Announcements

- Logical effort.
- Homework 3, problem 9 will be moved to Homework 4.
- Review DeMorgan’s Laws and gate design.
Examples

- $f(a) = a$
- $f(a) = \overline{a}$
- $f(a, b) = ab$
- $f(a, b) = ab$ (Check Figure 6-33 in J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Prentice-Hall, second edition, 2003!)
- $f(a, b, c) = ab + \overline{bc}$ (try both ways).

Derive and explain.
Lecture plan

1. Non-idealities
2. DCVSL
3. Dynamic CMOS
4. Charge sharing
5. Homework
Miller effect

- If $V_D$ switches in the opposite direction of $V_G$, the effect of $C_{GD}$ is doubled.
- Consider an inverter.
- Model by using a $2C_{GD}$ capacitor to ground.
Each series transistor drops the voltage seen by the next transistor.

\[ V_T = V_{T0} + \gamma \left( \sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|} \right) \]

\[ V_{Tn2} = V_{Tn0} + \gamma \left( \sqrt{|2\phi_F + V_{int}|} - \sqrt{|2\phi_F|} \right) \]
Dynamic hazards

- Potential for two or more spurious transitions before intended transition
- Results from uneven path delays in some multi-level circuits
Dynamic hazards
Eliminating dynamic hazards

- Some approaches allow preservation of multi-level structure
  - Quite complicated to apply
- Simpler solution – Convert to two-level implementation
Still have static hazards

Potential for transient change of output to incorrect value
Problems with glitches

- These transitions result in incorrect output values at some times.
- Also result in uselessly charging and discharging wire and gate capacitances through wire, gate, and channel resistances.
  - Increase power consumption.
Glitches increase power consumption
Detecting hazards

- The observable effect of a hazard is a glitch
  - A circuit that might exhibit a glitch has a hazard
- Whether or not a hazard is observed as a glitch depends on relative gate delays
- Relative gate delays change depending on a number of factors – Conditions during fabrication, temperature, age, etc.
- Best to use abstract reasoning to determine whether hazards might be observed in practice, under some conditions
Eliminating static hazards

- Ensure that the function has a term maintaining a 0 output for all $0 \rightarrow 0$ transitions.
- Ensure that the function has a term maintaining a 1 output for all $1 \rightarrow 1$ transitions.
- There are precisely defined algorithms for this, but they build on a knowledge of logic minimization.
Where do static hazards really come from?

- **Static-0**: $A \bar{A}$
- **Static-1**: $A + \bar{A}$

Assume SOP form has no product terms containing a variable in complemented and uncomplemented forms
  - Reasonable assumption, if true, drop product term
Where do static hazards really come from?

- Assume POS form has no sum terms containing a variable in complemented and uncomplemented forms
  - Reasonable assumption, if true, drop sum term
- Assume only one input switches at a time
- Conclusion: SOP has no 0-hazards and POS has no 1-hazards
  - In other words, if you are doing two-level design, you need not analyze the other form for hazards
Living with hazards

Sometimes hazards can be tolerated

- Combinational logic whose outputs aren't observed at all times
- Synchronous systems
- Systems without tight power consumption limits
Lecture plan

1. Non-idealities
2. DCVSL
3. Dynamic CMOS
4. Charge sharing
5. Homework
Differential cascode voltage switch logic
Differential cascode voltage switch logic example

\[\text{XOR-NXOR gate}\]
Differential cascode voltage switch logic response
NMOS-only wired and

\[ F = AB \]
Level restoration
Restorer sizing

![Graph showing voltage over time for different restorer sizes.](image-url)
Depletion mode $V_T = 0$ V pass transistor

Consider leakage.
Lecture plan

1. Non-idealities
2. DCVSL
3. Dynamic CMOS
4. Charge sharing
5. Homework
Static vs. dynamic logic

- Static logic relies only on steady-state behavior of system. Eventually the output converges to a correct result.
- Dynamic logic relies on transient behavior and is sensitive to timing. Reliable design is generally trickier. Why use it?
- Static logic requires \((k_P + k_N)\) transistors for \(k\)-input gate.
- Dynamic logic requires \(k_N + 2\) transistors for \(k\)-input gate.
Dynamic logic

Two-phase operation.
Dynamic logic example

![Dynamic logic diagram]

- **Clk**: Clock signal
- **M_p**: PMOS transistor
- **M_e**: NMOS transistor
- **A**: Input A
- **B**: Input B
- **C**: Output
- **Out**: Output signal

The diagram illustrates a basic dynamic (DCVSL) CMOS logic circuit, which utilizes charge sharing to perform logic operations. The circuit is designed to toggle between two states based on the clock signal and input states, resulting in a dynamic output.
Dynamic logic operating principles I

1. Can only discharge output node once per clock period.
2. Inputs must make only one transition during evaluation.
3. Output can be in the high impedance state during and after evaluation.
4. Logic function is implemented by the pull-down network only.
5. Requires only \( k_N + 2 \) transistors.
6. Full swing outputs.
7. Non-ratioed - sizing of the devices does not affect the logic levels.
8. Reduced load capacitance due to lower input capacitance.
9. Reduced load capacitance due to smaller output loading. No Isc, so all the current provided by PDN goes into discharging CL.
Power consumption usually higher than static CMOS.

- Good: No static current.
- Good: No glitching.
- Bad: Higher transition probabilities.
- Bad: More load on clock distribution network.

\[ V_M = V_{IH} = V_{IL} = V_{TN} \] so noise margin is low.

Needs precharge and evaluation cycle.
Upcoming topics

- Example problems on recently covered material.
- Latches and flip-flops.
Review

- What are dynamic hazards?
- What are static hazards?
- What problems do hazards cause?
- What is the root cause of static hazards?
- Let’s implement a function using DCVSL.

Derive and explain.
Lecture plan

1. Non-idealities
2. DCVSL
3. Dynamic CMOS
4. Charge sharing
5. Homework
Dynamic logic charge leakage
Dynamic logic charge leakage timing diagram

CLK

V_{Out}

Evaluate

Precharge
Leakage prevention
Charge sharing
Charge sharing model

1. Determine condition by setting $\Delta V_{out} = V_{Tn}$.
2. This yields $\frac{C_a}{C_L} = \frac{V_{Tn}}{V_{DD} - V_{Tn}}$. 

$$
\text{Determine condition by setting } \Delta V_{out} = V_{Tn}.
\text{This yields } \frac{C_a}{C_L} = \frac{V_{Tn}}{V_{DD} - V_{Tn}}.
$$
Charge sharing equations

\[ \Delta V_{\text{out}} = \begin{cases} 
V_{\text{out}}^{(\text{final})} + V_{\text{DD}} &= -\frac{C_a}{C_L} \left( V_{\text{DD}} - V_{Tn}^{(V_X)} \right) & \text{if } \Delta V_{\text{out}} < V_{Tn} \\
- V_{\text{DD}} \frac{C_a}{C_a + C_L} &= & \text{if } \Delta V_{\text{out}} > V_{Tn} 
\end{cases} \]

Note: The book has a sign error when deriving the boundary point.
Preventing charge sharing problems
Transition from combinational to sequential circuits
Upcoming topics

- Sense amplifiers.
- A more formal approach to gate sizing.
Lecture plan

1. Non-idealities
2. DCVSL
3. Dynamic CMOS
4. Charge sharing
5. Homework
Homework assignment

- 7 November: Project 4.