Pseudo-nMOS Logic
( Ratioed Logic)
Introduction

• What makes a circuit fast?
  – \( I = C \frac{dV}{dt} \rightarrow t_{pd} \propto \frac{C}{I} \Delta V \)
  – 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...
Ratioed Logic

Goal: to reduce the number of devices over complementary CMOS
**Ratioed Logic**

- N transistors + Load
- \( V_{OH} = V_{DD} \)
- \( V_{OL} = \frac{R_{PN}}{R_{PN} + R_L} \)
- Assymetrical response
- Static power consumption
- \( t_{pL} = 0.69 R_L C_L \)
Active Loads

Depletion Load

$V_{DD}$

$V_T < 0$

$F$

$V_{SS}$

depletion load NMOS

PMOS Load

$V_{DD}$

$V_{SS}$

$F$

$V_{SS}$

pseudo-NMOS
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
  – Ratio issue
  – Make pMOS about \( \frac{1}{4} \) effective strength of pulldown network
Pseudo-nMOS

(a) Circuit diagram

(b) Graph showing relationship between $V_{in}$ and $V_{out}$ for different $P$ values

(c) Graph showing relationship between $V_{in}$ and $V_{out}$ for different $P$ values

(d) Graph showing relationship between $V_{in}$ and $I_{ds}$ for different $P$ values
Pseudo-NMOS VTC
# Pseudo-nMOS Design

<table>
<thead>
<tr>
<th>Size of PMOS</th>
<th>$V_{OL}$</th>
<th>Static Power Dissipation</th>
<th>$t_{PLH}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.693 V</td>
<td>564 $\mu$W</td>
<td>14 ps</td>
</tr>
<tr>
<td>2</td>
<td>0.273 V</td>
<td>298 $\mu$W</td>
<td>56 ps</td>
</tr>
<tr>
<td>1</td>
<td>0.133 V</td>
<td>160 $\mu$W</td>
<td>123 ps</td>
</tr>
<tr>
<td>0.5</td>
<td>0.064 V</td>
<td>80 $\mu$W</td>
<td>268 ps</td>
</tr>
<tr>
<td>0.25</td>
<td>0.031 V</td>
<td>41 $\mu$W</td>
<td>569 ps</td>
</tr>
</tbody>
</table>
Pseudo-nMOS Gates

- Design for unit current on output to compare with unit inverter.
- pMOS fights nMOS
- $I_{out} = 4I/3 - I/3$

Inverter

$$\begin{align*}
g_u &= \\
g_d &= \\
g_{avg} &= \\
p_u &= \\
p_d &= \\
p_{avg} &=
\end{align*}$$

NAND2

$$\begin{align*}
g_u &= \\
g_d &= \\
g_{avg} &= \\
p_u &= \\
p_d &= \\
p_{avg} &=
\end{align*}$$

NOR2

$$\begin{align*}
g_u &= \\
g_d &= \\
g_{avg} &= \\
p_u &= \\
p_d &= \\
p_{avg} &=
\end{align*}$$
Pseudo-nMOS Gates

• Design for unit current on output to compare with unit inverter.

• pMOS fights nMOS

<table>
<thead>
<tr>
<th>Inverter</th>
<th>NAND2</th>
<th>NOR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_u = \frac{4}{3} )</td>
<td>( g_u = \frac{8}{3} )</td>
<td>( g_u = \frac{4}{3} )</td>
</tr>
<tr>
<td>( g_d = \frac{4}{9} )</td>
<td>( g_d = \frac{8}{9} )</td>
<td>( g_d = \frac{4}{9} )</td>
</tr>
<tr>
<td>( g_{avg} = \frac{8}{9} )</td>
<td>( g_{avg} = \frac{16}{9} )</td>
<td>( g_{avg} = \frac{8}{9} )</td>
</tr>
<tr>
<td>( p_u = \frac{6}{3} )</td>
<td>( p_u = \frac{10}{3} )</td>
<td>( p_u = \frac{10}{3} )</td>
</tr>
<tr>
<td>( p_d = \frac{6}{9} )</td>
<td>( p_d = \frac{10}{9} )</td>
<td>( p_d = \frac{10}{9} )</td>
</tr>
<tr>
<td>( p_{avg} = \frac{12}{9} )</td>
<td>( p_{avg} = \frac{20}{9} )</td>
<td>( p_{avg} = \frac{20}{9} )</td>
</tr>
</tbody>
</table>
Pseudo-nMOS Design

- Ex: Design a k-input AND gate using pseudo-nMOS. Estimate the delay driving a fanout of H

- G = 1
- F = G
- P = 1 + (4+8k)/9 = (8k+13)/9
- N = 2
- D = NF^{1/N} + P =
Pseudo-nMOS Power

- Pseudo-nMOS draws power whenever $Y = 0$
  - Called static power $P = I_{DD}V_{DD}$
  - A few mA / gate * 1M gates would be a problem
  - Explains why nMOS went extinct

- Use pseudo-nMOS sparingly for wide NORs

- Turn off pMOS when not in use

```
A  |  B  |  C  
|   |   |   |
|   |   |   |
|   |   |   |
en
```

$\text{en}$
Ratio Example

• The chip contains a 32 word x 48 bit ROM
  – Uses pseudo-nMOS decoder and bitline pullups
  – On average, one wordline and 24 bitlines are high

• Find static power drawn by the ROM
  – $I_{on-p} = 36 \, \mu A$, $V_{DD} = 1.0 \, V$

$$P_{pull-up} = \frac{36 \, \mu A \times 24}{1.0 \, V} = 864 \, \mu W$$

• Solution:
Pseudo-NMOS Design

• Pseudo-nMOS gates will not operate correctly if $V_{OL} > V_{IL}$ of the driven gate.
• This is most likely in the SF corner.
• Conservative design requires extra weak pMOS.
• Another choice is to use replica biasing.
• Idea comes from analog design.
• Replica biasing allows $1/3$ the current ratio rather than the conservative $1/4$ ratio of earlier.
Replica Biasing

To other pseudo-nMOS gates

\[ V_{bias} \]

\[ \begin{align*}
\text{P1:} & \quad 2 \\
\text{N1:} & \quad \frac{1}{2} \\
\text{P2:} & \quad 2 \\
\text{N2:} & \quad \frac{3}{2} \\
\text{Y:} & \quad \text{A} \\
\text{\( g_u \):} & \quad 1 \\
\text{\( g_d \):} & \quad \frac{1}{2} \\
\text{\( g_{avg} \):} & \quad \frac{3}{4}
\end{align*} \]
Ganged CMOS

(a)  

(b)  

\[ g_u = 1 \]
\[ g_d = \frac{2}{3} \]
\[ g_{avg} = \frac{5}{6} \]
# Ganged CMOS

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>N1</th>
<th>P1</th>
<th>N2</th>
<th>P2</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>OFF</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>~0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
<td>~0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>0</td>
</tr>
</tbody>
</table>
Improved Loads

Adaptive Load
Improved Loads

Differential Cascode Voltage Switch Logic (DC VSL)
Improved Loads (2)

Differential Cascode Voltage Switch Logic (DCVSL)
DCVSL Example

XOR-NXOR gate
DCVSL Example

(c)
DCVSL Transient Response
Pass-Transistor Logic

- N transistors
- No static consumption
Example: AND Gate

\[ F = AB \]
NMOS-Only Logic
NMOS-Only Switch
NMOS Only Logic:
Level Restoring Transistor

- Advantage: Full Swing
- Restorer adds capacitance, takes away pull down current at X
- Ratio problem
Restorer Sizing
LEAP

- **LEA**n integration with **P**ass transistors
- Get rid of pMOS transistors
  - Use weak pMOS feedback to pull fully high
  - Ratio constraint