CMOS Technology and Logic Gates

Only 15,432,758 more mosfets to do...
Quality of Design

Quality of a hardware design primarily judged by:
- Price
- Performance
- Power and/or Energy

Other important metrics can include:
- Operating range
  - Temperature, voltage, background radiation
- Reliability
  - Mean-time between failures (MTBF)
- Form factor
  - Size, weight
- Flexibility
  - Tolerance to changes in specification

Need to understand implementation technology to understand tradeoffs among these attributes.
System-Level Impacts

Chips do not exist in a vacuum, e.g., 2/2.5G cell phone contains:

• RISC Application Processor (ARM)
• Digital Signal Processor
• SRAM/DRAM Chips
• Flash Memory Chips
• Analog Chips
  - E.g. headphone amplifier
• Radio Chips
• Power Management Subsystem
• Passive components
  - resistors, capacitors and inductors

Need to consider quality of a design in context of target system.
- E.g., design alternative might have twice the performance but require 10x off-chip memory bandwidth.

[Buss, ISSCC 2002]
Digital Technology Generations

- Electromechanical Relays
- Vacuum Tubes
- Bipolar Transistors
- CMOS/FET Transistors
  - ~10,000nm gates originally, now down to 90nm in production
  - scaling will stop somewhere below 30nm (over 100 billion trans./chip)
- Future:
  - 3D CMOS (10 trillion transistors/system?)
  - Carbon Nanotubes?
  - Molecular Electronics?

CMOS VLSI is the digital implementation technology of choice for the foreseeable future (next 10-20 years)
- Excellent energy versus delay characteristics
- High density of wires and transistors
- Monolithic manufacturing of devices and interconnect, cheap!
Abstraction Levels in Design

Gap too large to bridge in one step

but there are exceptions, e.g. magnetic compass
Hardware Design Abstraction Levels

- Application
- Algorithm
- Unit-Transaction Level (UTL) Model
- Guarded Atomic Actions (Bluespec)
- Register-Transfer Level (Verilog RTL)
- Gates
- Circuits
- Devices
- Physics
CMOS Fabrication

Starting wafer is pure silicon crystal.
Multiple process steps deposit new materials and etch existing layers using photolithography (light focused through masks).
Modern logic chips fabricated on 20cm (8”) wafers, ~100s chips/wafer.
Wafer sawed into separate chips after fabrication.
Chips then placed into packages (see packaging lecture later in course)
Basic CMOS Fabrication Steps

Growing silicon dioxide to serve as an insulator between layers deposited on the surface of the silicon wafer.

Doping the silicon substrate with acceptor and donor atoms to create p- and n-type diffusions that form isolating PN junctions and one plate of the MOS capacitor.

 Depositing material on the wafer to create masks, wires and the other plate of the MOS capacitor.

Etching deposited materials to create the appropriate geometric patterns.

Figures by MIT OCW.
Adapted from Maly, W.
Atlas of IC Technologies: An Introduction to VLSI Processes. (Ignore dimensions in figures)
Photoresist is spun onto wafer then exposed with UV light or X-rays through mask (or written with electron beam, no mask).

Performance Note: minimum feature size often determined by photoresist and etching process.

**Wet Etching**

**Dry Etching**
FET = Field-Effect Transistor

The four terminals of a fet (gate, source, drain and bulk) connect to conducting surfaces that generate a complicated set of electric fields in the channel region which depend on the relative voltages of each terminal.

A sufficiently strong vertical field will attract enough electrons to the surface to create a conducting n-type channel between the source and drain. If a channel exists, a horizontal field will cause a drift current from the drain to the source.
Multiple Levels of Interconnect

IBM photomicrograph (Si has been removed!)

Figure by MIT OCW.
• Design rules are an abstraction of the fabrication process that specify various geometric constraints on how different masks can be drawn.
• Design rules can be absolute measurements (e.g. in nm) or scaled to an abstract unit, the lambda. Lambda-based designs are scaled to the appropriate absolute units depending on the manufacturing process finally used.
Lambda-based Design Rules

One lambda ($\lambda$) = one half of the “minimum” mask dimension. Typically the length of a transistor channel is $2\lambda$. Usually all edges must be “on grid”, e.g., in the MOSIS scalable rules, all edges must be on a lambda grid.

The length of the transistor channel is usually the feature that sets the process technology name (e.g., 0.18$\mu$m has 0.18$\mu$m transistor length)

- diffusion (active)
- poly
- metal1
- contact
Static CMOS Gates

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

Figure by MIT OCW.
Simplified FET Model

Binary logic values represented by voltages:

"High" = Supply Voltage, "Low" = Ground Voltage

Supply Voltage = $V_{DD}$

PFET connects $S$ and $D$ when
\[ G = \text{"low"}=0 \text{V} \]

PFET only good at pulling up

NFET connects $D$ and $S$ when
\[ G = \text{"high"} = V_{DD} \]

NFET only good at pulling down

Ground = GND = 0V
Generic Static CMOS Gate

For every set of input logic values, either pullup or pulldown network makes connection to VDD or GND

- If both connected, power rails would be shorted together
- If neither connected, output would float (tristate logic)
NAND Gate

- When both A and B are high, output is low
- When either A or B is low, output is high
NOR Gate

- When both A and B are low, output is high
- When either A or B is high, output is low
NAND Gate Layout

Parallel PMOS Transistors

P-Diffusion (in N-well)

Poly wire connects PMOS & NMOS gates

Metal 1-Diffusion Contact

Output on Metal-1

Series NMOS Transistors

(A.B)
Methodical Gate Building

Goal is to create a logic function \( f(x_1, x_2, \ldots) \)
- must be inverting for single level of CMOS logic

Pull up network should connect output to \( V_{DD} \) when
\[
f(x_1, x_2, \ldots) = 1
\]

Pull down network should connect output to GND when
\[
\overline{f}(x_1, x_2, \ldots) = 1
\]

Because PMOS is conducting with low inputs, useful to write pullup as function of inverted inputs
\[
p(\overline{x}_1, \overline{x}_2, \ldots) = f(x_1, x_2, \ldots)
\]
Pullup is Dual of Pulldown Network

For NAND gate, \( f = (A \cdot B) \)
\[
Pulldown f = A \cdot B \\
Pullup p = f = A \cdot \overline{B} = \overline{A} + B
\]

(De Morgan’s Laws)

For NOR gate, \( f = (A + B) \)
\[
Pulldown f = A + B \\
Pullup p = f = \overline{A + B} = \overline{A} \cdot \overline{B}
\]

\[
\text{parallel switches form OR} \\
\text{series switches form AND} \\
(A \cdot B)
\]

\[
\text{series switches form OR} \\
\text{parallel switches form AND} \\
(A + B)
\]
More Complex Example

\[ f = (A+B).C \]

pullup \[ p = (A+B).C \]
\[ = (A+B) + \overline{C} \]
\[ = (A \cdot B) + \overline{C} \]

pulldown \[ \overline{f} = (A+B).C \]
Nearly all transistors in digital CMOS circuits have minimum $L$
  - but might use slightly longer $L$ to cut leakage in parts of modern circuits

- Can scale transistor $R$ and $C$ parameters by width $W$
- Effective $R$ scales linearly with $1/W$
  - $\sim 4k\Omega \mu m$ NMOS, $\sim 9k\Omega \mu m$ PMOS, in $0.25\mu m$ technology
- Gate capacitance scales linearly with $W$
  - $\sim 2fF/\mu m$
- Diffusion capacitance scales linearly with $W$
  - sum contributions from perimeter and area, $\sim 2fF/\mu m$
Transistor Delay

When one gate drives another, all capacitance on the node must be charged or discharged to change voltage to new state. Delay is proportional to driving resistance and connected capacitance.

$$\text{Delay} \sim R_{onN}(C_{\text{drainN}}+C_{\text{drainP}}+C_{\text{gateP}}+C_{\text{gateN}})$$
Gate Layout Tricks

Long transistors don't lay out well, and have a lot of parasitic diffusion capacitance.

“Fold” transistor to reduce perimeter diffusion cap.

Diffusion has high resistance.

Use multiple contacts to diffusion to reduce resistance.
More Layout Tricks

Fold whole stack, not individual transistors

Pack series transistors as close as possible to minimize diffusion R&C parasitics
Even More Complex Gates?

Can build arbitrarily complex logic function into one gate, e.g.

\[ F = -(A+B).(C+D).E.G+H.(J+K) \]

- But don’t want to:
  - Usually less total delay using a few smaller logic gates rather than one large complex gate
  - Only want to design and characterize a small library of gates

- What’s the best way to implement a given logic function?
Method of Logical Effort
(Sutherland and Sproul)

- Easy way to estimate delays in CMOS process.
- Indicates correct number of logic stages and transistor sizes.
- Based on simple RC approximations.
- Useful for back-of-the-envelope circuit design and to give insight into results of synthesis.
Technology Speed Parameter: $\tau$

Characterize process speed with single delay parameter: $\tau$

$\tau$ is delay of inverter driving same-sized inverter, with no parasitics other than gate

$\tau \sim 16-20\text{ps}$ for $0.25\mu\text{m}$ process
Gate Delay Components

Delay = Logical Effort \times Electrical Effort + Parasitic Delay

![Diagram of Logic Gate with Cin, Logic Gate, and Cout]

Logical Effort
- Complexity of logic function (Invert, NAND, NOR, etc)
- Define inverter has logical effort = 1
- Depends only on topology not transistor sizing

Electrical Effort
- Ratio of output capacitance to input capacitance \( C_{\text{out}}/C_{\text{in}} \)

Parasitic Delay
- Intrinsic self-loading of gate
- Independent of transistor sizes and output load
Logical Effort for Simple Gates

Define Logical Effort of Inverter = 1

For other gates, Logical Effort is ratio of logic gate's input cap. to inverter's input cap., when gate sized to give same current drive as inverter.

**Inverter**
- Input Cap = 3 units
- L.E. = 1 (definition)

**NAND**
- Input Cap = 4 units
- L.E. = 4/3

**NOR**
- Input Cap = 5 units
- L.E. = 5/3
Electrical Effort

\[
\text{Electrical Effort} = \frac{C_{\text{out}}}{C_{\text{in}}}
\]

Ratio of output load capacitance over input capacitance:

Usually, transistors have minimum length

Input and output capacitances can be measured in units of transistor gate widths
Parasitic Delay

Main cause is drain diffusion capacitances.
These scale with transistor width so P.D. independent of transistor sizes.
Useful approximation:
\[ C_{\text{gate}} \approx C_{\text{drain}} \]
For inverter:
Parasitic Delay \( \approx 1.0 \tau \)
Inverter Chain Delay

For each stage:

\[
\text{Delay} = \text{Logical Effort} \times \text{Electrical Effort} + \text{Parasitic Delay}
\]

\[
= 1.0 \text{ (definition)} \times 1.0 \text{ (in = out)} + 1.0 \text{ (drain C)}
\]

\[
= 2.0 \text{ units}
\]
Optimizing Circuit Paths

Path logical effort, $G = \prod g_i$  
$\quad$ (gi = L.E. stage i)

Path electrical effort, $H = \frac{C_{out}}{C_{in}}$  
$\quad$ (hi = E.E. stage i)

Parasitic delay, $P = \sum p_i$  
$\quad$ (pi = P.D. stage i)

Path effort, $F = GH$

Minimum delay when each of N stages has equal effort

$Min. \ D = NF^{1/N} + P$

i.e. $g_i \ h_i = F^{1/N}$
Optimal Number of Stages

Minimum delay when:

stage effort = logical effort $\times$ electrical effort $\approx 3.4-3.8$

- Some derivations have $e = 2.718\ldots$ as best stage effort - this ignores parasitics
- Broad optimum, stage efforts of 2.4-6.0 within 15-20% of minimum

Fan-out-of-four (FO4) is convenient design size ($\sim 5\tau$)

FO4 delay: Delay of inverter driving four copies of itself
Using Logical Effort in Design

For given function, pick candidate gate topology

Determine optimal stage effort
- equal for all stages

Starting at last gate
- output load is known
- logical effort is known (from gate topology)
- calculate transistor size to give required stage effort
- gives output load for preceding stage
- lather, rinse, repeat...

Can modify stage efforts up or down to reduce area, power, or to fit fixed set of library cells
- optimal sizing has broad optimum

In 6.884, we’ll just let synthesis tool handle gate sizing, but it’s useful to know why the tool makes certain decisions.