**Logic Design with MOSFETs**

- For logic gates, we can think of MOSFETs as switches:

  ![Diagram](image)

  An NMOS FET works like an "assert-high" switch:
  
  ![Diagram](image)

  A PMOS FET works like an "assert-low" switch:
  
  ![Diagram](image)

- **Note:** CMOS logic uses complementary sets of PMOS & NMOS switches to insures that there is either:
  
  a) A path from VDD to the output, or
  
  b) A path from GND to the output, but never both!
EXAMPLE: A "NOT" GATE (INVERTER)

\[ Y = \overline{A} \]

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

TRUTH TABLE

TWO CASES

\[ A = 0 \quad Y = 1 \]

\[ A = 1 \quad Y = 0 \]
NOW, MOSFETS HAVE THRESHOLD VOLTAGES WHICH MUST BE OVERCOME BEFORE THE FET TURNS ON

\[ \begin{align*}
&VA \\ &\rightarrow \\ &M_N \\ &\rightarrow \\ &V_{PP} \\ &A = 1 \Rightarrow M_N = "ON" \\
&V_{IN} \\ &A = 0 \Rightarrow M_N = "OFF" \\
&0
\end{align*} \]

AND, SINCE WE DON'T WANT BOTH THE NMOS & PMOS FETS ON AT THE SAME TIME!

\[ \begin{align*}
&VA \\ &\rightarrow \\ &V_{PP} \\ &A = 1 \Rightarrow M_P = "OFF" \\
&V_{PD} - (V_{TH}) \\ &A = 0 \Rightarrow M_P = "ON" \\
&0
\end{align*} \]

NOTE: IF BOTH NMOS & PMOS FETS ARE ON AT THE SAME TIME:

1) OUTPUT LEVEL = UNKNOWN
2) SHORT BETWEEN VPP AND GND \Rightarrow LOTS OF CURRENT FLOWS!
   (I DEPENDS ON MOS RON's)

\} \text{ BAD!}
PASS CHARACTERISTICS OF MOS SWITCHES

- MOSFETS MAKE VERY GOOD SWITCHES, BUT THEY STILL NEED \( |VGS| > |VTH| \) TO TURN THEM ON.

**IDEAL CASE**

- NMOS SWITCH

- PMOS SWITCH

\[ \text{→ NMOS SWITCHES PASS OV CORRECTLY, BUT CAN ONLY PASS VDD AS VDD - VTN BEFORE THEY TURN OFF.} \]

\[ \text{→ PMOS SWITCHES PASS VDD CORRECTLY, BUT CAN ONLY PASS OV AS |VTP| BEFORE THEY TURN OFF.} \]

(NMOS PASSES "WEAK" LOGIC "1"

(PMOS PASSES "WEAK" LOGIC "0")
CMOS Transmission Gates ("T-Gates")

- To solve the "weak" 1 from NMOS switches, and "weak" 0 from PMOS switches, use both?

![MOS Circuit and Symbol]

- Here, at least one of the MOSFETs stay on when $S=1$, $\bar{S}=0$ regardless of the value of $X$.

  $\Rightarrow$ For: $X < |V_{TP}|$ $\Rightarrow$ NMOS on, PMOS off

  $|V_{TP}| < X < V_{DD} - V_{TN}$ $\Rightarrow$ Both NMOS, PMOS on

  $X > V_{TN}$ $\Rightarrow$ NMOS off, PMOS on

  $\Rightarrow$ T-Gates pass all voltages from 0 to VDD!

- Note that the source arrow is arbitrary. Here since MOSFETs are symmetrical (typically), the source & drain switch places depending on which one is at the higher voltage.

  $\Rightarrow$ T-Gates are "bi-directional" switches, they tie $X$ and $Y$ together, but either $X$ or $Y$ can be the input!

- Digital designers typically avoid using T-Gates, because their CAD tool can't handle...
INVERTERS ("NOT" GATES)

MOS CIRCUIT

\[ F(x) = \text{NOT}(x) = \overline{x} \]

Truth Table

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
NAND GATES ("NOT-AND" GATES)

2-INPUT

MOS CIRCUIT

\[ Y = \overline{A \cdot B} \]

TRUTH TABLE

3-INPUT

MOS CIRCUIT

(MORE INPUTS POSSIBLE: 4, 5, ...)

TRUTH TABLE
NOR GATES ("NOT-OR" GATES)

2-INPUT

\[ Y = A + B \]

\[ \text{Symbol} \]

MOS CIRCUIT

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

(More inputs are possible: 4, 5, ...)

3-INPUT

\[ Y = A + B + C \]

\[ \text{Symbol} \]

MOS CIRCUIT

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
**Complex Logic Gates**

- Can build virtually any complex logic function in a single CMOS logic gate
  - But, not usually done, because:
    1. Most digital blocks are built out of standard cells (not custom)
    2. Complex logic gates can be slow

- If we want to build a complex gate, then one way is to:
  1. Draw the NMOS part first, with "and" = series FETs; "or" = parallel FETs
  2. Draw the PMOS part as the NMOS "dual" series → parallel; parallel → series

**Example:** \( F = A \cdot B + C \cdot D \)

![Diagram](image)

- NMOS part

"Dual" = PMOS part!

\( A, B \) in series for "and"

\( A, B \) in parallel with \( C, D \) for "or"

\( C, D \) in series for "and"

\( A + B \) in series with \( C + D \)
**Final Gate!**

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

"AND-OR-INVERT" or "AOI" Gate

**Symbol**

- AOI gates are so useful that they are usually included in standard cell libraries.
"OR-AND-INVERT" or "OA" Gate

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

* ALSO VERY POPULAR!

* NOTE THE SIMILARITIES BETWEEN AOI & OA

GENERAL NAMING CONVENTION (AS PER TEXT)

AOI22

AOI321

OA1221

TYPE OF GATE

# OF INPUTS FOR 1ST AND

# OF INPUTS FOR 2ND AND

'1' = BYPASS OR (BRAND FOR AOI)

NOTE: MANY DIFFERENT NAMING CONVENTIONS EXIST!
**XOR AND XNOR GATES**

\[ A \oplus B \]

\[
\begin{array}{c|c|c}
A & B & A \oplus B \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

- OUTPUT = "1" IF EITHER \( A = 1 \) OR \( B = 1 \), BUT NOT BOTH

**THIS CAN BE WRITTEN AS:**

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

**OR**

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

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

**ALSO,**

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

**BOTH CAN BE BUILT FROM ADI GATES!**

---

**EXCLUSIVE-OR**

**EXCLUSIVE-NOR**

*NOTE: FIG. 2.57 IN TEXT IS WRONG!*
The XOR can thus be expressed as

\[ a \oplus b = \overline{(a \cdot \overline{b} + a \cdot \overline{b})} \]

which has AOI structure. Using the circuit in Figure 2.48(a) gives the basic AOI XOR circuit shown in Figure 2.57(a). Since the XOR gate has inputs of \((a, b)\) only, two inverters are needed to provide the 4-input set \((a, b, \overline{a}, \overline{b})\) in this circuit.

To obtain an XNOR circuit, we just complement the XOR SOP equation to write

\[ \overline{a \oplus b} = \overline{a \cdot \overline{b} + a \cdot \overline{b}} \]

Interchanging \(a\) and \(\overline{a}\) in the XOR circuit thus gives the XNOR gate in Figure 2.57(b). Switching the \(b\) and \(\overline{b}\) variables would have given the same result.

**Figure 2.57** AOI XOR and XNOR gates

*Fig 2.57 is wrong! For both (a) & (b) what is the output voltage?*
Transmission Gate Circuits

\[ S = \text{"select" control} \]
\[ S = 1 \Rightarrow y = B \]
\[ S = 0 \Rightarrow y = A \]

2-to-1 Multiplexor (MUX) (Can have 3, 4, ... inputs)

\[ S = 1 \Rightarrow y_2 = A \]
\[ S = 0 \Rightarrow y_1 = A \]

2-to-1 De-MUX

* The "non-active" output is high-impedance

* Good for driving 1 bus with >1 input
TRANSMISSION GATE CIRCUITS (CONT.)

![Diagram of a TG-based OR gate](image)

**Figure 2.61** A TG-based OR gate

![Diagram of XOR and XNOR circuits](image)

(a) XOR circuit  
(b) XNOR circuit

**Figure 2.62** TG-based exclusive-OR and exclusive-NOR circuits

![Diagram of an XNOR gate using TGs and FETs](image)

**Figure 2.64** An XNOR gate that uses both TGs and FETs
CLOCKING AND DATAFLOW CONTROL

Figure 2.65 Complementary clocking signals

Figure 2.68 Block-level system timing diagram

Figure 2.69 Control of binary words using clocking planes
CLOCKED T-GATES

(a) Closed switch

(b) Open switch

Figure 2.66 Behavior of a clocked TG

Figure 2.67 Data synchronization using transmission gates
DATA PATH CONTROL WITH LATCHES

(a) Logic diagram

(b) CMOS circuit

Figure 2.70 SR latch

(a) Logic diagram

(b) CMOS circuit

Figure 2.71 Clocked SR latch