EE3230 Lecture 7: Combinational Circuit Design

Ping-Hsuan Hsieh (謝秉璇)

Delta Building R908

EXT 42590

phsieh@ee.nthu.edu.tw

Outline

- **Static CMOS**
- Ratioed circuits
- Cascode voltage switch logic
- Dynamic circuits
- Pass-transistor circuits
- Circuit pitfalls

Static CMOS

- Bubble pushing
- Compound gates
- Logical effort example
- Input ordering
- Asymmetric gates
- Skewed gates
- Best P/N ratio

Example I

```
module mux(input  s, d0, d1,
           output y);

   assign y = s ? d1 : d0;
endmodule
```

1) Sketch a design using AND, OR, and NOT gates

Example II

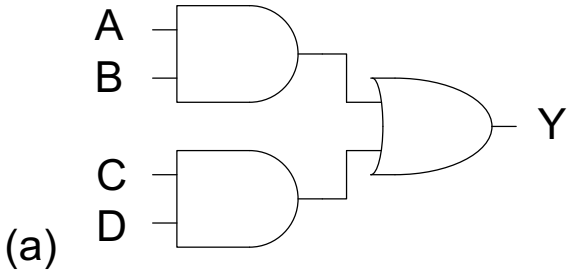
2) Sketch a design using NAND, NOR, and NOT gates. Assume $\overline{S}$ is available

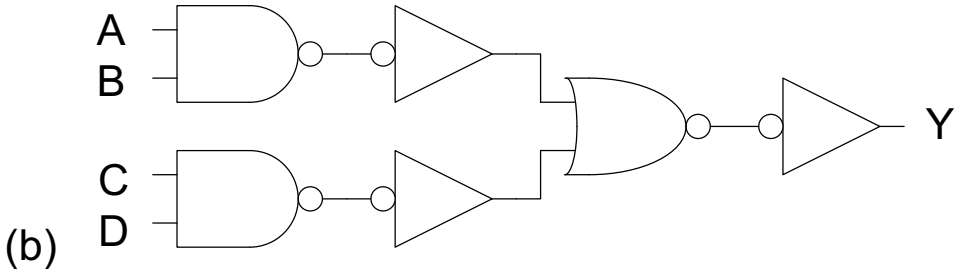
Bubble Pushing

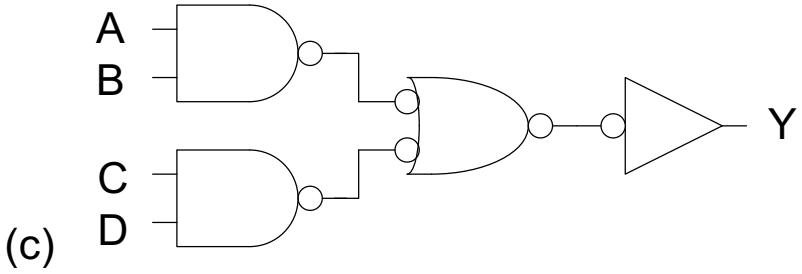
- Start with network of AND / OR gates
- Convert to NAND / NOR + inverters
- Push bubbles around to simplify logic
  - Remember DeMorgan's Law

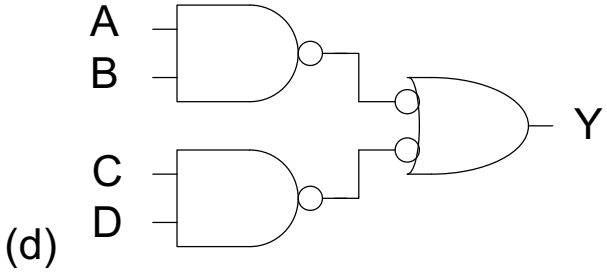
Bubble Pushing

- Y = AB + CD









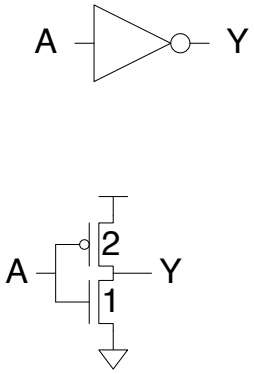
Example III

3) Sketch a design using one compound gate and one NOT gate. Assume $\overline{S}$ is available.

Logical Effort of Compound Gates

unit inverter

$$Y = \overline{A}$$

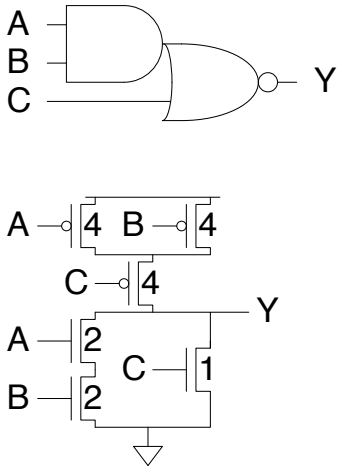


$g_A = 3/3$

$p = 3/3$

AOI21

$$Y = \overline{A \bullet B + C}$$



$g_A = 6/3$

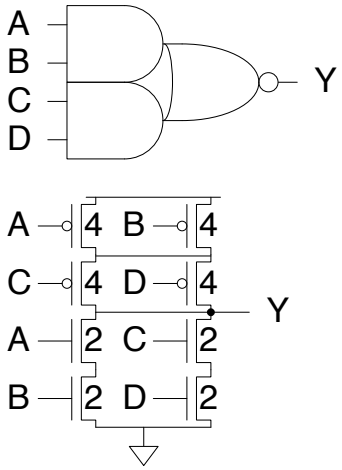
$g_B = 6/3$

$g_C = 5/3$

$p = 7/3$

AOI22

$$Y = \overline{A \bullet B + C \bullet D}$$



$g_A = 6/3$

$g_B = 6/3$

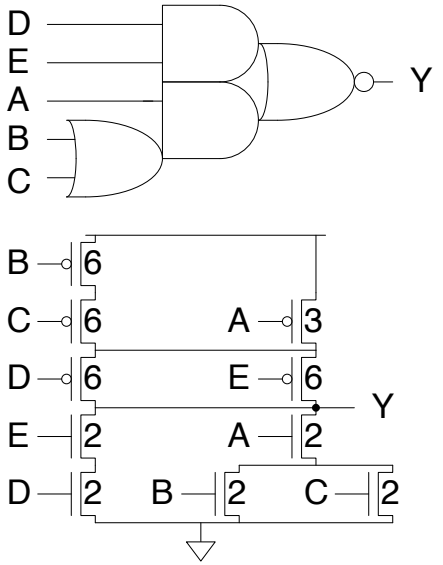
$g_C = 6/3$

$g_D = 6/3$

$p = 12/3$

Complex AOI

$$Y = \overline{A \bullet \left(B + C\right) + D \bullet E}$$



$g_A = 5/3$

$g_B = 8/3$

$g_C = 8/3$

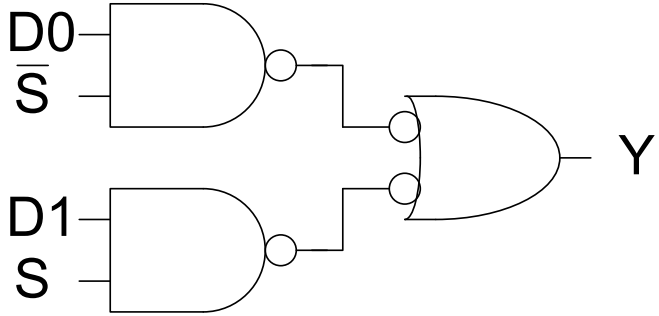
$g_D = 8/3$

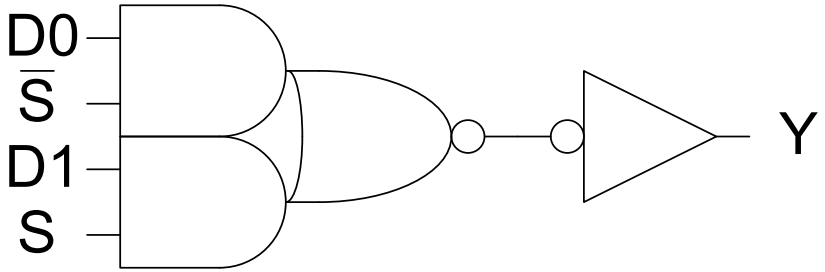
$g_E = 8/3$

$p = 16/3$

Example IV

- The multiplexer has a maximum input capacitance of 16 units on each input. It must drive a load of 160 units. Estimate the delay of the NAND and compound gate designs





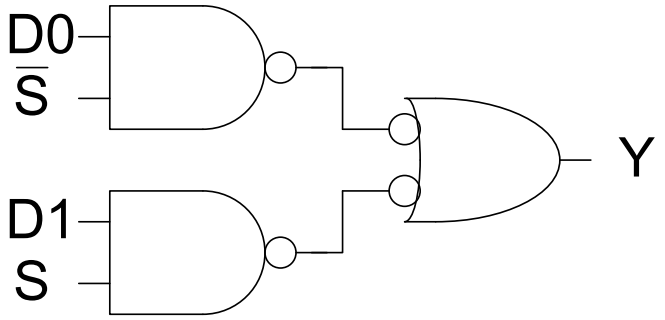
H =

B =

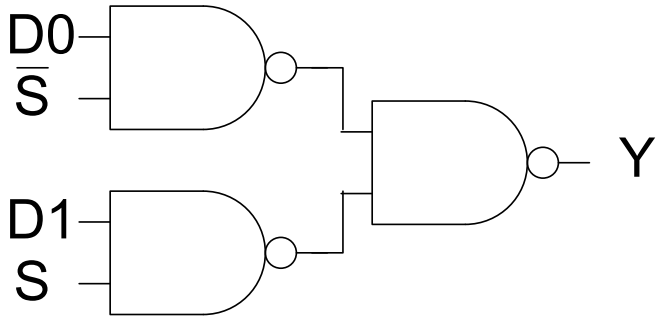
N =

NAND Solution

- P =
- G =
- F =
- $\hat{f}$ =
- D =



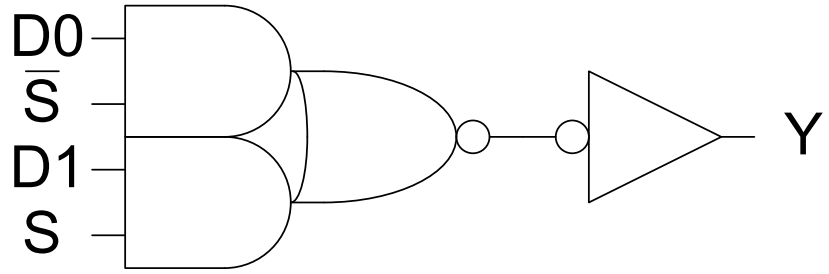
DeMorgan's Law



Compound Solution

- P =
- G =
- F =
- $\hat{f}$ =
- D =

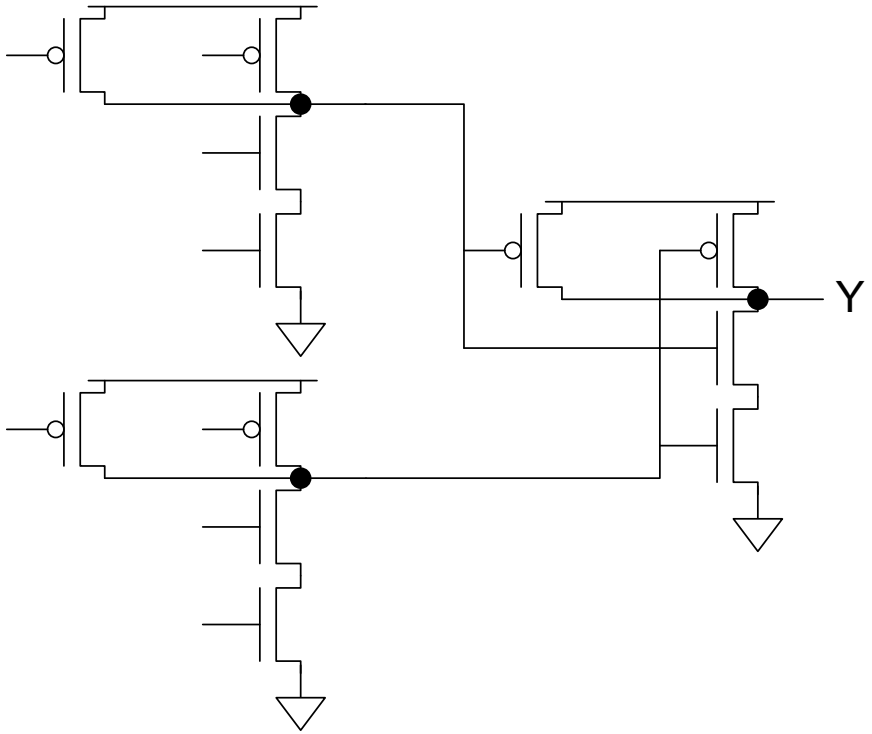
AOI22 + INV



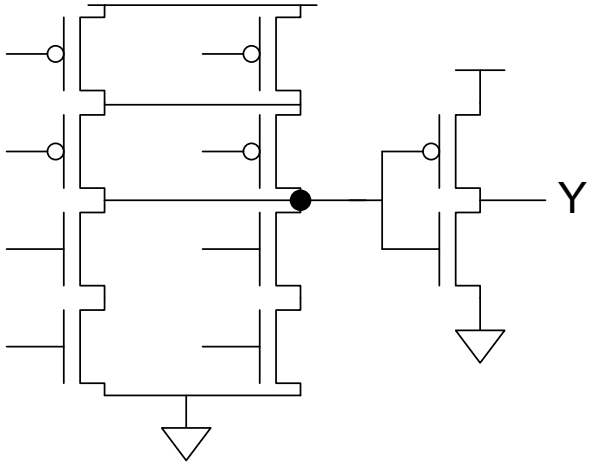
Example V

- Annotate your designs with transistor sizes that achieve this delay

NAND solution

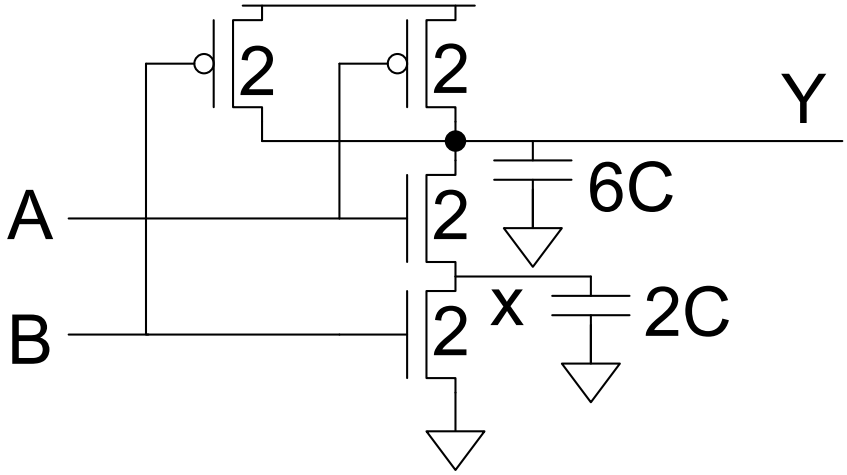


Compound solution



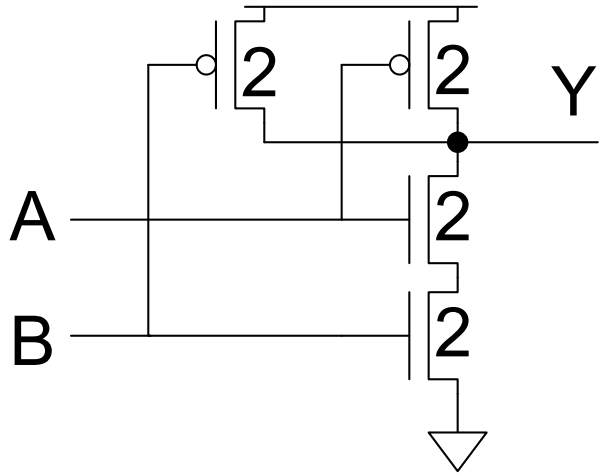
Input Order

- Our parasitic delay model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest?
    - If B arrives latest?



Inner and Outer Inputs

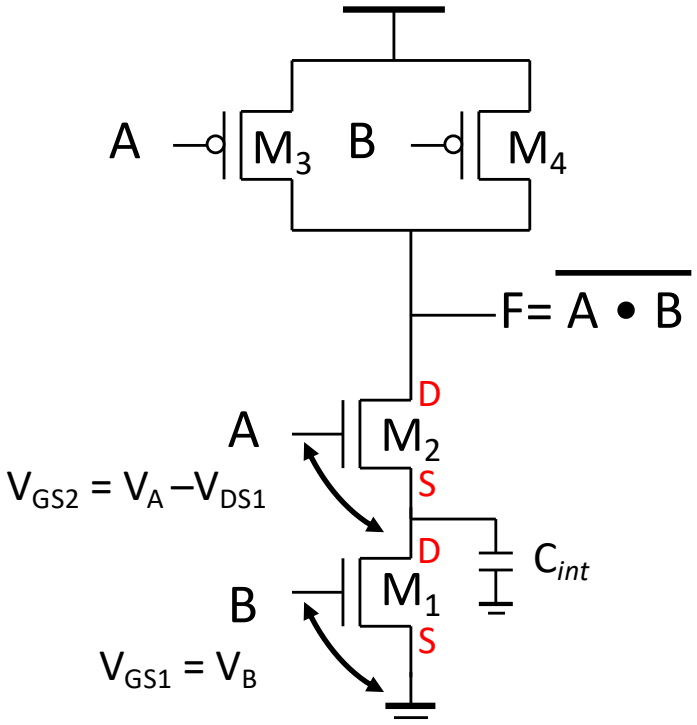
- *Outer* inputs are closer to rail (B)
- *Inner* inputs are closer to output (A)

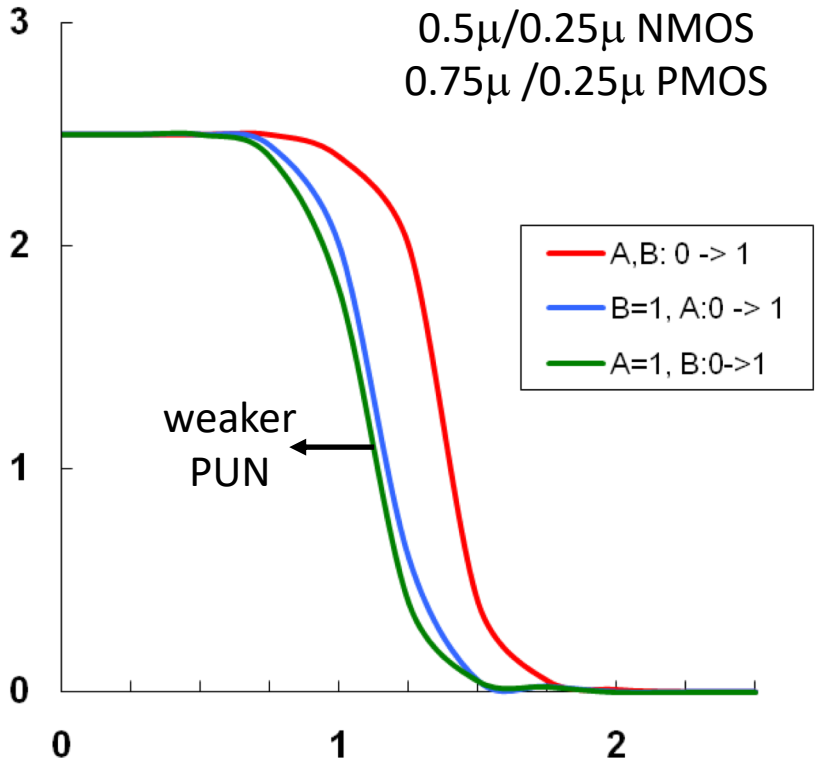


- If input arrival time is known
  - Connect latest input to inner terminal

Data-Dependent VTC

- The threshold voltage of M_2 is higher than M_1 due to **body effect**
- Since V_{SB} of M_2 is not zero (when $V_B = 0$)



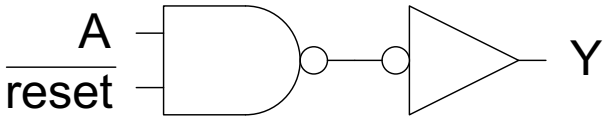


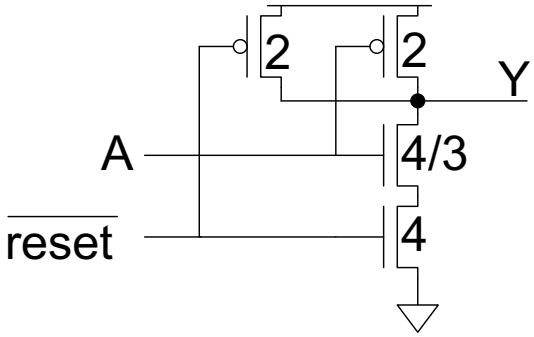
Symmetric Gates

- Inputs can be made perfectly symmetric

Asymmetric Gates

- Asymmetric gates favor one input over another
- **Ex:** suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same
  - g_A =
  - g_B =
  - $g_{total} = g_A + g_B$

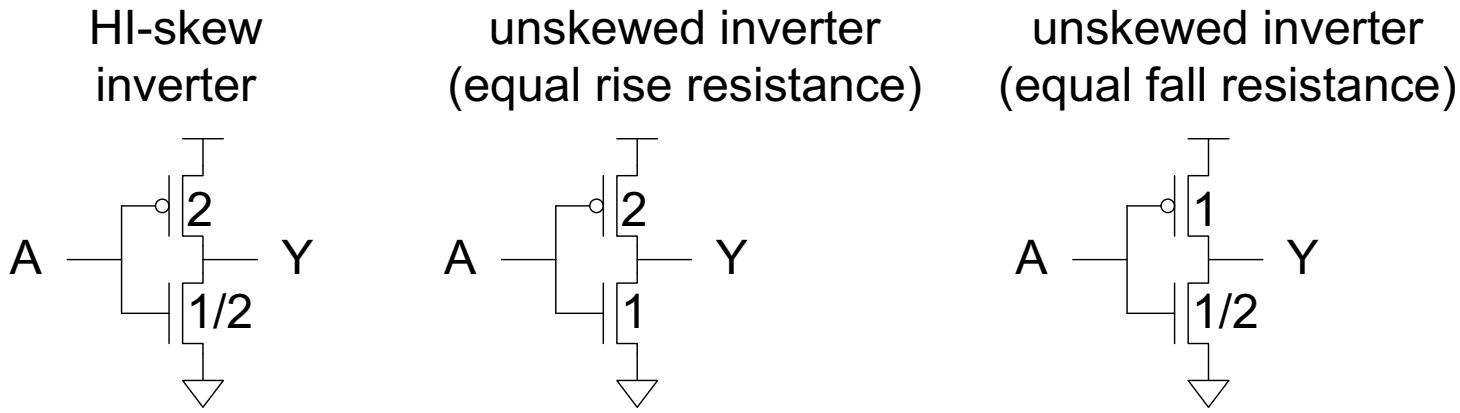




- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up

Skewed Gates

- Skewed gates favor one edge over another
- **Ex:** suppose rising output of inverter is most critical
  - Downsize noncritical NMOS transistor

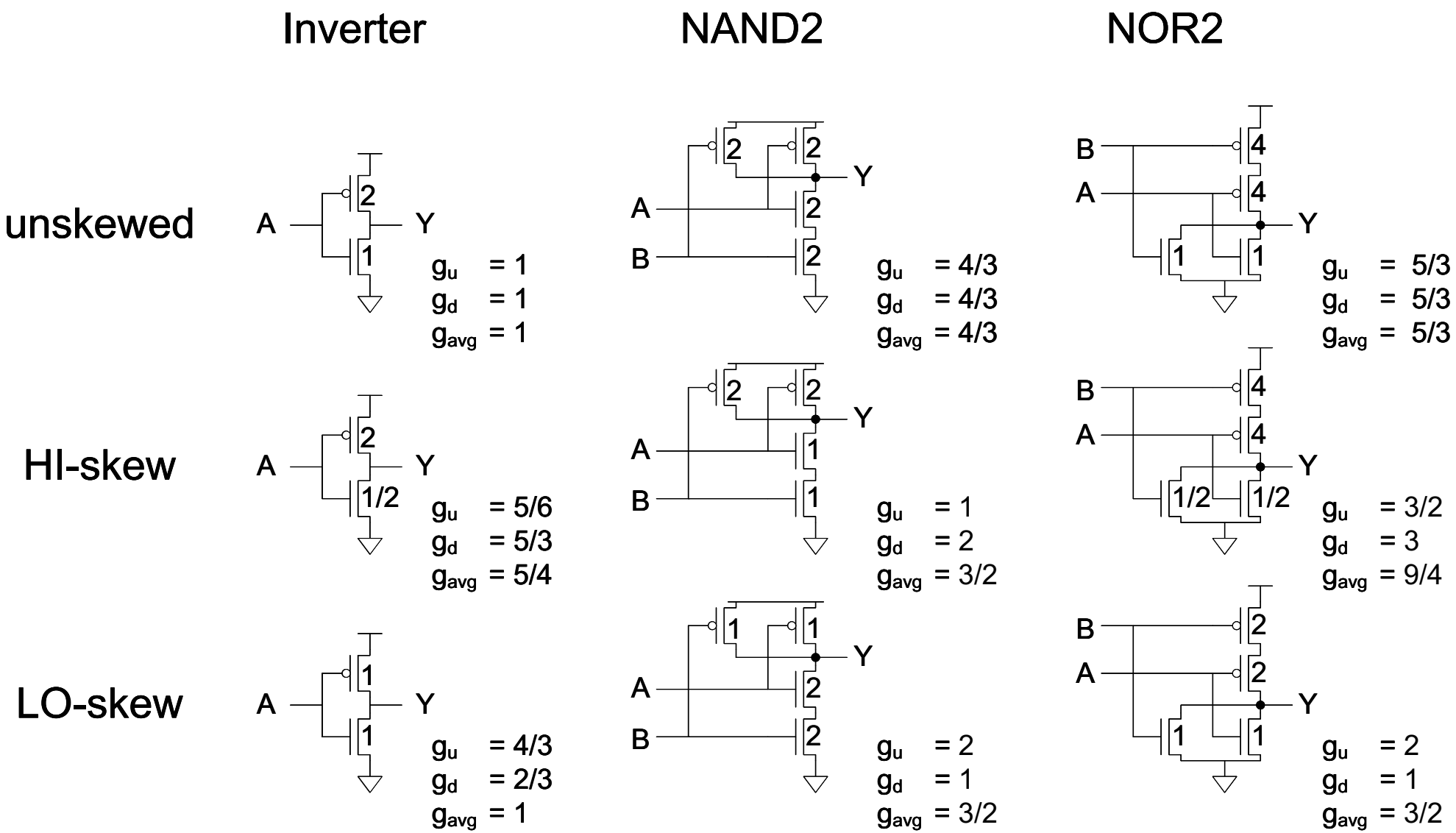


- Calculate logical effort by comparing to unskewed inverter with same effective R on that edge.
  - g_u =
  - g_d =

HI- and LO- Skew

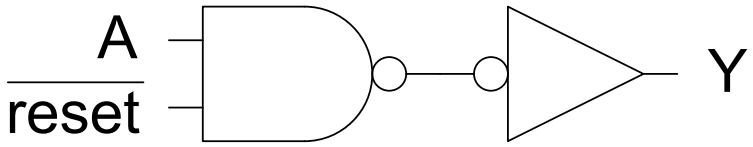
- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.
- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small NMOS)
  - LO-skew gates favor falling output (small PMOS)
- Logical effort is smaller for favored direction
- But larger for the other direction

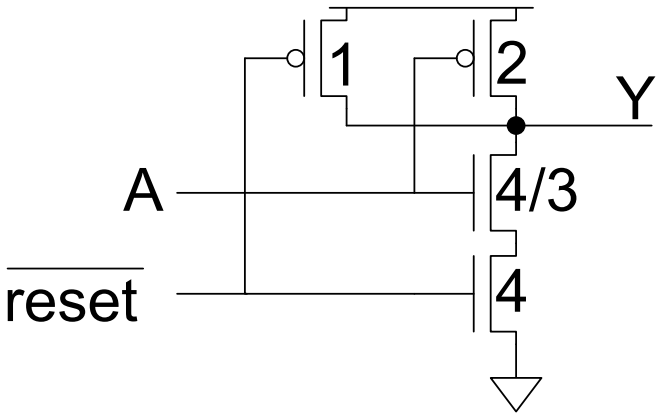
Catalog of Skewed Gates



Asymmetric Skew

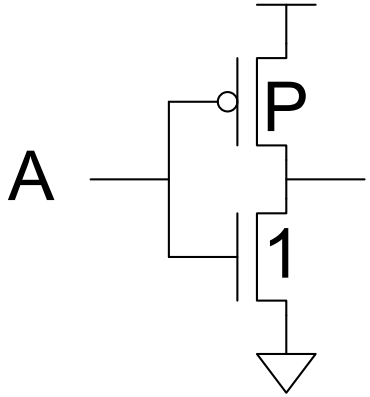
- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input





Best P/N Ratio

- We have selected P/N ratio for unit rise and fall resistance (μ = 2-3 for an inverter).
- Alternative: choose ratio for **least average delay**
- **Ex:** inverter
  - Delay driving identical inverter
  - t_{pdf} =
  - t_{pdr} =
  - t_{pd} =
  - Differentiate t_{pd} w.r.t. P
  - Least delay for P =

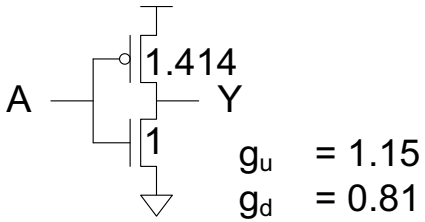


P/N Ratios

- In general, best P/N ratio is **square root of that giving equal delay**
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power

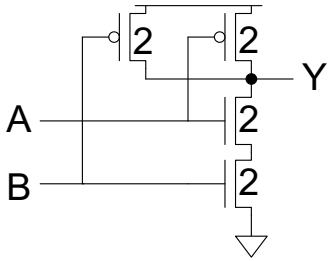
fastest P/N ratio

Inverter



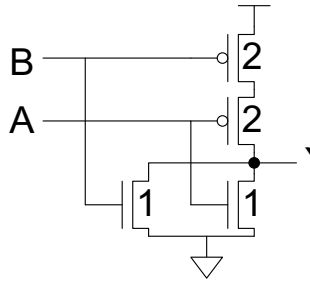
$g_u = 1.15$
$g_d = 0.81$
$g_{avg} = 0.98$

NAND2



$g_u = 4/3$
$g_d = 4/3$
$g_{avg} = 4/3$

NOR2



$g_u = 2$
$g_d = 1$
$g_{avg} = 3/2$

Observations

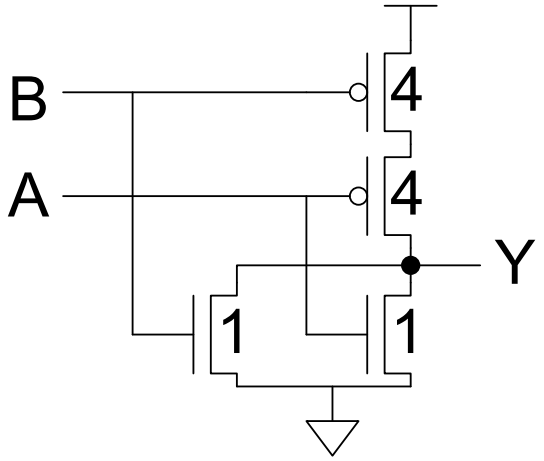
- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages

Outline

- Static CMOS
- **Ratioed circuits**
- Cascode voltage switch logic
- Dynamic circuits
- Pass-transistor circuits
- Circuit pitfalls

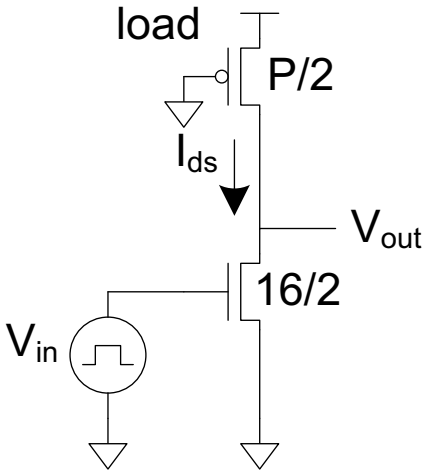
Introduction

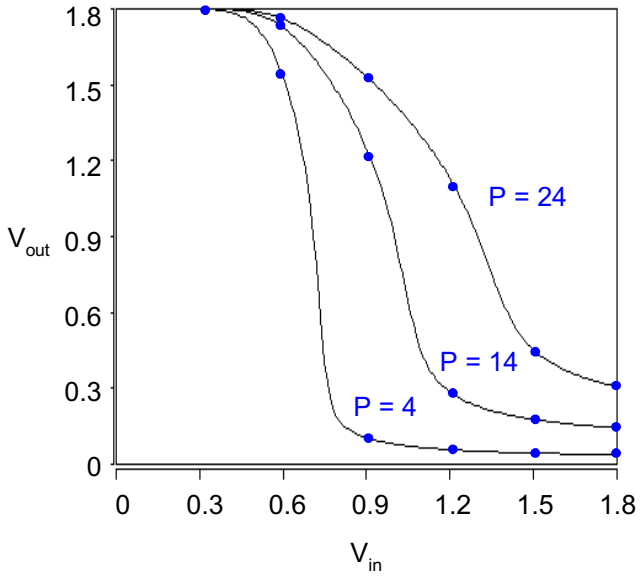
- What makes a circuit fast?
  - $I = C\,dV/dt \rightarrow t_{pd} \propto (C/I)\,\Delta V$
  - low capacitance
  - high current
  - small swing
- Logical effort is proportional to C
- PMOS are the enemy!
  - High capacitance for a given current
- Can we take the PMOS capacitance off the input?
- Various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



Pseudo-NMOS

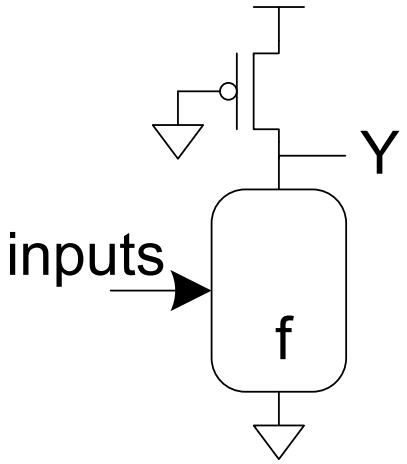
- In the old days, NMOS processes had no PMOS
  - Instead, use pull-up transistor that is always ON
- In CMOS, use a PMOS that is always ON
  - *Ratio* issue, Make PMOS about 1/3~1/6 effective strength of pulldown network



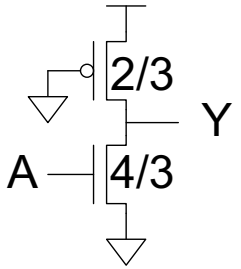


Pseudo-NMOS Gates

- Design for unit current on output to compare with unit inverter.
- Choose PMOS size between 1/3 ~ 1/6 the effective width (pick 1/3)

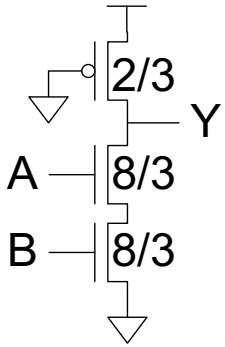


Inverter



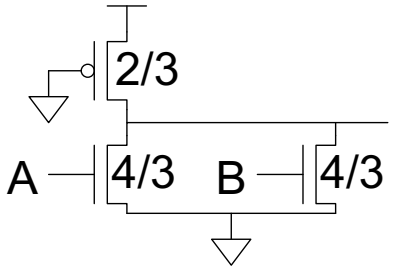
$g_u = 4/3$
$g_d = 4/9$
$g_{avg} = 8/9$
$p_u = 6/3$
$p_d = 6/9$
$p_{avg} = 12/9$

NAND2



$g_u = 8/3$
$g_d = 8/9$
$g_{avg} = 16/9$
$p_u = 10/3$
$p_d = 10/9$
$p_{avg} = 20/9$

NOR2

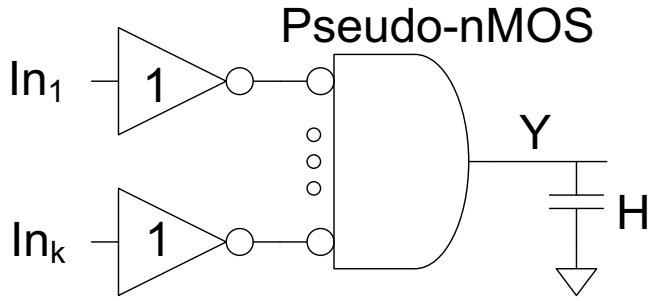


$g_u = 4/3$
$g_d = 4/9$
$g_{avg} = 8/9$
$p_u = 10/3$
$p_d = 10/9$
$p_{avg} = 20/9$

Pseudo-NMOS Design

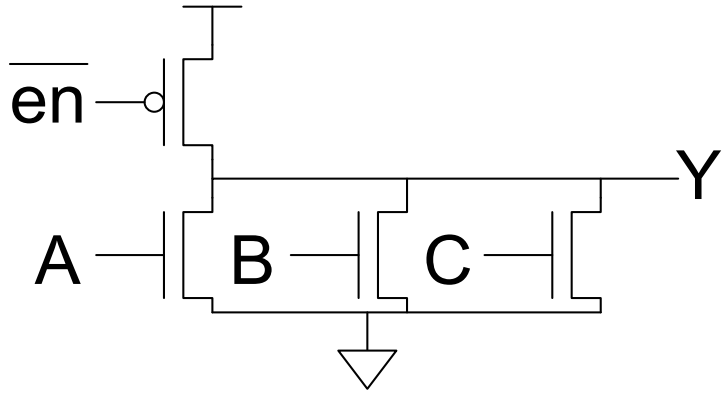
- Ex: Design a k-input AND gate using INV+ pseudo-NMOS NOR. Find delay driving a fanout of H

- G =
- F =
- P =
- N =
- D =



Pseudo-NMOS Power

- Pseudo-nMOS draws power whenever Y = 0
  - Called static power $P = I*V_{DD}$
  - A few mA/gate * 1M gates would be a problem
  - This is why NMOS process went extinct!
- Use pseudo-NMOS sparingly for wide NORs
- Turn off PMOS when not in use

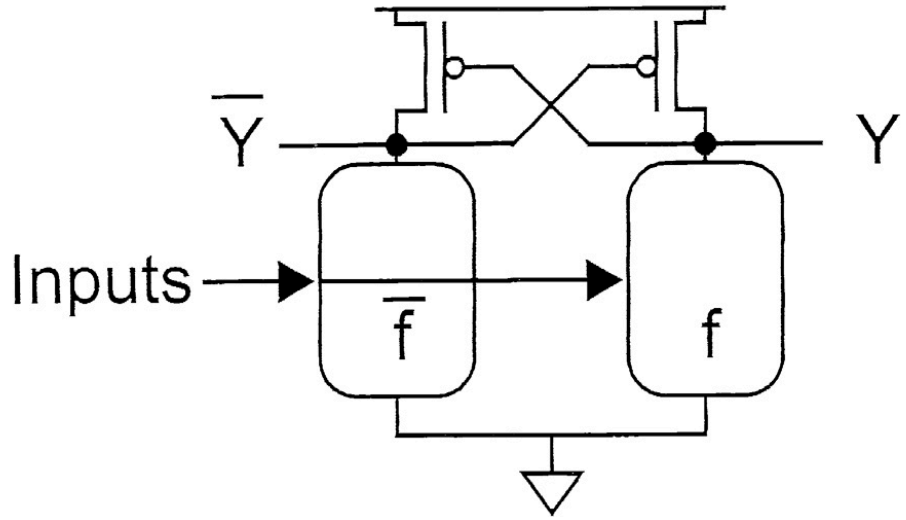


Outline

- Static CMOS
- Ratioed circuits
- **Cascode voltage switch logic**
- Dynamic circuits
- Pass-transistor circuits
- Circuit pitfalls

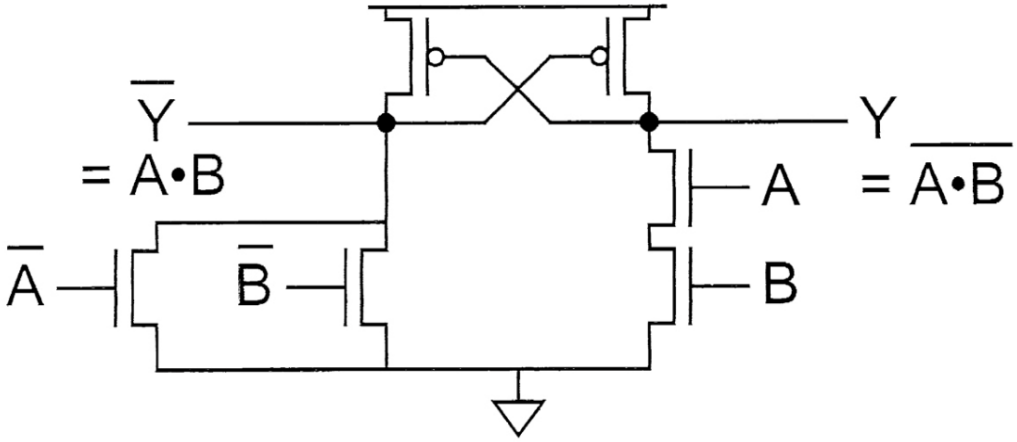
Cascode Voltage Switch Logic

- **D**ifferential **C**ascode **V**oltage **S**witching **L**ogic
  - Seeks the performance of ratioed circuits without the static power consumption
  - Use both true and complementary input signals and compute both true and complementary outputs

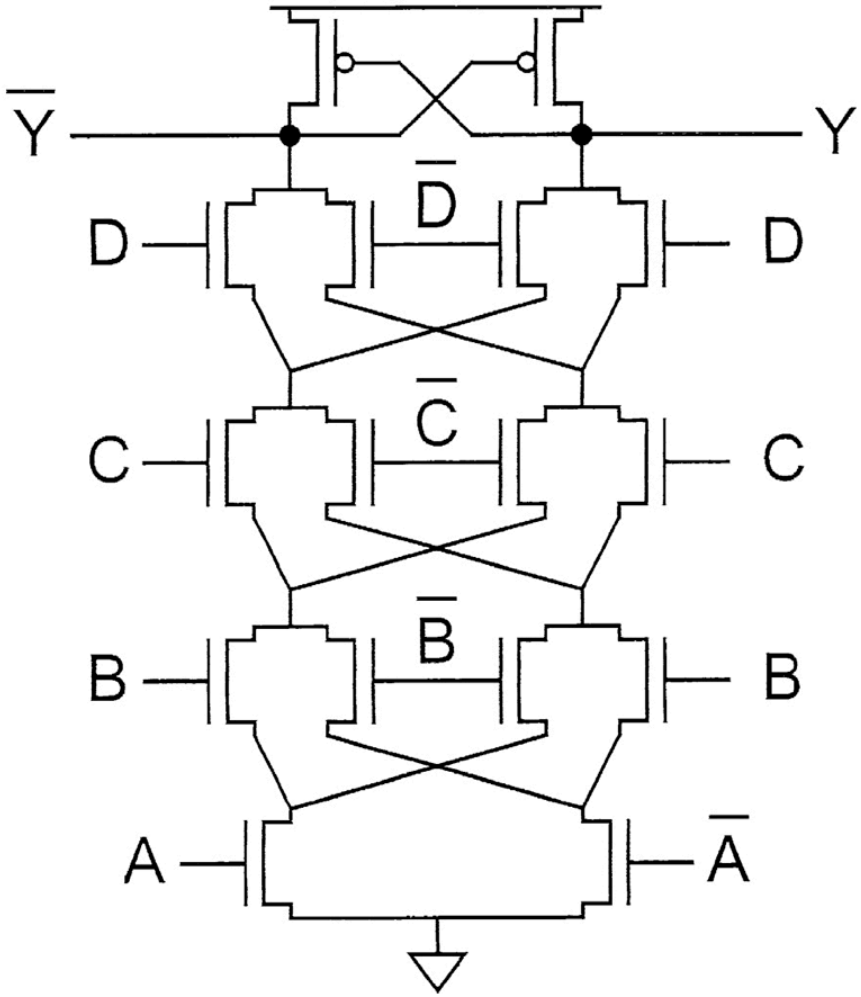


Cascode Voltage Switch Logic

2-input AND/NAND gate



4-input XOR/XNOR gate



Outline

- Static CMOS
- Ratioed circuits
- Cascode voltage switch logic
- **Dynamic circuits**
- Pass-transistor circuits
- Circuit pitfalls

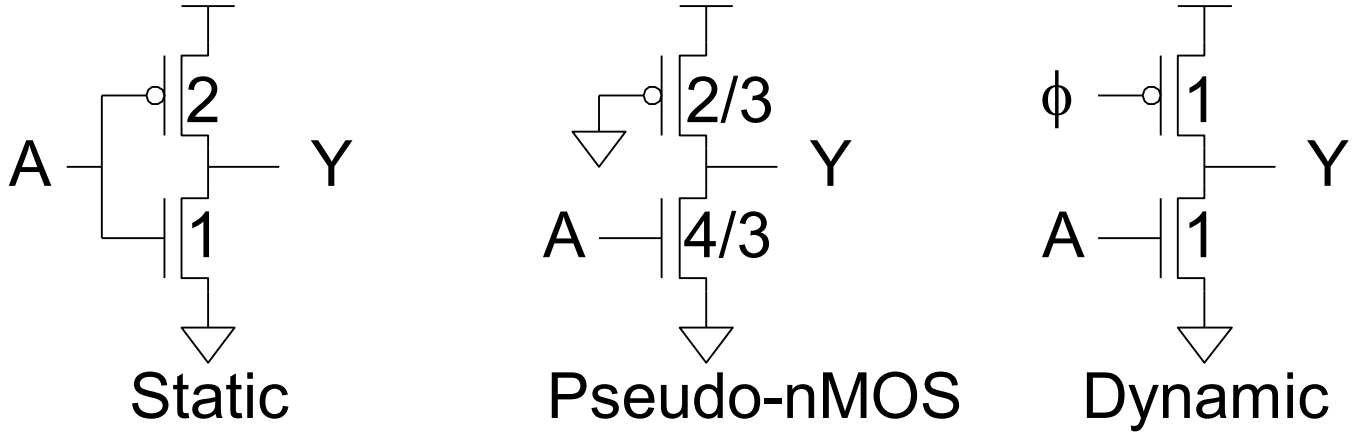
Dynamic CMOS

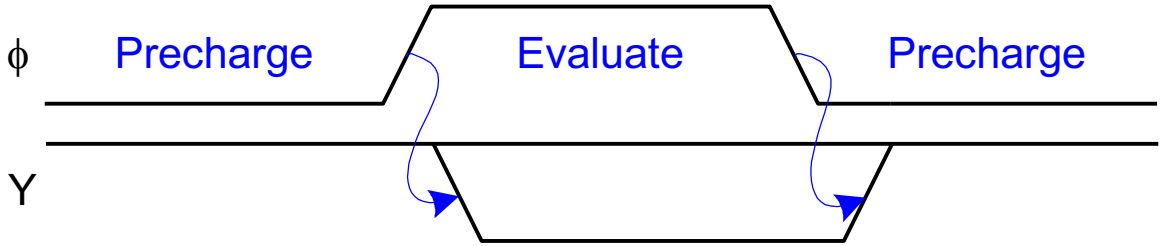
- In **static** circuits at every point in time (except when switching) the output is connected to either GND or V_{DD} via a low resistance path.
  - Fan-in of N requires 2N devices

- **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 Logic

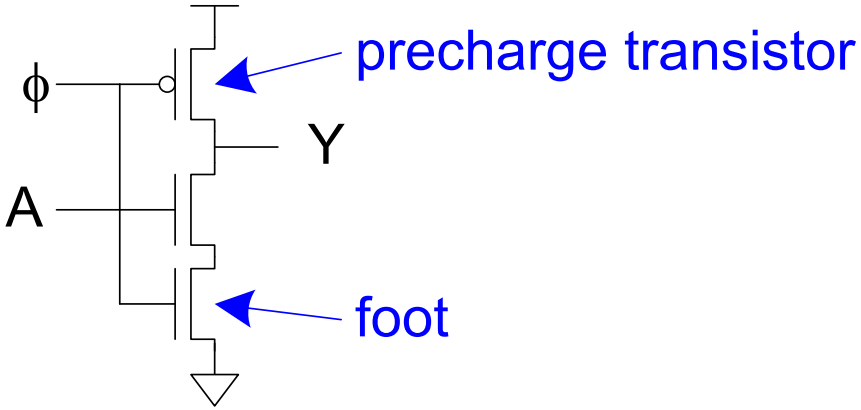
- **Dynamic** gates uses a clocked PMOS pullup
- Two modes: precharge and evaluate

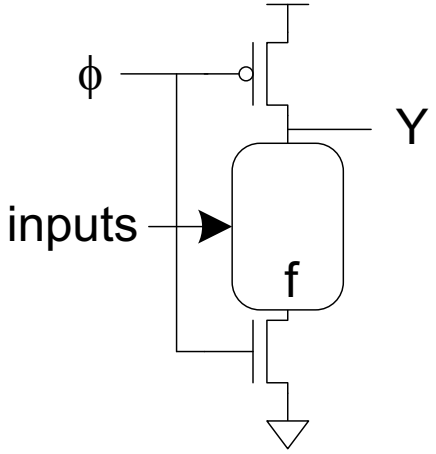




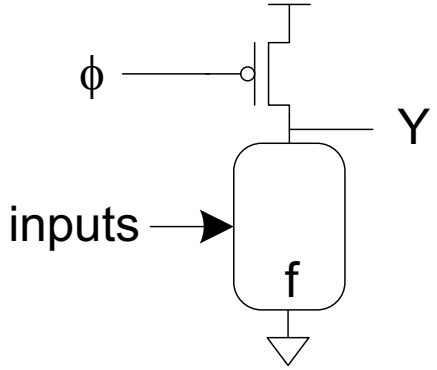
The Foot

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



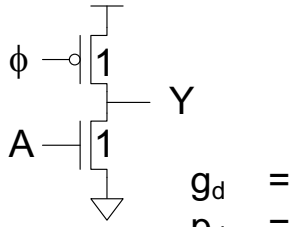
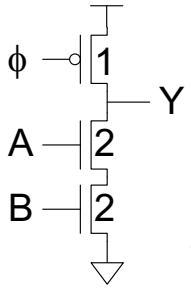
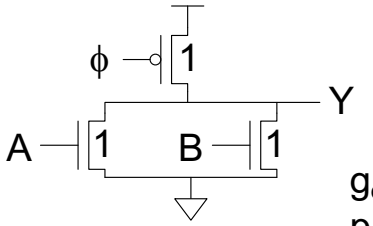
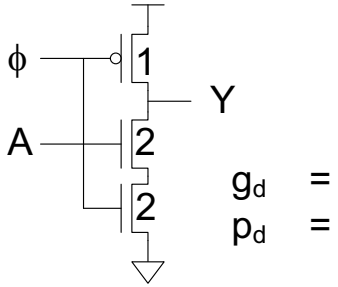
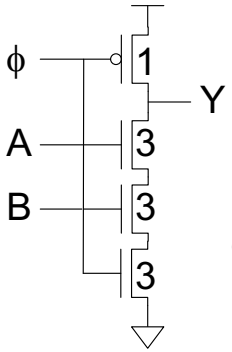
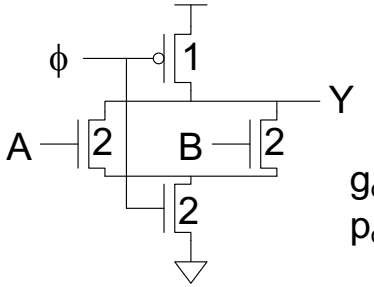


footed



unfooted

Logical Effort

	Inverter	NAND2	NOR2
unfooted	 g_d = p_d =	 g_d = p_d =	 g_d = p_d =
footed	 g_d = p_d =	 g_d = p_d =	 g_d = p_d =

Conditions on Output

- Once output of a dynamic gate is discharged, it cannot be charged again until next precharge phase
- 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

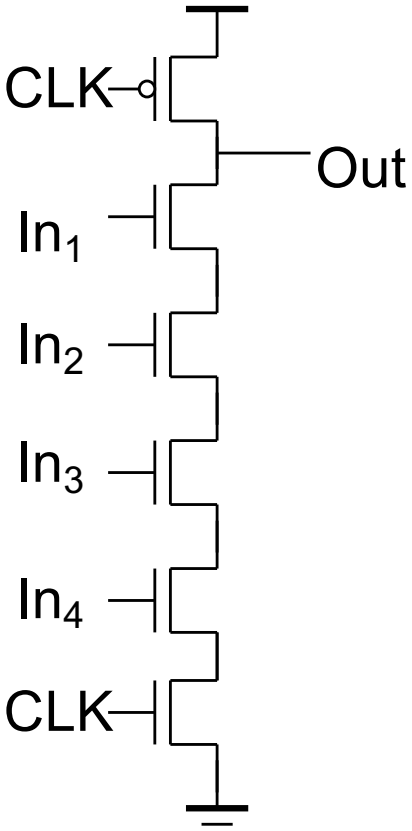
Properties of Dynamic Gates (I)

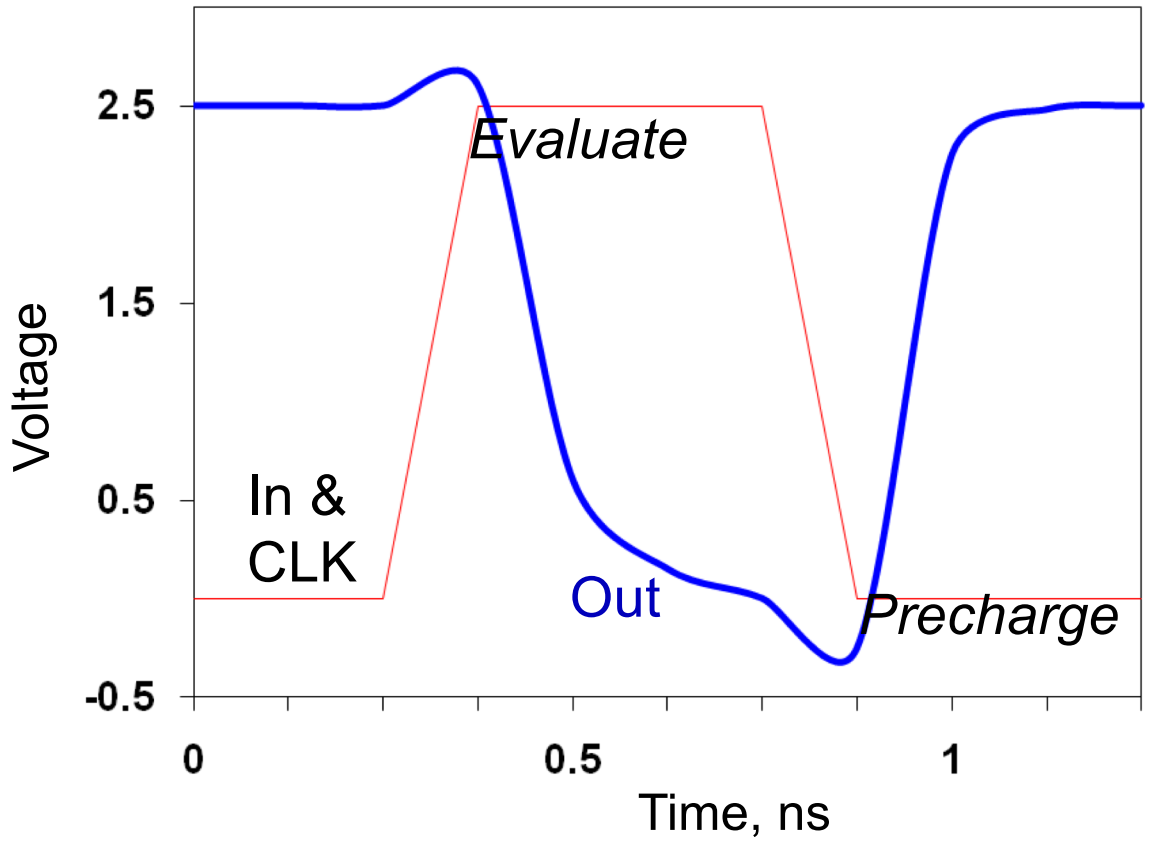
- Logic function is implemented by the PDN only
  - Number of transistors is N + 2 (versus 2N for static CMOS)
  - Should be smaller in area than static CMOS
- Full output swing (V_{OL} = GND and V_{OH} = V_{DD})
- Nonratioed – size of devices is not important for proper functioning (only for performance)
- Faster switching speed
  - Reduced load capacitance due to lower number of transistors per gate (C_{int}), leading to reduced logical effort
  - Reduced load capacitance due to smaller fan-out (C_{ext})
  - No I_{sc}, so all the current provided by PDN goes into discharging C_L
  - t_{PLH} = 0 by ignoring the time for precharge, but the presence of evaluation transistor slows down t_{PHL}

Properties of Dynamic Gates (II)

- Lower power dissipation
  - Consumes only dynamic power – no short-circuit current since the pull-up path is not ON during evaluation
  - lower capacitance – both C_{int} (since there are fewer transistors connected to the drain output) and C_{ext} (since there the output load is one per connected gate, not two)
  - At most one transition per cycle by construction – no glitching
- But power dissipation can be significantly higher
  - Higher transition probabilities
  - Extra load on CLK
- PDN starts to work as soon as the input signals exceed V_{Tn}, so set V_M, V_{IH} and V_{IL} all equal to V_{Tn}
  - Low noise margin (NM_L)
- Need a precharge clock

Dynamic Behavior

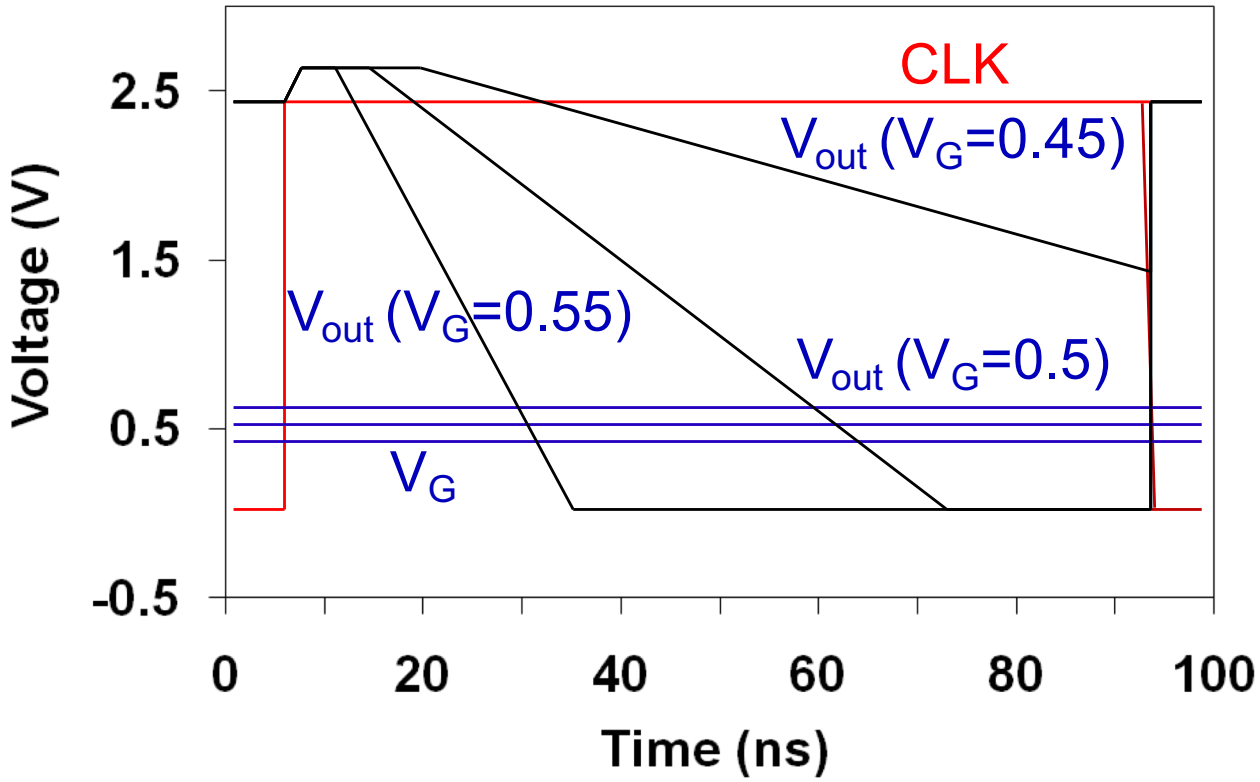




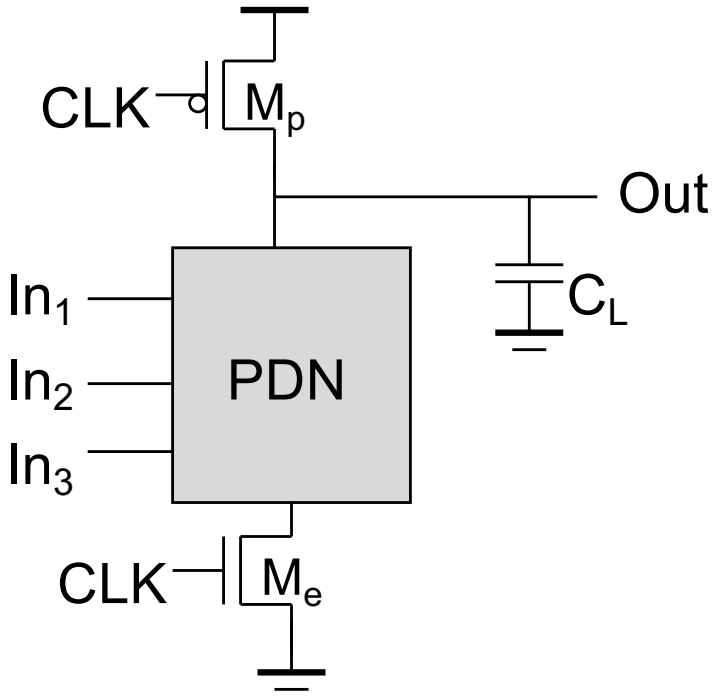
#Trns	V_{OH}	V_{OL}	V_M	NM_H	NM_L	t_{pHL}	t_{pLH}	t_p
6	2.5V	0V	V_{Tn}	2.5-V_{Tn}	V_{Tn}	110ps	0ns	83ps

Time-Independent Gate Parameters

- The amount by which the output voltage drops depends strongly on input voltage and available evaluation time
  - Noise needed to corrupt the signal increases for short evaluation time, i.e., the switching threshold is truly time-independent.



Power Consumption



Only dissipate power when previous output is LOW

Data Dependent Dynamic Power

Dynamic 2-input NOR Gate

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

Assume signal probabilities

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

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

and transition probability

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

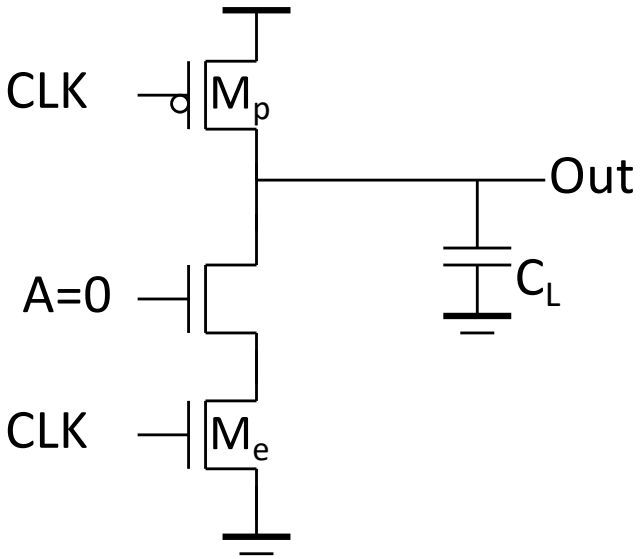
$= 3/4 \times 1$

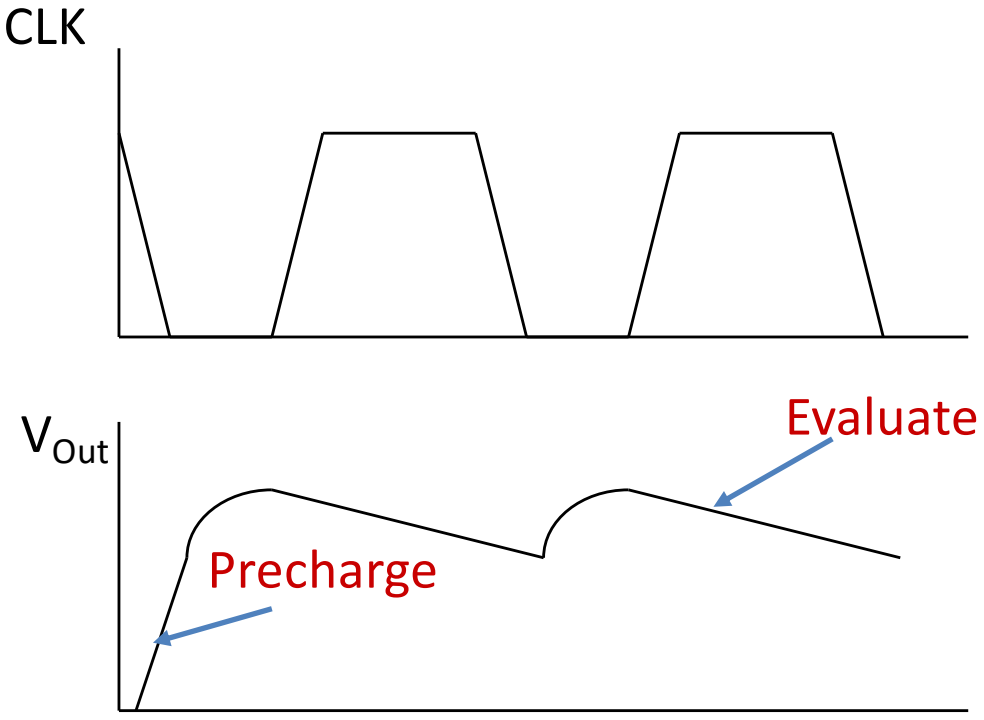
$= 3/4$

Switching activity can be higher in dynamic gates!

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

Charge Leakage

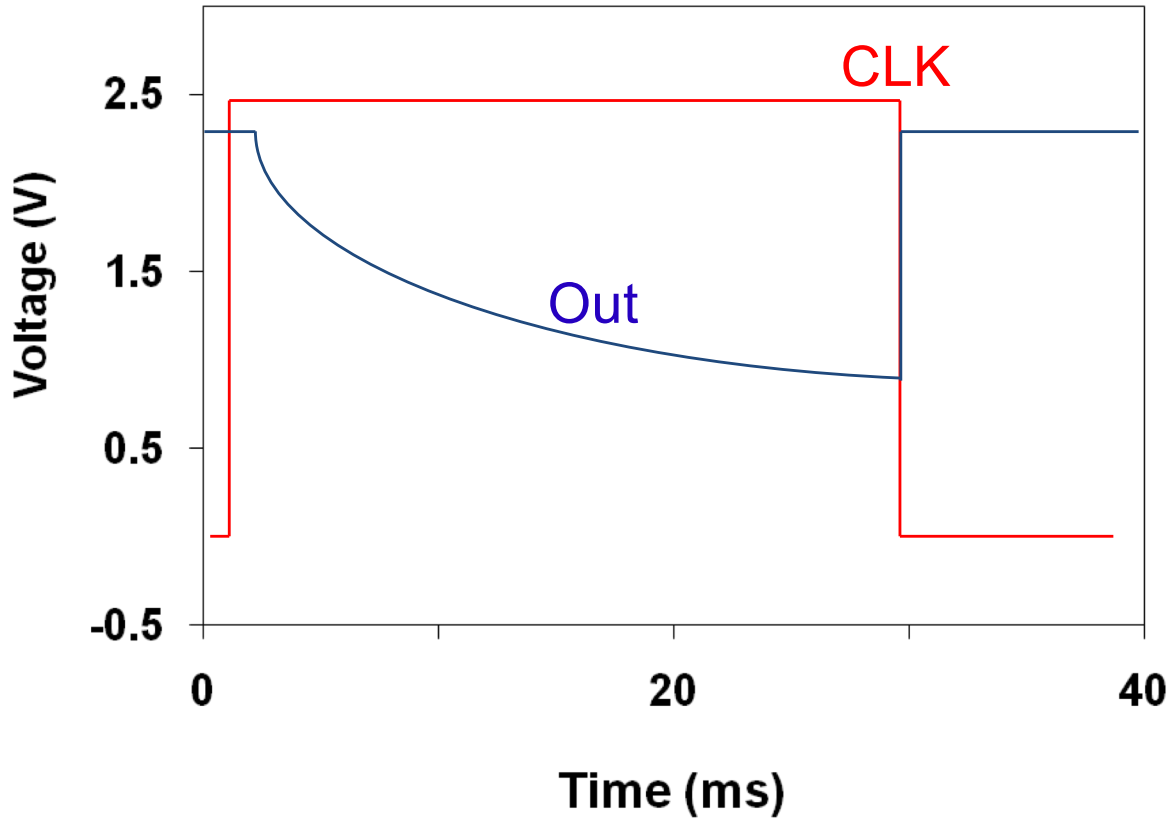




Minimum clock rate of a few kHz

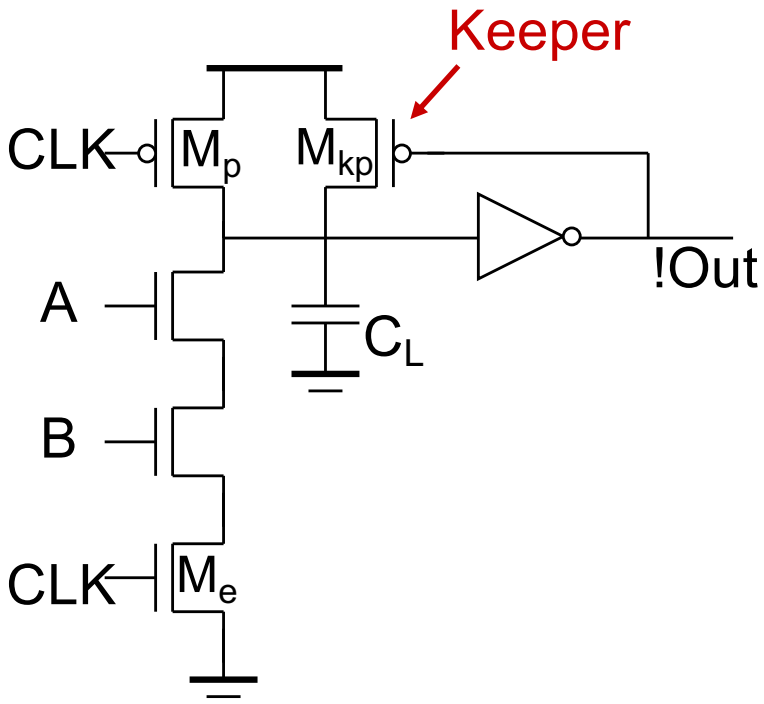
Impact of Charge Leakage

- 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



One Solution to Charge Leakage

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



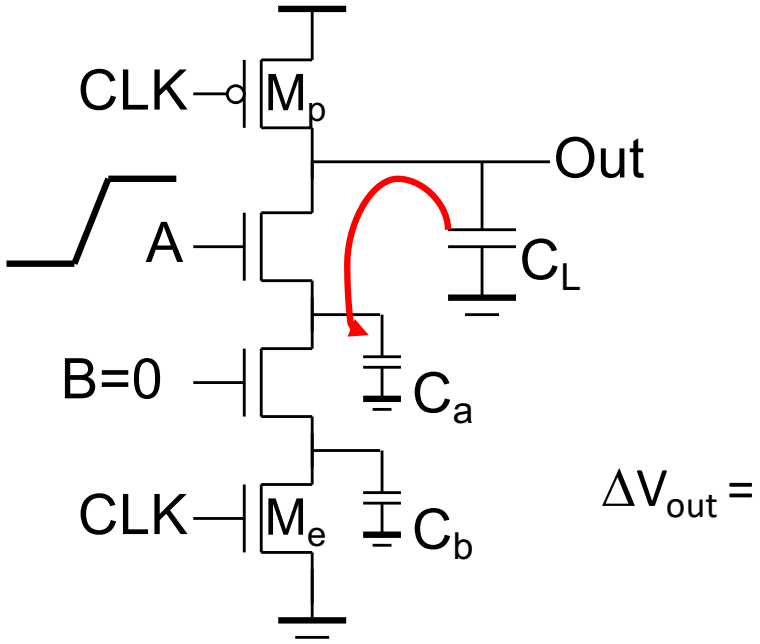
Same approach as level restorer for pass transistor logic

Summary: Leakage

- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{OFF} \neq 0$)
  - Dynamic data leak away over time
  - Formerly miliseconds, now nanoseconds!
- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation

Charge Sharing

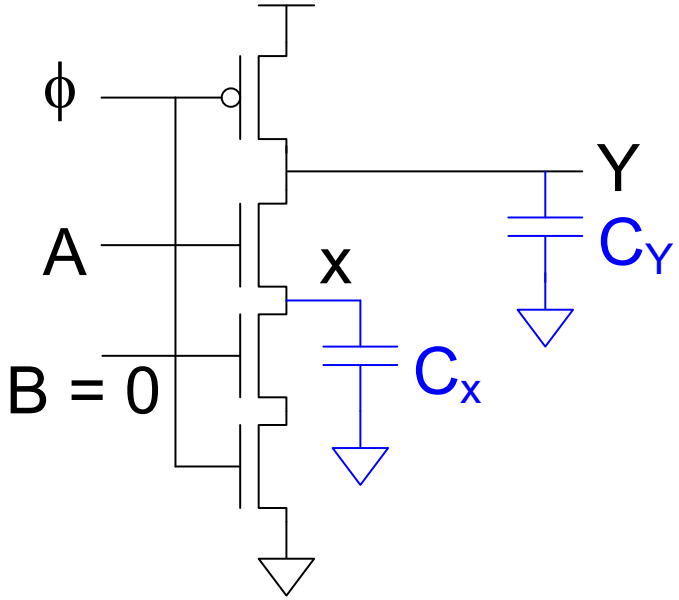
- Charge stored originally on C_L is redistributed (shared) over C_L and C_a leading to static power consumption by downstream gates and possible circuit malfunction

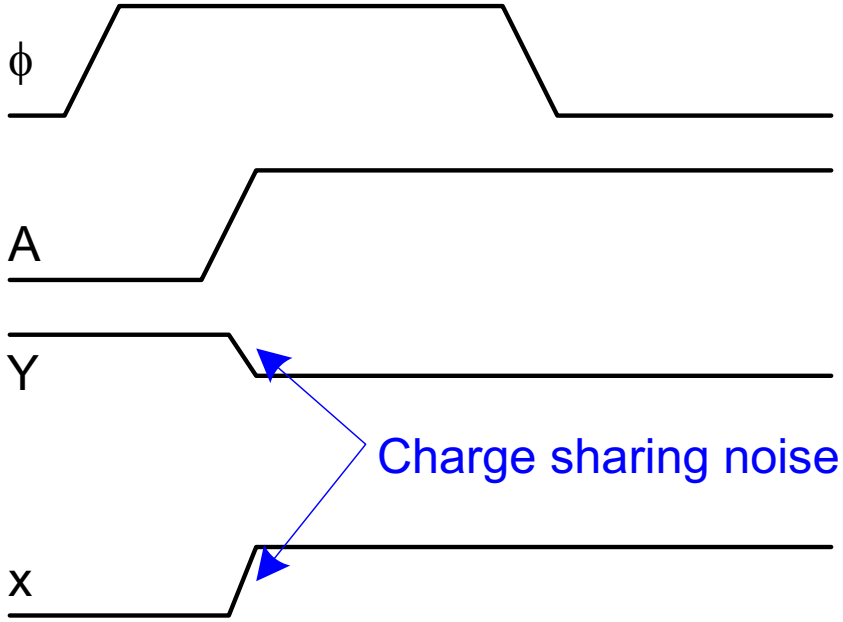


- V_{out} drops below the switching threshold of following gate and causes malfunction

Charge Sharing

- Dynamic gates suffer from charge sharing

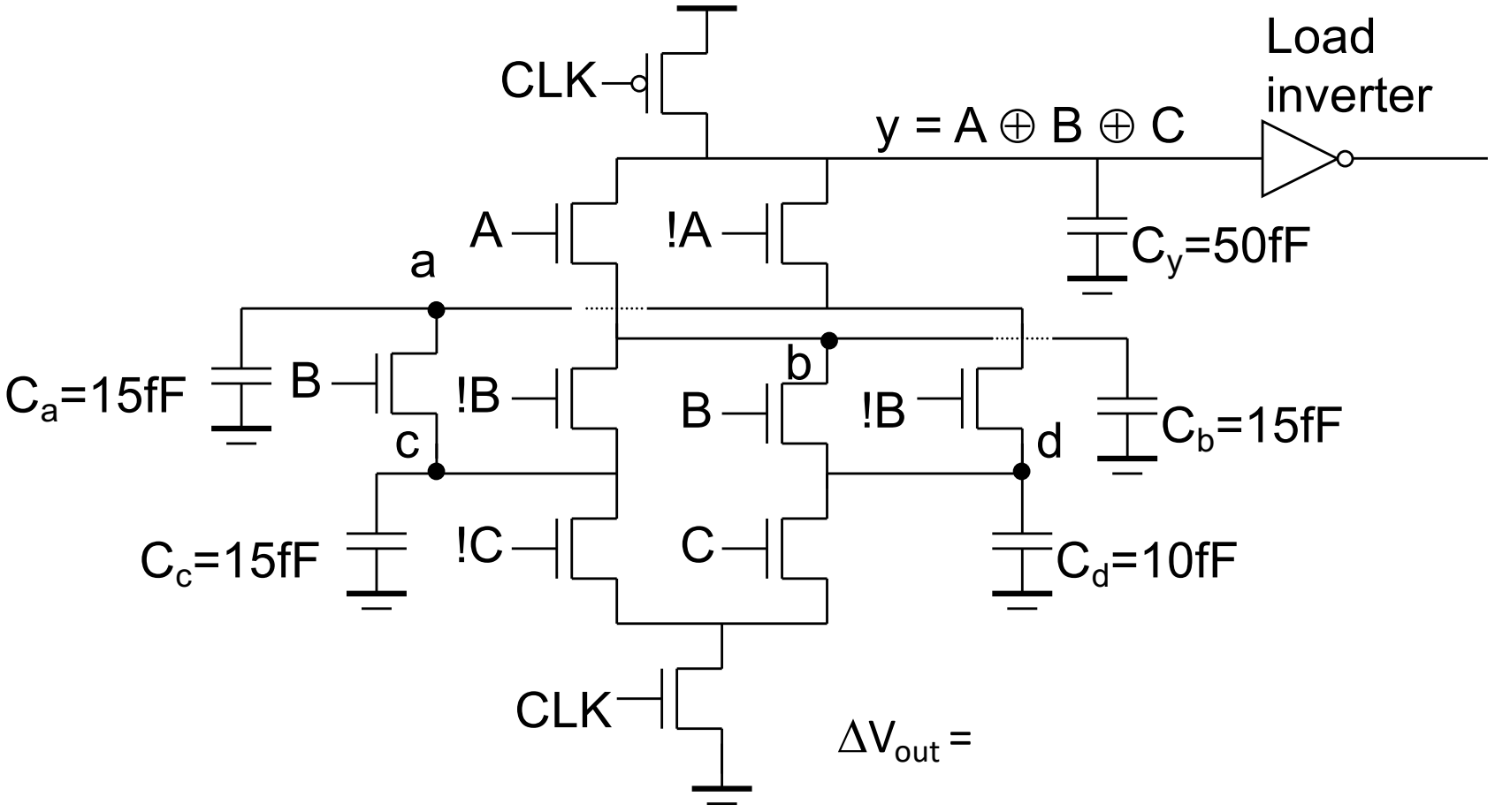




$$V_x = V_Y = \frac{C_Y}{C_x + C_Y} V_{DD}$$

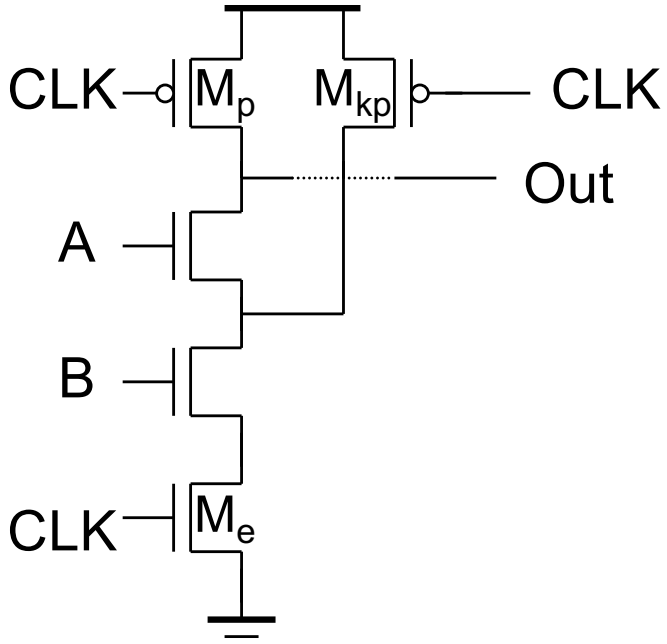
Charge Sharing Example

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



Solution to Charge Redistribution

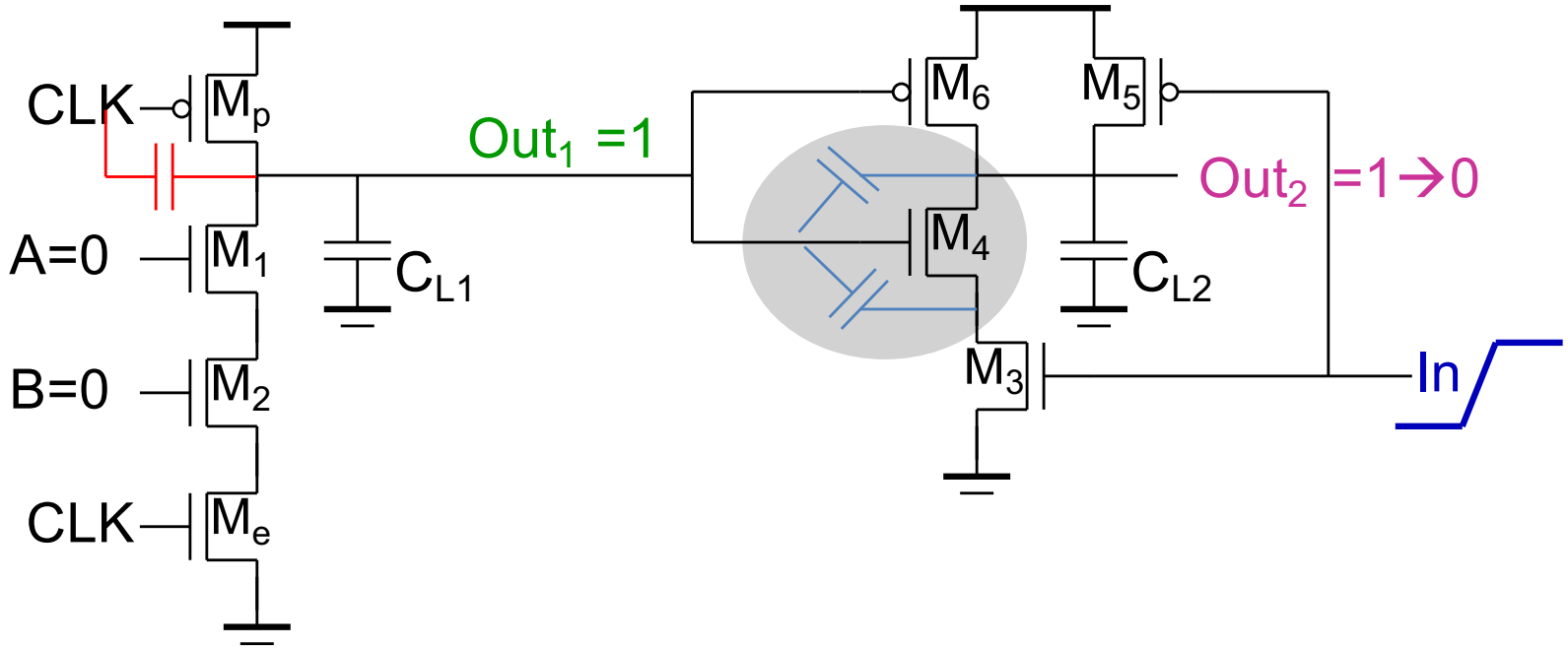
- Precharge internal nodes using a clock-driven transistor (secondary precharge transistor)
  - Typically need to precharge every other node
  - At the cost of increased area and power



- Large load capacitance C_{out} helps as well

Backgate Coupling

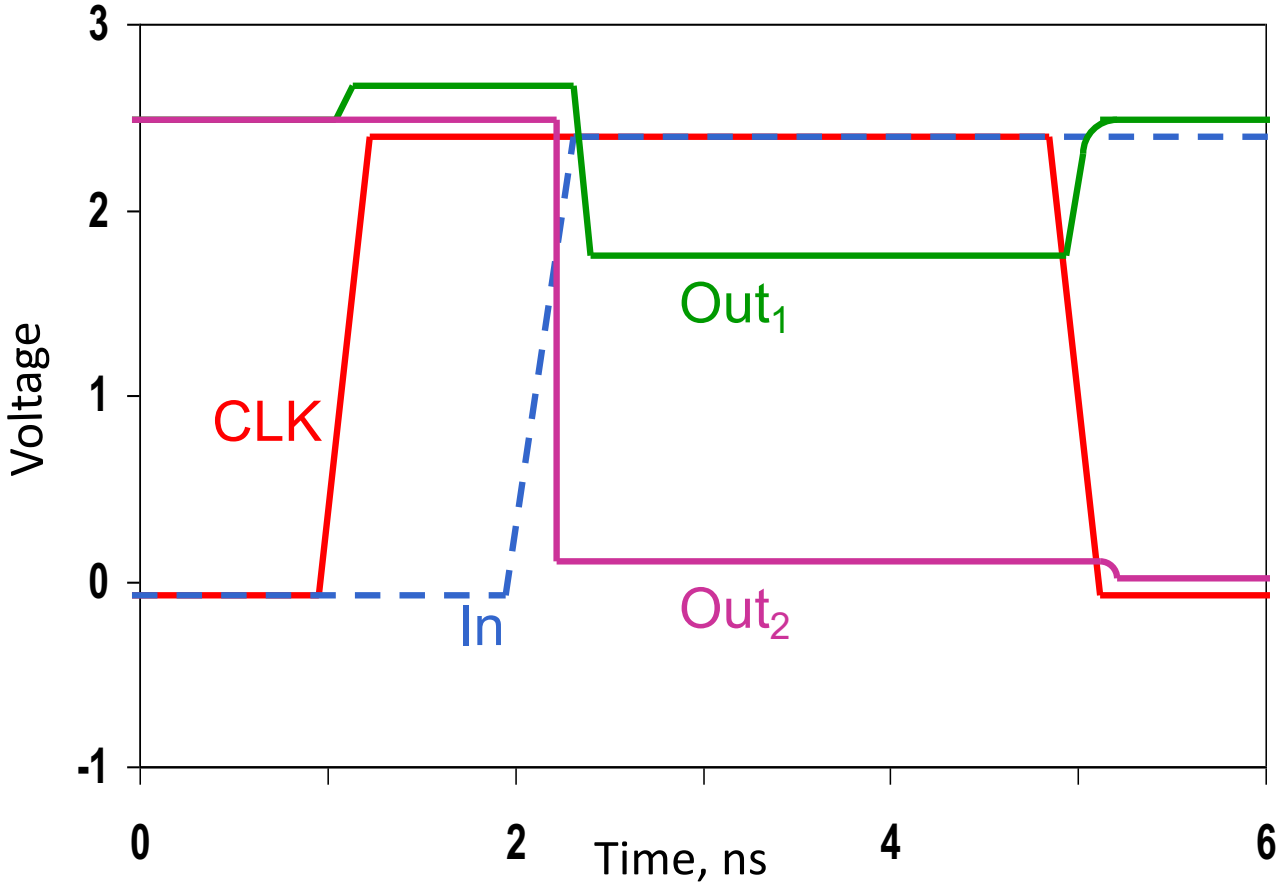
- Susceptible to crosstalk due to 1) high-impedance output node and 2) backgate capacitive coupling
  - Out_2 couples with Out_1 capacitively through the gate-source and gate-drain capacitances of M_4



Dynamic NAND Static NAND

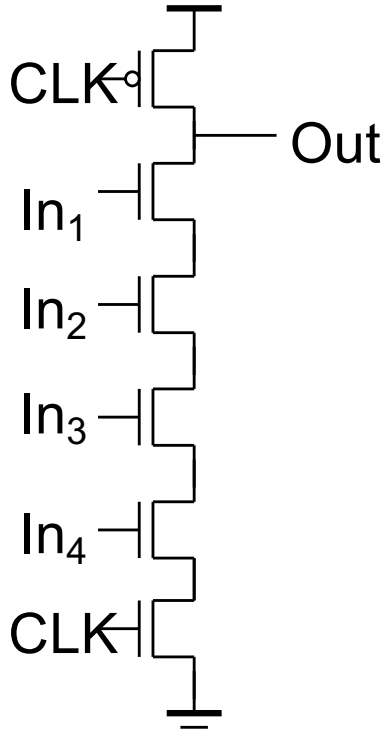
Impact of Backgate Coupling

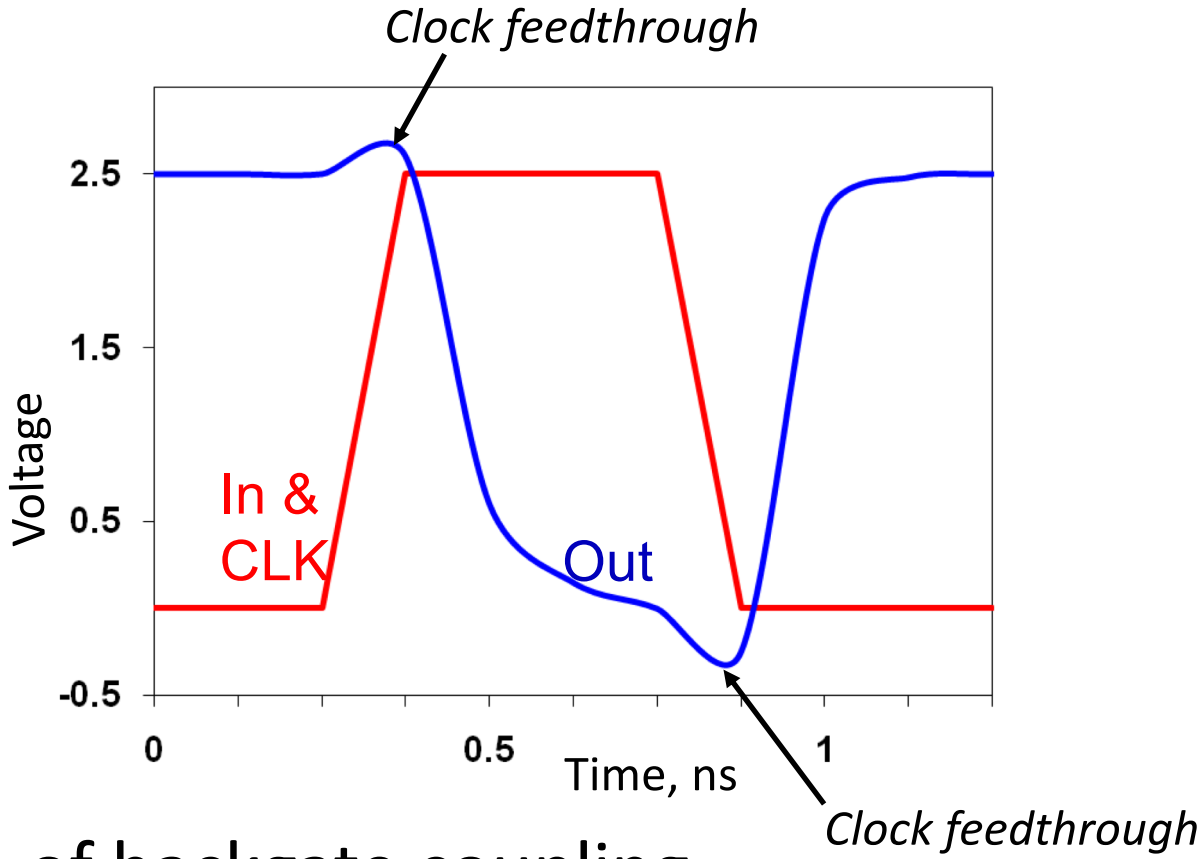
- Capacitive coupling means Out_1 drops significantly so Out_2 doesn't go all the way to ground



Clock Feedthrough

- The fast rising (and falling) edges of CLK couples to Out so that voltage of Out can rise above V_{DD} (or below GND)

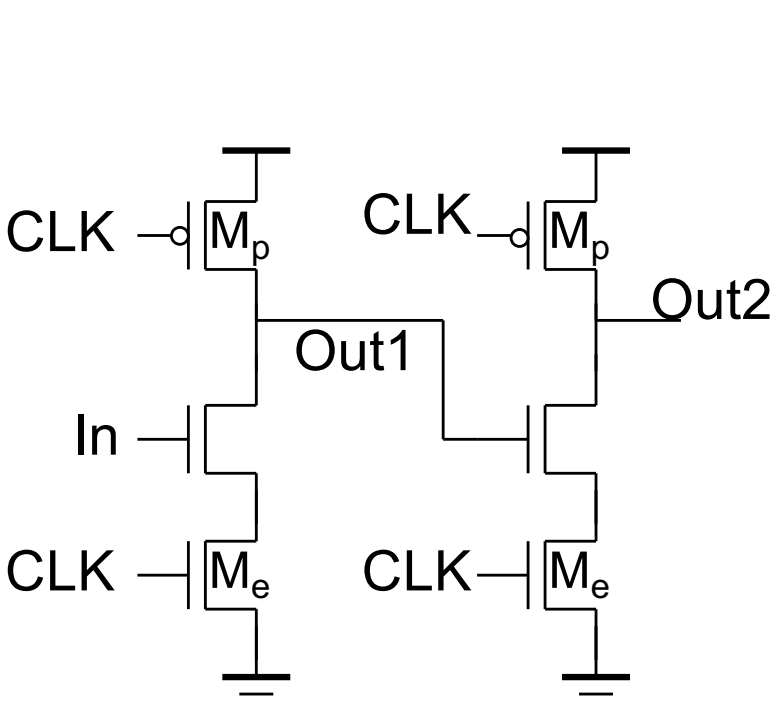


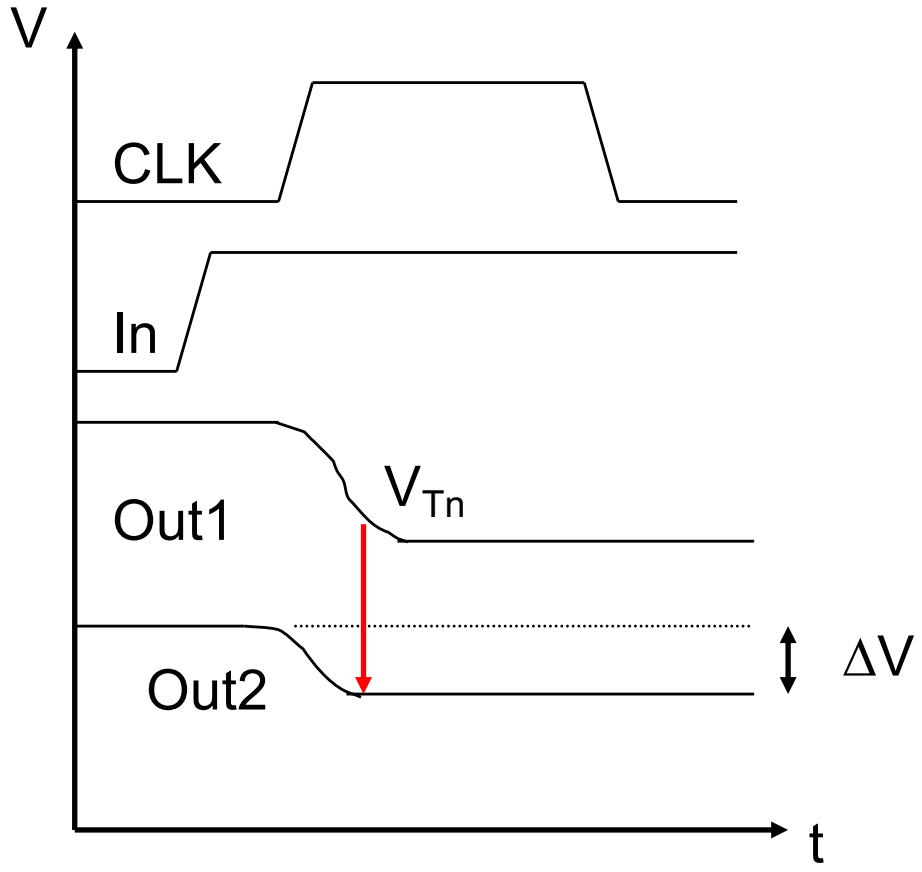


- A special case of backgate coupling

Cascading Gates

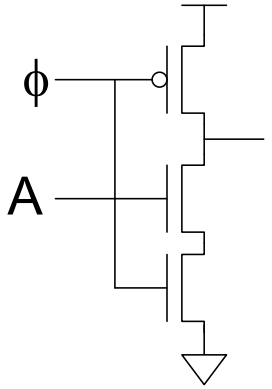
- Only one single 0→1 transition allowed at the inputs during evaluation phase

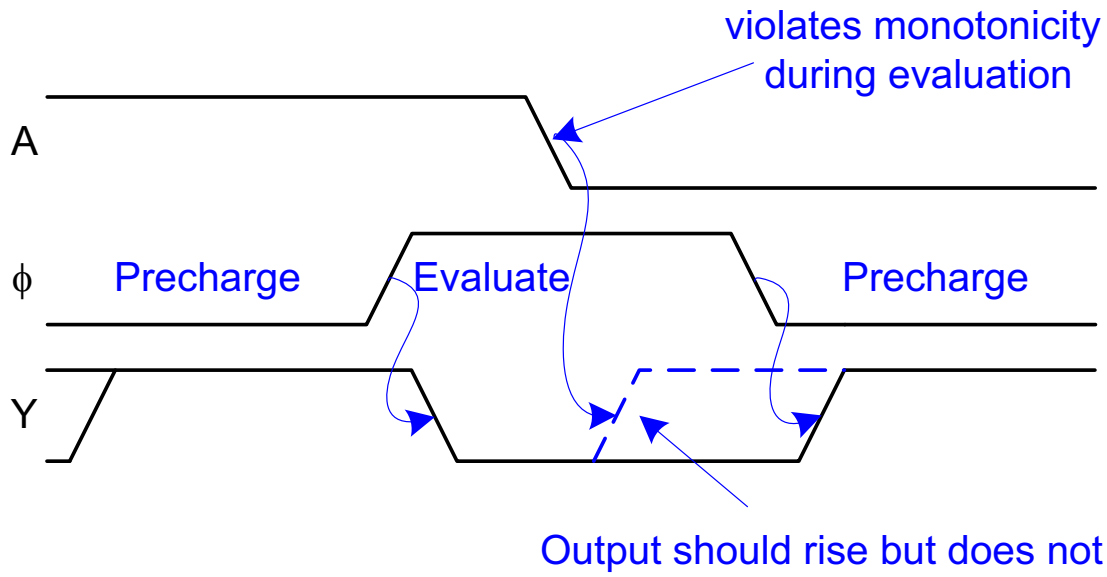




Monotonicity

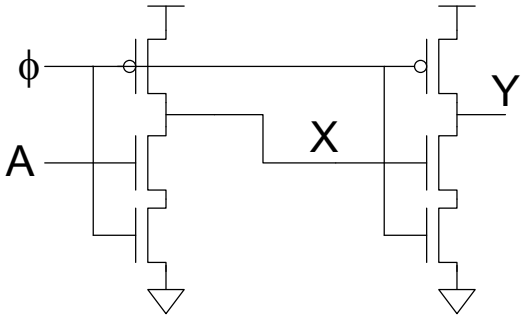
- Dynamic gates require *monotonically rising* inputs during evaluation
  - 0→0
  - 0→1
  - 1→1
  - But NOT 1→0

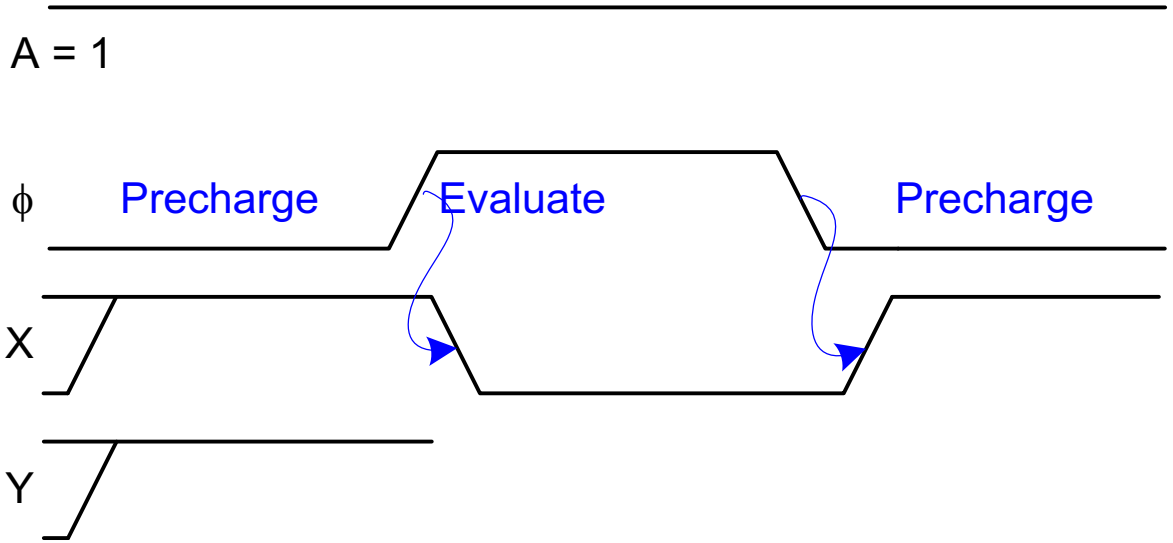




Monotonicity Woes

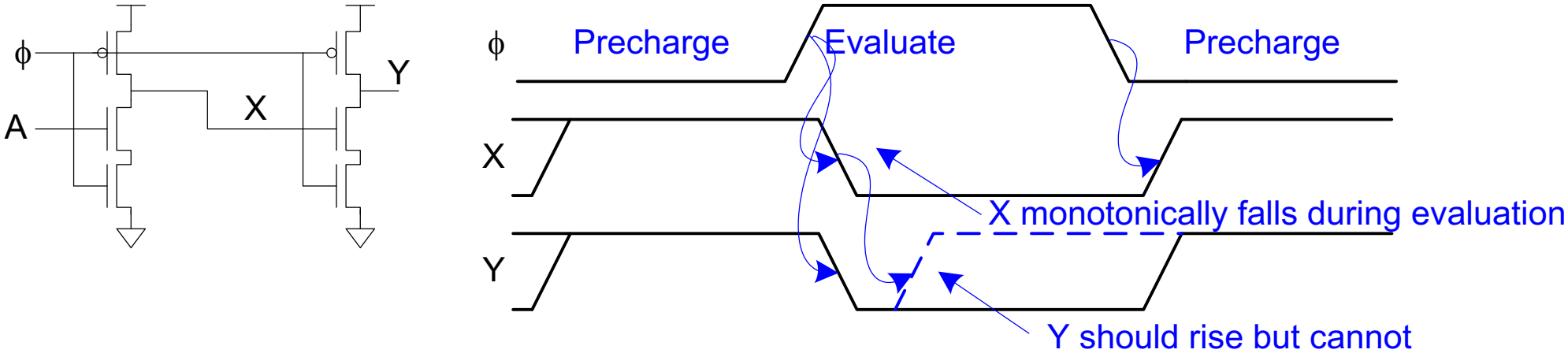
- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!





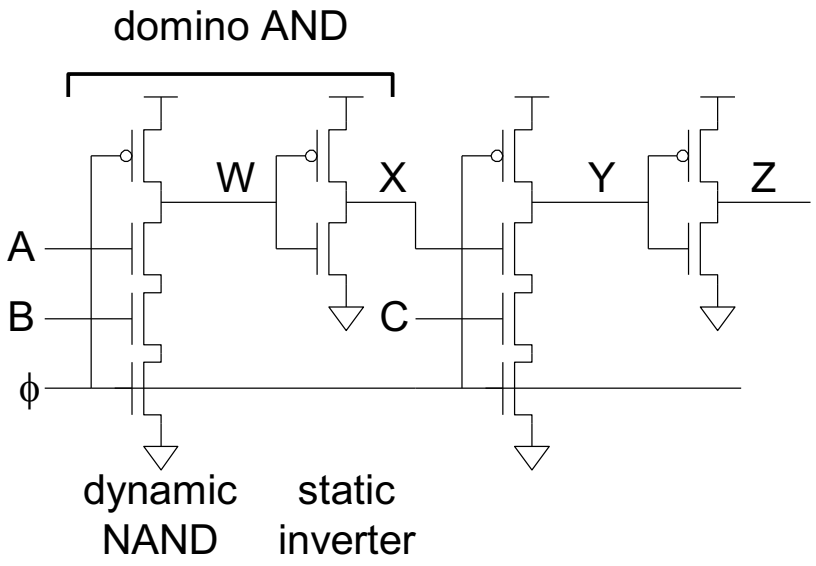
Monotonicity Woes

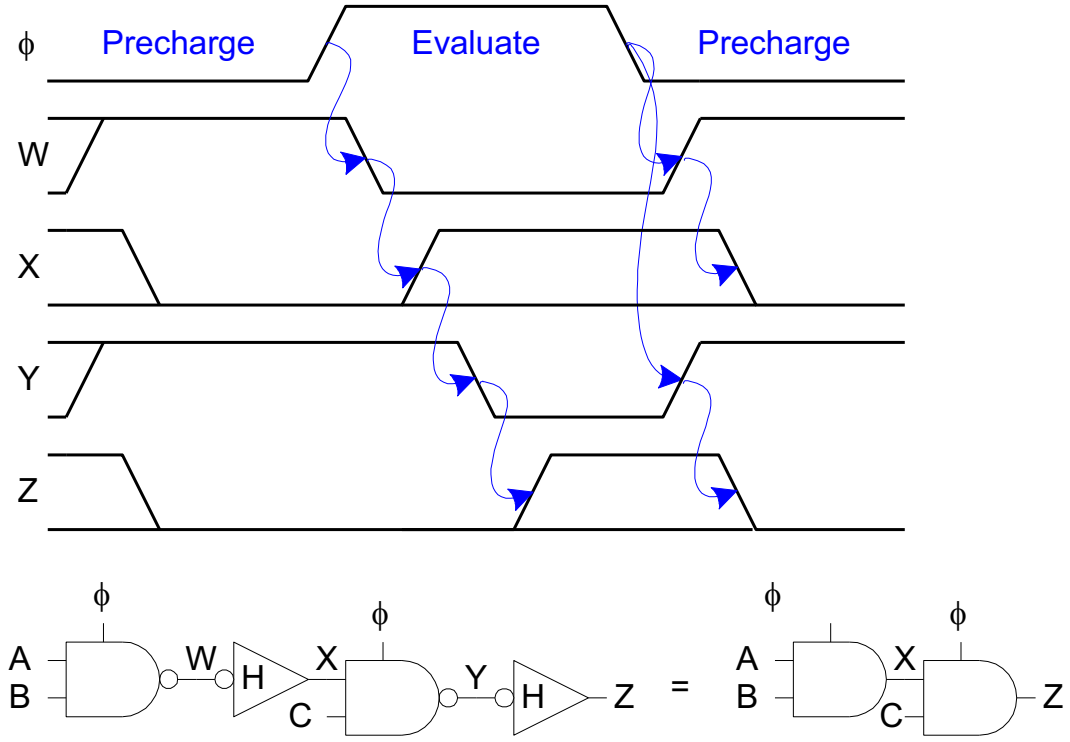
- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!



Domino Gates

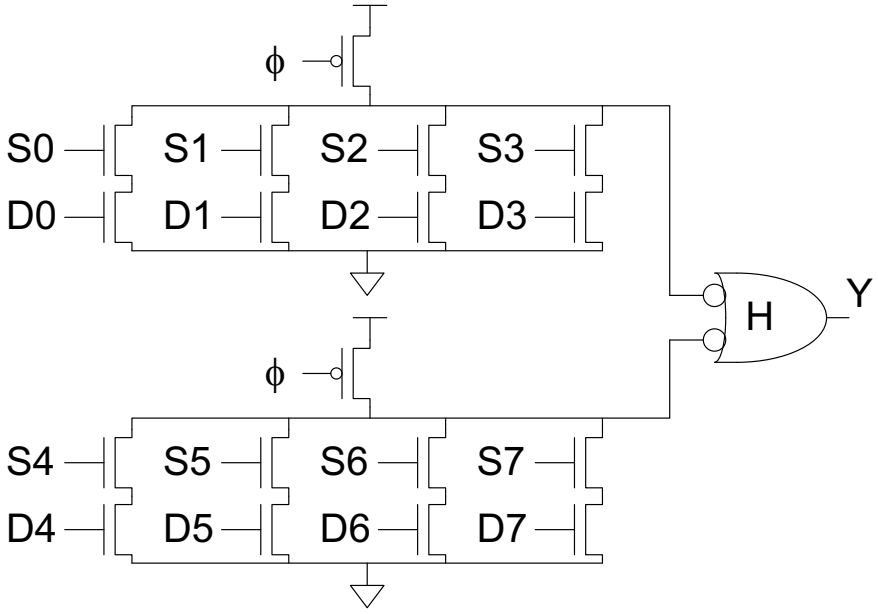
- Dynamic stage followed by inverting static gate
  - Dynamic/static pair is called domino gate





Domino Optimizations

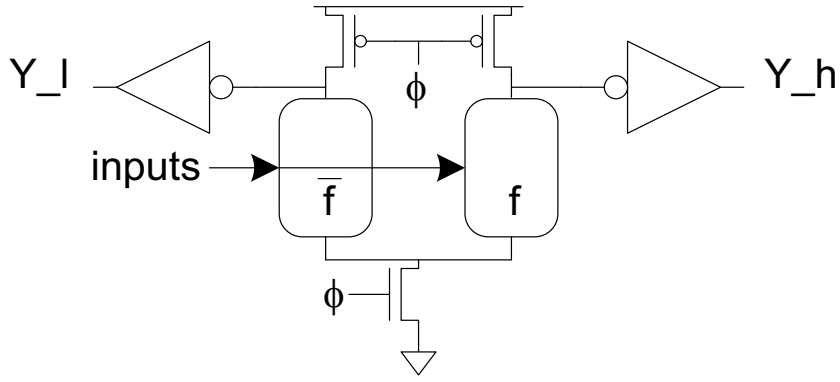
- Each domino gate triggers next one, like a string of dominos toppling over
- Gates precharge in parallel, evaluate sequentially
  - Evaluation is more critical than precharge
- HI-skewed static stages can perform logic



Dual-Rail Domino

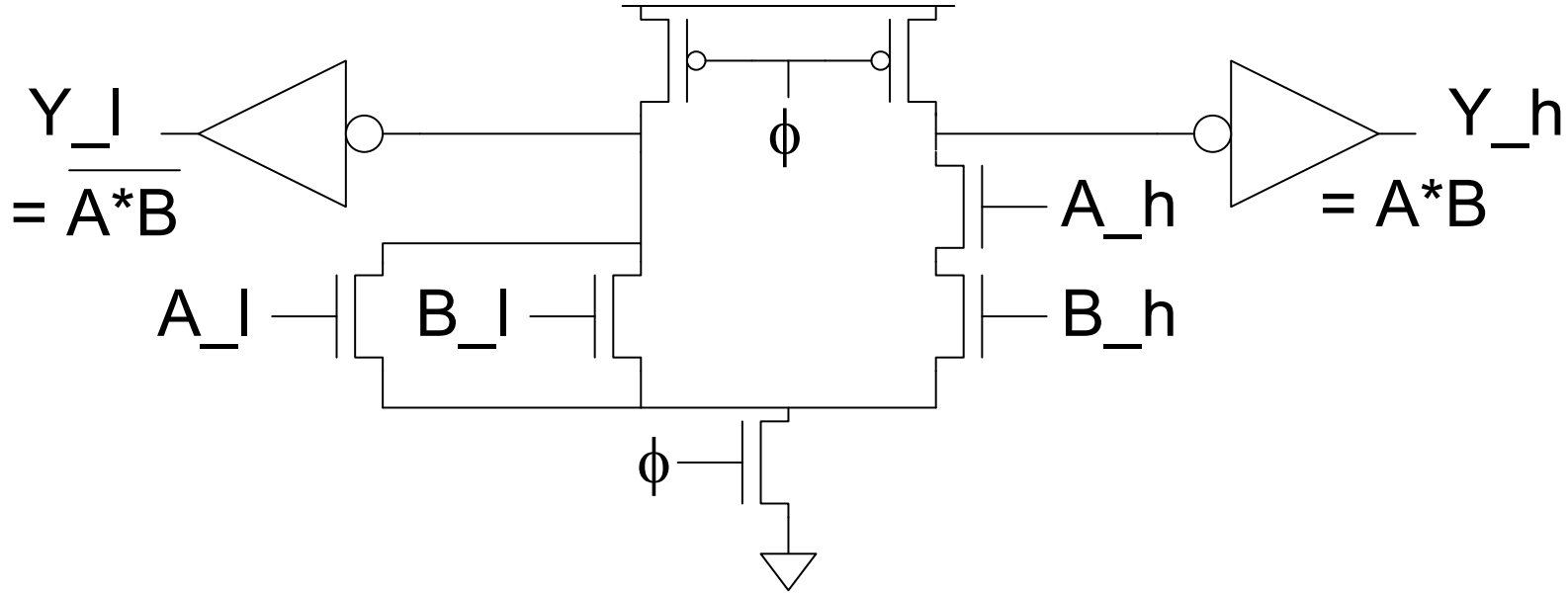
- Domino only performs non-inverting functions:
  - AND, OR but not NAND, NOR, or XOR
- Dual-rail domino solves this problem
  - Takes true and complementary inputs
  - Produces true and complementary outputs

sig_h	sig_l	Meaning
0	0	Precharged
0	1	‘0’
1	0	‘1’
1	1	invalid



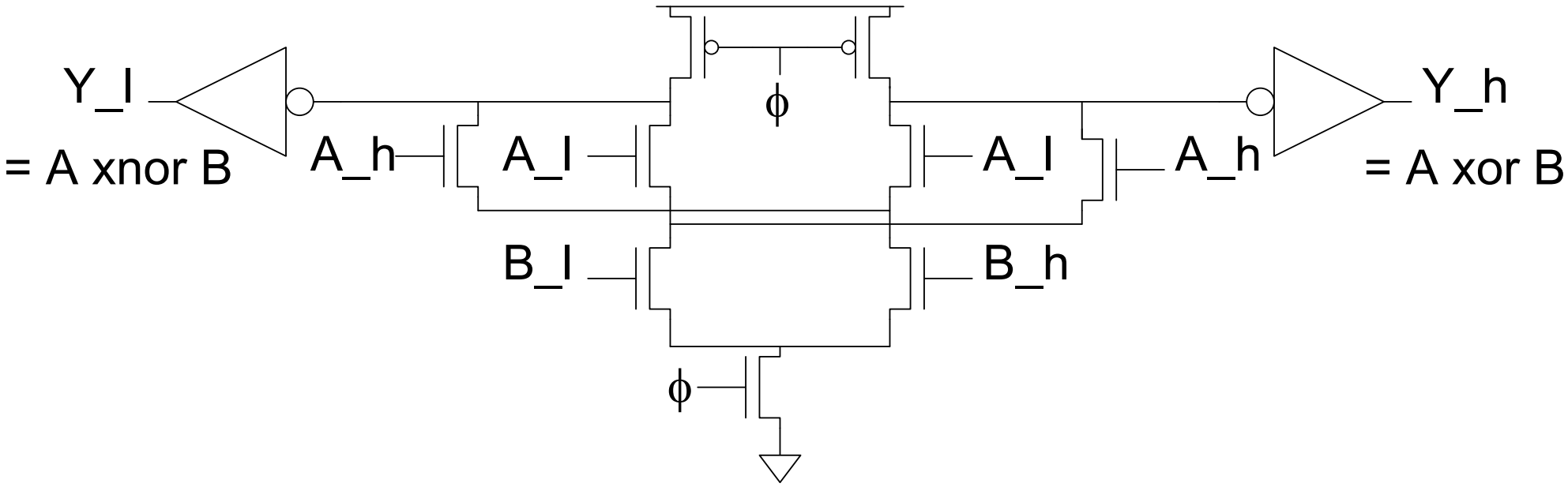
Example: AND/NAND

- Given A_h, A_l, B_h, B_l
- Compute Y_h = AxB,
  Y_l = ~(AxB)
- PDNs perform complementary conduction



Example: XOR/XNOR

- Sometimes possible to share transistors



Noise Sensitivity

- Dynamic gates are very sensitive to noise
  - Inputs: $V_{IH} \approx V_{tn}$
  - Outputs: floating output susceptible noise
- Noise sources
  - Capacitive crosstalk
  - Charge sharing
  - Power supply noise
  - Feedthrough noise
  - And more!

Summary: Domino

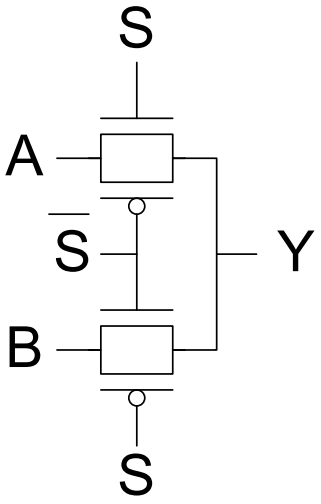
- Domino logic is attractive for high-speed circuits
  - 1.5 – 2x faster than static CMOS
  - But many challenges:
    - Monotonicity
    - Leakage
    - Charge sharing
    - Noise
- Widely used in high-performance microprocessors

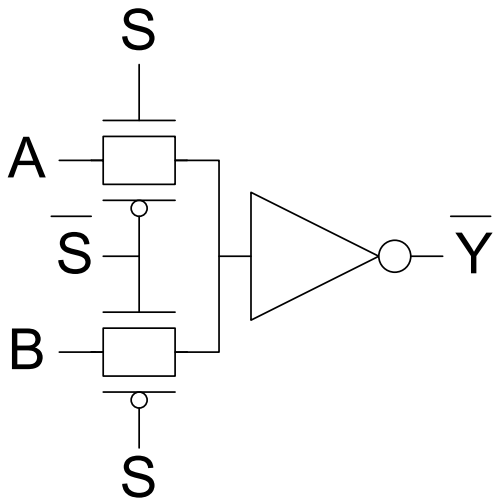
Outline

- Static CMOS
- Ratioed circuits
- Cascode voltage switch logic
- Dynamic circuits
- **Pass-transistor circuits**
- Circuit pitfalls

Pass Transistor Circuits

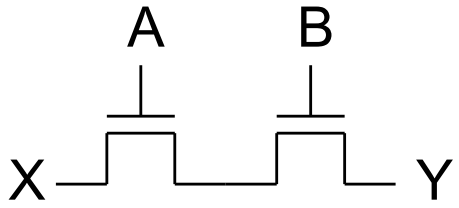
- Use pass transistors like switches to perform logic
- Inputs drive gates as well as diffusion terminals
- CMOS + Transmission Gates:
  - 2-input multiplexer
  - Gates should be restoring



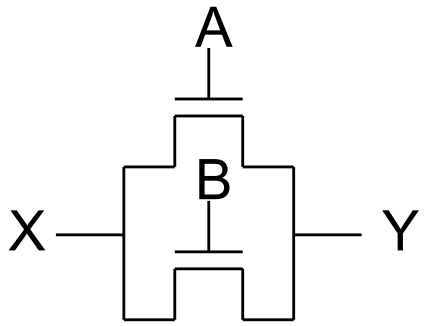


NMOS Transistors in Parallel/Series

- Primary inputs drive both gate and source/drain terminals
- NMOS switch turns ON when the gate voltage is high



X = Y if A and B

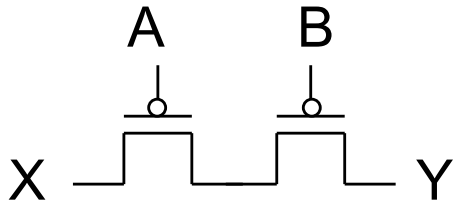


X = Y if A or B

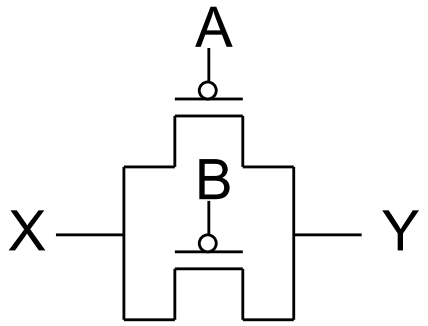
- Remember –
  NMOS transistors pass a strong 0 but a weak 1

PMOS Transistors in Parallel/Series

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



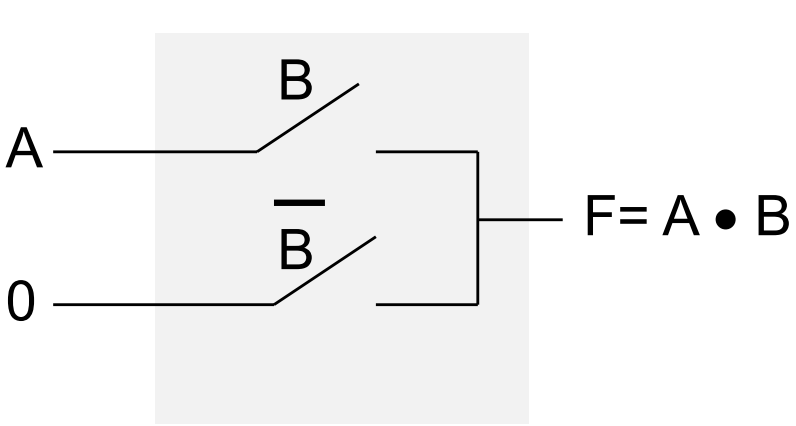
$X = Y$ if $\overline{A}$ and $\overline{B} = \overline{A + B}$

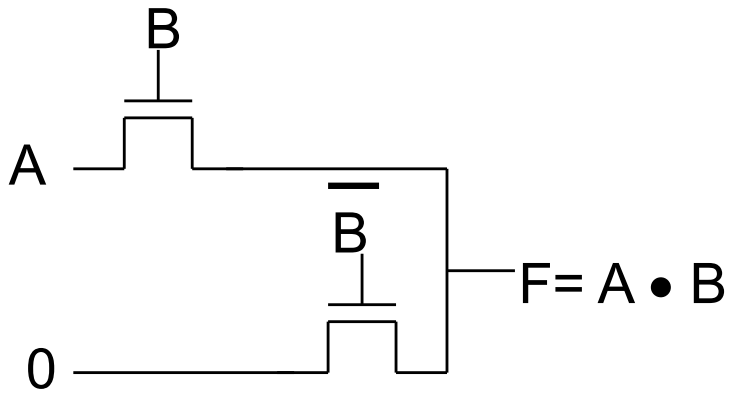


$X = Y$ if $\overline{A}$ or $\overline{B} = \overline{A \bullet B}$

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

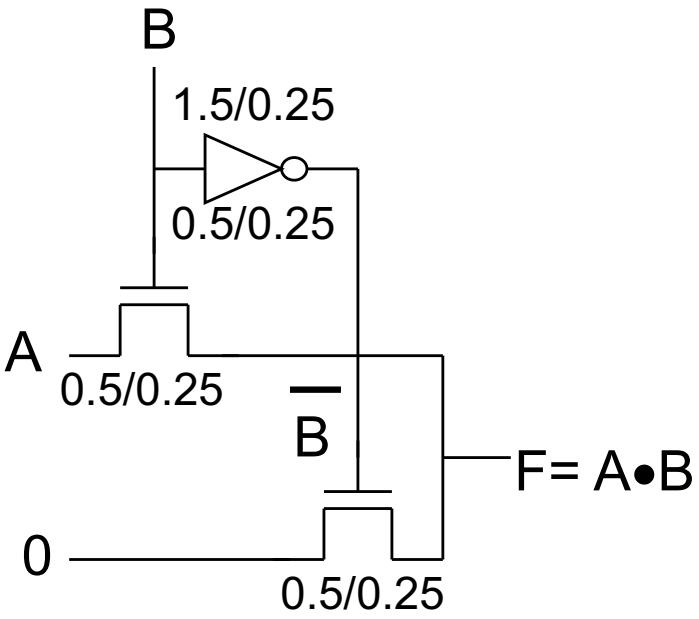
Pass Transistor (PT) Logic

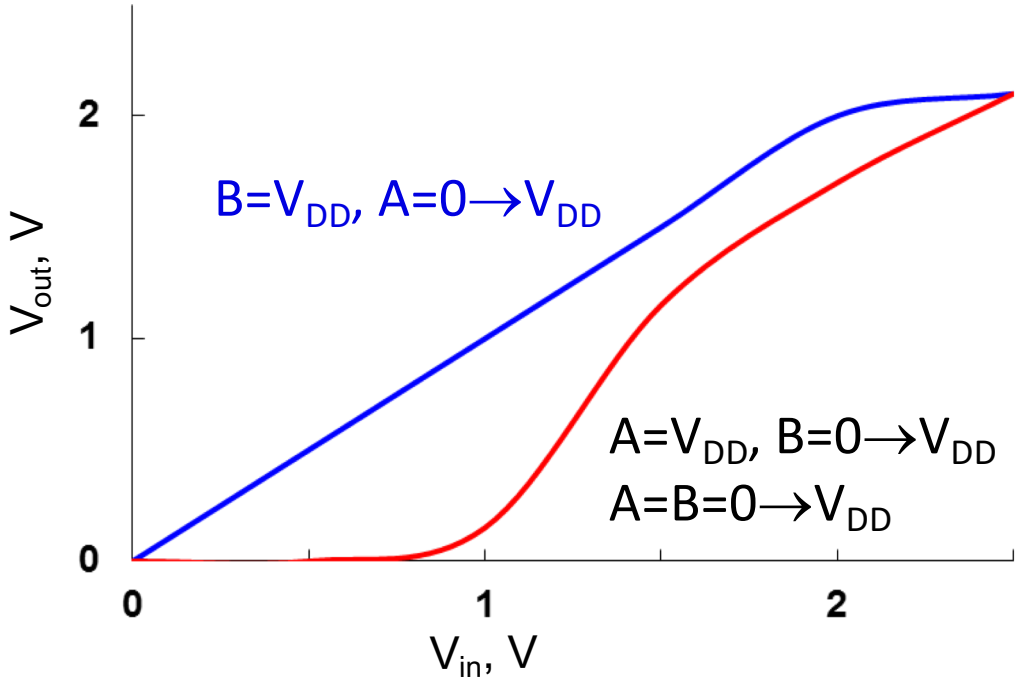




- Static – low-impedance paths exist to either supply rail under all circumstances
- N transistors instead of 2N
- No static power consumption
- Ratioless
- Bidirectional (versus undirectional)

VTC of PT AND Gate

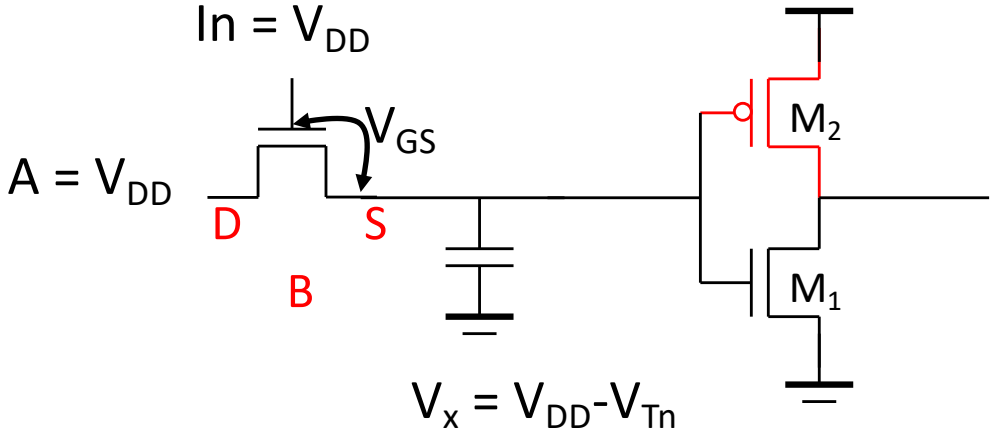




- Pure PT logic is not regenerative
  Signals gradually degrade after passing through a few stages (can fix by inserting static CMOS inverters)

NMOS Only PT Driving an Inverter

- V_x does not pull up to V_{DD}, but $V_{DD} - V_{Tn}$

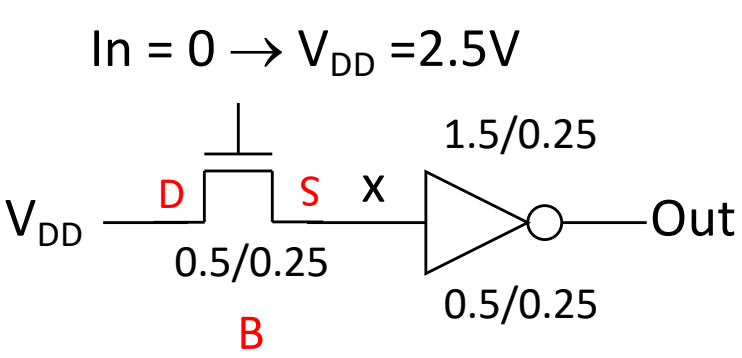


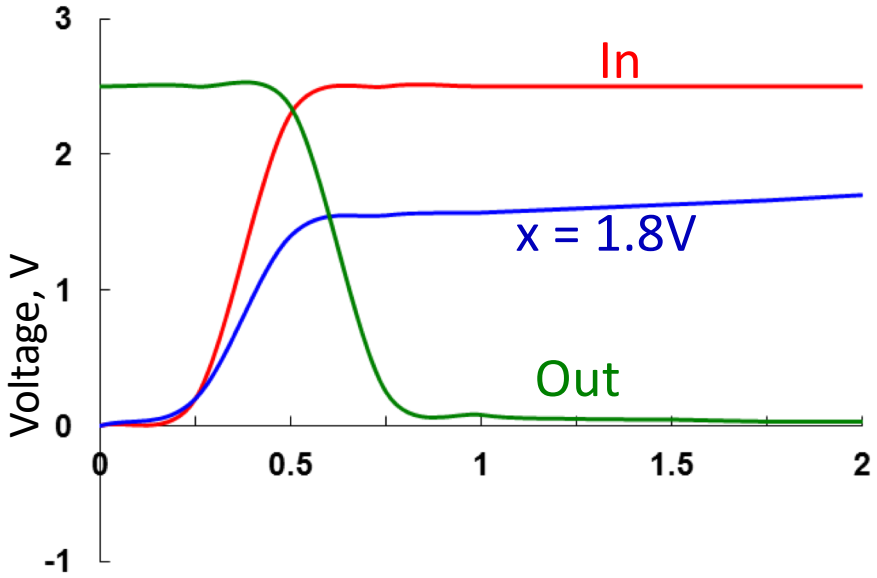
- V_{Tn} voltage drop causes static power consumption
  - M_2 may conduct weakly, forming a path from V_{DD} to GND
- V_{Tn} increases for pass transistors due to body effect

Voltage Swing of PT Driving and Inverter

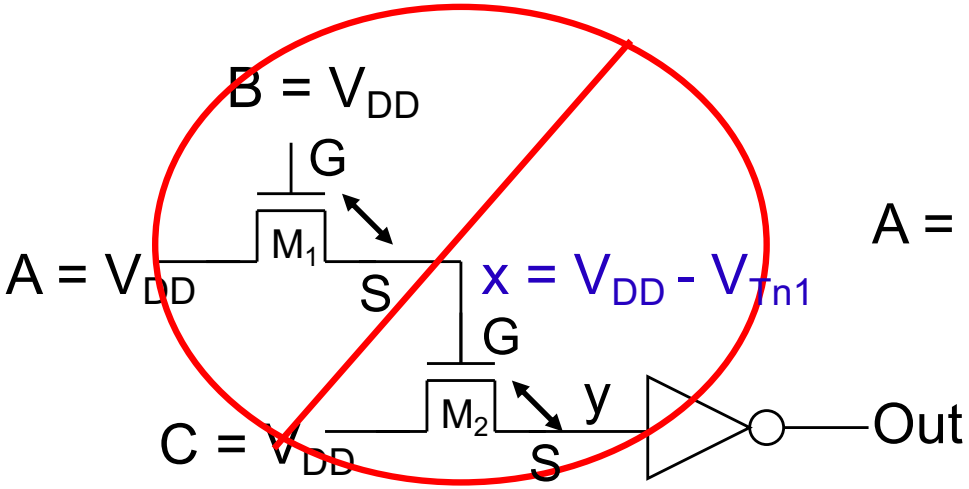
- Body effect – large V_{SB} at node x when pulling high (B is tied to GND and S charged up close to V_{DD})
- So the voltage drop is even worse

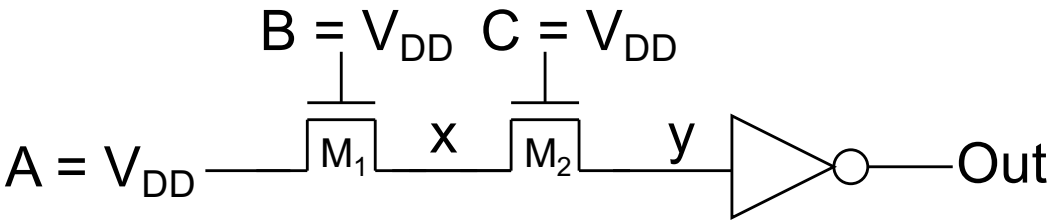
$$V_x = V_{DD} - (V_{Tn0} + \gamma(\sqrt{(|2\phi_f| + V_x)} - \sqrt{|2\phi_f|}))$$





Cascaded NMOS Only PTs





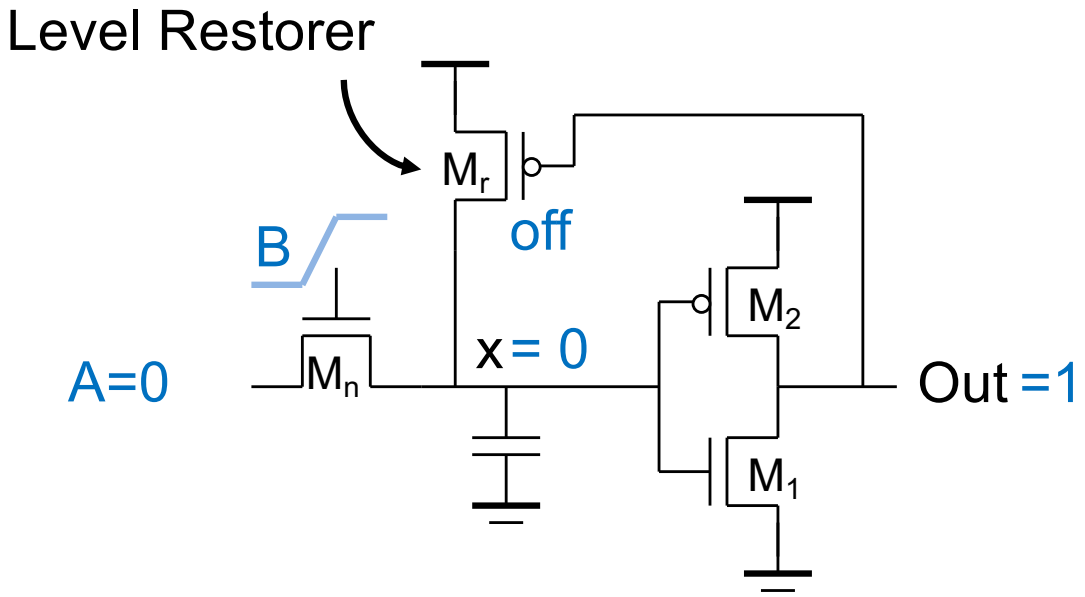
Swing on y = V_{DD} - V_{Tn1} - V_{Tn2}

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

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

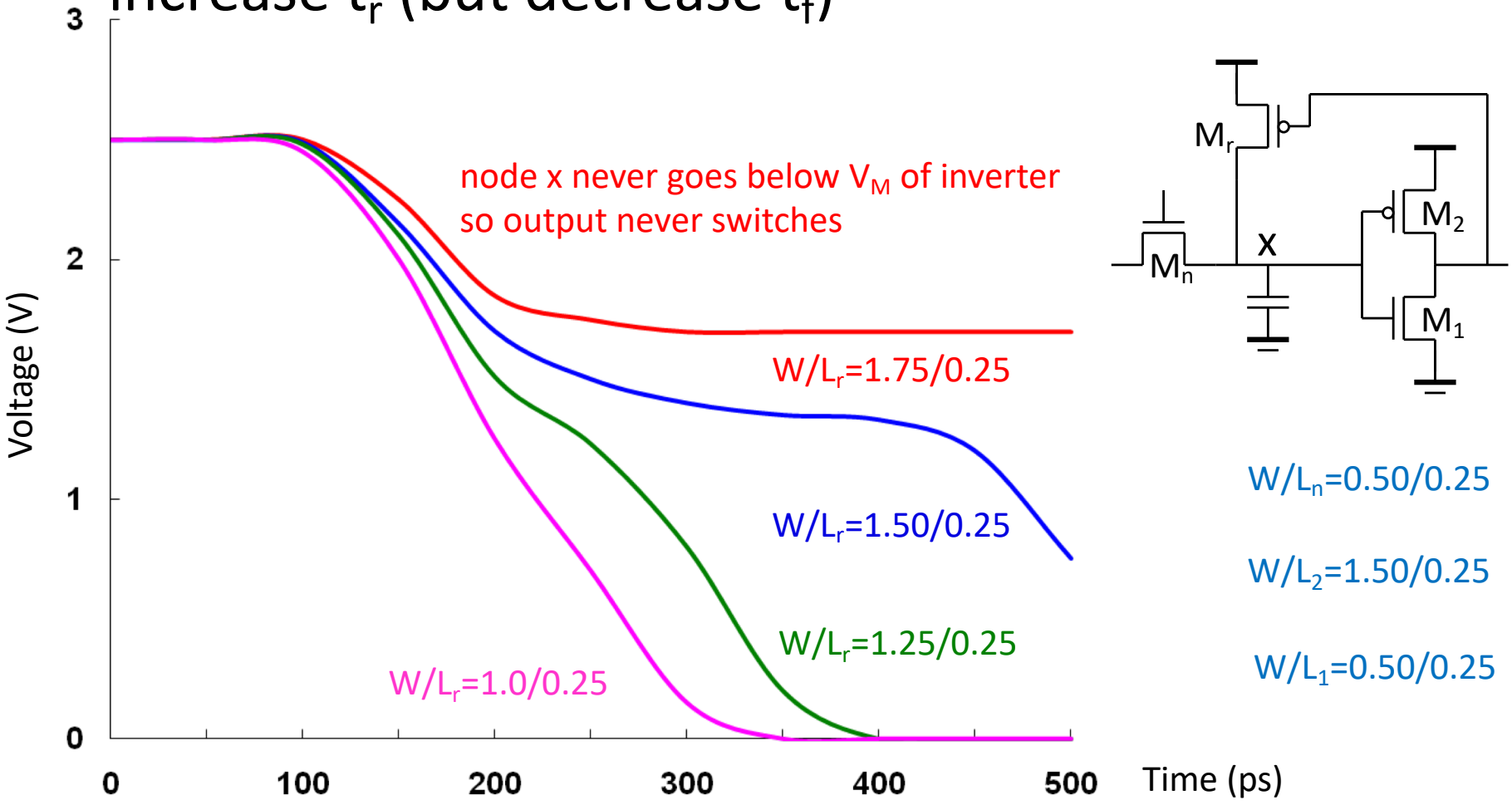
Solution #1: Level Restorer

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



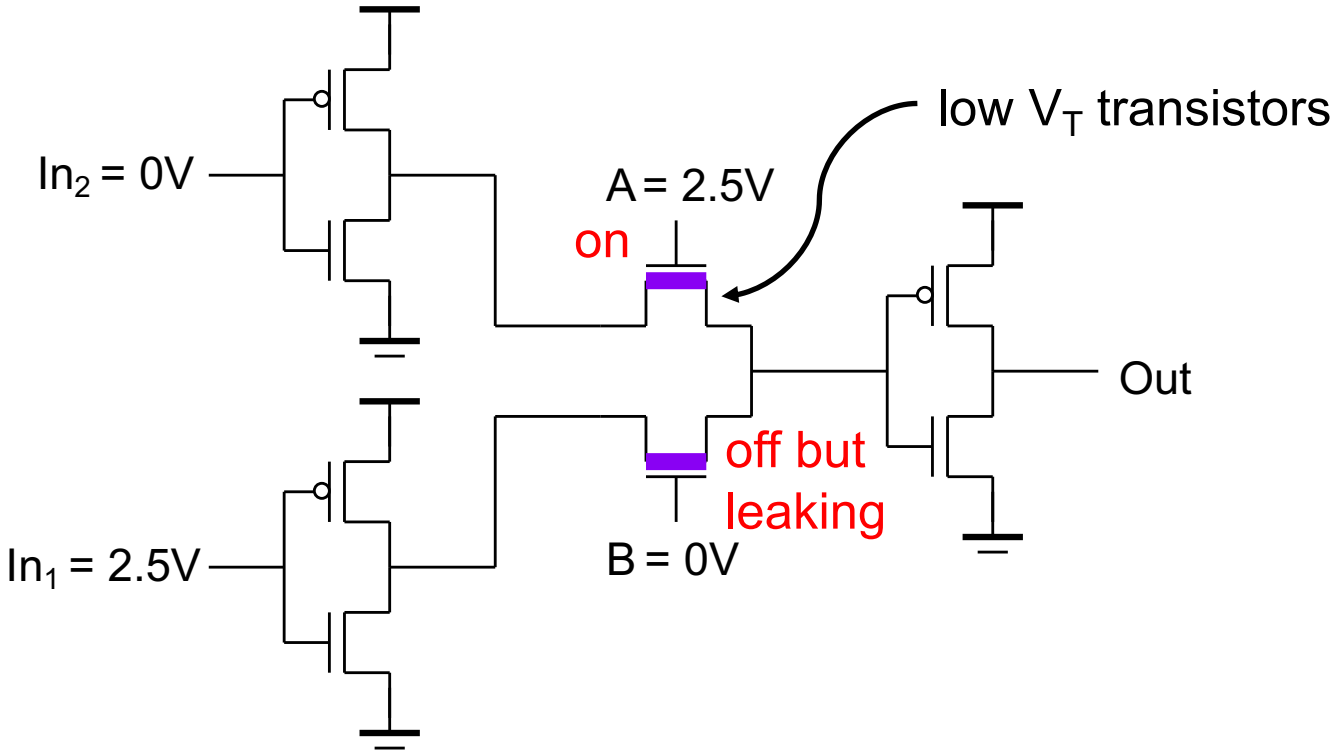
Restorer Circuit Transient Response

- Restorer has speed and power impacts: increase capacitance at x, slow down the gate, increase t_r (but decrease t_f)



Solution #2: Multiple V_{TH} Transistors

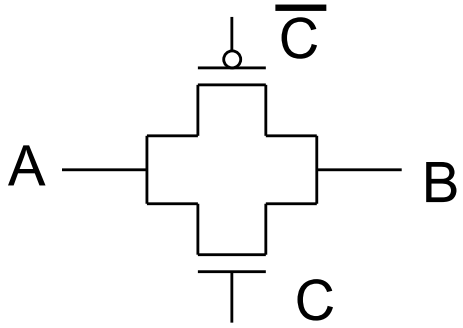
- Technology solution
  Use (near) zero V_T devices for NMOS PTs to eliminate *most* of the threshold drop (body effect still prevents full swing to V_{DD})

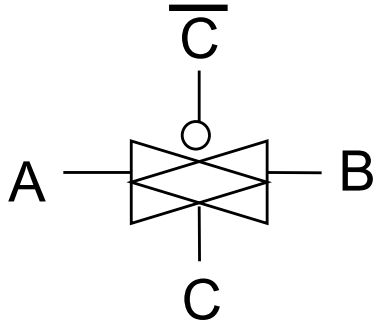


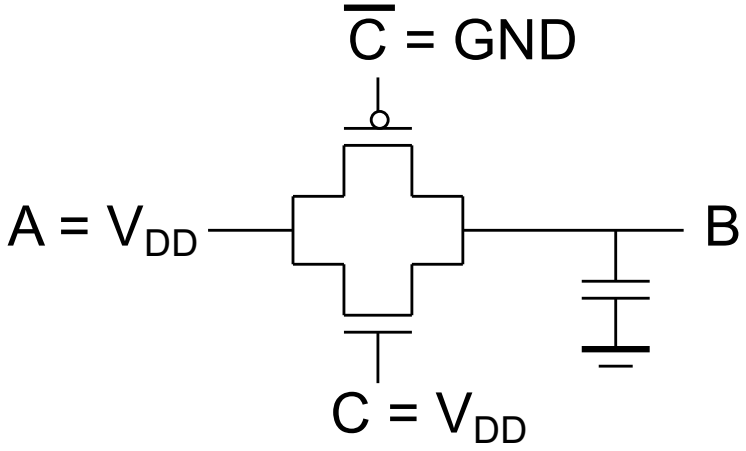
- Impacts static power consumption due to subthreshold currents flowing through the PTs (even if V_{GS} is below V_T)

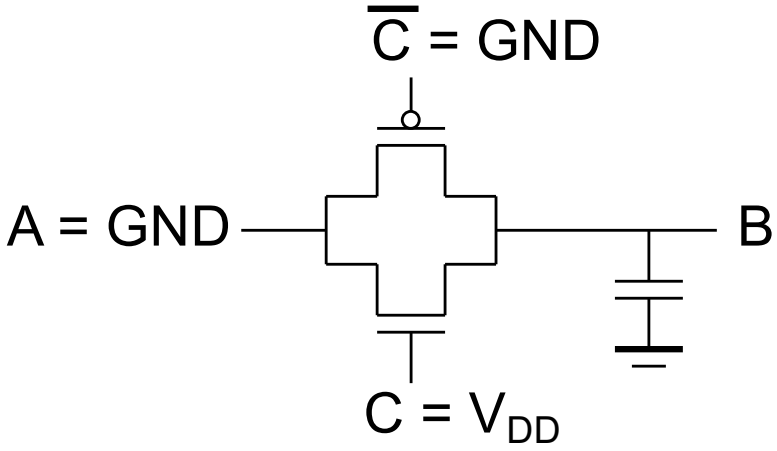
Solution #3: Transmission Gates (TGs)

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



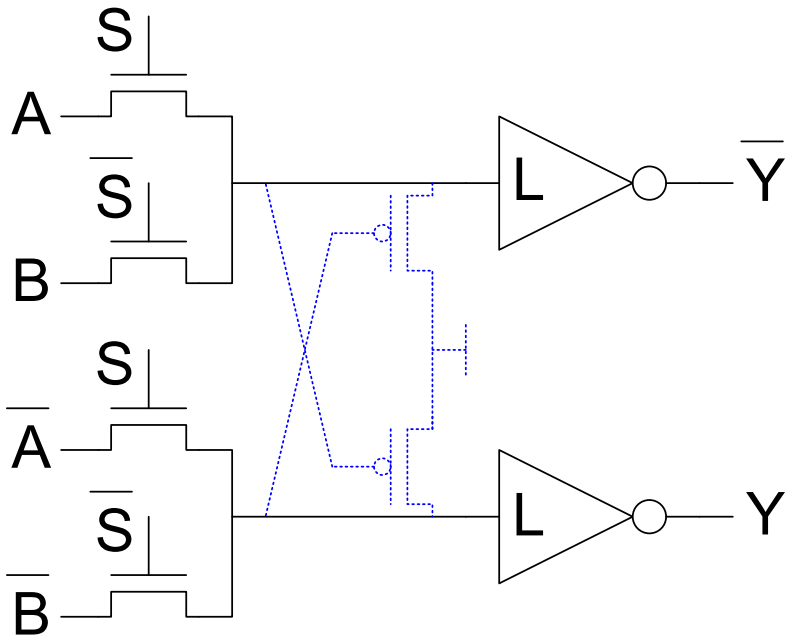






Solution #4: CPL

- **C**omplementary **P**ass-transistor **L**ogic
  - Dual-rail form of pass transistor logic
  - Avoid need for ratioed feedback
  - Optional cross-coupling for rail-to-rail swing



Outline

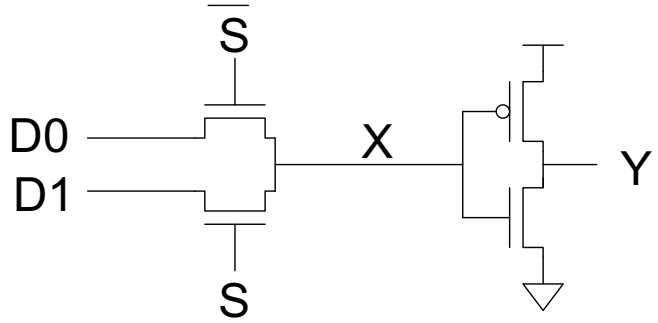
- Static CMOS
- Ratioed circuits
- Cascode voltage switch logic
- Dynamic circuits
- Pass-transistor circuits
- **Circuit pitfalls**

Introduction

- Circuit Pitfalls
  - Detective puzzle
  - Given circuit and symptom, diagnose causes and recommend solutions
  - All these pitfalls have caused failures in real chips
- Noise Budgets
- Reliability

Threshold Drop

- Circuit
  - 2:1 multiplexer

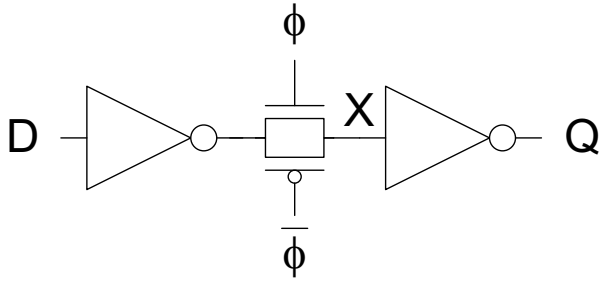


- Symptom
  - Mux works when selected input is 0 but not 1
  - Or fails at low V_{DD}
  - Or fails in SF corner

- Principle: Threshold drop
  - X never rises above V_{DD}-V_{th}
  - V_{thn} increased due to body effect
  - V_{th} drop worsens as V_{th} becomes a greater fraction of V_{DD}
- Solution: Use transmission gates, not pass transistors

Leakage

- Circuit
  - Latch

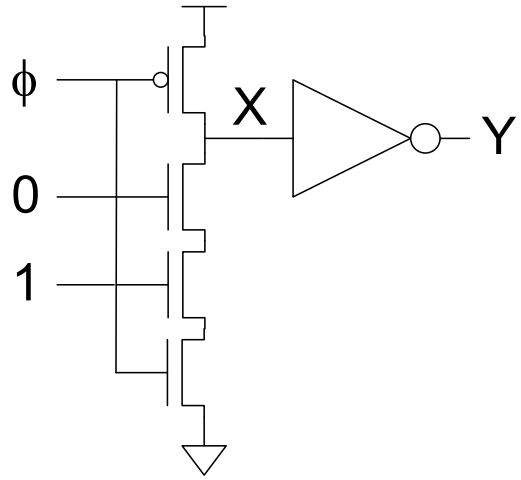


- Symptom
  - Load a 0 into Q
  - Set $\phi = 0$
  - Eventually Q spontaneously flips to 1

- Principle: Leakage
  - X is a dynamic node holding value as charge on the node
  - Eventually subthreshold leakage may disturb charge
- Solution: Stabilize node with feedback
  - Or periodically refresh node (requires fast clock, not practical with huge leakage)

Leakage

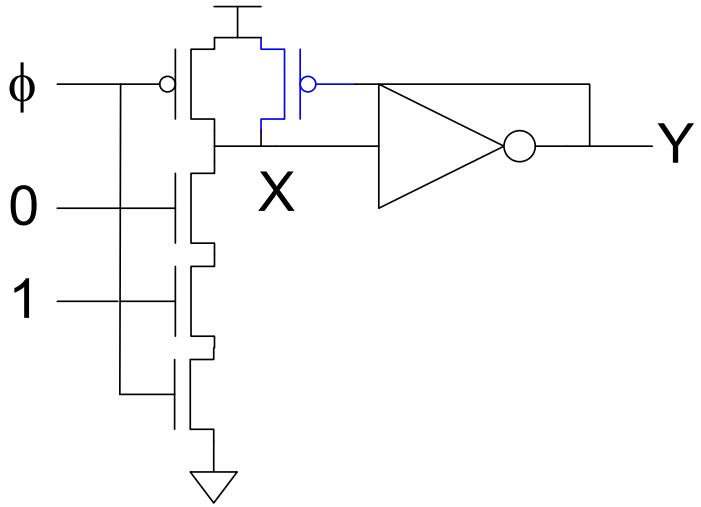
- Circuit
  - Domino AND



- Symptom
  - Precharge gate (Y=0)
  - Then evaluate
  - Eventually Y spontaneously flips to 1

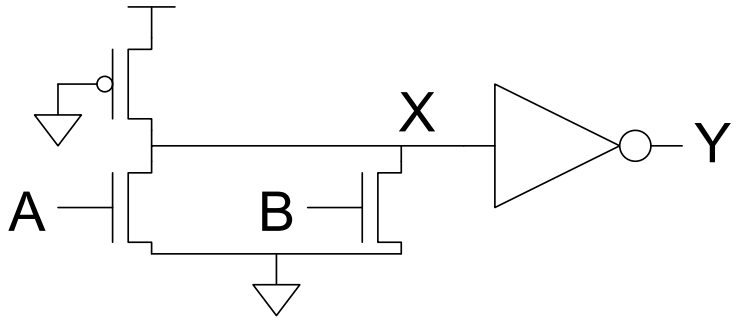
❑ Principle: Leakage
  - X is a dynamic node holding value as charge on the node
  - Eventually subthreshold leakage may disturb charge

❑ Solution: Keeper



Ratio Failure

- Circuit
  - Pseudo-NMOS OR

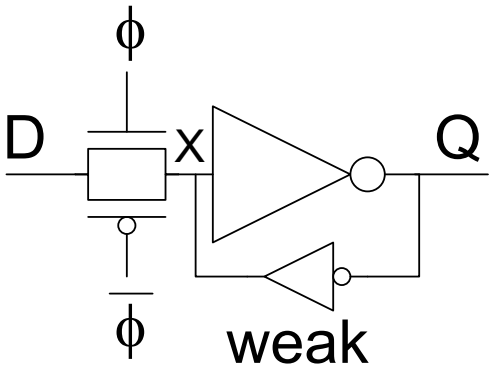


- Symptom
  - When only one input is true, Y = 0
  - Seems happening in SF corner

- Principle: Ratio Failure
  - nMOS and pMOS fight each other.
  - If the pMOS is too strong, nMOS cannot pull X low enough.
- Solution: Check that ratio is satisfied **in all corners**

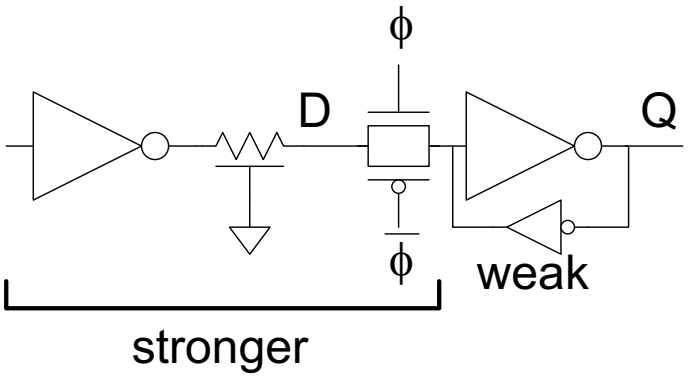
Ratio Failure

- Circuit
  - Latch



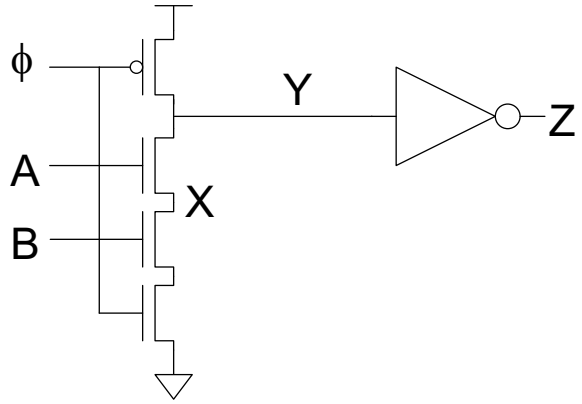
- Symptom
  - Q stuck at 1
  - Seems only happening for certain latches where input driver is weak (small gate or located far away)

- ❑ Principle: Ratio Failure (again)
  - Series resistance of D driver, wire resistance, and Tgate must be much less than weak feedback inverter.
- ❑ Solutions: Check relative strengths
  - Avoid unbuffered diffusion inputs where driver is unknown



Charge Sharing

- Circuit
  - Domino AND gate



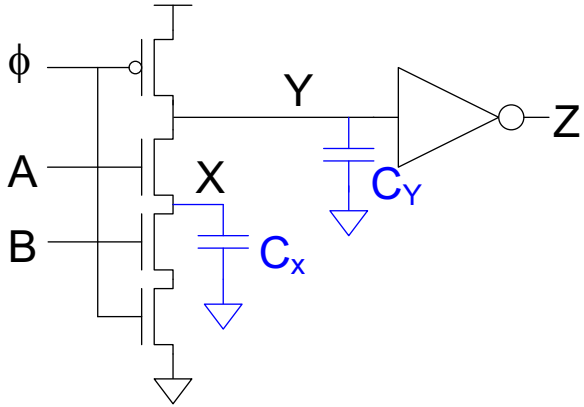
- Symptom
  - Precharge gate while A = B = 0, so Z = 0
  - Set ϕ = 1 and A rises
  - Z is observed to sometimes rise

❑ Principle: Charge Sharing
  - If X was low, it shares charge with Y

❑ Solutions: Limit charge sharing

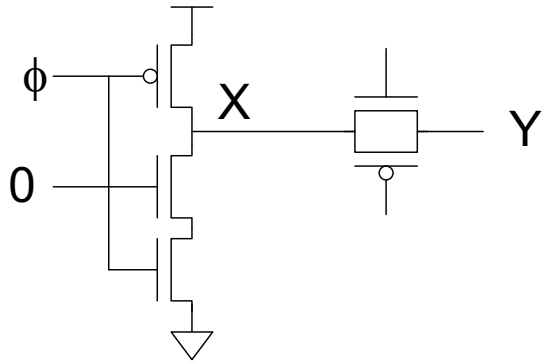
$$V_x = V_Y = \frac{C_Y}{C_x + C_Y} V_{DD}$$

  - Safe if $C_Y >> C_X$
  - Or precharge node X too



Charge Sharing

- Circuit
  - Dynamic gate
    + latch

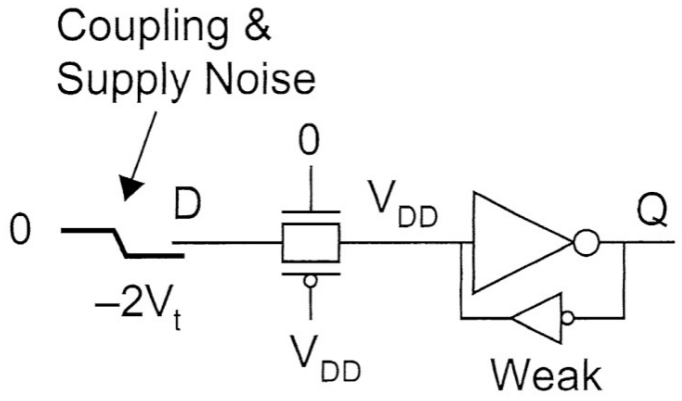


- Symptom
  - Precharge gate while transmission gate latch is opaque
  - Evaluate
  - When latch becomes transparent, X falls

- Principle: Charge Sharing
  - If Y was low, it shares charge with X
- Solution: Buffer dynamic nodes before driving transmission gate

Diffusion Input Noise

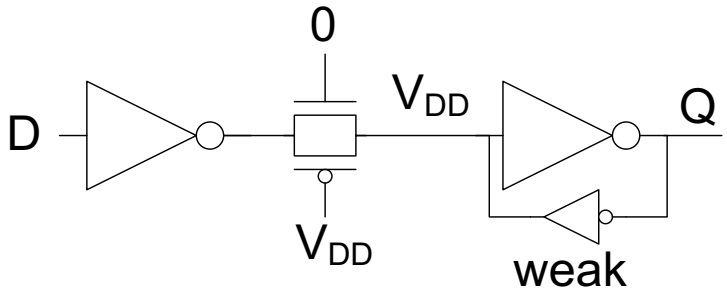
- Circuit
  - Latch



- Symptom
  - Q changes while latch is opaque
  - Especially if D comes from a far-away driver

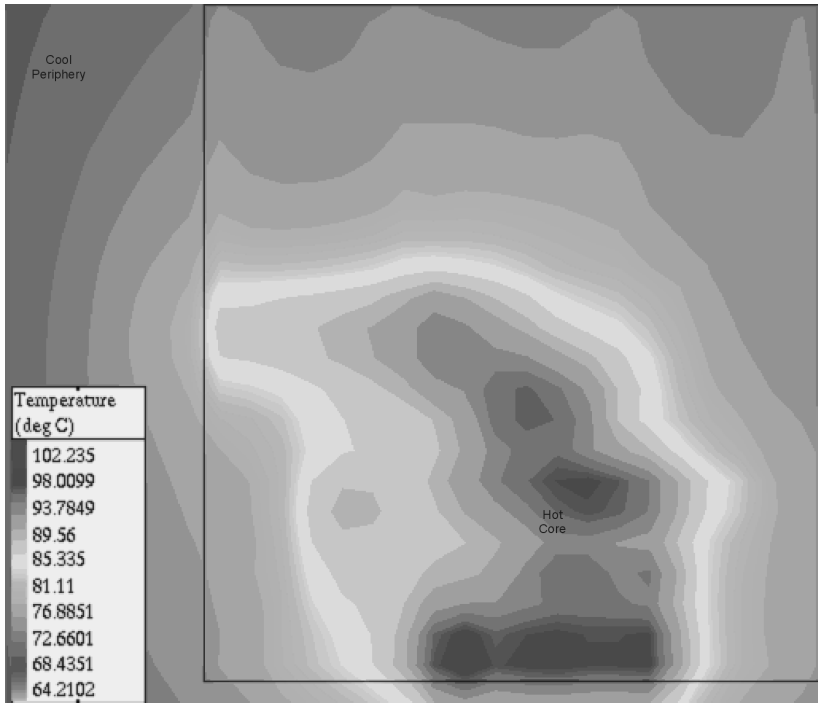
❑ Principle: Diffusion Input Noise Sensitivity
  - If $D < -V_{thn}$ → transmission gate turns on
  - Most likely because of power supply noise or coupling on D

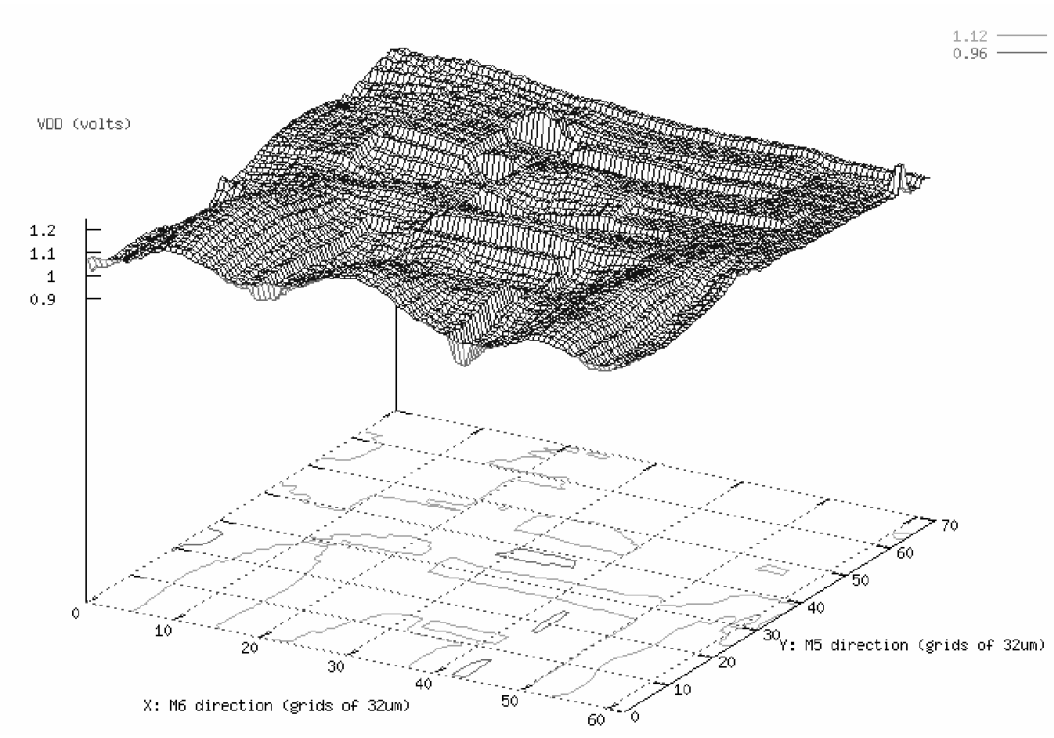
❑ Solution: Buffer D locally



Hot Spot

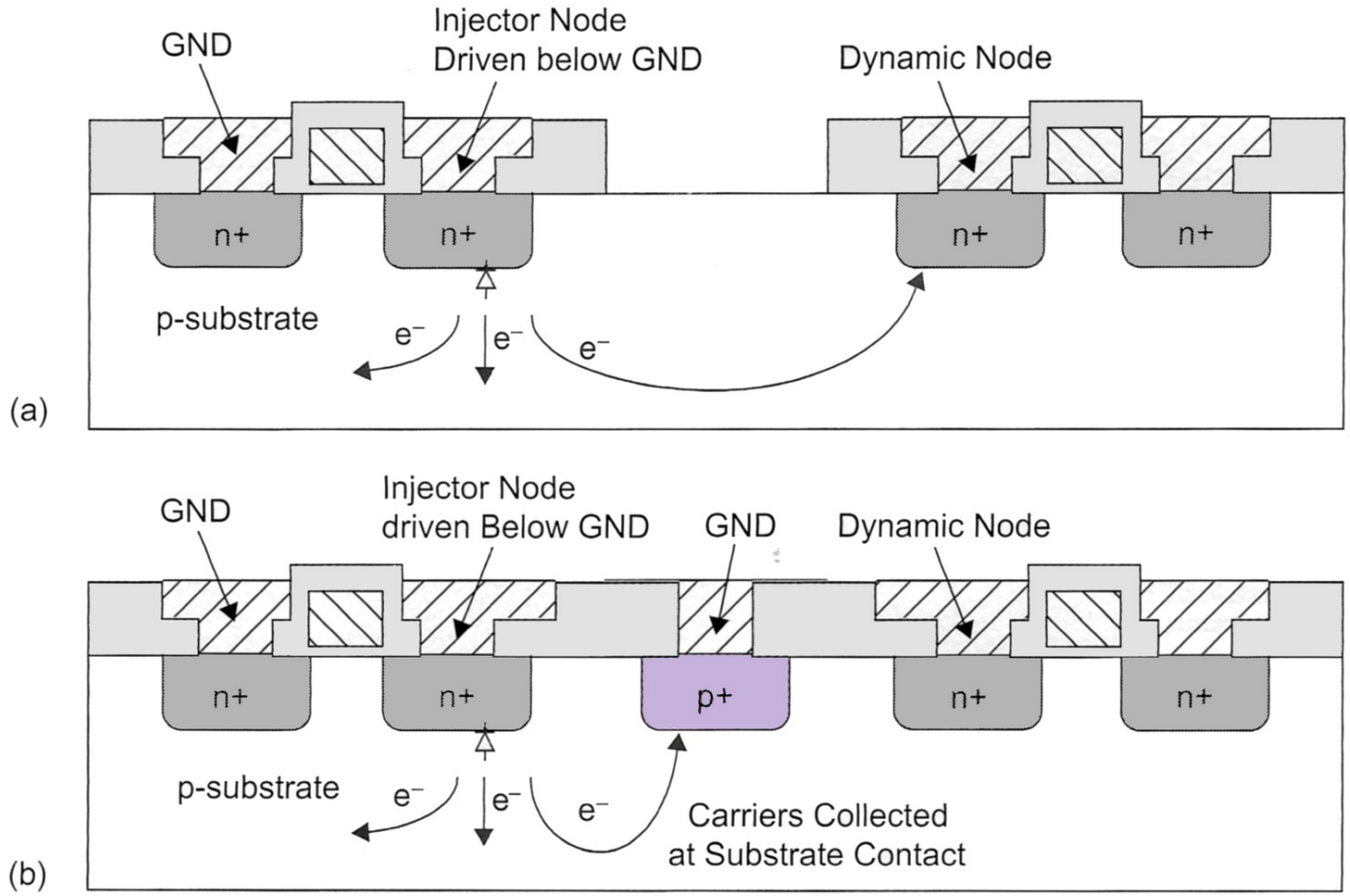
- Nonuniform power dissipation
  (even within overall power budget)





Minority Carrier Injection

- Minority injection caused by forward biased p-n junction
- Solution: Use guard ring to collect the excess minority carriers



Backgate Coupling

- Dynamic gates drive multiple-input static CMOS gates
- Solution: Drive input closer to the rail