Project Phase 1 Done – Thanks for the timely response.
Phase 2 to be announced We – Launched on Fr.
Hw 6 due on Fr.
Class Material

- Last lecture
  - Pass transistor logic
  - CMOS Layout
- Today’s lecture
  - Ratioed Logic
  - Dynamic Logic
- Reading (Ch 6)
Goal: build gates faster/smaller than static complementary CMOS
Ratioed Logic LE

- Rising and falling delays aren’t the same
  - Calculate LE for the two edges separately

  \[ L_{EH} = \begin{cases} 
  \frac{3}{2}W C_G & \text{for } C_{\text{gate}} = W C_G \\
  W C_G & \text{for } \text{inv} = (3/2)W C_G 
\end{cases} \]

- For tpLH:
  - \[ \begin{align*}
  C_{\text{gate}} &= W C_G \\
  C_{\text{inv}} &= (3/2)W C_G \\
  L_{EH} &= \end{align*} \]

Ratioed Logic LE (pull-down edge)

- What is LE for \( t_{pHL} \)?
- Switch model would predict \( R_{eff} = R_n || R_p \)
  - Would that give the right answer for LE?
Time constant is smaller, but it takes more time to complete 50% \( V_{DD} \) transient (arguably)

- Rp actually takes some current away from discharging C

**Ratioed Logic Pull-down Delay**

- Think in terms of the current driving \( C_{load} \)

- When you have a conflict between currents
  - Available current is the difference between the two
  - In pseudo-nMOS case:
    \[
    R_{drive} = \frac{1}{\frac{1}{Rn} - \frac{1}{Rp}} \quad \rightarrow \quad R_{drive} = \frac{Rn}{1 - \left(\frac{Rn}{R_P}\right)}
    \]
  - (Works because \( Rp >> Rn \) for good noise margin)
Ratioed Logic LE (pull-down edge)

- For $t_{\text{pHL}}$ (assuming $R_{sqp} = 2R_{sqn}$):
  - $R_{\text{gate}} = R_n/(1-R_n/R_p) = 2R_n$
  - $C_{\text{gate}} = W C_G$
  - $R_{\text{inv}} = R_n$
  - $C_{\text{inv}} = 3W C_G$
- LE is lower than an inverter!
  - But have static power dissipation…

Improved Loads

- Adaptive Load

$M_1 >> M_2$
**Improved Loads (2)**

**Differential Cascode Voltage Switch Logic (DCVSL)**

**DCVSL Transient Response**
**DCVSL Example 1: AND**

**DCVSL Example 2**

XOR/XNOR gate
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 $2n (n \text{ N-type} + n \text{ P-type})$ devices

Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.

- Dynamic circuits only require $n + 2 (n+1 \text{ N-type} + 1 \text{ P-type})$ transistors
**Dynamic Gate**

Two phase operation

- **Precharge** (Clk = 0)
- **Evaluate** (Clk = 1)

**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 \((V_{OL} = \text{GND} \text{ and } V_{OH} = V_{DD})\)
- Non-ratioed - sizing of the devices does not affect the logic levels
- Faster switching speeds
  - reduced capacitance due to lower input capacitance \((C_{in})\)
  - no \( I_{sc} \), so all the current provided by PDN goes into discharging \( C_L \)

LE of Dynamic Gates

\[
C_{gate} = \frac{1}{LE} \]

\[
C_{gate} = \frac{1}{LE} \]
**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 ($N_{ML}$)
- Needs a precharge/evaluate clock

**Issues in Dynamic Design 1: Charge Leakage**

*Dominant component is subthreshold current*
**Solution to Charge Leakage**

Same approach as level restorer for pass-transistor logic

**Dynamic Gate VTC**

[Diagram of Dynamic Gate VTC]
**Issues in Dynamic Design 2: Charge Sharing**

- Charge initially stored on $C_L$
  - $C_A$ previously discharged

- When $A$ rises, this charge is redistributed (shared) between $C_L$ and $C_A$
- Makes $Out$ drop below $V_{DD}$
**Charge Sharing**

- Two cases:
  - \( M_a \) stays on – complete charge share
  - \( M_a \) turns off – incomplete charge share

**Complete charge share:**
- \( Q_{Ca} = V_{Out} C_a \)
- \( \Delta Q_{CL} = -V_{Out} C_a \)
- \( \Delta V_{Out} = -V_{DD} C_a / (C_a + C_L) \)

**Incomplete charge share:**
- \( Q_{Ca} = (V_{DD} - V_{TN^*}) C_a \)
- \( \Delta Q_{CL} = -(V_{DD} - V_{TN^*}) C_a \)
- \( \Delta V_{Out} = -(V_{DD} - V_{TN^*}) C_a / C_L \)

**Solution to Charge Sharing**

- Keeper helps a lot
  - Can still get failures if Out drops below inverter \( V_{sw} \)
- Another option: precharge internal nodes
  - Increases power and area
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.

Clock Feedthrough

In & Clk

Clock feedthrough

Clock feedthrough
Issues in Dynamic Design 4: Backgate Coupling

Dynamic NAND

Static NAND

Backgate Coupling Effect

Voltage

Time, ns

Clk

Out1

Out2

In
Other Effects

- Capacitive coupling
- Substrate coupling
- Minority charge injection
- Supply noise (ground bounce)

Domino Logic
Cascading Dynamic Gates

Only \( 0 \rightarrow 1 \) transitions allowed at inputs!

Domino Logic

\[ \Delta V \]
Why Named Domino?

Like falling dominos!

Properties of Domino Logic

- Only non-inverting logic can be implemented
- Very high speed
  - static inverter can be skewed, only L-H transition critical
  - Input capacitance reduced – smaller logical effort
**Domino Logic LE**

\[
\text{clk} \quad \begin{array}{c}
\text{A} \quad \text{C}_3 \\
\text{B} \quad \text{C}_3 \\
\text{C}_3
\end{array} \\
\text{LE}_{\text{inv}} = \\
\text{LE}_{\text{inv}} = 1 \\
\text{\neg LE} = \]

\[
\text{clk} \quad \begin{array}{c}
\text{A} \quad \text{C}_3 \\
\text{B} \quad \text{C}_3 \\
\text{C}_3
\end{array} \\
\text{LE}_{\text{inv}} = \\
\text{LE}_{\text{inv}} = 1 \\
\text{\neg LE} = \]

**Domino Logic LE (skewed static gate)**

\[
\text{clk} \quad \begin{array}{c}
\text{A} \quad \text{C}_3 \\
\text{B} \quad \text{C}_3 \\
\text{C}_3
\end{array} \\
\text{LE}_{\text{inv}} = \\
\text{LE}_{\text{inv}} = 1 \\
\text{\neg LE} = \]

Reference inverter: 

\[
\begin{array}{c}
\text{C}_3 \\
\text{C}_3
\end{array}
\]
Buffer “Average” LE

\[ \text{Domino buffer:} \]

\[ \text{LE}_{\text{DOM}} = \frac{2}{3} \]
\[ \text{LE}_{\text{SINV}} = \frac{5}{6} \]
\[ \text{TILE} = \frac{10}{18} \]
"Average" LE = \[\sqrt{\frac{10}{18}} \approx \frac{3}{4}\]

Designing with Domino Logic

Inputs = 0 during precharge

Can be eliminated