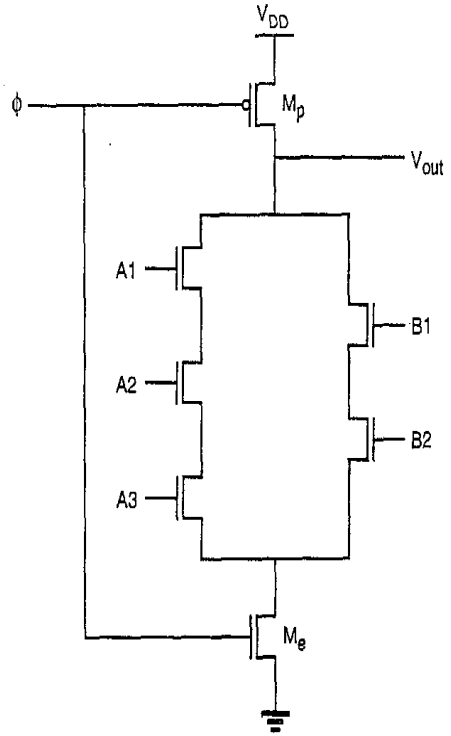
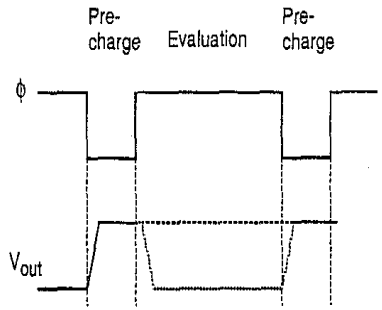
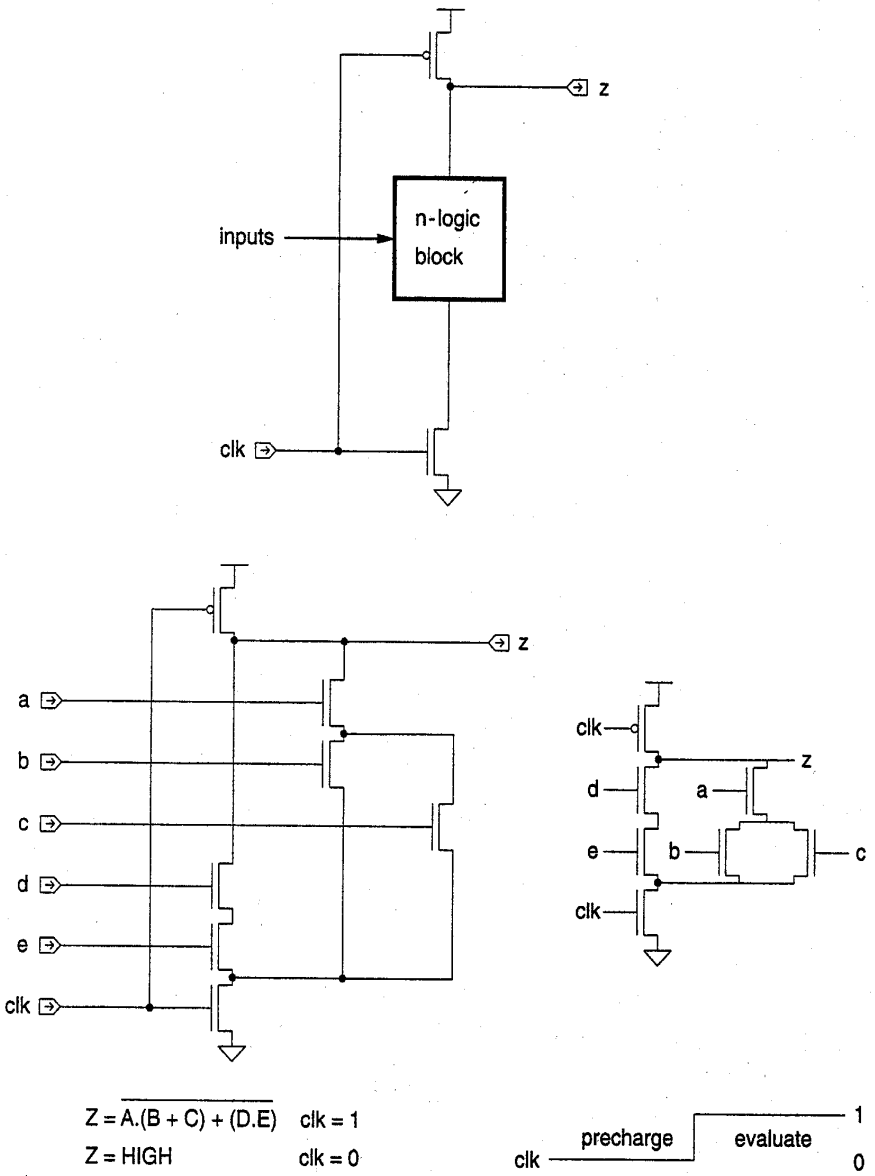
Dynamic CMOS Logic Gate





- In dynamic CMOS logic a single clock ϕ can be used to accomplish both the pre-charge and evaluation operations
  - When ϕ is low, PMOS pre-charge transistor Mp charges Vout to Vdd, since it remains in its linear region during final pre-charge
    - During this time the logic inputs A1 … B2 are active; however, since Me is off, no charge will be lost from Vout
  - When ϕ goes high again, Mp is turned off and the NMOS evaluate transistor Me is turned on, allowing for Vout to be selectively discharged to GND depending on the logic inputs
    - If A1 … B2 inputs are such that a conducting path exists between Vout and Me, then Vout will discharge to 0
    - Otherwise, Vout remains at Vdd

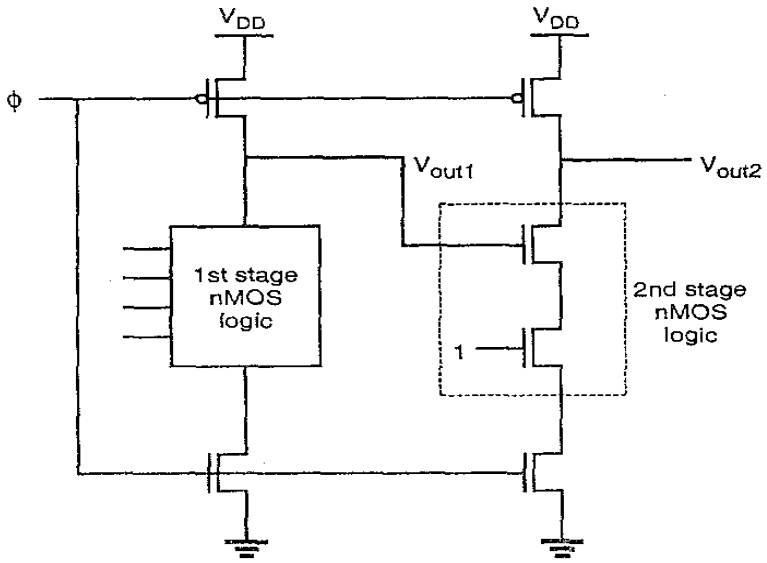
Dynamic CMOS Logic Circuits

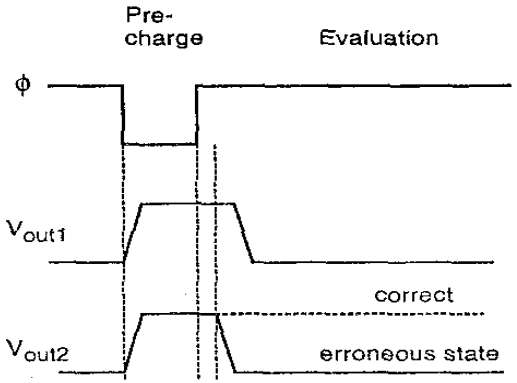


- Dynamic CMOS Logic circuits require a clock to precharge the output node and then to pull down the logic tree (assuming the logic inputs provide a path for current to flow)
  - Precharge Phase: clock is down turning on the P precharge transistor; N pull-down transistor is off. Output capacitance C_N charges to Vdd.
  - Evaluation Phase: clock goes high turning on the N pull down transistor and turning off the P precharge transistor. If logic inputs are such that neg Z is true, then output capacitance C_N discharges to ground.
  - No dc current flows during either the precharge or the evaluate phase.
  - Power is dynamic and is given by $P = C_N V_{dd}^2 f \alpha$ where C_N represents an equivalent total capacitance on the output, f = clock frequency, α =logic repetition rate

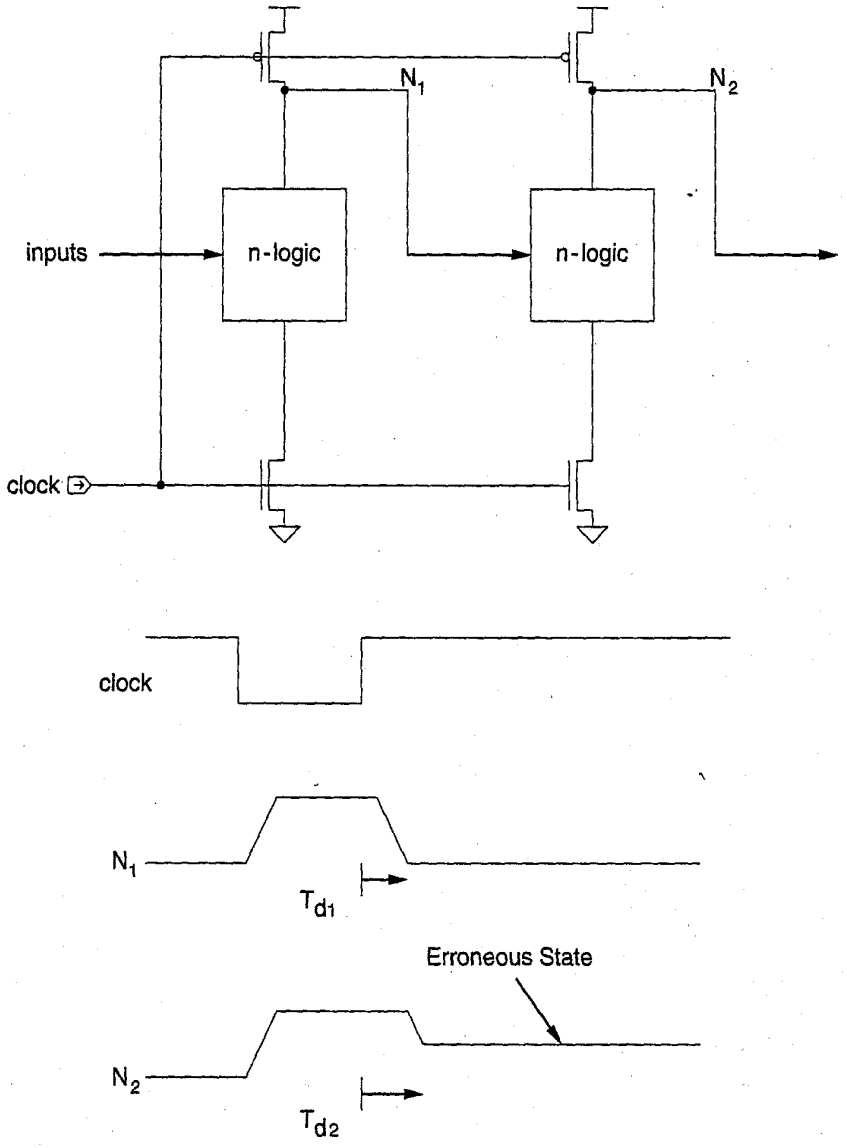
Cascading Problem in Dynamic CMOS Logic

- If several stages of the previous CMOS dynamic logic circuit are cascaded together using the same clock ϕ, a problem in evaluation involving a built-in "race condition" will exist
- Consider the two stage dynamic logic circuit below:
  - During **pre-charge**, both Vout1 and Vout2 are pre-charged to Vdd
  - When ϕ goes high to begin **evaluate**, all inputs at stage 1 require some finite time to resolve, but during this time charge may erroneously be discharged from Vout2
    - e.g. assume that eventually the 1st stage NMOS logic tree conducts and fully discharges Vout1, but since all the inputs to the N-tree all not immediately resolved, it takes some time for the N-tree to finally discharge Vout1 to GND.
    - If, during this time delay, the 2nd stage has the input condition shown with bottom NMOS transistor gate at a logic 1, then Vout2 will start to fall and discharge its load capacitance until Vout1 finally evaluates and turns off the top series NMOS transistor in stage 2
  - The result is an error in the output of the 2nd stage Vout2



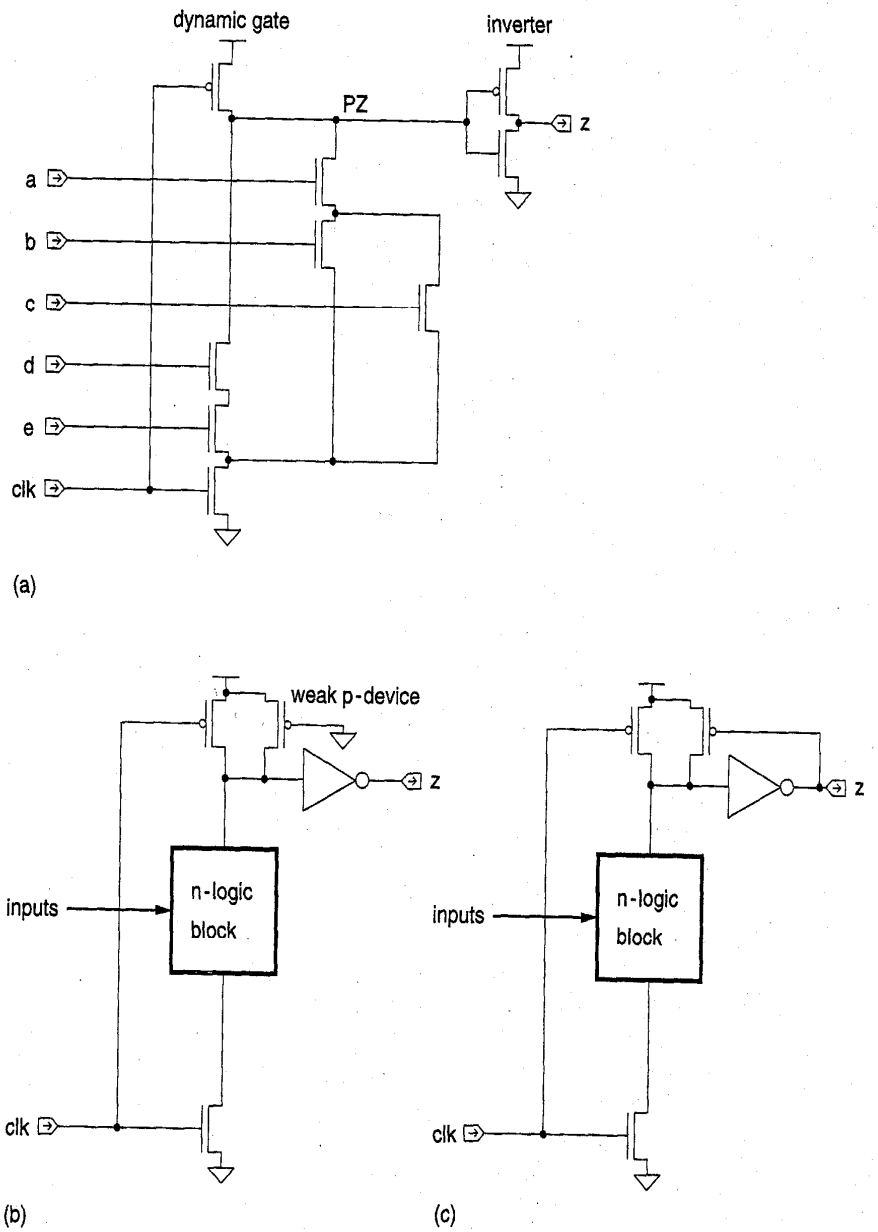


Cascaded Dynamic CMOS Logic Gates: Evaluate Problem



- With simple cascading of dynamic CMOS logic stages, a problem arises in the evaluate cycle:
  - The pre-charged high voltage on Node N2 in stage 2 may be inadvertently (partially) discharged by logic inputs to stage 2 which have not yet reached final correct (low) values from the stage 1 evaluation operation.
  - Can not simply cascade dynamic CMOS logic gates without preventing unwanted bleeding of charge from pre-charged nodes
- Possible Solutions:
  - two phase clocks
  - use of inverters to create Domino Logic
  - NP Domino Logic
  - Zipper/NORA logic

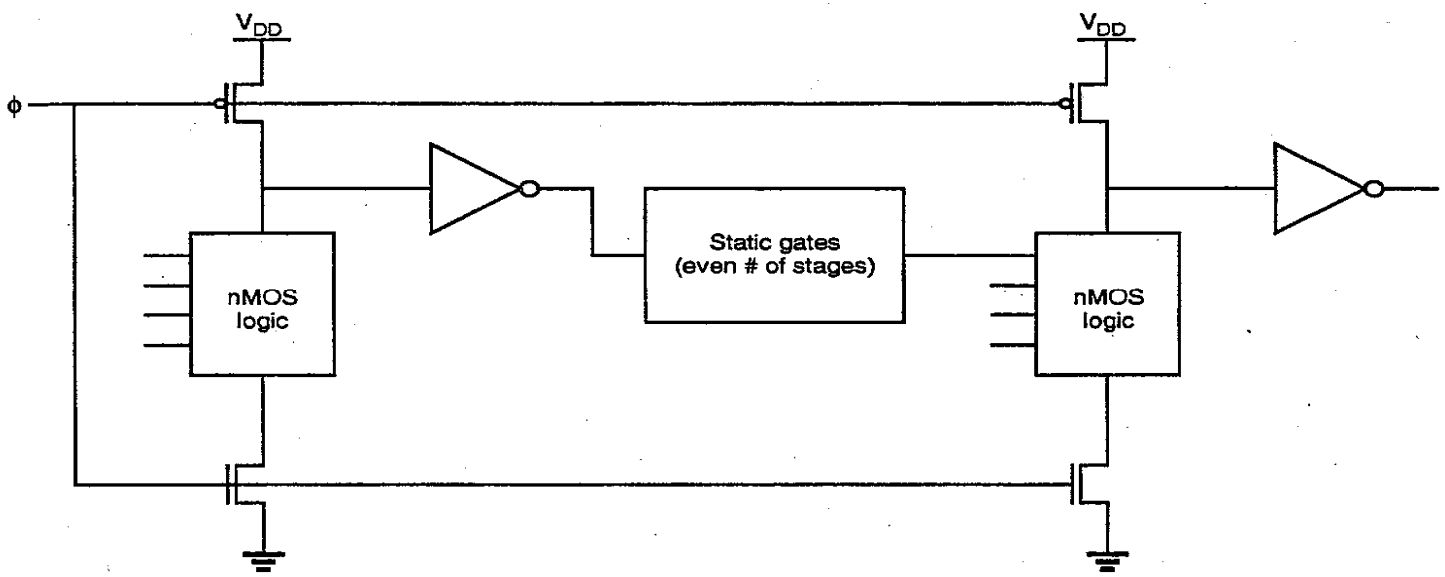
CMOS Domino Logic



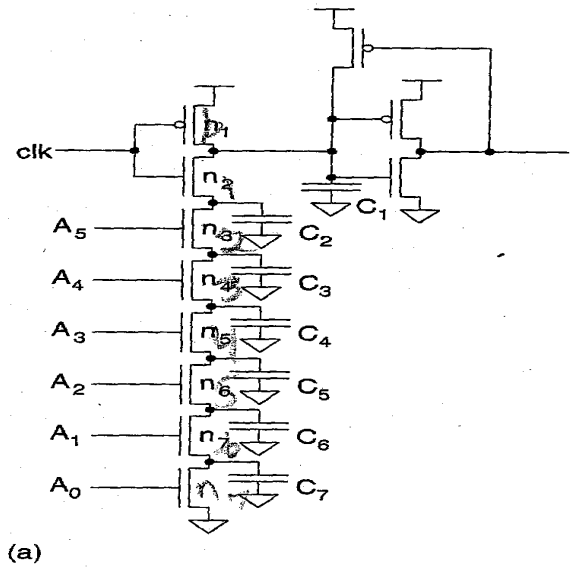
- The problem with faulty discharge of precharged nodes in CMOS dynamic logic circuits can be solved by placing an inverter in series with the output of each gate
  - All inputs to N logic blocks (which are derived from inverted outputs of previous stages) therefore will be at zero volts during precharge and will remain at zero until the evaluation stage has logic inputs to discharge the precharged node PZ.
  - This circuit approach avoids the race problem of “vanilla” cascaded dynamic CMOS
  - However, all circuits only provide non-inverted outputs
- In (b) a weak P device compensates for charge loss due to charge sharing and leakage at low frequency clock operation
- In (c) the weak P device can be used to latch the output high

Mixing Domino CMOS Logic with Static CMOS Logic

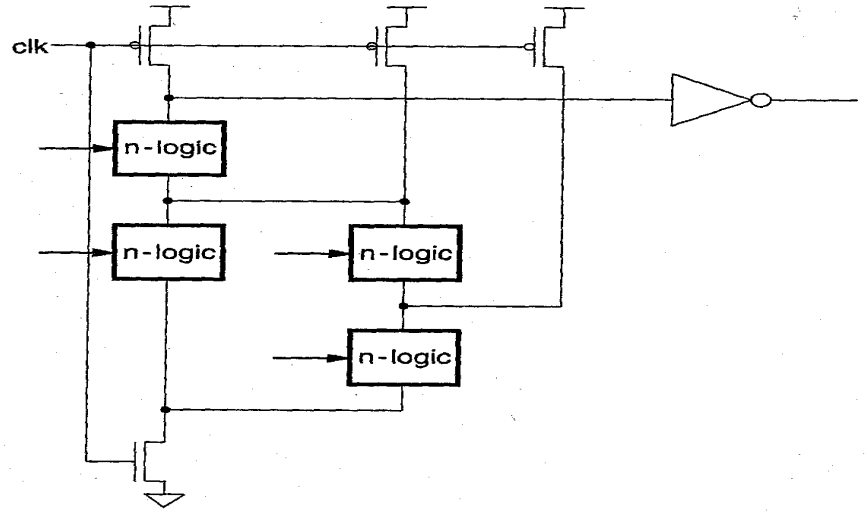
- We can add an **even** number of static CMOS inverting logic gates after a Domino logic stage prior to the next Domino logic stage
  - Even number of inverting stages guarantees that inputs to the second Domino logic stage experience only 0-to-1 transitions (since 1-to-0 transitions may cause an erroneous logic level as discussed in prior charts 5-67 and 5-68)
- In the cascaded **Domino** logic structure, the evaluation of each stage ripples through the cascaded stages similar to a chain of Dominos (from which it takes the name)
  - The evaluate cycle must be of sufficient duration to allow all cascaded logic stages (between latches) to complete their evaluation process within the **clock evaluation interval**



CMOS Domino Logic Design Hazards



(a)

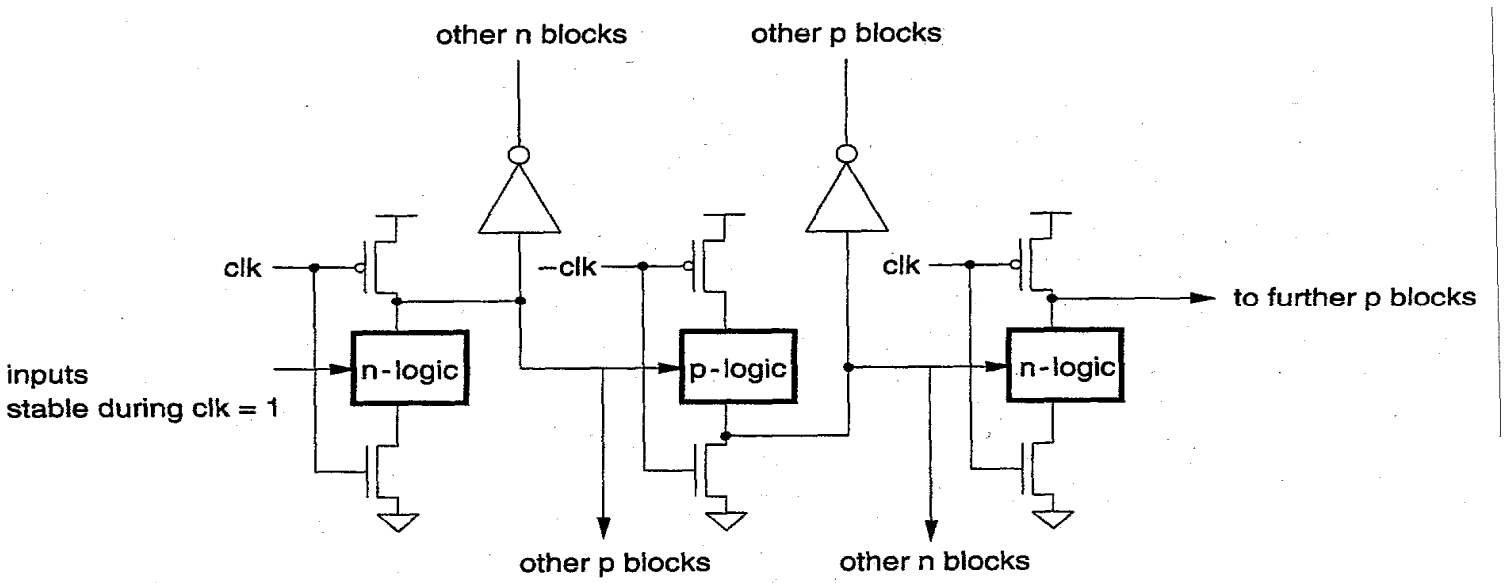


(b)

- In (a) the N evaluate transistor is placed nearest to the output C1 node (poor design)
  - 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, some or all of capacitors C2-C7 will bleed charge from the larger node capacitor C1, thus reducing the voltage on C1.
    - 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 thus allowing the possibility of getting C2-C7 charged during the precharge phase
- Using additional precharge P transistors (as in b) to charge intermediate nodes in a complex logic tree will help with the charge sharing problem.

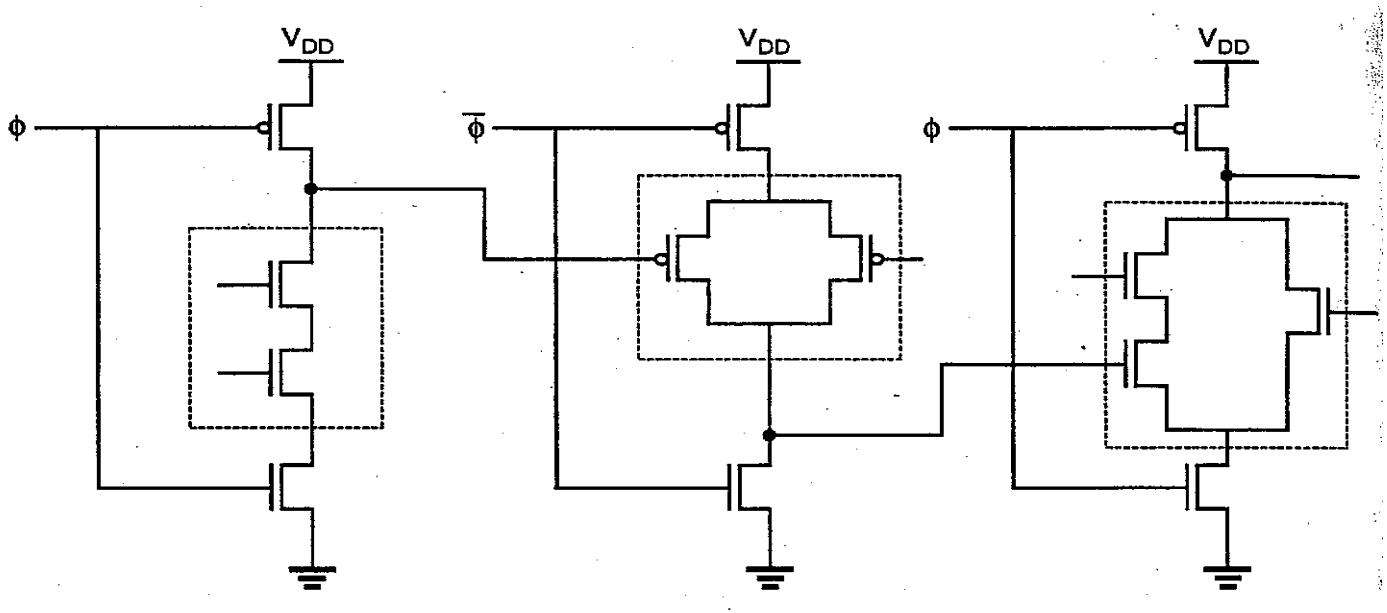
NP Domino Logic (NORA Logic)

- An elegant solution to the dynamic CMOS logic "erroneous evaluation" problem is to use NP Domino Logic (also called NORA logic) as shown below.
  - Alternate stages of N logic with stages of P logic
    - N logic stages use true clock, normal precharge and evaluation phases, with N logic tree in the pull down leg. P logic stages use a complement clock, with P logic stage tied above the output node.
    - During precharge clk is low (-clk is high) and the P-logic output precharges to ground while N-logic outputs precharge to Vdd.
    - During evaluate clk is high (-clk is low) and both type stages go through evaluation; N-logic tree logically evaluates to ground while P-logic tree logically evaluates to Vdd.
- Inverter outputs can be used to feed other N-blocks from N-blocks, or to feed other P-blocks from P-blocks.

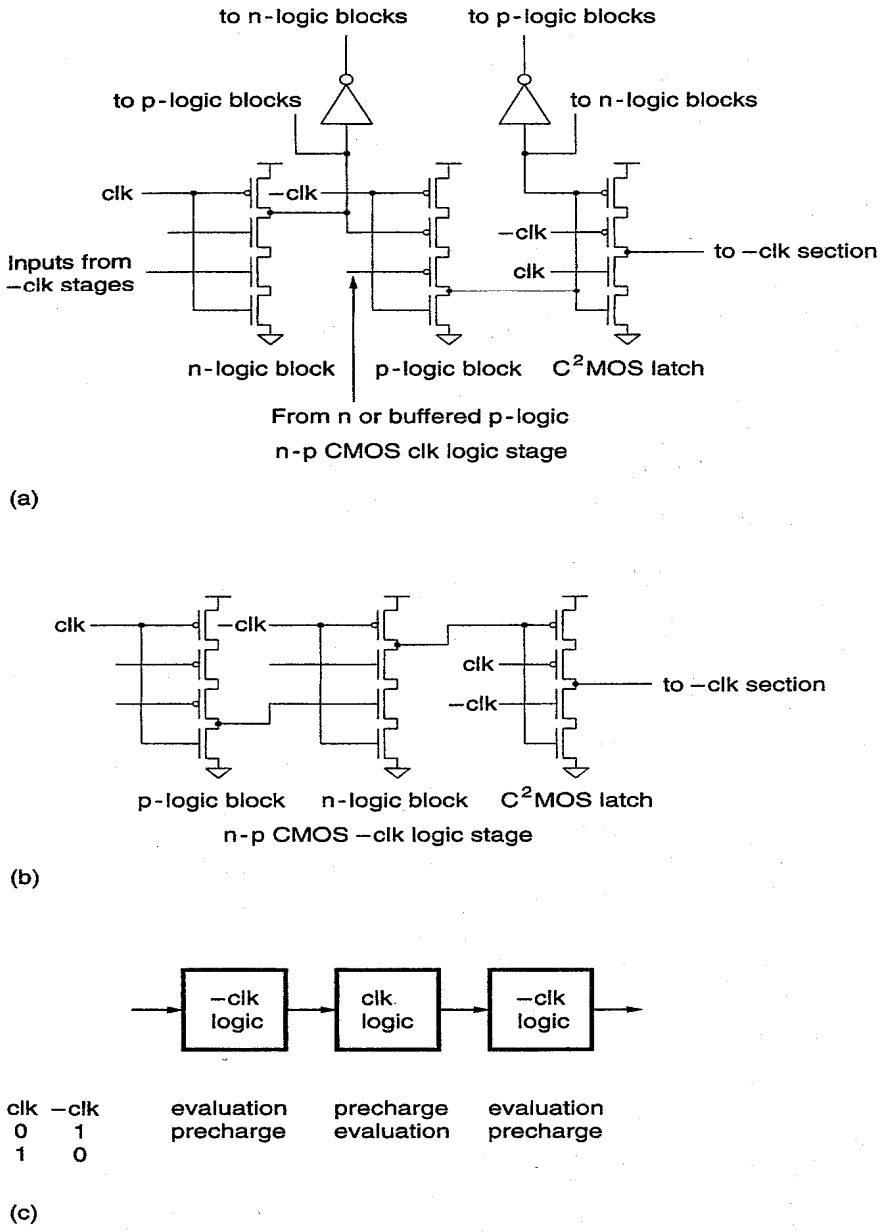


NORA CMOS Logic Circuit Example

- An example of NP or NORA (No Race) logic is shown below:
- During ϕ low (ϕ' high), each stage pre-charges
  - N logic stages pre-charge to Vdd; P logic stages pre-charge to GND
- When ϕ goes high (ϕ' low), each stage enters the evaluation phase
  - N logic evaluates to GND; P logic stages evaluate to Vdd
  - All NMOS and PMOS stages evaluate one after another in succession, as in Domino logic
- Logic below:
  - Stage 1 is X = (A · B)'
  - Stage 2 is G = X' + Y'
  - Stage 3 is Z = (F · G + H)'

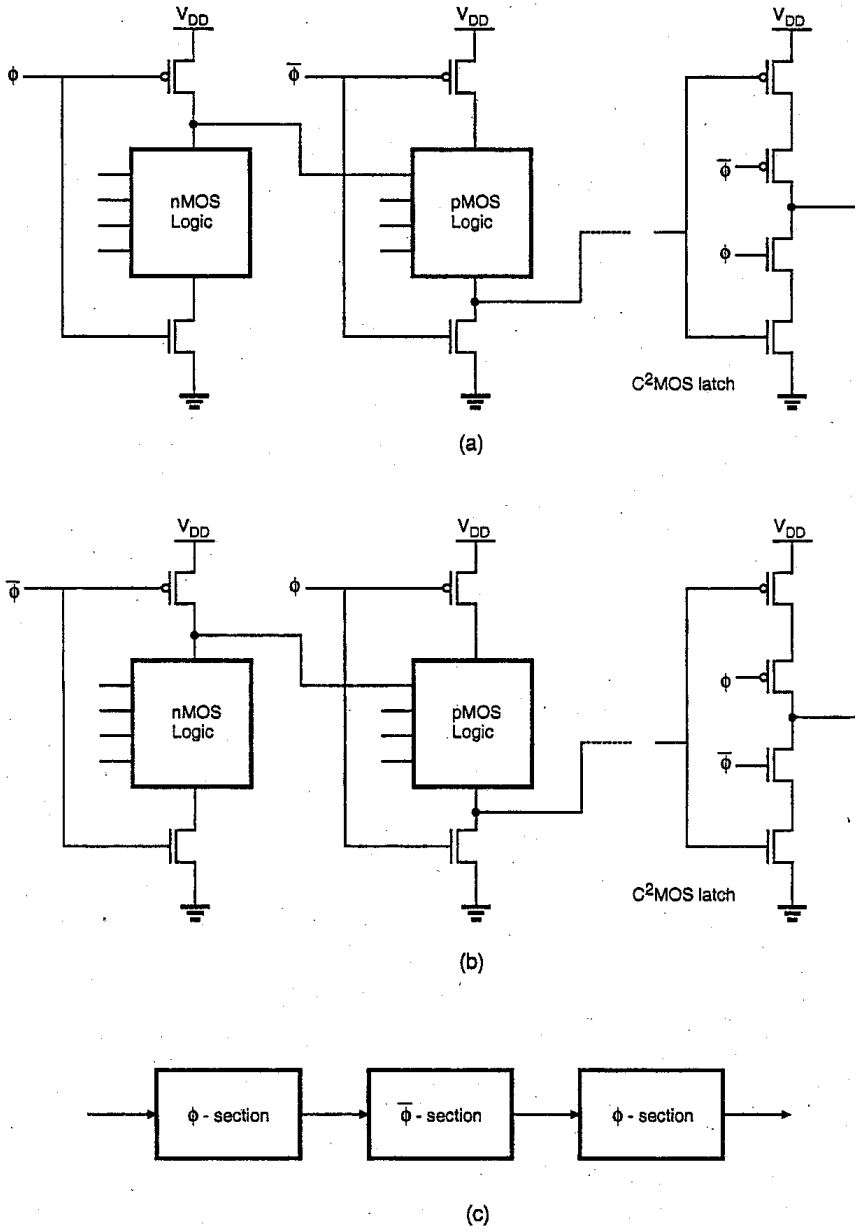


Single-Phase NP Dynamic Logic Structures



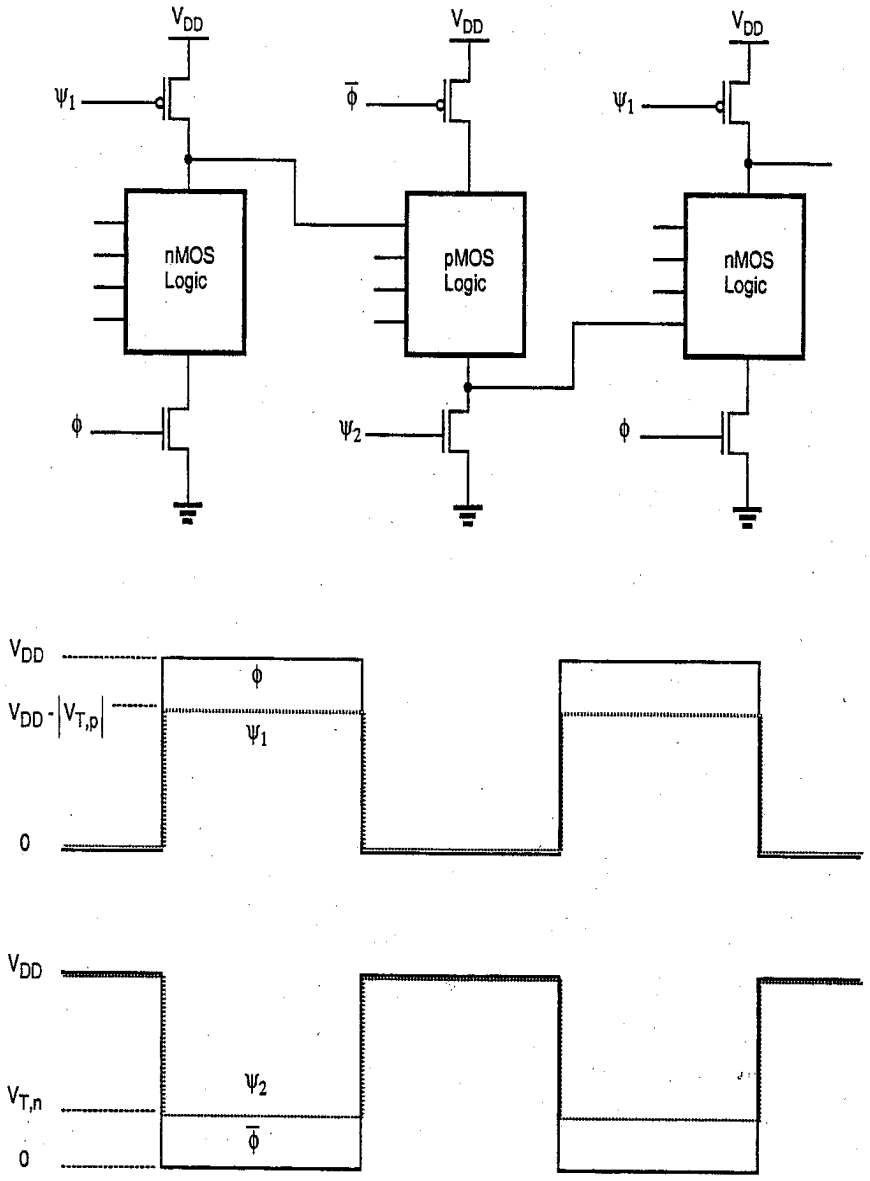
- Combines NP Domino logic sections with C^2MOS latch
  - n-logic block can drive p-logic block or another n-logic block with a static inverter
  - similarly for a p-logic block
  - Must end in a C^2MOS latch
- clk logic: (a) prechrg on clk=0, eval clk=1
- -clk logic: (b) pre on clk=1, eval on clk=0
- clk logic can feed –clk logic & vice-versa
- can mix static logic with NP domino logic
- Rules to avoid race conditions:
  - During precharge, logic blocks are OFF
  - During eval, internal inputs make only one transition
- Pipeline design:
  - Even # of inversions between C^2MOS, or
  - at least 1 dynamic stage and even # inversions prior to it

Pipelined NORA CMOS Circuit Operation



- With pipelined NORA CMOS logic design
  - one can alternate N and P stages between C^2MOS latches where ϕ high is used for evaluation as shown in (a)
  - Or, one can alternate N and P stages similarly between C^2MOS 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 C^2MOS latch

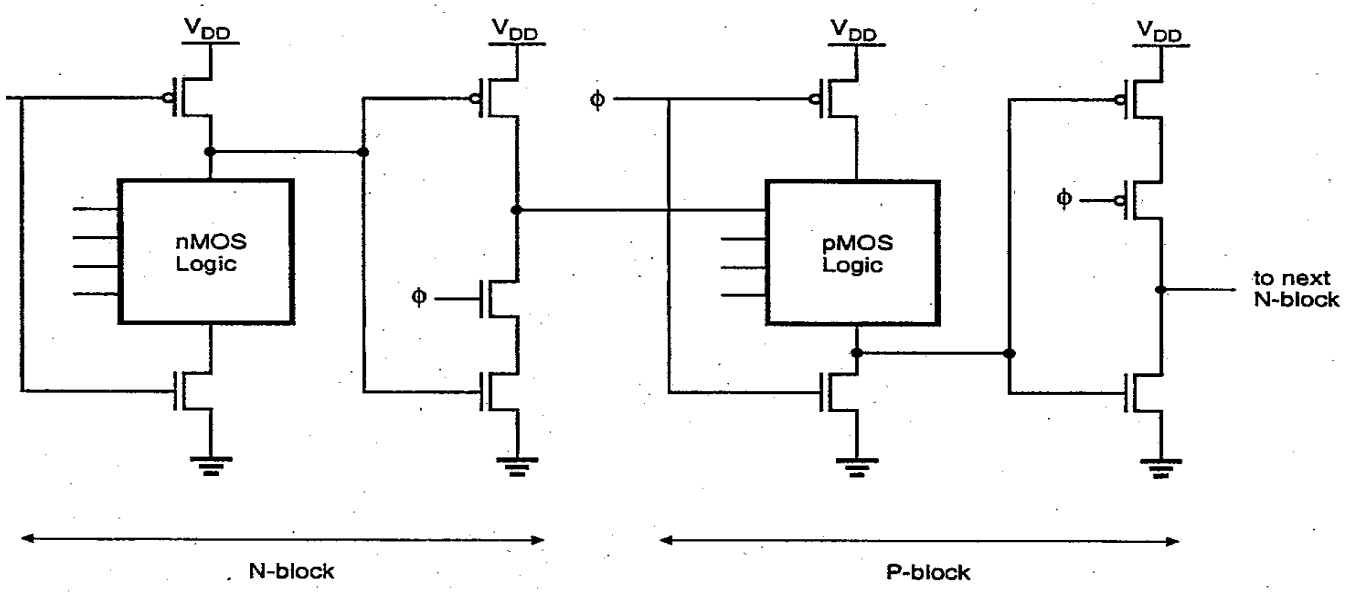
Zipper CMOS Dynamic Logic



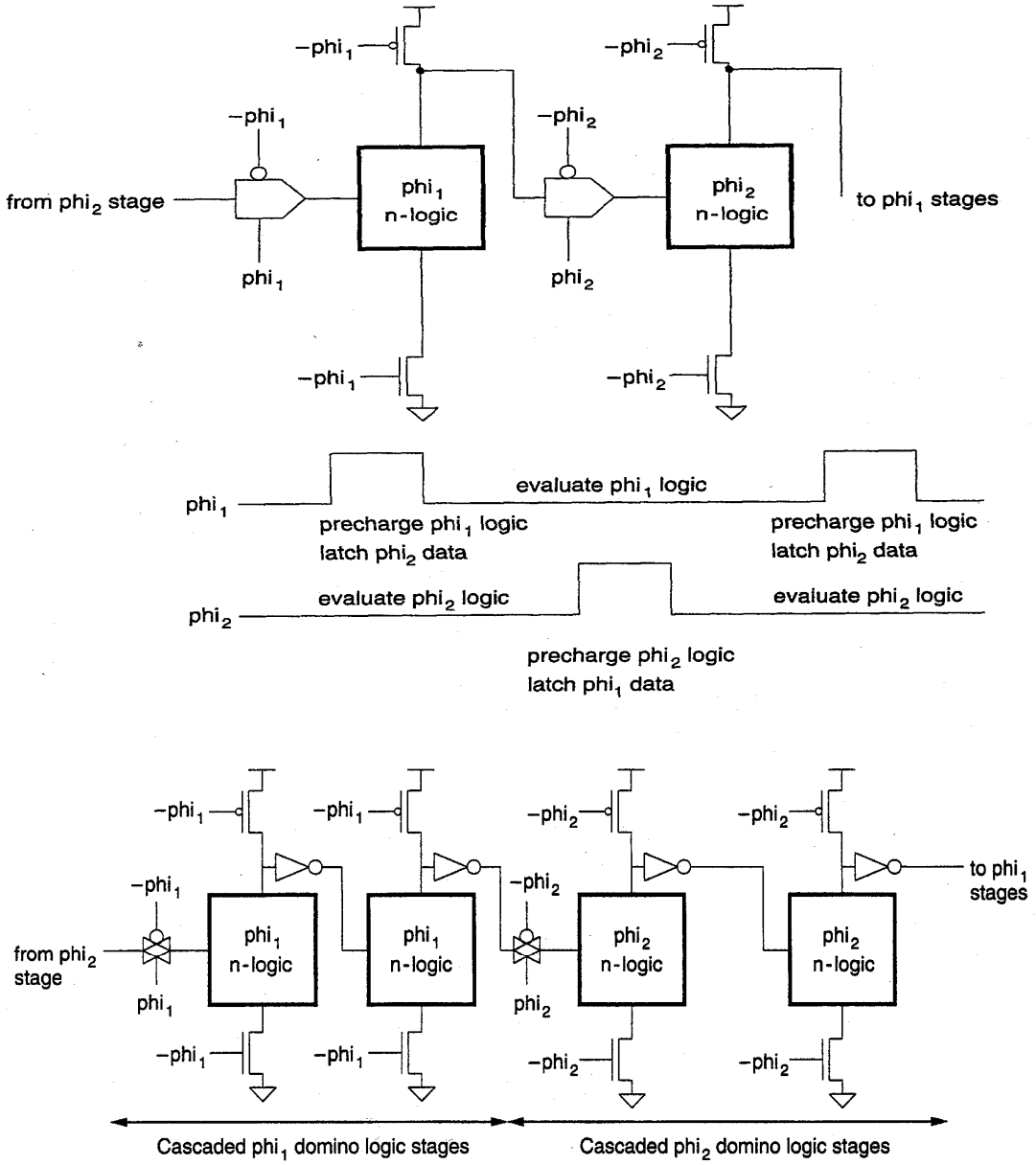
- **Zipper CMOS logic** is a scheme for improving charge leakage and charge sharing problems
- Pre-charge transistors receive a slightly modified clock where the clock pulse (during pre-charge off time) holds the pre-charge transistor at weak conduction in order to provide a trickle pre-charge current 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

Pipelined True Single Phase Clock (TSPC) CMOS

- A true single phase clock 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



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