EEL 5764 Computer Architecture

Sandip Ray
Department of Electrical and Computer Engineering
University of Florida
sandip@ece.ufl.edu
http://sandip.ece.ufl.edu

Lecture 1-2:
Introduction and Technology Trends

About This Course

- **Principles of modern computer architectures**
  - →All the "things" behind modern computers
- **Quantitative analysis approaches to design tradeoffs in terms of**
  - →Cost, performance, power/energy efficiency etc.
- **Why this course?**
  - →Have better sense of future by learning from the past
  - →Design better computers to solve real-world problems
  - →Write better software
  - →Get a good grade in order to continue your grad. study

Course Objectives

- **Understand various aspects of modern computer architecture design**
  - →Inst set design, mem hierarchy, pipelining, multi-processing, etc
- **Understand quantitative approaches to analysis and evaluation of design tradeoffs**
  - →Cost, performance, power/energy, etc
- **Understand the interplays between the computer architecture & software**
  - →How they affect each other

Instructor

- **Dr. Sandip Ray**
  - → Office: BEN 323
  - → Office Hours: 2:00-3:00pm Tue (or by Appointment)
  - → Phone: (352) 392-1605
  - → Email: sandip@ece.ufl.edu
- **Research interests:**
  - → **S**ecure **I**ntelligent, **R**esilient Architectures for **I**nternet of Things Applications
  - → **Current research projects:**
    - ➤ Robust Autonomous vehicles,
    - ➤ System-on-Chip Security,
    - ➤ Trustworthy Machine-learning Systems
- **Past Experience**
  - → Microarchitecture design, post-silicon validation at Intel labs
  - → Automotive architecture (and validation) at NXP

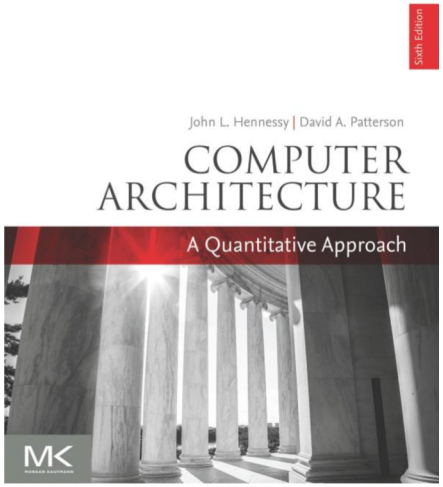
Course Administration

- **Teaching Assistant:** Kshitij Raj
  - → Office: TBD
  - → Office Hours: TBD
- **Teaching Assistant:** Atul Prasad Deb Nath
  - → Office: TBD
  - → Office Hours: TBD
- **Teaching Assistant:** Jiaqi Zhang
  - → Office: TBD
  - → Office Hours: TBD

- **Please contact teaching staff through Canvas.**
- **For common questions, use the Discussion tab.**

Course Material

- **Text:**
  - → Computer Architecture: A Quantitative Approach by Hennessy & Patterson, 6th Edition, Morgan Kaufmann (**Required**).
  - → Computer Organization and Design RISC-V Edition, Morgan Kaufman (Additional Material)
- **Course Web page:**
  - → **Public:** http://sandip.ece.ufl.edu/classes/eel-5764-19f/index.html
  - → Canvas: Will contain lecture slides and other class information



It is your responsibility monitor Canvas regularly for various information and announcements

Evaluation and Grading Information

Assignment	Percentage of Final Grade
Homework Sets (3-4)	20%
Midterm Exam 1	25%
Midterm Exam 2	25%
Term Paper	30%
TOTAL	100%

- For homework grades talk to the TAs.
  →All HW must be submitted electronically to Canvas.
- Other queries: TAs and Instructor
- Mid-term dates will be announced by mid-September
- Term Paper topics will be discussed around third week of September
- Any grade dispute should be reported within one week.

Final Grading Scale

Percent	Grade
93.0 - 100	A
90.0 – 92.9	A-
87.0 – 89.9	B+
83.0 – 86.9	B
80.0 – 82.9	B-
77.0 – 79.9	C+
73.0 – 76.9	C
70.0 – 72.9	C-
67.0 – 69.9	D+
63.0 – 66.9	D
60.0 – 62.9	D-

No curves!

What You Should Know before the course

- **Basic logic design & computer organization**
  - → Information representation, Boolean logic
  - → Logic building blocks: flip-flops, multiplexers, decoders, shift registers
  - → Datapaths, register transfer level design
  - → Knowledge of digital systems in terms of computer organization and logic level design
  - → Instruction sets: opcodes, operands, memory addressing, etc
  - → Input/output and communication; memory hierarchy.
- **Basic knowledge of C/C++ and Linux**
- **If you think you do not have the requisite background but still want to take the course, talk to me asap**

Expectations from You

- Do all homework sets (start early!).
- Additional problem sets will be provided (not graded)
  → But you should work on them to do well in mid-terms
- Attend lectures (come **on time**).
- Take notes whenever necessary.
- Prepare for class.
- Ask questions, participate in class.
- For exams, read notes, book and supplementary materials.
- The course will be moderately heavy – work hard!
- Be honest to yourselves!

More on Homework Assignments

- Do them to truly understand the material, not just to get the grade.
- Content from lectures, readings, discussions.
- All homework writeups ***must* be your own work**, written up individually and independently.
  - →Peer discussions are encouraged.
  - →Copying each others' work or from previous work is strictly forbidden!
- There is a fine line between plagiarism and collaboration.
- **No late homework submissions are accepted.**
  - →But the homework with lowest grade will be dropped.

Cheating and Academic Dishonesty

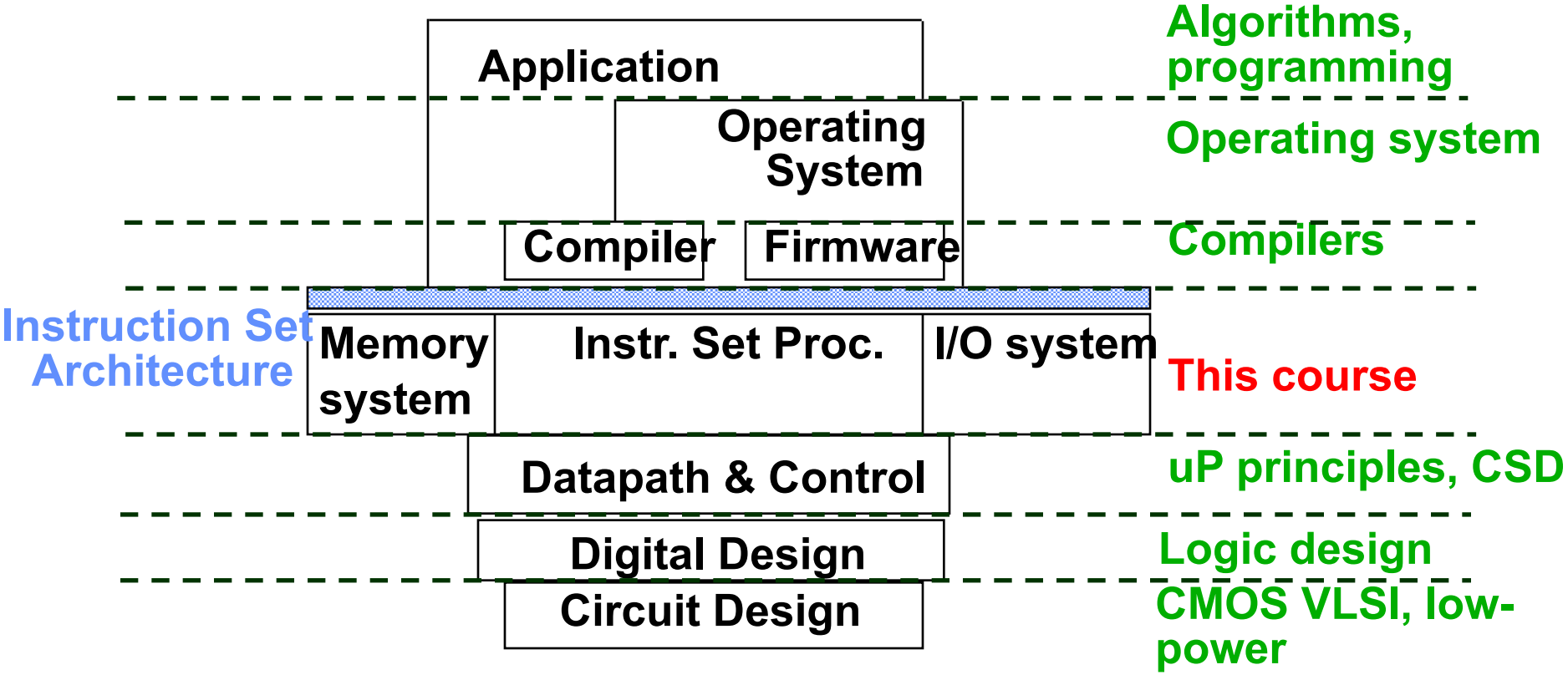
- **Absolutely no form of cheating will be tolerated**
- You are all adults and we will treat you so
- See syllabus, UF Policy, and UF Honor code
- Cheating → failing grade together with referral to UF Honor Court (no exceptions)

Course Content

Introduction	Ch. 1
Memory hierarchy design	Ch. 2, App. B
Instruction set principles	App. A
Instruction level parallelism and pipelining	Ch. 3, App. C
Data level parallelism and vector processing	Ch 4, App. G
Thread-level parallelism and multiprocessors	Ch. 5
Interconnection networks	App. F

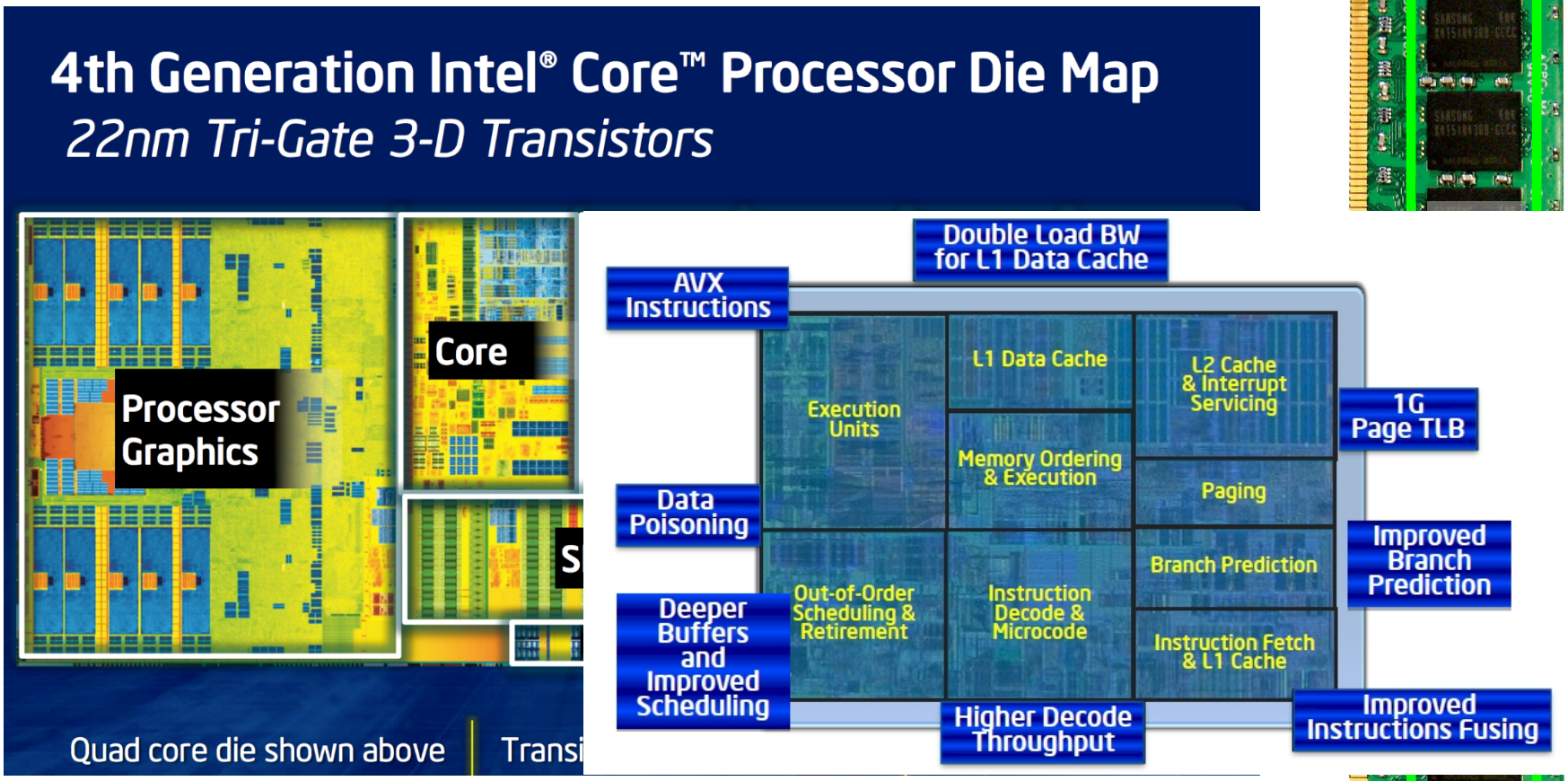
- Other current topics may be introduced if time permits.

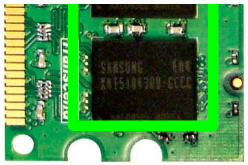
How Do the Pieces Fit Together?



- Coordination of many *levels of abstraction*
- Under a rapidly changing set of forces
- Design, measurement, *and* evaluation

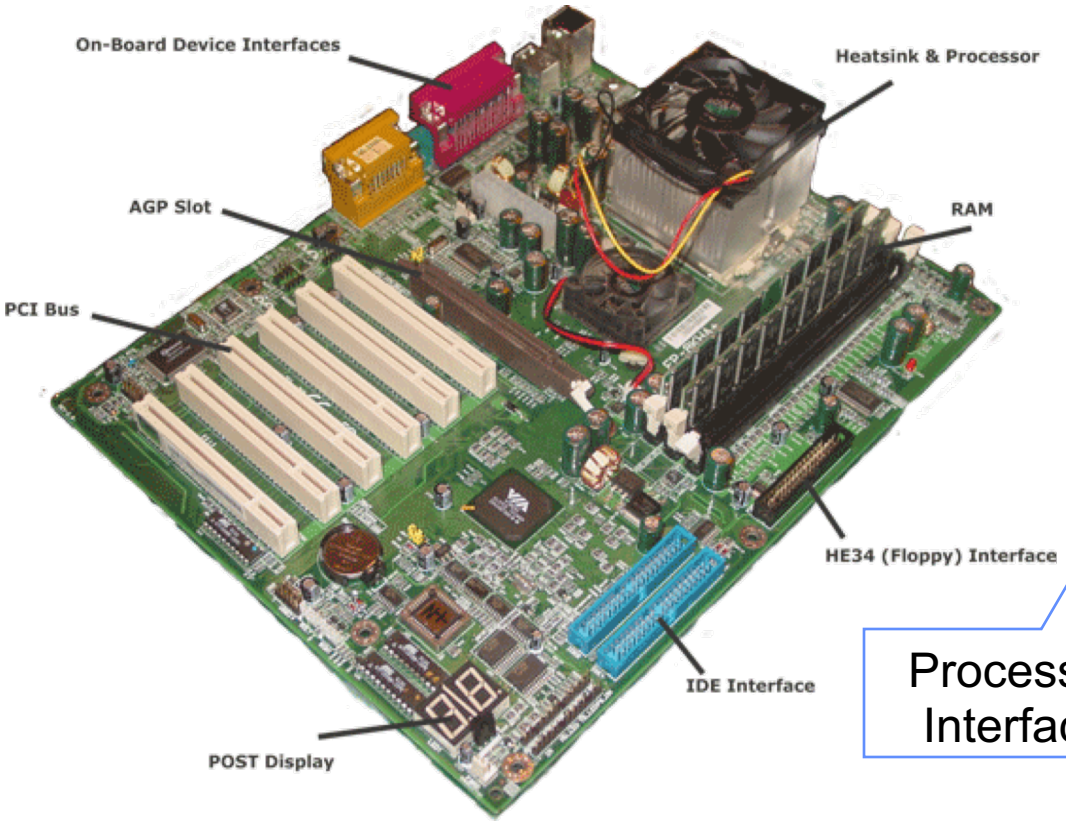
An Example

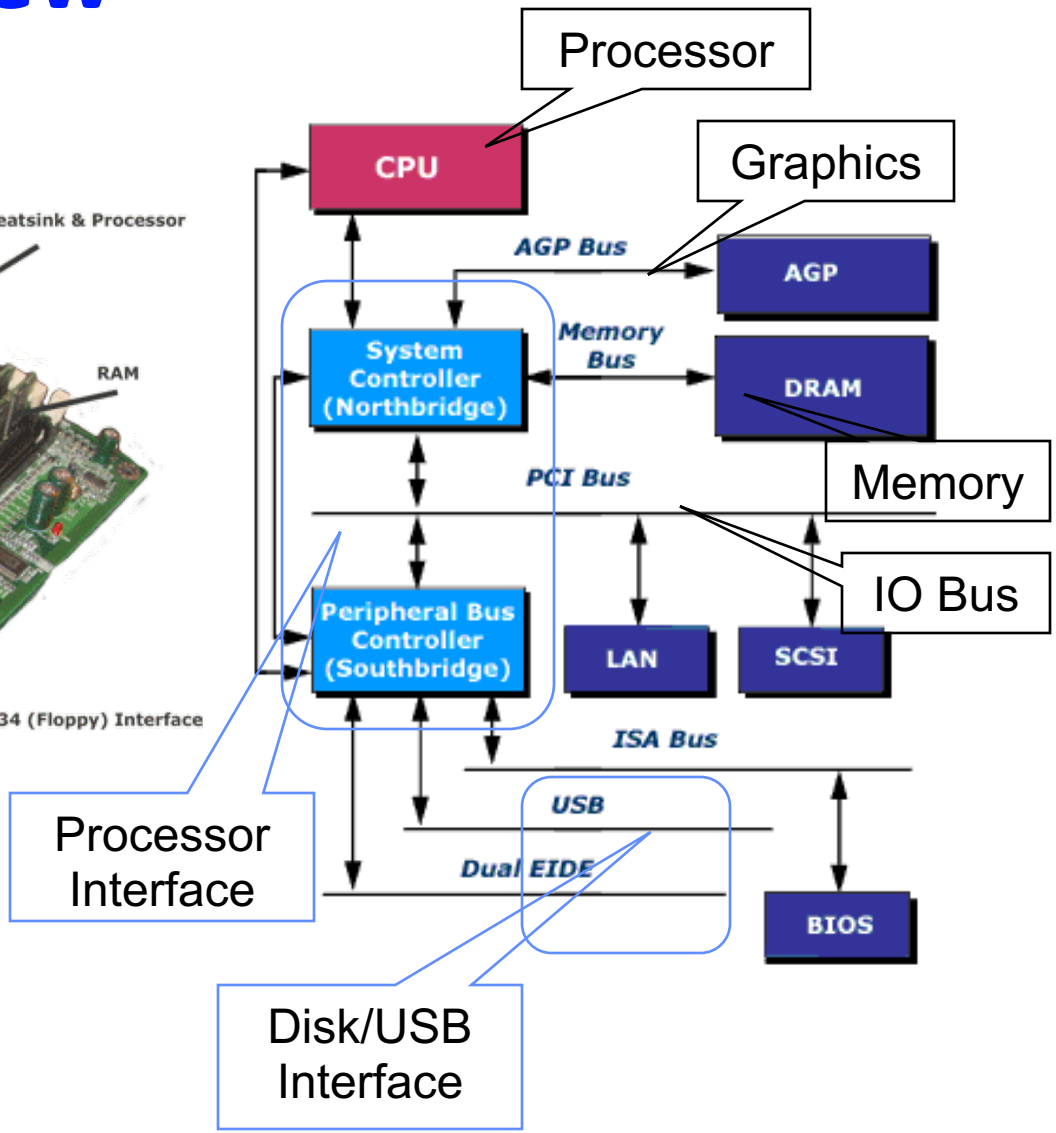




- The objective of this course is to touch each of these components

A System-Level View





Computer Architecture

- What is it?
- Why is it important?

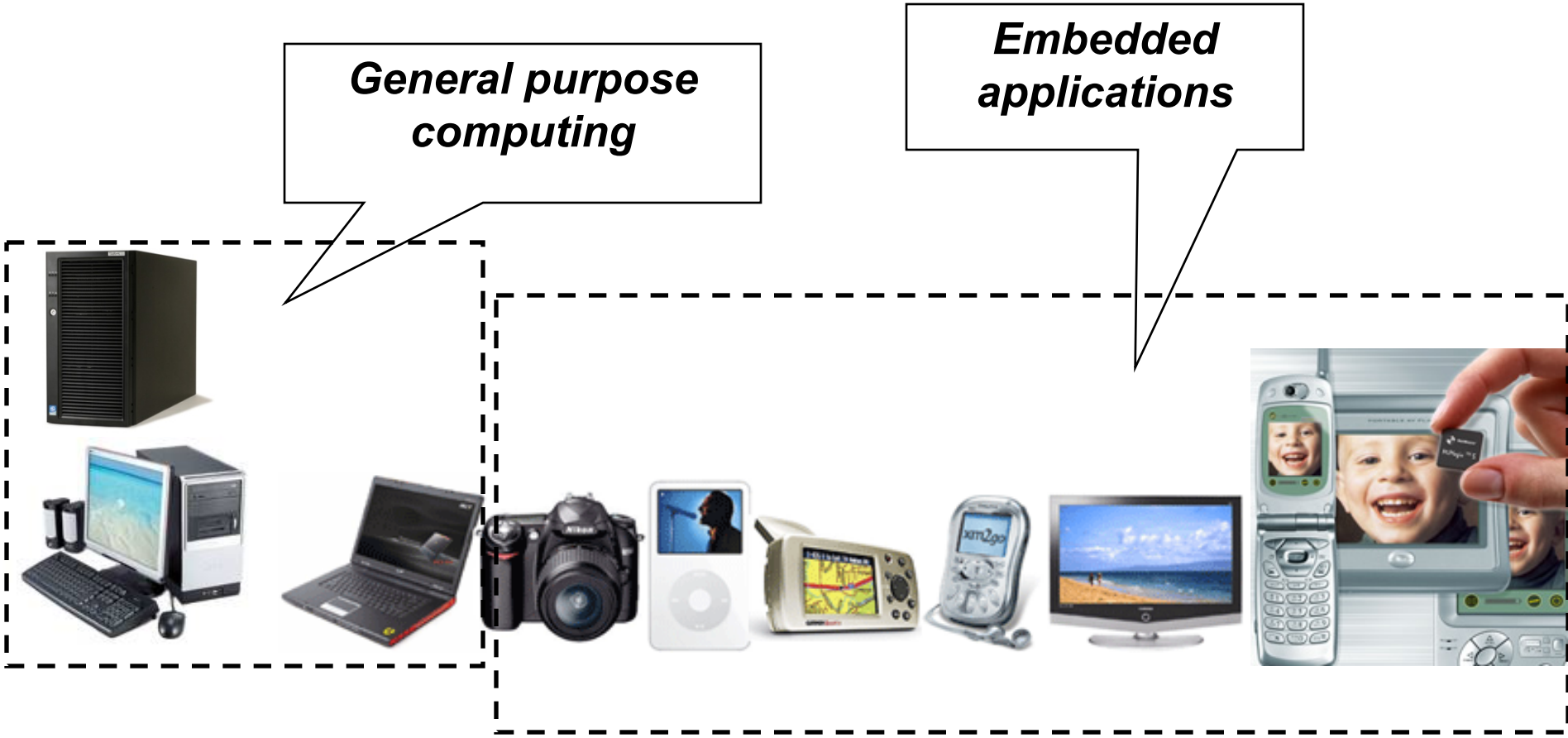
What is Computer Architecture?

"Computer architecture is the science and art of <u>selecting</u> and <u>interconnecting</u> hardware components to create a computer that meets functional, performance, and cost goals." *

** G. Sohi, UW*

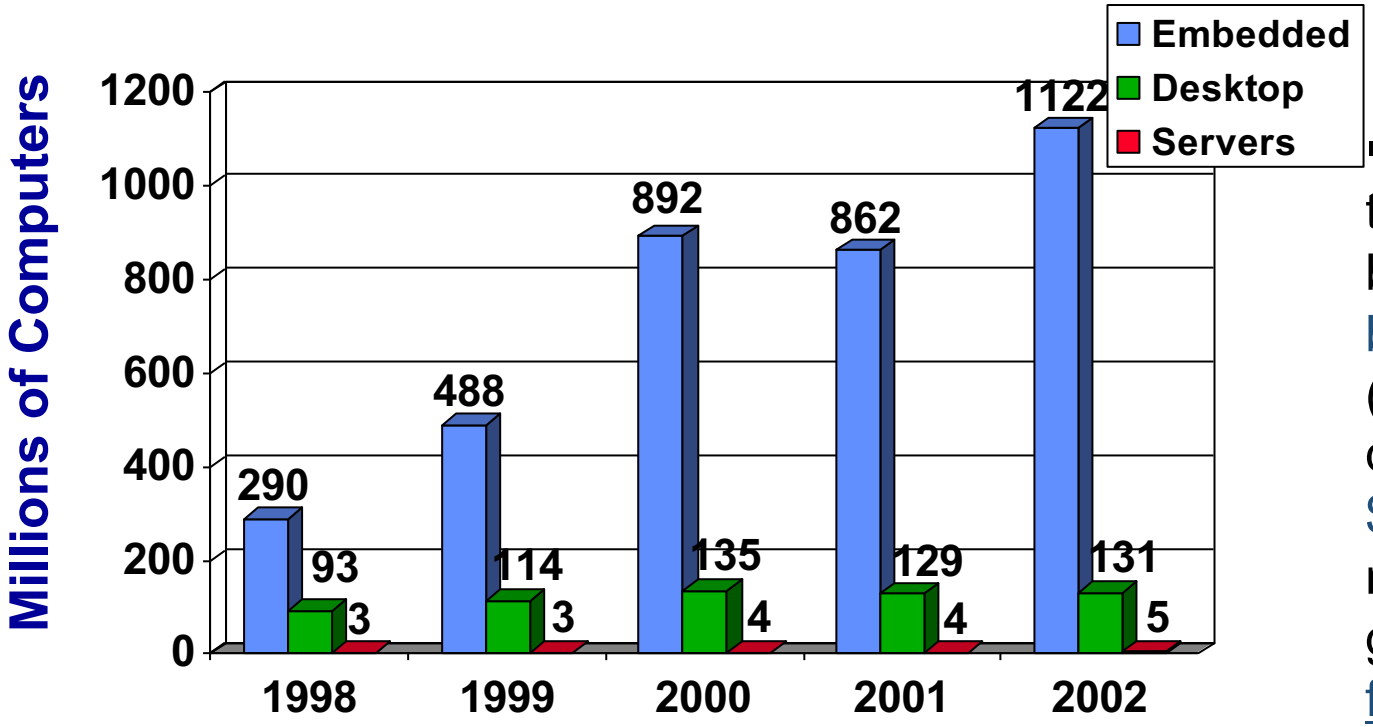
Think about architecture of buildings!

Ubiquitous Usage of Computers



- Today's high-end computers have
  - → Billions of transistors in a chip
  - → Tera-flops (10^12 FPOS) processing power

Where is the Market?



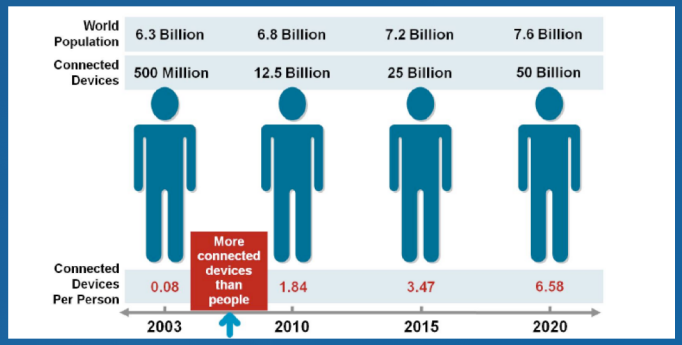
- It is estimated that there will be between 15 and 20 billion devices (small embedded devices) with a $900 Billion USD market, which is growing twice as fast as the PC market.

- **Embedded applications:** mobile electronics, automobile electronics, home electronics, communications and networking, healthcare products, sensor networks, gaming hardware, defense applications and many others!

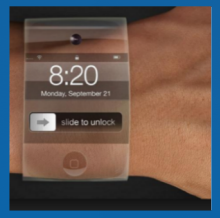
The Internet of Things Regime

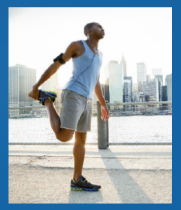
The IoT Regime is the **point of time** when the number of connected computing devices exceeds the human population. --- Cisco White Paper, 2011

IoT involves a **connected network of physical objects** or "things" embedded with electronics, software, and sensors, to enable it to **behave smartly** and achieve greater value. --- Wikipedia



Source: Cisco





Smart Dust

Computers smaller than a grain of sand can be spread anywhere to measure chemicals in soil or problems in human body

Source: Intel

Smart City

Fixed and mobile sensors dispersed throughout the city of Dublin are already creating a picture of what's happening and will help the city in the time of crisis.

Source: Intel

Classes of Computers

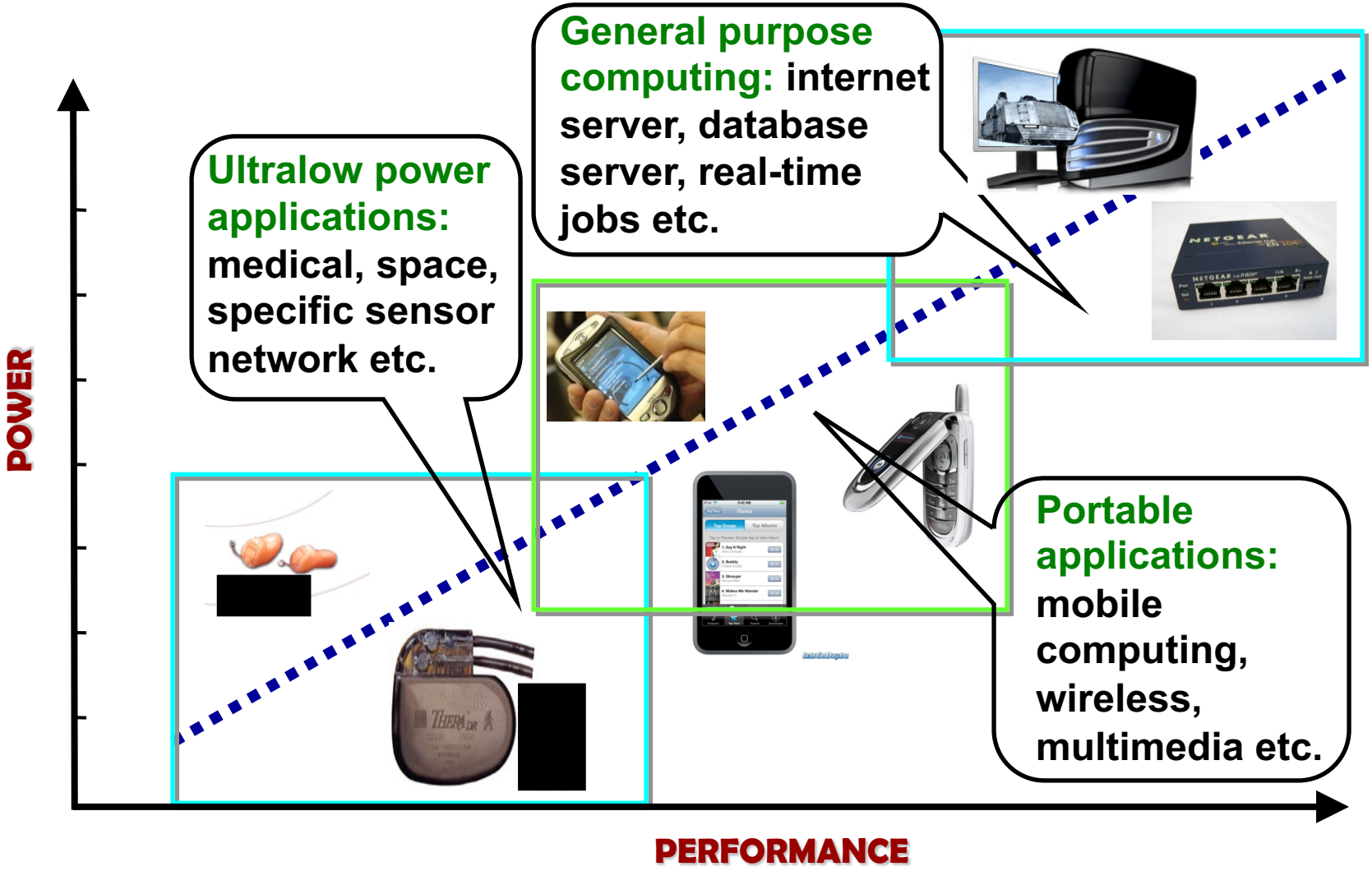
Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	$100–$1000	$300–$2500	$5000–$10,000,000	$100,000–$200,000,000	$10–$100,000
Price of micro-processor	$10–$100	$50–$500	$200–$2000	$50–$250	$0.01–$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

- Shipments: 1.9B PMD, 350M Desktop, 19B Embedded.
- Desktop has largest $ market share.

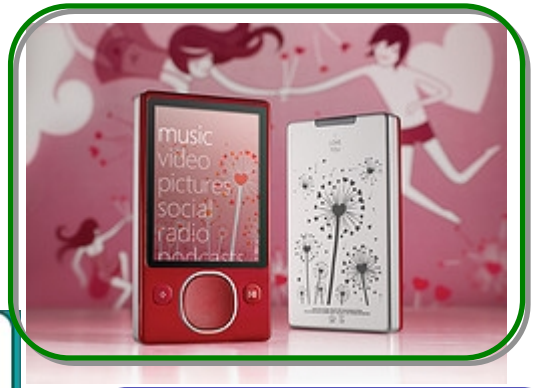
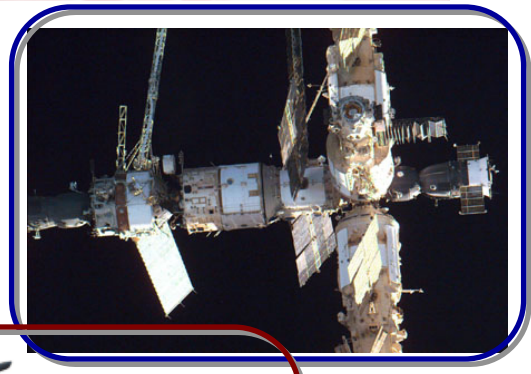
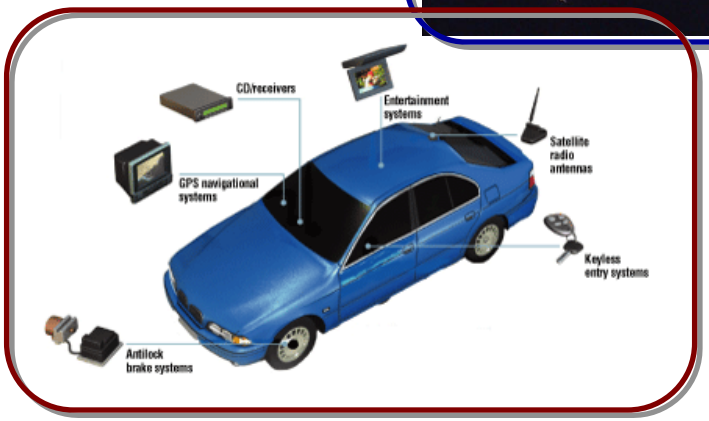
How can we design complex computing systems in this ecosystem to work efficiently, reliably, and according to specification?

- Under complex power/performance requirements
- In the presence of malicious devices
- When developed under very aggressive time-to-market requirements

Computer Applications



Different applications have different power-performance demands

System Design Parameters

- Performance (Speed)
- Cost
- Power (static + dynamic)
  - →Peak power
  - →Average power
- Robustness
  - →Noise-tolerance
  - →Radiation-hardness
- Testability
- Reconfigurability
- Time-to-market etc.