دانشگاه صنعتی امیرکبیر

دانشکده مهندسی برق

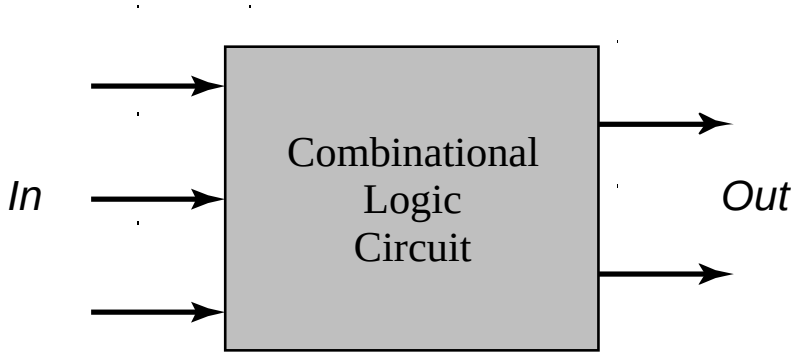
طراحی مدارهای VLSI

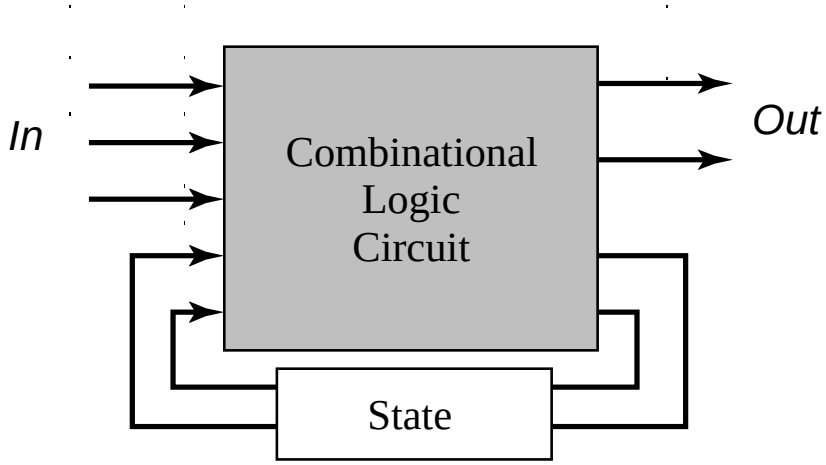
فصل پنجم: مدار های منطقی ترکیبی

مجید شالچیان

majid.shalchian@gmail.com

Combinational vs. Sequential Logic





Combinational

Sequential

Output = f(In)

Output = f(In, Previous In)

منطق CMOS ایستا و پویا

❑ منطق مکمل ایستای CMOS

خروجی بجز در حالت سوییچینگ از طریق یک مسیر با امپدانس کم به زمین یا VDD وصل می شود.

حاشیه نویز زیاد است و $V_{OH} = V_{DD}$ و $V_{OL} = 0\ V$

امپدانس خروجی کم و امپدانس ورودی زیاد

تلفات ایستا نداریم

زمان های فراز و فرود نسبتا متقارن (با تنظیم ابعاد ترانزیستور ها)

❑ منطق پویای CMOS

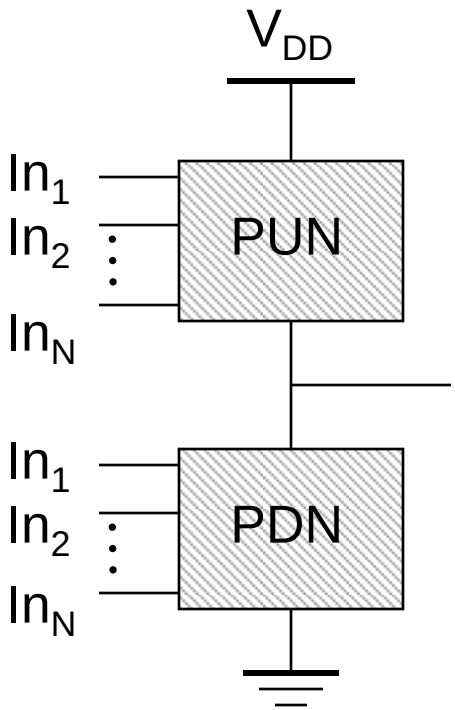
ذخیره مقدار سیگنال بصورت موقت روی خازن گره خروجی با امپدانس بالا

مدار ساده تر و سریعتر

حساسیت به نویز بیشتر است

Static Complementary CMOS

- Pull-up network (PUN) and pull-down network (PDN)



PMOS transistors only

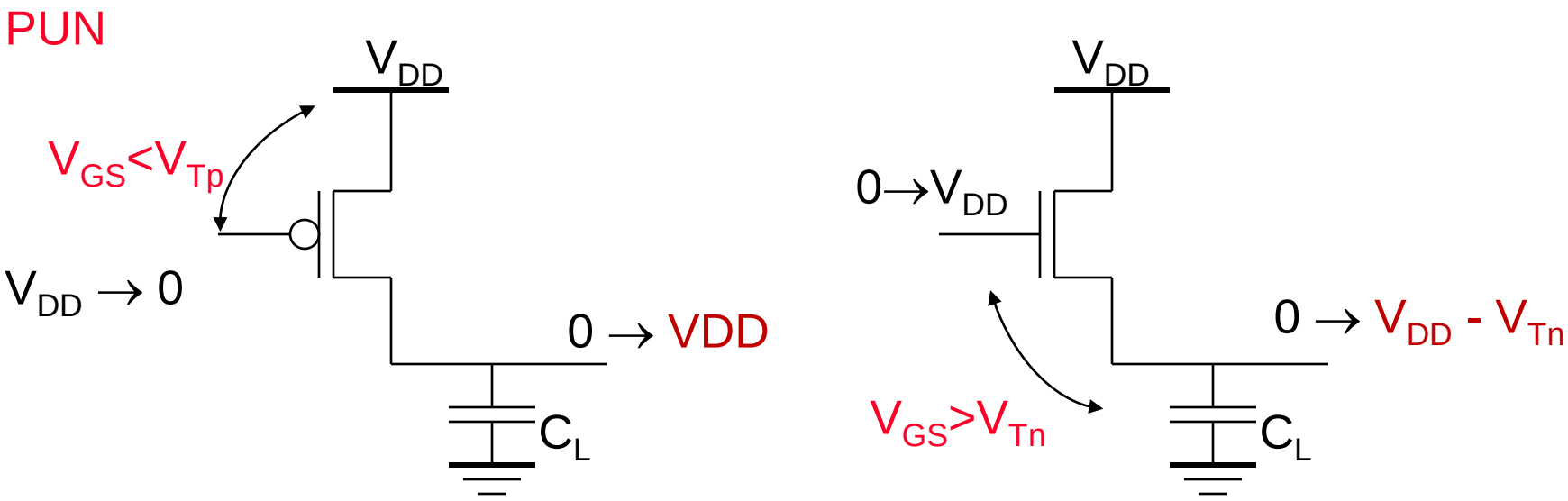
pull-up: make a connection from V_{DD} to F when $F(In_1,In_2,\ldots In_N) = 1$

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

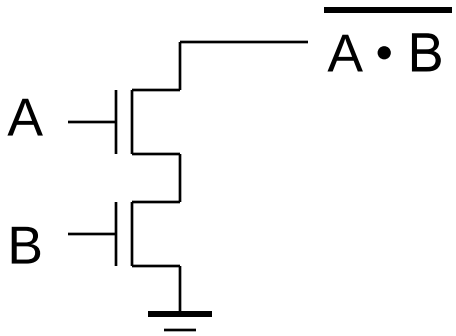
چرا PMOS درمسیر PUN



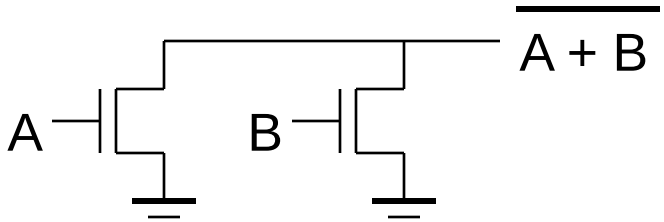
- PMOS در حالت روشن می تواند “1” قوی ایجاد کند اما NMOS “1” ضعیف ایجاد می کند

Construction of PDN

- NMOS devices in series implement a NAND function



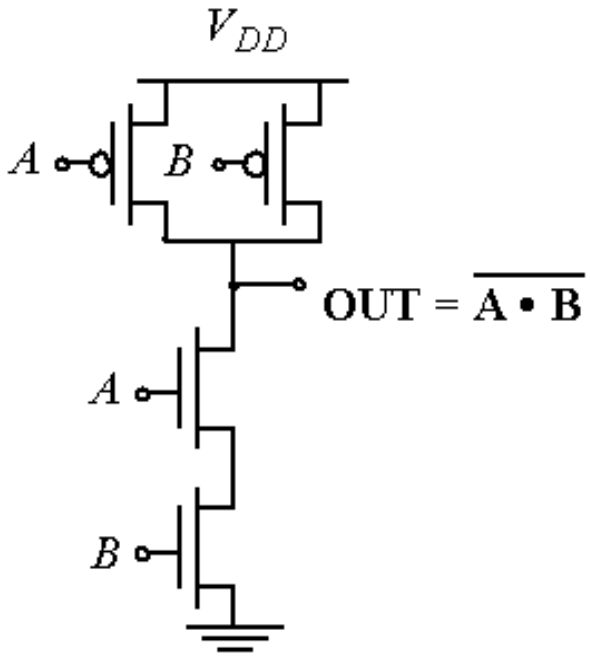
- NMOS devices in parallel implement a NOR function



Example Gate: NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table of a 2 input NAND gate



PDN: G = A B $\Rightarrow$ Conduction to GND

PUN: F= $\overline{A} + \overline{B} = \overline{AB} \Rightarrow$ Conduction to V_{DD}

$$\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

دوگان بودن شبکه پایین کش و بالا کش

❑ قضیه دمورگان

$$\overline{A + B} = \overline{A} \cdot \overline{B} \quad [!(A + B) = !A \cdot !B \text{ or } !(A \mid B) = !A \,\&\, !B]$$

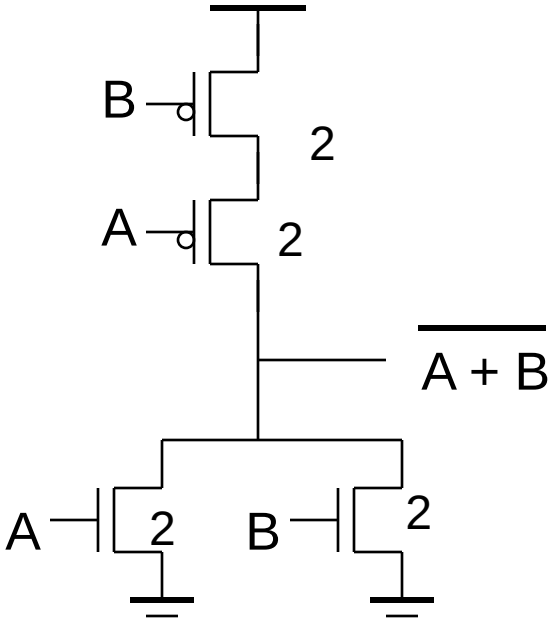
$$\overline{A \cdot B} = \overline{A} + \overline{B} \quad [!(A \cdot B) = !A + !B \text{ or } !(A \,\&\, B) = !A \mid !B]$$

- اتصال موازی در شبکه پایین کش تبدیل به اتصال سری در شبکه بالاکش می شود
- شبکه پایین کش را با وارون کردن خروجی پیاده می کنیم
- شبکه بالاکش را بر اساس قضیه دمورگان پیاده می کنیم (PMOS ورودی را مکمل می کند)

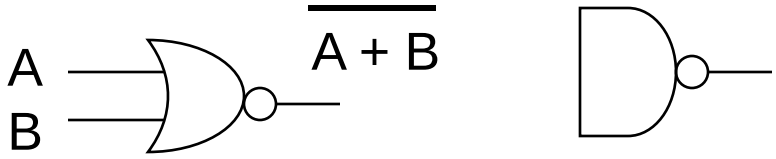
❑ گیت های CMOS مکمل ایستا همیشه خروجی مکمل (وارون) تولید می کنند (NAND, NOR)

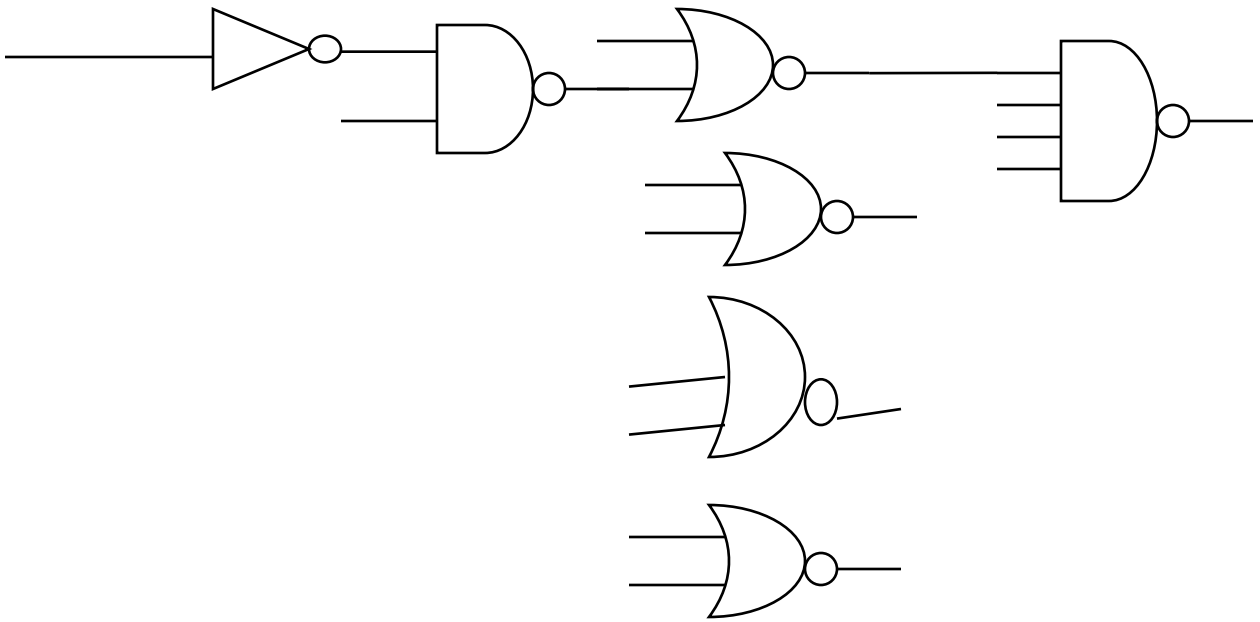
❑ تعداد ترانزیستور ها برای مدار N ورودی 2N می باشد.

Example Gate: NOR

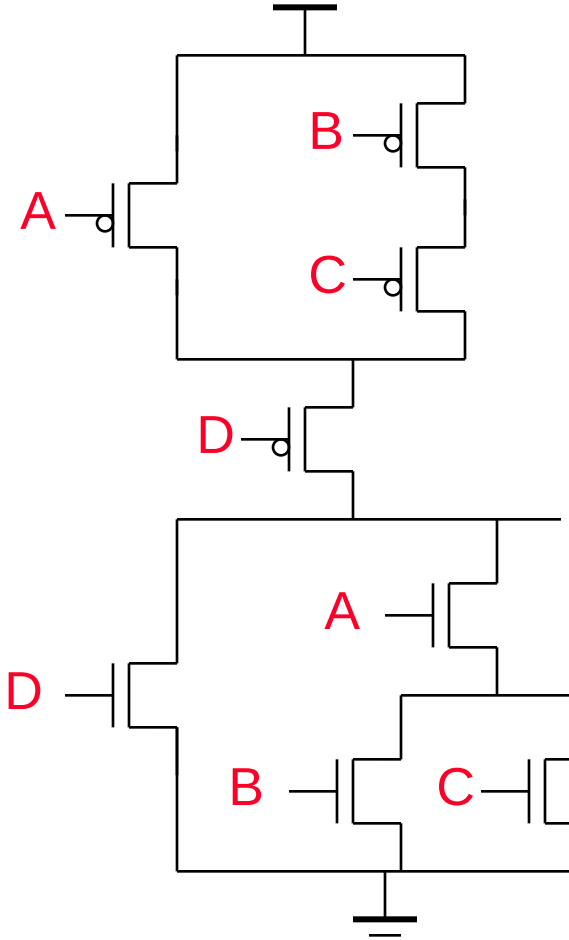


A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



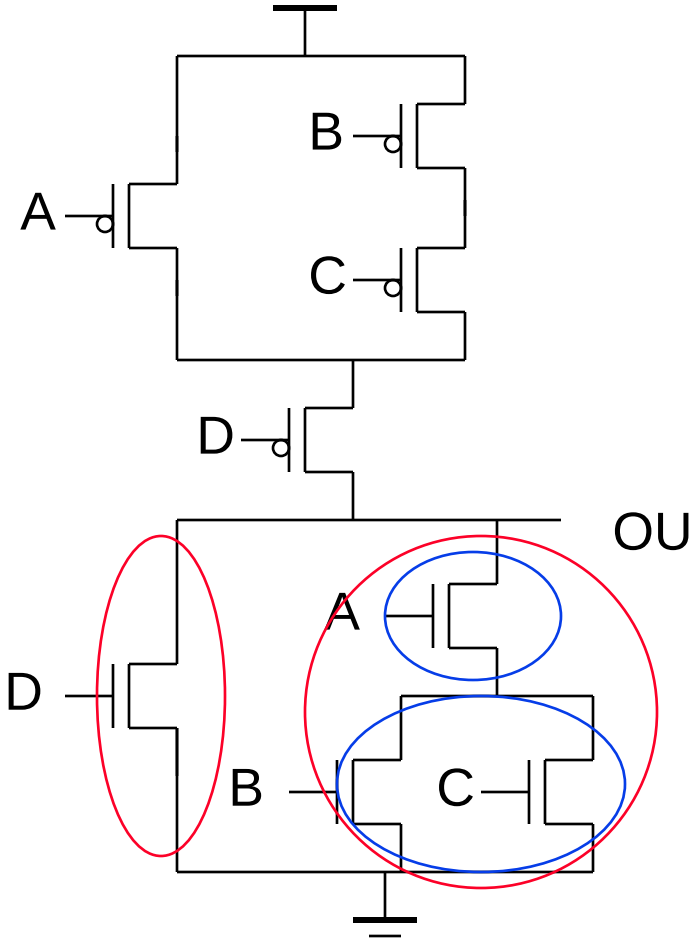


Complex CMOS Gate



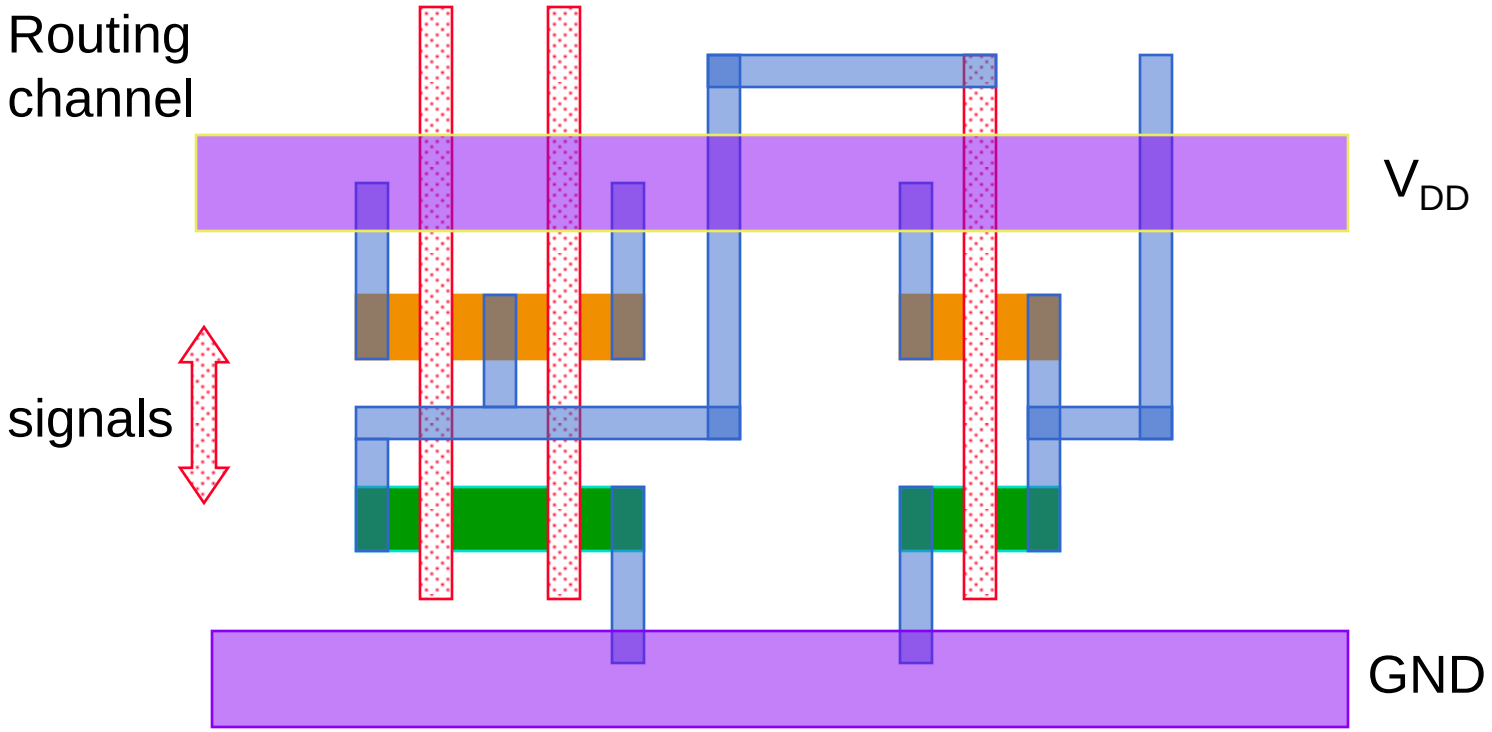
$$OUT = \overline{D + A \bullet (B + C)}$$

Complex CMOS Gate



$$OUT = \overline{(D + A \bullet (B + C))}$$

لی اوت گیت مکمل ایستا

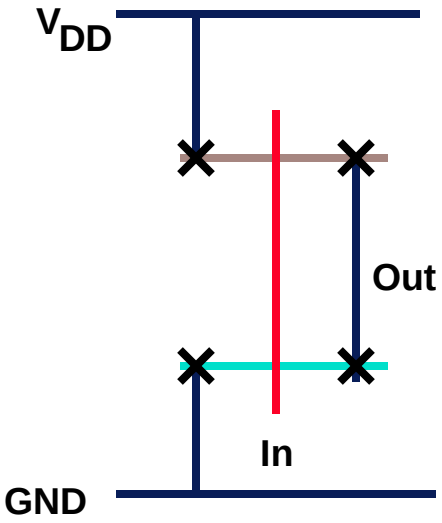


What logic function is this?

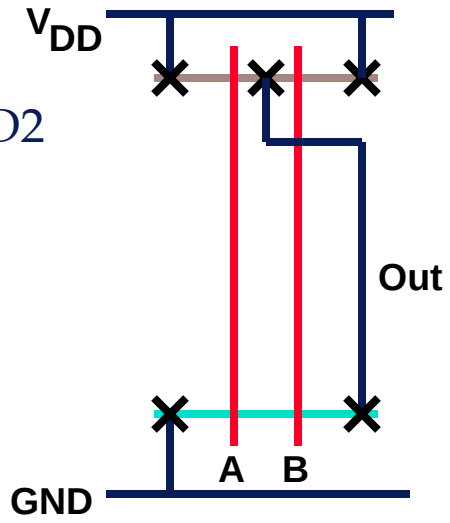
نمودار میله ای

جانمایی ترانزیستور ها و طراحی ساختار layout

Inverter

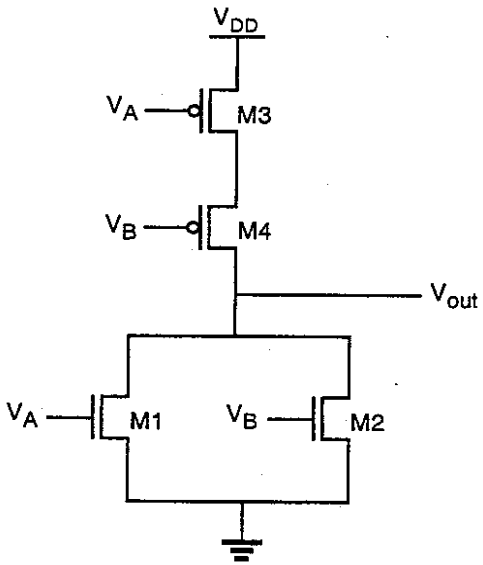


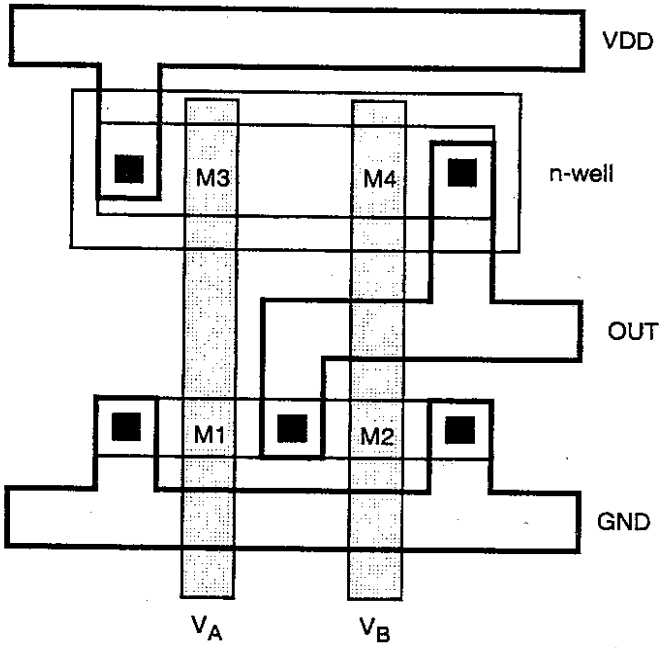
NAND2

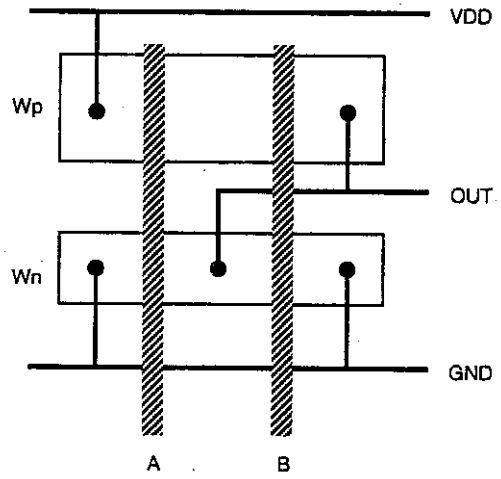


CMOS 2-Input NOR with Layout

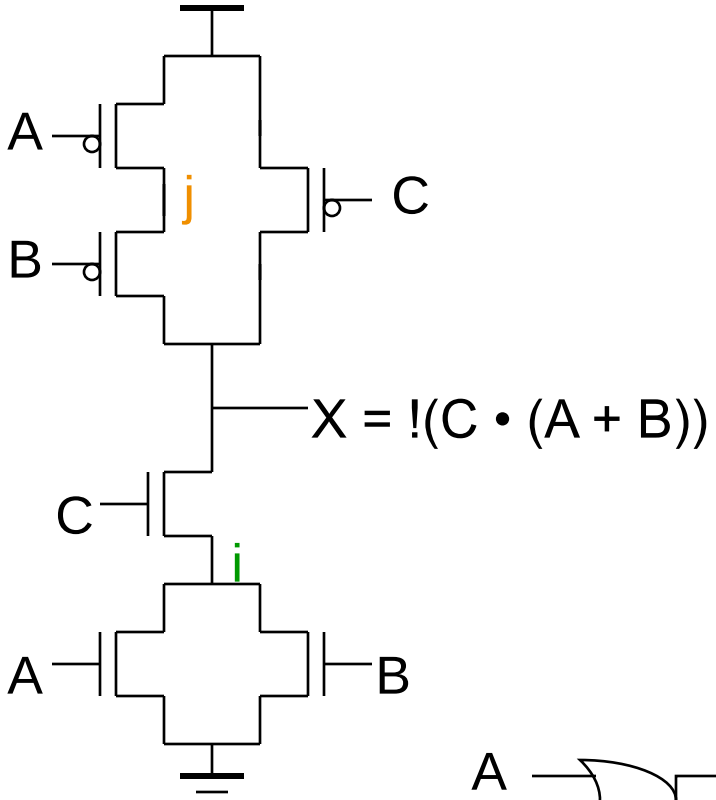
- Features of the layout are similar to the 2-input NAND
  - Single vertical poly lines for each input
  - Single active shapes for N and P devices, respectively
  - Metal busing running horizontal
- Also shown is a stick figure diagram for the NOR2 which corresponds directly to the layout, but does not contain W and L information
  - Stick figure diagram is useful for planning optimum layout topology

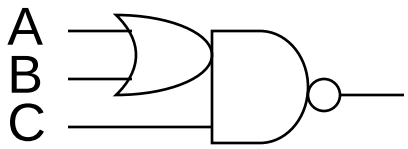


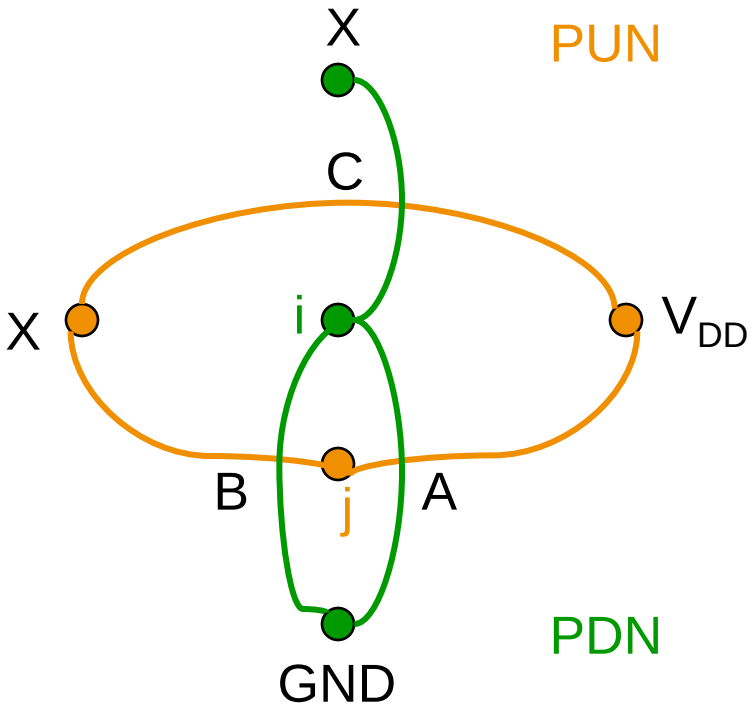




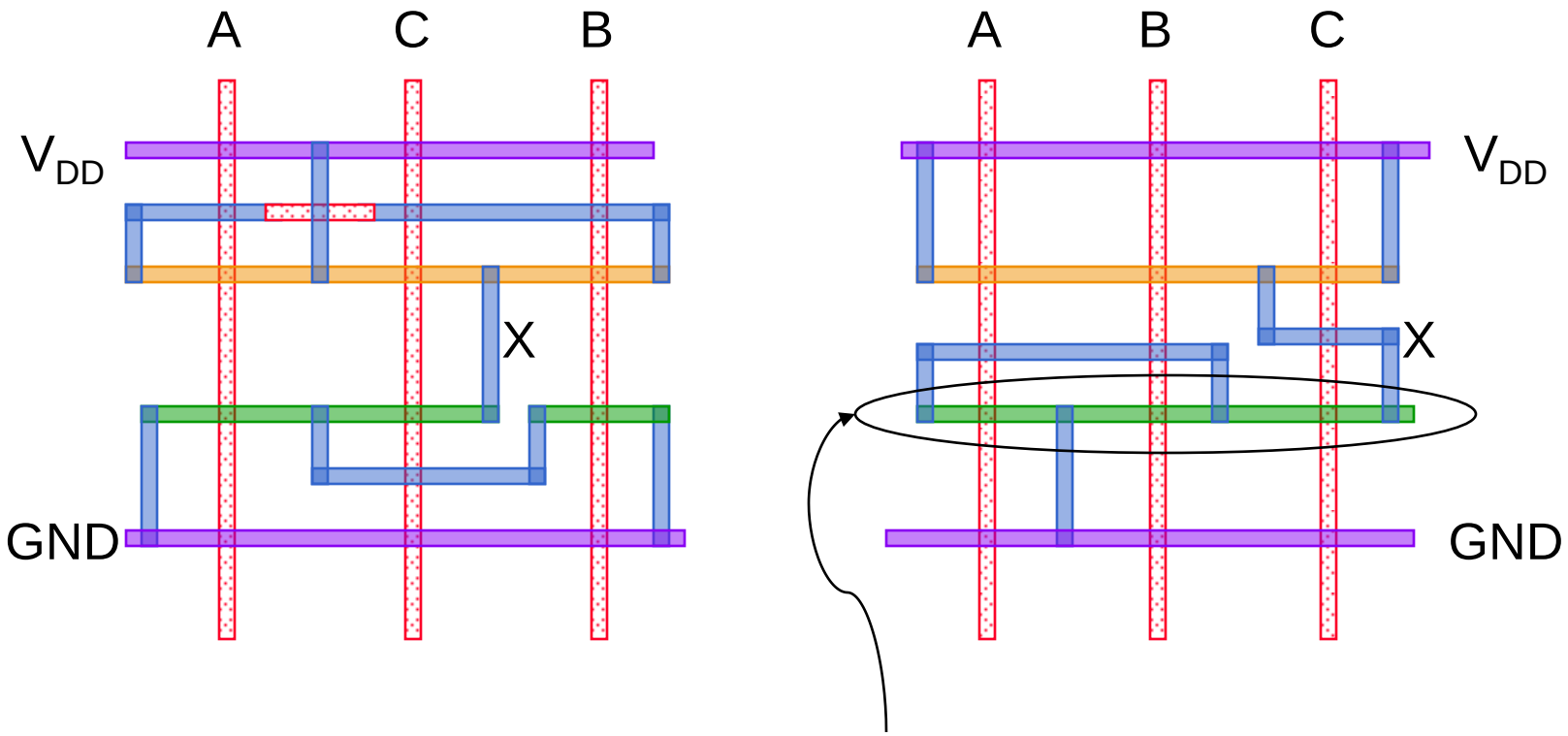
OAI21 Logic Graph





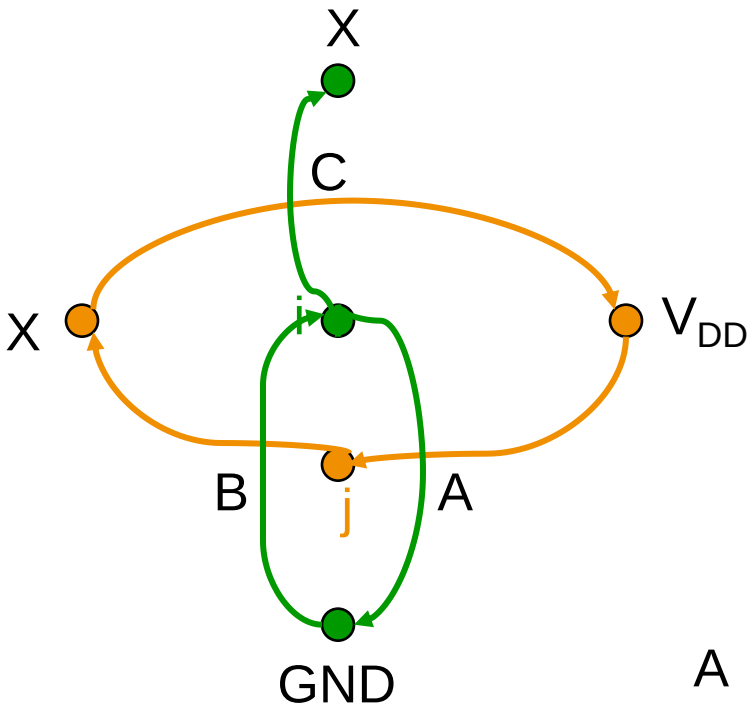


Two Stick Layouts of !(C • (A + B))



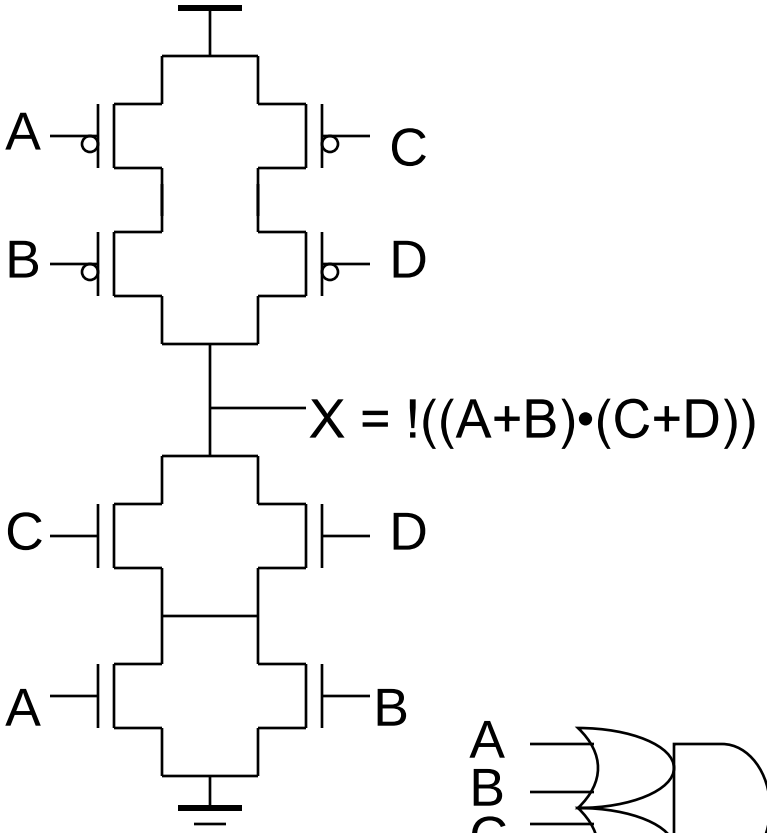
Consistent Euler Path

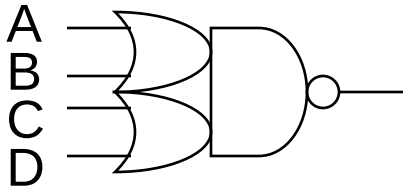
- An uninterrupted diffusion strip is possible only if there exists a Euler path in the logic graph
  - Euler path: a path through all nodes in the graph such that each edge is visited once and only once.

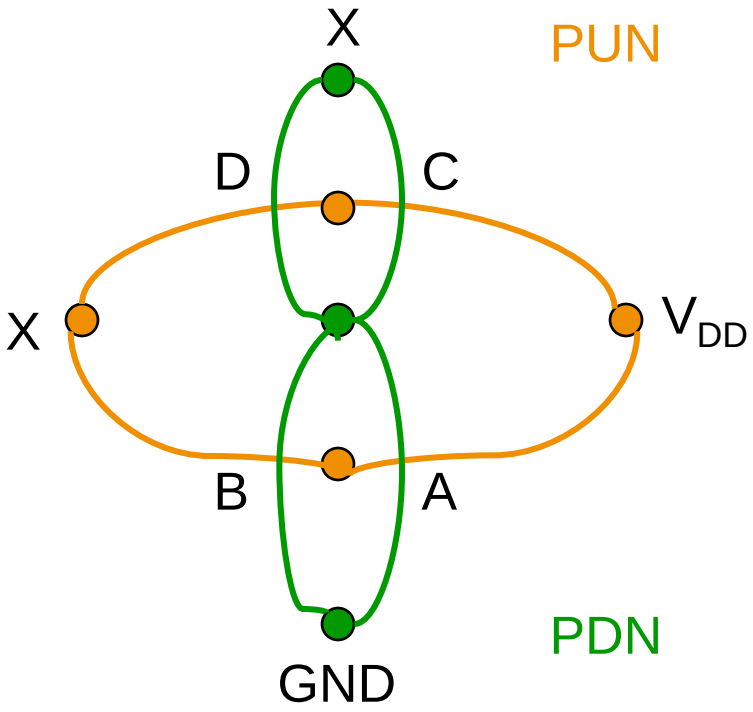


- For a single poly strip for every input signal, the Euler paths in the PUN and PDN must be consistent (the same)

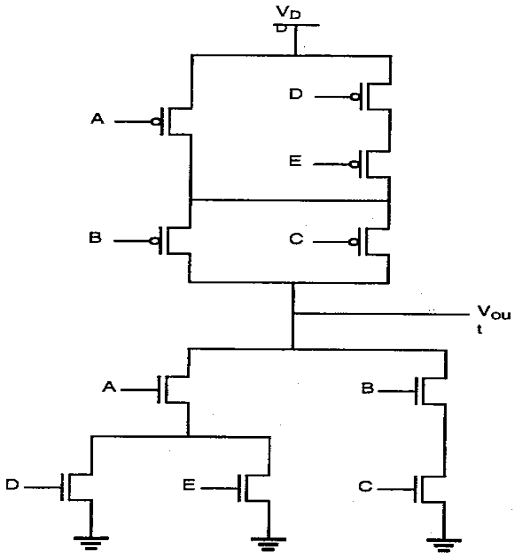
OAI22 Logic Graph

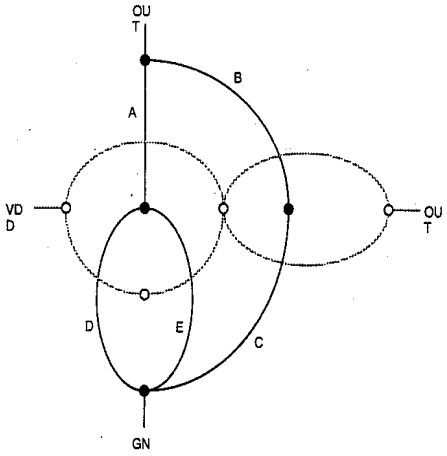


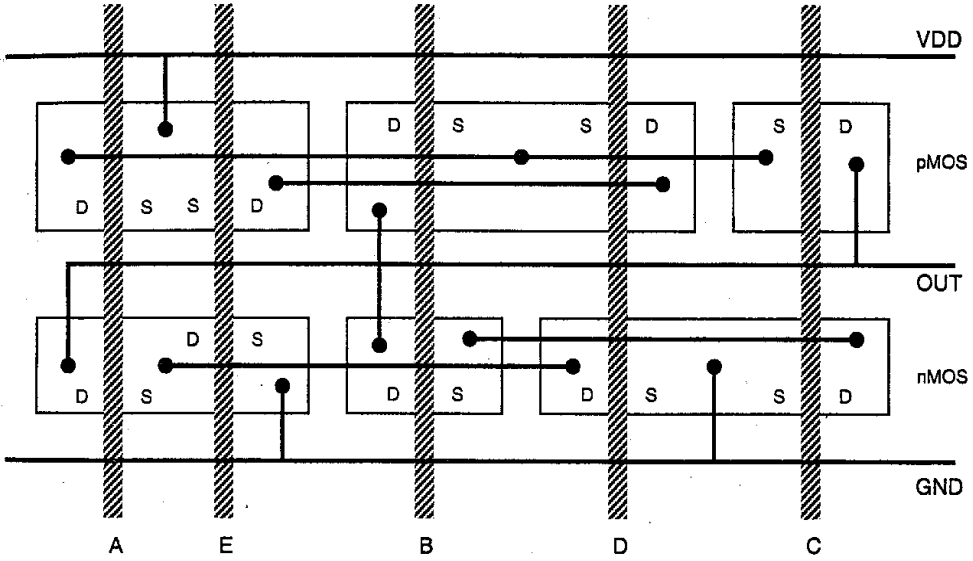




Layout Technique using Euler Graph Method

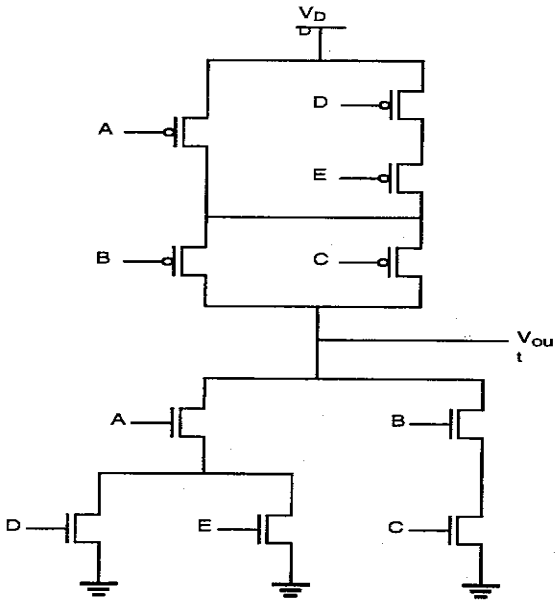




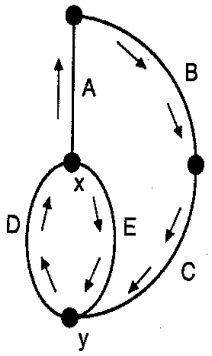


- Euler Graph Technique can be used to determine if any complex CMOS gate can be physically laid out in an optimum fashion
  - Start with either NMOS or PMOS tree (NMOS for this example) and connect lines for transistor segments, labeling devices, with vertex points as circuit nodes.
  - Next place a new vertex within each confined area on the pull-down graph and connect neighboring vertices with new lines, making sure to cross each edge of the pull-down tree only once.
  - The new graph represents the pull-up tree and is the dual of the pull-down tree.
- The stick diagram at the left (done with arbitrary gate ordering) gives a very non-optimum layout for the CMOS gate above.

Layout with Optimum Gate Ordering



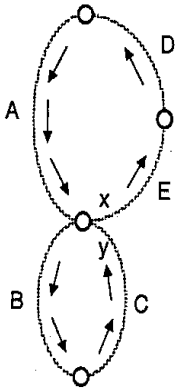
- By using the Euler path approach to re-order the polysilicon lines of the previous chart, we can obtain an optimum layout.
- Find a Euler path in both the pull-down tree graph and the pull-up tree graph with identical ordering of the inputs.
  - Euler path: traverses each branch of the graph exactly once!
- By reordering the input gates as E-D-A-B-C, we can obtain an optimum layout of the given CMOS gate with single
(belov



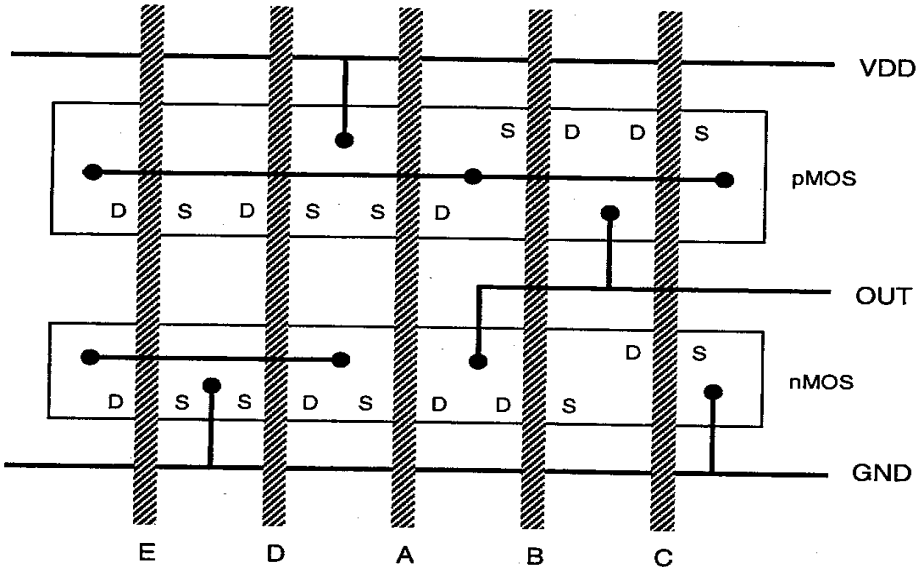
nMOS network

Common Euler path :

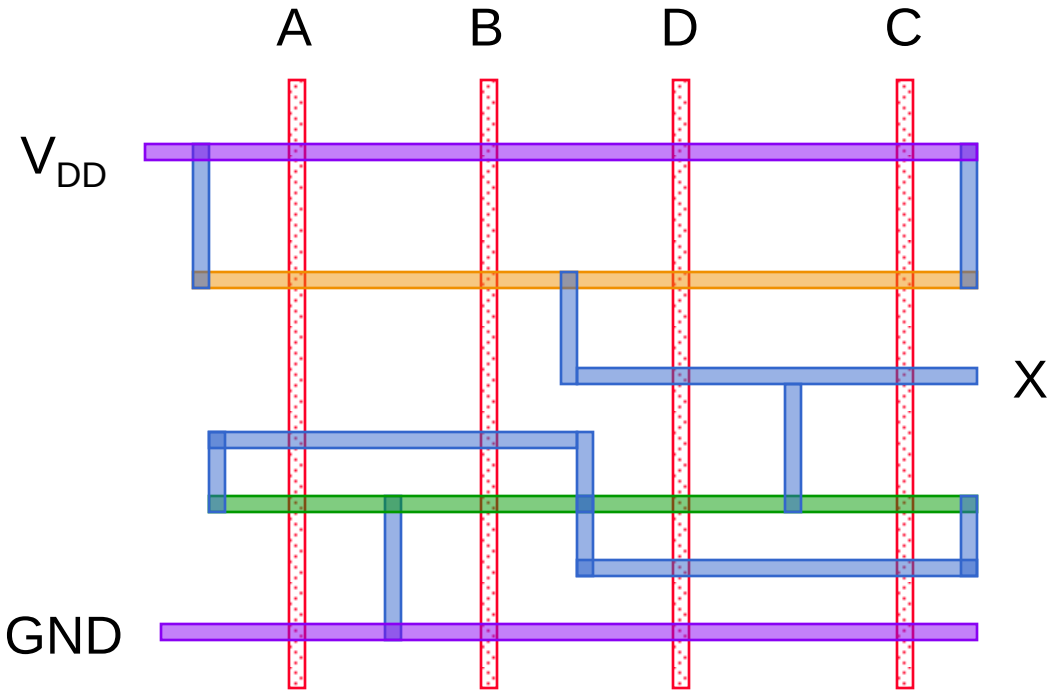
E - D - A - B - C



pMOS network



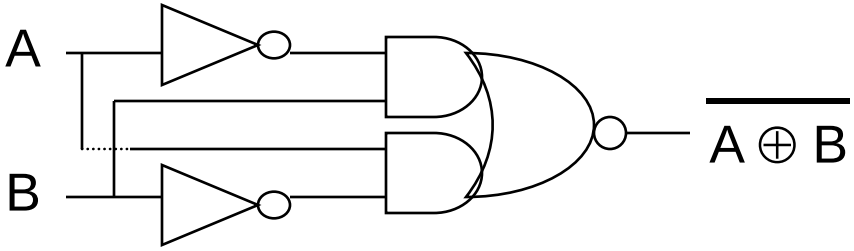
OAI22 Layout



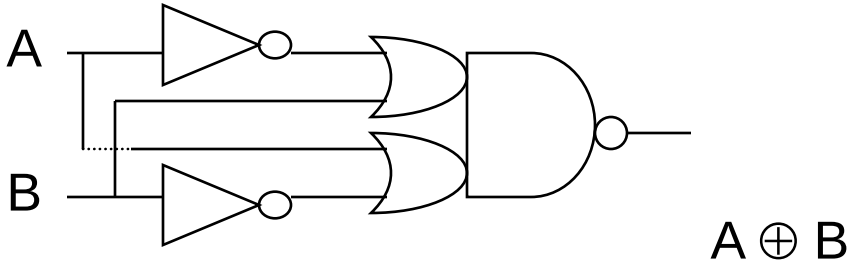
- Some functions have no consistent Euler path like x = !(a + bc + de) (but x = !(bc + a + de) does!)

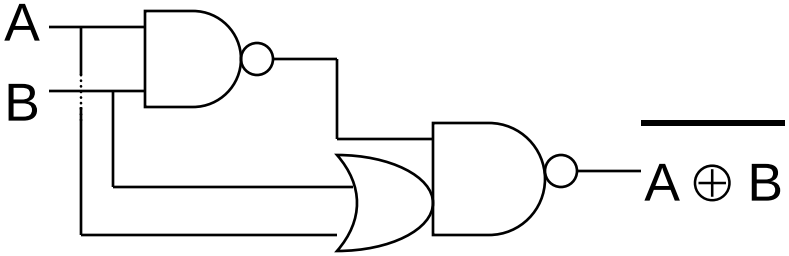
XNOR/XOR Implementation

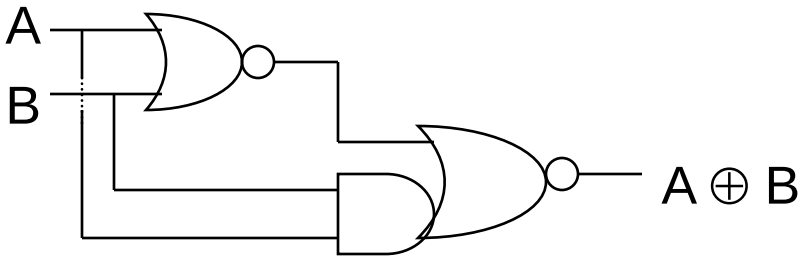
XNOR



XOR



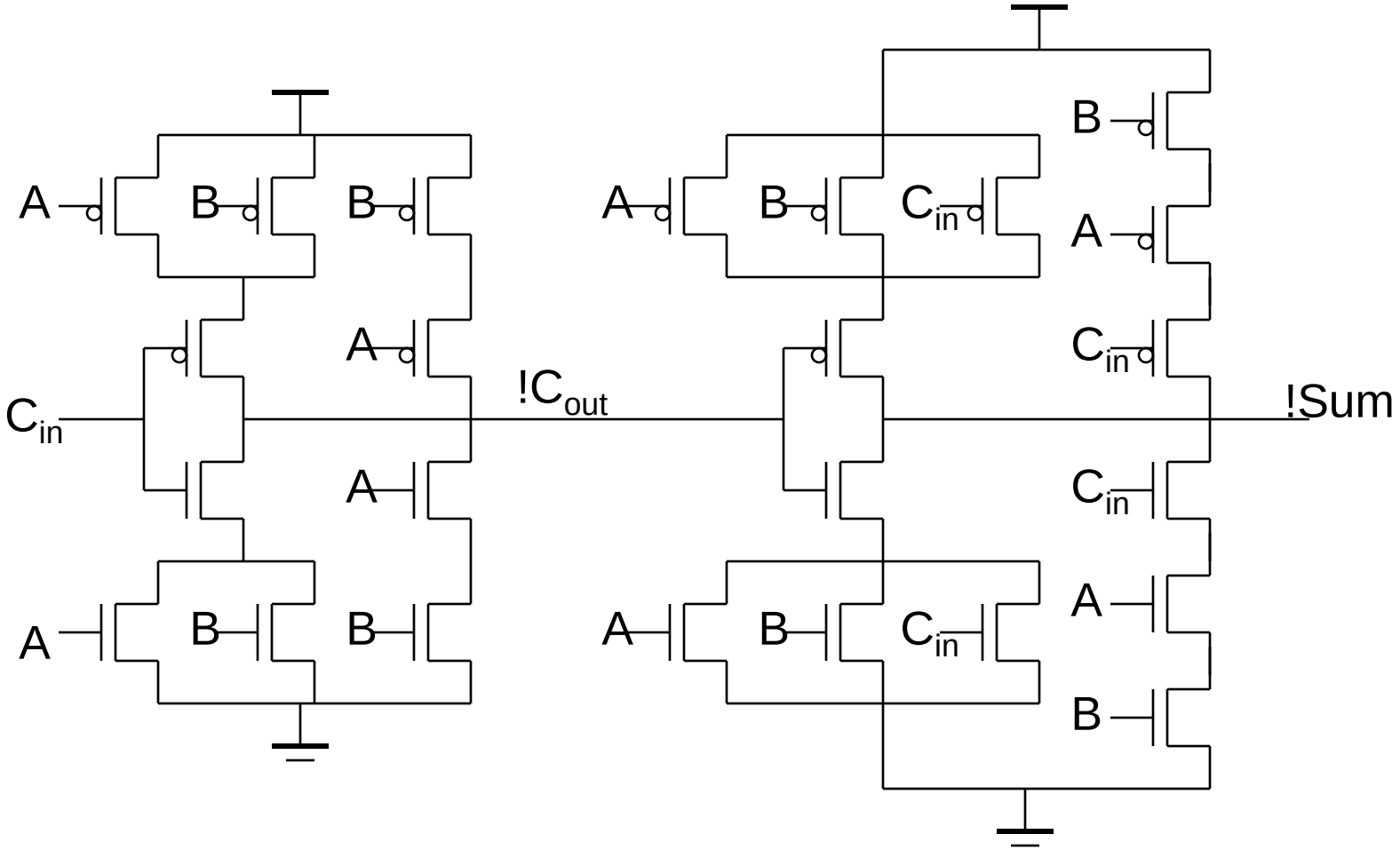




- How many transistors in each?

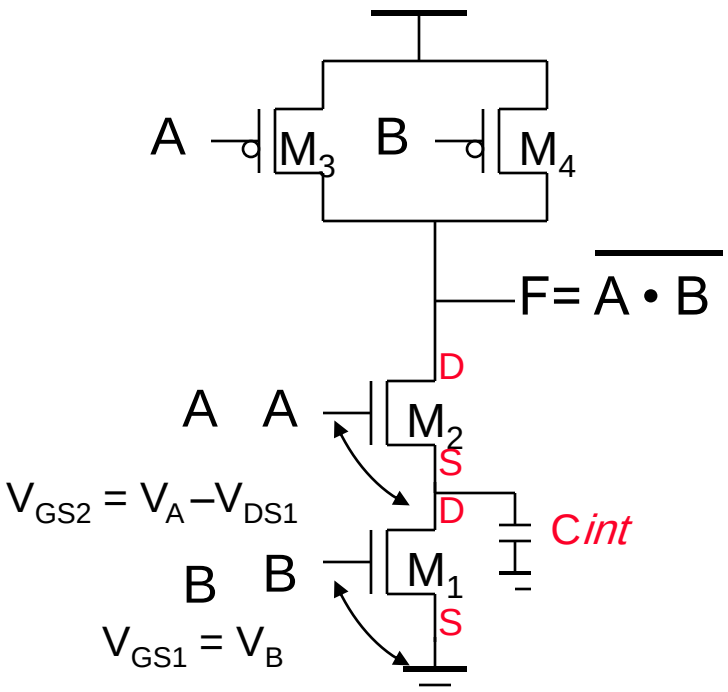
Static CMOS Full Adder Circuit

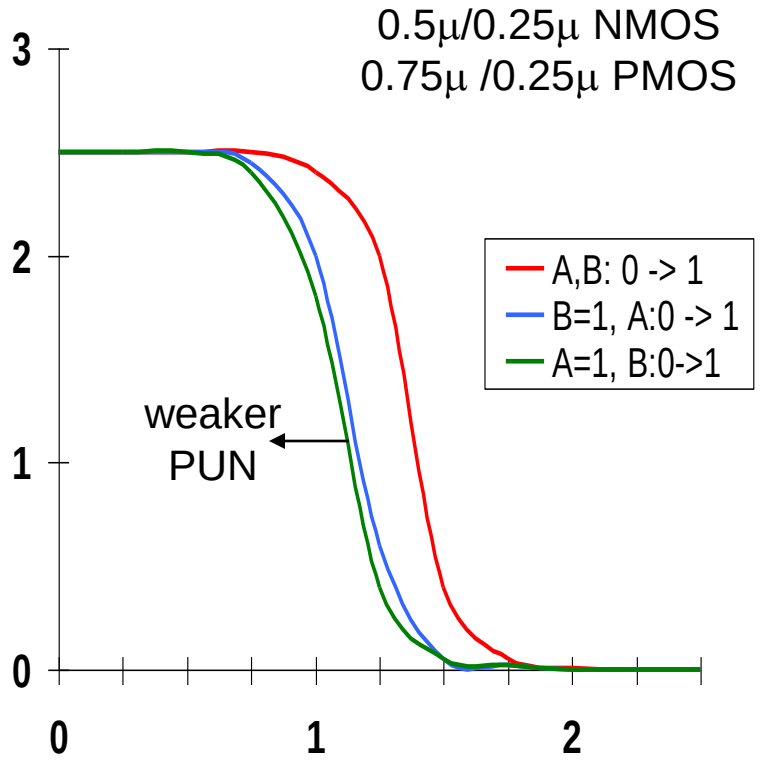
$!C_{out} = !C_{in} \,\&\, (!A \mid !B) \mid (!A \,\&\, !B)$ $!Sum = C_{out} \,\&\, (!A \mid !B \mid !C_{in}) \mid (!A \,\&\, !B \,\&\, !C_{in})$



$C_{out} = C_{in} \,\&\, (A \mid B) \mid (A \,\&\, B)$ $Sum = !C_{out} \,\&\, (A \mid B \mid C_{in}) \mid (A \,\&\, B \,\&\, C_{in})$

مشخصه انتقالی گیت NAND





- The threshold voltage of M_2 is higher than M_1 due to the body effect (γ)

$$V_{Tn1} = V_{Tn0}$$

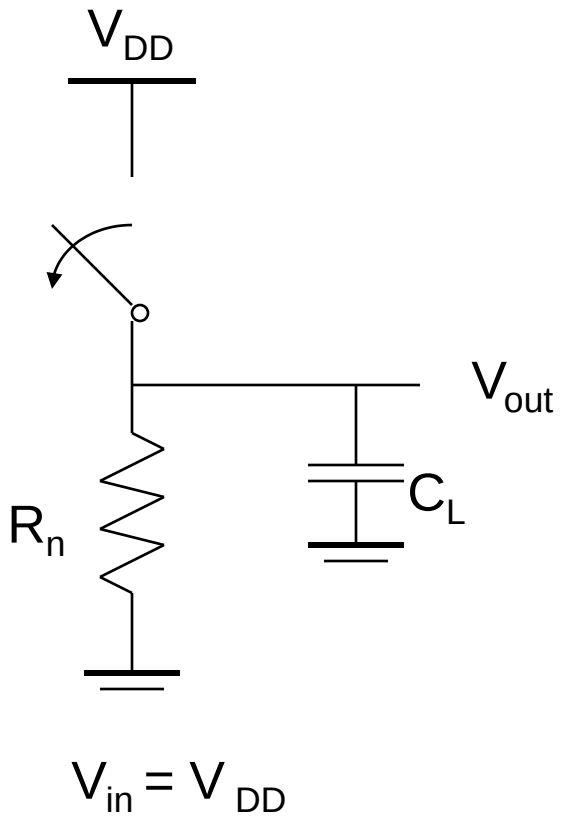
$$V_{Tn2} = V_{Tn0} + \gamma(\sqrt{(|2\phi_F| + V_{int})} - \sqrt{|2\phi_F|})$$

since V_{SB} of M_2 is not zero (when $V_B = 0$) due to the presence of C*int*

طراحی گیت های CMOS ایستا

تلاش منطقی

یادآوری تاخیر در وارونگر CMOS



$$t_{pHL} = f(R_n, C_L)$$

$$t_{pHL} = 0.69\, R_{eqn}\, C_L$$

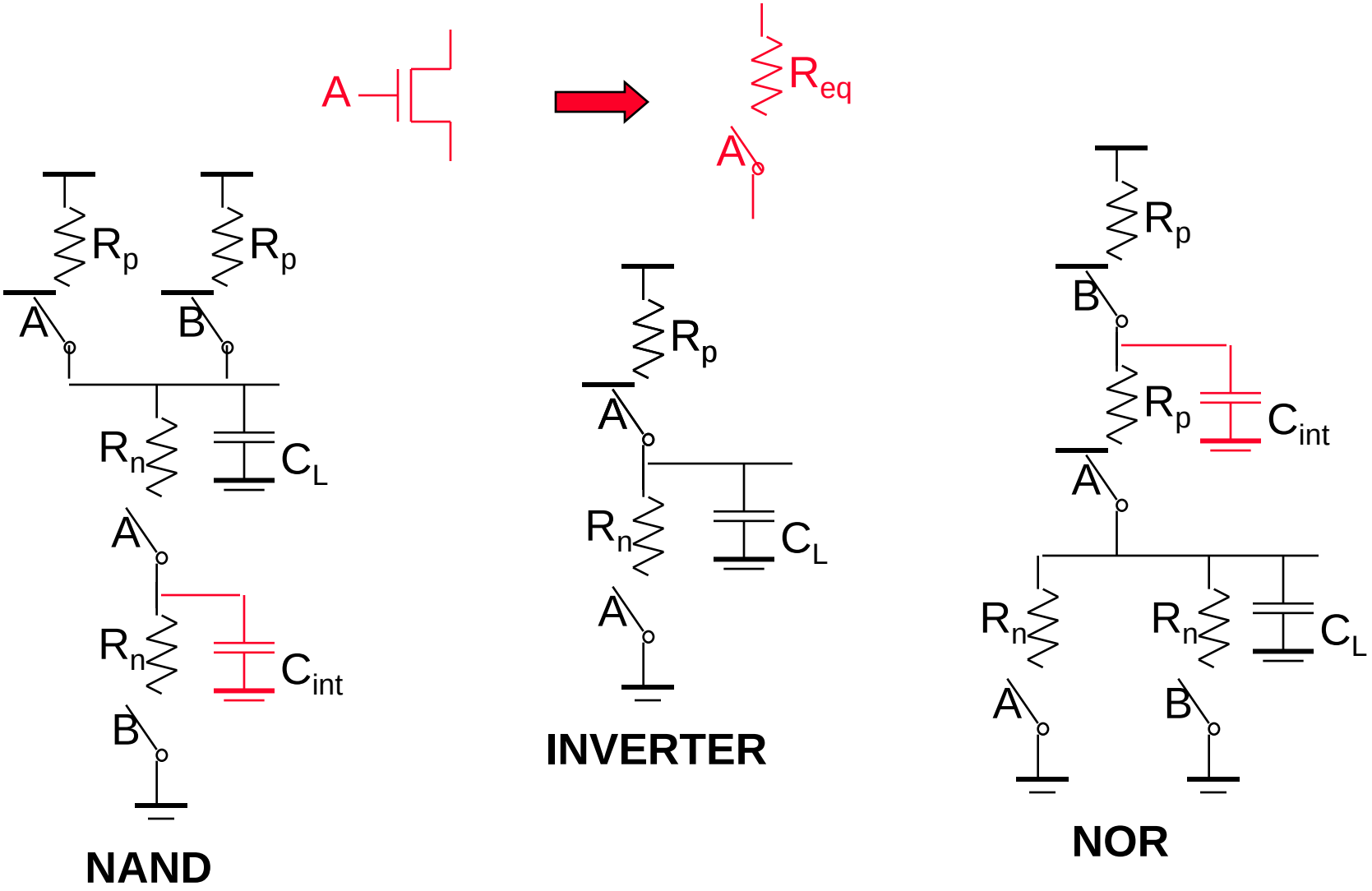
$$t_{pHL} = 0.69\, (3/4\, (C_L\, V_{DD}) / I_{DSATn})$$

$$= 0.52\, C_L / (W/L_n\, k'_n\, V_{DSATn})$$

راهکار های کاهش تاخیر

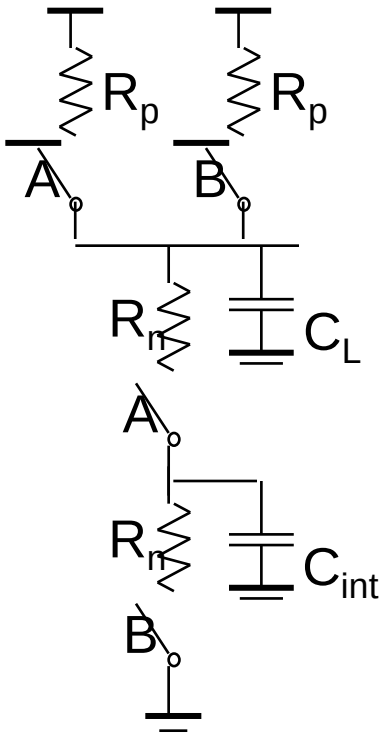
- کاهش C_L
  - کاهش سطح سورس و درین
  - کاهش خازن اتصالات
  - کاهش Fanout
- افزایش نسبت W/L
  - موثر ترین روش در اختیار طراح
  - توجه به مساله Self Loading
- افزایش V_{DD}
  - تاثیر جزیی روی عملکرد و افزایش تلفات توان
- Slope engineering
  - تلاش برای آنکه شیب زمان صعود و نزول سیگنال از شیب سیگنال ورودی کمتر یا مساوی آن باشد.

مدل سوییچ مقاومتی ترانزیستور



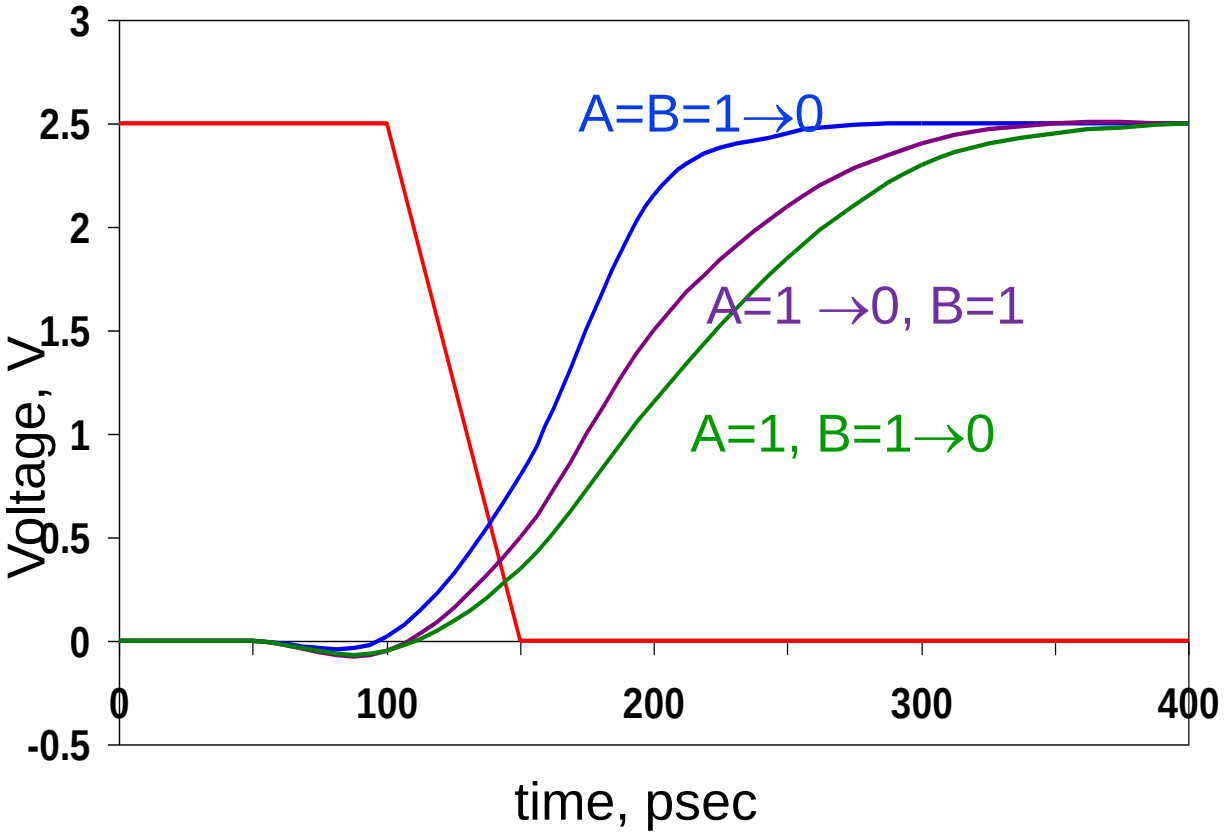
تاثیر داده ورودی روی تاخیر

- تاخیر بستگی به داده های ورودی دارد/
- تاخیر Low to High (خروجی)
  - اگر هر دو ورودی همزمان low شوند:
- delay is 0.69 $R_p/2$ C_L since two p-resistors are on in parallel
  - اگر فقط یک ورودی Low شود.
- delay is 0.69 R_p C_L
- تاخیر High to low (خروجی)
  - فقط وقتی هر دو وروی High شوند.
- delay is 0.69 $2R_n$ C_L
- سری کردن **ترانزیستور** ها بدون افزایش ابعاد باعث افزایش تاخیر می شود.



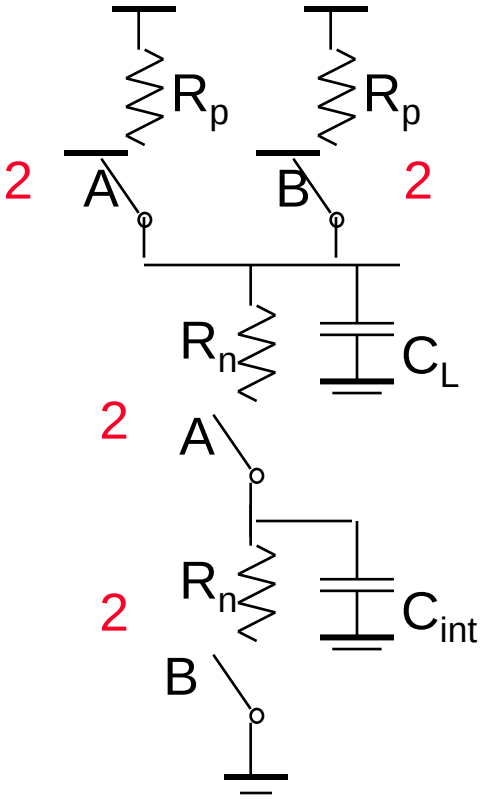
بستگی تاخیر به سیگنال های ورودی

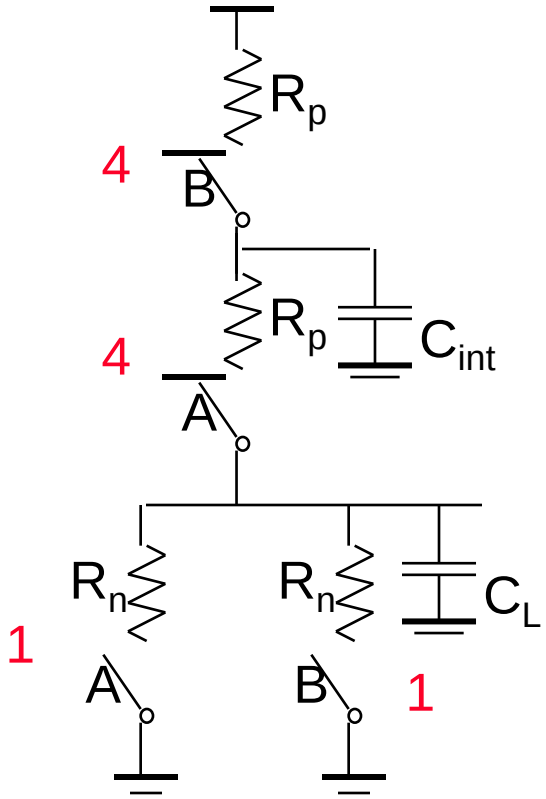
2-input NAND with
NMOS = 0.5μm/0.25 μm
PMOS = 0.75μm/0.25 μm
C_L = 10 fF



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

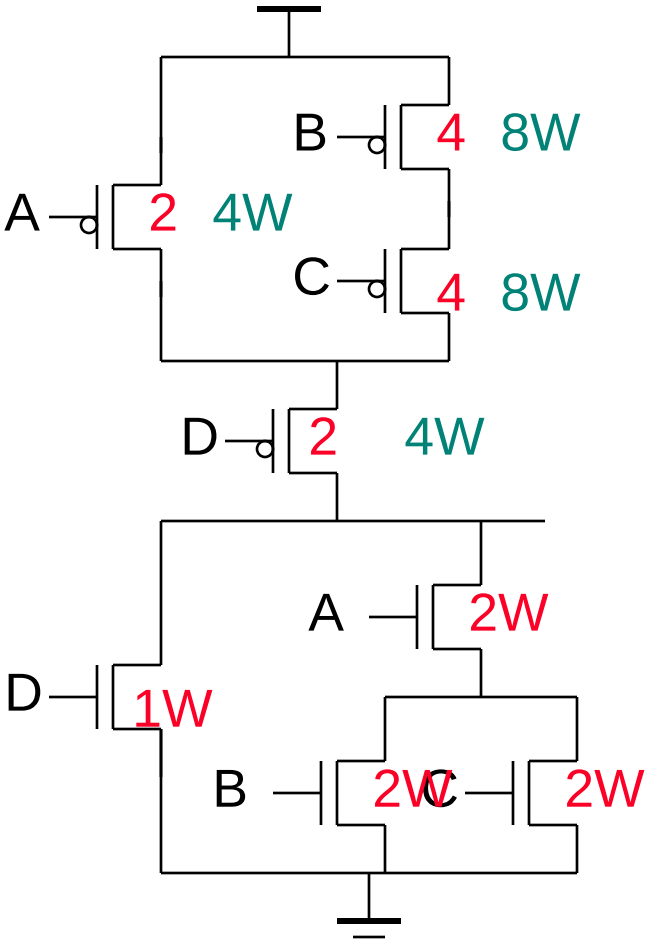
انتخاب ابعاد ترانزیستور ها





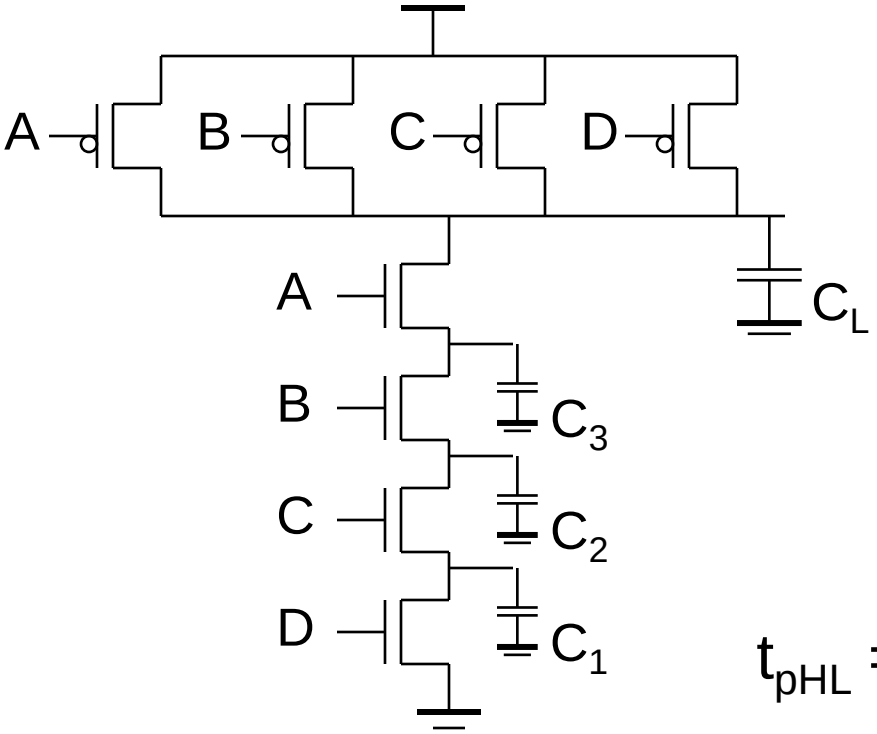
مثال : انتخاب ابعاد ترانزیستور ها

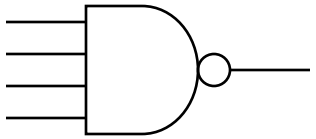
Wp(min)=2Wn(min)=2W



OUT = !(D + A • (B + C))

Fan-In Considerations



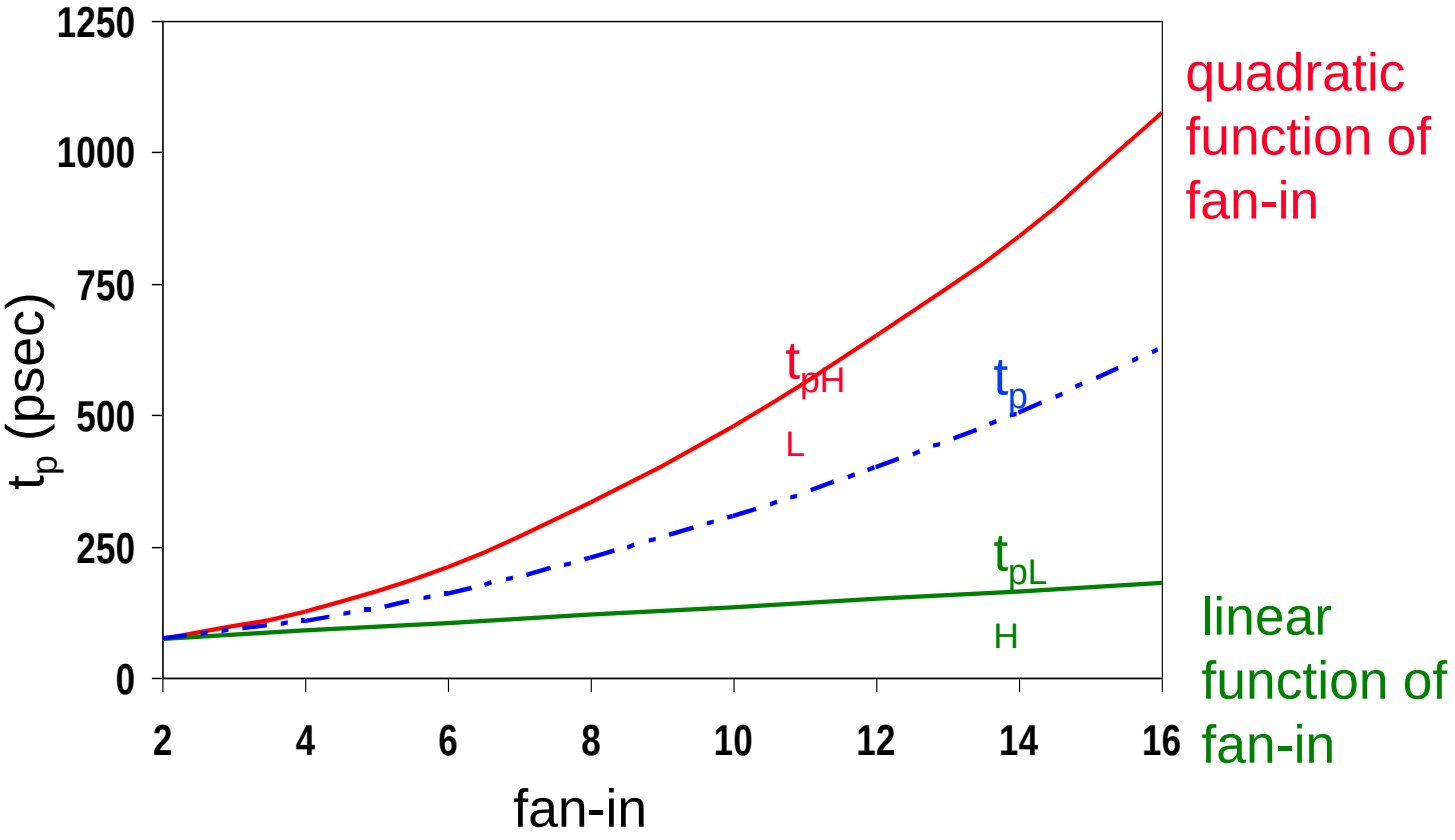


Distributed RC model
(Elmore delay)

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

Propagation delay increases as a function of fan-in – quadratic ally in the worst case.

t_p as a Function of Fan-In

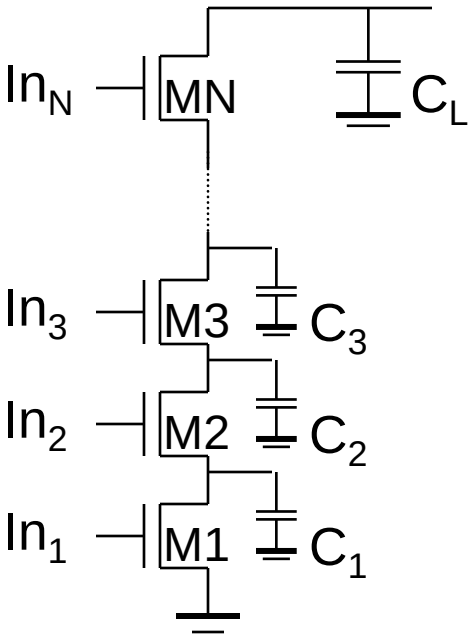


- Gates with a fan-in greater than 4 should be avoided.

Fast Complex Gates: Design Technique 1

- Transistor sizing
  - as long as fan-out capacitance dominates

- Progressive sizing



Distributed RC line

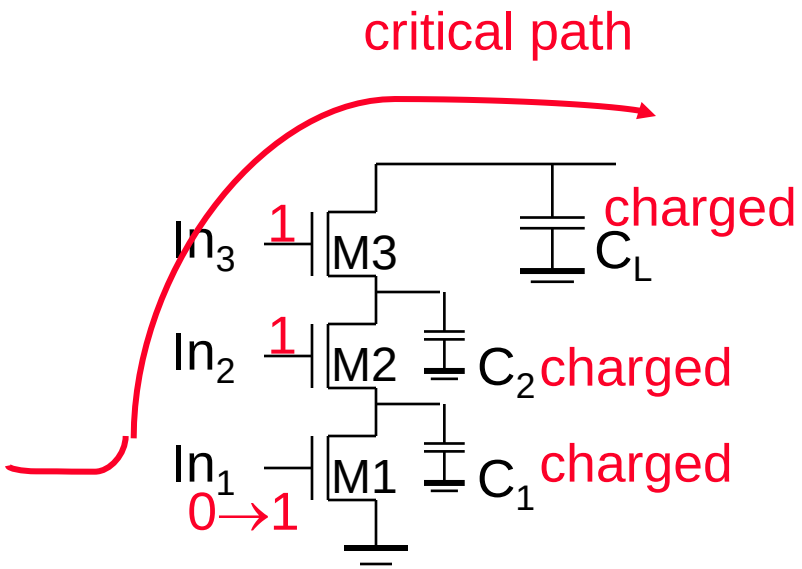
M1 > M2 > M3 > … > MN

(the fet closest to the output should be the smallest)

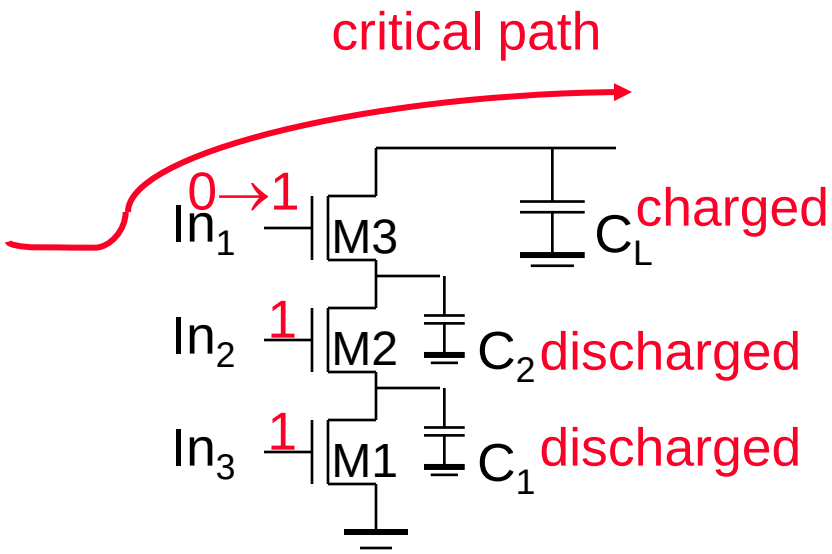
Can reduce delay by more than 20%; decreasing gains as technology shrinks

Fast Complex Gates: Design Technique 2

- Input re-ordering
  - when not all inputs arrive at the same time

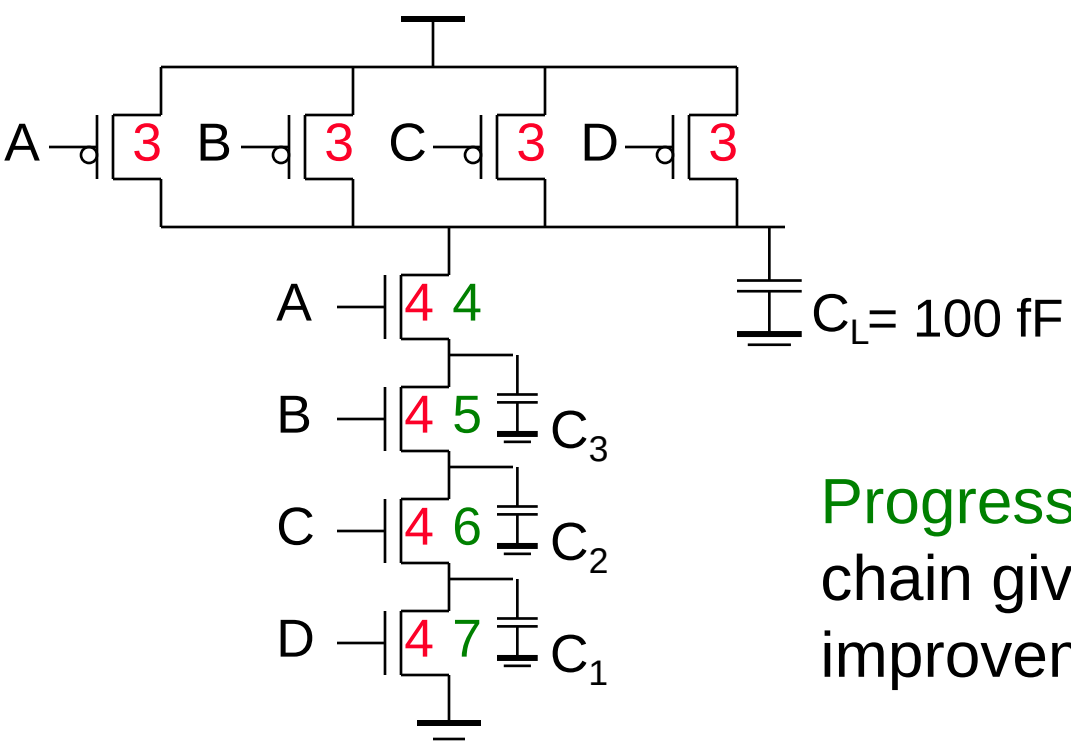


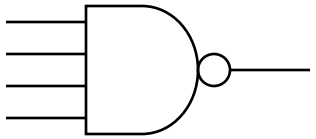
delay determined by time to discharge C_L, C_1 and C_2



delay determined by time to discharge C_L

Sizing and Ordering Effects





Progressive sizing in pull-down chain gives up to a 23% improvement.

Input ordering saves 5%

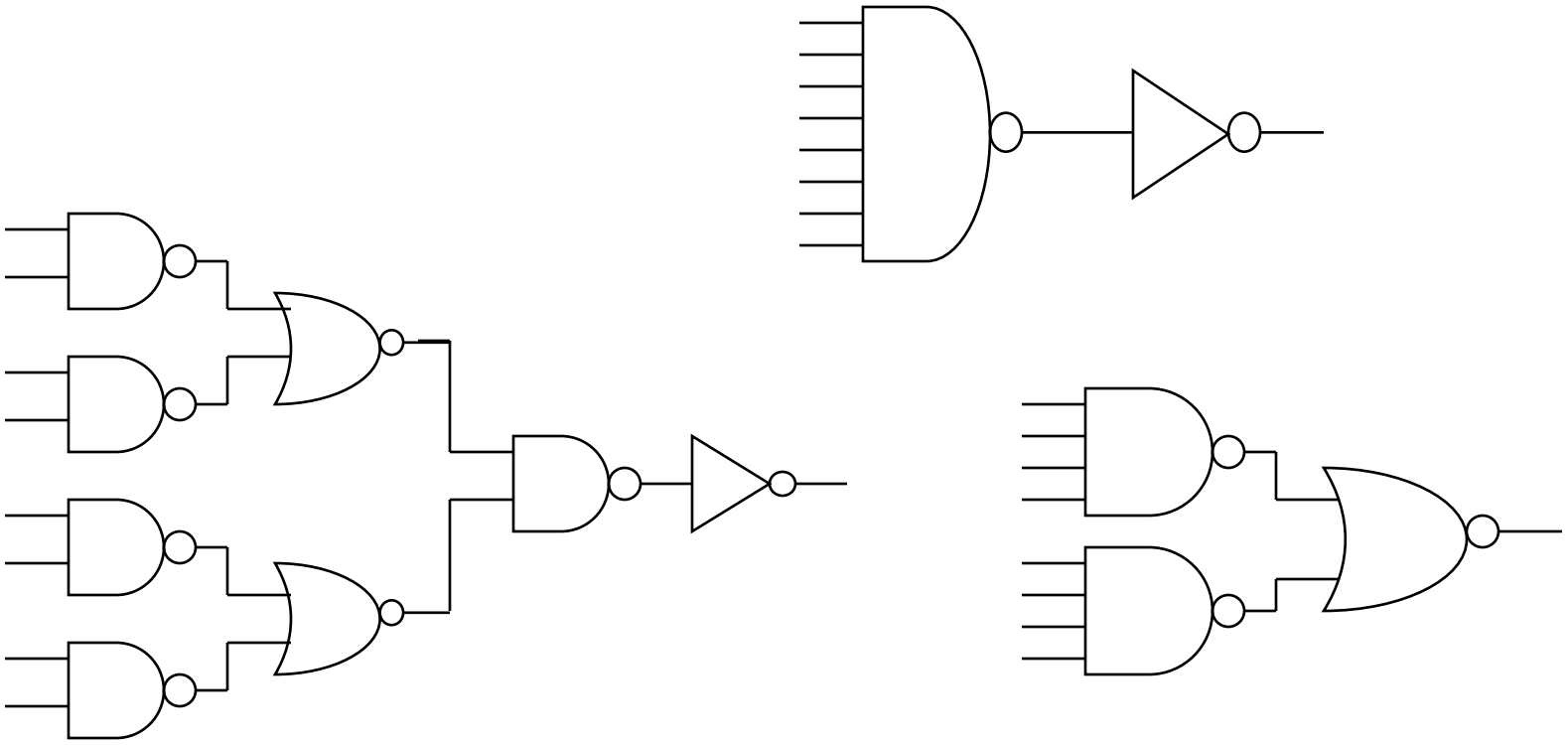
critical path A – 23%

critical path D – 17%

Fast Complex Gates: Design Technique 3

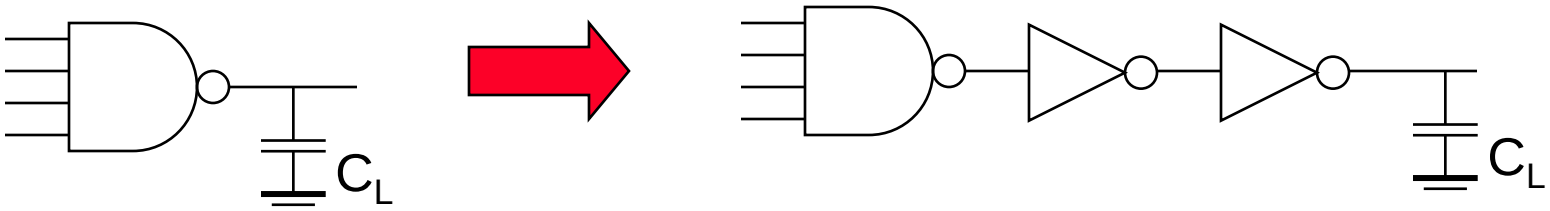
- Alternative logic structures

F = ABCDEFGH



Fast Complex Gates: Design Technique 4

- Isolating fan-in from fan-out using buffer insertion



- Real lesson is that optimizing the propagation delay of a gate in isolation is misguided.

Fast Networks: Design Technique 5 - Logical Effort

- The optimum fan-out for a chain of N inverters driving a load C_L is

$$f = \sqrt[N]{(C_L/C_{in})}$$

  - so, if we can, keep the fan-out per stage around 4.

- Can the same approach (logical effort) be used for any combinational circuit?
  - For a complex gate, we expand the inverter equation

$$t_p = t_{p0}\,(1 + C_{ext}/\gamma C_g) = t_{p0}\,(1 + f/\gamma)$$

to

$$t_p = t_{p0}\,(p + g\,f/\gamma)$$

- t_{p0} is the intrinsic delay of an inverter
- f is the effective fan-out (C_{ext}/C_g) – also called the electrical effort
- p is the ratio of the instrinsic (unloaded) delay of the complex gate and a simple inverter (a function of the gate topology and layout style)
- g is the logical effort

Intrinsic Delay Term, p

- The more involved the structure of the complex gate, the higher the intrinsic delay compared to an inverter

Gate Type	p
Inverter	1
n-input NAND	n
n-input NOR	n

Ignoring second order effects such as internal node capacitances

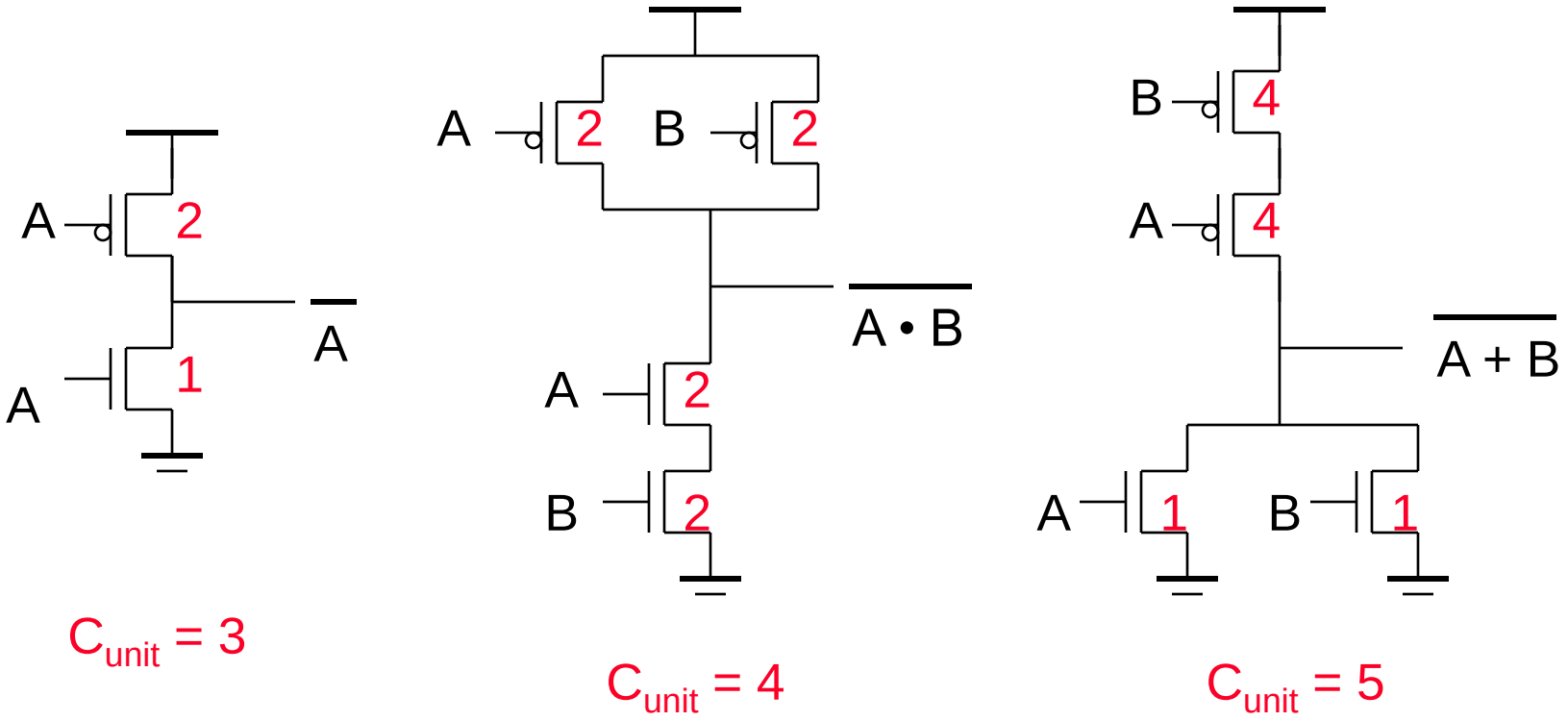
Logical Effort Term, g

- g represents the fact that, for a given load, complex gates have to work harder than an inverter to produce a similar (speed) response
  - the logical effort of a gate tells how much worse it is at producing an output current than an inverter (how much more input capacitance a gate presents to deliver it same output current)

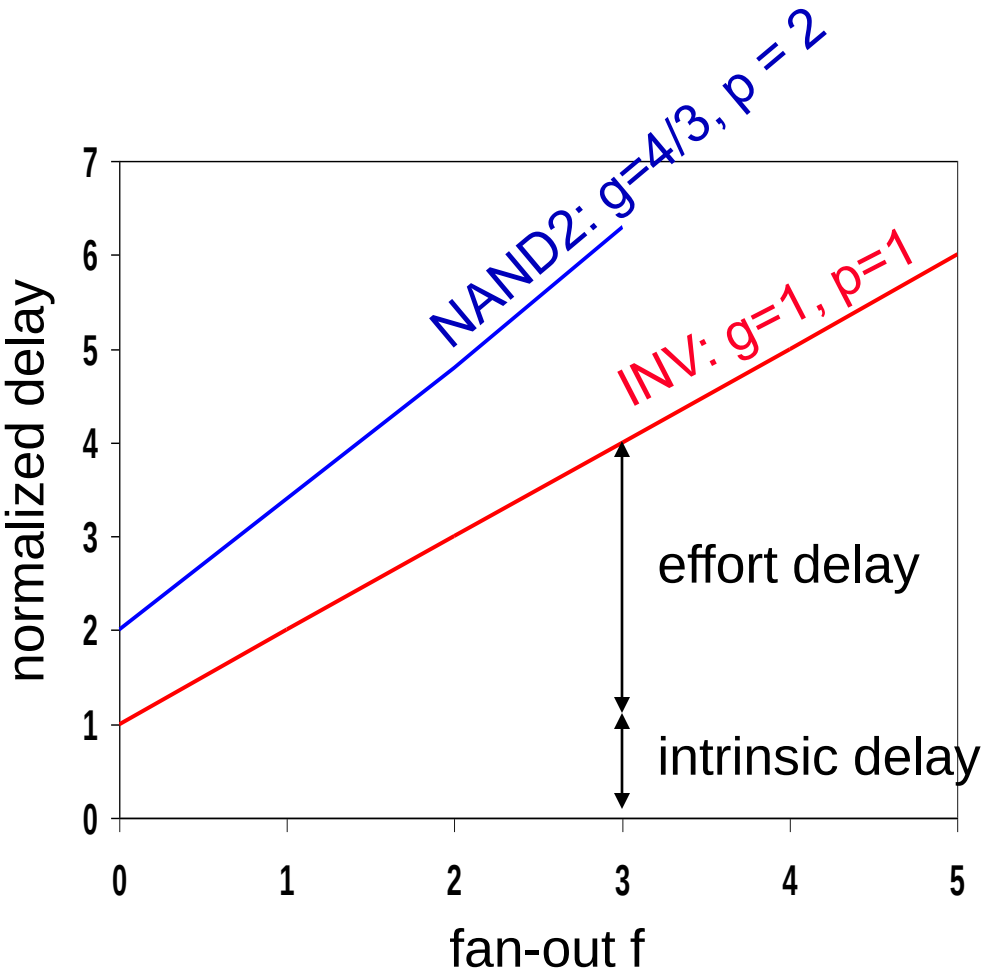
Gate Type	g (for 1 to 4 input gates)			
	1	2	3	4
Inverter	1			
NAND		4/3	5/3	(n+2)/3
NOR		5/3	7/3	(2n+1)/3

Example of Logical Effort

- Assuming a pmos/nmos ratio of 2, the input capacitance of a minimum-sized inverter is three times the gate capacitance of a minimum-sized nmos (C_{unit})



Delay as a Function of Fan-Out



- The slope of the line is the logical effort of the gate
- The y-axis intercept is the intrinsic delay
- Can adjust the delay by adjusting the effective fan-out (by sizing) or by choosing a gate with a different logical effort
- Gate effort: h = fg

Path Delay of Complex Logic Gate Network

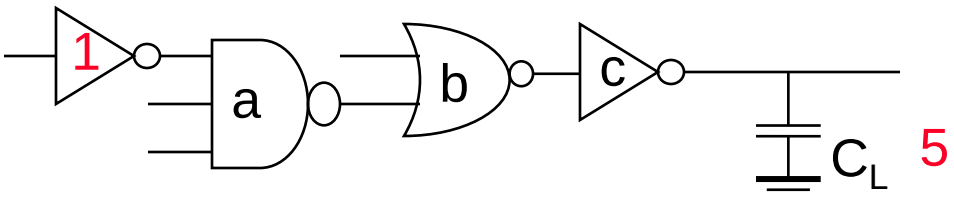
- Total path delay through a combinational logic block

$$t_p = \sum t_{p,j} = t_{p0} \sum(p_j + (f_j\, g_j)/\gamma\ )$$

- So, the minimum delay through the path determines that each stage should bear the same gate effort

$$f_1 g_1 = f_2 g_2 = \ldots = f_N g_N$$

- Consider optimizing the delay through the logic network



how do we determine a, b, and c sizes?

Path Delay Equation Derivation

- The path logical effort, $G = \prod g_i$
- And the path effective fan-out (path electrical effort) is $F = C_L/g_1$
- The branching effort accounts for fan-out to other gates in the network

$$b = (C_{on\text{-}path} + C_{off\text{-}path})/C_{on\text{-}path}$$

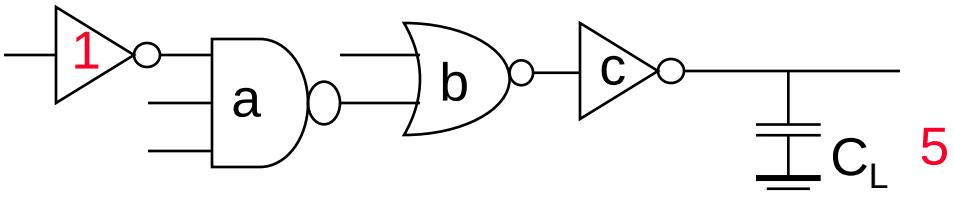
- The path branching effort is then $B = \prod b_i$
- And the total path effort is then $H = GFB$
- So, the minimum delay through the path is

$$D = t_{p0} \left(\sum p_j + (N \sqrt[N]{H})/ \gamma \right)$$

Path Delay of Complex Logic Gates, con't

- For gate i in the chain, its size is determined by

$$s_i = (g_1 s_1)/g_i \prod_{j=1}^{i-1} (f_j/b_j)$$



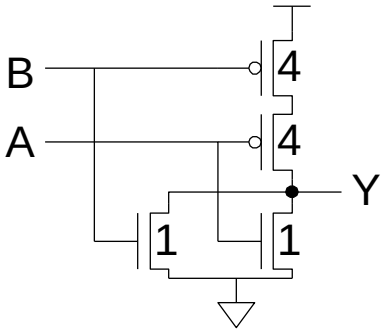
- For this network
  - $F = C_L/C_{g1} = 5$
  - $G = 1 \times 5/3 \times 5/3 \times 1 = 25/9$
  - $B = 1$ (no branching)
  - $H = GFB = 125/9$, so the optimal stage effort is $\sqrt[4]{H} = 1.93$
    - Fan-out factors are f_1=1.93, f_2=1.93 x 3/5 = 1.16, f_3 = 1.16, f_4 = 1.93
  - So the gate sizes are $a = f_1 g_1/g_2 = 1.16$, $b = f_1 f_2 g_1/g_3 = 1.34$ and $c = f_1 f_2 f_3 g_1/g_4 = 2.60$

منطق نسبتی:

Ratioed logic

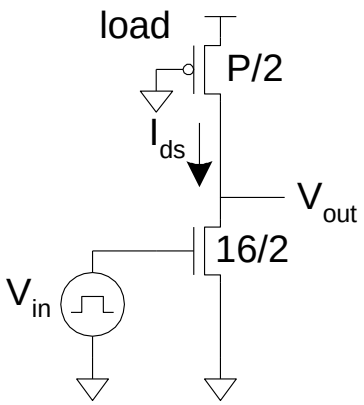
Introduction

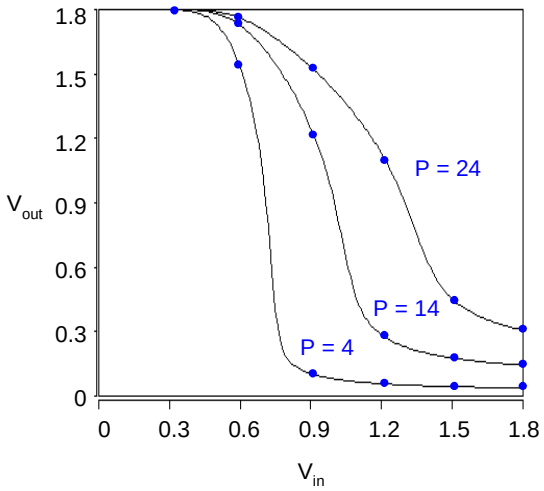
- What makes a circuit fast?
  - $I = C\, dV/dt \quad \rightarrow \quad t_{pd} \propto (C/I)\, \Delta V$
  - low capacitance
  - high current
  - small swing
- Logical effort is proportional to C/I
- pMOS are the enemy!
  - High capacitance for a given current
- Can we take the pMOS capacitance off the input?
- Various circuit families try to do this…



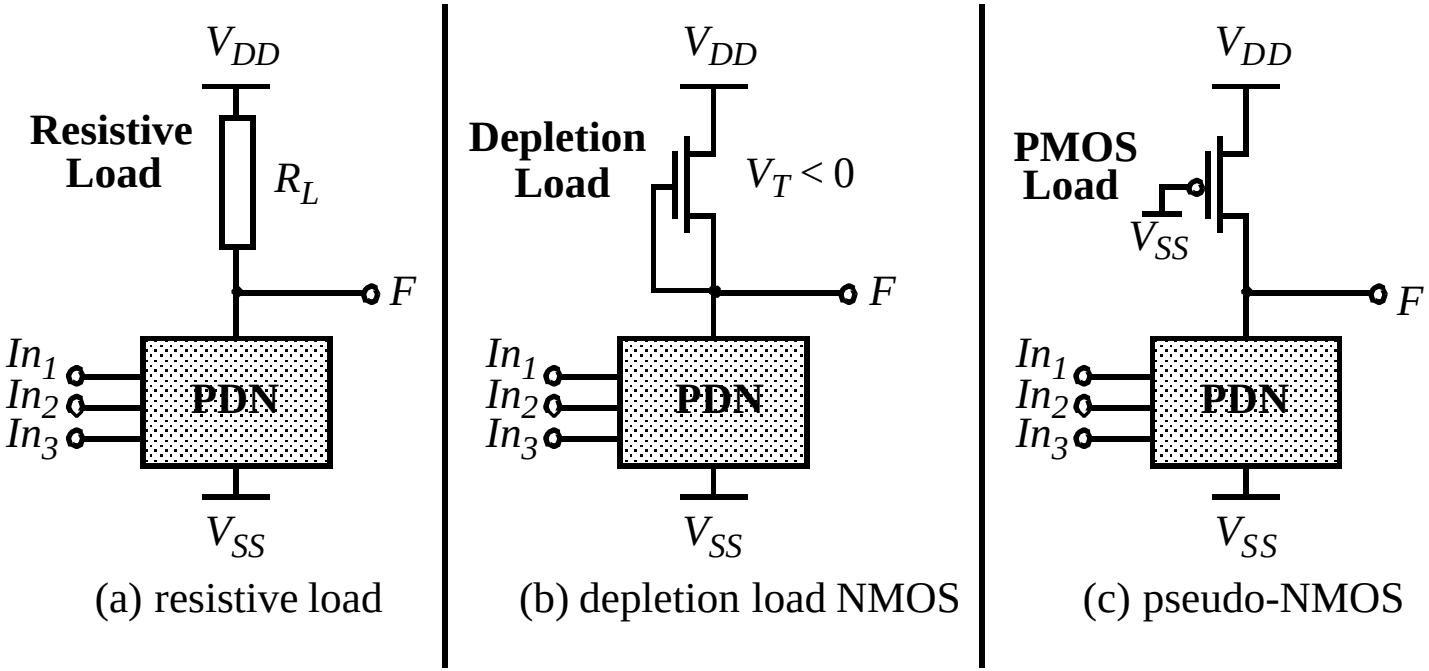
Pseudo-nMOS

- In the old days, nMOS processes had no pMOS
  - Instead, use pull-up transistor that is always ON
- In CMOS, use a pMOS that is always ON
  - *Ratio* issue
  - Make pMOS about ¼ effective strength of pulldown network



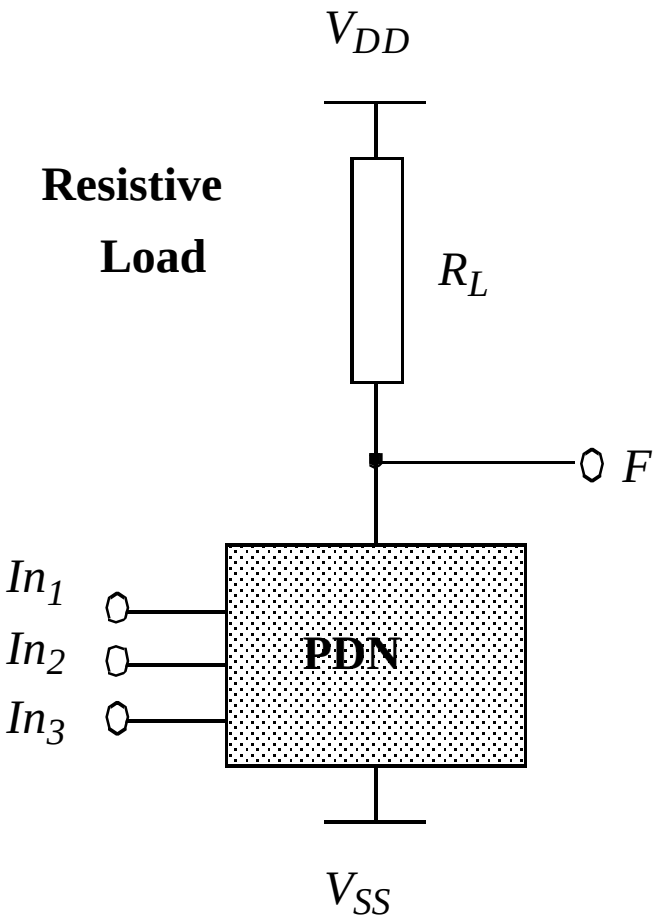


Ratioed Logic



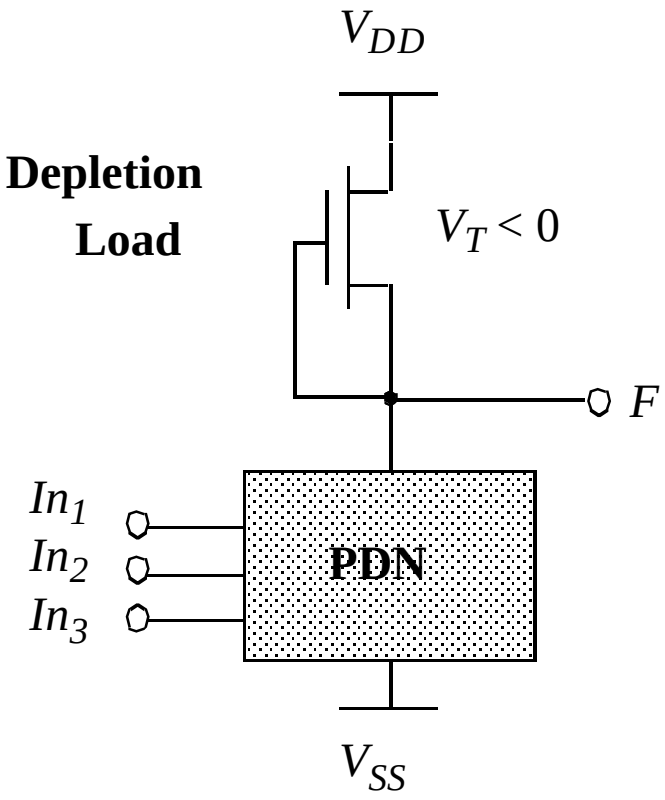
Goal: to reduce the number of devices over complementary CMOS

Ratioed Logic

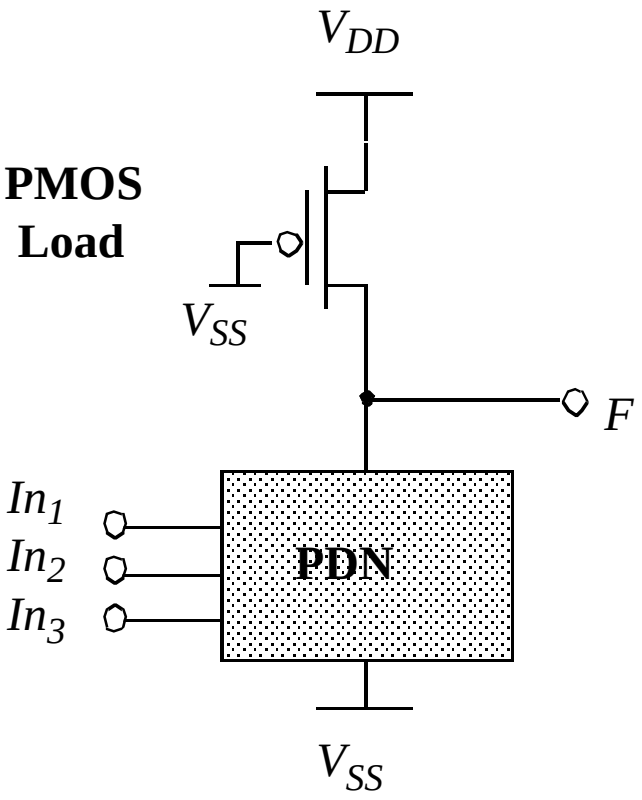


- N transistors + Load
- $V_{OH} = V_{DD}$
- $V_{OL} = \dfrac{R_{PN}}{R_{PN} + R_L}$
- Assymetrical response
- Static power consumption
- $t_{pL} = 0.69\, R_L C_L$

Active Loads



depletion load NMOS



pseudo-NMOS

VOL

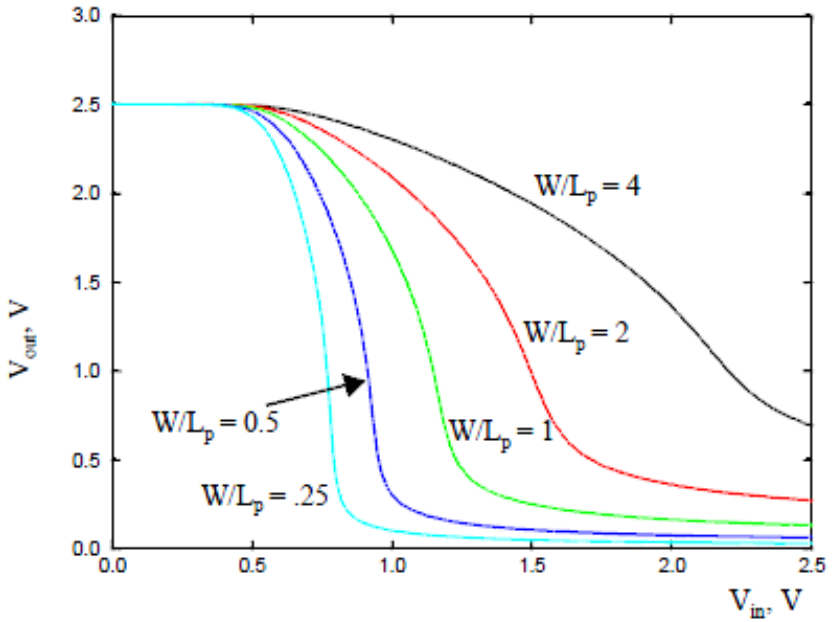
it is reasonable to assume that the NMOS device resides in linear mode (since the output should ideally be close to 0V), while the PMOS load is saturated.

$$k_n\left((V_{DD}-V_{Tn})V_{OL}-\frac{V_{OL}^2}{2}\right)=k_p\left((-V_{DD}-V_{Tp})\cdot V_{DSAT}-\frac{V_{DSAT}^2}{2}\right)$$

$$V_{OL}\approx\frac{k_p(-V_{DD}-V_{Tp})\cdot V_{DSAT}}{k_n(V_{DD}-V_{Tn})}\approx\frac{\mu_p\cdot W_p}{\mu_n\cdot W_n}\cdot|V_{DSAT}|$$

In order to make V_{OL} as small as possible, the PMOS device should be sized much smaller than the NMOS pull-down devices. Unfortunately, this has a negative impact on the *propagation delay* for charging up the output node since the current provided by the PMOS device is limited.

Pseudo-NMOS VTC

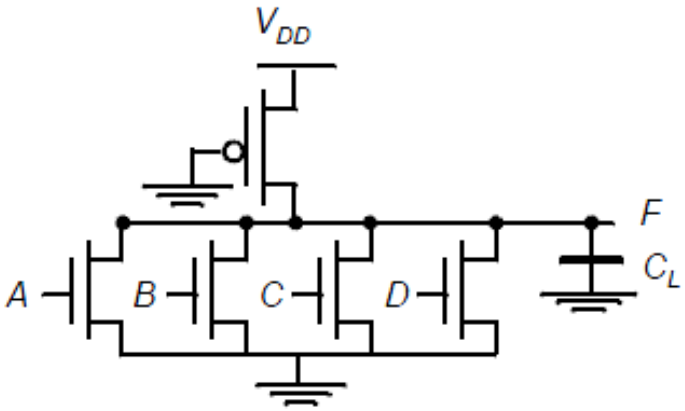


Voltage-transfer curves of the pseudo-NMOS inverter as a function of the PMOS size.

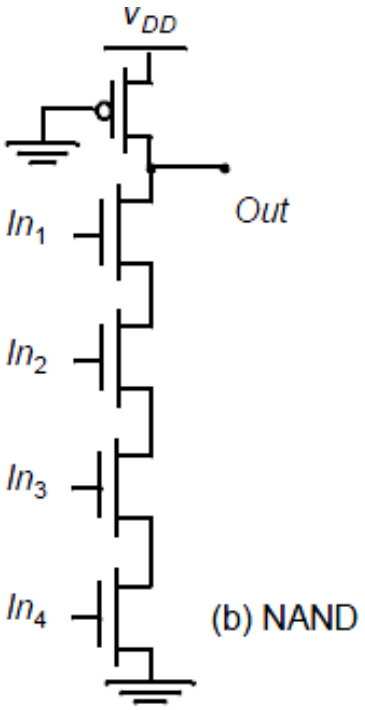
Performance of a pseudo-NMOS inverter.

Size	V_{OL}	Static Power Dissipation	t_{plh}
4	0.693V	564μW	14ps
2	0.273V	298μW	56ps
1	.133V	160μW	123ps
0.5	0.064V	80μW	268ps
0.25	0.031V	41μW	569ps

NAND or NOR (Large Fan in)

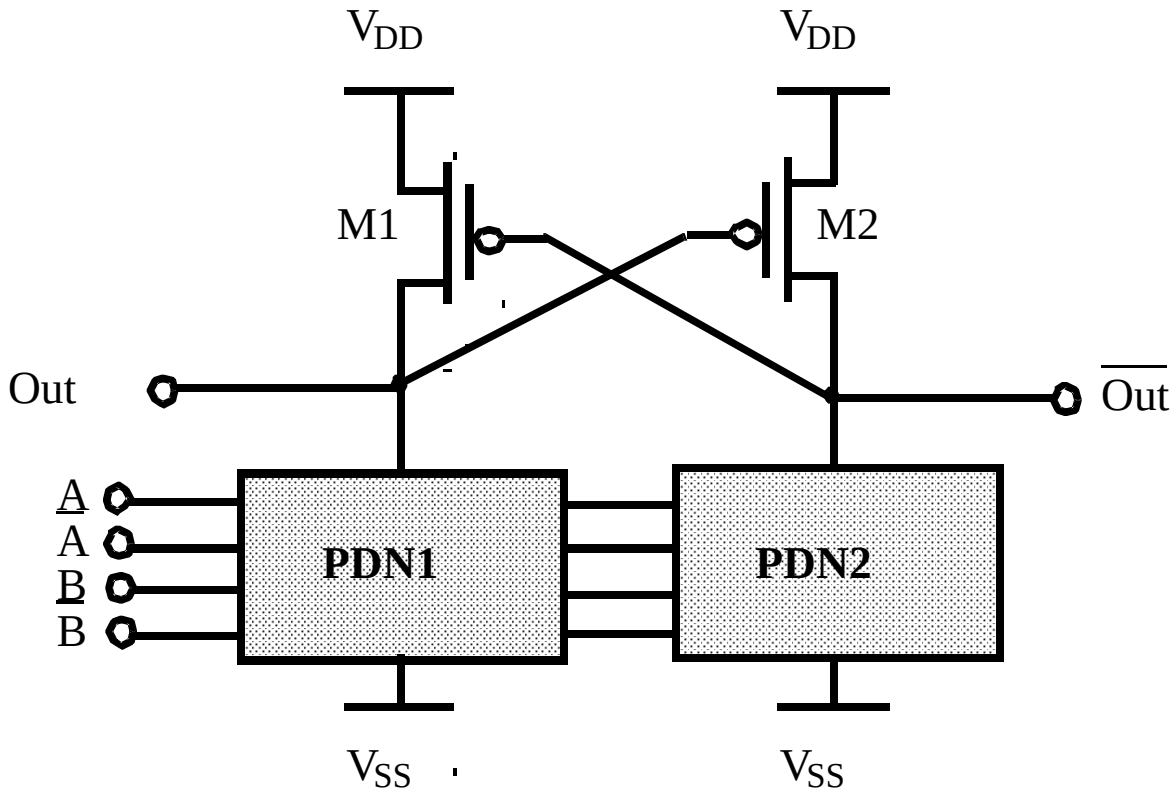


(a) NOR



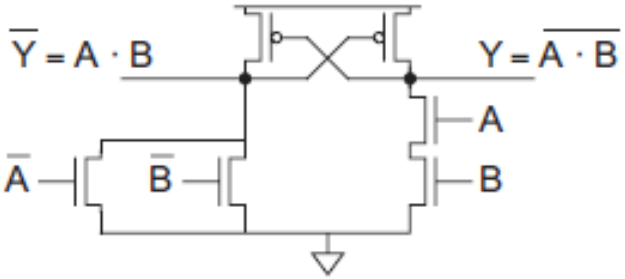
(b) NAND

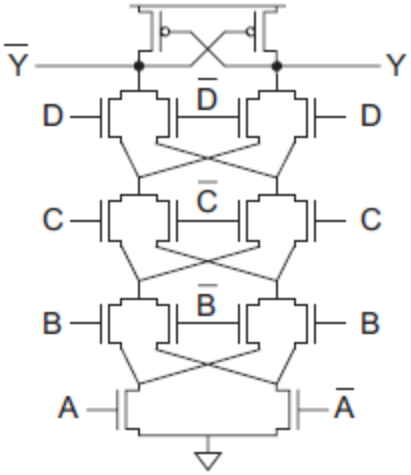
Improved Load



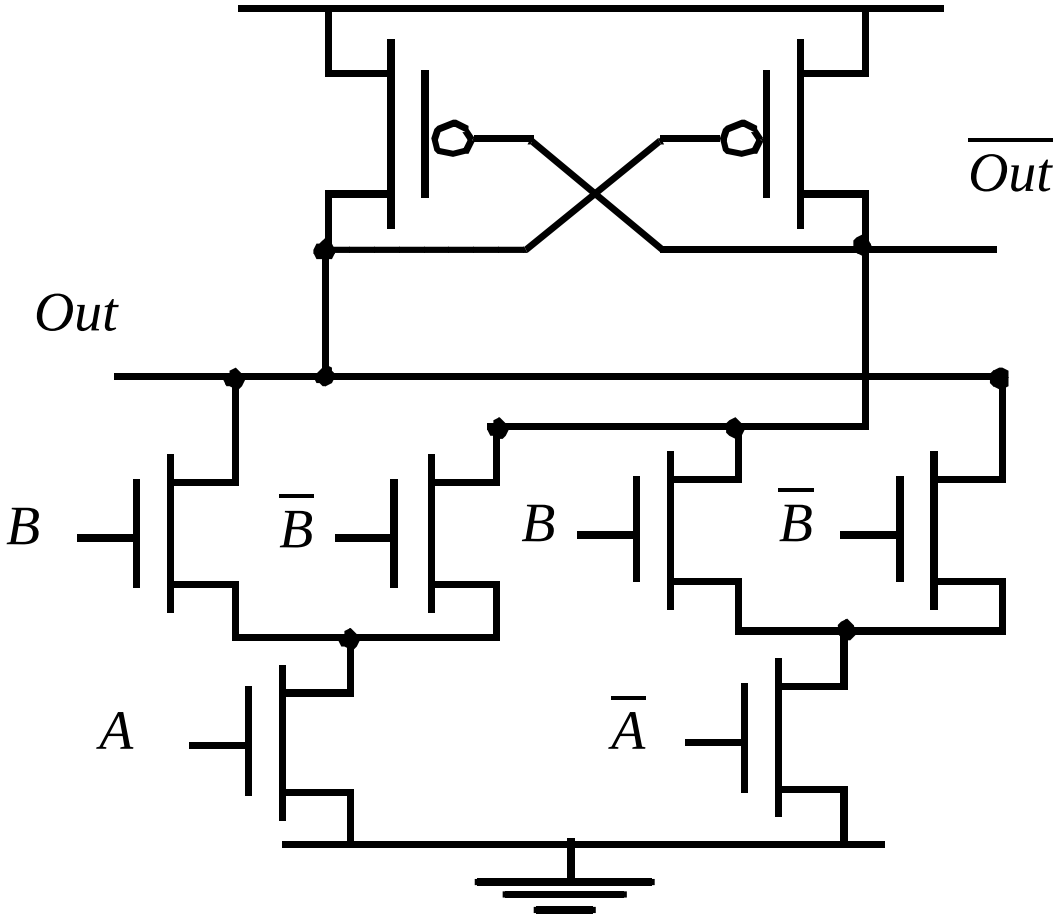
Differential Cascode Voltage Switch Logic (DCVSL)

CVSL has a potential speed advantage because all of the logic is performed with nMOS transistors, thus reducing the input capacitance. As in pseudo-nMOS, the size of the pMOS transistor is important. It fights the pulldown network, so a large pMOS transistor will slow the falling transition. Unlike pseudo-nMOS, the feedback tends to turn off the pMOS, so the outputs will settle eventually to a legal logic level.



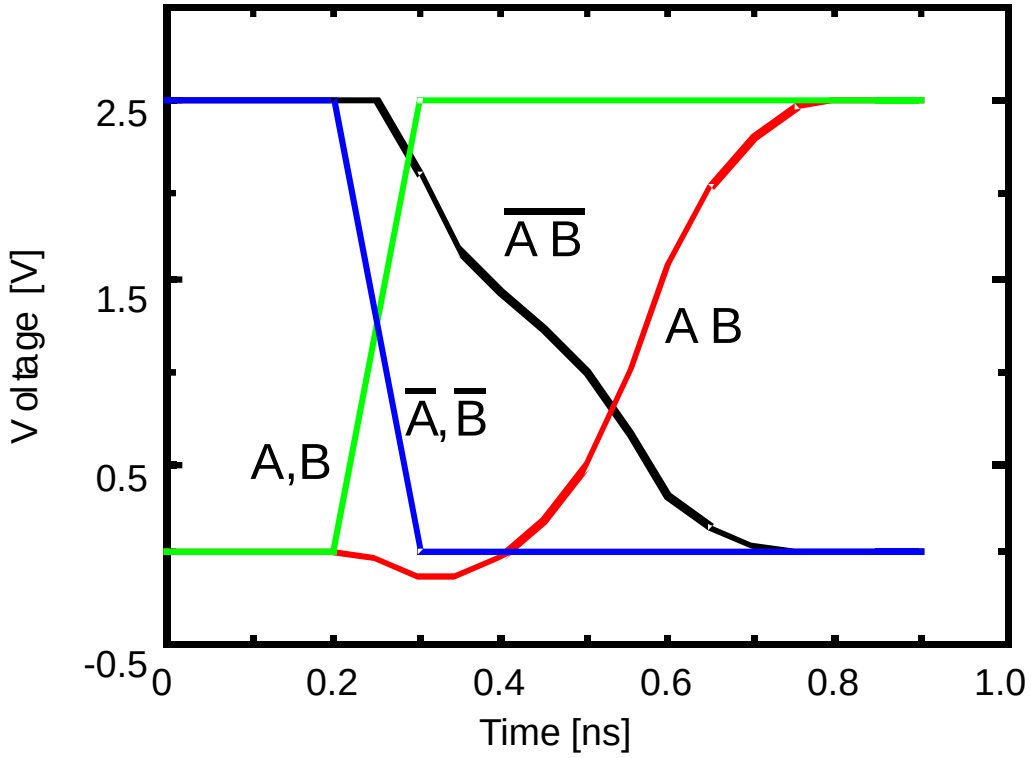


DCVSL Example



XOR-NXOR gate

DCVSL Transient Response

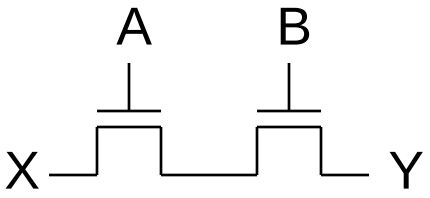


منطق ترانزیستور عبوری

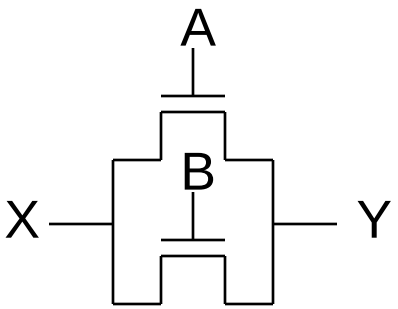
Pass-Transistor Logic

NMOS Transistors in Series/Parallel

- Primary inputs drive both gate and source/drain terminals
- NMOS switch closes when the gate input is high



X = Y if A and B

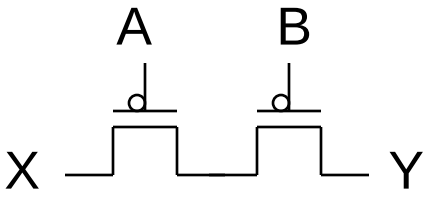


X = Y if A or B

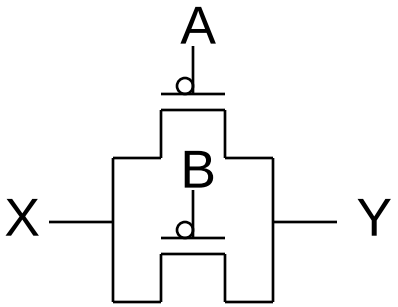
- Remember - NMOS transistors pass a strong 0 but a weak 1

PMOS Transistors in Series/Parallel

- Primary inputs drive both gate and source/drain terminals
- PMOS switch closes when the gate input is low



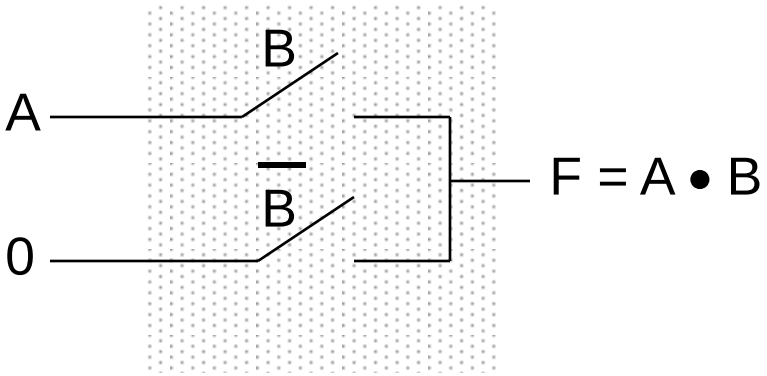
$X = Y$ if $\overline{A}$ and $\overline{B} = \overline{A + B}$

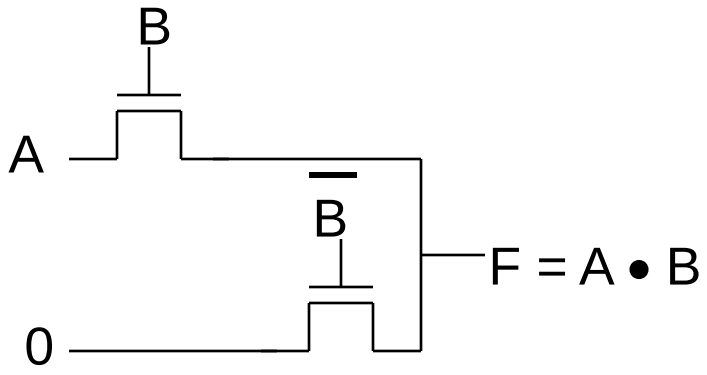


$X = Y$ if $\overline{A}$ or $\overline{B} = \overline{A \bullet B}$

- Remember - PMOS transistors pass a strong 1 but a weak 0

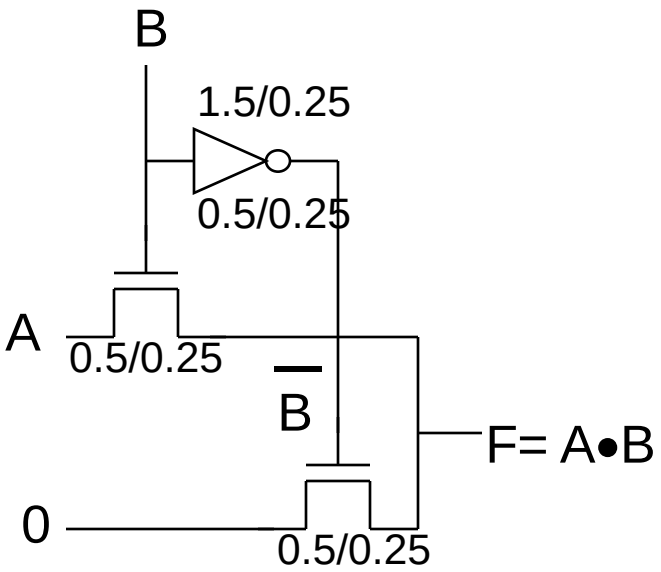
Pass Transistor (PT) Logic

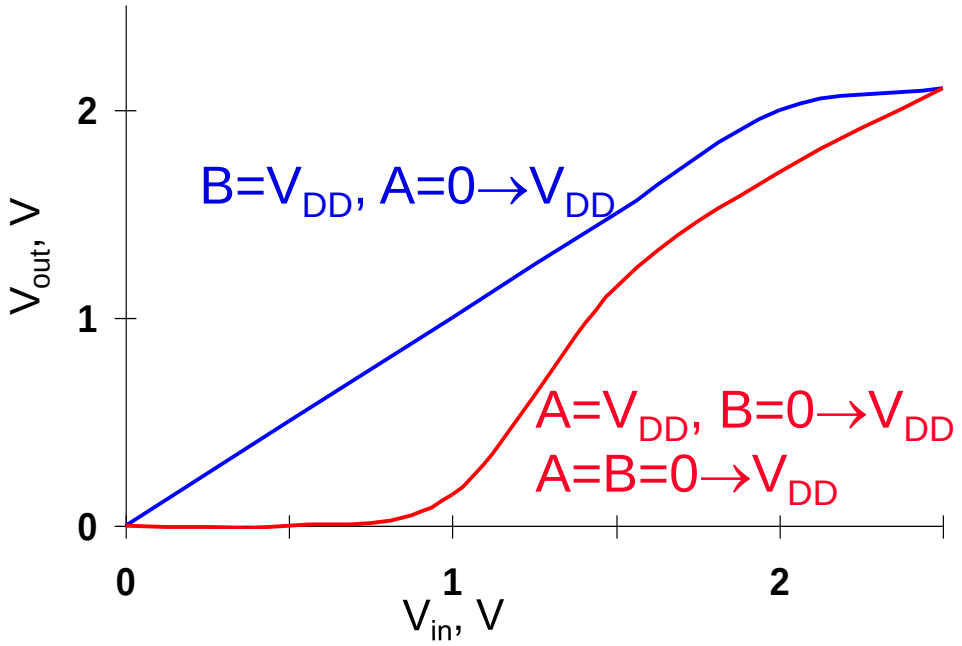




- Gate is static – a low-impedance path exists to both supply rails under all circumstances
- N transistors instead of 2N
- No static power consumption
- Ratioless
- Bidirectional (versus undirectional)

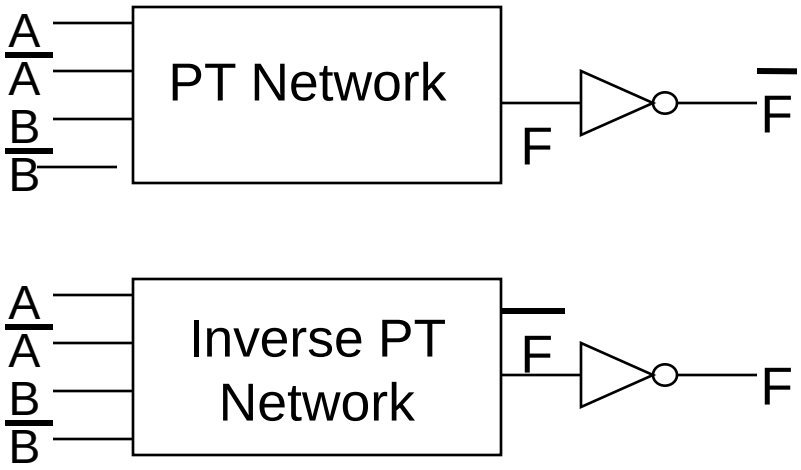
VTC of PT AND Gate

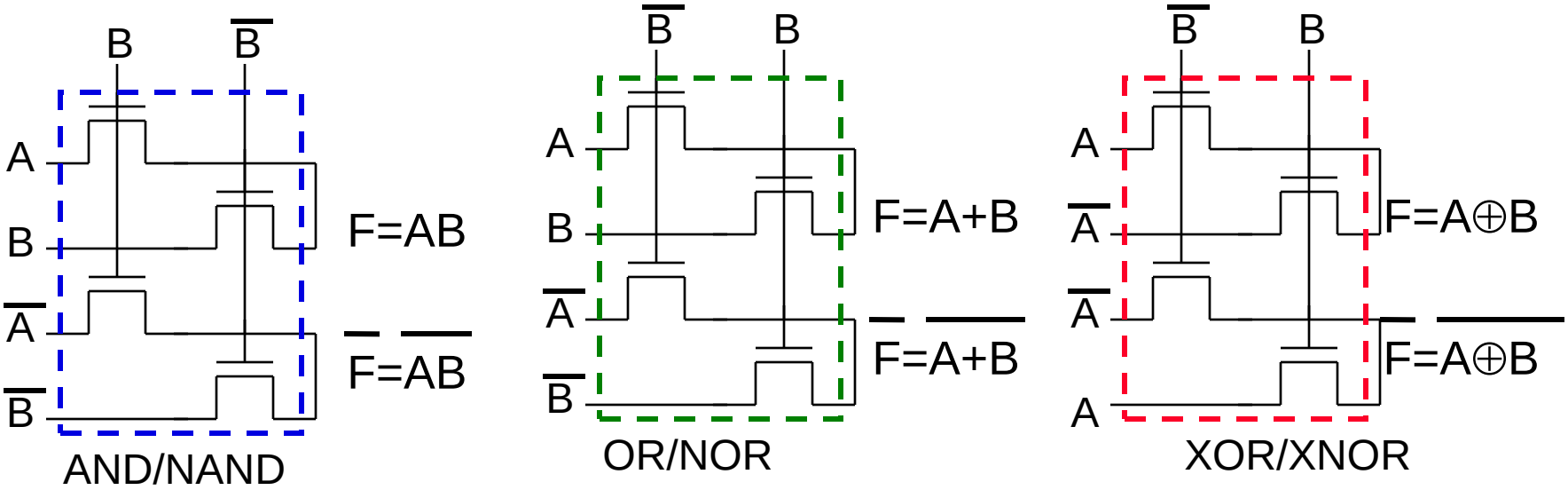




- Pure PT logic is not regenerative - the signal gradually degrades after passing through a number of PTs (can fix with static CMOS inverter insertion)

Differential PT Logic (CPL)



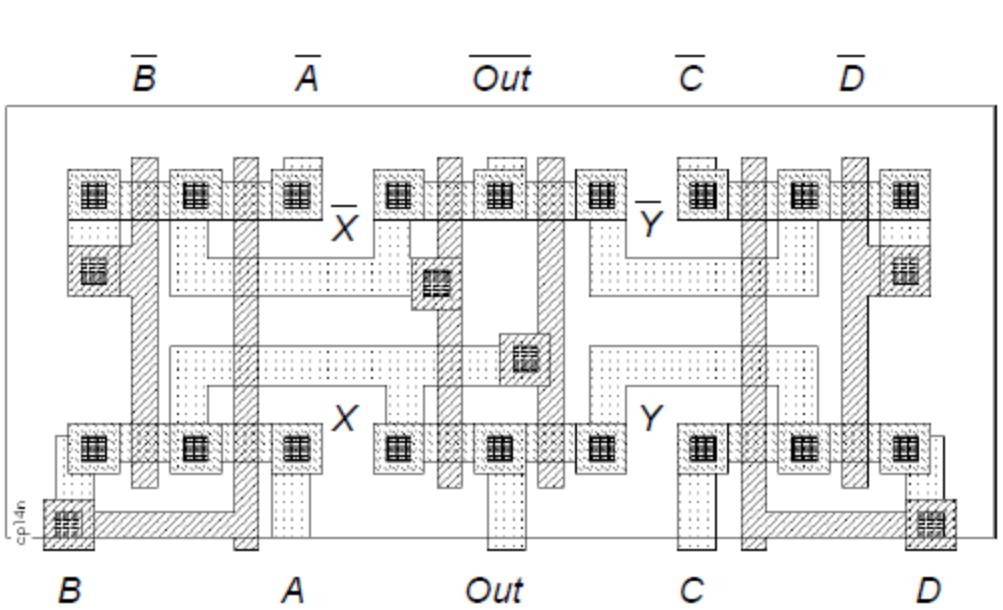


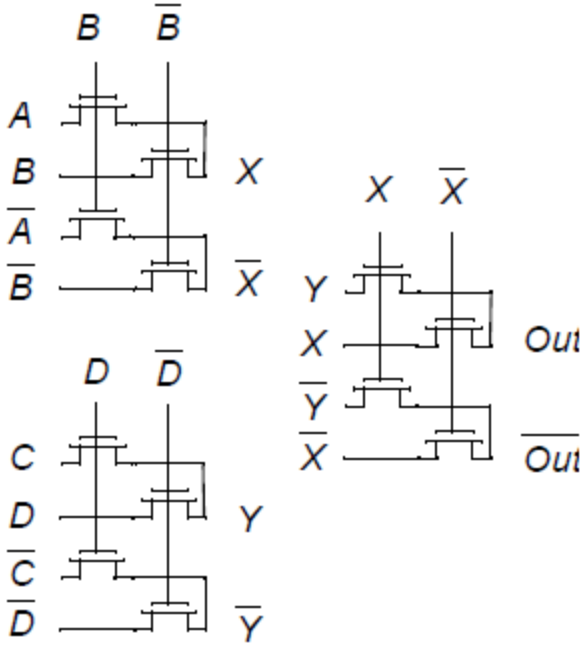
CPL Properties

- Differential so complementary data inputs and outputs are always available (so don't need extra inverters)
- Still static, since the output defining nodes are always tied to V_{DD} or GND through a low resistance path
- Design is modular; all gates use the same topology, only the inputs are permuted.
- Simple XOR makes it attractive for structures like adders
- Fast (assuming number of transistors in series is small)
- Additional routing overhead for complementary signals
- Still have static power dissipation problems

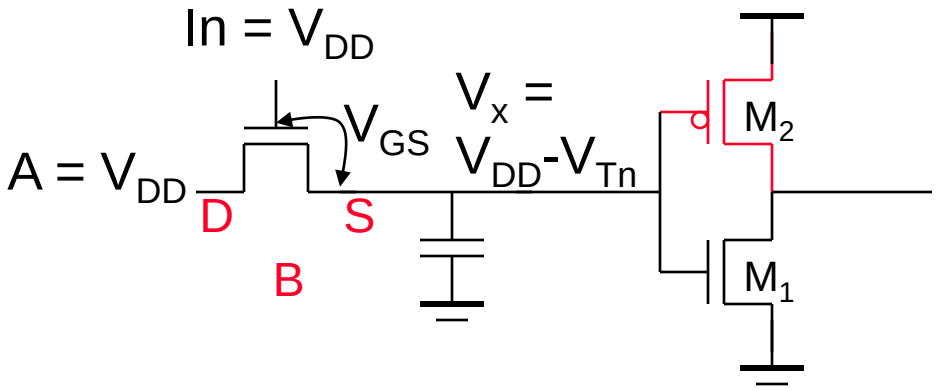
4 input NAND in CPL

four-input AND/NAND gate



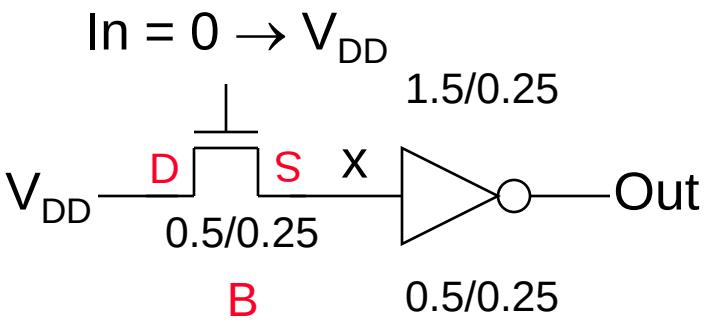


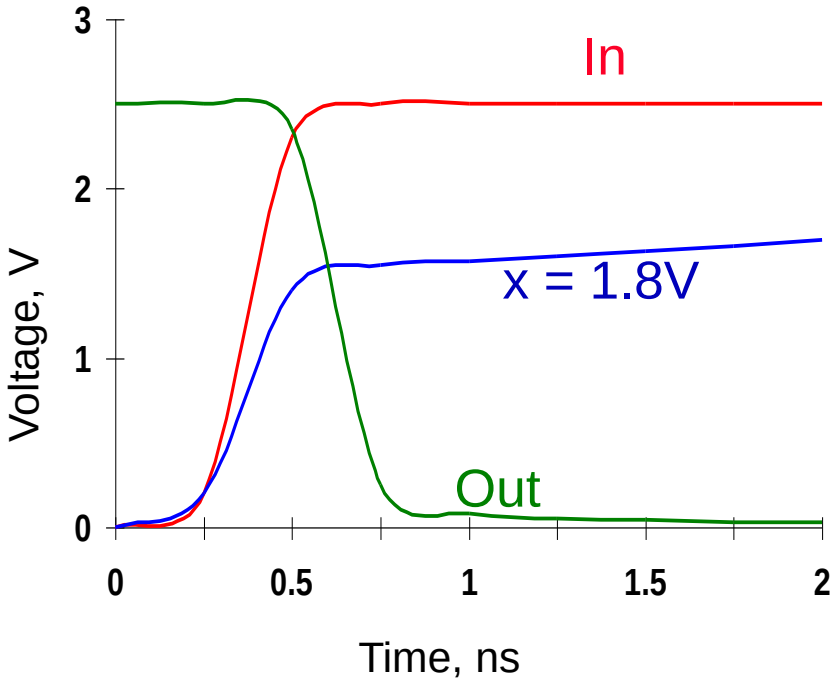
NMOS Only PT Driving an Inverter



- V_x does not pull up to V_{DD}, but $V_{DD} – V_{Tn}$
- Threshold voltage drop causes static power consumption (M_2 may be weakly conducting forming a path from V_{DD} to GND)
- Notice V_{Tn} increases of pass transistor due to body effect (V_{SB})

Voltage Swing of PT Driving an Inverter

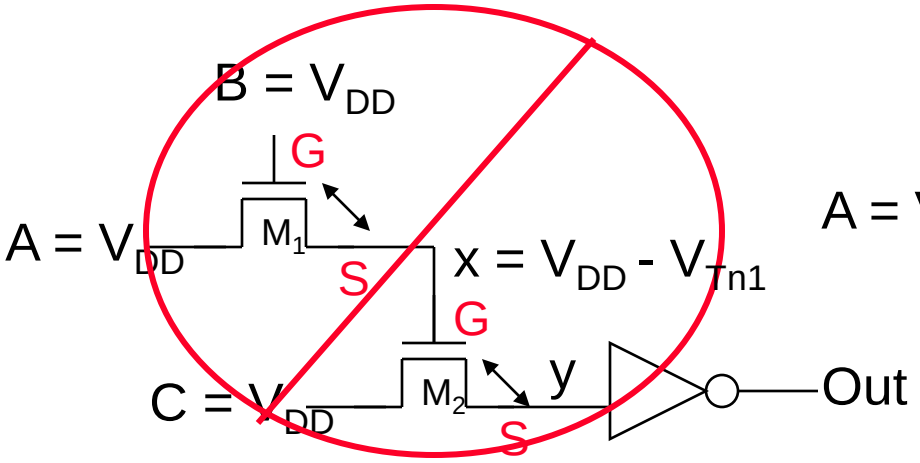


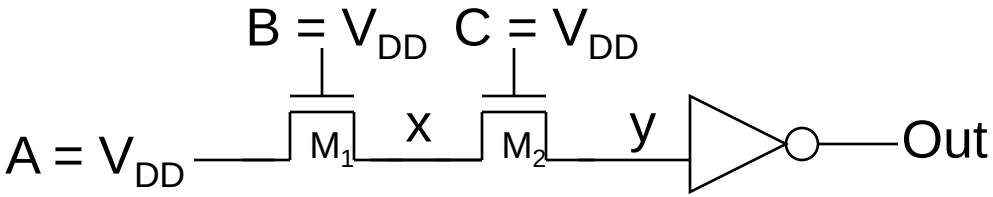


- Body effect – large V_{SB} at x - when pulling high (B is tied to GND and S charged up close to V_{DD})
- So the voltage drop is even worse

$$V_x = V_{DD} - (V_{Tn0} + \gamma(\sqrt{(|2\phi_f| + V_x)} - \sqrt{|2\phi_f|}))$$

Cascaded NMOS Only PTs



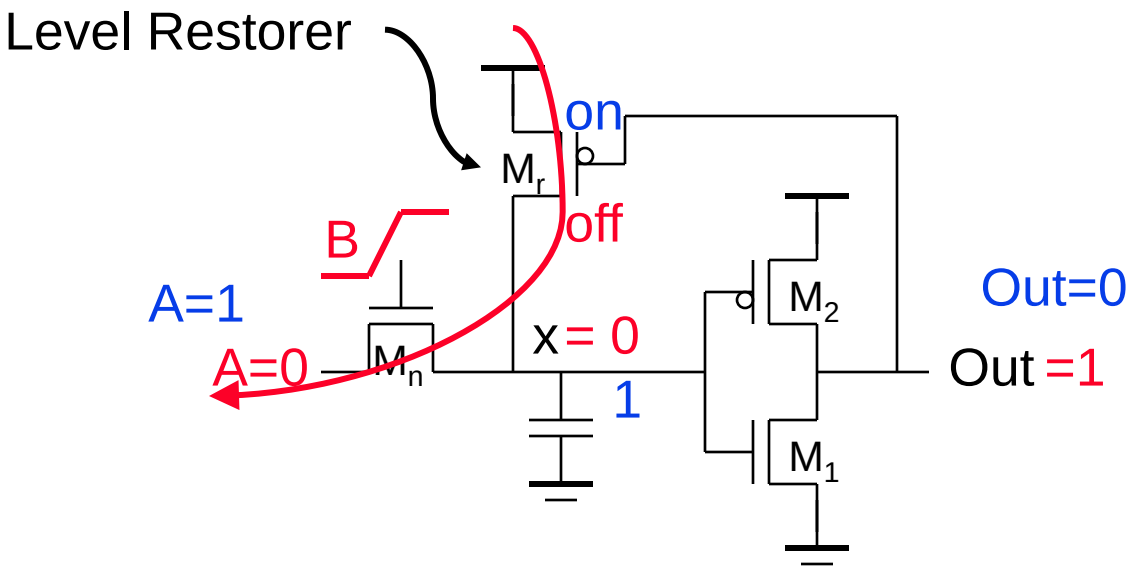


Swing on y = $V_{DD} - V_{Tn1} - V_{Tn2}$

Swing on y = $V_{DD} - V_{Tn1}$

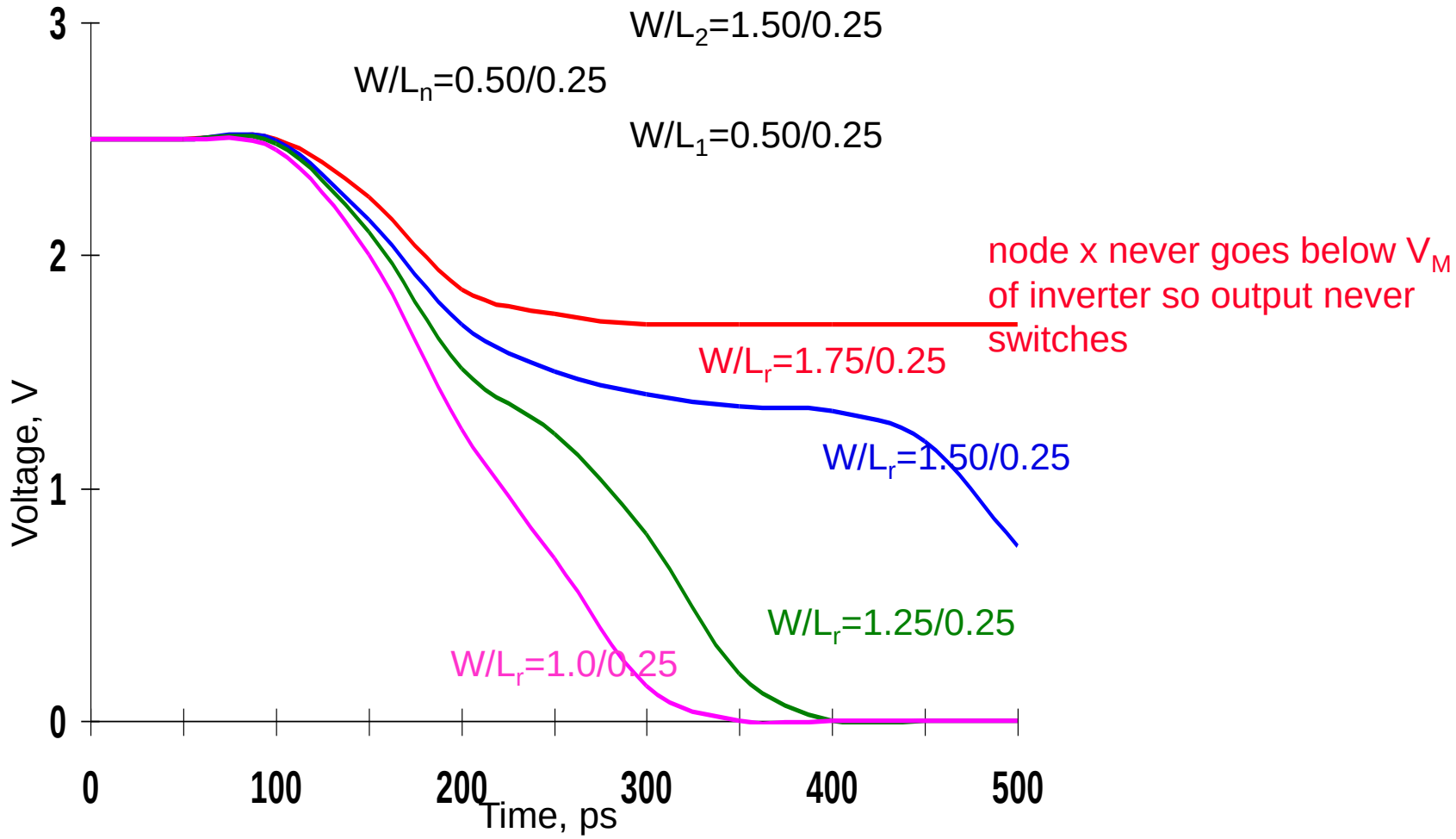
- Pass transistor gates should never be cascaded as on the left
- Logic on the right suffers from static power dissipation and reduced noise margins

Solution 1: Level Restorer



- Full swing on x (due to Level Restorer) so no static power consumption by inverter
- No static backward current path through Level Restorer and PT since Restorer is only active when A is high
- For correct operation M_r must be sized correctly (ratioed)

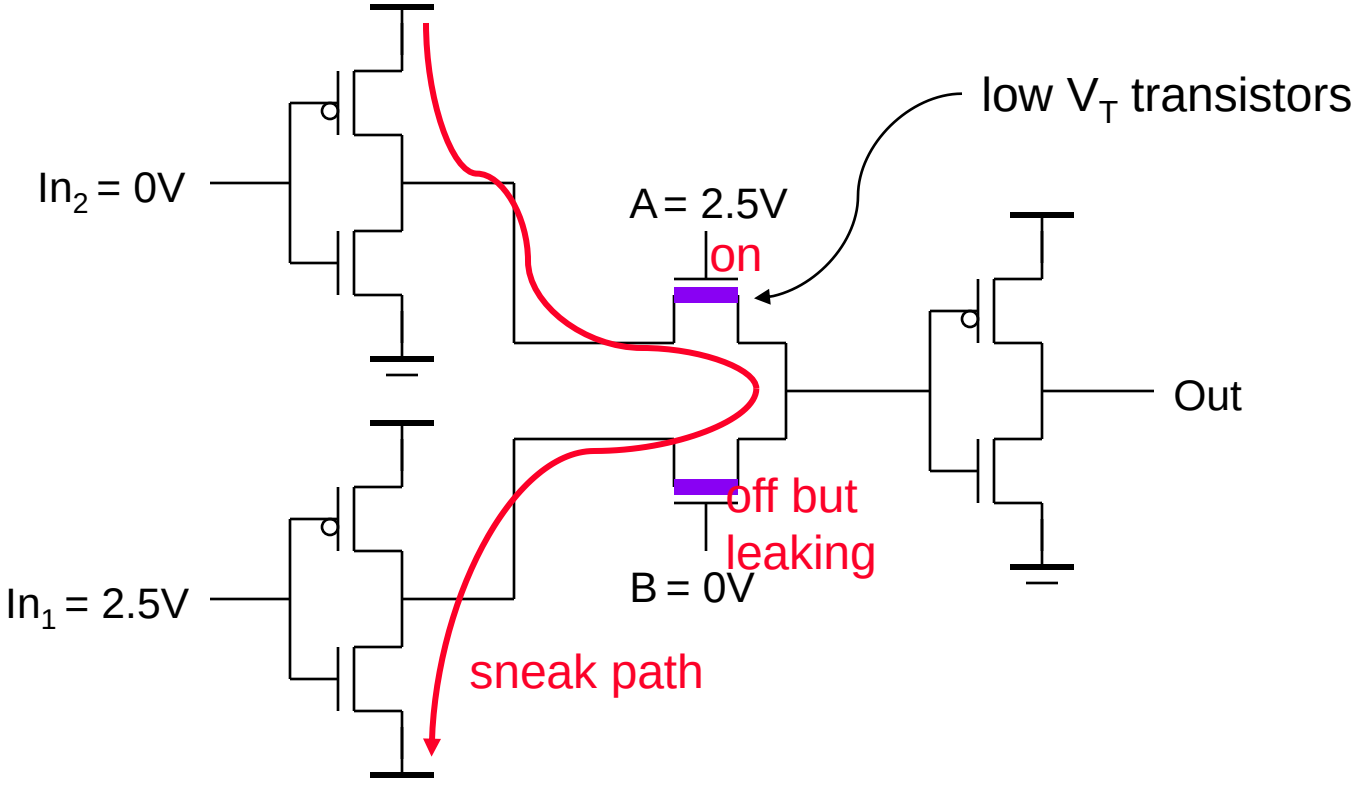
Transient Level Restorer Circuit Response



- Restorer has speed and power impacts: increases the capacitance at x, slowing down the gate; increases t_r (but decreases t_f)

Solution 2: Multiple V_T Transistors

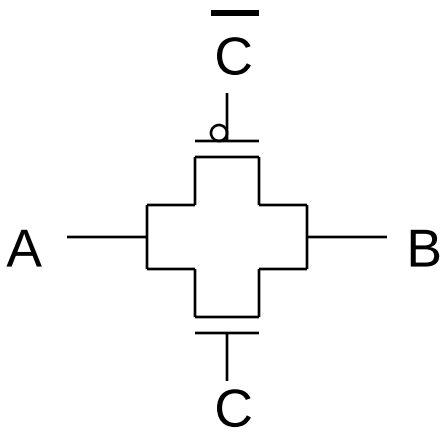
- Technology solution: Use (near) zero V_T devices for the NMOS PTs to eliminate *most* of the threshold drop (body effect still in force preventing full swing to V_{DD})

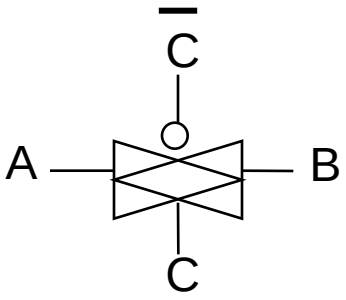


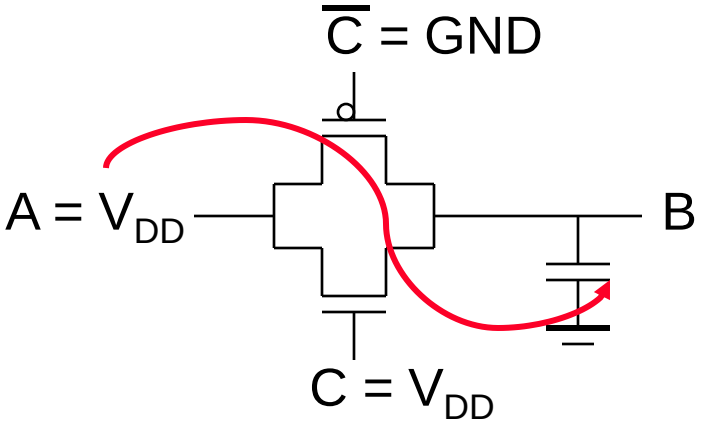
- Impacts static power consumption due to subthreshold currents flowing through the PTs (even if V_{GS} is below V_T)

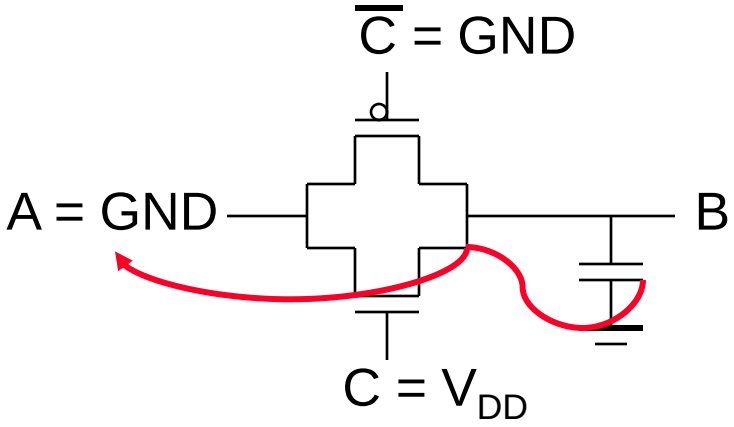
Solution 3: Transmission Gates (TGs)

- Most widely used solution



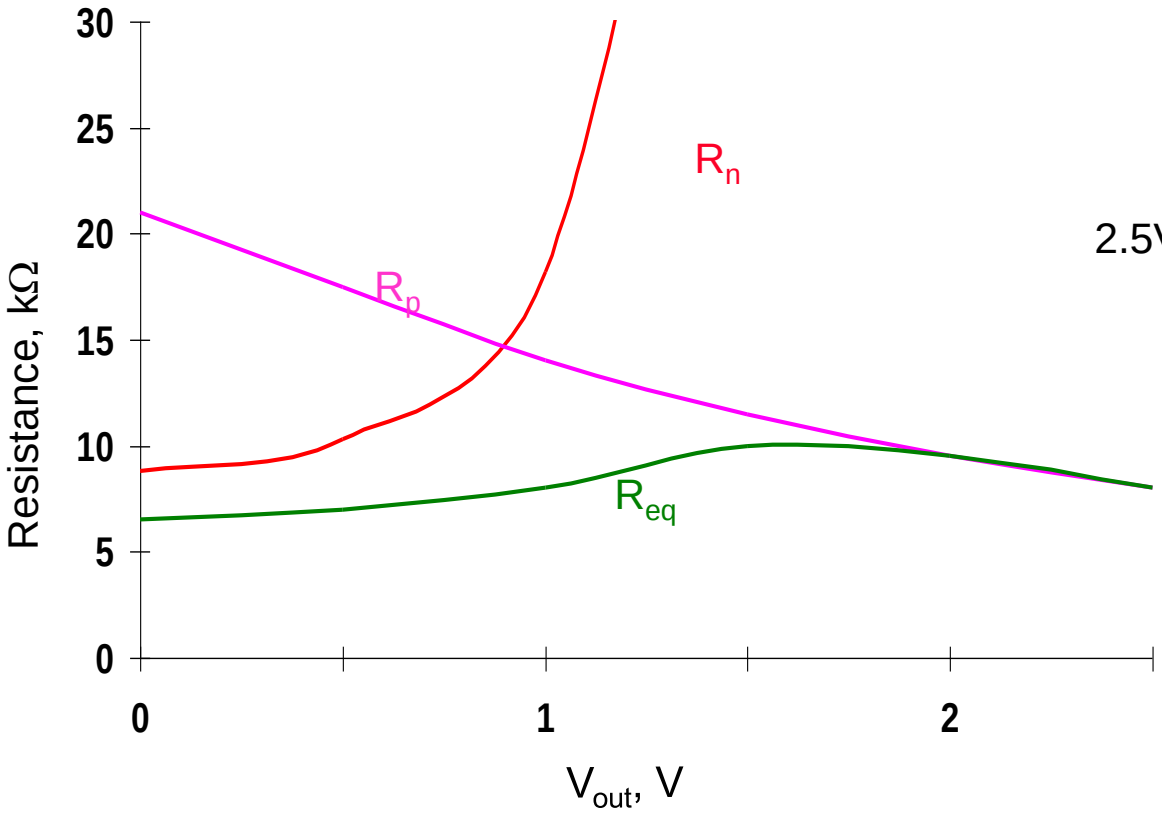


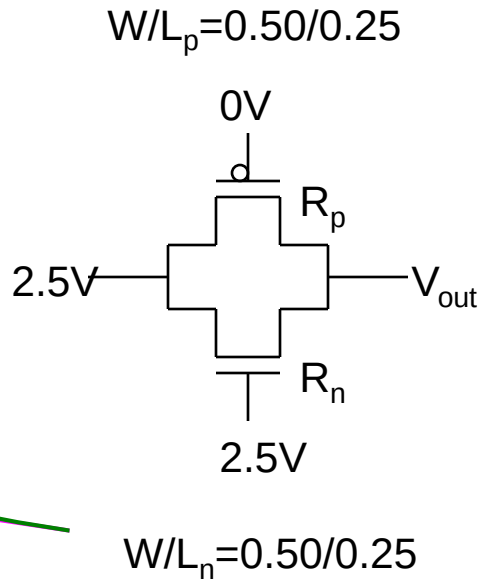




- Full swing *bidirectional* switch controlled by the gate signal C, A = B if C = 1

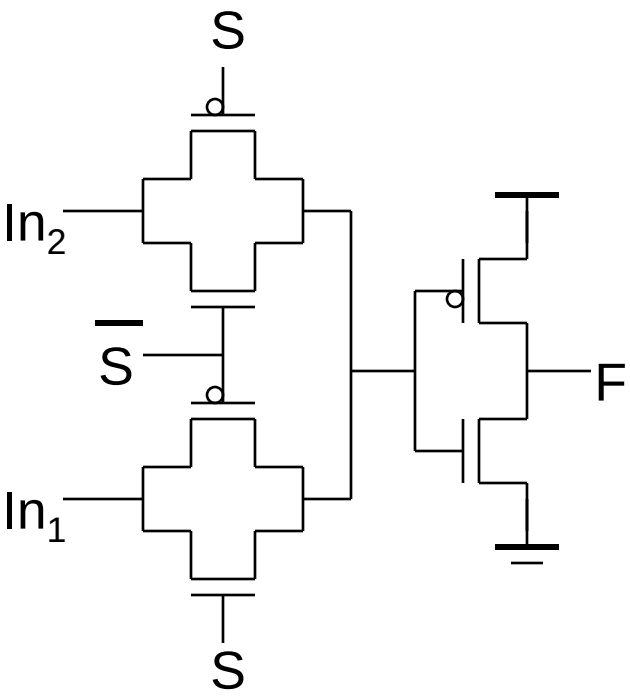
Resistance of TG



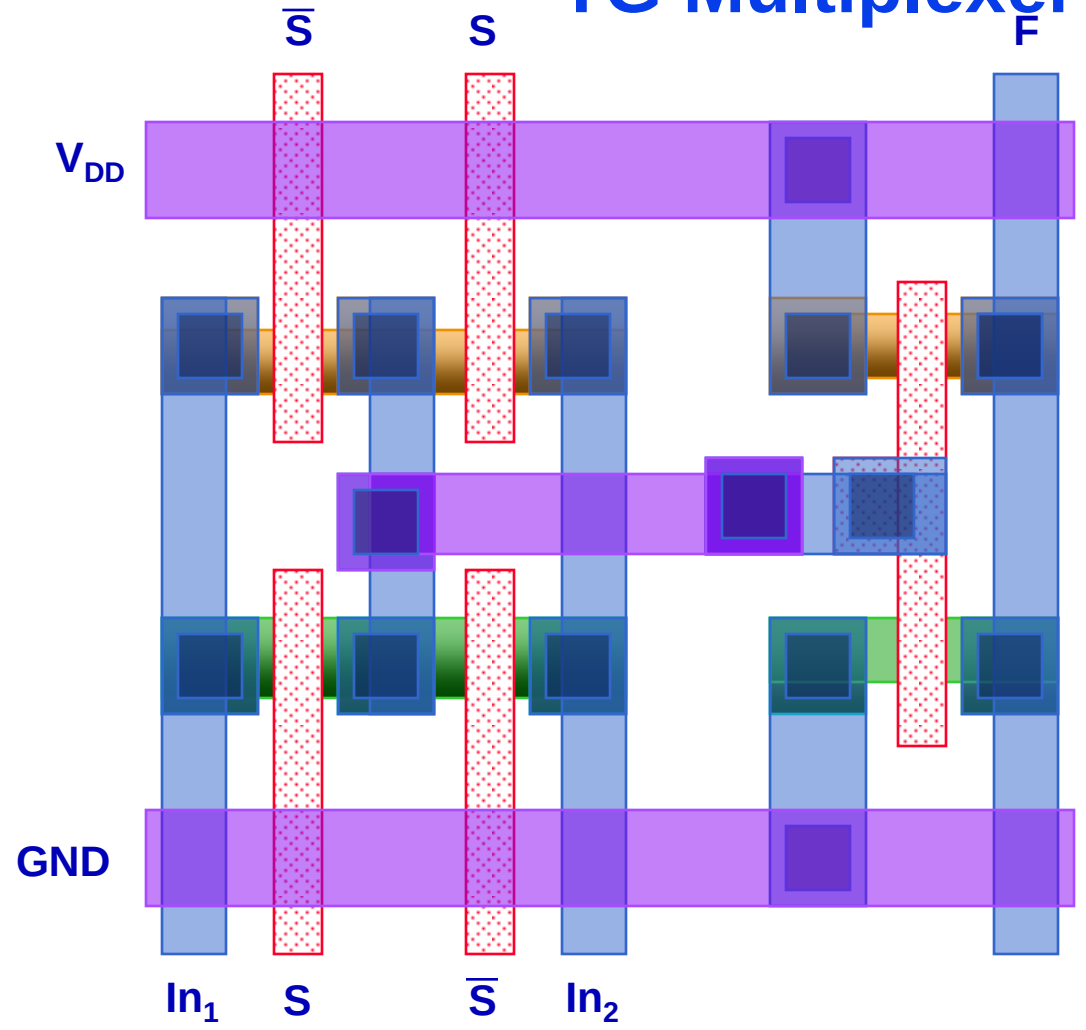


کاربرد های گیت انتقالی

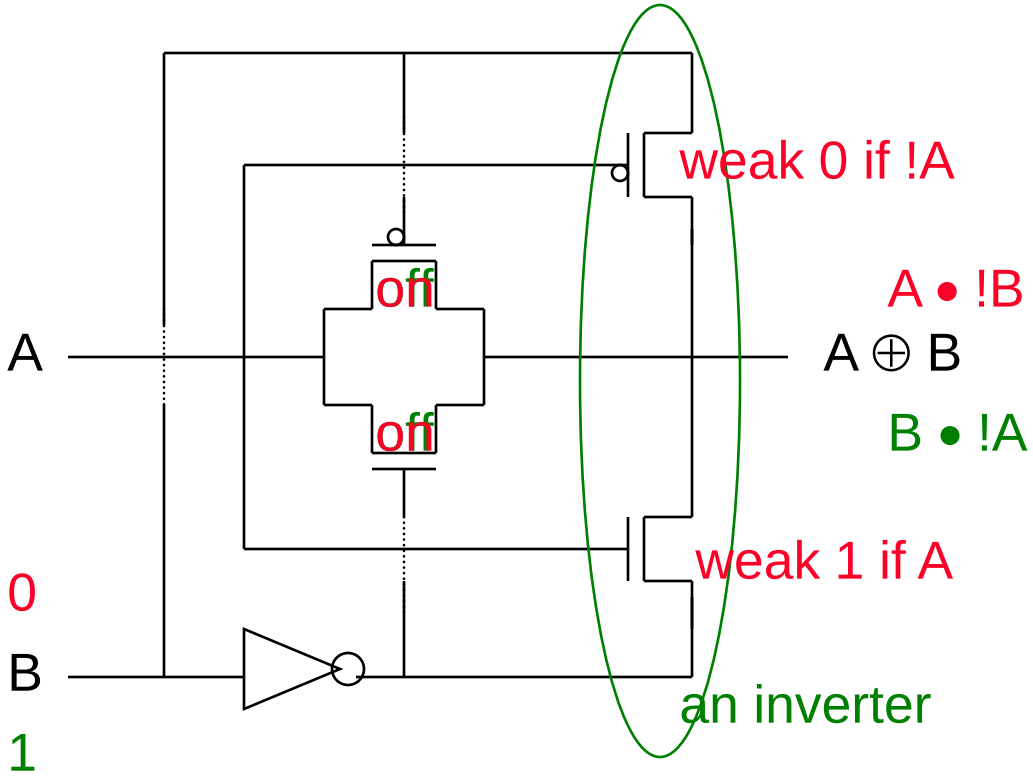
TG Multiplexer



$$F = !(In_1 \bullet S + In_2 \bullet \overline{S})$$

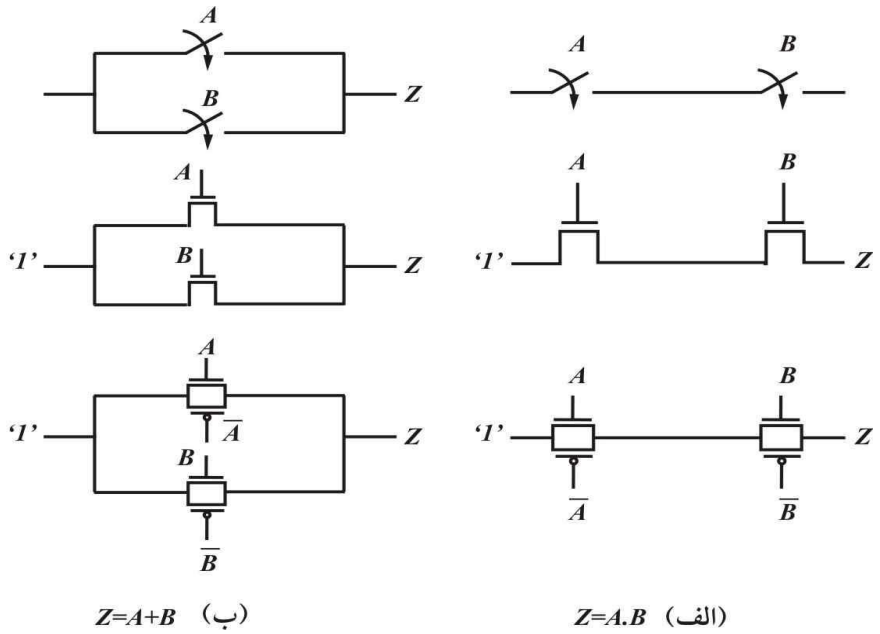


Transmission Gate XOR

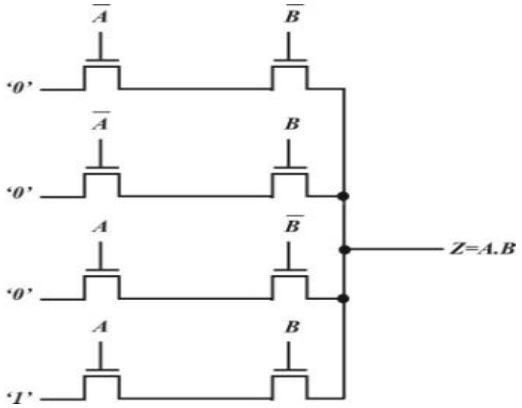


طراحی در منطق گیت انتقالی

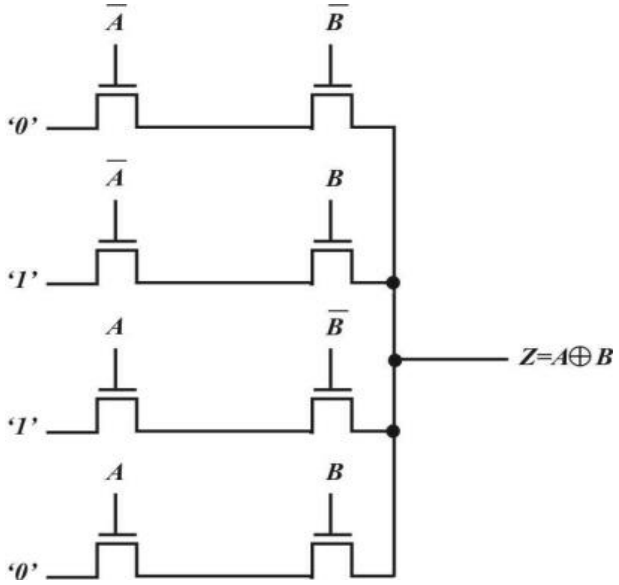
در منطق ترانزیستور های عبوری یا گیت انتقالی، ترانزیستور به عنوان سوییچ قابل کنترل استفاده می شود. به عنوان مثال با استفاده از ترکیب گیت های انتقالی مطابق شکل زیر می توان گیت هـای AND و OR را پیاده کرد.



پیاده سازی AND

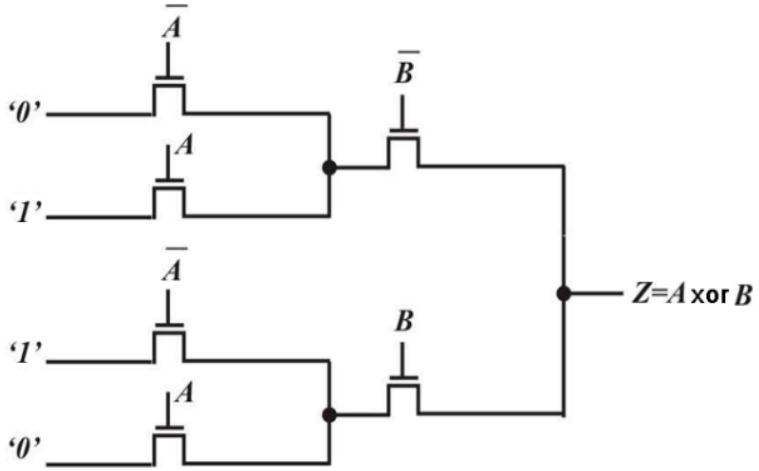


پیاده سازی XOR



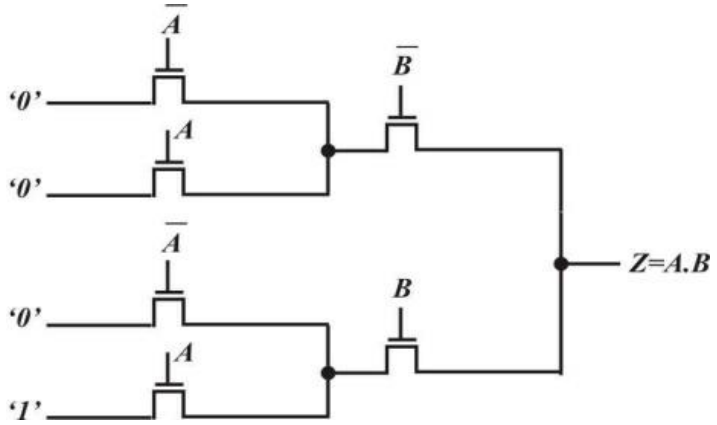
. اگر در مسیر های ۱و ۳ ترانزیستور های دوم را ادغام کنیم و در مسیر های ۲ و ۴ ترانزیستور های B را ادغام کنـیم.

مدار فوق را می توان با استفاده از ۶ ترانزیستور مطابق شکل زیر پیاده سازی کرد.

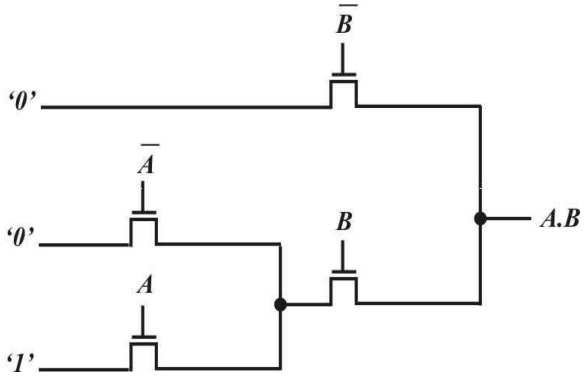


بطور کلی می توان نشان داد برای nورودی در روش اول $n2^n$ ترانزیستور مورد نیاز می باشد و در روش دوم $2(2^n-1)$ ترانزیستور

کاهش تعداد

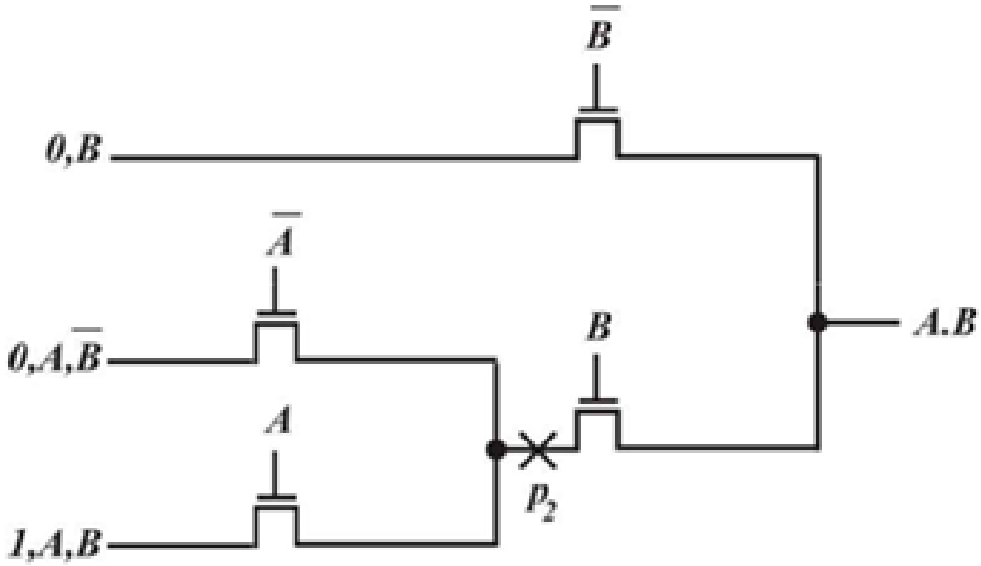


مشاهده می شود که در مسیر های بالا چه $A=1$ باشد و چه $A=0$ در هر دو حالت خروجی صفر است پس نیازی به ترانزیستور های A و $\bar{A}$ نداریم



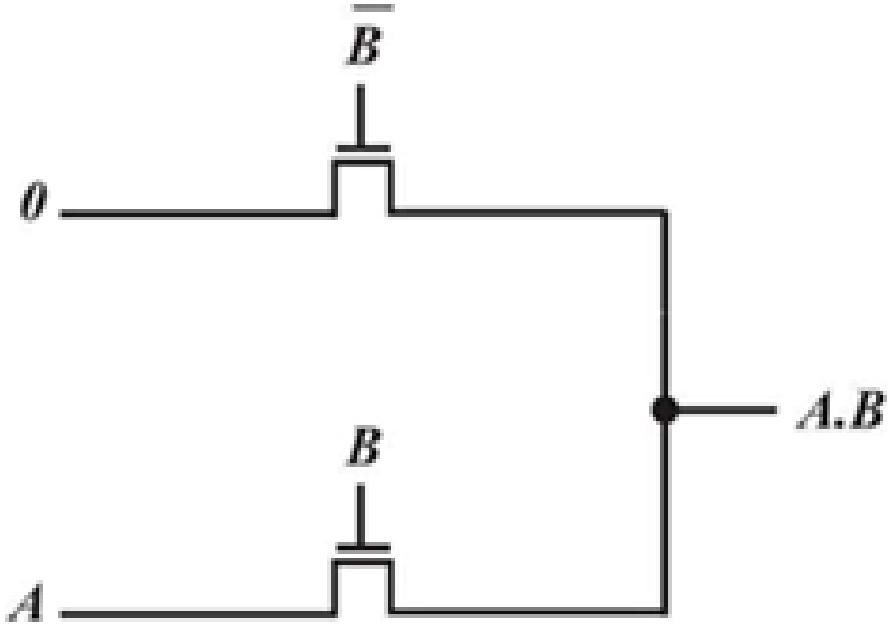
کاهش تعداد

به همین ترتیب در دو مسیر دیگر می توانیم ورودی را تغییر دهیم مثلا در مسیر $A.B$ وقتی فعال است که $A=0$ و $B=1$ باشد. بنابراین می توانیم ورودی آن را به یکی از مقادیر 0 یا A یا همینطور . ہوصل نماییم $\bar{B}$ ورودی مسیر $A.B$ را می توانیم به یکی از مقادیر 1یا A یا B مطابق شکل زیر وصل نماییم:

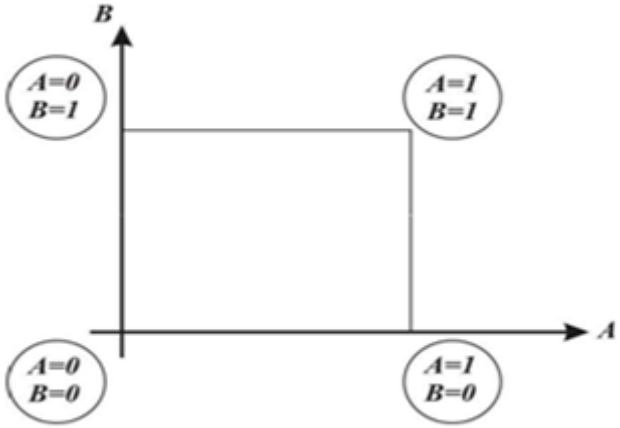


کاهش تعداد

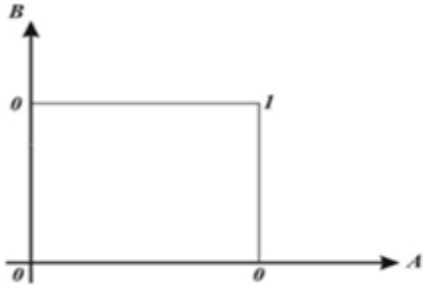
حال اگر ورودی A را انتخاب نماییم چون این ورودی بین دو مسیر مشترک است در صورتیکه $A=1$ یا $A=0$ باشد در هر حال گره $p_2 = A$ می شود بنابراین می توانیم ترانزیستورهای A و $\bar{A}$ را حذف کنیم و مدار AND را با دو ترانزیستور مطابق شکل زیر پیاده سازی کنیم:



کاهش تعداد



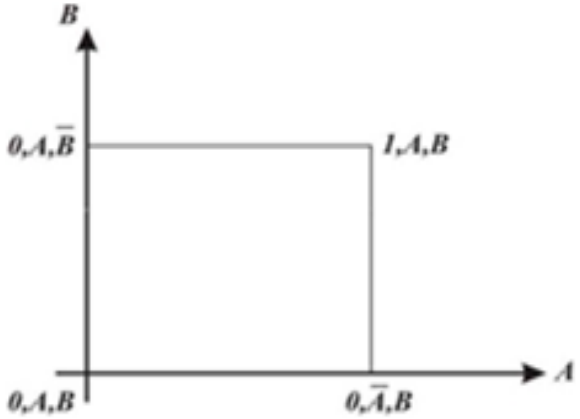
کافی است روی هر نقطه از مدار تابع را به ازای هر حالت A و B قرار دهیم. مثلا برای تابع AND داریم:



A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

کاهش تعداد

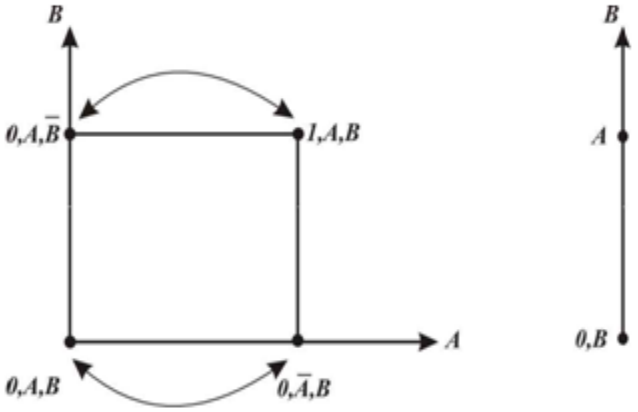
حال علاوه بر *0* و *1* مقادیر دیگری را که می توانند بر حسب *A* و *B* در خروجی قرار بگیرند را اضافه می کنیم



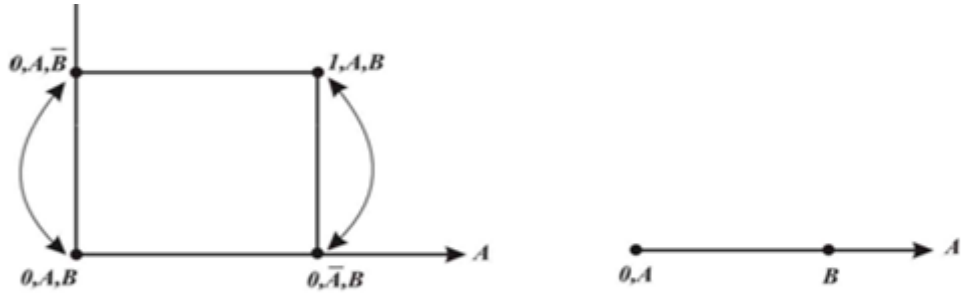
A	B	A.B
0	0	0, A, B
0	1	0, A, B'
1	0	0, A' , B
1	1	1, A, B

کاهش تعداد

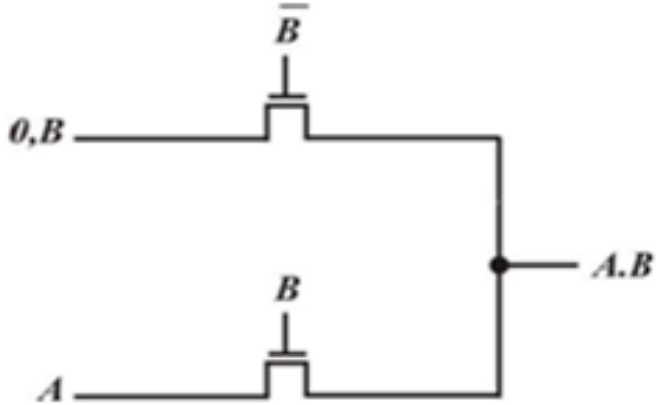
هر نقطه روی نمایش هندسی بیانگر یکی از مسیر های عبوری می باشد. در فرایند بهینه سازی هر بار در جهت یکی از محور ها نقاط مجاور را بر هم منطبق می کنیم و در صورت وجود گزینه مشترک آنرا روی نقطه حاصل می نویسیم. اگر گزینه مشترک وجود نداشت از علامت "-" استفاده می شود. پس از انطباق نقاط مجاور در همه جهات یکی از نمایش های به دست آمده را پیاده سازی می کنیم . به عنوان نمونه برای مدار AND ابتدا در جهت محور A نقاط را منطبق می کنیم و سپس در جهت محور B که هر دو حالت در شکل زیر نشان داده شده اند.



کاهش تعداد

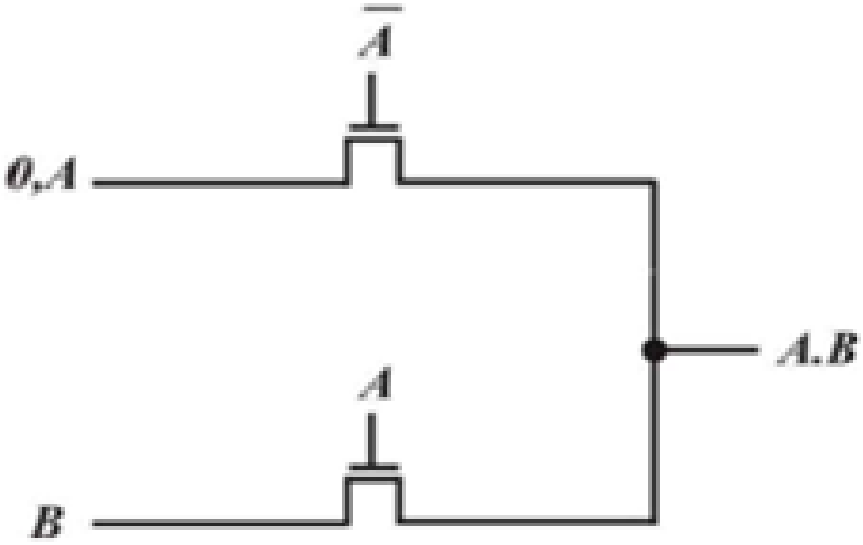


اگر در هر یک از نمایش های فوق مجددا بخواهیم انطباق ایجاد کنیم نقطه مشترکی به دست نمی آید. حال می خواهیم شکل اول را پیاده سازی کنیم. در این شکل نقطه پایینی به ازای حالت $B=0$ است که در این حالت خروجی می تواند یکی از مقادیر 0 یا B باشد و نقطه بالایی حالت $B=1$ است که در این حالت خروجی برابر با A است. پیاده سازی این مدار در شکل زیر دیده می شود.

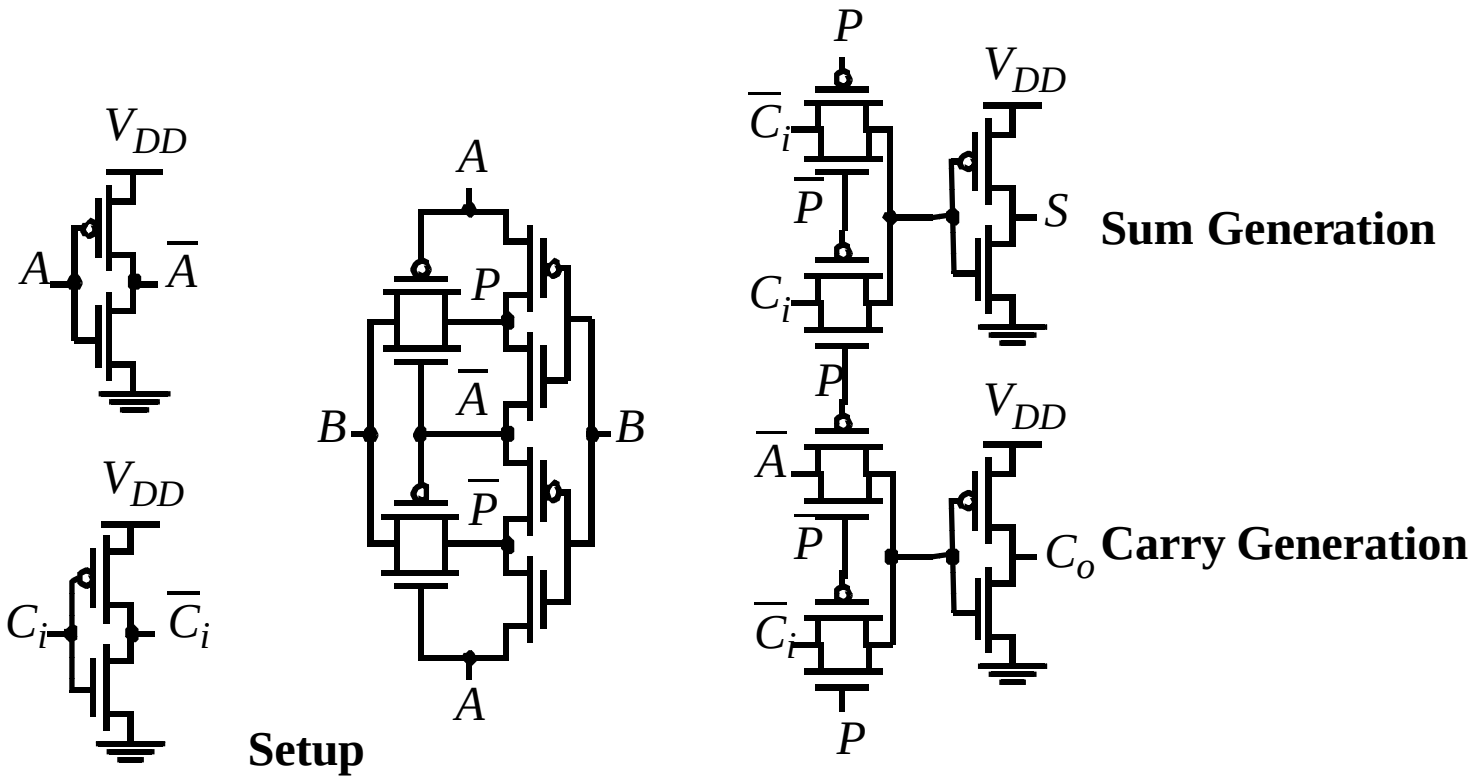


کاهش تعداد

به طور مشابه می توان پیاده سازی گیت را با سوئیچ های A و $\bar{A}$ مطابق شکل زیر انجام داد:



Transmission Gate Full Adder

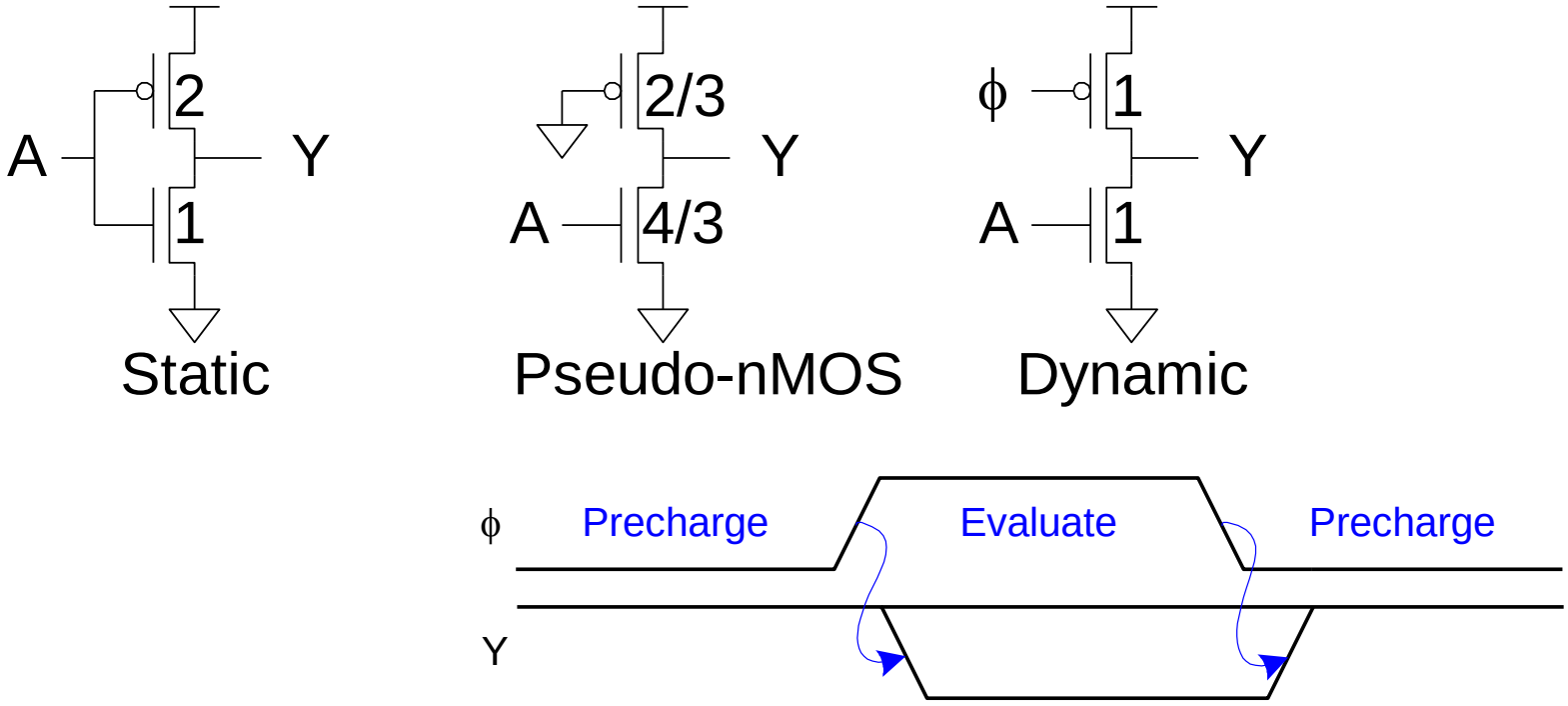


Similar delays for sum and carry

Dynamic Logic

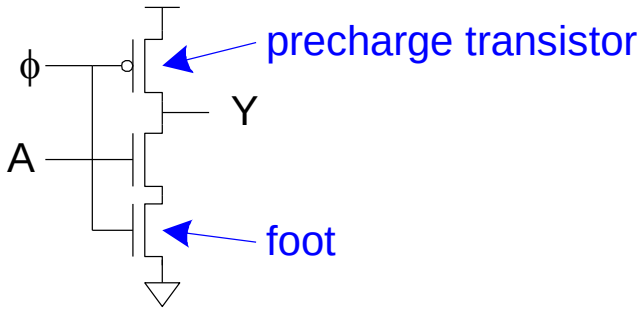
Dynamic Logic

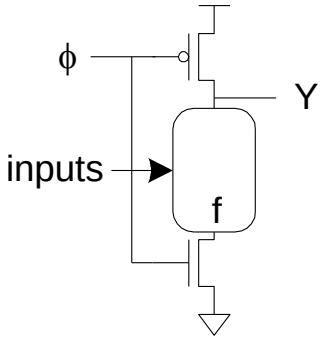
- *Dynamic* gates uses a clocked pMOS pullup
- Two modes: *precharge* and *evaluate*



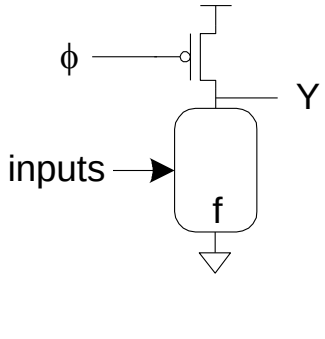
The Foot

- What if pulldown network is ON during precharge?
- Use series evaluation transistor to prevent fight.





footed



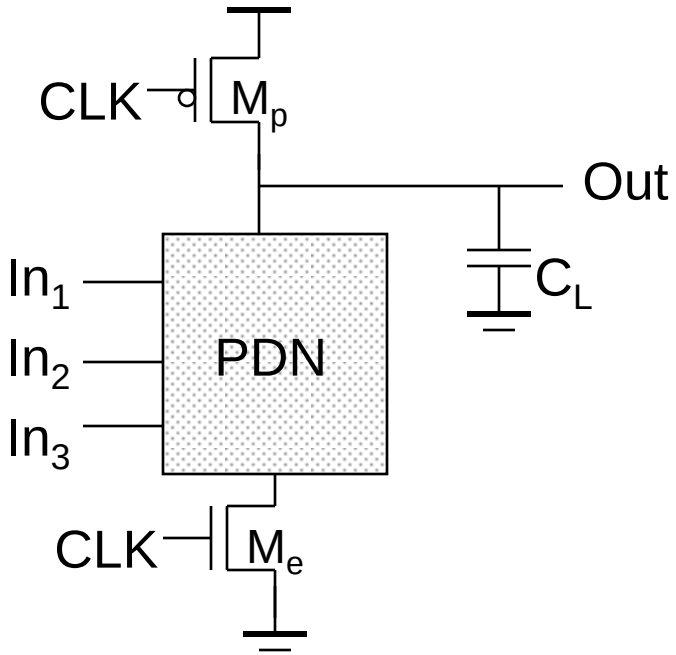
unfooted

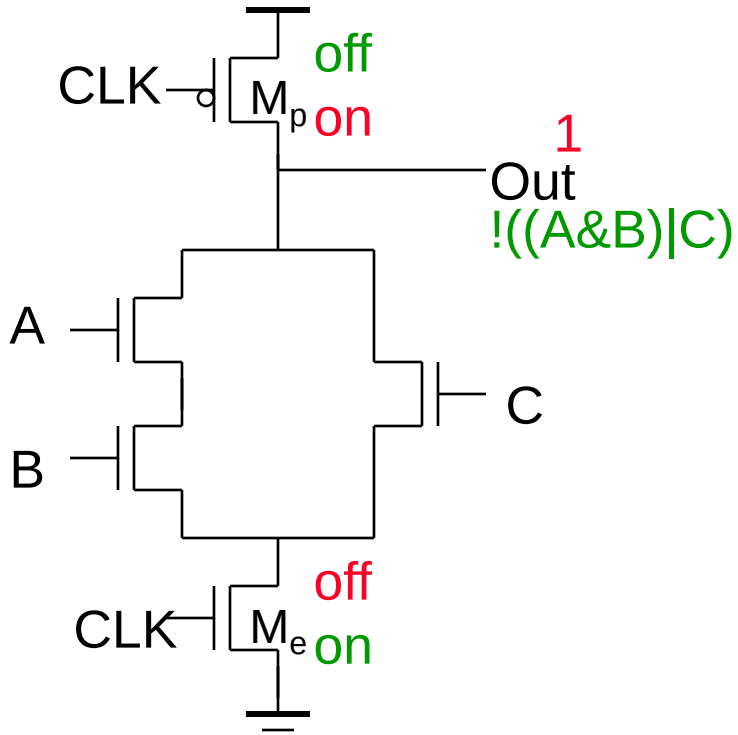
Dynamic CMOS

- 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.
  - fan-in of N requires 2N 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 of precharge and conditional evaluation phases to realize logic functions

Dynamic Gate





Two phase operation

Precharge (CLK = 0)

Evaluate (CLK = 1)

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

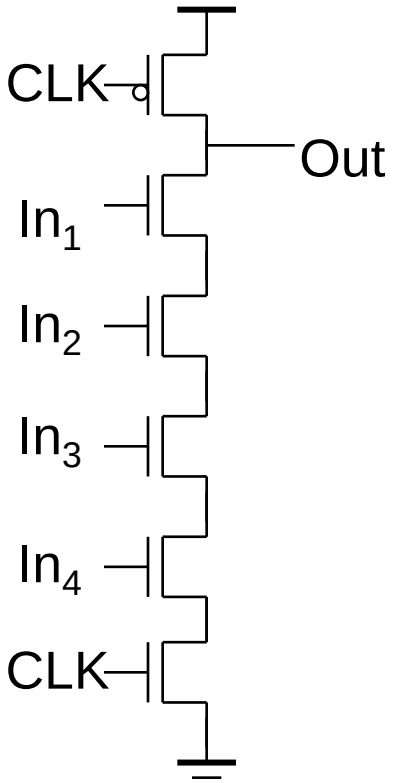
Properties of Dynamic Gates

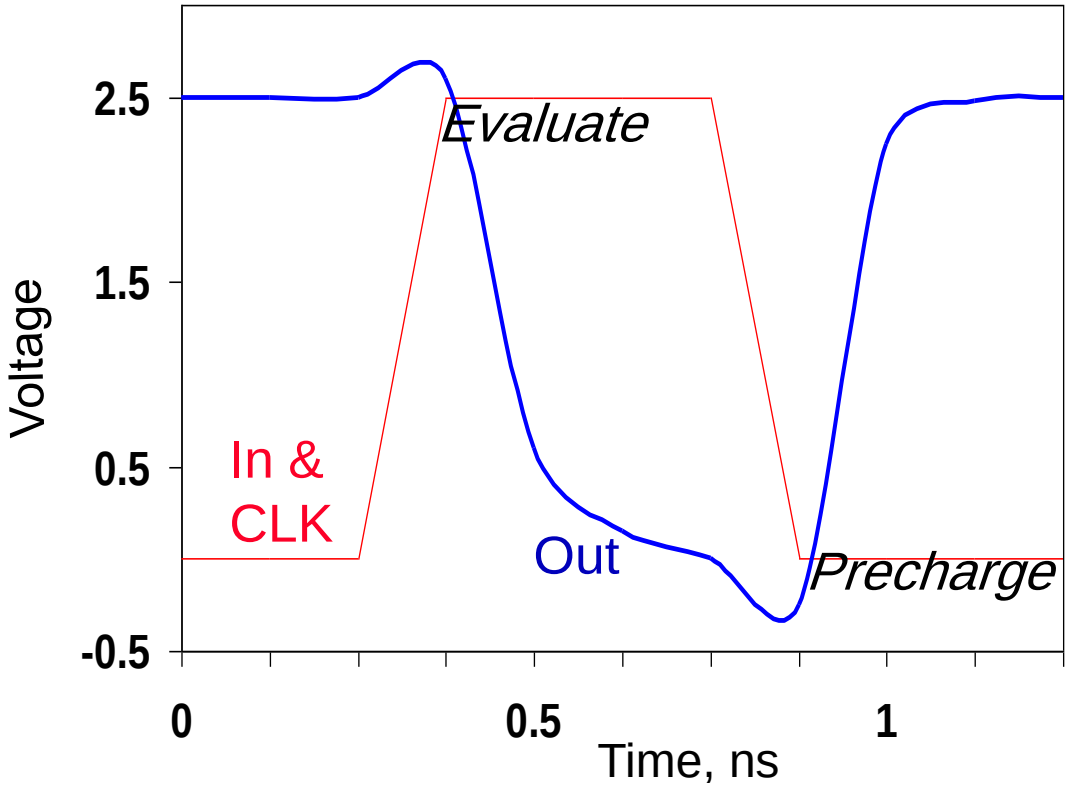
- Logic function is implemented by the PDN only
  - number of transistors is N + 2 (versus 2N for static complementary CMOS)
- Full swing outputs (V_{OL} = GND and V_{OH} = V_{DD})
- Non-ratioed - sizing of the devices does not affect the logic levels
- Faster switching speeds
  - reduced load capacitance due to lower input capacitance (C_{in})
  - reduced load capacitance due to smaller output loading (Cout)
  - no I_{sc}, so all the current provided by PDN goes into discharging C_L

Properties of Dynamic Gates

- Overall power dissipation usually higher than static CMOS
  - no static current path ever exists between V_{DD} and GND (including P_{sc})
  - no glitching
  - higher transition probabilities
  - extra load on Clk
- PDN starts to work as soon as the input signals exceed V_{Tn}, so V_M, V_{IH} and V_{IL} equal to V_{Tn}
  - low noise margin (NM_L)
- Needs a precharge/evaluate clock

Dynamic Behavior

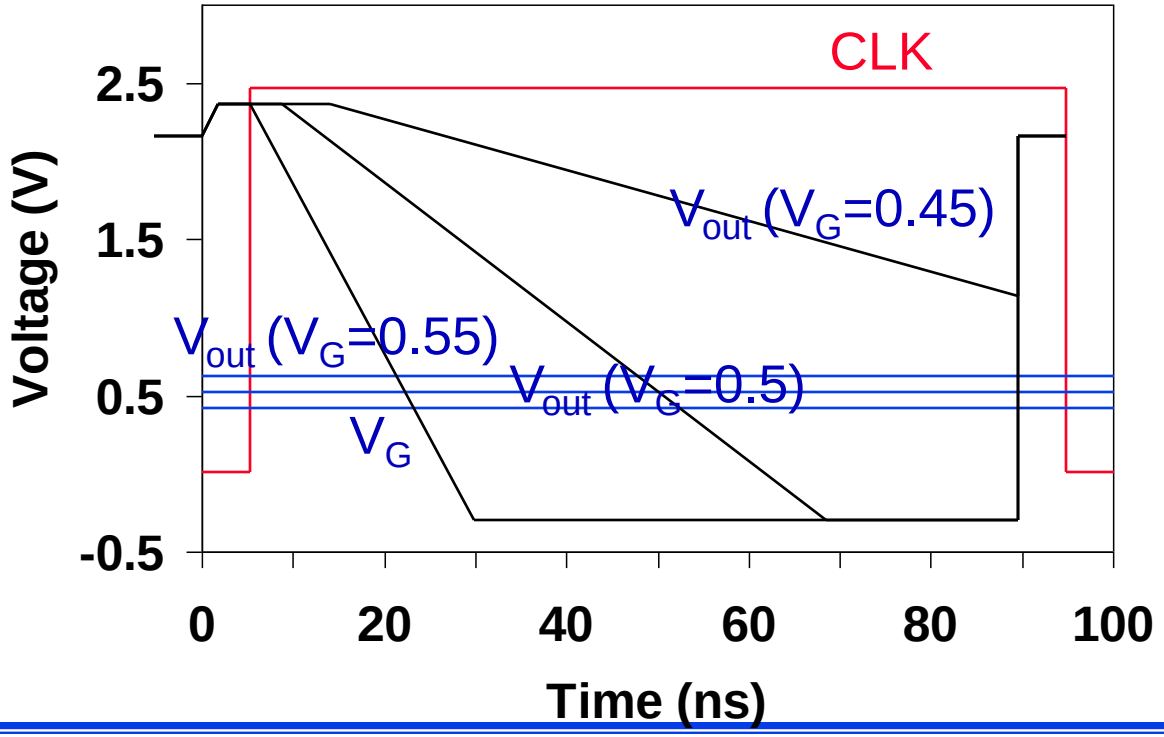




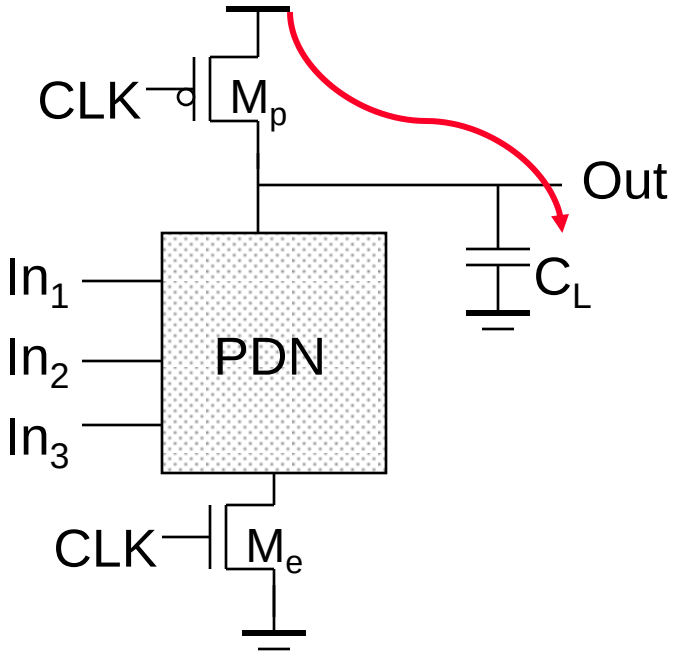
#Trns	V_{OH}	V_{OL}	V_M	NM_H	NM_L	t_{pHL}	t_{pLH}	t_p
6	2.5V	0V	V_{Tn}	2.5-V_{Tn}	V_{Tn}	110ps	0ns	55ps

Gate Parameters are Time Independent

- The amount by which the output voltage drops is a strong function of the input voltage and the available evaluation time.
  - Noise needed to corrupt the signal has to be larger if the evaluation time is short – i.e., the switching threshold is truly time independent.



Power Consumption of Dynamic Gate



Power only dissipated when previous Out = 0

Dynamic Power Consumption is Data Dependent

Dynamic 2-input NOR Gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Assume signal probabilities

$P_{A=1} = 1/2$

$P_{B=1} = 1/2$

Then transition probability

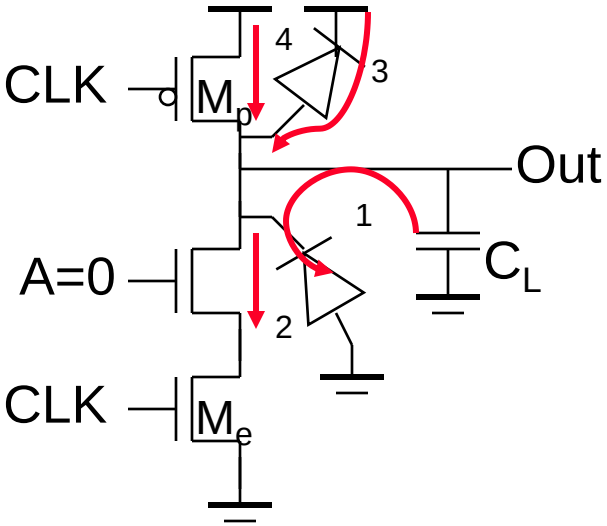
$P_{0\rightarrow1} = P_{out=0} \times P_{out=1}$

$= 3/4 \times 1 = 3/4$

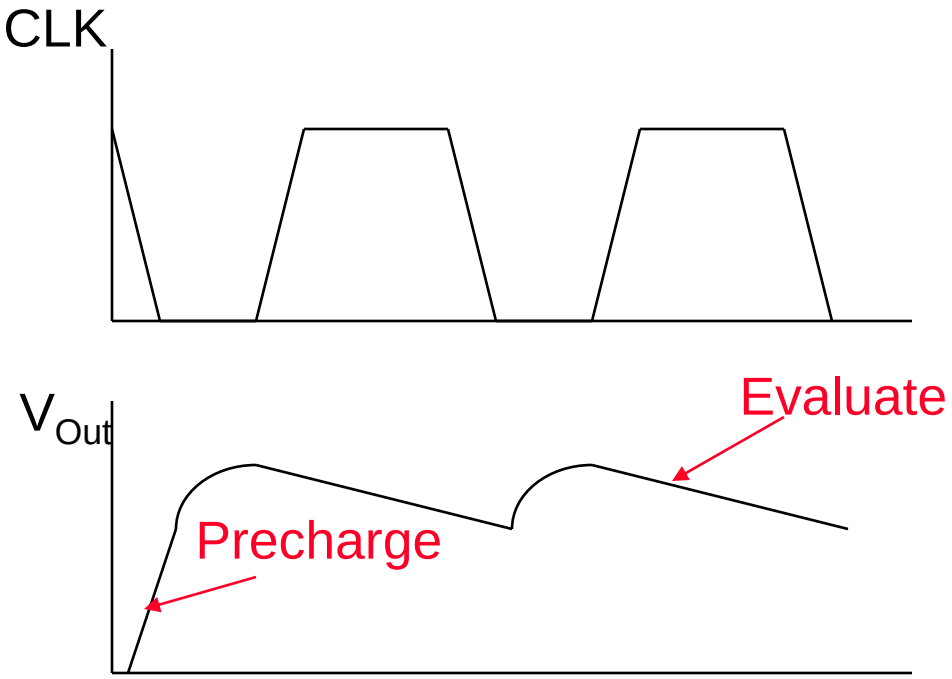
Switching activity can be higher in dynamic gates!

$P_{0\rightarrow1} = P_{out=0}$

Issues in Dynamic Design 1: Charge Leakage



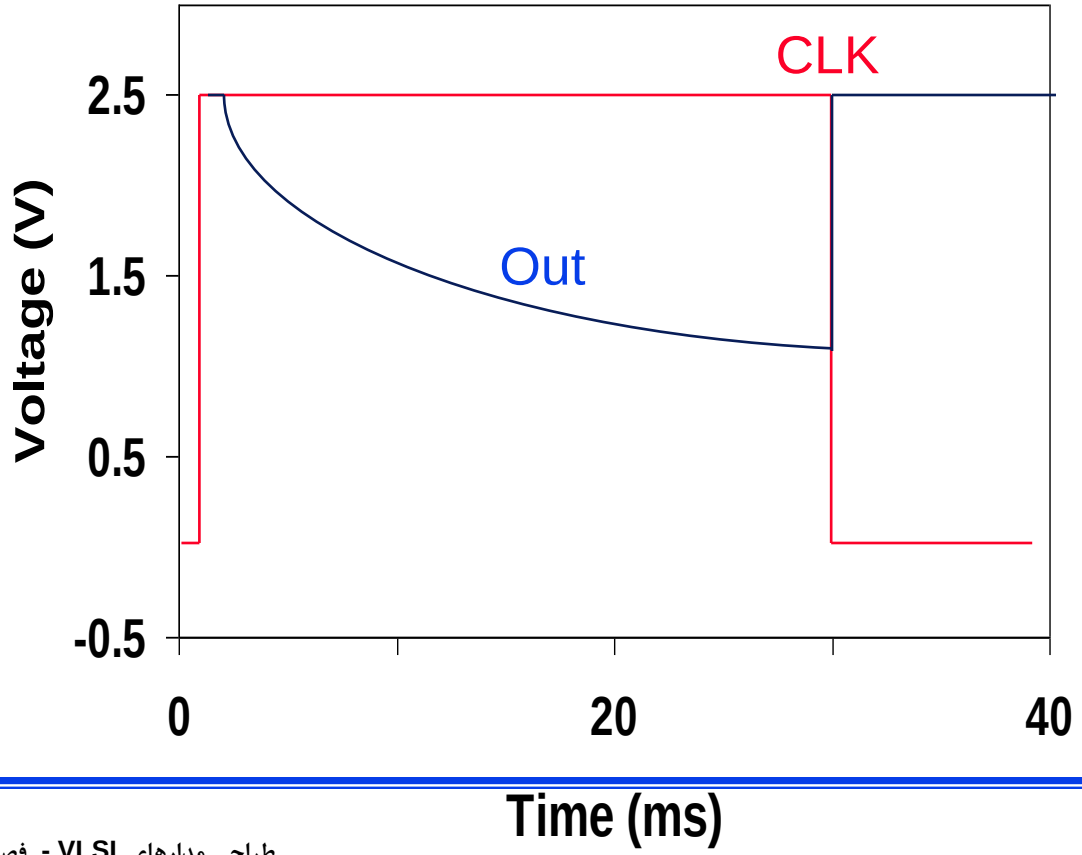
Leakage sources



Minimum clock rate of a few kHz

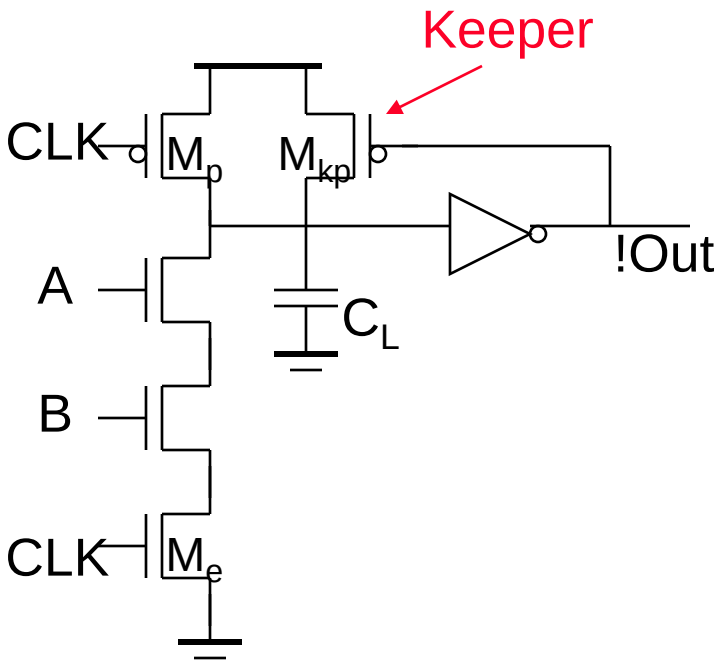
Impact of Charge Leakage

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



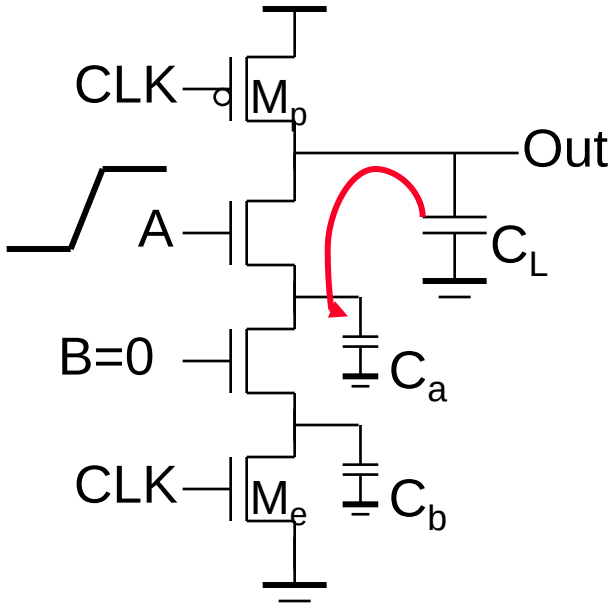
A Solution to Charge Leakage

- Keeper compensates for the charge lost due to the pull-down leakage paths.



Same approach as level restorer for pass transistor logic

Issues in Dynamic Design 2: Charge Sharing

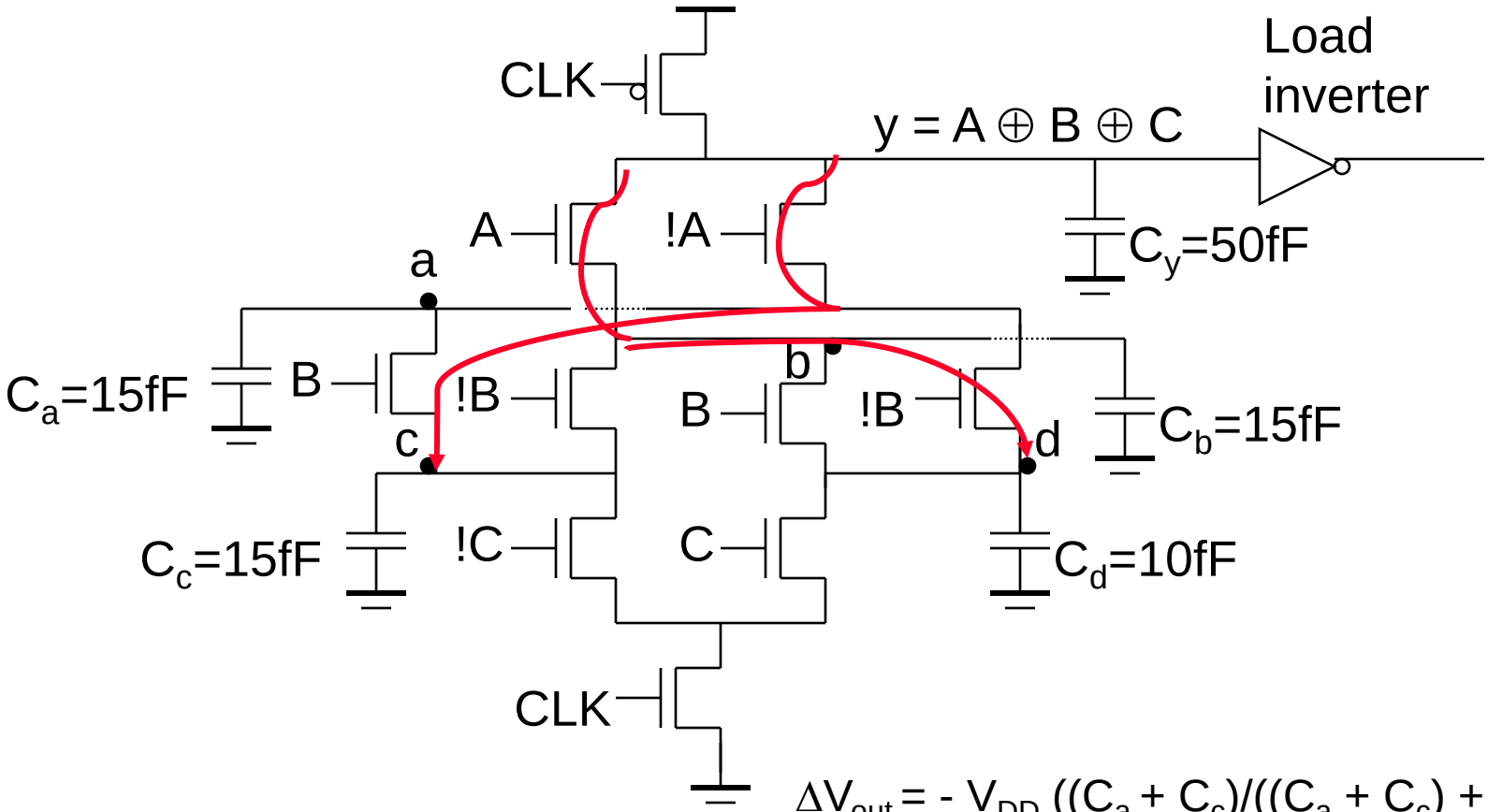


Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to static power consumption by downstream gates and possible circuit malfunction.

When $\Delta V_{out} = - V_{DD} (C_a / (C_a + C_L))$ the drop in V_{out} is large enough to be below the switching threshold of the gate it drives causing a malfunction.

Charge Sharing Example

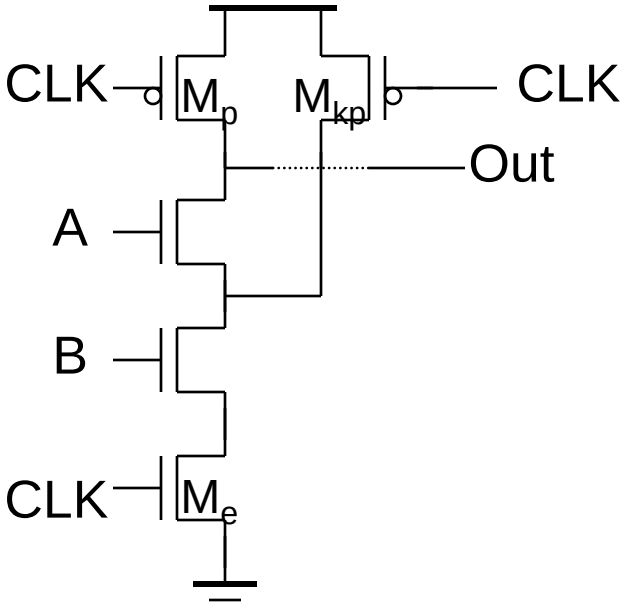
What is the worst case voltage drop on y? (Assume all inputs are low during precharge and that all internal nodes are initially at 0V.)



$$\Delta V_{out} = - V_{DD} ((C_a + C_c)/((C_a + C_c) + C_y))$$

$$= - 2.5V*(30/(30+50)) = -0.94V$$

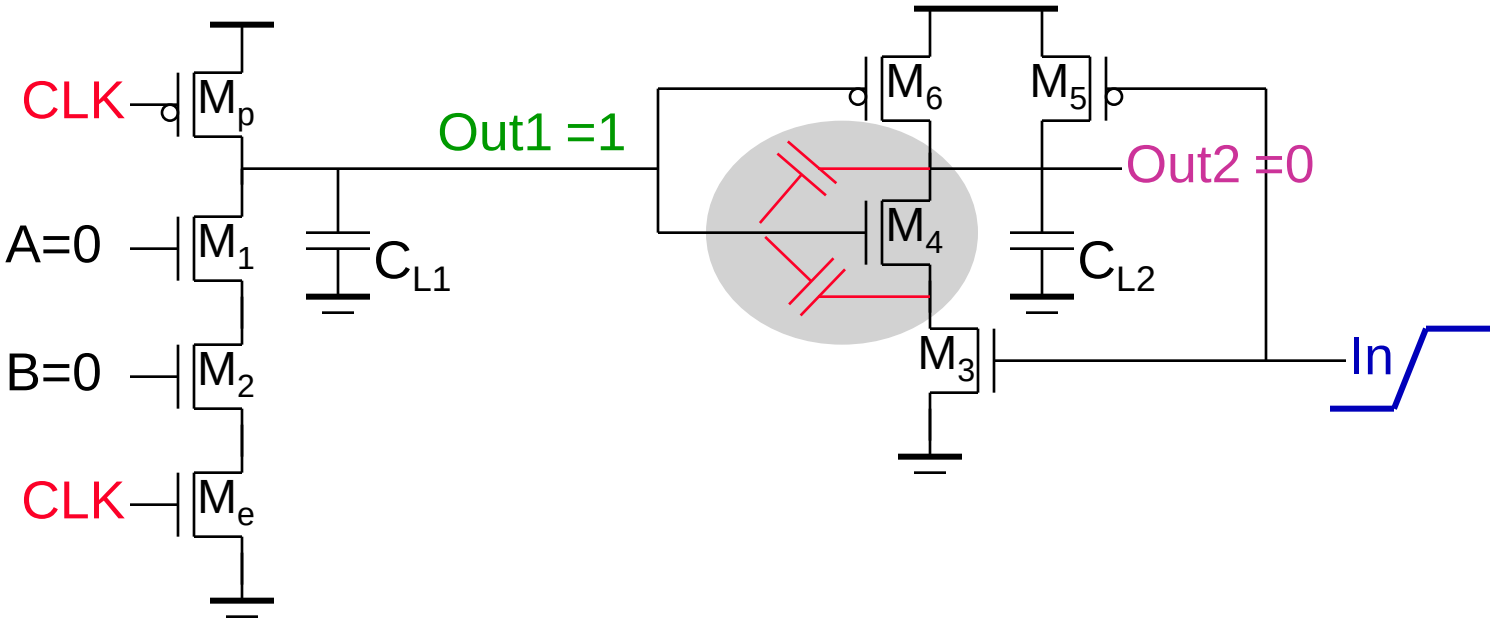
Solution to Charge Redistribution



Precharge internal nodes using a clock-driven transistor (at the cost of increased area and power)

Issues in Dynamic Design 3: Backgate Coupling

- Susceptible to crosstalk due to 1) high impedance of the output node and 2) capacitive coupling
  - Out2 capacitively couples with Out1 through the gate-source and gate-drain capacitances of M4

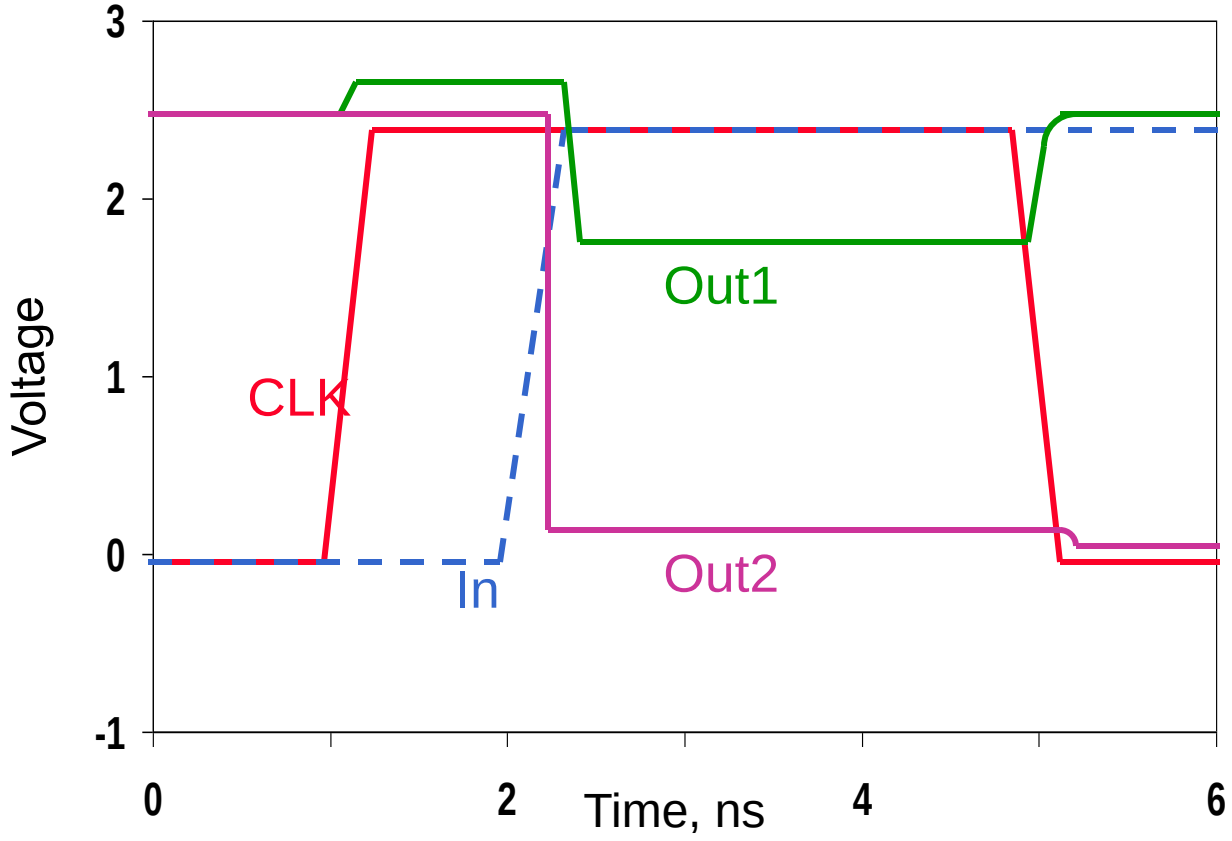


Dynamic NAND

Static NAND

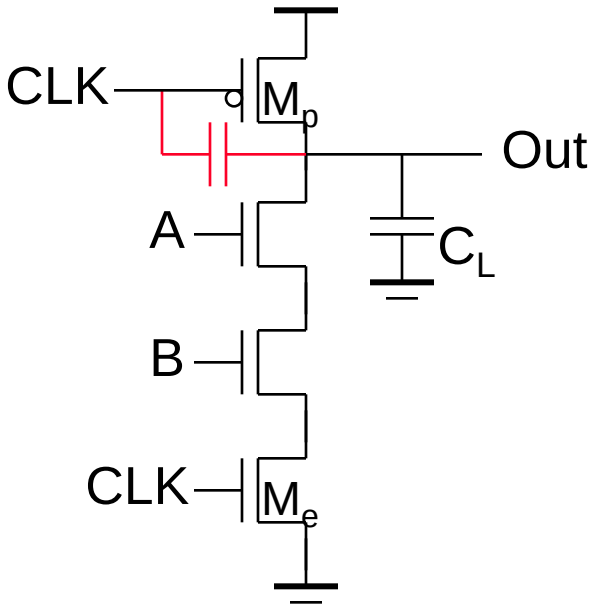
Backgate Coupling Effect

- Capacitive coupling means Out1 drops significantly so Out2 doesn't go all the way to ground



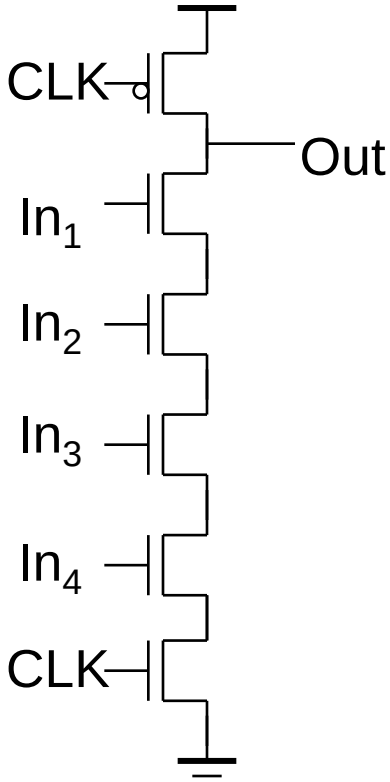
Issues in Dynamic Design 4: Clock Feedthrough

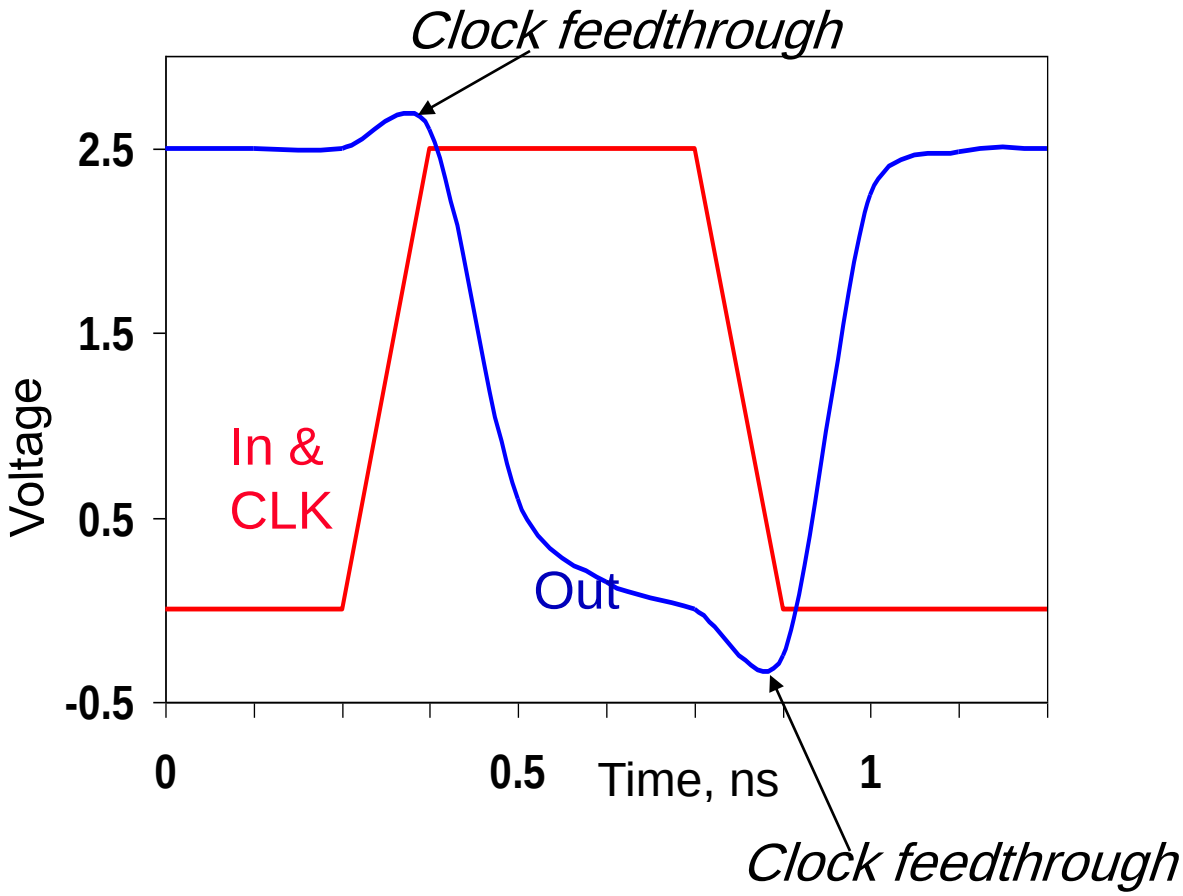
- A special case of capacitive coupling between the clock input of the precharge transistor and the dynamic output node



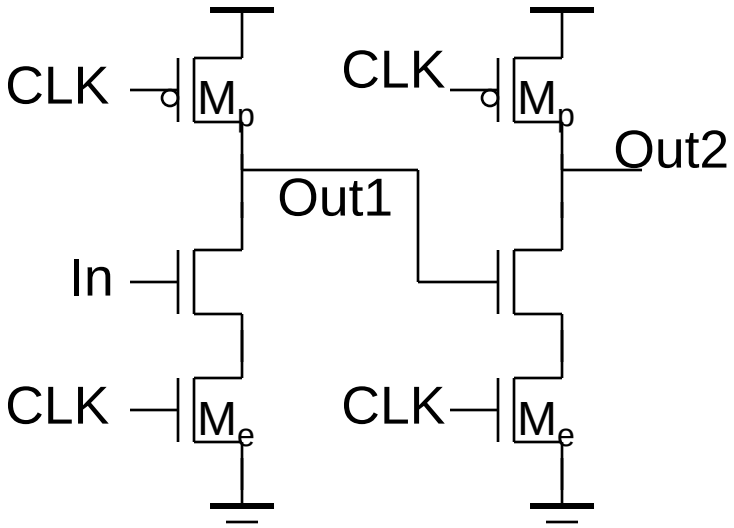
Coupling between Out and CLK input of the precharge device due to the gate-drain capacitance. So voltage of Out can rise above V_{DD}. The fast rising (and falling edges) of the clock couple to Out.

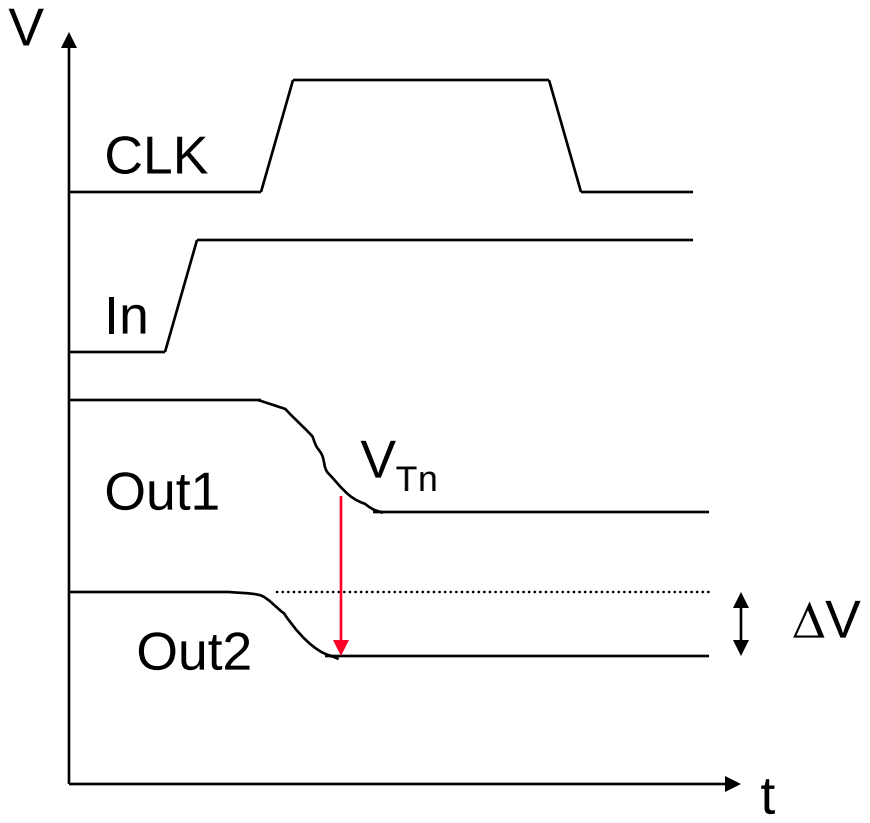
Clock Feedthrough





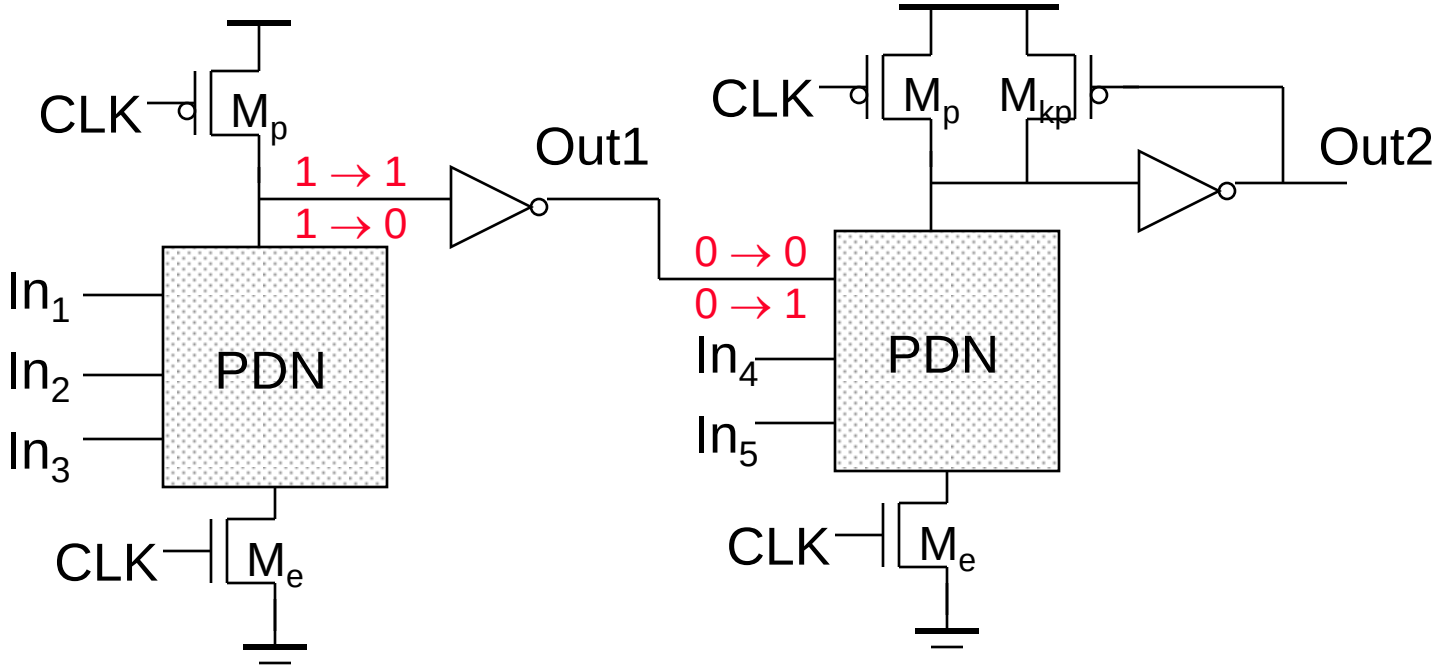
Cascading Dynamic Gates



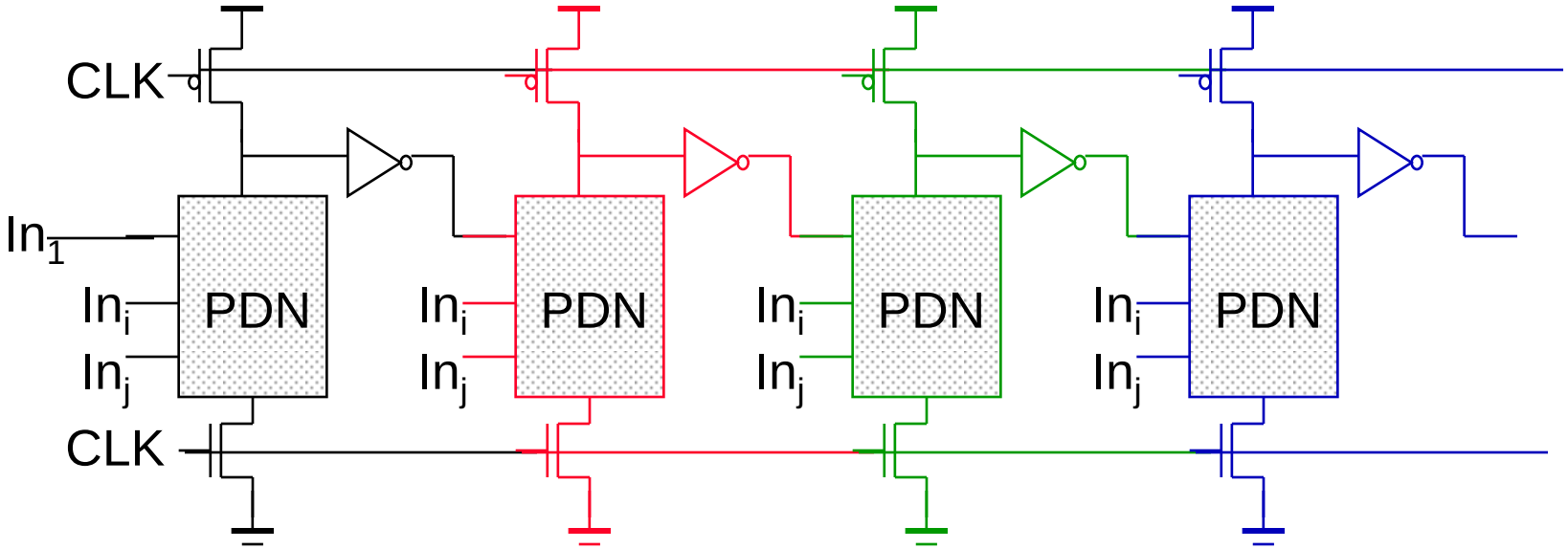


Only a single 0 → 1 transition allowed at the inputs during the evaluation period!

Domino Logic

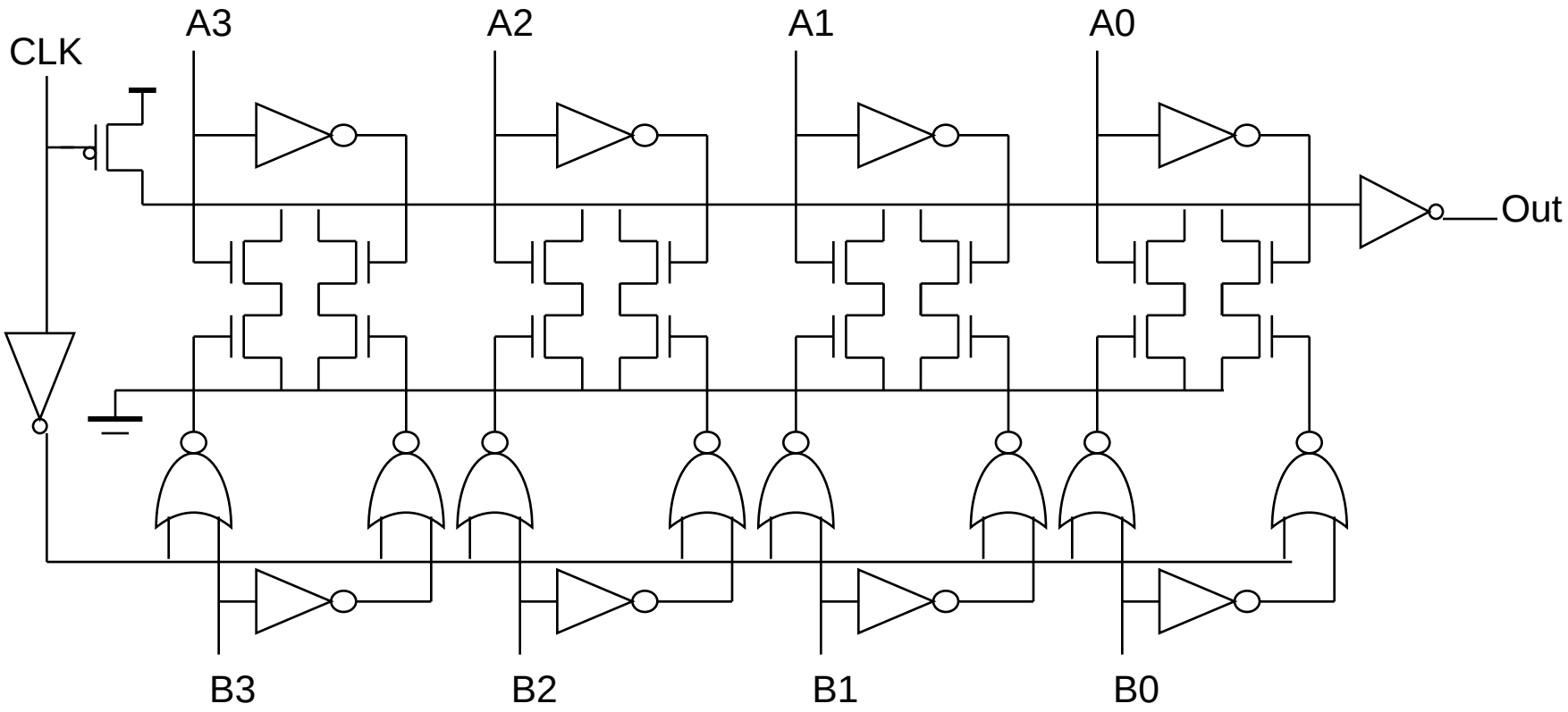


Why Domino?



Like falling dominos!

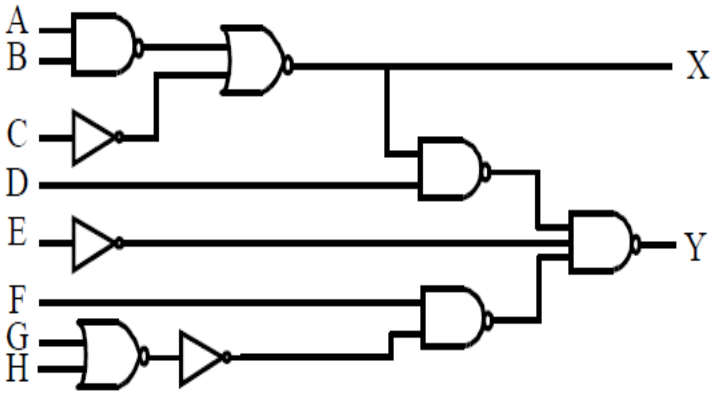
Domino Comparator



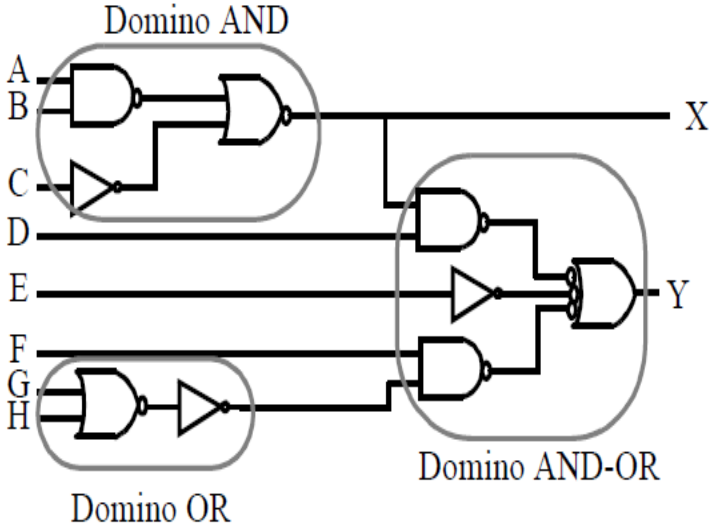
Properties of Domino Logic

- Only non-inverting logic can be implemented, fixes include
  - can reorganize the logic using Boolean transformations
  - use differential logic (dual rail)
  - use np-CMOS (zipper)
- Very high speed
  - $t_{pHL} = 0$
  - static inverter can be optimized to match fan-out (separation of fan-in and fan-out capacitances)

Transformation



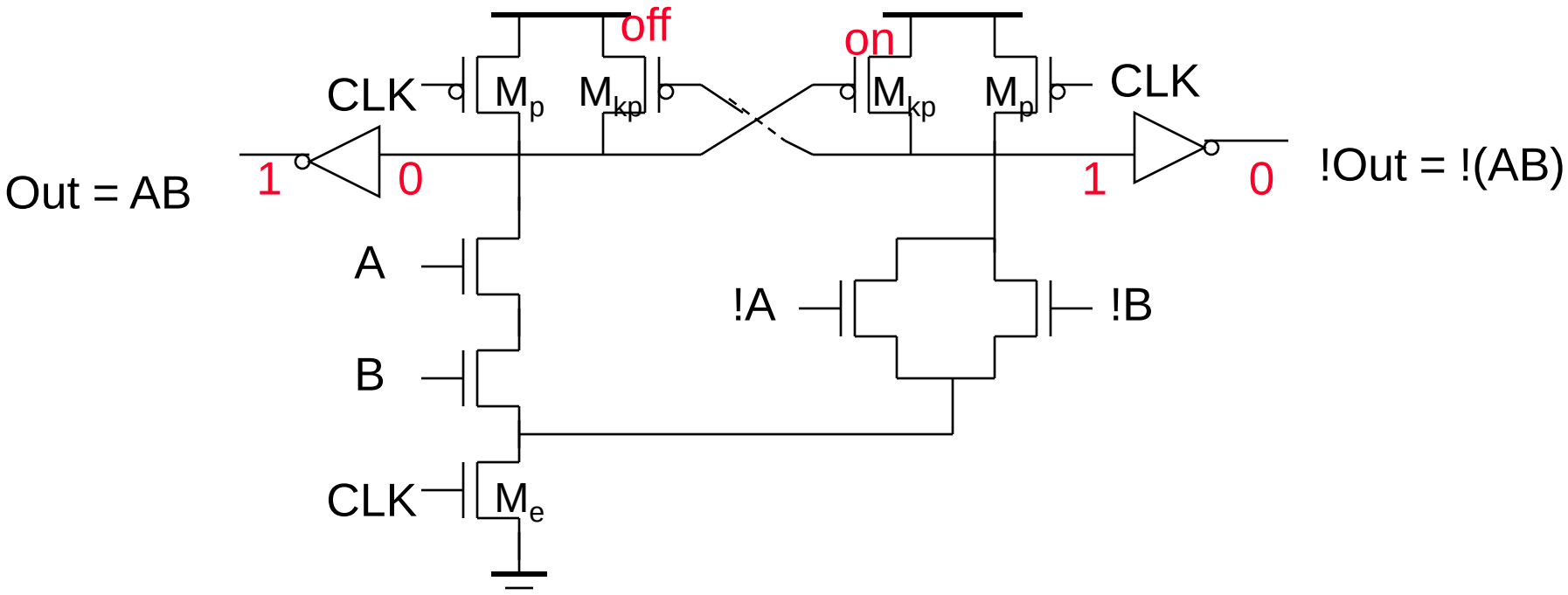
(a) before logic transformation



(b) after logic transformation

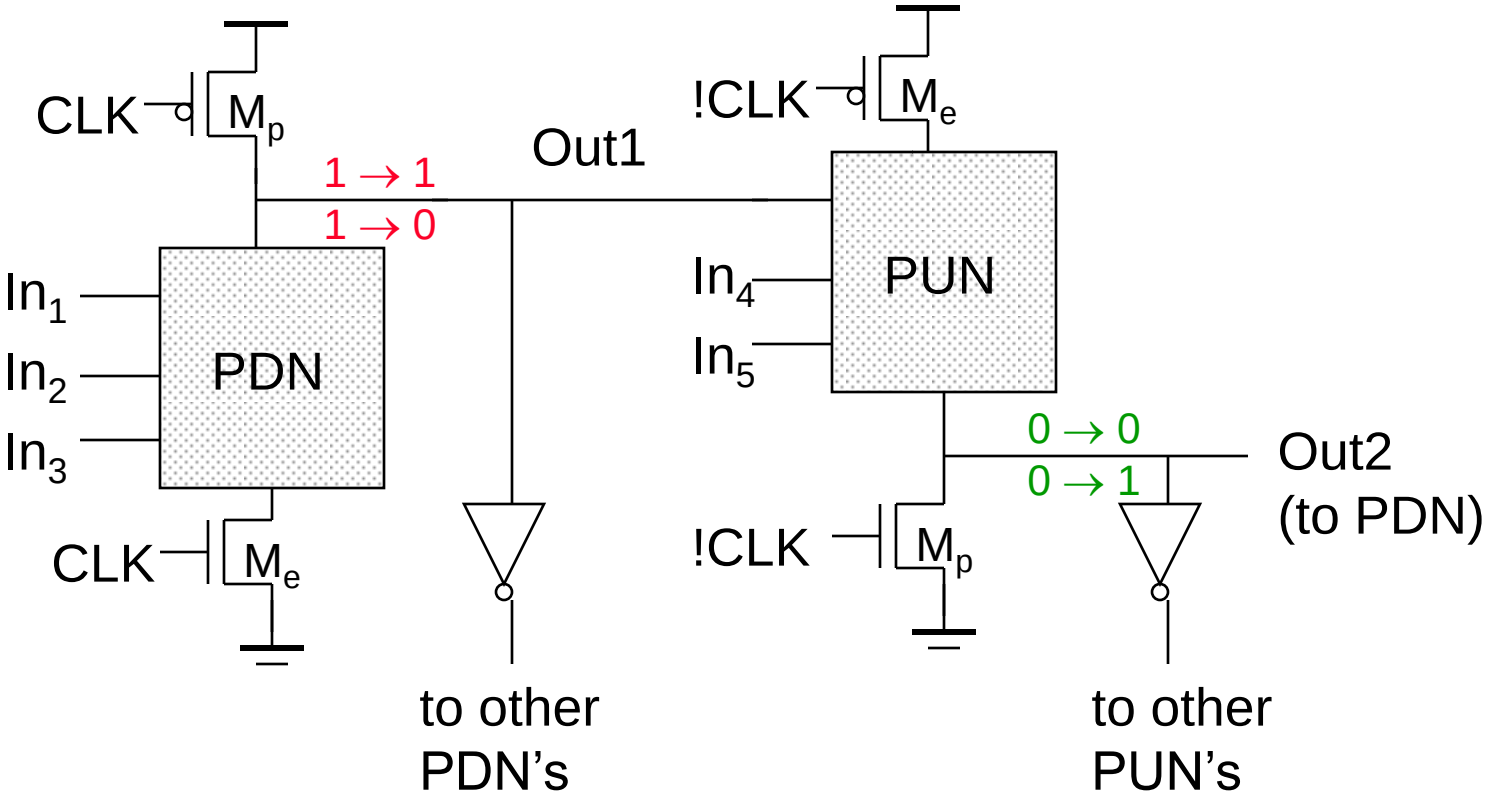
Not always possible

Differential (Dual Rail) Domino



Due to its high-performance, differential domino is very popular and is used in several commercial microprocessors!

np-CMOS (Zipper)



Only $0 \rightarrow 1$ transitions allowed at inputs of PDN

Only $1 \rightarrow 0$ transitions allowed at inputs of PUN

How to Choose a Logic Style

- Must consider ease of design, robustness (noise immunity), area, speed, power, system clocking requirements, fan-out, functionality, ease of testing

4-input NAND

Style	# Trans	Ease	Ratioed?	Delay	Power
Comp Static	8	1	no	3	1
CPL*	12 + 2	2	no	4	3
domino	6 + 2	4	no	2	2 + clk
DCVSL*	10	3	yes	1	4

* Dual Rail

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

جلسه آینده

- درس بعدی
  - اتصالات میانی