Lecture 1: Circuits & Layout
Outline

- A Brief History
- CMOS Gate Design
- Pass Transistors
- CMOS Latches & Flip-Flops
- Standard Cell Layouts
- Stick Diagrams
A Brief History

- 1958: First integrated circuit
  - Flip-flop using two transistors
  - Built by Jack Kilby at Texas Instruments

- 2010
  - Intel Core i7 μprocessor
    - 2.3 billion transistors
  - 64 Gb Flash memory
    - > 16 billion transistors
Growth Rate

- 53% compound annual growth rate over 50 years
  - No other technology has grown so fast so long
- Driven by miniaturization of transistors
  - Smaller is cheaper, faster, lower in power!
  - Revolutionary effects on society

[Moore65]
Electronics Magazine
Annual Sales

- \( >10^{19} \) transistors manufactured in 2008
  - 1 billion for every human on the planet
Invention of the Transistor

- Vacuum tubes ruled in first half of 20\textsuperscript{th} century
  Large, expensive, power-hungry, unreliable
- 1947: first point contact transistor
  - John Bardeen and Walter Brattain at Bell Labs
  - See *Crystal Fire*
    by Riordan, Hoddeson
Transistor Types

- Bipolar transistors
  - npn or pnp silicon structure
  - Small current into very thin base layer controls large currents between emitter and collector
  - Base currents limit integration density

- Metal Oxide Semiconductor Field Effect Transistors
  - nMOS and pMOS MOSFETS
  - Voltage applied to insulated gate controls current between source and drain
  - Low power allows very high integration
MOS Integrated Circuits

- 1970’s processes usually had only nMOS transistors
  - Inexpensive, but consume power while idle

Intel 1101 256-bit SRAM

1980s-present: CMOS processes for low idle power

Intel 4004 4-bit μProc

[Vadasz69] © 1969 IEEE.
Moore’s Law: Then

1965: Gordon Moore plotted transistors per chip
   Fit straight line on semi-log scale
   Transistor counts have doubled every 26 months

Integration Levels
SSI: 10 gates
MSI: 1000 gates
LSI: 10,000 gates
VLSI: > 10k gates
And Now...
Feature Size

- Minimum feature size shrinking 30% every 2-3 years
Corollaries

- Many other factors grow exponentially
  - Ex: clock frequency, processor performance
Activity:

- Sketch a 4-input CMOS NOR gate
Complementary CMOS

- Complementary CMOS logic gates
  - nMOS pull-down network
  - pMOS pull-up network
  - a.k.a. static CMOS

<table>
<thead>
<tr>
<th></th>
<th>Pull-up OFF</th>
<th>Pull-up ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull-down OFF</td>
<td>Z (float)</td>
<td>1</td>
</tr>
<tr>
<td>Pull-down ON</td>
<td>0</td>
<td>NO!</td>
</tr>
</tbody>
</table>
Series and Parallel

- nMOS: 1 = ON
- pMOS: 0 = ON
- Series: both must be ON
- Parallel: either can be ON
Conduction Complement

- Complementary CMOS gates always produce 0 or 1
- Ex: NAND gate
  - Series nMOS: Y=0 when both inputs are 1
  - Thus Y=1 when either input is 0
  - Requires parallel pMOS

- Rule of Conduction Complements
  - Pull-up network is complement of pull-down
  - Parallel -> series, series -> parallel
Compound Gates

- **Compound gates** can do any inverting function
- Ex: $Y = \text{NOT}(AB + CD)$ (AND-AND-OR-INVERT, AOI22)

(a) \hspace{2cm} (b)

(c) \hspace{2cm} (d)

(e) \hspace{2cm} (f)
Example: O3Al

\[ Y = \text{NOT} \{(A + B + C) \cdot D\} \]
Signal Strength

- **Strength** of signal
  - How close it approximates ideal voltage source
- $V_{DD}$ and GND rails are strongest 1 and 0
- nMOS pass strong 0
  - But degraded or weak 1
- pMOS pass strong 1
  - But degraded or weak 0
- Thus nMOS are best for pull-down network
Pass Transistors

- Transistors can be used as switches

\[ g = 0 \]
\[ s \rightarrow d \]
\[ g = 1 \]
\[ s \rightarrow d \]

Input: 0 → strong 0
Output: 0 → degraded 0

Input: 1 → degraded 1
Output: 1 → strong 1
Transmission Gates

- Pass transistors produce degraded outputs
- Transmission gates pass both 0 and 1 well

\[
\begin{align*}
\text{Input} & \quad \text{Output} \\
g = 0, \ gb = 1 & \quad g = 1, \ gb = 0 \\
a \rightarrow b & \quad 0 \rightarrow \text{strong 0} \\
g = 1, \ gb = 0 & \quad g = 1, \ gb = 0 \\
a \rightarrow b & \quad 1 \rightarrow \text{strong 1}
\end{align*}
\]
Tristates

- *Tristate buffer* produces Z when not enabled

<table>
<thead>
<tr>
<th>EN</th>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></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>
Nonrestoring Tristate

- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y

```
EN
  /
 /  
A   Y
  /
  EN
```
Tristate Inverter

- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output

EN = 0
Y = 'Z'

EN = 1
Y = \bar{A}
Multiplexers

- 2:1 multiplexer chooses between two inputs

<table>
<thead>
<tr>
<th>S</th>
<th>D1</th>
<th>D0</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Gate-Level Mux Design

- \( Y = SD_1 + \bar{SD}_0 \) (too many transistors)
- How many transistors are needed?
Transmission Gate Mux

- Nonrestoring mux uses two transmission gates
  - Only 4 transistors
Inverting Mux

- Inverting multiplexer
  - Use compound AOI22
  - Or pair of tristate inverters
  - Essentially the same thing
- Noninverting multiplexer adds an inverter
4:1 Multiplexer

- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 2:1 muxes
  - Or four tristates
D Latch

- When CLK = 1, latch is **transparent**
  - D flows through to Q like a buffer
- When CLK = 0, the latch is **opaque**
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

CLK

Latch

D

Q

CLK

D

Q

a.k.a. transparent latch or level-sensitive latch
D Latch Design

- Multiplexer chooses D or old Q
D Latch Operation

CLK = 1

CLK = 0

CLK

D

Q
D Flip-flop

- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. positive edge-triggered flip-flop, master-slave flip-flop
D Flip-flop Design

- Built from master and slave D latches

![D Flip-flop Circuit Diagram](image)
D Flip-flop Operation

- **CLK = 0**
  - D is applied to the circuit.
  - The output QM is inverted.
  - The output Q remains unchanged.

- **CLK = 1**
  - D is applied to the circuit.
  - The output QM is inverted.
  - The output Q is updated to the new value of D.

CLK waveform shows the timing of D and Q signals relative to the clock signal.
Race Condition

- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*
Nonoverlapping Clocks

- Nonoverlapping clocks can prevent races
  - As long as nonoverlap exceeds clock skew
- We may use them in this class for safe design
  - Industry manages skew more carefully instead

![Nonoverlapping Clocks Diagram]
Gate Layout

- Layout can be very time consuming
  - Design gates to fit together nicely
  - Build a library of standard cells

- Standard cell design methodology
  - $V_{DD}$ and GND should abut (standard height)
  - Adjacent gates should satisfy design rules
  - nMOS at bottom and pMOS at top
  - All gates include well and substrate contacts
Example: Inverter

(a) 

(b) Substrate T

Well Tap
Example: NAND3

- Horizontal N-diffusion and P-diffusion strips
- Vertical polysilicon gates
- Metal1 $V_{DD}$ rail at top
- Metal1 GND rail at bottom
- 32 $\lambda$ by 40 $\lambda$
Stick Diagrams

- Stick diagrams help plan layout quickly
  - Need not be to scale
  - Draw with color pencils or dry-erase markers

```
V_{DD} \rightarrow A
    \downarrow
    \times
    \uparrow
Y
GND \rightarrow INV
```

```
V_{DD} \rightarrow A
    \downarrow
    \times
    \uparrow
Y
GND \rightarrow NAND3
```

Legend:
- blue: metal1
- red: poly
- green: ndiff
- yellow: pdiff
- x: contact
A wiring track is the space required for a wire
- 4 $\lambda$ width, 4 $\lambda$ spacing from neighbor = 8 $\lambda$ pitch
Transistors also consume one wiring track
Well spacing

- Wells must surround transistors by 6 \( \lambda \)
  - Implies 12 \( \lambda \) between opposite transistor flavors
  - Leaves room for one wire track
Area Estimation

- Estimate area by counting wiring tracks
  - Multiply by 8 to express in $\lambda$
Example: O3AI

- Sketch a stick diagram for O3AI and estimate area

\[ Y = \text{NOT}(A + B + C) \cdot D \]
Summary

- A Brief History
- CMOS Gate Design
- Pass Transistors
- CMOS Latches & Flip-Flops
- Standard Cell Layouts
- Stick Diagrams