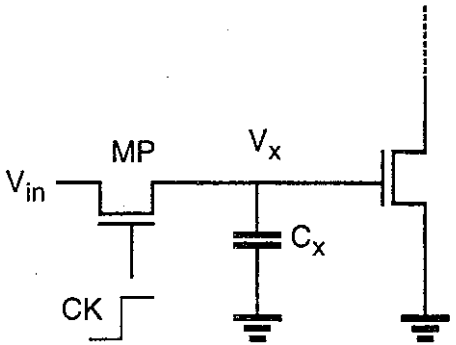
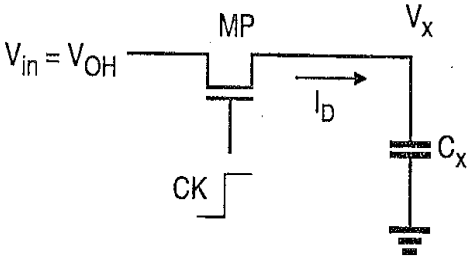
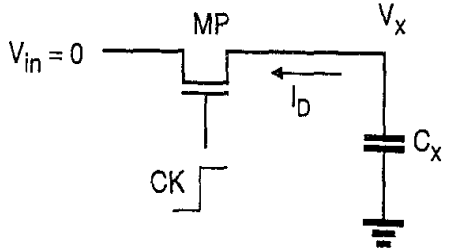
Dynamic Logic Circuits

- 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

NMOS Dynamic Logic Basic Circuit

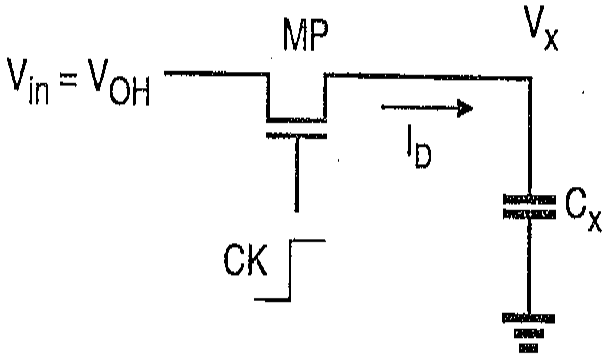






- The basic **dynamic logic gate concept** is shown at left (top)
  - the pass transistor MP is an NMOS device, but could also be implemented with a transmission gate TG
  - C_x represents the equivalent capacitance of the input gate of the second NMOS device (part of an inverter or logic gate) as well 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

Dynamic NMOS Logic: 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:

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

  - Solution:

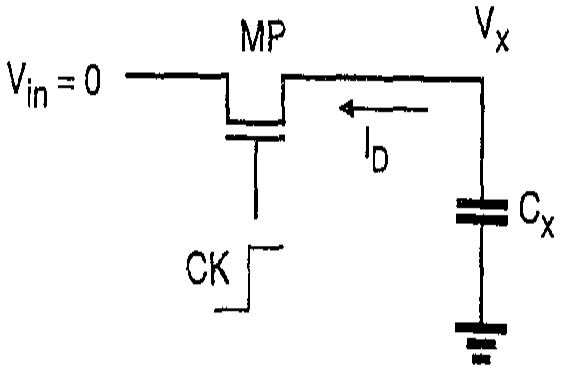
$$\mathbf{t = (2Cx/\,\beta_n)[\{1/(Vdd - Vx - Vtn) - 1/(Vdd - Vtn)\}]}$$

or, solving for $V_x(t)$

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

- As t → infinity, $V_x(t)$ → Vdd – Vtn
- Solve for time needed to reach 90% (Vdd – Vtn):
  - Set $V_x(t) = 0.9\,(Vdd - Vtn)$ → $\mathbf{t_{90\%} = 18\,C_x/\,\beta_n(Vdd - Vtn)}$
    - i.e. 18 time constants

Dynamic NMOS Logic: 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:

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

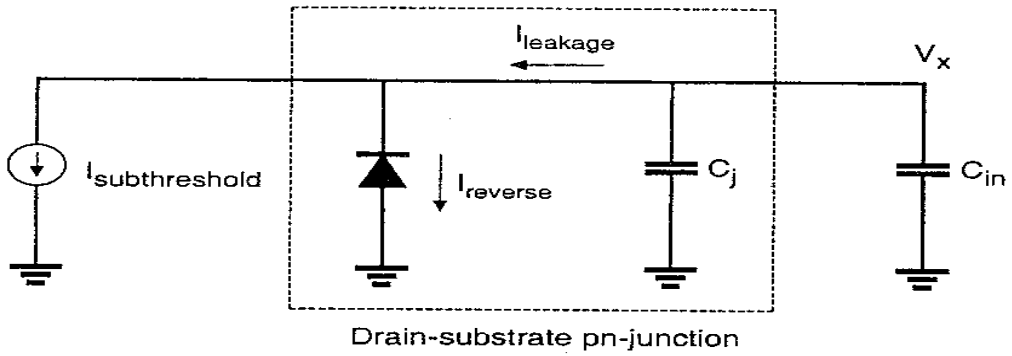
  - Solution:

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

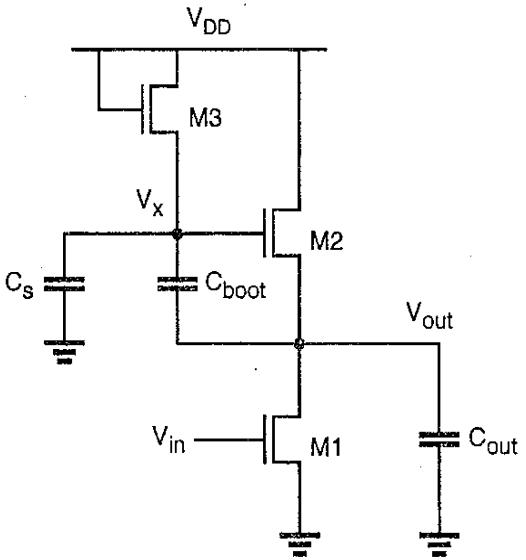
- Solve for time needed for Vx to fall to 10% (Vdd – Vtn):
  - Set $V_x(t) = 0.1\,(Vdd - Vtn)$ ➔ $\mathbf{t_{10\%} = 2.9\,C_x/\,\beta_n(Vdd - Vtn)}$
    - i.e. 2.9 time constants
- Therefore, the time to discharge Cx with an NMOS MP pass transistor is much shorter than the time to charge Cx due to the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

Leakage and Subthreshold Current in Dynamic Pass Gate

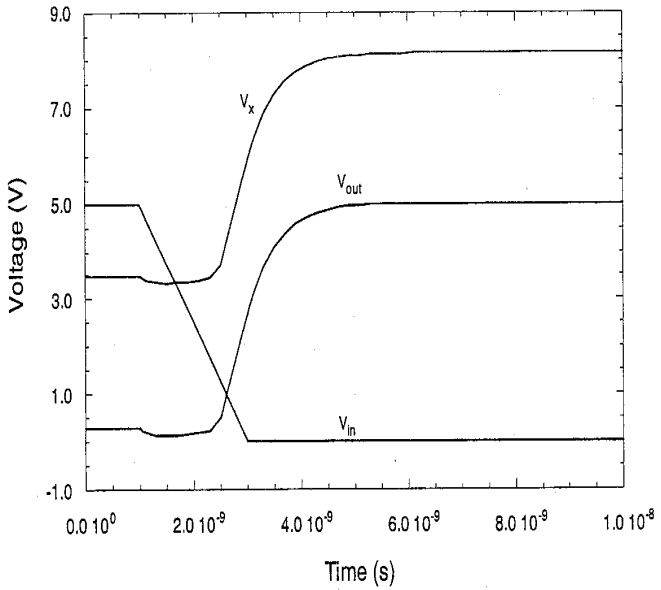
- Charge can leak off the storage capacitor Cx mainly from two sources:
  - PN junction leakage of the NMOS drain (source) junction
  - Subthreshold current (I_{OFF}) through MP when its gate is down at zero volts
- One can solve for the maximum amount of time Δt that charge can be retained on Cx using 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 $\Delta V \sim Vdd - Vtn - \frac{1}{2} Vdd = \frac{1}{2} Vdd - Vtn$
- The minimum frequency of operation can be found from $f \sim 1/(2\ \Delta t)$

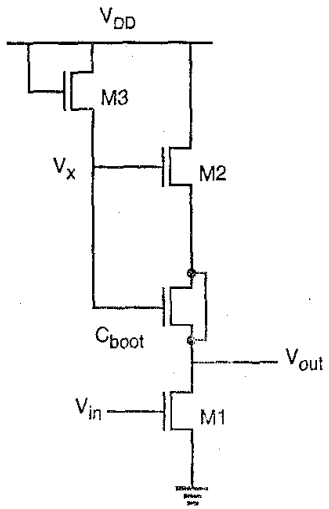


Dynamic Bootstrapping Technique



- **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.

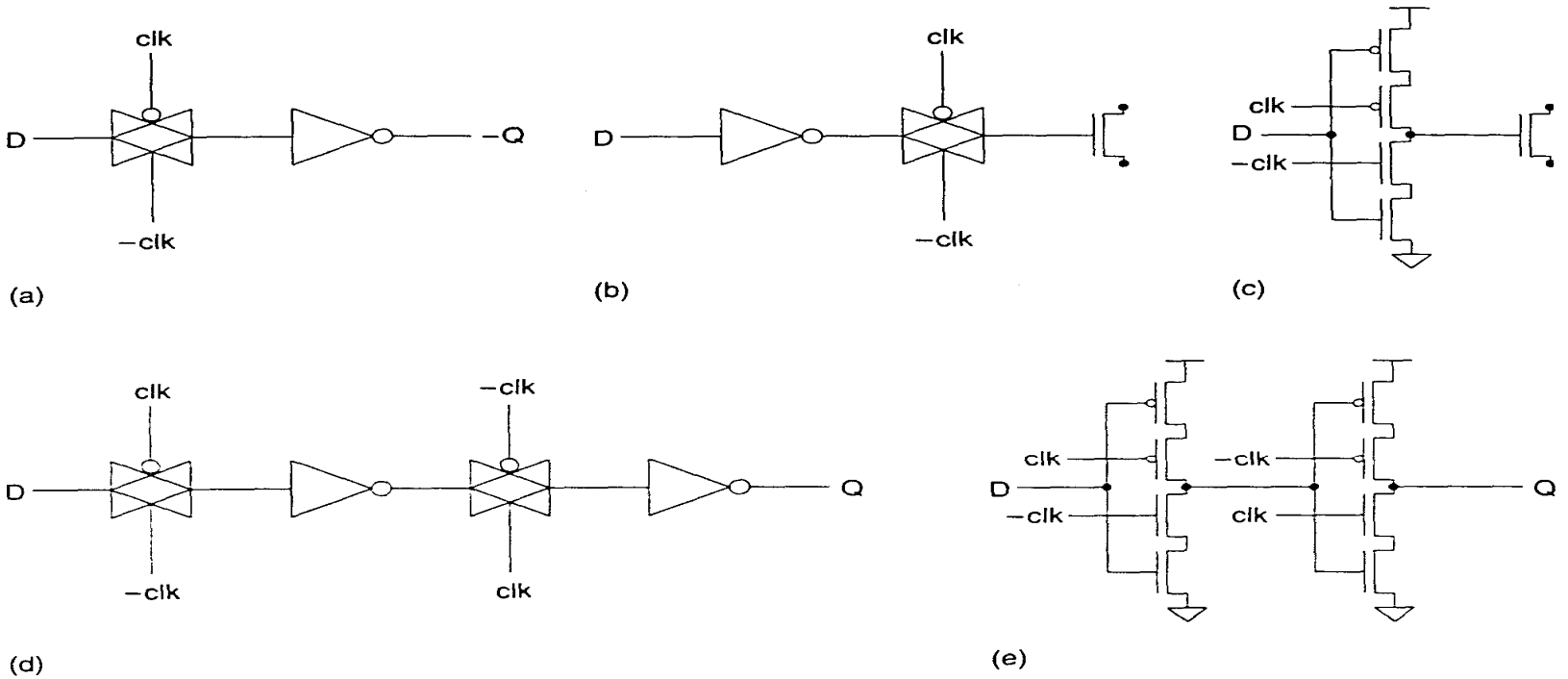




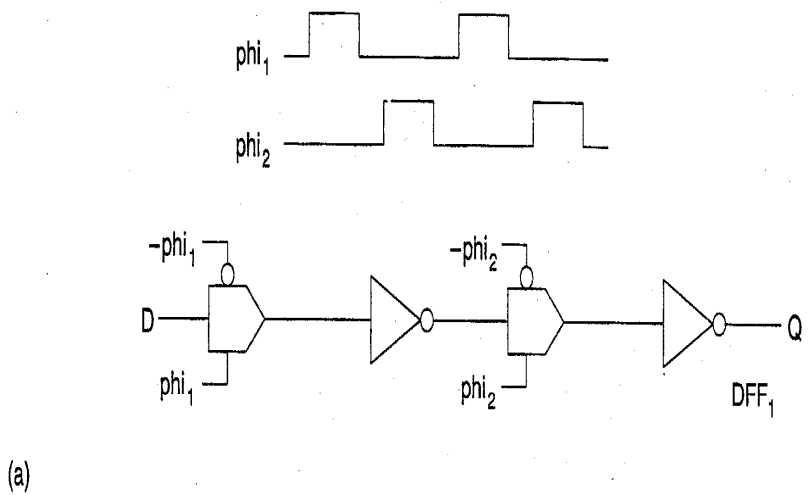
- It is desired to obtain Vx > Vdd + Vt in order to keep M2 linear, to allow Vout to be charged fully to Vdd.
- Parasitic capacitor C_s bleeds some of the charge off C_{boot}, limiting the max voltage on Vx (charging coupling eq.)
- At left C_{boot} is implemented with a transistor having source tied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

Dynamic Latches with a Single Clock

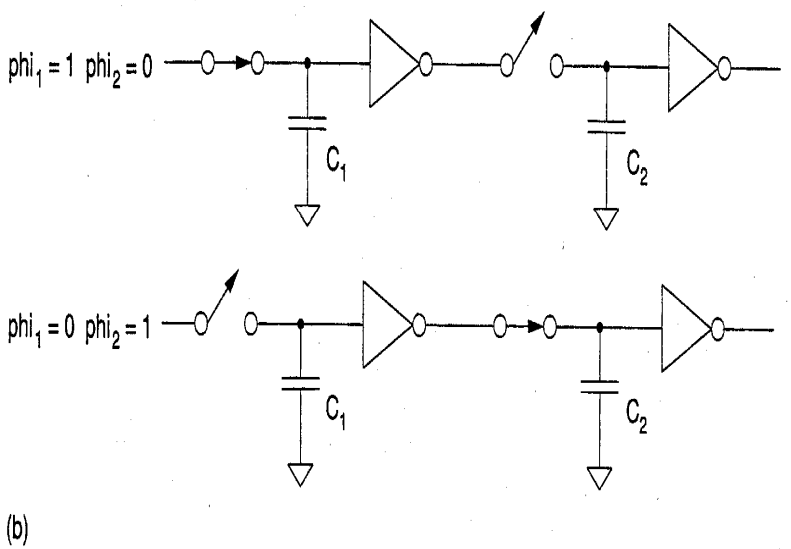
- Dynamic latches eliminate dc feedback leg by storing data on gate capacitance of inverter (or logic gate) and switching charge in or out with a transmission gate
  - Minimum frequency of operation is typically of the order of 50-100 KHz so as not to lose data due to junction or gate leakage from the node
  - Can be clocked at high frequency since very little delay in latch elements
- Examples:
  - (a) or (b) show simple transmission gate latch concept
  - (c) tri-state inverter dynamic latch holds data on gate when clk is high
  - (d) and (e) dynamic D register



Dynamic Registers with Two Phase Clocks



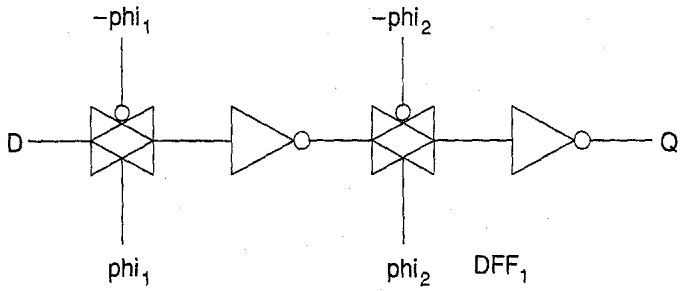
(a)

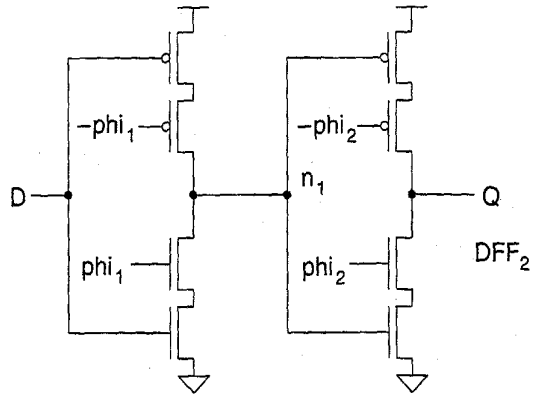


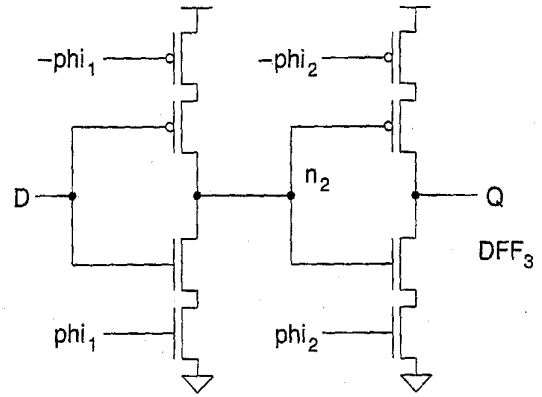
(b)

- 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!

Two Phase Dynamic Registers (Compact Form)







Both of these dynamic registers have to drive a local storage gate.

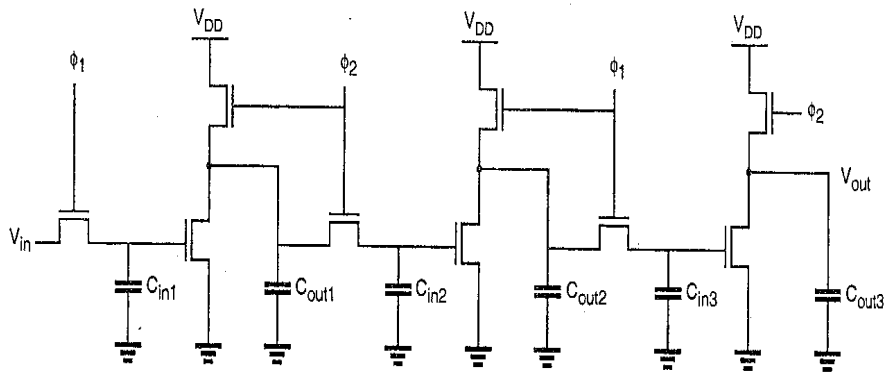
Compact implementation of of two phase dynamic registers shown at left using a tri-state buffer form.

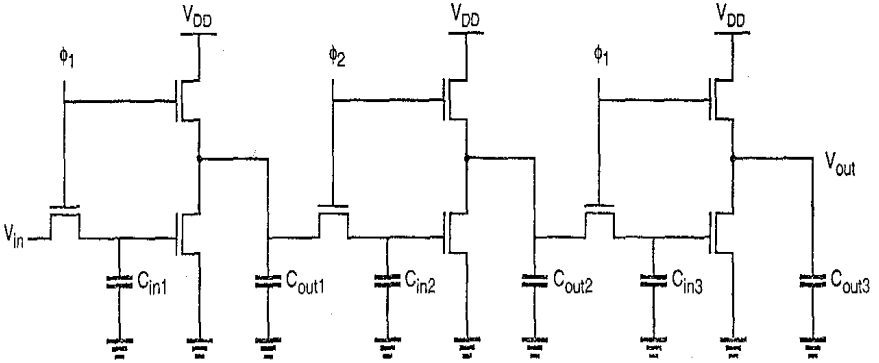
- Transmission gate and inverter integrated into one circuit
- Two versions:
  - Pass devices closest to output
  - Inverter devices closest to output

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





- At left (top) is a dynamic shift register implemented with a technique named "**ratioed 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.