Lecture 4&5
CMOS Circuits

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese566/
Worst-Case $V_{OL}$

a) Identify the worst-case input combination(s) for $V_{OL}$.
b) Calculate the worst-case value of $V_{OL}$. (Assume that all pull-down transistors have the same body bias and initially, that $V_{OL} \approx 5\% \ V_{DD}$.)

Fig. 3
Combinational Logic (Delay Analysis)
Sequential Circuits
Memory
RC Delay

- Lumped Model
  - C only
  - RC model

Figure 4.11  Distributed versus lumped capacitance model of wire. $C_{lumped} = L \times c_{wire}$, with $L$ the length of the wire and $c_{wire}$ the capacitance per unit length. The driver is modeled as a voltage source and a source resistance $R_{driver}$.

$$C_{lumped} \frac{dV_{out}}{dt} + \frac{V_{out} - V_{in}}{R_{driver}} = 0$$

$$\tau = R_{driver} \times C_{lumped}$$

$$V_{out}(t) = (1 - e^{-t/\tau}) V$$

$t_{50\%} = 0.69 \times 10 \text{ K}\Omega \times 11 \text{ pF} = 76 \text{ nsec}$

$t_{90\%} = 2.2 \times 10 \text{ K}\Omega \times 11 \text{ pF} = 242 \text{ nsec}$
Elmore Delay Formula

- **Assumptions regarding the RC network**
  - the network has a single input node
  - all the capacitors are between a node and the ground
  - the network does not contain any resistive loops (tree)

- **Unique resistive path**
  - path resistance
  \[ R_{44} = R_1 + R_3 + R_4 \]
  - shared path resistance
  \[ R_{ik} = \sum R_j \Rightarrow (R_j \in \text{path}(s \rightarrow i) \cap \text{path}(s \rightarrow k)) \]

Figure 4.12 Tree-structured RC network.
Example: RC Ladder/Chain

![RC Ladder/Chain Diagram]

**Figure 4.13** RC chain.

\[
\tau_{DN} = \sum_{i=1}^{N} C_i \sum_{j=1}^{i} R_j = \sum_{i=1}^{N} C_i R_{ii}
\]

**What if distributed?**

\[
\tau_{DN} = \left(\frac{L}{N}\right)^2 (rc + 2rc + \ldots + Nrc) = (rcL^2)\frac{N(N+1)}{2N^2} = RC\frac{N+1}{2N}
\]

\[
\tau_{DN} = \frac{RC}{2} = \frac{rcL^2}{2}
\]
Distributed RC Line

\[ V_{in} \text{ r\Delta L} V_{i-1} \text{ r\Delta L} V_i \text{ r\Delta L} V_{i+1} \text{ r\Delta L} \ldots \text{ r\Delta L} V_{out} \]

(a) Distributed model

\[ c\Delta L \frac{\partial V_i}{\partial t} = \frac{V_{i+1} - V_i + (V_{i-1} - V_i)}{r\Delta L} \]

(b) Schematic symbol for distributed line

\[ V_{in} \text{ (r,c,L)} \text{ V_out} \]

\[ rc \frac{\partial^2 V}{\partial t^2} = \frac{\partial^2 V}{\partial x^2} \]

\[ V_{out}(t) = 2erfc\left(\sqrt{\frac{RC}{4t}}\right) \]

\[ t \ll RC \]

\[ V_{out}(t) = 1.0 - 1.366e^{-2.5359 \frac{t}{RC}} + 0.366e^{-9.4641 \frac{t}{RC}} \]

\[ t \gg RC \]
Calculate Wire Delay

- **Rule of Thumb**
  - RC delay should only be considered when \( t_{pRC} \gg t_{pgate} \)

\[
L_{crit} \gg \sqrt[0.38]{t_{pgate} \over \sqrt{0.38rc}}
\]

- RC delay should only be considered when the rise/fall time at the line input is smaller than RC: \( t_{rise} < RC \)

<table>
<thead>
<tr>
<th>Voltage range</th>
<th>Lumped RC network</th>
<th>Distributed RC network</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 50% ((t_p))</td>
<td>0.69 RC</td>
<td>0.38 RC</td>
</tr>
<tr>
<td>0 → 63% ((t))</td>
<td>( RC )</td>
<td>0.5 RC</td>
</tr>
<tr>
<td>10% → 90% ((t_r))</td>
<td>2.2 RC</td>
<td>0.9 RC</td>
</tr>
<tr>
<td>0% → 90%</td>
<td>2.3 RC</td>
<td>1.0 RC</td>
</tr>
</tbody>
</table>
Inverter Propagation Delay

- Simplified switch model
  - find equivalent resistance

\[
R_{eq} = \frac{1}{V_{DD}/2} \int_{V_{DD}/2}^{V_{DD}} \frac{V}{I_{DSAT}(1 + \lambda V)}dV 
\approx \frac{3V_{DD}}{4I_{DSAT}} \left( 1 - \frac{7}{9} \lambda V_{DD} \right)
\]

with \( I_{DSAT} = k' \frac{W}{L} \left( (V_{DD} - V_T) V_{DSAT} - \frac{V_{DSAT}^2}{2} \right) \)

- apply RC delay

\[
t_{pHL} = \ln(2)R_{eqn}C_L = 0.69R_{eqn}C_L
\]
\[
t_{pLH} = 0.69R_{eqp}C_L
\]
\[
t_p = \frac{t_{pHL} + t_{pLH}}{2} = 0.69C_L \left( \frac{R_{eqn} + R_{eqp}}{2} \right)
\]
Minimize Inverter Delay

- Reduce $C_L$
  - keep the drain diffusion areas as small as possible
- Increase W/L ratio
  - will minimize the delay until intrinsic capacitance dominate $\Rightarrow$ “self-loading”
- Increase $V_{DD}$
  - reliability concerns
- NMOS/PMOS ratio

\[
C_L = (C_{dp1} + C_{dn1}) + (C_{gp2} + C_{gn2}) + C_W
\]

\[
C_L = (1 + \beta)(C_{dn1} + C_{gn2}) + C_W
\]

\[
t_p = \frac{0.69}{2} \frac{(1 + \beta)(C_{dn1} + C_{gn2}) + C_W}{R_{eqn} + \frac{R_{eqp}}{\beta}}
\]

\[
t_p = 0.345((1 + \beta)(C_{dn1} + C_{gn2}) + C_W) R_{eqn} \left(1 + \frac{r}{\beta}\right)
\]

\[
\beta_{opt} = \sqrt{r \left(1 + \frac{C_W}{C_{dn1} + C_{gn2}}\right)}
\]
Sizing Inverter for Performance

- Inverter delay model

\[
    t_p = 0.69 R_{eq} (C_{int} + C_{ext}) \\
    = 0.69 R_{eq} C_{int} (1 + C_{ext}/C_{int}) = t_{p0} (1 + C_{ext}/C_{int})
\]

- Size scaling factor (S)

\[
    t_p = 0.69 (R_{ref} / S)(SC_{iref})(1 + C_{ext} / (SC_{iref})) \\
    = 0.69 R_{ref} C_{iref} \left( 1 + \frac{C_{ext}}{SC_{iref}} \right) = t_{p0} \left( 1 + \frac{C_{ext}}{SC_{iref}} \right)
\]
Sizing Inverter Chain

- Intrinsic delay

\[ C_{\text{int}} = \gamma C_g \]

\[ t_p = t_{p0} \left( 1 + \frac{C_{\text{ext}}}{\gamma C_g} \right) = t_{p0} \left( 1 + f/\gamma \right) \]

- Inverter delay chain

\[ t_{p,j} = t_{p0} \left( 1 + \frac{C_{g,j+1}}{\gamma C_{g,j}} \right) = t_{p0} \left( 1 + f_j/\gamma \right) \]

\[ t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0} \sum_{j=1}^{N} \left( 1 + \frac{C_{g,j+1}}{\gamma C_{g,j}} \right), \text{ with } C_{g,N+1} = C_L \]

\[ f = N \sqrt[4]{C_L/C_{g,1}} = N \sqrt[4]{F} \]

\[ t_p = N t_{p0} \left( 1 + N \sqrt[4]{F/\gamma} \right). \]
Optimal Number of Inverters in the Chain

\[ \gamma + N \sqrt{\frac{F}{N}} - \frac{N \sqrt{F} \ln F}{N} = 0 \]

or equivalently

\[ f = e^{(1 + \gamma/f)} \]

(a) Optimum effective fanout \( f \) (or inverter scaling factor) as a function of the self-loading factor \( \gamma \) in an inverter chain.

(b) Normalized propagation delay \( \frac{t_p}{t_{prop}} \) as a function of the effective fanout \( f \) for \( \gamma = 1 \).

Table 5.3 \( t_{opt}/t_{p0} \) versus \( x \) for various driver configurations.

<table>
<thead>
<tr>
<th>( F )</th>
<th>Unbuffered</th>
<th>Two Stage</th>
<th>Inverter Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>8.3</td>
<td>8.3</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>22</td>
<td>16.5</td>
</tr>
<tr>
<td>1000</td>
<td>1001</td>
<td>65</td>
<td>24.8</td>
</tr>
<tr>
<td>10,000</td>
<td>10,001</td>
<td>202</td>
<td>33.1</td>
</tr>
</tbody>
</table>
Examples: Inverter Sizing and Delay

\[ C_L = 64 \, C_{g,1}. \]

\[ \frac{4C_{g,2}}{C_{g,1}} = \frac{4C_{g,3}}{C_{g,2}} = \frac{C_L}{C_{g,3}} \]

\[
\begin{align*}
t_p &= 0.69R_{dr}C_{int} + (0.69R_{dr} + 0.38R_w)C_w + 0.69(R_{dr} + R_w)C_{fan} \\
&= 0.69R_{dr}(C_{int} + C_{fan}) + 0.69(R_{dr}c_w + r_wC_{fan})L + 0.38r_wc_wL^2
\end{align*}
\]
Propagation Delay of Complex Logic Gates

- Depend on inputs

<table>
<thead>
<tr>
<th>Input Data Pattern</th>
<th>Delay (psec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A = B = 0 \rightarrow 1$</td>
<td>69</td>
</tr>
<tr>
<td>$A = 1, B = 0 \rightarrow 1$</td>
<td>62</td>
</tr>
<tr>
<td>$A = 0 \rightarrow 1, B = 1$</td>
<td>50</td>
</tr>
<tr>
<td>$A = B = 1 \rightarrow 0$</td>
<td>35</td>
</tr>
<tr>
<td>$A = 1, B = 1 \rightarrow 0$</td>
<td>76</td>
</tr>
<tr>
<td>$A = 1 \rightarrow 0, B = 1$</td>
<td>57</td>
</tr>
</tbody>
</table>

- Internal cap matters

\[ t_{pHL} = 0.69(R_1 \cdot C_1 + (R_1 + R_2) \cdot C_2 + (R_1 + R_2 + R_3) \cdot C_3 + (R_1 + R_2 + R_3 + R_4) \cdot C_L) \]
Sizing Combinational Network for Performance

- **Inverter delay**
  \[ t_p = t_{p0}(1 + \frac{C_{ext}}{\gamma C_g}) = t_{p0}(1 + f/\gamma) \]

- **Complex logic delay**
  - \( p \): ratio of the intrinsic (unloaded) delay of the complex gate and the simple inverter. Affected by both topology and layout style
  - \( g \): logic effort
  - \( f \): electrical effort

<table>
<thead>
<tr>
<th>Gate type</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>( n )-input NAND</td>
<td>( n )</td>
</tr>
<tr>
<td>( n )-input NOR</td>
<td>( n )</td>
</tr>
<tr>
<td>( n )-way multiplexer</td>
<td>( 2n )</td>
</tr>
<tr>
<td>XOR, NXOR</td>
<td>( n2^{n-1} )</td>
</tr>
</tbody>
</table>
Logic Effort (g)

- For a given capacitive load, complex gates have to work harder than an inverter to produce similar response.

Table 6.4: Logic efforts of common logic gates, assuming a PMOS/NMOS ratio of 2.

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAND</td>
<td>4/3</td>
<td>5/3</td>
<td>(n+2)/3</td>
<td></td>
</tr>
<tr>
<td>NOR</td>
<td>5/3</td>
<td>7/3</td>
<td>(2n+1)/3</td>
<td></td>
</tr>
<tr>
<td>Multiplexer</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>XOR</td>
<td>4</td>
<td>12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optimal Sizing of Combinational Network

- Gate effort
  - \( h = f^g \)

\[
t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0} \sum_{j=1}^{N} \left( p_j + \frac{f^g_j}{\gamma} \right)
\]

- Optimal delay condition

\[
f_1 g_1 = f_2 g_2 = \ldots = f_N g_N
\]

\[
F = f_1 f_2 \ldots f_N = C_L / C_{g1}
\]

\[
G = g_1 g_2 \ldots g_N
\]

\[
h = N \sqrt{FG} = N \sqrt{H},
\]

\[
D = t_{p0} \left( \sum_{j=1}^{N} p_j + \frac{N(N\sqrt{H})}{\gamma} \right)
\]

Graphical representation of the network with a formula for calculating \( G \).

\[
G = 1 \times \frac{5}{3} \times \frac{5}{3} \times 1 = \frac{25}{9}
\]
Outline

Combinational Logic

Interconnect

Sequential State
Level-Sensitive Latch
SR Latch

- Basic NOR latch

R (reset)  \[ Q \]
S (set)  \[ \overline{Q} \]

\[ S \quad 0 \]
\[ R \quad 0 \]
\[ Q \quad 0 \]
\[ \overline{Q} \quad 1 \]

\[ t_{pd} \]

Set  \[ \text{No change} \]  \[ \text{Reset} \]  \[ \text{not allowed} \]  \[ \text{unstable} \]
Other SR Latches

- Clocked

- NAND SR latch
Edge-Sensitive Flip-Flop
Outline

Combinational Logic (Delay Analysis)

Sequential Circuits

Memory
Static RAM

- **Applications**
  - CPU register file, cache, embedded memory, DSP

- **Characteristics**
  - 6 transistor per cell, other topologies
  - no need to refresh
  - access time ~ cycle time
  - no charge to leak
  - faster, more area, more expensive
SRAM Operation

- **Standby**
  - word line de-asserted

- **Read**
  - precharge bit lines
  - assert WL
  - BL rise/drop slightly

- **Write**
  - apply value to BL
  - assert WL
  - input drivers stronger
SRAM Architecture

Column Drivers

Bitcell Array

Bitlines for bit n

Sense Amplifiers

source: semiengineering.com
Multi-Bank Layout

source: semiengineering.com
Questions?

Comments?

Discussion?
Homework #3

- Posted on class website
- Due on 2/6 at 2:30pm
- Solution will be posted on 2/5 evening
- Use it as an exercise to prepare for exam
- Will release excerpts from textbook on BlackBoard
In-Class Exam

• 2/6 in the lecture room
• Starts at 2:40pm and ends at 4:00pm
• Designed to be completed in 60min
• 75% material similar to HW0 and HW1
• 25% material similar to HW2 and HW3
Design Tool Tutorials

• Standard-cell based design flow
Design Tool Tutorials

• Functional Simulation
  - tool: Synopsys VCS
  - simulate your HDL (eg. Verilog) code to verify functionality

• Logic Synthesis
  - tool: Synopsys Design Compiler (DC)
  - convert/synthesize behavioral/RTL level HDL to gate-level netlist (i.e. connectivity list)

• Physical Design (Place & Route)
  - tool: Cadence Encounter
  - given the gate-level netlist, place and route the design to complete an IC chip in its final physical form
Lab1: Design Tool Tutorials

- Will be posted on 2/7 before the Wed lecture
- TA will give hand-on introduction on 2/8
- Please bring your laptop
- Please set up your SEAS account
- Please send your Github ID to Yunfei
- Please walk through the Linuxlab tutorial
- Please read Lab1 before the lecture, so you can ask questions
- Due on 2/22 at 2:30pm
Acknowledgement

Cornell University, ECE 5745