COMP60001 / COMP70086
Advanced Computer Architecture
Chapter 1.1

Introduction
Is this course for you? How will it work? What will you learn?

October 2026

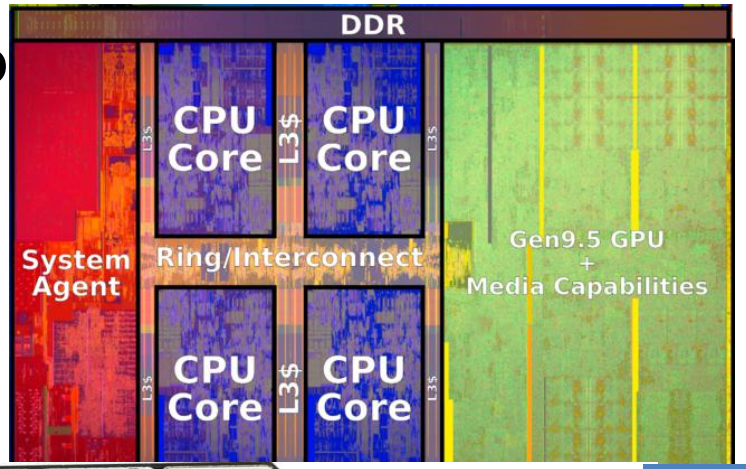
Paul H J Kelly

These lecture notes are partly based on the course text, Hennessy and Patterson's *Computer Architecture, a quantitative approach, and on the lecture slides of David Patterson's Berkeley course (CS252)*

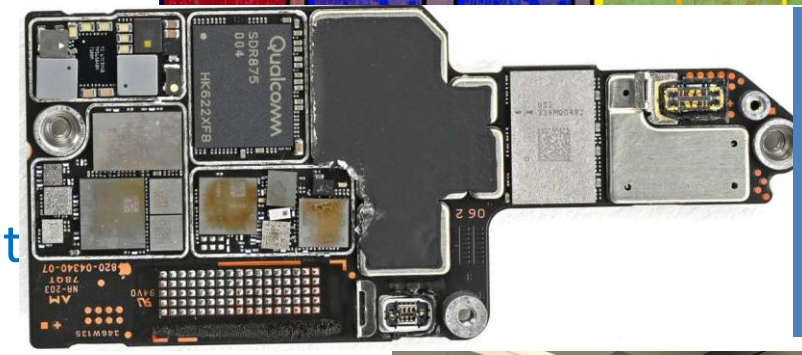
Course materials online on
https://scientia.doc.ic.ac.uk/2627/modules/60001/materials and
https://www.doc.ic.ac.uk/~phjk/AdvancedCompArchitecture/aca20/

What is this course about?

- How the latest microprocessors *work*
- Why they are built that way – and what are the alternatives?
- How you can make software that uses the hardware in the best possible way
- How you can make a compiler that does it for you
- How you can design a computer for *your* problem
- What does a *big* computer look like?
- What are the fundamental big ideas and challenges in computer architecture?
- What is the scope for theory?



Intel Kaby Lake (my old laptop)
https://en.wikichip.org/wiki/intel/core_i7/i7-8650u

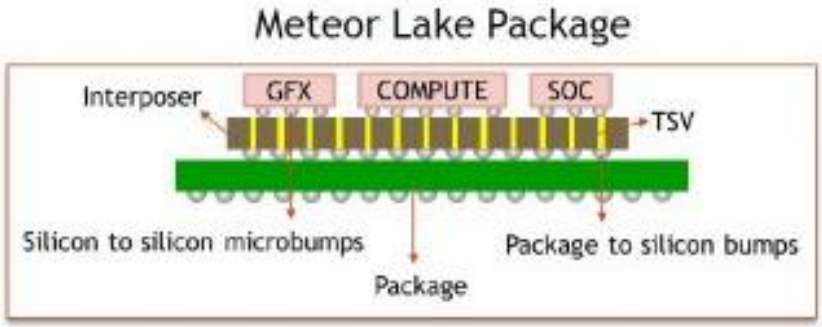


Apple iPhone 18 Pro Max
https://www.ifixit.com/News/119329/inside-the-tiny-unfixable-eye-iphone-18-pro-and-pro-max-teardown

El Capitan supercomputer at Lawrence Livermore National Labs
https://en.wikipedia.org/wiki/El_Capitan_(supercomputer)

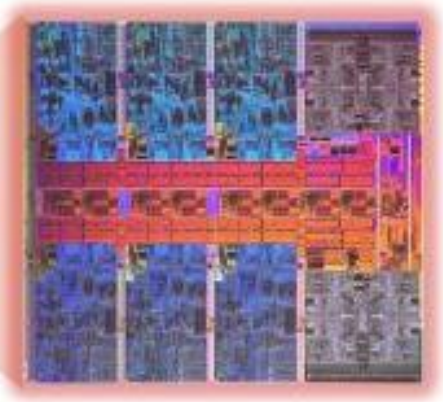
Google Data Centre, Waltham Cross, Herts
https://www.youtube.com/watch?v=_GP58P9SnNw

My laptop (LG Gram Pro 17, Intel Ultra 7 155H)



Compute tile:

- 6 “performance” cores
- 8 “efficient” cores
- 2 “low-power efficient” cores
- 24MB cache



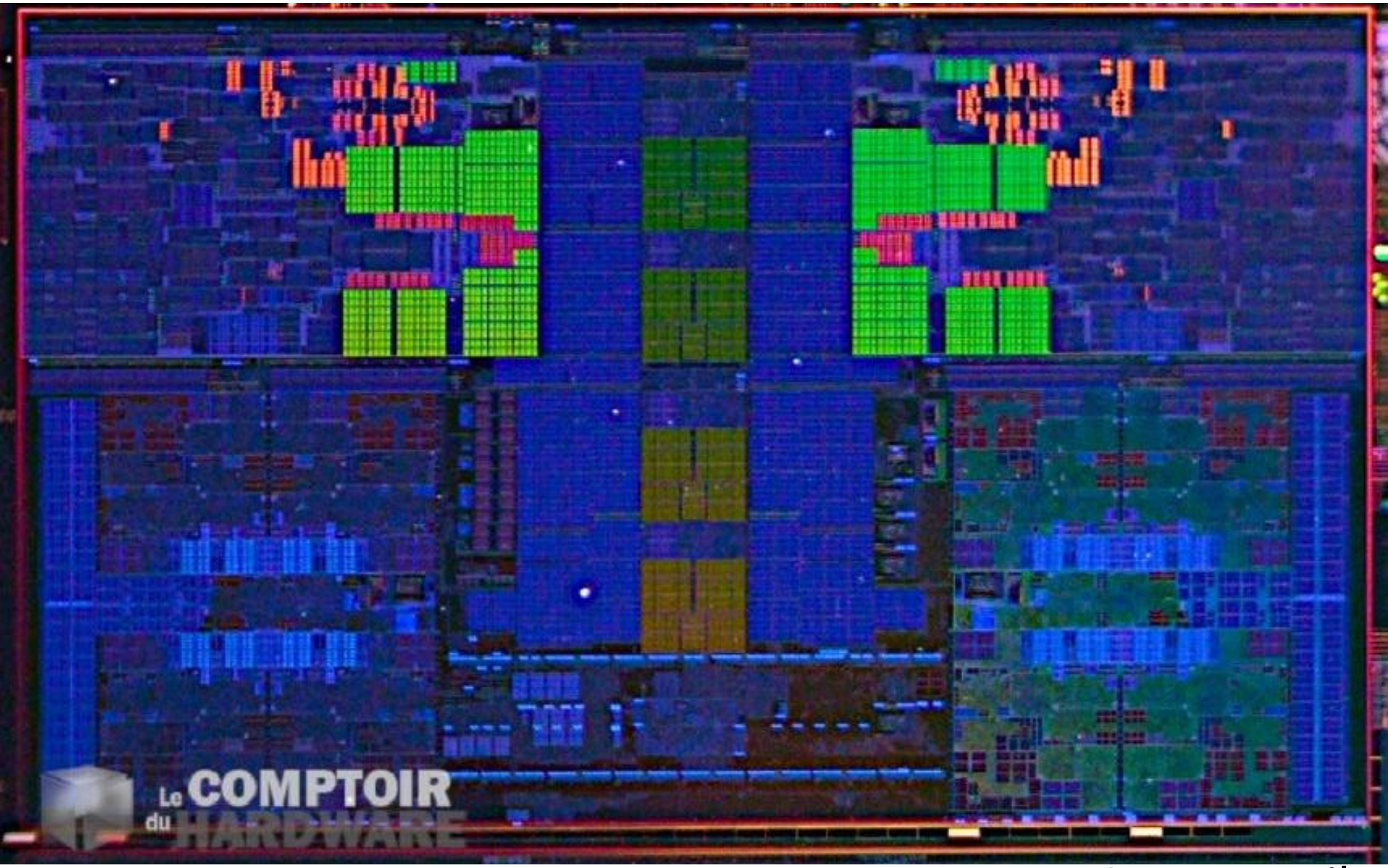
- Four separate chips (“dies”): Compute, Graphics, SOC, IO
- Packaged on an “interposer” chip
- Interposer connects the four chips together
- All inside the packaged chip product

Image from:
https://semianalysis.com/2022/05/26/meteor-lake-die-shot-and-architecture/#

See also:
https://www.hc34.hotchips.org/assets/program/conference/day2/Mobile%20and%20Edge/Meteor_Lake_Hotchips%20_%20Wilfred%20_%20final_submit%20(1).pdf

Meteor Lake mobile (Intel Core Ultra 5 115U)

(a smaller chip than in my laptop, 2 P-cores, 4 E-cores, 2 LP-cores)

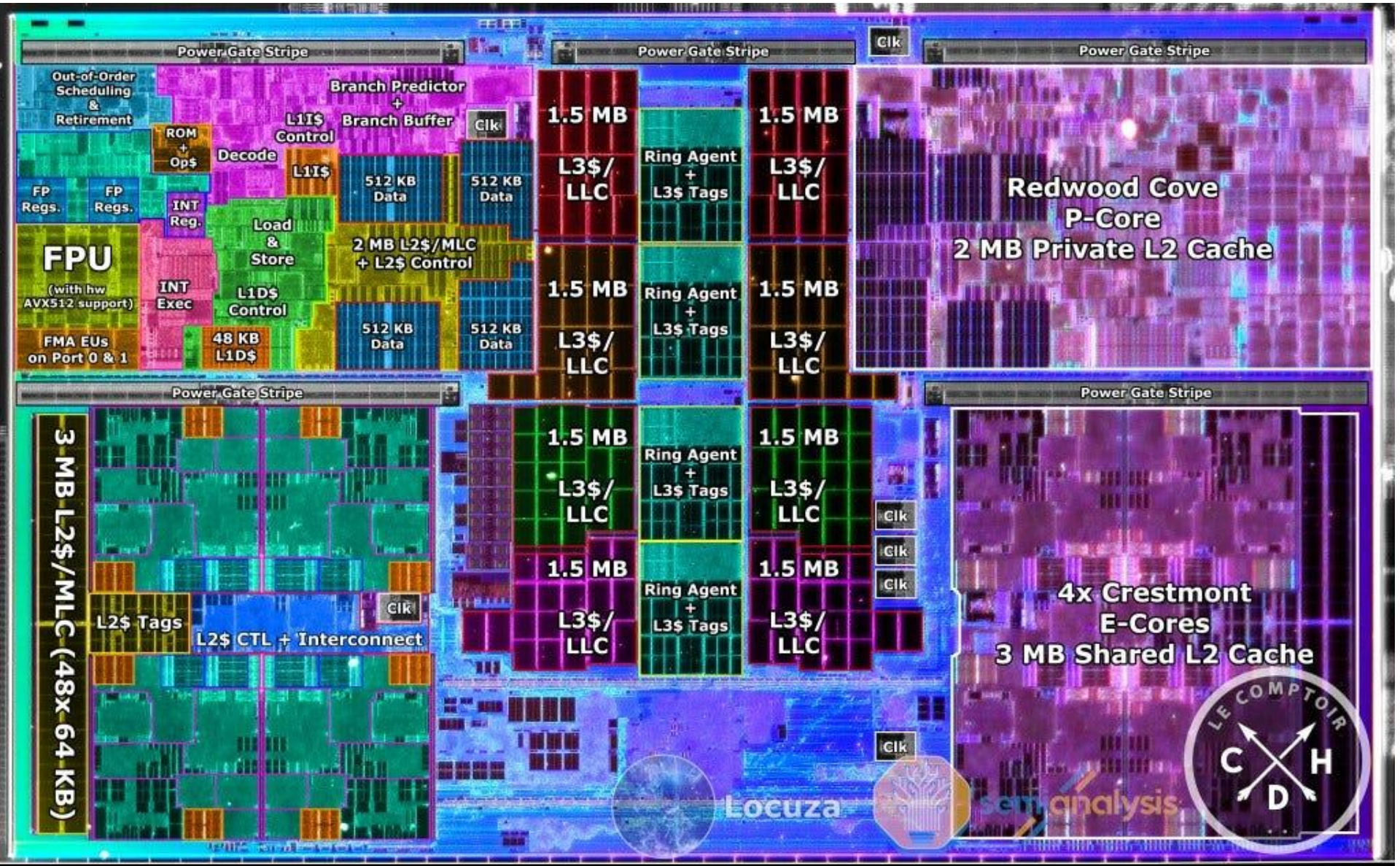


https://semianalysis.com/2022/05/26/meteor-lake-die-shot-and-architecture/#

Compute tile

Meteor Lake mobile (Intel Core Ultra 5 115U)

(a smaller chip than in my laptop, 2 P-cores, 4 E-cores, 2 LP-cores)

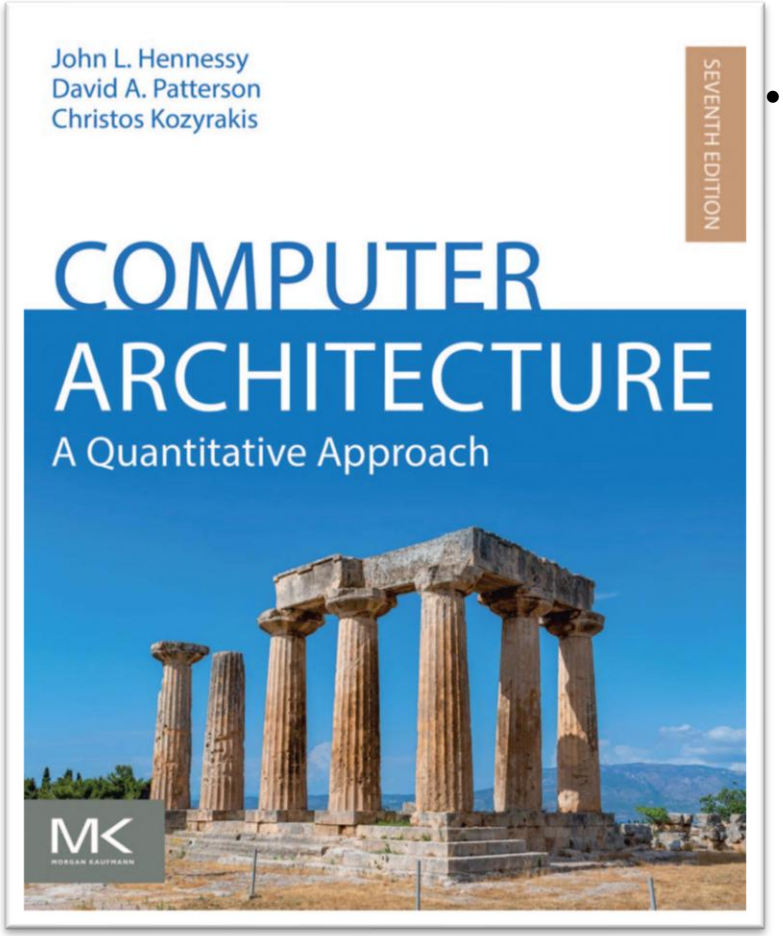


https://semianalysis.com/2022/05/26/meteor-lake-die-shot-and-architecture/#

Meteor Lake die shot by Nicolas Derumigny / comptoir-hardware.com - www.comptoir-hardware.com

Die shot interpretation by Locuza - https://twitter.com/Locuza_

This course's starting point:



- Computer Architecture: A Quantitative Approach (7th Edition)

  John L. Hennessy, David A. Patterson, Christos Kozyrakis

  - 936 pages. Morgan Kaufmann (2025)
  - ISBN: 9780128119051
  - Price: around £70 (shop around!)
  - Publisher's companion web site:
    - https://www.educate.elsevier.com/book/details/9780443154065
    - Textbook includes some vital introductory material as appendices:
    - Appendix C: tutorial on pipelining (read it NOW)
    - Appendix B: tutorial on memory hierarchy (read it NOW)
  - Further appendices (some in book, some online) cover more advanced material (some very relevant to parts of the course), eg
    - Networks
    - Parallel applications
    - Embedded systems
    - Storage systems
    - VLIW
    - Datacentre architecture
    - Domain-specific architectures
    - Computer arithmetic (esp floating point)

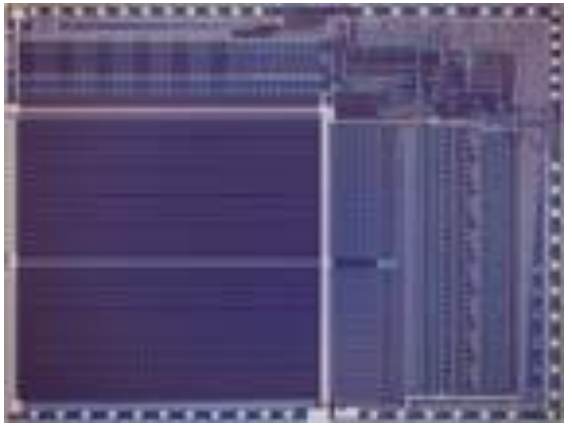
Who are these guys anyway and why should I read their book?

- John Hennessy:
  - Founder, MIPS Computer Systems
  - President (2000-2016), Stanford University
  - Board member, Cisco, chair of Alphabet Inc (parent company of Google)
  - The “godfather of Silicon Valley” (Wikipedia)

RAID-I (1989) consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software.

- David Patterson
  - Leader, Berkeley RISC project
  - RAID (redundant arrays of inexpensive disks)
  - Professor, University of California, Berkeley
  - President of ACM 2004-6
  - Served on Information Technology Advisory Committee to the US

By Peg Skorpinski - Subject of pictures emailed it upon request, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=3207893



http://www.cs.berkeley.edu/~pattrsn/Arch/prototypes2.html

RISC-I (1982) Contains 44,420 transistors, fabbed in 5 micron NMOS, with a die area of 77 mm^2, ran at 1 MHz. This chip is probably the first VLSI RISC.

Joint winners of the 2017 ACM Turing Award

“For pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry”

Who are these guys anyway and why should I read their book?

- Christos Kozyrakis:
  - Professor at Stanford
  - University of Crete, UC Berkeley, EPFL, Stanford, NVIDIA, various startups
  - Phd 2002 “Scalable Vector Media-processors for Embedded Systems”

https://engineering.stanford.edu/news/2015-maurice-wilkes-award-presented-christos-kozyrakis

Alternative textbook:

- **Next-Gen Computer Architecture: Till the End of Silicon**

  Smruti R Sarangi (IIT Delhi), ISBN 8119510143

  - Full text available for free download at:
    - https://www.cse.iitd.ac.in/~srsarangi/advbook/index.html
  - Slides and lecture videos also available
  - Often more detailed that Hennessy and Patterson
    - Particularly on subtle details of out-of-order CPU architecture, for example
    - Section III on Advanced Topics provides excellent coverage of
      - » Power and Temperature
      - » Reliability
      - » Secure architectures
      - » Side-channel vulnerabilities

Course organisation

- Lecturers:
  - Paul Kelly – Leader, Software Performance Optimisation research group
  - David Schall – new lecturer working in Computer Architecture
  - Jacky Wong – PhD student, Teaching Scholar, Course Support Leader
    - With help from other PhD students as well
- Nominally four lecture hours per week, up to two hours “synchronous”
- In the last couple of lectures we will spend some time on exam preparation

- Assessment:
  - Exam
    - The exam will take place in last week of term
    - The goal of the course is to teach you how to think about computer architecture
    - The exam is designed to test your thinking, your understanding – not your memory – have a look at the past papers
  - Coursework
    - You will be assigned two coursework exercises
    - You will learn about using simulators, and experimentally evaluating hypotheses to understand system performance
    - You will get introduced to the research frontier in the field

Course overview (plan)

- Ch1
  - Review of pipelined, in-order processor architecture and simple cache structures
- Ch2
  - Dynamic scheduling, out-of-order
  - Register renaming
  - Speculative execution
- Ch3
  - Branch prediction
- Ch4
  - Caches in more depth
  - Software techniques to improve cache performance
  - Virtual memory and protection
- Ch5
  - Side-channel vulnerabilities
- Ch6 non-examinable
  - Static instruction scheduling
  - Software pipelining
  - instruction-set support for speculation and register renaming
- Ch7
  - Multi-threading
- Ch8
  - Data-parallelism, SIMD and vector
- Ch9
  - Graphics processors - GPUs
- Ch10
  - Shared-memory multiprocessors
  - Cache coherency
  - Atomicity, consistency
  - Large-scale cache-coherency; ccNUMA. COMA

- Lab-based coursework exercise:
  - “Exploration”: Simulation study
  - “Evaluation”: summarise and evaluate a recent research paper in computer architecture
- Exam:
  - Partially based on recent processor architecture article, which we will study in advance (see past papers)

Major milestones

- Why might it help to re-schedule instructions?
- Why might it help to re-order loop iterations?
- How can we dynamically reschedule instructions when a cache miss occurs?
- What happens when we mis-predict a branch?
- How can we speculatively pass data from a store to a load from the same address?
- How can we support translation from virtual addresses to physical addresses?

- Parallelism:
  - Instruction-level parallelism
  - Vectorisation
  - Memory parallelism
  - Shared-memory multicore parallelism
  - Distributed-memory parallelism
- Cache coherency
  - What guarantees do we have about what one core can observe about the progress of another core?

Spectre and Meltdown:

Major security vulnerabilities

Present in almost all modern CPUs

Some mitigations are possible

GPUs: graphics processing units

For graphics

For compute

We will see how GPUs are “just a point in the CPU architecture design space”

External Students – Registration for DoC Courses

1. Apply at: `https://dbc.doc.ic.ac.uk/externalreg/`
2. Then,
   - Your department's endorser will approve/reject your application
3. If approved,
   - DoC's External Student Liaison will approve/reject your application
4. If approved (again!),
   - Students will get access to DoC resources (DoC account, CATE, ...)
   - No access after a few days? Check status of approval and contact relevant person(s)

Key Dates

- Exams for DoC $3^{rd}/4^{th}$ yr. courses take place at the end of the Term in which the course is taught
- Registration for exams opens in November for Autumn courses and end January for Spring term courses

If in doubt, read the guidelines available at the link above ☺

Main points:

- If you have studied computer architecture before, you should be able to take this course
  - Do you know what a pipeline stall is? Do you know what a cache miss is?
  - You will also need to read some C code, do a little Linux command-line and bash-scripting
- By the end you will understand the main features and design alternatives in computer architectures widely used today
  - The “microarchitecture” of single cores, for both low power and high performance
  - Multicore systems, including how the cores are connected and how memory consistency is maintained
  - Graphics processors – at least from a compute point of view (rather than graphics)
  - Large-scale computer systems, supercomputers
- The course’s examinable content is defined by the lecture slides, but you will benefit from reading more widely
- The textbook provides both more depth and more breadth

- There will be two assessed coursework exercises:

  (1)“Exploration” – find a single-core microarchitecture configuration that runs a given program with the lowest total energy

  (2)“Evaluation” – write a brief summary and evaluation of an article from one of this year’s main computer architecture conferences

- The final exam will be partly based on an article about a recent architecture, which we will study in detail in class

- Participate
- Interact
- Do your homework
- Reflect on the learning experience

- If you get behind with watching the videos, **do not skip the class!** Skim the slides and come anyway!