CMOS Digital Gates:

\[ \text{Cloud} = C_{\text{wire}} + C_{\text{gate total}} \text{ (of next logic gates)} \]

(INVERTER)

1. \( \text{IN} \) \( (L \rightarrow H) \)
   \( \text{OUT} \) \( (H \rightarrow L) \)
   \( \text{NMOS} \) \( \text{off} \rightarrow \text{on} \) \( \Rightarrow \text{RN} \)
   \( \text{PMOS} \) \( \text{on} \rightarrow \text{off} \)

\[ t_{\text{delay}} = \ln(2) \times \text{RN} \times \text{Cloud} \]

2. \( \text{IN} \) \( (H \rightarrow L) \)
   \( \text{OUT} \) \( (L \rightarrow H) \)
   \( \text{NMOS} \) \( \text{on} \rightarrow \text{off} \)
   \( \text{PMOS} \) \( \text{off} \rightarrow \text{on} \) \( \Rightarrow \text{RP} \)

\[ t_{\text{delay}} = \ln(2) \times \text{RP} \times \text{Cloud} \]
What about a NAND gate or a NOR gate?

<table>
<thead>
<tr>
<th>AB</th>
<th>N1</th>
<th>N2</th>
<th>P1</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>01</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>10</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>11</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>off</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AB</th>
<th>OUT</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
<td>Pp/Rp</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>Rp</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>Rp</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>RN+RN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OUT</th>
<th>Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rp+Rp</td>
</tr>
<tr>
<td>0</td>
<td>RN</td>
</tr>
<tr>
<td>0</td>
<td>RN</td>
</tr>
<tr>
<td>0</td>
<td>RN/YN</td>
</tr>
</tbody>
</table>

What about XOR, AND, OR and XNOR gates?
(These are good exercises for practice!)
Voltage Transfer Characteristic

how can we make inverter operates at point C?

3 stable operating points A, B & C.
- Points A & B are used for digital logic level.
- Point C should be avoided.
(because it draws a lot of static power = $I_{sc} \times V_{DD}$)

Very attractive property of CMOS → No power dissipation when idle!

Switching power = $(C_{load})(V_{DD}^2)(frequency)$ [units of W]
Energy stored = $\frac{1}{2}C_{load}V_{DD}^2$ [units of J]

Noise Margin

\[ NM_H = V_{OH} - V_{IH} \]
\[ NM_L = V_{IL} - V_{OL} \]
Driver outputs $H > \text{Voh}$ such that receiver recognizes that as logic $H$, since $\text{Voh}$ always $> \text{Vih}$.

Example:

- \( \text{Voh} = 4\text{V} \)
- \( \text{Vih} = 3\text{V} \)
- \( \text{Vil} = 2\text{V} \)

Since logic $H$ voltage $> 4\text{V}$, receiver sees it as $H$ ($4\text{V} > 3\text{V}$).

Level in between is undefined.

Sequential logic

Has memory through feedback...

Can you replace this with a wire?

DRAM is essentially a wire + C. But this has no direct path to VDD & GND, unlike cross-coupled inverters or SRAM. DRAM needs refreshing...
D Flip-Flop

- D
- Q
- CLK

- this is edge trigger ⇒

- D: 0 → 1 (this will not cause Q to change since there is no edge in clock)
- Clk: 0 → 1
- Q: 0 → 1
- t_delay = ?

Latch is level sensitive

- D: 0 → 1
- Clk: 0 → 1
- Q follows D when Clk = 1

Combinational logic

- D depends on IN2 & Q (from previous cycle!)

L"
Even

\[
\begin{align*}
\text{latch} & \quad \text{odd.} & \quad \text{ring oscillator} & \quad \text{latch}
\end{align*}
\]

\[
\begin{align*}
in & \quad \text{out} & \quad \text{in} & \quad \text{out} & \quad \text{in} & \quad \text{out} & \quad \text{in} & \quad \text{out}
\end{align*}
\]

\[
\begin{align*}
\text{inverter} & \quad \text{inverter} & \quad \text{inverter}
\end{align*}
\]

\[
\begin{align*}
1 \cdot A &= A \\
1 + A &= 1 \\
0 + A &= 0 \\
\overline{AB} &= \overline{A + B} \\
A + \overline{B} &= (A \overline{B})
\end{align*}
\]

\[
\begin{align*}
A \cdot A &= A \\
A + A &= A \\
A \cdot \overline{A} &= 0 \\
\overline{A B} + \overline{A \overline{B}} &= A \oplus \overline{B} \\
A B + \overline{A \overline{B}} &= A \oplus B
\end{align*}
\]

\[
\begin{align*}
\overline{A B} &= \overline{A + B} \quad \text{(De Morgan's)}
\end{align*}
\]

\[
\begin{align*}
3\text{-input NAND}
\end{align*}
\]

Try to come up with transistor implementation of these 3-input gates?