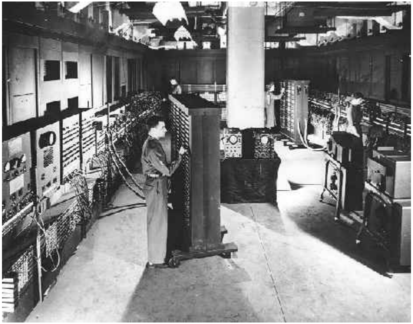
# Embedded System Design

Stephen A. Edwards

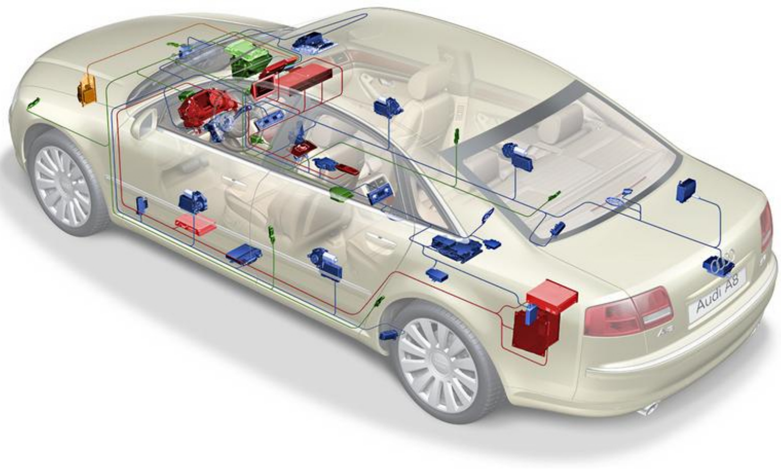
Columbia University

Spring 2015

# Spot the Computer



# Cars These Days...



# Embedded Systems: Ubiquitous Computers

iPhone

Laser Keyboard


Nikon D300

Video Watch

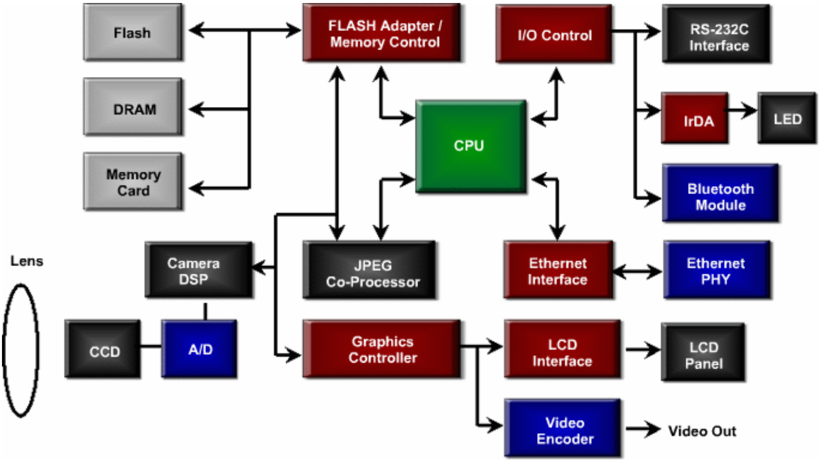

GPS


Playstation 3

PC Keyboard


SD Card

# Inside a Digital Camera



# Want an Optimal Device that Meets Constraints On

Price

Performance

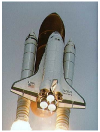
Power

Maintainability

Functionality

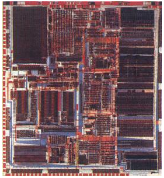
Size

Time-to-market

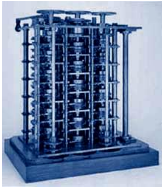


Safety

# Embedded System Technologies



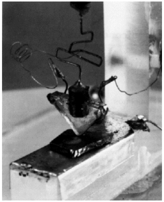
Integrated Circuits



Processing elements

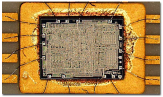
Design tools

# IC Technology

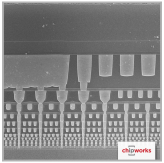


1947: First transistor (Shockley, Bell Labs)

1958: First integrated circuit (Kilby, TI)



1971: First microprocessor (4004: Intel)

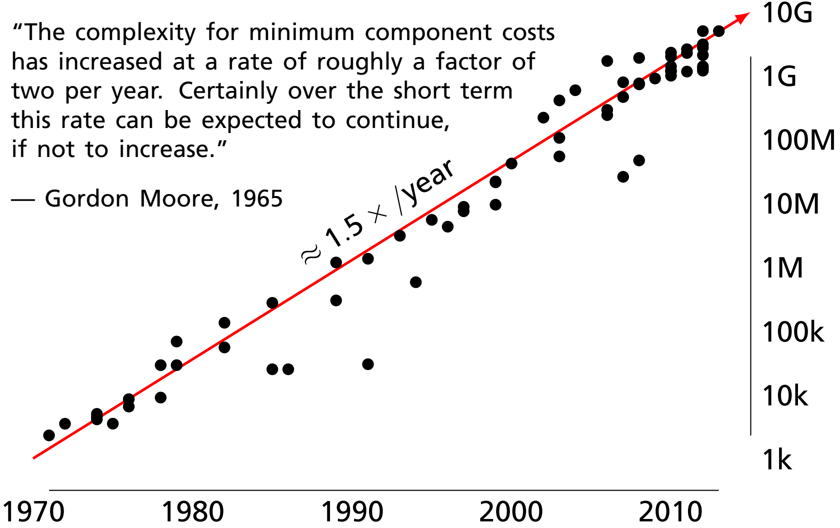


2015: 14 nm features, 13 layers (Intel, Broadwell)

# Moore's Law: Transistors per chip

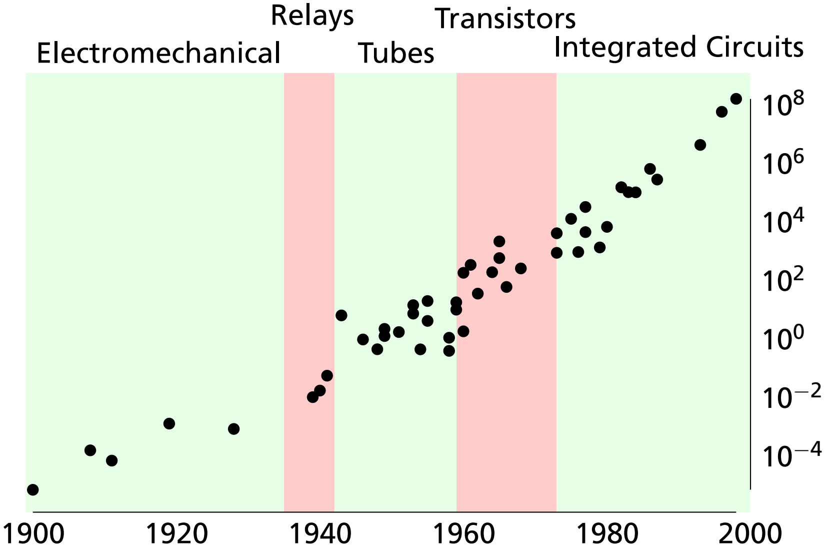
"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase."

— Gordon Moore, 1965



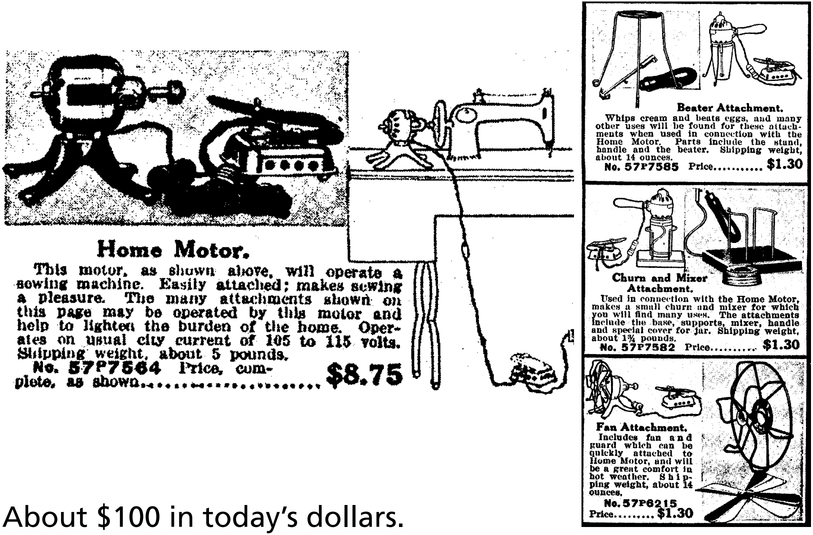

Source: Intel/Wikipedia

# $1000 Buys You This Many Cycles per Second




Source: Ray Kurzweil, *The Age of Spiritual Machines*

# 1918 Sears Roebuck Catalog



**Home Motor.**

This motor, as shown above, will operate a sewing machine. Easily attached; makes sewing a pleasure. The many attachments shown on this page may be operated by this motor and help to lighten the burden of the home. Operates on usual city current of 105 to 115 volts. Shipping weight, about 5 pounds.

No. 57P7564 Price, complete, as shown.................... $8.75

**Beater Attachment.**

Whips cream and beats eggs, and many other uses will be found for these attachments when used in connection with the Home Motor. Parts include the stand, handle and the beater. Shipping weight, about 14 ounces.

No. 57P7585 Price.......... $1.30

**Churn and Mixer Attachment.**

Used in connection with the Home Motor, makes a small churn and mixer for which you will find many uses. The attachments include the base, supports, mixer, handle and special cover for jar. Shipping weight, about 1¾ pounds.

No. 57P7582 Price.......... $1.30

**Fan Attachment.**

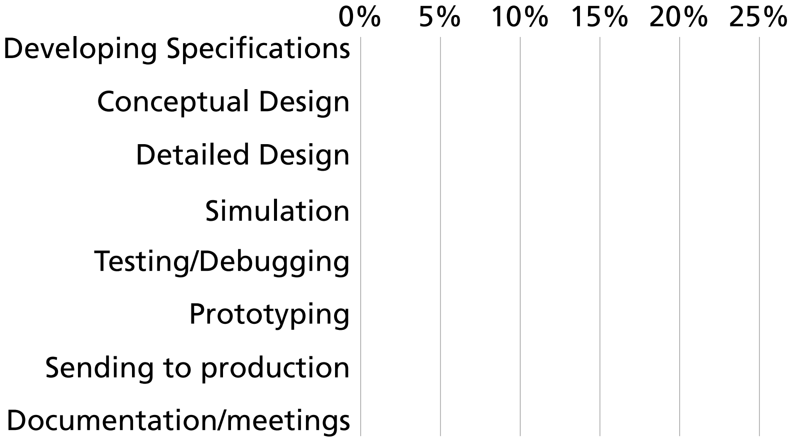
Includes fan and guard which can be quickly attached to Home Motor, and will be a great comfort in hot weather. Shipping weight, about 14 ounces.

No. 57P6215 Price......... $1.30

About $100 in today's dollars.

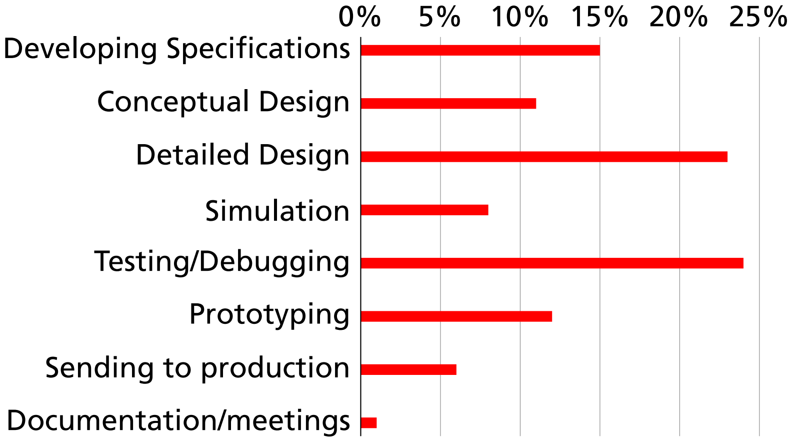
From Donald Norman, *The Invisible Computer*, 1998.

# What Percentage of Time Do You Spend...




Source: 2009 Embedded Market Study

# What Percentage of Time Do You Spend...




Source: 2009 Embedded Market Study

# Does Your Current Project Contain FPGAs?

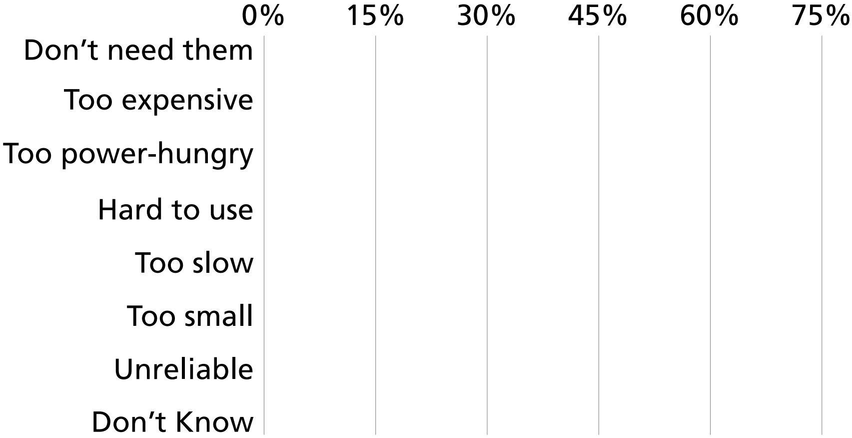
# Does Your Current Project Contain FPGAs?

45% Yes

55% No

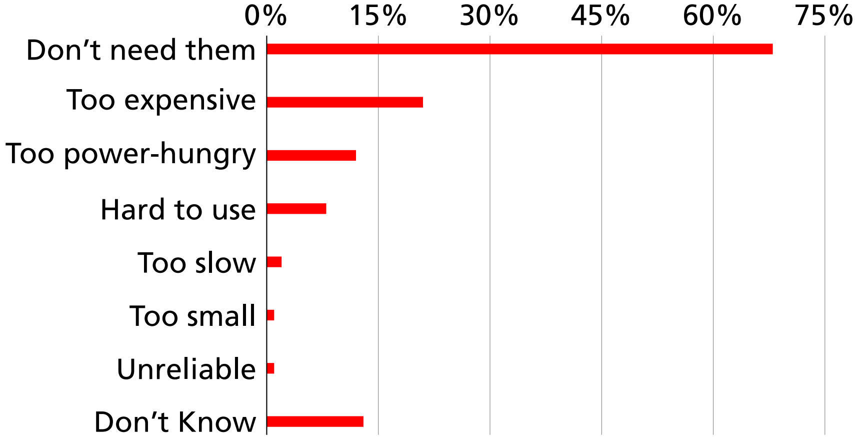
Source: 2009 Embedded Market Study

# Why Won't Your *Next* Project Use FPGAs?



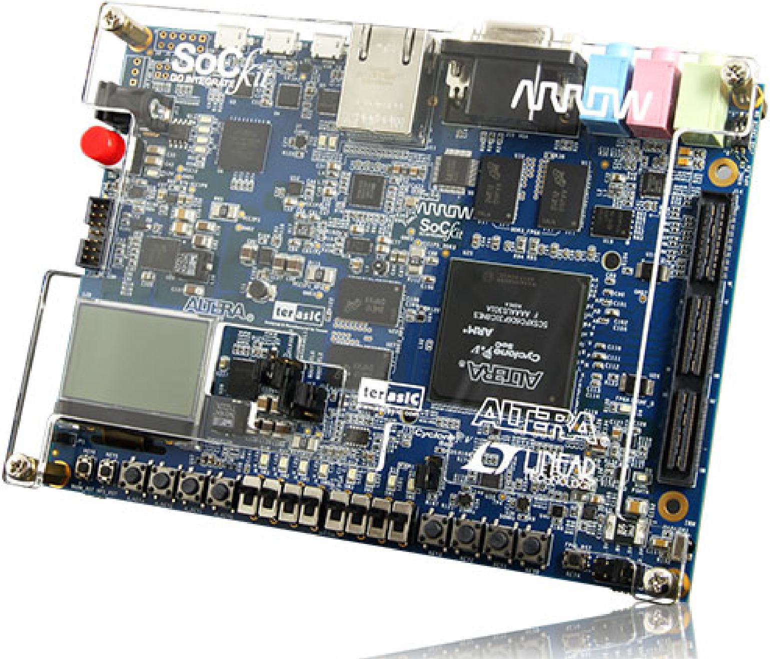

Source: 2009 Embedded Market Study

# Why Won't Your *Next* Project Use FPGAs?

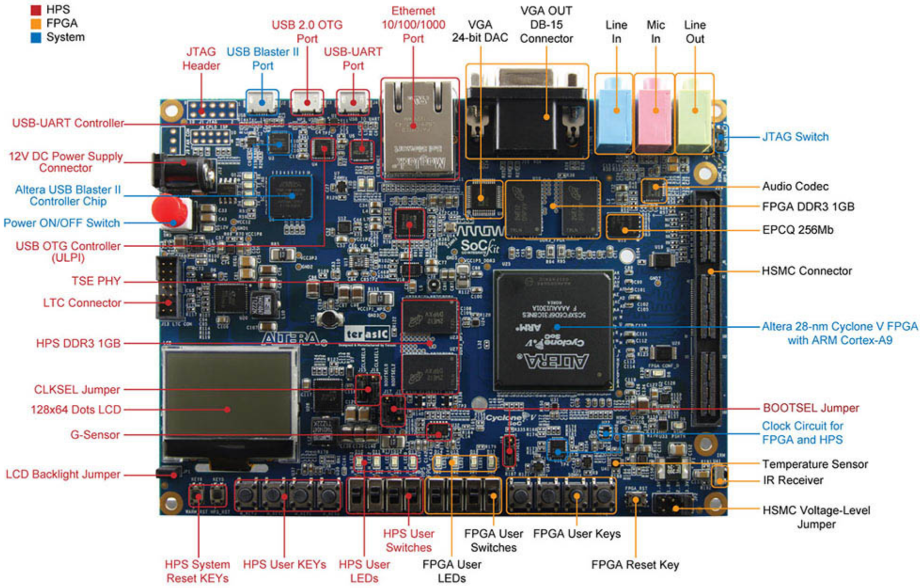



Source: 2009 Embedded Market Study

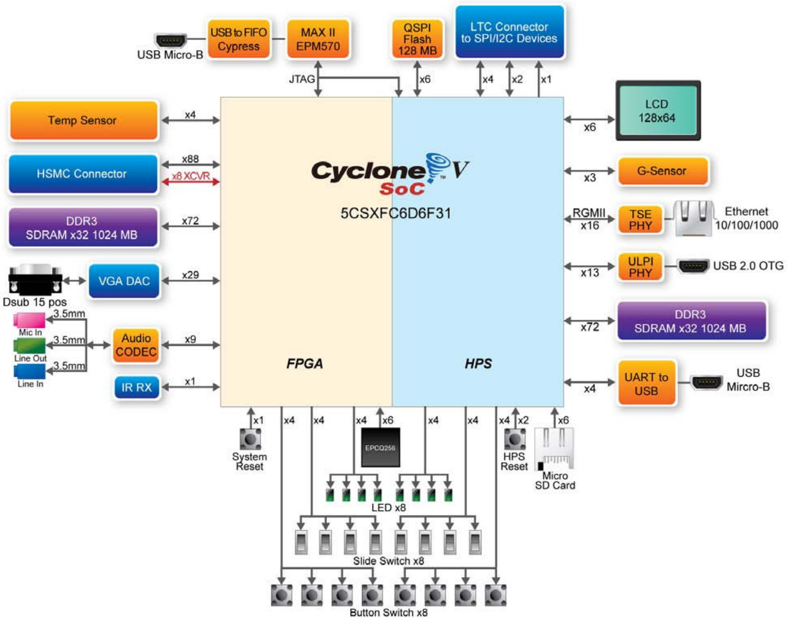
# Your Nemesis: The SoCKit Board



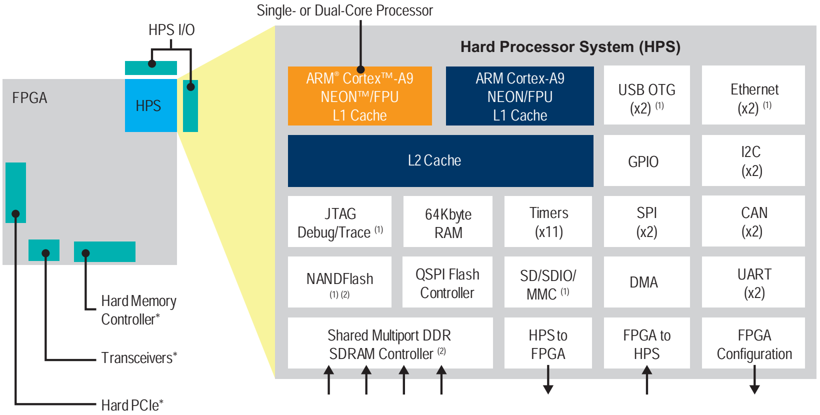
# Components and Peripherals



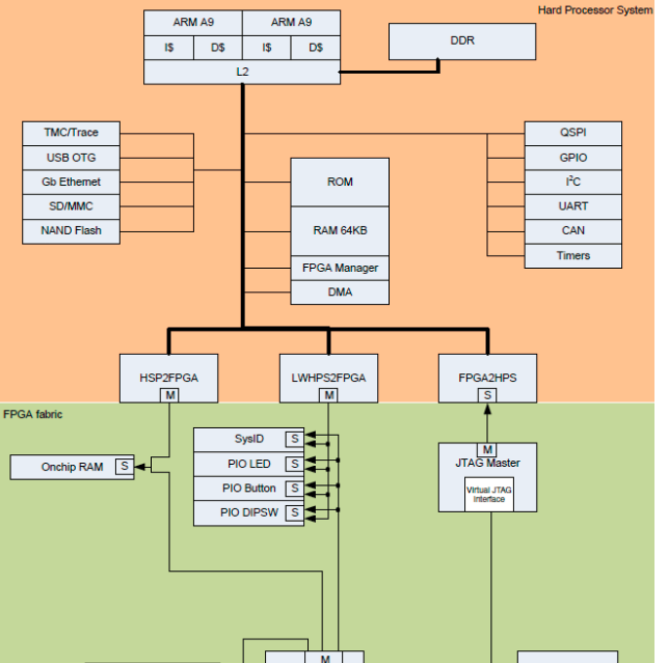
# Dual ARM Cortex-A9 and Programmable Logic



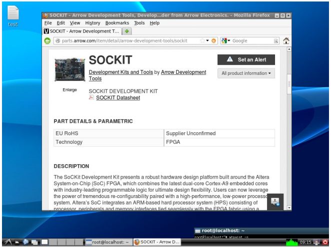
# Inside the Cyclone V: Dual ARM processors + FPGA

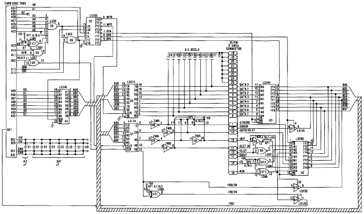


# An Example System



# Linux + Custom Hardware



# Class Structure

Three Introductory Labs: 2 weeks each

Work in pairs

1. Hardware: Access, modify, and display memory
2. Software: A simple Internet chat client
3. HW + SW: A video bouncing ball

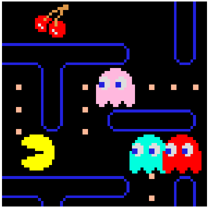
---

The project: **Design-your-own**

Work in groups of four

Broadly: C + SystemVerilog + peripheral(s)

# Broad Project Idea: Video Game

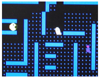
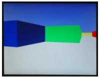
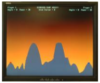
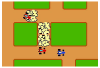
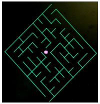
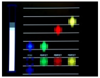
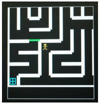


Implement graphics in custom hardware

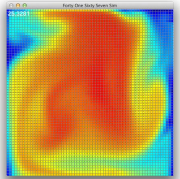
Put game logic in software

Interface with USB HID (Joystick, etc.)

E.g., Pac-man, 2.5D maze game, tank, worms

# Broad Project Idea: Computational Accelerator



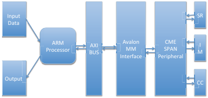

Pick a computationally intensive algorithm

Implement its core in custom hardware

Write software and device drivers that pass data to and from the accelerator

E.g., Smoke simulator, inverse kinematics for robotics, Bitcoin miner

# Broad Project Idea: Network Accelerator




Pick a simple network processing problem, e.g., from finance

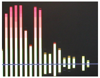
Implement part of existing software algorithm in hardware

Interface hardware with network controller; processor

E.g., TCP/IP, tickerplant, margin calculations, memcached, FIX protocol parser

# More Ideas


Digital tone control

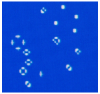

Spectrum analyzer

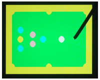

Internet radio

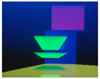
Speech Synthesizer


MIDI synthesizer

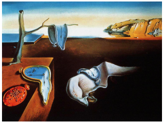

Accelerated JPEG


Game of Life

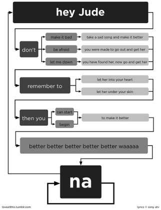

Pool game


Real-time ray tracer

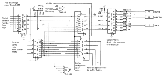
# The Three Main Challenges of Embedded Systems



Coping with Real-World Sensor Data

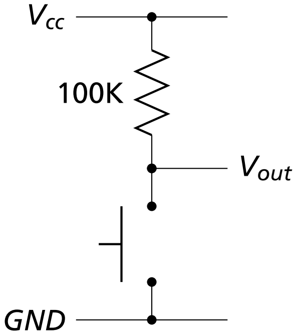



Algorithm Design

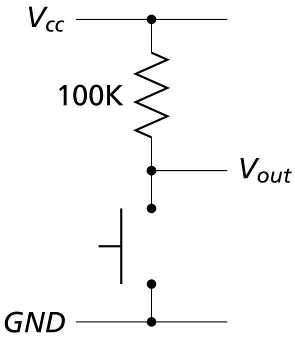


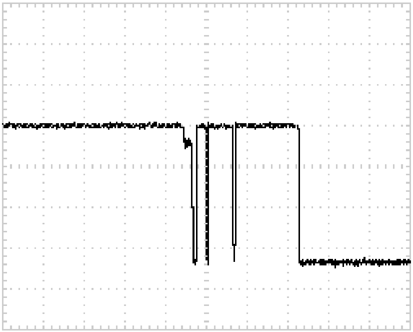
Implementation Details

# What Happens When You Press the Switch?

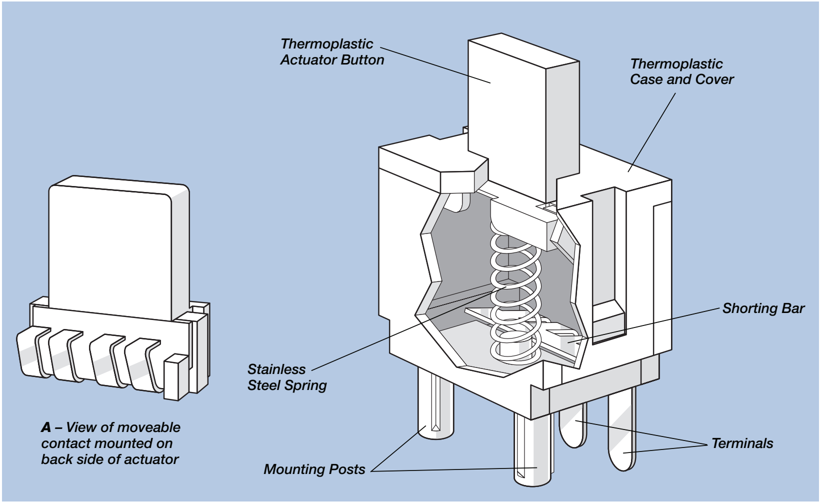


# What Happens When You Press the Switch?



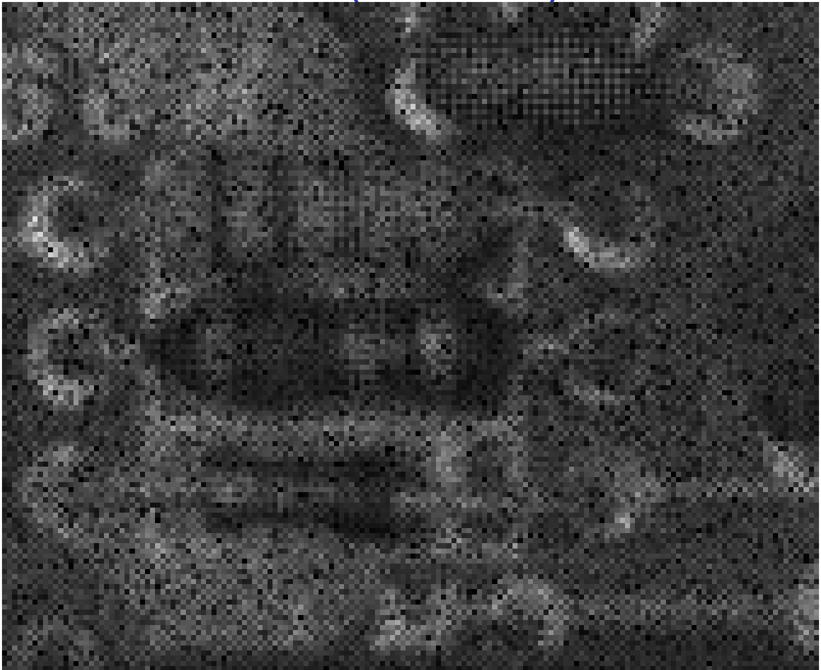



# Inside a Pushbutton Switch

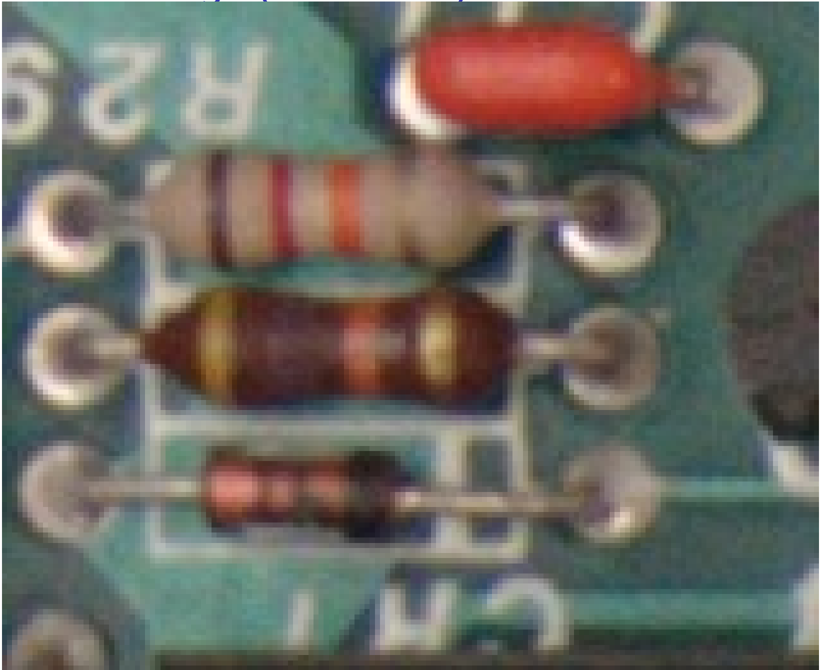



Source: Cherry CS series data sheet

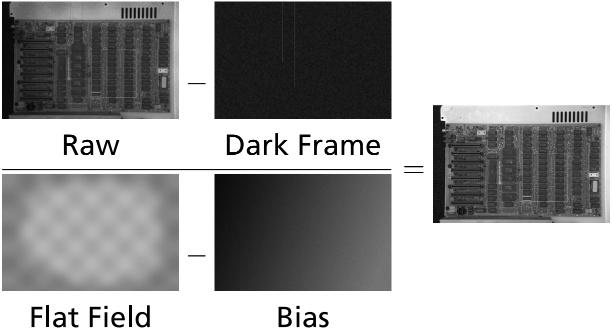
# Raw Data from a CCD (zoomed in)

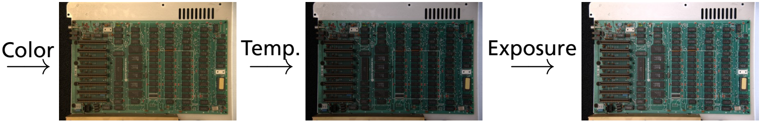


# Corrected Image (zoomed in)

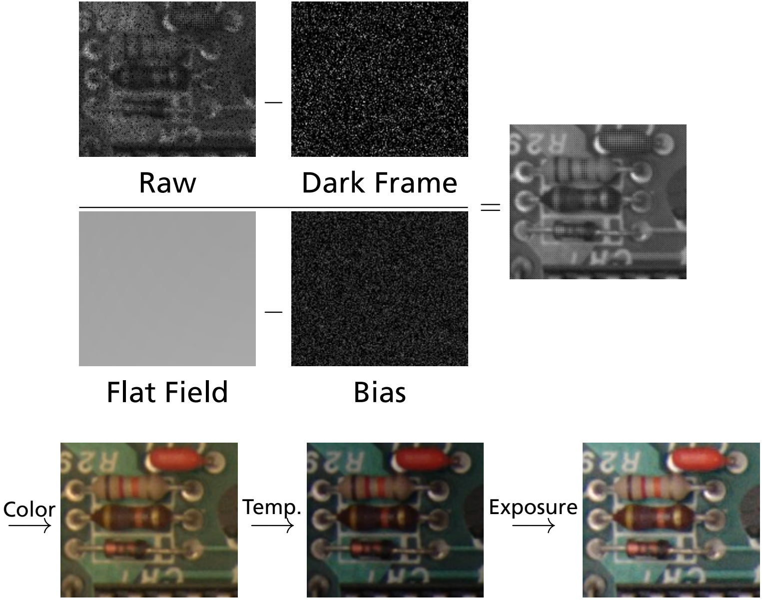


# Correcting Data from CCDs

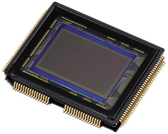


# Correcting Data from CCDs



# Where Does This Noise Come From?

Nikon D300: 23.6 mm $\times$ 15.8 mm
12.3 megapixel CMOS sensor

Pixels are 5.5 $\mu$m on a side

A/D sampling of 12 bits per pixel measures

| ISO: | LO | 200 | 400 | 800 | 1600 | 3200 |
|---|---|---|---|---|---|---|
| G | 7.1 | 5.5 | 2.7 | 1.3 | 0.65 | 0.33 |
| B | 5.8 | 4.6 | 2.3 | 1.1 | 0.55 | 0.27 |
| R | 4.7 | 4.5 | 2.2 | 1.1 | 0.54 | 0.26 |

The units: *electrons per ADU* (digital unit).

Emil Martinec, *A comparison of the Nikon D300 and Canon 40D sensors*, 2007.

# Development Plan

1. Obtain some representative raw sensor data
2. Develop an algorithmic prototype using your favorite language (e.g., Java, C, Matlab)
3. Plan how to implement it
4. Implement while constantly testing