CMOS Technology
A Deep Dive Into a Chip

Packaged chip

Source: Intel
A Deep Dive Into a Chip

Packaged chip

Silicon die (100-400mm²)

Source: Intel
A Deep Dive Into a Chip

Packaged chip

Silicon die (100-400mm²)

Die cross-section

6-15 metal layers (wires)

Source: Intel

October 3, 2019
A Deep Dive Into a Chip

Packaged chip

Silicon die (100-400mm²)

Die cross-section

Transistor (FET)

6-15 metal layers (wires)

Source: Intel

October 3, 2019
MIT 6.004 Fall 2019
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties: nFET and pFET.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties: nFET and pFET.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties: nFET and pFET.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties: nFET and pFET.

```
          D
         /|
        / G
       /  |
      /    |
     /     |
    /      |
   /       |
  nFET    S (source) (gate)
```
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are **voltage-controlled switches**
- FETs come in two varieties: nFET and pFET

```
  nFET
     D (drain)
   G (gate)
     S (source)
```
Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

FETs come in two varieties: nFET and pFET.

- **nFET**
  - **D** (drain)
  - **G** (gate)
  - **S** (source)

A high voltage at gate creates conducting path between source and drain.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties: nFET and pFET.

A high voltage at gate creates conducting path between source and drain.
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches

- FETs come in two varieties: nFET and pFET

A high voltage at gate creates conducting path between source and drain

A low voltage at gate creates conducting path between source and drain
Labeling Source and Drain

- There is no physical difference between source and drain, called the **diffusion terminals**
Labeling Source and Drain

- There is no physical difference between source and drain, called the **diffusion terminals**
- By convention, we label diffusion terminals as source or drain depending on their voltages:
Labeling Source and Drain

- There is no physical difference between source and drain, called the diffusion terminals.
- By convention, we label diffusion terminals as source or drain depending on their voltages:
  - On nFETs, source = diffusion terminal at lower voltage.
Labeling Source and Drain

- There is no physical difference between source and drain, called the **diffusion terminals**
- By convention, we label diffusion terminals as source or drain depending on their voltages:
  - On nFETs, source = diffusion terminal at lower voltage
  - On pFETs, source = diffusion terminal at higher voltage
Labeling Source and Drain

- There is no physical difference between source and drain, called the **diffusion terminals**
- By convention, we label diffusion terminals as source or drain depending on their voltages:
  - On nFETs, source = diffusion terminal at lower voltage
  - On pFETs, source = diffusion terminal at higher voltage

```
D (higher voltage)
```

```
G
```

```
S (lower voltage)
```
Labeling Source and Drain

- There is no physical difference between source and drain, called the **diffusion terminals**.
- By convention, we label diffusion terminals as source or drain depending on their voltages:
  - On nFETs, source = diffusion terminal at lower voltage
  - On pFETs, source = diffusion terminal at higher voltage

```
D (higher voltage)                     S (higher voltage)
G
S (lower voltage)                     G
D (lower voltage)
```
There is no physical difference between source and drain, called the **diffusion terminals**.

By convention, we label diffusion terminals as source or drain depending on their voltages:
- On nFETs, source = diffusion terminal at lower voltage
- On pFETs, source = diffusion terminal at higher voltage

This convention lets us define the behavior of FETs using the voltage between gate and source.
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
- pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a **threshold voltage** $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
- pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise

\[ V_{GS} < V_{TH} \quad \text{OFF} \quad V_{GS} > V_{TH} \quad \text{ON} \]

\[ G \quad S \quad D \]

\[ V_{GS} \]

\[ G \quad S \quad D \]

\[ V_{SG} \]
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
- pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
- pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise
FET Switching Model

- FETs have a threshold voltage $V_{TH}$
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise
- pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise

- This is a very simplified model, but it is sufficient to build logic gates
What Does This Circuit Compute?
What Does This Circuit Compute?
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$
What Does This Circuit Compute?

Assume $V_{TH} < \frac{V_{DD}}{2}$

$V_{IN} < V_{TH}$

$V_{OUT}$
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

$V_{IN} < V_{TH}$

$V_{DD}$

$V_{OUT}$
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

$V_{IN} < V_{TH}$
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

- $V_{IN} < V_{TH}$
- $V_{IN} > V_{DD} - V_{TH}$
What Does This Circuit Compute?

Assume $V_{\text{TH}} < V_{\text{DD}}/2$

- $V_{\text{IN}} < V_{\text{TH}}$
- $V_{\text{IN}} > V_{\text{DD}} - V_{\text{TH}}$
What Does This Circuit Compute?

![Circuit Diagram]

Assume $V_{TH} < V_{DD}/2$

- $V_{IN} < V_{TH}$
- $V_{IN} > V_{DD} - V_{TH}$

MIT 6.004 Fall 2019
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

<table>
<thead>
<tr>
<th>$V_{IN}$</th>
<th>$V_{OUT}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD}$</td>
<td>$V_{DD}$</td>
</tr>
<tr>
<td>$V_{IN}$</td>
<td>$V_{OUT}$</td>
</tr>
</tbody>
</table>

$V_{IN} < V_{TH}$

$V_{IN} > V_{DD} - V_{TH}$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

- $V_{IN} < V_{TH}$
- $V_{IN} > V_{DD} - V_{TH}$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

Assume $V_{TH} < V_{DD}/2$

- $V_{IN} < V_{TH}$
- $V_{IN} > V_{DD} - V_{TH}$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

CMOS inverter

Assume $V_{TH} < V_{DD}/2$

$V_{IN} < V_{TH}$  $V_{IN} > V_{DD} - V_{TH}$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Note on Terminology

- MOSFETs (metal-oxide-semiconductor field-effect transistors) are the most common type of FET
Note on Terminology

- MOSFETs (metal-oxide-semiconductor field-effect transistors) are the most common type of FET

- nFET and pFET are sometimes abbreviated as nMOS and pMOS
Note on Terminology

- MOSFETs (metal-oxide-semiconductor field-effect transistors) are the most common type of FET
  
- nFET and pFET are sometimes abbreviated as nMOS and pMOS
  
- CMOS stands for complementary MOS
What Does This Circuit Compute?

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</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>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```
What Does This Circuit Compute?

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

October 3, 2019
What Does This Circuit Compute?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</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></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</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></td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</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>
What Does This Circuit Compute?

CMOS NAND gate

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</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>

CMOS NAND gate
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

```
pullup  pulldown  F(inputs)
```
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven “1”</td>
</tr>
</tbody>
</table>

Diagram:
- Power supply
- Pullup circuit
- Pulldown circuit
- Inputs
- Output
- Ground
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven (\text{&quot;1&quot;})</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>driven (\text{&quot;0&quot;})</td>
</tr>
</tbody>
</table>

Diagram:
- Power supply
- Pullup circuit
- Pulldown circuit
- Ground
- Inputs
- Output
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven “1”</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>driven “0”</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>driven “X”</td>
</tr>
</tbody>
</table>
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven “1”</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>driven “0”</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>driven “X”</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>no connection</td>
</tr>
</tbody>
</table>
CMOS Logic

- CMOS gates have complementary pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven “1”</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>driven “0”</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>driven “X”</td>
</tr>
</tbody>
</table>
| off    | off      | no connection      

- CMOS uses pFETs to implement the pullup network and nFETs to implement the pulldown network.
Some Questionable Gates

- What can go wrong with the following gates?
Some Questionable Gates

- What can go wrong with the following gates?
Some Questionable Gates

- What can go wrong with the following gates?
Some Questionable Gates

- What can go wrong with the following gates?

A=0 B=1 or A=1 B=0 connect supply and ground
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks

A=0 B=1 or A=1 B=0 connect supply and ground
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks

A=0 B=1 or A=1 B=0 connect supply and ground
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks

A=0 B=1 or A=1 B=0 connect supply and ground
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks

A=0 B=1 or A=1 B=0 connect supply and ground

pFET doesn’t pull down
V_{OUT} below V_{TH}
nFET doesn’t pull up
V_{OUT} above V_{DD} – V_{TH}
Some Questionable Gates

- What can go wrong with the following gates?

   - CMOS Rule #1: Complementary pullup and pulldown networks

     - pFET doesn’t pull down $V_{OUT}$ below $V_{TH}$
     - nFET doesn’t pull up $V_{OUT}$ above $V_{DD} - V_{TH}$

     - A=0 B=1 or A=1 B=0 connect supply and ground
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks

  - A=0 B=1 or A=1 B=0 connect supply and ground
  - pFET doesn’t pull down $V_{OUT}$ below $V_{TH}$
  - nFET doesn’t pull up $V_{OUT}$ above $V_{DD} - V_{TH}$
Some Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks
- CMOS Rule #2: pFETs in pullup, nFETs in pulldown

A=0 B=1 or A=1 B=0 connect supply and ground

pFET doesn’t pull down
V_{OUT} below V_{TH}
nFET doesn’t pull up
V_{OUT} above V_{DD} - V_{TH}
CMOS Complements

A conducts when A is high

A conducts when A is low: $\overline{A}$
CMOS Complements

- Conducts when $A$ is high: $A$
- Conducts when $A$ is low: $\overline{A}$
- Conducts when $A$ is high and $B$ is high: $A \cdot B$
- Conducts when $A$ is low or $B$ is low: $\overline{A} + \overline{B} = \overline{A \cdot B}$
CMOS Complements

conducts when A is high

conducts when A is low: $\overline{A}$

conducts when A is high and B is high: $A \cdot B$

conducts when A is low or B is low: $\overline{A} + \overline{B} = A \cdot B$

conducts when A is high or B is high: $A + B$

conducts when A is low and B is low: $\overline{A} \cdot \overline{B} = A + B$
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

$$F = \overline{A} + \overline{B} \times \overline{C}$$

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets.

Step 3. Combine pFET pullup network from Step 1 with nFET pulldown network from Step 2 to form the CMOS gate.
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets

Step 3. Combine pFET pullup network from Step 1 with nFET pulldown network from Step 2 to form the CMOS gate.
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets

Step 3. Combine pFET pullup network from Step 1 with nFET pulldown network from Step 2 to form the CMOS gate.

Can CMOS gates implement arbitrary functions?
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

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

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets

Step 3. Combine pFET pullup network from Step 1 with nFET pulldown network from Step 2 to form the CMOS gate.

Can CMOS gates implement arbitrary functions? No
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and vice versa.
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and viceversa.

- On a rising input,
  - nFETs go OFF→ON, so pulldown may connect output to ground.
  - pFETs go ON→OFF, so pullup may disconnect output from $V_{DD}$. 
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0 → 1) lead to falling outputs (1 → 0) and vice versa.

- On a rising input,
  - nFETs go OFF → ON, so pulldown may connect output to ground.
  - pFETs go ON → OFF, so pullup may disconnect output from $V_{DD}$.
  - Output either stays the same or falls.
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and vice versa.

- On a rising input,
  - nFETs go OFF→ON, so pulldown may connect output to ground
  - pFETs go ON→OFF, so pullup may disconnect output from $V_{DD}$
  - Output either stays the same or falls

- Corollary: Cannot build non-inverting logic using a single CMOS gate
  - Example: AND
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0 → 1) lead to falling outputs (1 → 0) and vice versa.

- On a rising input,
  - nFETs go OFF → ON, so pulldown may connect output to ground.
  - pFETs go ON → OFF, so pullup may disconnect output from $V_{DD}$.
  - Output either stays the same or falls.

- Corollary: Cannot build non-inverting logic using a single CMOS gate.
  - Example: AND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>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>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>
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and vice versa.

- On a rising input,
  - nFETs go OFF→ON, so pulldown may connect output to ground
  - pFETs go ON→OFF, so pullup may disconnect output from $V_{DD}$
  - Output either stays the same or falls

- Corollary: Cannot build non-inverting logic using a single CMOS gate
  - Example: AND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$A \cdot B$</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>
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and vice versa.

- On a rising input,
  - nFETs go OFF→ON, so pulldown may connect output to ground
  - pFETs go ON→OFF, so pullup may disconnect output from $V_{DD}$
  - Output either stays the same or falls

- Corollary: Cannot build non-inverting logic using a single CMOS gate
  - Example: AND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>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>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>

rising input  rising output
Analyzing the Delay, Area, and Power of CMOS Gates

NOTE: Demystification, will not be on the quiz
MOSFET Physical Structure

nFET

gate

source  drain
MOSFET Physical Structure
MOSFET Physical Structure

nFET

source

gate

drain

source

gate

drain

L

W
MOSFET Physical Structure

nFET

gate

source

drain

source

gate

drain

nFET

gate

source

drain

n

n

p
MOSFET Physical Structure

- **source**
- **drain**
- **gate**

**nFET**

- **Metal**
- **Oxide (dielectric)**
- **Semiconductor**

*Image showing the physical structure of a MOSFET with labeled components: source, drain, gate, semiconductor, metal, oxide (dielectric).*
MOSFET Electrical View

With $V_{GS} < V_{TH}$, almost no current flows between source and drain.

Diagram showing a MOSFET with labels for source (S), drain (D), gate (G), and depletion region.
MOSFET Electrical View

With $V_{GS} < V_{TH}$, almost no current flows between source and drain.

As $V_{GS}$ reaches $V_{TH}$, a channel forms between source and drain.

[Diagram showing depletion region and channel formation]
MOSFET Electrical View

With $V_{GS} < V_{TH}$, almost no current flows between source and drain.

As $V_{GS}$ reaches $V_{TH}$, a channel forms between source and drain.

The shape of the channel (and its resistance) also depends on the voltage at the drain. But a low-resistance channel will exist while $V_{GS} > V_{TH}$.
FET *First-Order Electrical Model*
FET *First-Order* Electrical Model

![Diagram of a MOSFET with labels G, S, D, and V_{GS}](image-url)
FET *First-Order* Electrical Model

\[ R_{channel} = \begin{cases} 
R_{OFF} & \text{if } V_{GS} < V_{TH} \\
R_{ON} & \text{if } V_{GS} \geq V_{TH} 
\end{cases} \]
FET First-Order Electrical Model

\[ V_{GS} \]

\[ R_{channel} = \begin{cases} 
R_{OFF} & \text{if } V_{GS} < V_{TH} \\
R_{ON} & \text{if } V_{GS} \geq V_{TH} 
\end{cases} \]

\[ R_{ON} \ll R_{OFF} \]
FET *First-Order* Electrical Model

\[ R_{\text{channel}} = \begin{cases} 
R_{\text{OFF}} & \text{if } V_{GS} < V_{TH} \\
R_{\text{ON}} & \text{if } V_{GS} \geq V_{TH} 
\end{cases} \]

\[ R_{\text{ON}} \ll R_{\text{OFF}} \]
FET *First-Order* Electrical Model

\[
R_{\text{channel}} = \begin{cases} 
R_{\text{OFF}} & \text{if } V_{GS} < V_{TH} \\
R_{\text{ON}} & \text{if } V_{GS} \geq V_{TH} 
\end{cases}
\]

\[
R_{\text{ON}} \ll R_{\text{OFF}}
\]
FET *First-Order* Electrical Model

- Simplest possible model that lets us reason about delay, area, and power. Not very accurate!

\[ R_{channel} = \begin{cases} 
R_{OFF} & \text{if } V_{GS} < V_{TH} \\
R_{ON} & \text{if } V_{GS} \geq V_{TH} 
\end{cases} \]

\[ R_{ON} \ll R_{OFF} \]
Consider the following circuit. Given \( V_{IN}(t) \), can you derive \( V_{OUT}(t) \)?
Consider the following circuit. Given $V_{\text{IN}}(t)$, can you derive $V_{\text{OUT}}(t)$?
Consider the following circuit. Given $V_{IN}(t)$, can you derive $V_{OUT}(t)$?

For $t > 0$, $V_{OUT}(t)$
CMOS Gate Delay

Consider the following circuit. Given $V_{IN}(t)$, can you derive $V_{OUT}(t)$?

For $t > 0$, $V_{OUT}(t)$
Consider the following circuit. Given $V_{IN}(t)$, can you derive $V_{OUT}(t)$?

For $t > 0$,
Consider the following circuit. Given \( V_{\text{IN}}(t) \), can you derive \( V_{\text{OUT}}(t) \)?

For \( t > 0 \),
Consider the following circuit. Given $V_{IN}(t)$, can you derive $V_{OUT}(t)$?

For $t > 0$, 

$$V_{OUT}(t) = VOU_T(0)e^{-t/RC}$$
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

\[ V_{IN} \]

\[ V_{IL} \quad V_{IH} \]

\[ V_{OUT} \]

\[ V_{OL} \quad V_{OH} \]
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

![Graph showing input and output voltages](image)
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

$V_{IN}$

$V_{IH}$

$V_{IL}$

$V_{OUT}$

$V_{OH}$

$V_{OL}$
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

$V_{IN}$

$V_{IH}$

$V_{IL}$

$V_{OUT}$

$V_{OH}$

$V_{OL}$
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

$V_{IN}$

$V_{IH}$

$V_{IL}$

$V_{OUT}$

$V_{OH}$

$V_{OL}$
Propagation delay (\(t_{PD}\)): Upper bound on the delay from valid inputs to valid outputs.

\[
V_{IN} < V_{IL} < V_{IH} < V_{OUT}
\]

\[
V_{OL} < V_{OH} < V_{OUT}
\]
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

To minimize $t_{PD}$, must keep resistances and capacitances low.
Contamination Delay

Contamination delay ($t_{CD}$): Lower bound on the delay from any invalid input to an invalid output.
Contamination Delay

Contamination delay ($t_{CD}$): Lower bound on the delay from any invalid input to an invalid output

\[ V_{\text{IN}} \]
\[ V_{\text{IH}} \]
\[ V_{\text{IL}} \]
\[ V_{\text{OUT}} \]
\[ V_{\text{OH}} \]
\[ V_{\text{OL}} \]
Contamination Delay

Contamination delay ($t_{CD}$): Lower bound on the delay from any invalid input to an invalid output.
Contamination delay ($t_{CD}$): Lower bound on the delay from any invalid input to an invalid output

Contamination Delay

- $V_{IN}$
- $V_{IH}$
- $V_{IL}$
- $V_{OUT}$
- $V_{OH}$
- $V_{OL}$
Contamination Delay

Contamination delay ($t_{CD}$): Lower bound on the delay from any invalid input to an invalid output

\[ V_{IN} \geq t_{CD} \geq V_{OUT} \]
MOSFET Sizing
MOSFET Sizing

\[ \text{source} \quad \text{gate} \quad \text{drain} \]

\[ W \quad L \]

\[ R_{\text{channel}} \quad C_{\text{gate}} \]

\[ D \quad S \]
MOSFET Sizing

How do $C_{\text{gate}}$ and $R_{\text{channel}}$ change with L and W?
MOSFET Sizing

How do $C_{\text{gate}}$ and $R_{\text{channel}}$ change with $L$ and $W$?

$C_{\text{gate}} \propto $
MOSFET Sizing

How do $C_{gate}$ and $R_{channel}$ change with $L$ and $W$?

$C_{gate} \propto L \cdot W$
MOSFET Sizing

How do $C_{gate}$ and $R_{channel}$ change with $L$ and $W$?

$C_{gate} \propto L \cdot W$

$R_{channel} \propto$
How do $C_{gate}$ and $R_{channel}$ change with $L$ and $W$?

- $C_{gate} \propto L \cdot W$
- $R_{channel} \propto L/W$
MOSFET Sizing

- CMOS gates use MOSFETs with smallest possible L and choose W to set performance
  - Wider FETs drive more current (lower R), but their gates are harder to drive (higher C) and they take more area.
Standard Cell Libraries

- A standard cell library provides implementations of common gates (NAND, NOR, XOR, etc.) for a specific implementation technology
Standard Cell Libraries

- A standard cell library provides implementations of common gates (NAND, NOR, XOR, etc.) for a specific implementation technology.

- Each gate includes:
  - Electrical parameters (e.g., Rs and Cs)
  - Physical layout
Standard Cell Libraries

- A standard cell library provides implementations of common gates (NAND, NOR, XOR, etc.) for a specific implementation technology.

- Each gate includes:
  - Electrical parameters (e.g., Rs and Cs)
  - Physical layout
Standard Cell Libraries

- A standard cell library provides implementations of common gates (NAND, NOR, XOR, etc.) for a specific implementation technology.

- Each gate includes:
  - Electrical parameters (e.g., Rs and Cs)
  - Physical layout

- Synthesis tools use gates from the standard library instead of sizing and placing individual transistors.

Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pull-down}}$ or $R_{\text{pull-up}}$. Instead, use trees of smaller devices...

Example: 8-input NAND
Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pulldown}}$ or $R_{\text{pullup}}$. Instead, use trees of smaller devices...

Example: 8-input NAND
Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pulldown}}$ or $R_{\text{pullup}}$. Instead, use trees of smaller devices...

Example: 8-input NAND

How does $t_{PD}$ grow with the number of inputs $N$?
Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pulldown}}$ or $R_{\text{pullup}}$. Instead, use trees of smaller devices...

Example: 8-input NAND

How does $t_{PD}$ grow with the number of inputs $N$?

If we use a single CMOS gate, $t_{PD} \propto N$
Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pull\down}}$ or $R_{\text{pull\up}}$. Instead, use trees of smaller devices...

Example: 8-input NAND

How does $t_{\text{PD}}$ grow with the number of inputs $N$?

If we use a single CMOS gate, $t_{\text{PD}} \propto N$

If we use a tree of gates, $t_{\text{PD}} \propto \log(N)$
CMOS Power Dissipation

- Total power dissipation: \( P = P_{\text{dynamic}} + P_{\text{static}} \)
- Dynamic power: Caused by 0\(\leftrightarrow\)1 transitions of nodes in the circuit
  - Charging/discharging each capacitor consumes \( \frac{1}{2} CV_{DD}^2 \) energy
  - If on average \( C_S \) capacitance across the chip switches each cycle, and there are \( f_{CLK} \) cycles per second
    \[
    P_{\text{dynamic}} = \frac{1}{2} C_S V_{DD}^2 f_{CLK}
    \]
- Static power: Caused by
  - Subthreshold leakage: Even when the FET is off, a very small current flows from source to drain (\( R_{OFF} < \infty \))
  - Tunneling current: Gate and channel are separated by a very thin (<1nm) dielectric, so some electrons tunnel through
    \[
    P_{\text{static}} = I_{\text{static}} V_{DD}
    \]
  - Static power is typically 10-30% of total power
Summary

- FETs behave as voltage-controlled switches

- CMOS gates:
  - Use complementary pullup and pulldown networks
  - Use pFETs in pullup, nFETs in pulldown network

- CMOS gates are inverting (rising inputs can only cause falling outputs, and vice versa)
Thank you!

Next lecture:
Sequential logic