EECS 373

Introduction to Embedded System Design

Website: https://www.eecs.umich.edu/courses/eecs373/

Robert Dick
University of Michigan

Lecture 1: Introduction, ARM ISA

11 January 2024

Many slides based on slides from other teachers, e.g., Mark Brehob and Alanson Sample.

Teacher

Robert Dick
http://robertdick.org/
dickrp@umich.edu

- EECS Associate Professor.
- Co-founder, CEO of Stryd athletic wearable electronics company (www.stryd.com).
- Visiting Professor at Tsinghua University.
- Graduate studies at Princeton.
- Visiting Researcher at NEC Labs, America, where technology went into their smartphones.
- 100 research papers on embedded system design, cited by 11,000 other papers.

Lab instructor

Matthew Smith

- matsmith@umich.edu
- Head lab instructor.
- 20 years of 373 experience.
- He has probably seen it before… but he'll make you figure it out yourself, anyway.

TAs




- GSI: Guthrie Tabios <tabiosg@umich.edu>
- IA: Anna Huang <ahuangg@umich.edu>
- IA: Joseph Maffetone <jmaff@umich.edu>
- IA: Rajin Nagpal <rajinn@umich.edu>

Course goals

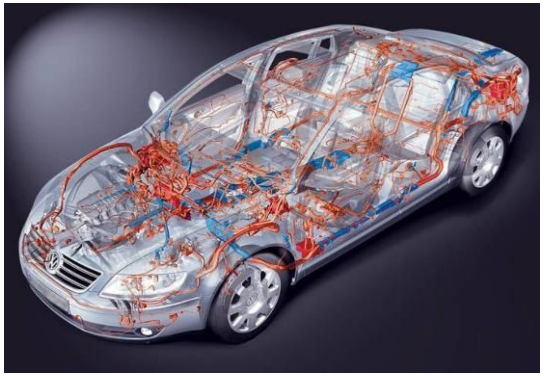
Teach you

- **embedded system design**,
- **debugging complex systems**, and
- a little communication and marketing.

- For some, provide a head start on a new product or research idea.

What is an embedded system?

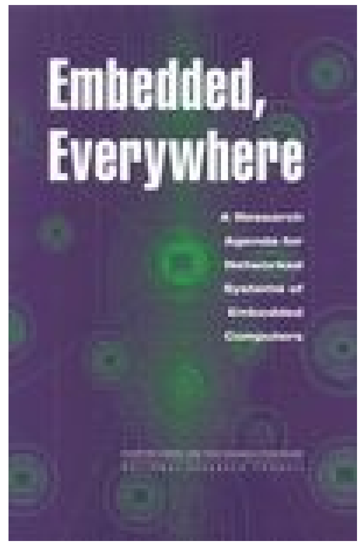
An (application-specific) computer
within something else
that is not generally regarded as a computer.

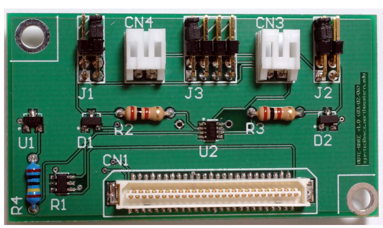



NOKIA


eZ430-Chronos
Wireless Development Tool
Texas Instruments
MSP430


iPod nano
iPod classic
iPod touch


Embedded, Everywhere
A Research Agenda for Networked Systems of Embedded Computers


CN4
CN3
J1
J3
J2
R2
R3
D1
D2
U1
U2
CN1
R4
R1

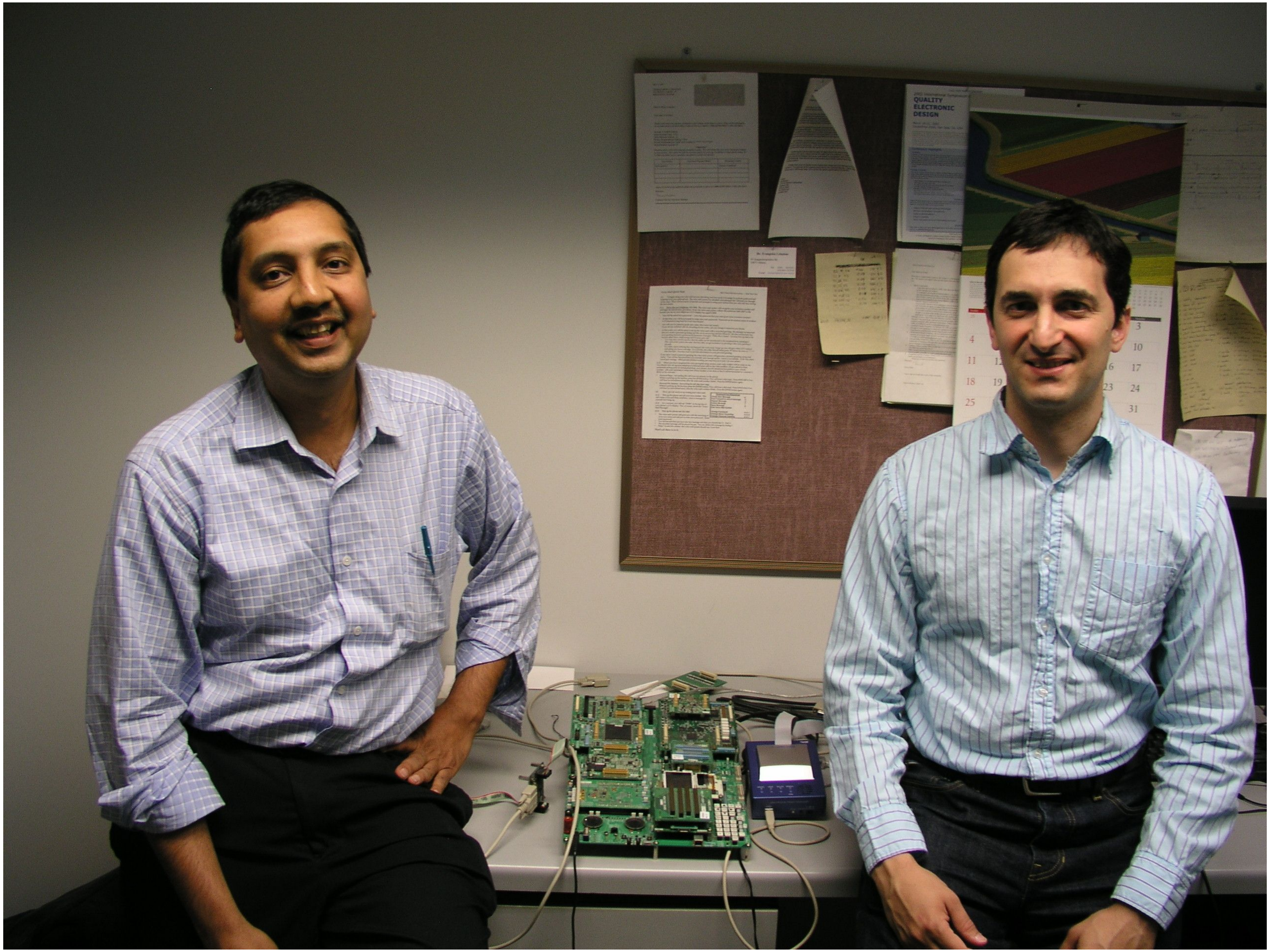
Embedded systems market

- Dominates general-purpose computing market in volume.
- Similar in monetary size to general-purpose computing market.
- Historically grows at 15% per year, 10% for general-purpose computing.
- Car example: half of value in embedded electronics, from zero a few decades ago.

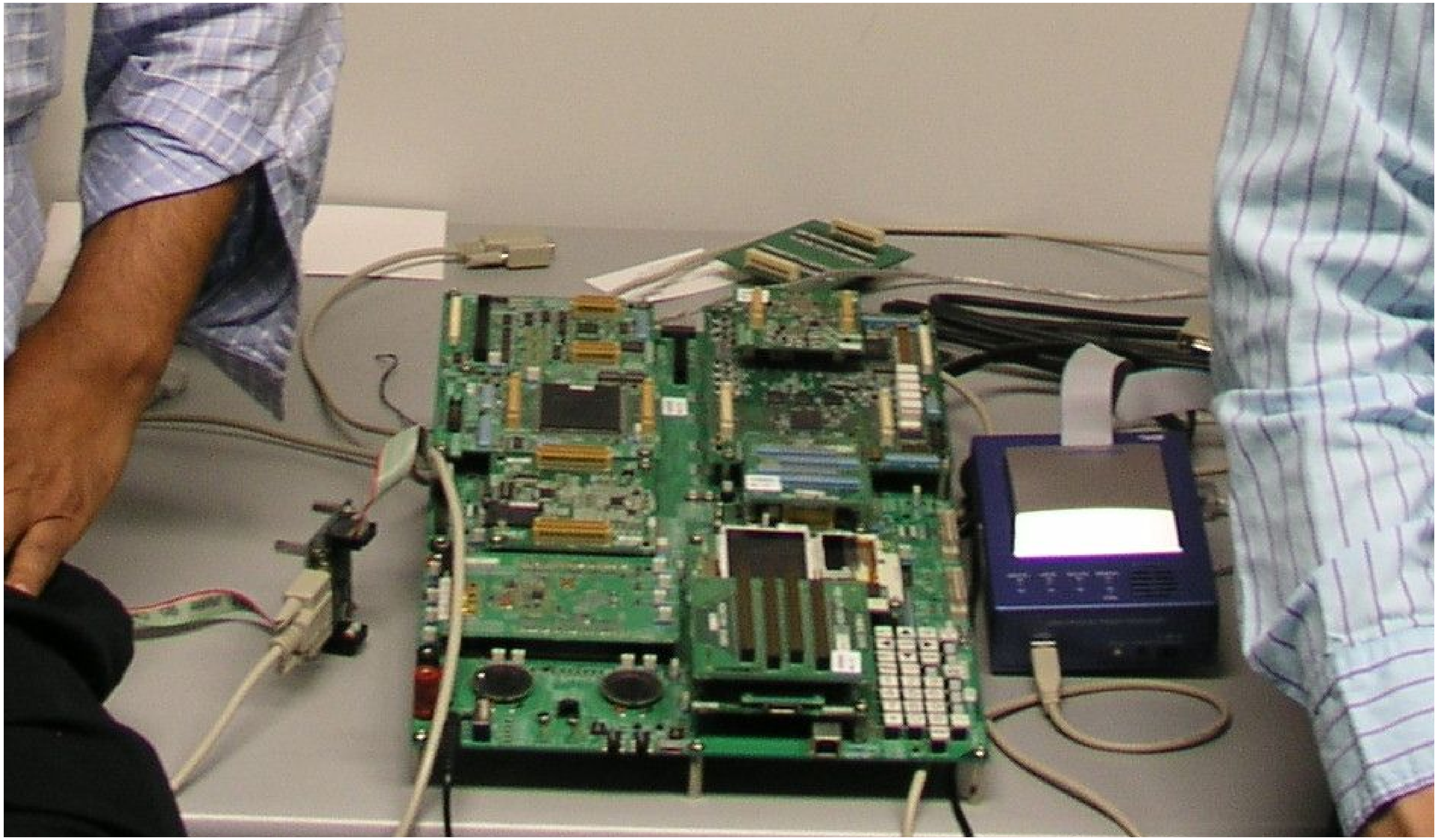
Common requirements

- Timely (hard real-time)
- Wireless
- Reliable
- First time correct
- Rapidly implemented
- Low price
- High performance
- Low power
- Embodying deep domain knowledge
- Beautiful

Example design process



Example design process



What is driving the
embedded everywhere trend?

Technology trends

Application innovations

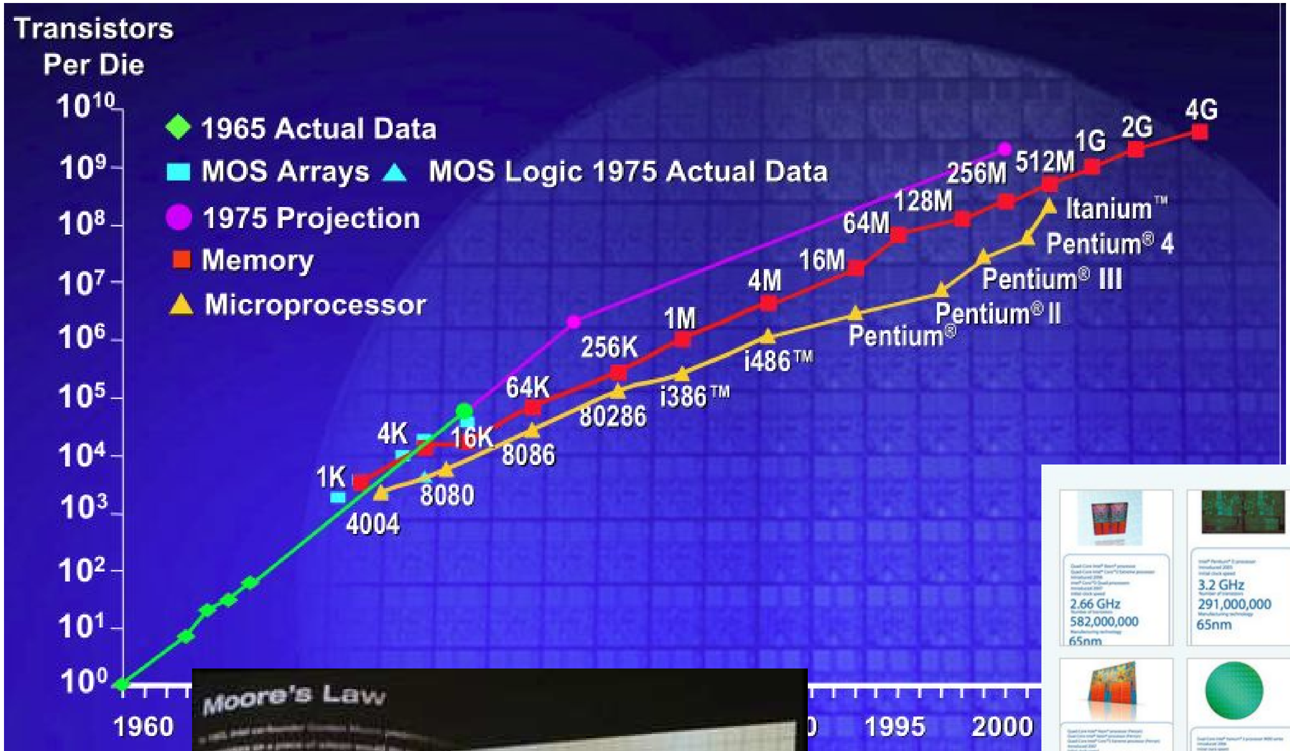
Outline

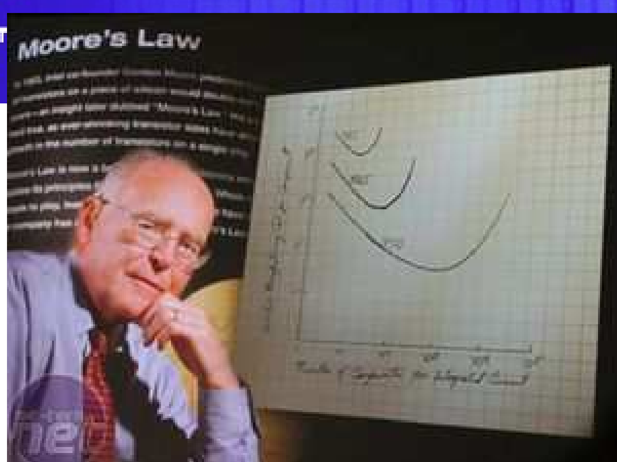
Technology Trends

Course Description/Overview

Review, Tools Overview, ISA

Moore's Law (a statement about economics): IC transistor count doubles every 18-24 months





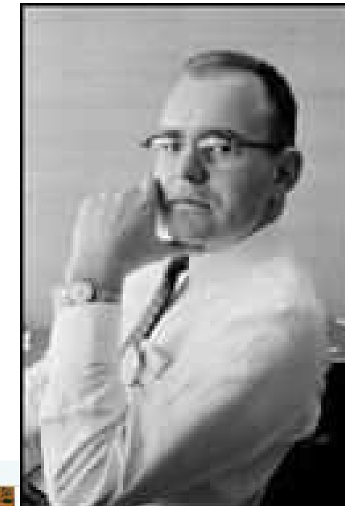
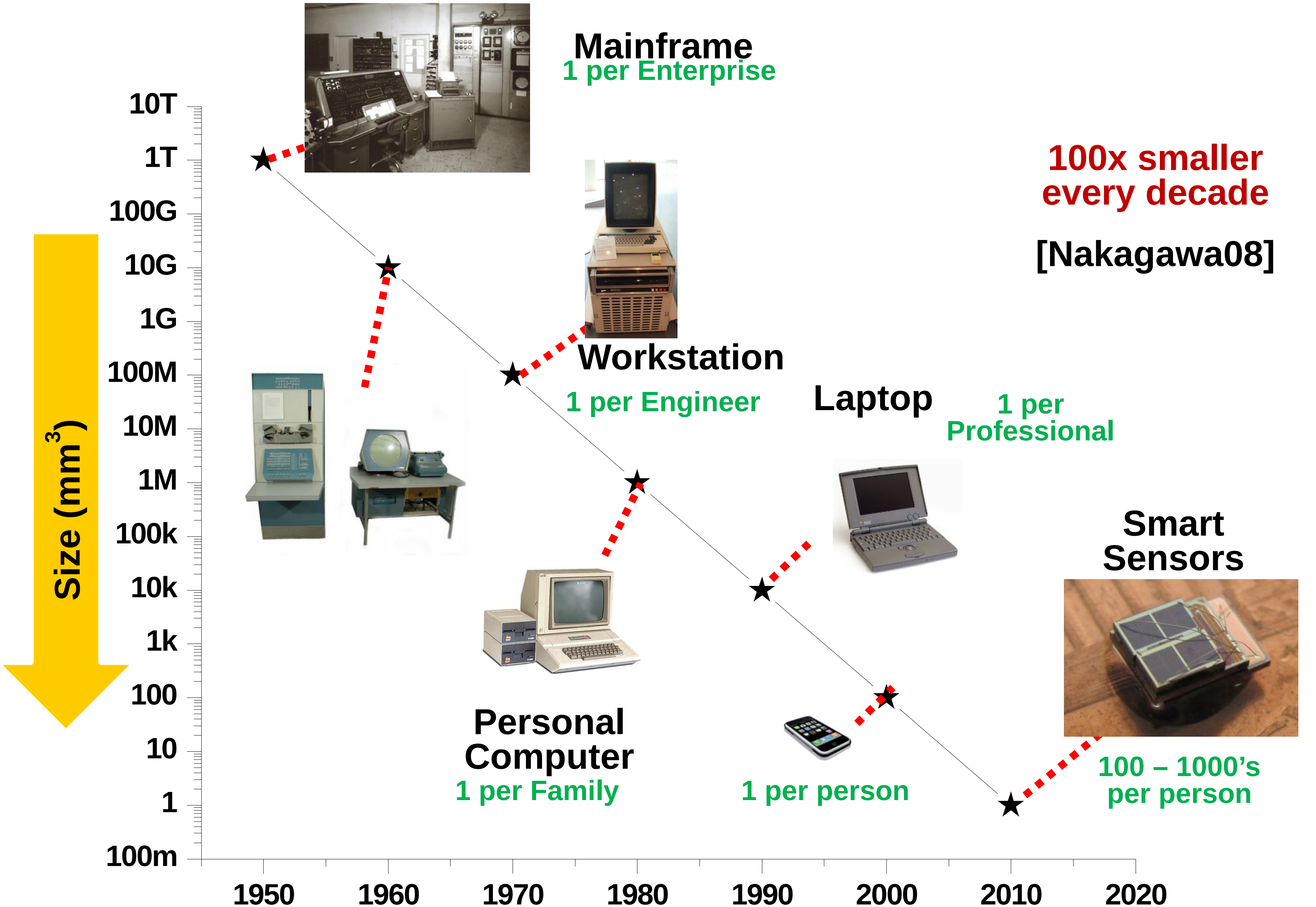
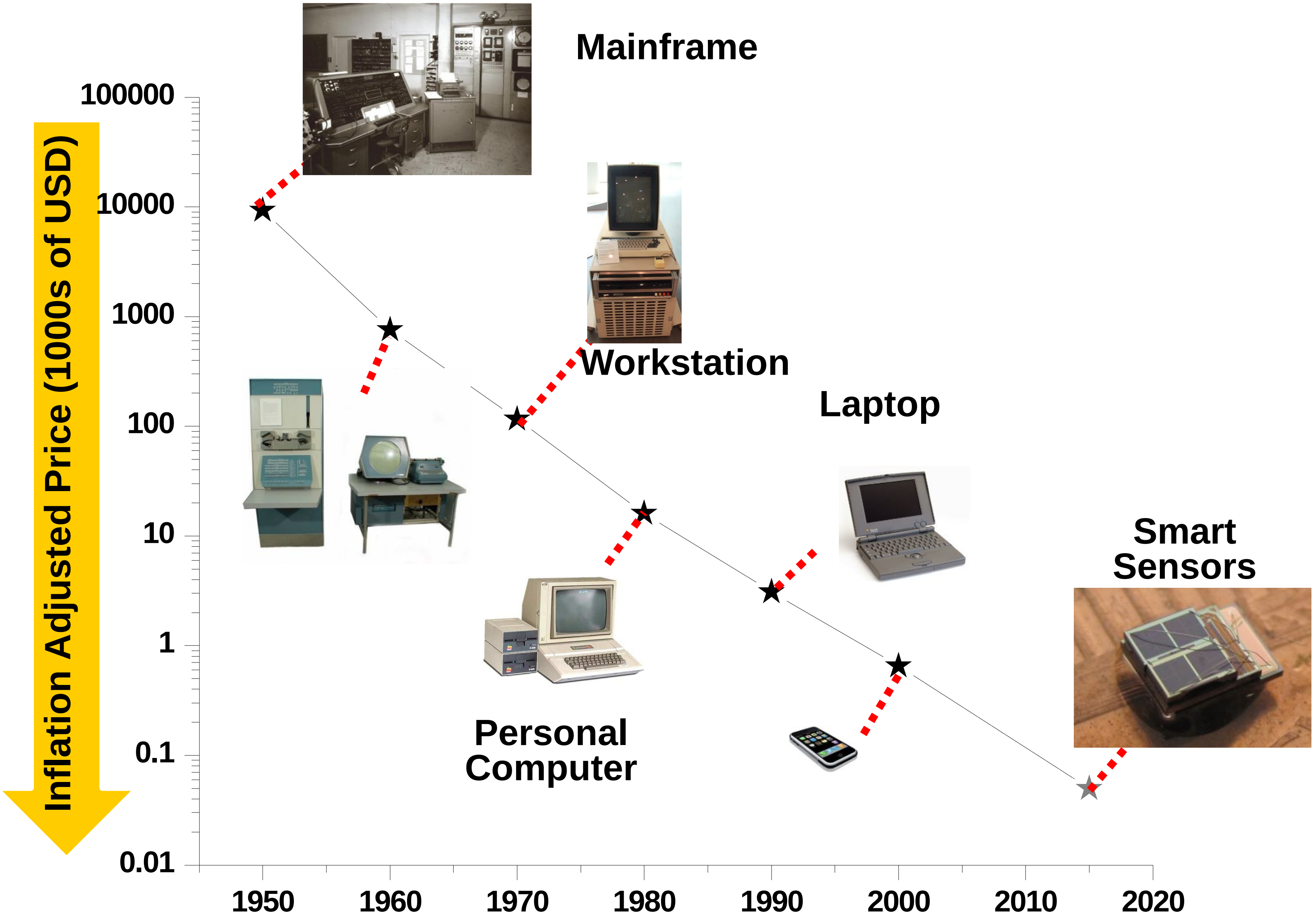


Photo Credit: Intel

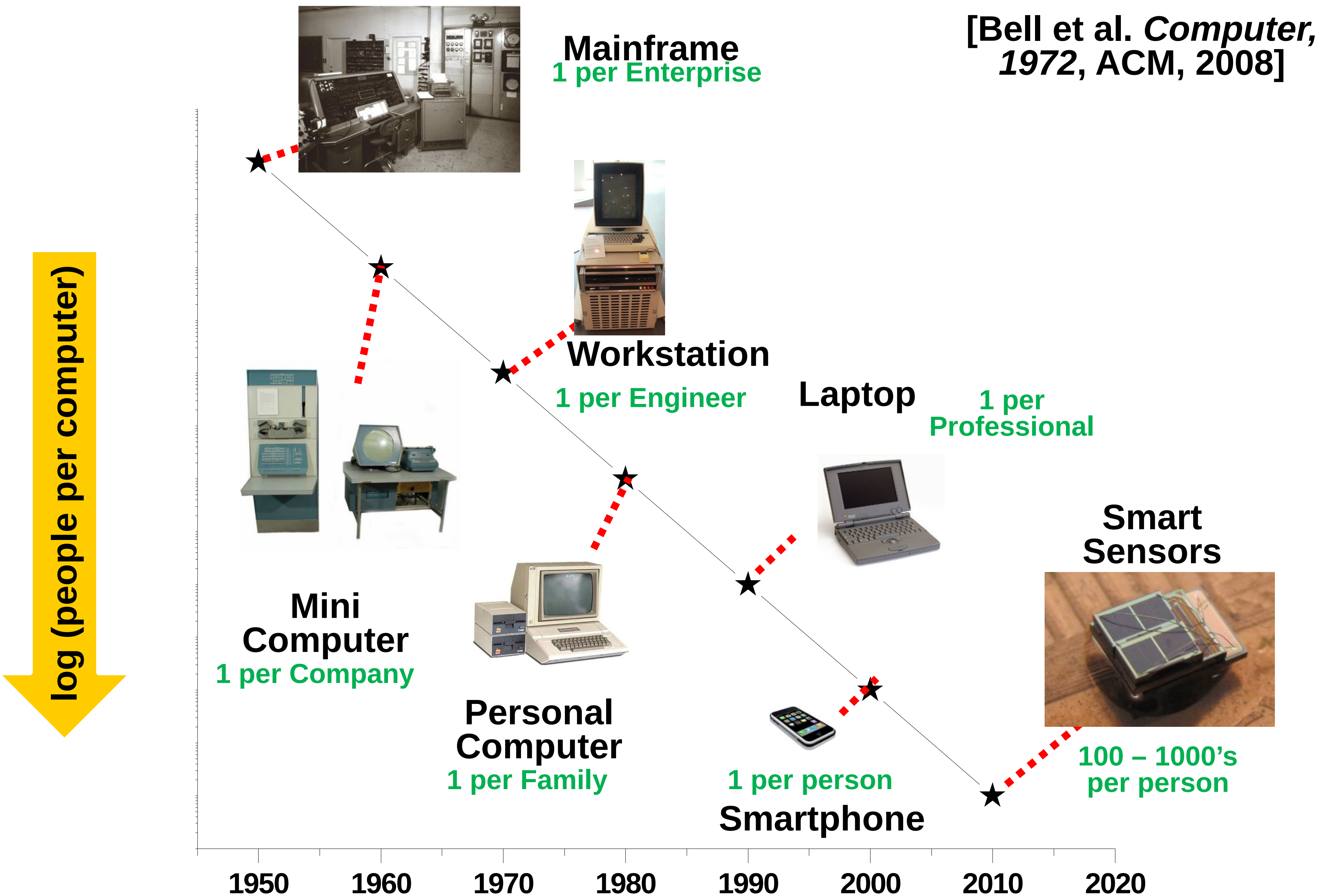
Computer volume shrinks by 100x every decade



Price falls dramatically, and enables new applications



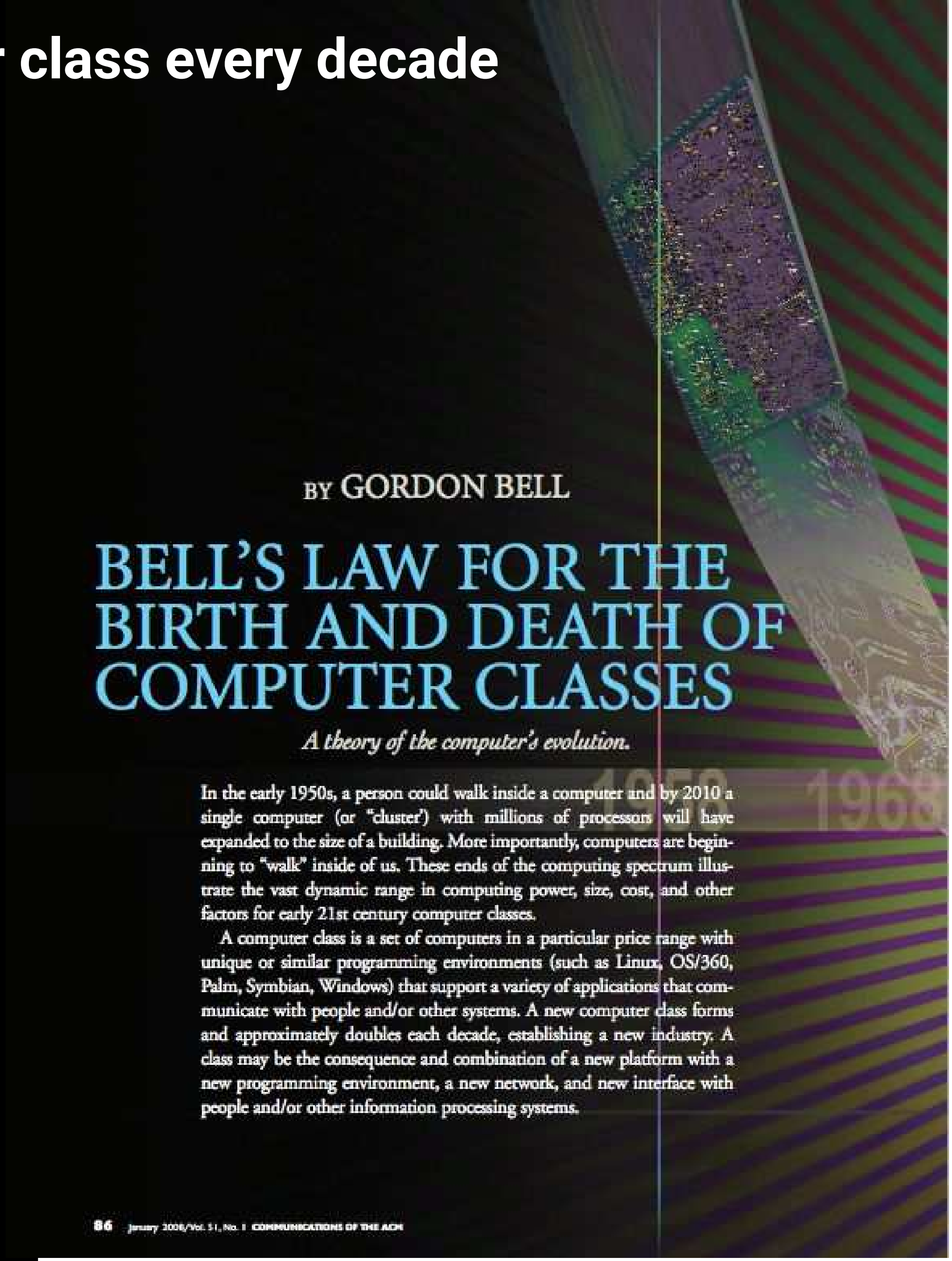
Computers per person



Bell's Law: A new computer class every decade

"Roughly every decade a new, lower priced computer class forms based on a new programming platform, network, and interface resulting in new usage and the establishment of a new industry."

- Gordon Bell [1972,2008]



BY GORDON BELL

BELL'S LAW FOR THE BIRTH AND DEATH OF COMPUTER CLASSES

A theory of the computer's evolution.

In the early 1950s, a person could walk inside a computer and by 2010 a single computer (or "cluster") with millions of processors will have expanded to the size of a building. More importantly, computers are beginning to "walk" inside of us. These ends of the computing spectrum illustrate the vast dynamic range in computing power, size, cost, and other factors for early 21st century computer classes.

A computer class is a set of computers in a particular price range with unique or similar programming environments (such as Linux, OS/360, Palm, Symbian, Windows) that support a variety of applications that communicate with people and/or other systems. A new computer class forms and approximately doubles each decade, establishing a new industry. A class may be the consequence and combination of a new platform with a new programming environment, a new network, and new interface with people and/or other information processing systems.

What is driving Bell's Law?

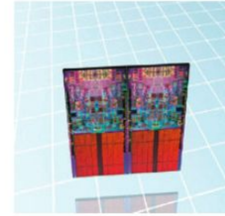
Technology Scaling

- Moore's Law
  - Made transistors cheap
- Dennard Scaling
  - Made them fast
  - But power density undermines
- Result
  - Fixed transistor count
    - Exponentially lower cost
    - Exponentially lower power
  - Small, cheap, and low-power
    - Microcontrollers
    - Sensors
    - Memory
    - Radios

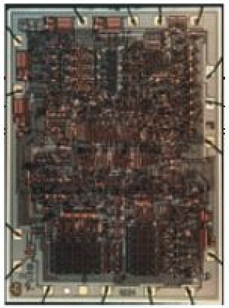
Technology Innovations

- MEMS technology
  - Micro-fabricated sensors
- New memories
  - New cell structures (11T)
  - New tech (FeRAM, FinFET)
- Near-threshold computing
  - Minimize active power
  - Minimize static power
- New wireless systems
  - Radio architectures
  - Modulation schemes
- Energy harvesting

Corollary to Moore's Law

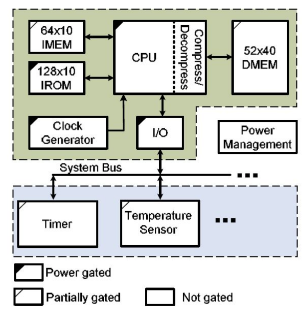


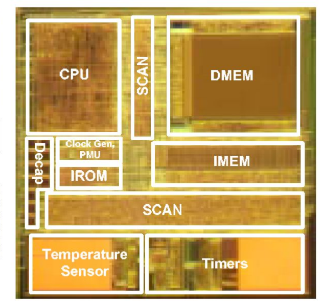
Quad-Core Intel® Xeon® processor
Quad-Core Intel® Core™2 Extreme processor
Introduced 2006
Intel® Core™2 Quad processors
Introduced 2007
Initial clock speed
2.66 GHz
Number of transistors
582,000,000
Manufacturing technology
65nm



Intel® 4004 processor
Introduced 1971
Initial clock speed
108 KHz
Number of transistors
2,300
Manufacturing technology
10µ

15x size decrease
40x transistors
55x smaller λ



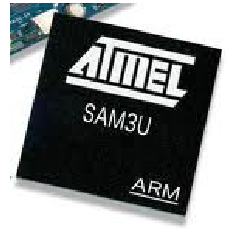


UMich Phoenix Processor
Introduced 2008
Initial clock speed
106 kHz @ 0.5V Vdd
Number of transistors
92,499
Manufacturing technology
0.18 µ

Photo credits: Intel, U. Michigan

Broad availability of inexpensive, low-power, 32-bit MCUs (with enough memory to do interesting things)

Actel®
POWER MATTERS



ATMEL®

CYPRESS
PERFORM

DUST
NETWORKS

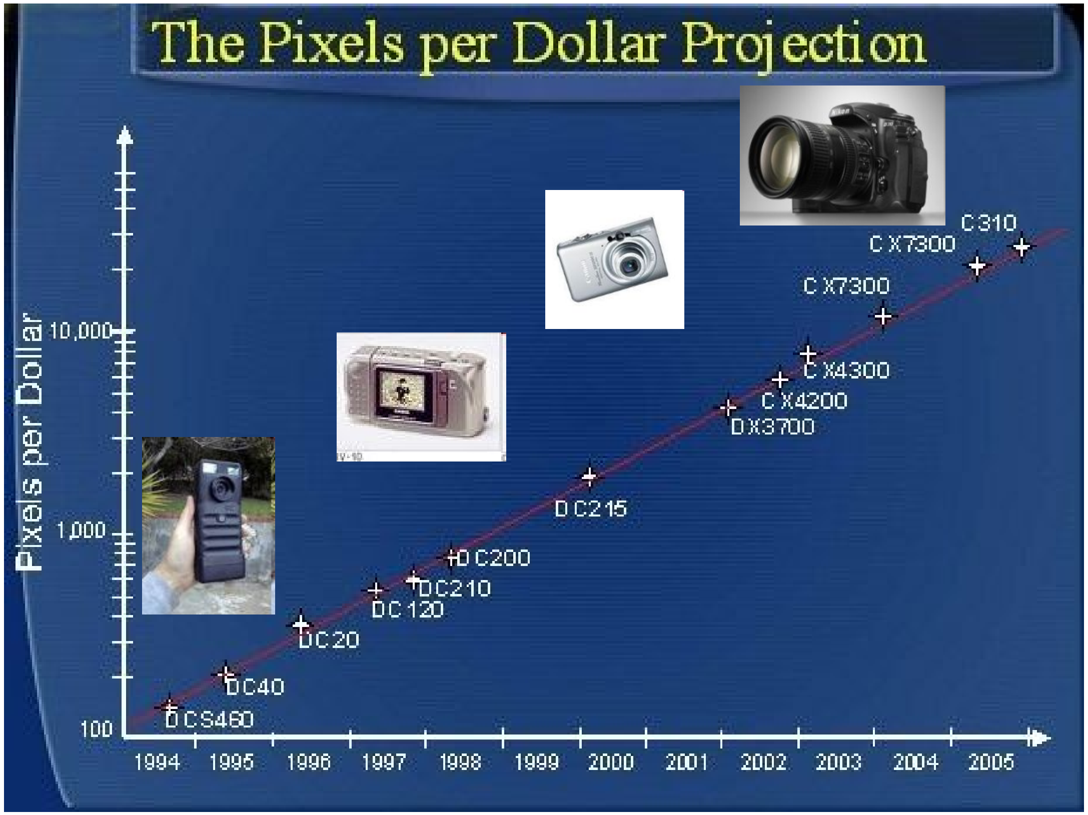
ember

NXP

ST

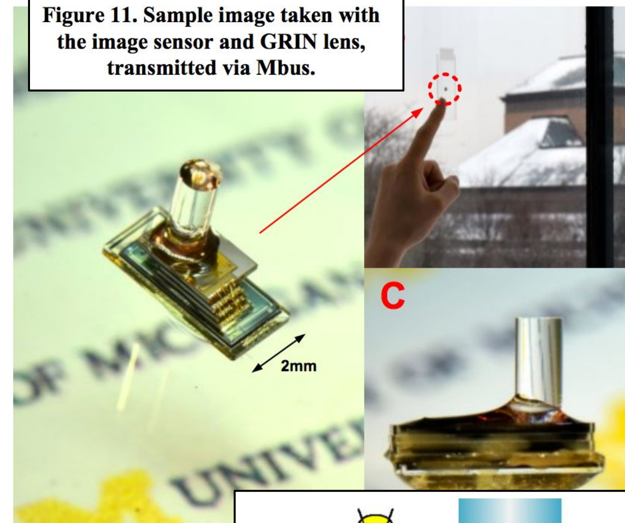
TEXAS INSTRUMENTS

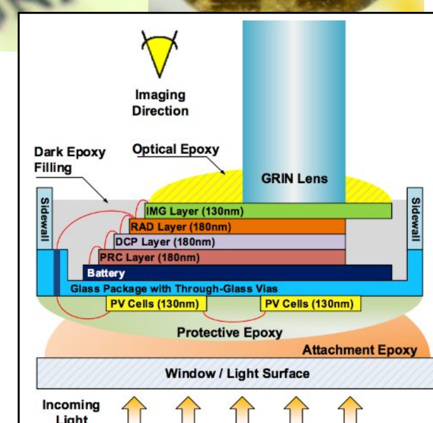
Hendy's "Law":
Pixels per dollar doubles annually



Credit: Barry Hendy/Wikipedia

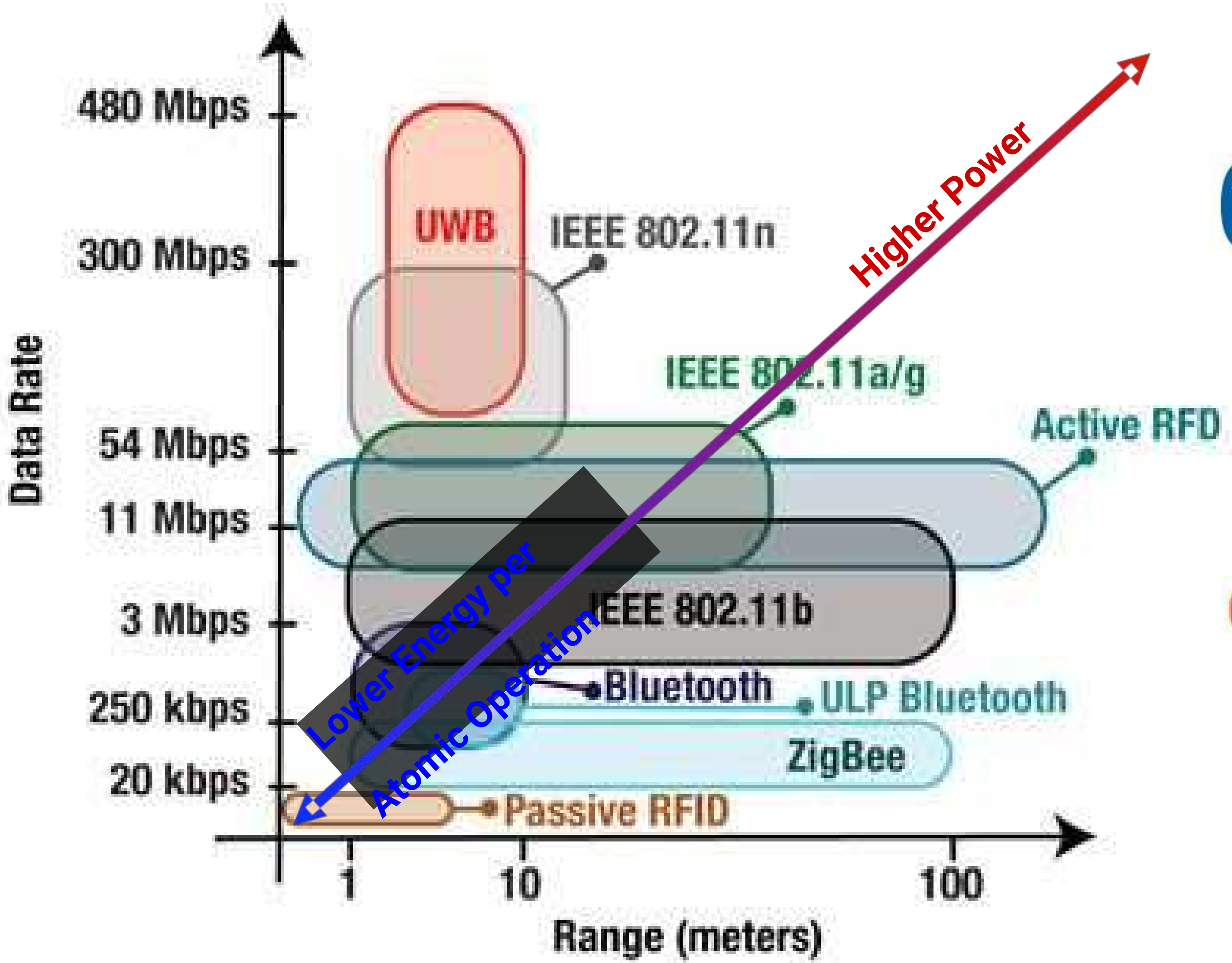
Figure 11. Sample image taken with the image sensor and GRIN lens, transmitted via Mbus.

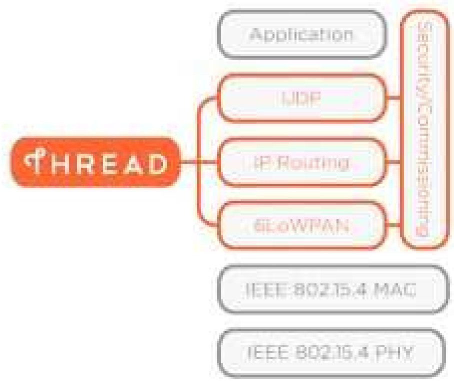




G. Kim, Z. Foo, Y, Lee, P. Pannuto, Y-S. Kuo, B. Kempke, M. Ghaed, S. Bang, I. Lee, Y. Kim, S. Jeong, P. Dutta, D. Sylvester, D. Blaauw, "A Millimeter-Scale Wireless Imaging System with Continuous Motion Detection and Energy Harvesting,
In Symposium of VLSI Technology (VLSI'14), Jun. 2014.

Radio technologies enabling pervasive computing, IoT



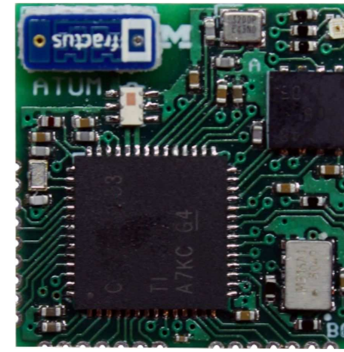


Source: Steve Dean, Texas Instruments
http://eecatalog.com/medical/2009/09/23/current-and-future-trends-in-medical-electronics/

Established commun interfaces: 802.15.4, BLE, NFC

IEEE 802.15.4 (a.k.a. “ZigBee” stack)

- Workhorse radio technology for sensornets
- Widely adopted for low-power mesh protocols
- Middle (6LoWPAN, RPL) and upper (CoAP layers)
- Can last for years on a pair of AA batteries



Bluetooth Smart

- Short-range RF technology
- On phones and peripherals
- Can beacon for years on coin cells

Near-Field Communications (NFC)

- Asymmetric backscatter technology
- Small (mobile) readers in smartphones
- Large (stationary) readers in infrastructure
- New: ambient backscatter communications

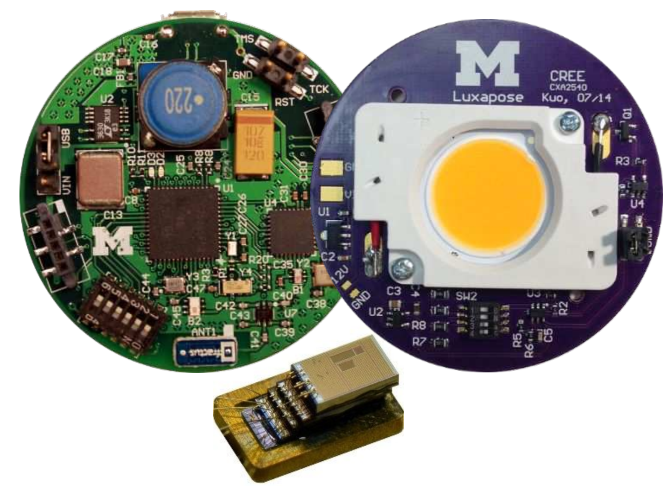
LPWAN

- 10 km + range.
- Battery lifespan of a decade, e.g., 10 mW for LoRa.
- Typically very low data rate, e.g., 20 kb/s for LoRa.

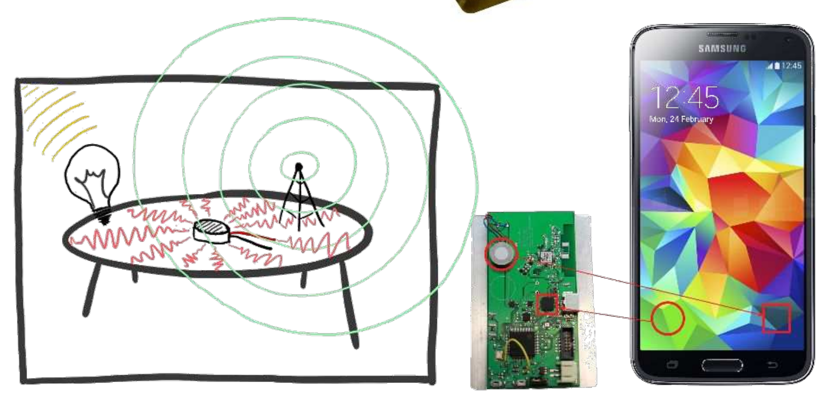
Emerging interfaces: ultrasonic, light, vibration

- Ultrasonic
- Small, low-power, short-range
- Supports very low-power wakeup
- Can support pairwise ranging of nodes

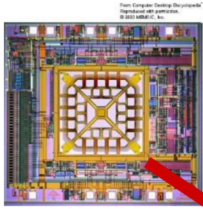
- Visible Light
- Enabled by pervasive LEDs and cameras
- Supports indoor localization and comms
- Easy to modify existing LED lighting



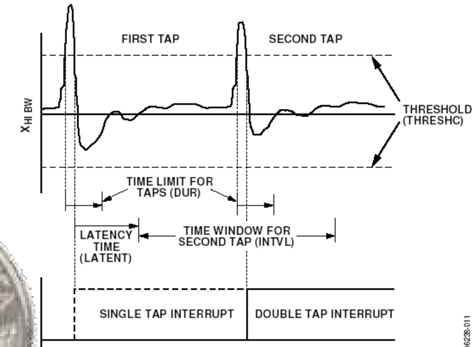
- Vibration
- Pervasive accelerometers
- Pervasive Vibration motors
- Bootstrap desktop area context



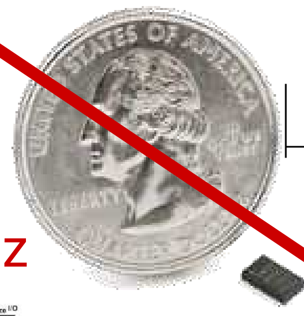
MEMS Sensors:
rapidly falling price and power of accelerometers



O(mA)



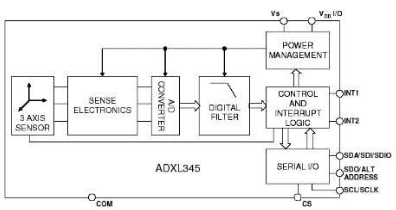
[Analog Devices, 2009]
ADXL345



25 µA @ 25 Hz

Price

Power



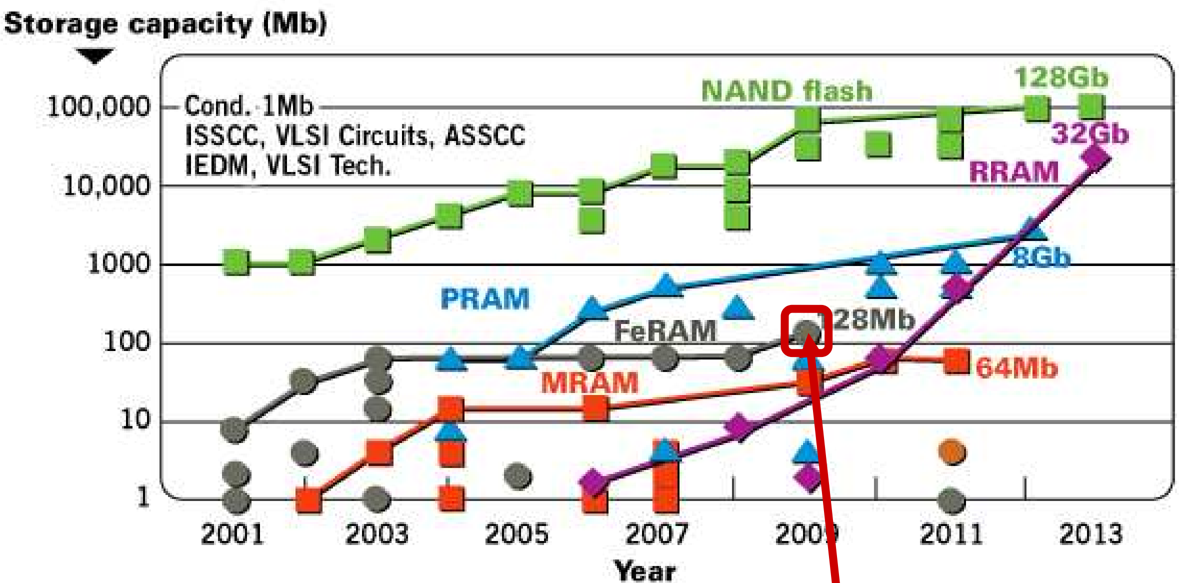
10 µA @ 10 Hz @ 6 bits
[ST Microelectronics, 2009]

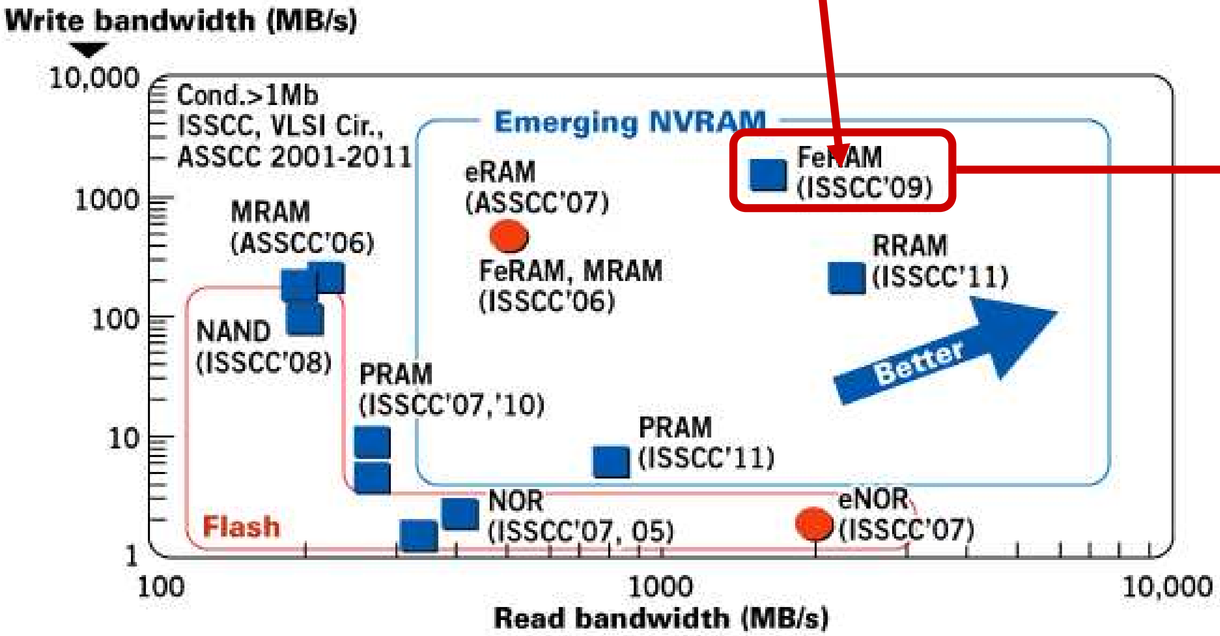
ADXL362

1.8 µA @ 100 Hz @ 2V
300 nA wakeup mode

[Analog Devices, 2012]

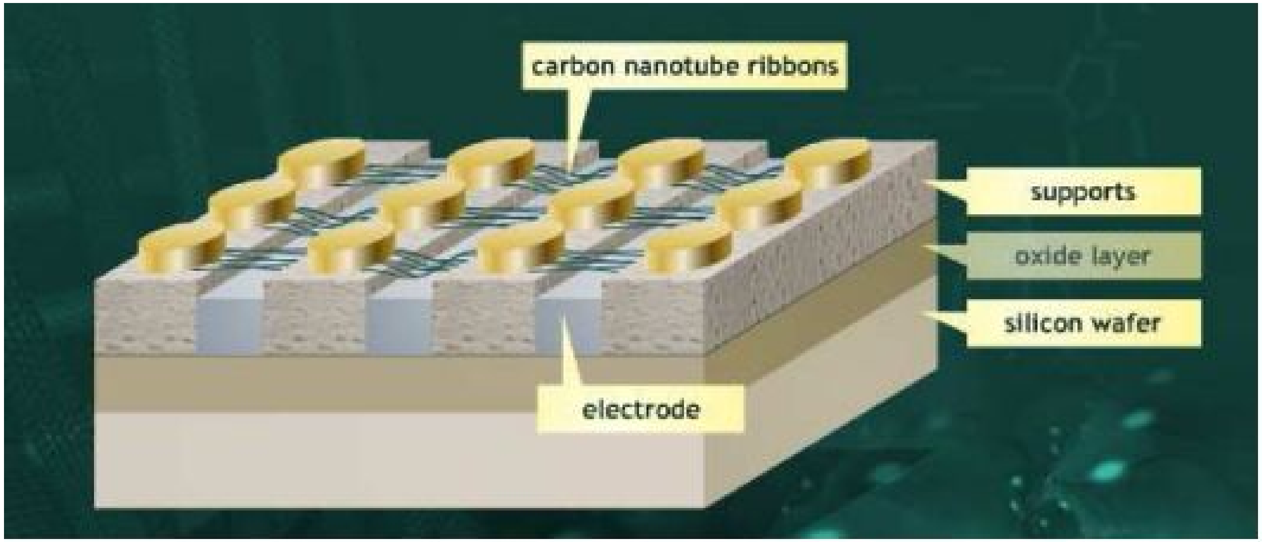
Non-volatile memory capacity & read/write bandwidth

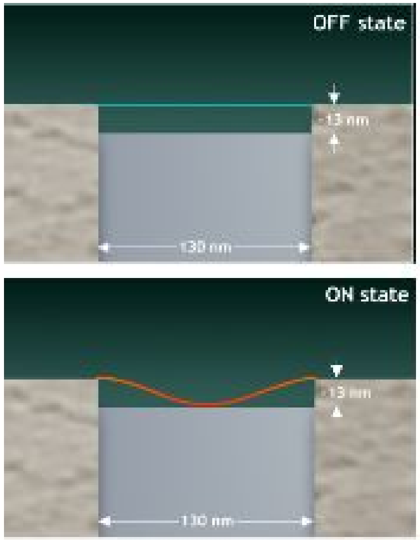




Lower capacity but
Higher R/W speeds
and
Lower energy per
atomic operation
and
High write
endurance

NRAM

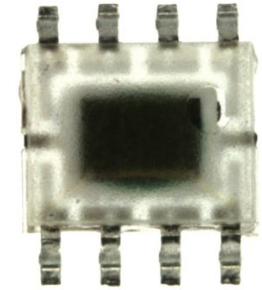




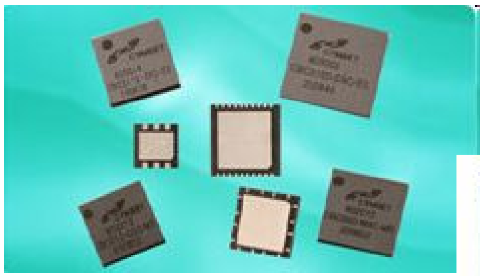
- Nonvolatile
- Fast as DRAM
- Vapor(hard)ware
- May happen

Energy harvesting and storage

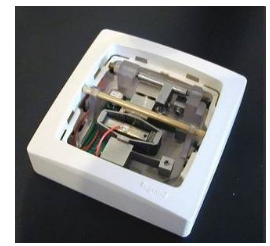
RF [Intel]



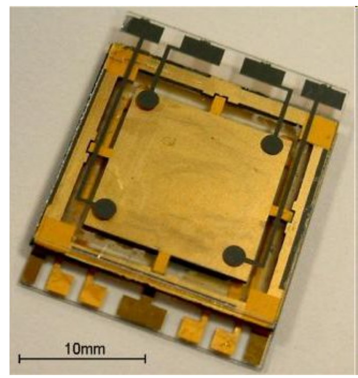
Clare Solar Cell



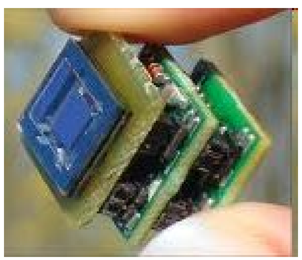
Thin-film batteries



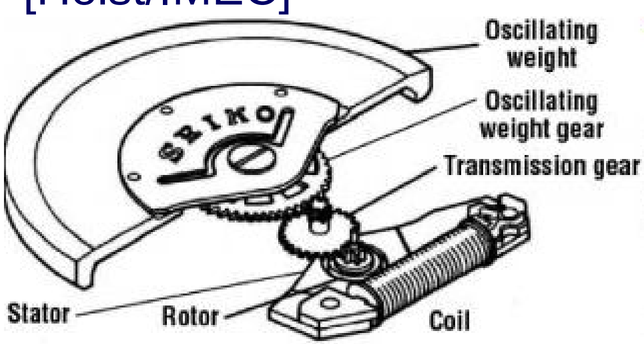
Shock Energy Harvesting
CEDRAT Technologies

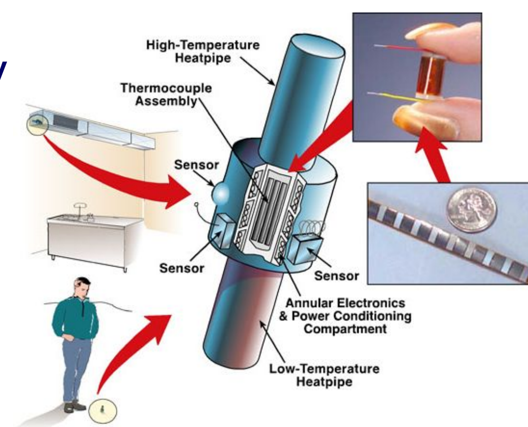


Electrostatic Energy
Harvester [ICL]



Piezoelectric
[Holst/IMEC]





Thermoelectric Ambient
Energy Harvester [PNNL]

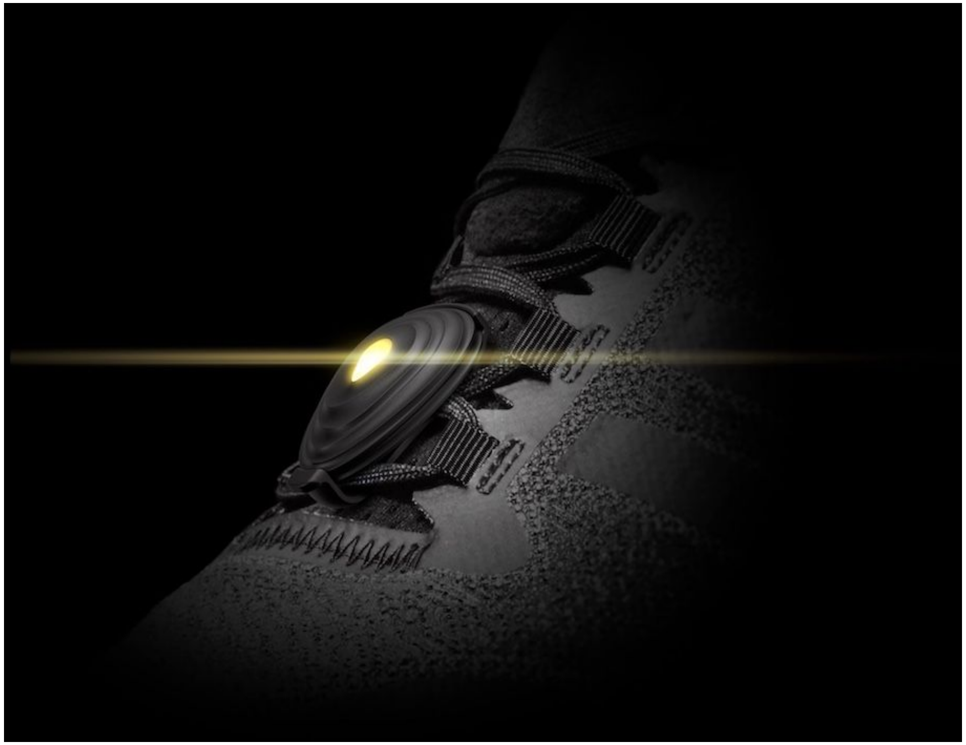
Growing application domains

- Wearable
- Social
- Location-aware
- IoT: integrated with physical world, networked
- Automated transportation
- Medical

My observations

- Every new class of computer systems will initially be seen as a toy by most.
- As it becomes socially and commercially important, nearly everybody will act as if it was always obvious...
- even those who claimed it would always be a toy.
- If logic dictates something, ignore the naysayers.
- But that logic better consider potential customers.

Embedded, Everywhere Example - Stryd



What?

- Tiny wearable embedded system
- Wireless communication
- Integrated signal processing
- Careful power management
- Unconventional sensors

Embedded, Everywhere Example - Stryd

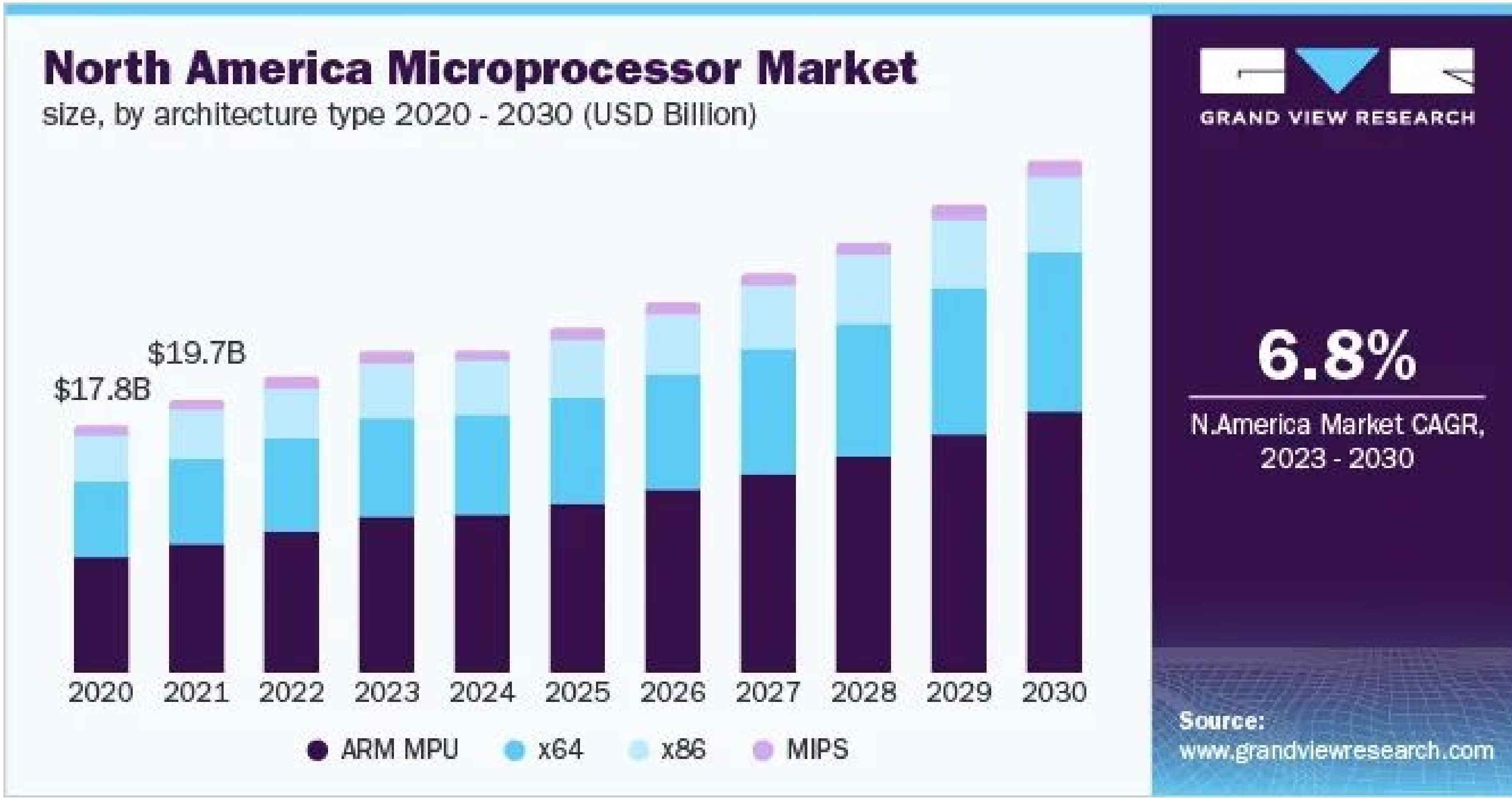
Lionel Sanders setting Ironman Triathlon World Record wearing Stryd

Why?

- Lets athletes precisely control effort when training and racing so they can develop faster and reduce race time.

Why study the ARMs and FPGAs?

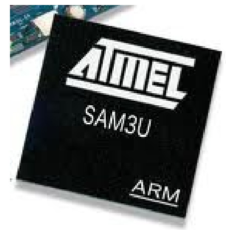
No single dominant architecture, but ARMs are popular



Many manufacturers ship ARM products

MICHIGAN

Actel
POWER MATTERS



ATMEL

CYPRESS
PERFORM

DUST NETWORKS

ember

NXP

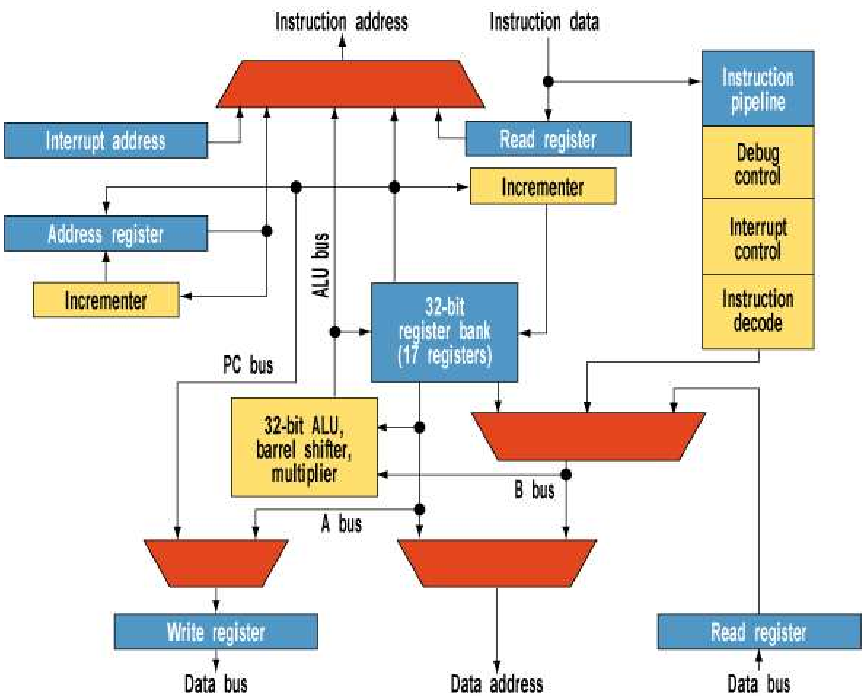
ST

TEXAS INSTRUMENTS

ARM

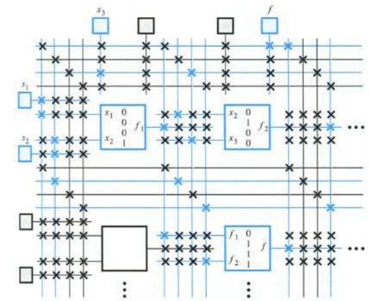
- ARM has a huge market share
  - Around $10B in 2023
  - >90% of smartphone market
  - 10% of notebooks
- Heavy use in general embedded systems
  - Inexpensive
  - ARM appears to get an average of $0.10 per device (averaged over cheap and expensive chips)
  - Flexible: spin your own designs
- Intel history

Microcontroller

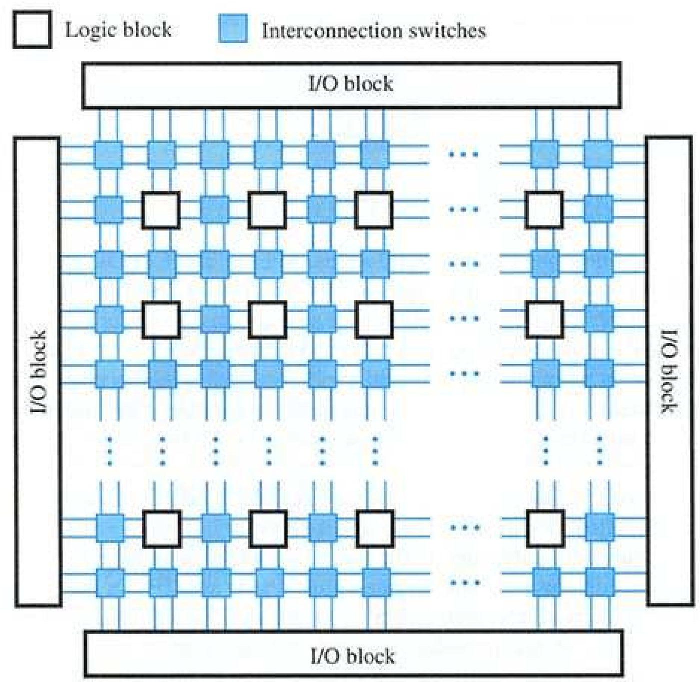


The Cortex M3's Thumbnail architecture looks like a conventional Arm processor. The differences are found in the Harvard architecture and the instruction decode that handles only Thumb and Thumb 2 instructions.

FPGA

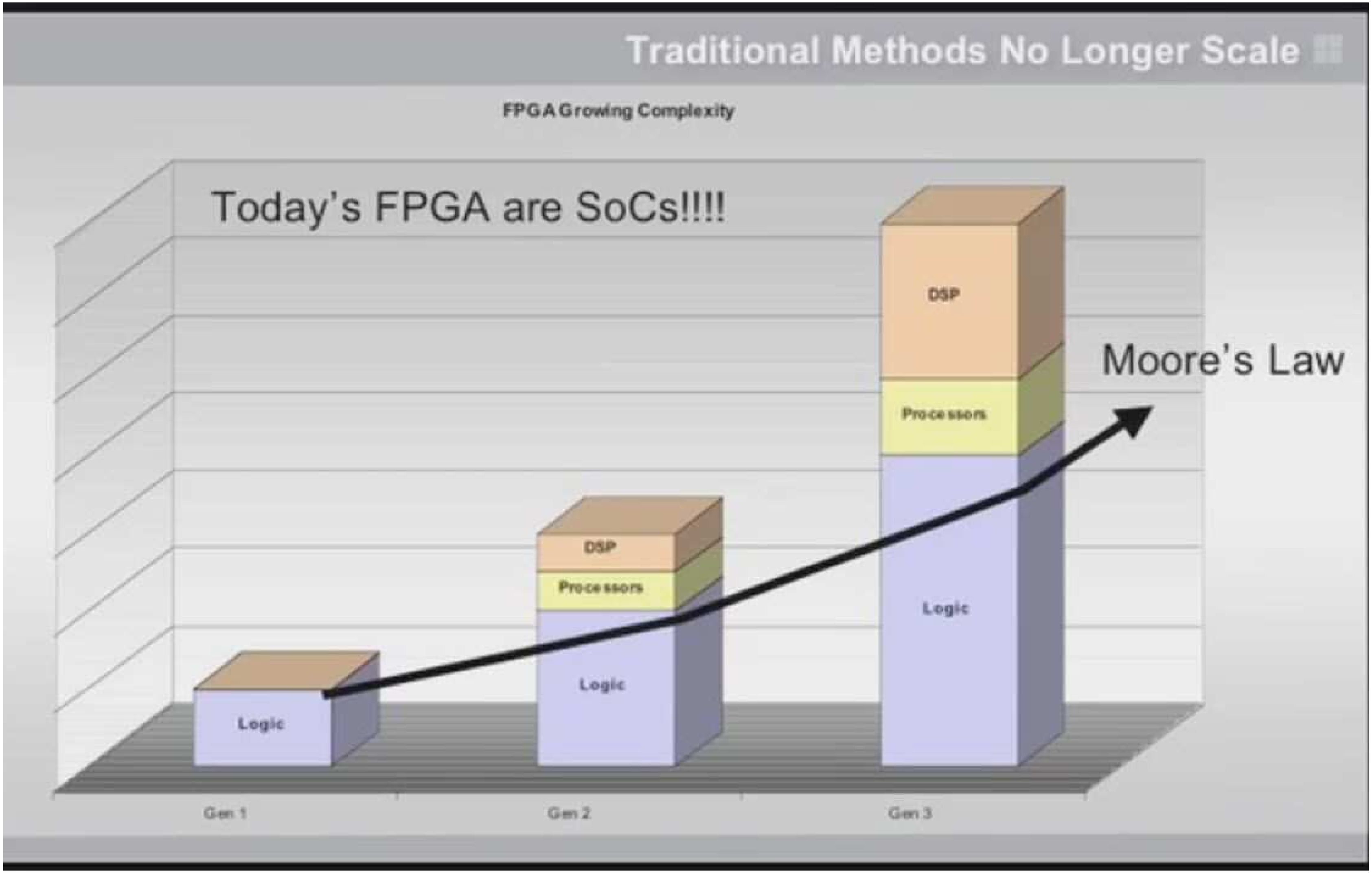


A section of a programmed FPGA



General structure of an FPGA

Modern FPGAs: best of both worlds!



Outline

~~Technology Trends~~

Course Description/Overview

Review, Tools Overview, ISA start

Website, etc.

- https://www.eecs.umich.edu/courses/eecs373/
  - Made for efficiency, not beauty.
  - Email me if you see problems.
  - It will grow during the semester.
  - Will have links to everything else you need.
- Piazza for Q&A.
  - Ask questions publicly unless it is about a private matter: other students can benefit.
- We'll use Gradescope for submitting assignments and grading exams.
- I don't plan to do much with Canvas.

Grades

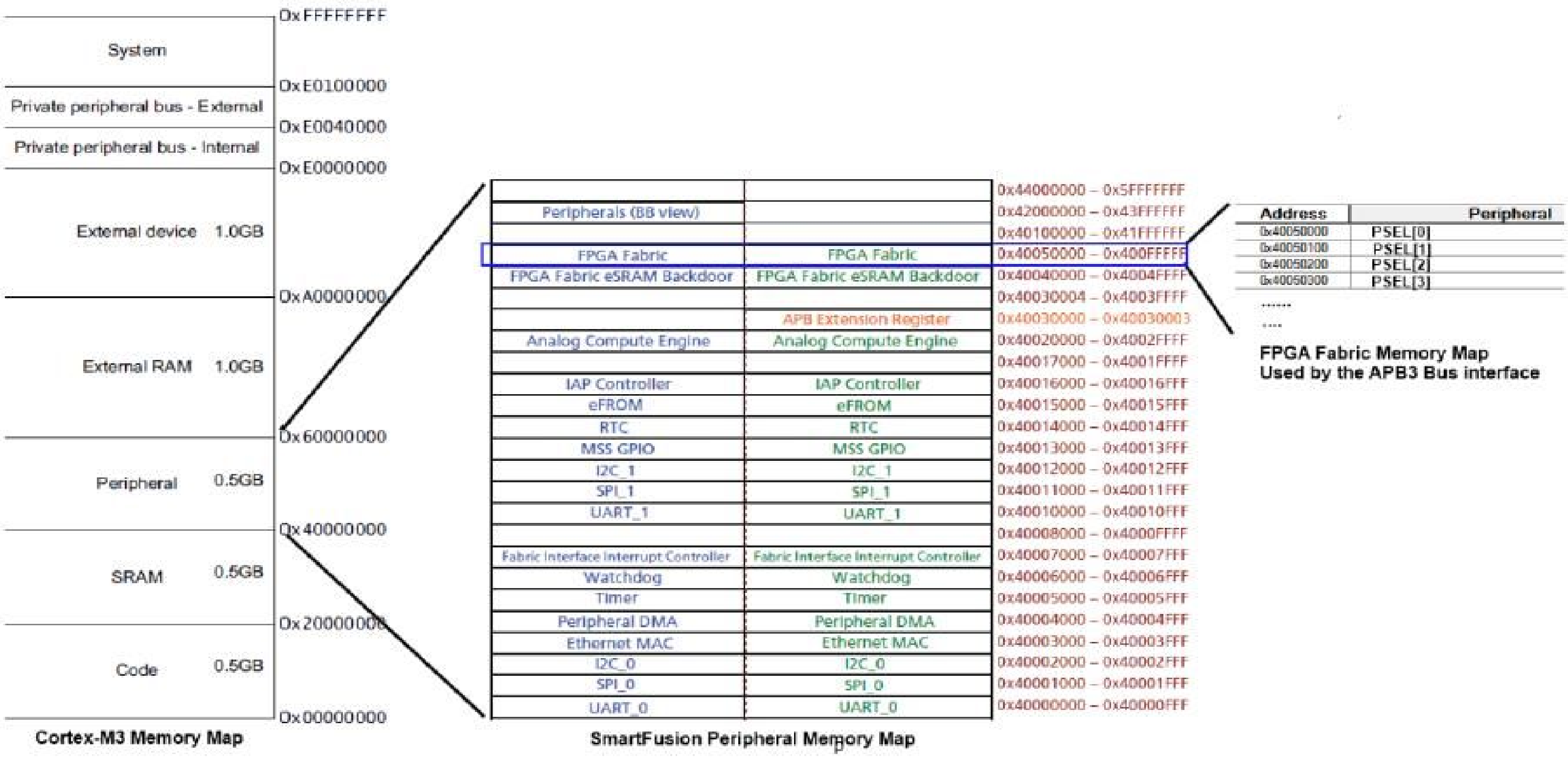
- Project and Exams tend to be the major differentiators.
- Class median is generally B

Labs	24%
Project	30%
Midterm 1	16%
Midterm 2	16%
Homework	7%
Presentations	7%

Prerequisites

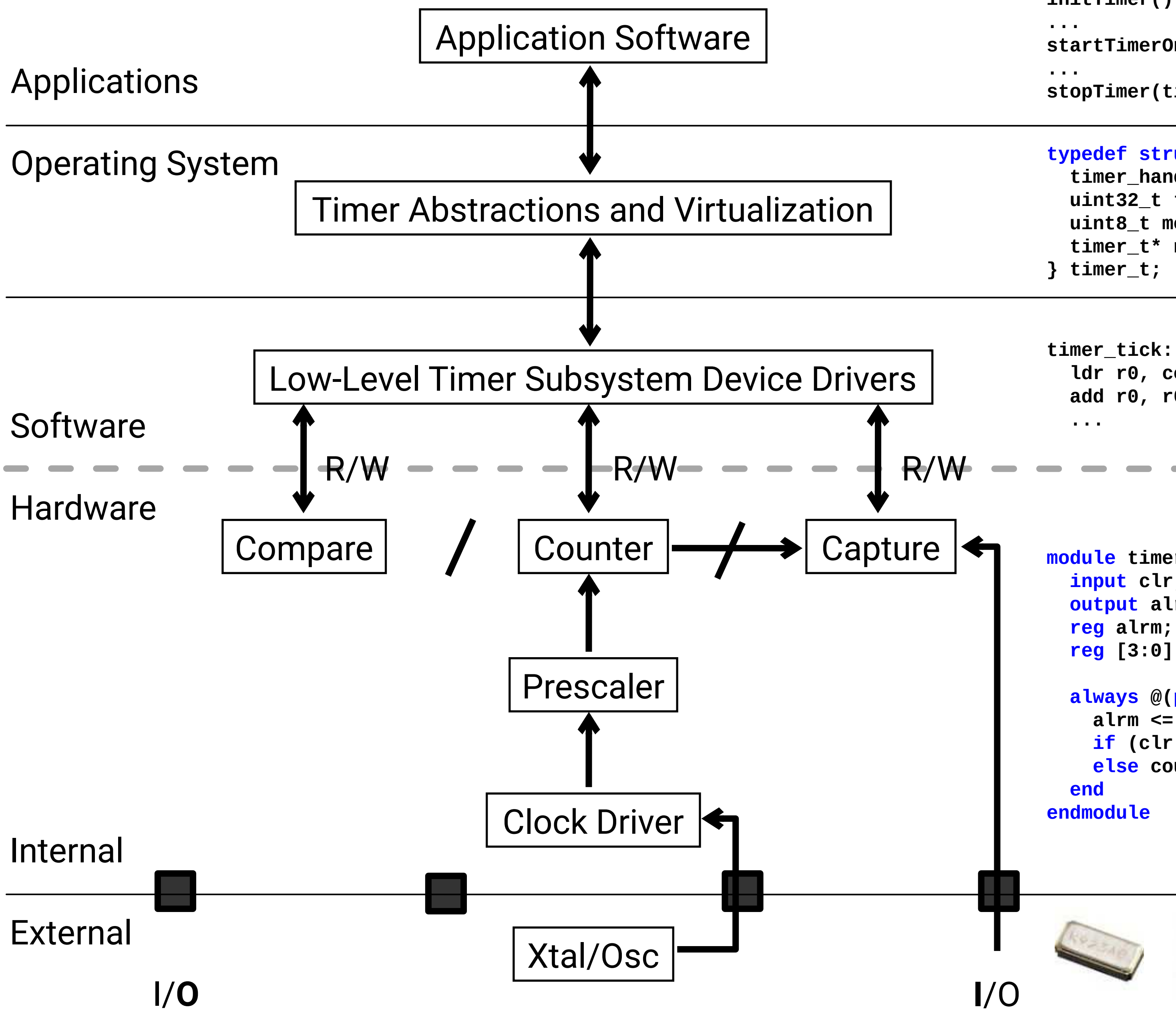
- EECS 270: Introduction to Logic Design
  - Combinational and sequential logic design
  - Logic minimization, propagation delays, timing
  - HDL
- EECS 370: Introduction to Computer Organization
  - Basic computer architecture
  - CPU control/datapath, memory, I/O
  - Compiler, assembler

Example: Memory-mapped I/O



- Enables program to communicate directly with hardware
  - Will use in Lab 3
  - Write memory to control motor
  - Read memory to read sensors

Example: Anatomy of a timer system



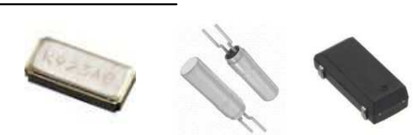

```
...
timer_t timerX;
initTimer();
...
startTimerOneShot(timerX, 1024);
...
stopTimer(timerX);
```

```
typedef struct timer {
  timer_handler_t handler;
  uint32_t time;
  uint8_t mode;
  timer_t* next_timer;
} timer_t;
```

```
timer_tick:
  ldr r0, count;
  add r0, r0, #1
  ...
```

```
module timer(clr, ena, clk, alrm);
  input clr, ena, clk;
  output alrm;
  reg alrm;
  reg [3:0] count;

  always @(posedge clk) begin
    alrm <= 0;
    if (clr) count <= 0;
    else count <= count+1;
  end
endmodule
```



Topics

- Memory-mapped I/O
  - The idea of using memory addressed to talk to input and output devices.
  - Switches, LEDs, hard drives, keyboards, motors
- Interrupts
  - How to get the processor to become "event driven" and react to things as they happen.
- Working with analog inputs
  - Interfacing with the physical world.
- Common devices and interfaces
  - Serial buses, timers, etc.

Time

- This is a time-consuming class
  - 2-3 hours/week in lecture
  - 8-12 hours/week working in lab
    - Expect more during project time; some labs are a bit shorter.
  - ~20 hours (total) working on homework
  - ~20 hours (total) studying for exams.
  - ~8 hour (total) on your oral presentation
- Averages out to about 15-20 hours/week pre-project and about 20 during the project…
  - This is more than I would like, but we've chosen to use industrial-strength tools, which take time to learn.

Labs

- 7 labs done in teams
  - FPGA + Hardware Tools
  - MCU + Software Tools
  - Memory + Memory-Mapped I/O
  - Interrupts
  - Timers and Counters
  - Serial Bus Interfacing
  - Data Converters (e.g., ADCs/DACs)

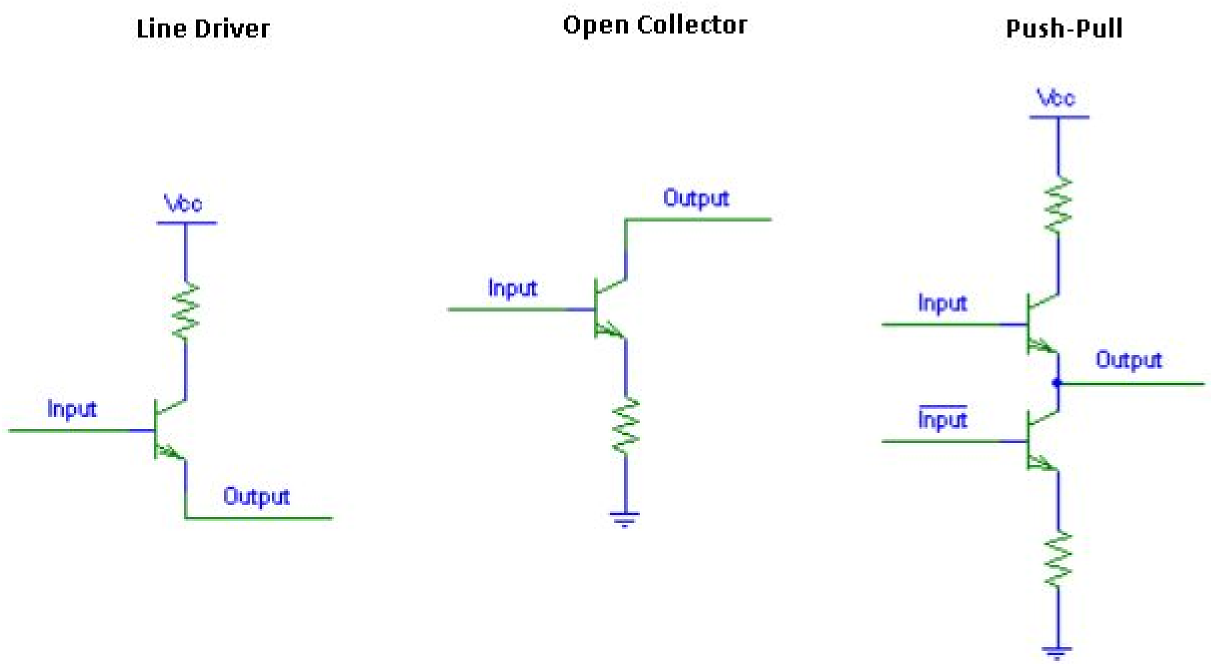
- Difficulty ramps up for higher labs.

- Labs are very time consuming.
  - As noted, students estimated 8-12 hours per lab with one lab (which varied by group) taking longer.

Lab 1

- Labs start on 16 January.
- We'll post the first lab shortly and announce.
- In Lab 1, you'll get familiar with the development environment.
- Develop a Hello World program.
- Use GPIO.
- Prelab normally required before lab section.
- For Lab 1, do as much of prelab as you are able.
  - We'll help if you are stuck.
- The prelab section is tutorial, but a few concepts will help.
- Control by writing values to specific memory locations.
- Hi-Z output state.
- Masking
  - 0101011**1** & 00000001 = 1
  - 0101011**0** & 00000001 = 0
- Multiple ports.
- C adequate. No assembly required.

Lab 1: drive type



Credit National Instruments for figure.

- Open collector: Output floats when input low, low when input high.
- Push-pull: Output high when input high, low when input low.

Open-ended project

- Goal: learn how to build embedded systems
  - By building an embedded system
  - Work in teams
  - You design your own project
- Will provide list.
- Can define own goal.
- Major focus of the last third of the course.
- Important to start early.
  - After labs end, some slow down.
  - That's fatal.
- This is the purpose and focus of the course.

Homework

- Around 5 assignments.
- Review material that will be useful in lab.
- Reinforce material from lecture.
- Most will have a one-week deadline.
- First assignment will be posted this week.

Exams

- Two midterm exams.
  - 20 February.
  - 26 March.
- Done when focus switches to project.
- 32% of grade.
- Higher (grade, not time) variance than project.
- We plan to offer review sessions.
- Also plan to post midterms from prior semesters as examples.

Office hours

- Robert Dick: 4:30-5:00 Tu, Th in 2417-E EECS.
  - Typically right after class.
  - Will extend duration if there is demand.
  - Typically, if there are people discussing or waiting at 5pm, I stay.
- Other office hours will be posted to the website.
- In Google Sheet.
  - Need to be logged in with UMich account.

Outline

~~Technology Trends~~

~~Course Description/Overview~~

Review, Tools overview, ISA start

Verilog

- Not covered in course.
- Review 270 material.
- Understand key differences w. SW languages (e.g., C).
  - E.g., nonblocking statement semantics.

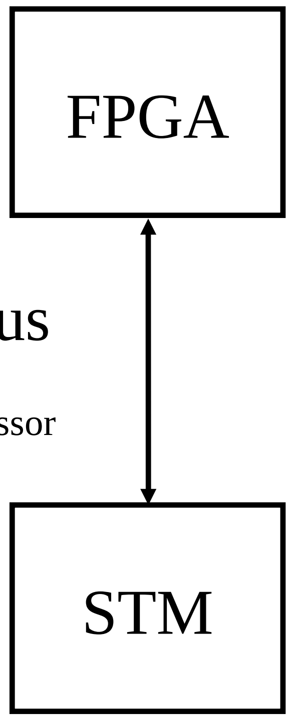
Net states

- What is a bus?
- What does “drive” mean?
- What does Hi-Z (high impedance) mean?
- Check EECS 270 notes and get help on Piazza.

Crash course in debugging

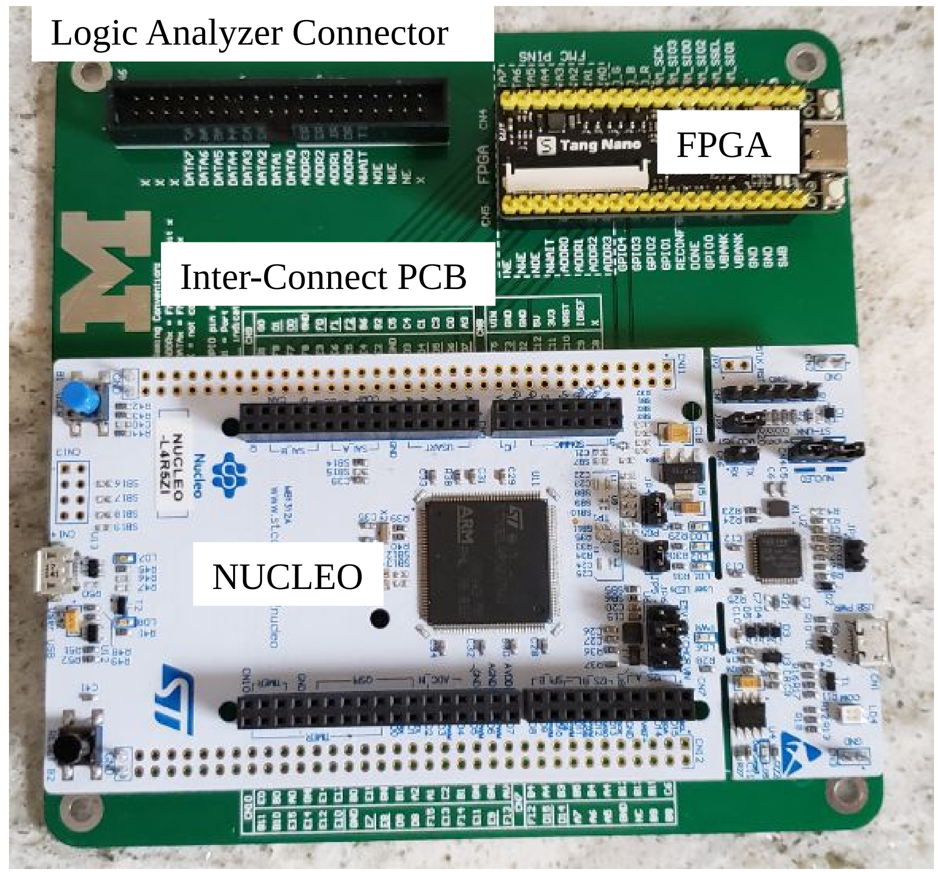
- Add minimal, independently tested units to system.
- Test after each small change.
- Minimize unnecessary interactions among components.
- Maybe you have heard these. Why do they exist?
- We’ll cover this in more detail later.
- First, I want to cover material that will help on the first few labs.
- More on HW/SW system debugging later, it’s one of the more useful skills you will improve in this class.

STM Kit + FPGA Kit



Parallel Bus

FPGA looks like memory to processor



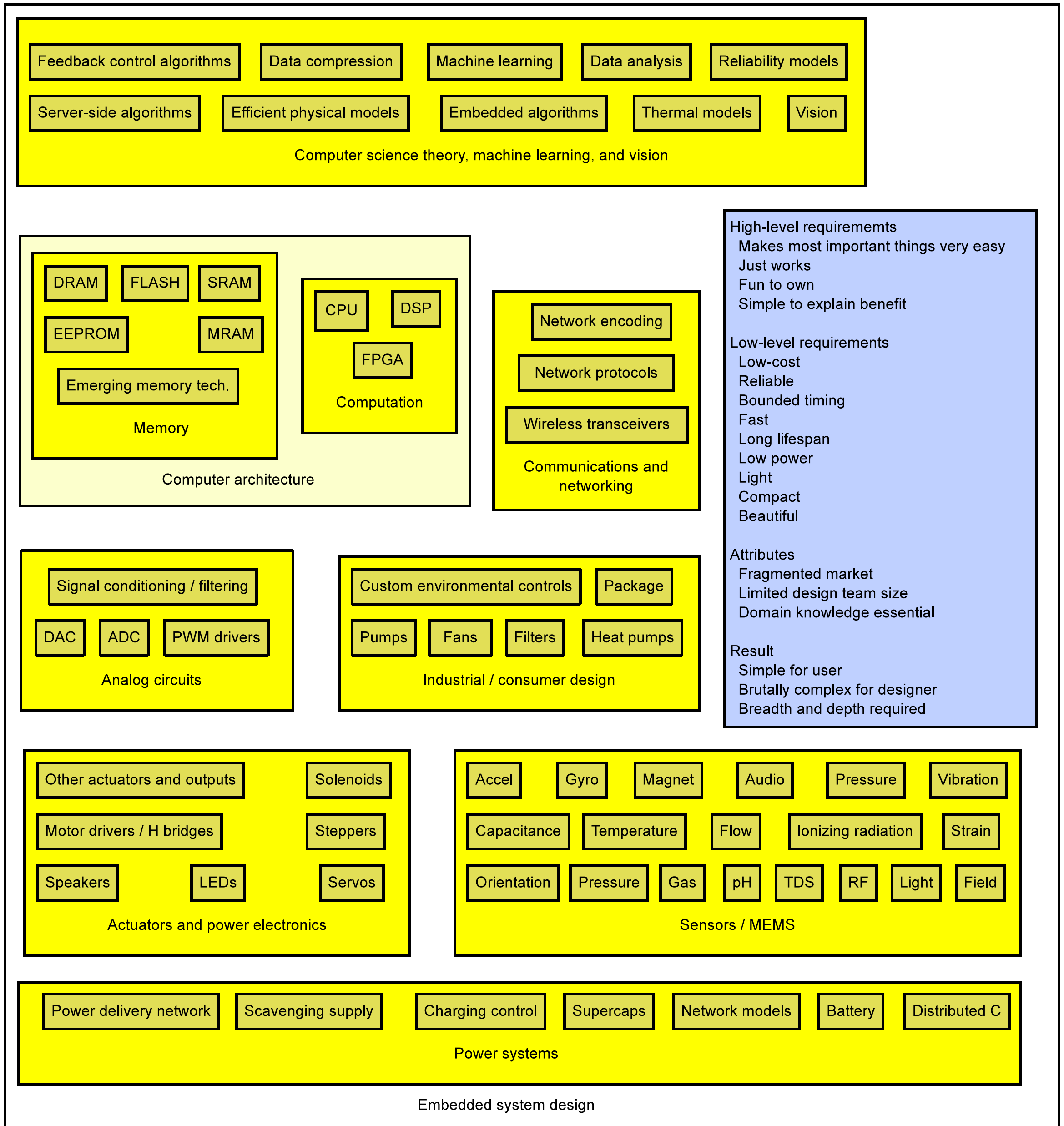
The Hardware

- STM32L4R5ZI
  - STM has a family of many ARM based processors.
  - 32 bit Cortex M4 ARM processor
  - Microprocessor system with IO, memory, etc
- Development Kit
  - Provided by manufactures like STM for developers.
  - Has debugger interface
  - Basic IO like switches and LEDs

The CAD Software

- STM Development Software CubeIDE
  - Environment to edit, compile and debug code
  - GUI is a common general purpose development interface called Eclipse
  - The tools under the GUI doing the compiling, debugging etc are a freeware set based on the GNU tool chain.

An embedded system



Done.