Administration

- First Midterm Thursday

- *Decided on closed book (sorry), but will give you extra time. We will make an exam that is expected to take 90 minutes, but will give you 3 hours.*
  - 5-8PM
  - 405 Soda
  - Material: Everything up to slide 35 of Thursdays lecture (Lecture 8) – that is, CMOS logic is included.
Static Complementary CMOS

V\text{DD}

\begin{align*}
\text{PUN} & \quad \text{Inverting switches} \\
F(\text{In}_1, \text{In}_2, \ldots, \text{In}_N) \\
\text{PDN} & \quad \text{Non-Inverting switches}
\end{align*}

PUN and PDN are dual logic networks
PUN and PDN functions are complementary

- Full rail-to-rail swing
- Symmetrical VTC
- No (...) static power dissipation
- Direct path current during switching
Switch (Transmission Gate Logic)

Static:
Output always defined by GND or VDD, never both

Network of switches

$\text{In}_1$
$\text{In}_2$
$\text{In}_3$
$\text{In}_4$

No connections to GND or VDD

$F$
Tri-state Buffers

Tri-state Buffer:

<table>
<thead>
<tr>
<th>OE</th>
<th>N</th>
<th>CUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

“high impedance” (output disconnected)

Variations:

Inverting buffer

Inverted enable

transmission gate useful in implementation
Tri-state Buffers

Tri-state buffers enable “bidirectional” connections.

Tri-state buffers are used when multiple circuits all connect to a common wire. Only one circuit at a time is allowed to drive the bus. All others “disconnect” their outputs, but can “listen”.

Tri-state Based Multiplexor

If $s=1$ then $c=a$

Transistor Circuit for inverting multiplexor:
Latches and Flip-flops

Positive Level-sensitive \textit{latch}:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{latch_diagram}
\end{figure}

Positive Edge-triggered \textit{flip-flop} built from two level-sensitive latches:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{flip-flop_diagram}
\end{figure}

\textbf{Latch Implementation}:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{Latch_Implementation_diagram}
\end{figure}
Digital abstraction
Noise and Digital Systems

- Circuit needs to works despite “analog” noise
  - Digital gates can and must reject noise
  - This is actually how digital systems are defined

- Digital system is one where:
  - Discrete values mapped to analog levels and back
  - Elements (gates) can reject noise
    - For “small” amounts of noise, output noise is less than input noise
  - Thus, for sufficiently “small” noise, the system acts as if it was noiseless
  - This is called regeneration
Bridging the digital and the analog worlds

- How to represent 0’s and 1’s in a world that is analog?

**The Static Definition**

- **Logic 0:** $V_{\text{MIN}} \leq V \leq V_{\text{OL}}$
- **Logic 1:** $V_{\text{OH}} \leq V \leq V_{\text{MAX}}$
- **Undefined logic value:** $V_{\text{OL}} \leq V \leq V_{\text{OII}}$
Ideal Inverter

Circuit representation and ideal transfer function:

Define *switching point* or *logic threshold*:

- \( V_M \equiv \text{input voltage for which } V_{\text{OUT}} = V_{\text{IN}} \)
  - For \( 0 \leq V_{\text{IN}} < V_M \) \( \Rightarrow V_{\text{OUT}} = V^+ \)
  - For \( V_M < V_{\text{IN}} \leq V^+ \) \( \Rightarrow V_{\text{OUT}} = 0 \)

Ideal inverter returns well defined logical outputs (0 or \( V^+ \)) even in the presence of considerable noise in \( V_{\text{IN}} \) (from voltage spikes, crosstalk, etc.) \( \Rightarrow \) signal is *regenerated*!
“Real Inverter”

- **Logic 0:**
  - \( V_{\text{MIN}} \) = output voltage for which \( V_{\text{IN}} = V^+ \)
  - \( V_{\text{OL}} \) = smallest output voltage where slope = -1

- **Logic 1:**
  - \( V_{\text{OH}} \) = largest output voltage where slope = -1
  - \( V_{\text{MAX}} \) = output voltage for which \( V_{\text{IN}} = 0 \)
If range of output values $V_{OL}$ to $V_{OH}$ is wider than the range of input values $V_{IL}$ to $V_{IH}$, then the inverter exhibits some noise immunity. (|Voltage gain| > 1)

Quantify this through noise margins.
Definition of Noise Margins

Noise margin high: \( NM_H = V_{OH} - V_{IH} \)

Noise margin low: \( NM_L = V_{IL} - V_{OL} \)
Simulated Inverter VTC (Spice)

- $V_{OH} =$
- $V_{OL} =$
- $V_{IL} =$
- $V_{IH} =$
- $N_{MH} =$
- $N_{ML} =$
- $V_M =$
Transient properties
The Switch – Dynamic Model

\[ |V_{GS}| \geq |V_T| \]
The Switch – Dynamic Model (Simplified)

\[ |V_{GS}| \geq |V_T| \]
The Switch Inverter: Transient Response

\[ V(t) = V_0 e^{-t/RC} \]

\[ t_{1/2} = \ln(2) \times RC \]

\[ t_{pHL} = f(R_{on}C_L) = 0.69 R_n C_L \]

(a) Low-to-high

(b) High-to-low
Switch Sizing

What happens if we make a switch $W$ times larger (wider)

$$|V_{GS}| \geq |V_T|$$

$C_{GW}$

$C_{SW}$

$R_{on}/W$
Switch Parasitic Model

The pull-down switch (NMOS)

Minimum-size switch

Sizing the transistor (factor $W$)

We assume transistors of minimal length (or at least constant length). $R$’s and $C$’s in units of per unit width.
**PMOS Sizing**

The PMOS challenge:

For the same voltages, it provides less current (approximately 2 times less)
Switch Parasitic Model

The pull-up switch (PMOS)

Minimum-size switch

Sized for symmetry

General sizing
Inverter Parasitic Model

\[ C_{in} = 3WC_G \]

\[ C_{int} = 3WC_D = 3W\gamma C_G \]

Drain and gate capacitance of transistor are directly related by process \((\gamma \approx 1)\)

\[ C_D = \gamma C_G \]

\[ t_p = 0.69\left(\frac{R_N}{W}\right)(3W\gamma C_G) = 0.69(3\gamma)R_NC_G \]

Intrinsic delay of inverter independent of size
**Inverter with Load Capacitance**

\[ V_{in} = 3WC_G \]

\[ C_{int} = \frac{3W\gamma C_G}{R_N} \]

\[ V_{out} = C_L \]

\[ t_p = 0.69 \left( \frac{R_N}{W} \right) (C_{int} + C_L) \]

\[ = 0.69 \left( \frac{R_N}{W} \right) (3W\gamma C_G + C_L) \]

\[ = 0.69(3C_G R_N)(\gamma + \frac{C_L}{C_{in}}) \]

\[ = t_{inv}(\gamma + \frac{C_L}{C_{in}}) = t_0(\gamma + f) \]

\[ f = \text{fanout} = \text{ratio between load and input capacitance of gate} \]
**Inverter Delay Model**

\[ t_p = t_{inv}(\gamma + f) \]

- \( t_{inv} \): Technology constant
  - Can be dropped from expression
  - Delay unit-less variable (expressed in unit delays)

**Question:** how does transistor sizing (W) impact delay?
Inverter Delay Optimization
Inverter Chain

- For some given $C_L$:
  - How many stages are needed to minimize delay?
  - How to size the inverters?
- Anyone want to guess the solution?
Careful about Optimization Problems

- Get fastest delay if build one **very** big inverter
  - So big that delay is set only by self-loading

- Likely not the problem you’re interested in
  - Someone has to drive this inverter…
Engineering Optimization Problems in General

- Need to have a set of constraints
- Constraints key to:
  - Making the result useful
  - Making the problem have a ‘clean’ solution

- For sizing problem:
  - Need to constrain size of first inverter
Delay Optimization Problem #1

- You are given:
  - A fixed number of inverters
  - The size of the first inverter
  - The size of the load that needs to be driven

- Your goal:
  - Minimize the delay of the inverter chain

- Need model for inverter delay vs. size
Apply to Inverter Chain

$$t_p = t_{p1} + t_{p2} + \ldots + t_{pN}$$

$$t_{pj} = t_{inv} \left( \gamma + \frac{C_{in,j+1}}{C_{in,j}} \right)$$

$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{inv} \sum_{i=1}^{N} \left( \gamma + \frac{C_{in,j+1}}{C_{in,j}} \right), \quad C_{in,N+1} = C_L$$
Delay equation has $N-1$ unknowns, $C_{in,2} \ldots C_{in,N}$

To minimize the delay, find $N-1$ partial derivatives:

\[ t_p = \ldots + t_{inv} \frac{C_{in,j}}{C_{in,j-1}} + t_{inv} \frac{C_{in,j+1}}{C_{in,j}} + \ldots \]

\[ \frac{dt_p}{dC_{in,j}} = t_{inv} \frac{1}{C_{in,j-1}} - t_{inv} \frac{C_{in,j+1}}{C_{in,j}^2} = 0 \]
Result: every stage has equal fanout (f):

\[
\frac{C_{in,j}}{C_{in,j-1}} = \frac{C_{in,j+1}}{C_{in,j}}
\]

Size of each stage is geometric mean of two neighbors:

\[
C_{in,j} = \sqrt{C_{in,j-1}C_{in,j+1}}
\]

Equal fanout \(\rightarrow\) every stage will have same delay
When each stage has same fanout $f$:

$$f^N = F = \frac{C_L}{C_{in,1}}$$

Fanout of each stage:

$$f = \sqrt[N]{F}$$

Minimum path delay:

$$t_p = N t_{inv} \left( \gamma + \sqrt[N]{F} \right)$$
Example

\[ C_L = 8 \ C_1 \]

\( C_L/C_1 \) has to be evenly distributed across \( N = 3 \) stages:
Delay Optimization Problem #2

- You are given:
  - The size of the first inverter
  - The size of the load that needs to be driven
- Your goal:
  - Minimize delay by finding optimal number and sizes of gates
- So, need to find $N$ that minimizes:

$$t_p = N t_{inv} \left( \gamma + \sqrt[4]{\frac{C_L}{C_{in}}} \right)$$
Untangling the Optimization Problem

- Rewrite \( N \) in terms of fanout/stage \( f \):

\[
N = \frac{\ln \left( \frac{C_L}{C_{in}} \right)}{\ln f}
\]

\[
t_p = N t_{inv} \left( \left( \frac{C_L}{C_{in}} \right)^{1/N} + \gamma \right) = t_{inv} \ln \left( \frac{C_L}{C_{in}} \right) \left( \frac{f + \gamma}{\ln f} \right)
\]

\[
\frac{\partial t_p}{\partial f} = t_{inv} \ln \left( \frac{C_L}{C_{in}} \right) \frac{\ln f - 1 - \gamma / f}{\ln^2 f} = 0
\]

\[
f = \exp \left( 1 + \gamma / f \right)
\]

(no explicit solution)

For \( \gamma = 0, f = e, N = \ln \left( \frac{C_L}{C_{in}} \right) \)
Optimum Effective Fanout $f$

- Optimum $f$ for given process defined by $\gamma$

$$ f = \exp(1 + \gamma / f) $$

$\gamma$-axis

For $\gamma = 1$

$$ f_{opt} = 3.6 $$
In Practice: Plot of Total Delay

- Why the shape?
- Curves very flat for $f > 2$
  - Simplest/most common choice: $f = 4$

[Hodges, p.281]
Normalized Delay As a Function of F

\[ t_p = N t_{inv} \left( \gamma + \frac{N}{\sqrt{F}} \right), \quad F = \frac{C_L}{C_{in}} \]

<table>
<thead>
<tr>
<th></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>

(\(\gamma = 1\))

[Rabaey: page 210]
Buffer Design

<table>
<thead>
<tr>
<th>( N )</th>
<th>( f )</th>
<th>( t_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Logical Effort
Question #1

How to best combine logic and drive for a big capacitive load?
Question #2

- All of these are “decoders”
  - Which one is “best”?
Method to answer both of these questions

- Extension of buffer sizing problem
- Logical effort
Complex Gate Sizing
Complex Gate Sizing: NAND-2 Example

- \(C_{gnand} = 4C_G = \left(\frac{4}{3}\right)C_{ginv}\)
- \(C_{dnand} = 6C_D = 6\gamma C_G = 2\gamma C_{ginv}\)
- \(f = C_L/C_{gnand} = \left(\frac{3}{4}\right)C_L/C_{ginv}\)

\[
t_{pNAND} = kR_N(C_{dnand} + C_L) = kR_N(2\gamma C_{ginv} + C_L) = kR_N C_{ginv} \left(2\gamma + C_L/C_{ginv}\right) = t_{inv} \left(2\gamma + \left(\frac{4}{3}\right)f\right)
\]
Logical Effort

- Defines ease of gate to drive external capacitance
- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates
- Logical effort LE is defined as:
  - \((R_{eq,\text{gate}}C_{in,\text{gate}})/(R_{eq,\text{inv}}C_{in,\text{inv}})\)
  - Easiest way to calculate (usually):
    - Size gate to deliver same current as an inverter, take ratio of gate input capacitance to inverter capacitance
- LE increases with gate complexity
Logical Effort

$$t_{pgate} = t_{inv} (p + LEf)$$

Measure everything in units of $t_{inv}$ (divide by $t_{inv}$):

$p$ – intrinsic delay - gate parameter $\neq f(W)$

$LE$ – logical effort – gate parameter $\neq f(W)$

$f$ – electrical fanout $= \frac{C_L}{C_{in}} = f(W)$

Normalize everything to an inverter:

$LE_{inv} = 1$, $p_{inv} = \gamma$
Delay of a Logic Gate

Gate delay:

\[ \text{Delay} = EF + p \] (measured in units of \( t_{inv} \))

- effective fanout
- intrinsic delay

Effective fanout:

\[ EF = LE \cdot f \]

- logical effort
- electrical fanout = \( C_L/C_{in} \)

Logical effort is a function of topology, independent of sizing
Effective fanout is a function of load/gate size
Logical Effort of Gates

\[ p = \gamma \cdot \text{Fan-in (for top input)} \]

Normalized delay (d) vs. Fan-out (f)

- \( t_{p\text{NAND-2}} \)
- \( t_{p\text{INV}} \)

LE =

\[ p = d = \]
1. Size for same resistance as inverter
2. LE = ratio of input cap of gate versus inverter

Intrinsic capacitance \( (C_{dnor}) = \)

\( t_{pint} \) (NOR) =
Any logic function can be implemented using NOR gates only or NAND gates only!

Which of the two approaches is preferable in CMOS (from a performance perspective)?
## Logical Effort

[From Sutherland, Sproull, Harris]

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>Number of Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>NAND</td>
<td>4/3</td>
</tr>
<tr>
<td>NOR</td>
<td>5/3</td>
</tr>
<tr>
<td>Multiplexer</td>
<td>2</td>
</tr>
<tr>
<td>XOR</td>
<td>4</td>
</tr>
</tbody>
</table>
Optimizing Complex Combinational Logic
Multistage Networks

\[ \text{Delay} = \sum_{i=1}^{N} (p_i + \text{LE}_i \cdot f_i) \]

Effective fanout: \( \text{EF}_i = \text{LE}_i f_i \)

Path delay \( D = \sum d_i = \sum p_i + \sum \text{EF}_i \)

Path electrical fanout: \( F = \frac{C_L}{C_{in}} = \prod f_i \)

Path logical effort: \( \Pi \text{LE} = \text{LE}_1 \text{LE}_2 \ldots \text{LE}_N \)

Path effort: \( PE = \Pi \text{LE} F \)

Only for tree networks
Adding branching

Branching effort: 

\[ b = \frac{C_{L,\text{on\_path}} + C_{L,\text{off\_path}}}{C_{L,\text{on\_path}}} \]
Multistage Networks

\[ Delay = \sum_{i=1}^{N} (p_i + LE_i \cdot f_i) \]

Effective fanout: \( EF_i = LE_i f_i \)

Path delay \( D = \Sigma d_i = \Sigma p_i + \Sigma EF_i \)

Path electrical fanout: \( F = \frac{C_L}{C_{in}} \)

Branching effort: \( \Pi B = b_1 b_2 \ldots b_N \)

\[ \Pi f_i = \Pi B F \quad (\text{assuming all paths in the tree are important}) \]

Path logical effort: \( \Pi LE = LE_1 LE_2 \ldots LE_N \)

Path effort: \( PE = \Pi LE \ \Pi B \ F \)
Optimum Effort per Stage

When each stage bears the same effort (effective fanout):

\[ EF^N = PE \]

\[ EF = \sqrt[N]{PE} \]

Effective fanouts: \( LE_1 f_1 = LE_2 f_2 = \ldots = LE_N f_N \)

Minimum path delay

\[ \hat{D} = \sum_{i=1}^{N} (LE_i f_i + p_i) = N \cdot PE^{1/N} + \sum_{i=1}^{N} p_i \]
Optimal Number of Stages

For a given load, and given input capacitance of the first gate, find optimal number of stages and optimal sizing

\[ D = N \cdot PE^{1/N} + \sum P_i \]

Remember: we can always add inverters to the end of the chain

The ‘best effective fanout’ \( EF = PE^{1/\hat{N}} \) is still around 4 (3.6 with \( \gamma = 1 \))
Method of Logical Effort: Summary

- Compute the path effort: \( \text{PE} = (\prod \text{LE})B\text{F} \)
- Find the best number of stages \( N \sim \log_4 \text{PE} \)
- Compute the effective fanout/stage \( \text{EF} = \text{PE}^{1/N} \)
- Sketch the path with this number of stages
- Work either from either end, find sizes:
  \( C_{in} = C_{out} \times \text{LE}/\text{EF} \)

Optimizing Complex Combinational Logic: Examples
Example 1: No branching

Electrical fanout, $F = \Pi LE = PE = EF/\text{stage} = a = b = c = \frac{5}{c}$
Example 1: No branching

Electrical fanout, $F = 5$

\[ \Pi LE = \frac{25}{9} \]

\[ PE = \frac{125}{9} \]

\[ EF/\text{stage} = 1.93 \]

\begin{align*}
\text{From the back} & \\
5/c &= 1.93 \\
(5/3)c/b &= 1.93 \\
(5/3)b/a &= 1.93
\end{align*}

$a, b, c$ are input capacitances normalized to the unit inverter

The diagram shows the connectivity of the circuit with the following LE values:

- LE of the first inverter: 1
  - $f = a$
- LE of the second inverter: \( \frac{5}{3} \)
  - $f = b/a$
- LE of the third inverter: \( \frac{5}{3} \)
  - $f = c/b$
- LE of the fourth inverter: 1
  - $f = 5/c$
Our old problem: which one is better?

\[
LE = 10/3 \quad 1 \\
\Pi \Pi LE = 10/3 \\
P = 8 \quad + \quad 1
\]

\[
LE = 2 \quad 5/3 \\
\Pi \Pi LE = 10/3 \\
P = 4 \quad + \quad 2
\]

\[
LE = 4/3 \quad 5/3 \quad 4/3 \quad 1 \\
\Pi \Pi \Pi \Pi LE = 80/27 \\
P = 2 \quad + \quad 2 \quad + \quad 2 \quad + \quad 1
\]
Adding Branching

\[
LE = 1 \\
F = \frac{90}{5} = 18 \\
PE = 18 \text{ (wrong!)}
\]

\[
EF_1 = \frac{(15+15)}{5} = 6 \\
EF_2 = \frac{90}{15} = 6 \\
PE = 36, \text{ not 18!}
\]

Better: \( PE = F \cdot LE \cdot B = 18 \cdot 1 \cdot 2 = 36 \)
Select gate sizes $y$ and $z$ to minimize delay from $A$ to $B$

Logical Effort: $LE =$

Electrical Fanout: $F =$

Branching Effort: $B =$

Path Effort: $PE =$

Best Effective Fanout: $EF =$

Delay: $D =$
Example 2 with Branching

Select gate sizes \( y \) and \( z \) to minimize delay from \( A \) to \( B \)

**Logical Effort:** \( LE = (4/3)^3 \)

**Electrical Fanout:** \( F = \frac{C_{out}}{C_{in}} = 9 \)

**Branching Effort:** \( B = 2 \cdot 3 = 6 \)

**Path Effort:** \( PE = \prod LE \cdot F \cdot B = 128 \)

**Best Effective Fanout:** \( EF = \frac{PE^{1/3}}{3} \approx 5 \)

**Delay:** \( D = 3 \cdot 5 + 3 \cdot 2 = 21 \)

Work backward for sizes:

\[
\begin{align*}
\text{z} &= \frac{9C \cdot (4/3)}{5} = 2.4C \\
\text{y} &= \frac{3z \cdot (4/3)}{5} = 1.9C
\end{align*}
\]