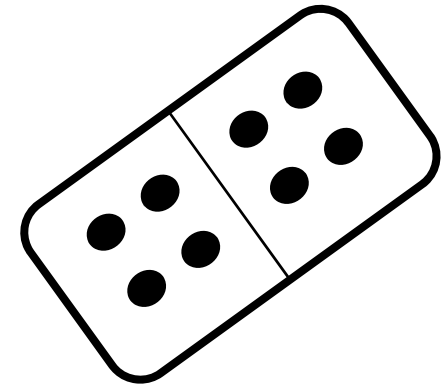
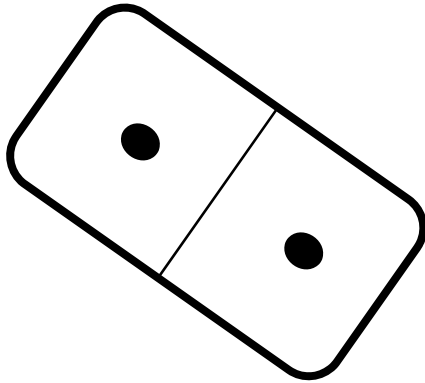


EE-382M

VLSI-II



Domino Circuit Design

Mark McDermott
Kevin Nowka, IBM

Outline

- **Derivation of Domino**
 - Motivation for domino circuits
 - Tradeoffs compared to static circuits
- **Textbook vs Industry Domino**
 - Use of static logic gates
 - Keepers
 - Footless gates and delayed clocks
 - Domino timing constraints
 - Set Dominant Latches (SDLs)
- **Summary**

Motivation for Domino - SPEED

- Performance of CMOS gates
 - Delay varies directly with output load
 - Delay varies inversely with device size
 - Sizing up: Current stage gets faster, previous stage gets slower
- Higher gain gates have a speed advantage
 - Gain = C_{out}/C_{in} for a particular delay or drive strength

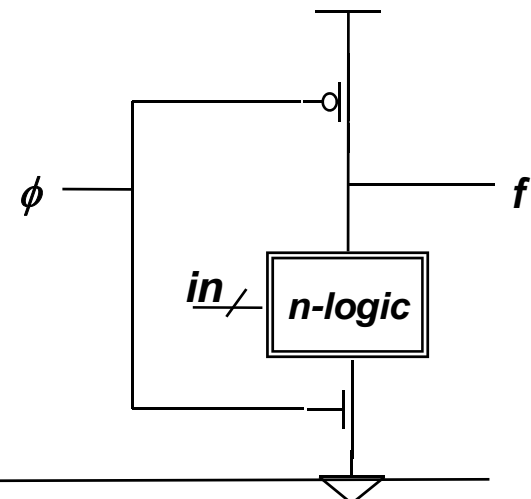
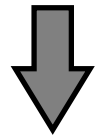
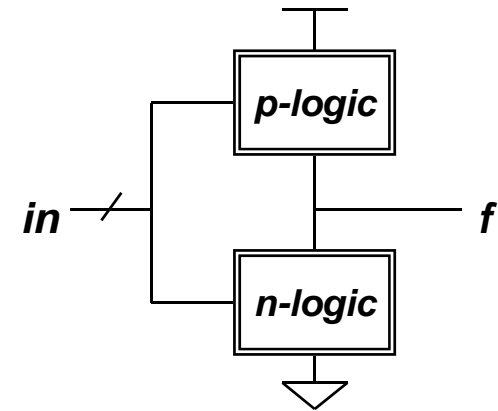
Basic precharge/discharge structures

Fully CMOS logic gates:

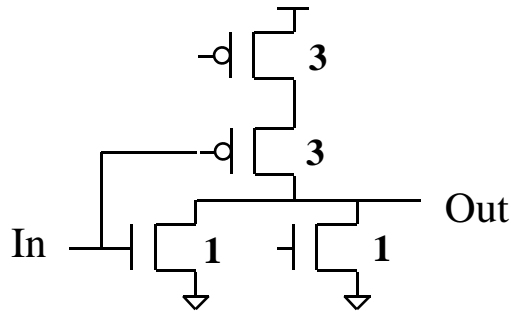
- ☹ Two logic blocks (N & P).
- ☹ Both up and down transitions could be critical.
- ☹ The input capacitance is that of both an N and a P device.
- ☹ The Pfets generally wider than NFETS (a.k.a. Beta ratio). Especially problematic when high stacks are present (then the area/speed/Cin penalty is big).

So, let's get rid of the p-logic block !

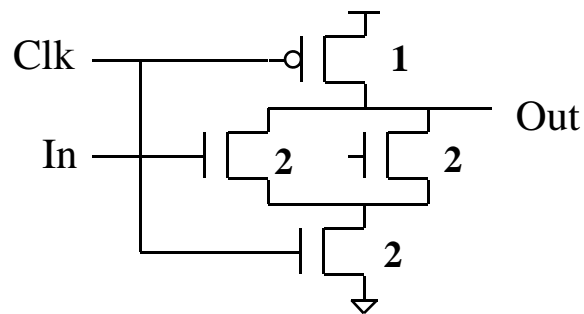
- ☺ A single logic block remains.
- ☺ A single transition should be optimized for the logic function evaluation!
- ☺ The input capacitance is only that of the N device!



Anatomy of a Domino Gate



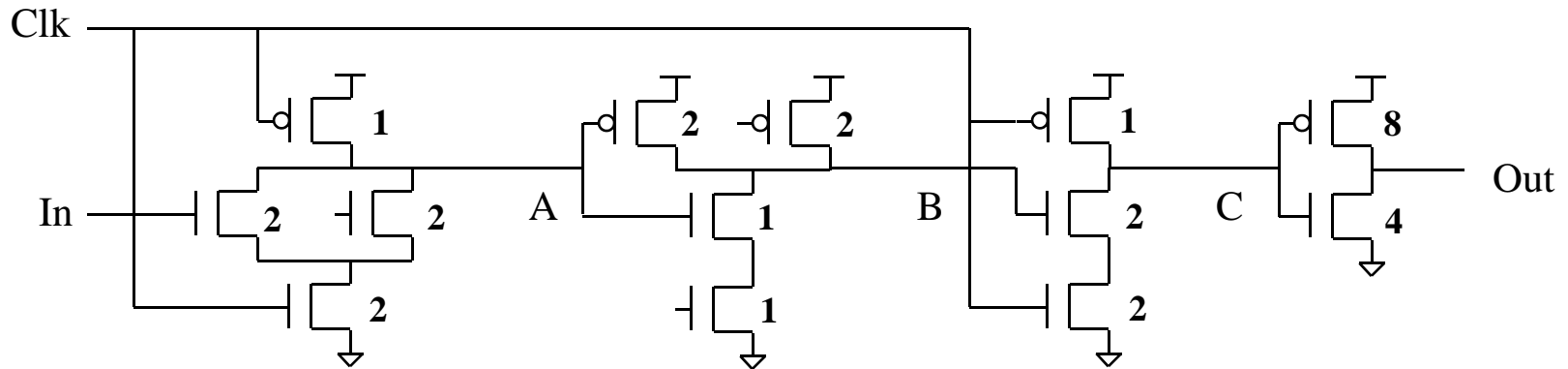
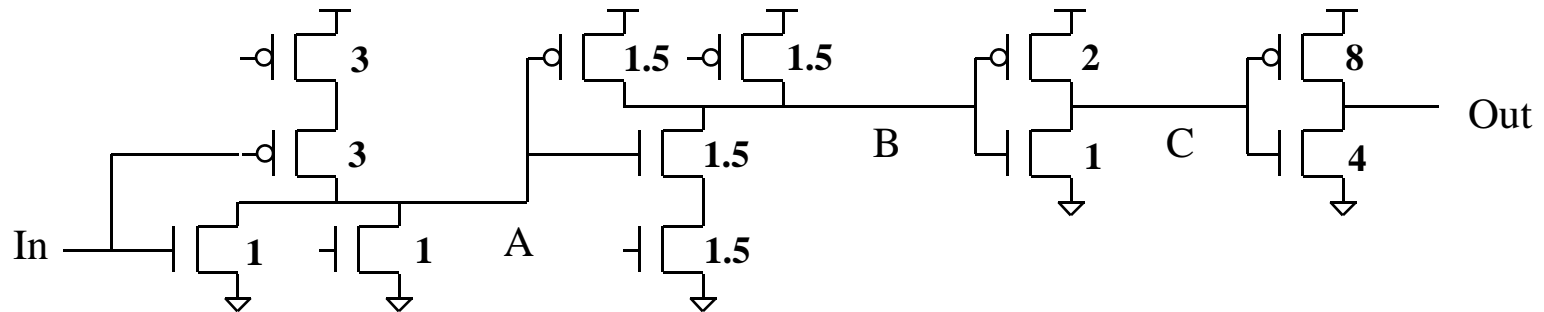
Static CMOS NOR



Domino NOR

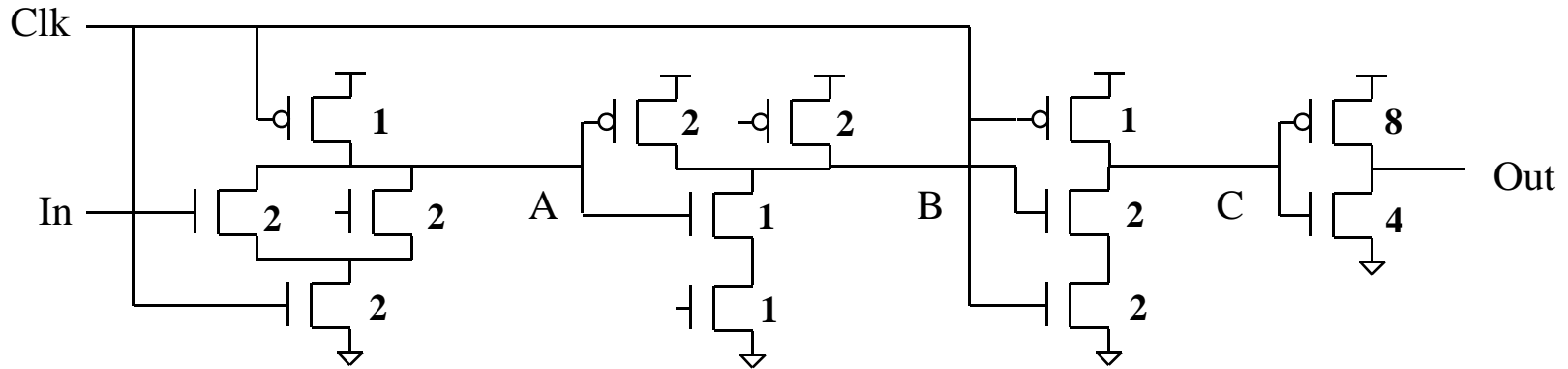
- Features of a simple domino gate
 - Single clocked P-FET
 - Clocked “foot” N-FET device
 - Only presents N device load to its inputs
 - N network is identical to static gate except for foot
- For the same pull-down strength, the domino gate gives approximately 2x the gain (in this case)
- Trip point of the gate is now just V_{tn} instead of based on P/N ratio

Static CMOS Path vs Domino Path

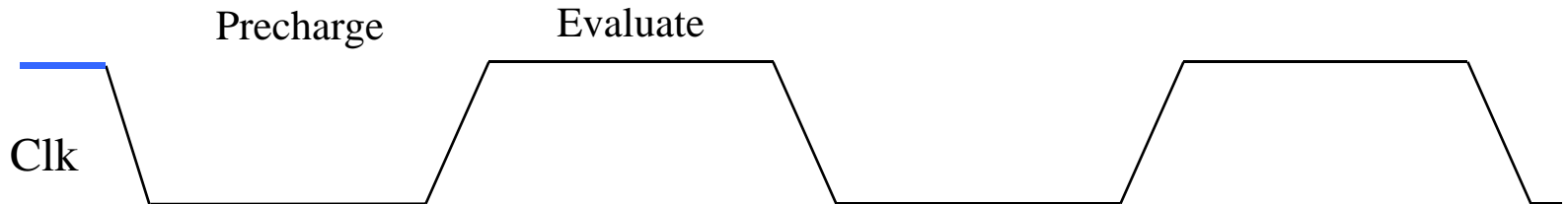


Domino performs the same logic as static CMOS with significantly less delay.

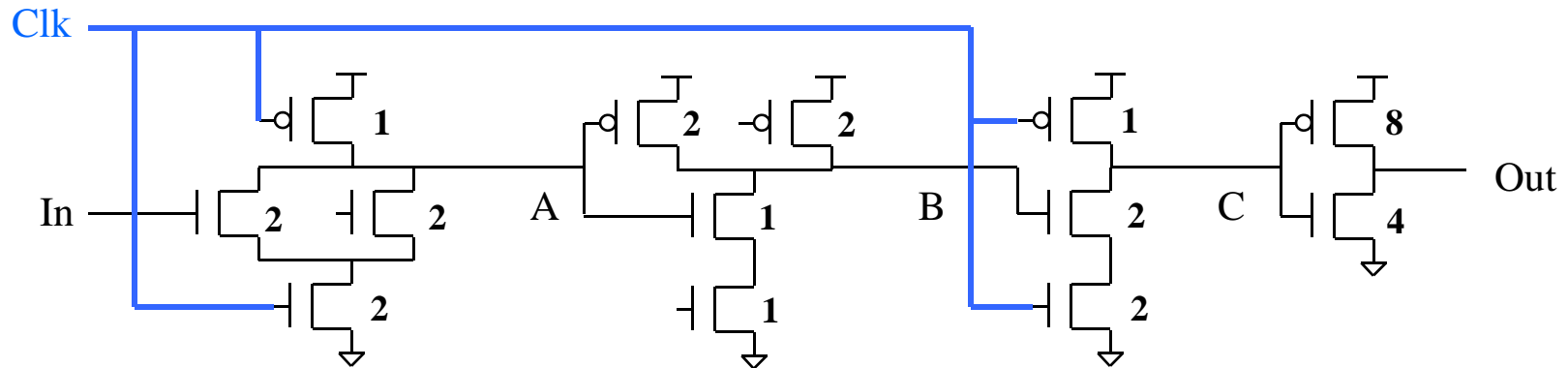
Domino timing



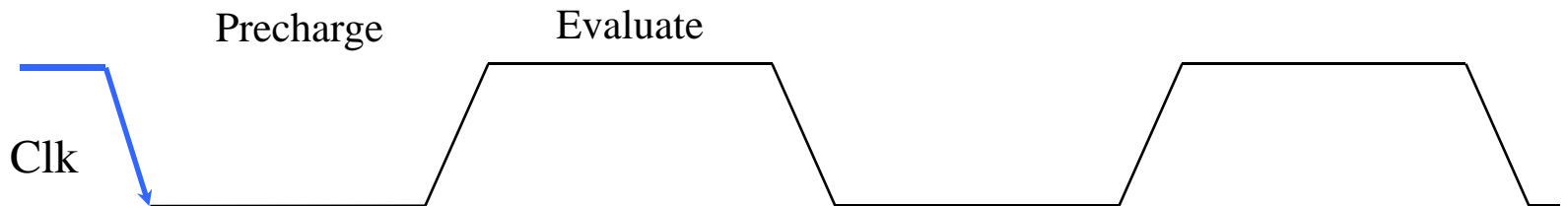
Domino Precharge



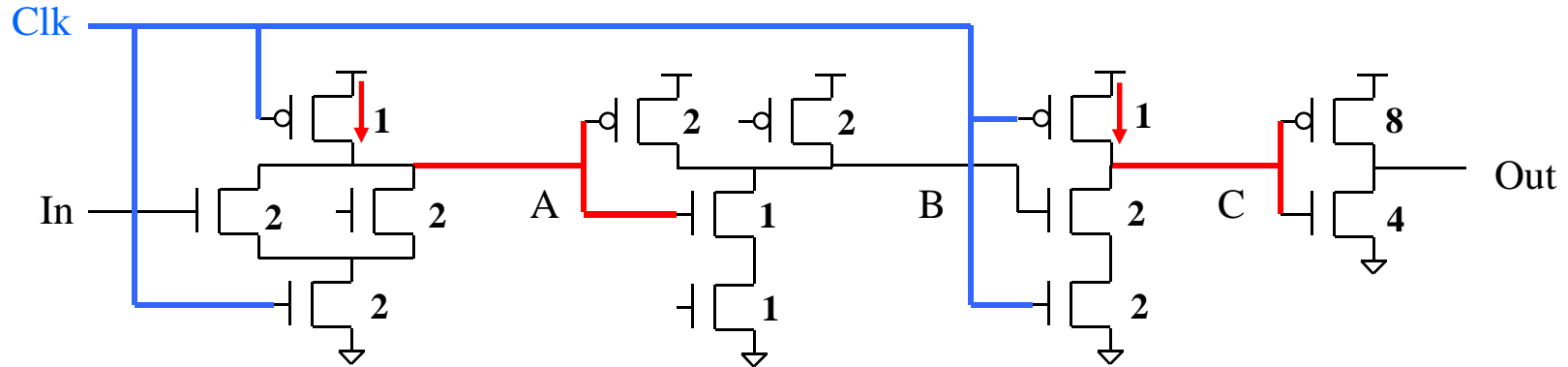
Domino timing



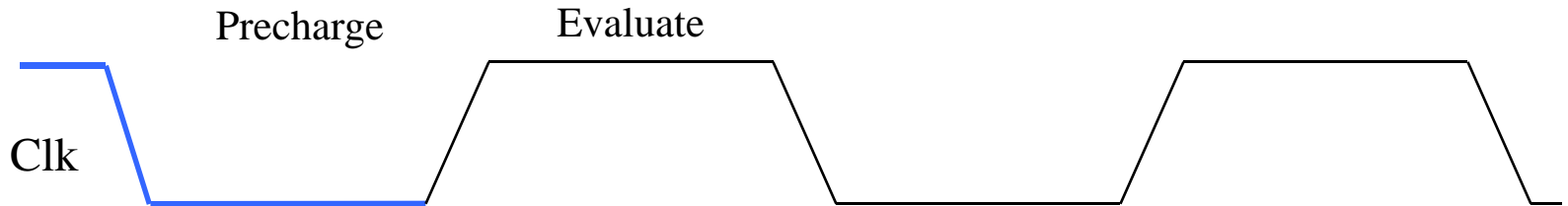
Domino Precharge



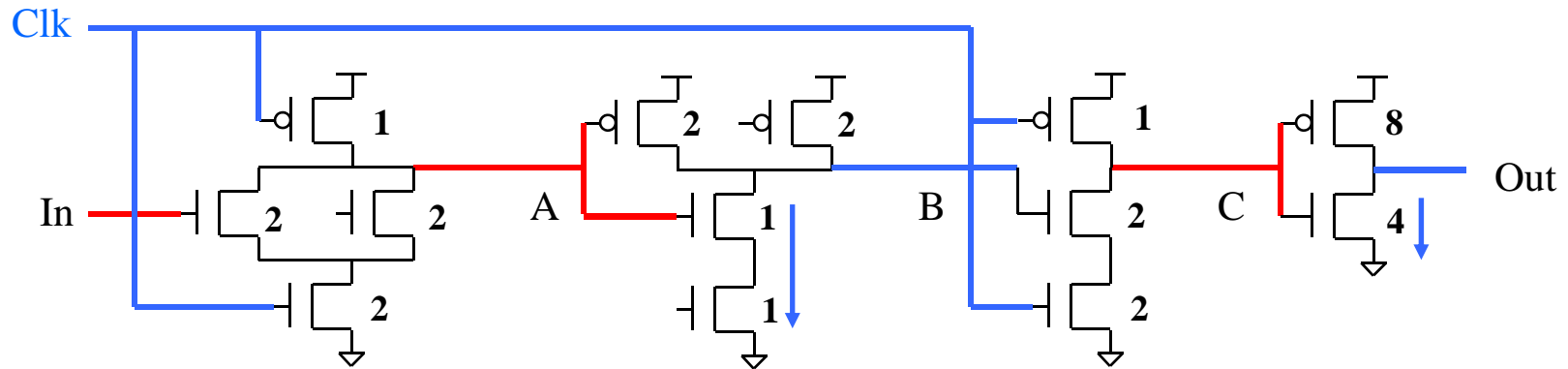
Domino timing



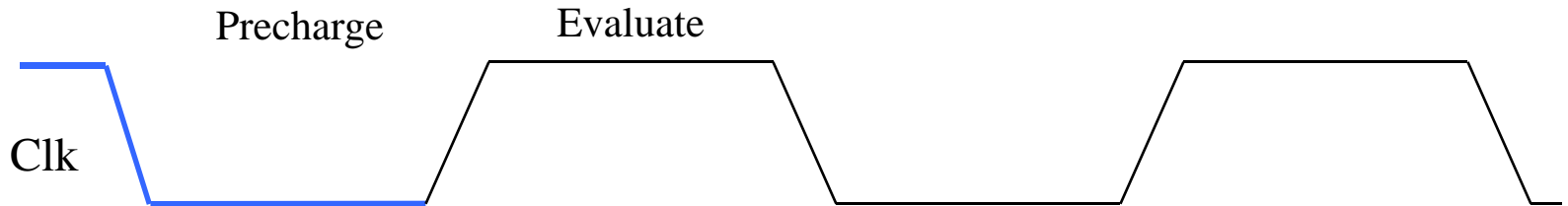
Domino Precharge



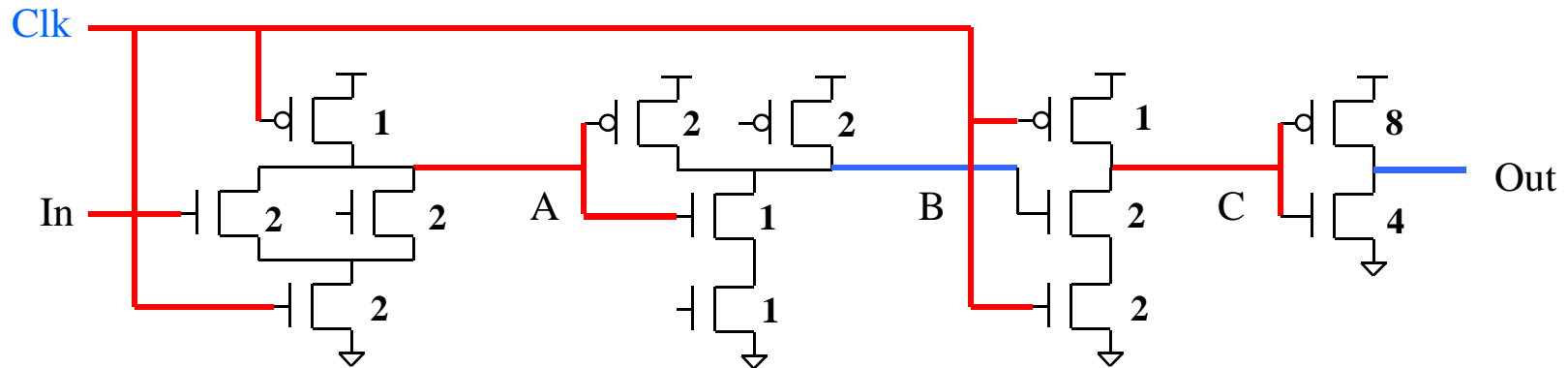
Domino timing



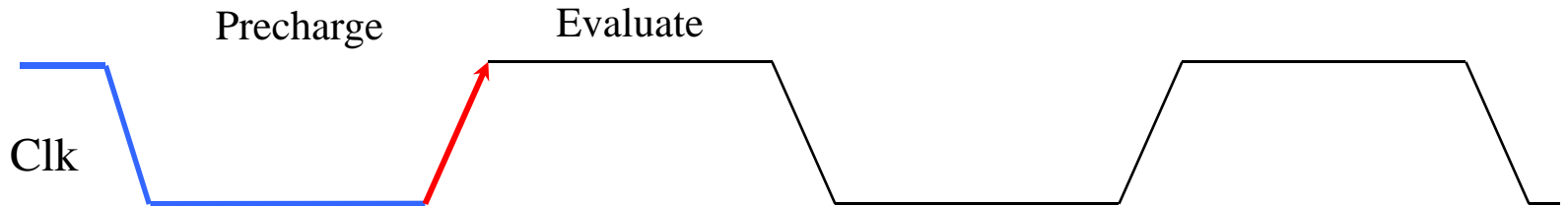
Domino Precharge



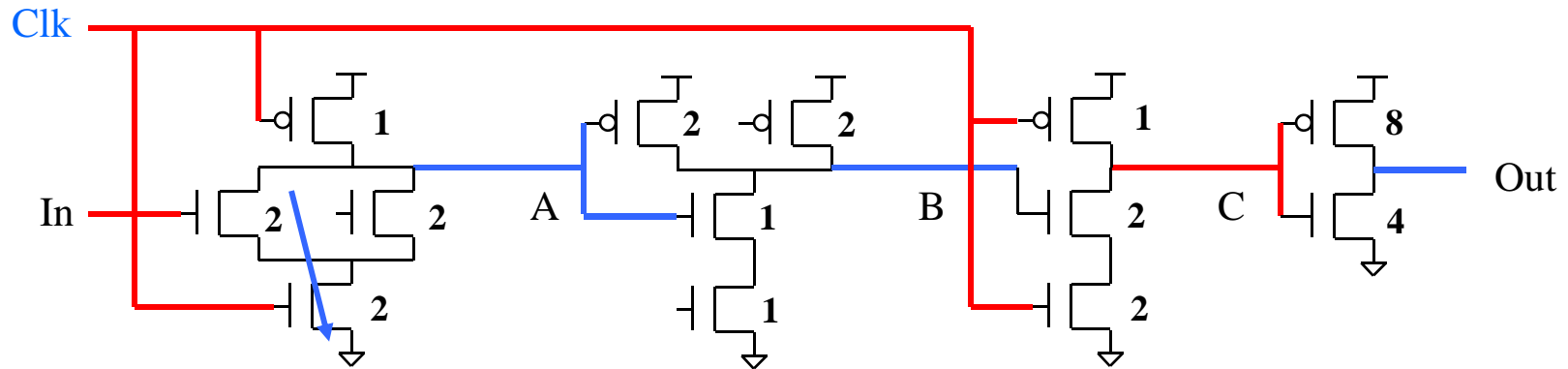
Domino timing



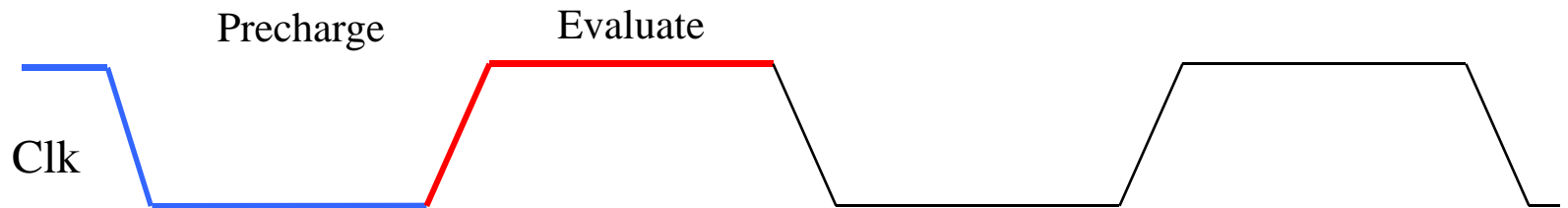
Domino Evaluate



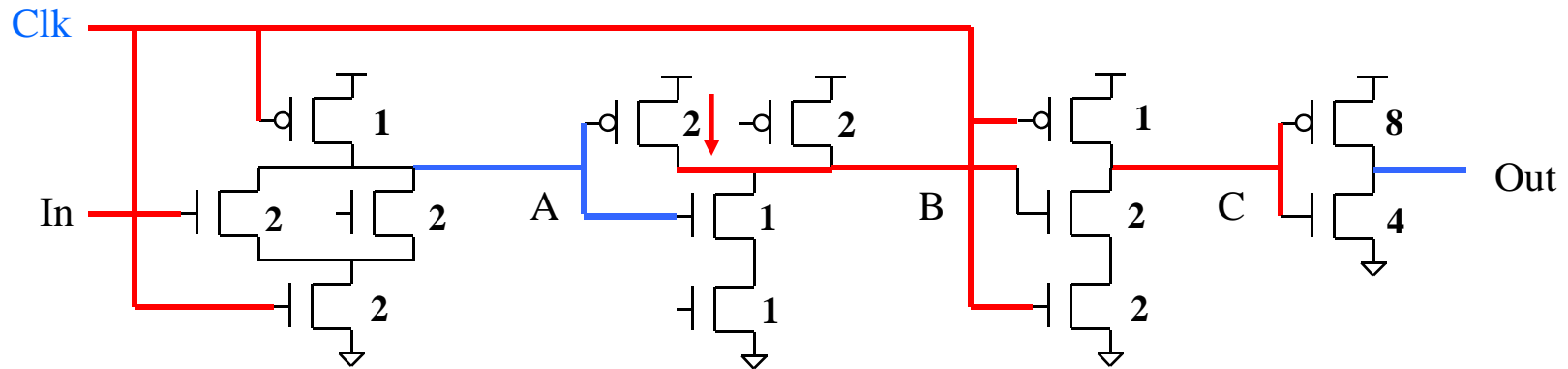
Domino timing



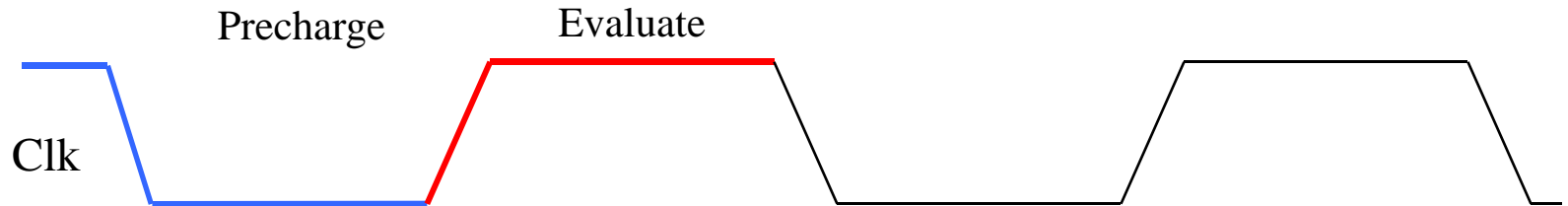
Domino Evaluate



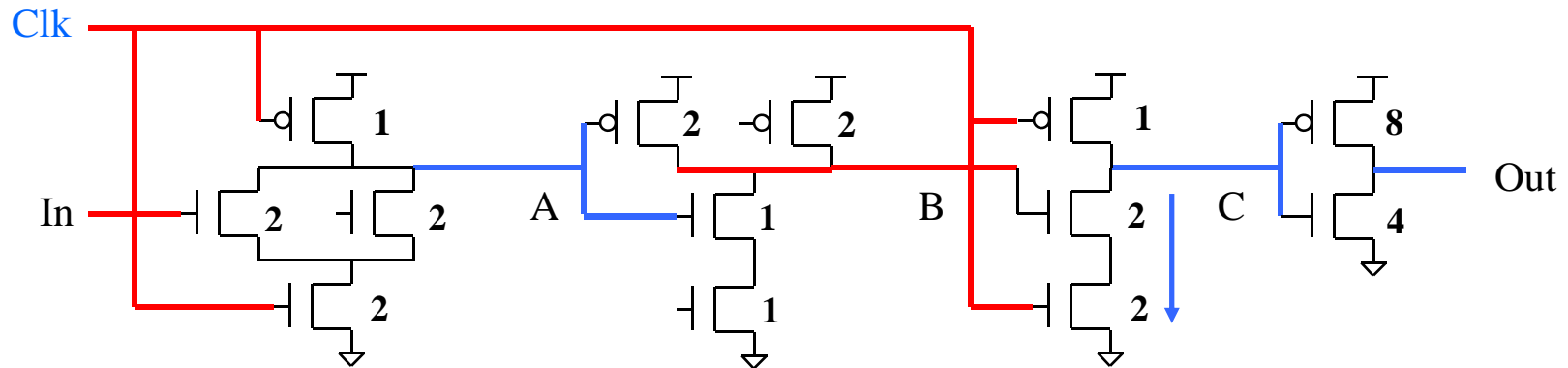
Domino timing



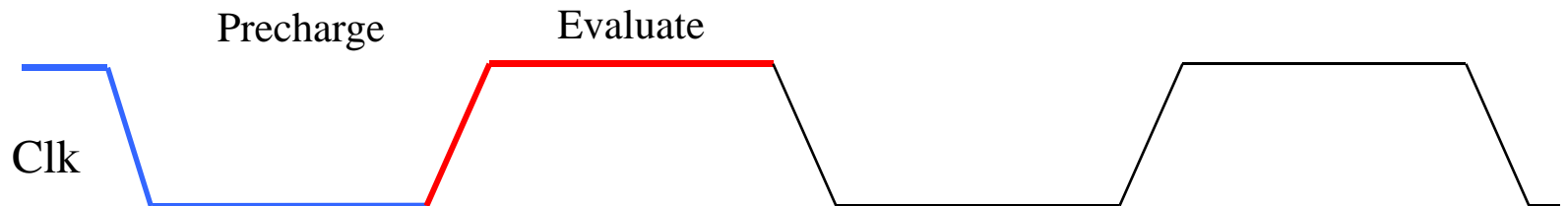
Domino Evaluate



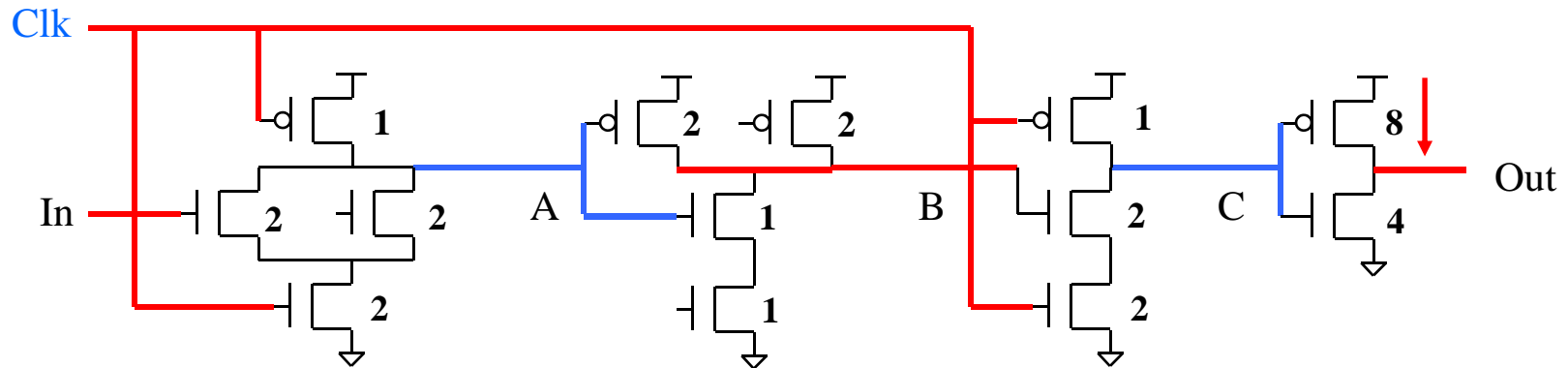
Domino timing



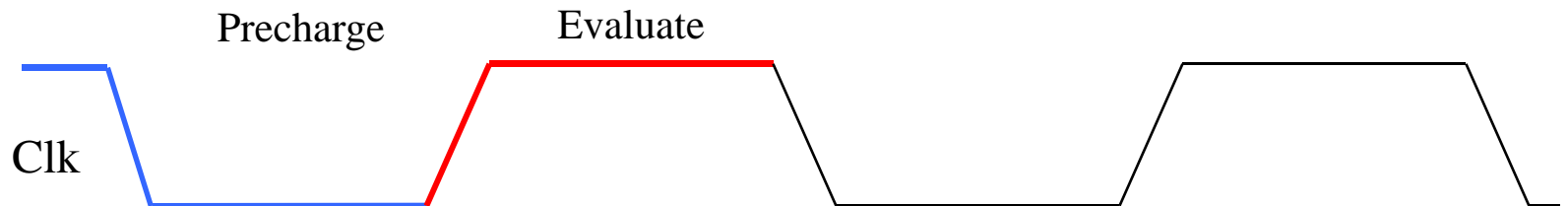
Domino Evaluate



Domino timing



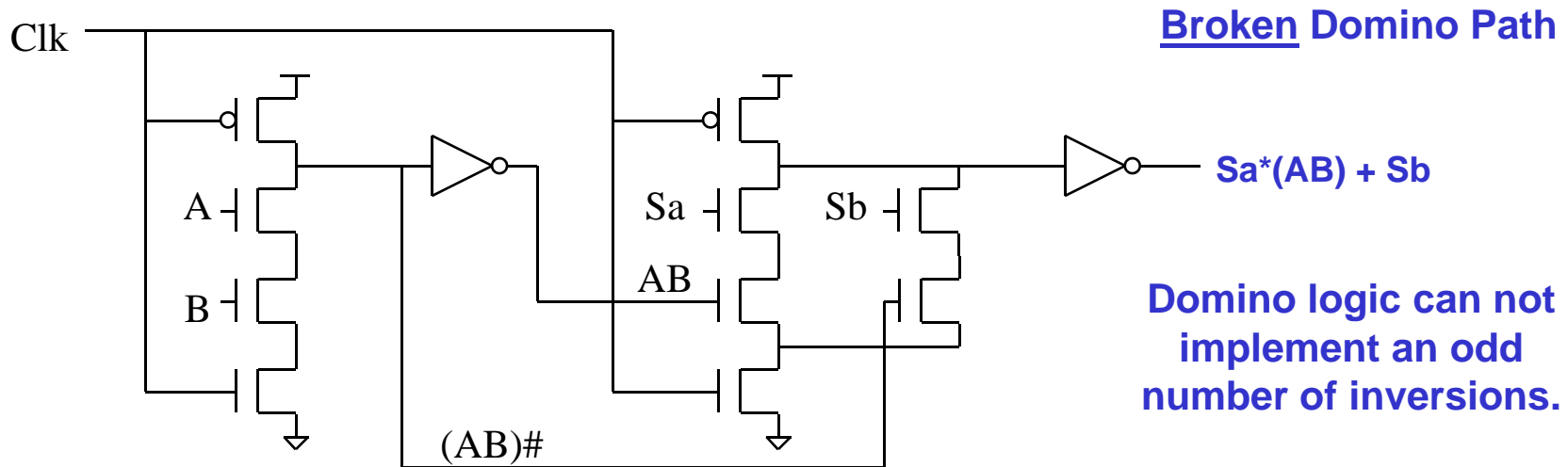
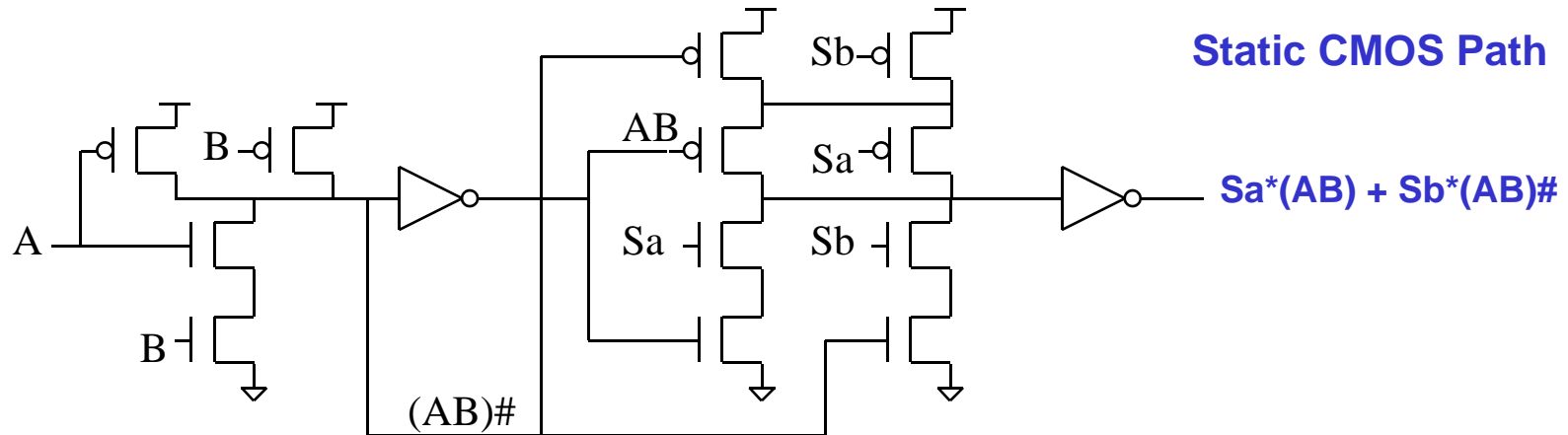
Domino Evaluate



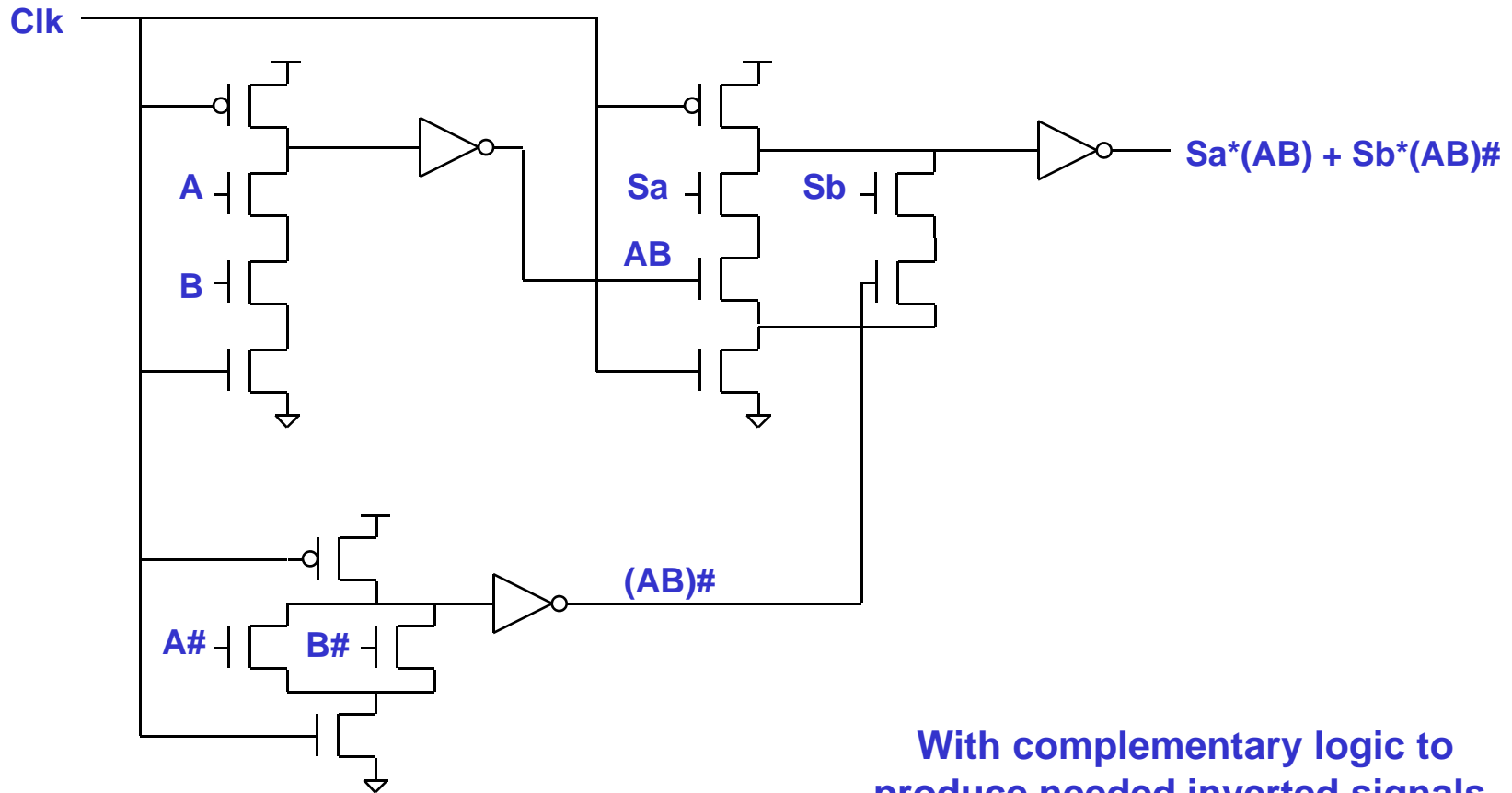
Domino Timing Notes

- During evaluate domino is sensitive to “up glitches”
 - No recovery from incorrect discharge
 - “Hard” setup requirement for falling input edges
 - Hopefully, the falling edge is not critical
 - “Soft” setup requirement for rising input edges
 - Rising edges may be late provided they still trip the gate
- Domino chains must follow strict domino-static structure
 - Every even stage must be static.
 - Can’t have an odd number of stages to a domino input.
 - Why not?

The Problem of Inverting Logic



Dual Rail Domino



With complementary logic to produce needed inverted signals, domino can implement any function.

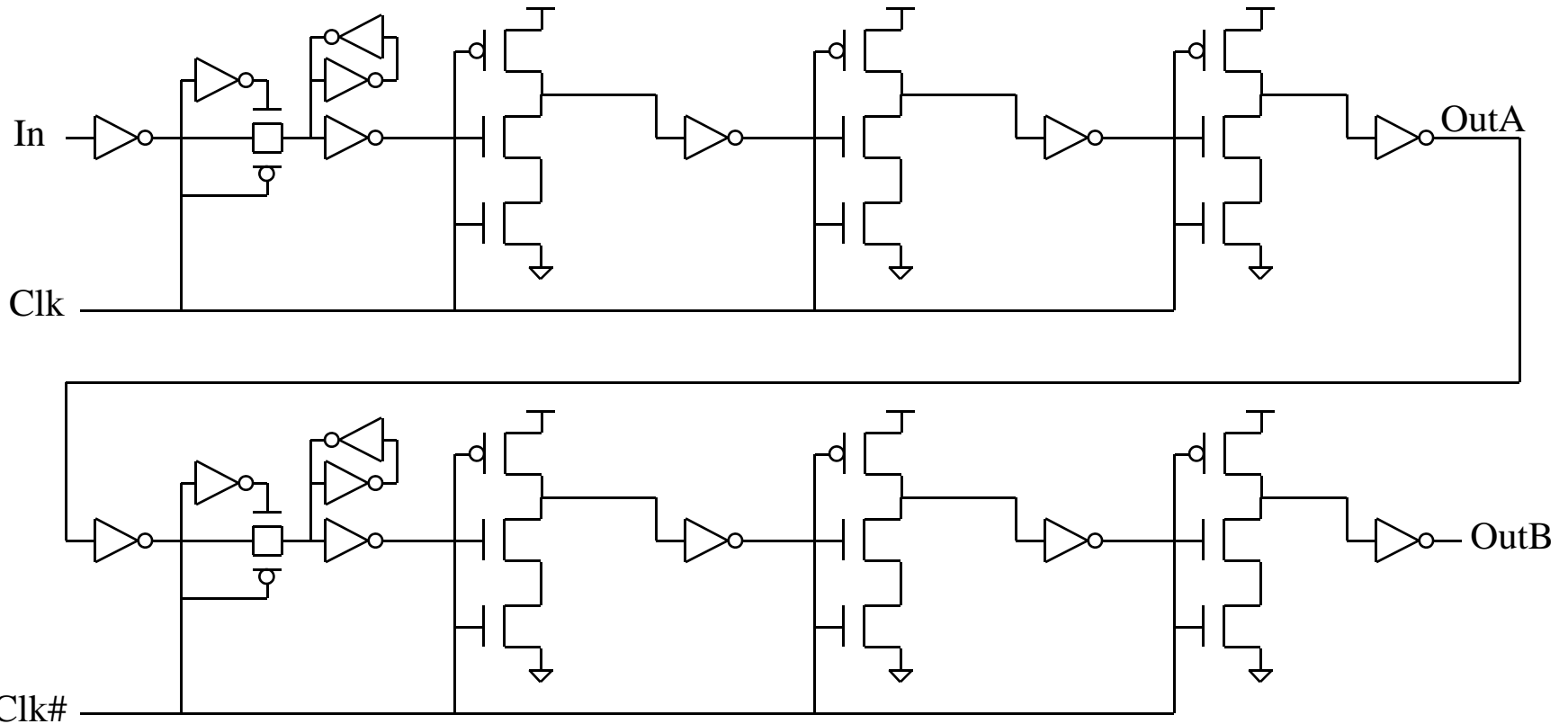
Tradeoffs of Domino

- Faster Gates
 - Allows more logic per cycle with less delay
 - Allows for more complex gates (no dual P network)
- Increased Area
 - Dual rail domino can double the number of gates
- Increased Power
 - Precharging increases activity factors
 - Increased clock load
- Increased Noise Sensitivity
 - High gain means low noise immunity
 - Charge sharing in complex gates
- Increased Design Time
 - Added timing checks
 - CAD tools less automated

Domino typically only used in the most timing critical paths.

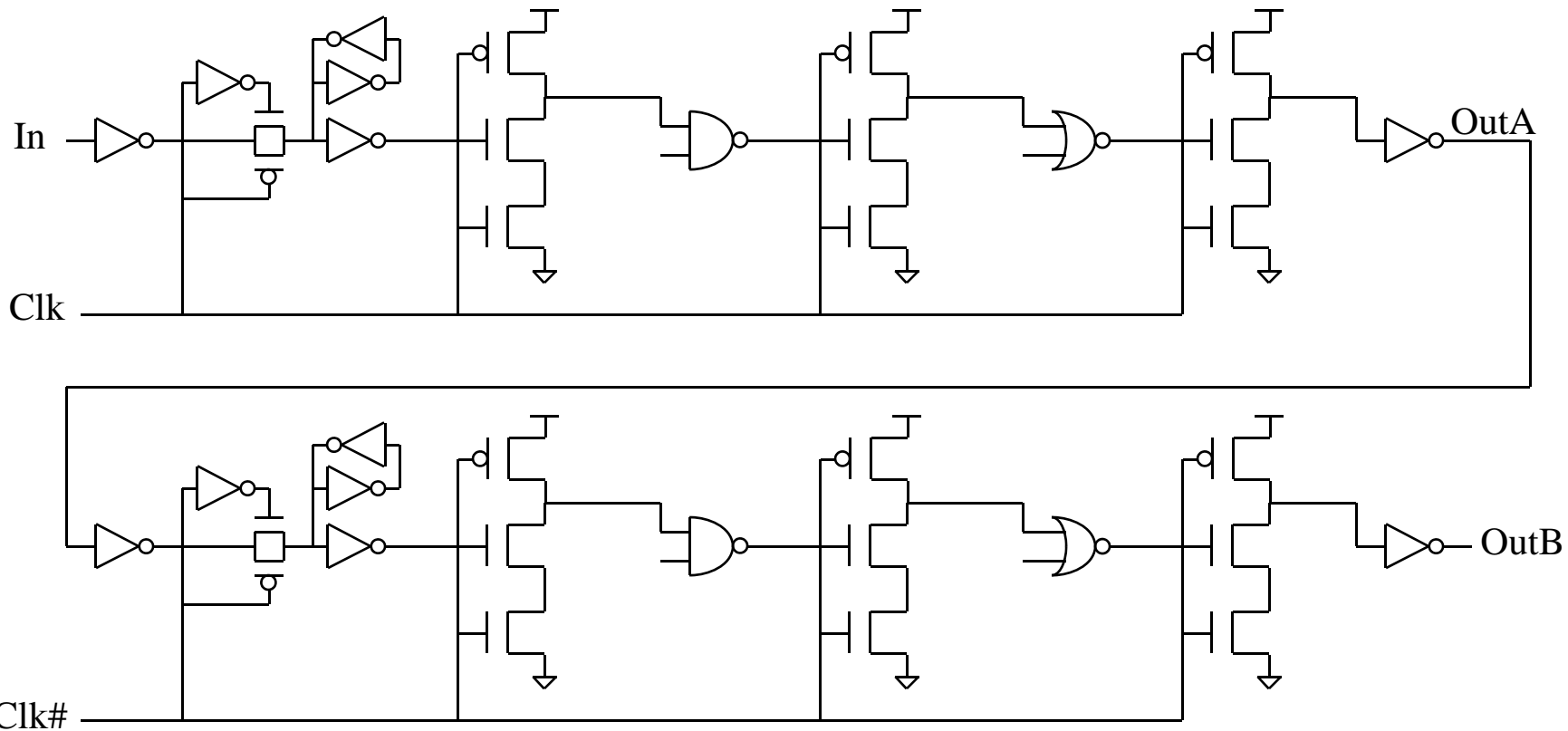
Textbook vs Industry Domino

Textbook Domino Cycle



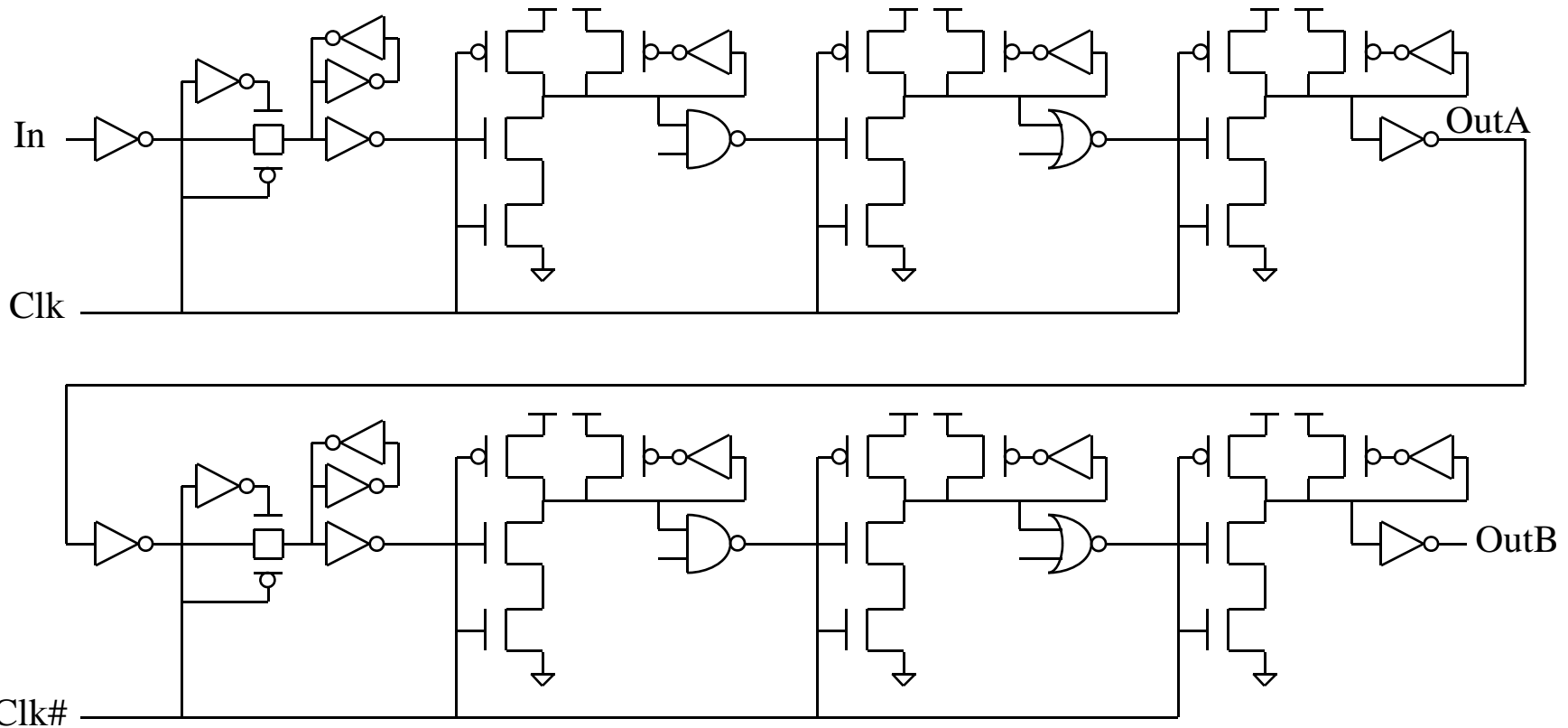
In two phase domino design half of logic precharges while other half evaluates.

Addition #1: Static Gates Used for Logic (a.k.a. Compound Domino)



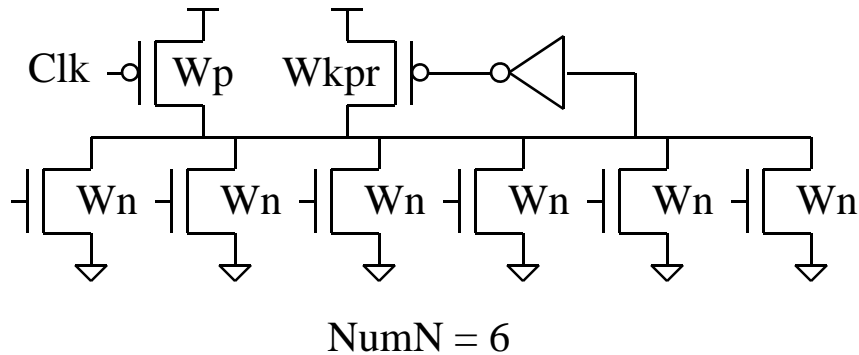
Any inverting static gate may be used between domino gates.

Addition #2: Add Half Latches (domino no longer dynamic)



P-keepers hold domino nodes high during stop clock and improve noise margins.

Sizing Keepers



$$\text{NumN} \cdot W_n / 20 < W_{kpr}$$

Min keeper size set by
noise requirements

$$W_{kpr} < W_n / 2$$

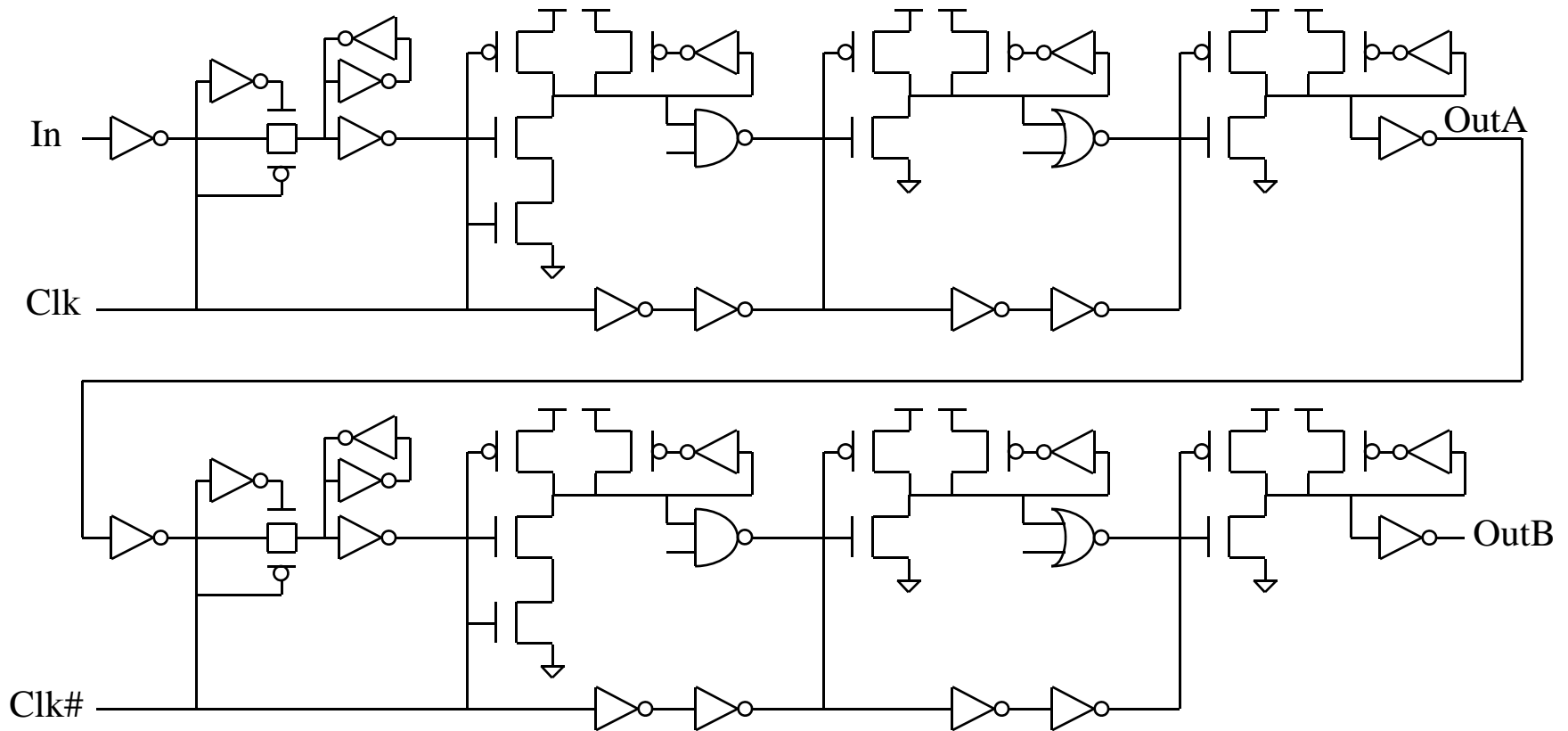
Max keeper size set by
delay degradation

$$\text{NumN} \cdot W_n / 20 < W_{kpr} < W_n / 2$$

$$\text{NumN} < 10$$

Maximum noise sensitivity and delay limit number of parallel stacks.

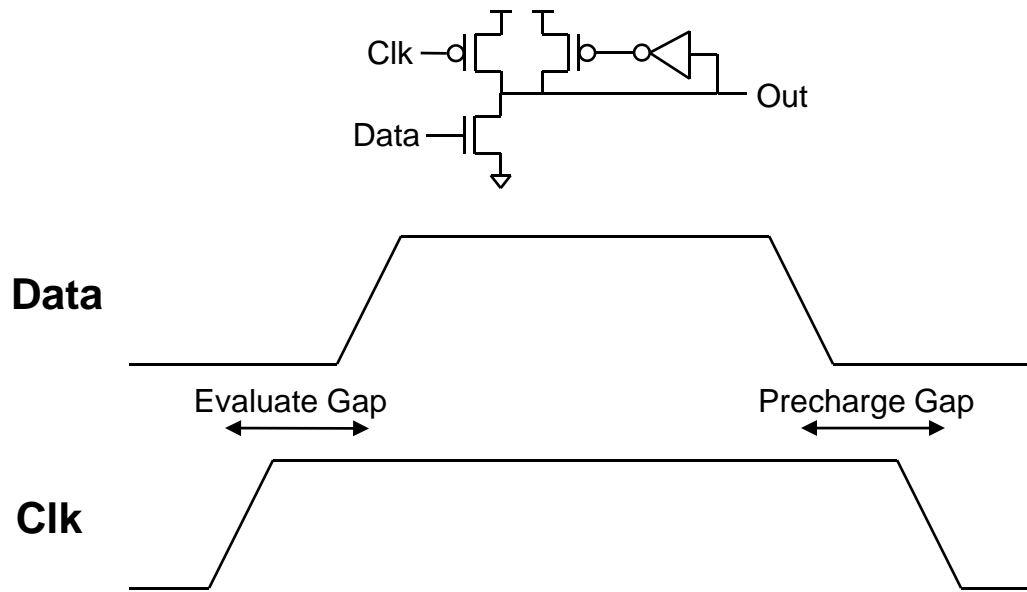
Addition #3: Add Footless Domino Gates (a.k.a. unfooted delayed reset)



Footless domino gates improve speed and allow more logic per gate.

Power Races

In footless domino gates Clk must be high when Data is high.



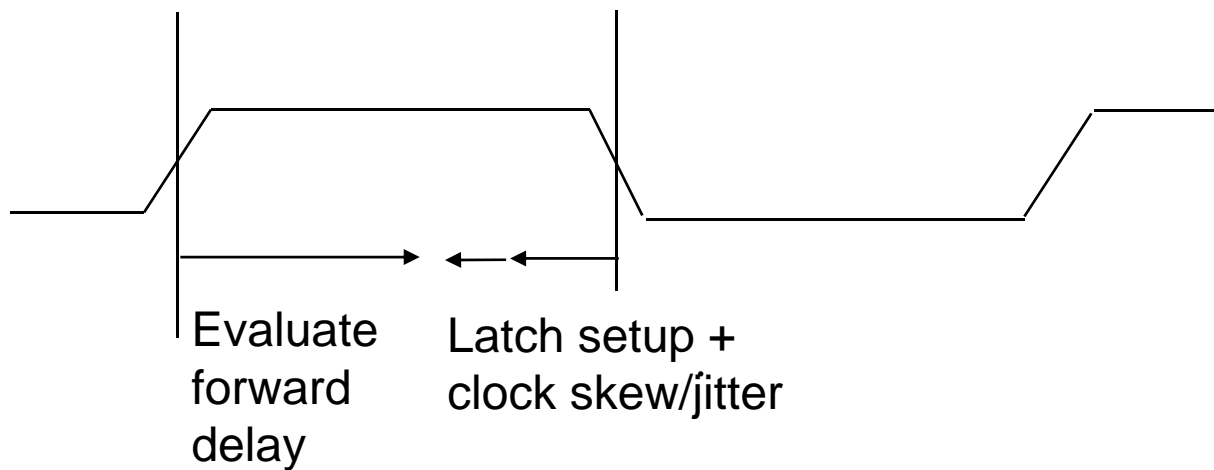
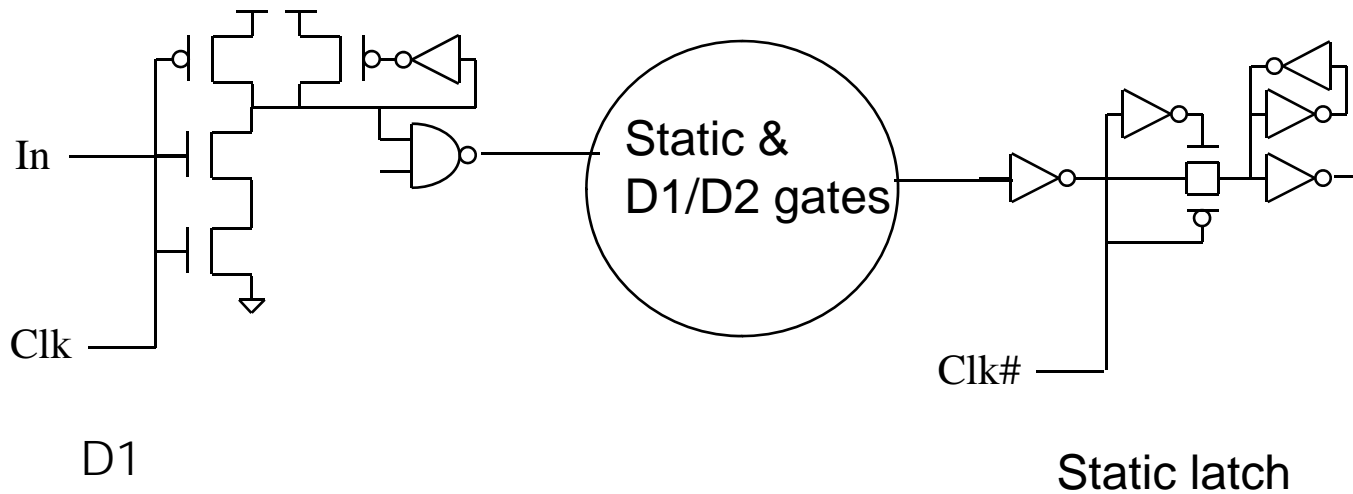
Negative Evaluate Gap:

- More Power
- Evaluate Delay Push Out

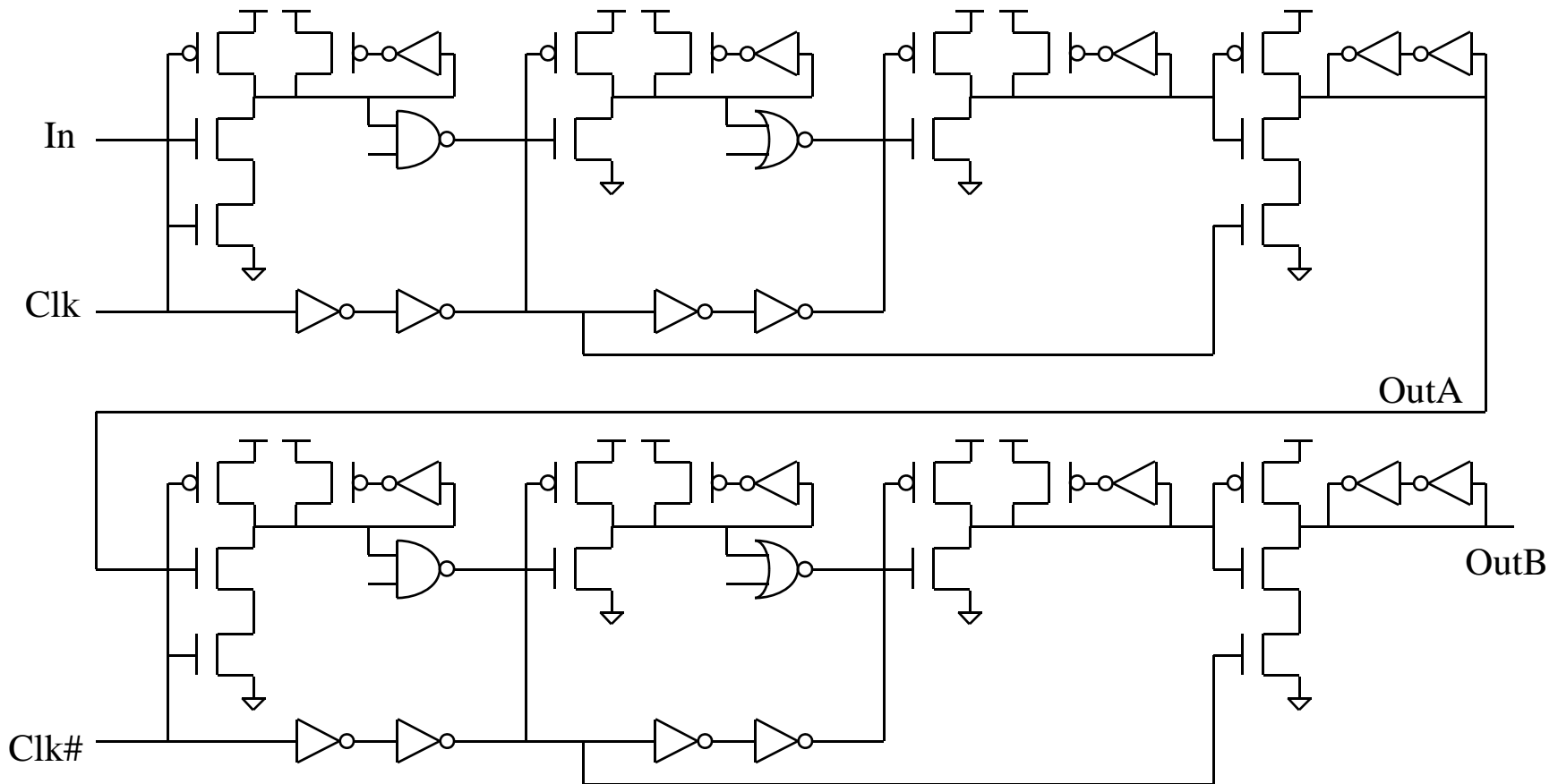
Negative Precharge Gap:

- More Power
- Precharge Delay Push Out

Domino max timing constraints with static latches



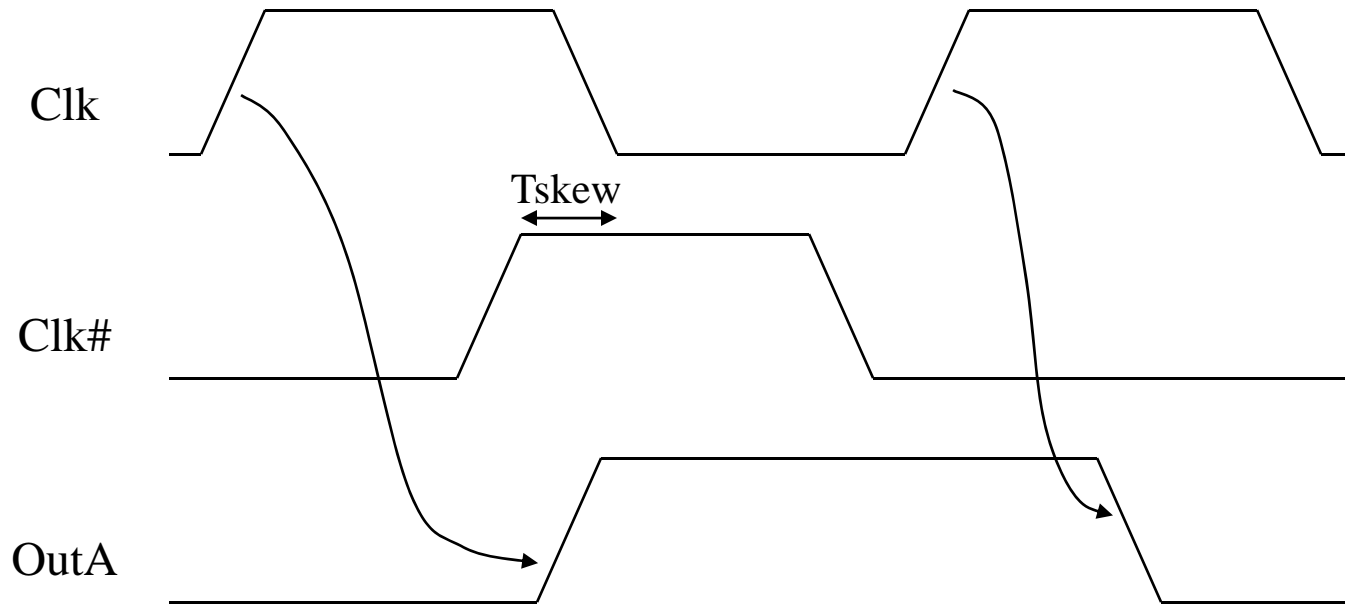
Addition #5: Add Set Dominant Latches (SDLs)



SDLs convert domino signals into more static-like behavior

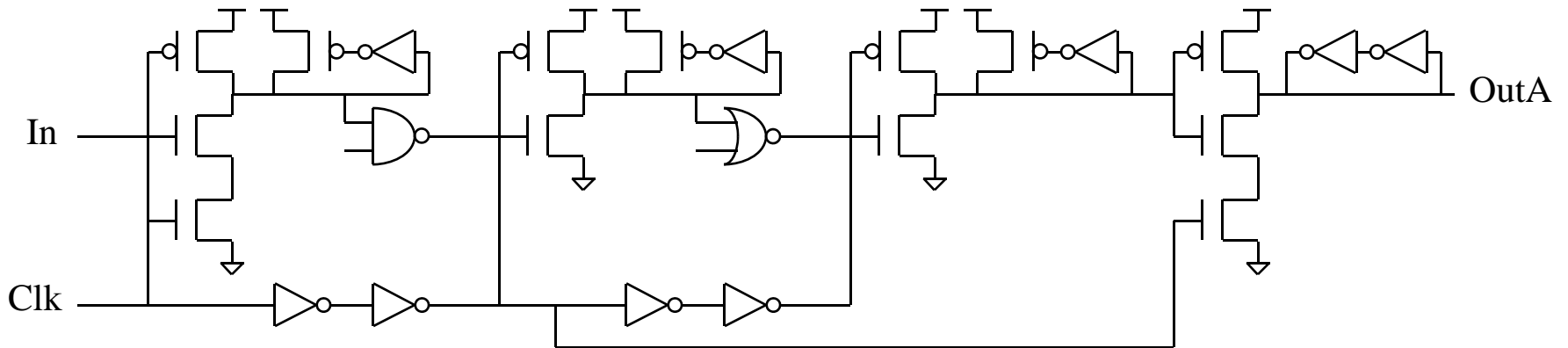
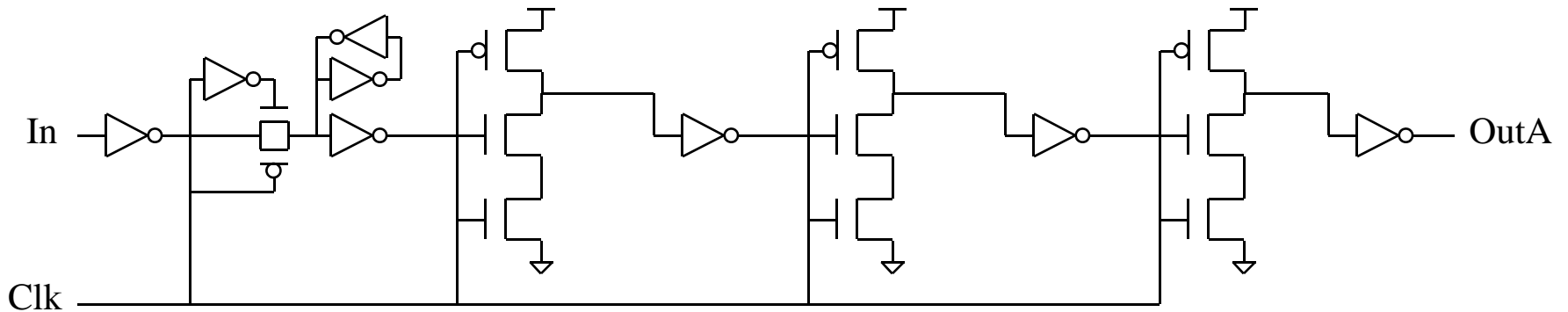
SDL Timing

With SDLs precharge transition is delayed until start of next evaluate.



Overlap of Clk# and OutA is now frequency dependent.

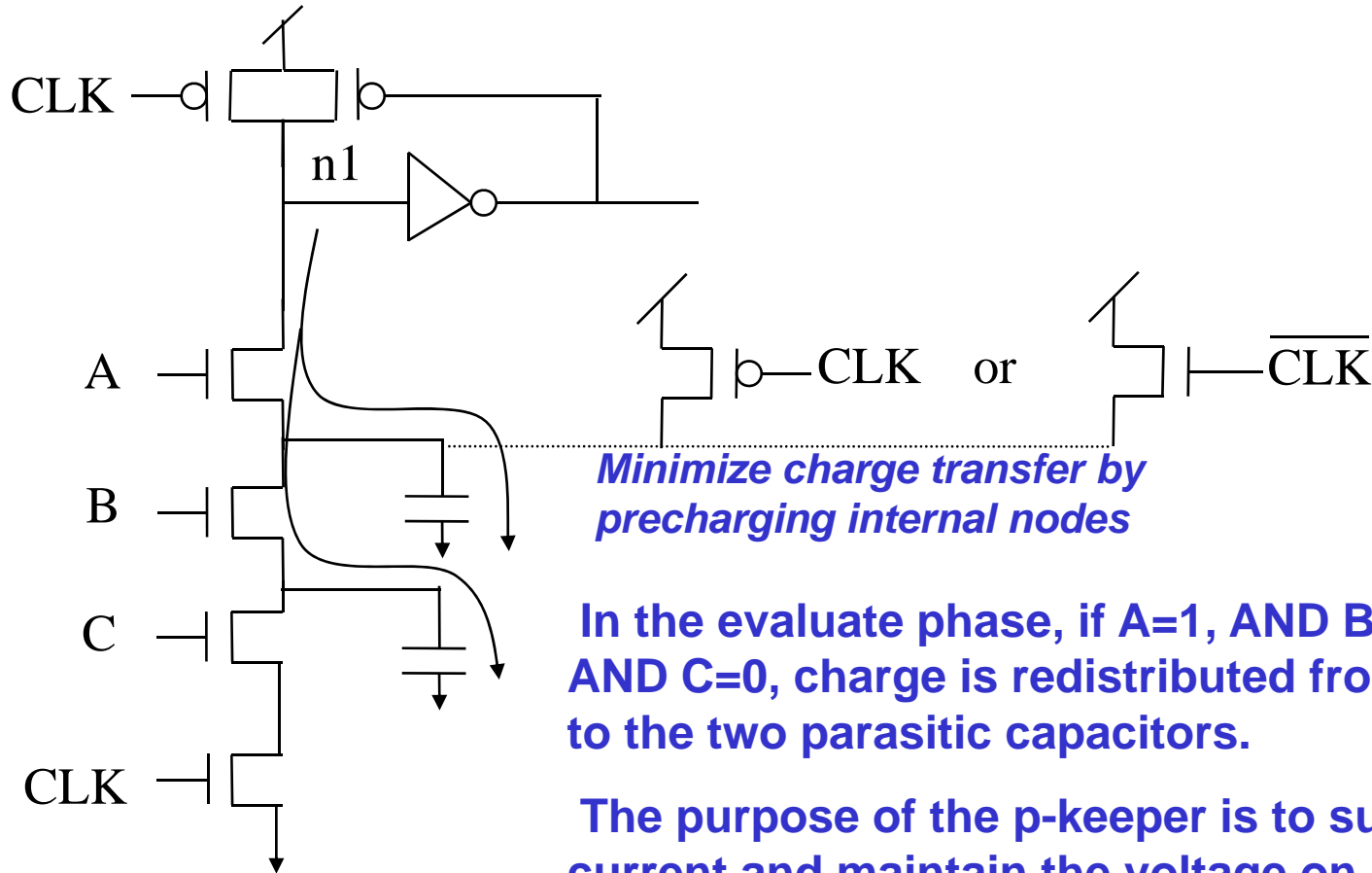
Textbook vs Industry Domino



Domino Circuit Summary

- **Domino reduces delay by favoring one transition and making the other non-critical by construction**
- **The price of this speed is generally:**
 - **Greater Noise Sensitivity**
 - **More Power**
 - **More Design Time**
- **Industry domino circuits make use of:**
 - **Different static logic gates**
 - **Keepers**
 - **Footless domino gates**
 - **Transparency**
 - **SDLs**

Noise: Leakage and Charge Sharing



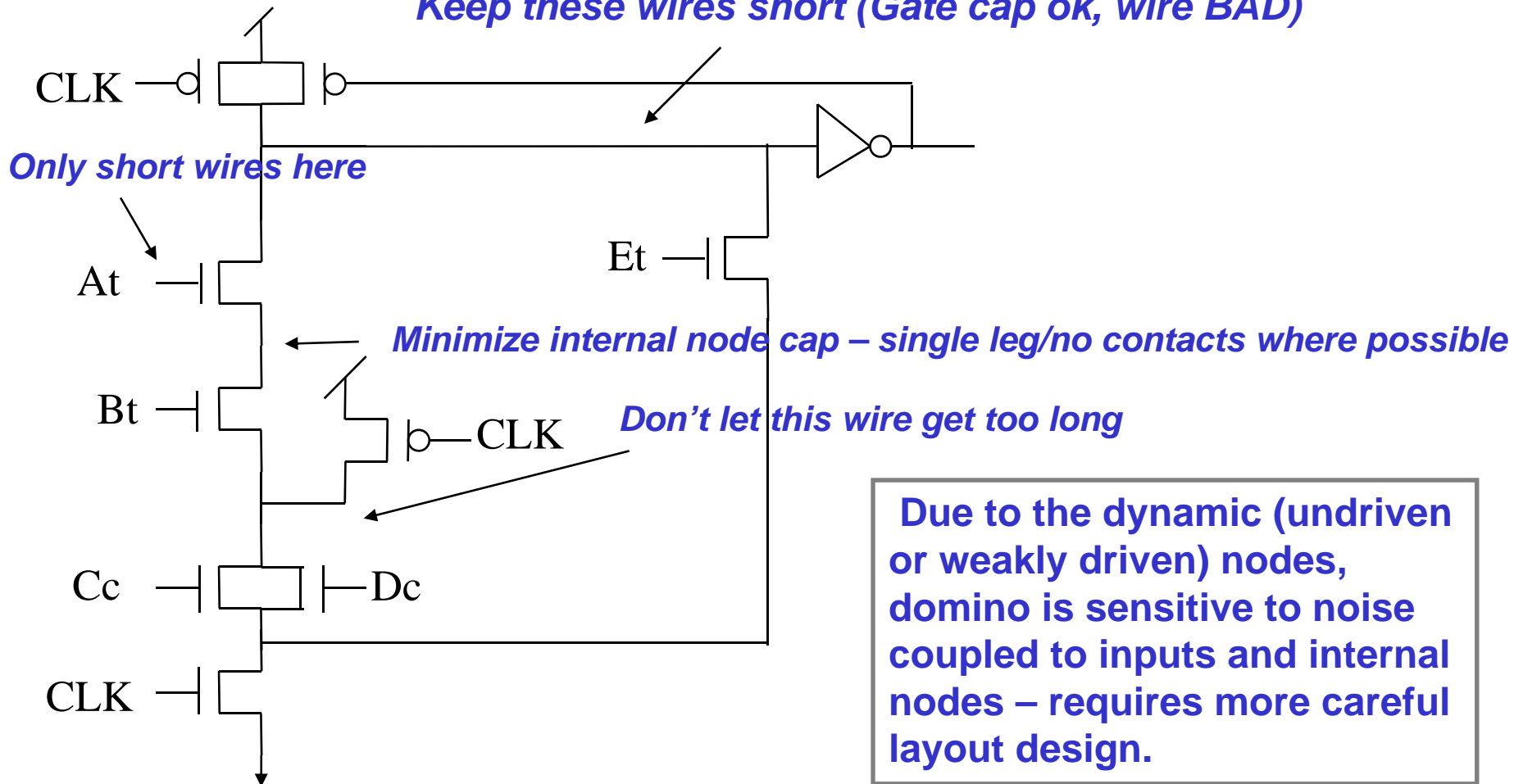
Minimize charge transfer by precharging internal nodes

In the evaluate phase, if $A=1$, AND $B=1$ AND $C=0$, charge is redistributed from n1 to the two parasitic capacitors.

The purpose of the p-keeper is to supply current and maintain the voltage on n1 so that the inverter does not flip erroneously. Domino is unforgiving unlike static CMOS that permits spurious transitions.

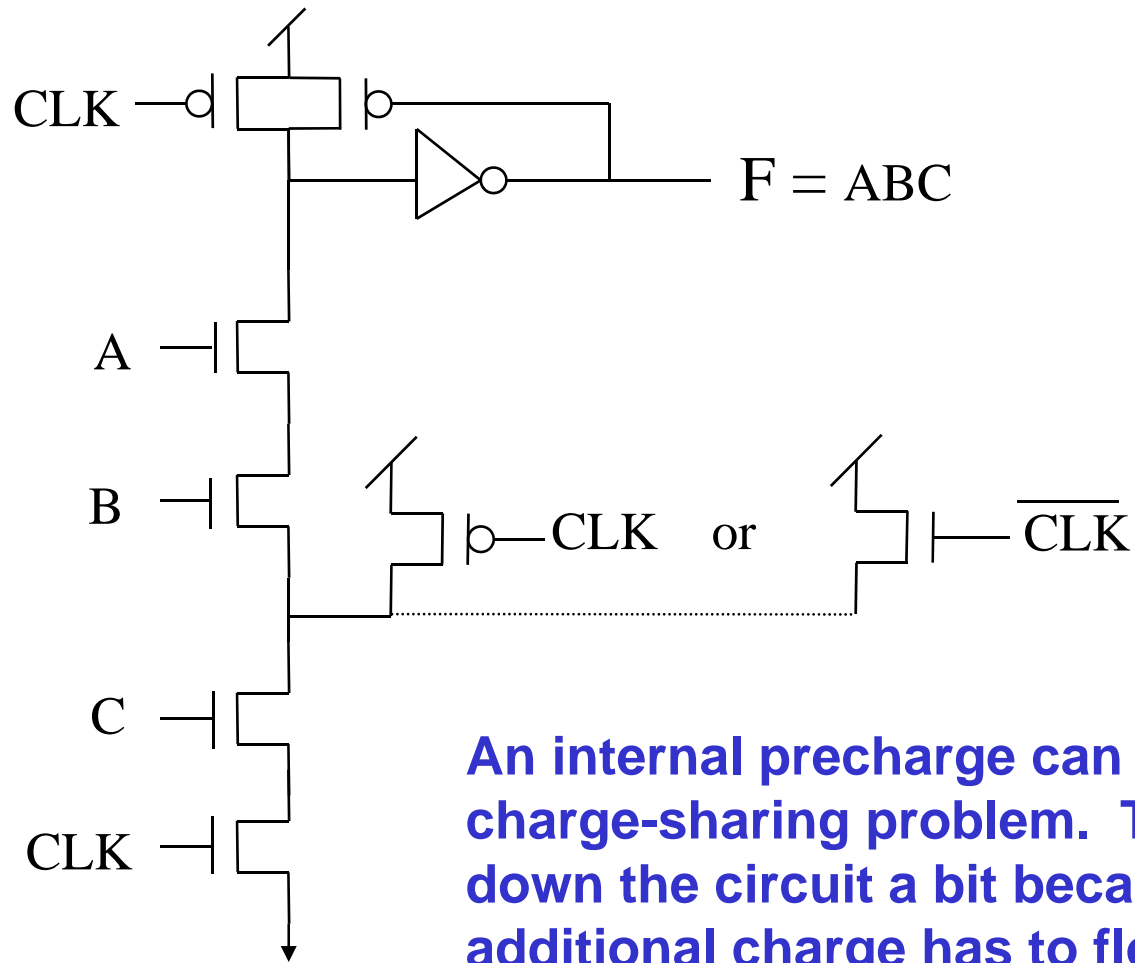
Noise Optimization – layout considerations

Keep these wires short (Gate cap ok, wire BAD)



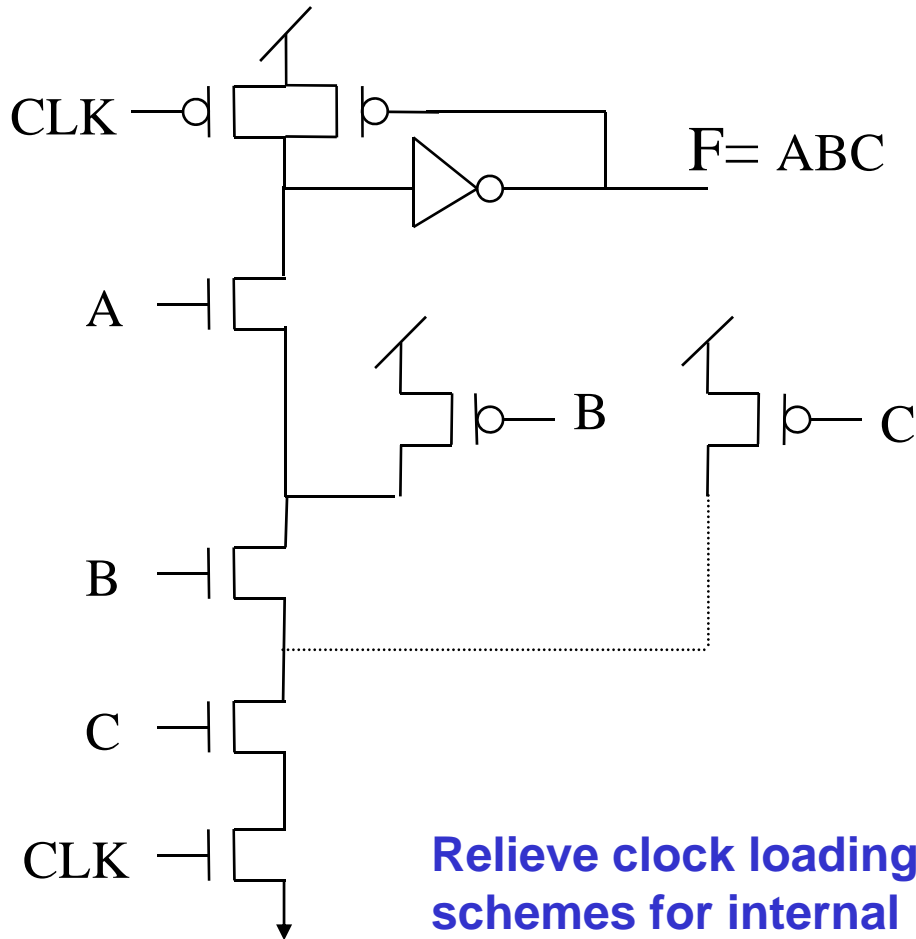
Due to the dynamic (undriven or weakly driven) nodes, domino is sensitive to noise coupled to inputs and internal nodes – requires more careful layout design.

Internal pre-charge scheme - I



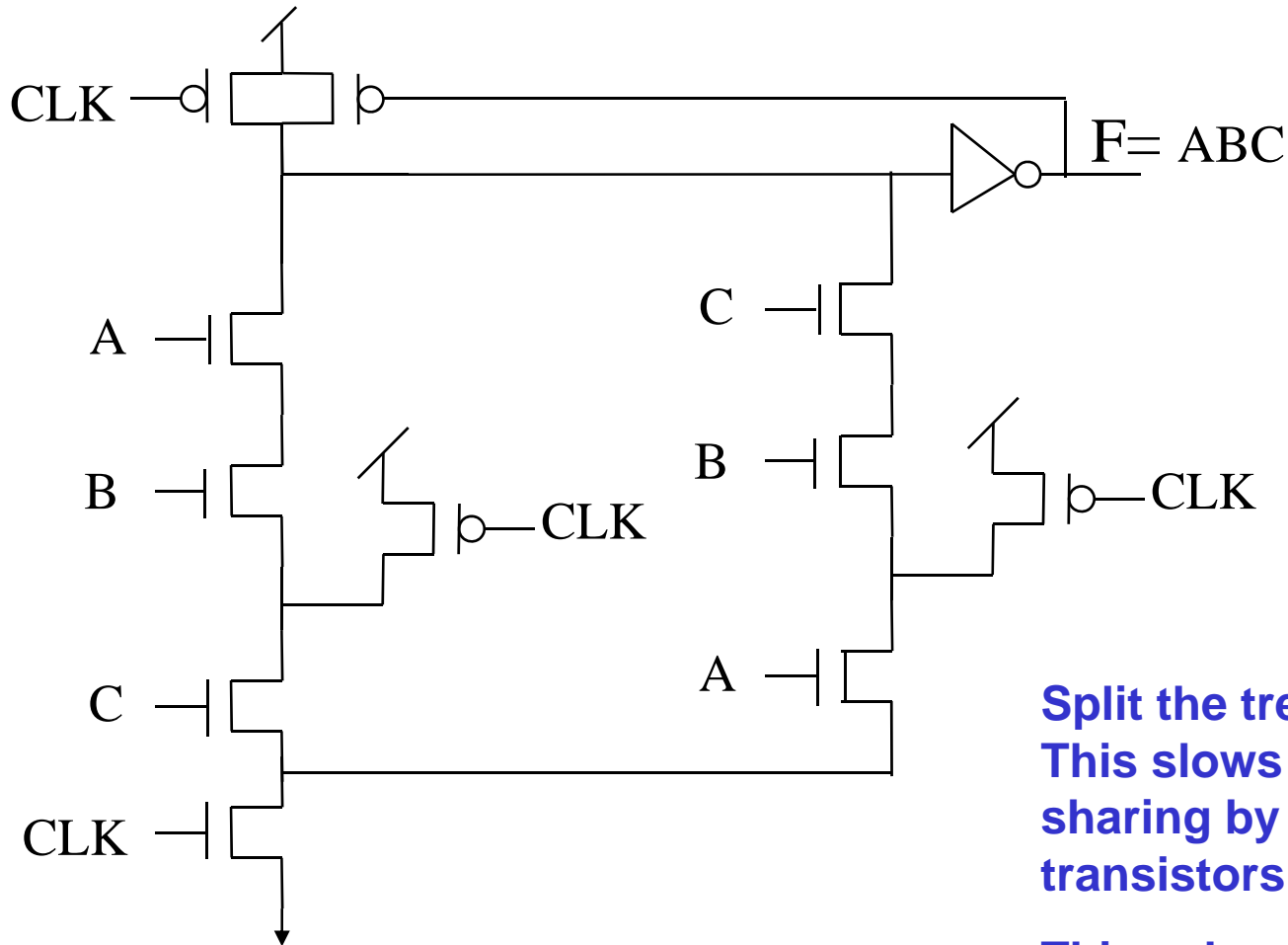
An internal precharge can mitigate the charge-sharing problem. This also slows down the circuit a bit because now the additional charge has to flow to ground when $A=B=C=1$.

Internal pre-charge scheme - II



Relieve clock loading with alternate precharge schemes for internal nodes.

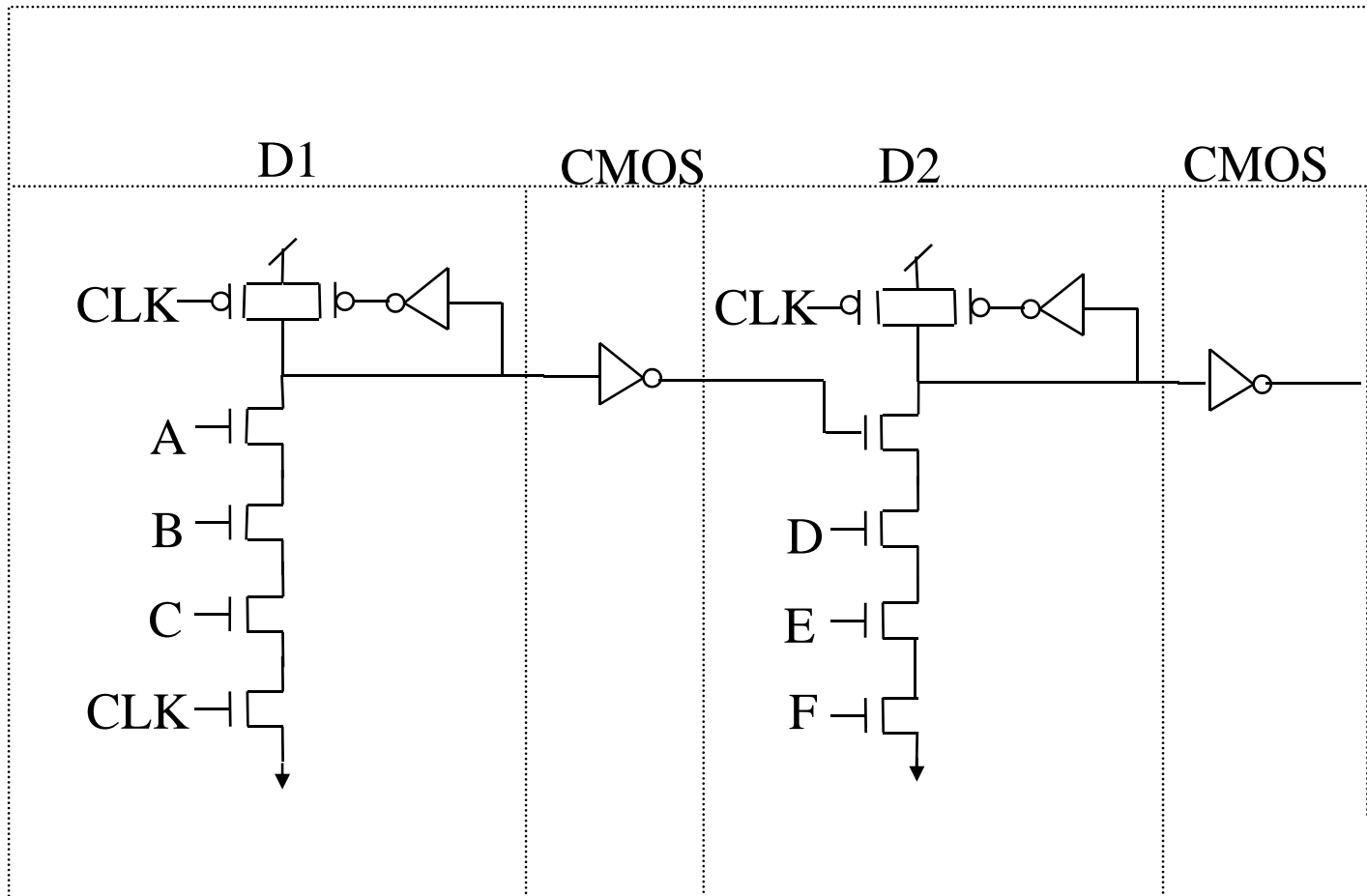
Internal pre-charge scheme - III



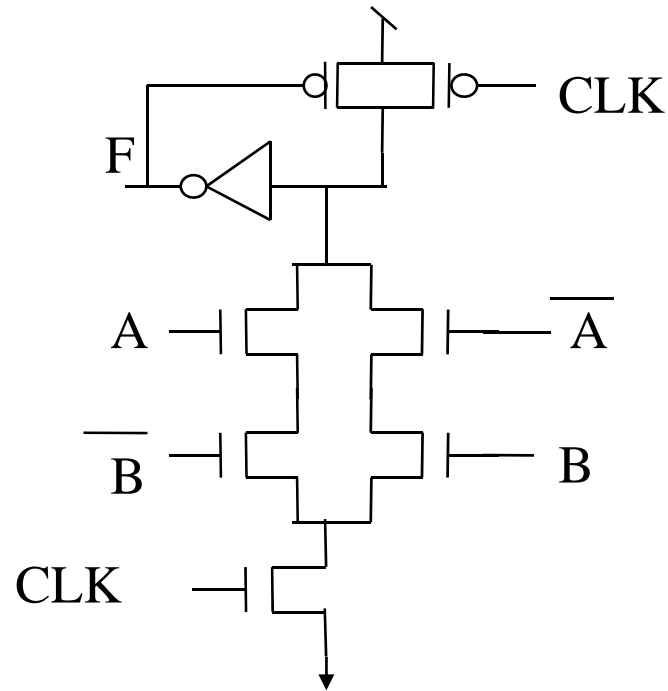
Split the tree into two parts.
This slows the charge-sharing by making the transistors smaller.

This scheme also evens out the delay between all three inputs.

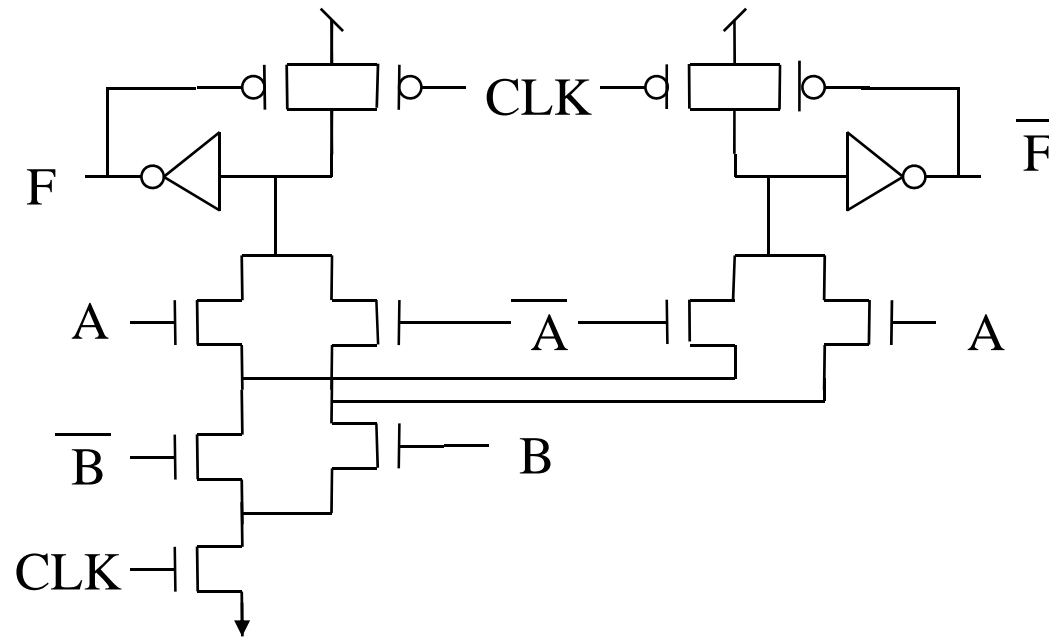
D1 & D2 (footed & footless)



Single Rail Domino: 2-input XOR

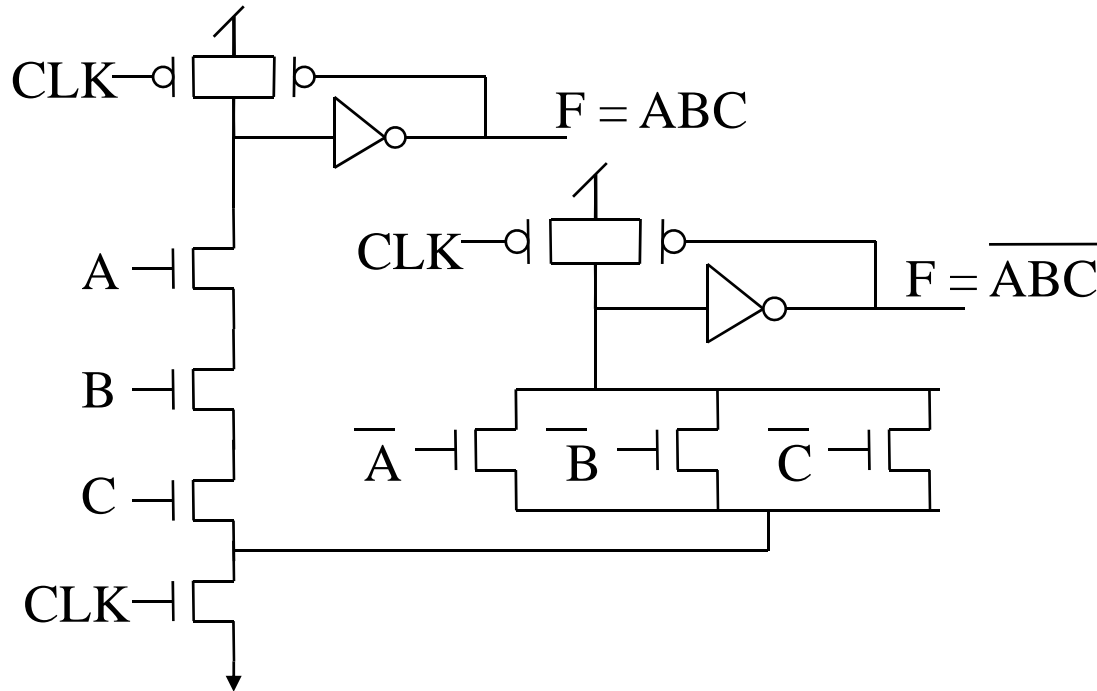


Dual rail domino:2-input XOR



Make use of intermediate nodes to form dual rail domino.
Be careful of “sneak paths”.

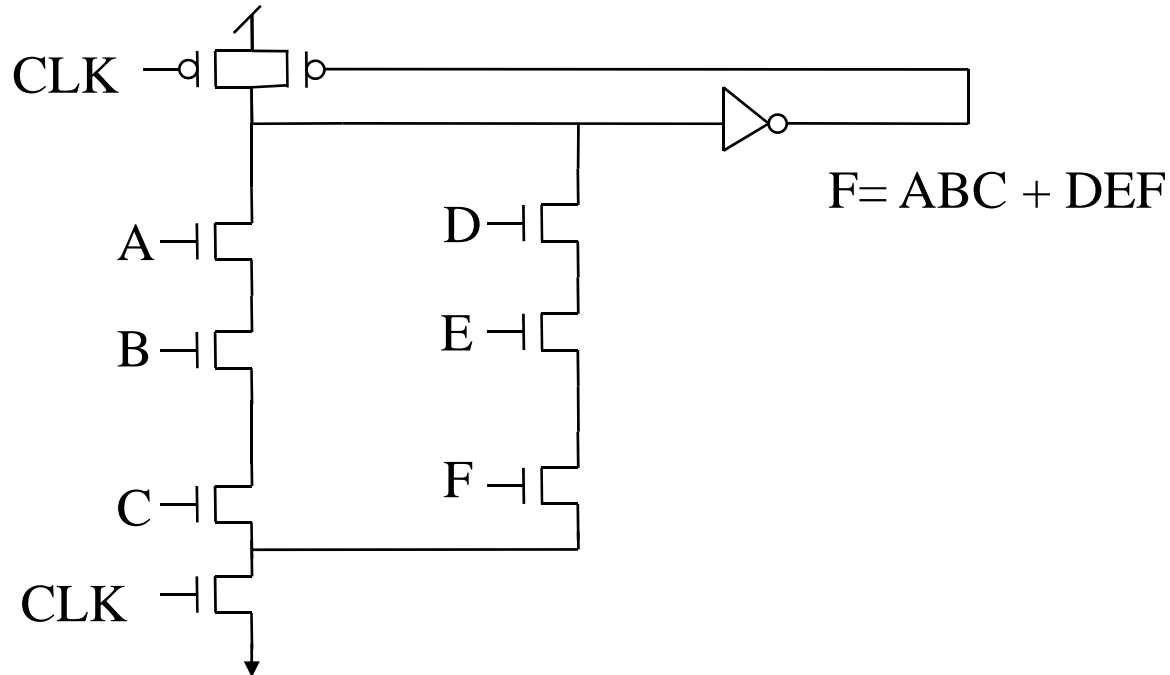
Dual Rail Domino: 3-input AND



Can't always make use of intermediate nodes.

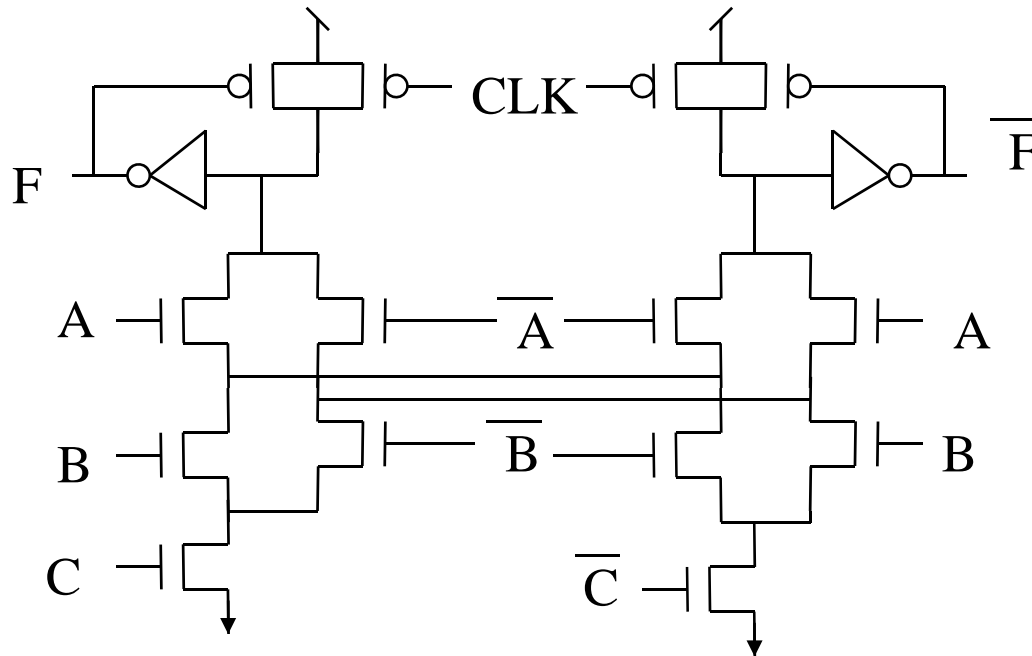
In dual rail, one rail can be faster than the other.

Complex function: AND-OR



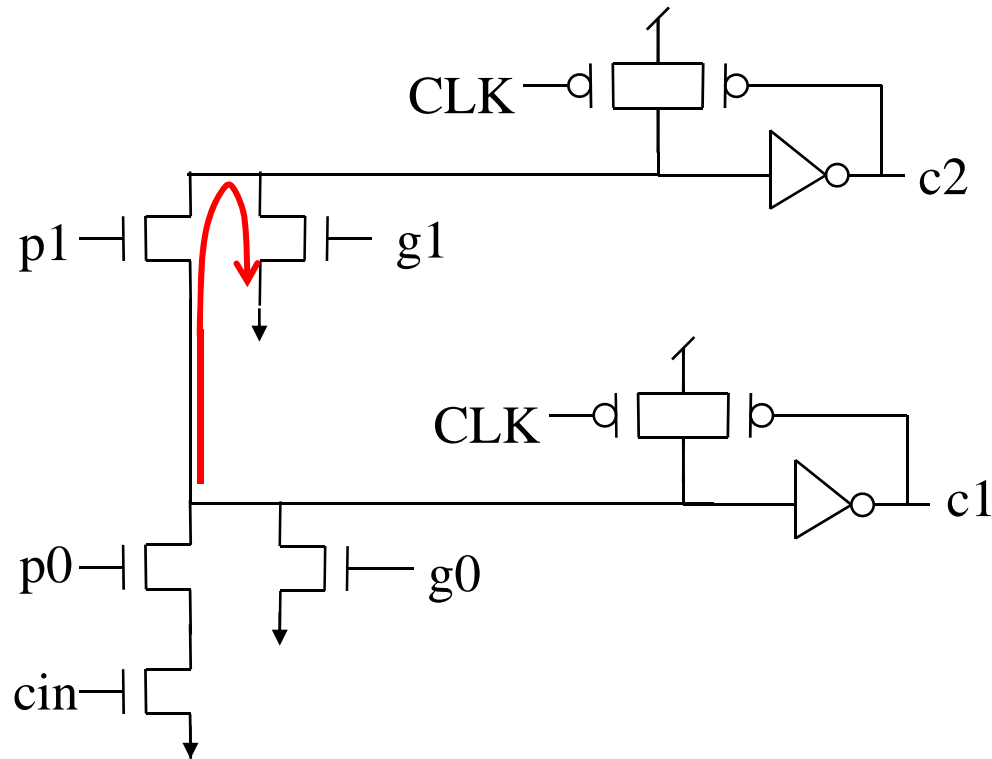
Complex gates can be formed just by adding N-channel transistors.

Complex function: 3-input XOR



In complex gates, make use of intermediate nodes. Again, be careful of “sneak paths”.

sneak path example: carry chain



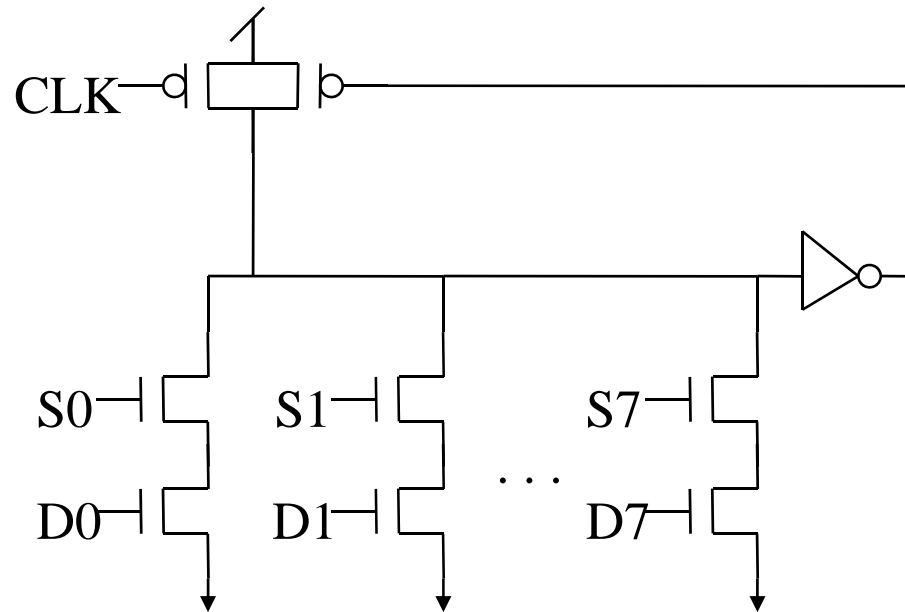
$$g = ab \quad p = a + b$$

$$c1 = g0 + p0cin$$

$$c2 = g1 + p1c1 = g1 + p1g0 + p1p0cin$$

So what happens when $a1=b1=1$ and $a0=b0=0$? Well, $c2=1$ and $c1=?$

8:1 Mux

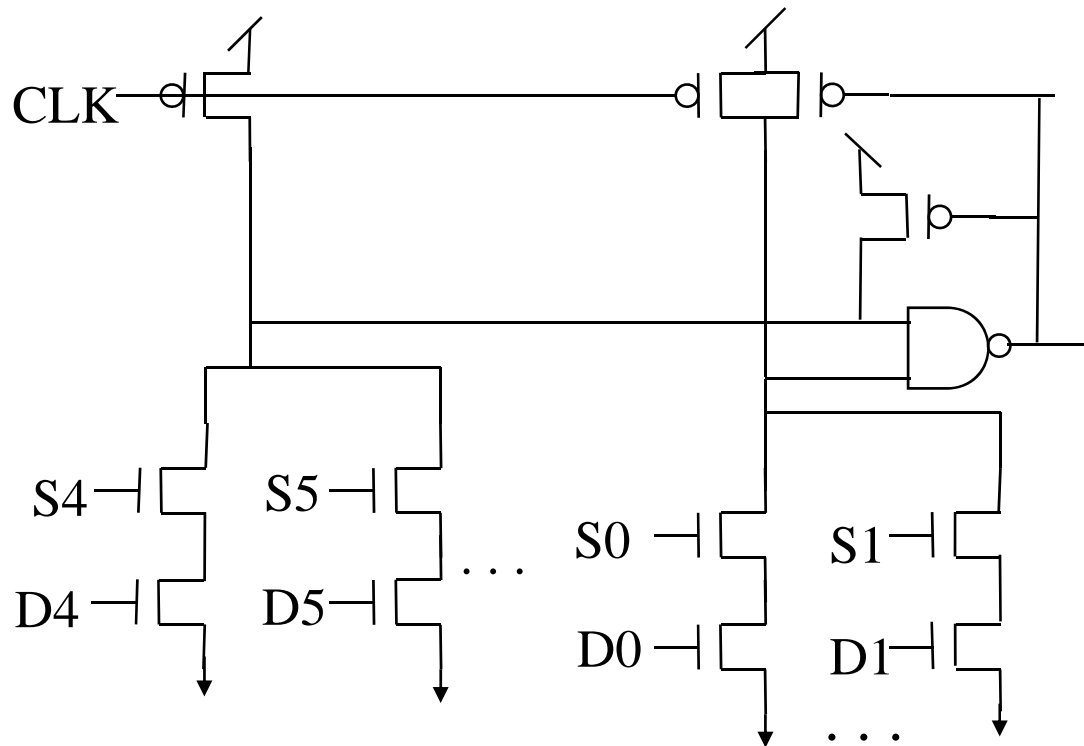


Although this looks OK, the dynamic node is heavily loaded with all of the parasitic diode capacitance.

The heavily loaded dynamic node is good for noise immunity; lousy for performance.

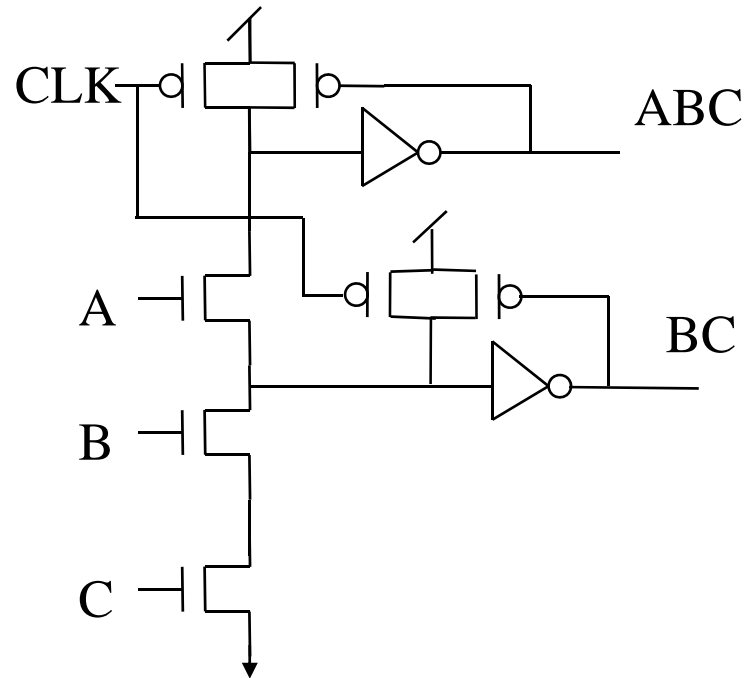
Select signals are nearest the dynamic node for charge-sharing reasons. Only one (few?) select line should ever be high.

Compound Domino



When the dynamic node is heavily loaded sometimes it's best to split it.
This circuit is like two 4:1 muxes melded together.

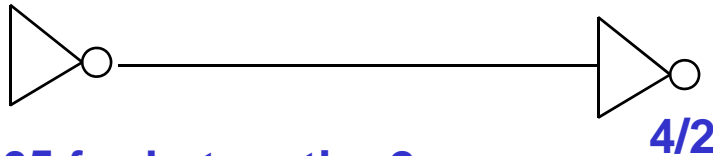
Multiple output domino



Sometimes you can make use of intermediate nodes to form new outputs.
The extra circuitry does slow down the 3-input AND function.

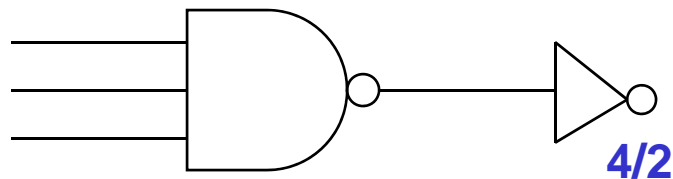
Sizing-I

“equivalent inverter”



1.3/0.65 for beta ratio=2,
fanout=3($w_p+w_n=2$)

1.2/0.8 for beta ratio=1.5

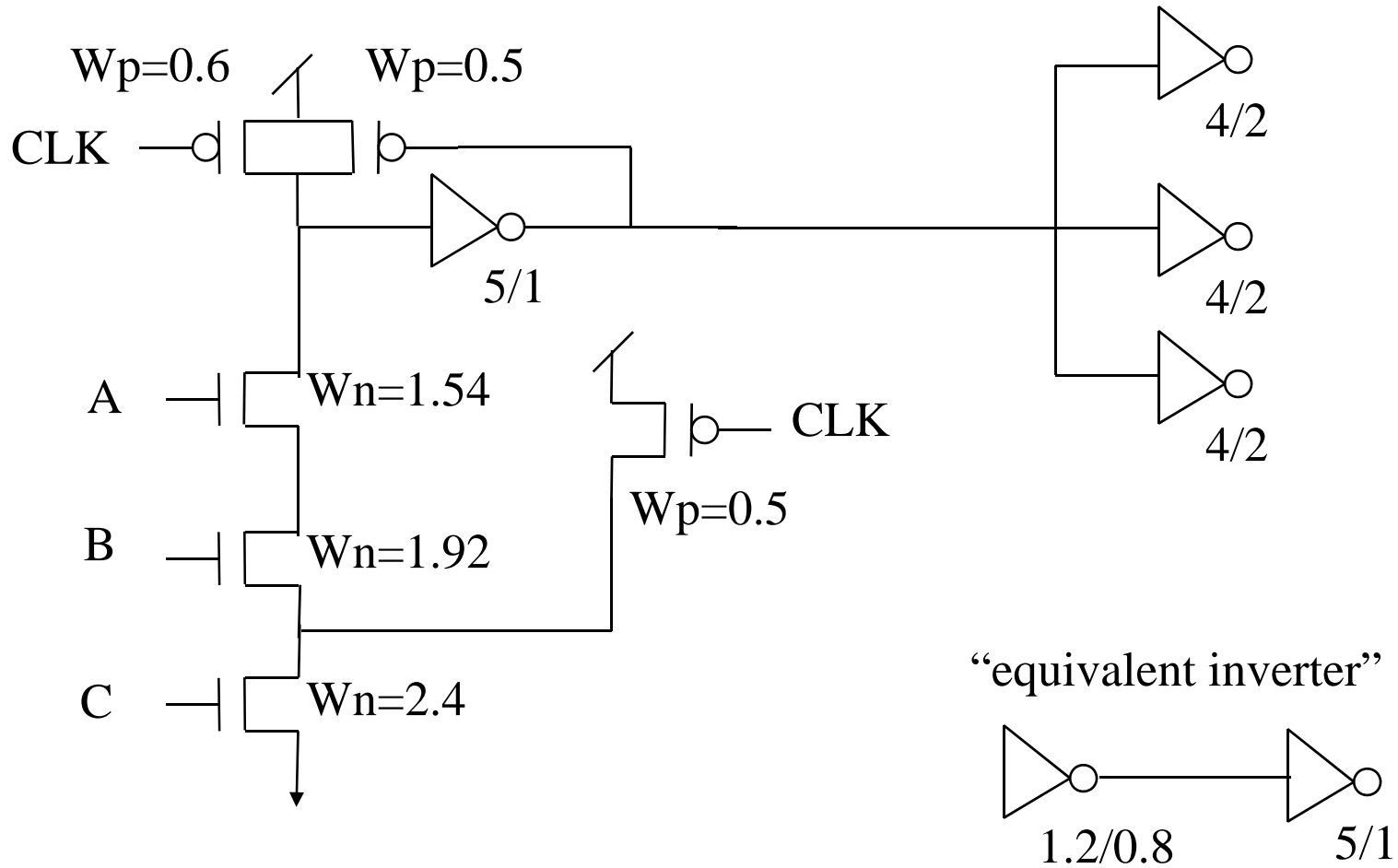


1.3/0.65*3 \rightarrow 1.3/1.95 for beta ratio =2, 1.2/2.4
for beta ratio =1.5

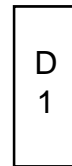
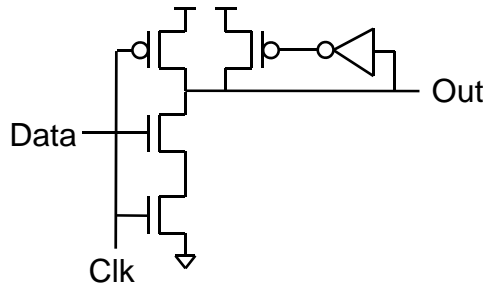
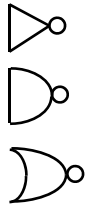
Sizing-II

- Start with the output load, and size the inverter(ratio 4:1 or 5:1) for a fanout of three.
- Size the NMOS tree for a fanout of three and taper.
- Make the p-keeper big enough to overcome leakage and the PMOS precharge device equal to fanout of five or six.
- Increase the size the p-keeper for worst case charge sharing, and/or add internal precharge FETs if necessary.

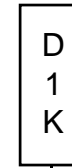
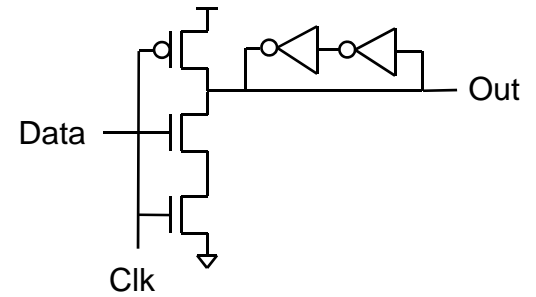
Sizing-III



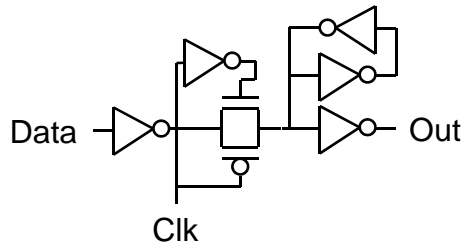
Circuit Elements



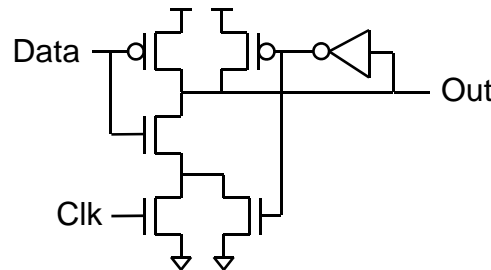
Clk



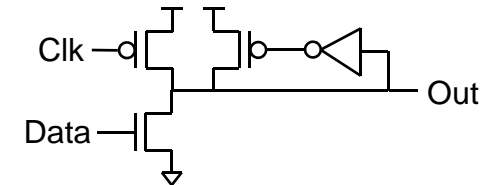
Clk



Clk



Clk



Clk

Bibliography

Domino References

K. Bernstein, et al., *High Speed CMOS Design Styles*, Kluwer Academic Publishers, 1998.

A good modern survey of different circuit styles.

R. Gu, et al., *High-Performance Digital VLSI Circuit Design*, Kluwer Academic Publishers, 1996.

Discusses many different types of digital logic including several flavors of domino.

D. Harris and M. Horowitz, "Skew-Tolerant Domino Circuits", *IEEE Journal of Solid State Circuits*, pp. 1702-1711, 1997.

Describes removing static latches between domino phases in order to allow time borrowing.

D. Harris, *Skew-Tolerant Circuit Design*, Morgan Kaufmann Publishers, 2001.

Harris has expanded his JSSC paper into a very good very accessible textbook on time borrowing for both static and domino circuits.

EE-382M

VLSI-II

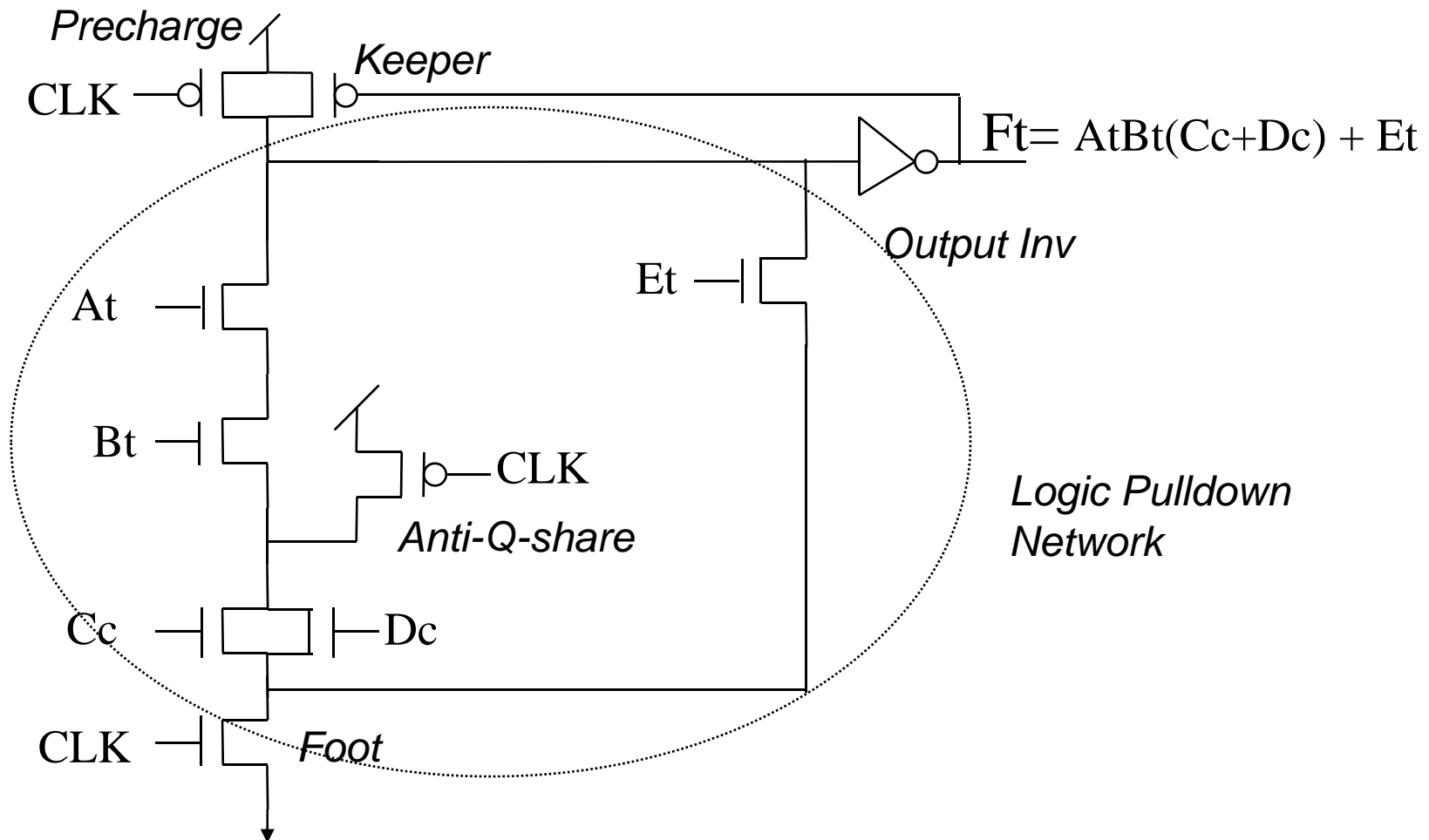
**Dynamic Circuits:
Beyond basic domino**

Kevin Nowka, IBM

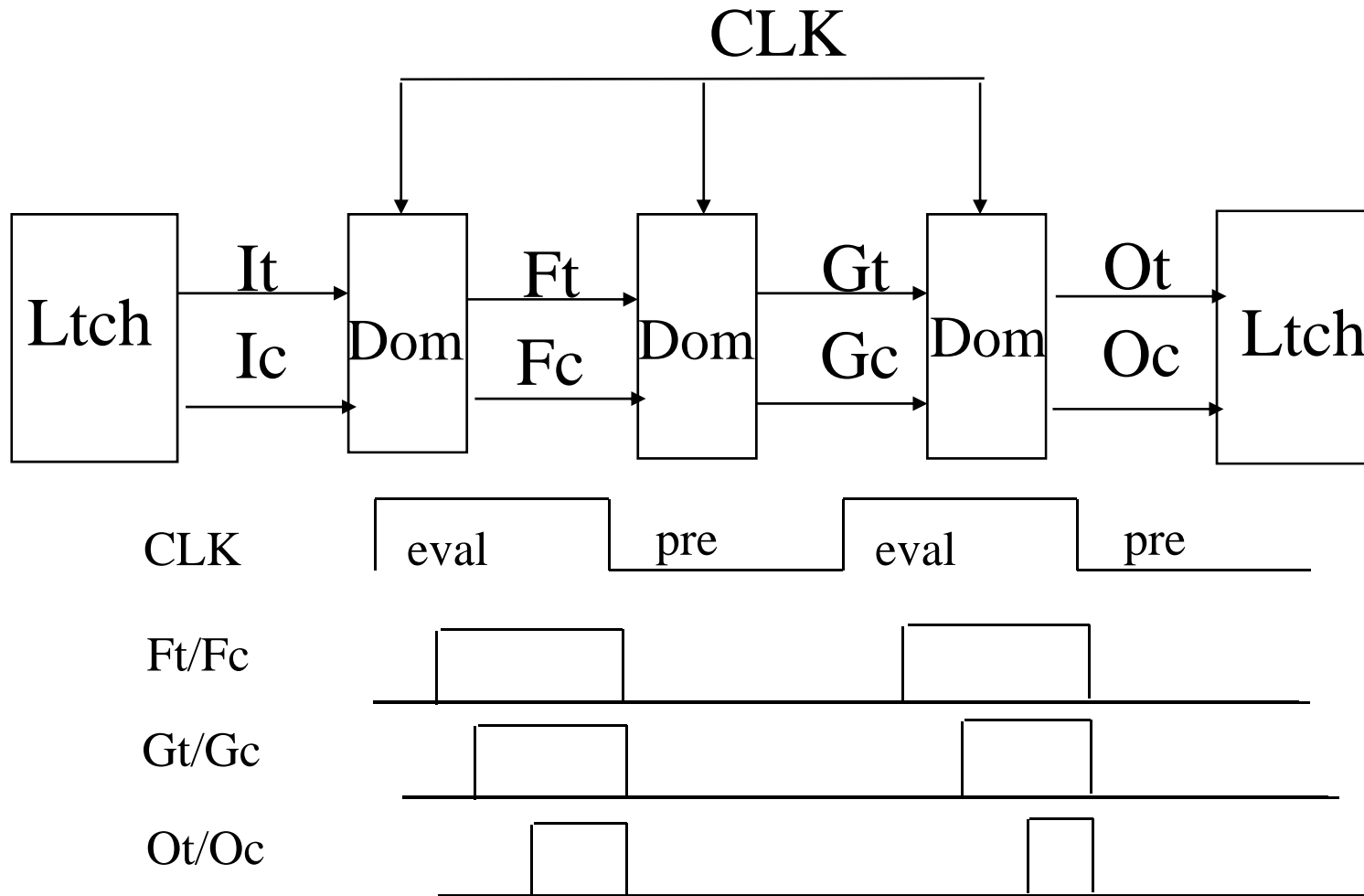
Agenda

- Review of domino
- Performance Enhancements
- Robustness Enhancements
- Power Enhancements
- Technology and dynamic logic
- Summary

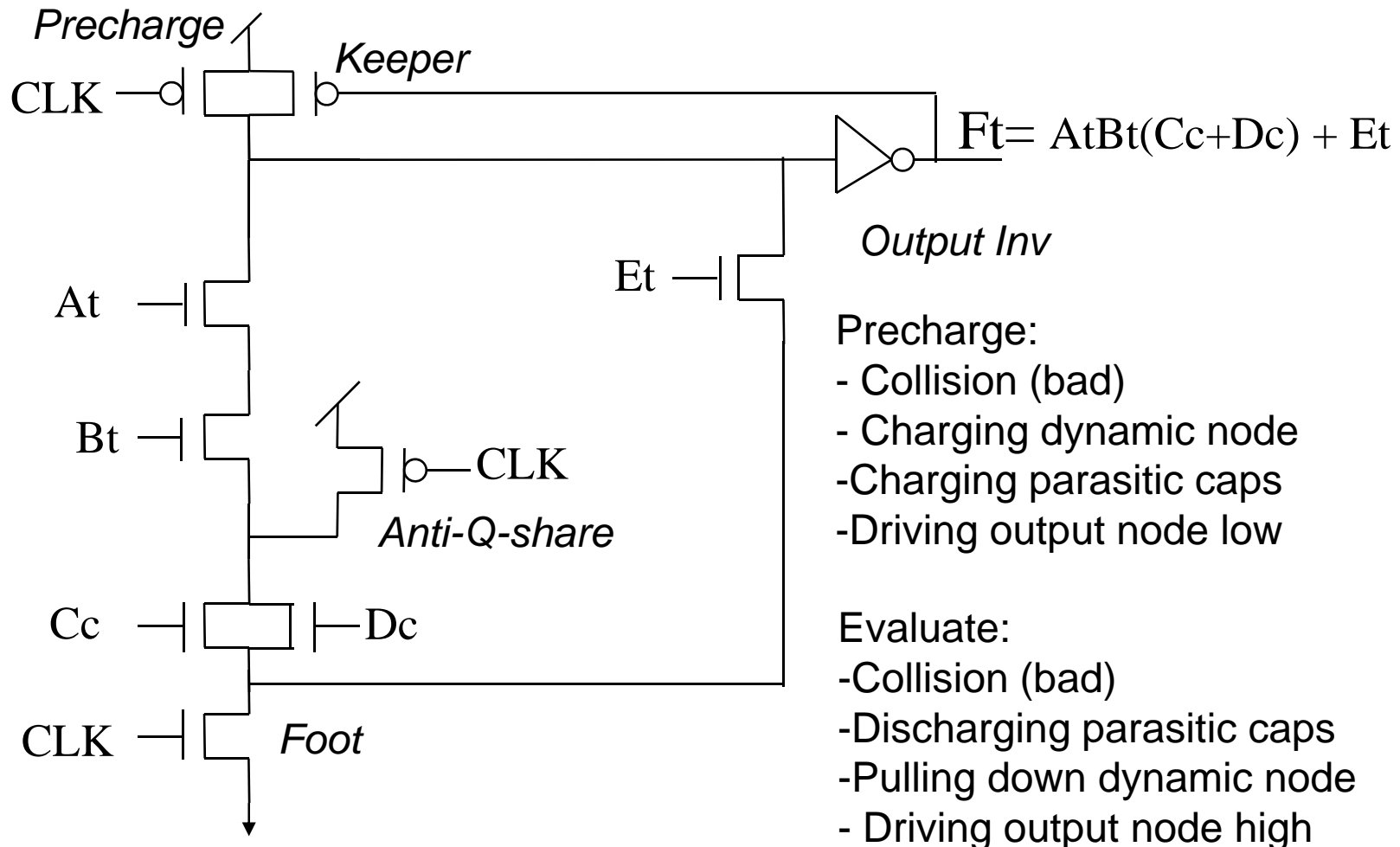
Review of Domino Gate



Review of Domino Timing



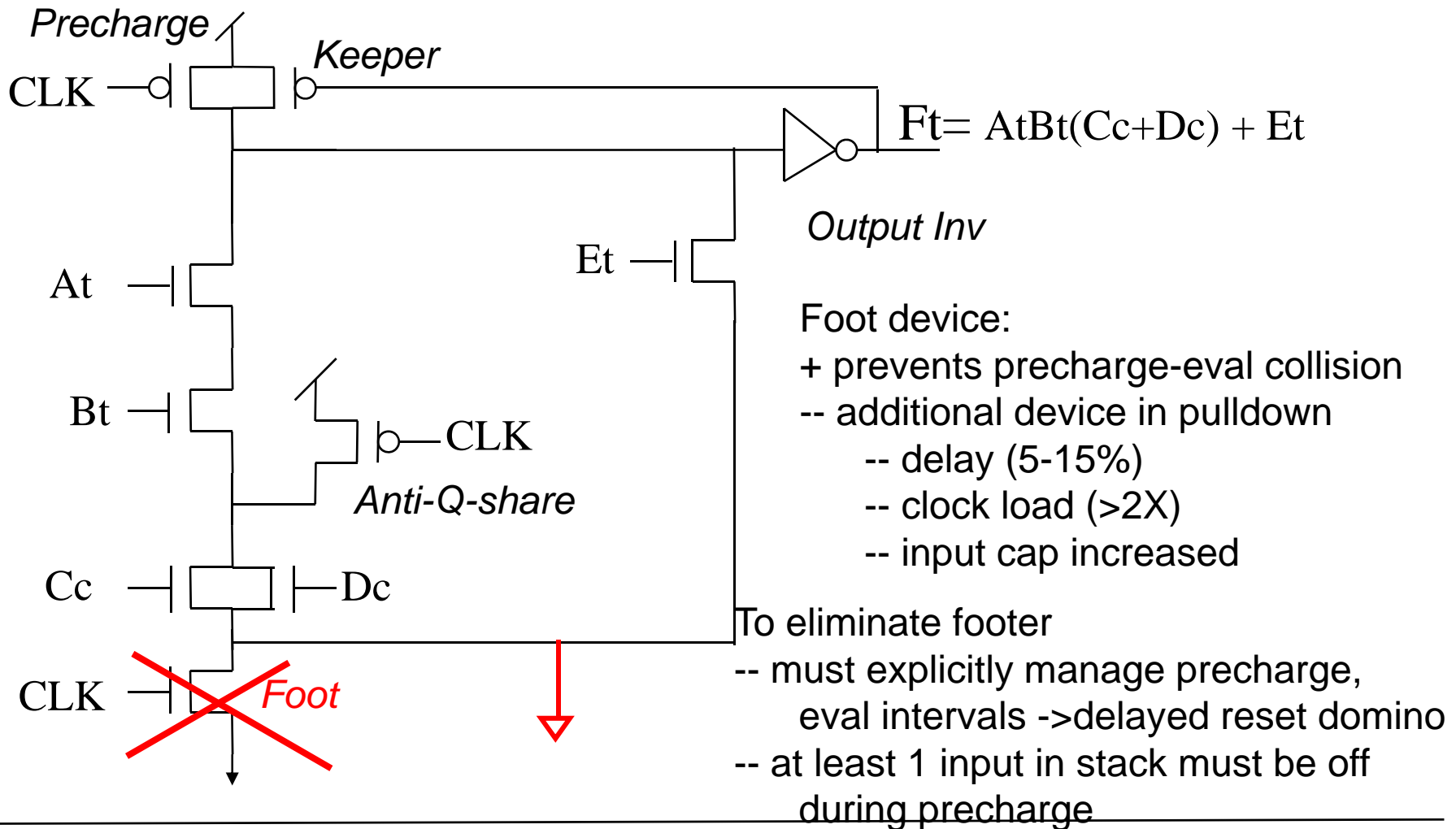
Performance – where does the time go?



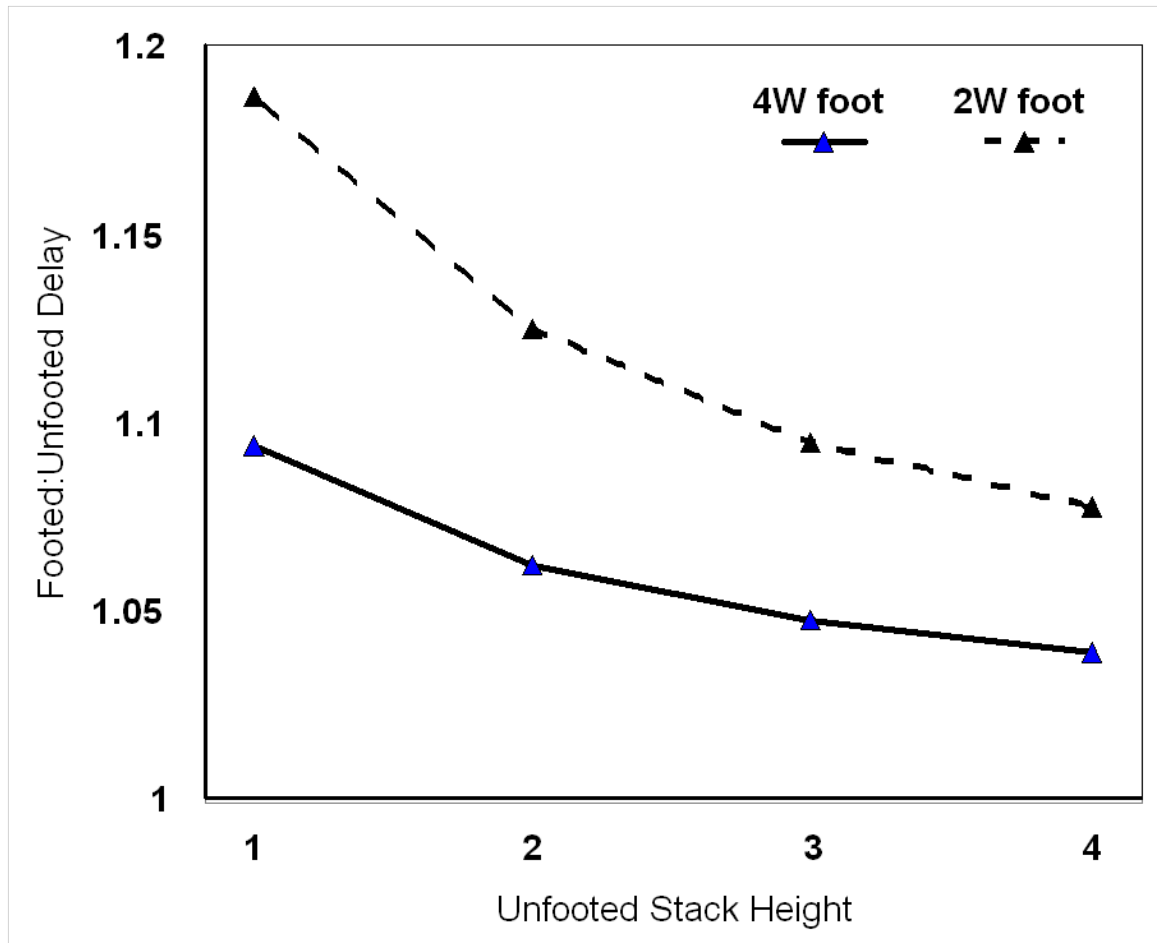
Performance Enhancements

- **No foot**
- **Beta skew**
- **Predischarge**
- **Reset assist**
- **Tapering**
- **MODL (already covered)**

Performance Optimization -- unfooted

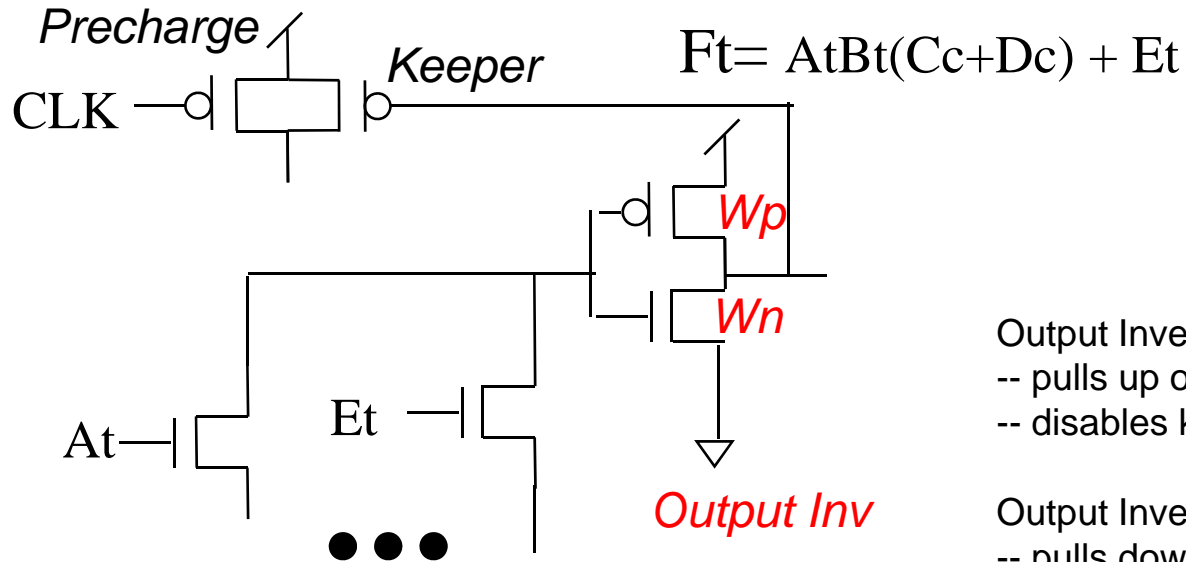


Footed vs. Unfooted Domino



Carry Merge Foot vs. Unfooted Delay

Performance Optimization – beta ratio



Output Inverter PFET:

- pulls up output during evaluate: critical
- disables keeper

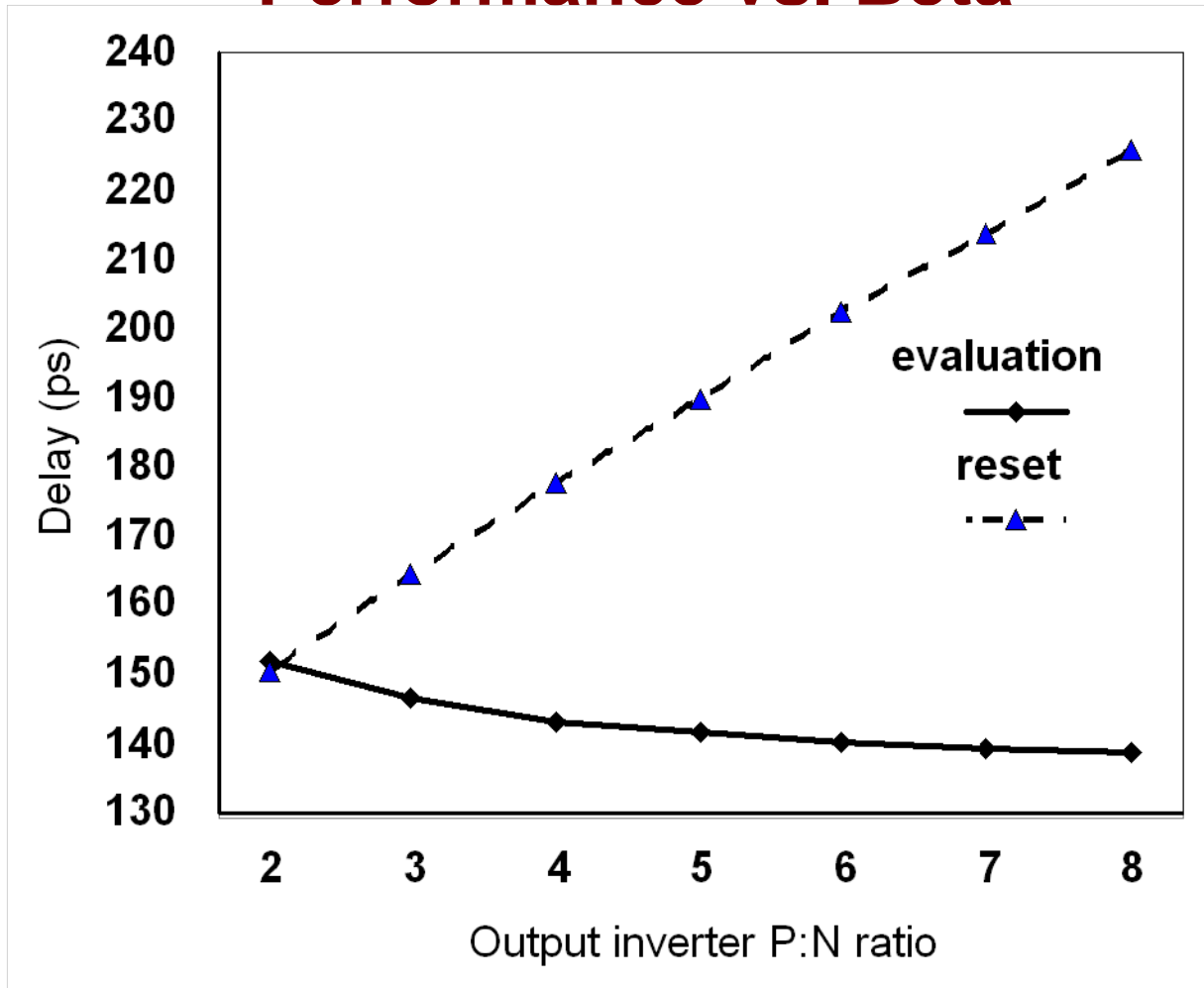
Output Inverter NFET

- pulls down output during precharge
- holds output low during standby
- enables keeper during standby

Skew beta ratio of inverter in favor of PFET

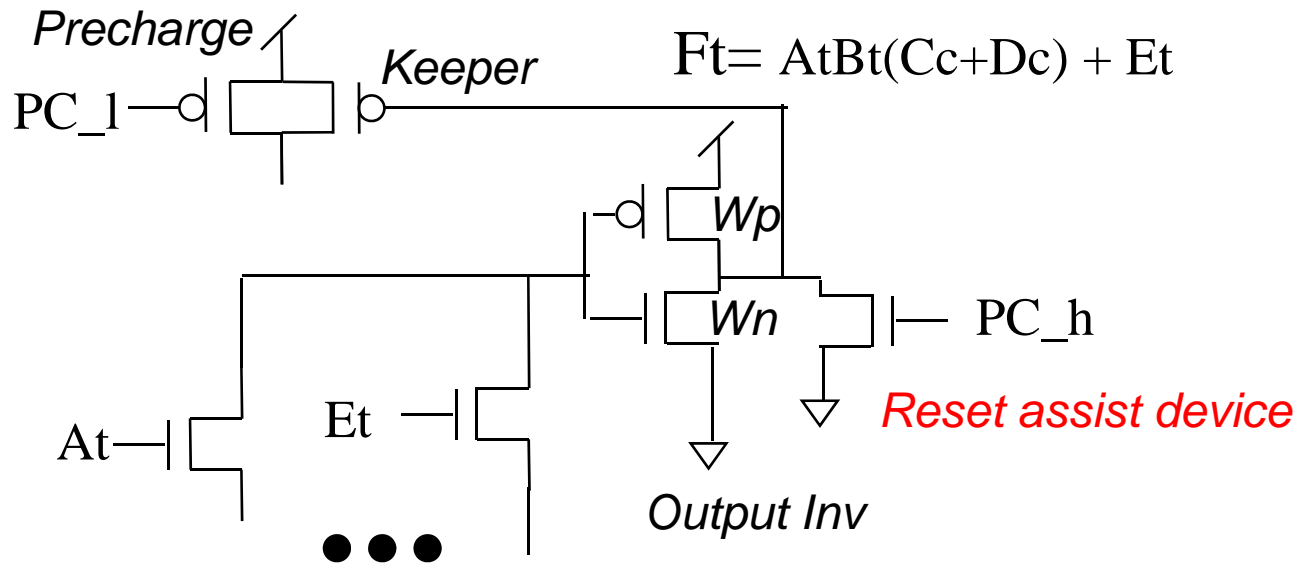
- faster
- reduced noise margin
- slow reset of output
 - can be mitigated with additional reset assist device

Performance vs. Beta

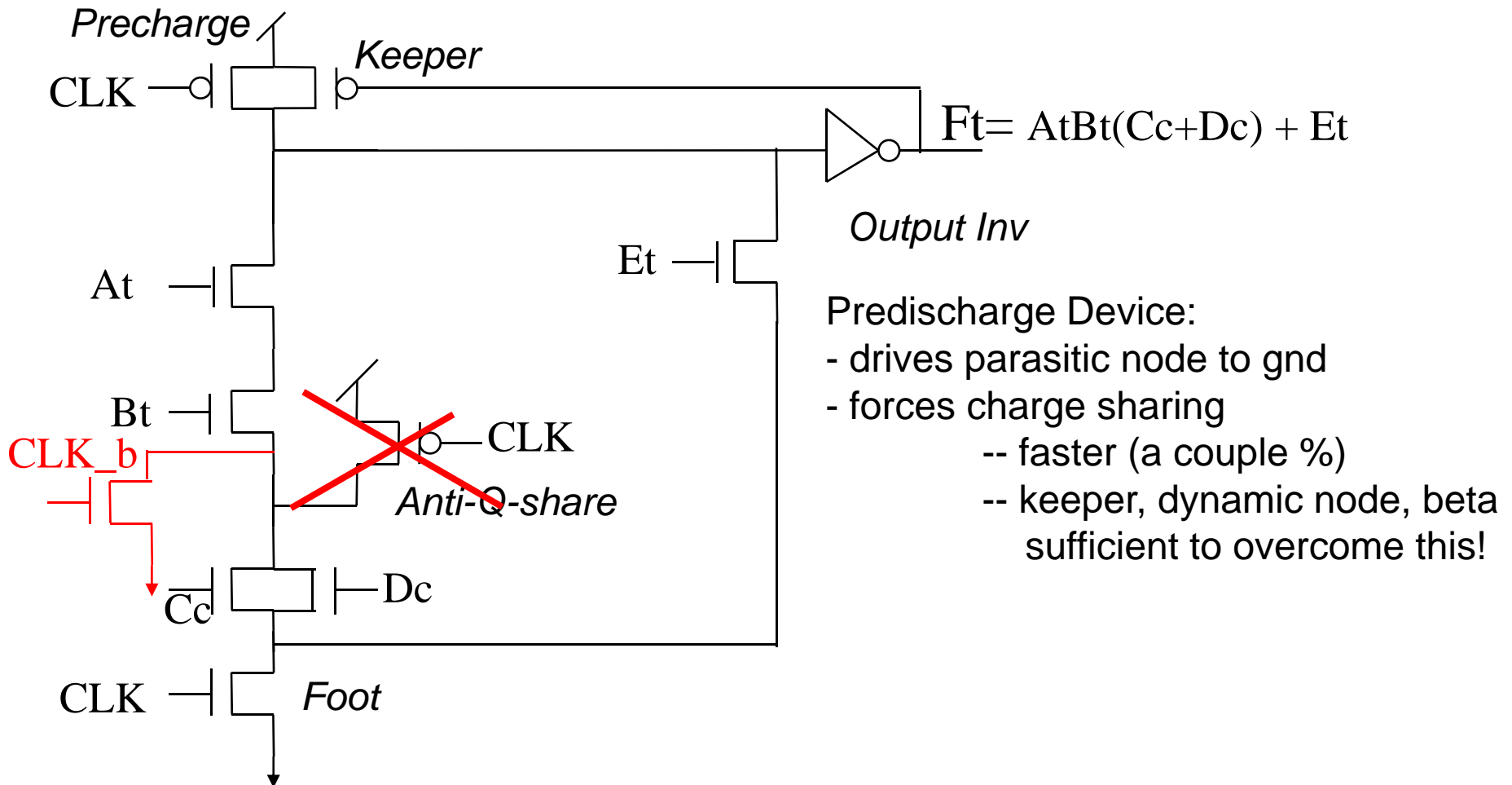


Carry merge circuit delay vs. Beta

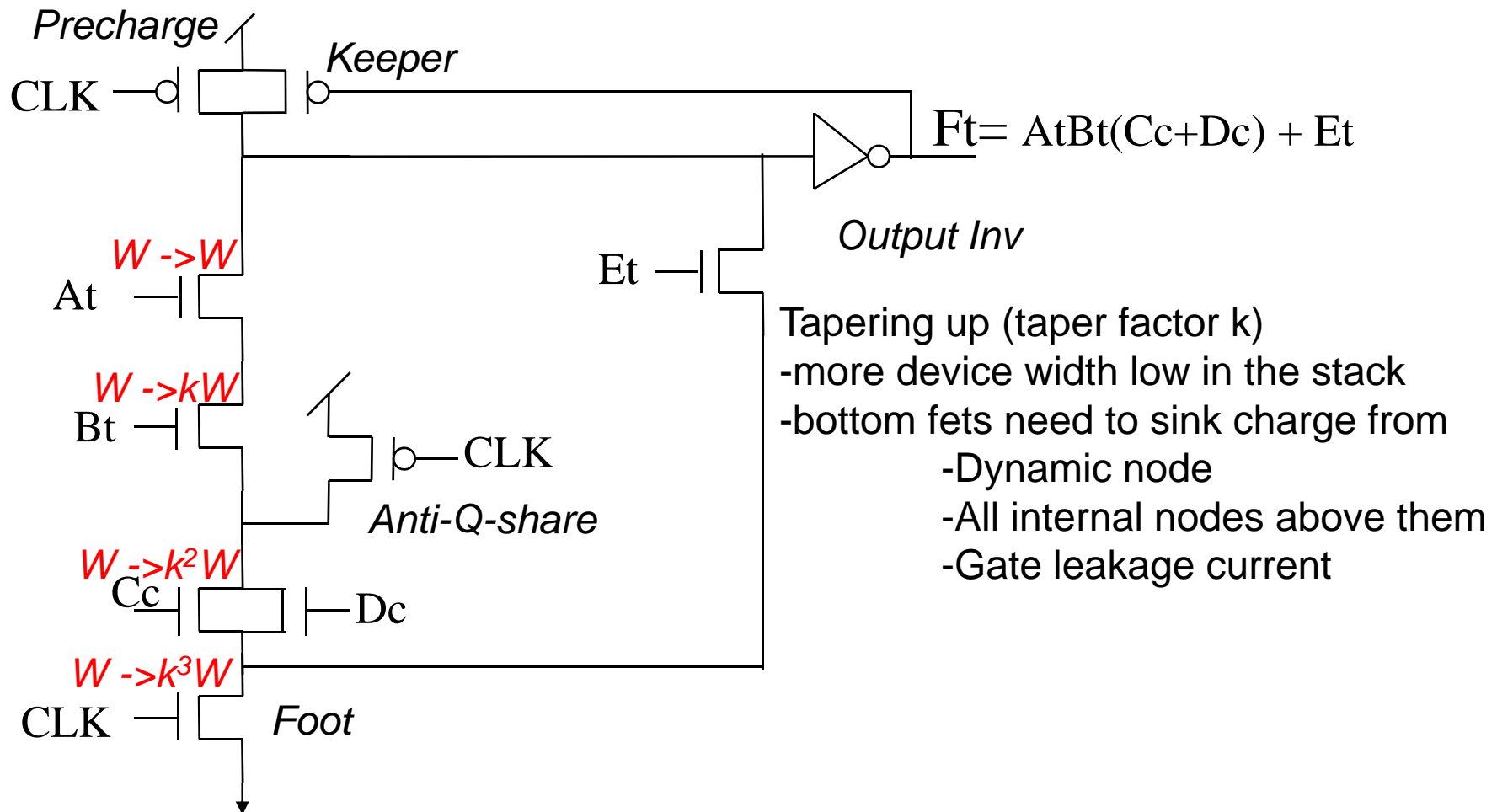
Performance Enhancement – high beta w/ reset assist



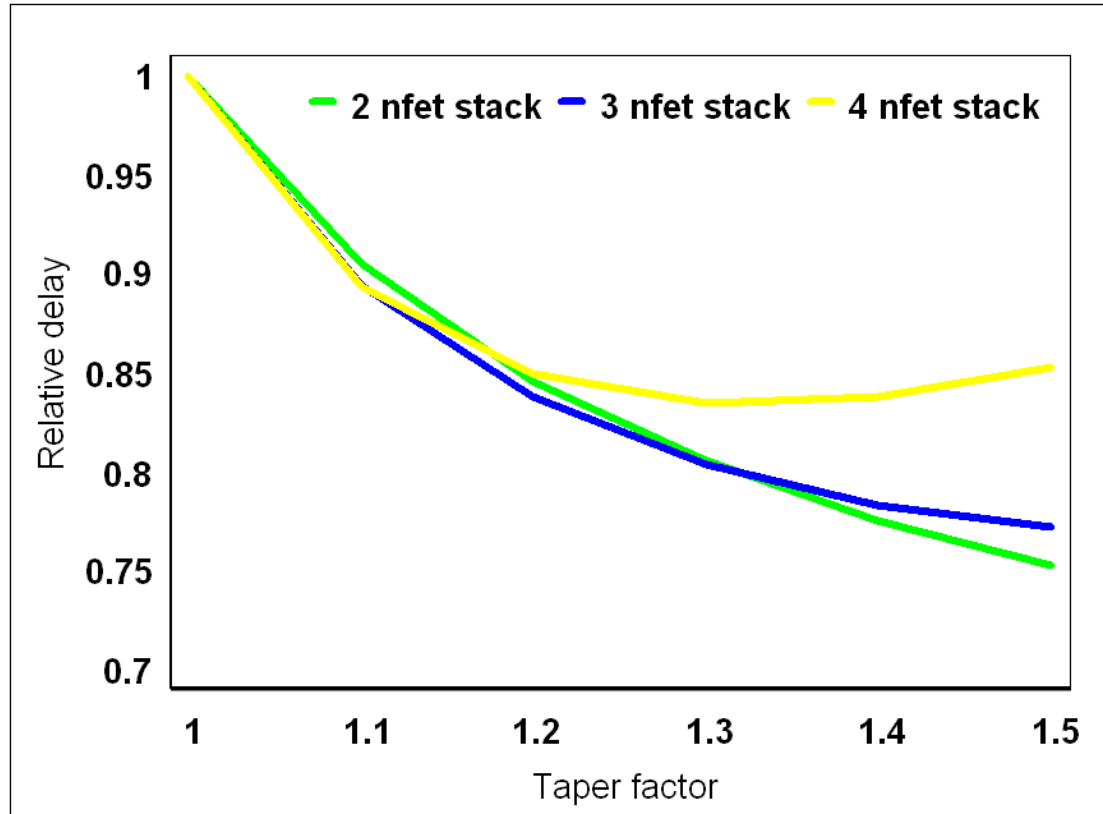
Performance Enhancement -- predischarge



Performance Enhancement – tapering up

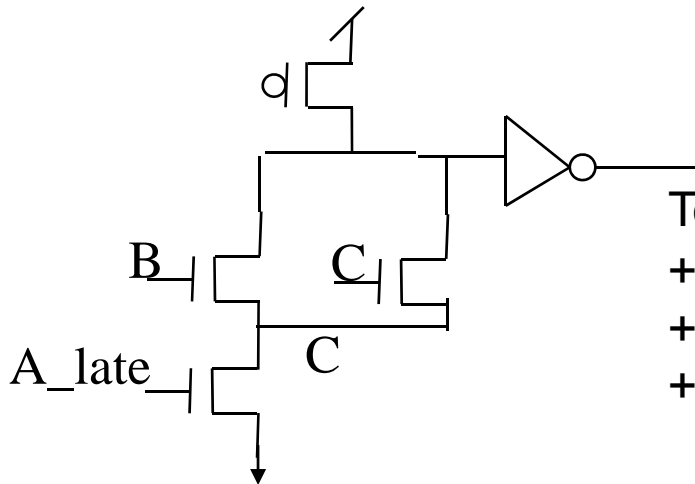


Performance vs. Tapering UP

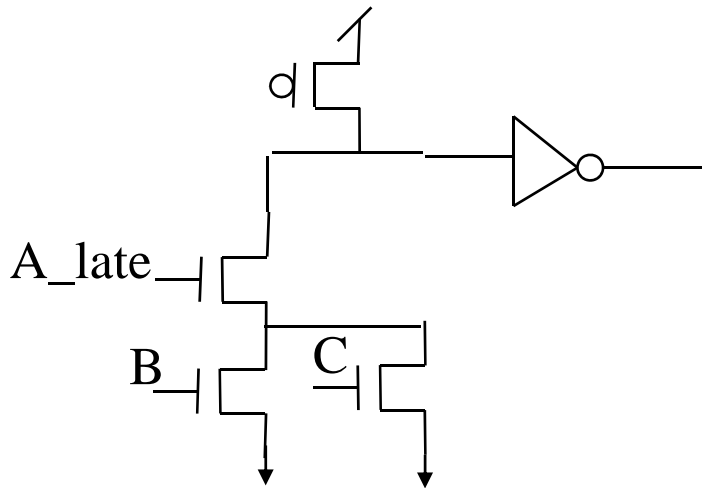


0.18um bulk CMOS process

Performance Optimization – topology changes



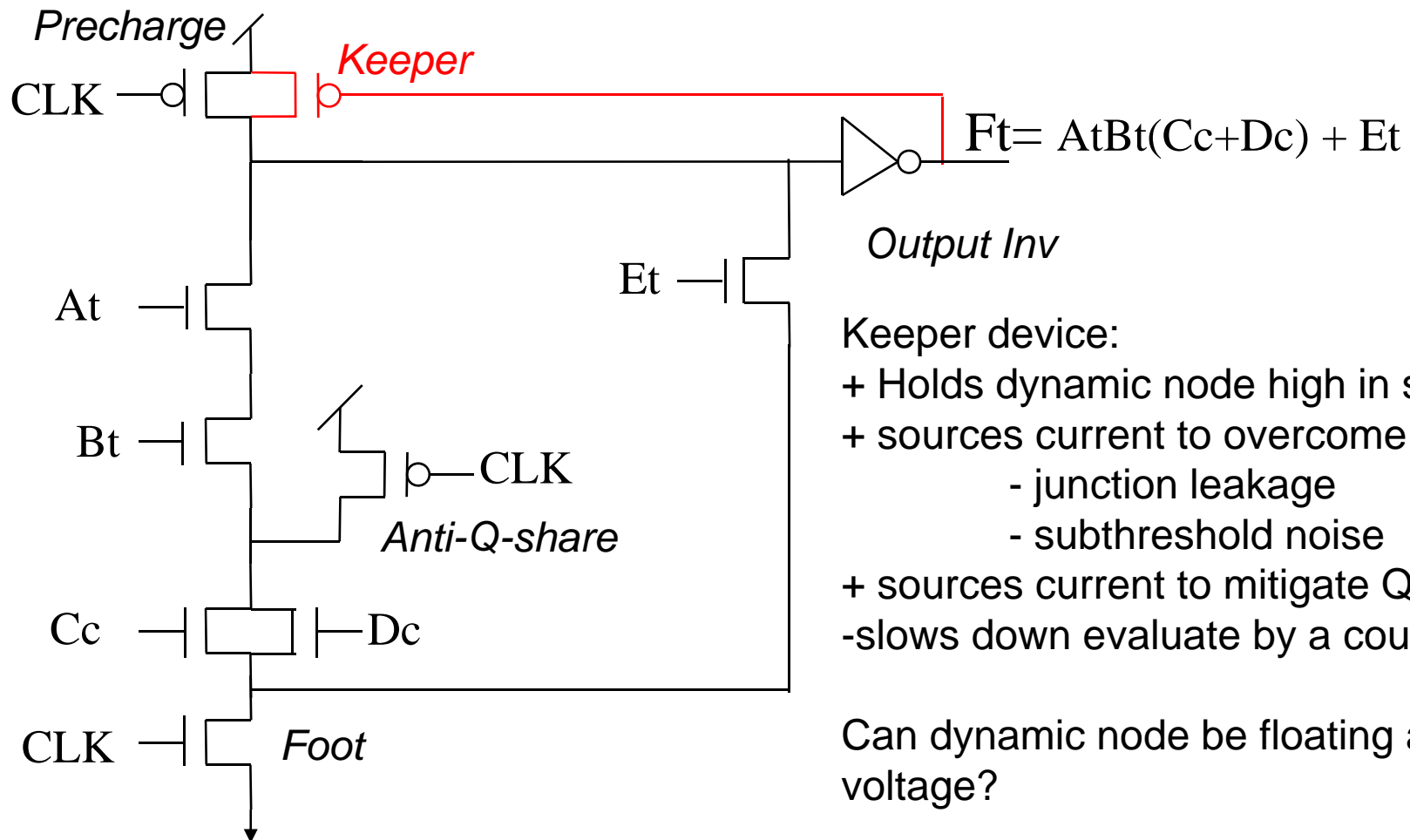
Topology change:
+ latest arriving signal nearest dynamic node
+ decrease dynamic node cap
+ increase charge sharing!



Robustness Enhancements

- **Keepers**
- **Anti-Q-sharing devices**
- **Beta ratios**
- **Topology changes**
- **Layout considerations**

Robustness Optimization -- keeper

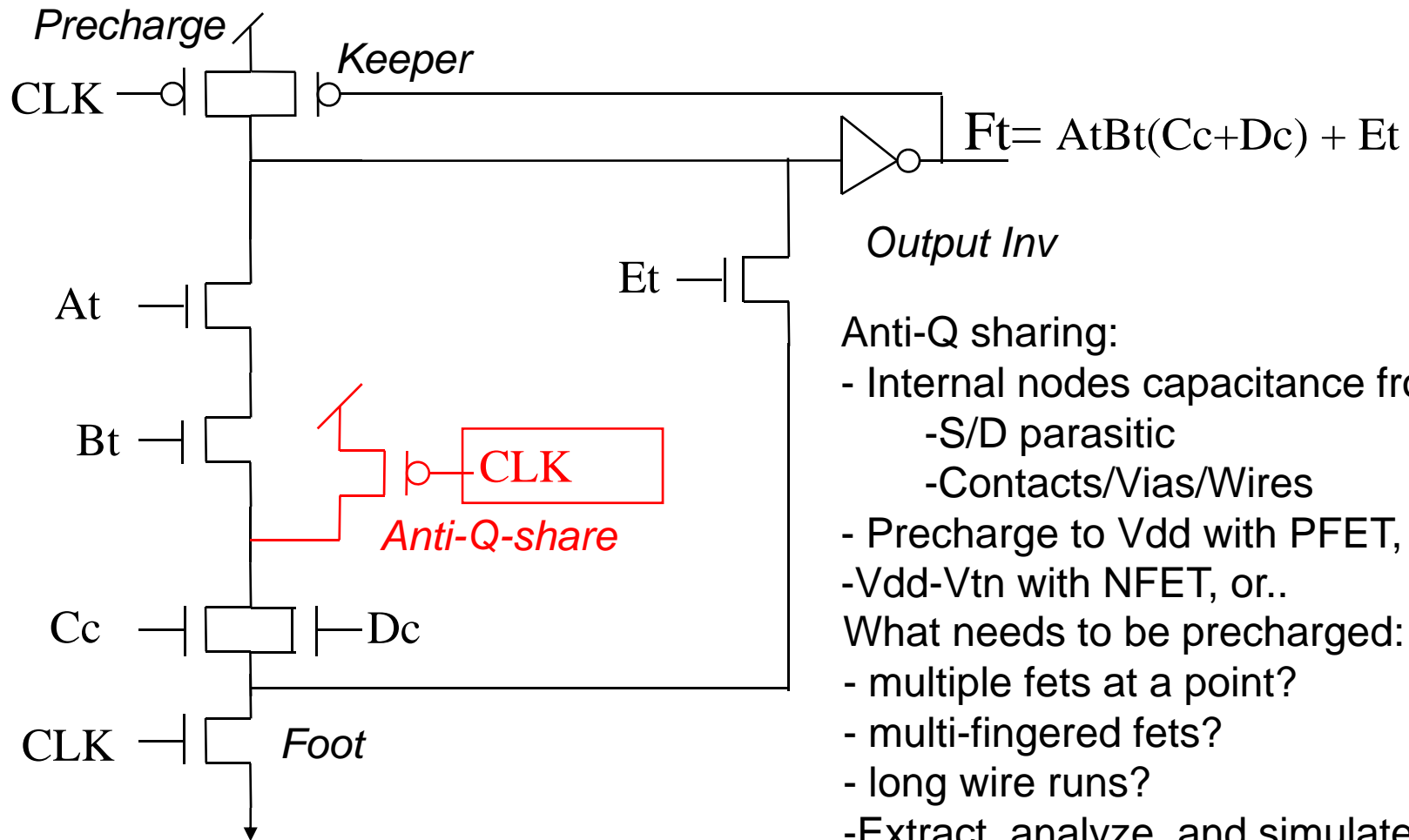


Keeper device:

- + Holds dynamic node high in standby
- + sources current to overcome leakage
 - junction leakage
 - subthreshold noise
- + sources current to mitigate Q sharing
- slows down evaluate by a couple %

Can dynamic node be floating at a low voltage?

Robustness Enhancement – anti-Q-share



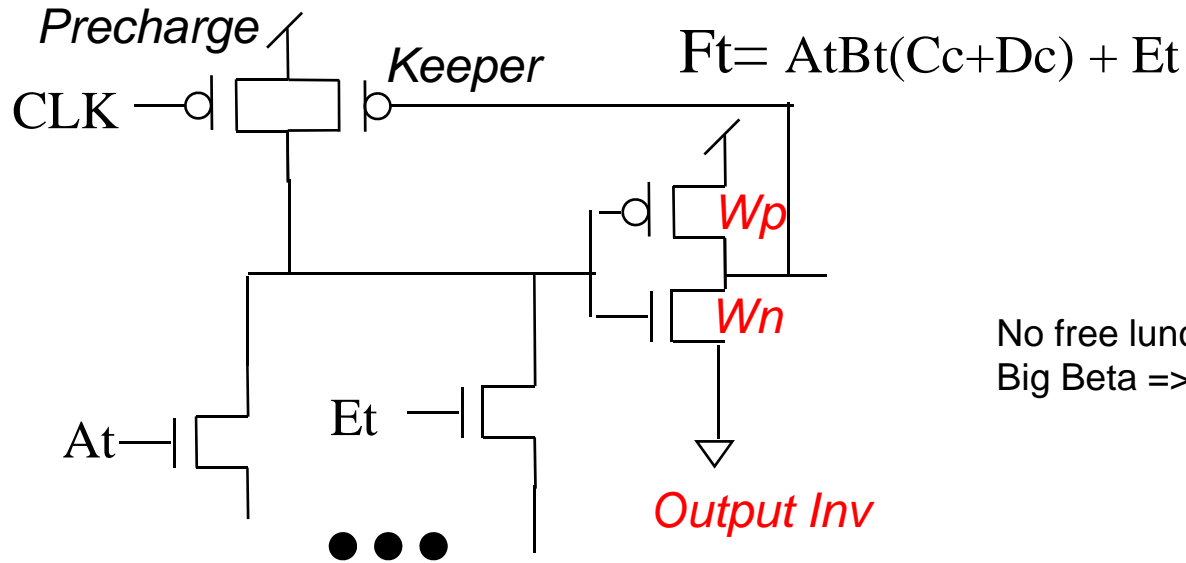
Anti-Q sharing:

- Internal nodes capacitance from
 - S/D parasitic
 - Contacts/Vias/Wires
- Precharge to Vdd with PFET, or
- Vdd-Vtn with NFET, or..

What needs to be precharged:

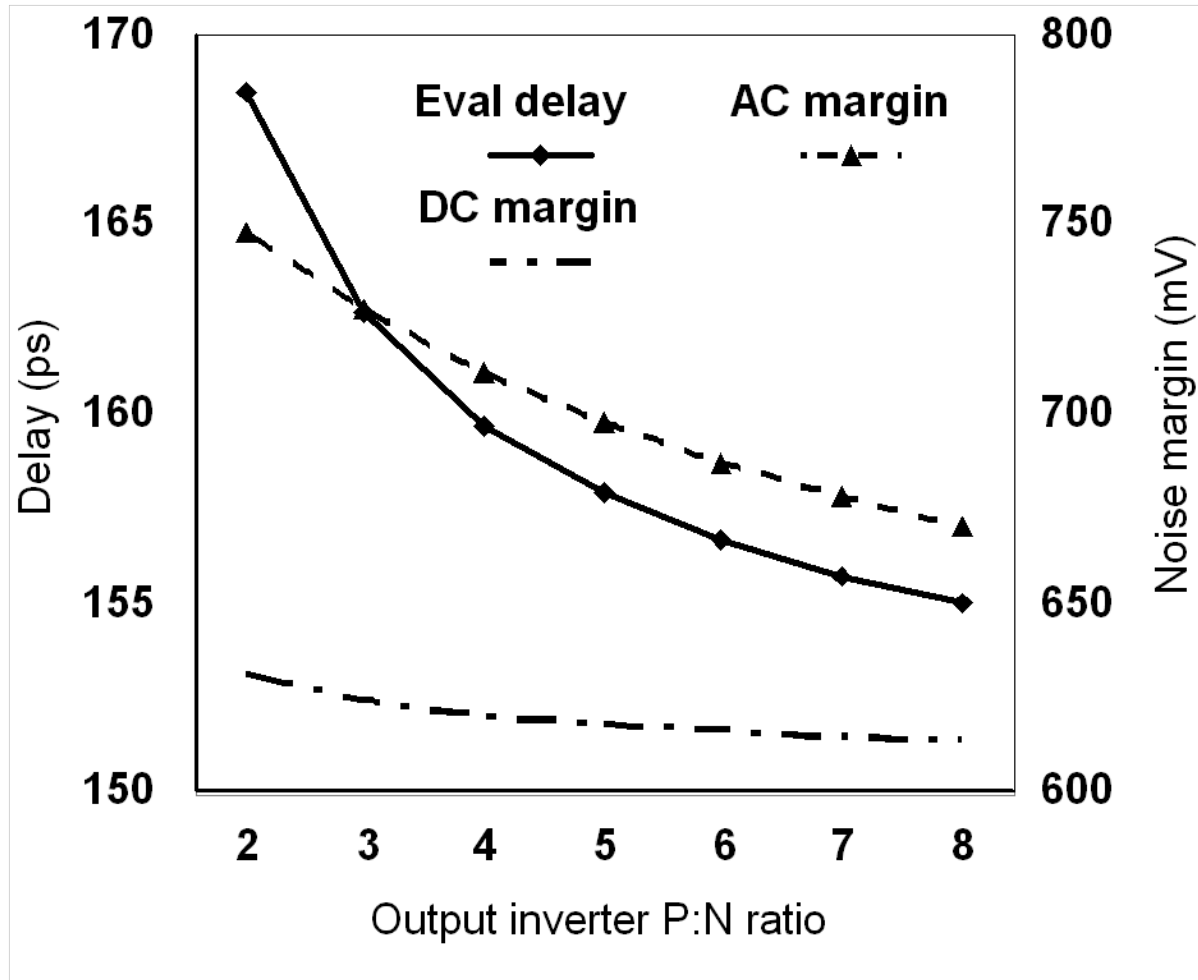
- multiple fets at a point?
- multi-fingered fets?
- long wire runs?
- Extract, analyze, and simulate!!

Robustness Optimization – Beta redux



No free lunch –
Big Beta => Lower Noise Margin

DC, AC Noise margin vs. Beta



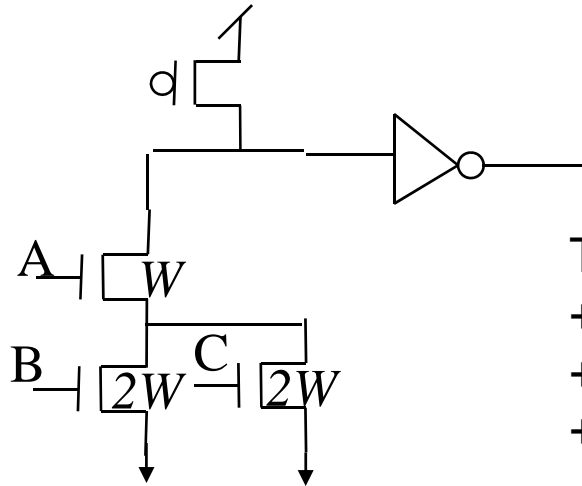
Source: Nowka, Galambos ICCD'98

Robustness Optimization -- topology

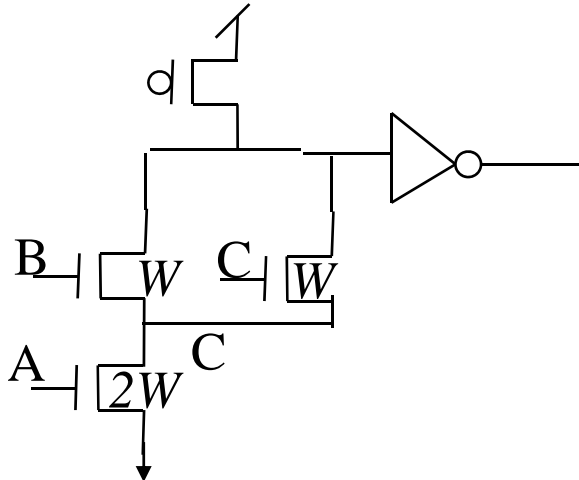
Topology changes:

- + to decrease leakage paths
- + to decrease charge sharing
- + to increase dynamic node cap
- + to change noise margin of output stage

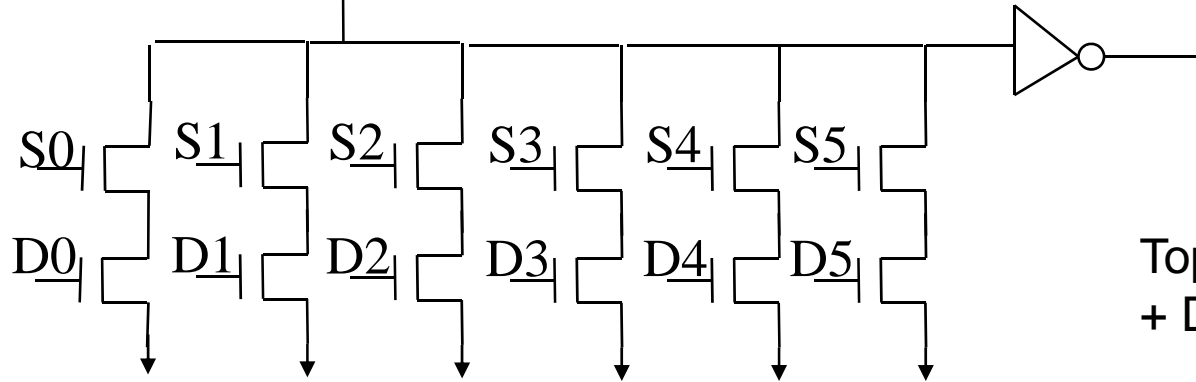
Robustness Optimization -- topology



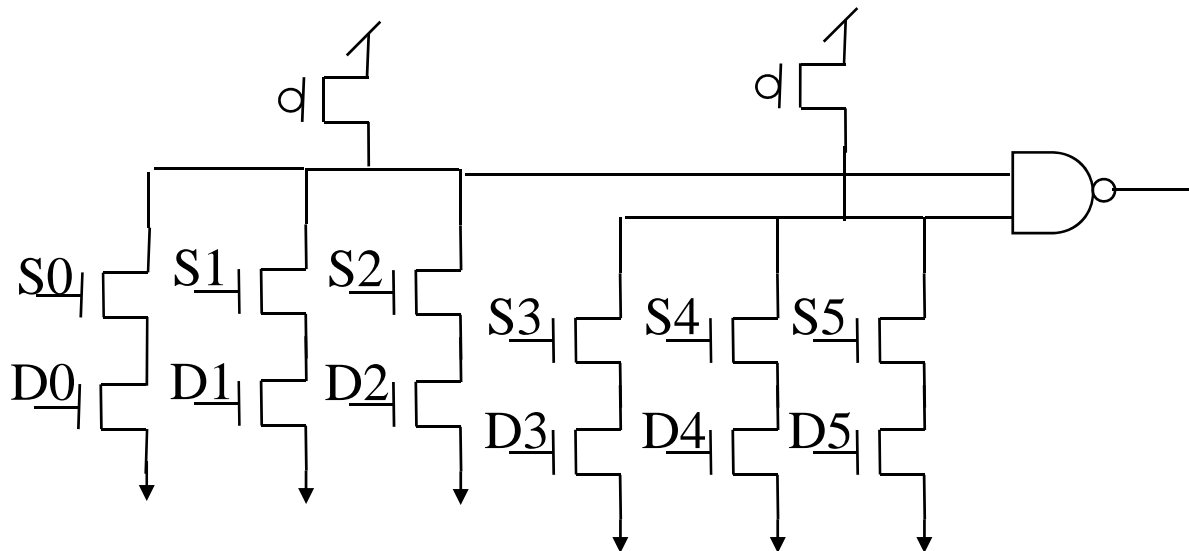
Topology change:
+ to decrease leakage paths
+ decreases charge sharing
+ increases dynamic node cap



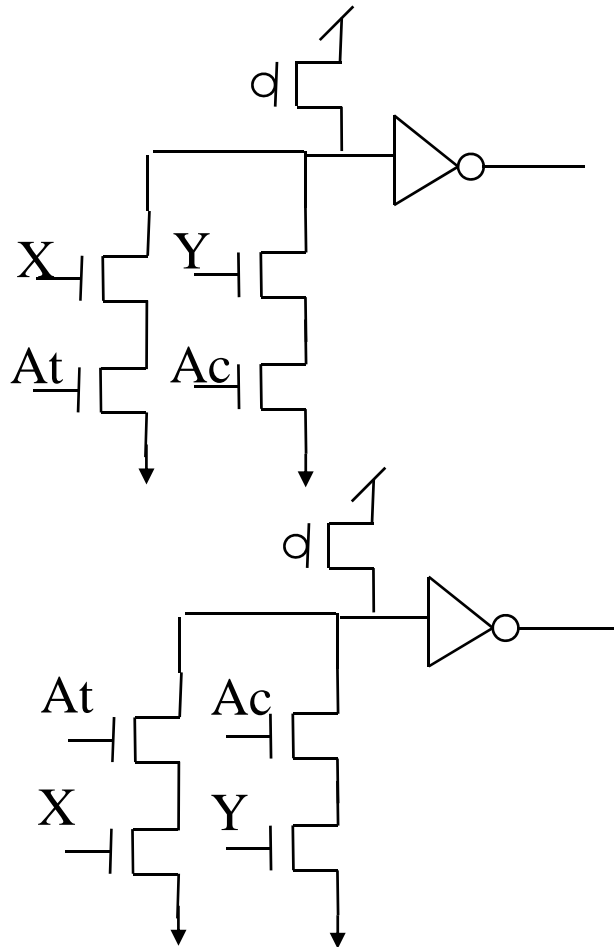
Robustness Optimization -- topology



Topology changes:
+ Decrease charge sharing
- or does it?
+ Minimize leakage paths
- or does it?

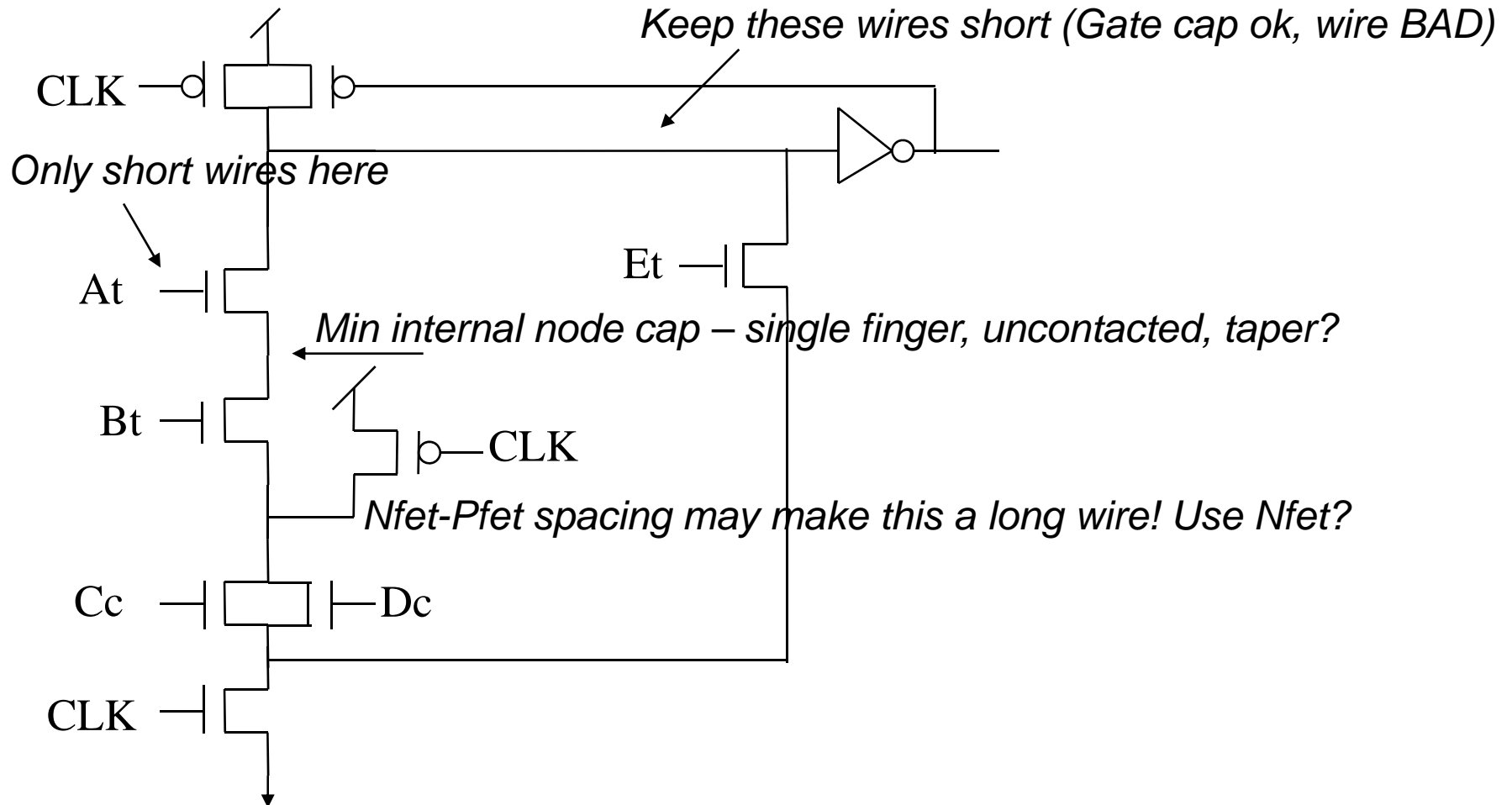


Robustness Optimization -- topology



Topology changes:
+ Decrease charge sharing

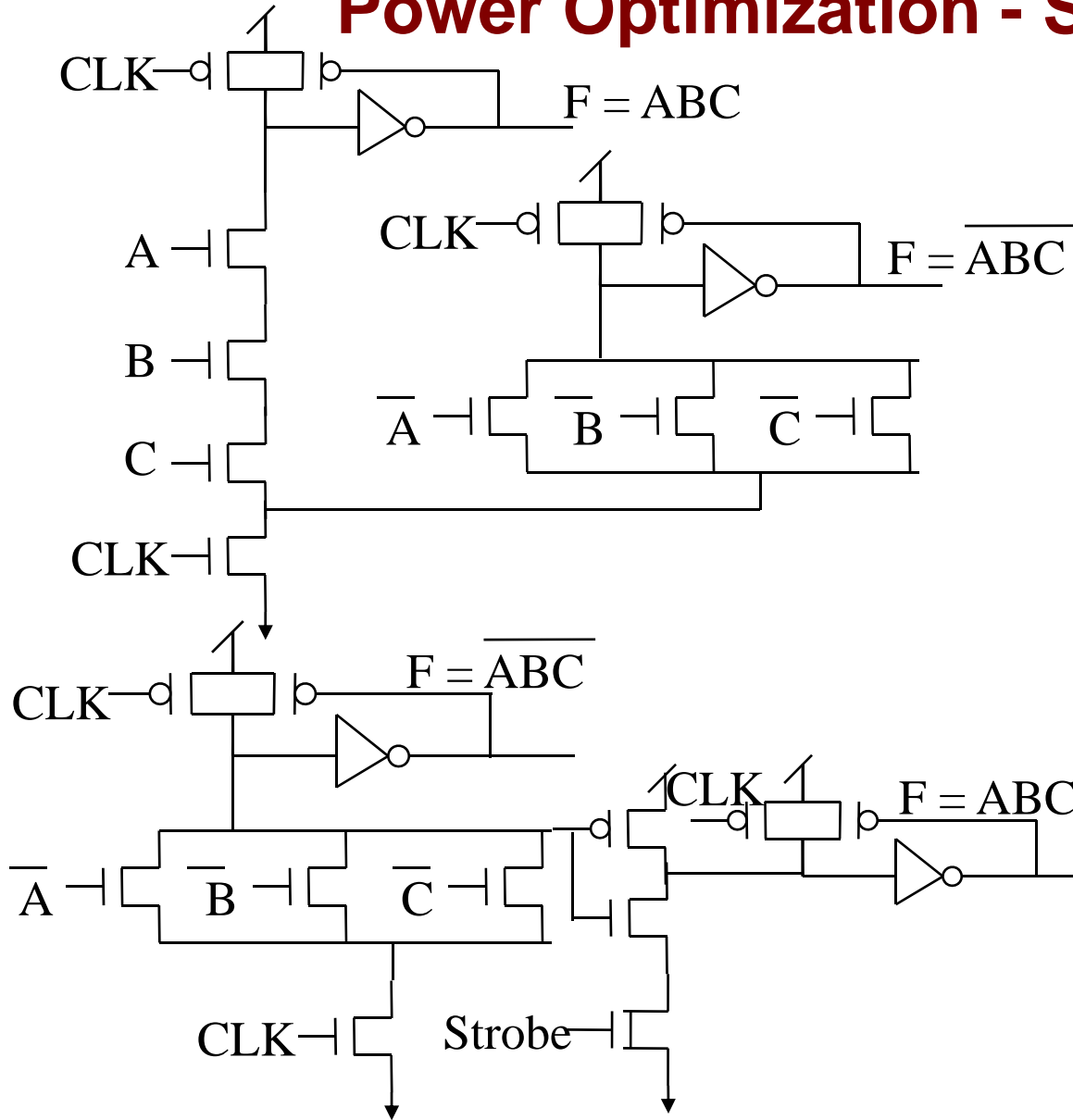
Robustness Optimization – layout considerations



Power (Energy!) Enhancements

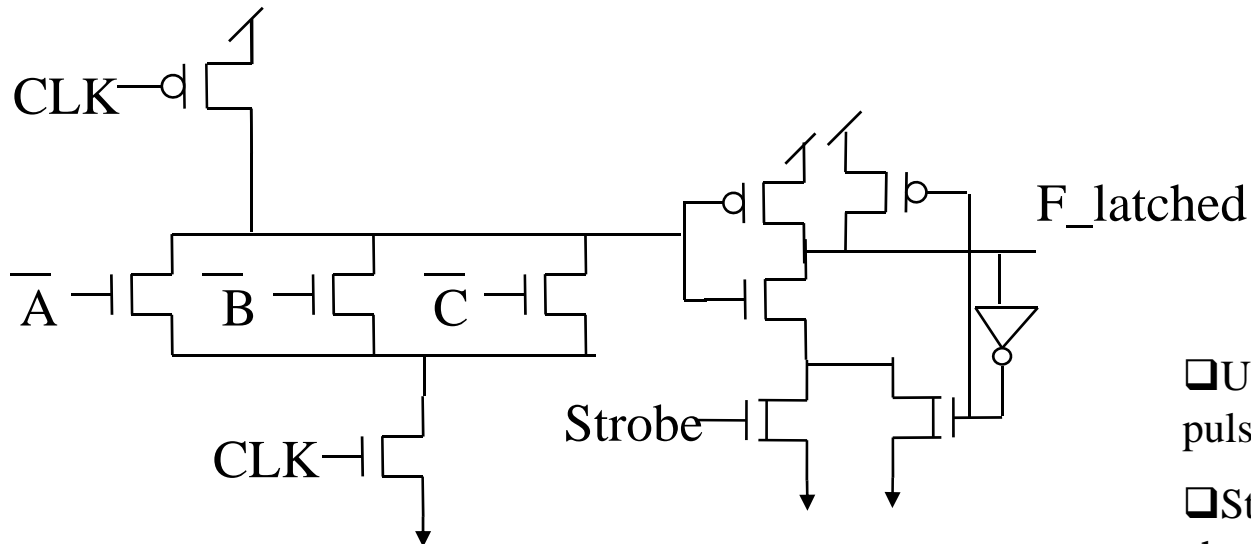
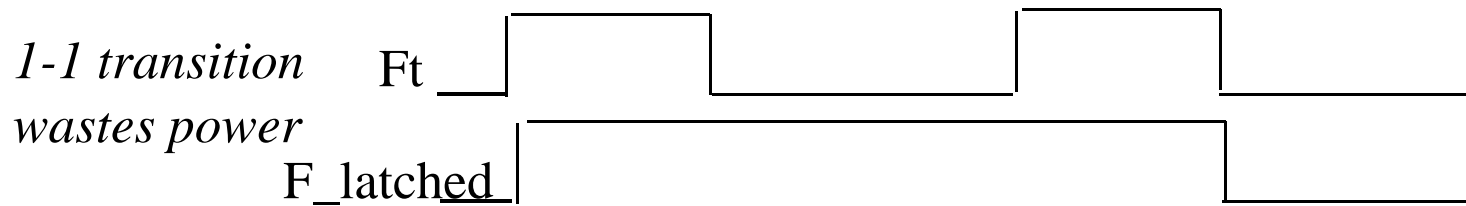
- **Strobing**
- **Latched output**
- **No foot**
- **No tapering**
- **Gated precharge**
- **MODL (already covered)**

Power Optimization - Strobing



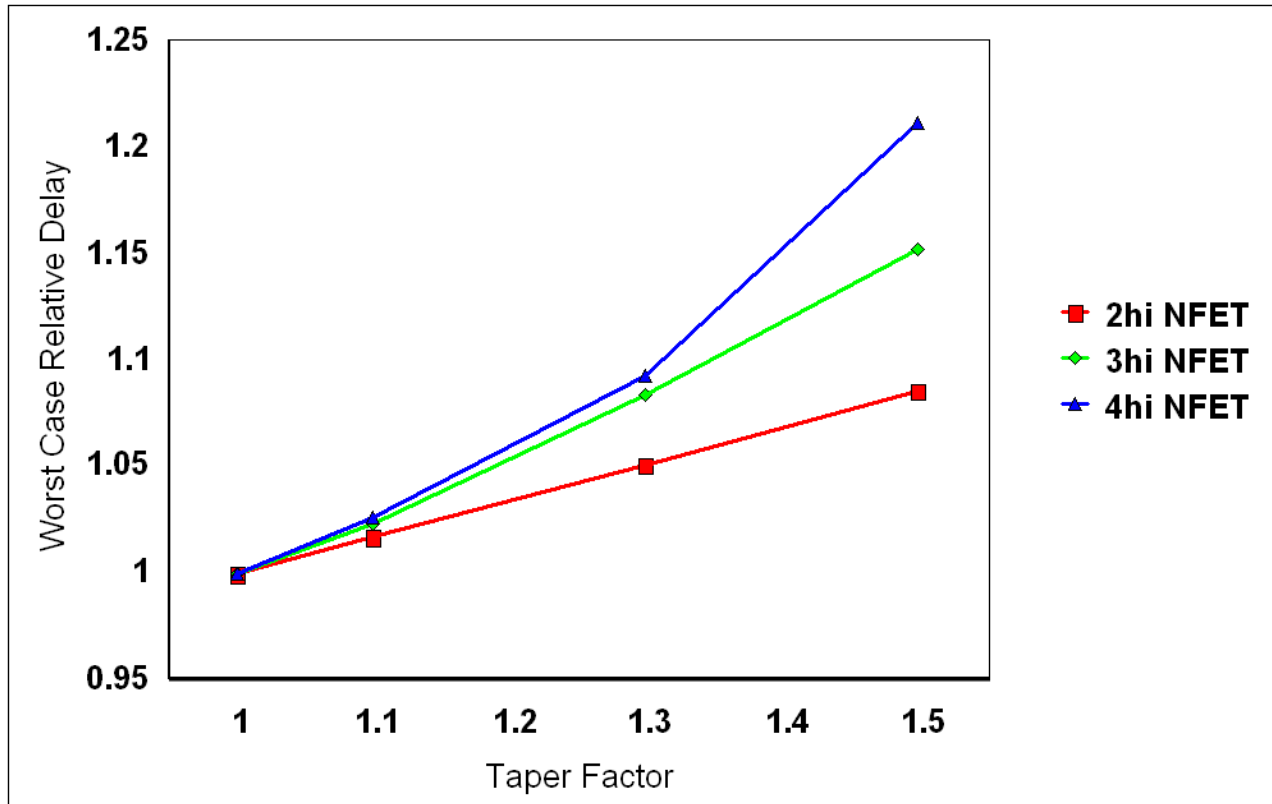
- ❑ Dual rail => large capacitance for two pulldown networks, 2 output wires
- ❑ Strobing – use a pulse to invert function
- ❑ Don't strobe too early!
- ❑ Generally slower than DR
- ❑ Do this where timing well controlled – memories, PLAs, latches

Power Optimization – Latched output



- ❑ Use a strobe to avoid 2 pulses on a 1-1 transition
- ❑ Strobe can be CLK or chopped CLK
- ❑ Don't latch a precharge

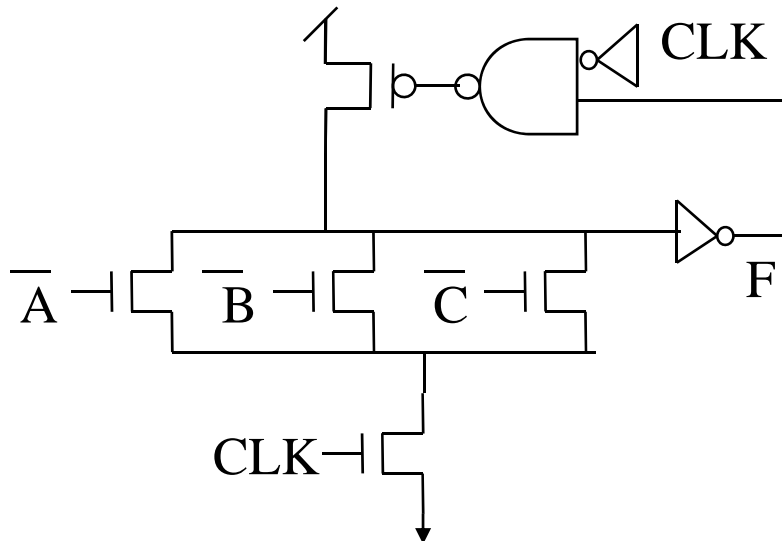
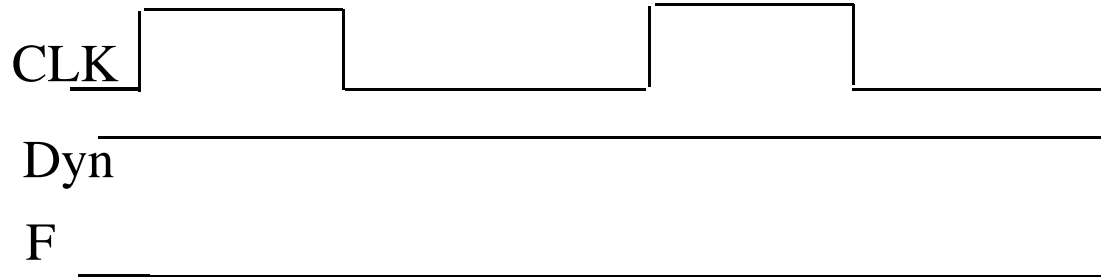
Performance vs. Tapering BACK



0.13um bulk CMOS process

Power Optimization – Gated precharge

*Precharge with
dynamic node hi
wastes power*

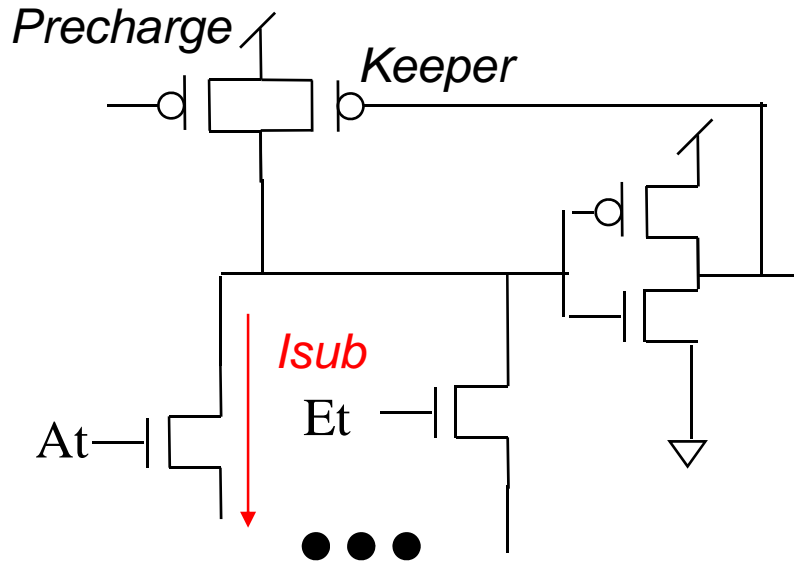


- ❑ Lots of added devices for very little savings
- ❑ Only useful in limited situations
- ❑ May need to buffer output F before the NAND

Effects of technology on dynamic circuits

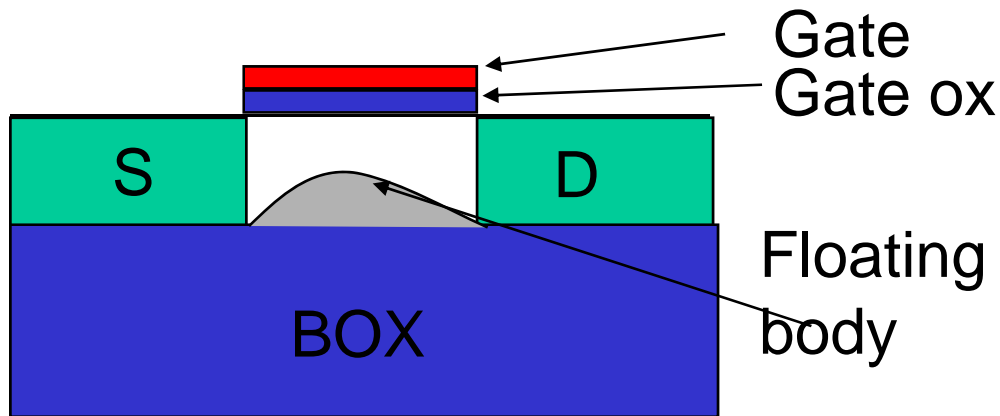
- Leakage and noise trends
 - Subthreshold
 - SOI and bipolar leakage
 - Gate leakage
- Low k inter and intra layer dielectrics, dual gate, active-well, hi-V_ts and lo-V_ts,

Subthreshold leakage

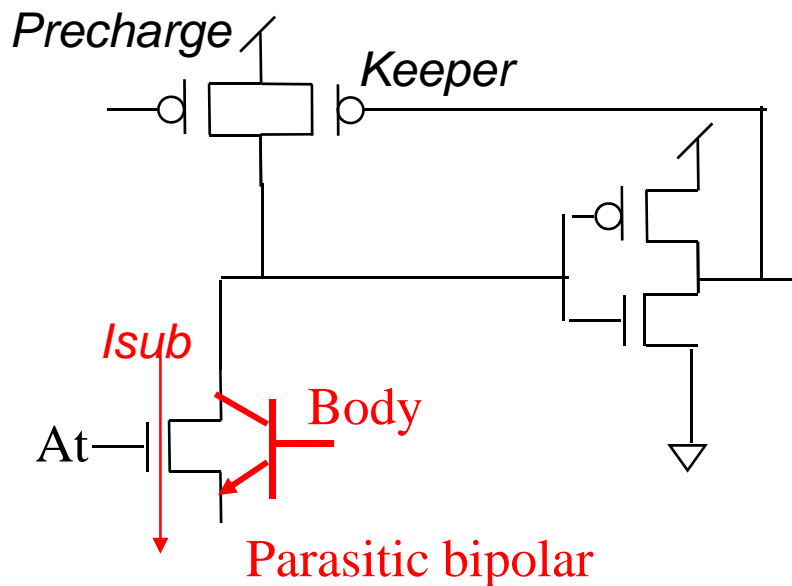


- **Subthreshold trends**
 - V_t decreasing, but not according to scaling theory
 - V_{dd} decreasing
 - Traditional 5-order magnitude $I_{on}:I_{off}$ degrading to 3-4 O.M.
 - Subthreshold really an issue for low- V_t s and SOI

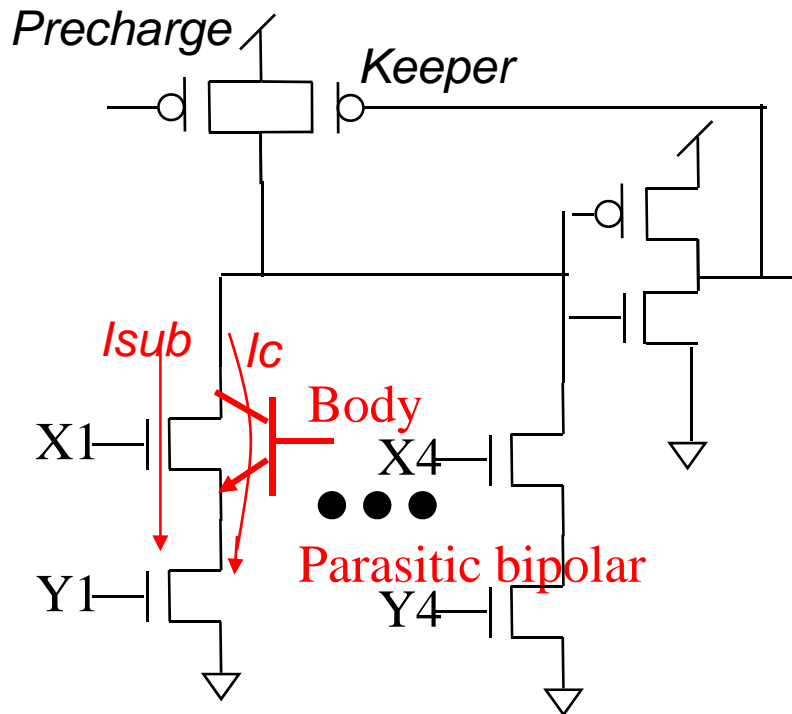
Partially depleted SOI and leakage



- Floating body collects charge due to impact ionization at drain.
- Voltage on body is potential on base terminal of bipolar
- Voltage on body also lowers (or raises) V_t of MOSFET
- Increased leakage!!



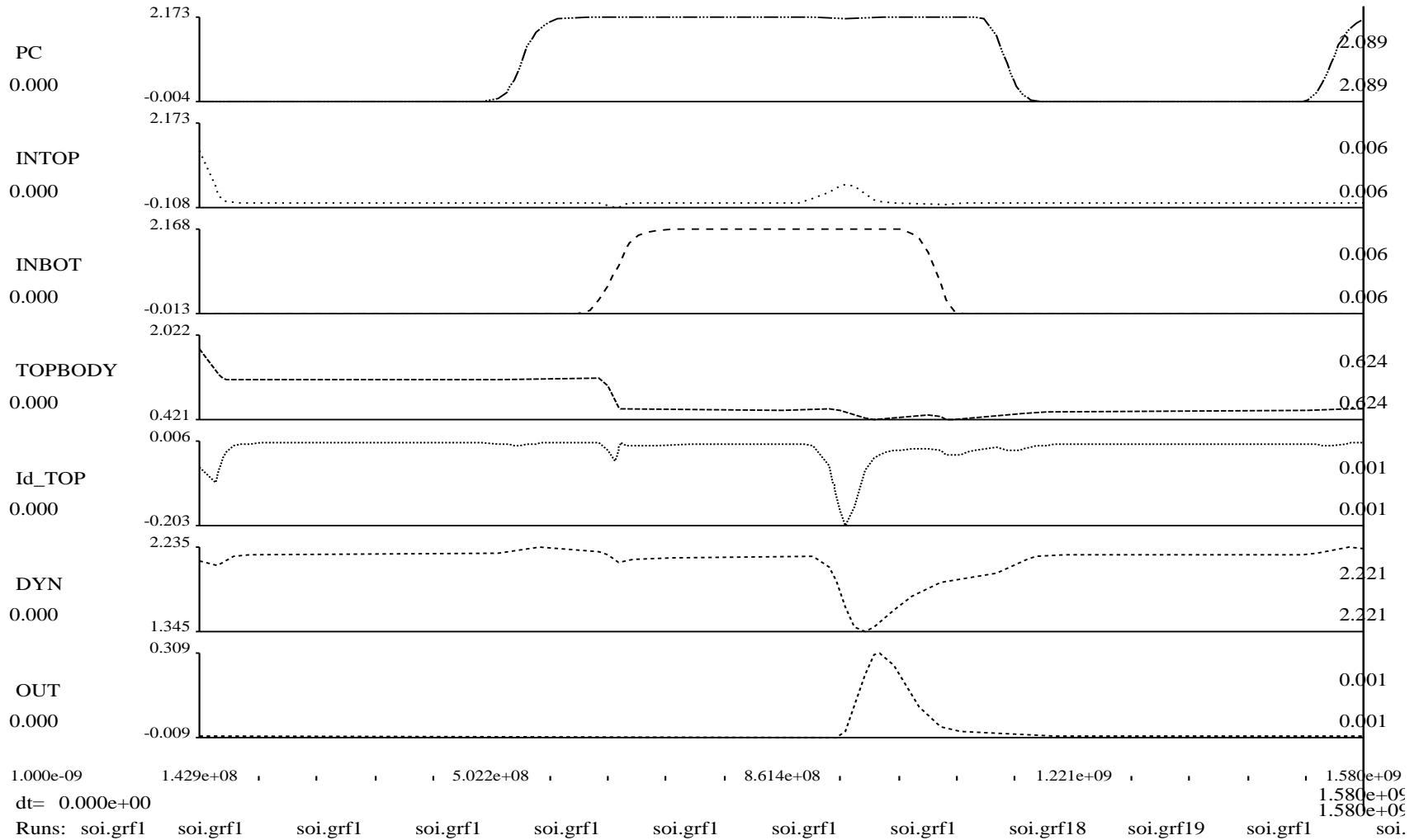
Effects of floating body on 2-and-4-or gate



1. X1..X4 high, Y1..Y4 low
 - MOS V_t falls
2. Bodies of T_{x1} to T_{x4} float high
 - Bipolar current flows
3. X1..X4 low, Y1..Y4 high
 - MOS subthreshold leakage
4. Noise on X1..X4
 - MOS subthreshold leakage

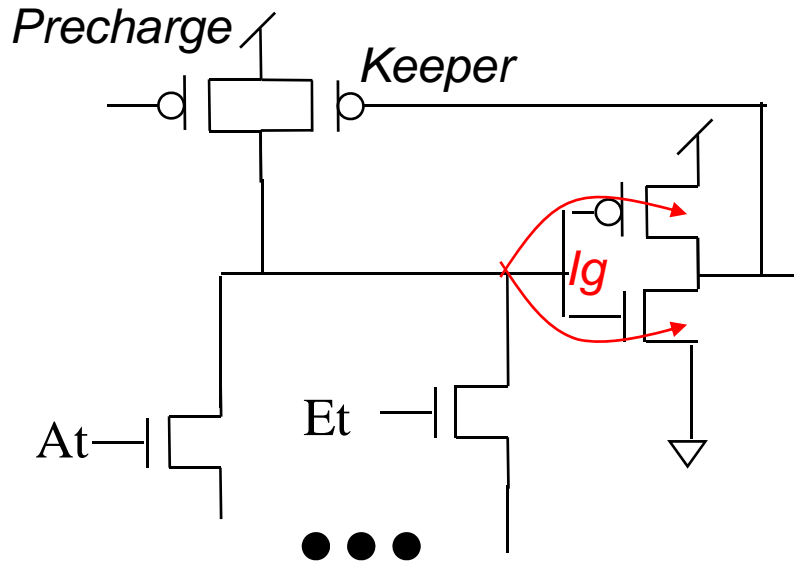
Increase keepers, change topologies, predischarge intermediate nodes...

Waveform of this scenerio



Source: CRC Computer Engr. Handbook, Ed. Oklobdzija 2002

Gate leakage



- Below 2.2 nm gate oxides gate oxide tunneling becomes a MAJOR issue.
- Unlike subthreshold, bipolar leakage, not much we can do:
 - Keep T_{ox} thicker
 - High k oxide materials
 - Lower voltage
- This is especially a thorn for dynamic circuits

Other technology trends

- **Low k dielectrics in wiring => decrease in coupled noise at inputs and dynamic nodes**
- **Body biases, well biases, dual gate => can be used for increased performance or to lower leakage**
- **Multiple V_{ts} => more options in performance vs. power/noise/leakage**

Domino logic: Summary

1. Numerous ways to improve the performance of dynamic
2. Numerous ways to improve the robustness of dynamic
3. Numerous ways to improve the power consumption of dynamic
4. 1-3 not easily mutually satisfied....
5. HIGH PERFORMANCE domino is hand-crafted to avoid noise and power problems as well as get performance
6. Dynamic logic which is not high-performance doesn't have a significant advantage over simpler static design.
7. Thus, today domino is largely full-custom, but many are trying to automate the design process.
8. Technology trends are a little disturbing – even more challenging