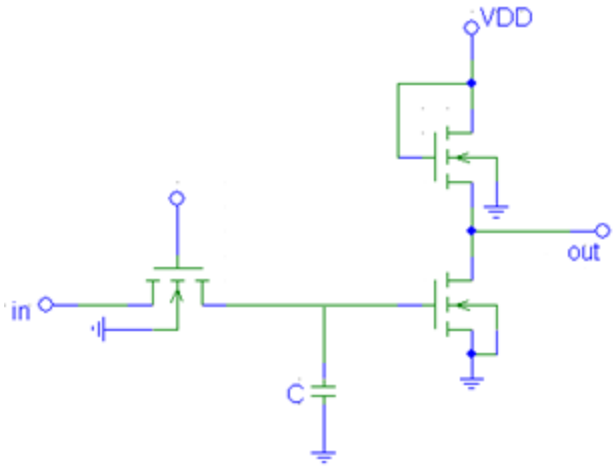
گیت های دینامیک CMOS

Nasser Mozayani
School of Computer Engineering
Iran University of Science and Technology

عملکرد گیتهای دینامیک

- وقتی تاخیر گیتها با اهمیت شده و مساحت سیلیسوم بار اضافه تلقی می شود به سراغ گیتهای دینامیک می رویم
- معمولا حدود ۲ برابر سریعتر و مساحت کمتر دارند
- ولی مصرف آنها می تواند کمتر یا بیشتر باشد

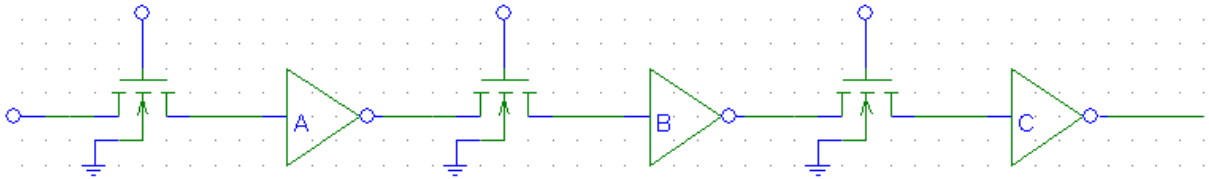
البته چون فشرده سازی بیشتر است مصرف بیشتر است



- عملکرد بر اساس پیش شارژ و ارزیابی است
- به گره ای که بتواند یک سطح منطقی را

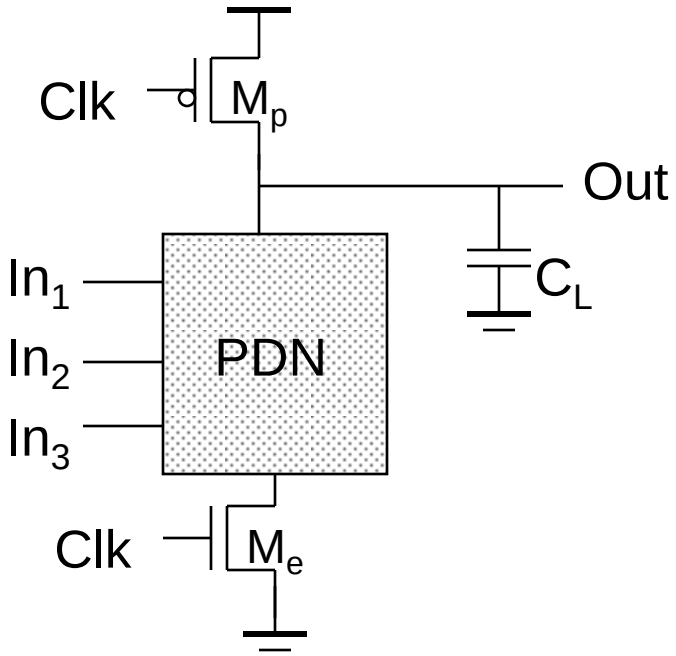
به وسیله ی ذخیره ی بار در خود نگه دارد

Soft Node گویند



- طراحی گیتهای بزرگ بصورت سنکرون است پس مدار دینامیک سرعت بهتری دارد

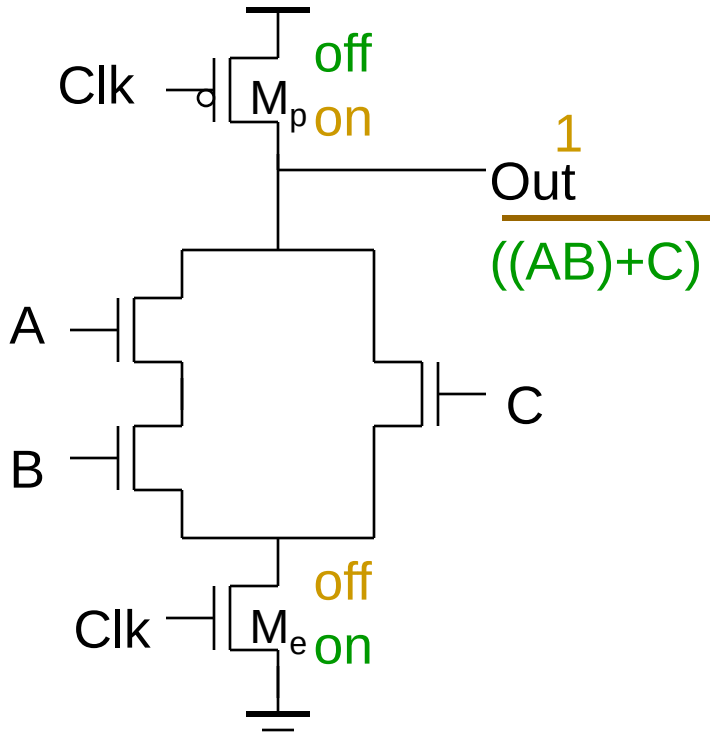
Dynamic Logic Gates



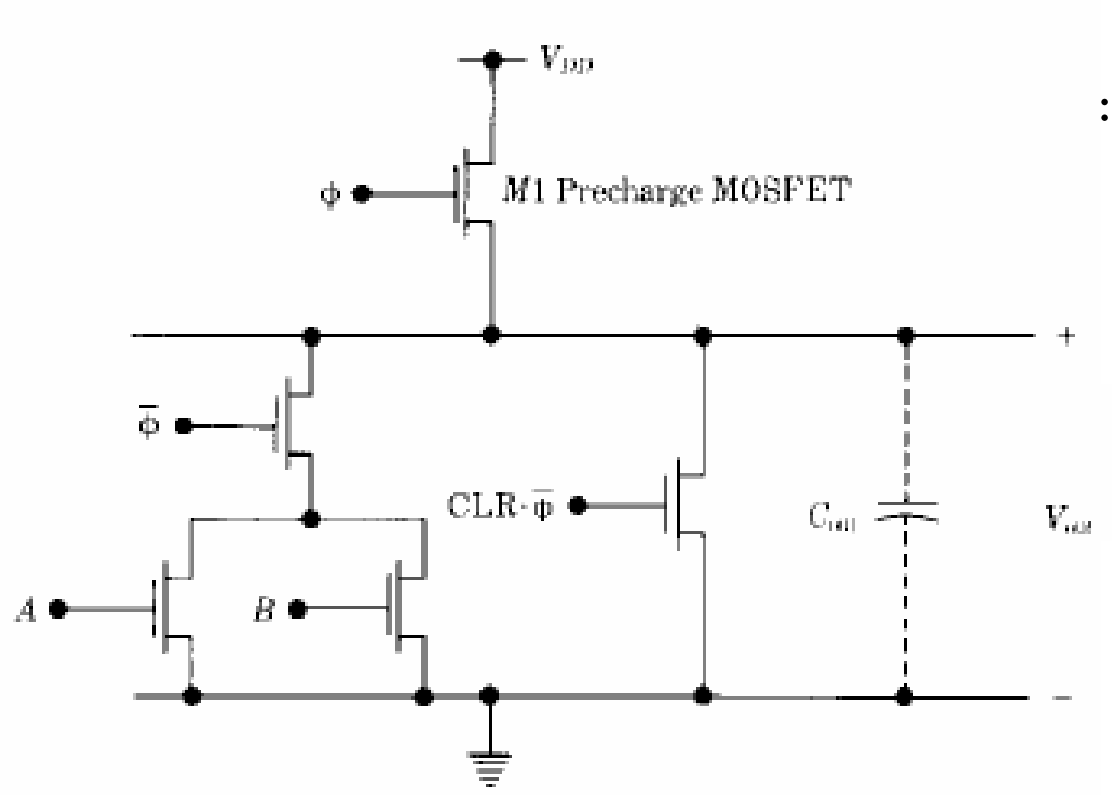
Two phase operation

Precharge (Clk = 0)

Evaluate (Clk = 1)



precharging



- افزایش سرعت سوئیچینگ:

تا خیر انتشار نزولی از

رابطه زیر تعیین می شود

$$t_{HL} \approx \frac{2.94C_{out}}{\beta_D(V_{DD} - V_{TD})},$$

و تاخیر انتشار صعودی

(که کند بود) حل می شود

- و کاهش مصرف توان (مسیر مستقیم بین زمین و Vdd اصلا وجود ندارد)

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 2*n* (*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 *n* + 2 (*n*+1 N-type + 1 P-type) transistors

Conditions on Output

- Once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.

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

Properties of Dynamic Gates

- Logic function is implemented by the PDN only
  - number of transistors is N + 2 (versus 2N for static complementary CMOS)
- Full swing outputs (V_{OL} = GND and V_{OH} = V_{DD})
- Non-ratiod - 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 (C_{out})
  - no I_{sc}, so all the current provided by PDN goes into discharging C_L

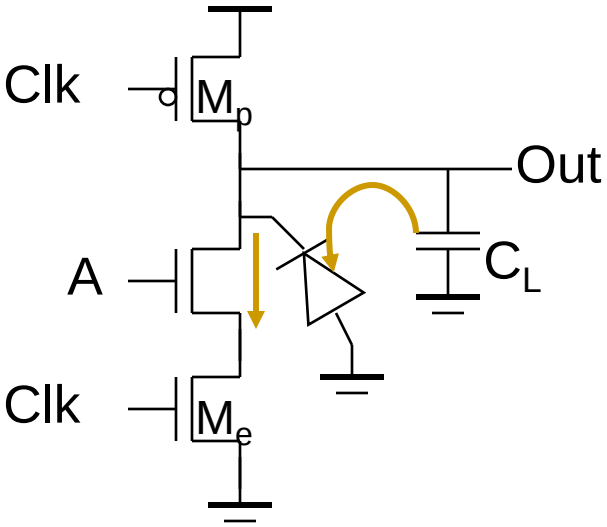
Properties of Dynamic Gates

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

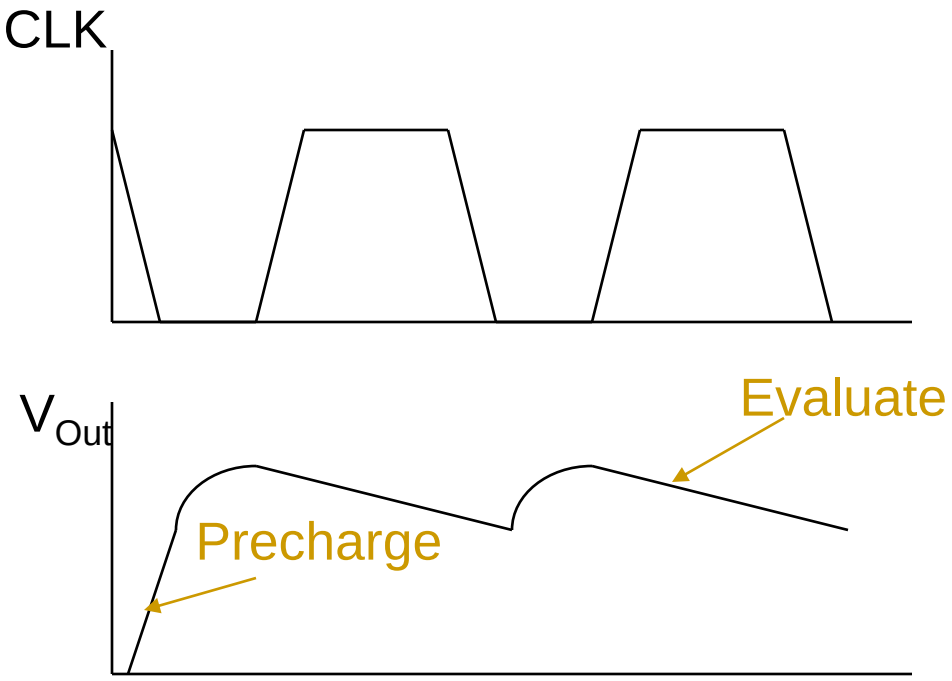
Dynamic CMOS

- Advantages over static logic:
  - Avoids duplicating logic twice as both N-tree and P-tree, as in standard CMOS
  - The input capacitance is only that of the N device
  - Typically can be used in very high performance applications
  - Very simple sequential memory circuits; amenable to synchronous logic
  - High density achievable
  - Consumes less power (in some cases)
- Disadvantages compared to static logic:
  - Problems with clock synchronization and timing
  - Design is more difficult

Issues in Dynamic Design 1: Charge Leakage

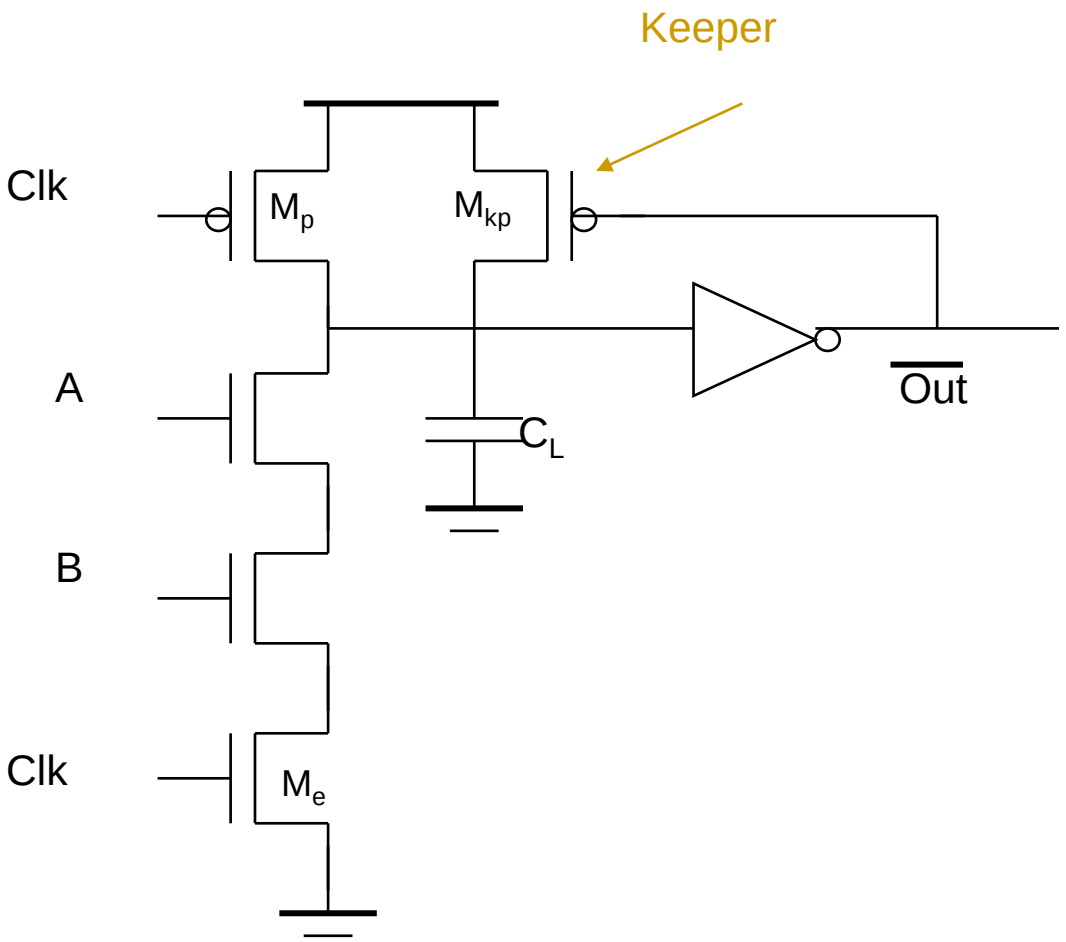


Leakage sources



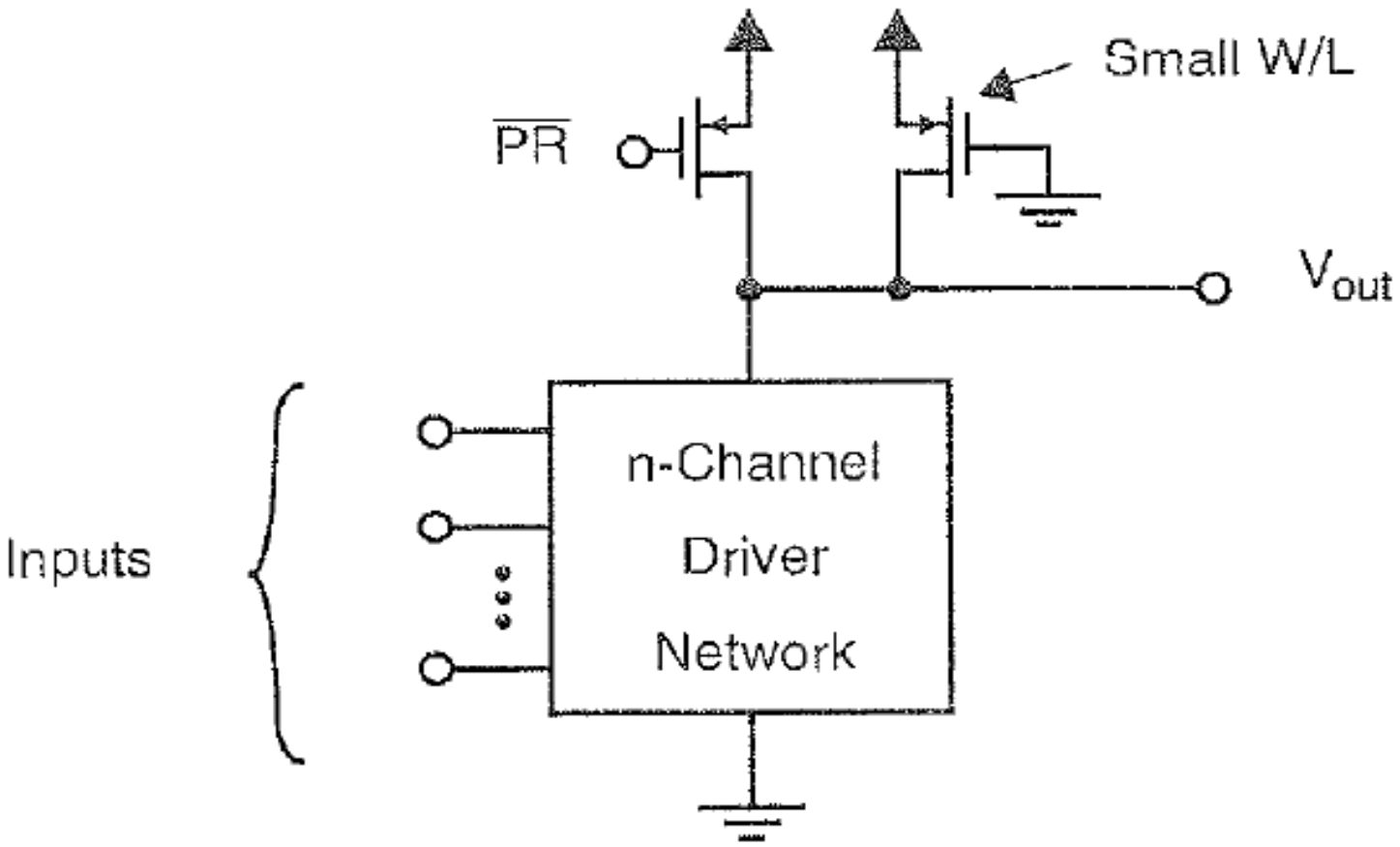
Dominant component is subthreshold current

Solution to Charge Leakage

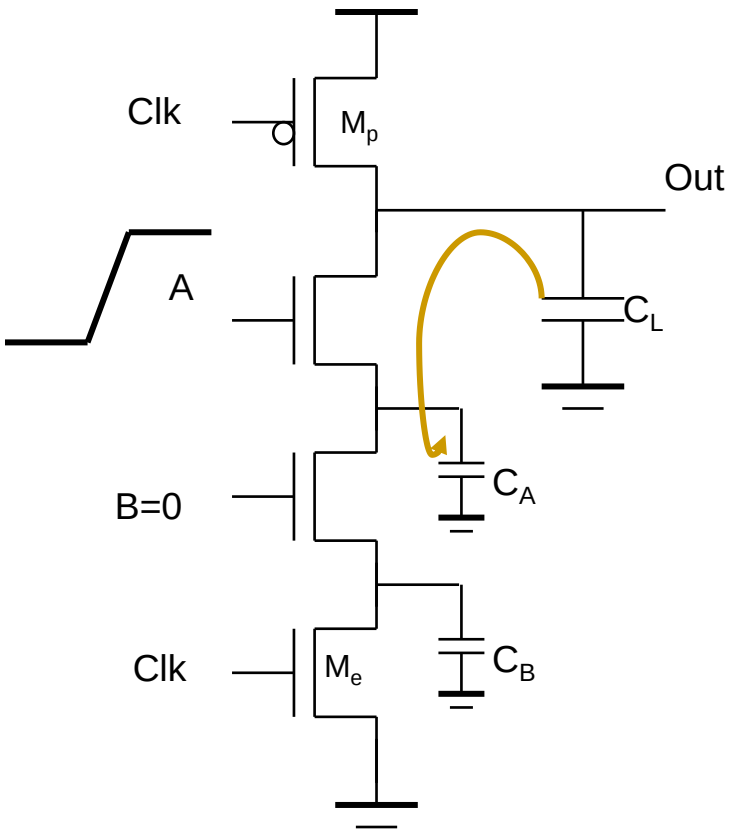


Same approach as level restorer for pass-transistor logic

Solution 2 to Charge Leakage

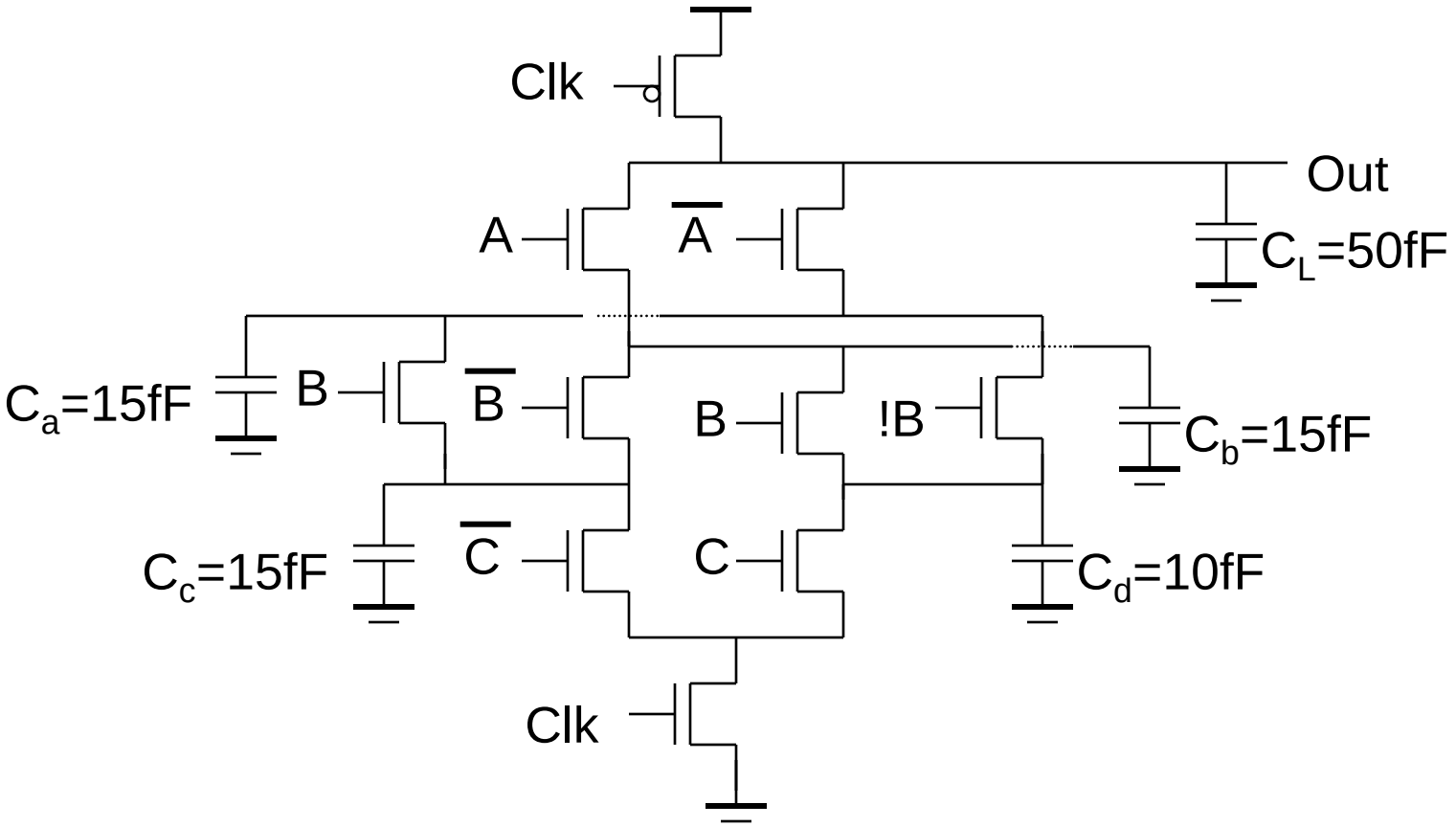


Issues in Dynamic Design 2: Charge Sharing

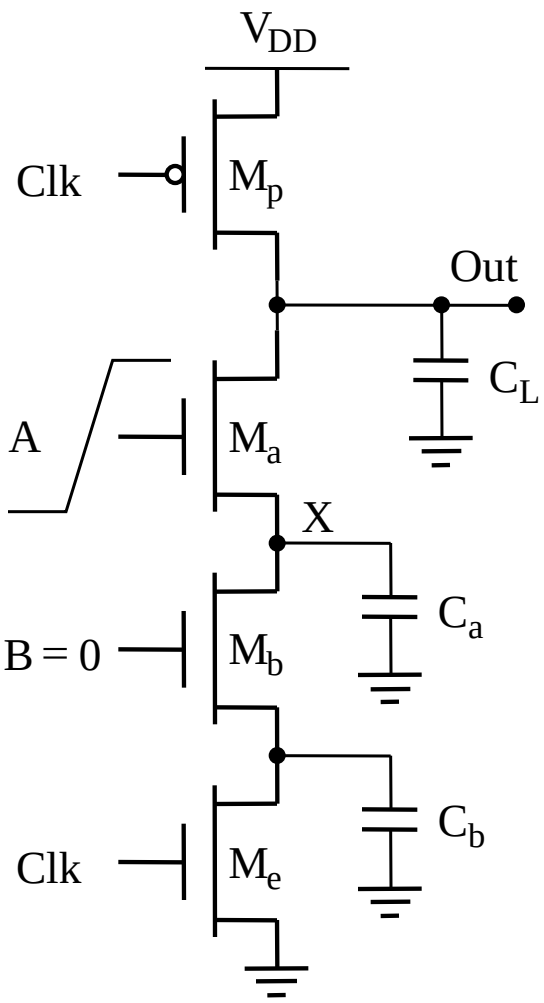


Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to reduced robustness

Charge Sharing Example



Charge Sharing



case 1) if $\Delta V_{out} < V_{Tn}$

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

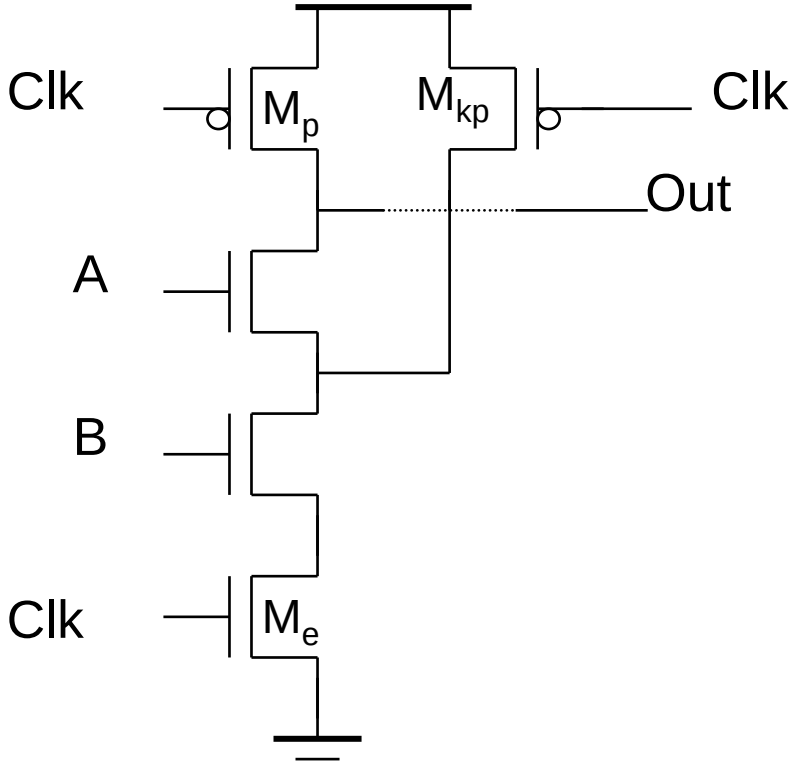
or

$$\Delta V_{out} = V_{out}(t) - V_{DD} = -\frac{C_a}{C_L}(V_{DD} - V_{Tn}(V_X))$$

case 2) if $\Delta V_{out} > V_{Tn}$

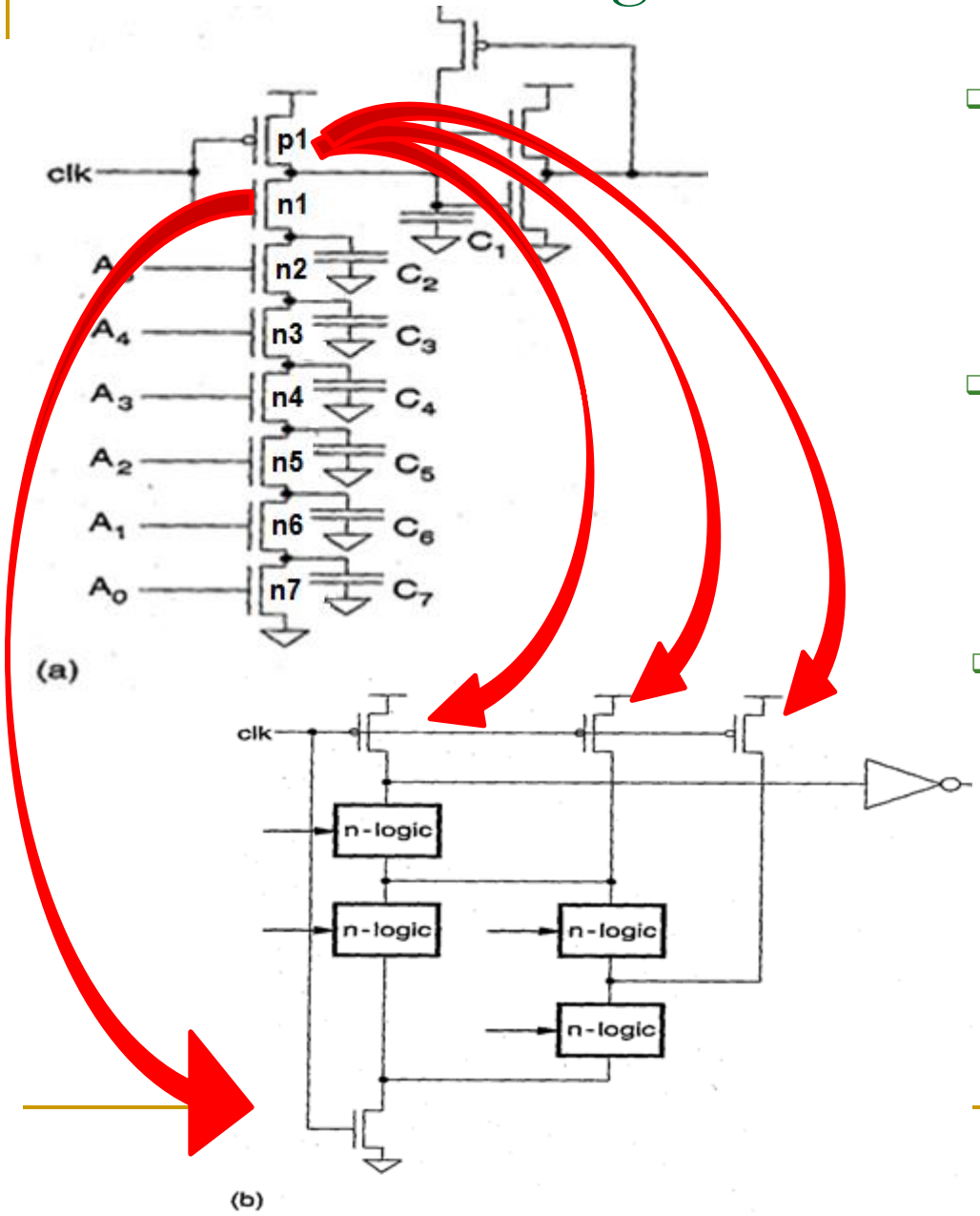
$$\Delta V_{out} = -V_{DD}\left(\frac{C_a}{C_a + C_L}\right)$$

Solution to Charge Redistribution



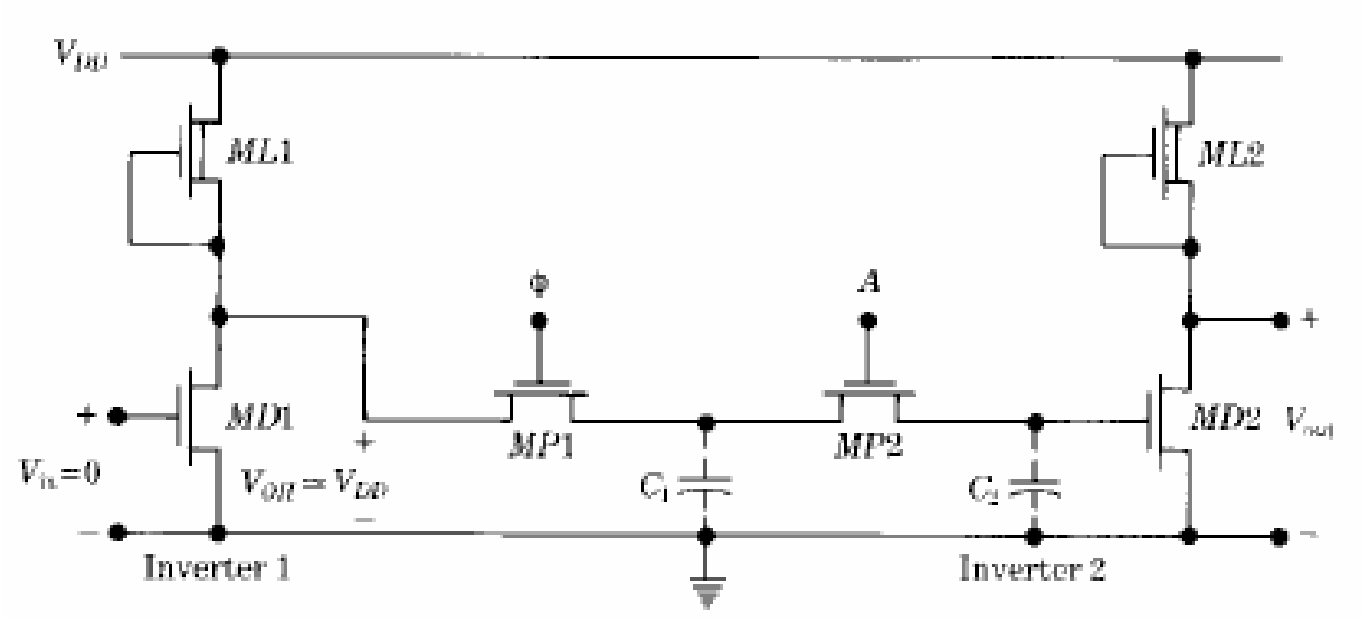
Precharge internal nodes using a clock-driven transistor (at the cost of increased area and power)

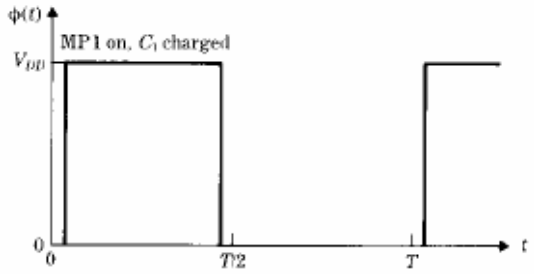
Solution to Charge Redistribution

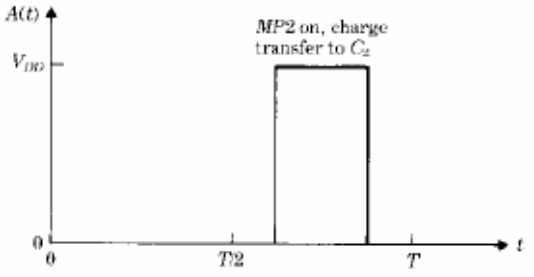


- During precharge C1 is charged high to Vdd, but C2-C7 do not get charged and may be sitting at ground potential.
- When the clock goes high for the evaluate phase: VC1 may reduce to Vdd(C1/(C1 + C2 + C3 + C4 + C5 + C6 + C7)) in the worst case
- The solution is to put the discharge transistor N1 at the bottom of the logic tree and reapeat P1 for every C2 to C7 thus allowing the possibility of getting C2-C7 charged during the precharge phase

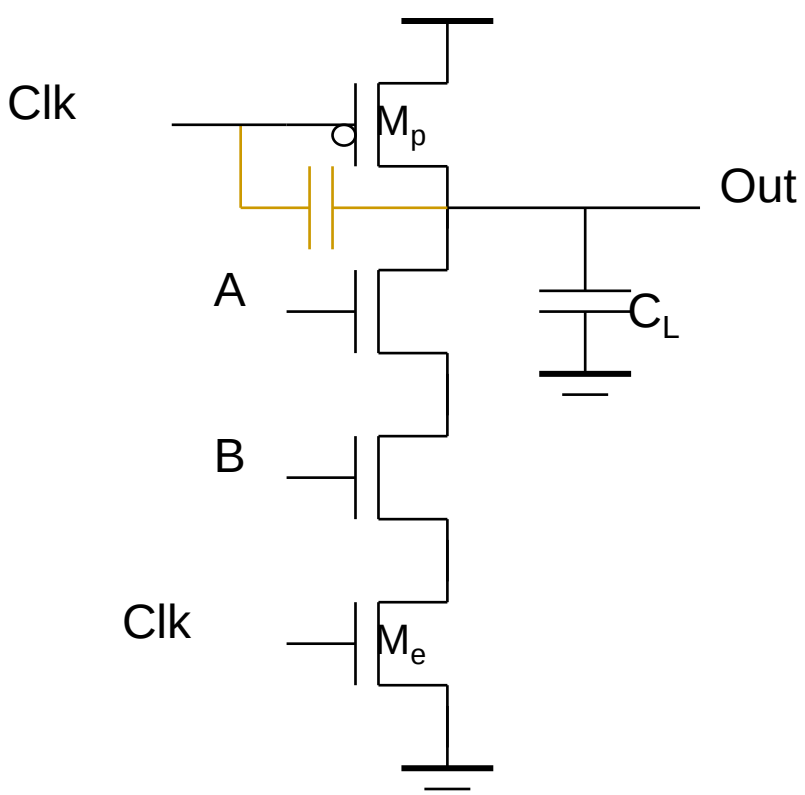
charge sharing in pass transistors







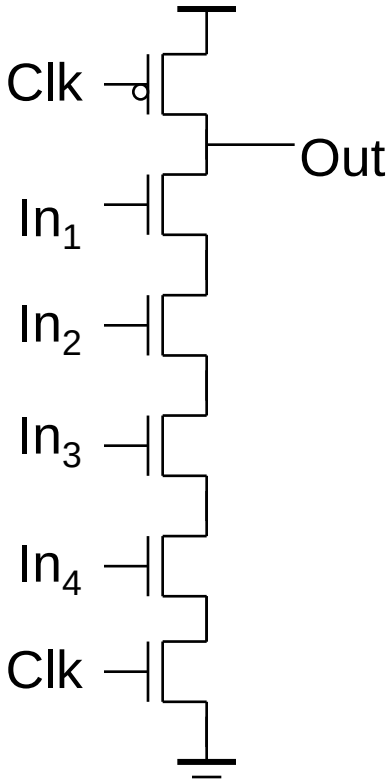
Issues in Dynamic Design Clock Feedthrough

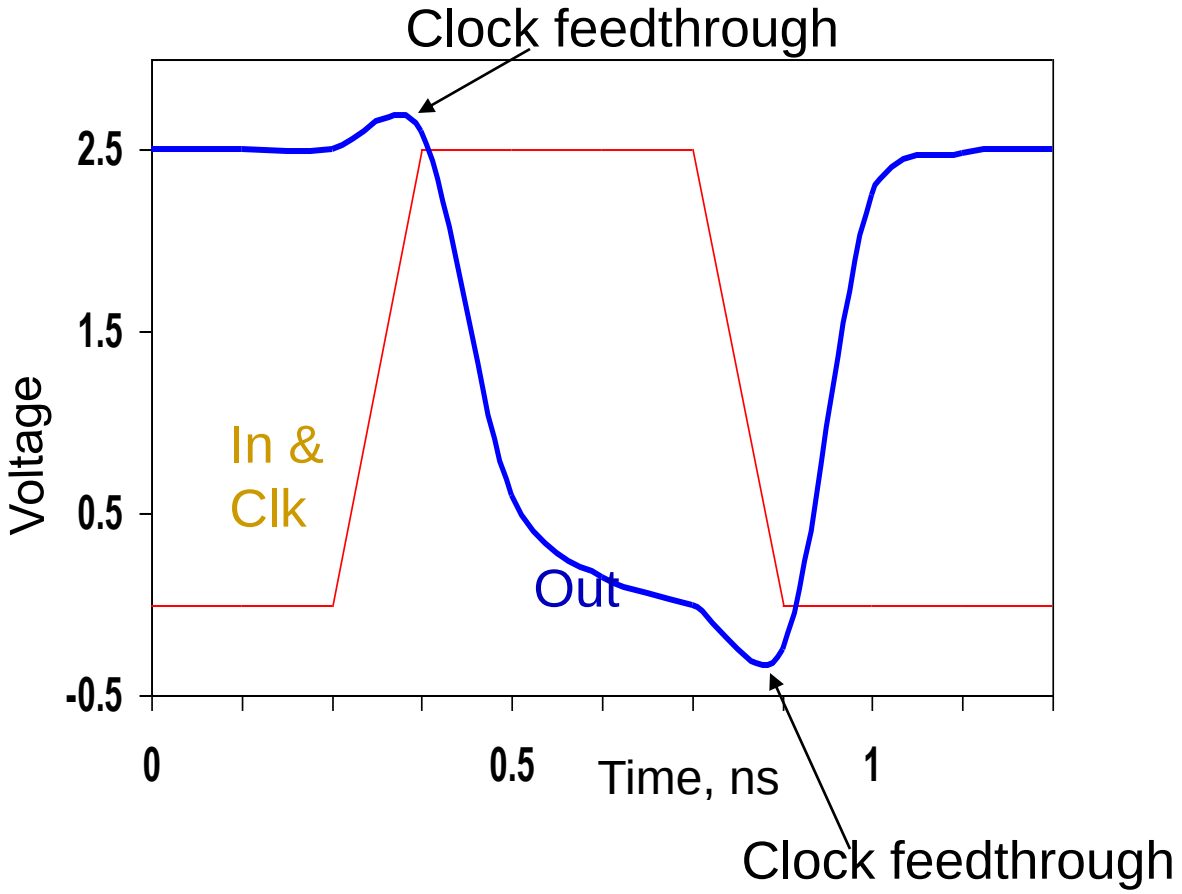


Coupling between Out and Clk input of the precharge device due to the gate to drain capacitance. So voltage of Out can rise above V_{DD}. The fast rising (and falling edges) of the clock couple to Out

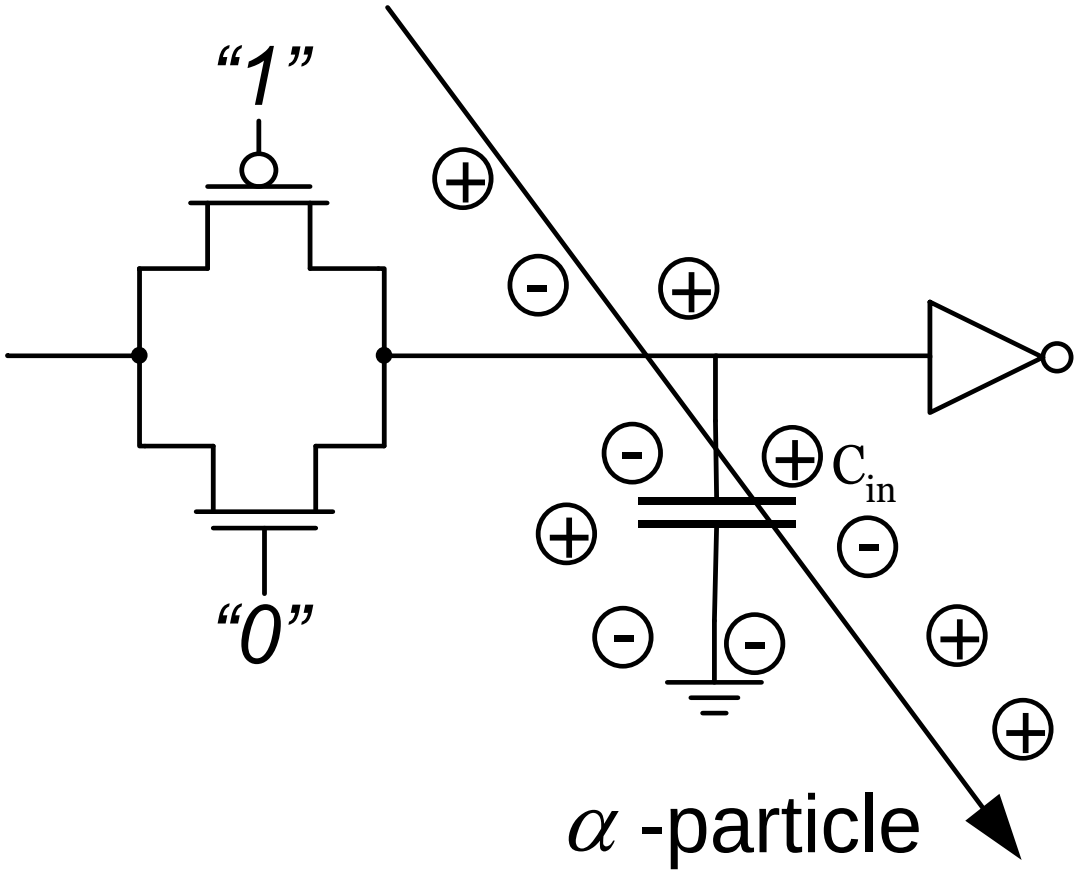
Danger is that signal levels can rise enough above VDD that the normally reverse-biased junction diodes become forward-biased causing electrons to be injected into the substrate.

Clock Feedthrough



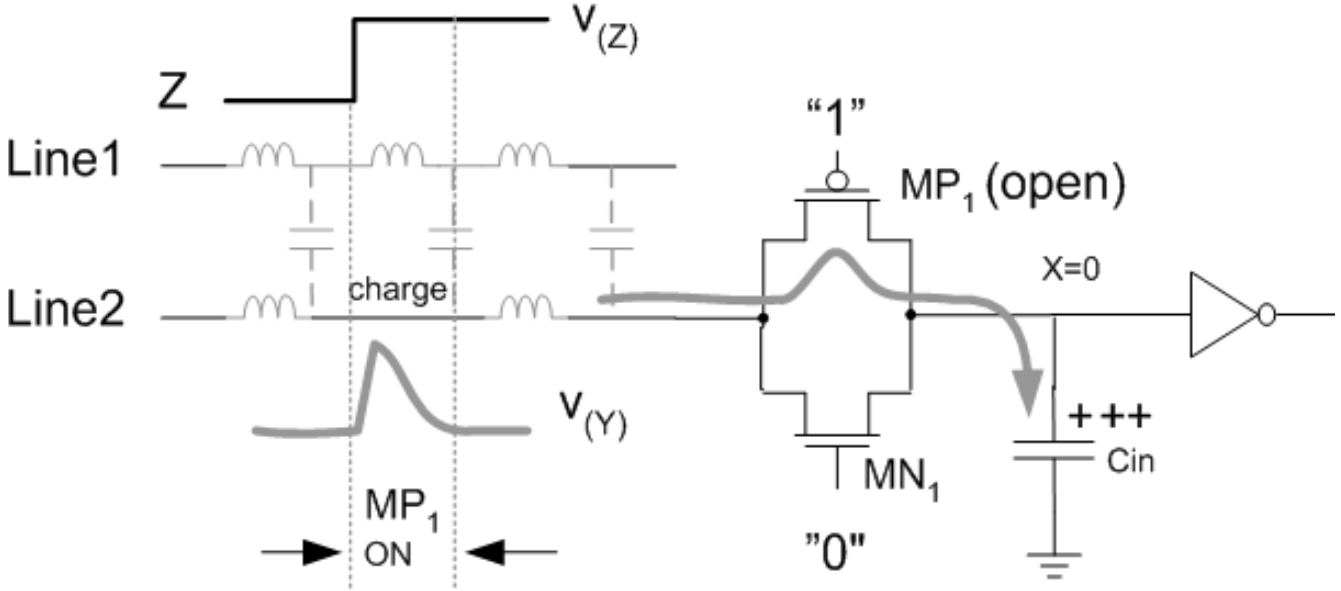


Radiation induced charge

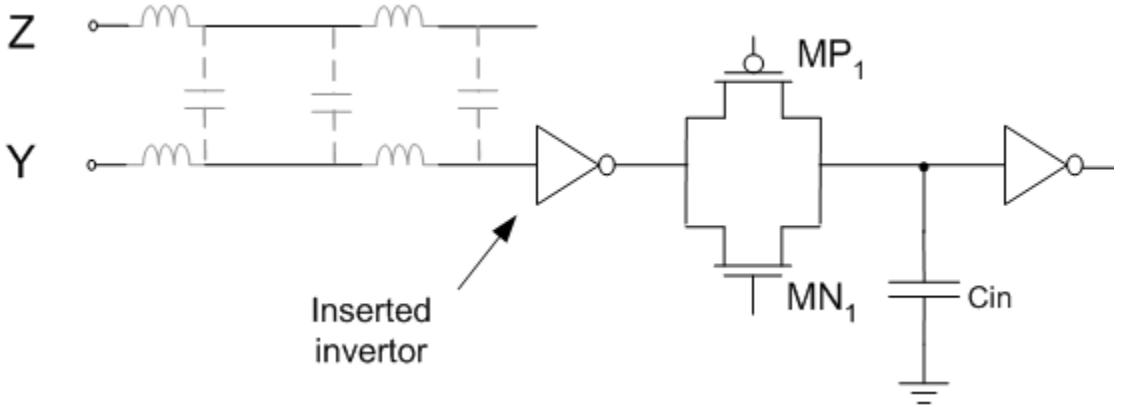


Crosstalk

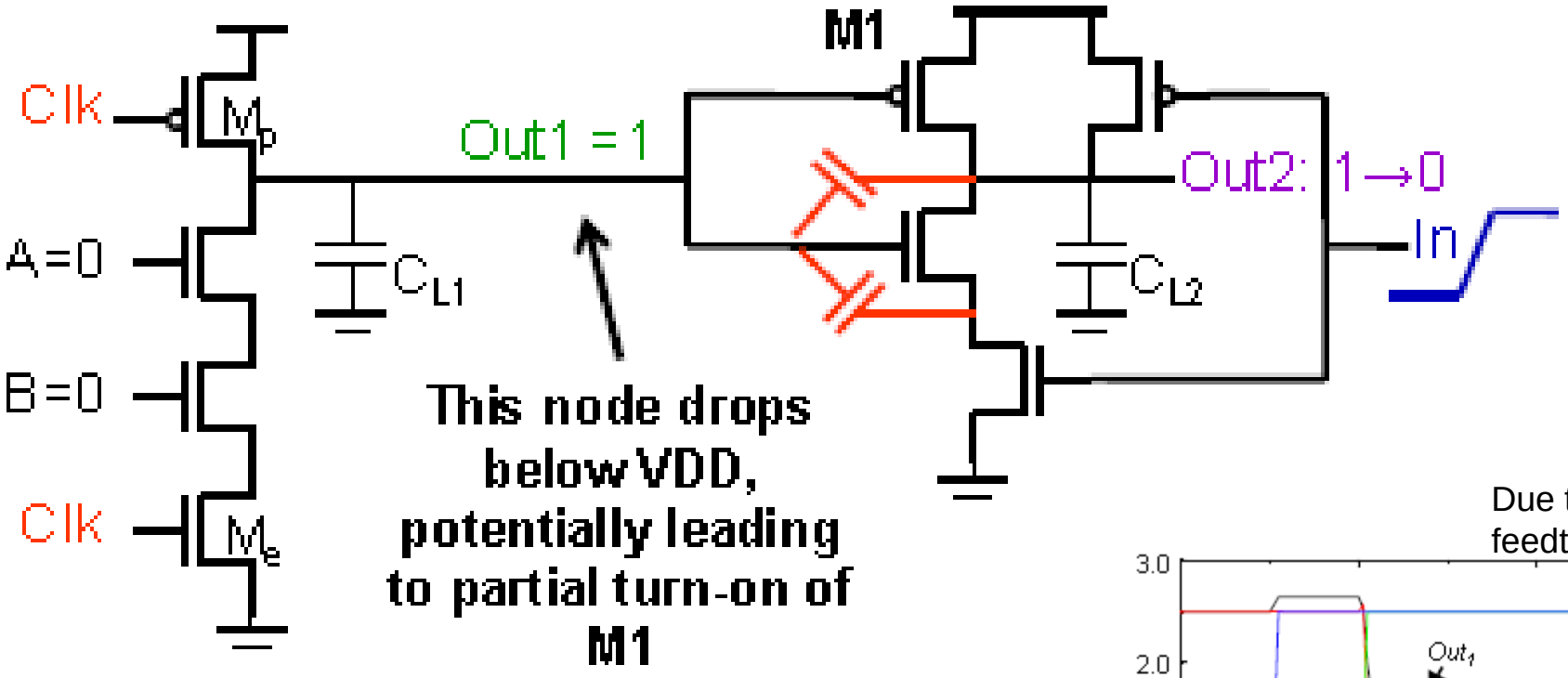
Accidental charge caused by capacitive or inductive coupling between the signal lines Y and Z. (a)



Prevention by inserting and inverter between the affected line and the pass-transistor switch (b)

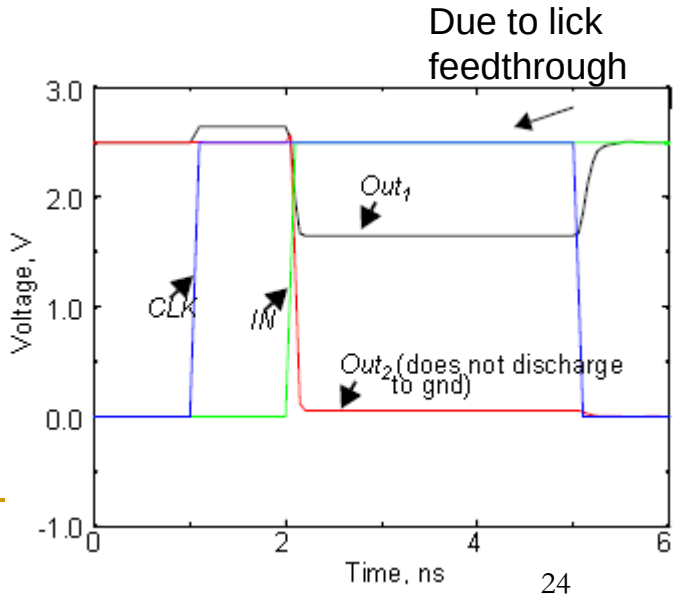


Backgate coupling (capacitive coupling)

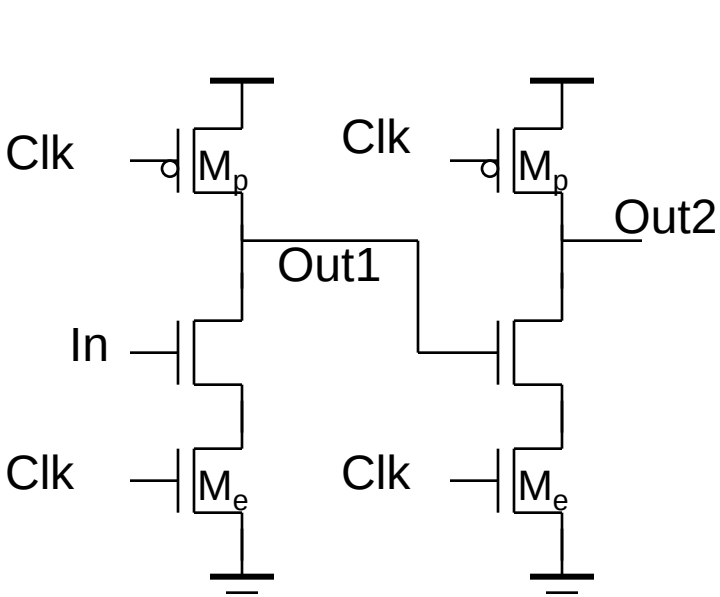


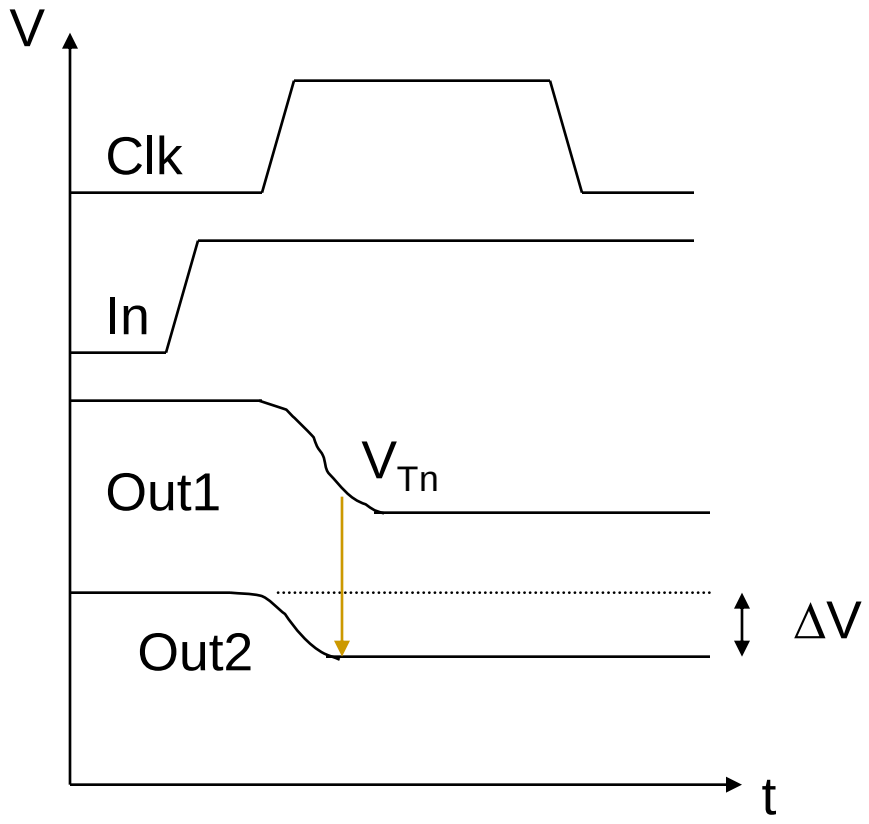
Dynamic gate

Static gate



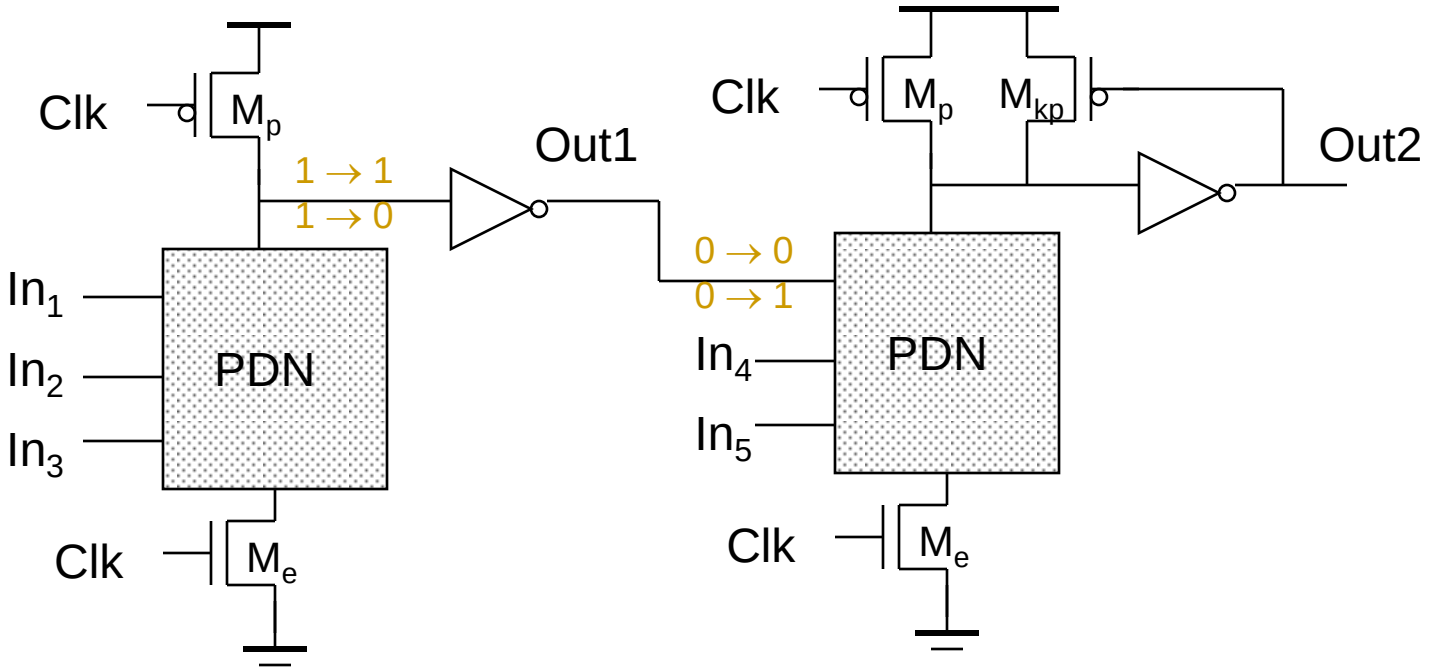
Cascading Dynamic Gates



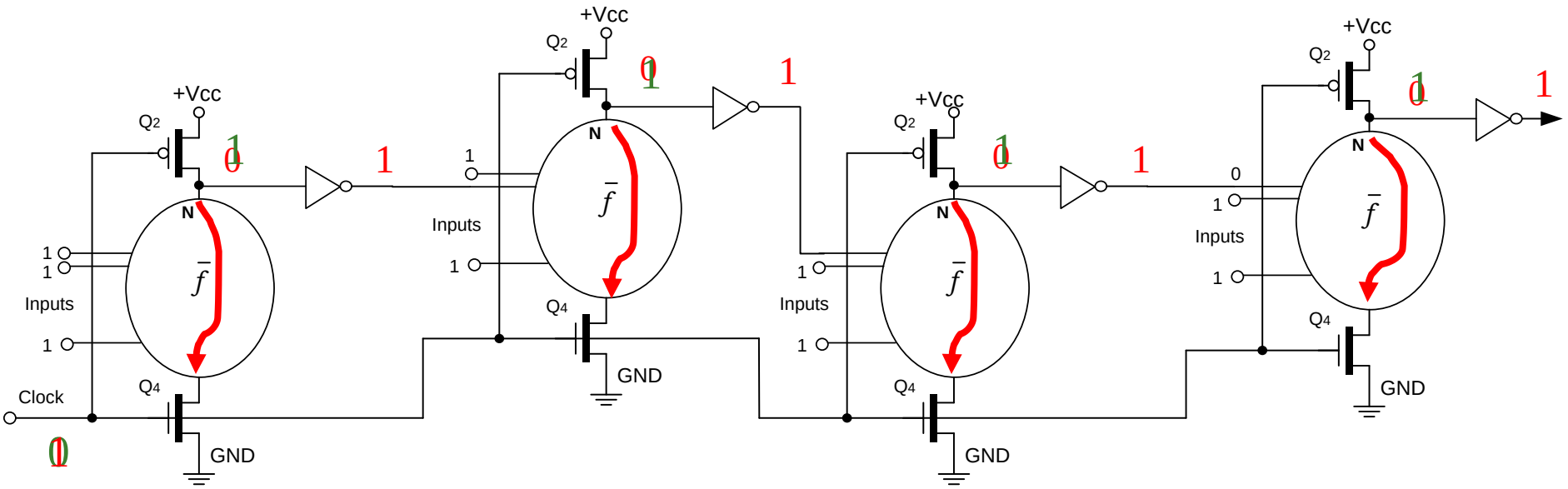


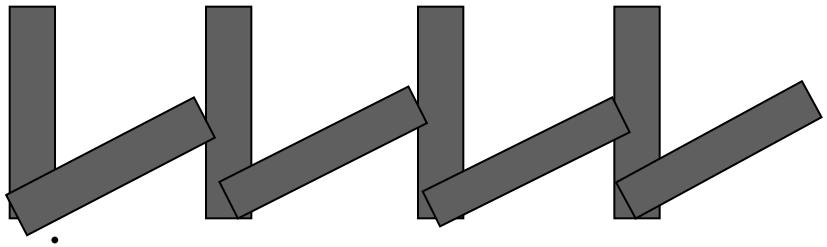
Only 0 → 1 transitions allowed at inputs!

Domino Logic



CMOS Domino Logic Operation





Dominos

Properties of Domino Logic

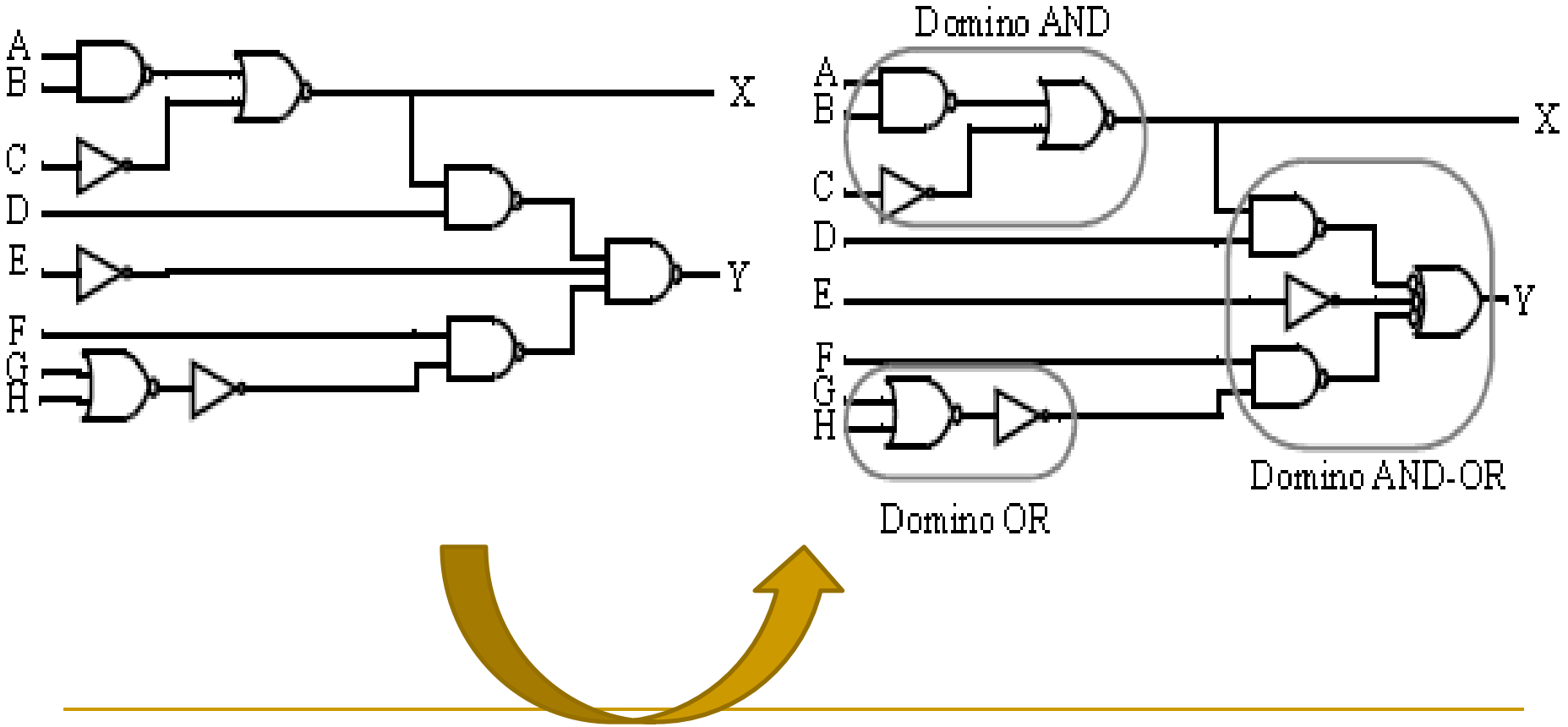
- Only non-inverting logic can be implemented
- Very high speed
- $tp_{HL} = 0$
  - static inverter can be skewed, only L-H transition
  - Input capacitance reduced – smaller logical effort
  - static inverter can be optimized to match fan-out (separation of fan-in and fan-out capacitances)

  - *First 32 bit micro (BellMAC 32) was designed in Domino logic*

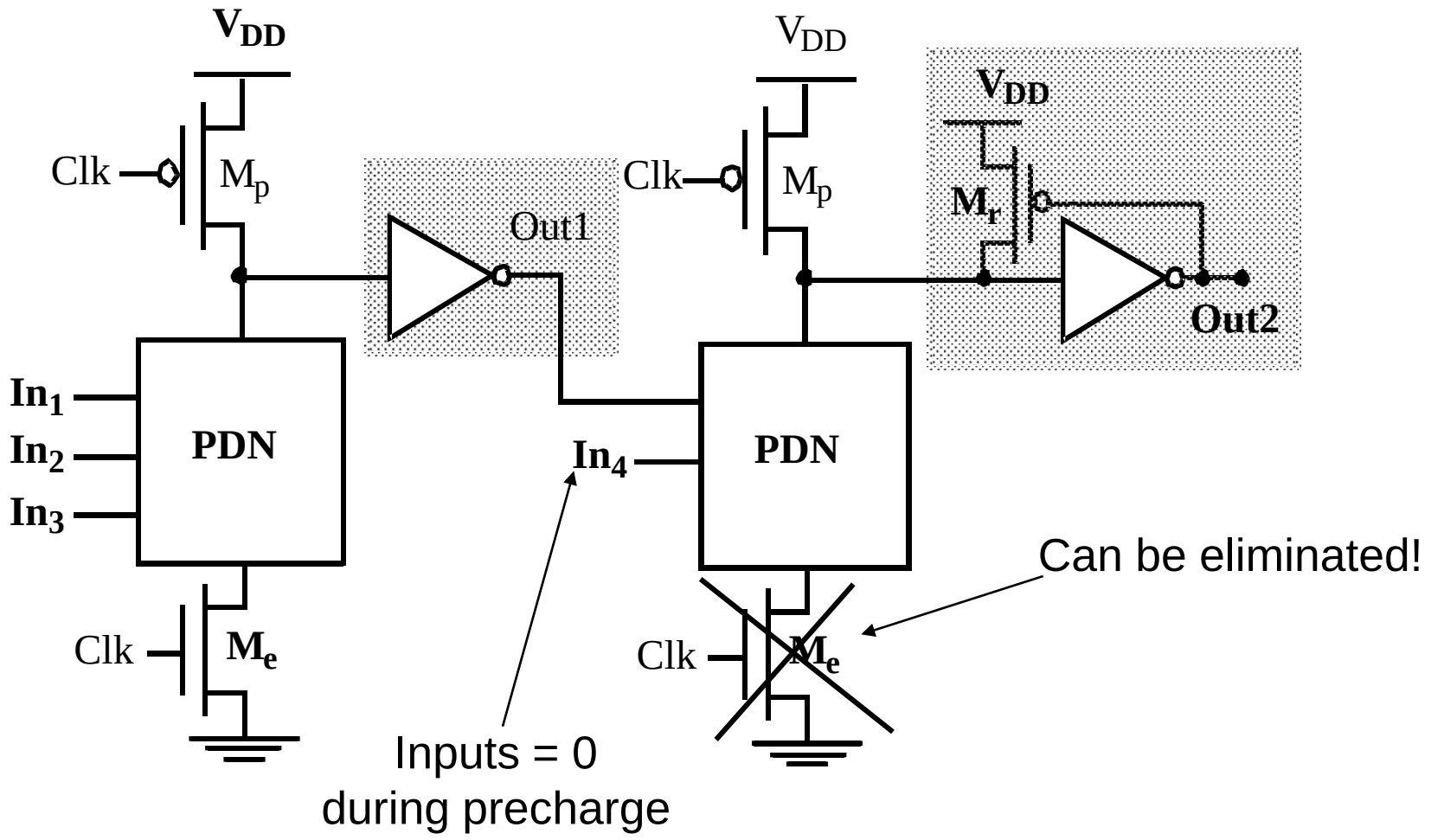
Properties of Domino Logic

- Faster Gates:
  - Allows more logic per cycle with less delay
  - Allows for more complex gates (no dual P network)
- 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.**

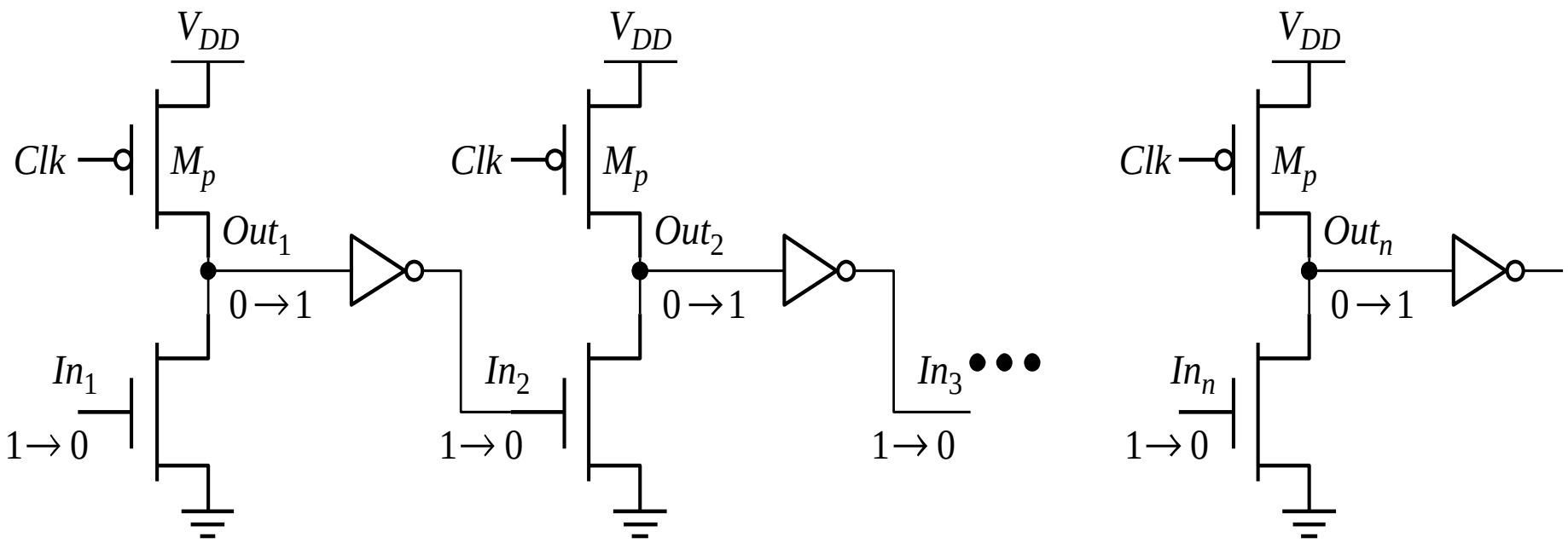
Restructuring logic to enable implementation using non-inverting Domino Logic



Designing with Domino Logic

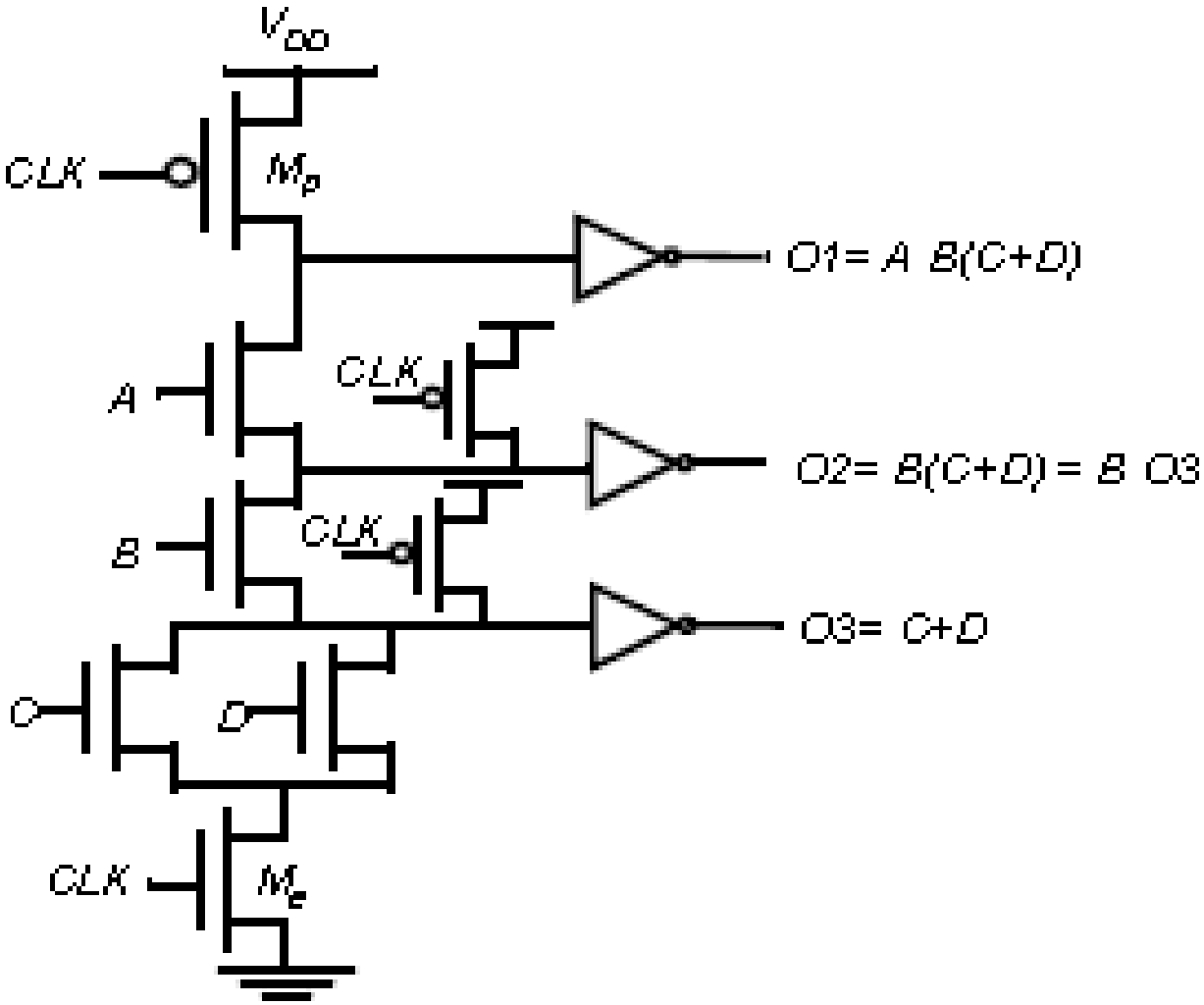


Footless Domino

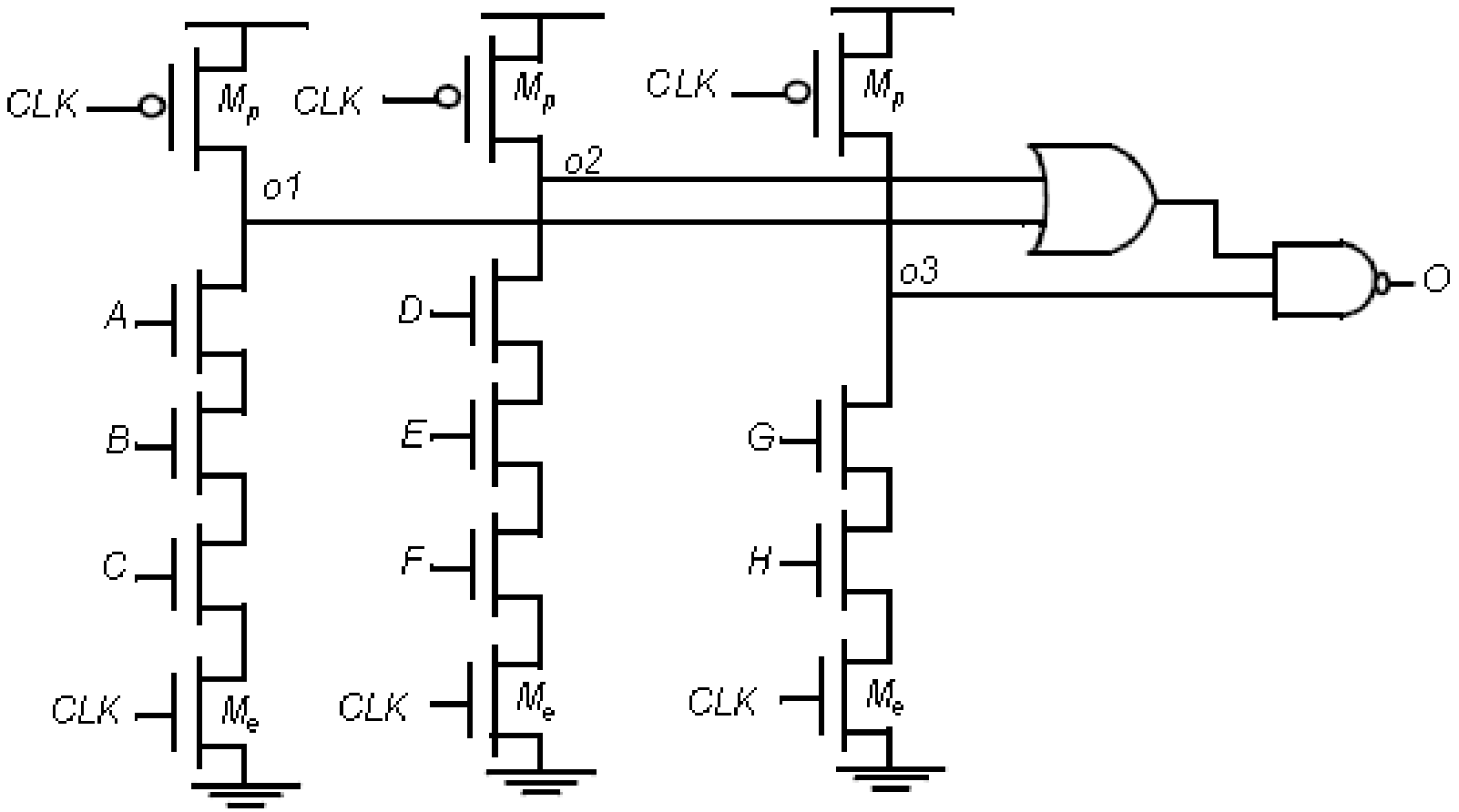


- The first gate in the chain needs a foot switch
- Precharge is rippling – short-circuit current
- A solution is to delay the clock for each stage

Multiple output Domino



Compound Domino logic

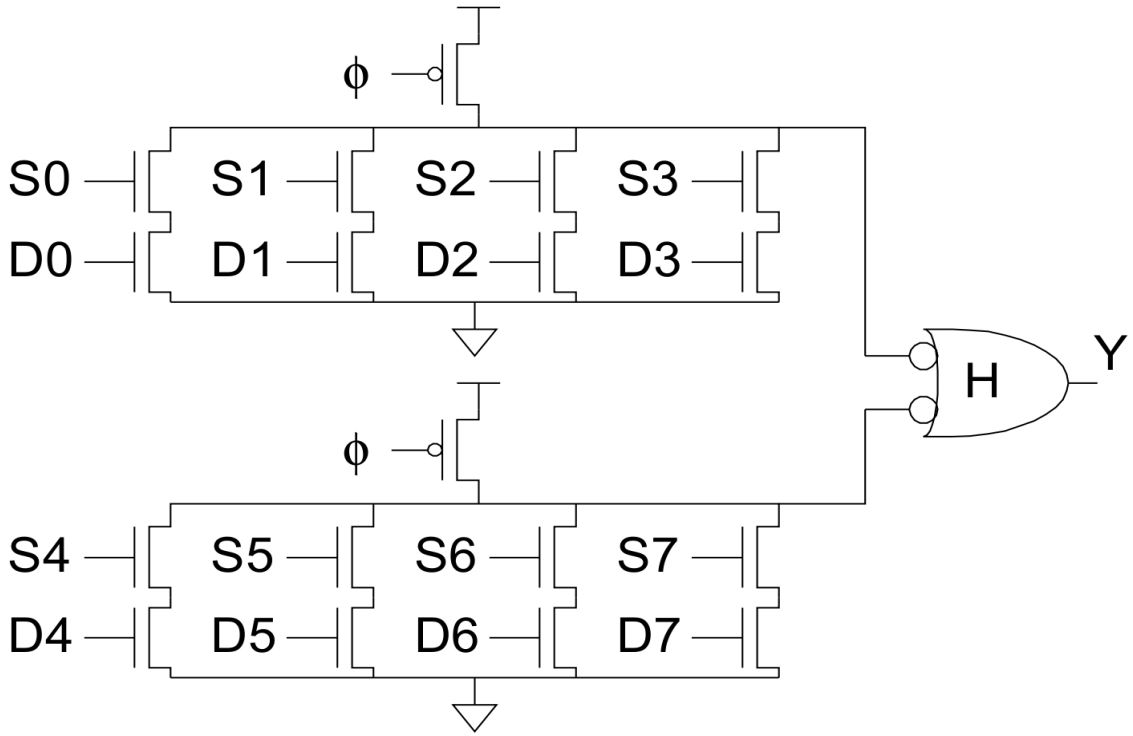


Compound Domino logic where complex static gates can be placed at the output of dynamic gates.

Domino optimizations

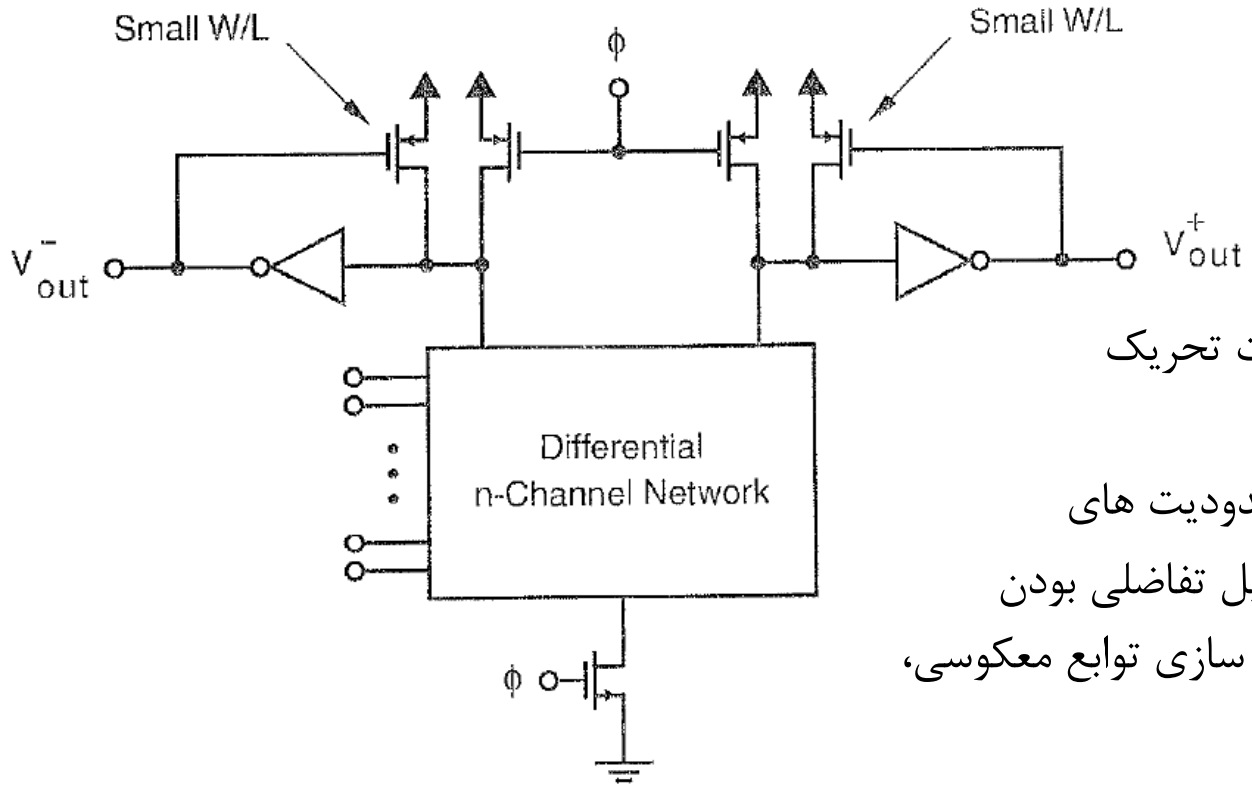
Each domino gate triggers next one, like a string of dominos toppling over
Gates evaluate sequentially, precharge in parallel
Evaluation is more critical than precharge
HI-skewed static stages can perform logic (Skewed gates favor one edge over another)

Using a higher or lower P/N ratio favors rising or falling outputs, respectively. For example, with a P/N ratio of 4/1, the input does not have to fall as far as VDD/2 before the output could switch. We call such a circuit a “high skewed” gate and use it on paths where the critical transition is a rising output



منطق تفاضلی Domino
Differential (Dual Rail) Domino

ترکیب طراحی منطق تفاضلی و تکنیک های پویا منطق تفاضلی دامینو را ایجاد می کند



اتلاف توان dc آن بسیار کم است

در حالی که سرعت آن مناسب است

به دلیل وجود بافرهای خروجی، قابلیت تحریک

خروجی آن بسیار مناسب است

و مهمتر از همه، یکی از مهمترین محدودیت های

منطق Domino این است که به دلیل تفاضلی بودن

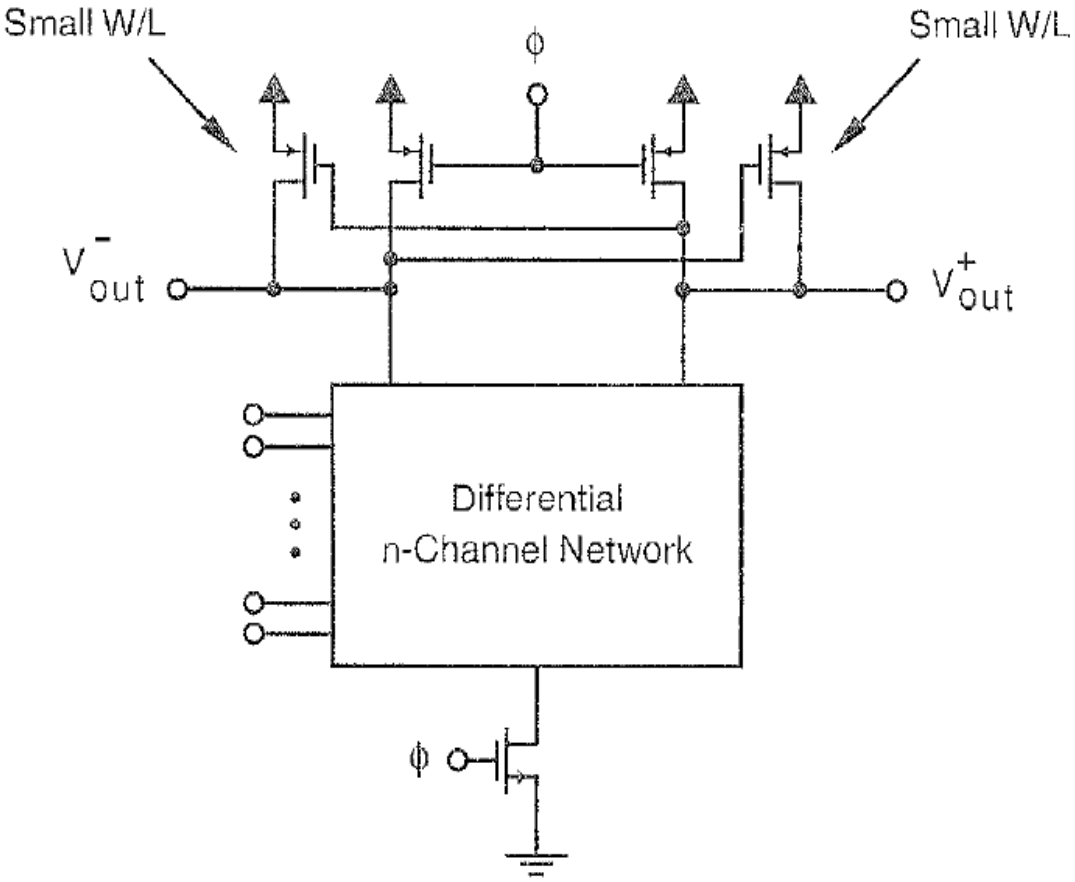
ذاتی این منطق، پیچیدگی های پیاده سازی توابع معکوسی،

حذف شده است

در منطق تفاضلی Domino این نوع توابع به سادگی با پیاده سازی تابع منطقی غیرمعکوسی و تعویض خطوط خروجی آن پیاده سازی می شود

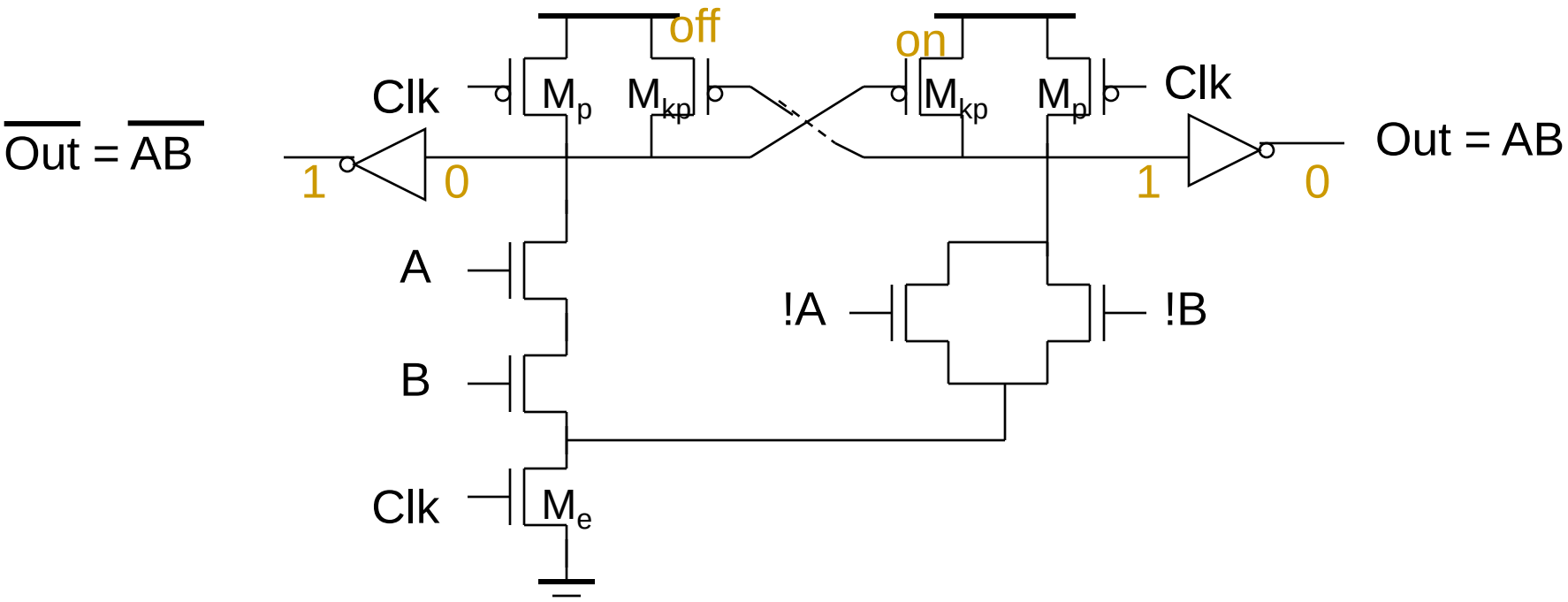
منطق تفاضلی Domino
Differential (Dual Rail) Domino

هنگامی که ظرفیت بار خروجی کوچک است، معکوس کننده ها در خروجی را می توان حذف نمود



این روش پیاده سازی دارای تأخیر و حاصل ضرب توان- تأخیر بسیار اندک است

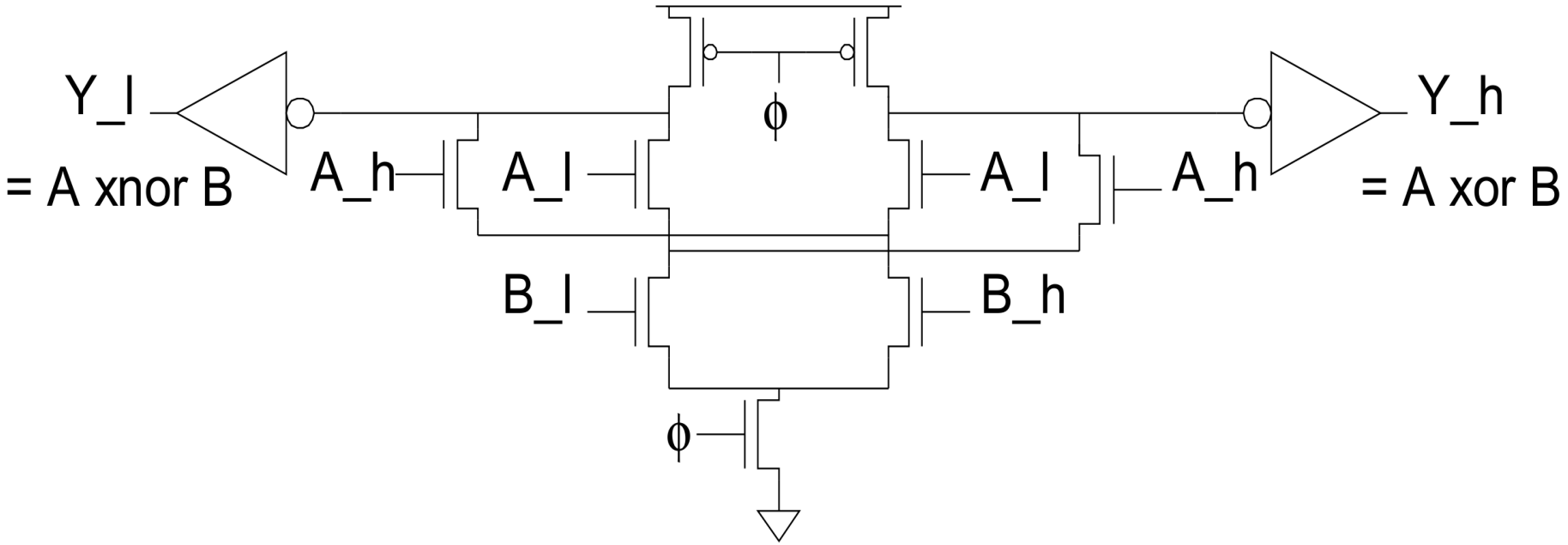
Differential (Dual Rail) Domino



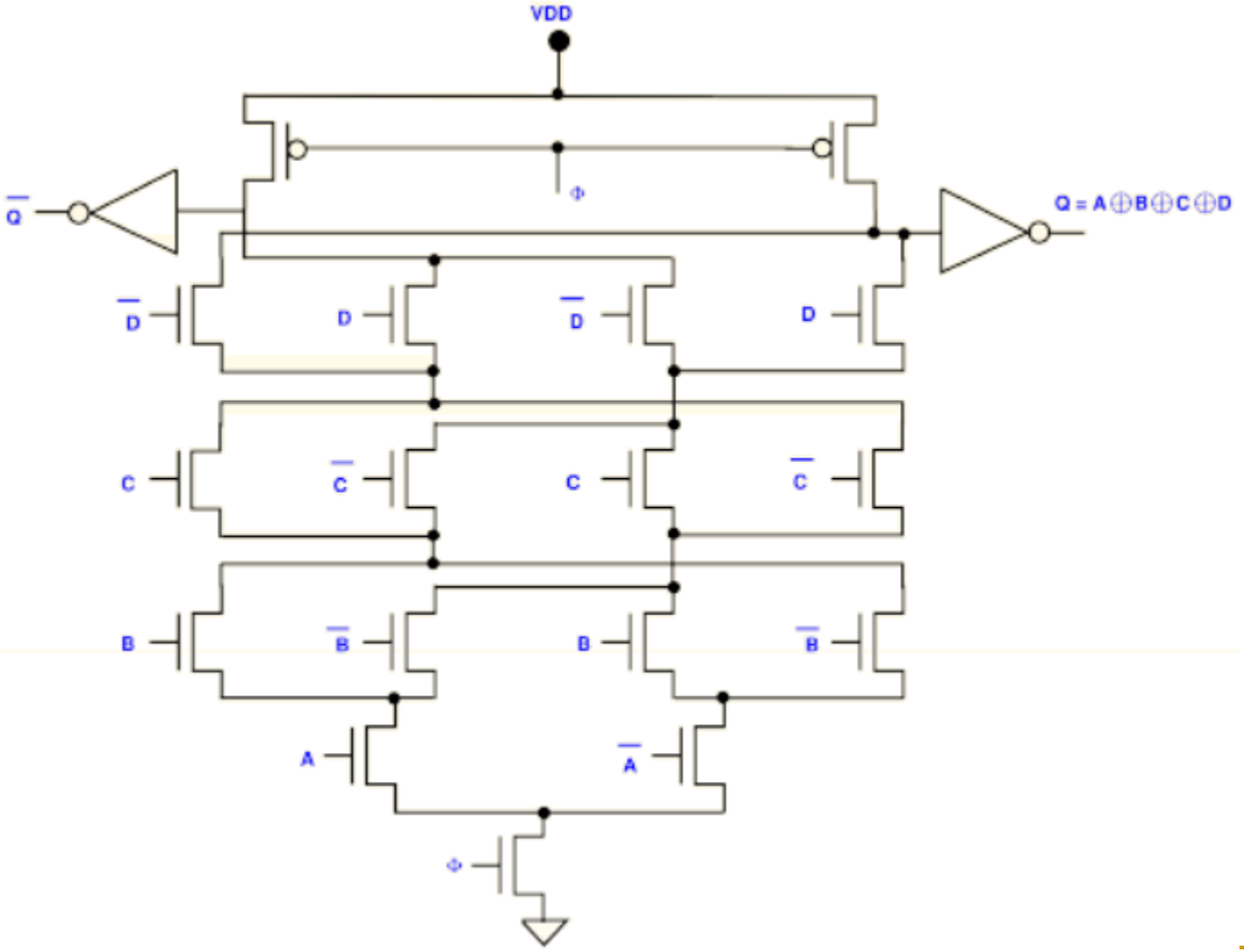
Differential (Dual Rail) Domino

Sometimes possible to share transistors

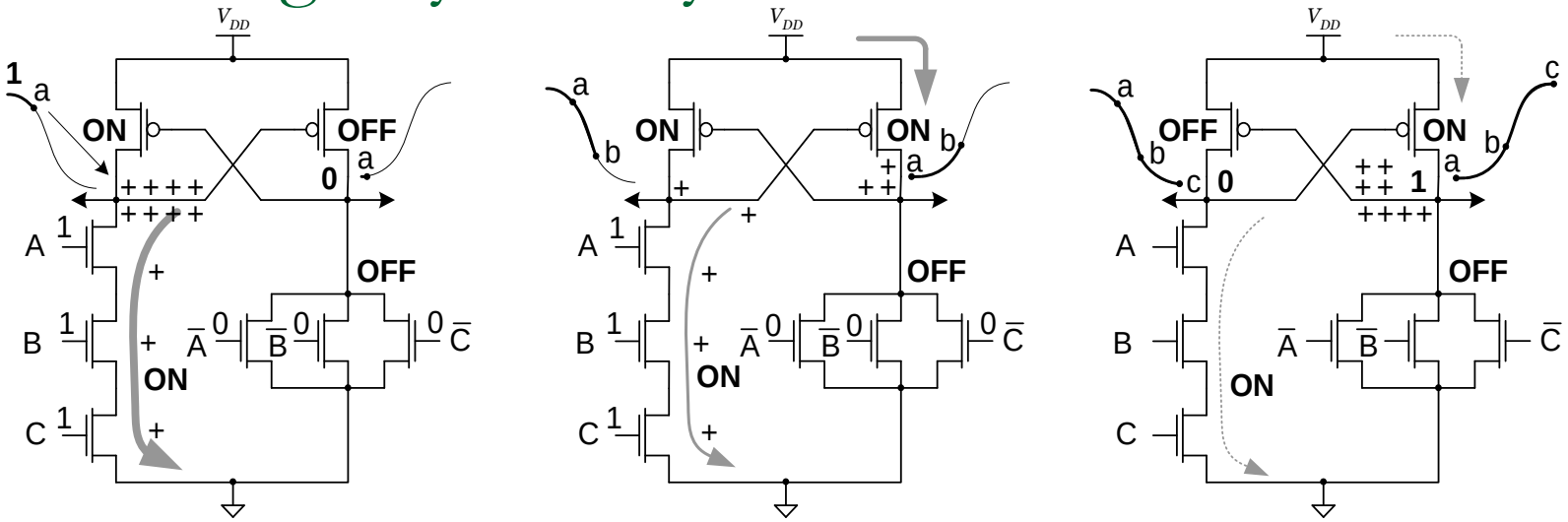
Sharing works well in implementations of symmetric functions

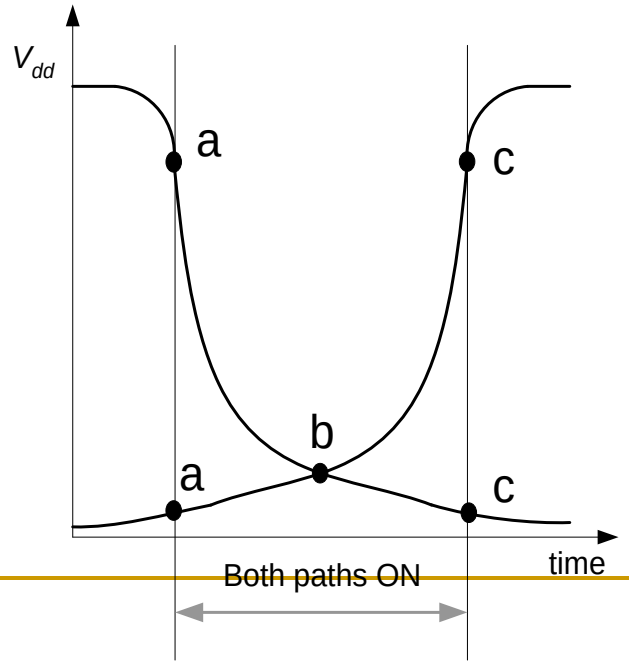


Dynamic CVSL (Cascade Voltage Switch Logic) XOR



Switching Asymmetry in DCVSL

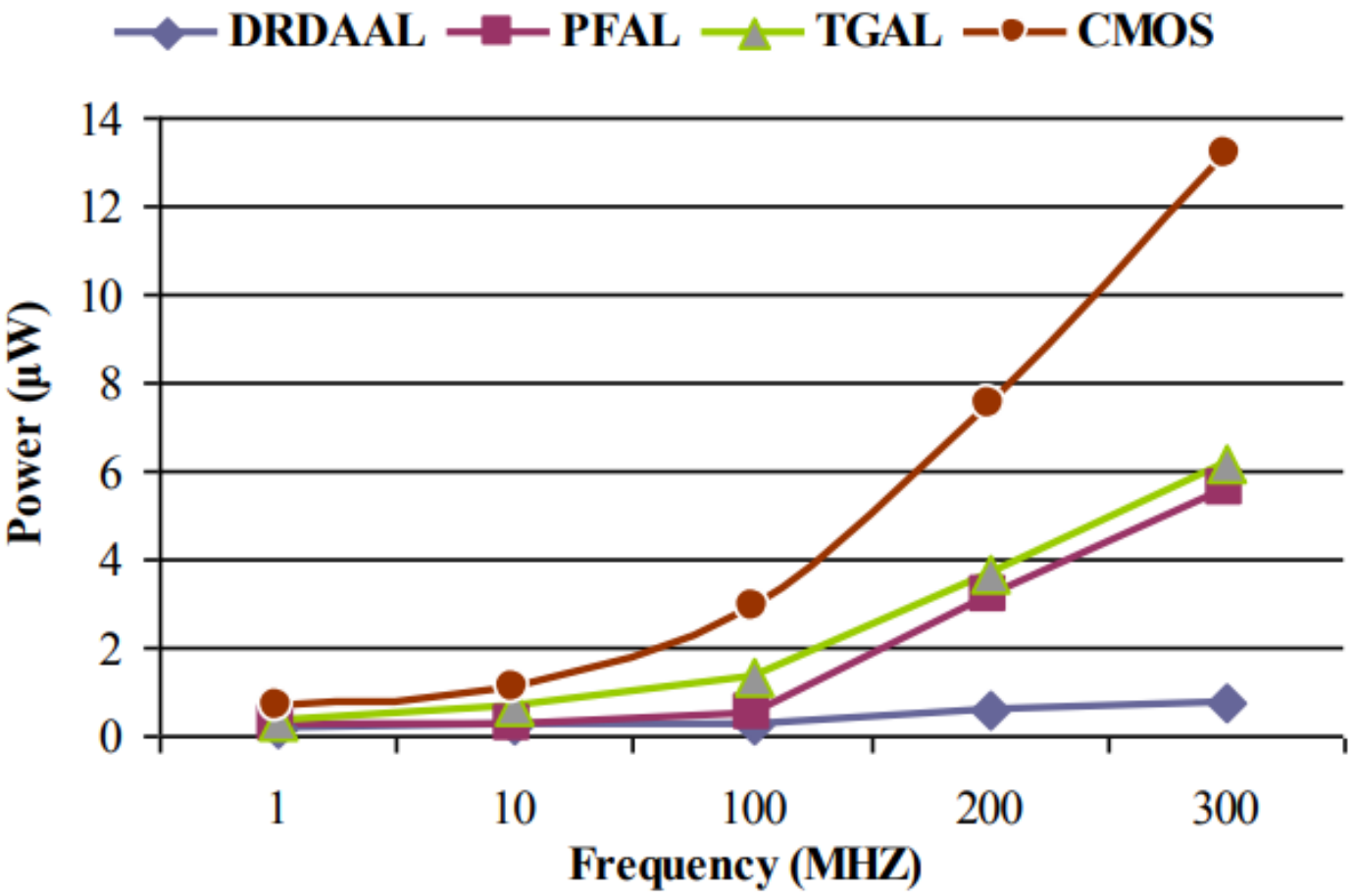




This asymmetry causes current spikes and increased power consumption !

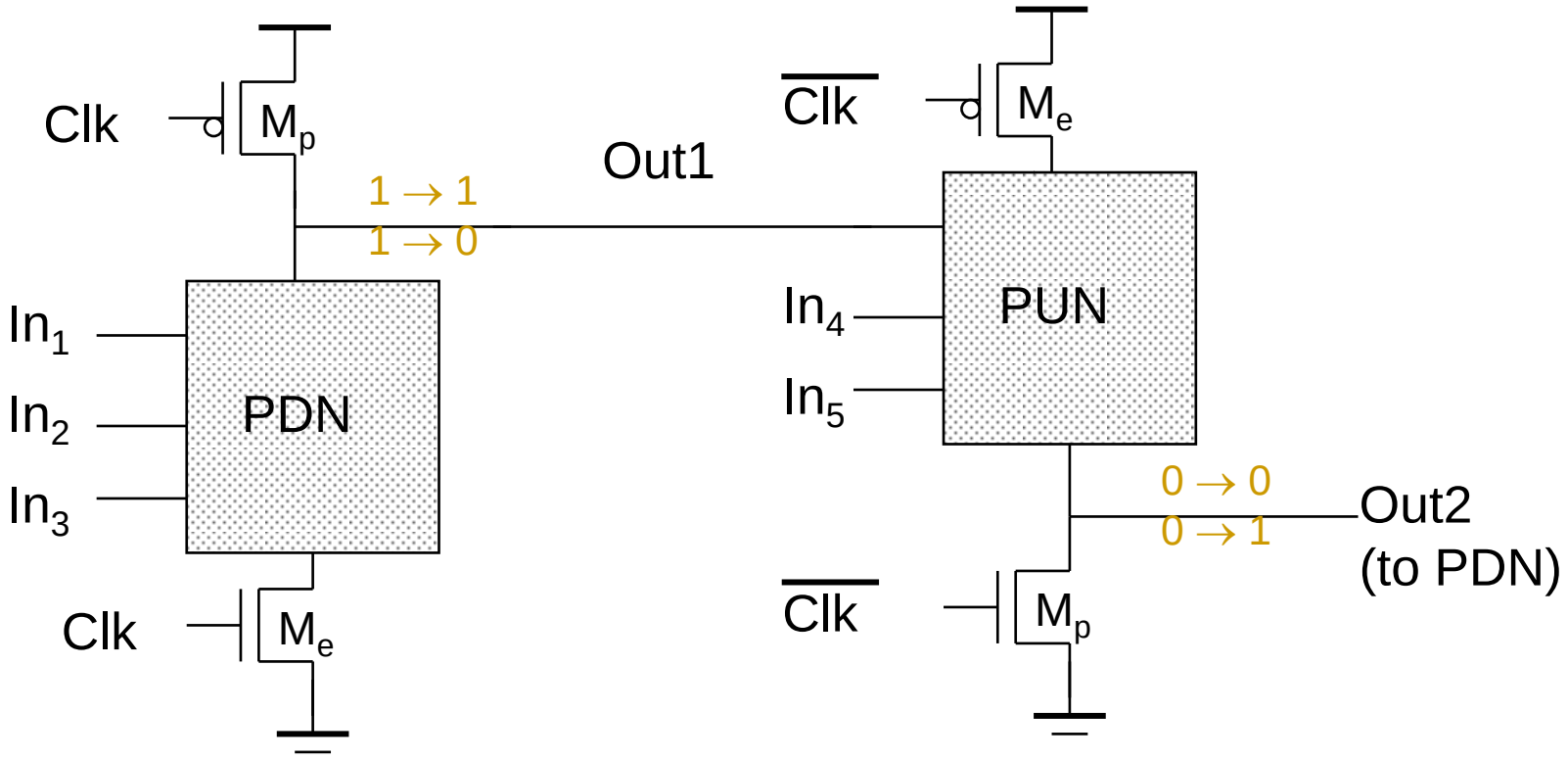
Drawback: area, power

DRDAAL power consumption



Power comparison of DRDAAL full adder with different logic designs

np-CMOS

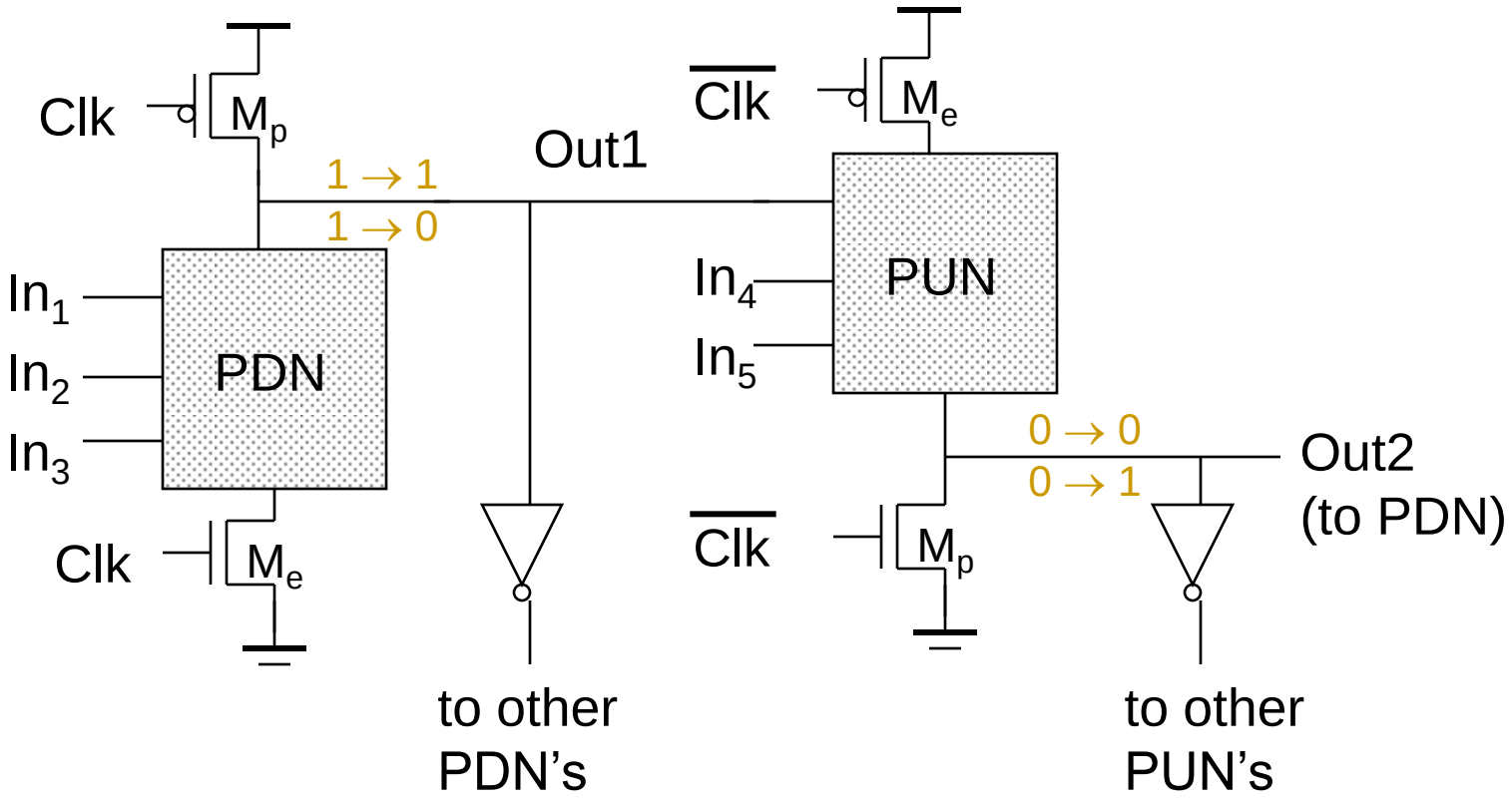


Only $0 \rightarrow 1$ transitions allowed at inputs of PDN Only $1 \rightarrow 0$ transitions allowed at inputs of PUN

Really dense layouts and very high speed (20% faster than domino with the correct sizing)

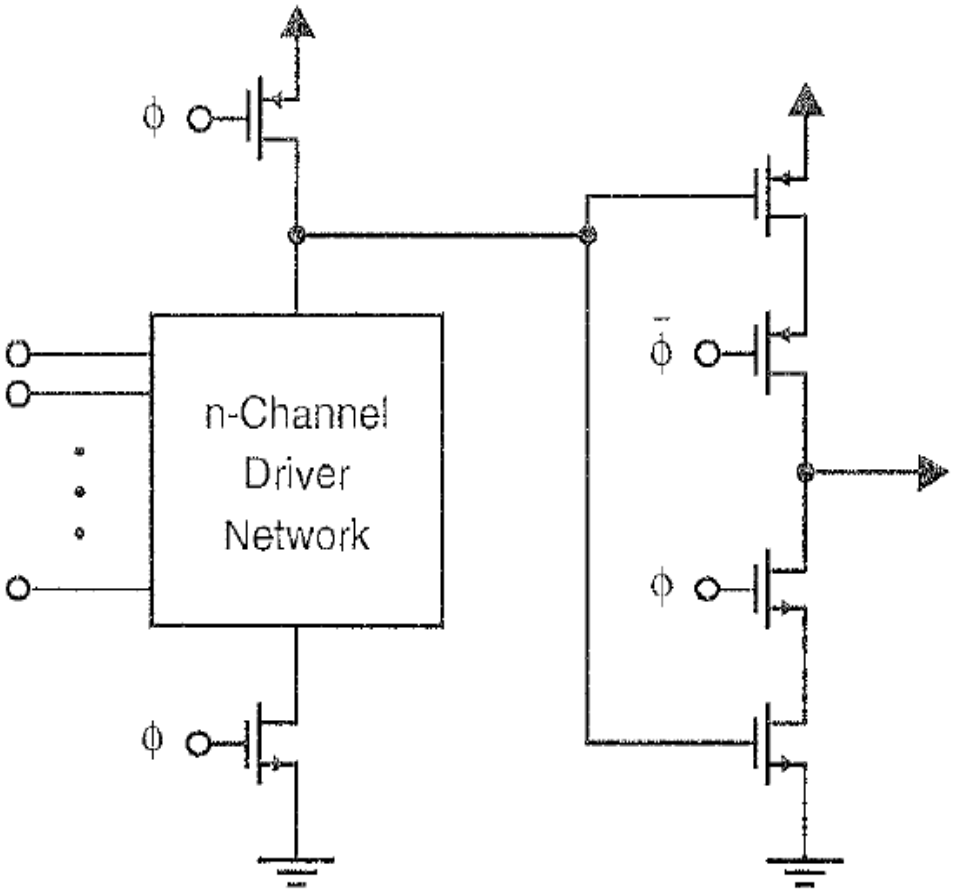
DEC alpha uses np-CMOS logic (Dobberpuhl)

NORA Logic



WARNING: Very sensitive to noise!

NORA Logic (Domino with Clocked Latch)



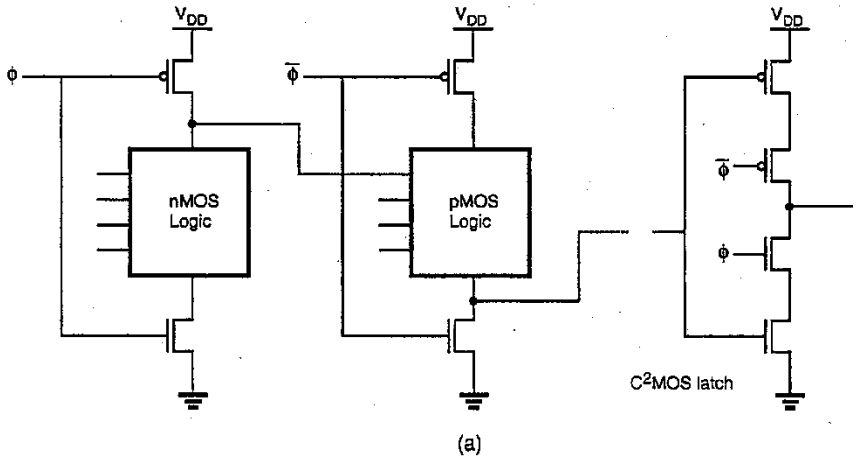
Combines NP Domino logic sections with C²MOS latch

- n-logic block can drive p-logic block or another n-logic block with a static inverter
- similarly for a p-logic block
- Must end in a C²MOS latch

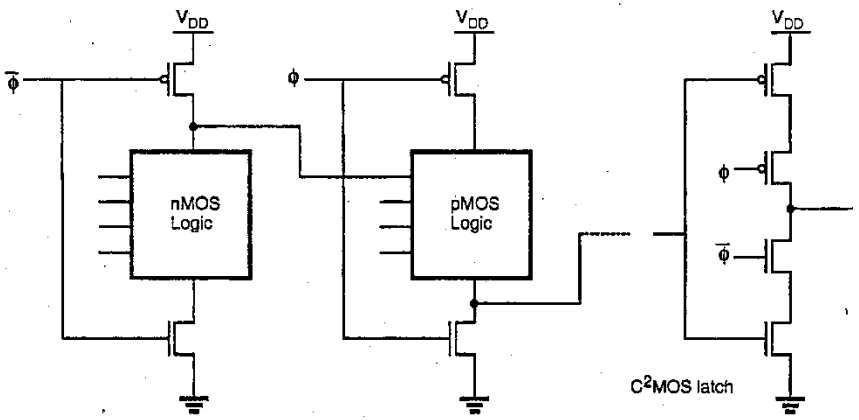
Rules to avoid race conditions:

- During precharge, logic blocks are OFF
- During eval, internal inputs make only one transition

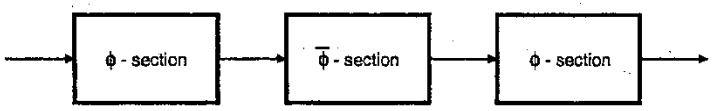
Pipelined NORA CMOS Circuit Operation



(a)



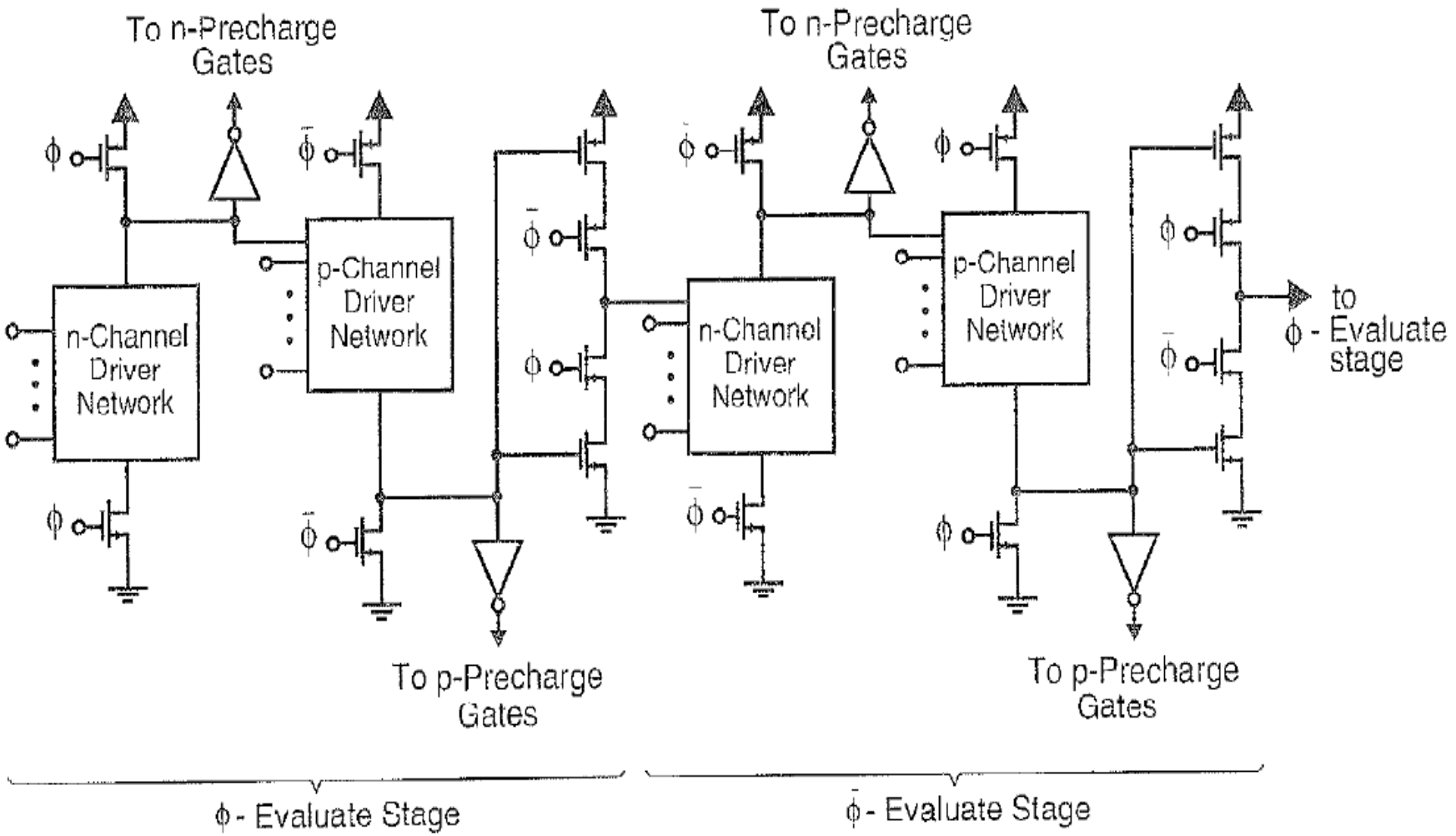
(b)



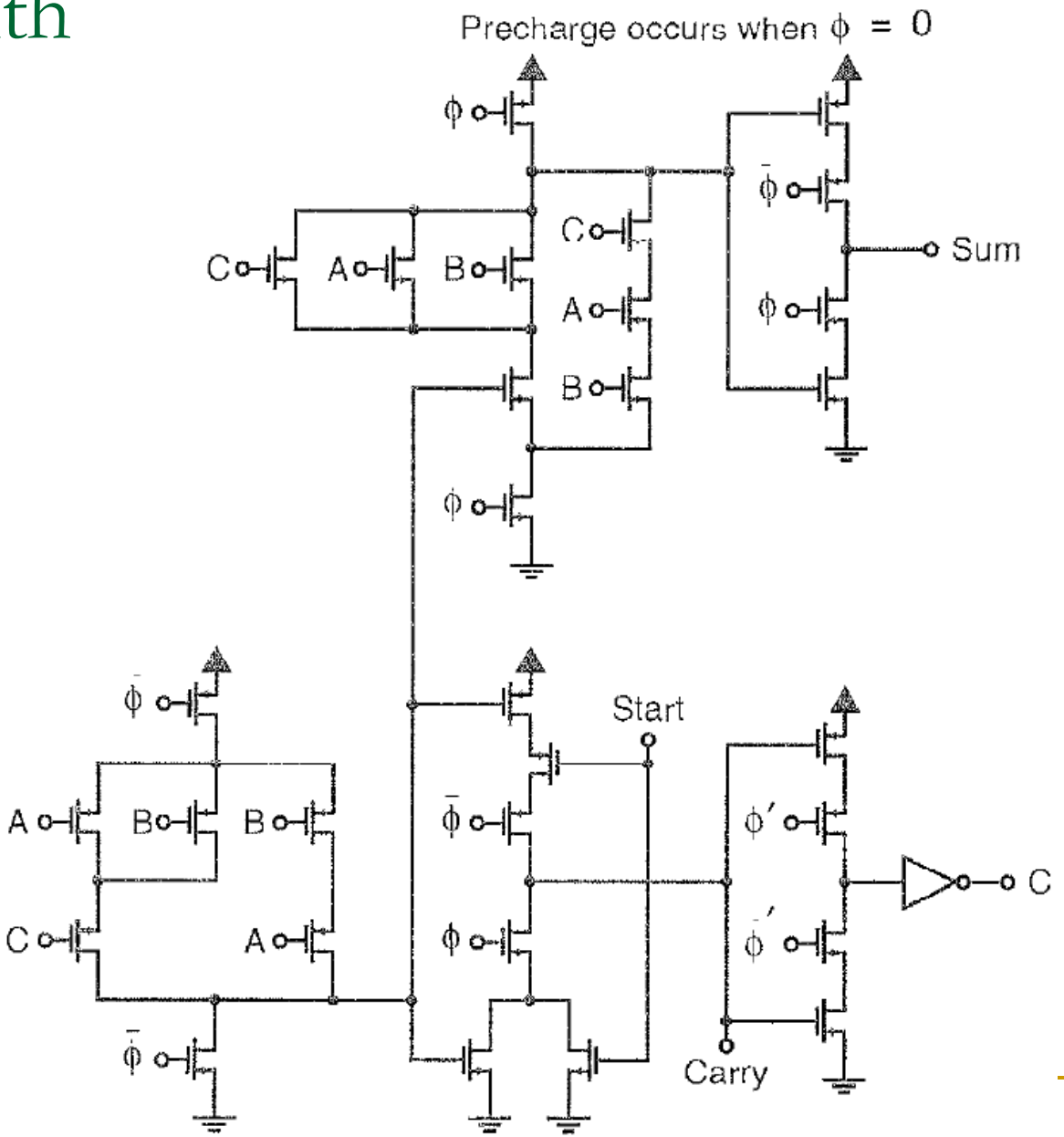
(c)

- With pipelined NORA CMOS logic design
  - one can alternate N and P stages between C²MOS latches where ϕ high is used for evaluation as shown in (a)
  - Or, one can alternate N and P stages similarly between C²MOS latches with ϕ' high used for evaluation as in (b)
  - ϕ sections may be alternately cascaded with ϕ' sections as shown in (c)
- During the evaluation phase, the logic ripples through each stage in succession up to the next C²MOS latch

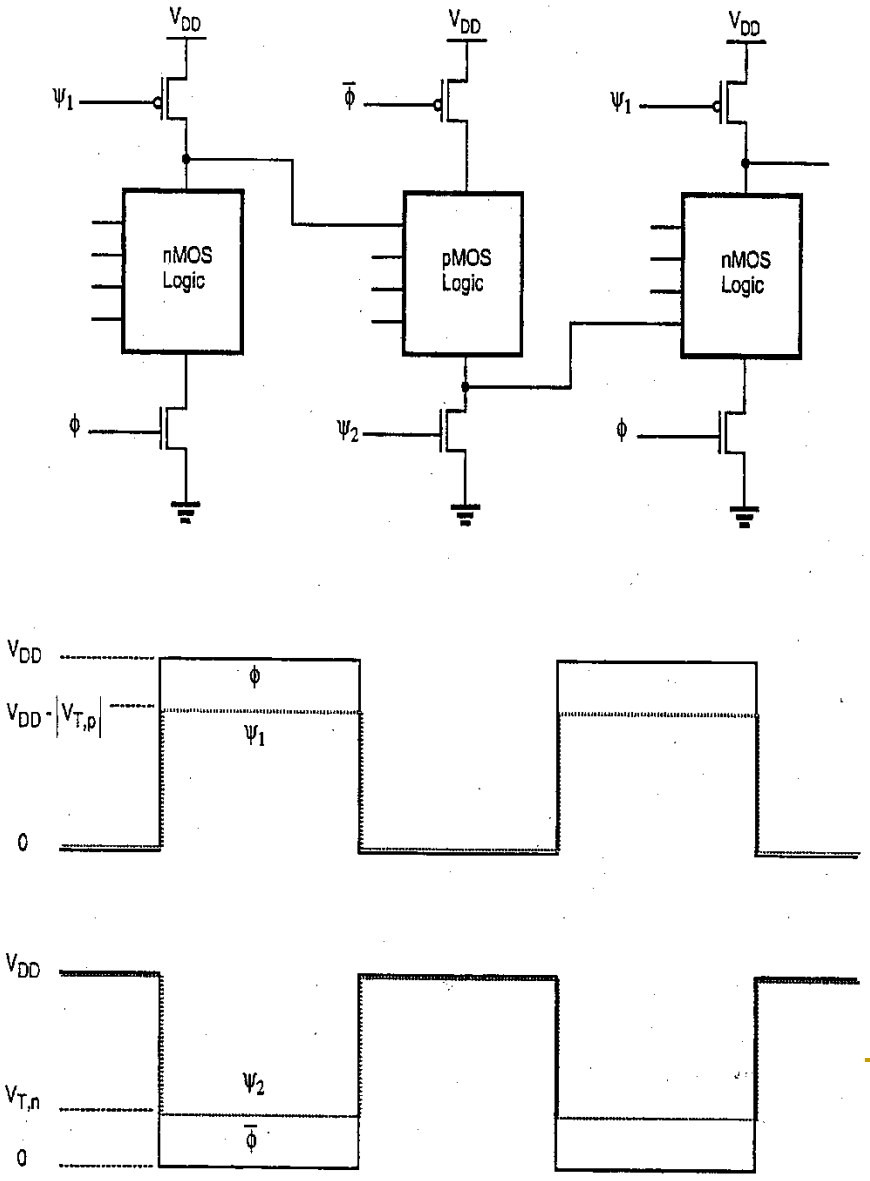
No-Race


To n-Precharge Gates
To n-Precharge Gates
n-Channel Driver Network
p-Channel Driver Network
n-Channel Driver Network
p-Channel Driver Network
to ϕ - Evaluate stage
To p-Precharge Gates
To p-Precharge Gates
ϕ - Evaluate Stage
ϕ̄ - Evaluate Stage

Full Adder with No-Race

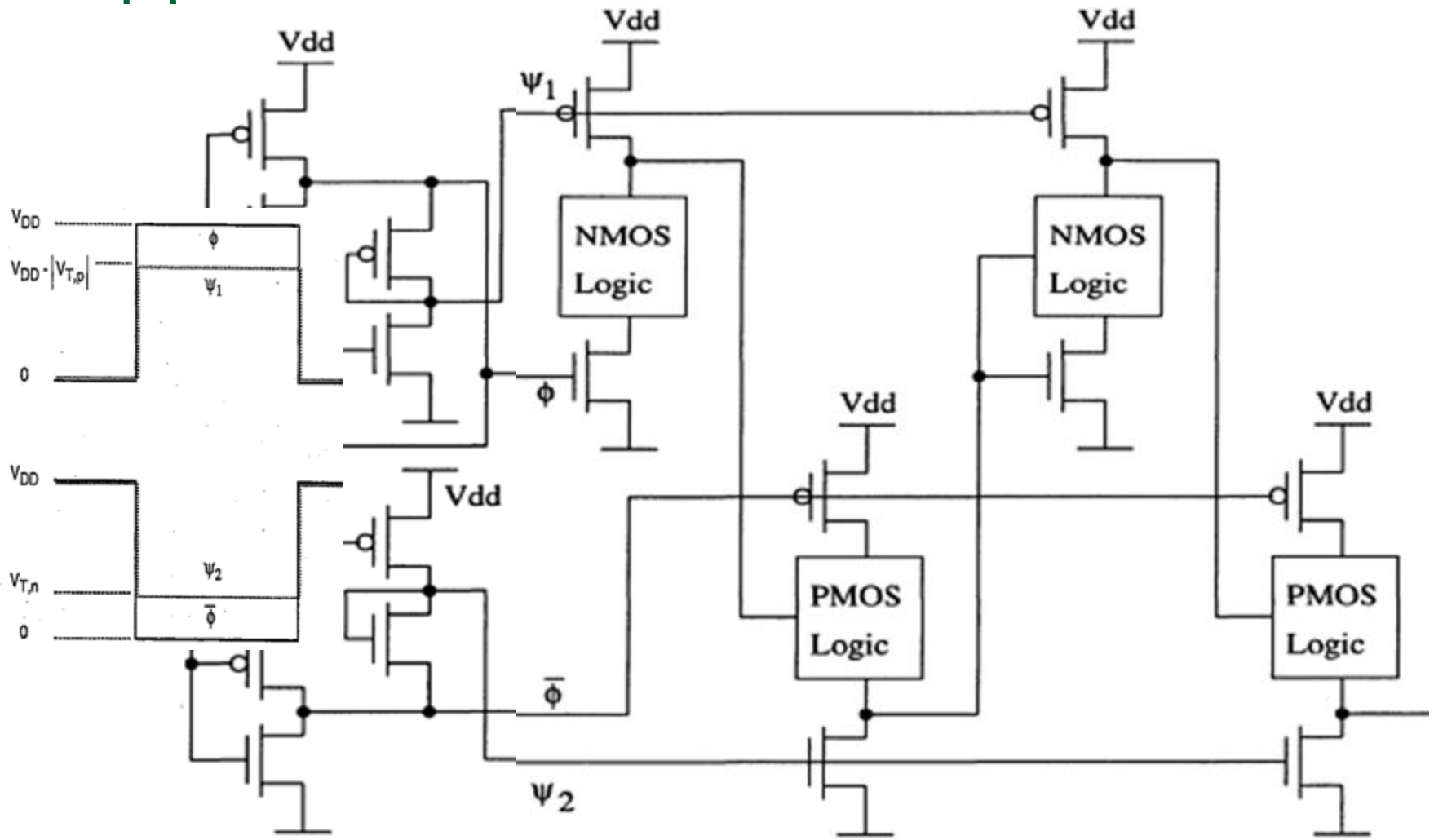


Zipper CMOS Dynamic Logic



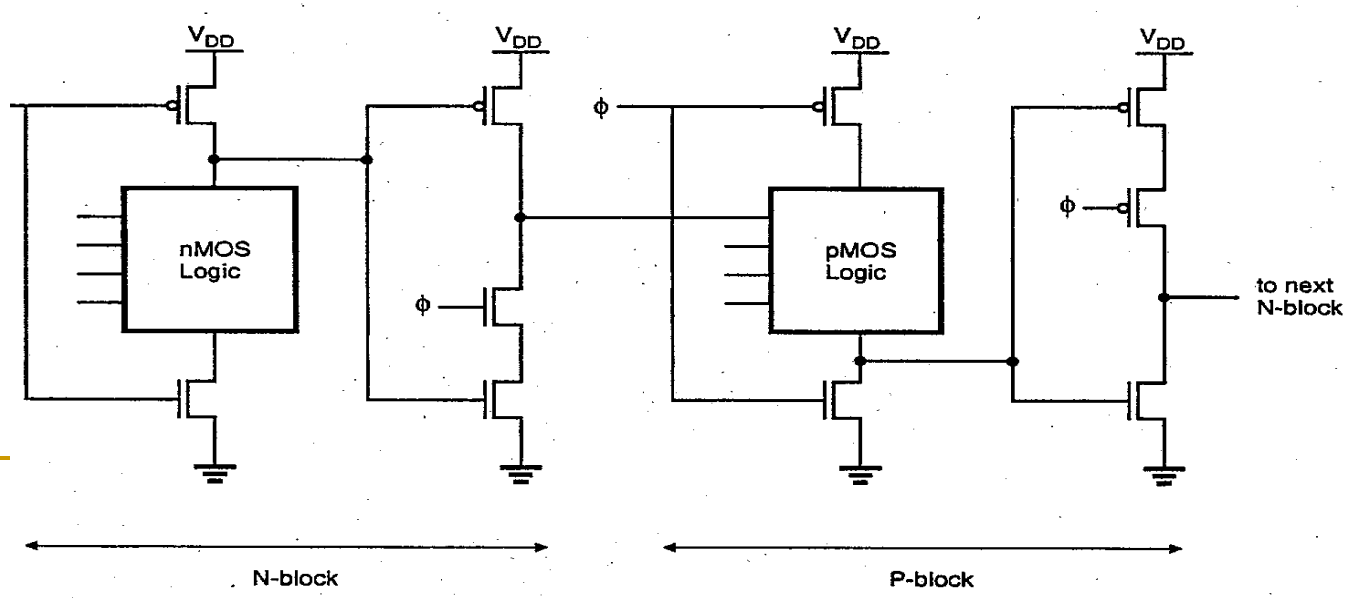
- **Zipper CMOS logic** is a scheme for improving charge leakage and charge sharing problems
- Pre-charge transistors receive a slightly modified clock where the clock pulse (during pre-charge off time) holds the pre-charge transistor at weak conduction in order to provide a trickle pre-charge current during the evaluation phase
  - PMOS pre-charge transistor gates are held at Vdd - |Vtp|
  - NMOS pre-charge transistor gates are held at Vtn above GND

Zipper

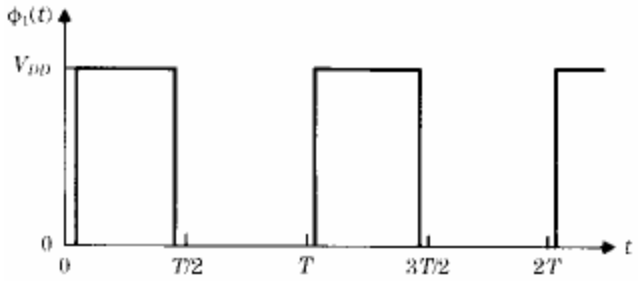

Vdd
Vdd
Vdd
Vdd
Vdd
Vdd
ψ1
NMOS
Logic
NMOS
Logic
PMOS
Logic
PMOS
Logic
ϕ
ϕ̄
ψ2
VDD
VDD - |VT,p|
0
VDD
VT,n
0

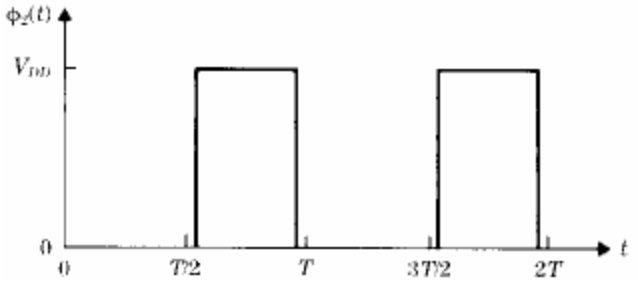
Pipelined True Single Phase Clock (TSPC) CMOS

- A TSPC system (without any inverted clocks required) can be built as shown below
- Each NMOS and PMOS stage is followed by a dynamic latch (inverter) built with only the single phase clock ϕ
- The single phase clock ϕ is used for both NMOS and PMOS stages
  - NMOS logic stages pre-charge when ϕ is low and evaluate when ϕ is high
  - PMOS logic stages pre-charge when ϕ is high and evaluate when ϕ is low
- With inverter latches between each stage, an erroneous evaluate condition can not exist
- Attractive circuit for use in pipelined, high performance processor logic

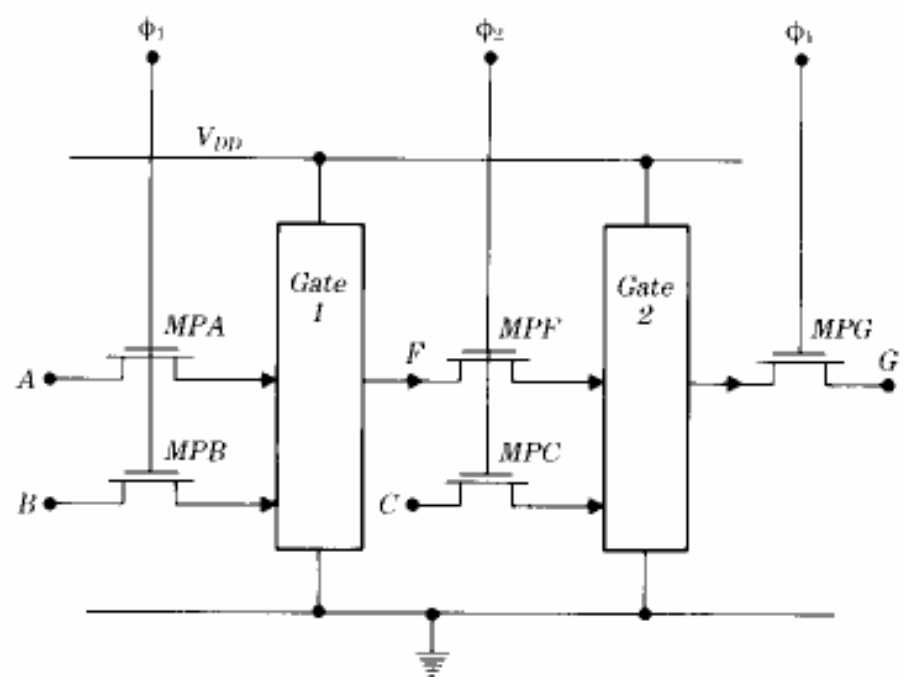


پالس های ساعت دو فازی

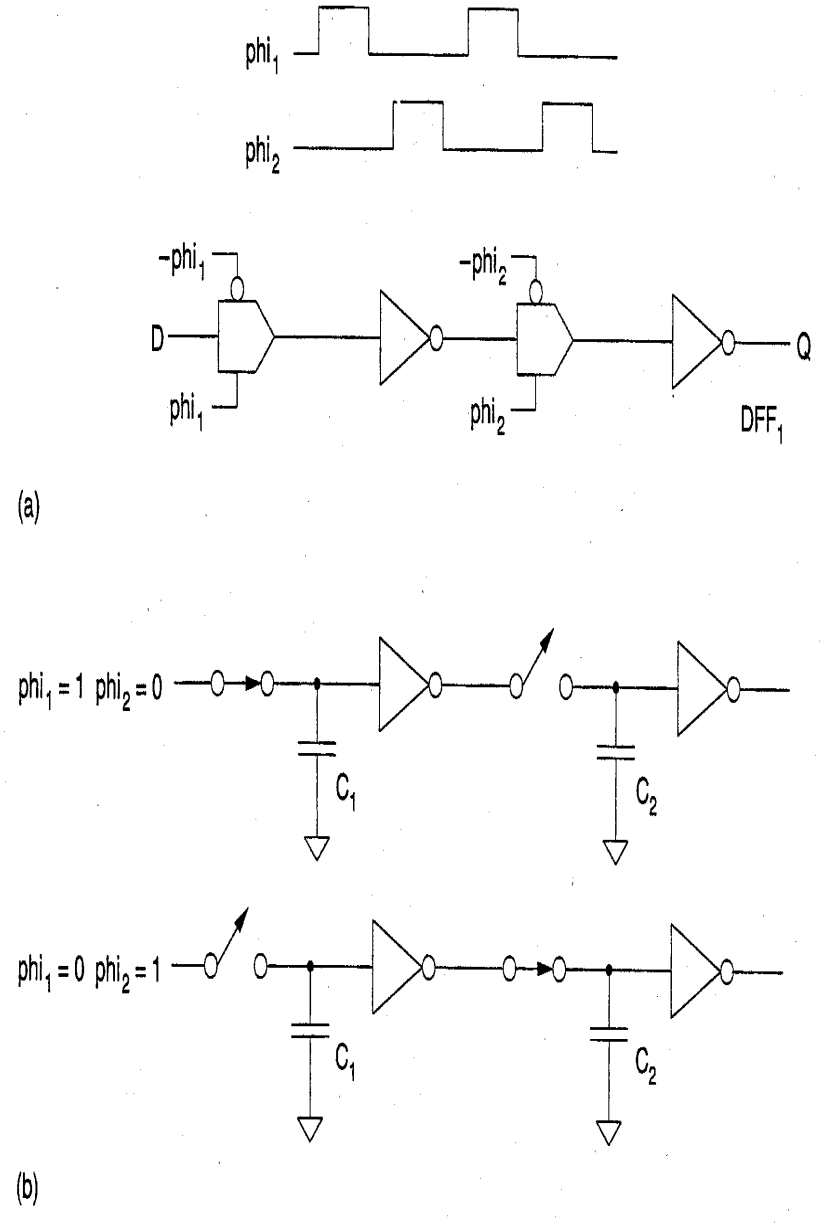




$$\phi_1(t)\phi_2(t) = 0$$

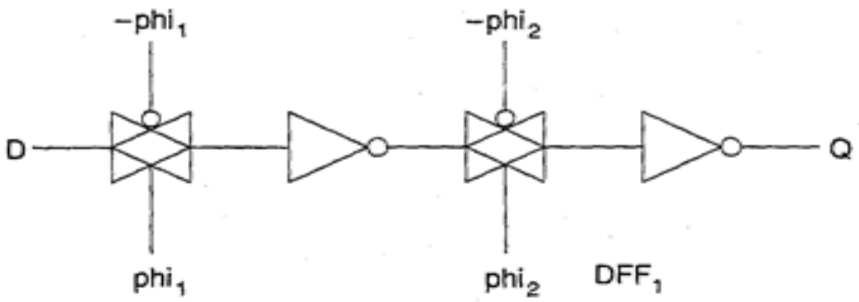


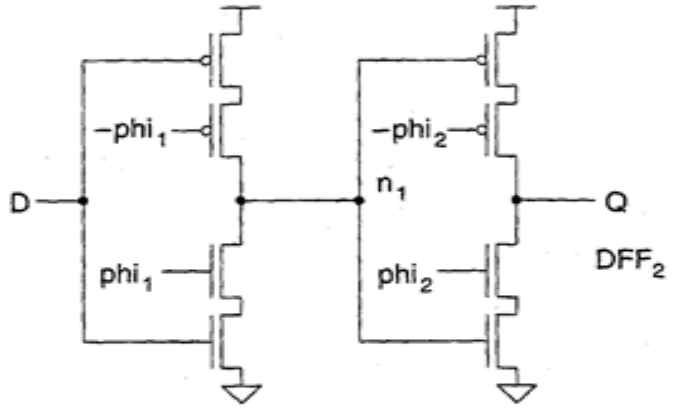
Dynamic Registers with Two Phase Clocks

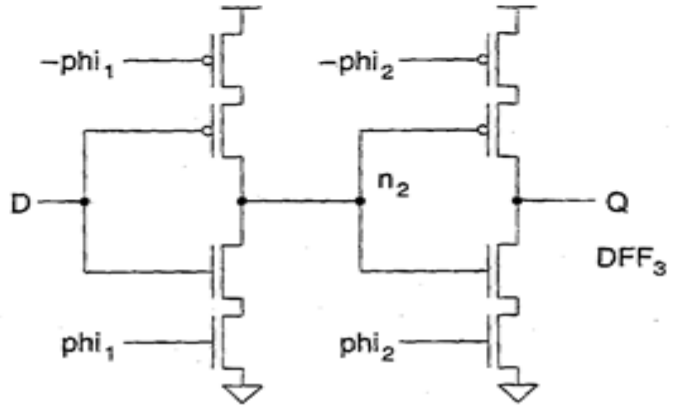


- Dynamic register with pass gates and two phase clocking is shown
  - Clocks *phi1* and phi2 are non-overlapping
  - When phi1 is high & phi2 is zero,
    - 1st pass gate is closed and D data charges gate capacitance C1 of 1st inverter
    - 2nd pass gate is open trapping prior charge on C2
  - When phi1 is low and phi2 is high,
    - 1st pass gate opens trapping D data on C1
    - 2nd pass gate closes allowing C2 to charge with inverted D data
- If clock skew or sloppy rise/fall time clock buffers cause overlap of phi1 and phi2 clocks,
  - Both pass gates can be closed at the same time causing mixing of old and new data and therefore loss of data integrity!

Two Phase Dynamic Registers (Compact Form)

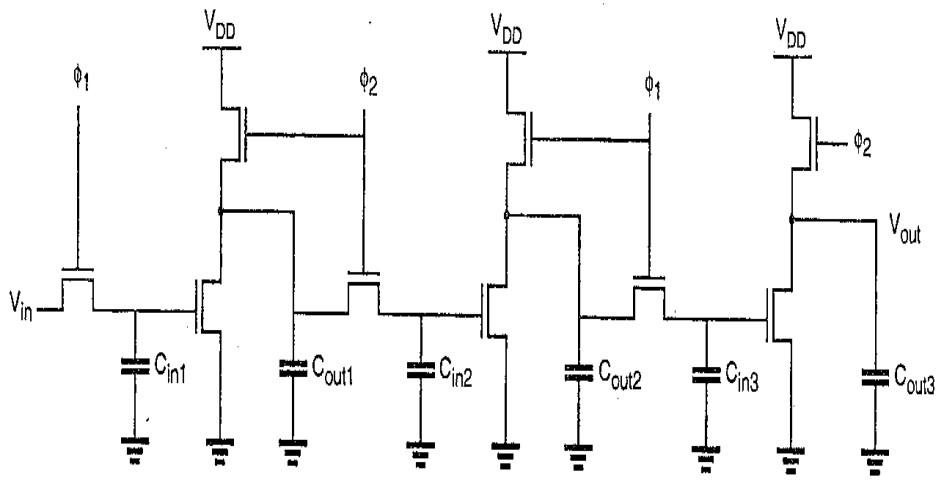


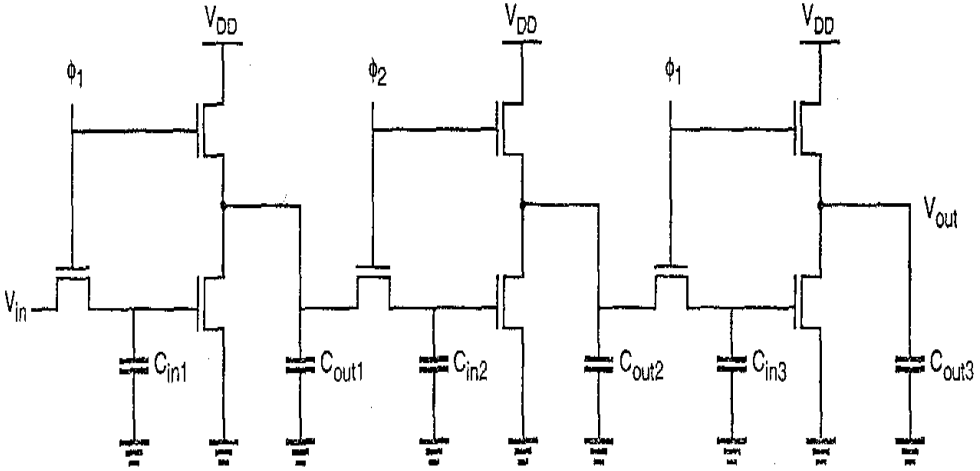




- Compact implementation of two phase dynamic registers shown at left using a tri-state buffer form.
  - Transmission gate and inverter integrated into one circuit
  - Two versions:
    - Pass devices closest to output
    - Inverter devices closest to output
- Two phase dynamic registers and logic is often preferred over single phase because
  - Due to finite rise and fall times, the CLK and CLK' are not truly non-overlapping
  - Clock skew often is a problem due to the fact that CLK' is usually generated from CLK using an inverter circuit and also due to the practical problem of distributing clock lines without any skew

Dynamic Shift Registers with Enhancement Load



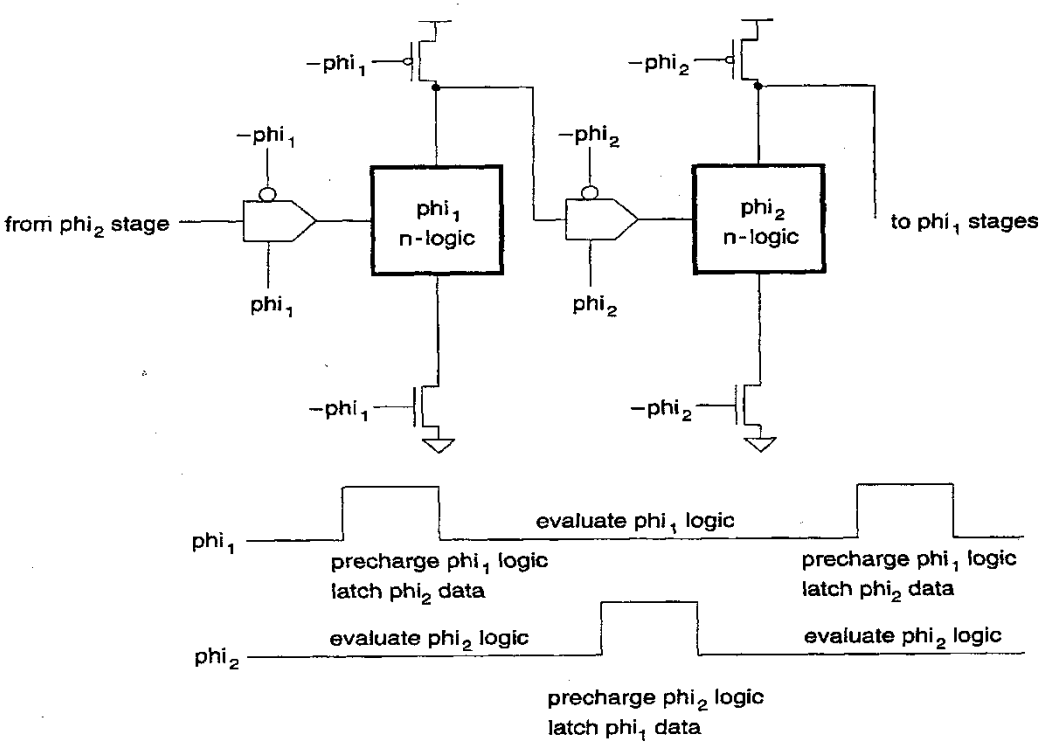


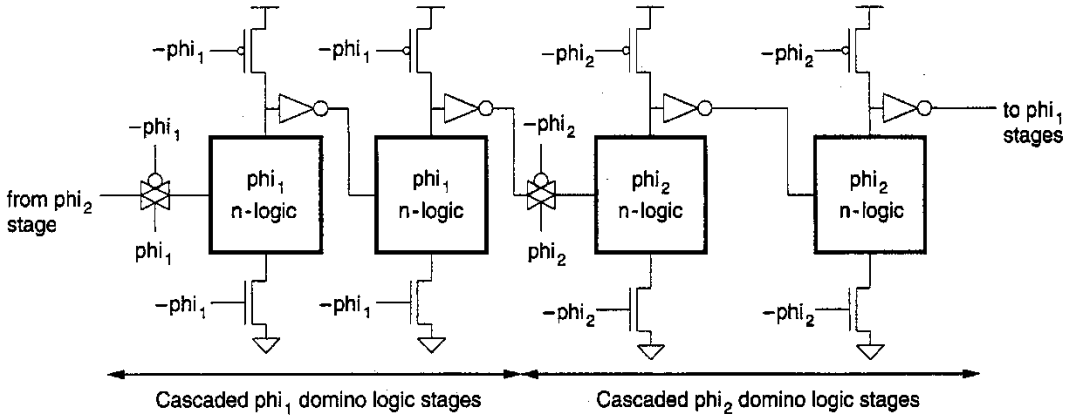
- a dynamic shift register implemented with a technique named "**ratiod dynamic logic**".
  - $\phi 1$ and $\phi 2$ are non-overlapping clocks
  - When $\phi 1$ is high, Cin1 charges to Vdd – Vt if Vin is high or to GND if Vin is low
  - When $\phi 1$ drops and $\phi 2$ comes up, the input data is trapped on Cin1 and yields a logic output on Cout1 which is transferred to Cin2
  - When $\phi 2$ drops and $\phi 1$ comes up again, the logic output on Cout1 is trapped on Cin2, which yields a logic output on Cout2, which is transferred to Cin3, etc.
  - To avoid losing too much voltage on the logic high level, $Cout_n >> Cin_{n+1}$ is desired
  - Each inverter must be ratioed to achieve a desired V_{OL} (e.g. when $\phi 2$ is high on 1st inv)

The bottom left dynamic shift register is a "**ratioless dynamic logic**" circuit

When $\phi 2$ is high transferring data to stage 2, $\phi 1$ has already turned off the stage 1 load transistor, allowing a $V_{OL} = 0$ to be obtained without a ratio condition between load and driver transistors.

Two-Phase Dynamic Logic

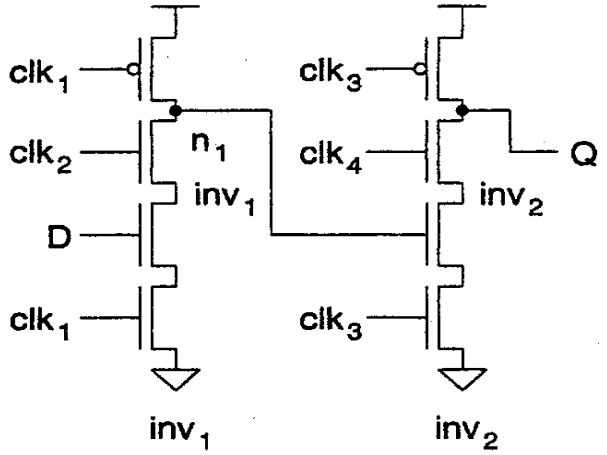




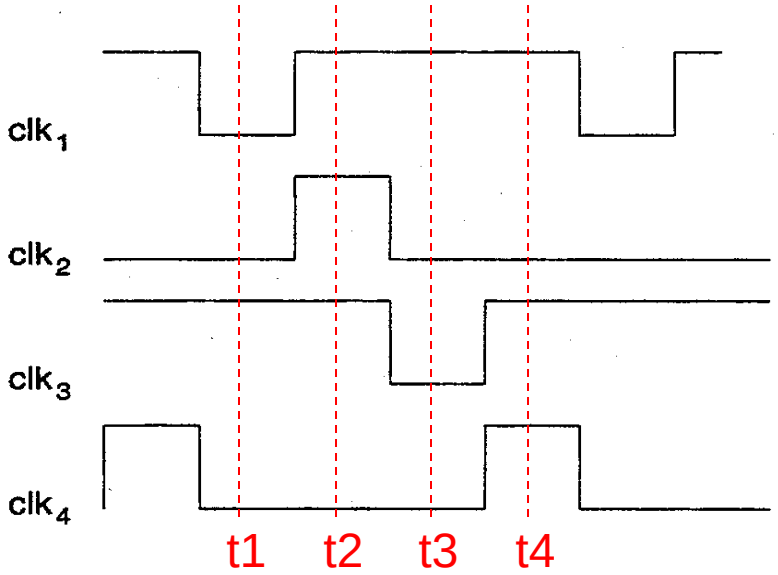
- Two phase dynamic logic similar to two phase dynamic register circuits
- Top figure shows n type logic stages with two phase non-overlapping clocks
  - phi1 high: precharge phi1 logic, evaluate phi2 logic
  - phi2 high: precharge phi2 logic, evaluate phi1 logic
- Bottom figure shows use of Domino logic having both phi1 and phi2 logic stages
  - Each block is separated from other by a clocked pass gate register/latch to store the logic result
  - Note that inverters must be used between successive stages of the same clock logic

Four Phase Clocking and Registers

- Four phase logic adds an evaluation phase to the existing precharge and evaluation phases of two-phase structures.
- Simple four-phase structure below illustrates operation:
  - during *t1* inverter1 is in **precharge** phase; node n1 charges to Vdd
  - during *t2* inverter1 **evaluates** since both NFET devices in n-tree leg are ON
  - during *t3* inverter2 precharges and inverter1 is in **hold** phase (i.e. both N and P devices are OFF isolating node n1
  - during *t4* inverter2 evaluates while inverter1 continues in **hold** phase
- Note that the hold phase is really two clock phases long
- Due to charge sharing during the *t2* phase, clk2 is sometimes replaced by clk12 (and clk4 is replaced by clk34) by keeping clk12 high during both *t1* and *t2* phases.

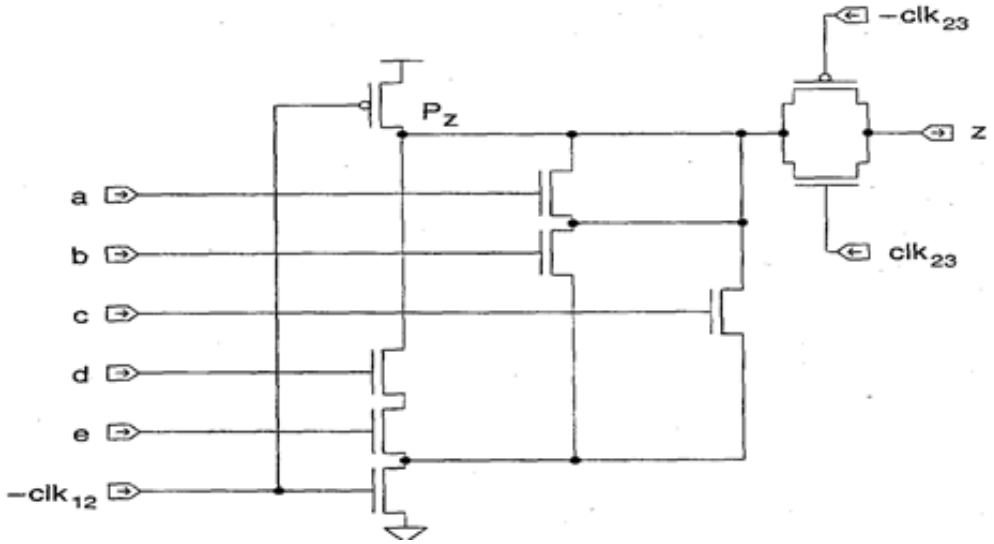


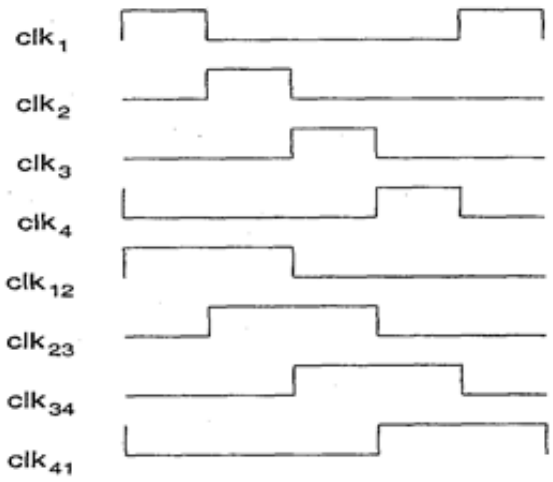
(a)



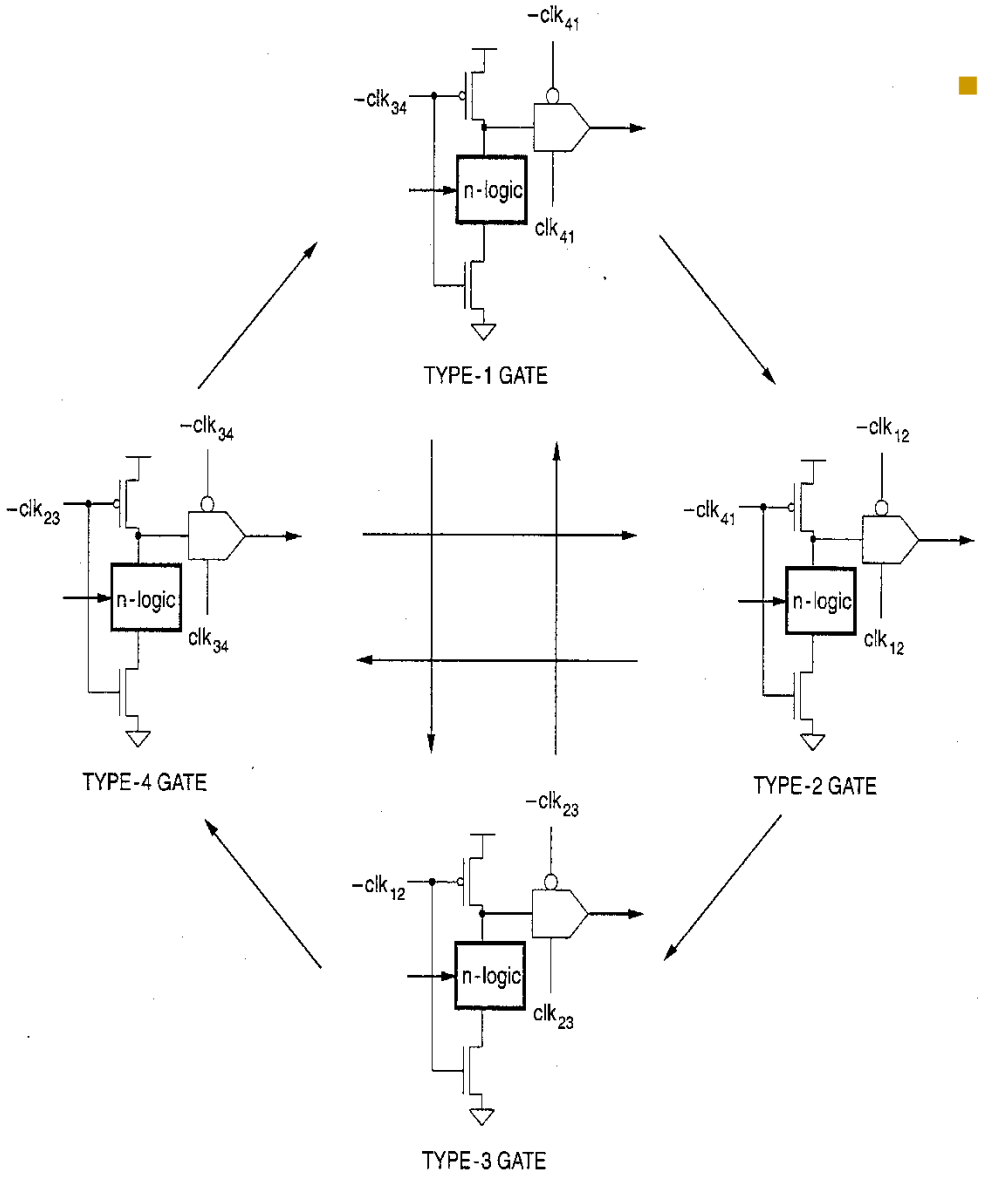
Four-Phase Logic Structures

- Four phase logic structure shown using transmission gate to isolate data on z during hold time:
  - during clk1 time, -clk12 is down causing Pz to be precharged to Vdd
  - during clk2 time, -clk12 is still down keeping precharge active, but clk23 goes high thus precharging node z
  - during clk3 time, precharge of node Pz ends and evaluation begins with Xgate still closed
  - during clk4 time, the transmission gate opens and the correct data is isolated on node z
- For the gate shown, z is valid during phases 4 and 1





Four Phase Logic: Allowable Interconnections



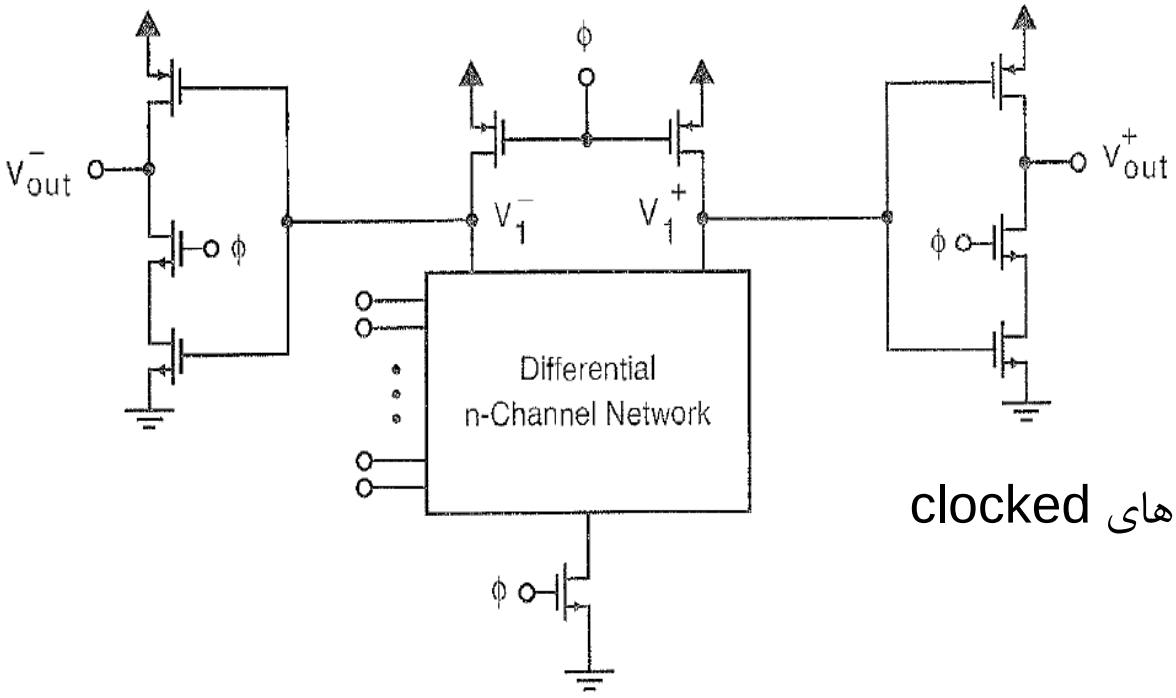
- Using four different type logic gates as shown in previous chart (where Type refers to the evaluation phase time), four phase logic can be used in pipelined logic structures where each type must be used per the allowable interconnection diagram at the left
  - a Type 1 gate can feed Type 2 or Type 3 gates
  - a Type 2 gate can feed Types 3 and 4
  - a Type 3 gate can feed Types 4 and 1
  - a type 4 gate can feed Types 1 and 2

Two-Phase Clock Generator

- Phi1 and Phi2 clocks may be generated from a master clock using a two-phase clock generator circuit
  - RS type cross-coupled latch with delay built into each feedback loop
  - Use an even number of inverters in each feedback loop
    - The delay built into the feedback loop sets the non-overlap period in the two out-of-phase clocks
  - NOR (or NAND) gates used to synchronize the generator with a master clock input
- Alternately, it may be desired to bring both phases on the chip as inputs and distribute both clocks globally

منطق تفاضلی NoRA

استفاده از لچ های clocked و منطق NORA

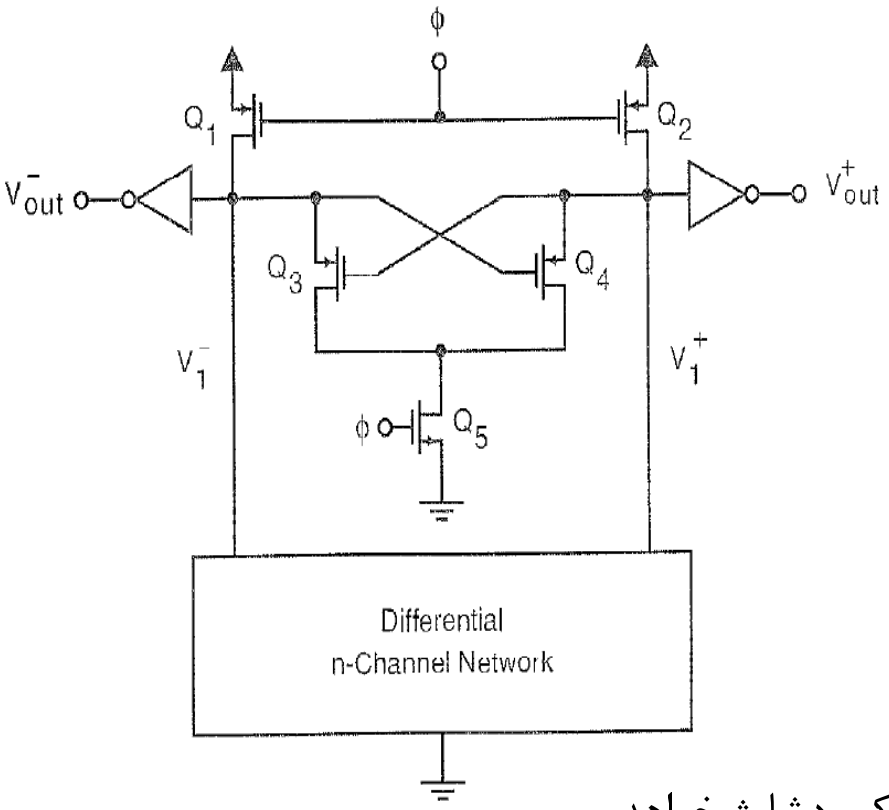


معکوس کننده های خروجی با لچ های clocked جایگزین شده است

یک ترانزیستور کانال p در هر لچ clocked حذف شده است چون وقتی گیت های تفاضلی در حال پیش شارژ هستند گره های V_1^+وV_1^- و در ولتاژهای بالا هستند و بنابر این لچ ها در وضعیت امپدانس بالای خود هستند، تضمین شده است بنابراین لچ های clocked دارای یک V_{DD} در امپدانس بالا می باشند بدون این که ترانزیستور کانال p اضافی وجود داشته باشد.

منطق تفاضلی احیا کننده regenerative

اگر مدارهای منطقی دارای شبکه های تحریک بزرگ با گره ها و ترانزیستورهای سری زیاد باشند، بسیار کند خواهند بود لذا از تقویت کننده های حساس برای تسریع آن استفاده می شود



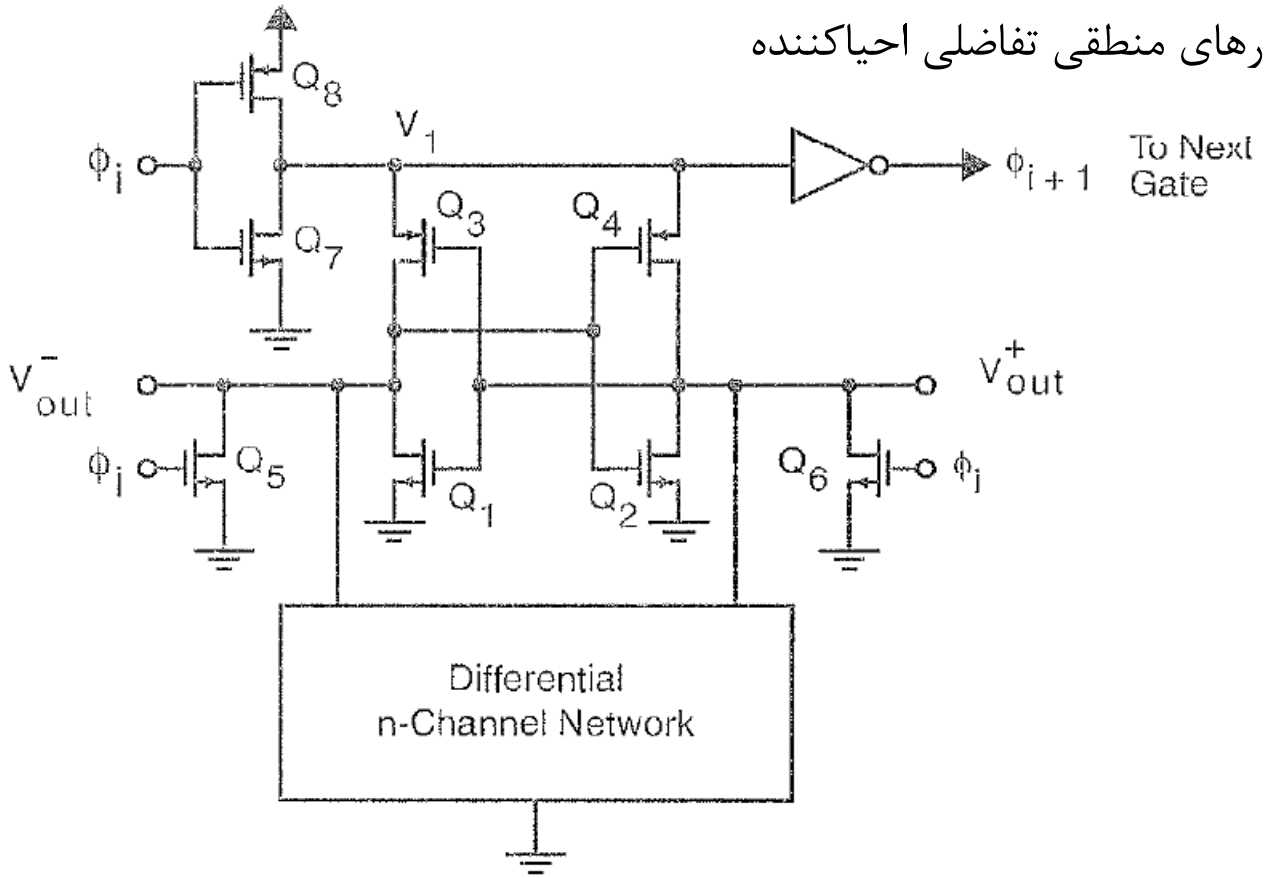
وقتی $\varphi=0$ ترانزیستورهای Q_1 و Q_2 روشن می شوند و چون ترانزیستورهای عریض هستند هر دو گره V_1^+وV_1^- را حدود V_{DD} می برند.

بسته به این که کدام طرف شبکه درختی کانال n تفاضلی دارای امپدانس کم باشد، یکی از گره های فوق در سطح ولتاژ پایینتری نسبت به دیگری قرار خواهد گرفت

وقتی $\varphi=1$، ترانزیستور بار کانال p پویا خاموش می شود و تقویت کننده حساس متشکل از Q_3، Q_4 و Q_5 به کار می افتد. این مدار تزویجی به سرعت یکی از گره های مذکور را که در ابتدا در ولتاژ کمتری قرار داشتند دشارژ می کند. گره دیگر که در ابتدا در V_{DD} قرار داشت تنها اندکی دشارژ خواهد

منطق تفاضلی احیا کننده

روش دیگری برای پیاده سازی مدارهای منطقی تفاضلی احیاکننده



تمامی گره ها زمانی که $\varphi_i = "1"$ است به سطح پایین پیش شارژ می شوند. زمانی که $\varphi_i = "0"$ شد V_1 به "۱" می رود که معکوس کننده های تزویجی متشکل از Q_1، Q_2، Q_3 و Q4 را روشن می کند که بسته به این که کدام طرف شبکه کانال n تفاضلی دارای امپدانس کم است موجب می شود V_{out}^{+} یا V_{out}^{-} بالا برود.

$\varphi_i + 1$ بلافاصله پس از این که خروجی ها به مقدار نهایی خود رسیدند پایین می رود. که به عنوان یک سیگنال ساعت برای گیت احیا ساز بعدی مورد استفاده قرارمی گیرد، آنگاه گیت بعدی را بلافاصله پس از این که ورودی آن ثابت شد فعال خواهد ساخت. نوعی منطق خود زمان ده Self-Timed logic

کدام خانواده منطقی را انتخاب کنیم؟

- Each of the circuit styles has its advantages and disadvantages
- Which one to select depends upon the primary requirement: ease of design, robustness, area, speed, or power dissipation, system clocking requirements, fan-out, functionality, ease of testing
- No single style optimizes all these measures at the same time
- Even more, the approach of choice can vary from logic function to logic function

منطق استاتیک؟

- The static approach has the advantage of being robust in the presence of noise
- This makes the design process rather trouble-free and amenable to a high degree of automation
- This ease-of-design does not come for free: for complex gates with a large fan-in, complementary CMOS becomes expensive in terms of area and performance
- Alternative static logic styles have therefore been devised Pseudo-NMOS is simple and fast at the expense of a reduced noise margin and static power dissipation

منطق دینامیک یا ترانزیستور عبور؟

- Pass-transistor logic is attractive for the implementation of a number of specific circuits, such as multiplexers and XOR- dominated logic such as adders
- Dynamic logic, on the other hand, makes it possible to implement fast and small complex gates
- This comes at a price. Parasitic effects such as charge sharing make the design process a precarious job
- Charge leakage forces a periodic refresh, which puts a lower bound on the operating frequency of the circuit

ترجیح غالب کدام است؟

- Current trend is towards an increased use of complementary static CMOS
- This tendency is inspired by the increased use of design-automation tools at the logic design level
- These tools emphasize optimization at the logic rather than the circuit level and put a premium on robustness
- Another argument is that static CMOS is more amenable to voltage scaling than some of other approaches

یک مقایسه

4-input NAND

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

* Dual Rail