Static CMOS Circuits

- Conventional (ratio-less) static CMOS
  - Covered so far
- Ratio-ed logic (depletion load, pseudo nMOS)
- Pass transistor logic

Example 1

```verilog
module mux(input s, d0, d1, output y);
    assign y = s ? d1 : d0;
endmodule
```

1) Sketch a design using AND, OR, and NOT gates.
Example 1

module mux(input s, d0, d1, 
            output y);

    assign y = s ? d1 : d0;
endmodule

1) Sketch a design using AND, OR, and NOT gates.

Example 2

2) Sketch a design using NAND, NOR, and NOT gates. Assume ~S is available.
Example 2

2) Sketch a design using NAND, NOR, and NOT gates. Assume ~S is available.

![](image)

Bubble Pushing

- Start with network of AND / OR gates
- Convert to NAND / NOR + inverters
- Push bubbles around to simplify logic
  - Remember DeMorgan’s Law

(a) (b)

(c) (d)
Example 3

3) Sketch a design using one compound gate and one NOT gate. Assume \sim S is available.
Compound Gates

• Logical Effort of compound gates

Example 4

• The multiplexer has a maximum input capacitance of 16 units on each input. It must drive a load of 160 units. Estimate the delay of the NAND and compound gate designs.
NAND Solution

\[ P = 2 + 2 = 4 \]
\[ G = \left(\frac{4}{3}\right) \left(\frac{4}{3}\right) = 16/9 \]
\[ F = GBH = 160/9 \]
\[ \hat{f} = \sqrt[3]{F} = 4.2 \]
\[ D = \hat{Nf} + P = 12.4\tau \]
\[ P = 4 + 1 = 5 \]
\[ G = \frac{6}{3} \cap \{1\} = 2 \]
\[ F = GBH = 20 \]
\[ \hat{f} = \sqrt[14]{F} = 4.5 \]
\[ D = N\hat{f} + P = 14\tau \]
Example 5

- Annotate your designs with transistor sizes that achieve this delay.

Homework exercise!

Input Order

- Our parasitic delay model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest?
    - If B arrives latest?
Input Order

- Our parasitic delay model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest? $2\tau$
    - If B arrives latest? $2.33\tau$

Inner & Outer Inputs

- *Outer* input is closest to rail (B)
- *Inner* input is closest to output (A)

If input arrival time is known
  - Connect latest input to inner terminal
**Asymmetric Gates**

- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same
- \( g_A = \frac{10}{9} \)
- \( g_B = 2 \)
- \( g_{\text{total}} = g_A + g_B = \frac{28}{9} \)
- Asymmetric gate approaches \( g = 1 \) on critical input
- But total logical effort goes up

**Symmetric Gates**

- Inputs can be made perfectly symmetric
Skewed Gates

- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical nMOS transistor
    
    \[
    g_u = \frac{\frac{1}{2}}{\frac{2}{A_Y}} = \frac{5}{6}
    \]
    
    \[
    g_d = \frac{\frac{2.5}{1.5}}{A_Y} = \frac{5}{3}
    \]

- Calculate logical effort by comparing to unskewed inverter with same effective resistance on that edge.
HI- and LO-Skew

- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.

- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)

- Logical effort is smaller for favored direction
- But larger for the other direction

---

Catalog of Skewed Gates

**Inverter**

<table>
<thead>
<tr>
<th>Type</th>
<th>( g_u )</th>
<th>( g_d )</th>
<th>( g_{avg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unskewed</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>HI-skew</td>
<td>( \frac{5}{6} )</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{4} )</td>
</tr>
<tr>
<td>LO-skew</td>
<td>( \frac{4}{3} )</td>
<td>( \frac{2}{3} )</td>
<td>( \frac{1}{4} )</td>
</tr>
</tbody>
</table>

**NAND2**

<table>
<thead>
<tr>
<th>Type</th>
<th>( g_u )</th>
<th>( g_d )</th>
<th>( g_{avg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unskewed</td>
<td>( \frac{4}{3} )</td>
<td>( \frac{4}{3} )</td>
<td>( \frac{4}{3} )</td>
</tr>
<tr>
<td>HI-skew</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
</tr>
<tr>
<td>LO-skew</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
</tr>
</tbody>
</table>

**NOR2**

<table>
<thead>
<tr>
<th>Type</th>
<th>( g_u )</th>
<th>( g_d )</th>
<th>( g_{avg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unskewed</td>
<td>( \frac{4}{3} )</td>
<td>( \frac{4}{3} )</td>
<td>( \frac{4}{3} )</td>
</tr>
<tr>
<td>HI-skew</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
</tr>
<tr>
<td>LO-skew</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
<td>( \frac{5}{3} )</td>
</tr>
</tbody>
</table>
Catalog of Skewed Gates

Inverter

unskewed
- $g_u = 1$
- $g_d = 1$
- $g_{avg} = 1$

HI-skew
- $g_u = \frac{5}{6}$
- $g_d = \frac{5}{3}$
- $g_{avg} = \frac{5}{4}$

LO-skew
- $g_u = \frac{4}{3}$
- $g_d = \frac{2}{3}$
- $g_{avg} = 1$

NAND2

unskewed
- $g_u = \frac{4}{3}$
- $g_d = \frac{4}{3}$
- $g_{avg} = \frac{4}{3}$

HI-skew
- $g_u = \frac{5}{3}$
- $g_d = \frac{5}{3}$
- $g_{avg} = \frac{5}{3}$

LO-skew
- $g_u = \frac{4}{3}$
- $g_d = \frac{2}{3}$
- $g_{avg} = \frac{4}{3}$

NOR2

unskewed
- $g_u = \frac{2}{3}$
- $g_d = 1$
- $g_{avg} = \frac{3}{2}$

HI-skew
- $g_u = \frac{3}{2}$
- $g_d = 3$
- $g_{avg} = \frac{9}{4}$

LO-skew
- $g_u = \frac{2}{3}$
- $g_d = \frac{1}{1}$
- $g_{avg} = \frac{3}{2}$
Asymmetric Skew

- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input

![Asymmetric Skew Diagram]

Best P/N Ratio

- We have selected P/N ratio for unit rise and fall resistance ($\mu = 2-3$ for an inverter).
- Alternative: choose ratio for least average delay
- Ex: inverter
  - Delay driving identical inverter
    - $t_{pdr}$ =
    - $t_{pd}$ =
    - $t_{pdf}$ =
  - Differentiate $t_{pd}$ w.r.t. $P$
  - Least delay for $P =$
Best P/N Ratio

- We have selected P/N ratio for unit rise and fall resistance ($\mu = 2-3$ for an inverter).
- Alternative: choose ratio for least average delay
- Ex: inverter
  - Delay driving identical inverter
  - $t_{pdr} = (P+1)$
  - $t_{pd} = (P+1)(\mu/P)$
  - $t_{pd} = (P+1)(1+\mu/P)/2 = (P + 1 + \mu + \mu/P)/2$
  - Differentiate $t_{pd}$ w.r.t. $P$
  - Least delay for $P = \sqrt{\mu}$

P/N Ratios

- In general, best P/N ratio is sqrt of that giving equal delay.
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power

<table>
<thead>
<tr>
<th>Inverter</th>
<th>NAND2</th>
<th>NOR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_u = 2$</td>
<td>$g_u = 4/3$</td>
<td>$g_u = 2$</td>
</tr>
<tr>
<td>$g_d = 1$</td>
<td>$g_d = 4/3$</td>
<td>$g_d = 1$</td>
</tr>
<tr>
<td>$g_{avg} = 3/2$</td>
<td>$g_{avg} = 4/3$</td>
<td>$g_{avg} = 3/2$</td>
</tr>
</tbody>
</table>
Observations

- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages

Combinational vs. Sequential Logic

(a) Combinational
\[ \text{Output} = f(\text{In}) \]

(b) Sequential
\[ \text{Output} = f(\text{In}, \text{Previous In}) \]
At every point in time (except during the switching transients) each gate output is connected to either $V_{DD}$ or $V_{SS}$ via a low-resistive path.

The outputs of the gates assume at all times the value of the Boolean function, implemented by the circuit (ignoring, once again, the transient effects during switching periods).

This is in contrast to the *dynamic* circuit class, which relies on temporary storage of signal values on the capacitance of high impedance circuit nodes.

**Static CMOS Circuit (Review)**

![Diagram of Static CMOS Circuit](image)

- **PMOS Only**
  - $F = \overline{G}$
- **NMOS Only**
- **PUN and PDN are Dual Networks**
Properties of Complementary CMOS Gates (Review)

High noise margins:
$V_{OH}$ and $V_{OL}$ are at $V_{DD}$ and $GND$, respectively.

No static power consumption:
There never exists a direct path between $V_{DD}$ and $V_{SS}$ ($GND$) in steady-state mode.

Comparable rise and fall times:
(under the appropriate scaling conditions)

Influence of Fan-In and Fan-Out on Delay

$\text{Fan-Out: Number of Gates Connected}$
Every fanout (output) adds two gate capacitances (pMOS and nMOS)

$\text{FanIn: Quadratic Term due to:}$
1. Resistance Increasing
2. Capacitance Increasing

$t_p = a_1 F_I + a_2 F_I^2 + a_3 F_O$
Fast Complex Gate-Design Techniques

- **Transistor Sizing:**
  As long as Fan-out Capacitance dominates

- **Progressive Sizing:**

\[
\begin{array}{c}
\text{In}_N \\
\text{In}_3 \\
\text{In}_2 \\
\text{In}_1 \\
\text{Out}
\end{array}
\]

\[\begin{align*}
C_1 &> C_2 & C_3 &> C_L \\
M_1 &> M_2 &> M_3 &> M_N
\end{align*}\]

Fast Complex Gate - Design Techniques

- **Transistor Ordering**

\[
\begin{array}{c}
\text{In}_3 \\
\text{In}_2 \\
\text{In}_1 \\
\text{critical path}
\end{array}
\]

\[
\begin{array}{c}
\text{In}_3 \\
\text{In}_2 \\
\text{In}_1 \\
\text{critical path}
\end{array}
\]

(a) (b)
Fast Complex Gate - Design Techniques

- Improved Logic Design

![Improved Logic Design Diagram]

Ratioed Logic

(a) resistive load  
(b) depletion load NMOS  
(c) pseudo-NMOS

Goal: to reduce the number of devices over complementary CMOS

Careful design needed!
**Ratioed Logic**

- $V_{OH} = V_{DD}$
- $V_{OL} = \frac{R_{PDN}}{R_L + R_{PDN}} V_{DD}$

Desired: $R_L \gg R_{PDN}$ (to keep noise margin low)

$\tau_{PLH} = 0.69R_L C_L$

Problems:
1) Static power dissipation
2) Difficult to implement a large resistor, eg 40kΩ resistor (typical value) needs 3200 $\mu^2$ of n-diff, i.e. 1,000 transistors!

---

**Active Loads**

- Depletion-mode transistor has negative threshold
- On if $V_{GS} = 0$
- Body effect may be a problem!
**Pseudo-nMOS**

- No problems due to body effect
- N-input gate requires only N+1 transistors
- Each input connects to only a single transistor, presenting smaller load to preceding gate
- Static power dissipation (when output is zero)
- Asymmetric rise and fall times

Example: Suppose minimal-sized gate consumes 1 mW of static power. 100,000 gate-circuit: 50 W of static power (plus dynamic power)! (half the gates are in low-output state)

- Effective only for small subcircuits where speed is important, e.g., address decoders in memories

---

**Pseudo-NMOS NAND Gate**

\[ C_{L,\text{pseudo}} = 0.5 \times C_{L,\text{CMOS}} \text{ (Fan-out of 1)} \]
Pass-Transistor Logic

Inputs

Out

A

B

Is this transmission gates necessary?

AND gate

Need a low impedance path to ground when B = 0

- No static consumption

Pass-Transistor Based Multiplexer

\[ \overline{F} = AS + BS \]

\[ V_{DD} \]

\[ V_{DD} \]

Out F

In1 S

In2 S

GND

ECE 261 Krish Chakrabarty

ECE 261 Krish Chakrabarty
Transmission Gate XOR

6 transistors only!

Case 1:
B = 1, M3/M4 turned off, F = \overline{AB}

Case 2:
B = 0, M3/M4 turned on, F = AB

F always has a path to V_{DD} or Gnd, hence low impedance node.
If not, node would be dynamic, requiring refresh due to charge leakage.

Delay in Transmission Gate Networks

Insert buffers after every m switches.
Delay in Transmission Gate Networks

Consider Kirchoff’s Law at node $V_i$

$$\frac{V_{i+1} - V_i}{R_{eq}} + \frac{V_{i-1} - V_i}{R_{eq}} = C \frac{dV_i}{dt}$$

Therefore,

$$\frac{dV_i}{dt} = \frac{V_{i+1} + V_{i-1} - 2V_i}{R_{eq}C}$$

Propagation delay can be determined using Elmore delay analysis.

---

Delay Optimization

- **Delay of RC chain**

  $$t_p = 0.69 \sum_{k=0}^{n} CR_{eq}^k = 0.69CR_{eq} \frac{n(n+1)}{2}$$

  Delay can be reduced by adding buffers after $m$ stages

  $$t_{buf} = \text{delay of a buffer}$$

- **Delay of Buffered Chain**

  $$t_p = \left(0.69 \left[CR_{eq} \frac{n(n+1)}{2} + \left(\frac{n}{m} - 1\right) t_{buf}\right] \right) + \left(0.69 \left[CR_{eq} \frac{m(m+1)}{2} + \left(\frac{n}{m} - 1\right) t_{buf}\right] \right)$$

  $$m_{opt} = 1.7 \sqrt{\frac{t_{pbuf}}{CR_{eq}}}$$
Transmission Gate Full Adder

NMOS Only Logic: Level Restoring Transistor

- Advantage: Full Swing
- Disadvantage: More Complex, Larger Capacitance
- Other approaches: reduced threshold NMOS
Single Transistor Pass Gate with $V_T=0$

```
VDD

0V   5V

VDD

5V

Out

WATCH OUT FOR LEAKAGE CURRENTS
```

Complimentary Pass Transistor Logic

```
(a) Pass-Transistor Network

F = A ∨ B

(b) Inverse Pass-Transistor Network

F = A ⊕ B

AND/NAND

OR/NOR

EXOR/NEXOR
```
4 Input NAND in CPL