VLSI Design EE 523
Spring 2024

Shahid Masud

Lecture 21

Topics for lecture 21

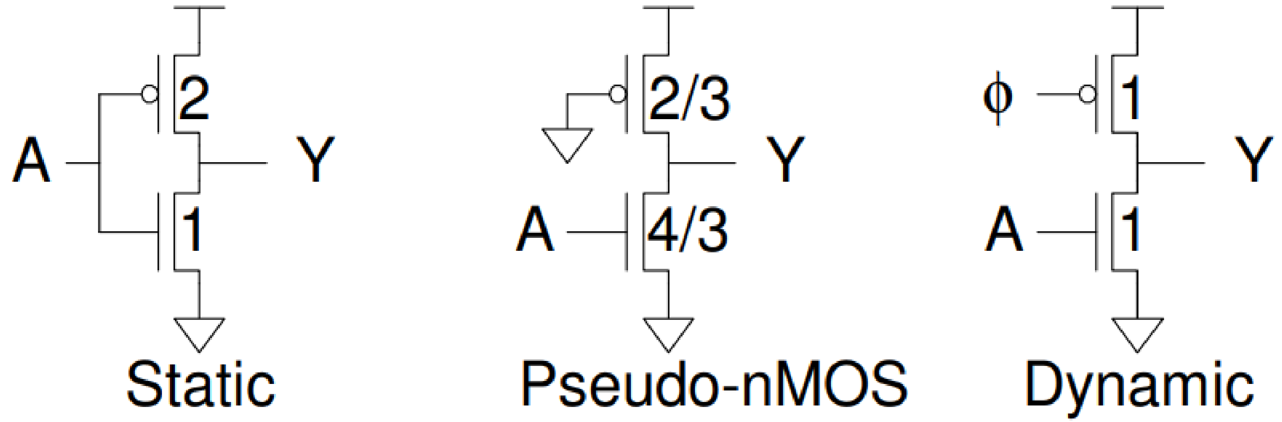
- Dynamic Logic Circuits
- Cascading, Monotonicity
- Charge Sharing Problems
- Domino Circuit Design
- **Quiz 4 was held**

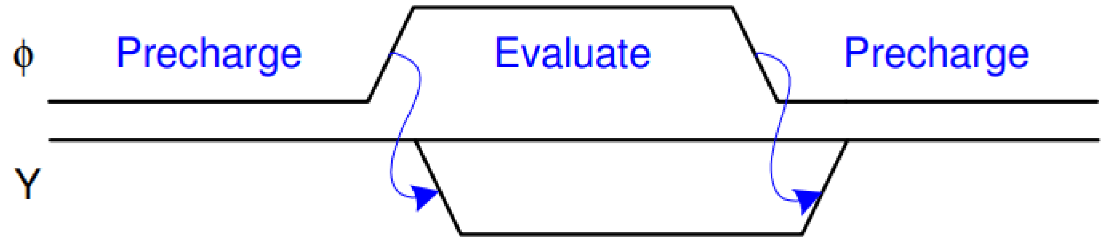
DYNAMIC CMOS LOGIC

What is Dynamic Logic?

Dynamic Logic

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

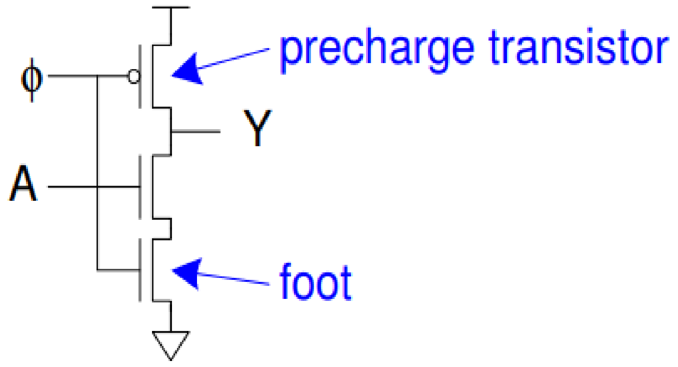


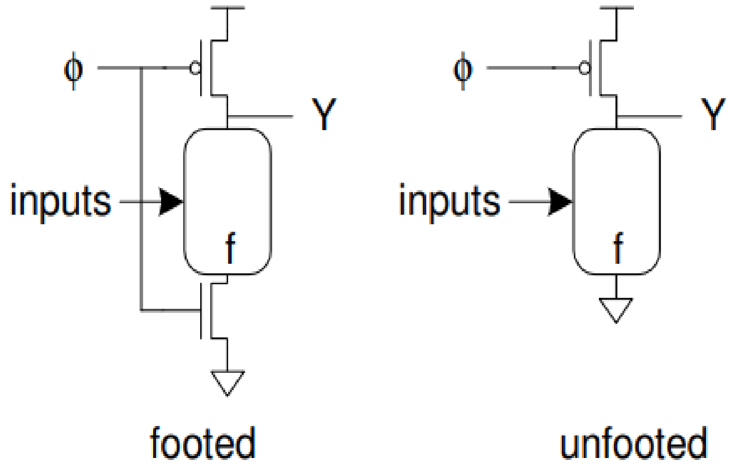


The FOOT Transistor

The Foot

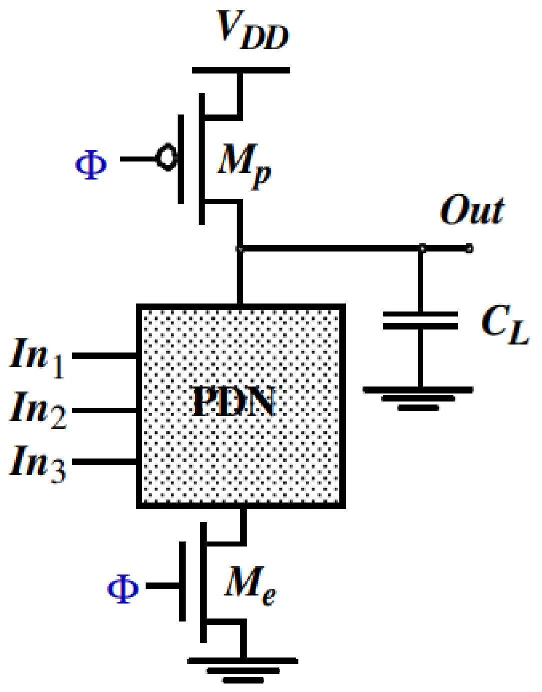
- What if pulldown network is ON during precharge?
- Use series evaluation transistor to prevent fight.



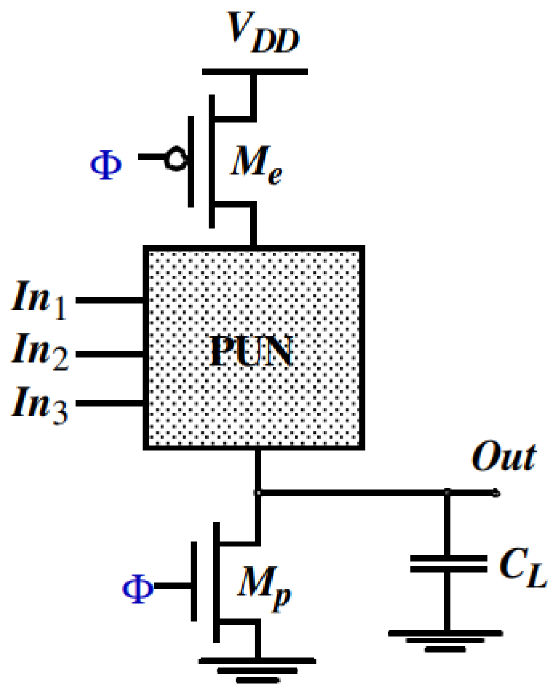


Dynamic Logic Operation

Dynamic Logic



Φn network



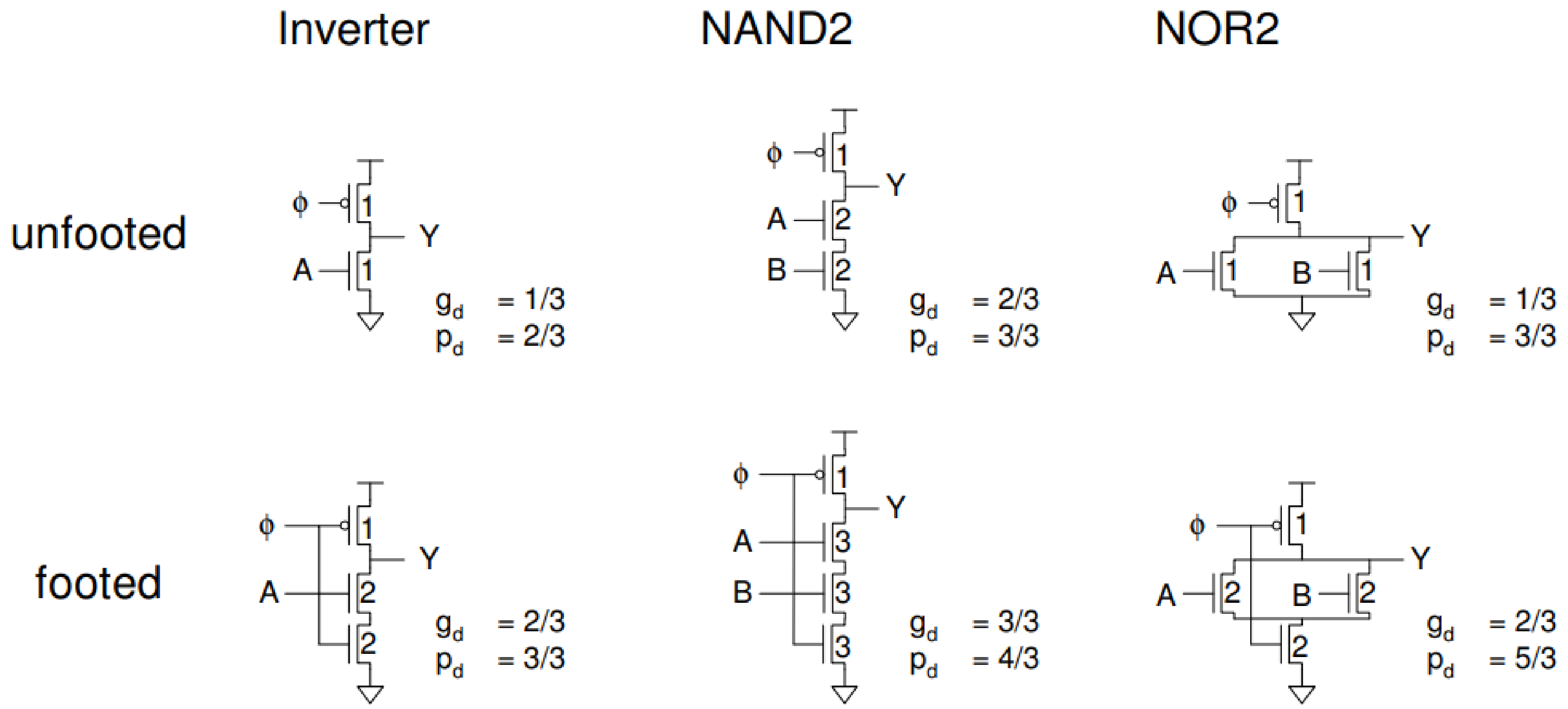
Φp network

2 phase operation:

- Precharge
- Evaluation

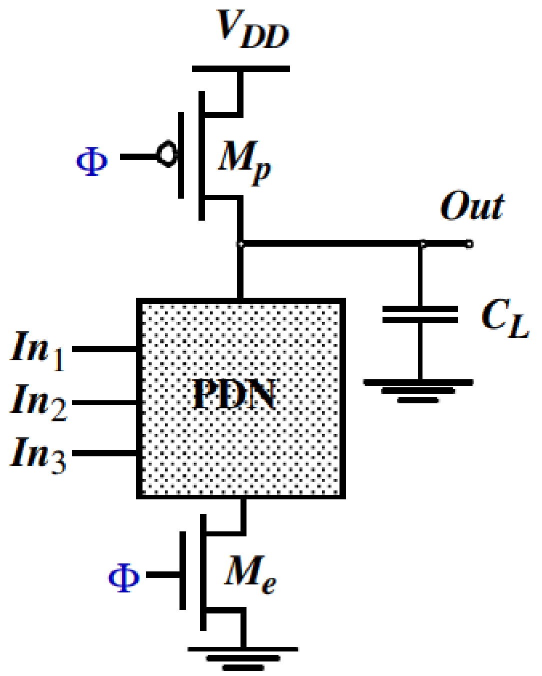
LE in Dynamic Logic

Logical Effort



Dynamic Logic Principles

Dynamic Logic: Principles



- **Precharge**

  $\Phi = 0$, *Out* is precharged to V_{DD} by M_p. M_e is turned off, no dc current flows (regardless of input values)

- **Evaluation**

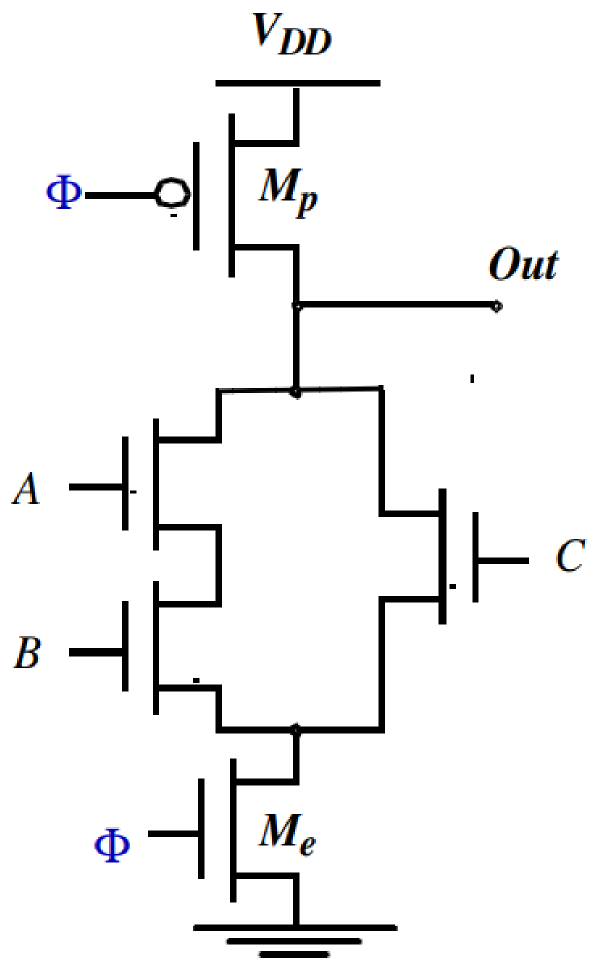
  $\Phi = 1$, M_e is turned on, M_p is turned off. Output is pulled down to zero depending on the values on the inputs. If not, precharged value remains on C_L.

Important: Once *Out* is discharged, it cannot be charged again!
Gate input can make only one transition during evaluation

- Minimum clock frequency must be maintained
- Can M_e be eliminated?

Dynamic Logic Example

Example



- Ratioless
- No Static Power Consumption
- Noise Margins small (NM_L)
- Requires Clock

Evolution from Static Gate to Dynamic Gate

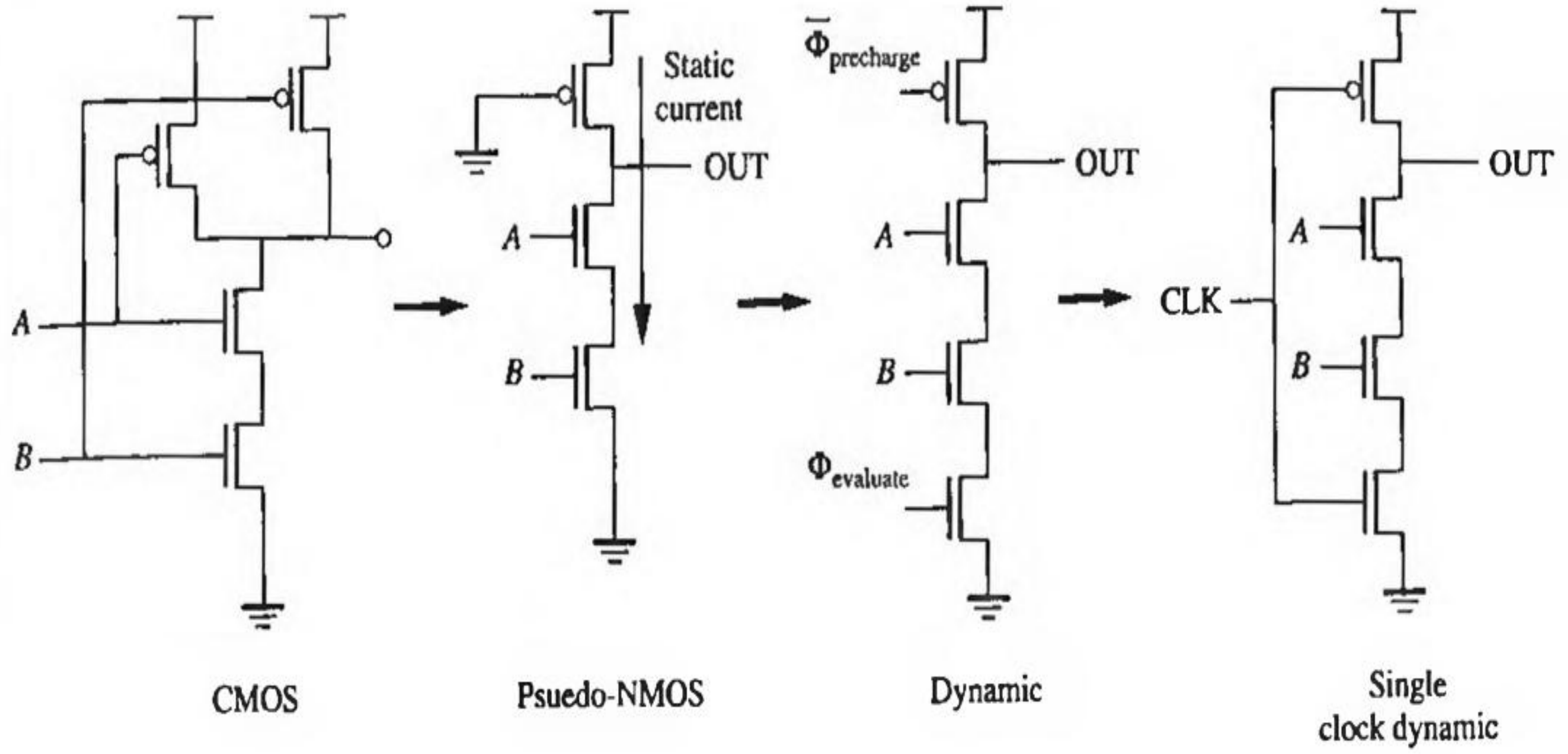
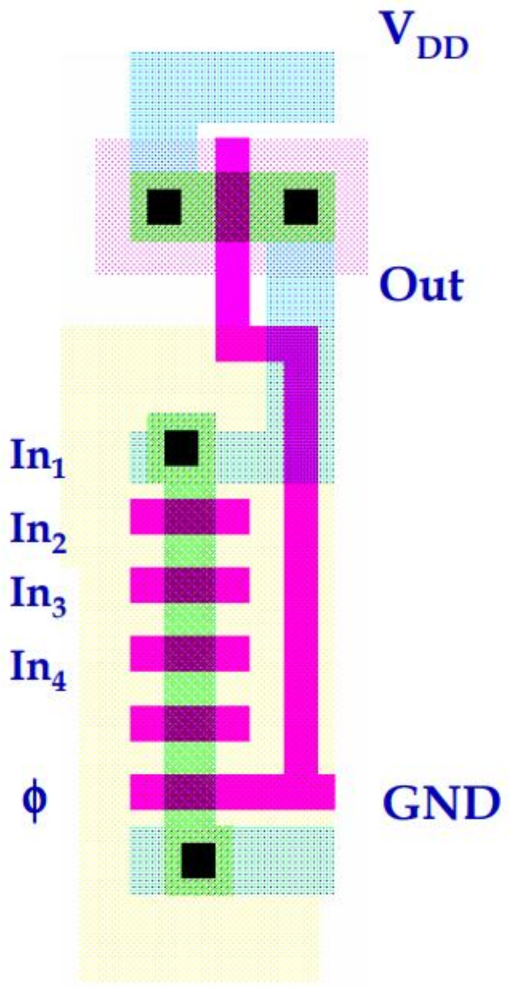
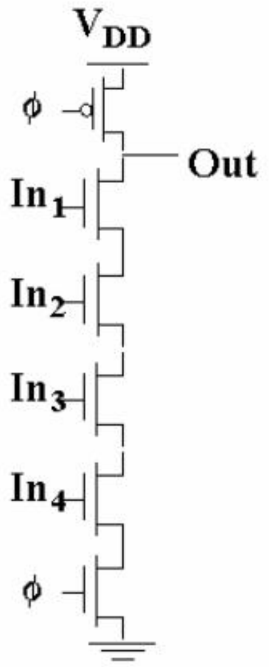


Figure 7.25
Evolution from static gate to dynamic gate.

Dynamic 4 Input NAND Gate

Dynamic 4 Input NAND Gate





Design of Dynamic Gate

Dynamic Gate Implementations

Example 7.6

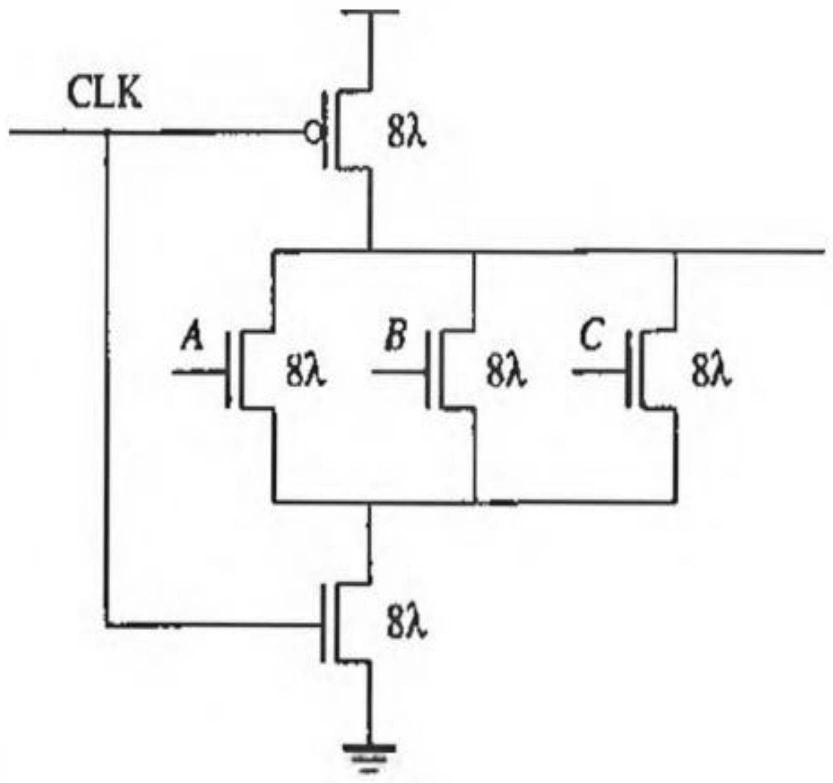
Problem:

Implement a 3-input NOR gate in dynamic logic and explain its operation. Size the transistors to deliver the same delays as a conventional CMOS inverter (PMOS 8λ:2λ, NMOS 4λ:2λ).

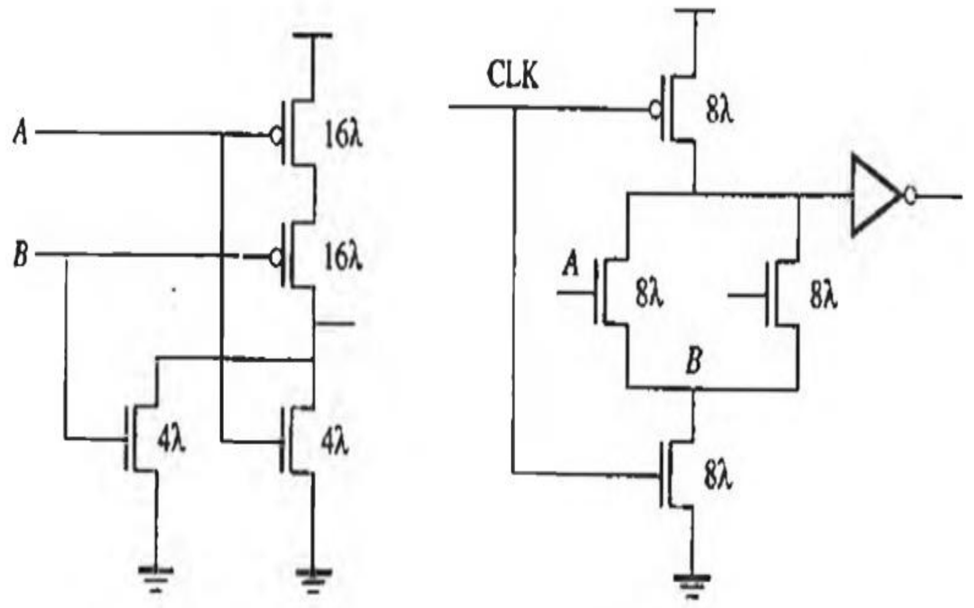
Solution:

The dynamic 3-input NOR gate requires three parallel NMOS transistors, plus two transistors connected to the CLK input: one PMOS pull-up and one NMOS foot transistor. When CLK is low, the precharge transistor pulls the output to V_{DD}. When the clock goes high, the evaluate transistor turns on. If *A* or *B* or *C* is high, the output will be discharged to Gnd. If all three inputs are low, the output will remain high.

The pull-up requires a width of 8λ to deliver the same rise delay as the CMOS inverter. This is somewhat irrelevant since all gates are precharged simultaneously. The pull-down tree should also be sized with 8λ transistors to produce the equivalent 4λ device to deliver the same fall delay as the CMOS inverter. The final sizes are shown below:



LE of Static and Dynamic Gates



(a) Static NOR gate (b) Dynamic NOR gate

Figure 7.29
Comparison of static and dynamic logical effort.

For the static gate, we already determined its logical effort to be 5/3. This can be derived by comparing the input capacitance of the static NOR to a corresponding inverter. The inverter size of 8λ for the pull-up and 4λ for the pull-down produces a ratio of

$$LE_{NOR} = \frac{\text{(NOR input cap.)}}{\text{(Inverter input cap.)}} = \frac{16\lambda + 4\lambda}{8\lambda + 4\lambda} = 5/3$$

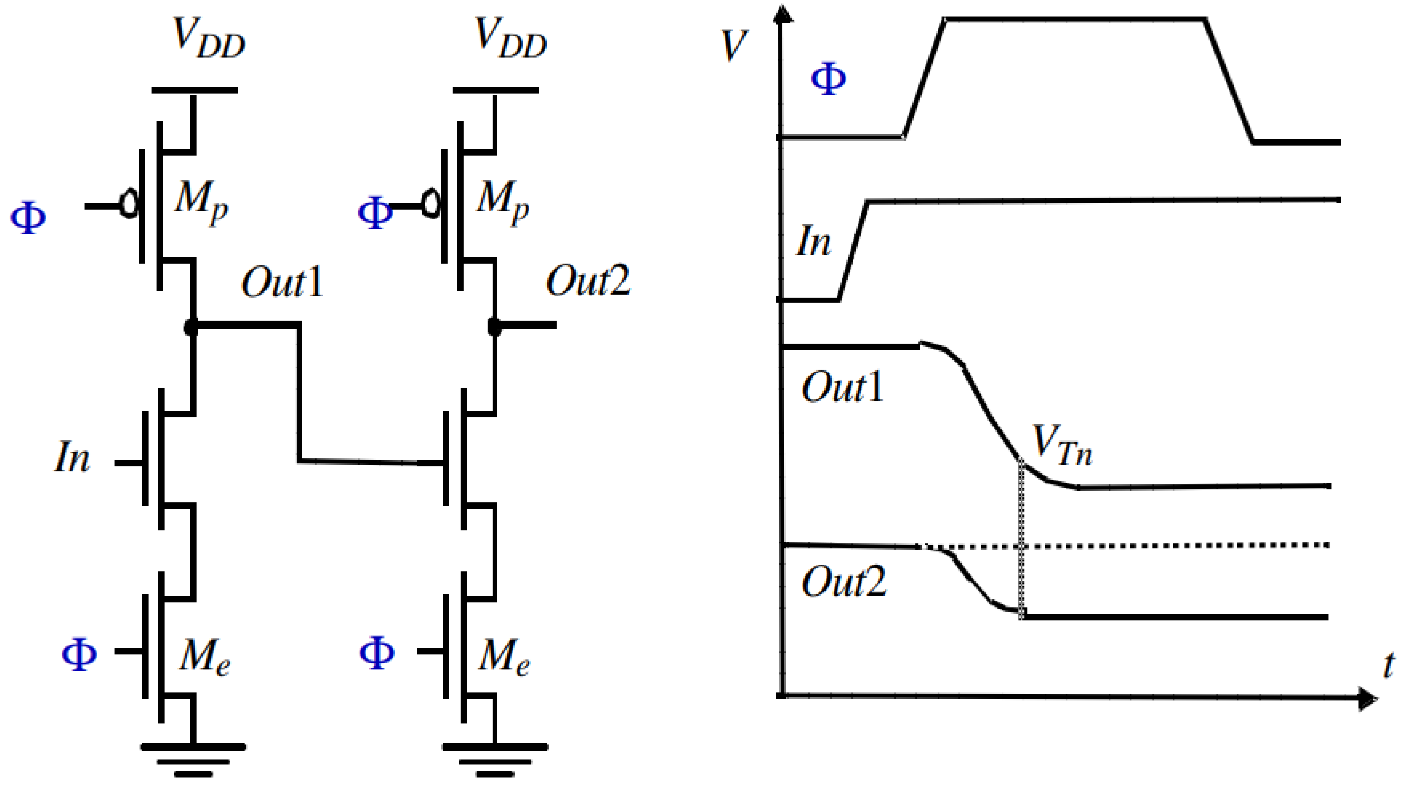
For the domino circuit, the input capacitance is simply 8λ since it is only connected to NMOS devices. Then,

$$LE_{dyn\_NOR} = \frac{\text{(Dynamic input cap.)}}{\text{(Inverter input cap.)}} = \frac{8\lambda}{12\lambda} = 2/3$$

Since the extra inverter in domino logic has an LE = 1, the average LE for each of the two gates is the geometric mean of the product of their logical efforts (i.e., $LE_{avg} = \sqrt{(2/3)\ (1)} \approx 0.8$). Therefore, the domino stage is better in terms of its overall drive capability and input capacitive loading.

Cascaded Dynamic Gates

Cascading Dynamic Gates

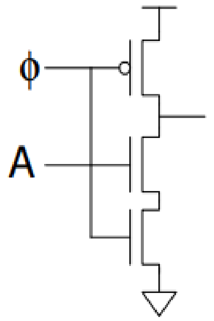


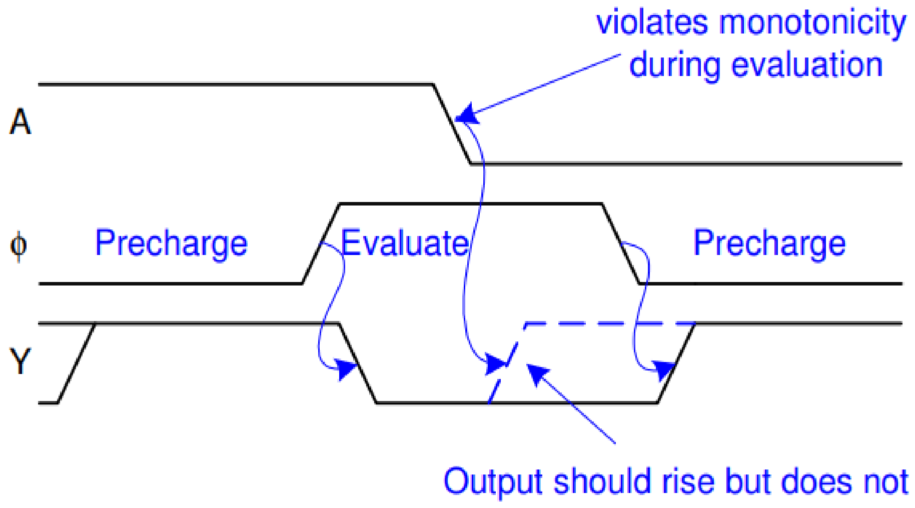
Internal nodes can only make 0-1 transitions during evaluation period

Monotonicity in Dynamic Gates

Monotonicity

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

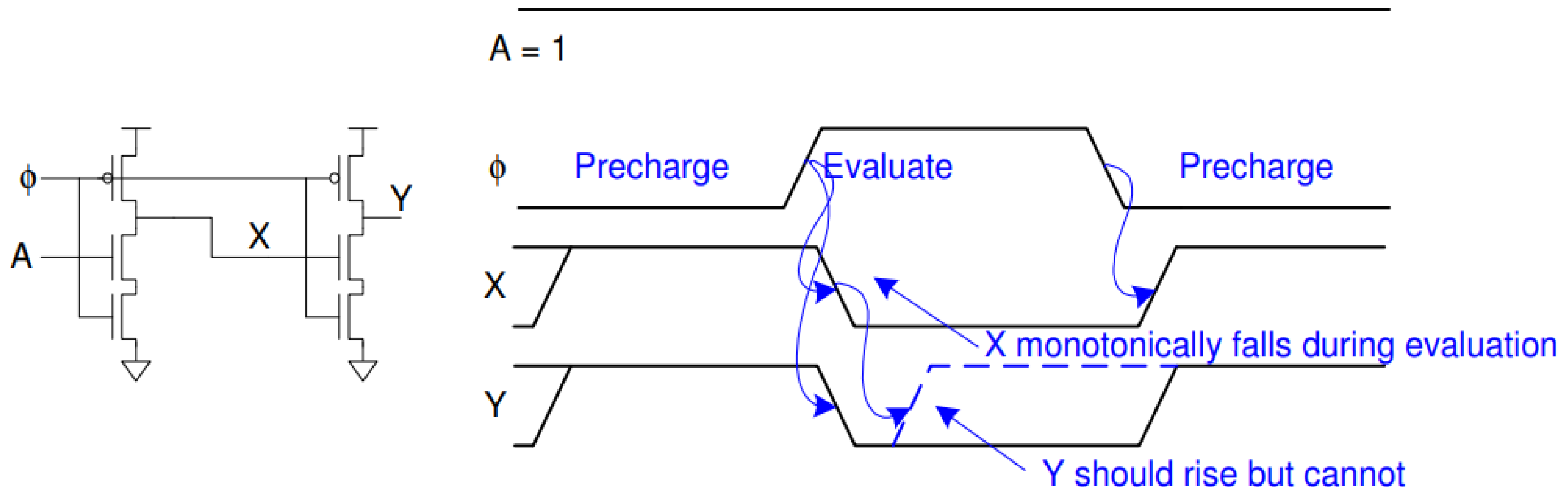



violates monotonicity during evaluation
A
ϕ
Precharge
Evaluate
Precharge
Y
Output should rise but does not

Monotonicity Problems in Dynamic Gates

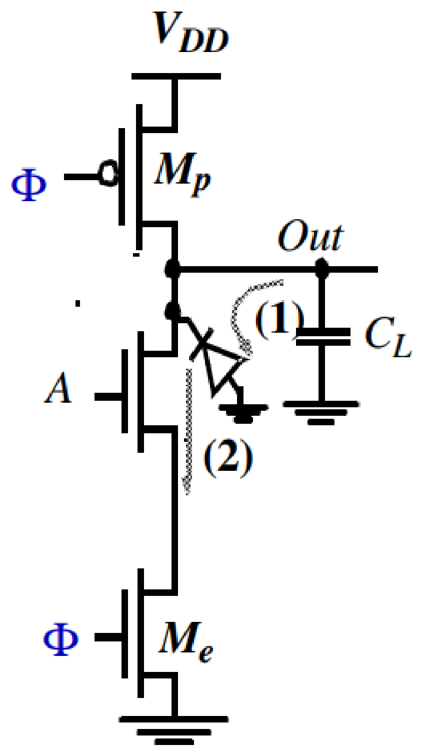
Monotonicity Woes

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

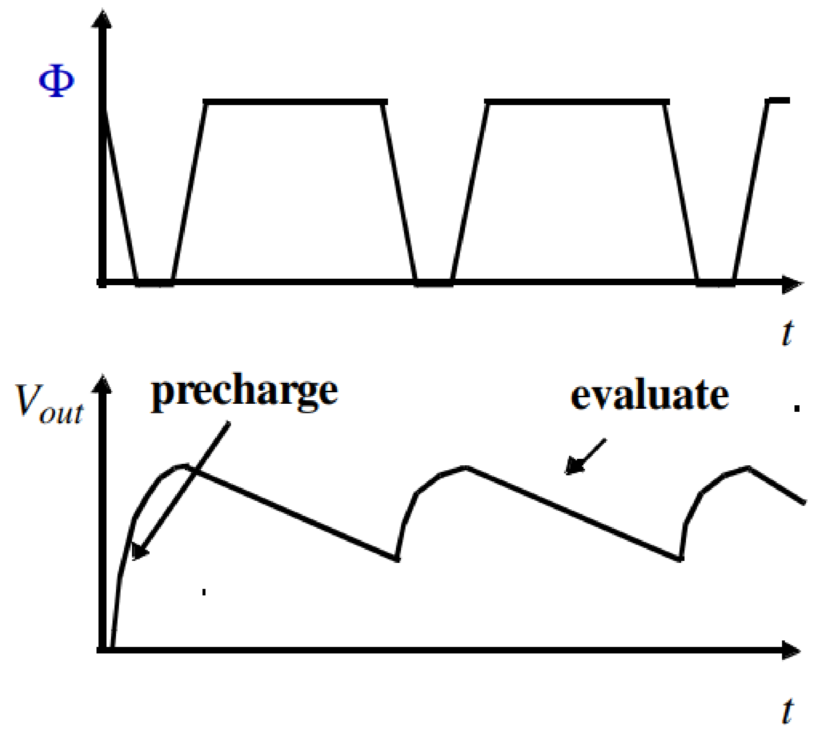


Reliability Issues in Dynamic Gates

Reliability Problems — Charge Leakage



(a) Leakage sources



(b) Effect on waveforms

A = 0

(1) Leakage through reverse-biased diode of the diffusion area

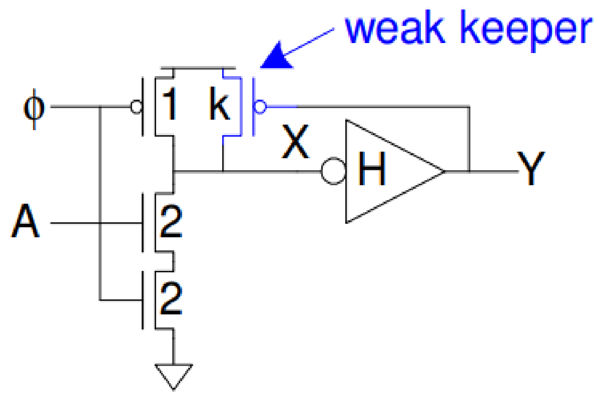
(2) Subthreshold current from drain to source

Minimum Clock Frequency: > 1 MHz

Charge Leakage in Dynamic Gates

Leakage

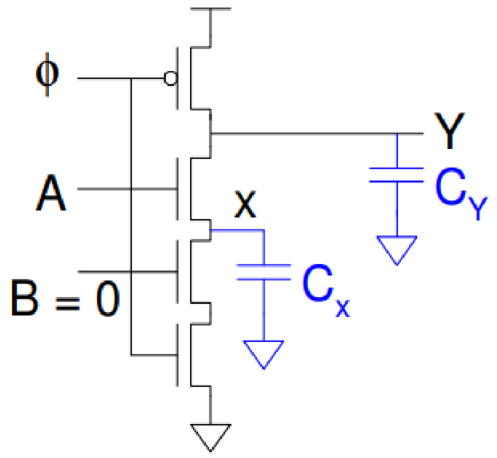
- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{OFF} \neq 0$)
  - Dynamic value will leak away over time
  - Formerly miliseconds, now nanoseconds!
- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation

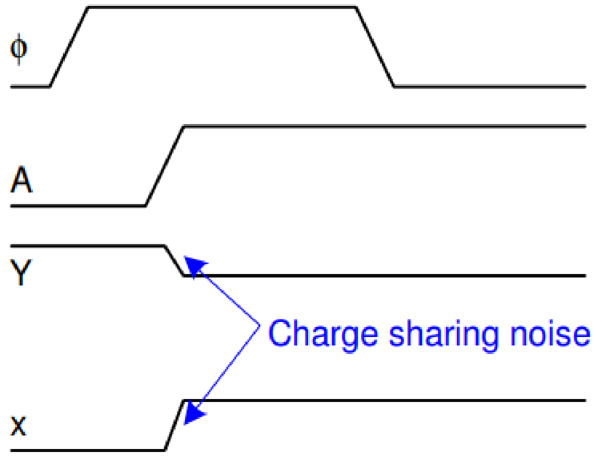


What is Charge Sharing?

Charge Sharing

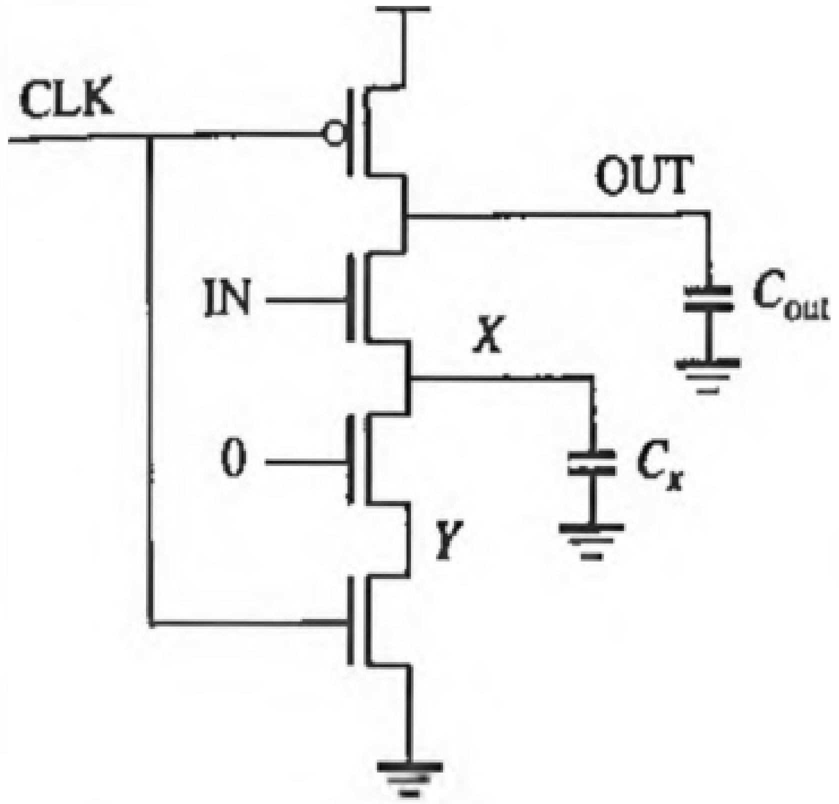
- Dynamic gates suffer from charge sharing





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

Charge Sharing Example in Dynamic Logic



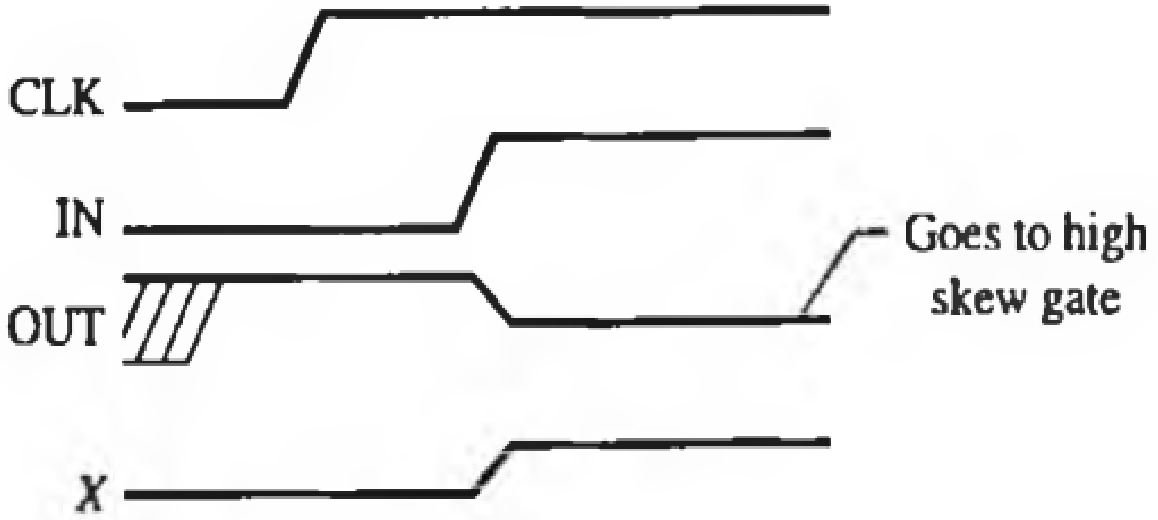
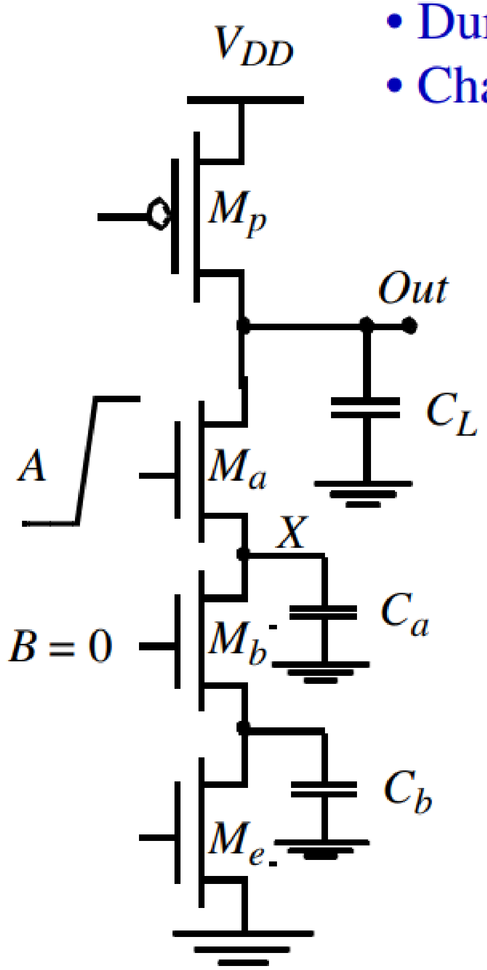


Figure 7.30
Charge-sharing example in domino logic.

Charge Sharing in Dynamic Gates

Charge Sharing (redistribution)

- Assume: during precharge, A and B are 0, C_a is discharged
- During evaluation, B remains 0 and A rises to 1
- Charge stored on C_L is now redistributed over C_L and C_a



$$C_L V_{DD} = C_L V_{out}(t) + C_a V_X$$

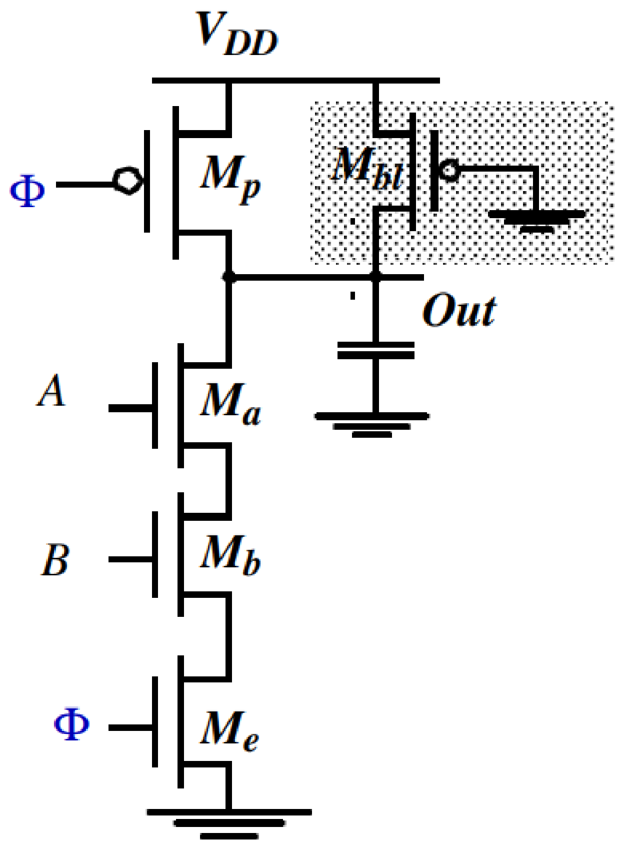
$V_X = V_{DD} - V_t$, therefore

$$\delta V_{out}(t) = V_{out}(t) - V_{DD} = -\frac{C_a}{C_L}(V_{DD} - V_t)$$

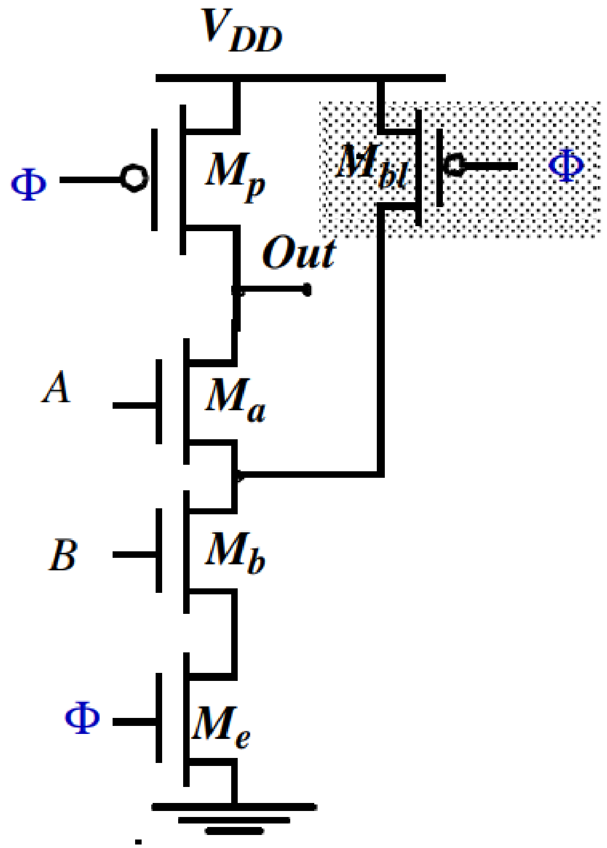
Desirable to keep the voltage drop below threshold of pMOS transistor (why?) $\Rightarrow C_a/C_L < 0.2$

Charge Redistribution

Charge Redistribution - Solutions



(a) Static bleeder

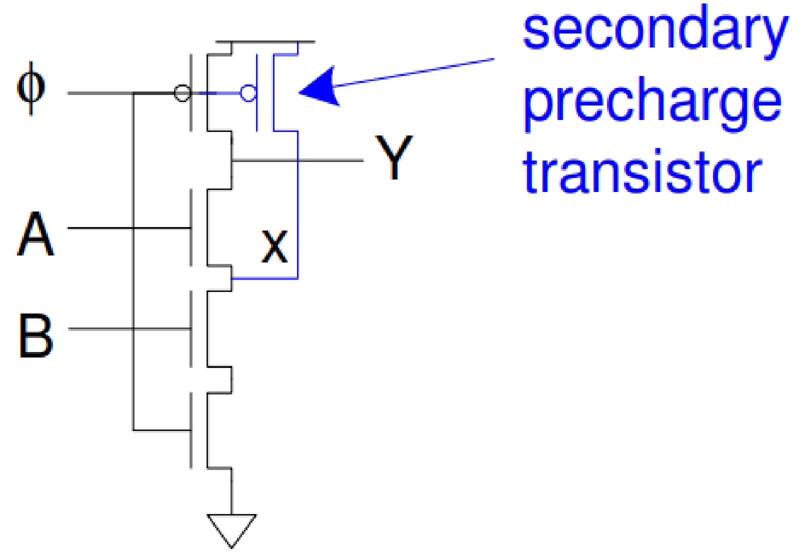


(b) Precharge of internal nodes

Secondary Precharge in Dynamic Gates

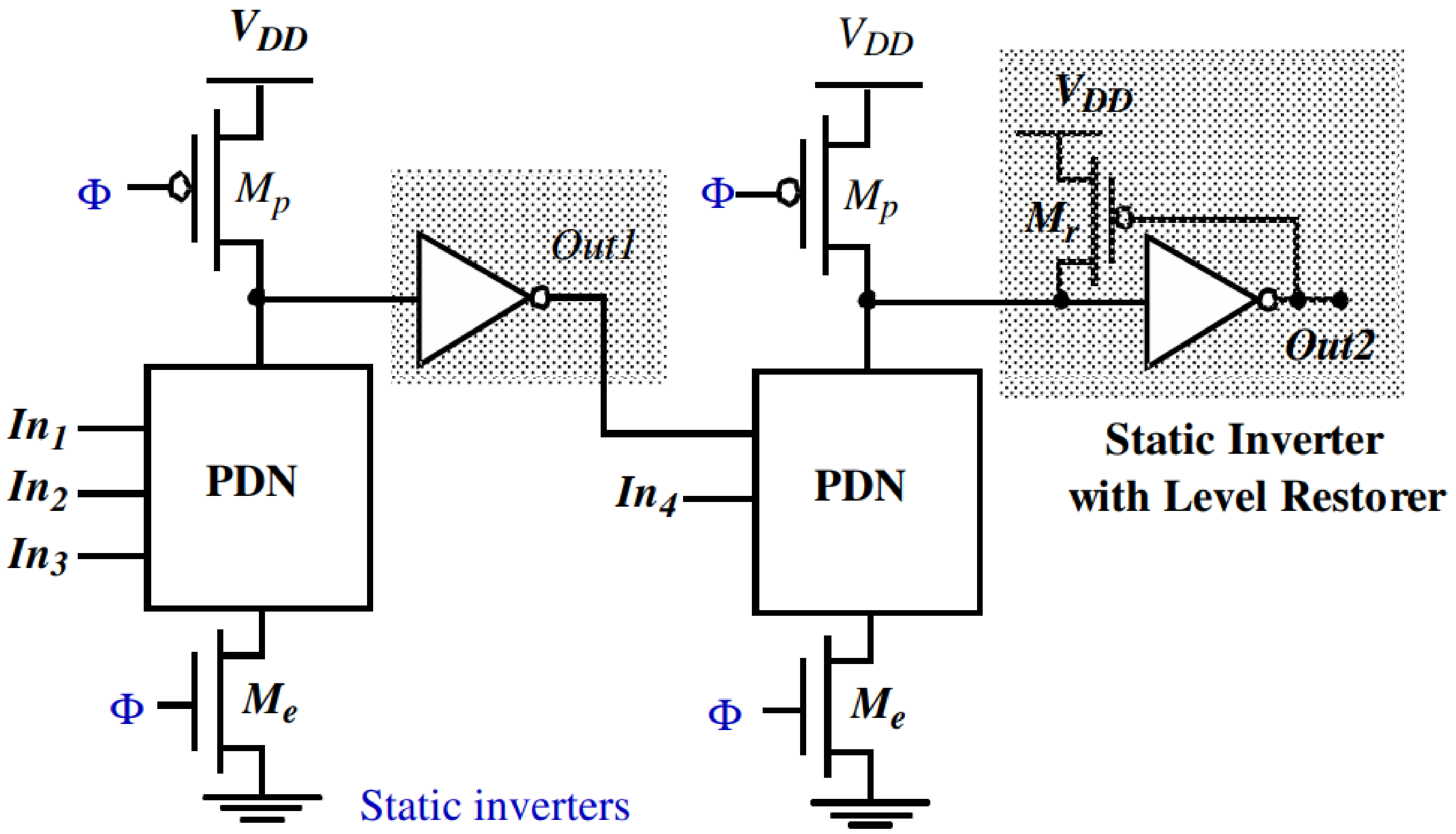
Secondary Precharge

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



Domino Logic

What is Domino Logic?



Static inverters between dynamic stages

Charge Retention Using Keeper Transistors

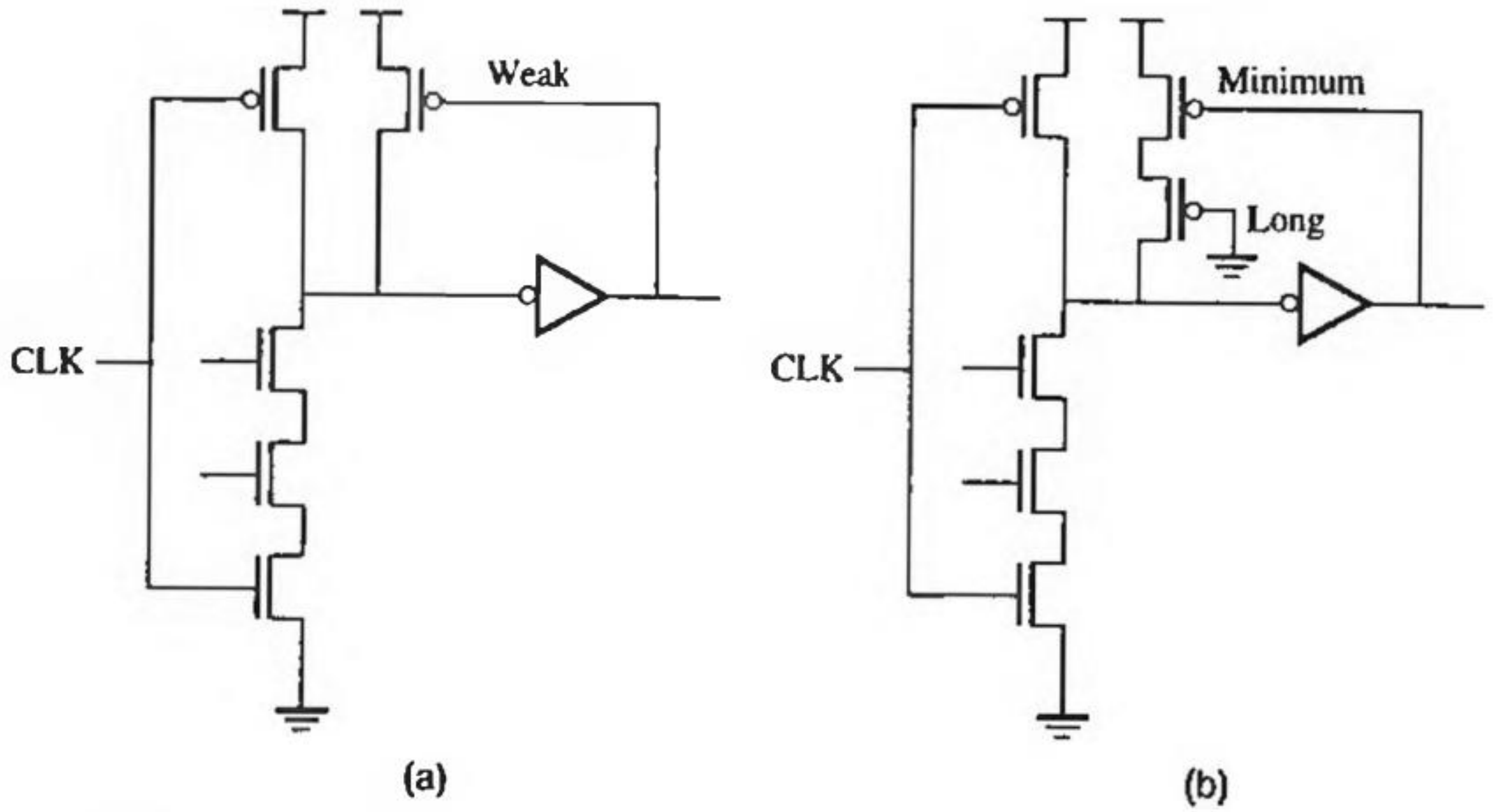


Figure 7.31
Minimizing the effects of charge sharing using keepers.

Domino Cascaded Gates

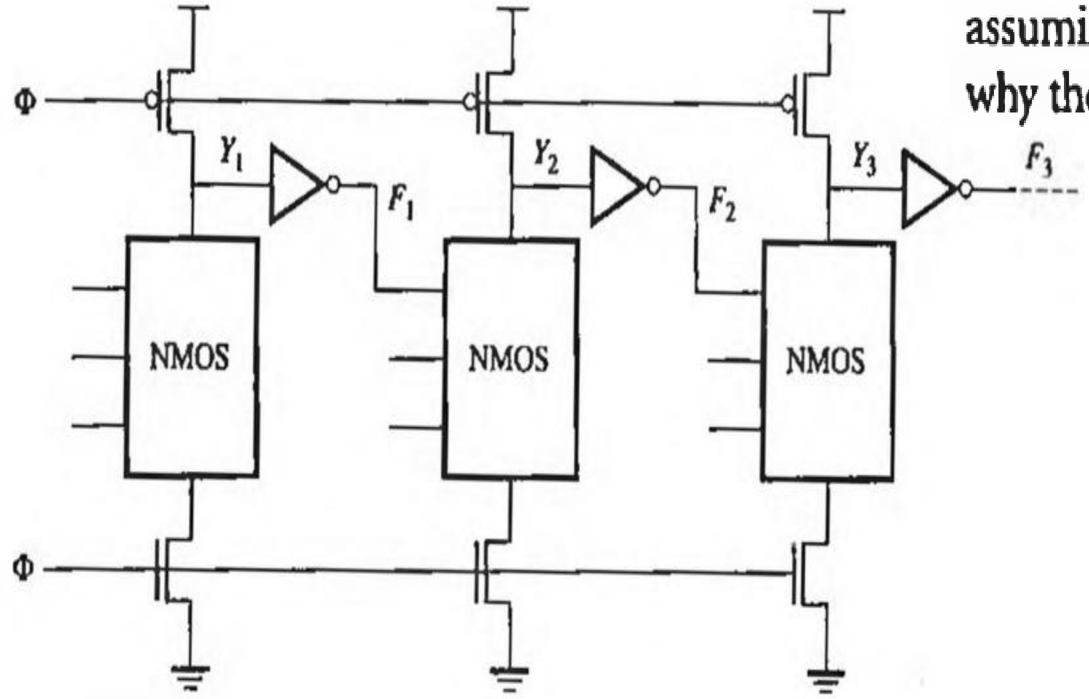


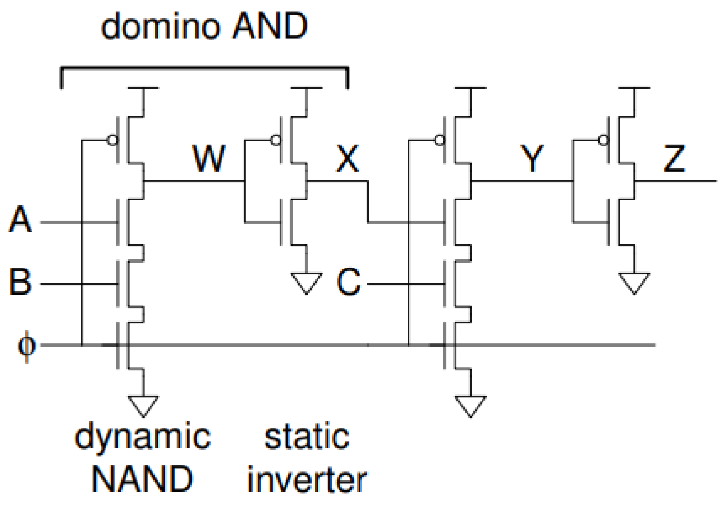
Figure 7.28
Domino cascaded gates.

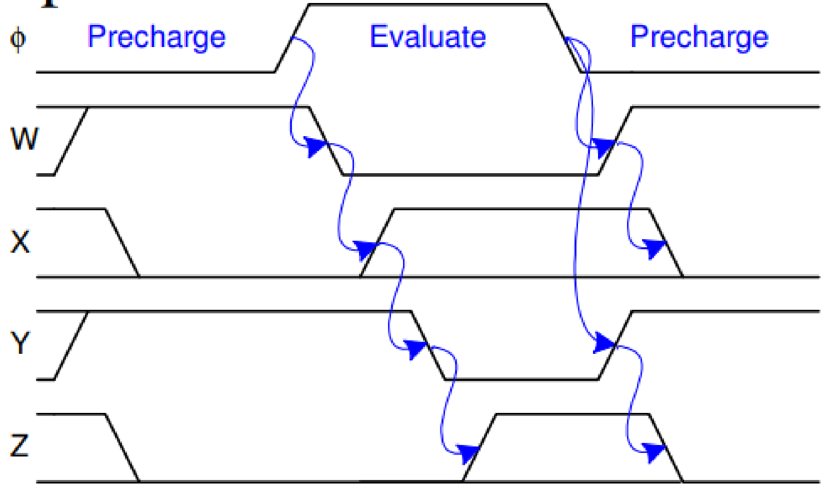
A cascade of three domino logic stages is shown in Figure 7.28. All three stages are precharged high when Φ is low. The clock, Φ, need not be low for very long since all stages are precharged simultaneously. In fact, the clock should only be low until all inverter outputs are low. When Φ goes high, we enter the evaluation phase where the internal nodes (labeled Y_1, Y_2, Y_3) will fall like dominos in order from left to right, assuming there is a path to the Gnd through their respective *n*-complexes. This is why the name *domino* is used to describe logic circuits implemented in this fashion.

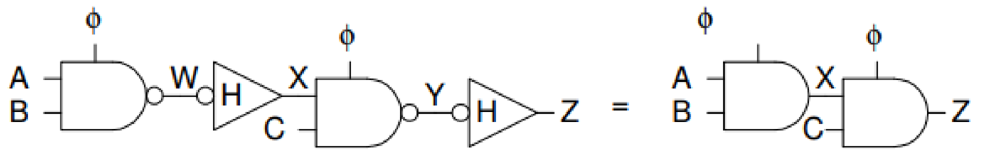
Domino Gate Operation

Domino Gates

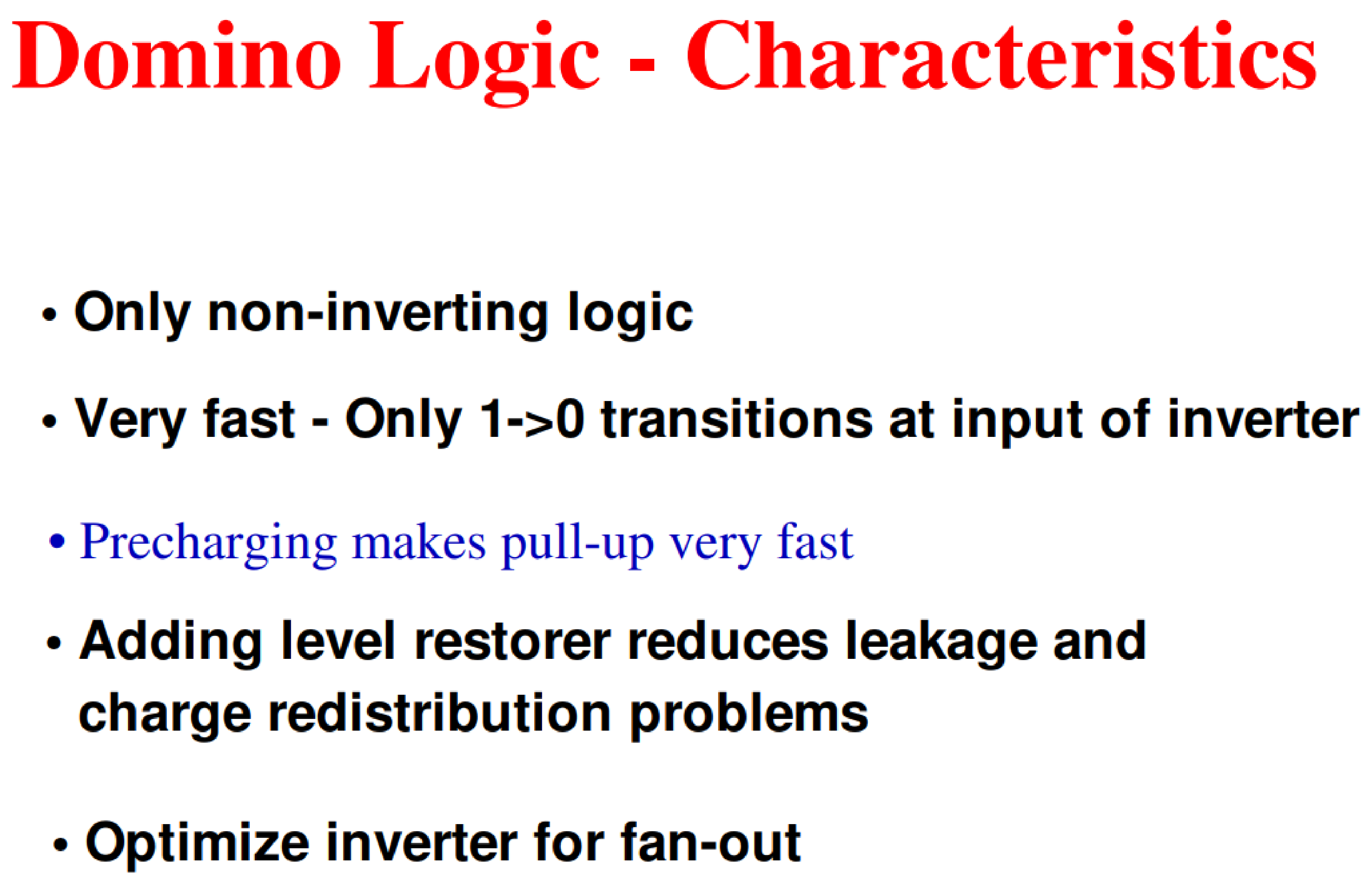
- Follow dynamic stage with inverting static gate
  - Dynamic / static pair is called domino gate
  - Produces monotonic outputs







Domino Logic Characteristics



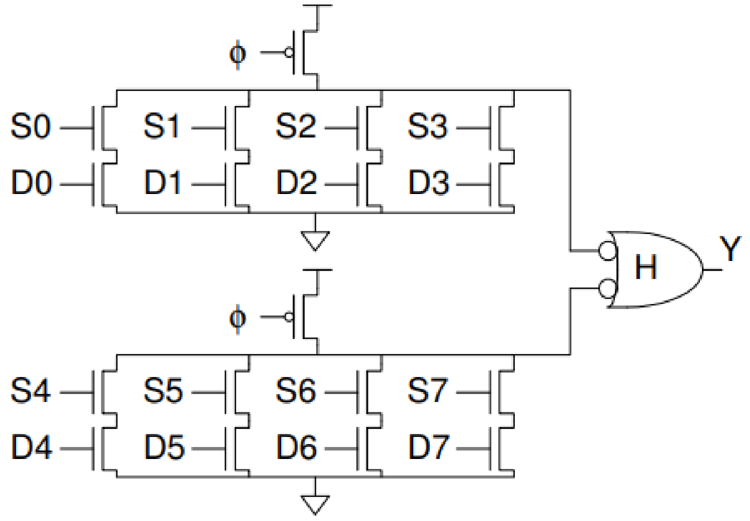
Domino Logic - Characteristics

- Only non-inverting logic
- Very fast - Only 1->0 transitions at input of inverter
- Precharging makes pull-up very fast
- Adding level restorer reduces leakage and charge redistribution problems
- Optimize inverter for fan-out

Domino Gates Optimization

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

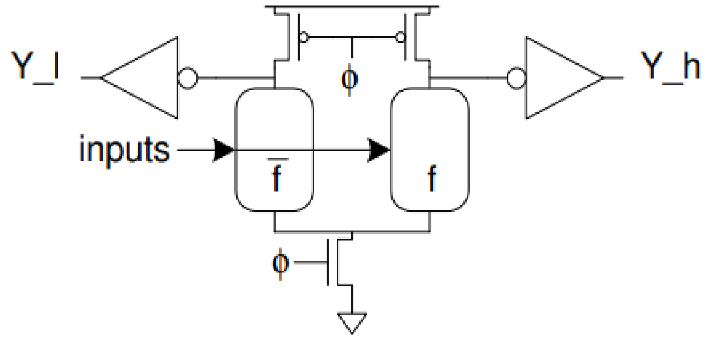


Dual Rail Domino

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

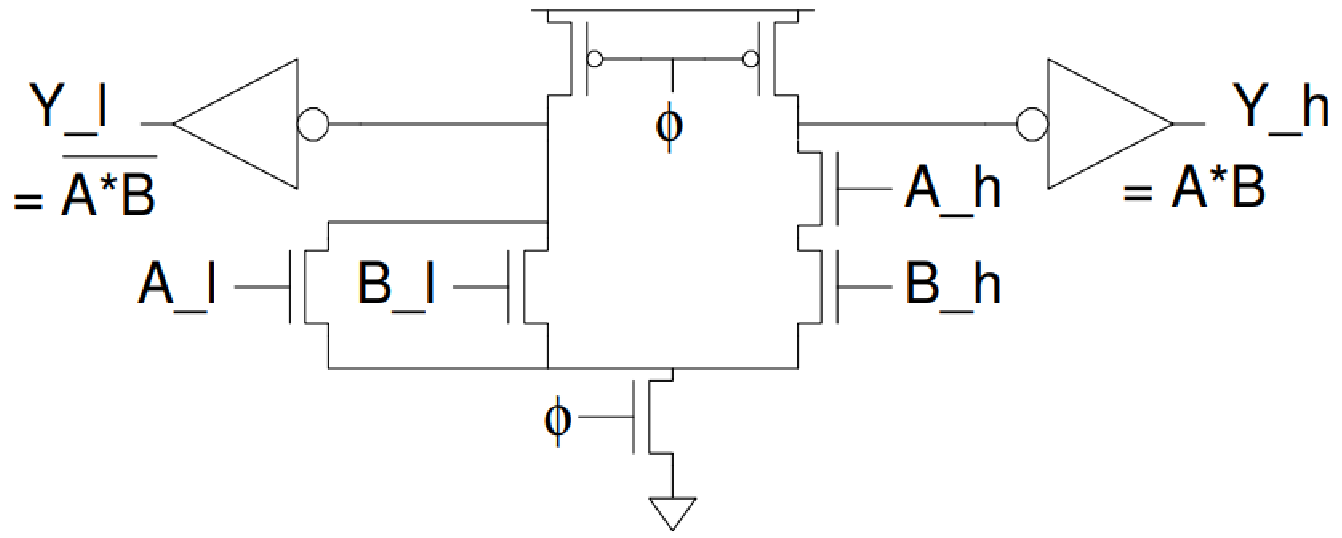
sig_h	sig_l	Meaning
0	0	Precharged
0	1	'0'
1	0	'1'
1	1	invalid



Dual Rail AND/NAND

Example: AND/NAND

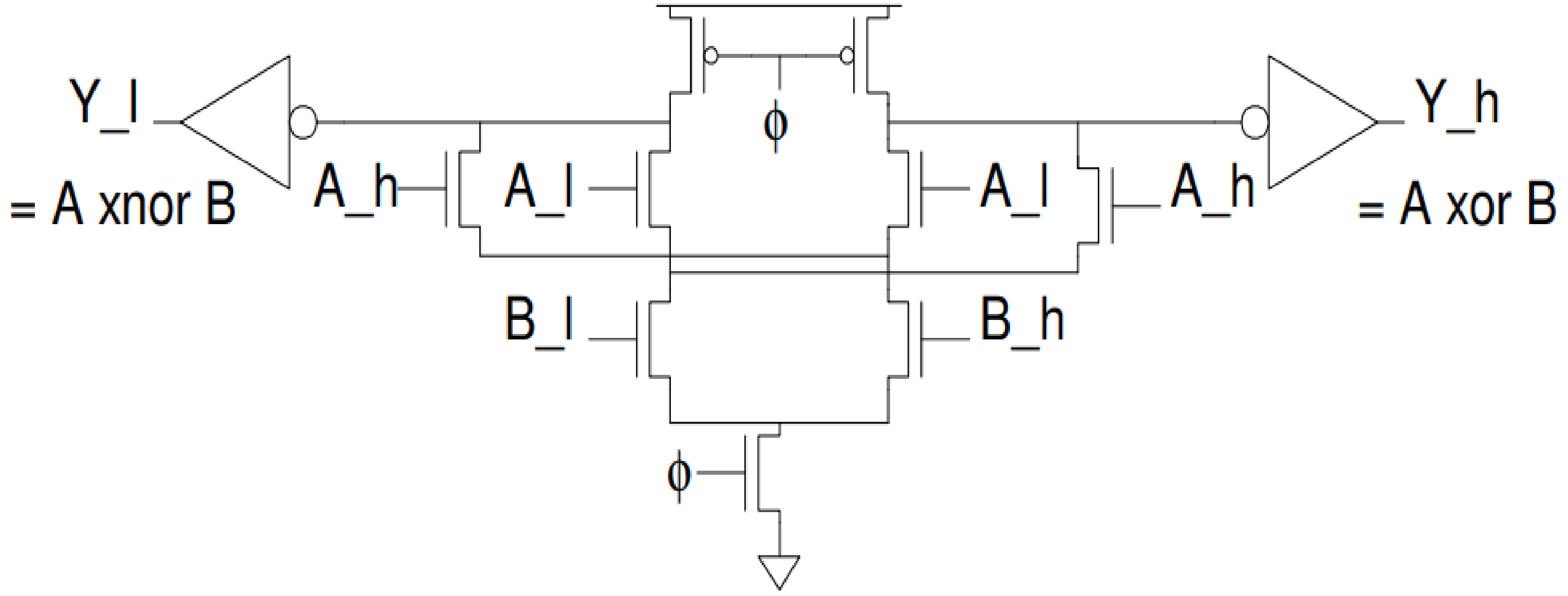
- Given A_h, A_l, B_h, B_l
- Compute Y_h = A * B, Y_l = ~(A * B)
- Pulldown networks are conduction complements



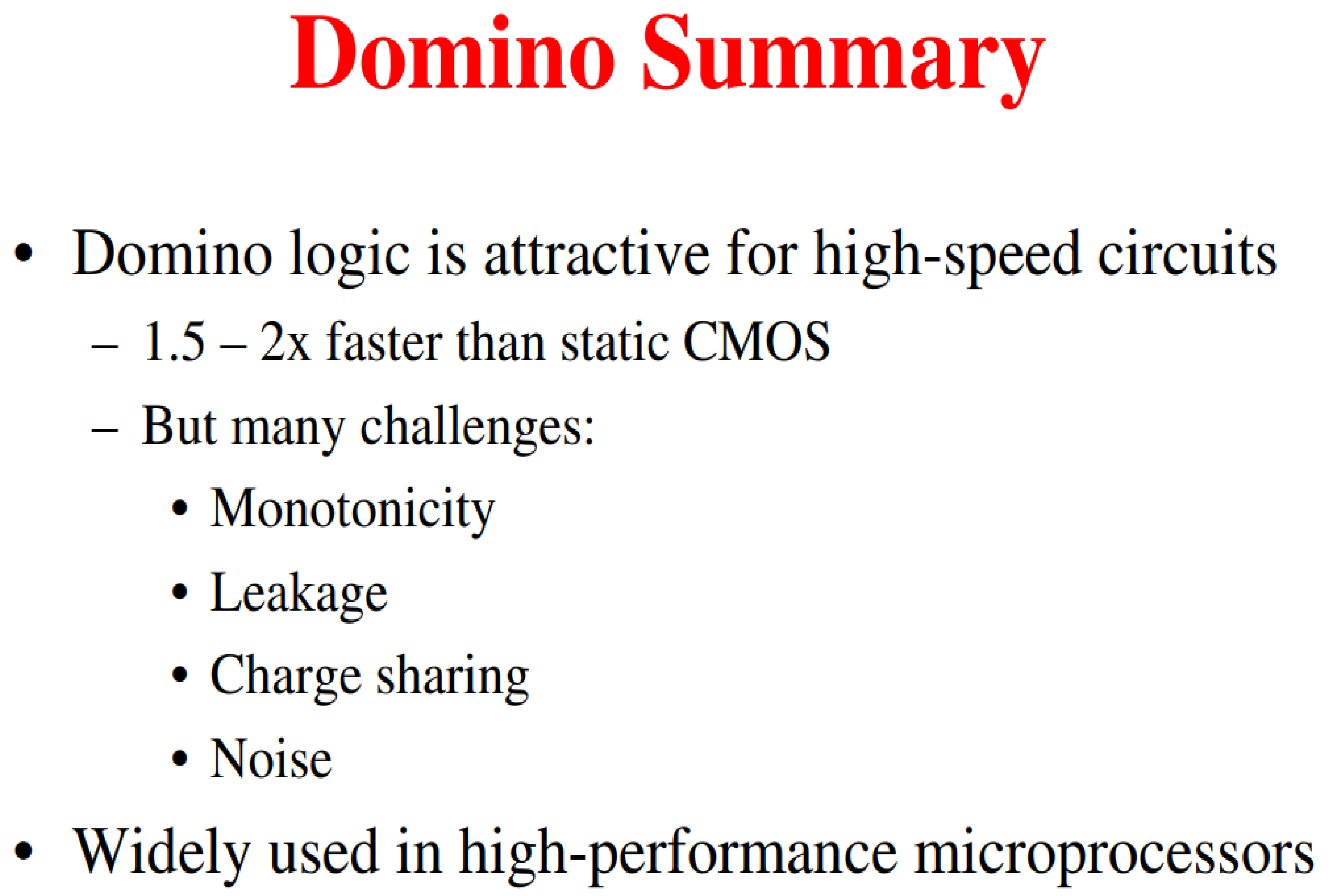
Dual Rail XOR / XNOR

Example: XOR/XNOR

- Sometimes possible to share transistors



Domino Characteristics Summary

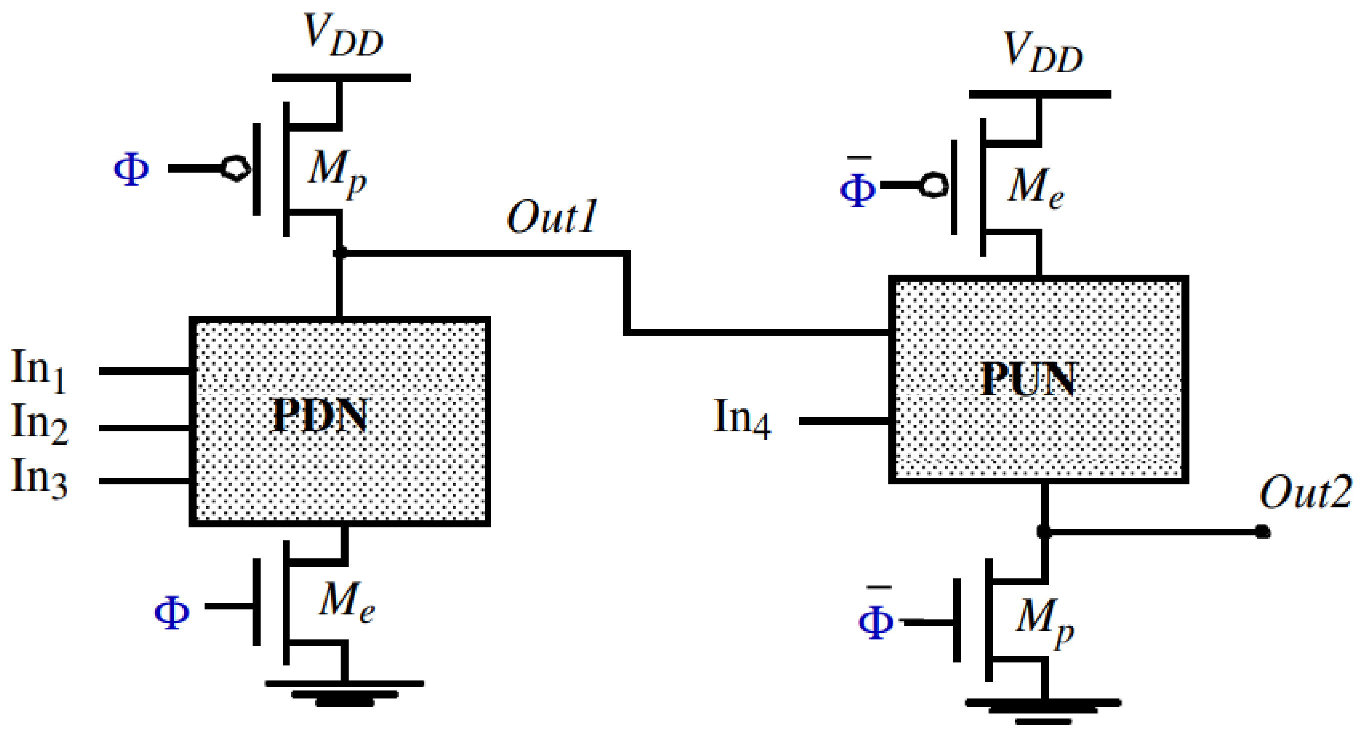


Domino Summary

- 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

NP Zipper CMOS

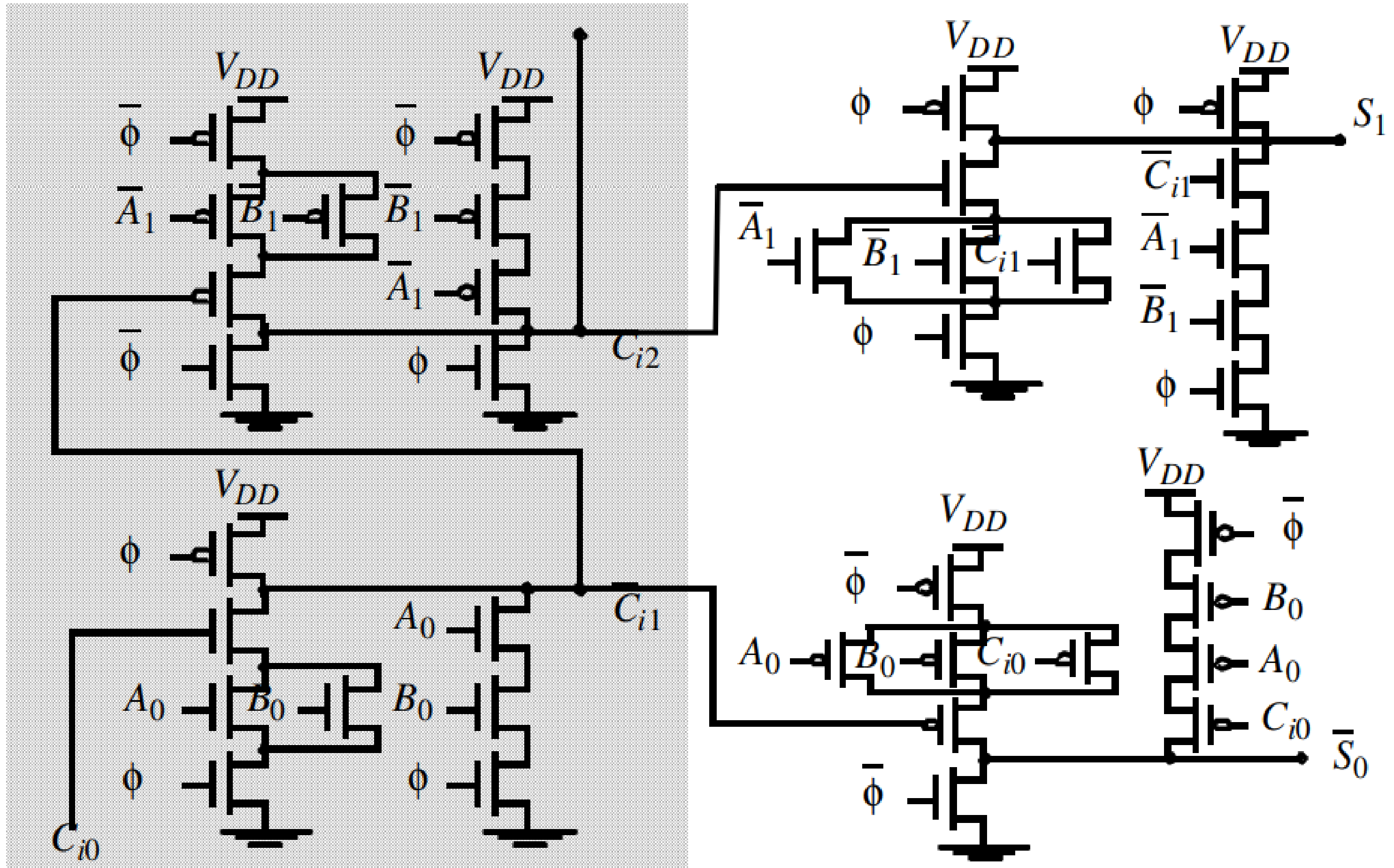
np-CMOS (Zipper CMOS)



- Only 1-0 transitions allowed at inputs of PUN
- Used a lot in the Alpha design

NP Zipper CMOS Adder

np CMOS Adder



Carry Path

Readings

- Combinational Circuit Families are discussed in Chapter 9 of course textbook CMOS VLSI Design by Weste and Harris, 4th Ed.