Lecture 10
Logic Gate Families and Layout

• XOR gates
• Critical Paths
• Logic Layouts
• Transmission Gate Layouts
• Clocked CMOS Logic
• Pass Transistor Logic
• Summary
Material from: *Principles of CMOS VLSI Design*

By Neil E. Weste and Kamran Eshraghian

&

Amar Mukherjee; *Introduction to nMOS & VLSI System Design*; Prentice-Hall, 1986
nMOS Gates

- nMOS NOR gate

- nMOS NAND gate
**XOR Gates**

- **nMOS XOR gate:**

- **CMOS XOR gate:**

- Complement \( A \), \( B \), or \( OUT \) to get an equivalence gate.
Causes of Incorrect Gate Operation

- Insufficient or noisy power lines
- Gate input noise
- Faulty transistors
- Faulty transistor connections
- Incorrect transistor ratios
- Charge sharing or bad clocking of dynamic gates
- Ratioed and dynamic gates not used in *application specific integrated circuits* (ASIC’s)
  - *Used in microprocessors*
Critical Paths

- Slowest timing paths that limit a chip’s speed
- Affect critical paths at these levels:
  - Architecture
  - RTL/Logic gate levels
  - Circuit level
  - Layout level
- RTL/logic level – pipelining, gate types, fanin & fanouts
- Circuit level – resize transistors to speed up
- Layout level – speed up by changing physical layout
- False paths – appear to be critical paths, but cannot propagate transitions due to Boolean value conflicts
New Gate Notation

- Bubble on transistor closest to gate output
- *Stage ratio* – increase in transistor size in successive logic stages
- Use gates with # series inputs of 2-5 for best speed
Example

fan-out = 4
Worst-Case Rise Delay for m-Input NAND

\[ t_{dr} = \frac{R_p}{n} (mnC_d + C_r + kC_g) \]  

(5.2a)

where

- \( R_p \) = the effective resistance of p-device in a minimum-sized inverter
- \( n \) = width multiplier for p-devices in this gate
- \( k \) = the fan-out (number of inputs connected to gate output, say, in units of minimum-sized inverters)
- \( m \) = fan-in of gate
- \( C_g \) = gate capacitance of a minimum-sized inverter
- \( C_d \) = source/drain capacitance of a minimum-sized inverter (Note: the p and n contributions are added as an approximation (worst case) to model the effect of internal diffusion regions loading the gate as well as diffusions connected to the output)
- \( C_r \) = routing capacitance.
Reformulation

\[ t_{dr} = \frac{R_p}{n} (mnC_g + q(k)C_g + kC_g) \]

\[ = \frac{R_p C_g}{n} (mn + q(k) + k) \]

\[ = R_p C_g mr + \frac{R_p C_g}{n} q(k) + \frac{R_p C_g}{n} k, \]

(5.2b)

where

\[ r = \frac{C_d}{C_g}, \] the ratio of the intrinsic drain capacitance of an inverter to the gate capacitance,

and

\[ q(k) = \text{a function of the fan-out representing the routing capacitance as a multiplier times the gate capacitance.} \]
New Delay Model

- $t_{dr} = t_{internal-r} + k t_{output-r}$
- $t_{internal-r} = R_p C_g m r$
- $t_{output-r} = R_p C_g \left( \frac{1 + q(k)}{n} \right) \left( \frac{k}{k} \right)$
- $t_f$ equation is similar
- Assume equal-sized $n$ & $p$ transistors
Different Delay Model

- $t_r = t_f$
- NAND gate: $R_p = m R_n$
- $W_p = \frac{W_n}{m}$
- NOR gate: $R_n = m R_p$
- $W_n = \frac{W_p}{m}$
Delay of 1-3 Input Gates
Delay of 4-8 Input Gates

[Diagrams showing the delay of 4-8 input gates with corresponding graphs and data points for different capacitive loads.]
## NAND/NOR Delays Measured with SPICE

<table>
<thead>
<tr>
<th>GATE</th>
<th>$t_{\text{internal-f}}$ (ns)</th>
<th>$t_{\text{output-f}}$ (ns/pF)</th>
<th>$t_{\text{internal-r}}$ (ns)</th>
<th>$t_{\text{output-r}}$ (ns/pF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>.08</td>
<td>1.7</td>
<td>.08</td>
<td>2.1</td>
</tr>
<tr>
<td>ND2</td>
<td>.2</td>
<td>3.1</td>
<td>.15</td>
<td>2.1</td>
</tr>
<tr>
<td>ND3</td>
<td>.41</td>
<td>4.4</td>
<td>.2</td>
<td>2.1</td>
</tr>
<tr>
<td>ND4</td>
<td>.68</td>
<td>5.7</td>
<td>.25</td>
<td>2.1</td>
</tr>
<tr>
<td>ND8</td>
<td>2.44</td>
<td>10.98</td>
<td>.38</td>
<td>2.2</td>
</tr>
<tr>
<td>NR2</td>
<td>.135</td>
<td>1.75</td>
<td>.25</td>
<td>4.1</td>
</tr>
<tr>
<td>NR3</td>
<td>.14</td>
<td>1.83</td>
<td>.52</td>
<td>6.2</td>
</tr>
<tr>
<td>NR4</td>
<td>.145</td>
<td>1.88</td>
<td>.9</td>
<td>8.2</td>
</tr>
<tr>
<td>NR8</td>
<td>.19</td>
<td>1.8</td>
<td>3.35</td>
<td>16.4</td>
</tr>
</tbody>
</table>
**Effective Channel Resistances**

**TABLE 5.2  Effective Resistance Values for a Typical 1\(\mu\) CMOS Process (m = 1 – 4)**

<table>
<thead>
<tr>
<th>GATE</th>
<th>(R_n (\Omega))</th>
<th>(R_p (\Omega))</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>7.1K</td>
<td>8.5K</td>
</tr>
<tr>
<td>ND2</td>
<td>6.3K</td>
<td>8.6K</td>
</tr>
<tr>
<td>ND3</td>
<td>6.0K</td>
<td>8.7K</td>
</tr>
<tr>
<td>ND4</td>
<td>5.9K</td>
<td>8.8K</td>
</tr>
<tr>
<td>NR2</td>
<td>7.3K</td>
<td>8.4K</td>
</tr>
</tbody>
</table>
Choice of Fastest Gate Implementation
# Comparison Results for Different 8-Input AND Gates

<table>
<thead>
<tr>
<th>APPROACH</th>
<th>DELAY STAGE 1 (ns)</th>
<th>DELAY STAGE 2 (ns)</th>
<th>DELAY STAGE 3 (ns)</th>
<th>DELAY STAGE 4 (ns)</th>
<th>TOTAL DELAY (SPICE) (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.82</td>
<td>3.37</td>
<td>2.17</td>
<td>3.19</td>
<td>6.2</td>
</tr>
<tr>
<td>ND8-&gt;</td>
<td>ND8 falling</td>
<td>INV rising</td>
<td>ND2 falling</td>
<td>ND2 falling</td>
<td>3.46</td>
</tr>
<tr>
<td>INV</td>
<td>INV rising</td>
<td>ND2 falling</td>
<td>ND2 falling</td>
<td>INV rising</td>
<td>6.5</td>
</tr>
<tr>
<td>2</td>
<td>.88</td>
<td>4.36</td>
<td>2.17</td>
<td>3.19</td>
<td>5.24</td>
</tr>
<tr>
<td>ND4-&gt;</td>
<td>ND4 falling</td>
<td>NR2 rising</td>
<td>ND2 falling</td>
<td>ND2 falling</td>
<td>3.46</td>
</tr>
<tr>
<td>NR2</td>
<td>INV rising</td>
<td>ND2 falling</td>
<td>ND2 falling</td>
<td>INV rising</td>
<td>5.26</td>
</tr>
<tr>
<td>3</td>
<td>.31</td>
<td>.4</td>
<td>.31</td>
<td>2.17</td>
<td>3.19</td>
</tr>
</tbody>
</table>
Logic Design Guidelines

- Avoid long resistive charging / discharging paths in CMOS
- Use NAND’s wherever possible
- Place inverters at high fanout nodes
- Avoid NOR’s in high-speed circuits
  - With \( \text{fanin} > 4 \) and large \( \text{fanout} \)
- Keep fanout below 5-10
- Use minimum-sized gates on high fanout nodes to keep \( C_L \) down
- Keep rising (falling) edges sharp
- When power or area is a constraint, use large fanin static gates
CMOS Inverter Layouts
High Drive Inverters

(a)
NAND Gate Layout Guidelines

- Use metal3 or highest metal layer for power and ground
- Place transistors in complex gates:
  - Horizontal channels in diffusion
  - Vertical poly gates
  - Use “line of diffusion” rule for gate layout
NAND Gate Layouts
NOR Gate Layouts
Complex Logic Gate Layout

![Diagram of a complex logic gate layout with labeled nodes A, B, C, D, and V_DD, V_SS.]
XNOR Layouts
Standard (Sandia) Cells

- Fixed height (pitch)
- Width varies with gate function
- Automatically placed in rows and wires routed by Cadence Envisia tool
Example Standard Cell
More Standard Cell Layouts
More Standard Cell Layouts
Gate Array (Sea of Gates) Layout
Sea of Gates Layout
Gate Array Layouts

- Isolate gates from each other with vertical poly strips tied to $V_{SS}$ ($V_{DD}$)
**Full-Custom Density Improvement**

1. Route wires over cells
2. Use merged source/drain connections
3. Use white space in sparse gates
4. Use smaller, optimal device sizes
Graded Transistors

- Speed up old technologies 15-30%
- Only speed up deep submicron devices 2-4%
Reducing $C_L$

- Use better physical design
Inferior Gate Layout
Superior Gate Layout

- Less $C$ on output node
Transmission Gate Layouts
Two-Input MUX
Two-Input MUX
CMOS Reduced Supply Voltage

• Can reduce 5V $V_{DD}$ to 2.5V or lower for static CMOS
  – *TSMC 0.35 $\mu$m process requires* $V_{DD} = 3.3$ V
  – *Reduces power usage – main power saving method*
  – *Transistor still works because of subthreshold conduction*

• Requires an on-chip voltage regulator circuit to drop the supply
Voltage Regulator Circuit
Clocked CMOS Dynamic Logic

$C^2$MOS – Used to:

- Interface with dynamic CMOS logic
- Same input as static CMOS – longer $t_r$ & $t_f$
- Faster if clocking transistors at center
- Better for hot electron problems if clock transistor is next to ground
  - $CLOCK$ is last input to change
  - However, this defeats the isolation purpose of $C^2$MOS
Clocked CMOS Logic
Pass Transistor Logic

- Design with Variable-Entered Karnaugh Map
Example – XNOR Gate Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(A \oplus B)</th>
<th>PASS FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(-A + -B)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(A + -B)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(-A + B)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(A + B)</td>
</tr>
</tbody>
</table>
Corresponding Variable-Entered
Karnaugh Map

\[ F = -A(-B) + A(B) \]
Pass Transistor Logic

- Marvelous for low-power design

---

November 5, 2002
VLSI Design - I D. Misra Week-10
49
**Example Pass Transistor Logic**

- Dynamic pass transistor logic with optional latching
  - *Timing like* RC *delay line* – square law
  - *Good for low-power, high-performance logic*
nMOS Implementation

Diagram showing a circuit with inputs A, -A, B, and -B, and outputs F(A B). The circuit involves four transistors labeled P₁, P₂, P₃, and P₄.
**Principles of Pass Transistor Logic**

- **Normally OPEN** Closes when $X = 1$

- **Normally CLOSED** Opens when $X = 1$

- $F = X + \overline{Y}Z$
  - When IN = “1” and all paths are closed $F = 1$
  - $F$ is assumed to be “0” if all paths are OPEN i.e. $\overline{F}$
Principles (continues)

\[ F = \overline{X} (Y + \overline{Z}) \]

Amar Mukherjee; Introduction to nMOS & VLSI System Design; Prentice-Hall, 1986
Stick Diagram nMOS Implementation

\[ f = X + \overline{Y}Z \]
Vertical Metal Lines

\[ f = X + \overline{Y}Z \]
Shannon’s Expansion Theorem

\[ f = x + \overline{y}z \]
CMOS Implementation
Improved CMOS Layout
Static or Latched Version
Summary

- XOR gates
- Critical Paths
- Logic Layouts
- Transmission Gate Layouts
- Clocked CMOS Logic
- Pass Transistor Logic
- Summary
Dynamic CMOS Gate

![Diagram of Dynamic CMOS Gate](image-url)
Dynamic CMOS Gates

\[ Z = \overline{A(B + C)} + (D.E) \quad \text{clk} = 1 \]
\[ Z = \text{HIGH} \quad \text{clk} = 0 \]

Clock phases:
- **Precharge**
- **Evaluate**

Clock signal transitions:
- 0 to 1
- 1 to 0
**Dynamic CMOS Logic**

- *clk* is single-phase clock
- Input is half that of static CMOS
- Pullup time greatly improved by active precharge transistor
- Pulldown time slightly increased by evaluate transistor
- Eliminate evaluation transistor if inputs guaranteed to be 0 during precharge

**Problems:**
- Inputs can change only during precharge – MUST be stable during evaluation
  - Otherwise, charge redistribution gives wrong logic value
- Charge sharing at internal dynamic gate nodes can cause wrong logic value
Dynamic CMOS Logic

- *n*-block dynamic gate can only drive *p*-block dynamic gates
  - *For n*-block to drive *n*-block, *must have static inverter in between*

- **Advantages:**
  - *Uses far less area than static CMOS* – only 1 *p* transistor and simpler wiring
  - *Faster than static CMOS* – precharge period can be made very short by widening precharger to give it very high $\beta$
  - *Each gate input has only 1 transistor load rather than 2 as in static CMOS* – faster switching
  - *Easier to test than static CMOS* – less likely to have faults turning combinational logic gates into sequential circuits
Erroneous Dynamic Gate Evaluation

- Never connect gates this way!
- Signal at $N_1$ changes during 2\textsuperscript{nd} dynamic gate’s evaluation – erroneously discharges the gate
Erroneous Timing Behavior