ESE 566A
Modern System-on-Chip Design

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese566/

Course Overview

Objectives

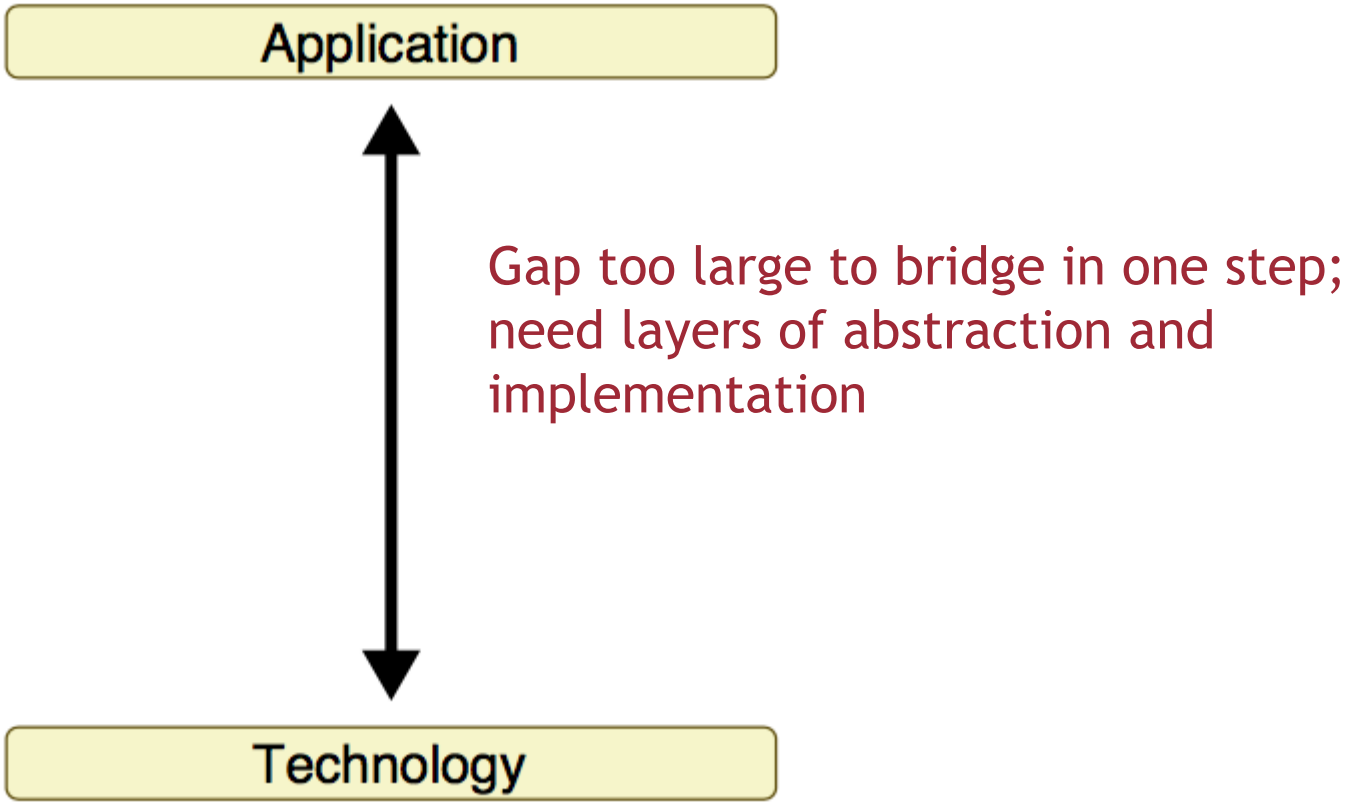
Motivations

Structure

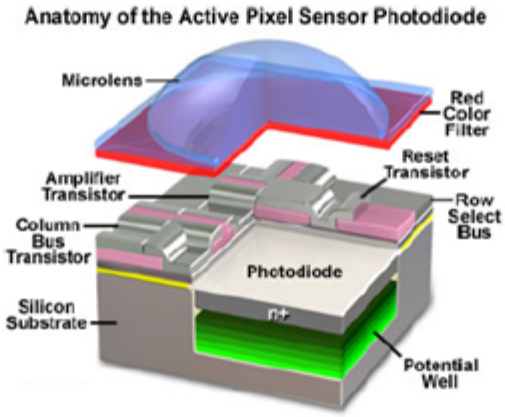
Logistics

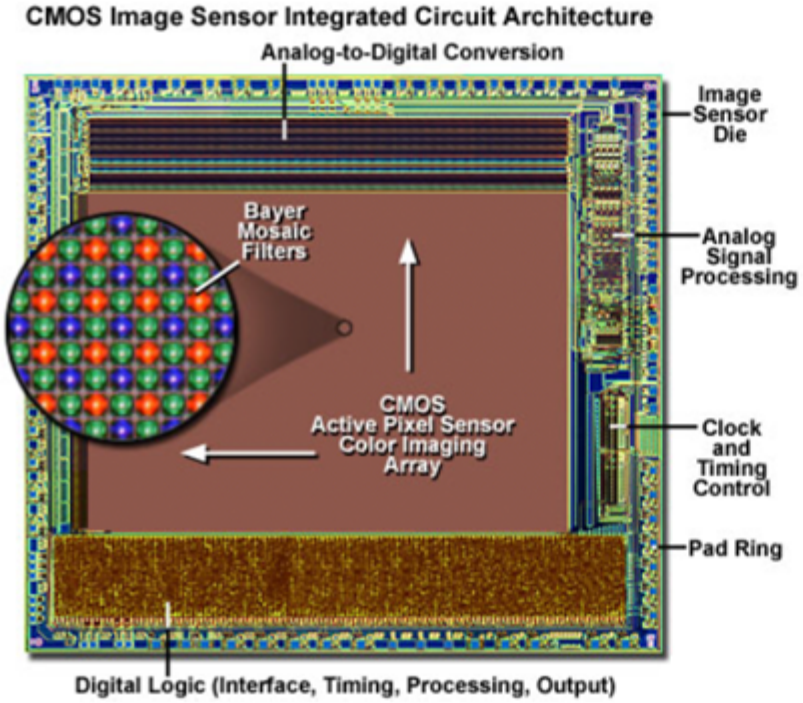
Computer Engineering Stack

- Bridge application and technology
  - examples
  - app: navigation (North), tech: magnetic compass
  - app: face recognition, tech: charge-coupled device (CCD)



Interface to the Physical World: The camera

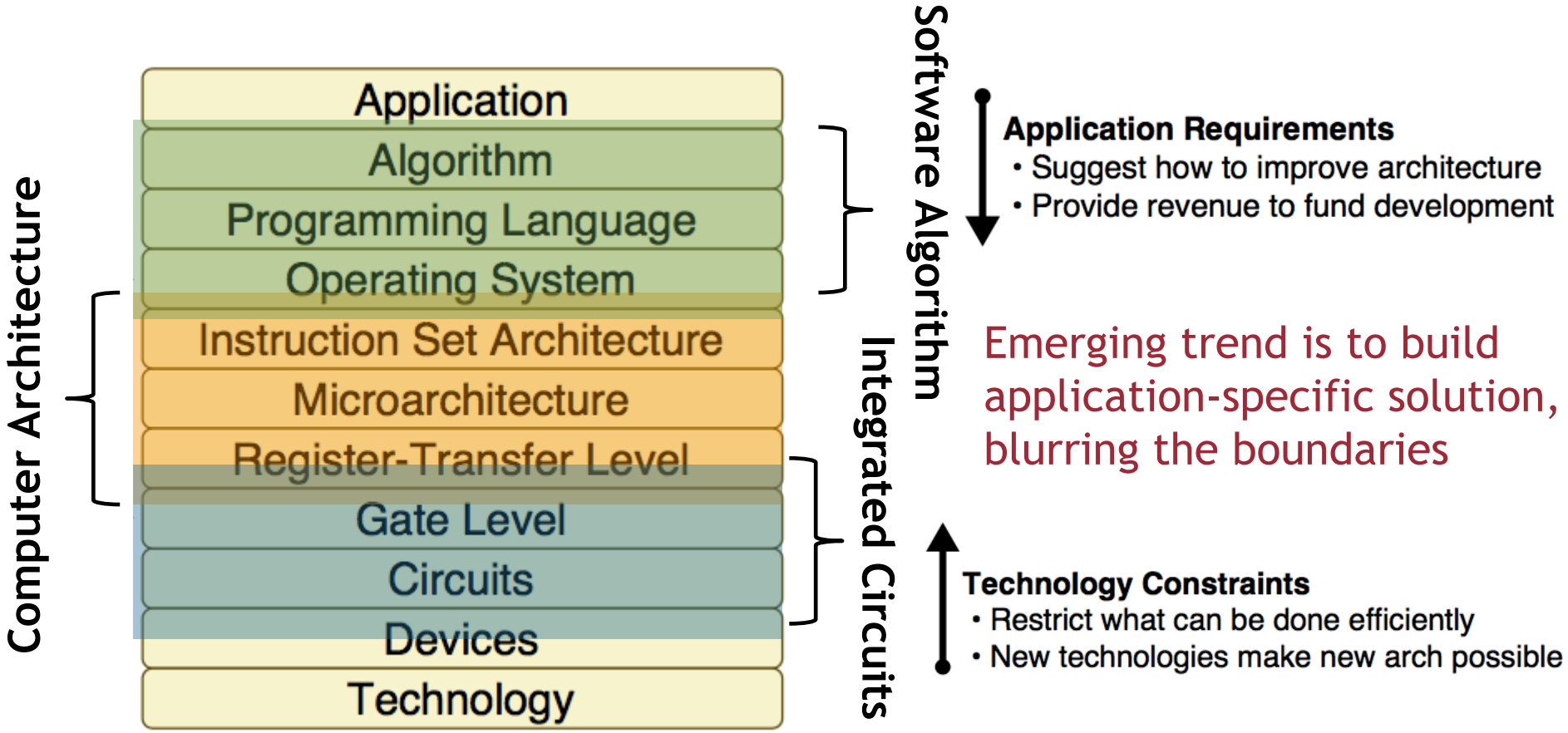




- Focus/Exposure Control
- Pre-processing
- White-balancing
- Demosaic
- Color Transform
- Post-processing
- Compression

Computer Engineering Stack

- Circuit, Architecture, and Algorithm
  - traditional division



What is a System-on-Chip (SoC)?

- Contain general-purpose processor
  - but also other computing units
- Designed for specific application
- Small, low power, portable
- Unique architecture
  - core, memory, accelerators, peripherals
- Combine digital, analog, mixed-signal
- Power management circuits
- Timing source
- Cost constrained

SoC is the perfect place where circuit, architecture, and algorithm meet; use it as a teaching example.

Course Objectives

- Understand basic IC technology and design flow
  - CMOS device and circuits
  - language: Verilog
  - tools: Synopsys, Cadence
  - process: design, simulation, synthesis, verification, test
  - principles: performance, power, other considerations
- Understand key architecture concepts and metrics
  - processor, memory, network
  - speed, power/energy, complexity/chip area
- Introduce advanced tools for future system design
  - high-level synthesis, hardware acceleration

Course Overview

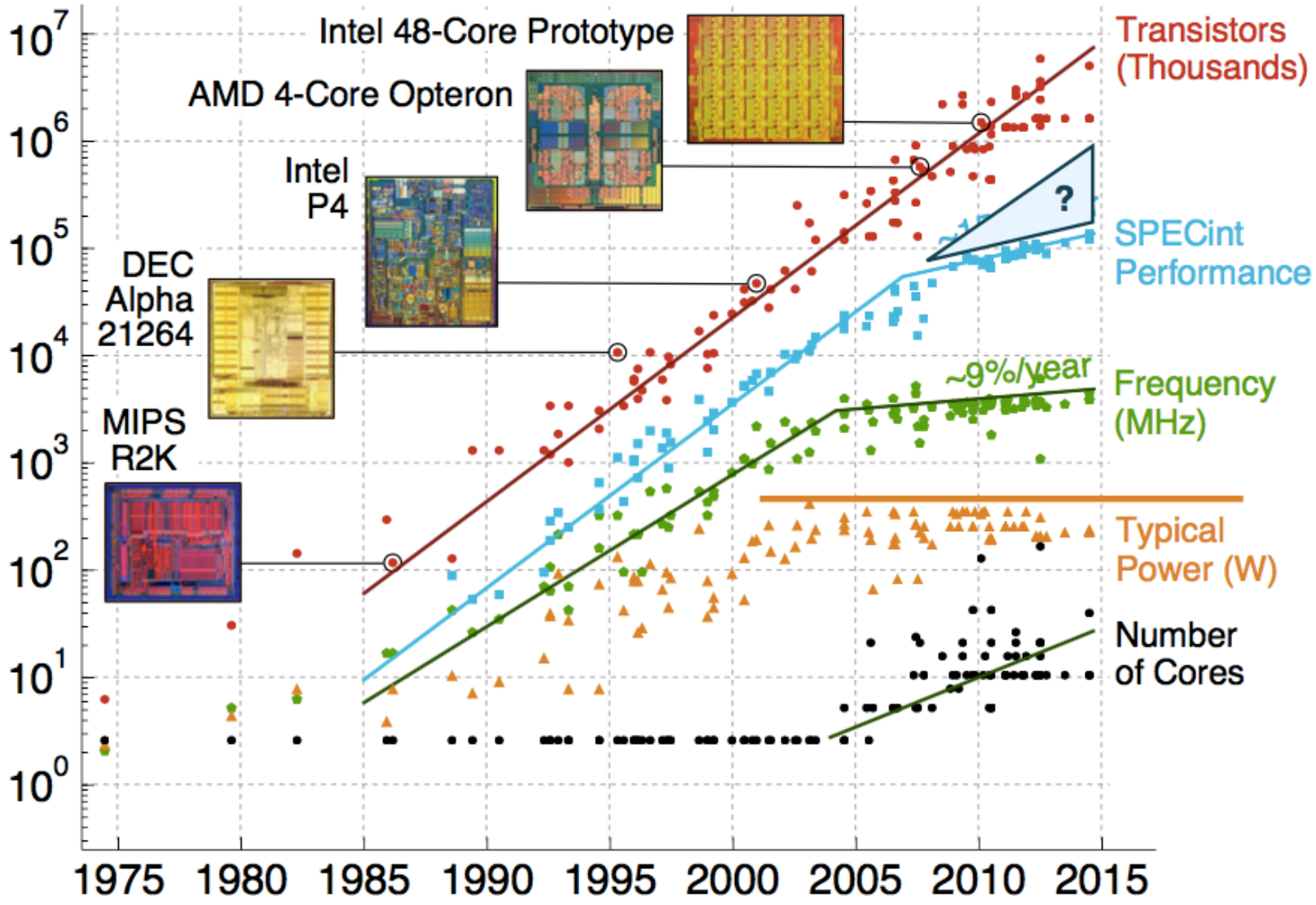
Objectives

Motivations

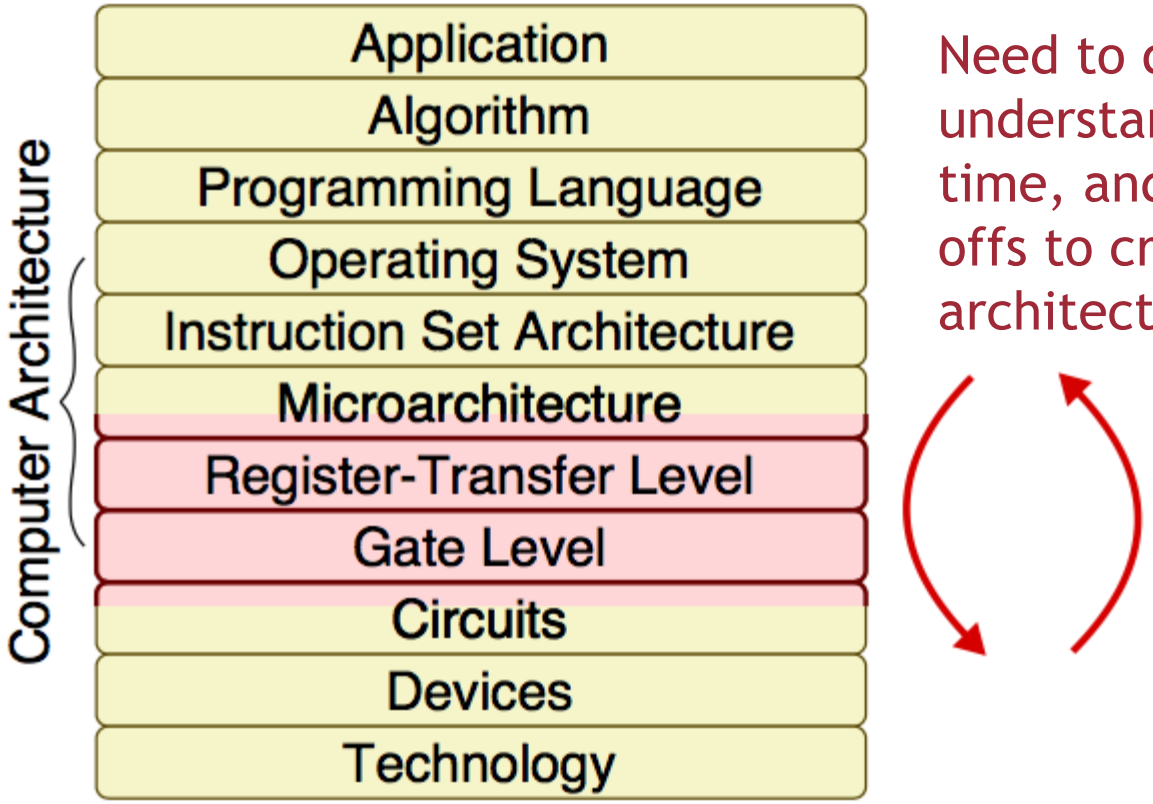
Structure

Logistics

Computer Architecture Research Perspective



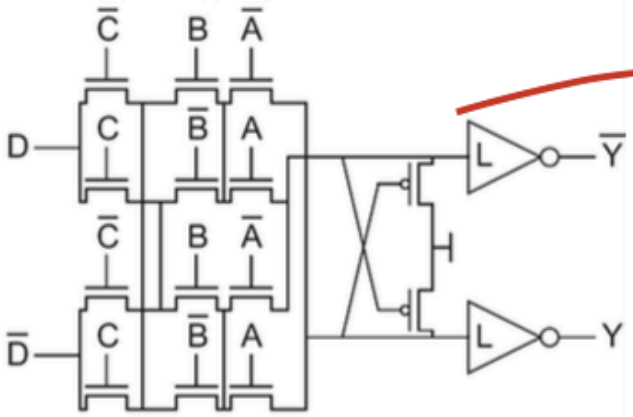
Cross-Layer Interaction is Critical



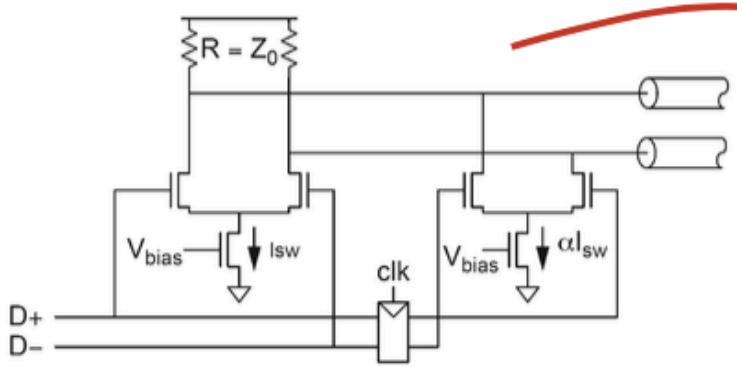
Need to quantitatively understand area, cycle time, and energy trade-offs to create new architectures

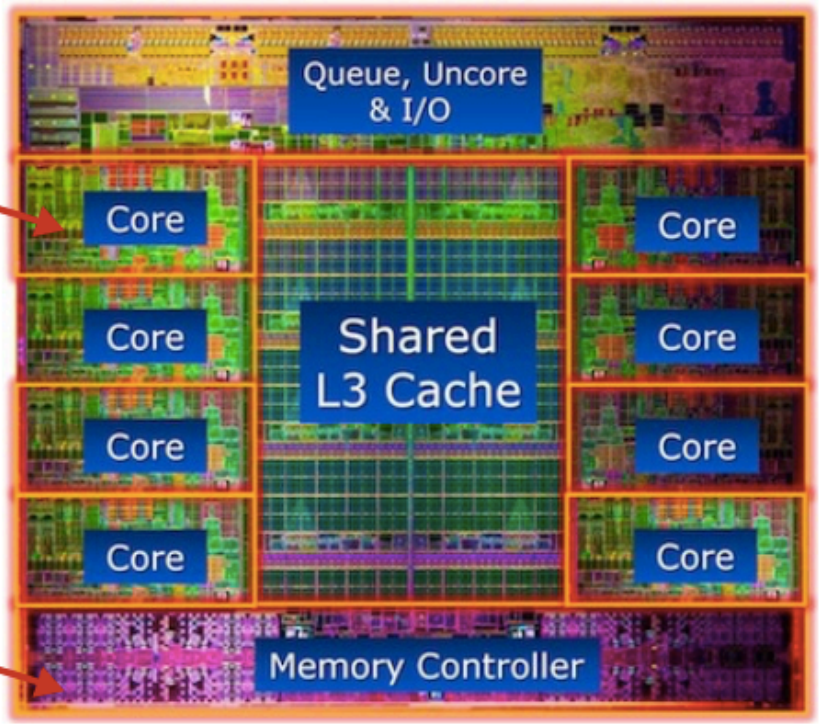
Circuit Research Perspective

Your Digital Circuit Here



Your Analog Circuit Here

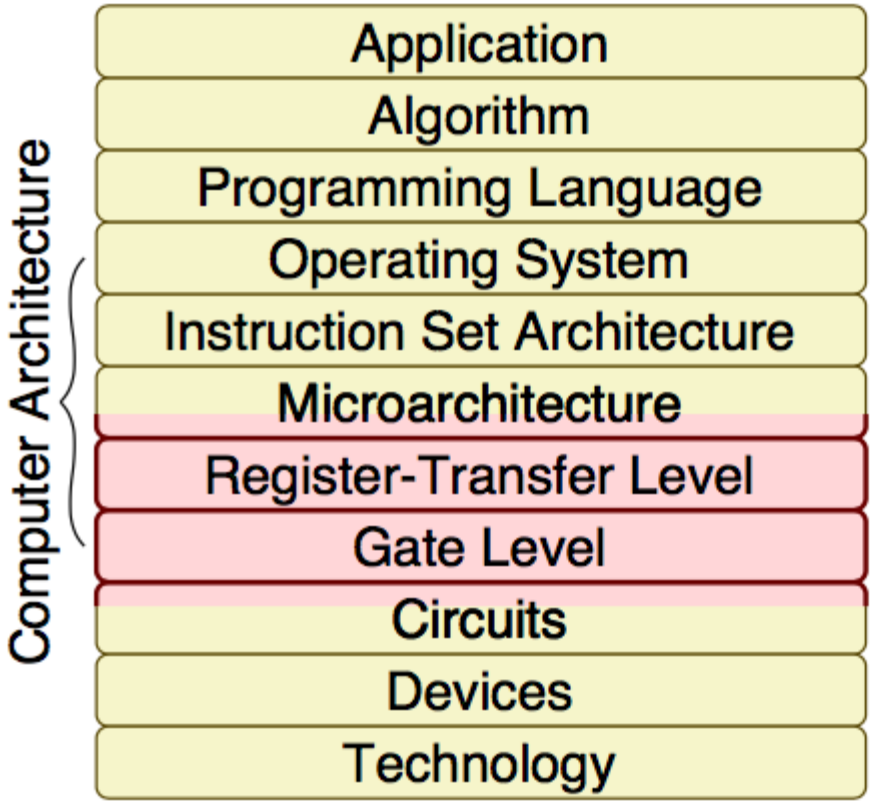




Fighter Airplane
~100,000 parts

Intel Sandy Bridge E
2.27 Billion transistors

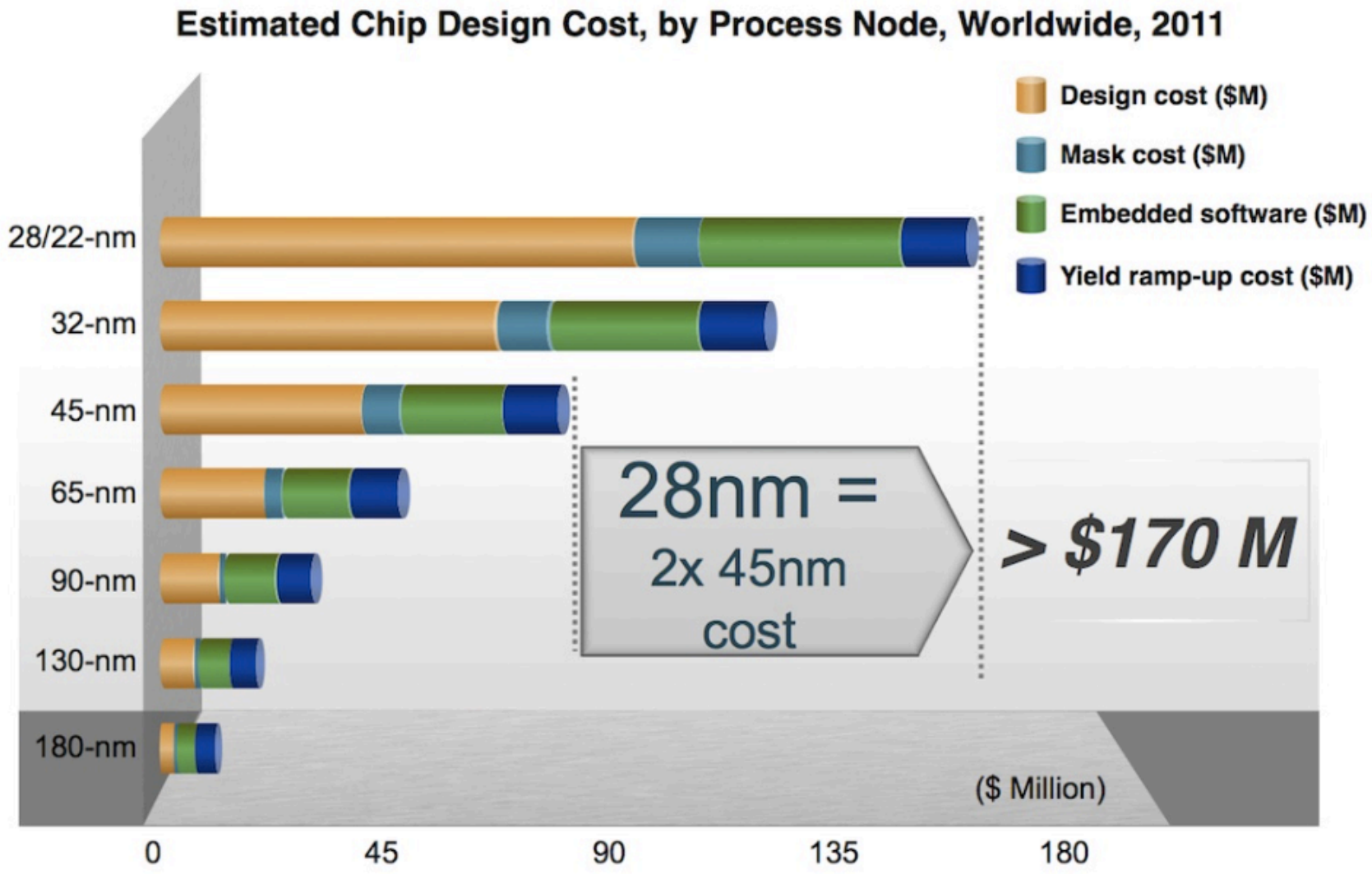
Cross-Layer Interaction is Critical



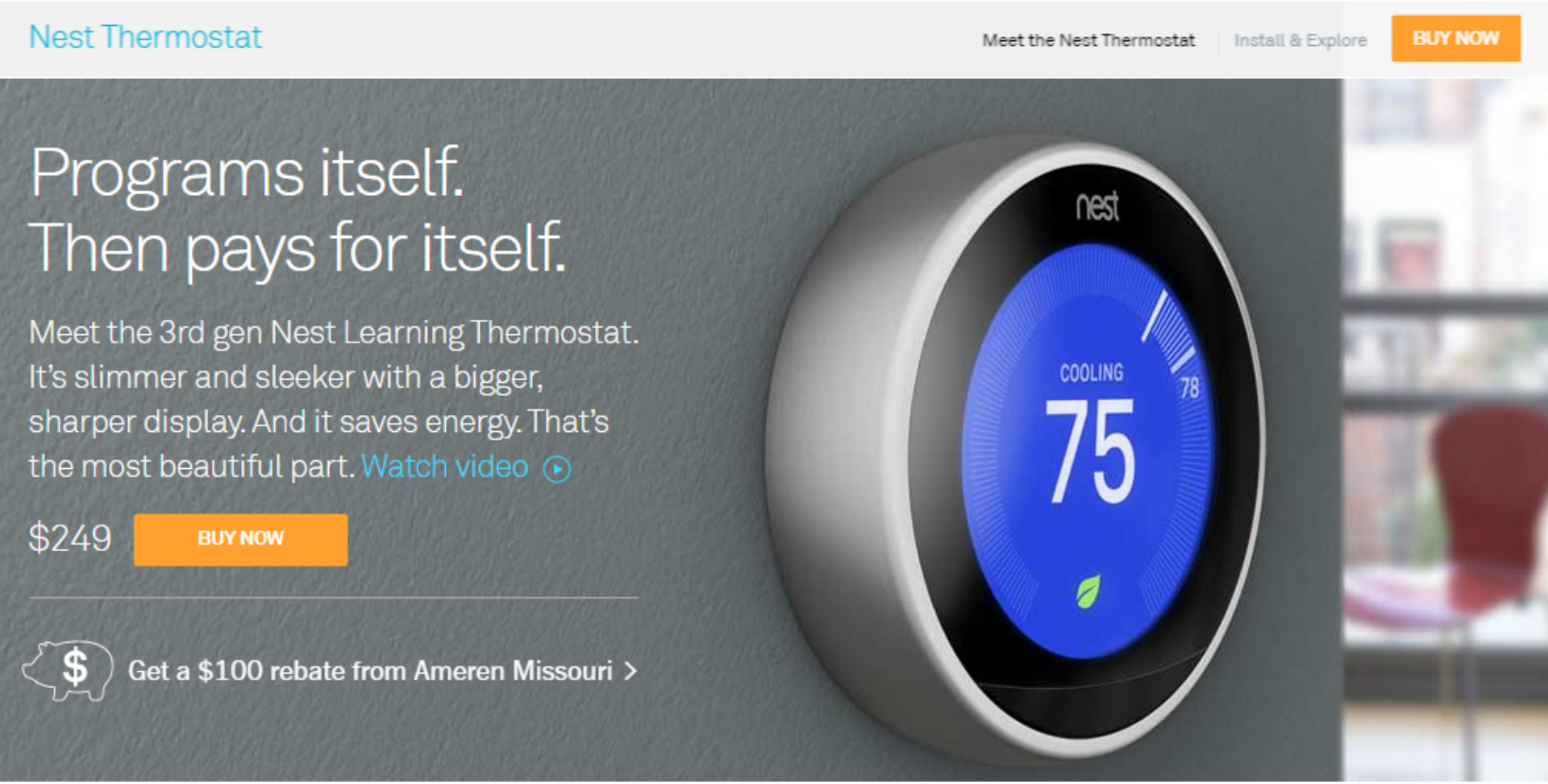
Need to appreciate the system-level context for their subsystems to identify exciting new opportunity

Industry Development Perspective

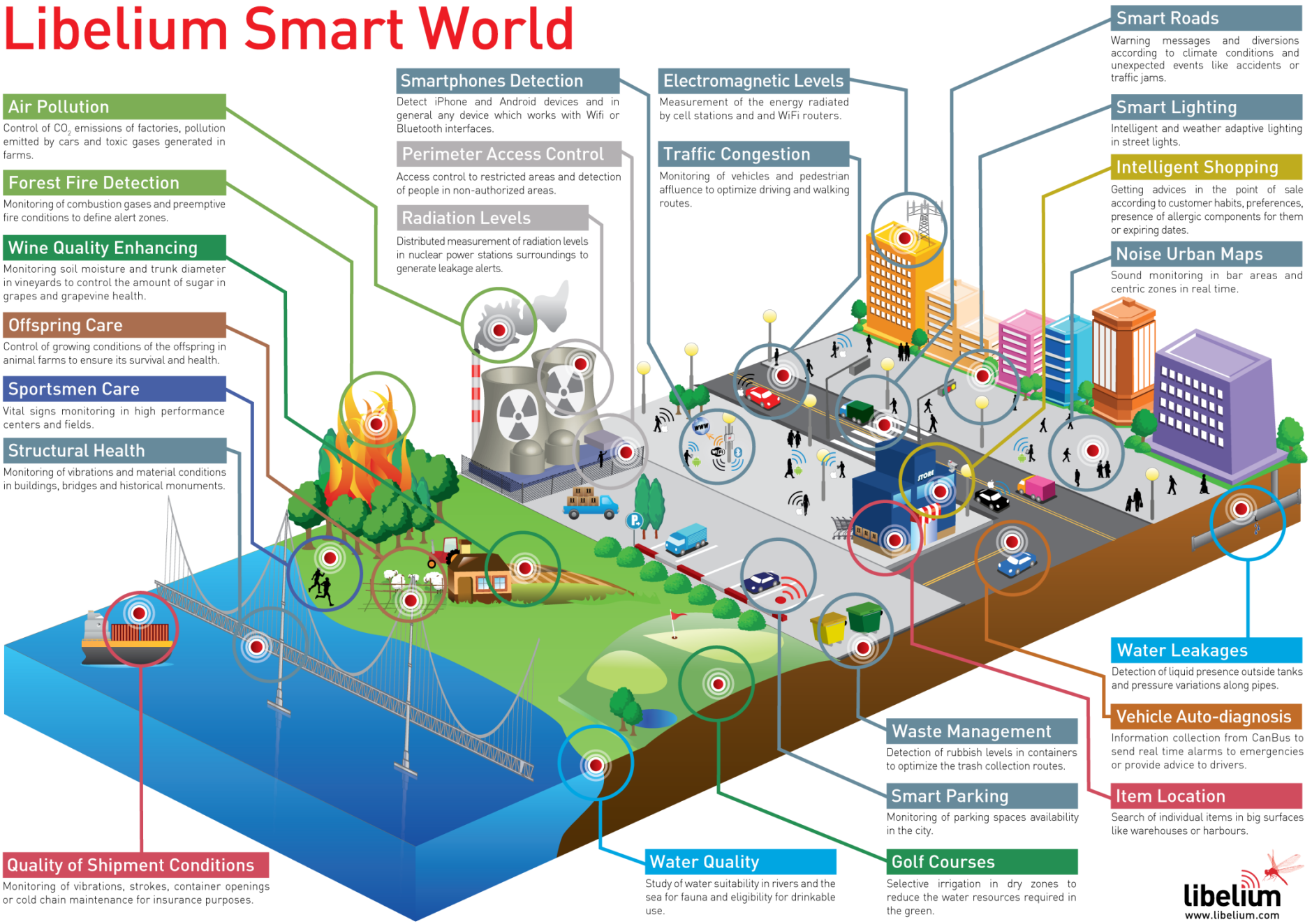
- Industry at a cross-road



Case Study: Internet-of-Things (IoT)



Libelium Smart World



Air Pollution
Control of CO_2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Forest Fire Detection
Monitoring of combustion gases and preemptive fire conditions to define alert zones.

Wine Quality Enhancing
Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

Offspring Care
Control of growing conditions of the offspring in animal farms to ensure its survival and health.

Sportsmen Care
Vital signs monitoring in high performance centers and fields.

Structural Health
Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

Quality of Shipment Conditions
Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.

Smartphones Detection
Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

Perimeter Access Control
Access control to restricted areas and detection of people in non-authorized areas.

Radiation Levels
Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

Electromagnetic Levels
Measurement of the energy radiated by cell stations and and WiFi routers.

Traffic Congestion
Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.

Water Quality
Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.

Waste Management
Detection of rubbish levels in containers to optimize the trash collection routes.

Smart Parking
Monitoring of parking spaces availability in the city.

Golf Courses
Selective irrigation in dry zones to reduce the water resources required in the green.

Smart Roads
Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

Smart Lighting
Intelligent and weather adaptive lighting in street lights.

Intelligent Shopping
Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.

Noise Urban Maps
Sound monitoring in bar areas and centric zones in real time.

Water Leakages
Detection of liquid presence outside tanks and pressure variations along pipes.

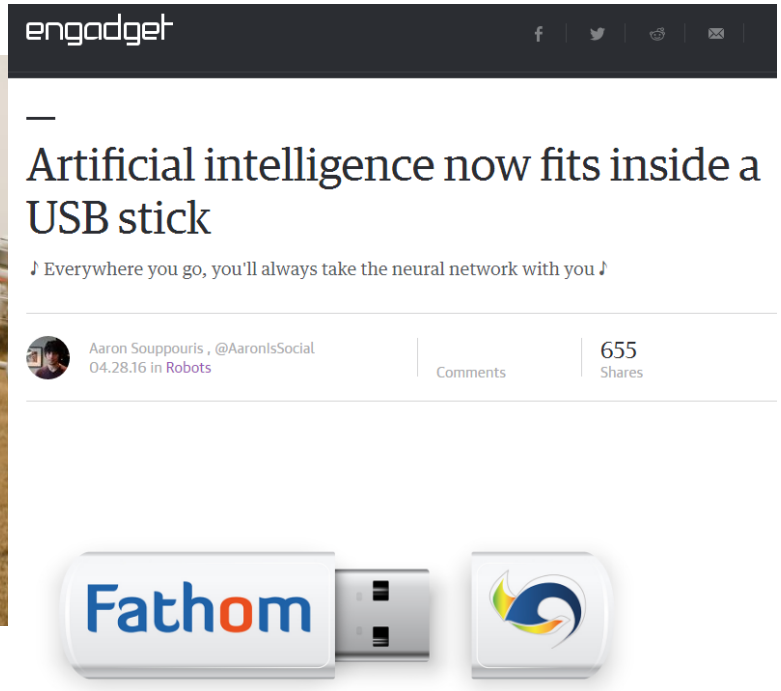
Vehicle Auto-diagnosis
Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

Item Location
Search of individual items in big surfaces like warehouses or harbours.

libelium
www.libelium.com

Case Study: Deep Learning Hardware

- Artificial Intelligence (AI)
- Machine Learning
  - a branch of machine learning
  - deep neural networks (DNN)
  - convolutional neural networks (CNN)
  - recurrent neural networks (RNN)



Case Study: Deep Learning Hardware

Movidius

Home About Applications Solutions Technology News Events

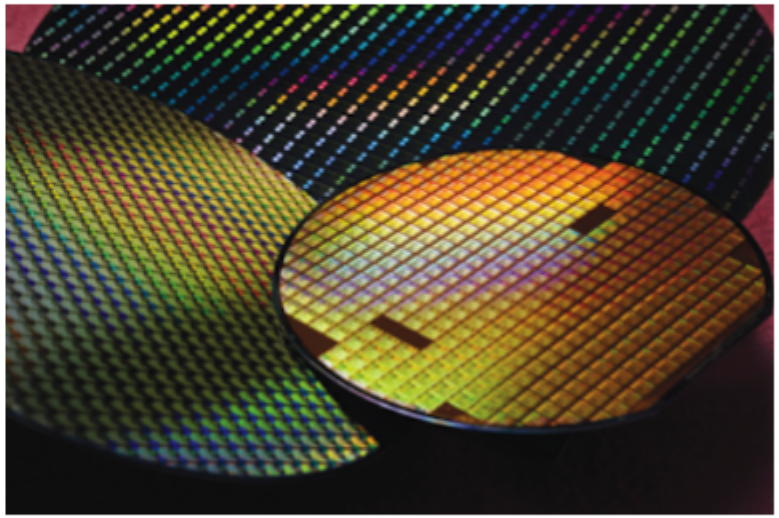
DEEPHI TECH

Main About Technology Product Video Team Contact Recruit Paper 中文版 | English

Discoverin

Nervana Engine delivers deep learning at ludicrous speed!

Nervana is currently developing the Nervana Engine, an application specific integrated circuit (ASIC) that is custom-designed and optimized for deep learning.



Training a deep neural network involves many compute-intensive operations, including matrix multiplication of tensors and convolution. Graphics processing units (GPUs) are more well-suited to these operations than CPUs since GPUs were originally designed for video games in which the movement of on-screen objects is governed by vectors and linear algebra. As a result, GPUs have become the go-to computing platform for deep learning. But there is much room for improvement — because the numeric precision, control logic, caches, and other architectural elements of GPUs were optimized for video games, not deep learning.

Industry Development Perspective

- Incumbent players and new comers

Course Overview

Objectives

Motivations

Structure

Logistics

Tentative Schedule

- W1: Intro to SoC and System Stack
- W2: CMOS Devices and Circuits
- W3: ASIC Design Methodology
- W4: In-Class Exam and Tool Tutorials

VLSI and ASIC (3 homework)

- W5: Fundamental Processor Concepts
- W6: Fundamental Memory Concepts
- W7: Fundamental Network Concepts

Architecture (3 labs)

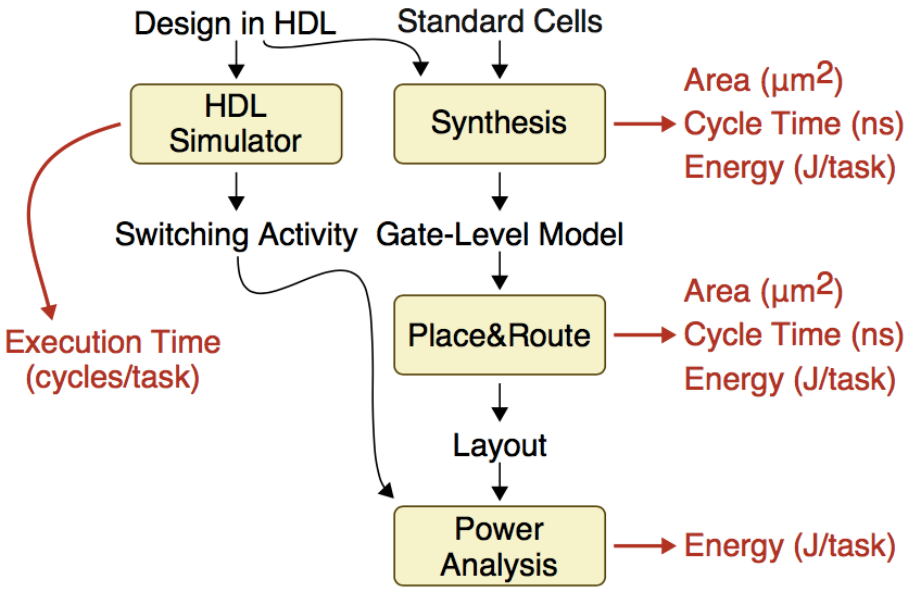
- W8: System Integration and Class Project Introduction
- W9: Spring Break
- W10: High-Level Synthesis (HLS)
- W11: HLS Tool and Techniques
- W12: Class Project Discussion
- W13: Reliability and Security

Advanced Top-Down Design (class project)

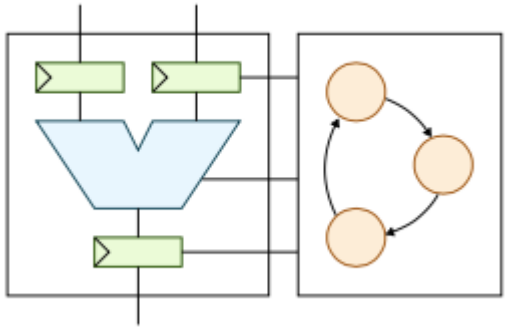
- W14: Future Research and Conclusion
- W15: Class Project Presentation

Labs: Deepen Understanding of Design and System

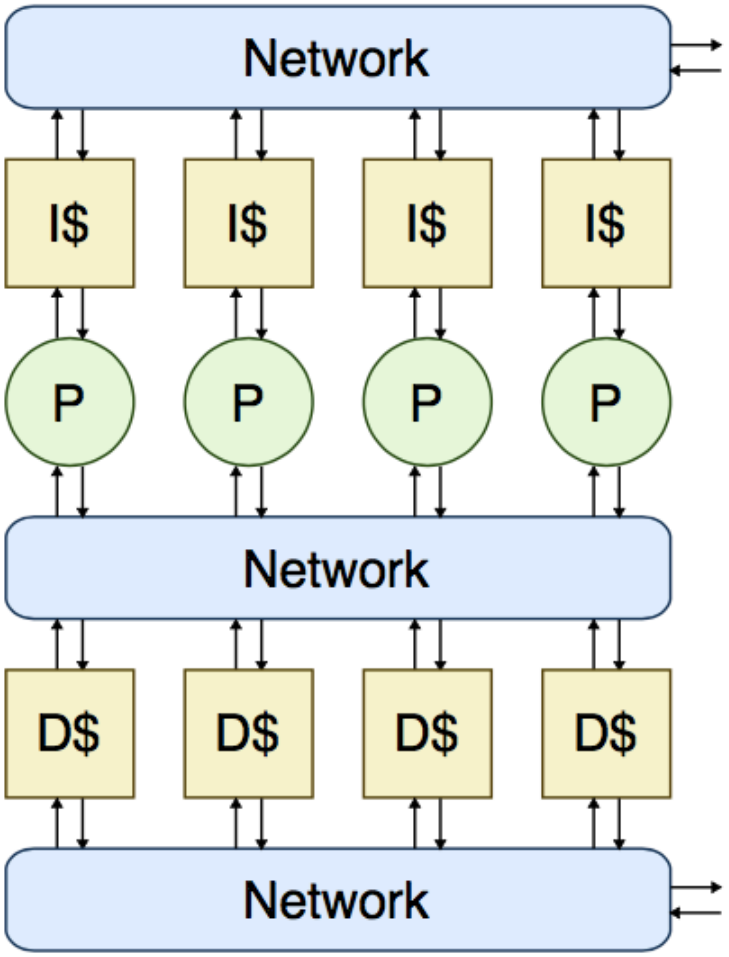
Lab 1: Design Flow



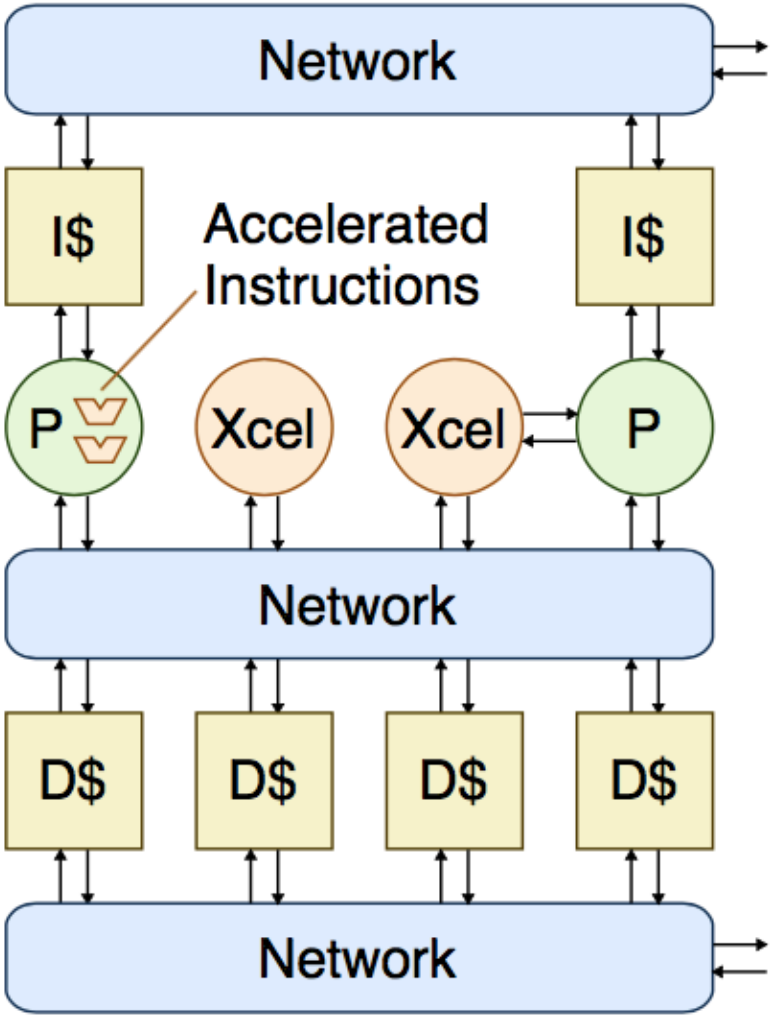
Lab 2: Multiplier

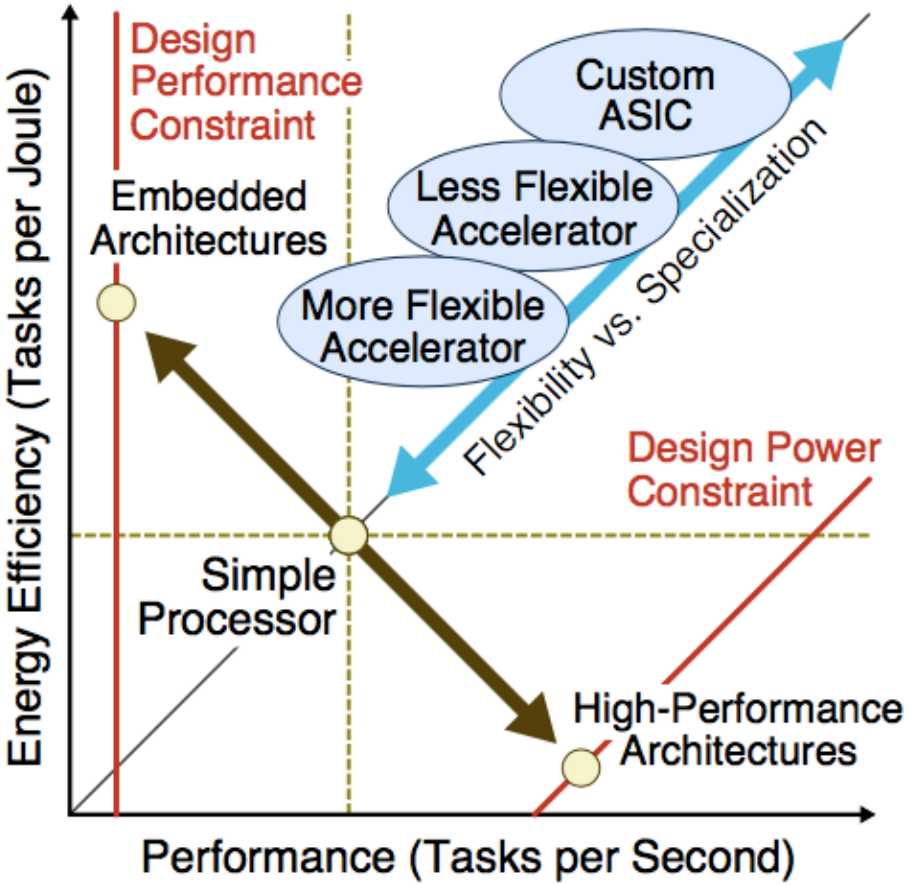


Lab 3: Memory Controller



Class Project: Application Specific Accelerator





Course Overview

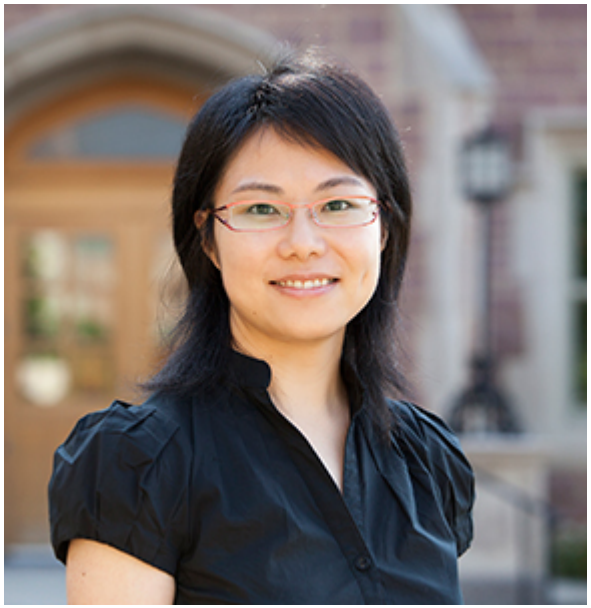
Objectives

Motivations

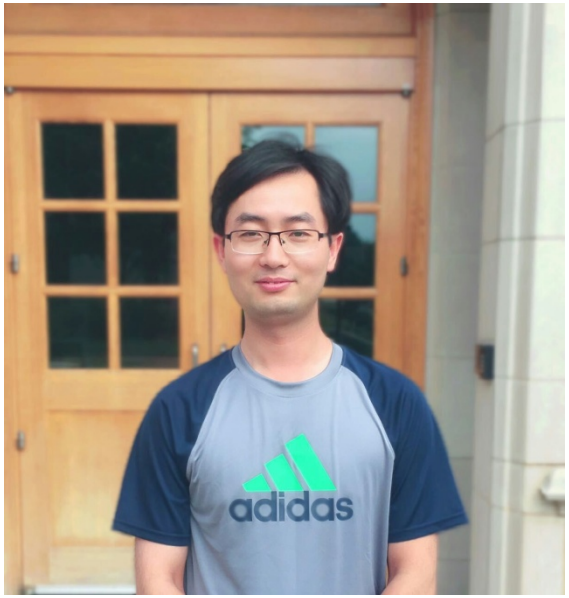
Structure

Logistics

Instructional Staff
(see homepage for contact info, office hours)



Xuan ‘Silvia’ Zhang
(Mon 4-5pm)



Dengxue Yan
(Thur 3:30-5pm)

Yunfei Gu
(Tue 3:30-5pm)

Prerequisites

- ESE 232: Introduction to Electronic Circuits
  - analysis and design of transistors
  - semiconductor memory devices
- ESE 260: Introduction to Digital Logic and Computer Design
  - combinational and sequential logic
  - logic minimization, propagation delays, timing
- Hardware description language (Verilog, VHDL)
- Recommended
  - basic computer architecture (CSE 362M)
  - basic design flow (ESE 461)
  - basic system implementation (CSE 462M)

Course Overview

- Course homepage:
  - http://classes.engineering.wustl.edu/ese566/
- Distribution
  - 30%: reading and learning
  - 70%: programming, debugging, design iteration
- Course load
  - one in-class exam
  - 4 homework (one per week)
  - 3 labs (one every two weeks)
  - one final project (3-person team)
- Philosophy
  - learner-directed instruction

Final Project

- Goal: learn by doing
  - Work in teams of 3
  - Topic fixed, implementation open-ended
  - Release around Week 9 (spring break)
  - Optimize design to meet/exceed performance goals
  - A custom designed IC chip as the end result

- Evaluation
  - Completion of the design flow
  - Performance achieved
  - Techniques applied
  - Proposal, mid-proj report, presentation, final report

Grading

- Engagement 5%
- In-Class Exam 10%
- Homework 20%
- Labs 30%
- Class Project 35%

- Policy:
  - 90% or above A
  - 80% - 89% B
  - 65% - 79% C
  - 45% - 64% D
  - 44% or below F

Policies

- Submission
  - labs and homework due Monday before lecture
  - fixed 3 slip days on homework and 3 slip days on lab
  - each team has 4 slip days on class project
- Discussion & Collaboration
  - learning through discussion
  - help classmates to understand concepts
  - sharing code or schematics not-allowed
- Plagiarism
  - zero tolerance
  - specify sources to avoid confusion

Textbook

- Lecture Slides and Notes
- Tutorials
- Documentations

- Recommended Textbook
  - “Computer System Design: System-on-Chip” by Michael J. Flynn and Wayne Luk
  - free ebook can be access via WashU lib

Is ESE 566A right for you?

- Advanced graduate-level fast-paced;
- 6~8 hours/week load (week1-5) ;
- 10~12 hours/week load (week6-14);
- lots of coding and debugging for lab and class projects;
- not the class to teach you coding;
- not the class to teach you Verilog;
- you have to work out your own solution.

Next Step After Lec #1

- Enrollment
  - decide to enroll or drop, capacity limited
- Piazza
  - sign up instruction to follow
  - questions relating to class will be directed to Piazza
- Github
  - sign up instruction to follow
  - all lab and class project submissions on Github
- SEAS account
  - design software on LinuxLab
  - tutorial to be released.
- Homework 1
  - will be posted on class website. due on 1/23 at 2:30pm.

Questions?

Comments?

Discussion?

Acknowledgement

Cornell University, ECE 5745