12. lecture

- Logic gates

Switches in Series

Truth Table (OFF/ON=0/1)

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>PATH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Function = ??

Switches in Parallel

Truth Table

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>PATH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>NO</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>YES</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>YES</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>YES</td>
</tr>
</tbody>
</table>

Function = Logic AND
Switches in Parallel

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>PATH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Function = ??

Switches in Parallel

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>PATH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Function = ??

Switches in Parallel

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>PATH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Function = Logic OR

CMOS Transistor

- Complementary MOS
  - P-channel MOS (pMOS)
  - N-channel MOS (nMOS)

- pMOS
  - P-type source and drain diffusions
  - N substrate
  - Mobility by holes

- nMOS
  - N-type source and drain diffusions
  - P substrate
  - Mobility by electrons

Pass Transistor using NMOS

- Assume capacitor \( C_L \) is initially discharged
- Gate=1, Vin=1
  - \( C_L \) begins to conduct and charges toward \( V_{dd} \) and stops at \( V_{dd}-V_t \)
  - Signal is degraded

- Gate=1, Vin=0
  - \( C_L \) begins to discharge toward 0
Transmission Degradation using Pass Transistor

\[ V_{dd} - V_t \]

\[ V_{dd} - 2V_t \]

\[ V_{dd} = V_{dd} - N*V_t \]

Still 1??

CMOS Signal Transfer Property

<table>
<thead>
<tr>
<th>Gate</th>
<th>Path</th>
<th>Transmits 1 well</th>
<th>Transmits 0 poorly</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Open</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CMOS Transmission Gate

- Transmit signal from INPUT to OUTPUT when Gate is closed

<table>
<thead>
<tr>
<th>Gate (complementary of Gate)</th>
<th>pMOS</th>
<th>nMOS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
<td>OFF</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
<td>ON</td>
<td>INPUT</td>
</tr>
</tbody>
</table>

Z : High-Impedance State, consider the terminal is "floating"

High Impedance

- When a path exists
  - Impedance is low to allow ample flow of current

- When no path
  - Impedance is high allowing almost no current flow between two terminals

Transmission Gates

Transmit Logic 0

Transmit Logic 1

Transmission Gate Symbol
**CMOS Inverter**

- Connect the following terminals of a PMOS and an NMOS
  - Gates
  - Drains

**I-V Characteristics**

- Make pMOS is wider than nMOS such that $\beta_n = \beta_p$

**Current vs. $V_{out}, V_{in}$**

- For a given $V_{in}$:
  - Plot $I_{ds}$ vs. $V_{out}$
  - $V_{out}$ must be where currents are equal in

**Load Line Analysis**

- $V_{in} = 0$
- $V_{in} = 0.2 V_{DD}$
Load Line Analysis

- $V_{in} = 0.4\ V_{DD}$

- $V_{in} = 0.6\ V_{DD}$

- $V_{in} = 0.8\ V_{DD}$

- $V_{in} = V_{DD}$

Load Line Summary

CMOS Inverter – Load line
DC Transfer Curve

- Transcribe points onto $V_{in}$ vs. $V_{out}$ plot

Operating Regions

- Revisit transistor operating regions

<table>
<thead>
<tr>
<th>Region</th>
<th>nMOS</th>
<th>pMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Cutoff</td>
<td>Linear</td>
</tr>
<tr>
<td>B</td>
<td>Saturation</td>
<td>Linear</td>
</tr>
<tr>
<td>C</td>
<td>Saturation</td>
<td>Saturation</td>
</tr>
<tr>
<td>D</td>
<td>Linear</td>
<td>Saturation</td>
</tr>
<tr>
<td>E</td>
<td>Linear</td>
<td>Cutoff</td>
</tr>
</tbody>
</table>

Noise Margins

- How much noise can a gate input see before it does not recognize the input?

Logic Levels

- To maximize noise margins, select logic levels at unity gain point of DC transfer characteristic

CMOS Invertor – Dynamic characteristic

- $S$

CMOS Invertor: Time response

\[ I_{PhL} = \frac{1}{0.69 R \cdot C_L} \]
Inverter Time Response

- Ex: find time response of inverter driving load capacitor

\[ V_{in}(t) = u(t-t_0)V_{DD} \]
\[ V_{out}(t < t_0) = V_{DD} \]
\[ \frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}} \]

\[ I_{dsn}(t) = \begin{cases} \frac{1}{2}(V_{in} - V_{out}(t))^2 & \text{if } t \leq t_0 \\ \frac{1}{2}(V_{in} - V_{out}(t))^2 & \text{if } t > t_0 \end{cases} \]

Dynamic losses
- Charging and discharging of parasitic capacities
- The short circuit current
  - Current flowing between the power supply and the ground during the switching
- Leakage current (Leakage)
  - Reverse current of PN junctions

Power losses of CMOS gates? Where the energy is lost?

Reverse leakage currents of PN junctions

- Dynamic losses
- Charging and discharging of parasitic capacitances
- The short circuit current
  - Current flowing between the power supply and the ground during the switching
- Leakage current (Leakage)
  - Reverse current of PN junctions

Reverse leakage currents of PN junctions

\[ I_{dsn} = J_s \times A \]

\[ J_s = 10-100 \, \text{pA/\mu m}^2 \, \text{by 25 \, \degree C (for 0.25\mu m CMOS)} \]
\[ J_s \, \text{doubling every 9 degrees!} \]
How CMOS logic gates are designed?

Rules for the design

CMOS network consists of a Pull-Up Network (PUN) and a Pull-Down Network (PDN)

PUN consists of a set of PMOS transistors

PDN consists of a set of NMOS transistors

PUN and PDN implementations are complimentary to each other

PMOS ↔ NMOS

Series topology ↔ Parallel topology

PUN/PDN of a CMOS Inverter

Gate Symbol of a CMOS Inverter

PUN/PDN of a NAND Gate

PUN/PDN of a NAND Gate
**NAND Gate Symbol**

Truth Table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ C = A \cdot B \]

**PUN/PDN of a NOR Gate**

<table>
<thead>
<tr>
<th>Pull-Up Network</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pull-Down Network</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**PUN/PDN of a NOR Gate**

<table>
<thead>
<tr>
<th>Pull-Up Network</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pull-Down Network</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**NOR Gate Symbol**

Truth Table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ C = A + B \]

**How about an AND gate**

<table>
<thead>
<tr>
<th>Vdd</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**An OR Gate**

<table>
<thead>
<tr>
<th>Vdd</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[ C = A + B \]
What's the Function of the following CMOS Network?

A B C
0 0 0
0 1 1
1 0 0
1 1 0

Function = XOR

Yet Another XOR CMOS Network

A B C
0 0 2
0 1 1
1 0 1
1 1 2

Function = XOR

Exclusive-OR (XOR) Gate

A B C
0 0 0
0 1 1
1 0 1
1 1 0

C = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B

How about XNOR Gate

Truth Table
A B C
0 0 1
0 1 0
1 0 0
1 1 1

C = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B

How do we draw the corresponding CMOS network given a Boolean equation?

A Systematic Method (I)
Start from Pull-Up Network

- Each variable in the given Boolean eqn corresponds to a PMOS transistor in PUN and an NMOS transistor in PDN
- Draw PUN using PMOS based on the Boolean eqn
  - AND operation drawn in series
  - OR operation drawn in parallel
- Invert each variable of the Boolean eqn as the gate input for each PMOS in the PUN
- Draw PDN using NMOS in complementary form
  - Parallel (PUN) to series (PDN)
  - Series (PUN) to parallel (PDN)
- Label with the same inputs of PUN
- Label the output
A Systematic Method (II)
Start from Pull-Down Network

- Each variable in the given Boolean eqn corresponds to a PMOS transistor in PUN and an NMOS transistor in PDN
- Invert the Boolean eqn
- With the Right-Hand Side of the newly inverted equation, Draw PDN using NMOS
  - AND operation drawn in series
  - OR operation drawn in parallel
- Label each variable of the Boolean eqn as the gate input for each NMOS in the PDN
- Draw PUN using PMOS in complementary form
  - Parallel (PUN) to series (PDN)
  - Series (PUN) to parallel (PDN)
- Label with the same inputs of PUN
- Label the output

Example 1 (Method I)

<table>
<thead>
<tr>
<th>In parallel</th>
<th>In series</th>
</tr>
</thead>
<tbody>
<tr>
<td>F = \overline{A} \cdot \overline{C} + B</td>
<td>Vdd</td>
</tr>
</tbody>
</table>

1) Draw the Pull-Up Network
2) Assign the complemented input
3) Draw the Pull-Down Network in the complementary form

Systematic Approaches

- Note that both methods lead to exactly the same implementation of a CMOS network
- The reason to invert Output equation in (II) is because
  - Output (F) is conducting to "ground", i.e. 0, when there is a path formed by input NMOS transistors
  - Inversion will force the desired result from the equation
- Example
  - F = \overline{A} \cdot \overline{C} + B: When (A=0 and C=1) or B=1, F=1. However, in the PDN (NMOS) of a CMOS network, F=0, i.e. an inverse result.
  - Revisit how a NAND CMOS network is implemented
- Inverting each PMOS input in (I) follow the same reasoning

Example 1 (Method I)

<table>
<thead>
<tr>
<th>In parallel</th>
<th>In series</th>
</tr>
</thead>
<tbody>
<tr>
<td>F = \overline{A} \cdot \overline{C} + B</td>
<td>Vdd</td>
</tr>
</tbody>
</table>

1) Draw the Pull-Up Network
2) Assign the complemented input
3) Draw the Pull-Down Network in the complementary form
**Example 1 (Method I)**

\[ F = A \cdot C + B \]

- **In series**
  - \( F = A \cdot C + B \)
- **In parallel**
  - \( F = A \cdot C + B \)

**In series**

\[
\begin{array}{cccc}
A & B & C & F \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]

**In parallel**

\[
\begin{array}{cccc}
A & B & C & F \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]

Label the output \( F \)

---

**Drawing the Schematic using Method II**

\[ F = A \cdot \overline{C} + B \]

\[ \overline{F} = A \cdot \overline{C} + B \]

\[ \overline{F} = A \cdot \overline{C} \cdot B \]

\[ F = (A + C) \cdot \overline{B} \]

---

**An Alternative for XNOR Gate (Method I)**

\[ C = \overline{A} \cdot \overline{B} + A \cdot B \]

**Truth Table**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**Example 3**

\[ F = \overline{A} \cdot D + \overline{B} \cdot (A + \overline{C}) \]

Start from the innermost term

\[
\begin{array}{cccc}
A & B & C & F \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]

---

**Example 3**

\[ F = \overline{A} \cdot D + \overline{B} \cdot (A + \overline{C}) \]

Start from the innermost term

\[
\begin{array}{cccc}
A & B & C & F \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\]
Example 3

\[ F = \overline{A} \cdot D + \overline{B} \cdot (A + \overline{C}) \]

Start from the innermost term

Example 4

\[ F = (E + \overline{D}) \cdot (\overline{A} \cdot D + \overline{B} \cdot (A + \overline{C})) \]

Start from the innermost term

Modely pro Zpoždění hradel

A\overline{NAND2} \leftrightarrow \overline{INV} \leftrightarrow \overline{NOR2}

Transistor sizing

Transistor sizing – complex logic

\[ \text{OUT} = D + A \cdot (B + C) \]
Logic Gates layout – standard cells

NAND standard cell

Combinational vs. sequential logic

Combinational logic circuits

Timing Diagram of an AND Gate (Output=AB)

- Static CMOS logic
  - Conventional static CMOS logic
  - Switching Transistors / transmission gate
- Dynamic of CMOS logic
  - Domino logic
  - np-of CMOS

Note that the Output change can occur "at any Time" for Combinational logic.
**Combinational Logic**

- Outputs, "at any time", are determined by the input combination
- We will discuss
  - Multiplexers / De-Multiplexers
  - Decoders / Encoders
  - Comparators
  - Parity Checkers / Generators
  - Binary Adders / Subtractors
  - Integer Multipliers

**Multiplexers (Mux)**

- **Functionality:** Selection of a particular input
- **Route 1 of N inputs** (A) to the output F
- **Require** selection bits (S)
- **En(able) bit** can disable the route and set F to 0

**Multiplexers (Mux) w/out Enable**

\[
F = \overline{S_1}S_0A_0 + \overline{S_1}S_0A_1 + S_1\overline{S_0}A_2 + S_1S_0A_3
\]
Logic Diagram of a 4-to-1 Mux

\[ F = S_1 S_0 A_0 + S_1 S_0 A_1 + S_0 S_0 A_2 + S_0 S_0 A_3 \]

4-to-1 Mux using Transmission Gates

4-to-1 Mux using Transmission Gates

4-to-1 Mux using Transmission Gates

4-to-1 Mux using Transmission Gates
4-to-1 Mux using Transmission Gates with Enable (F=0 when En=0)

<table>
<thead>
<tr>
<th>En</th>
<th>S1</th>
<th>S0</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>A0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>A1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>A2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A3</td>
</tr>
</tbody>
</table>