MOS Logic and Gate Circuits

[Diagram of a MOS logic circuit with labeled inputs A, B, AB, and output Y, indicating a wired OR function.]
Contents

- Introduction
  - NMOS Logic
    - Resistive Load
    - Saturated Enhancement Load
    - Linear Enhancement Load
    - Depletion Load
    - Some Gates
    - Transient in NMOS Circuit
    - Pseudo-NMOS
  - CMOS Logic
    - Static CMOS Logic Gates
      - NOT
      - NAND
      - NOR
      - Realization of More Complicated Gate Circuits
    - Transmission Gates Family
      - NMOS Only Switch
      - CMOS
    - Differential Cascode Voltage Switch Logic
    - Rules of Thumb
  - Summary
Introduction

- The MOS inverter is the basic circuit exhibits all of the essential features of MOS Logic. Extension of MOS inverter concepts to NOR and NAND Gate is very simple. In this lecture we will analysis for VTC, NM, PD,... Both NMOS and CMOS circuits are considered. Digital MOS circuits can be classified into two categories:
  - **Static Circuits:** require no clock or other periodic signal for operation. Clocks are required for static circuit in sequential logic
  - **Dynamic Circuits:** require periodic clock signals, synchronized with data signals, for proper operation even in combinational logic
NMOS Logic

- Resistive Load

\[ \frac{W}{L} \uparrow \Rightarrow VoL \downarrow \Rightarrow C \uparrow \Rightarrow Speed \downarrow \]

\[ P \downarrow \Rightarrow RL \uparrow \Rightarrow Area \uparrow \]
NMOS Logic

- Resistive Load Properties
  - \( N \) transistors + Load
  - \( V_{OH} = V_{dd} \)
  - \( V_{OL} = V_{dd} \left( \frac{r_N}{r_N + RL} \right) \)
  - Assymetrical response
  - Static power consumption
  - \( t_{PL} = 0.69RLCL \)
### NMOS Logic

- **Saturated Enhancement Load**

\[
V_{ds2} = V_{gs2} \implies \\
V_{ds2} > \left| V_{gs2} - V_{T2} \right| \implies \\
\text{M2 is in saturation}
\]
Saturated Enhancement Load

\( V_{IL} \)

\[ V_{IL} \geq V_{T1} \Rightarrow M1, M2 \text{ are in saturation} \]
\[ \Rightarrow K_1 (V_{gs1} - V_{T1})^2 = K_2 (V_{gs2} - V_{T2})^2, V_{T2} \approx \text{Cte} \]
\[ V_{gs1} = V_i, V_{gs2} = V_{dd} - V_o \Rightarrow \frac{\delta V_o}{\delta V_i} = -\sqrt{\beta_R}, \beta_R > 1 \]
\[ \Rightarrow \frac{\delta V_o}{\delta V_i} \neq -1 \Rightarrow V_{IL} \approx V_{T1} \]
NMOS Logic

- Saturated Enhancement Load
  - $V_{OL}$
    - $V_{OL}$ is difficult to obtain because it is the output voltage when input equal to $V_{OH}$, the resulting expression is a fourth order polynomial!

\[
\frac{\delta V_o}{\delta V_i} = -1
\]
Saturated Enhancement Load

- \( V_{IH} \)
  - M1 is in triode and M2 in saturation

\[
\frac{\delta V_o}{\delta V_i} = - \frac{\delta i_1}{\delta V_i} (r_{ds1}||r_{ds2})
\]
\[
r_{ds1}||r_{ds2} \approx r_{ds1} = \frac{\delta V_o}{\delta i_1}
\]
\[
i_1 = K_1 \left[ (V_{gs1} - V_{T1}) V_o - V_o^2/2 \right]
\]
\[
i_2 = \frac{1}{2} K_2 (V_{gs2} - V_{T2})^2
\]
\[
\Rightarrow \frac{\delta i_1}{\delta V_i} = K_1 V_o
\]

\[
\Rightarrow \frac{\delta i_1}{\delta V_o} = K_1 \left( V_{gs1} - V_{T1} - V_o \right)
\]
\[
\Rightarrow \frac{\delta V_o}{\delta V_i} = - \frac{K_1 V_o}{K_1 (V_{gs1} - V_{T1} - V_o)} = -1
\]
\[
\Rightarrow V_o = \frac{V_i - V_{T1}}{2}
\]
\[
V_i = V_{IH} \Rightarrow \frac{V_{IH} - V_{T1}}{2} = V_o
\]
**NMOS Logic**

- **Saturated Enhancement Load**
  - $V_{IH}$
    - M1 is in triode and M2 in saturation

\[
i_1 = i_2 \Rightarrow \\
K_1 \left[ (V_{IH} - V_{T1}) V_0 - V_0^2 / 2 \right] = \frac{1}{2} K_2 (V_{dd} - V_0 - V_{T2})^2 \\
\Rightarrow V_{IH} = \frac{2(V_{dd} - V_{T2})}{\sqrt{3\beta_R} + 1} + V_{T1}
\]
NMOS Logic

- **Saturated Enhancement Load**
  - **NM**

  \[
  NML = \overline{V_{IL}} - \overline{V_{OL}} \approx \text{Some tenth of volt}
  \]

  \[
  NMH = \overline{V_{OH}} - \overline{V_{IH}} = V_{dd} - V_{T2} - V_{IH}
  \]

- **Power**

  \[
  P_{\text{disH}} \approx 0, \quad P_{\text{disL}} = I_d V_{dd}
  \]

  \[
  \Rightarrow P_{\text{dis}} = \frac{1}{2} I_d V_{dd}
  \]
**NMOS Logic**

- **Linear Enhancement Load**
  - $V_{GG} \geq V_{dd} + V_{T2}$
  - $\implies M2$ is in triode

[Diagram of NMOS Logic]
**NMOS Logic**

- **Linear Enhancement Load**
  - **VTC**
    - By this circuit the $V_{OH}$ can be increased (or $V_{dd}$ can be decreased because $V_{o_{max}} = V_{dd}$)
NMOS Logic

- **Linear Enhancement Load**
  - \( V_{IL} \)
    - M1 is in saturation and M2 in triode

\[
\frac{\delta V_o}{\delta V_i} = -\frac{\delta i_1}{\delta V_i} \cdot rds_2 = -\frac{\delta i_1}{\delta V_i} \cdot \frac{\delta V_{ds2}}{\delta i_1}
\]

\[
i_2 = K_2 \left[ (V_{gs2} - V_{T2}) V_{ds2} - V_{ds2}^2/2 \right]
\]

\[
\frac{\delta i_2}{\delta V_{ds2}} = K_2 \left( V_{GG} - V_o - V_{T2} - V_{ds2} \right)
\]

\[
i_i = \frac{1}{2} K_1 (V_i - V_{T1})^2
\]

\[
\frac{\delta i_1}{\delta V_i} = K_1 \left( V_i - V_{T1} \right)
\]

\[
\frac{\delta V_o}{\delta V_i} = -1 = -\frac{-K_1 (V_i - V_{T1})}{K_2 (V_{GG} - V_o - V_{T2} - V_{ds2})}
\]

\[\Rightarrow V_{il} = f(V_o)\]

- **\( V_{IH} \)**
  - M1 and M2 are in triode

\[rds_2 \rightarrow rds_1 || rds_2\]
NMOS Logic

- Linear Enhancement Load
  - Disadvantages:
    - More chip area is required (since an extra voltage source VGG)
    - Additional interconnection on the chip is needed
    - The required value of $\beta_R$ is even larger than a saturated load
Depletion Load

Ion implantation processing step is needed to create depletion device, but overcome the disadvantages of the previous circuit.
NMOS Logic

- Depletion Load
  - VTC

\[
Vi = \text{Low} \Rightarrow i_1 = 0, \text{M1 in cutoff} \Rightarrow i_1 = i_2 = 0
\]

if M2 is in saturation then \( i_2 = \frac{K_2}{2} (V_{gs2} - V_{T2})^2, V_{gs2} = 0 \Rightarrow i_2 > 0 \)

- Therefore M2 is in triode
NMOS Logic

- Depletion Load
  - $V_{OL}$, $V_{IL}$, $V_{IH}$

\[
K_1 \left[ (V_{OH} - V_{T1}) V_{OL} - \frac{V_{OL}^2}{2} \right] = \frac{K_2}{2} (0 - V_{T2})^2 \Rightarrow V_{OL} = ?
\]

\[
\frac{\delta V_o}{\delta V_i} = -1 \Rightarrow \frac{K_1 (V_{gsl} - V_{T1})}{K_2 (V_{gsl2} - V_{T2} - V_{ds2})} = -1, \ i_1 = i_2 \Rightarrow V_{IL} = ?
\]

\[
i_1 = K_1 \left[ (V_{IH} - V_{T1}) V_{ds1} - \frac{V_{ds1}^2}{2} \right] = i_2 = \frac{K_2}{2} (0 - V_{T2})^2, \ \frac{\delta V_o}{\delta V_i} = -1 \Rightarrow V_{IH} = ?
\]
Some Gates

In all previous structures which different only in load, the following Gates can be implemented. Note that the NMOS Gates are not available as separately packaged individual circuits, but they are used extensively in LSI systems

- NOR Gates
- NAND Gates
Some Gates
- NOR Gate
NMOS Logic

- Some Gates
  - NAND Gate

![NAND Gate Circuit Diagram](image)
Some Gates

- In NOR Gate two transistors are paralleled but in NAND Gate two transistors are in series. Because of the need for increased area when adding NAND inputs, NAND logic with more than 2 inputs is not economically be attractive in NMOS. NOR logic is preferable.
- In NAND the M2 has body effect.
- In NOR we need the less interconnection (this can be shown from layout).
NMOS Logic

- Transient in NMOS Circuits
  - Saturated Enhancement Load

\[
C_{\text{tot}} = C_L + C_{\text{eq}_{d-\text{sub}1}} + C_{\text{eq}_{s-\text{sub}2}} + C_{\text{gs}2} + 2 \times C_{\text{gd}1}
\]
Super Buffer (1)

- If Fan-out is very large then $C_{tot}$ will be large. For reduction it and decrease the switching time the Super Buffer circuit is used.

- In this circuit if Vi is Low state then V1 will be high very more rapid than Vo. Thus the Gate of M2 is in high state very rapidly. Therefore M2 will be in saturation which result the reduction of switching time ($t_{on}$)
NMOS Logic

- Super Buffer (2)
  - It is non-Inverting
  - Describe the operation of this circuit!

![Super Buffer (2) Circuit]
What makes a circuit fast?
- \( I = C \frac{dV}{dt} \rightarrow t_{pd} \propto \frac{C}{I} DV \)
- low capacitance
- high current
- small swing

Logical effort is proportional to \( C/I \)

PMOS are the enemy!
- High capacitance for a given current

Can we take the PMOS capacitance off the input?

Various circuit families try to do this…
Pseudo-NMOS

In the old days, NMOS processes had no PMOS
  • Instead, use pull-up transistor that is always ON

In CMOS, use a PMOS that is always ON
  • Make PMOS about \( \frac{1}{4} \) effective strength of pulldown network
NMOS Logic

- Pseudo-NMOS
  - Uses a p-type as a resistive pullup, n-type network for pulldowns
Pseudo-NMOS Characteristics

- Compared to CMOS, this family has higher packing density, since for n inputs only n+1 transistors are required.
- The main disadvantages with Pseudo-NMOS Gates is the large static power dissipation that occurs whenever a pull-down path is activated.
- Has much smaller pullup network than static gate.
- Pulldown time is longer because pullup is fighting.

NMOS Logic
NMOS Logic

- Pseudo-NMOS Output Voltages
  - Logic 1 output is always at $V_{dd}$
  - Logic 0 output is above $V_{ss}$
  - $V_{OL} = 0.25 \,(V_{dd} - V_{ss})$ is one plausible choice
Pseudo-NMOS Design Topics

- For logic 0 output, pullup and pulldown form a voltage divider
- Must choose n, p transistor sizes to create effective resistances of the required ratio
- Effective resistance of pulldown network must be computed in worst case—series n-types means larger transistors
Pseudo-NMOS Transistor Ratio Calculation

- MOSFET sizing is important
- Need to have reasonable W/L ratios for circuit to work correctly
- $V_{OL} > V_{SS}$ but must be low enough to turn off/on next MOSFET in the chain
- Static current drain when “on”
- $V_{out}$ is a function of the number of parallel and series $N$ channels in the pull down network
NMOS Logic

- Pseudo-NMOS Transistor Ratio Calculation
  - For single supply

\[
V_{OH} = V_{dd}, \quad \text{For the worst case one NMOS to be on}
\]

\[
K_n \left[ \left( V_{dd} - V_{Tn} \right) V_{OL} - \frac{V_{OL}^2}{2} \right] = \frac{K_p}{2} \left( V_{dd} - |V_{TP}| \right)^2
\]

\[
V_{OL} = K_n (V_{dd} - V_{T}) \left[ 1 - \sqrt{1 - \frac{K_p}{K_n^2}} \right]
\]

Assuming that \( V_T = V_{Tn} = |V_{TP}| \)

\[V_{OL} \rightarrow 0 \Rightarrow K_p << K_n\]
NMOS Logic

- Pseudo-NMOS VTC ($W/L_n = 1$)
NMOS Logic

- Pseudo-NMOS Gates
  - Design for unit current on output
  - PMOS fights NMOS
Pseudo-NMOS Power

- Pseudo-NMOS draws power whenever \( Y = 0 \)
  - Called static power \( P = I \cdot V_{DD} \)
  - A few mA / gate \( \times \) 1M gates would be a problem
  - This is why NMOS went extinct!

- Use Pseudo-NMOS sparingly for wide NORs
- Turn off PMOS when not in use
NMOS Logic

- Pseudo-NMOS (NAND) Layout Example
CMOS Logic

- Static CMOS Logic Family
  - All of the circuits described in the previous sections have a large static power dissipation. This disadvantage can be overcome by using Static CMOS Logic Family
CMOS Logic

- Static CMOS Logic Family
  - NOT

![CMOS Logic Diagram](image)
CMOS Logic

- Static CMOS Logic Family
  - Two inputs NAND
CMOS Logic

- Static CMOS Logic Family
  - Two inputs NOR

![CMOS Logic Diagram](image)
CMOS Logic

- Static CMOS Logic Family
  - NAND is more suitable for CMOS because by suppose the equal W/L for NMOS and PMOS transistors, the PMOS transistor has more resistance respect to NMOS, therefore it is better to design circuit by paralleling the PMOS and cascading the NMOS.
  - The better Technology for digital circuit is N-Well, because in this Technology the NMOS transistors are made in the Sub. Which result better characteristic for transistor.
Realization of More Complicated Gate Circuits
   - a) $Y = A(B+C)$
   - It can be implemented in three levels:
     - Gate Level
     - Transistor Level
     - Layout Level
Realization of More Complicated Gate Circuits

a) $Y = A(B+C)$

- Gate Level
  - It consists of 10 transistors, 4 transistors for NOR, 2 for NOT and 4 for NAND
Realization of More Complicated Gate Circuits
- a) \( Y = A(B+C) \)
  - Transistor Level
    - It needs only 6 transistors
Realization of More Complicated Gate Circuits

a) \( Y = A(B+C) \)

- **Layout Level**
  - By proper construction of layout, the parasitic capacitors are also reduced and area can be saved.
Realization of More Complicated Gate Circuits
  
  b) XNOR (Y = AB + \overline{AB})
  
  • Gate Level
    – 16 transistors are needed
Realization of More Complicated Gate Circuits

b) XNOR (Y = AB + \overline{AB})

- Transistor Level (1)
  - How many transistors are needed?
Realization of More Complicated Gate Circuits

b) XNOR

- Transistor Level (2)
  - Previous circuit can be simplified by eliminating two wiring lines

\[
Y = AB + \overline{AB}
\]
Realization of More Complicated Gate Circuits

b) XNOR

- Transistor Level (3)
  - In this circuit we have static power dissipation in the state of (A=0, B=1) or (A=1, B=0)

\[ Y = AB + \overline{AB} \]
Realization of More Complicated Gate Circuits

b) XNOR

- Transistor Level (4) (For less dissipation)
  - Note to the state of $A = B = V_{dd}$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$V_{dd}$</td>
</tr>
<tr>
<td>0</td>
<td>$V_{dd}$</td>
<td>0</td>
</tr>
<tr>
<td>$V_{dd}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$V_{dd}$</td>
<td>$V_{dd}$</td>
<td>$V_{dd} - V_{th}$</td>
</tr>
</tbody>
</table>
Realization of More Complicated Gate Circuits

- c) XOR
  
  - As Previous, the Source and Drain of M3 (or M4) are replaced by each other in different states, for example in state of A=0, B=0 the b connection of M3 is Source.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Vth</td>
</tr>
<tr>
<td>0</td>
<td>Vdd</td>
<td>Vdd-Vth</td>
</tr>
<tr>
<td>Vdd</td>
<td>0</td>
<td>Vdd</td>
</tr>
<tr>
<td>Vdd</td>
<td>Vdd</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
Y = \overline{AB} + A\overline{B}
\]
Realization of More Complicated Gate Circuits

d) Tri-State Outputs

- A floating state at the output is needed
  - Non-Inverting

\[
\begin{align*}
\text{En} = 1 & \Rightarrow Y = D \\
\text{En} = 0 & \Rightarrow Y = \text{Hi – Z}
\end{align*}
\]
Realization of More Complicated Gate Circuits

d) Tri-State Outputs

- A floating state at the output is needed
  - Inverting

\[
\begin{align*}
\text{En} = 1 & \Rightarrow Y = \text{Hi} - Z \\
\text{En} = 0 & \Rightarrow Y = \overline{D}
\end{align*}
\]
Realization of More Complicated Gate Circuits

e) Schmitt Trigger

- M3 and M6 have minimum sized geometries
- With \( V_{in} = 0 \), the transistors M1 and M2 will be on but conducting negligible Drain current since M4 and M5 are off

\[
\Rightarrow V_y \approx V_x \approx V_{dd}
\]

\[
V_y \approx V_{dd} \Rightarrow M_6 \approx \text{on} \Rightarrow V_z = V_y - V_{TN}
\]
Realization of More Complicated Gate Circuits

e) Schmitt Trigger

- When $V_{\text{in}}$ rises to $V_{\text{TN}}$, M5 turns on, but M4 is off. M5 and M6 form an NMOS amplifier. Thus as $V_{\text{in}}$ rises, $V_{\text{z}}$ is falling and in the certain voltage M4 turns on. With both M4 and M5 conducting, Vy rapidly goes to zero turning off M6. With $V_{y}=0$, M3 turns on, which aids in turning off M2 as $V_{x}$ goes from $V_{dd}$ to $V_{y}-V_{TP}$
- As $V_{\text{in}}$ decrease from $V_{dd}$ to zero the operation is essentially similar. But now M1 turns on, in different voltage and ...
Transmission Gates Family

- Use pass transistors like switches to do logic
- Inputs drive diffusion terminals as well as gates
- $N$ transistors instead of $2N$
- No static power consumption
- Ratioless
- Bidirectional
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch

![Diagram of NMOS Only Switch](image)

![Graph of voltage over time](image)
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch
    - $V_B$ does not pull up to 2.5V, but $2.5-V_{TN}$
    - Threshold voltage loss causes ($M_2$ may be weakly conducting forming a path from Vdd to GND)
    - NMOS has higher threshold than PMOS (body effect)
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch
    - Pass transistor gates should never be cascaded as on the left
    - Logic on the right suffers from static power dissipation and reduced noise margins
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch
    - Solution1: Level Restoring Transistor

![Diagram of NMOS Only Switch with Level Restoring Transistor](image-url)
Transmission Gates Family

NMOS Only Switch
  - Solution1: Level Restoring Transistor

Advantages
  - Full swing on x (due to Level Restorer) so no static power consumption by inverter
  - No static backward current path through Level Restorer and PT since Restorer is only active when A is high
  - Restorer adds capacitance, takes away pull down current at X

For correct operation Mr must be sized correctly (ratioed)
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch
    - Solution 2: Multiple $V_T$ Transistors

![Diagram of NMOS Only Switch with sneak path and off but leaking state]
Transmission Gates Family

- NMOS Only Switch
  - Solution2: Multiple $V_T$ Transistors
- Technology solution: Use (near) zero $V_T$ devices for the NMOS TGs to eliminate most of the threshold drop (body effect still in force preventing full swing to Vdd)
- Impacts static power consumption due to subthreshold currents flowing through the TGs (even if VGS is below $V_T$)

![CMOS Logic diagram](image)
CMOS Logic

- Transmission Gates Family
  - NMOS Only Switch
    - Disadvantage:
      - It can be bad because the signal can be degraded
      - We do not allow a few gates in series for one signal (Pure TG logic is not regenerative, the signal gradually degrades after passing through a number of TGs)
    - Advantage:
      - Allow us to save transistor or less stage of logic

![Diagram of NMOS Only Switch]
CMOS Logic

- CMOS Transmission Gates Family
  - PMOS
    - CMOS
CMOS Logic

- CMOS Transmission Gates Family
  - There are many symbols for transmission gate

- Be careful, because it is bi-directional
CMOS Transmission Gates Family

- This circuit performs a function similar to that of the well-known diode bridge.

\[ C = V_{ss}, \overline{C} = 0 \Rightarrow \text{Both transistors are on} \Rightarrow A = Y \]

- The input voltage \( VA \) (which must be between \( V_{ss} \) and \( V_{dd} \)) is then connected to the output through the parallel on resistance of the channels of the two transistors. As \( VA \) approaches \( V_{dd} \), the N-channel device cuts off but the P-channel device remains non-saturated, as \( VA \) approaches \( V_{ss} \), the P-channel device cuts off but the N-channel device remains non-saturated. Therefore there is always a non-saturated transistor between input and output.
CMOS Logic

- CMOS Transmission Gates Family
  - For more understanding note to this circuit
  - We assume:

  \[ V_y(0^-) = 0, \ C = V_{dd}, \ \bar{C} = 0, \ |V_{TP}| < |V_{dd} - V_{TN}| \]
CMOS Logic

- CMOS Transmission Gates Family

\[ 0 \leq t \leq t_{V_y = V_{TP}} \Rightarrow \begin{cases} 
N = \text{sat} \\
P = \text{sat} 
\end{cases} \Rightarrow I_{CL} = I_N + I_P \]

\[ t_{V_y = V_{TP}} \leq t \leq t_{V_y = V_{dd} - V_{TN}} \Rightarrow \begin{cases} 
N = \text{sat} \\
P = \text{triode} 
\end{cases} \Rightarrow \tau \approx r_p CL \]

\[ t_{V_y = V_{dd} - V_{TN}} \leq t \leq t_{V_y = V_{dd}} \Rightarrow \begin{cases} 
N = \text{off} \\
P = \text{triode} 
\end{cases} \Rightarrow \tau \approx r_p CL \]
CMOS Logic

- CMOS Transmission Gates Family
  - It is normally assumed as a resistor

\[ V_A(t) = V_{dd}u(t) \]
\[ V_y(t) = V_{dd} \left[ 1 - \exp\left(\frac{-t}{\tau_{TG}}\right) \right] u(t) \]
\[ \tau_{TG} = R_{TG} \cdot CL \]
\[ R_{TG} = r_N || r_P \]
CMOS Logic

- CMOS Transmission Gates Family
  - $r_N$ and $r_P$ are approximately as follow (In saturation region)

\[
\begin{align*}
  r_N &\approx \frac{2V_{AN}}{\beta_N (V_{dd} - V_{TN})^2} \\
  r_P &\approx \frac{2V_{AP}}{\beta_P (V_{dd} - |V_{TP}|)^2}
\end{align*}
\]

$V_A$ is Early Voltage
CMOS Logic

- CMOS Transmission Gates Family
  - What to note about TG
    - The inputs must be able to give high current because they are connected directly to Drain and Source of transistors
    - Since each input is connected to an RC circuit, The delay can be considered directly
    - Limited Fan-in
    - Excessive Fan-out
    - Noise vulnerability (not restoring)
    - Supply voltage offset/bias vulnerability
    - Poor high voltage levels if NMOS-only
    - Body effect
CMOS Logic

- CMOS Transmission Gates Family
  - Rules of Thumb
    - Pass-Logic may consume half the power of static logic. But be careful of $V_T$ drop resulting in static leakage
    - Pass-Gate Logic is not appropriate when long interconnects separate logic stages or when circuits have high Fan-out load (use buffering)
CMOS Logic

- CMOS Transmission Gates Family
  - AND
CMOS Logic

- CMOS Transmission Gates Family
  - OR

\[
Y = A + \overline{AB} = (A + \overline{A})(A + B) = A + B
\]

Wired OR
CMOS Logic

- CMOS Transmission Gates Family
  - MUX

\[ Y = \begin{cases} 
  A & S = 1 \\
  B & S = 0 
\end{cases} \]
CMOS Logic

- CMOS Transmission Gates Family
  - MUX
    \[
    F = (\text{In}_1 S + \text{In}_2 \overline{S})
    \]
CMOS Logic

- CMOS Transmission Gates Family
  - XOR

\[ Y = A \oplus B \]
CMOS Logic

- CMOS Transmission Gates Family
  - XNOR

![CMOS Logic Diagram](image)

\[ Y = AB + \overline{AB} \]
CMOS Logic

- CMOS Transmission Gates Family
  - D Latch

1) Load \( Q_n = D_n \)
2) Hold \( Q_n = Q_{n-1} \)
CMOS Logic

- CMOS Transmission Gates Family
  - D Latch
    1) Load LD = 1
    2) Hold LD = 0
CMOS Logic

- CMOS Transmission Gates Family
  - D Latch (Simpler Realization)
    - If in Load Mode a level voltage opposite to the output of weak inverter is applied to the input by TG, $Q = D$ and weak inverter is not damaged!
CMOS Logic

- Delay in Transmission Gate

![Diagram of CMOS Logic with Delay in Transmission Gate]
CMOS Logic

- **Delay Optimization**
  - **Delay of RC chain**
    \[
    t_p = 0.69 \sum_{k=1}^{n} C R_{eq} \quad k = 0.69 C R_{eq} \frac{n(n+1)}{2}
    \]
  - **Delay of Buffered chain**
    \[
    t_p = 0.69 \left[ \frac{n}{m} C R_{eq} \frac{m(m+1)}{2} \right] + \left( \frac{n}{m} - 1 \right) t_{buf} \\
    = 0.69 \left[ C R_{eq} \frac{n(m+1)}{2} \right] + \left( \frac{n}{m} - 1 \right) t_{buf} \\
    m_{opt} = 1.7 \sqrt{\frac{t_{Pbuf}}{C R_{eq}}}
    \]
**CMOS Logic**

- **TG Points to Remember**
  - Stored charge leaks away due to reverse-bias leakage current
  - Stored value is good for about 1 ms
  - Value must be rewritten to be valid
  - If not loaded every cycle, must ensure that latch is loaded often enough to keep data valid
  - Capacitance comes primarily from inverter’s gate logic
CMOS Logic

- TG Layout
CMOS Logic

- **TG Properties**
  - Strong pull-up
  - Strong pull-down
  - May be difficult to design into a circuit (layout) because of close proximity of N and P devices (design rule separation)
  - Always requires 2N transistors for any N x TG design
  - Many logic functions (multiplexers in particular) are easily implemented using TG based designs
CMOS Logic

- Complementary Pass-transistor Logic (CPL) or Differential (+) TG Logic
  - Dual-rail form of pass transistor logic
  - Avoids need for ratioed feedback
CMOS Logic

- CPL

![Diagrams showing CMOS logic circuits for AND/NAND operations with expressions for F.](Image)
CMOS Logic

- 4 Input NAND in CPL
CMOS Logic

- CPL Advantages
  - Differential so complementary data inputs and outputs are always available (so don’t need extra inverters)
  - Still static, since the output defining nodes are always tied to Vdd or GND through a low resistance path
  - Design is modular, all gates use the same topology, only the inputs are permuted
  - Simple XOR makes it attractive for structures like adders
  - Fast (assuming number of transistors in series is small)
CMOS Logic

- CPL Disadvantages
  - Additional routing overhead for complementary signals
  - Still have static power dissipation problems
  - $V_{OH}$ is very weak! Then we need an inverter at the output
Differential Cascode Voltage Switch Logic (DCVSL)

- Compute both true and complementary outputs using a pair of complementary NMOS pull-down network
- The PMOS transistors are driven by the output of the complementary network
- No static power consumption
- Fast response
Differential Cascode Voltage Switch Logic (DCVSL)
- Example: NAND/AND

\[ Y = AB \]

\[ \overline{Y} = \overline{A} \cdot \overline{B} \]

\[ Y = \overline{A} + \overline{B} \]
**CMOS Logic**

- **Differential Cascode Voltage Switch Logic (DCVSL)**
  - **General Design**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

![Logic Diagram]
CMOS Logic

- Differential Cascode Voltage Switch Logic (DCVSL)
  - General Design

\[ x \rightarrow + \equiv \bar{x} \rightarrow + \]
\[ u = a \bar{x} + b \bar{x} \]

\[ x \rightarrow - + \equiv x \rightarrow - + \]

\[ x \rightarrow - + \equiv x \rightarrow - + \]
\[ u_1 \quad u_2 \]

\[ u_1 \quad u_2 \]
CMOS Logic

- Differential Cascode Voltage Switch Logic (DCVSL)
  - General Design
CMOS Logic

- Differential Cascode Voltage Switch Logic (DCVSL)
  - Example: XOR/XNOR
CMOS Logic

Rules of Thumb

- Step-up (alpha) ratio of 2.7 (“e”) produces minimum power-delay product
- P vs. N (beta) ratio of 2 balances pull-up and pull-down times and noise margins
- Approximately 75% of static logic are NAND stacks (limit stack to 3-4, use ordering and tapering for speed)
- Glitches consume approximately 15% of overall chip power
- Crossover (short-circuit) current consumes ~ 10% of a static chip’s total power (but is a function of input/output slews, ie sizing)
This lecture describes the basic MOS Logic Gates which require no clock or other periodic signal for operation and also implementation of them in three following levels:

- Gate Level
- Transistor level
- Layout Level
Contents

- Introduction
- Dynamic CMOS Logic
- CMOS Domino Logic
- CD Domino Logic
- Dynamic CVSL
- Sample-Set Differential Logic (SSDL)
- Summary
As mentioned before, Digital MOS circuits can be classified into two categories:

- **Static Circuits**: require no clock or other periodic signal for operation (except sequential logic). In these circuits at every point in time (except when switching) the output is connected to either GND or Vdd via a low resistance path. 
  - fan-in of N requires 2N devices ($n \text{ N-type} + n \text{ P-type}$)

- **Dynamic Circuits**: require periodic clock signals, synchronized with data signals, for proper operation even in combinational logic. These circuits rely on the temporary storage of signal values on the (parasitic) capacitance of high impedance nodes.
  - requires only $N + 2$ transistors ($n+1 \text{ N-type} + 1 \text{ P-type}$)
  - takes a sequence of precharge and conditional evaluation phases to realize logic functions
Introduction

- Why Dynamic Logic?
- In the area of high speed, higher Fan-in, extremely low power dissipation, other digital logic circuit have been considered. In this lecture, two of these alternatives to CMOS are described. The circuits are basically NMOS or CMOS Gates with slight improvements. These are:
  - Dynamic CMOS Logic
  - CMOS Domino Logic
- Each of them have specific operating advantages over NMOS or CMOS, but exhibit disadvantages in other areas
Dynamic CMOS Logic

- Dynamic Gates use a clocked PMOS pullup
- Two modes: precharge and evaluate
Dynamic CMOS Logic

The Foot

- What if pulldown network is ON during precharge?
- Use series evaluation transistor to prevent fight
Dynamic CMOS Logic

- To make the gate dynamic, a clock pulse is applied to the gate of complementary P and N channel devices. This gate consists of an NMOS Logic circuit whose output node is precharged to Vdd by the PMOS, when the clock is zero. The output node is discharged by the NMOS transistor connected to ground when the clock is high.
Dynamic CMOS Logic

- Dynamic CMOS Logic

![Dynamic CMOS Logic Diagram](image)

**Precharge (Clk = 0)**

**Evaluate (Clk = 1)**

Out

1

Out

((AB)+C)

C

Precharge (Clk = 0)

Evaluate (Clk = 1)
Dynamic CMOS Logic

- 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 $C_L$. 
Dynamic CMOS Logic

- Canonical Forms
Dynamic CMOS Logic

- Properties of Dynamic Gates
  - Logic function is implemented by the PDN only
    - should be smaller in area than static complementary CMOS
  - Full swing outputs \( V_{OL} = \text{GND} \) and \( V_{OH} = V_{DD} \)
  - Nonratioed - sizing of the devices is not important for proper functioning (only for performance)
  - Faster switching speeds
    - reduced load capacitance due to lower number of transistors per gate \( C_{\text{int}} \) so a reduced logical effort
    - reduced load capacitance due to smaller fan-out \( C_{\text{ext}} \)
    - no \( I_{\text{sc}} \), so all the current provided by PDN goes into discharging \( C_L \)
    - Ignoring the influence of precharge time on the switching speed of the gate, \( t_{pLH} = 0 \) but the presence of the evaluation transistor slows down the \( t_{pHL} \)
Dynamic CMOS Logic

- Properties of Dynamic Gates
  - Power dissipation should be better
    - consumes only dynamic power – no short circuit power consumption since the pull-up path is not on when evaluating
    - lower $C_L$ - both $C_{int}$ (since there are fewer transistors connected to the drain output) and $C_{ext}$ (since there the output load is one per connected gate, not two)
  - But power dissipation can be significantly **higher** due to
    - higher transition probabilities
    - extra load on CLK
  - PDN starts to work as soon as the input signals exceed $V_{Tn}$, so set $V_M$, $V_{IH}$ and $V_{IL}$ all equal to $V_{Tn}$
    - low noise margin ($NM_L$)
  - Needs a precharge/evaluate clock
Dynamic CMOS Logic

- **Leakage Sources**
  - **Subthreshold conduction**
  - Transistors can’t abruptly turn ON or OFF
  - Reverse-biased PN junction diode current
    - $I_s$ depends on doping levels and area and perimeter of diffusion regions, typically < 1 fA/μm²
- **Gate Leakage**
  - Carriers may tunnel through very thin gate oxides
  - Negligible for older processes
Leakage Sources

- Output settles to an intermediate voltage determined by a resistive divider of the pull-up and pull-down networks.
- Once the output drops below the switching threshold of the fan-out logic gate, the output is interpreted as a low voltage.
Leakage Sources

- Subthreshold leakage is dominant in modern transistors
Dynamic CMOS Logic

- Solution to Charge Leakage

![Diagram showing a circuit with transistors and capacitors labeled as Clk, Mp, Mkp, Me, A, B, and Out. The diagram includes a Keeper component.]

- The diagram illustrates a circuit with transistors and capacitors labeled as Clk, Mp, Mkp, Me, A, B, and Out. The Keeper component is highlighted in red.
Dynamic CMOS Logic

- **Charge Sharing**
  - Charge stored originally on CY is redistributed (shared) over CX leading to static power consumption by downstream gates and possible circuit malfunction.
  - When \( \Delta V_{\text{out}} = -V_{dd} \left( \frac{CX}{CX + CY} \right) \) the drop in Vout is large enough to be below the switching threshold of the gate it drives causing a malfunction.

\[
V_x = V_y = \frac{C_Y}{C_x + C_Y} V_{dd}
\]
Dynamic CMOS Logic

Solution to Charge Redistribution

- Add secondary precharge transistors (at the cost of increased area and power)
  - Typically need to precharge every other node
  - Secondary precharge transistors should be small because their diffusion capacitance slows the evaluation (increase delay)
  - Big load capacitance $C_Y$ helps as well
Charge Sharing Example

What is the worst case voltage drop on y? (Assume all inputs are low during precharge and that all internal nodes are initially at 0V)
Dynamic CMOS Logic

- Charge Sharing Example

\[ \Delta V_{\text{out}} = -V_{\text{dd}} \left[ \frac{(C_a + C_c)}{(C_a + C_c + C_y)} \right] \]

\[ = -2.5V \left( \frac{30}{30 + 50} \right) = -0.94V \]
Dynamic CMOS Logic

- Backgate Coupling
  - Susceptible to crosstalk due to
    - High impedance of the output node
    - Capacitive coupling
      - Out2 capacitively couples with Out1 through the gate-source and gate-drain capacitances of M4
Dynamic CMOS Logic

- **Backgate Coupling**
  - Capacitive coupling means Out1 drops significantly so Out2 doesn’t go all the way to ground

![Diagram showing voltage levels over time for Dynamic CMOS Logic with backgate coupling effects.](image-url)
Clock Feedthrough

- A special case of capacitive coupling between the clock input of the precharge transistor and the dynamic output node due to the gate to drain capacitance.

- So voltage of Out can rise above Vdd. The fast rising (and falling edges) of the clock couple to Out.
Dynamic CMOS Logic

- Clock Feedthrough

![Diagram of clock feedthrough and voltage levels](image-url)
Dynamic CMOS Logic

- Other Effects
  - Capacitive coupling
  - Substrate coupling
  - Minority charge injection
  - Supply noise (Ground bounce)
  - Floating output nodes
Dynamic CMOS Logic

- Floating output nodes
  - Solutions:
    - Only connect to gates
    - Add staticizer to refresh the charge
Dynamic CMOS Logic

- **Advantages:**
  - For n inputs, dynamic logic requires n+2 transistors
  - Have small area, high speed and compact layouts

- **Disadvantages:**
  - Circuit operation is more complex due to the required clock
  - The inputs can only change during the precharge phase and must be stable during the *evaluate* portion of the cycle
  - Need Monotonicity
    - Cannot be cascaded
Dynamic CMOS Logic

- **Monotonicity**
  - Dynamic gates require *monotonically rising* inputs during evaluation

![Diagram showing precharge, evaluate, and output phases with a waveform indicating violations of monotonicity during evaluation.](image)
Dynamic CMOS Logic

- **Monotonicity Woes**
  - Dynamic gates produce monotonically falling outputs during evaluation
  - Illegal for one dynamic gate to drive another!

\[
A1 = \]

\[\begin{array}{c}
 \text{Precharge} \\
 X \\
 \text{Evaluate} \\
 Y \\
 \text{Precharge} \\
 X \text{ monotonically falls during evaluation} \\
 Y \text{ should rise but cannot}
\end{array}\]
Dynamic CMOS Logic

- Cascading

Only 0 → 1 transitions allowed at inputs!
Dynamic CMOS Logic

- **Cascading**
  - Input going from high to low during evaluation
    - $a$ is 5V when precharge $b = 5V$, $c = 5V$
    - During evaluation:
      - Wanted: $b \to 0V$, $c \to 5V$
      - But, $b$ takes some time to drop to 0V
      - Consequently, $c$ may fall to some unknown value

- **Solution**
  - NP-CMOS
  - NORA Logic
  - Domino logic
Dynamic CMOS Logic

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

![Dynamic CMOS Logic Diagram]
Dynamic CMOS Logic

- NORA Logic
  - WARNING: Very sensitive to noise!
Dynamic CMOS Logic

- An example
Dynamic CMOS Logic

- Dynamic 4 Input NAND Gate
Dynamic CMOS Logic

- Power Consumption
  - Power only dissipated when previous Out = 0
Dynamic CMOS Logic

- **Power Consumption**
  - Dynamic Power Consumption is Data Dependent
    - Assume signal probabilities (Dynamic 2-input NOR Gate)
      - \( P_{A=1} = 1/2 \)
      - \( P_{B=1} = 1/2 \)
    - Then transition probability
      - \( P_{1\rightarrow0} = P_{\text{out}=0} = \frac{3}{4} \)

- Switching activity can be higher in dynamic gates!

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Dynamic CMOS Logic

Rules of Thumb

- Dynamic logic is best for wide OR/NOR structure (e.g. bit-lines), providing 50% delay improvement over static CMOS.
- Dynamic logic consumes 2x power due to its phase activity (unconditional pre-charging), not counting clock power.
Dynamic CMOS Logic

- Notes
  - No need to implement the complement of the function, leading to smaller area
  - We can avoid the long PMOS chains
  - Handle the charge sharing problem and floating output nodes
  - Input transistors should not change from on to off during evaluation
CMOS Domino Logic

- It is an extension of dynamic CMOS gates that allow cascading of stages.
- The simple modification entails incorporating a static CMOS inverter at the output of each logic gate.
CMOS Domino Logic

- Stage A should be precharged in $\Phi_1$ and evaluate in $\Phi_2$
- Stage B should be precharged in $\Phi_2$ and evaluate in $\Phi_1$
**CMOS Domino Logic**

- During precharge (clk=0), the output node of the dynamic gate is precharged high and the output node of the CMOS inverter is low. Then subsequent stages will be turned off during the precharge phase.

- When the clk=1, the output of the driving gate will conditionally discharge, allowing the output of the inverter to conditionally go high. Each connected gate output can then make a transition from low-to-high, in sequence.

- There is no restriction on the number of logic stages that can be cascaded provided that all stages can evaluate during one clock pulse.
CMOS Domino Logic

- \( pc = \Phi_1 \) and \( ev = \Phi_2 \)
Won’t work! If $pc = \Phi_2$ and $ev = \Phi_1$
CMOS Domino Logic

- Why Domino?
  - Like falling dominos!
CMOS Domino Logic

- Produces monotonic outputs

![Diagram of CMOS Domino Logic](image)
Domino Optimizations

- Each domino gate triggers next one, like a string of dominos toppling over
- Gates evaluate sequentially but precharge in parallel. Thus evaluation is more critical than precharge.
- HI-skewed static stages can perform logic.
- Static inverter can be optimized to match fan-out.
Leakage

- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{OFF} \neq 0$)
  - Dynamic value will leak away over time
  - Formerly milliseconds, now nanoseconds!

- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation
CMOS Domino Logic

- **Noise Sensitivity**
  - Dynamic gates are very sensitive to noise
    - Inputs: $V_{IH} \approx V_{tn}$
    - Outputs: floating output susceptible noise
  - Noise sources
    - Capacitive crosstalk
    - Charge sharing
    - Power supply noise
    - Feedthrough noise
    - And ... !
Designing with Domino Logic

- If all the inputs come from other domino gates, then all the inputs will be low during the precharge. You don’t need to explicitly evaluate transistors.
- Need to be a little careful. When precharge begins, the first gate’s output must precharge before the next gate can precharge. Both evaluate and precharge ripple in this scheme. But, if there is already a tall stack, transistor ratioing will let precharge win anyway. (but you waste power until the precharge ripples)
Example

- During precharge, $\overline{x}, \overline{y}, \overline{z} = 1$, $x, y = 0$
- During evaluation, $\overline{x} = 0$ when $a = b = 1$
- Therefore, $z = a \ b \ c \ d$
Advantages:
- Large Fan-in, fewer transistors (n+4 transistors, whereas CMOS requires 2n)
- Single clock can be used to precharge and evaluate all stages at the same time
- It is attractive for high-speed circuits
- 1.5 – 2x faster than static CMOS
- Widely used in high-performance microprocessors

Disadvantages:
- Each logic block must incorporate a separate inverter
- Each block performs only non-inverting logic
- Monotonicity
- Leakage
- Charge sharing
- Noise
CMOS Domino Logic

- Dual Rail Domino
  - Domino only performs noninverting functions
    - AND, OR but not NAND, NOR, or XOR
  - Dual-rail domino solves this problem
    - Takes true and complementary inputs
    - Produces true and complementary outputs

<table>
<thead>
<tr>
<th>sig_h</th>
<th>sig_l</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Precharged</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>‘0’</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>‘1’</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>invalid</td>
</tr>
</tbody>
</table>
CMOS Domino Logic

- Example AND/NAND
  - Given A_h, A_l, B_h, B_l
  - Compute Y_h = A * B, Y_l = ~(A * B)
- Pulldown networks are conduction complements
**Example XOR/XNOR**
- Sometimes possible to share transistors

\[
Y_\text{l} = A \text{xnor } B
\]
\[
Y_\text{h} = A \text{xor } B
\]
CMOS Domino Logic

- **Rules of Thumb**
  - Typical domino keepers have $W/L = 5\text{-}20\%$ of effective width of evaluate tree
  - Typical domino output buffers have a beta ratio of $\sim 6:1$ to push the switch point higher for fast rise-time
We noted that dynamic inputs never make 1 to 0 transitions while in evaluation.

Two solutions:
- Precharge outputs low using an inverting gate (standard domino).
- Delay the evaluate clock until inputs settle (CD domino).
CD Domino Logic

- Self-timed dynamic logic family
- Consists of a dynamic gate, and an optional delay element for the clock signal
CD Domino Logic

Advantages
- Uses single-rail circuits, rather than dual-rail for standard domino
- Provides both inverting and non-inverting functions
- High-speed, large fan-in NOR and OR circuits
CD Domino Logic

- **Delay Matching**
  - CD domino requires delay matching between the slowest dynamic gate at a level and a delay element.
  - A 20% margin is typically added to the delay of the fixed delay element to account for PVT variations.
  - Thus, 20% of the speed gain possible with CD domino is not realized.
  - Average speed gain of (60+20)% is theoretically possible.
- Use digitally programmable delay elements (PDEs) to reduce the margin and attain a speed improvement without affecting the reliability in the presence of variations.
CD Domino Logic

- **Clocking Scheme**
  - The circuits are fully levelized.
  - The delay element on each level is tuned to the slowest gate at its level, plus a 20% margin.
Dynamic CVSL

- Positive feedback does not exist
Sample-Set Differential Logic (SSDL)

- It is one type of Dynamic CVSL with positive feedback
- By this logic the low level output is guaranteed to zero in evaluation phase
Summary

- This lecture describes many basics CMOS Logic Gates which require clock or other periodic signal for operation.
- These circuits rely on the temporary storage of signal values on the (parasitic) capacitance of high impedance nodes.