Digital IC Design

Lec 4-3:

Combinational Circuits — Dynamic Logic

黃柏蒼 **Po-Tsang (Bug) Huang**
bughuang@nycu.edu.tw

International College of Semiconductor Technology
National Chiao Tung Yang Ming University

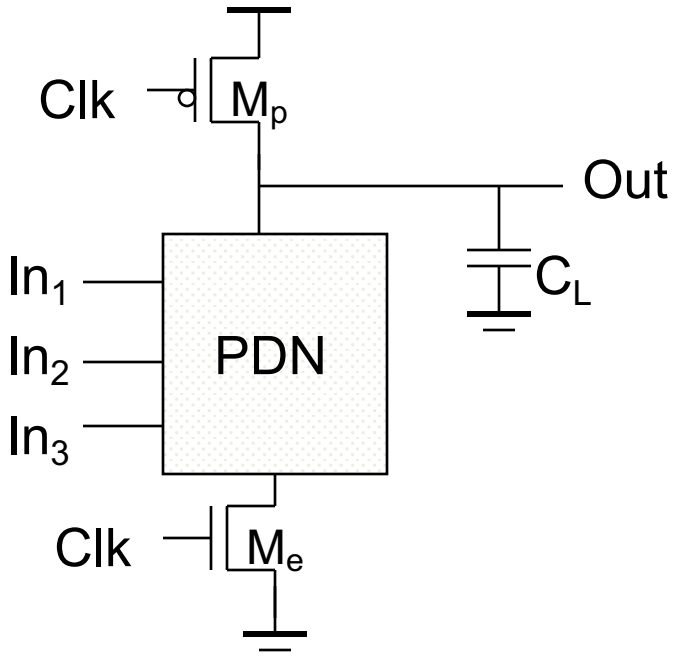
國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

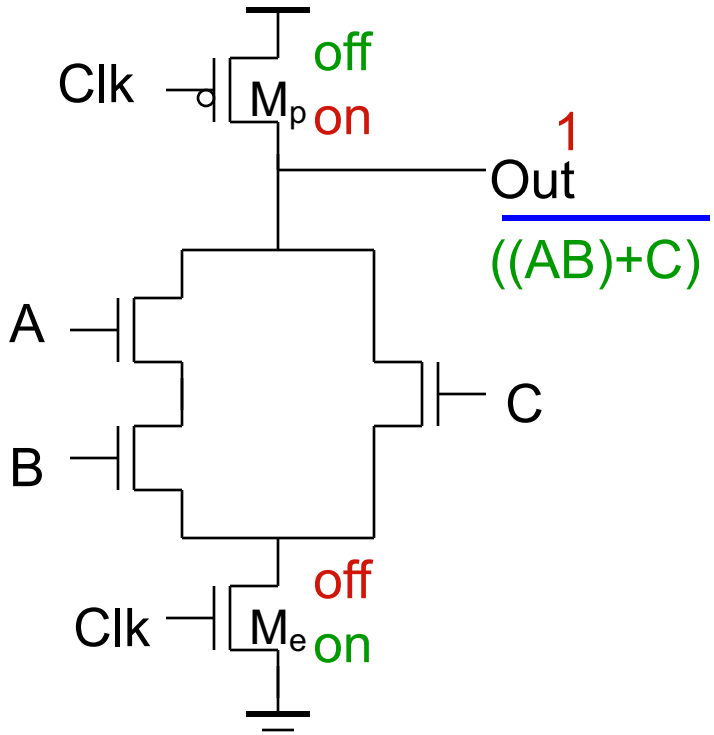
Dynamic and Static CMOS

- 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*** (*n* N-type + *n* P-type) devices

- Dynamic circuits rely on the ***temporary storage*** of signal values on the ***capacitance*** of high impedance nodes.
  - requires on *n* + 2 (*n*+1 N-type + 1 P-type) transistors
  - takes a sequence of precharge and conditional evaluation phases to realize logic functions

Dynamic Gate





Two phase operation

Precharge (Clk = 0)

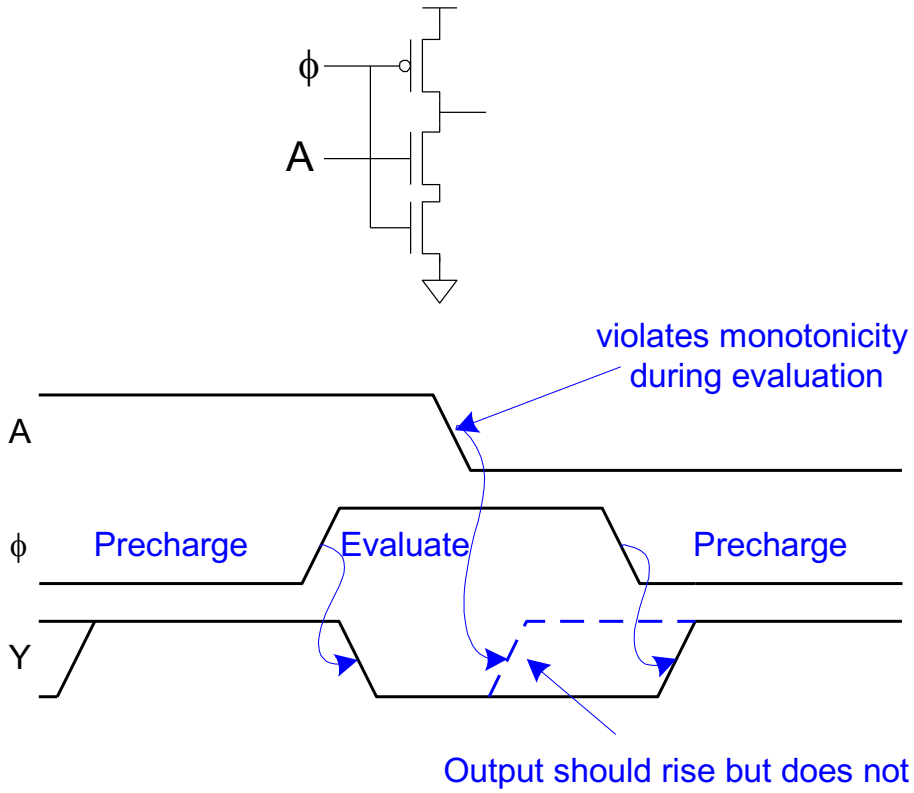
Evaluate (Clk = 1)

Problems of Dynamic Logic

- Cascading dynamic logic
  - Domino logic
    - Non-inverting logic
- Signal integrity problems
  - Charge leakage
    - Add keeper
  - Charge sharing
    - Precharge internal node
  - Capacitive coupling
  - Clock feedthrough

Cascade Dynamic Gates: Monotonicity

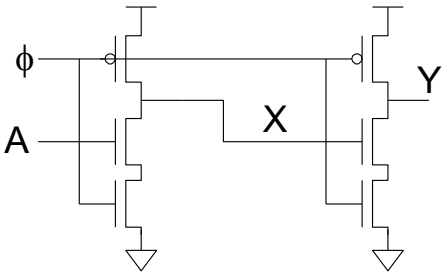
- Dynamic gates require *monotonically rising* inputs during evaluation
  - 0 -> 0
  - 0 -> 1
  - 1 -> 1
  - But not 1 -> 0

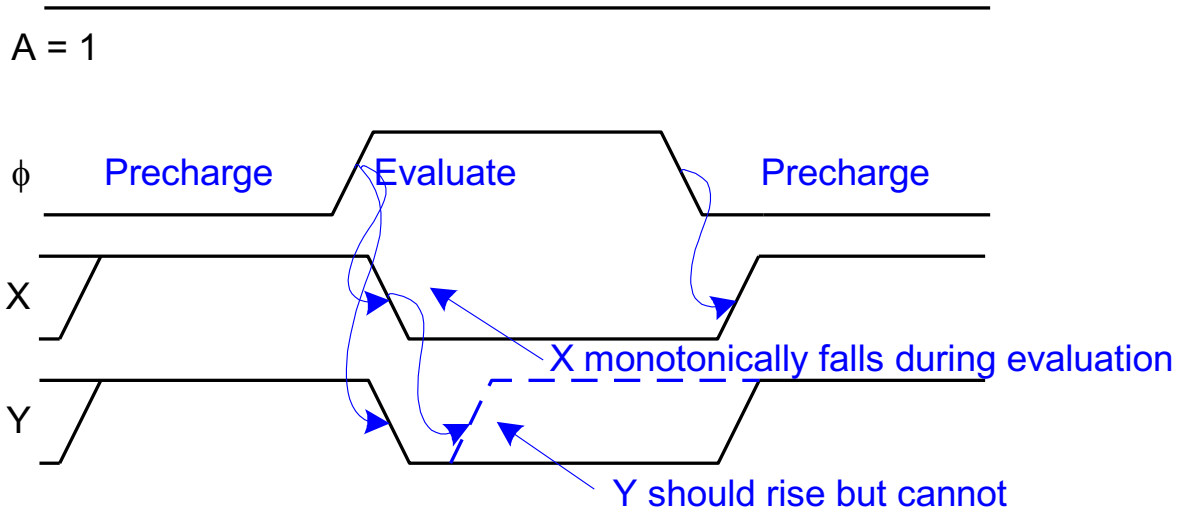


Only 0 → 1 transitions allowed at inputs!

Monotonicity Problem

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



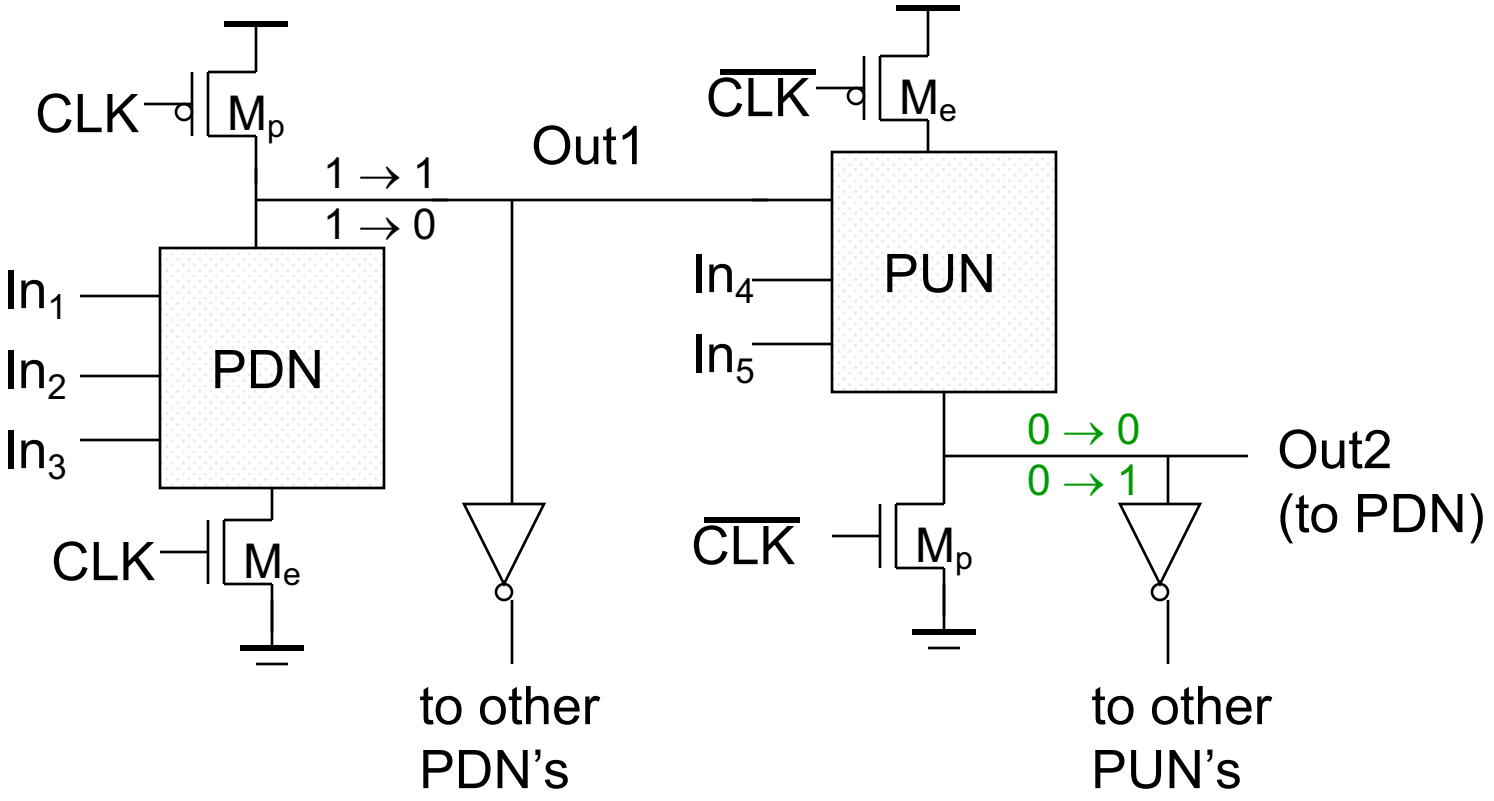


Conditions on Output

- 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.
  - This could be a problem when you cascade the dynamic logic
    - Extra transition will destroy the logic

- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on C_L

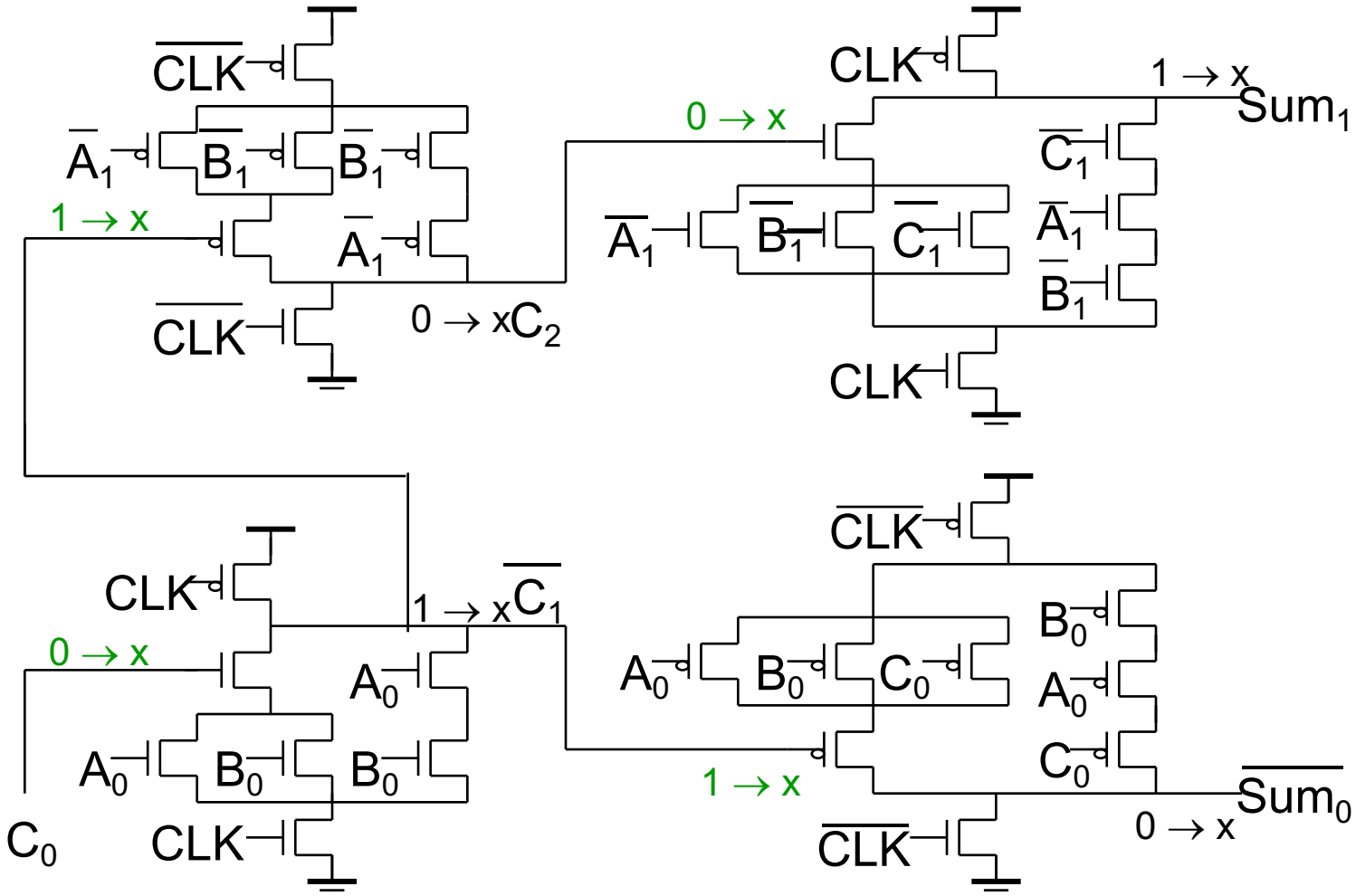
np-CMOS (Zipper)



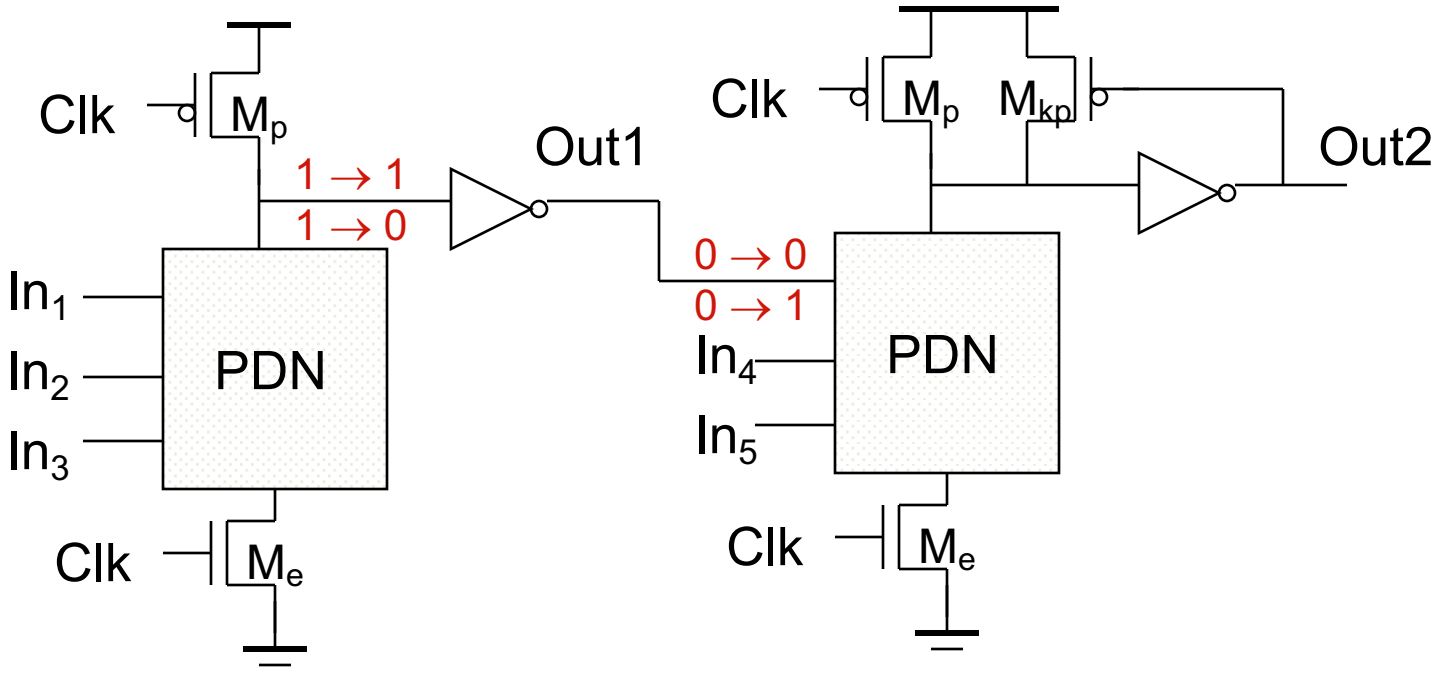
Only $0 \rightarrow 1$ transitions allowed at inputs of PDN

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

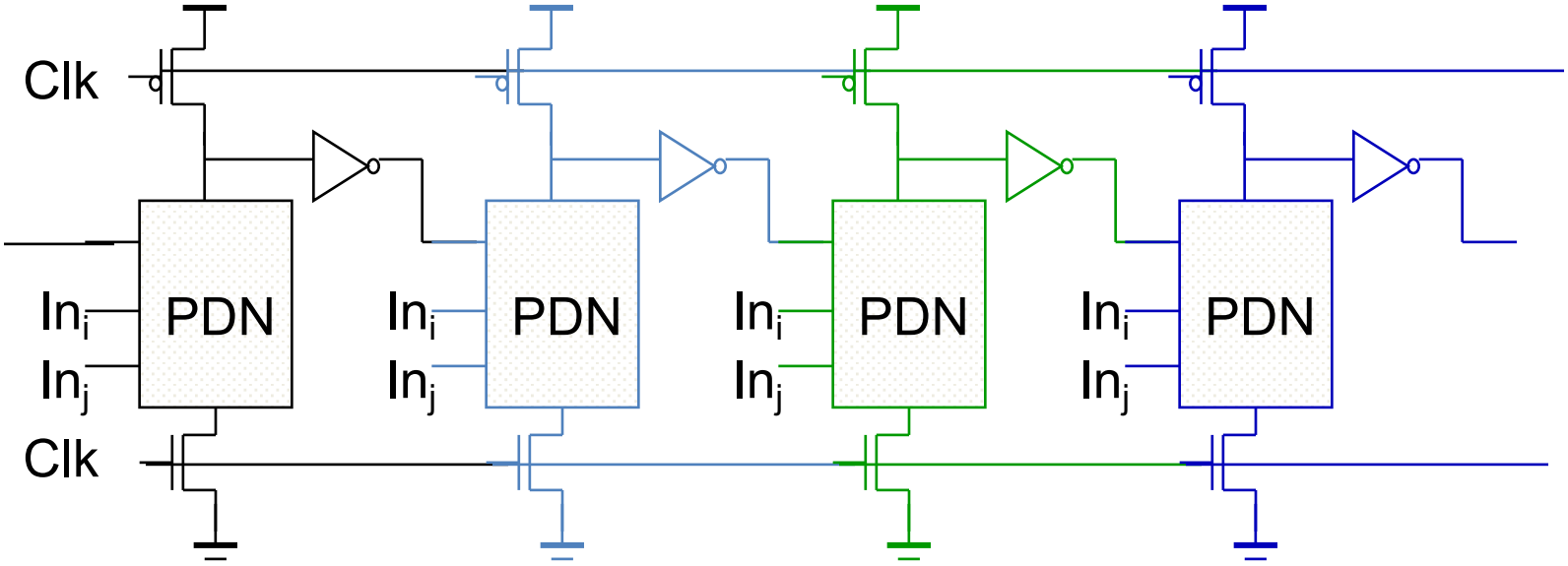
np-CMOS Adder Circuit



Solution: Domino Logic



Why Domino?



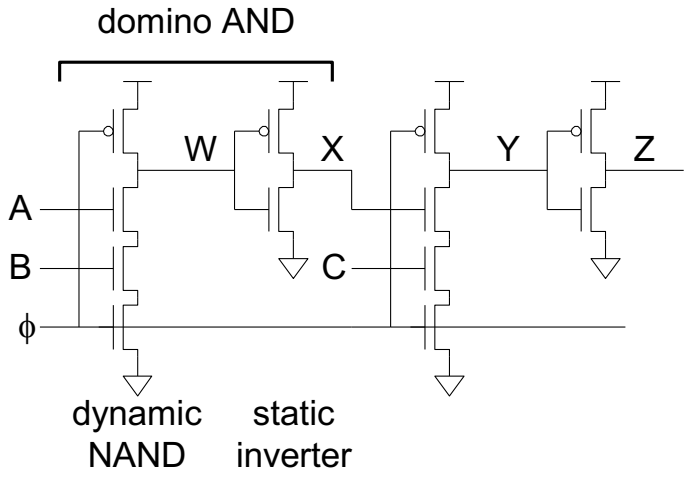
Like falling dominos!

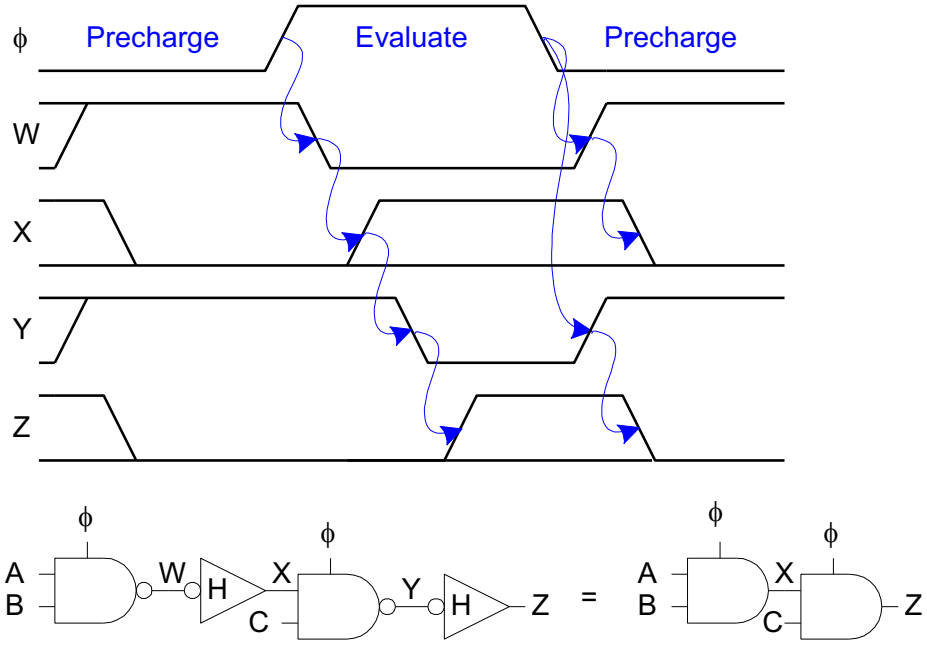
Properties of Domino Logic

- Only **non-inverting** logic can be implemented for dymaic circuits, 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)

Domino Gates

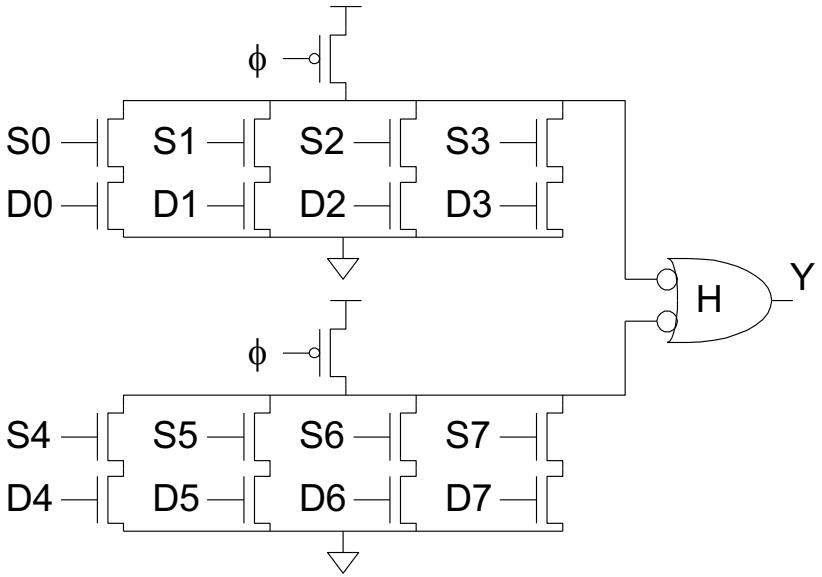
- Follow dynamic stage with inverting static gate
  - Dynamic / static pair is called domino gate
  - Produces monotonic outputs





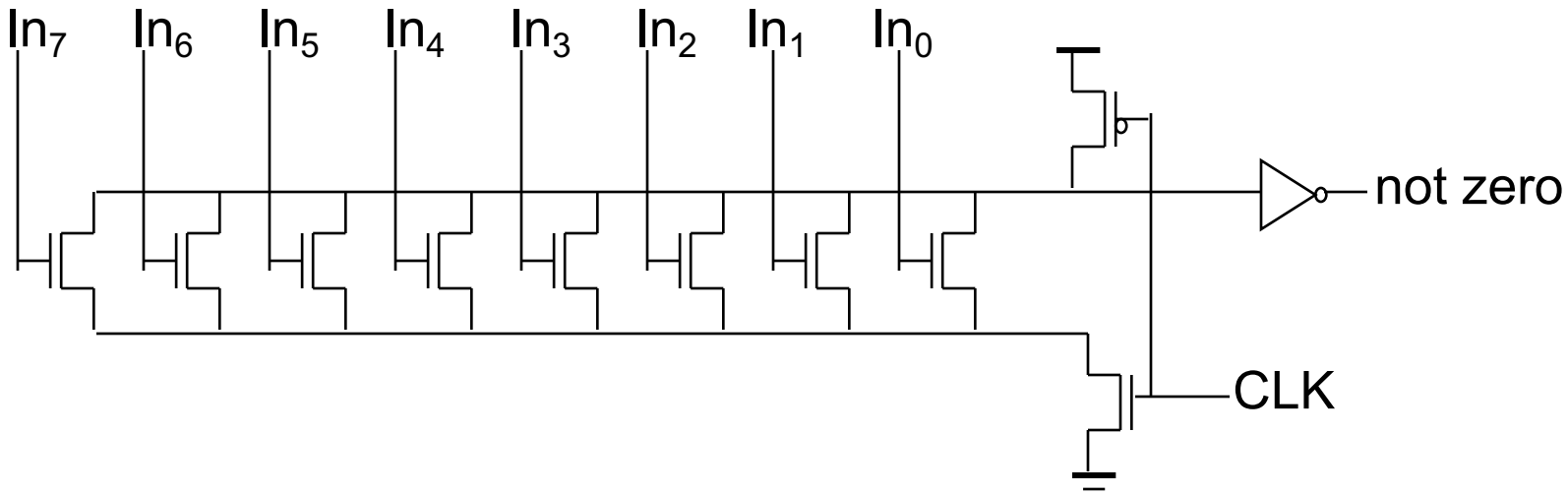
Domino Optimizations

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

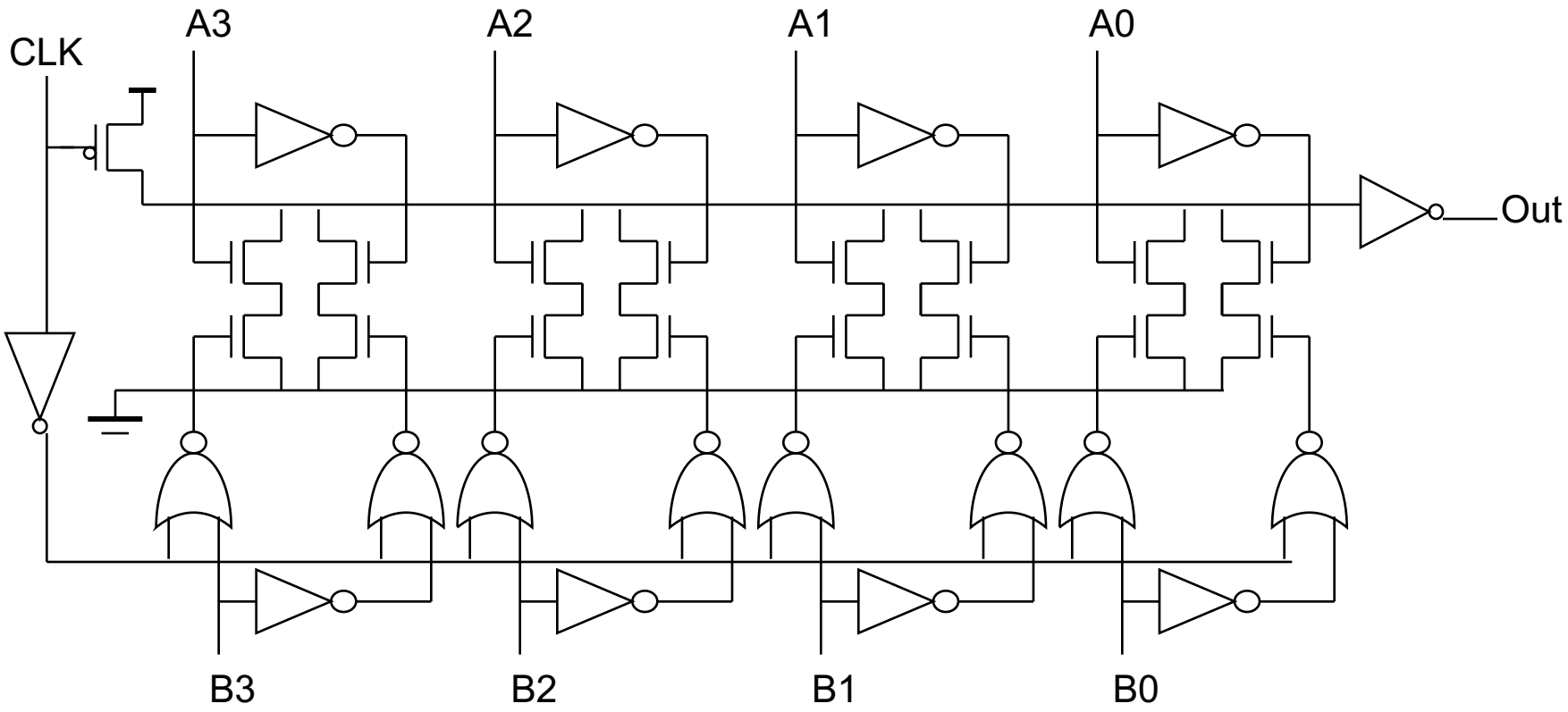


Domino Zero Detector

- Large Fan-in circuits



Domino Comparator



Properties of Domino Logic

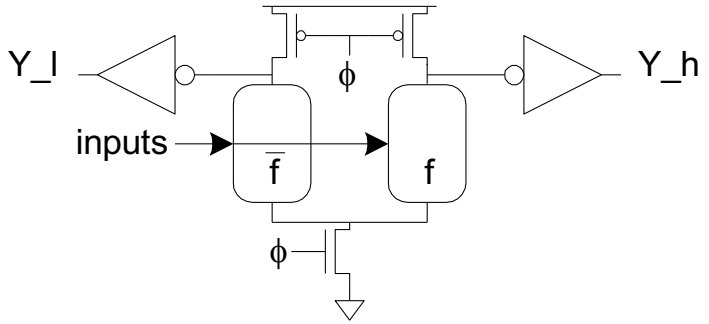
- Only non-inverting logic can be implemented, fixes include
  - can reorganize the logic using Boolean transformations
  - use differential logic (dual rail)
  - use np-CMOS (zipper)

- Very high speed
  - $t_{pHL} = 0$
  - static inverter can be optimized to match fan-out (separation of fan-in and fan-out capacitances)

Dual-Rail Domino

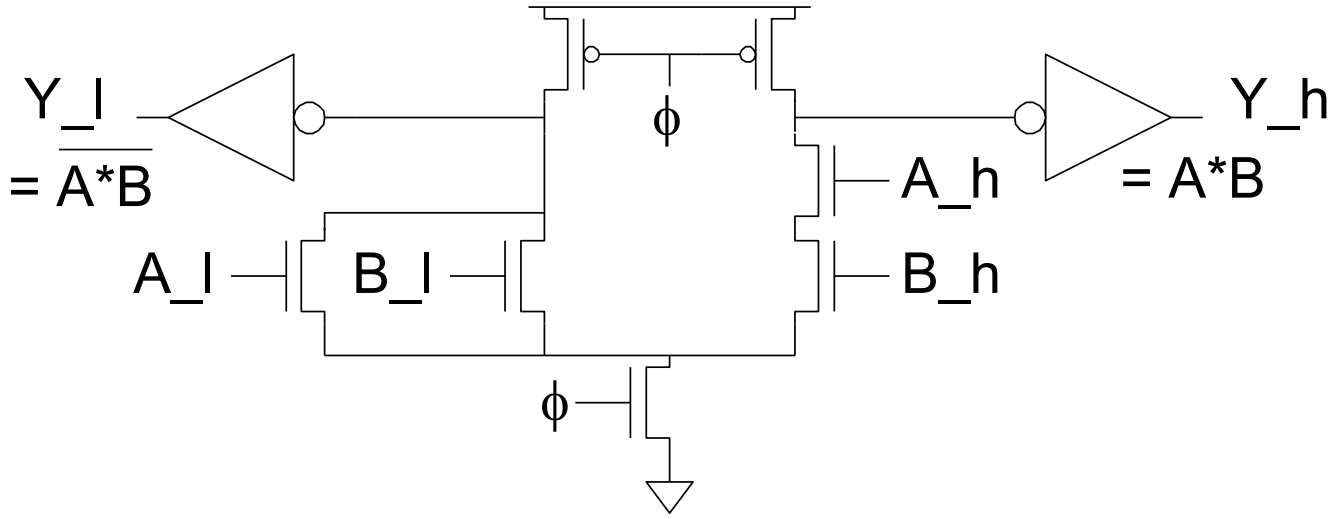
- Domino only performs noninverting functions:
  - AND, OR but not NAND, NOR, or XOR
- Dual-rail domino solves this problem
  - Takes true and complementary inputs
  - Produces true and complementary outputs

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



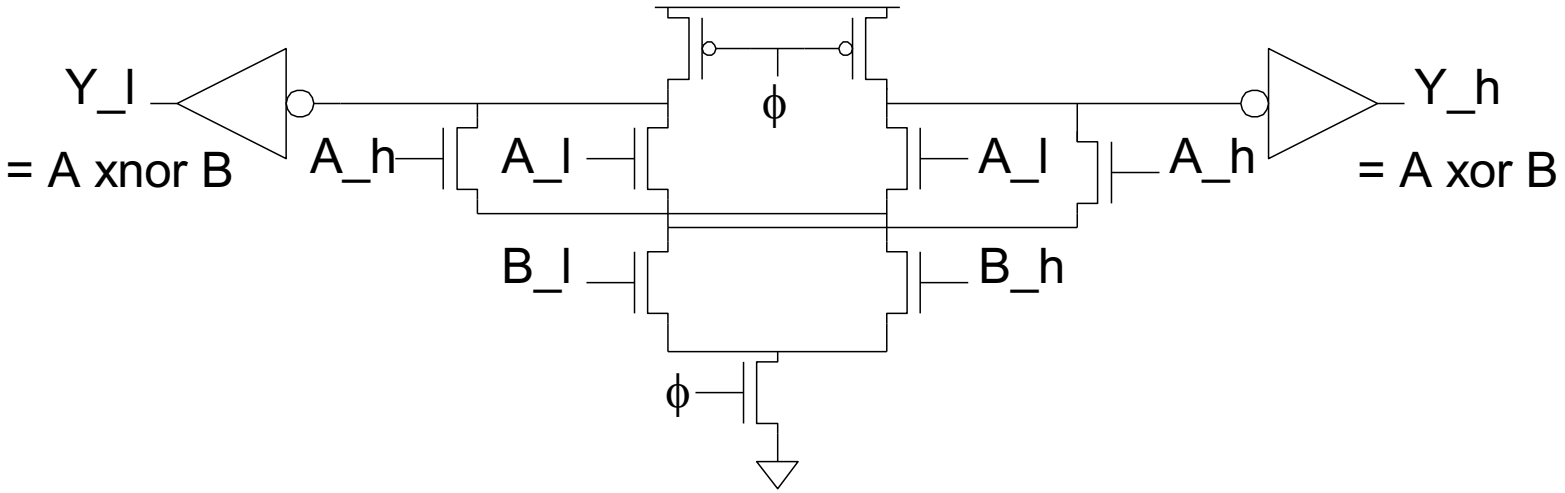
Example: AND/NAND

- Given A_h, A_l, B_h, B_l
- Compute Y_h = AB, Y_l = AB
- Pulldown networks are conduction complements

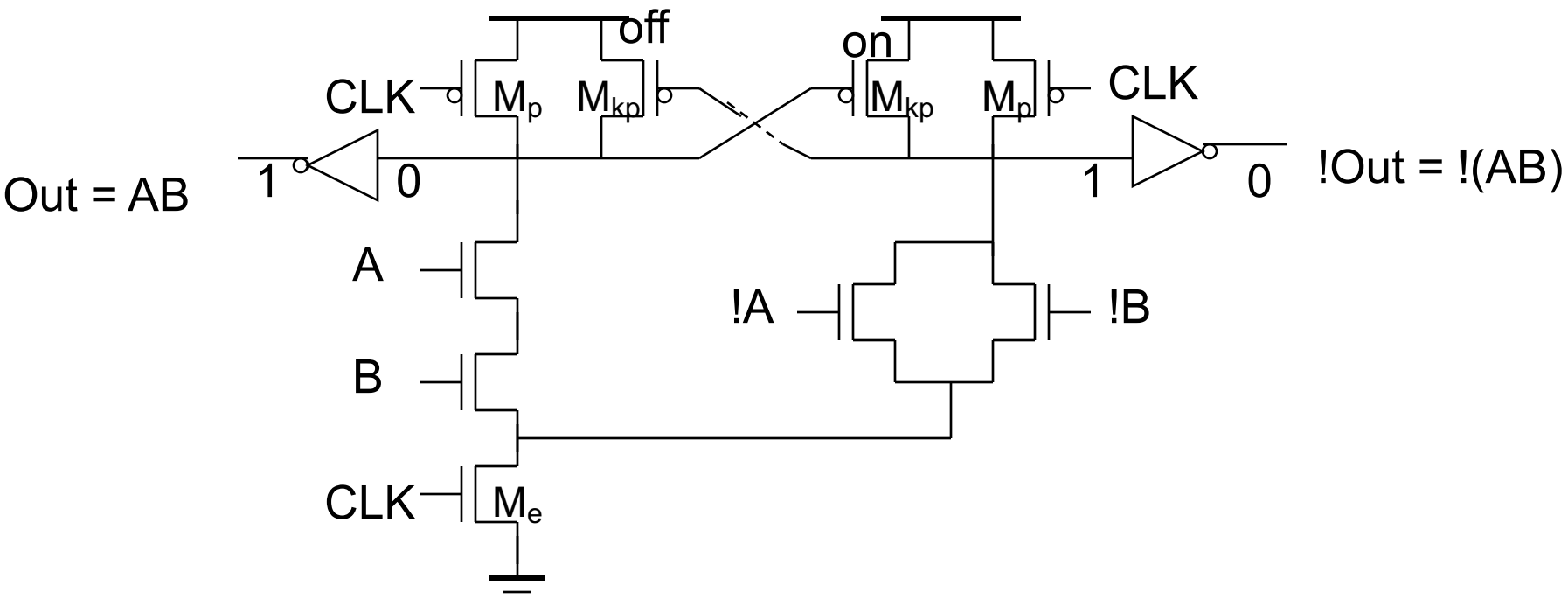


Example: XOR/XNOR

- Sometimes possible to share transistors



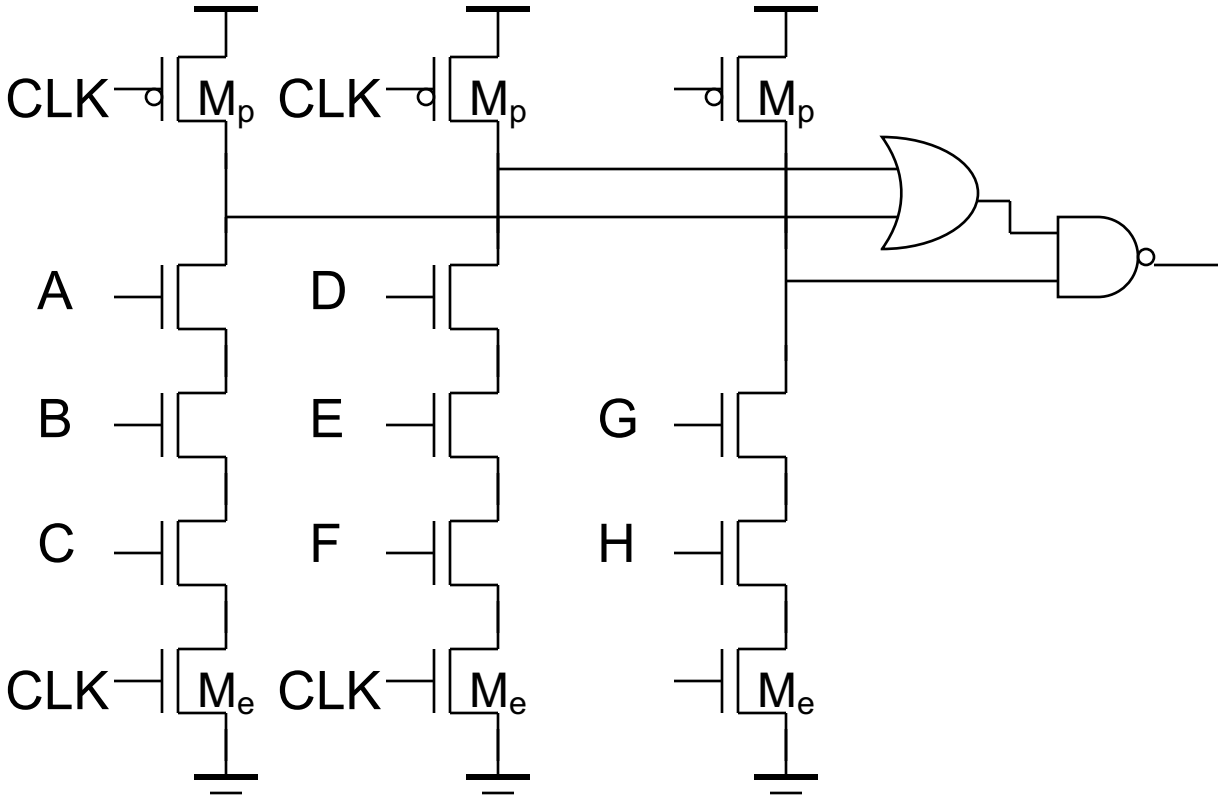
Differential (Dual Rail) Domino



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

Other Domino Variations

- Multiple output domino logic – exploits the fact that certain outputs are subsets of other outputs to generate a number of logic functions in a single gate.
- Compound domino

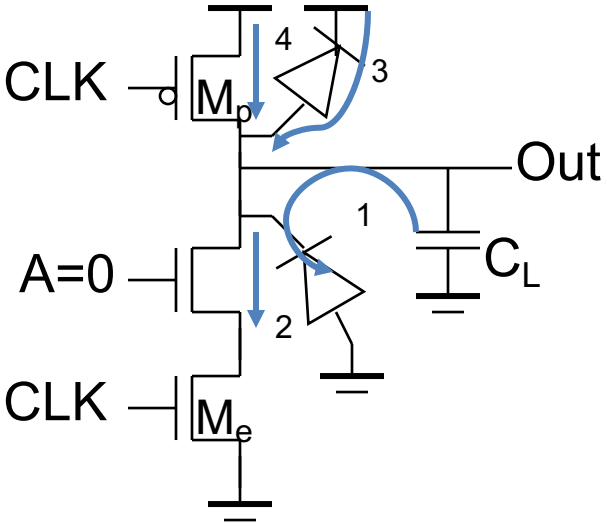


Domino Summary

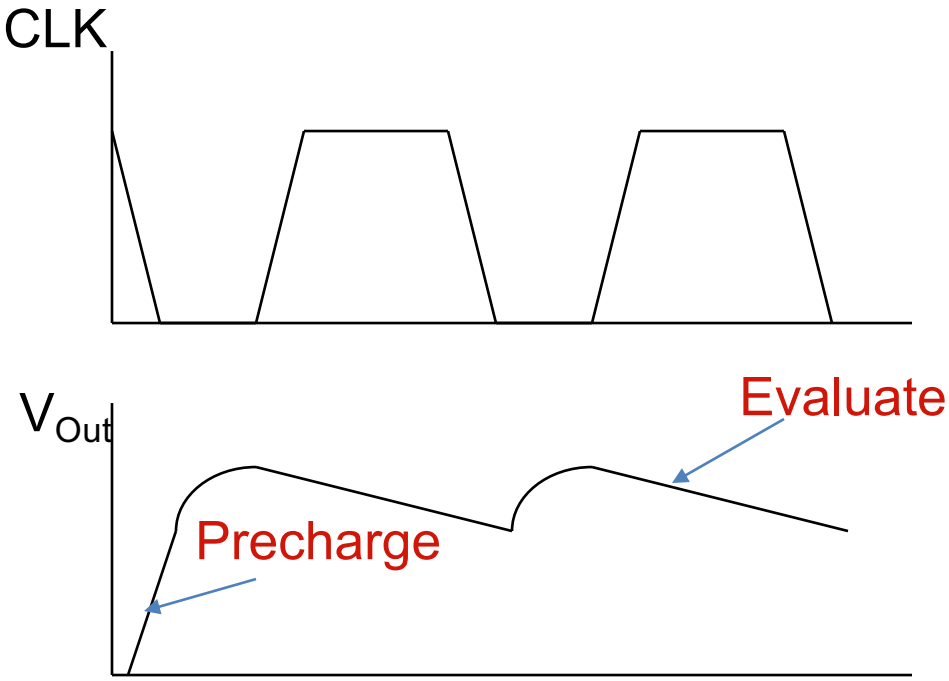
- Domino logic is attractive for high-speed circuits
  - 1.3 – 2x faster than static CMOS
  - But many challenges:
    - Monotonicity, leakage, charge sharing, noise
- Widely used in high-performance microprocessors in 1990s when speed was king
- Largely displaced by static CMOS now that power is the limiter
- Still used in memories for area efficiency

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.



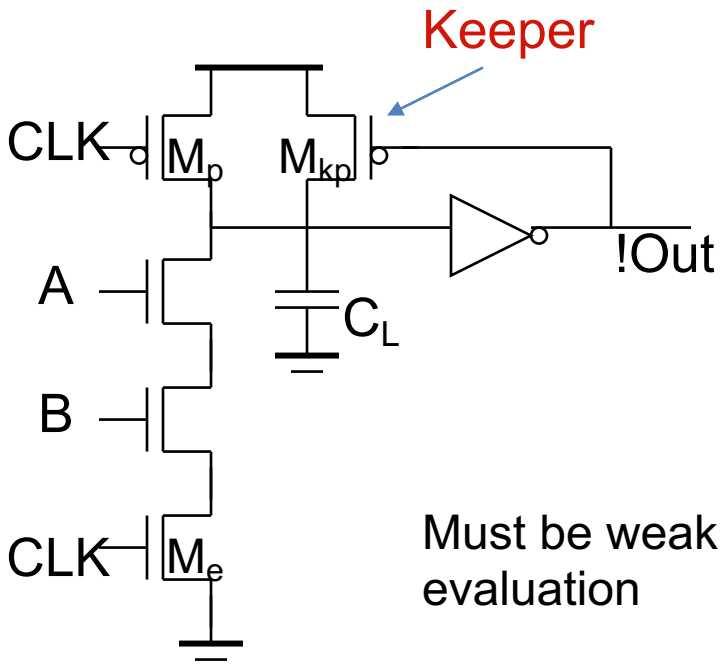
Leakage sources



Minimum clock rate!!!

A Solution to Charge Leakage

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

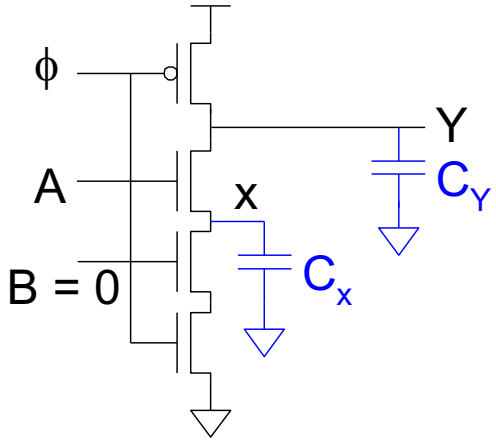


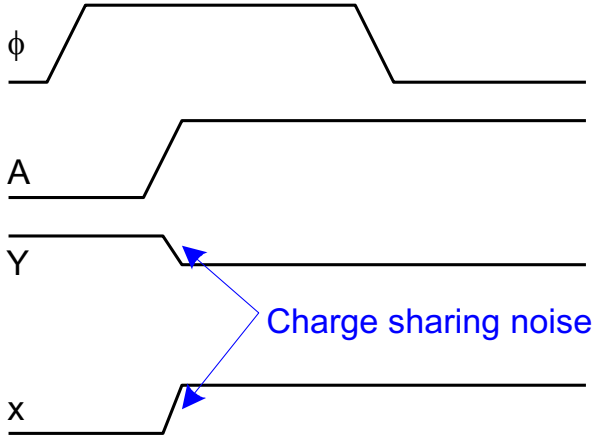
Must be weak enough not to fight evaluation

Same approach as level restorer for pass transistor logic

Charge Sharing

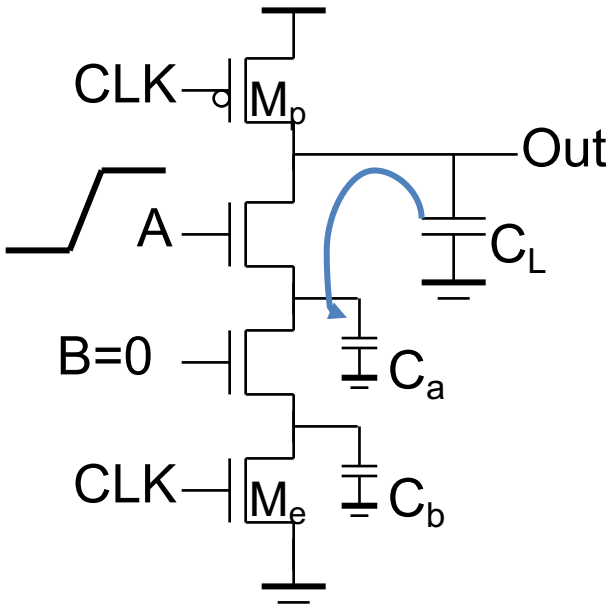
- Dynamic gates suffer from charge sharing





$$V_x = V_Y =$$

Malfunction of Charge sharing

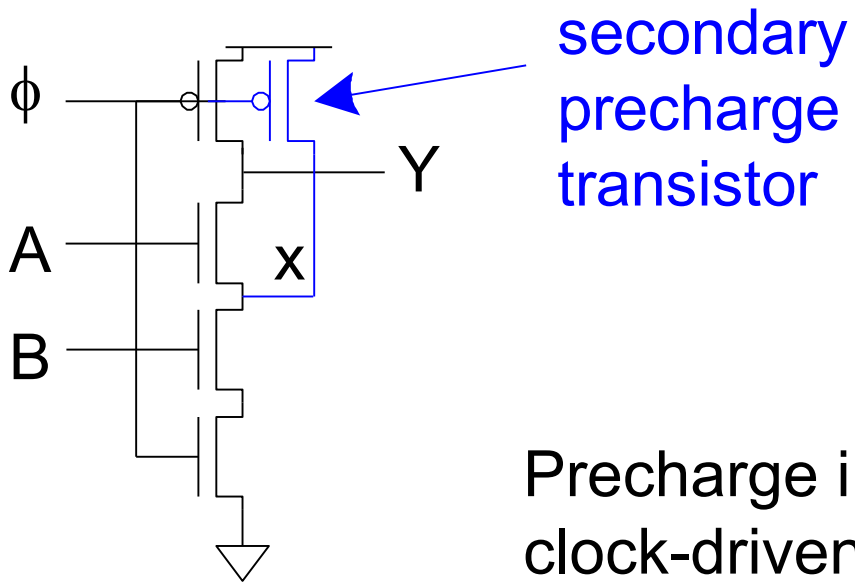


Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to static power consumption by downstream gates and possible circuit malfunction.

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.

Solution to Charge Redistribution

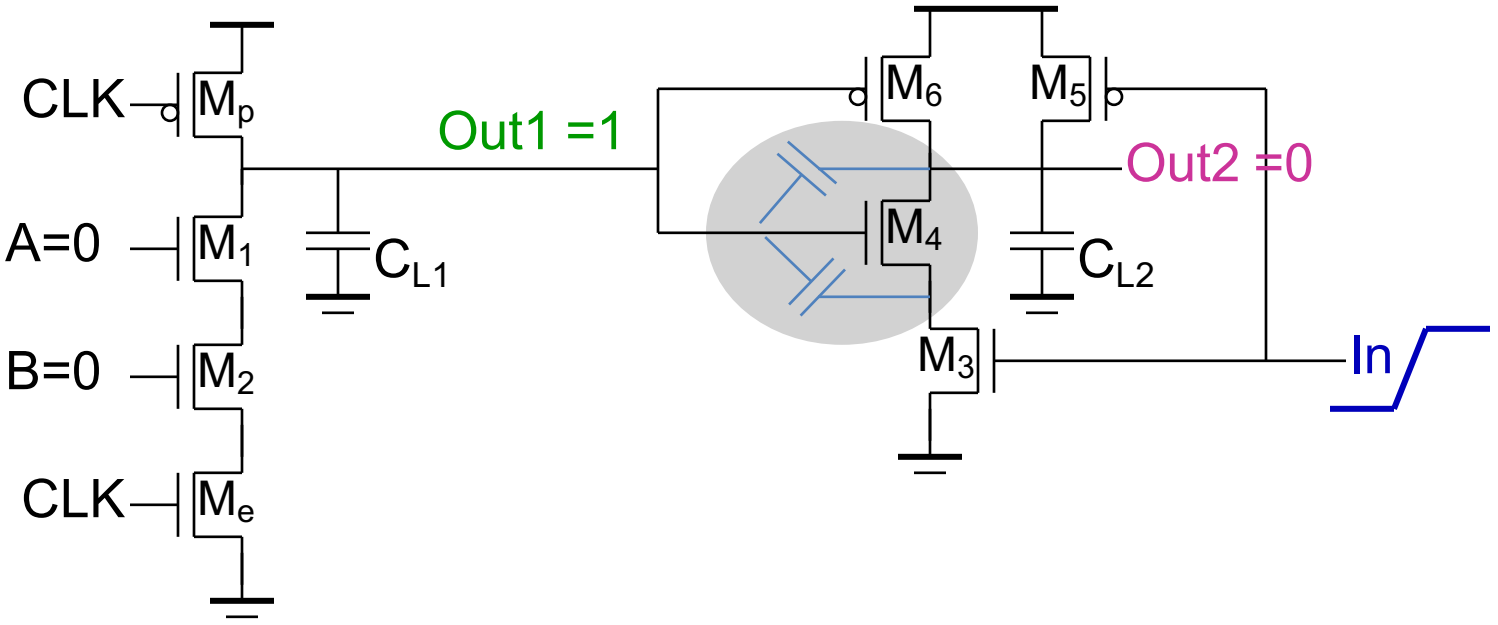
- Solution: add secondary precharge transistors
  - Typically need to precharge every other node
- Big load capacitance C_Y helps as well



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

Backgate Coupling

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

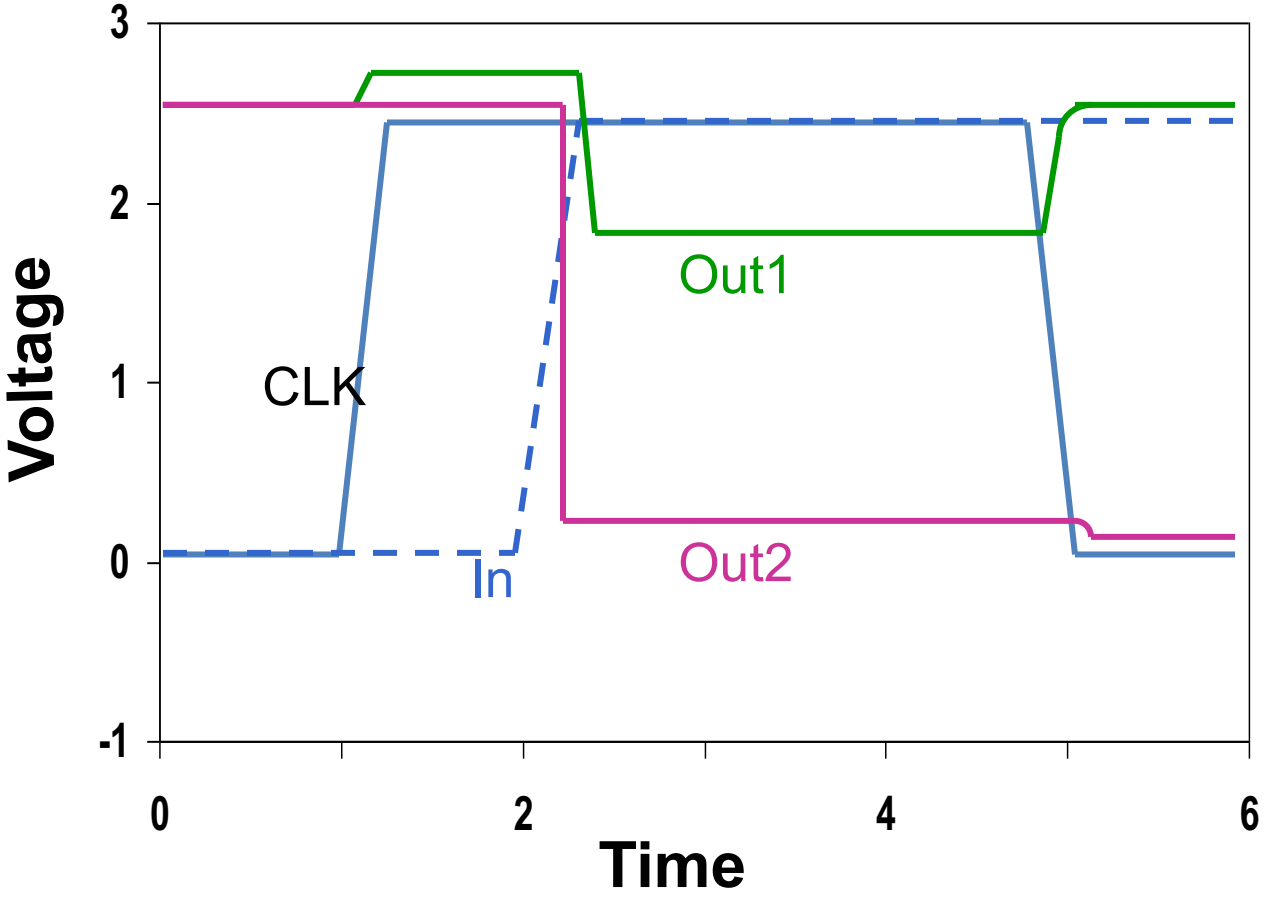


Dynamic NAND

Static NAND

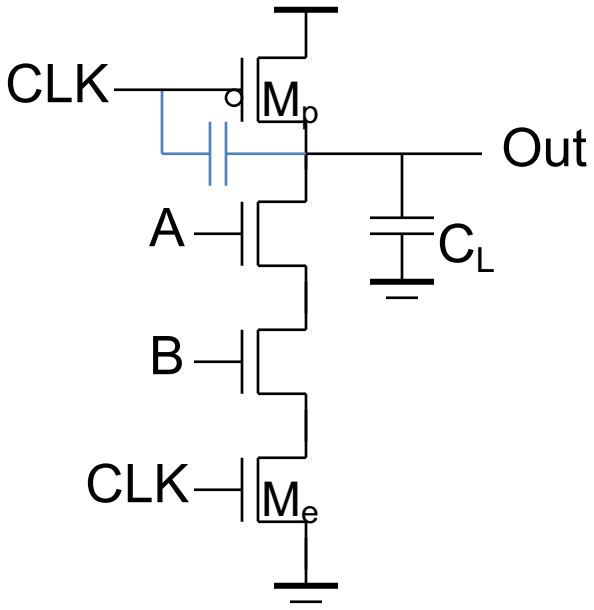
Backgate Coupling Effect

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



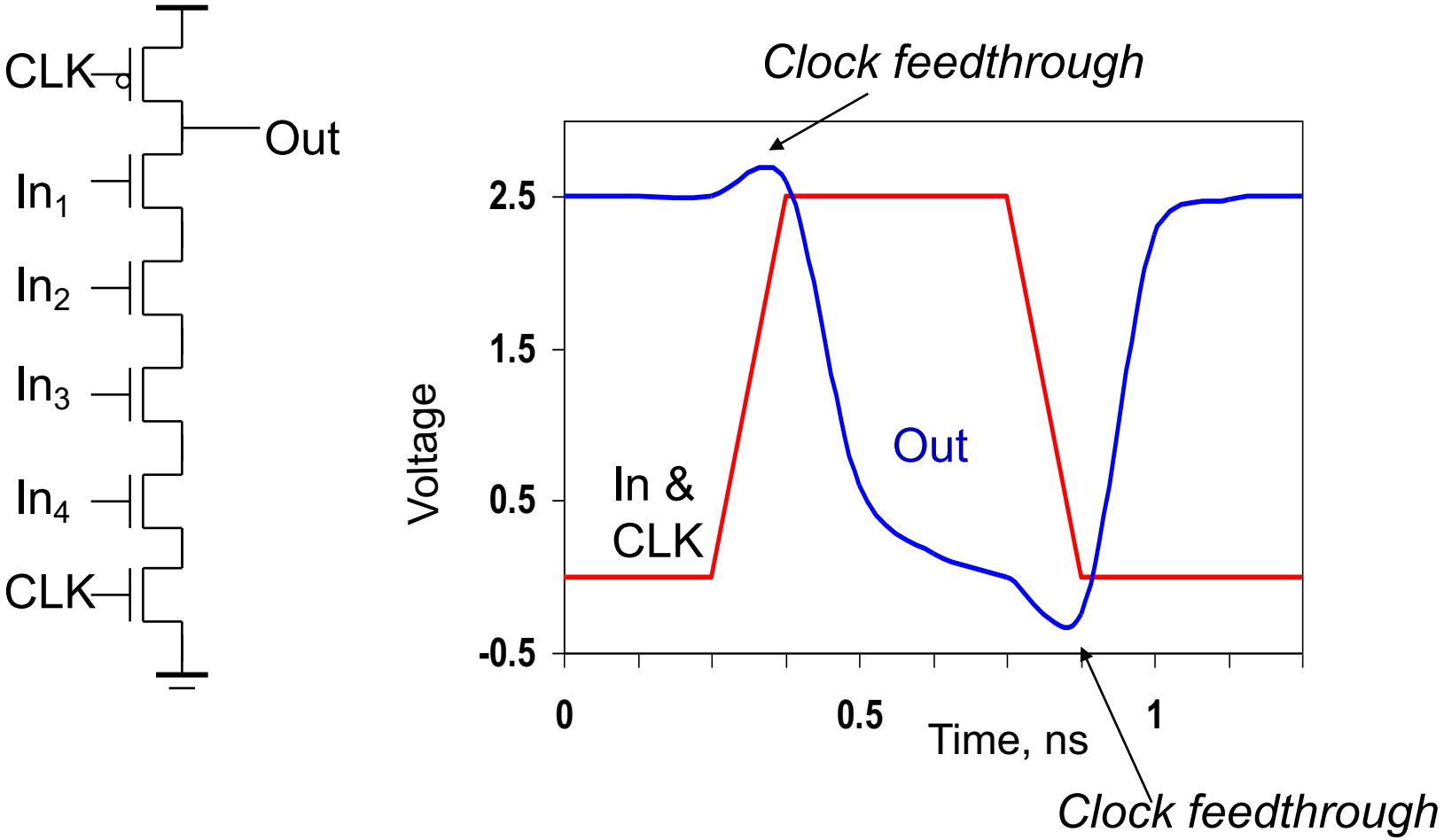
Clock Feedthrough

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

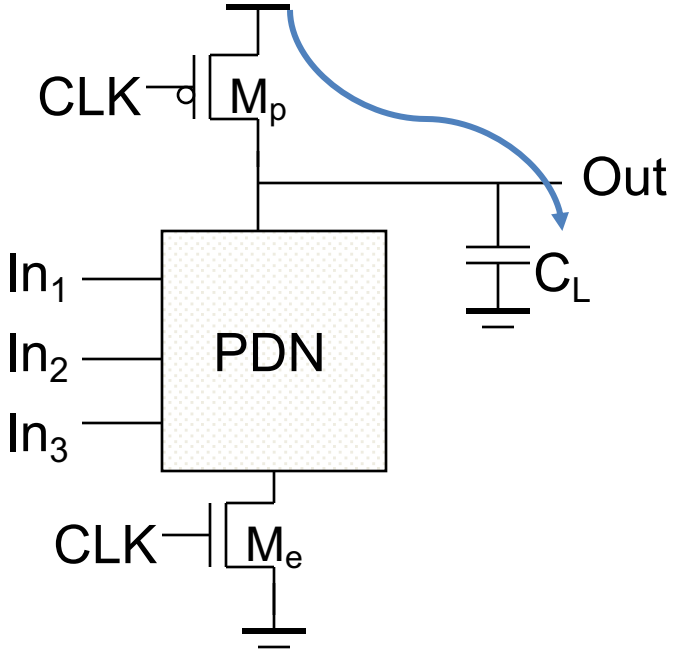


Coupling between Out and CLK input of the precharge device due to the gate-drain capacitance. So voltage of Out can rise above V_{DD}. The fast rising (and falling edges) of the clock couple to Out.

Clock Feedthrough



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!

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

Power

- Domino gates have high activity factors
  - Output evaluates and precharges
    - If output probability = 0.5, α = 0.5
      - Output rises and falls on half the cycles
  - Clocked transistors have α = 1
- Leads to very high power consumption

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!

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
  - Ignoring the influence of precharge time on the switching speed of the gate, t_{pLH} = 0 but the presence of the evaluation transistor slows down the t_{pHL}

Properties of Dynamic Gates

- Power dissipation should be better
  - consumes only dynamic power – no short circuit power consumption since the pull-up path is not on when evaluating
  - lower C_L- 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)
  - by construction can have at most one transition per cycle – no glitching
- But power dissipation can be significantly higher due to
  - ***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)
- Needs a precharge clock

How to Choose a Logic Style

- Depends on your area, speed, power and ease of design
- Static CMOS is the most robust to noise
  - Best general purpose logic style
  - Complementary CMOS with large fan-in and complex gates
    - high area cost, and lower performance
    - Pseudo NMOS
    - Pass-transistors for MUX and adders
- Dynamic logic is the fastest
  - Noise sensitive
  - A lot of design problems
  - Lower bound on freq. due to charge leakage

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