CMOS inverter dynamic power dissipation

High state equivalent circuit

Low state equivalent circuit
CMOS inverter dynamic power dissipation

Charging current

High state equivalent circuit

Discharging current

Low state equivalent circuit
CMOS inverter dynamic power dissipation

\[ p_{DD} = V_{DD} i_D(t) \]  

Instantaneous power dissipation

\[ E_{DD} = \int_{0}^{T_C} V_{DD} i_D(t) \]  

Energy delivered to capacitor

\[ = V_{DD} \int_{0}^{T_C} i_D(t) \]  

\[ = V_{DD} Q \]

\[ Q = CV_{DD} \]

\[ E_{DD} = CV_{DD}^2 \]
CMOS inverter dynamic power dissipation

\[ E_{DD} = CV_{DD}^2 \]

Energy delivered to capacitor
CMOS inverter dynamic power dissipation

\[ E_{DD} = CV_{DD}^2 \]  
Energy delivered to capacitor

\[ E_{\text{stored}} = \frac{1}{2} CV_{DD}^2 \]  
Stored energy

\[ E_{\text{dissipated}} = EC V_{\text{cycle}} \]
CMOS inverter dynamic power dissipation

\[ E_{DD} = CV_{DD}^2 \]  
Energy delivered to capacitor

\[ E_{stored} = \frac{1}{2} CV_{DD}^2 \]  
Stored energy

\[ E_{dissipated} = E_{DD} - E_{stored} = \frac{1}{2} CV_{DD}^2 \]  
Energy dissipation
CMOS inverter dynamic power dissipation

\[ E_{DD} = CV_{DD}^2 \]  
Energy delivered to capacitor

\[ E_{\text{stored}} = \frac{1}{2} CV_{DD}^2 \]  
Stored energy

\[ E_{\text{dissipated}} = E_{DD} - E_{\text{stored}} = \frac{1}{2} CV_{DD}^2 \]  
Energy dissipation

\[ E_{\text{dissipated}} / \text{cycle} = CV_{DD}^2 \]  
Energy dissipated in pull-up plus pull-down
CMOS inverter dynamic power dissipation

\[ E_{DD} = C V_{DD}^2 \]

Energy delivered to capacitor

\[ E_{\text{stored}} = \frac{1}{2} C V_{DD}^2 \]

Stored energy

\[ E_{\text{dissipated}} = E_{DD} - E_{\text{stored}} = \frac{1}{2} C V_{DD}^2 \]

Energy dissipation

\[ E_{\text{dissipated}} / \text{cycle} = C V_{DD}^2 \]

Energy dissipated in pull-up plus pull-down

\[ P_{\text{dynamic}} = f C V_{DD}^2 \]

Dynamic power dissipation
Inverter propagation delay

Input

\[ V_{DD} \]

Output

\[ V_{DD} \]

\[ \frac{V_{DD}}{2} \]

\[ t_{PHL} \]

\[ t_{PLH} \]
Inverter propagation delay

Time to discharge output capacitance

Time to charge output capacitance

Input

Output
Inverter propagation delay

\[ t_p = \frac{t_{pLH} + t_{pHL}}{2} \]

\[ T_{\text{min}} = t_{pLH} + t_{pHL} = 2t_p \]

\[ f_{\text{max}} = \frac{1}{T_{\text{min}}} = \frac{1}{2t_p} \]

Average propagation delay

Minimum period of square wave

Maximum switching speed
Source of propagation delay

- MOSFET capacitance
- Wiring capacitance
- Gate input capacitance
Performance metric

\[ PDP \equiv P_D t_p \]

Power delay product

\[ P_D = P_{dyn} = fCV_{DD}^2 \]

For CMOS

\[ PDP = fCV_{DD}^2 t_p \]

\[ PDP = \frac{1}{2} CV_{DD}^2 \]

For CMOS, at maximum switching speed
CMOS inverter

\[ V_{DD} \quad Q_P \quad i_{DP} \quad i_{DN} \quad Q_N \quad \]

\[ V_I \quad U_I \quad U_O \quad \]

\[ V_{OH} = V_{DD} \quad Q_N \text{ off} \quad A \quad \text{Slope} = -1 \]

\[ \left(\frac{V_{DD}}{2} + V_t\right) \quad \]

\[ \left(\frac{V_{DD}}{2} - V_t\right) \quad Q_N \text{ in triode region} \quad Q_P \text{ in saturation} \quad B \quad \text{Slope} = -1 \]

\[ V_O \quad Q_N \text{ and } Q_P \quad \]

\[ V_{OL} = 0 \quad V_t \quad V_{IL} \quad V_{IH} \quad (V_{DD} - V_t) \quad \]

\[ (V_{DD} - V_I) \quad Q_P \text{ off} \quad D \quad V_{DD} \quad V_M = \frac{V_{DD}}{2} \]

\[ v_O \quad v_I \]
CMOS inverter propagation delay
CMOS inverter dynamic power dissipation

\[ P_{\text{dyn}} = fCV_{DD}^2 \]

\[ I_{\text{peak}} = \frac{1}{2} \mu_n C_{ox} \left( \frac{W}{L} \right)_n \left( \frac{V_{DD}}{2} - V_{tn} \right)^2 \]
CMOS inverter dynamic power dissipation

\[ I_{\text{peak}} \]

\[ 0 \quad V_{\text{in}} \quad \frac{V_{\text{DD}}}{2} \quad V_{\text{DD}} - |V_{ip}| \quad V_{\text{DD}} \]

\[ i \]

\[ v_I \]
Performance metric: by logic family, design sizes

![Graph showing improving technologies over power and delay.]

- Power
- Delay
- Improving technologies
Speed/power tradeoffs

Power Dissipation $\sim CV^2$

- Reduce $C$
- Reduce size
- Reduced drive capability
- Reduced area for thermal dissipation

- Reduce $V$
- Reduced noise margin
Logic design evolution

Small Scale Integration (SSI)  ~1-10s gates/package  Fundamental logic function

Medium Scale Integration (MSI)  ~10s-100s gates/package  Simple logic systems

Large Scale Integration (LSI)  ~100s-1000s gates/package  Complete logic systems
Logic design evolution

Very Large Scale Integration (VLSI) ~1000s-100,000s gates/package Complete complex logic systems

Wafer Scale Integration (VLSI) ~100,000s-1,000,000s gates/package Multiple complete complex logic systems
Logic design options

Rapid prototyping, expensive manufacture

“Jelly bean” designs with SSI/MSI

Circuit boards with dozens of discrete logic devices

Custom ASICs for complex designs

Circuit boards with a few ASICs plus a few SSI/MSI “glue logic” devices

FPGAs and SoCs

Circuit boards with a one complex VLSI/WSI device plus a minimal “glue logic” devices

Rapid prototyping, reduced cost of manufacture

Long development interval, lowest manufacturing cost

Rapid prototyping, moderate cost of manufacture
Logic design options

- **Rapid prototyping, expensive manufacture**
- **“Jelly bean” designs with SSI/MSI**
- **Common functions built into LSI**
- **Circuit boards with dozens of discrete logic devices**
- **Circuit boards with several LSI devices plus serveral SSI/MSI for “glue logic”**

- **Rapid prototyping, reduced cost of manufacture**

- **Long development interval, lowest manufacturing cost**

- **Rapid prototyping, moderate cost of manufacture**
Logic design options

Rapid prototyping, expensive manufacture

Rapid prototyping, reduced cost of manufacture

Long development interval, lowest manufacturing cost

“Jelly bean” designs with SSI/MSI

Circuit boards with dozens of discrete logic devices

Common functions built into LSI

Circuit boards with several LSI devices plus serveral SSI/MSI for “glue logic”

Custom ASICs for complex designs

Circuit boards with a few ASICs plus a few SSI/MSI “glue logic” devices

Rapid prototyping, reduced cost of manufacture

Long development interval, lowest manufacturing cost

Rapid prototyping, expensive manufacture
Logic design options

- "Jelly bean" designs with SSI/MSI
  - Circuit boards with dozens of discrete logic devices

- Common functions built into LSI
  - Circuit boards with several LSI devices plus serveral SSI/MSI for "glue logic"

- Custom ASICs for complex designs
  - Circuit boards with a few ASICs plus a few SSI/MSI "glue logic" devices

- FPGAs and SoCs
  - Circuit boards with a one complex VLSI/WSI device plus a minimal "glue logic"

Rapid prototyping, expensive manufacture

Rapid prototyping, reduced cost of manufacture

Long development interval, lowest manufacturing cost

Rapid prototyping, moderate cost of manufacture

Rapid prototyping, reduced cost of manufacture

Long development interval, lowest manufacturing cost

Rapid prototyping, moderate cost of manufacture
Logic design options

- **“Jelly bean” designs with SSI/MSI**
  - Circuit boards with dozens of discrete logic devices
- **Common functions built into LSI**
  - Circuit boards with several LSI devices plus serveral SSI/MSI for “glue logic”
- **Custom ASICs for complex designs**
  - Circuit boards with a few ASICs plus a few SSI/MSI “glue logic” devices
- **FPGAs and SoCs**
  - Circuit boards with a one complex VLSI/WSI device plus a minimal “glue logic”

**Rapid prototyping**, expensive manufacture

**Rapid prototyping**, reduced cost of manufacture

**Long development interval**, lowest manufacturing cost

**Rapid prototyping**, moderate cost of manufacture

Rapid prototyping, reduced cost of manufacture
CMOS logic gates

Series AND functions
Parallel OR functions

PMOS transistors
NMOS transistors
### Pull-down networks

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

\( \overline{Y} = A + B \)

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

\( \overline{Y} = AB \)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></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>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Pull-up networks

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

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

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

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

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

<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>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</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>
Two-input NOR gate

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

\[ Y = \overline{A + B} = \overline{A} \cdot \overline{B} \]
# Two-input NAND gate

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

\[
Y = \overline{AB} = \overline{A} + \overline{B}
\]
Two-input XOR gate

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y = A ⊕ B</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>0</td>
</tr>
</tbody>
</table>

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