Dynamic Logic Circuits

A. Marzuki

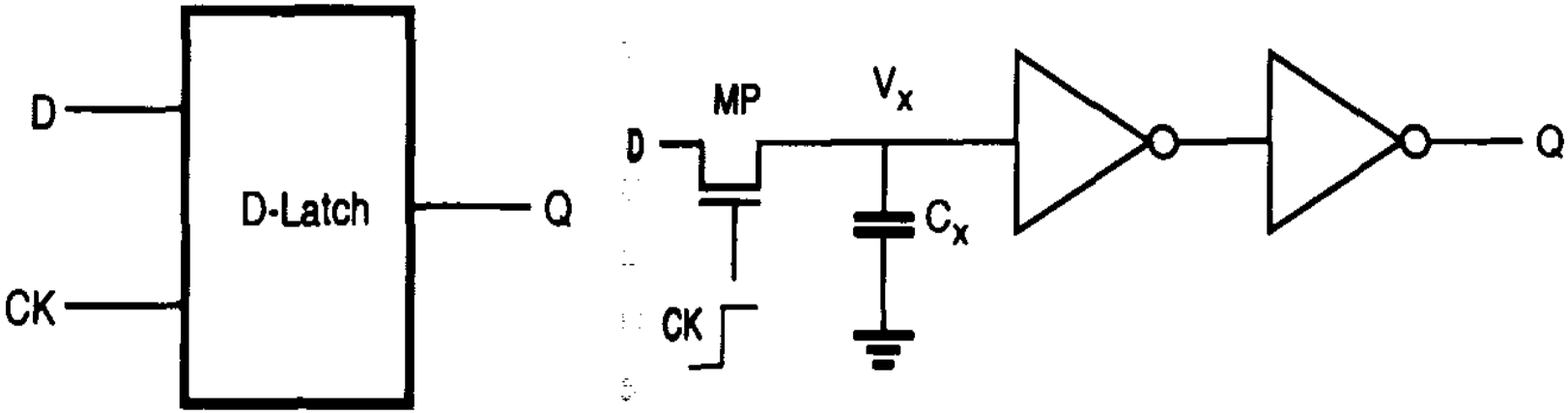
Introduction

- Introduction
- Pass Transistor Circuits
- Voltage Bootstrapping
- Synchronous Dynamic Circuit Techniques
- High-Performance Dynamic CMOS Circuits

Introduction

- Dynamic logic is temporary (**transient**) in that output levels will remain valid only for a certain period of time
  - Static logic retains its output level as long as power is applied
- Dynamic logic is normally done with charging and selectively discharging capacitance (i.e. capacitive circuit nodes)
  - **Precharge** clock to charge the capacitance
  - **Evaluate** clock to discharge the capacitance depending on condition of logic inputs
- Advantages over static logic:
  - Avoids duplicating logic twice as both N-tree and P-tree, as in standard CMOS
  - 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

Example



CK = 1 -> MP on

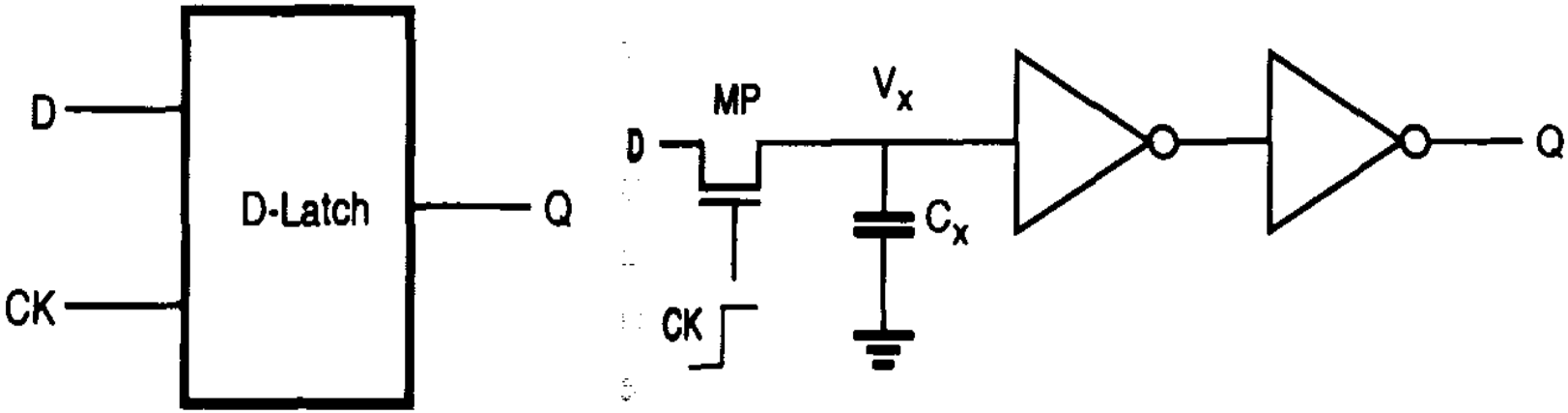
capacitor charge up or down

Q = D

CK = 0 -> MP off, capacitor isolated from D

No current path, Q = previous charge

Example



x soft node

Eg: $V_{DD} = V_{OH} = 5$ V, $V_{OL} = 0$ V, $V_{IL} = 2.1$ V, $V_{IH} = 2.9$ V

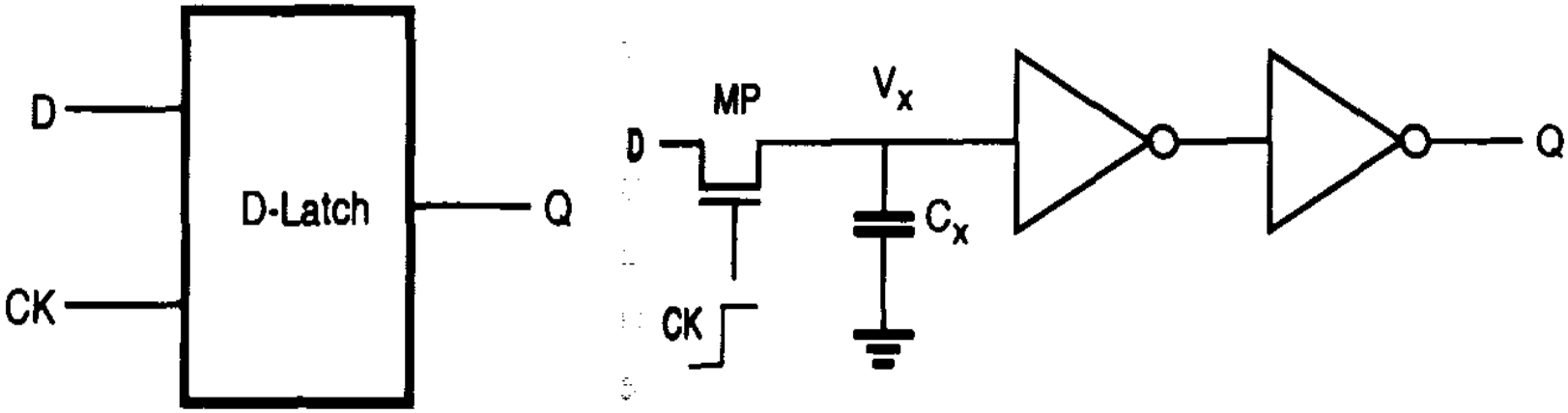
CK = 1 -> $V_{in} = 5$ V

C_x charge up

$V_x = 4.2$ V (poor conductor) -> $V_{out} = 0$ V (1st inverter)

$V_{out} = 5$ V (2nd inverter)

Example



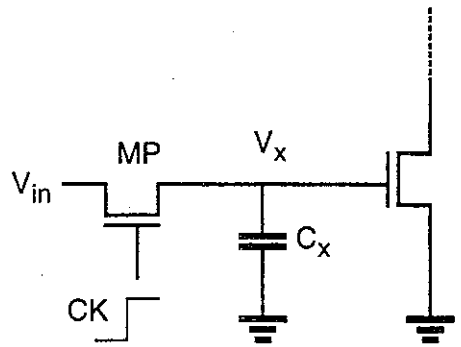
transistor turns off, logic high is preserved by C_x

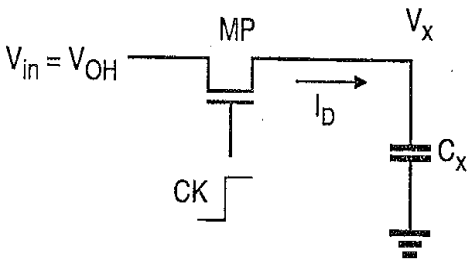
V_x drops from 4.2 V -> to whatever (charge leakage)

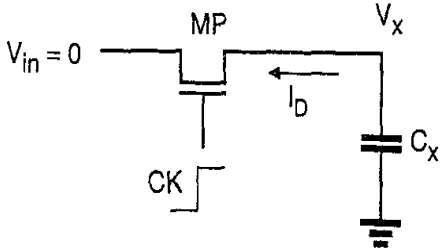
$V_Q = 5$ V requires $V_x > 2.9$ V

C_x must be restored

Basic Principle of Pass Transistor Circuits (PTC)

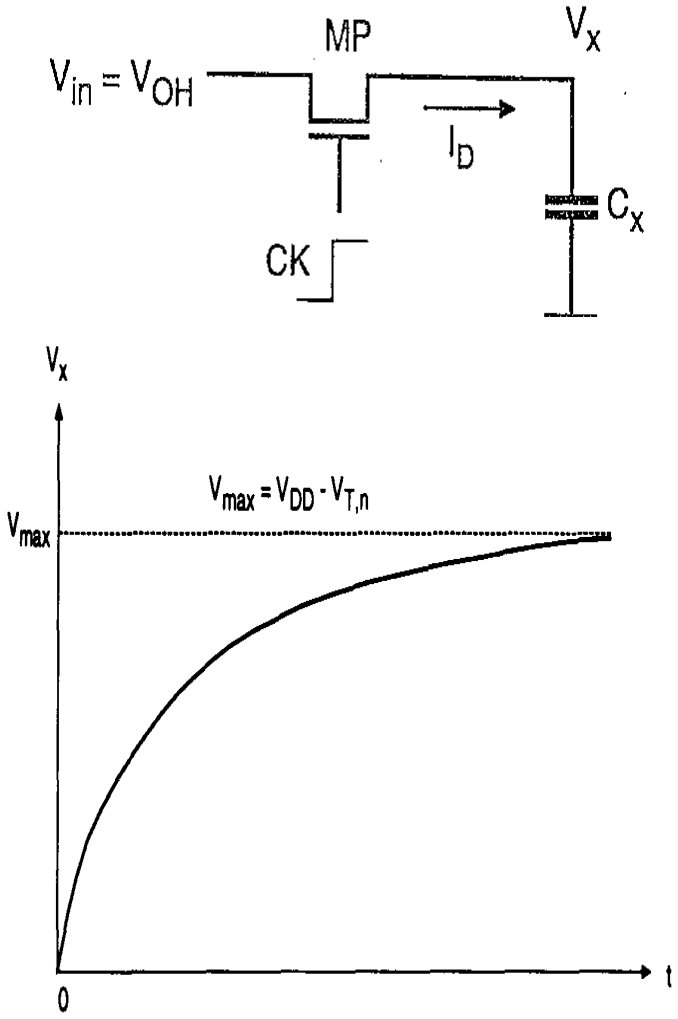






- The basic **dynamic logic gate concept** is shown at left (top)
  - the pass transistor MP is an NMOS device, but could also be implemented with a transmission gate TG
  - C_x represents the equivalent capacitance of the input gate of the second NMOS device (part of an inverter or logic gate) as well as the PN junction capacitance of MP's drain (source)
  - When clock CK goes high, MP is turned on and allows the input voltage Vin to be placed on capacitor Cx
    - Vin could be a high ("1") or a low ("0") voltage
  - When CK goes low, MP is turned off, trapping the charge on Cx
- Operation for a 1 or a 0:
  - If Vin is high (say V_{OH}), then MP will allow current to flow into Cx, charging it up to Vdd – Vtn (assume CK up level is Vdd)
  - If Vin is low (say GND), then MP will allow current to flow out of Cx, discharging it to GND
- Due to leakage from the drain (source) of MP, Cx can only retain the charge Q for a given period of time (called soft node)
  - If MP is NMOS, Cx will discharge to GND
  - If MP is PMOS, Cx will discharge to VDD
  - If MP is a TG, Cx could discharge in either direction

Transfer “1” Event



- Charging event with NMOS operating in source-follower mode:
  - MP will be saturated during transfer “1” transient
  - Max voltage attainable at Vx will be Vdd – Vtn, assuming that the CK pulse height is Vdd
  - Solve for increasing voltage Vx versus time:

    $$\mathbf{C_x\,(dV_x/dt) = \tfrac{1}{2}\,k_n(Vdd - V_x - Vtn)^2}$$

  - Solution:

$$\mathbf{t = (2Cx/\,k_n)[\{1/(Vdd - Vx - Vtn) - 1/(Vdd - Vtn)\}]}$$

or, solving for $V_x(t)$

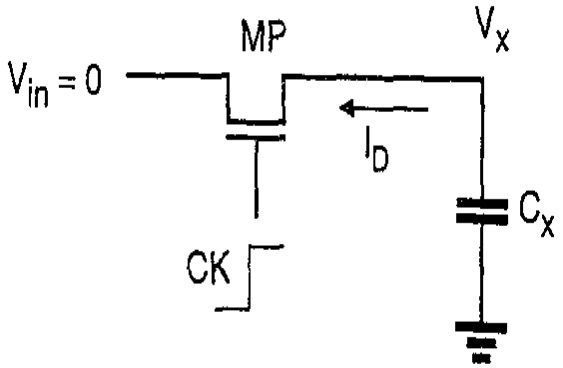
$$\mathbf{V_x(t) = (Vdd - Vtn)[1 - 1/\{1 + (Vdd - Vtn)(k_n/2C_x)t\}]}$$

As t → infinity, $V_x(t)$ → Vdd – Vtn

Solve for time needed to reach 90% (Vdd – Vtn):

- Set $V_x(t) = 0.9\,(Vdd - Vtn)$ ➔ $\mathbf{t_{90\%} = 18\,C_x/\,k_n(Vdd - Vtn)}$
  - i.e. 18 time constants

Transfer “0” Event



- On a transfer “0” event, the NMOS transfer device is in its common source configuration, i.e. the source is at GND and the drain is discharging Cx
  - MP is operating in the linear mode for the entire transient since the starting value is Vdd – Vtn
  - Solve for decreasing Vx with time:

  $$\mathbf{C_x\,(dV_x/dt) = -\,\beta_n\,V_x\,(Vdd - Vtn - \tfrac{1}{2}\,V_x)^2}$$

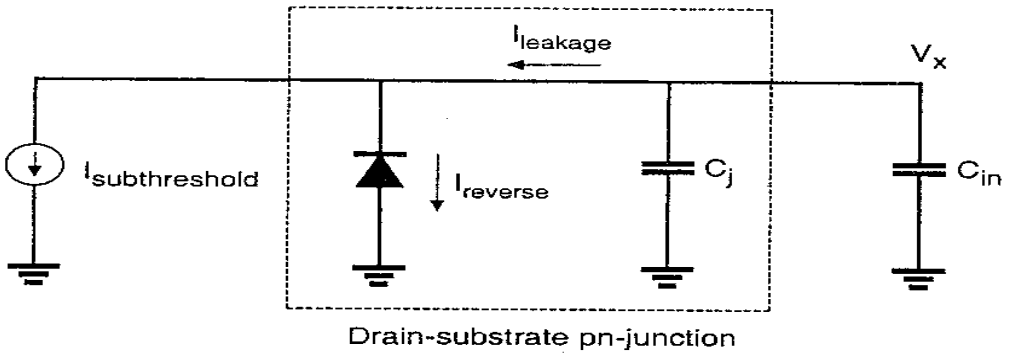
  - Solution:

  $$\mathbf{t = Cx/(\beta_n(Vdd - Vtn))\ ln\{(2(Vdd - Vtn) - Vx)/Vx\}}$$

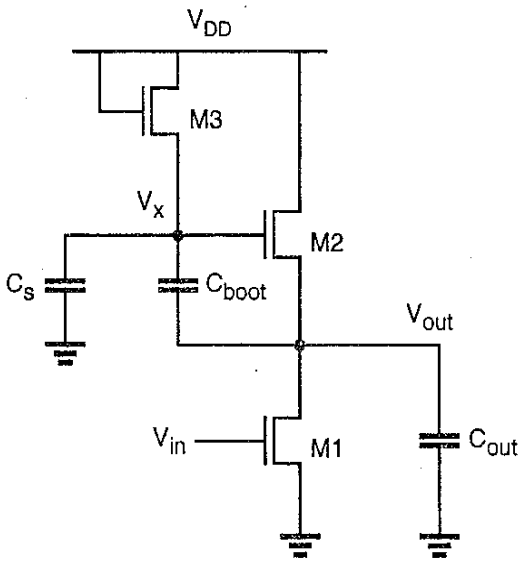
- Solve for time needed for Vx to fall to 10% (Vdd – Vtn):
  - Set $V_x(t) = 0.1\,(Vdd - Vtn)$ ➔ $\mathbf{t_{10\%} = 2.9\,C_x/\,\beta_n(Vdd - Vtn)}$
    - i.e. 2.9 time constants
- Therefore, the time to discharge Cx with an NMOS MP pass transistor is much shorter than the time to charge Cx due to the source-follower operation during charging.

Leakage and Subthreshold Current in Dynamic Pass Gate

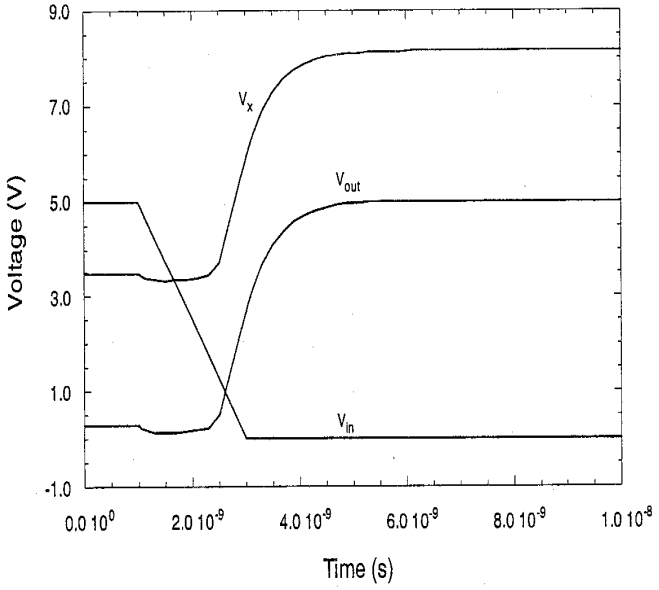
- Charge can leak off the storage capacitor Cx mainly from two sources:
  - PN junction leakage of the NMOS drain (source) junction
  - Subthreshold current (I_{OFF}) through MP when its gate is down at zero volts
- One can solve for the maximum amount of time Δt that charge can be retained on Cx using the differential equation C dv/dt = I, where
  - I is the total of the reverse PN junction leakage and the I_{OFF} current
  - C is the total load capacitance due to gate, junction, wire, and poly capacitance
  - the maximum allowable ΔV in order to preserve the logic "1" level is known
    - Typically ΔV ~ Vdd – Vtn – ½ Vdd = ½ Vdd – Vtn
- The minimum frequency of operation can be found from f ~ 1/(2 Δt)

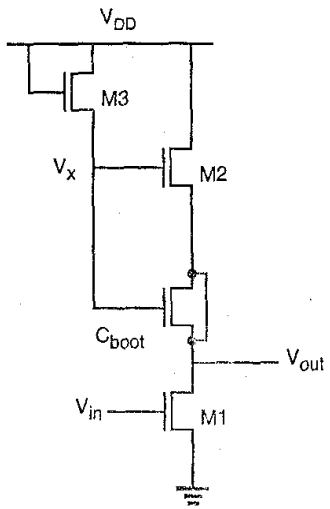


Dynamic Bootstrapping Technique



- **Bootstrapping** is a technique that is sometimes used to charge up a transistor gate to a voltage higher than Vdd when that transistor has to drive a line to the full Vdd
- At left is a NMOS bootstrap driver often used in memory circuits to drive a highly capacitive word line
- Operation:
  - When Vin = high, M1 is on holding Vout low while M3 charges Vx to Vdd – Vt. Thus, C_{boot} is charged to Vdd – Vt – V_{OL}
  - When Vin goes low, turning M1 off, M2 starts charging Vout high. If $C_{boot} > C_s$, most of the increase in Vout is "booted" to Vx, raising the voltage at Vx to well above Vdd.



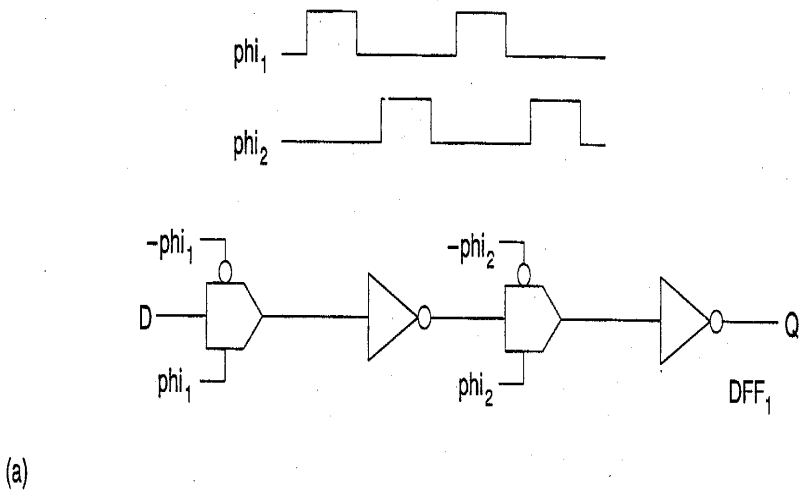


- It is desired to obtain Vx > Vdd + Vt in order to keep M2 linear, to allow Vout to be charged fully to Vdd.
- Parasitic capacitor C_s bleeds some of the charge off C_{boot}, limiting the max voltage on Vx (charging coupling eq.)
- At left C_{boot} is implemented with a transistor having source tied to drain.

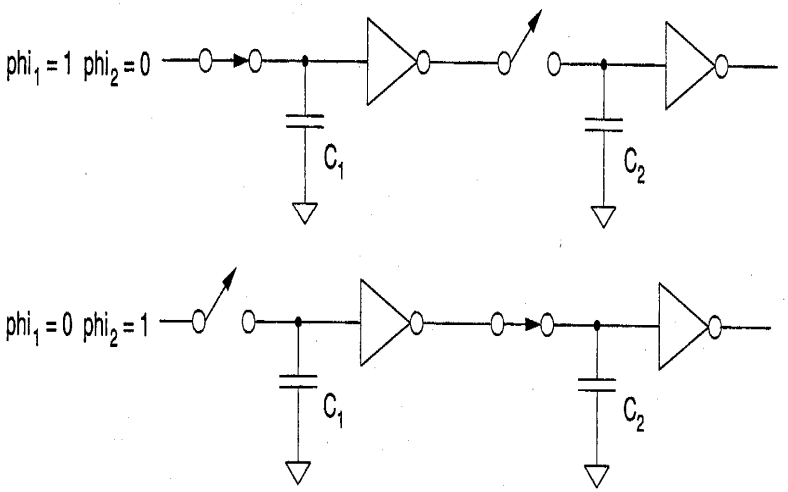
Dynamic Latches with a Single Clock

- Dynamic latches eliminate dc feedback leg by storing data on gate capacitance of inverter (or logic gate) and switching charge in or out with a transmission gate
  - Minimum frequency of operation is typically of the order of 50-100 KHz so as not to lose data due to junction or gate leakage from the node
  - Can be clocked at high frequency since very little delay in latch elements

Dynamic Registers with Two Phase Clocks



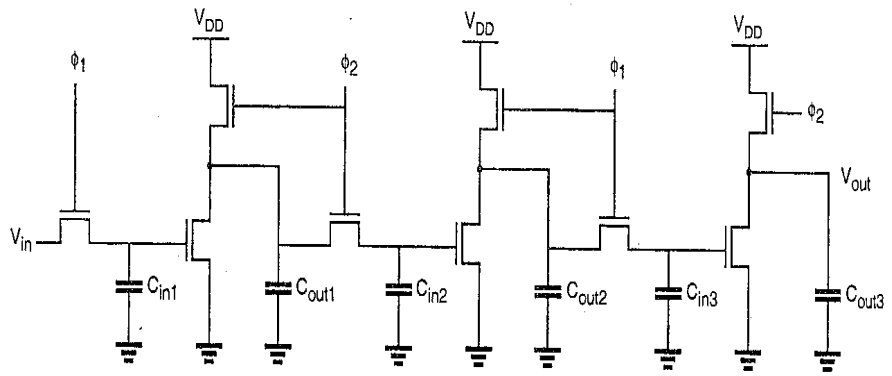
(a)

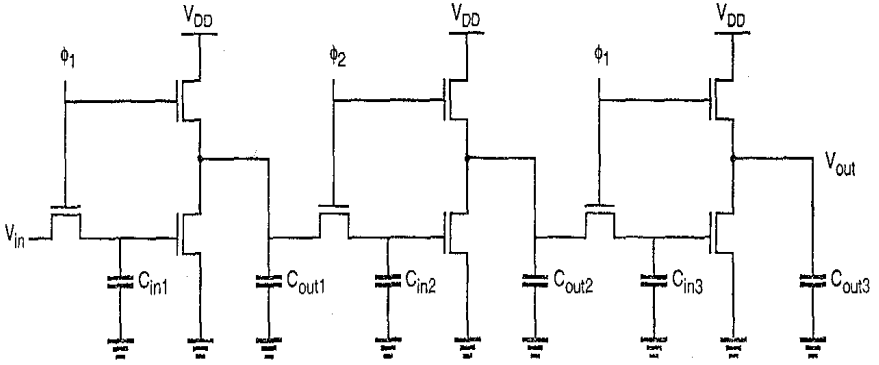


(b)

- 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!

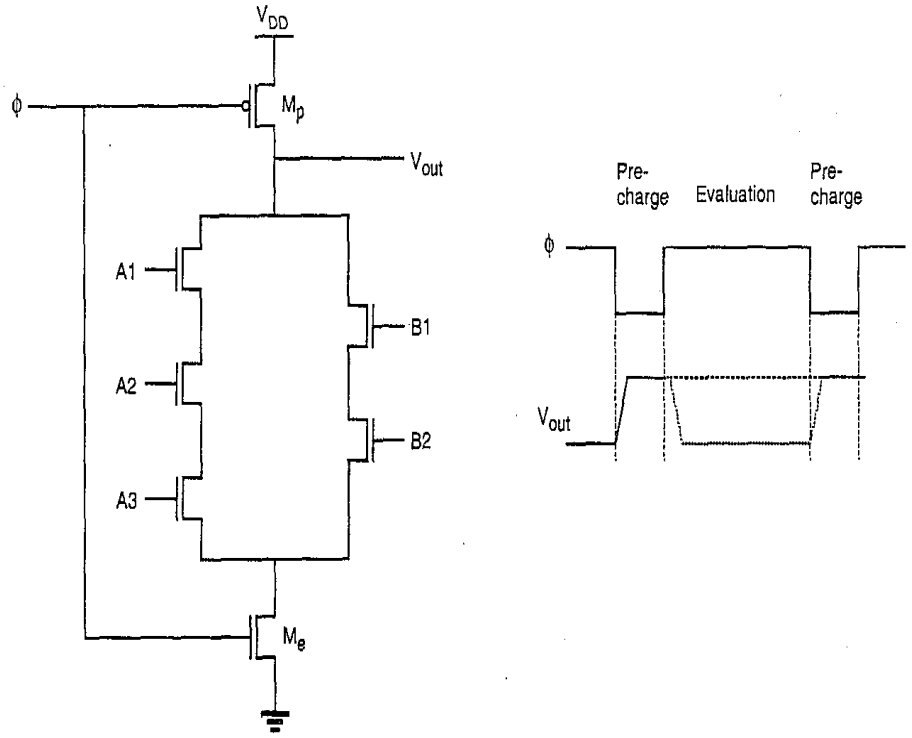
Dynamic Shift Registers with Enhancement Load





- At left (top) is a dynamic shift register implemented with a technique named "**ratioed dynamic logic**".
  - ϕ1 and ϕ2 are non-overlapping clocks
  - When ϕ1 is high, Cin1 charges to Vdd – Vt if Vin is high or to GND if Vin is low
  - When ϕ1 drops and ϕ2 comes up, the input data is trapped on Cin1 and yields a logic output on Cout1 which is transferred to Cin2
  - When ϕ2 drops and ϕ1 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 ϕ2 is high on 1st inv)
- The bottom left dynamic shift register is a "**ratioless dynamic logic**" circuit
  - When ϕ2 is high transferring data to stage 2, ϕ1 has already turned off the stage 1 load transistor, allowing a V_{OL} =

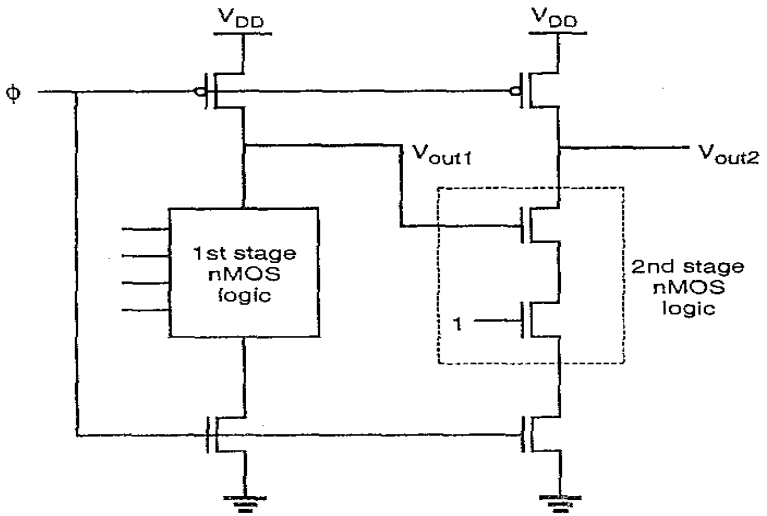
Dynamic CMOS Logic Gate

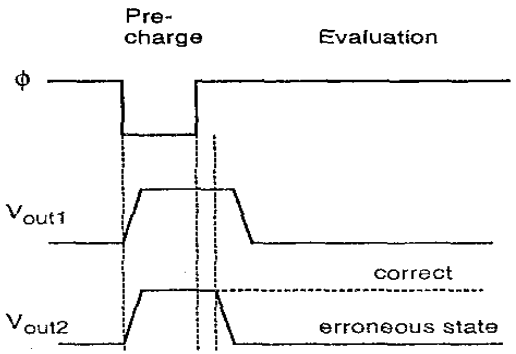


- In dynamic CMOS logic a single clock ϕ can be used to accomplish both the pre-charge and evaluation operations
  - When ϕ is low, PMOS pre-charge transistor Mp charges Vout to Vdd, since it remains in its linear region during final pre-charge
    - During this time the logic inputs A1 … B2 are active; 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

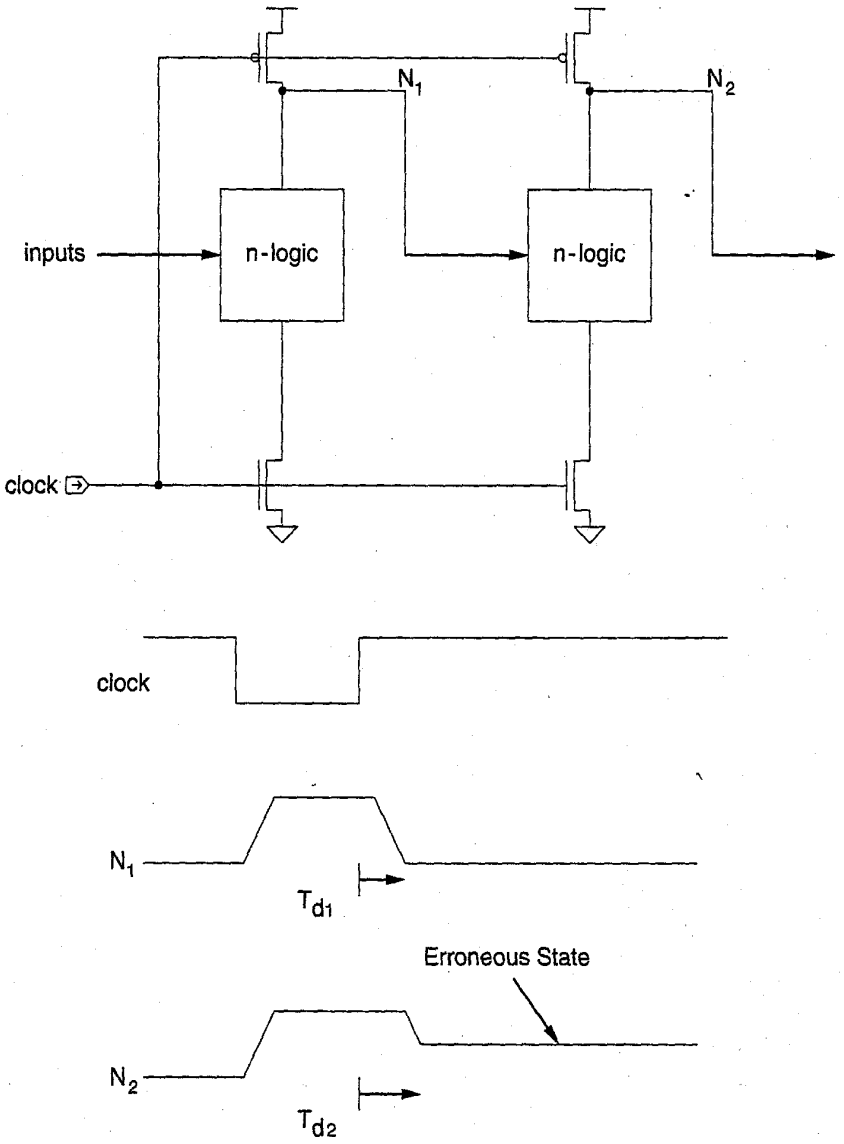
Cascading Problem in Dynamic CMOS Logic

- 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



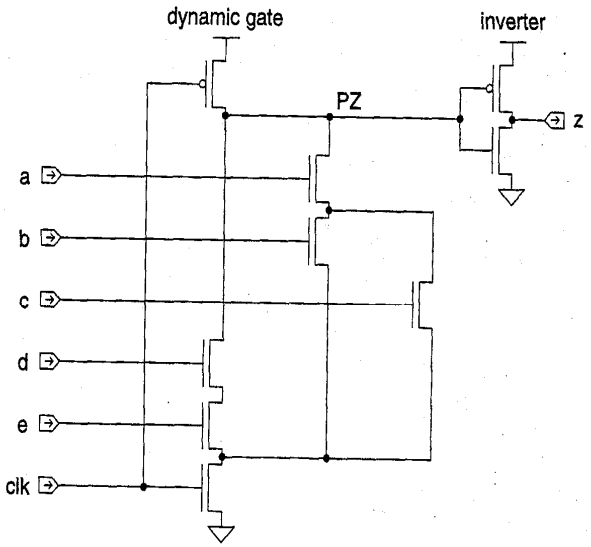


Cascaded Dynamic CMOS Logic Gates: Evaluate Problem



- 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

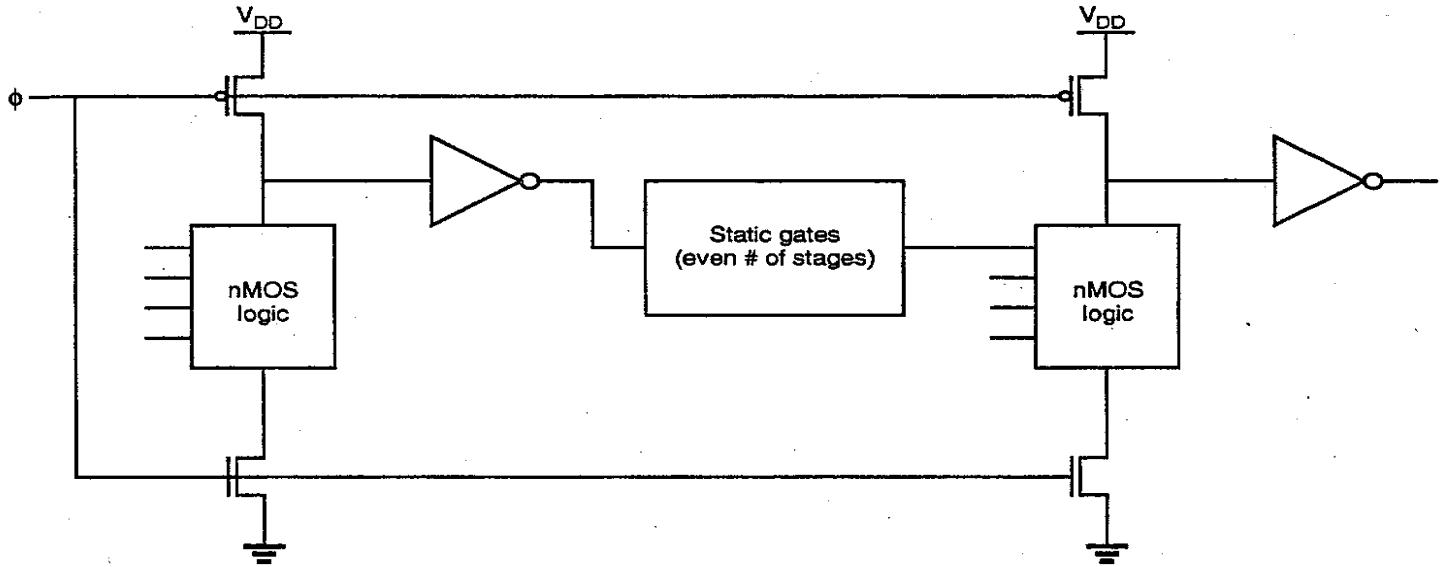
CMOS Domino Logic



- 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

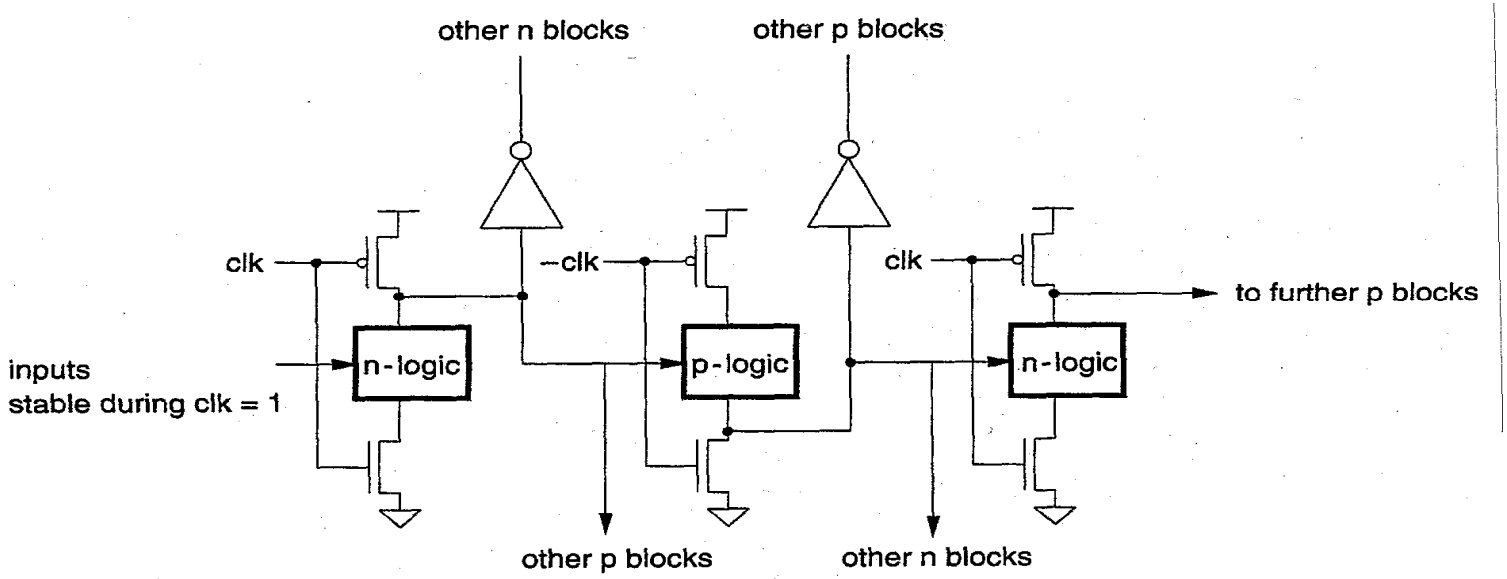
Mixing Domino CMOS Logic with Static CMOS Logic

- 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
- 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**



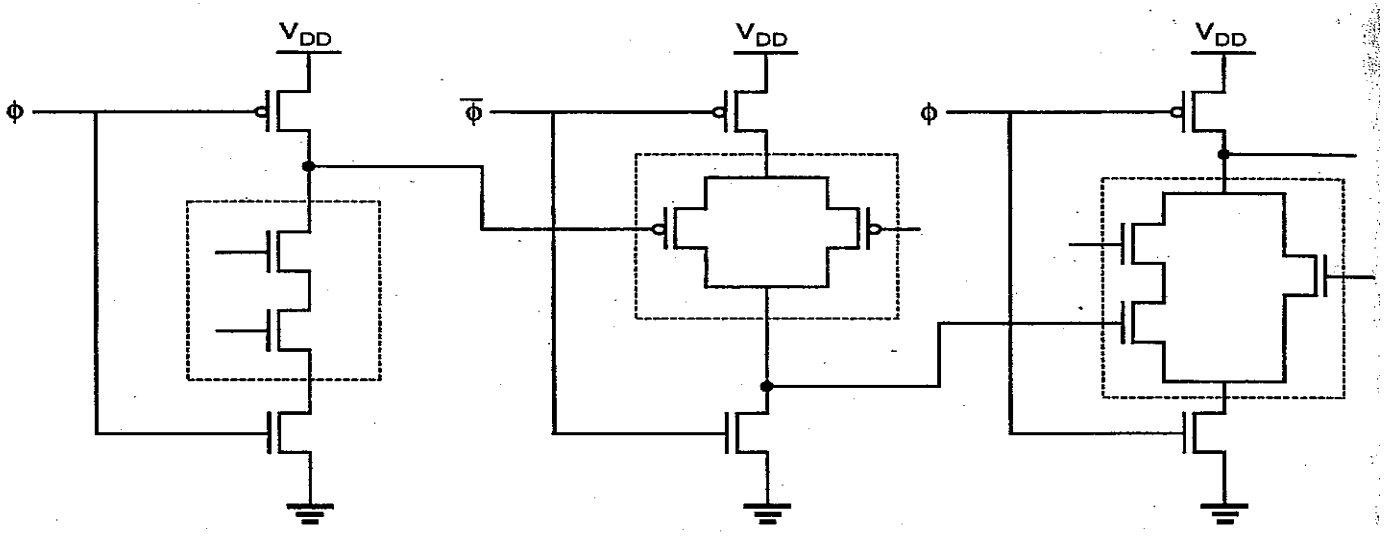
NP Domino Logic (NORA Logic)

- 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.

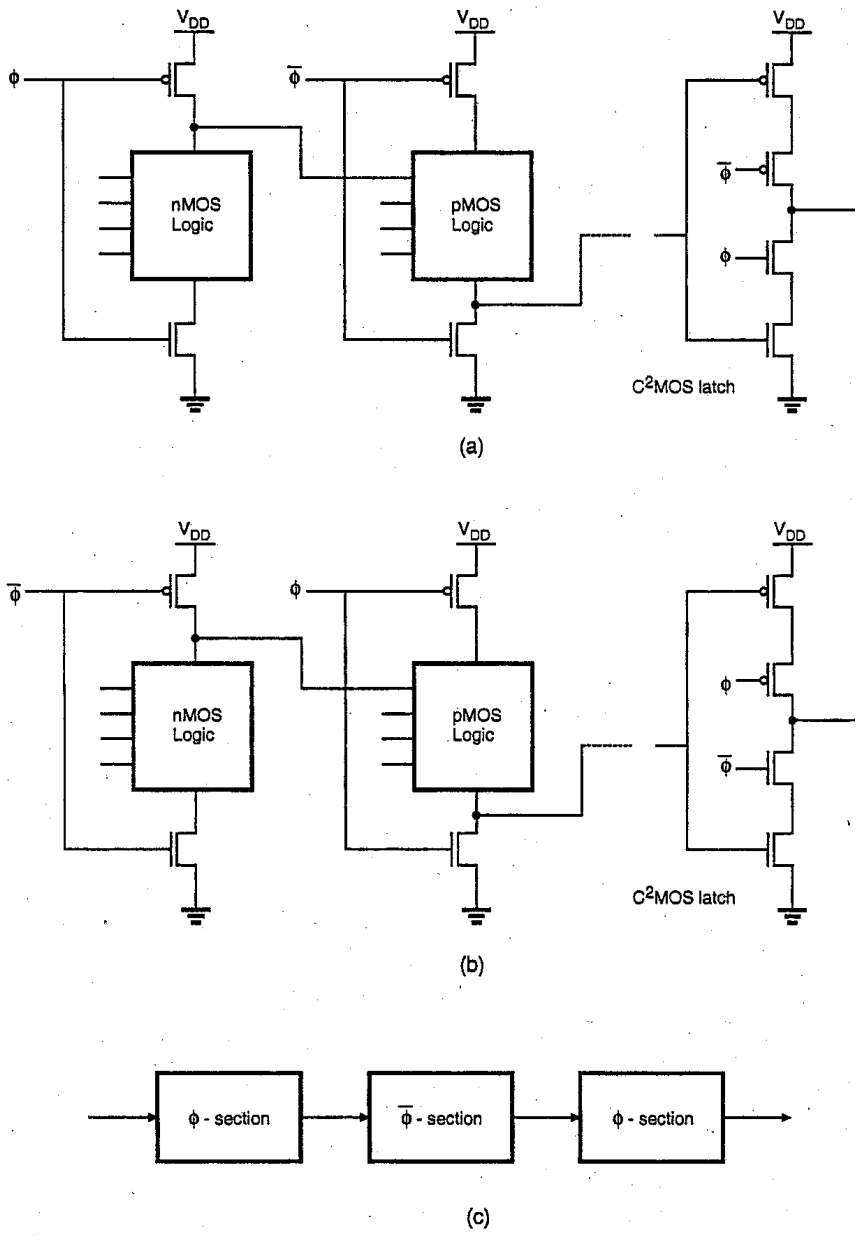


NORA CMOS Logic Circuit Example

- 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)'

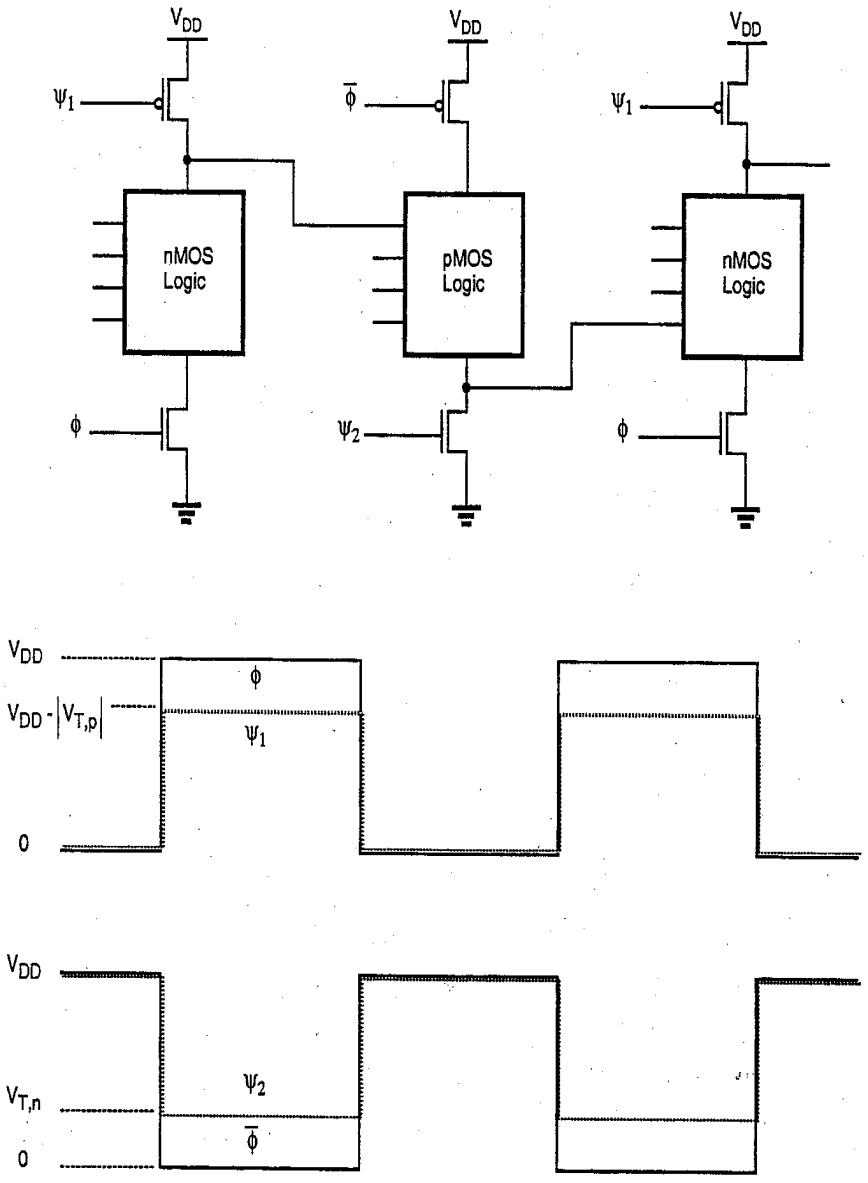


Pipelined NORA CMOS Circuit Operation



- 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

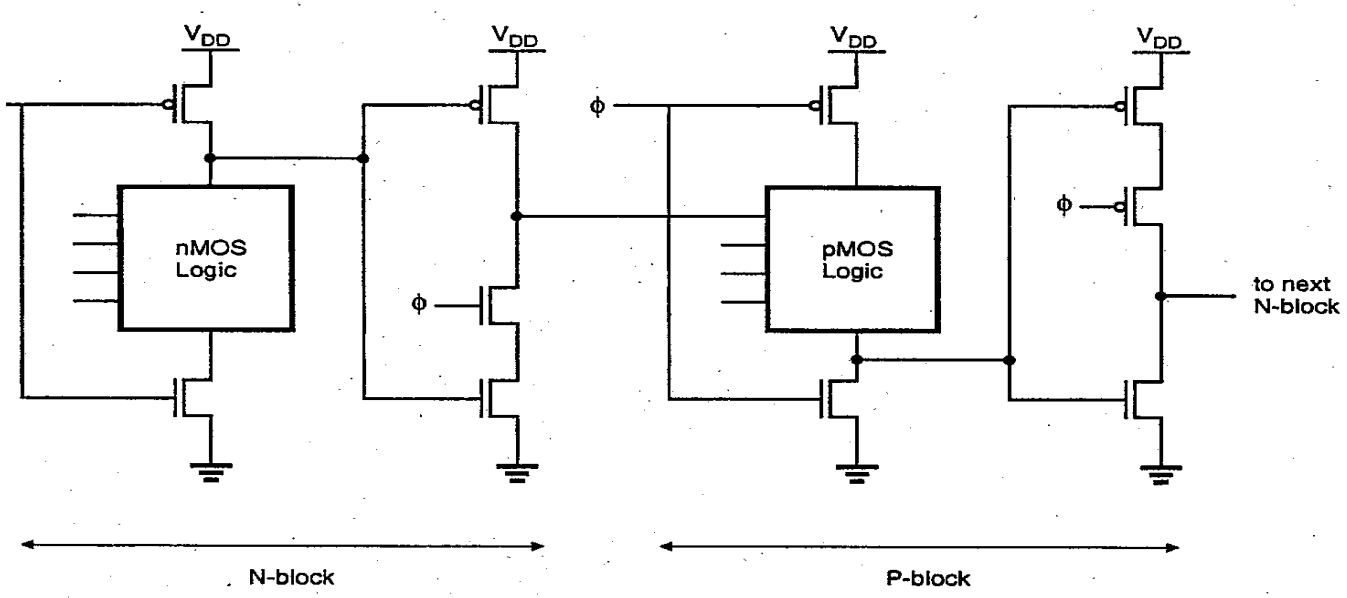
Zipper CMOS Dynamic Logic



- **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



References

- Lecture material R. W. Knepper SC571
- S-M. Kang and Y. Leblebici ,*CMOS Digital Integrated Circuits: Analysis and Design*,, 3rd edition