**Representation of Logic Values by Voltage Levels**

(Positive logic system)

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Logic value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD}$</td>
<td>1</td>
</tr>
<tr>
<td>$V_{I,min}$</td>
<td>Undefined</td>
</tr>
<tr>
<td>$V_{0,max}$</td>
<td>Logic value 0</td>
</tr>
<tr>
<td>$V_{SS}$ (Gnd)</td>
<td></td>
</tr>
</tbody>
</table>

**NMOS Transistor as a Switch**

- MOSFET: Metal Oxide Semiconductor Field-Effect Transistor
- NMOS: n-channel MOSFET

$x = \text{"low"}$

(a) A simple switch controlled by the input $x$

$x = \text{"high"}$

(b) NMOS transistor

(c) Simplified symbol for an NMOS transistor

**PMOS Transistor as a Switch**

- PMOS: p-channel MOSFET

$x = \text{"high"}$

(a) A switch with the opposite behavior of previous slide

$x = \text{"low"}$

(b) PMOS transistor

(c) Simplified symbol for an PMOS transistor

**Structure of a CMOS Circuit**

- CMOS: Complementary MOS Technology

**CMOS Realization of a NOT Gate**

(a) Circuit

(b) Truth table and transistor states

<table>
<thead>
<tr>
<th>x</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>off</td>
<td>on</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>on</td>
<td>off</td>
<td>0</td>
</tr>
</tbody>
</table>

**CMOS Realization of a NOR Gate**

(a) Circuit

(b) Truth table and transistor states

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>0</td>
</tr>
</tbody>
</table>
CMOS Realization of a NAND Gate

(a) Circuit

(b) Truth table and transistor states

Implement Compound Function Directly

- All variables need to appear in their complemented form.
- PUN can be derived by looking at \( f \) directly.
- PDN can be derived by looking at \( f' \).

Example:

\[
\begin{align*}
f &= x_1' + x_2' x_3' \\
f' &= x_1 (x_2 + x_3)
\end{align*}
\]
Types of Integrated Circuits (ICs)

- Standard Chips
- Programmable Logic Devices (PLDs):
  - Programmable Logic Array (PLA)
  - Programmable Array Logic (PAL)
  - Complex Programmable Logic Device (CPLD)
  - Field-Programmable Gate Array (FPGA)
- Non-Programmable Devices:
  - Custom Design
  - Standard-Cell Design
    - Application Specific Integrated Circuit (ASIC)
  - Gate-Array Design

Standard Chips

- A collection of specific gates in a chip
- Popular until mid-80s
- 7400-Series Standard Chips
  - 7404: NOT gates
  - 7408: AND gates
  - 7432: OR gates
  - 74244: Tri-State Buffers

Implementation of $f = x_1x_2 + x_2'x_3$

Programmable Logic Array (PLA)
Lookup Table (LUT)

(a) Circuit for a two-input LUT

(b) $f_1 = x_1 x_2$

(c) Storage cell contents in the LUT

Custom Design

- Intel 4004 Microprocessor 1971
- Intel Pentium 4 Processor 2000

Standard-Cell Design

Gate Array Design