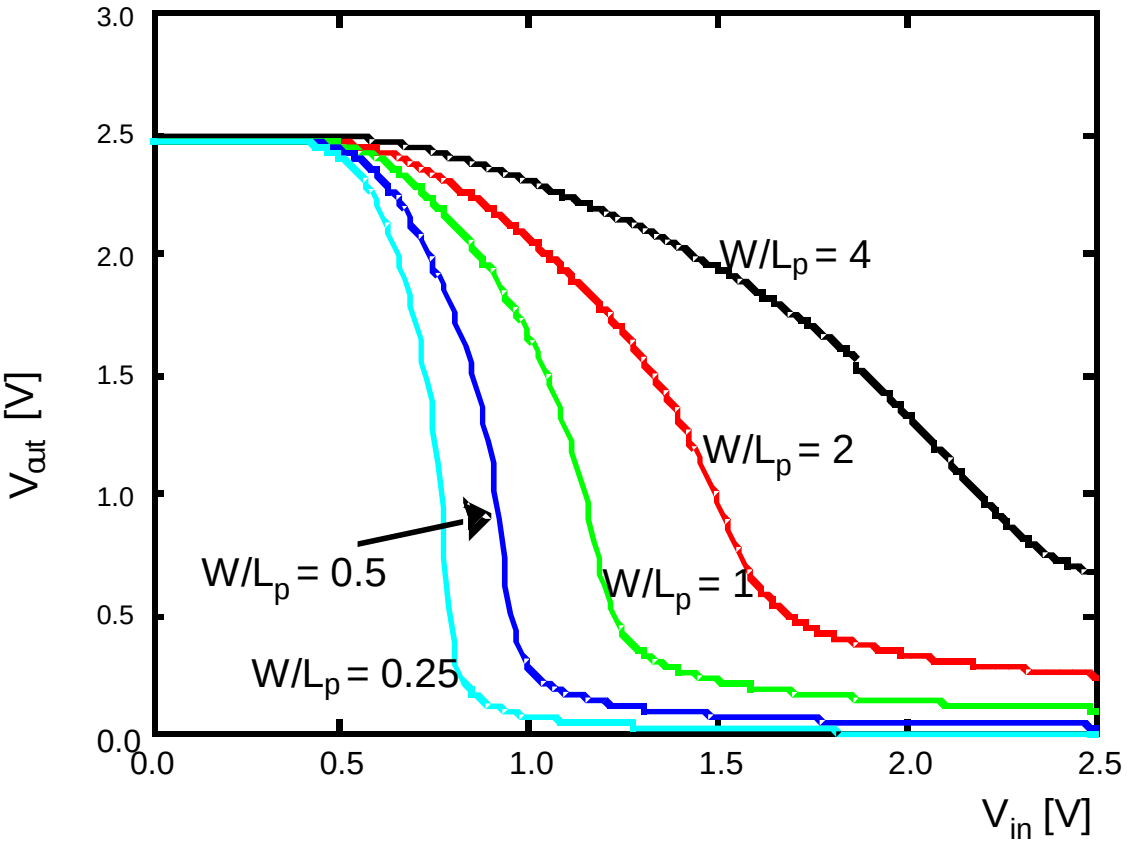
=> $W_p/L_p=2.5W_n/L_n$ (or $L_p/W_p=0.4L_n/W_n$)

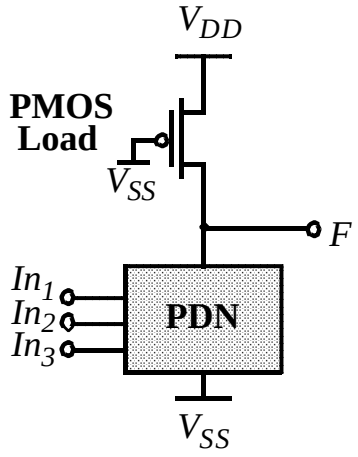
since equivalent resistance of MOS transistor $R_t=(L/W)R_{N,P}$, $R_N=0.2R_P$,

=> equivalent $R_{tp}=R_P\,(Lp/Wp)= 5\,R_N\,0.4L_n/W_n =2R_{tn}$, $\tau=R_tC_L$,

=> Pullup time is 2 times longer than pulldown for pseudo-NMOS.

Pseudo-NMOS VTC





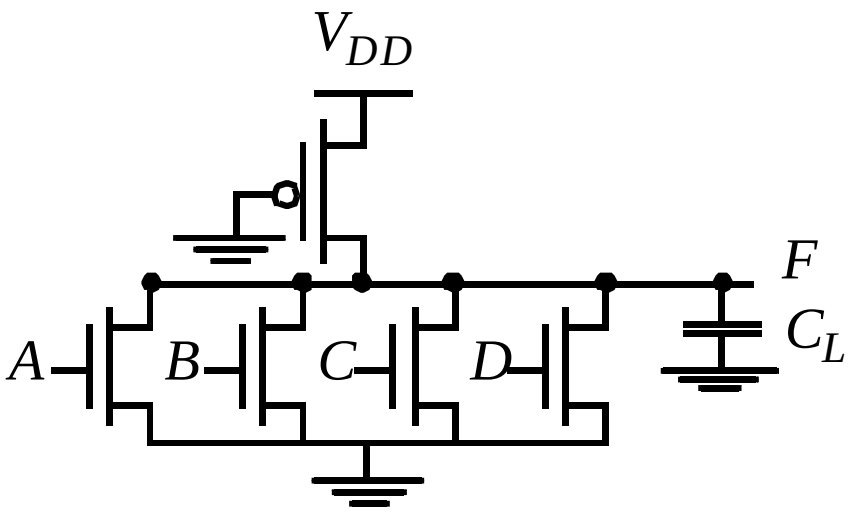
$$V_{OL} \approx \frac{\mu_p \cdot W_p}{\mu_n \cdot W_n} V_{DSATp}$$

- $V_{OL}\downarrow$ -> $(W/L)_P < (W/L)_P$
- Static power

Size	V_{OL}(V)	Power(μW)	t_{plh}(ps)
4	0.693	564	14
2	0.273	295	56
1	0.133	160	123
0.5	0.064	80	268
0.25	0.031	41	569

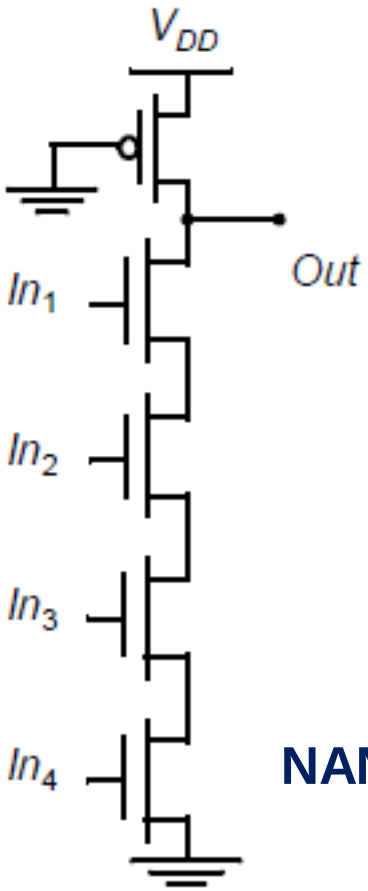
Pseudo-NMOS: 4-input NOR and NAND

- Small area: *2N -> N+1*



$V_{OH} = V_{DD}$ ***(similar to complementary CMOS)***

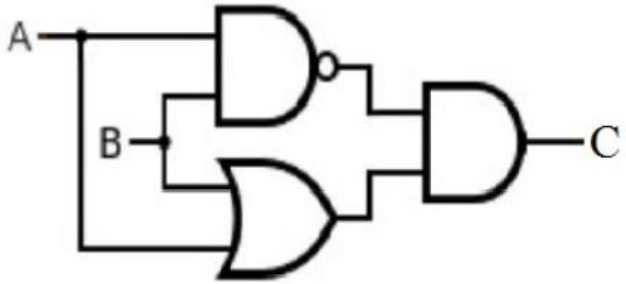
NOR



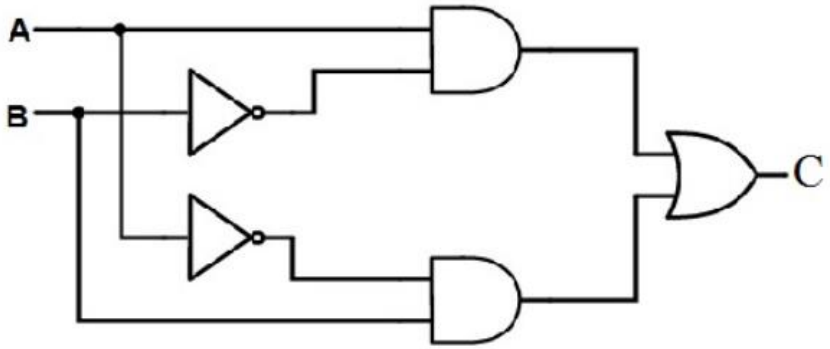
NAND

XOR gates

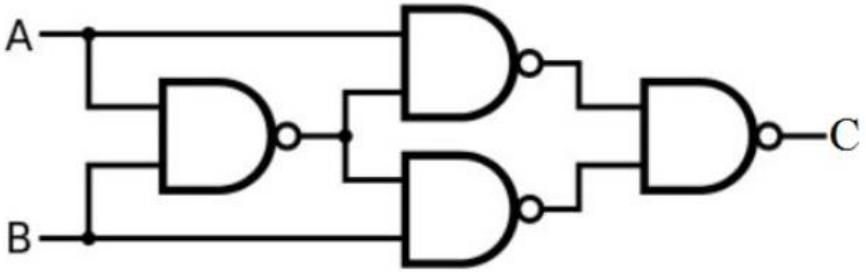
- Can be implemented with different gates

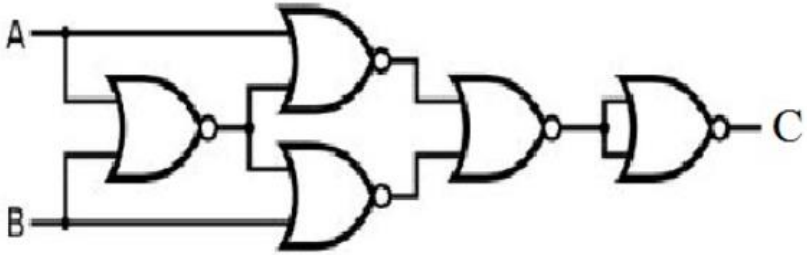


$(A + B) \cdot (\overline{A} + \overline{B})$



$A \cdot \overline{B} + \overline{A} \cdot B$





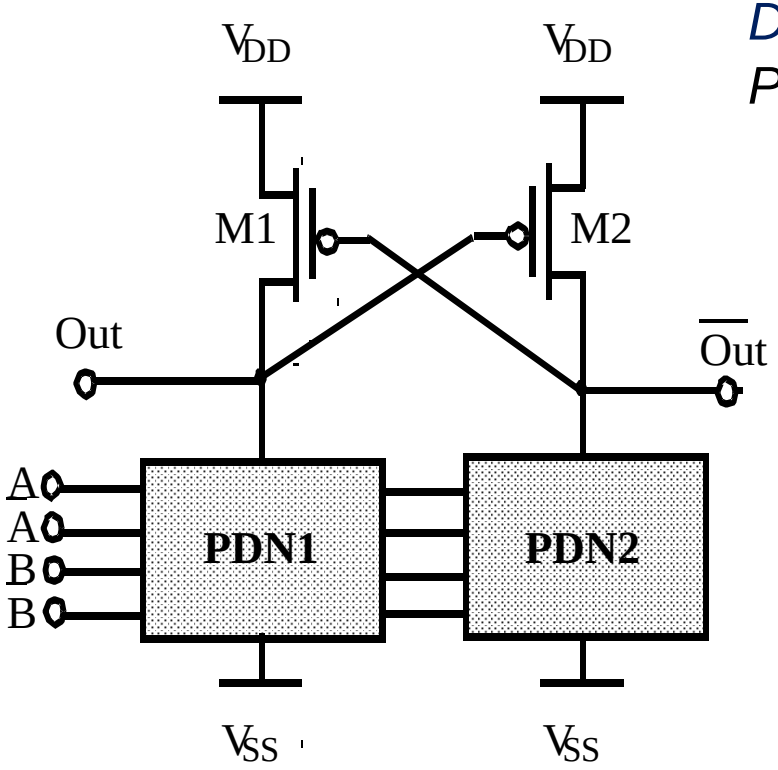
Improved Loads - DCVSL

- Differential Cascode Voltage Switch Logic (DCVSL)

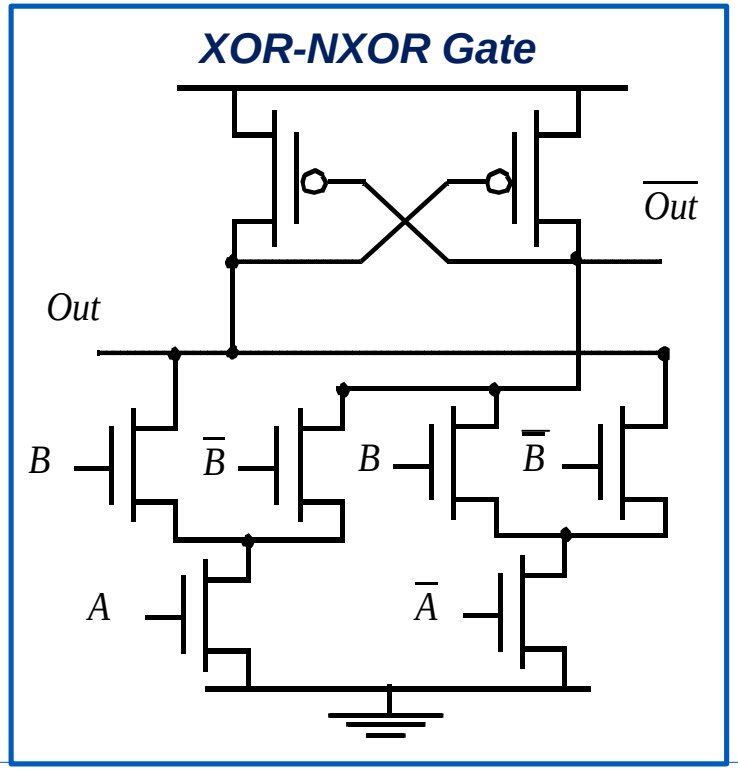
Differential logic +Positive feedback

PDN1 and PDN2: 1) NMOS

2)exclusive

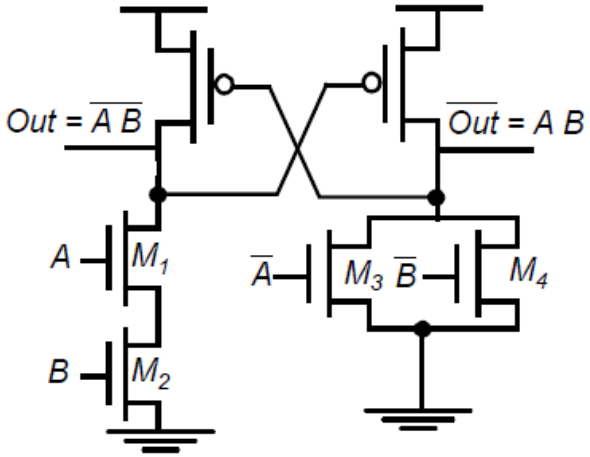


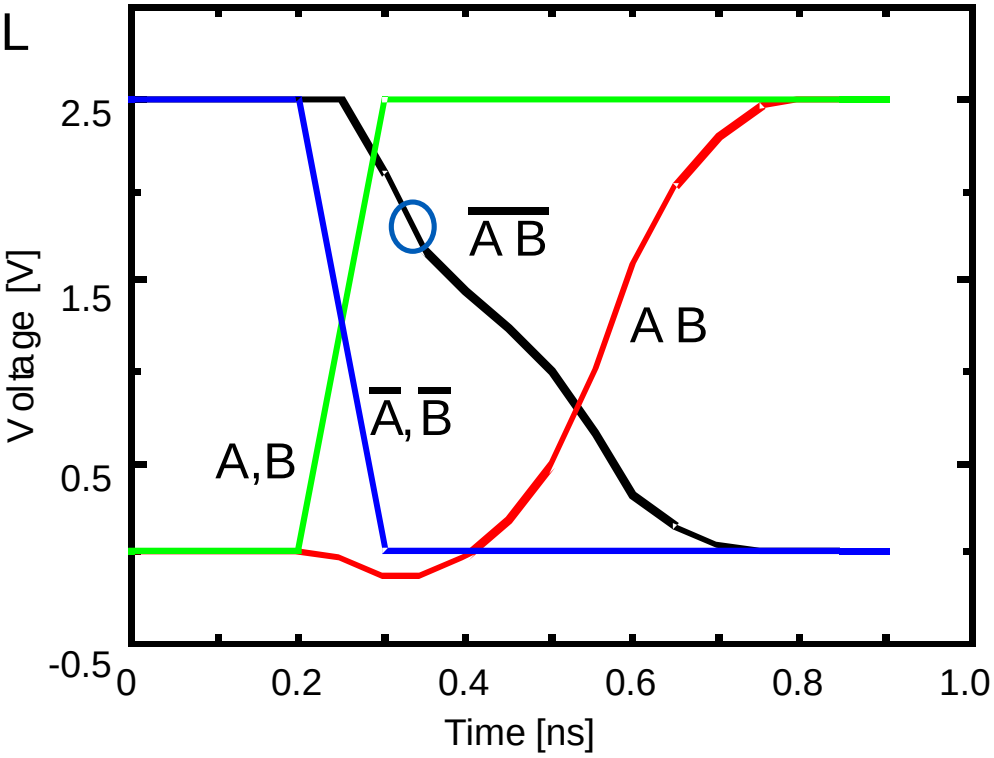
-- Differential Output



DCVSL Transient Response: AND/NAND

- AND/NAND gate in DCVSL





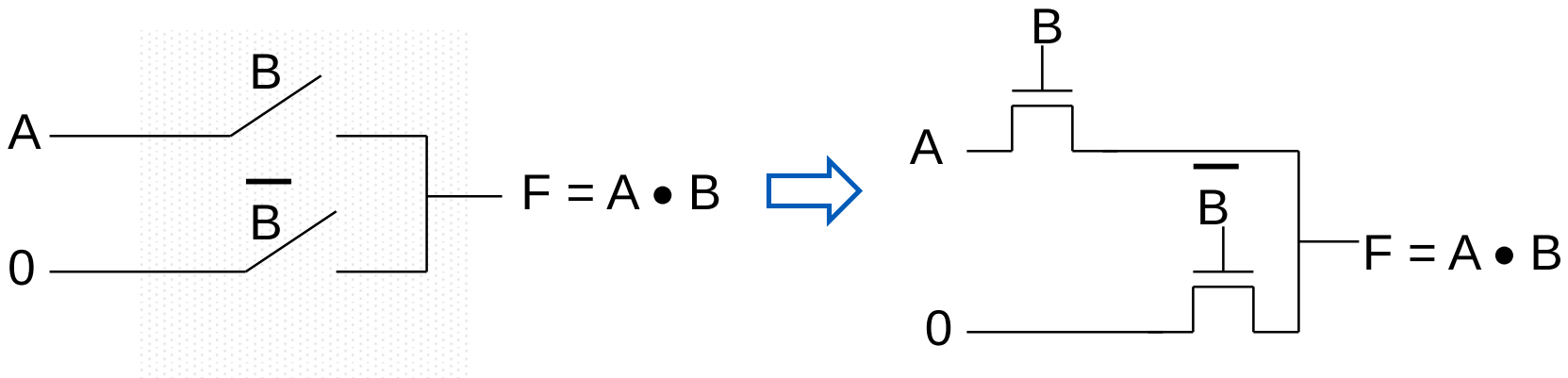
NOTE:

- As *Out* is pulled down to *VDD*-|*VTp*|, !*Out* starts to charge up to *VDD* quickly.
- The delay from the input to *Out* is 197 psec and to !*Out* is 321 psec.

Outline

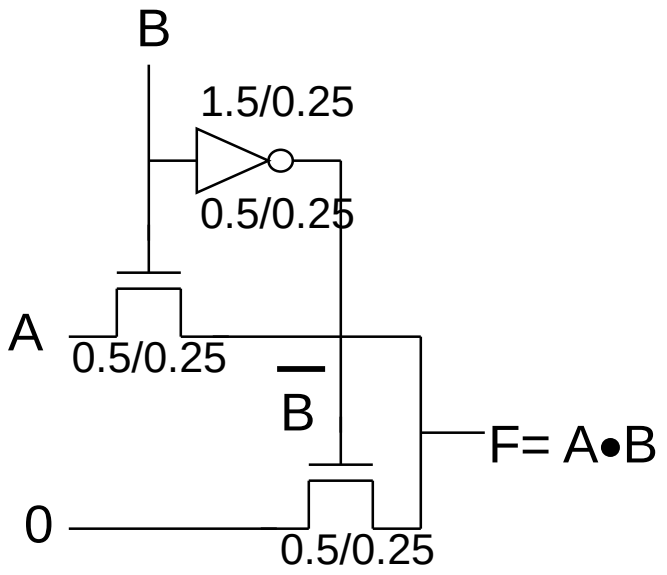
- **Static CMOS Design**
  - **Complementary CMOS**
  - **Ratioed Logic**
  - **Pass-Transistor Logic**
- **Dynamic CMOS Design**

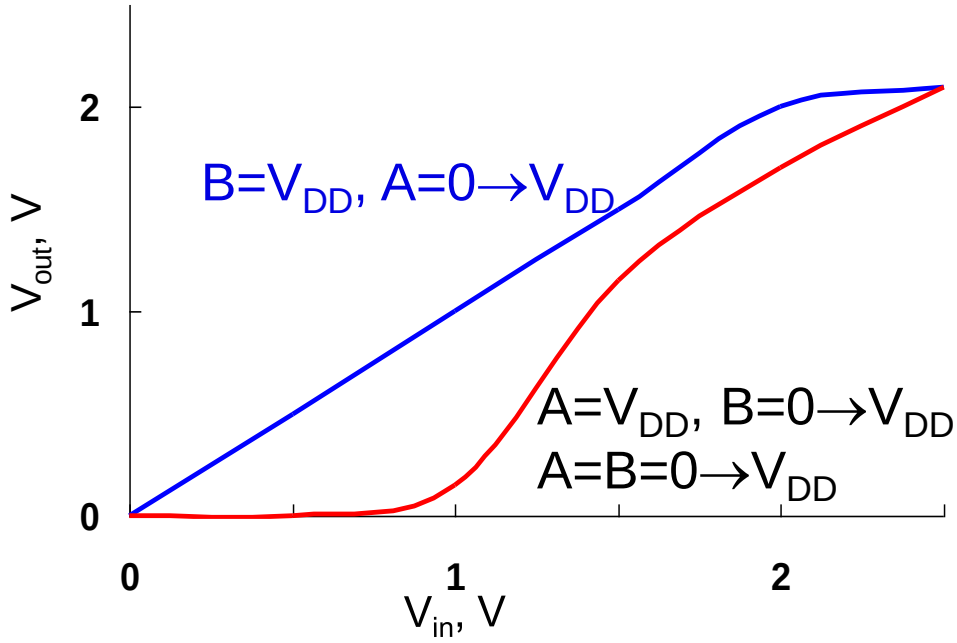
Pass-Transistor (PT) Logic



- Gate is static - a low-impedance path exists to both supply rails under all circumstances
- N transistors instead of 2N
- No static power consumption
- Ratioless

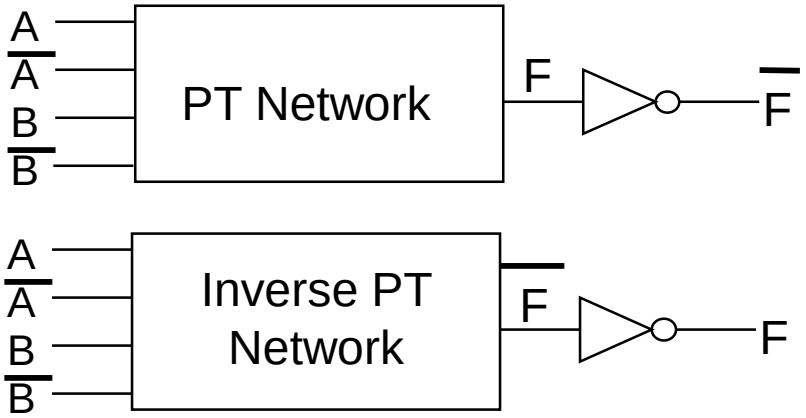
Pass-Transisitor AND Gate: VTC





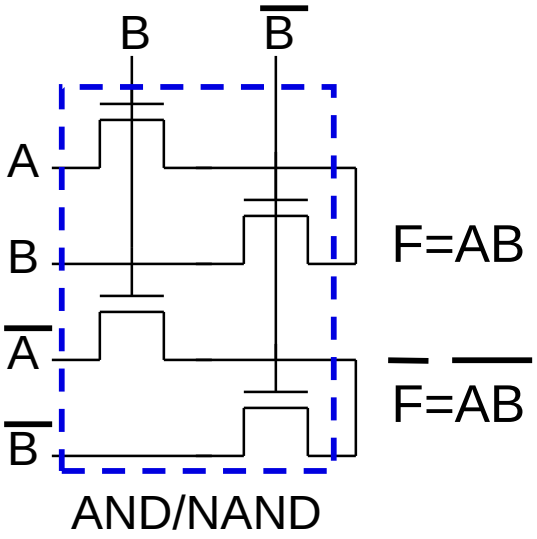
- Be data-dependent
- Pure PT logic is not regenerative - the signal gradually **degrades** after passing through a number of PTs (can fix with **static CMOS inverter** insertion)

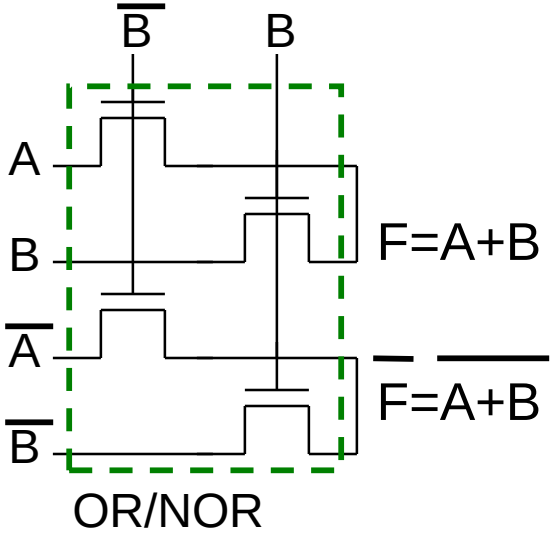
Differential PT Logic (CPL/DPL)

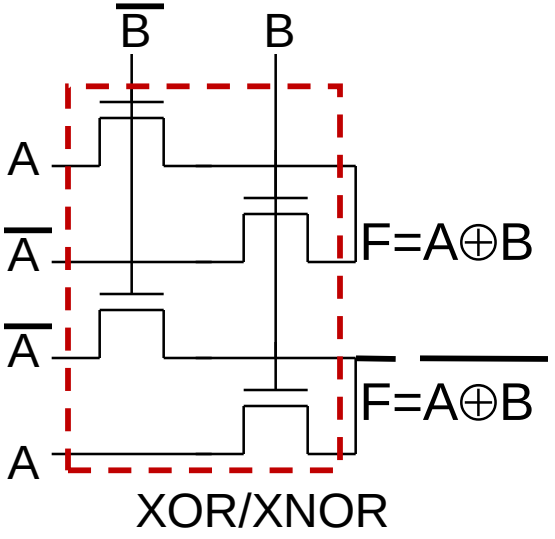


CPL：Complementary Pass-transistor Logic

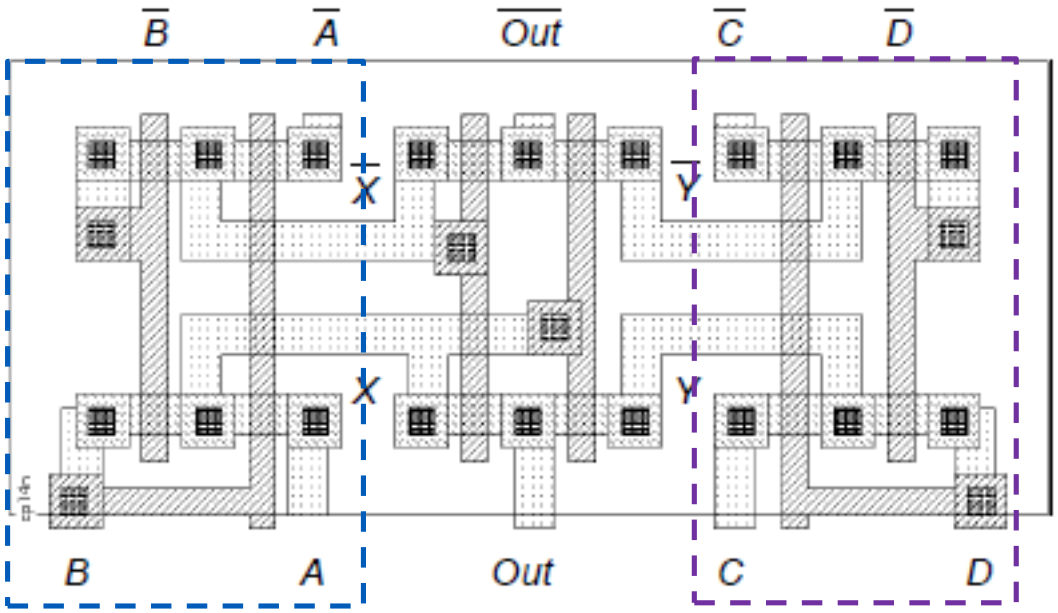
- Complementary data input/ output
- Static gates
- Modular

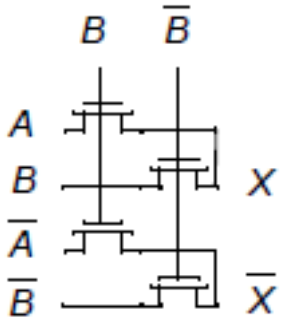


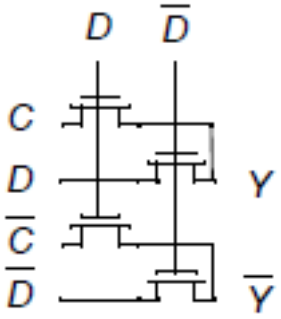


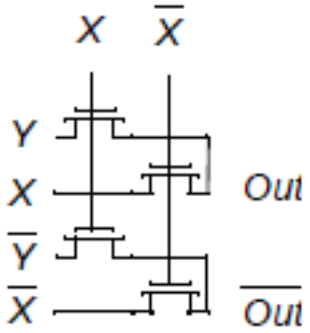


Four-input NAND in CPL

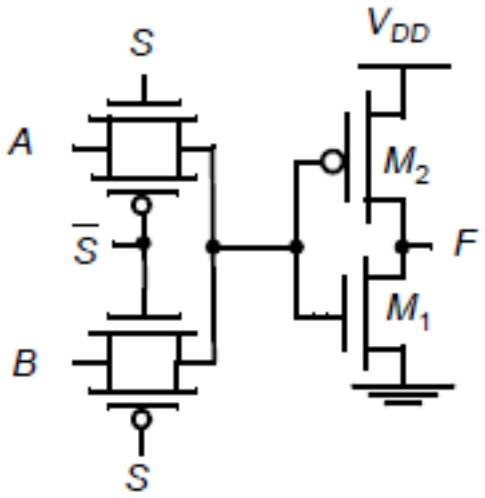




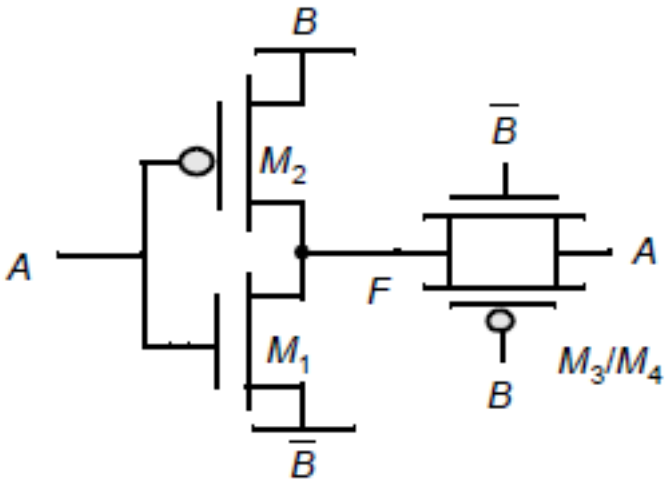




Transmission Gate: Example



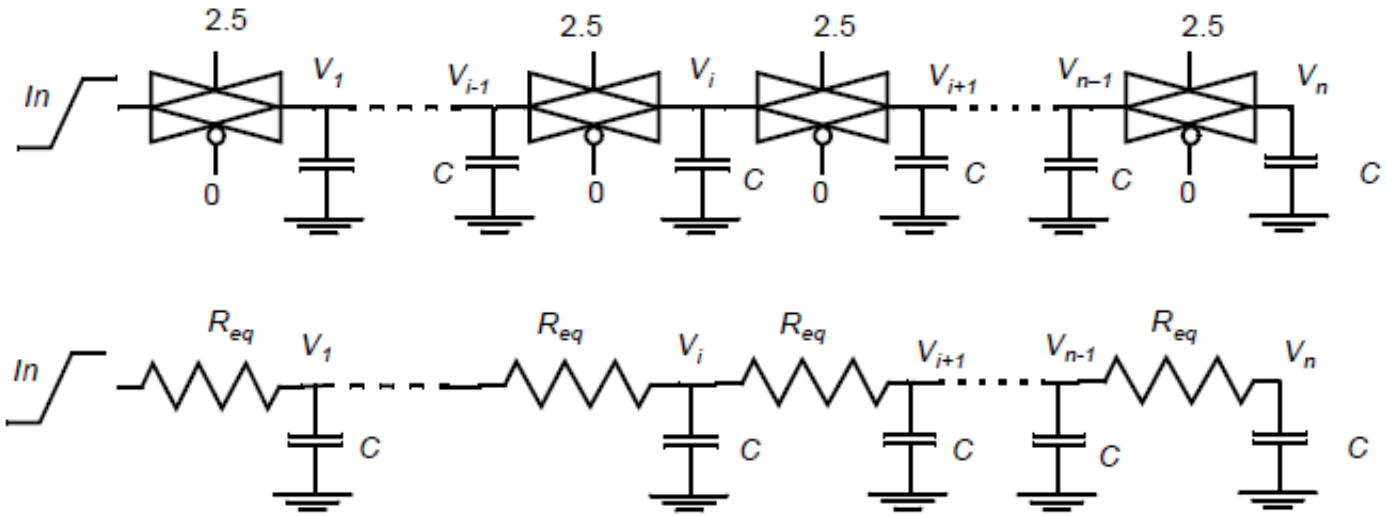
Transmission gate: Multiplexer



XOR

$$\bar{\bar{F}} = (A \cdot S + B \cdot \bar{S})$$

Delay in Transmission Gate Networks



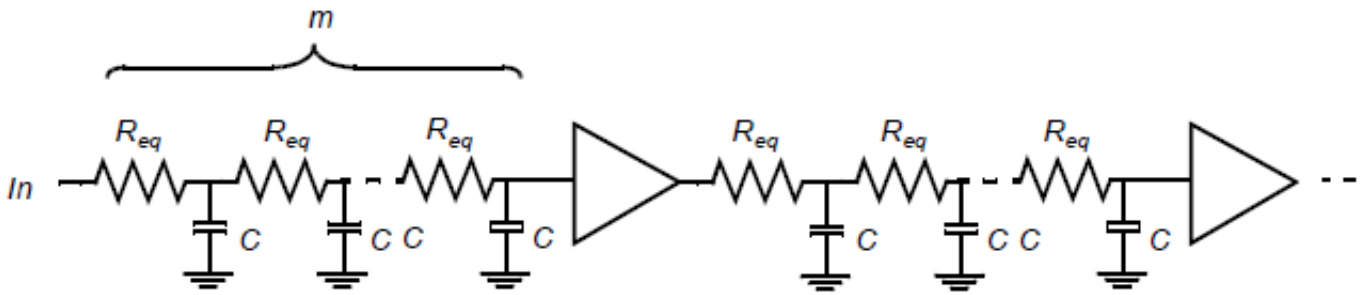
- Delay of RC Chain

$$t_p(V_n) = 0.69 \sum_{k=0}^{n} CR_{eq}k = 0.69CR_{eq}\frac{n(n+1)}{2}$$

The propagation delay is proportional to n^2 and increased rapidly with the switches in the chain.

- Delay optimization: break the chain and insert buffers every *m* switches

TG Networks : Delay Optimization



- Delay of Buffered Chain

$$t_p = 0.69\left[\frac{n}{m}CR_{eq}\frac{m(m+1)}{2}\right] + \left(\frac{n}{m} - 1\right)t_{buf}$$

$$= 0.69\left[CR_{eq}\frac{n(m+1)}{2}\right] + \left(\frac{n}{m} - 1\right)t_{buf} \Rightarrow m_{opt} = 1.7\sqrt{\frac{t_{pbuf}}{CR_{eq}}}$$

***m*=3 or 4**

The number of switches per segment grows with increasing values of t_{buf}.

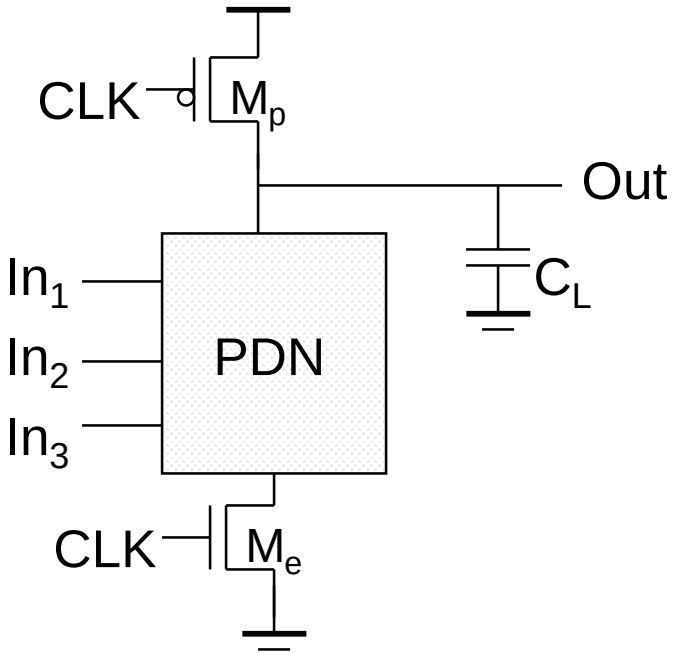
Outline

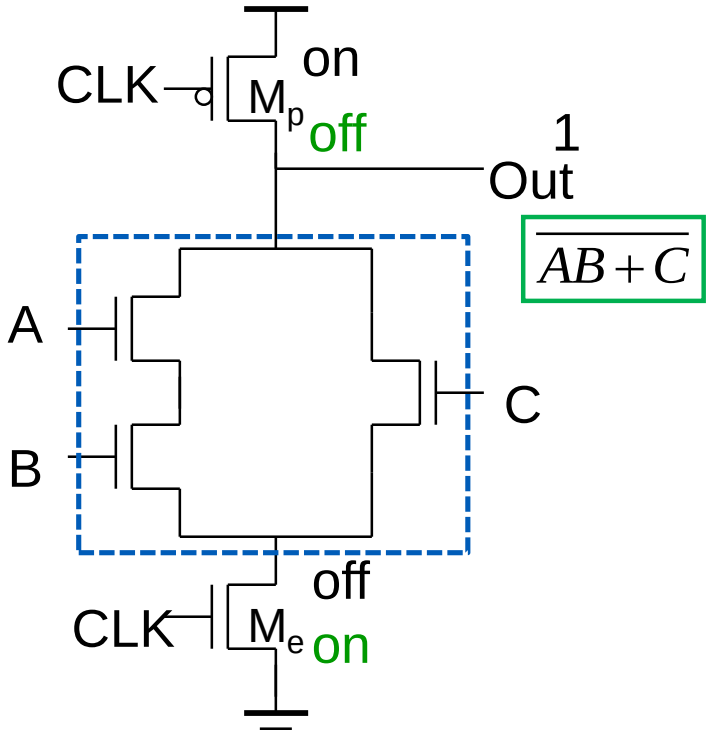
- Static CMOS Design
- Dynamic CMOS Design
  - Dynamic Logic: Basic Principles
  - Speed and Power Dissipation of Dynamic Logic
  - Cascading Dynamic Gates

Dynamic CMOS

- In static circuits at every point in time (except when switching) the output is connected to either GND or V_{DD} via a low resistance path.
  - Complementary: fan-in of N requires 2N devices
  - Pseudo-NMOS: fan-in of N requires N+1 devices
- Dynamic circuits rely on the **temporary storage** of signal values on the **capacitance** of high impedance nodes.
  - requires only N+2 transistors
  - takes a sequence: precharge phase, conditional evaluation phases

Dynamic Gate





Two phase operation:

-- Precharge (CLK = 0)

-- Evaluate (CLK = 1)

$$Out = \overline{CLK} + \overline{(AB + C)} \cdot CLK$$

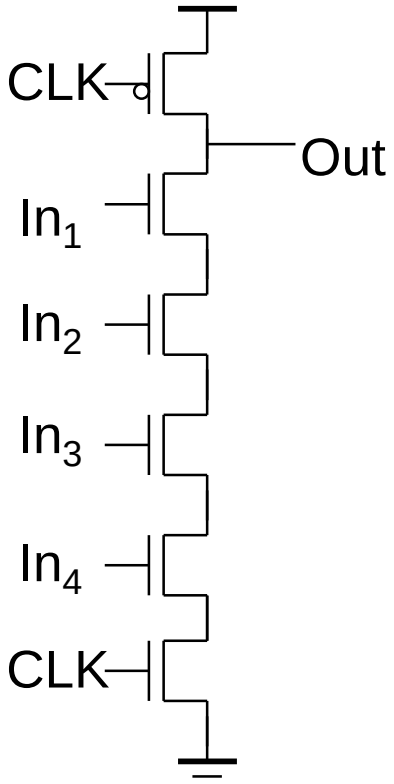
Conditions on Output

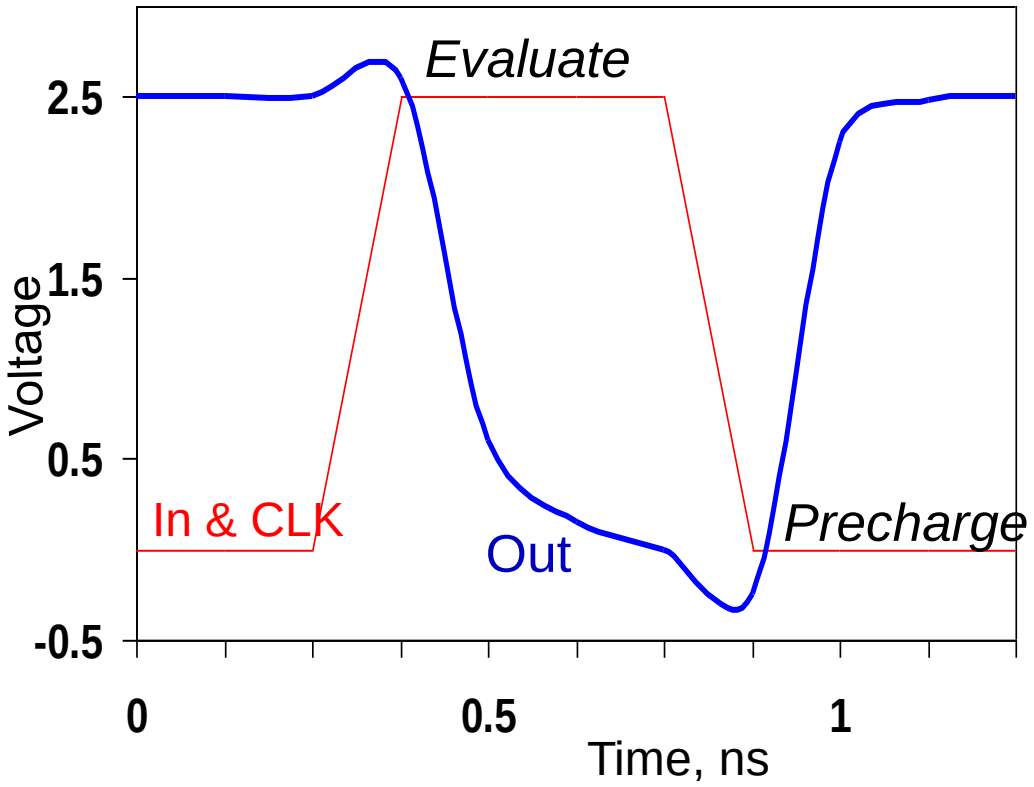
- Once the output of a dynamic gate is discharged, it **cannot be charged again** until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.
- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on C_L

Properties of Dynamic Gates

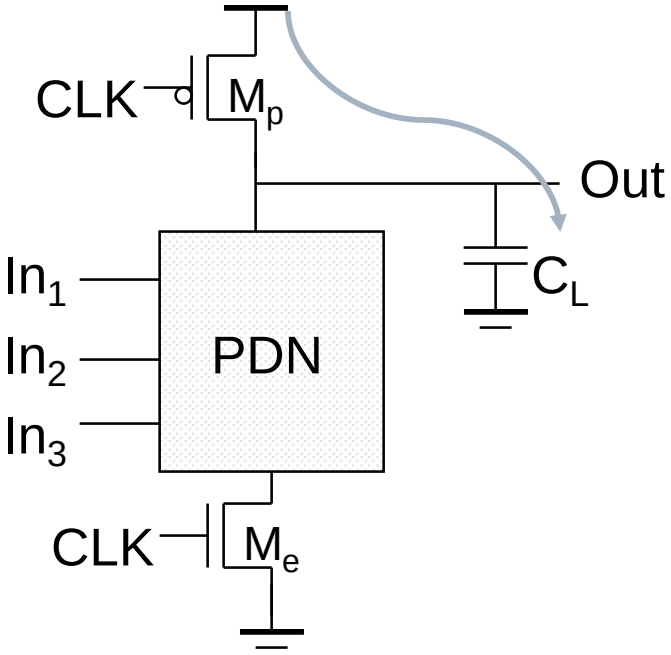
- Logic function is implemented by the PDN only
  - transistors number : **N+2** (vs. static complementary CMOS : 2N)
  - be **smaller** in area
- Full swing outputs: V_{OL}=GND, V_{OH}=V_{DD}
- Nonratioed - sizing of PMOS is not important for proper functioning (only for performance)
- Faster switching speeds *(charge leakage)*
- Power dissipation should be better
  - only dynamic power – no short circuit power consumption since the pull-up path is not on when evaluating
  - lower C_L- C_{int} (fewer transistors connected to the drain output)
    - C_{ext} (output load is one per connected gate, not two)
  - by construction can have at most one transition per cycle – no glitching
- Needs a precharge clock

Dynamic Behavior



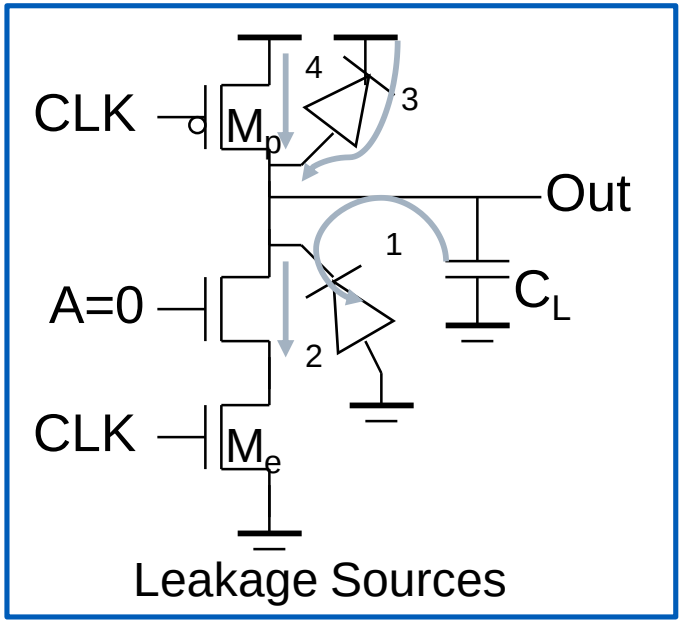


Power Consumption

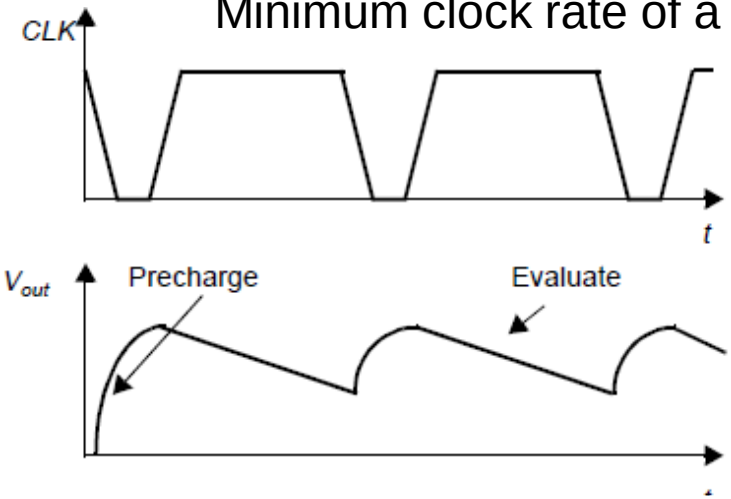


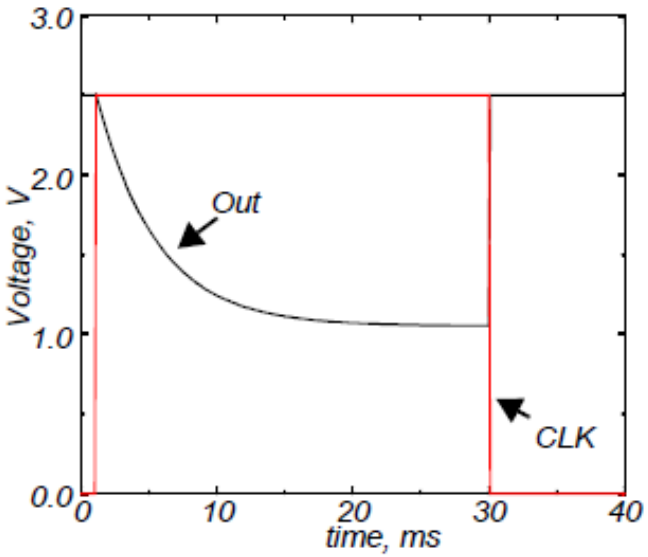
Power only dissipated
when previous Out = 0

Issues in Dynamic Design: Charge Leakage



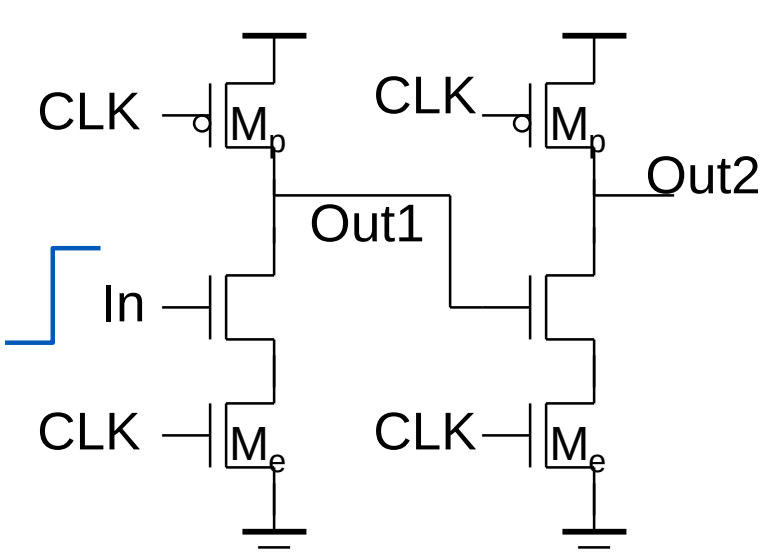
Minimum clock rate of a few kHz

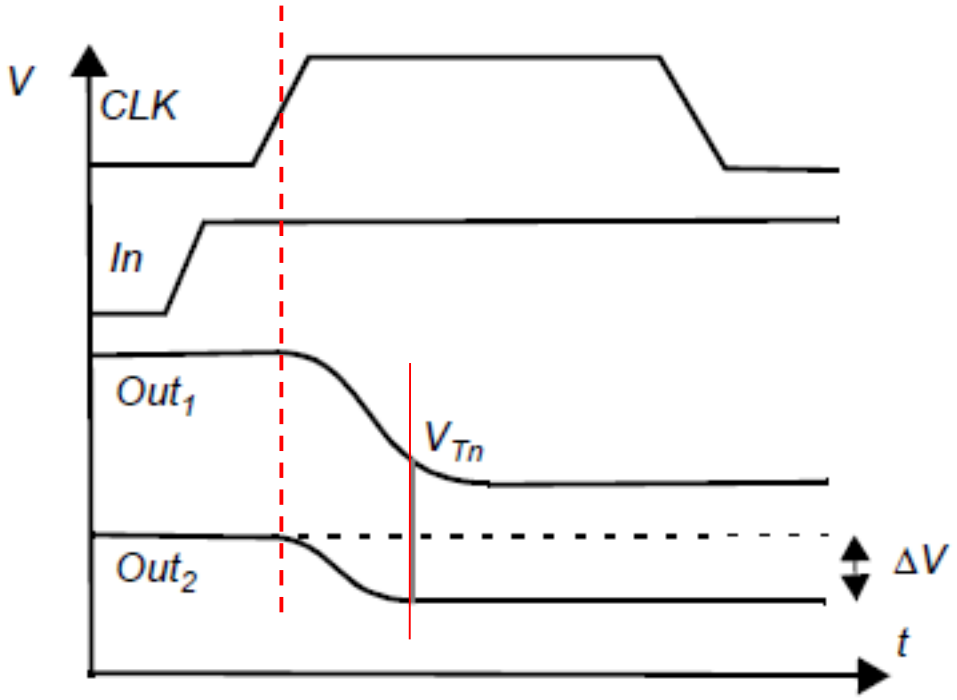




Output settles to an intermediate voltage determined by a resistive divider of the pull-up and pull-down networks *(Once the output drops below V_M of the fan-out logic gate, the output is interpreted as a low voltage)*

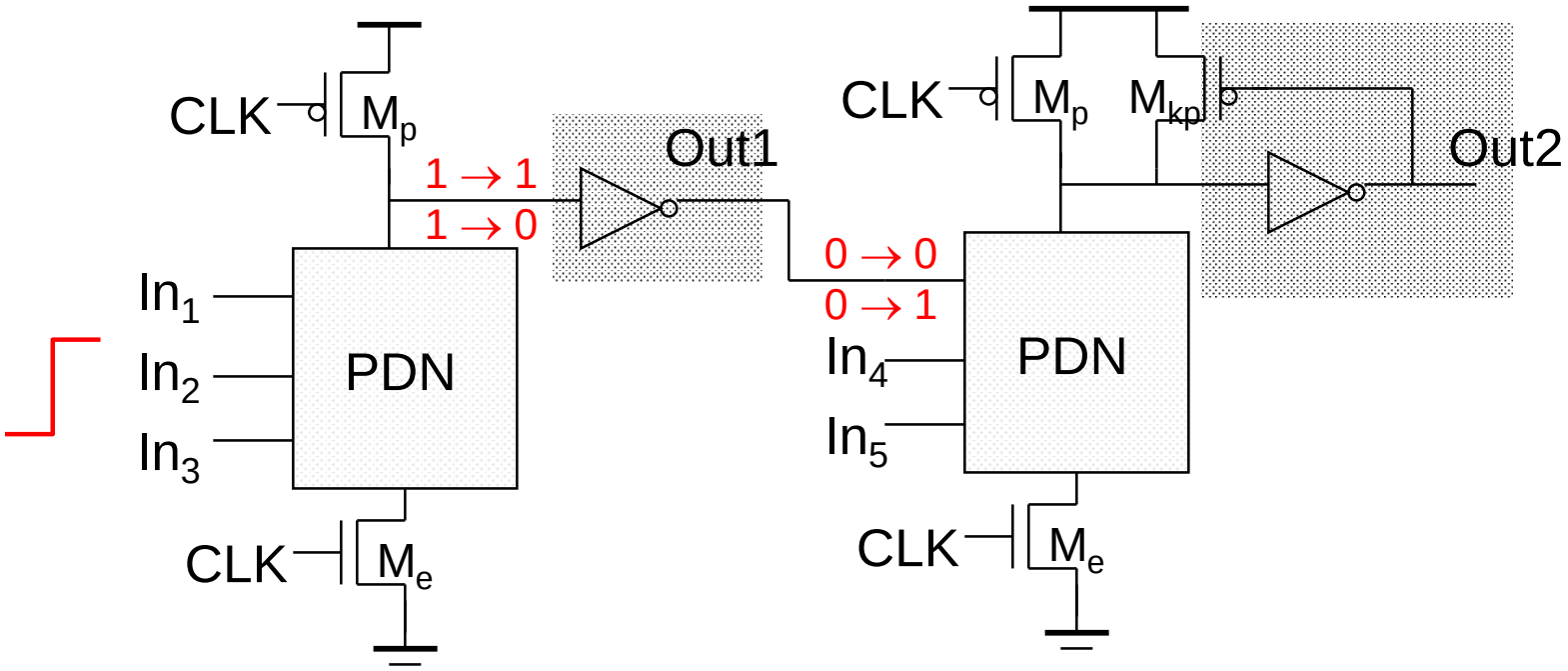
Cascading Dynamic Gates





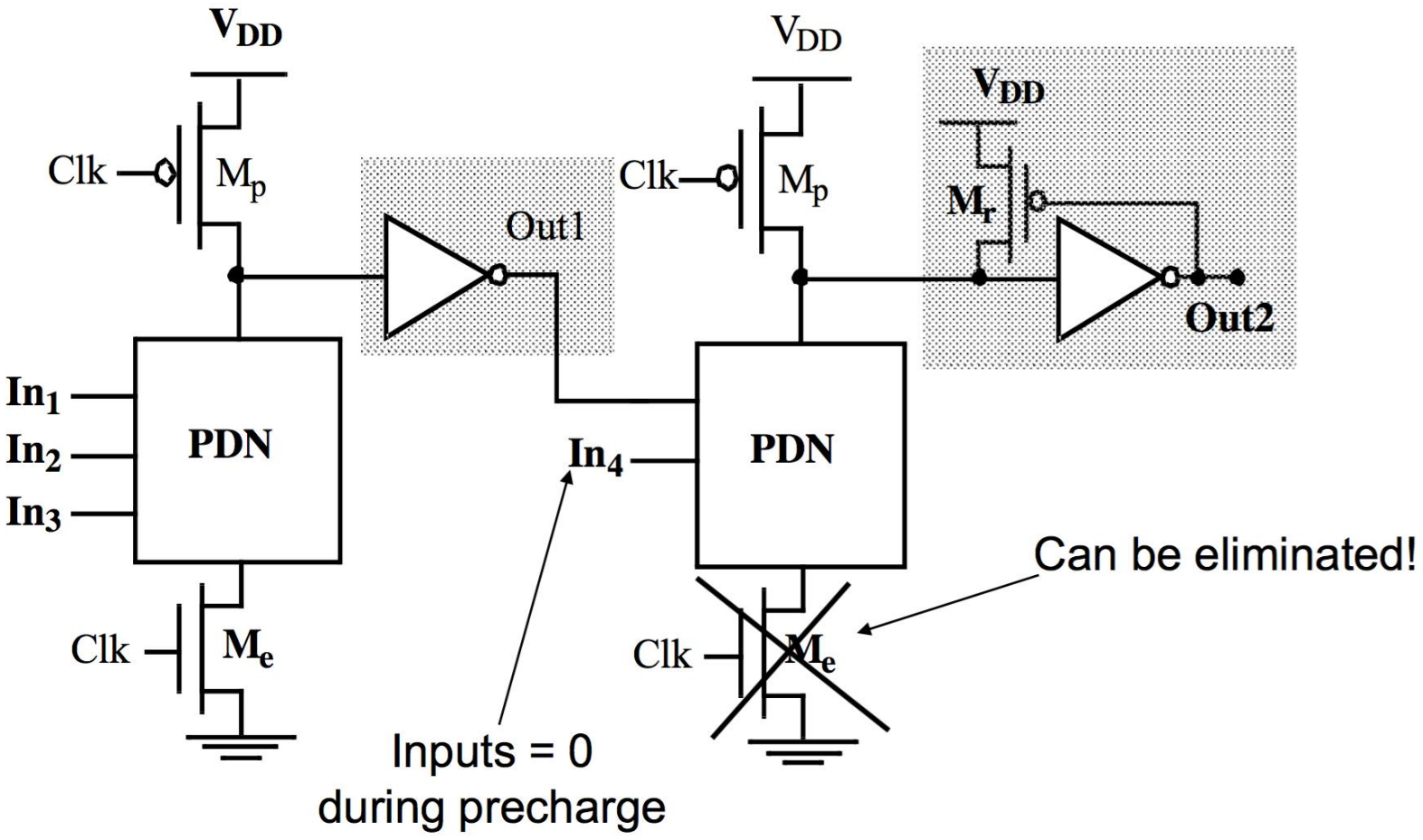
Only a single 0 → 1 transition allowed at the inputs during the evaluation period!

Domino Logic

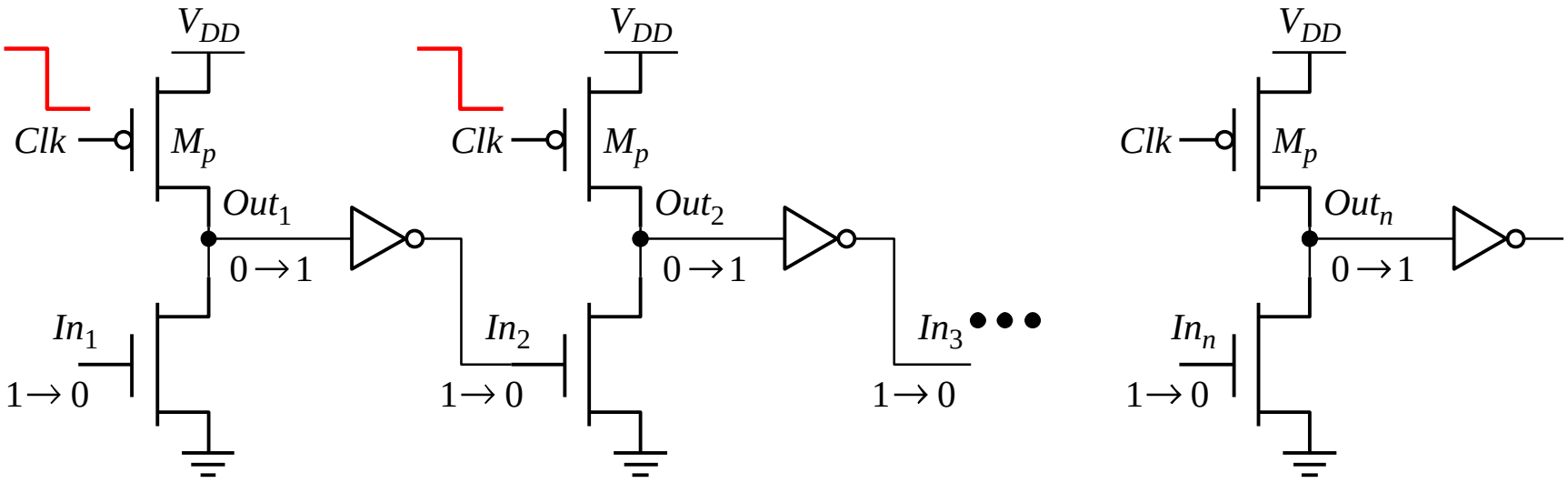


- Only non-inverting logic can be implemented
- Very high speed
  - static inverter can be skewed, only L-H transition
  - Input capacitance reduced – smaller logical effort

Designing with Domino Logic



Footless Domino



- The first gate in the chain needs a foot switch
  Precharge is rippling – short-circuit current
- A solution is to delay the clock for each stage

Properties of Domino Gates

- Logic function is implemented by the PDN only requires N + 2 transistors (vs. static complementary CMOS：2N)
- Be smaller in area (vs. static complementary CMOS)
- Full swing outputs (V_{OL} = GND and V_{OH} = VDD)
- Faster switching speeds:

  C_{int} reduced (fewer transistors connected to the drain output)

  C_{ext} (output load is one per connected gate, not two)
- **Inverter is required**
- Needs a precharge clock.

Summary

- Static **complementary static CMOS logic** combines dual PDN and PUN networks. The performance of a CMOS gate is a strong function of the fan-in, and also is a function of the fan-out.
- The ratioed logic style consists of an active pull-down network connected to a load device.
- **Pass-transistor** logic implements a logic gate as a simple switch network.
- Dynamic logic: a two-phase scheme consisting of a pre-charge followed by an evaluation step.

- Current trend is towards an increased use of **complementary static CMOS logic**: design support through EDA tools, robust, more amenable to voltage scaling.

集成电路原理与设计

11. 逻辑门

宋爽

shuangsonghz@zju.edu.cn