Informatics 3

# Computer Architecture


Dr. Boris Grot and Dr. Vijay Nagarajan

Institute for Computing Systems Architecture, School of Informatics

University of Edinburgh

# General Information

- Instructors:
  - Boris Grot (boris.grot@ed.ac.uk)
  - Vijay Nagarajan (vijay.nagarajan@ed.ac.uk )
- TAs/markers:
  - Vasilis Gavrielatos (s1687259@sms.ed.ac.uk)
  - Antonis Katsarakis (s1671850@sms.ed.ac.uk)
- Lectures slides: online before each lecture
- Book:
  - Hennessy & Patterson. Computer Architecture: A Quantitative Approach. Morgan Kaufmann, 5th ed
  - 4th ed. also OK
- Lecture slides + book = lecture notes

# General Information (con'd)

- Tutorials
  - Run in weeks 3, 4, 5, 7, 8
    - Week 6 (Feb 19-23) is Flexible Learning week – no class
  - Formative feedback

- Assignments (25%)
  - Assignment 1: Out in early Feb
  - Assignment 2: Out in late Feb (or early Mar)

Please interrupt with questions at any time

# General Information (contd.)

- Course web site: http://www.inf.ed.ac.uk/teaching/courses/car
  - Lecture slides
  - Tutorial problems
  - Courseworks
- Piazza discussion forum: http://piazza.com/ed.ac.uk/spring2018/car
  - Only course related please!
- Tutorials start in week 3

# What is Computer Architecture?

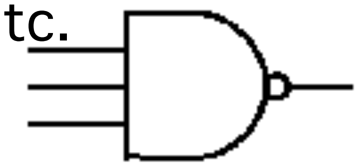
- Architecture
  - Science and art of interconnecting building materials to construct various buildings, subject to constraints
  - Materials: brick, concrete, glass etc.
  - Buildings: house, office, auditorium etc.
  - Constraints: cost, safety, time etc.

# What is Computer Architecture?

- Architecture
  - Science and art of interconnecting building materials to construct various buildings, subject to constraints
  - Materials: brick, concrete, glass etc.
  - Buildings: house, office, auditorium etc.
  - Constraints: cost, safety, time etc.

# What is Computer Architecture?

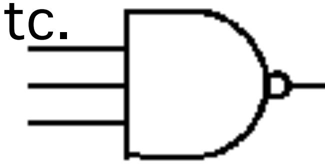
- Architecture
  - Science and art of interconnecting building materials to construct various buildings, subject to constraints
  - Materials: brick, concrete, glass, etc.
  - Buildings: house, office, auditorium, etc.
  - Constraints: cost, safety, time, etc.

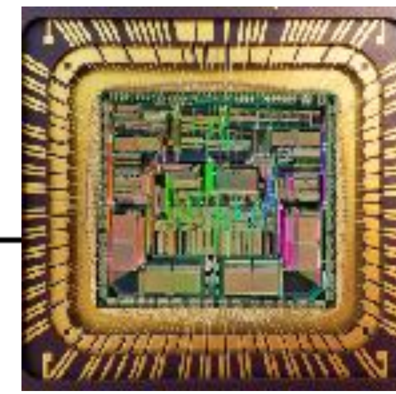


- Computer Architecture
  - Science and art of interconnecting hardware components to create computers, subject to constraints
  - Hardware components: circuits, gates, chips, etc.
  - Computers: desktop, server, mobile phone, etc.
  - Constraints: performance, energy, cost, etc.

# What is Computer Architecture?

- Architecture
  - Science and art of interconnecting building materials to construct various buildings, subject to constraints
  - Materials: brick, concrete, glass, etc.
  - Buildings: house, office, auditorium, etc.
  - Constraints: cost, safety, time, etc.





- Computer Architecture
  - Science and art of interconnecting hardware components to create computers, subject to constraints
  - Hardware components: circuits, gates, chips, etc.
  - Computers: desktop, server, mobile phone, etc.
  - Constraints: performance, energy, cost, etc.

# Three Aspects of Computer Architecture

- Instruction Set Architecture (ISA)
  - Programmer/compiler view
  - Instructions visible to the (system) programmer
  - Opcode, architectural registers, address translation, etc.
- Microarchitecture
  - Processor designer view
  - Logical organization that implements the ISA
  - Pipelining, functional units, caches, registers, etc.
- Circuits
  - Circuit/chip designer view
  - Detailed logic design and packaging technology
  - Gates, cells, CMOS process, etc.

# Three Aspects of Computer Architecture

- Instruction Set Architecture (ISA)
  - Programmer/compiler view
  - Instructions visible to the (system) programmer
  - Opcode, architectural registers, address translation, etc.
- Microarchitecture
  - Processor designer view
  - Logical organization that implements the ISA
  - Pipelining, functional units, caches, registers, etc.
- Circuits
  - Circuit/chip designer view
  - Detailed logic design and packaging technology
  - Gates, cells, CMOS process, etc.

Course focus

# What do we cover in this course?

- Introduction
  - What is Computer Architecture?
  - Historical trends and technology evolution
- Measuring and analyzing computer performance
- Instruction Set Architecture
- Processor microrchitecture
  - Pipelining, hazards, branch prediction, dynamic scheduling, superscalars
- Memory systems
  - Cache hierarchies
- Introduction to parallel architectures (time-permitting)

# The Rise of the Electronic Computer

- Today's microprocessors evolved over a period of 50 years
- Key developments were:
  - First transistor in 1947
  - First integrated circuit in 1958
  - First microprocessor in 1972
- Since the first electronic computers were created in the late 1940's, performance has increased at a dramatic rate, due to:
  - Advances in integrated circuit technology
  - Computer Architecture
  - Compiling Techniques
  - Algorithm Development

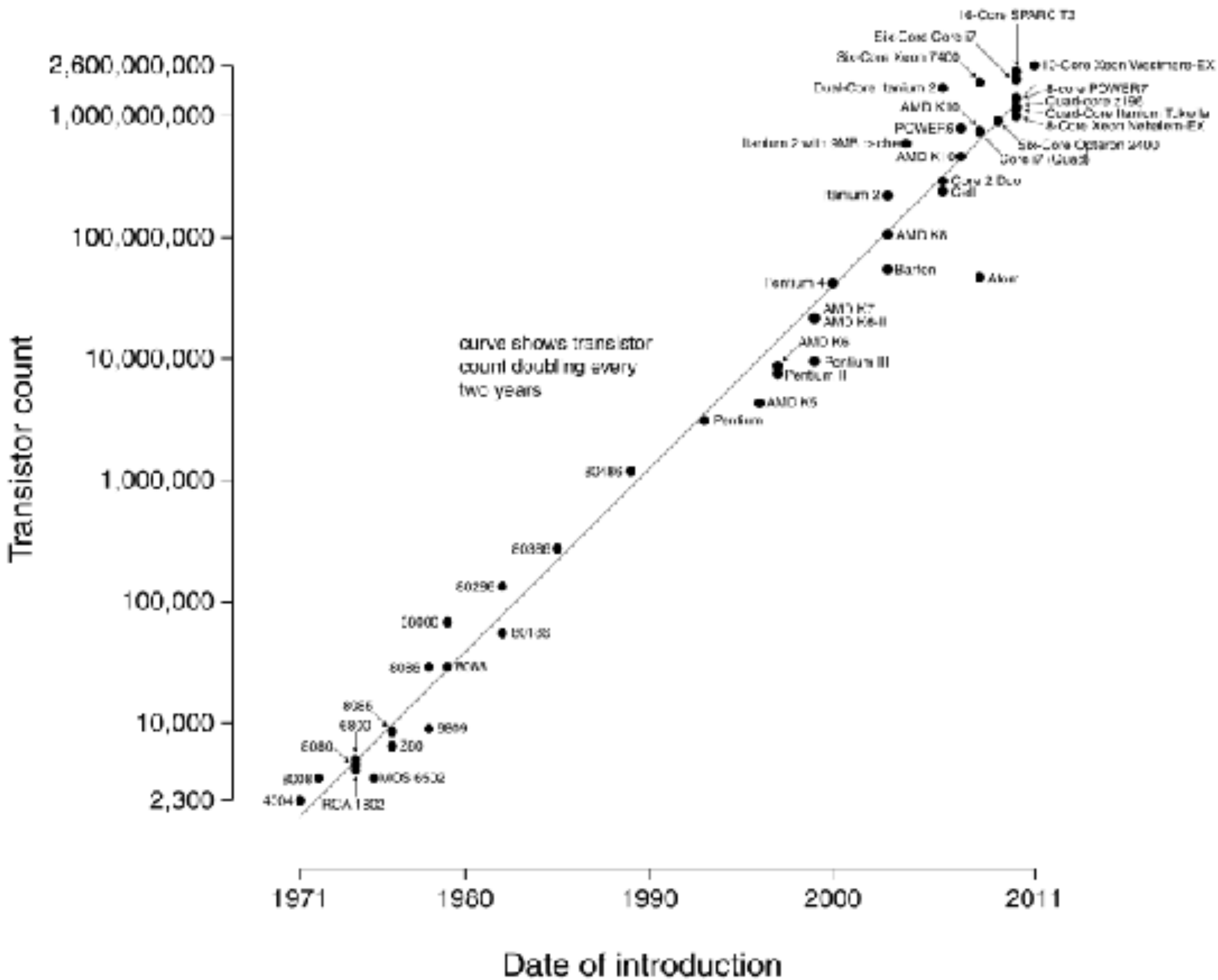
**12-core IBM Power 8**

**William Shockley, and the first transistor (1947)**

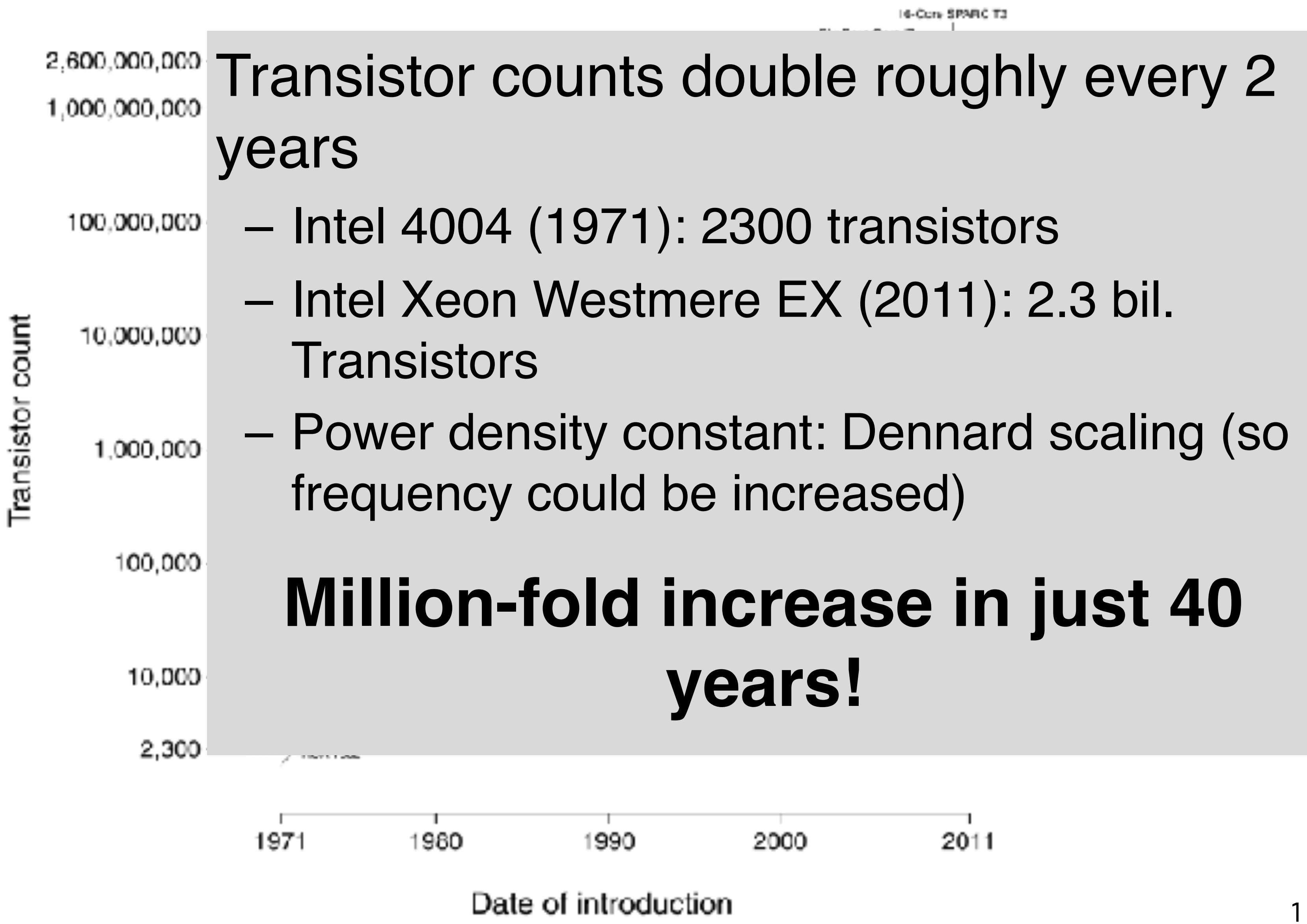
**Jack Kilbey with the first IC (1958)**

**Intel 4004 (1972)**

# Moore’s Law: the engine of computer technology



# Moore's Law: the engine of computer technology

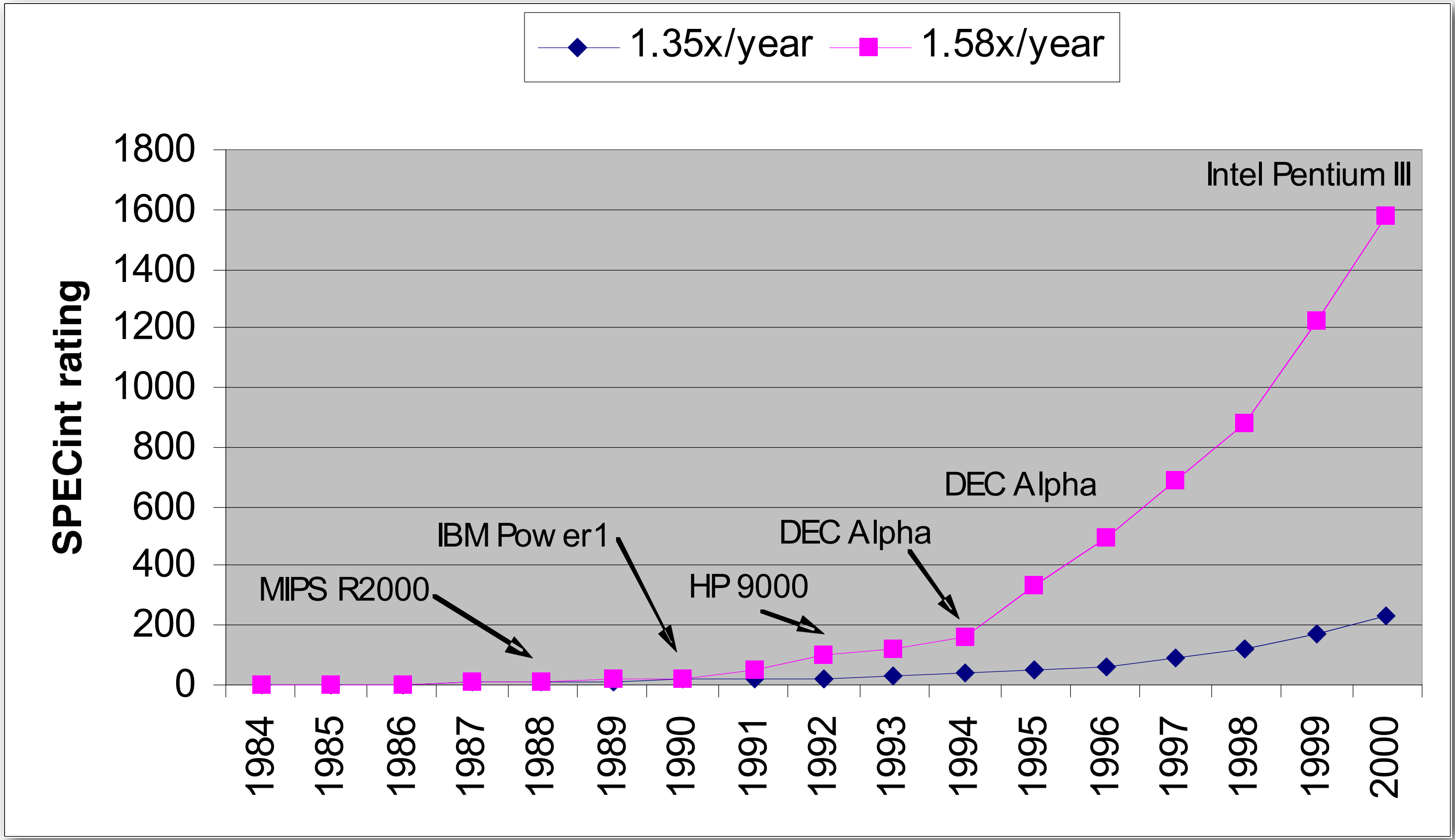



Transistor counts double roughly every 2 years

– Intel 4004 (1971): 2300 transistors
– Intel Xeon Westmere EX (2011): 2.3 bil. Transistors
– Power density constant: Dennard scaling (so frequency could be increased)

**Million-fold increase in just 40 years!**

# Tracking Technology: The role of CA




H&P
Fig. 1.1

Bottom-line: architectural innovation complements technological improvements

# History of Computer Architecture in 1 slide

- Earliest computers – simplest possible implementation
  - Think INF2C-CS
- CDC 6600, 7600 and IBM S/360 model 91 (late 60's)
  - Pipelining
  - Parallel execution units
  - Hardware detects parallelism between instructions
- "Scientific" computing
  - Driven mostly by US military requirements
  - Development of vector processing (70's)
- Emergence of Microprocessor (70's)
  - Driven by commercial and desktop use
  - Not initially a threat to mainframes
  - Eventually rendered mainframes obsolete
- Microprocessor (r)evolution (70's – )
  - Started simple, just like early computers
  - Re-used concepts from 6600, 7600 and 360/91
  - But also integrated fast memory close to CPU
- What drives CA in 2018?

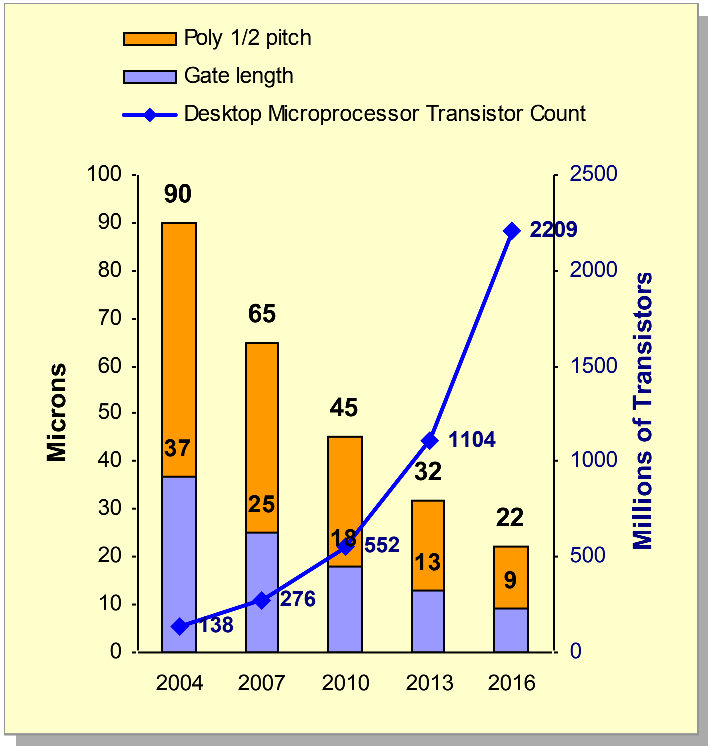

**CDC 6600**


**IBM S/360 model 91**


**Cray 1**

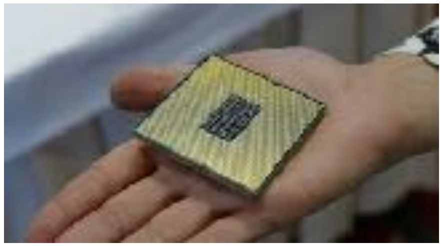
# Future Technology Predictions (2003)

- Moore's Law will continue to ~2016
- Procs. will have 2.2 billion transistors
- DRAM capacity to reach 128 Gbit
- Procs. clocks should reach 40 GHz?




**Source: International Technology Roadmap for Semiconductors, 2003**

# State-of-the-art - January 2018

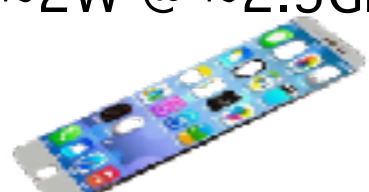


Qualcomm Centriq 2400

- 48cores
- 18B transistors, 398mm2
- ~120W @ 2.2GHz


Apple A11

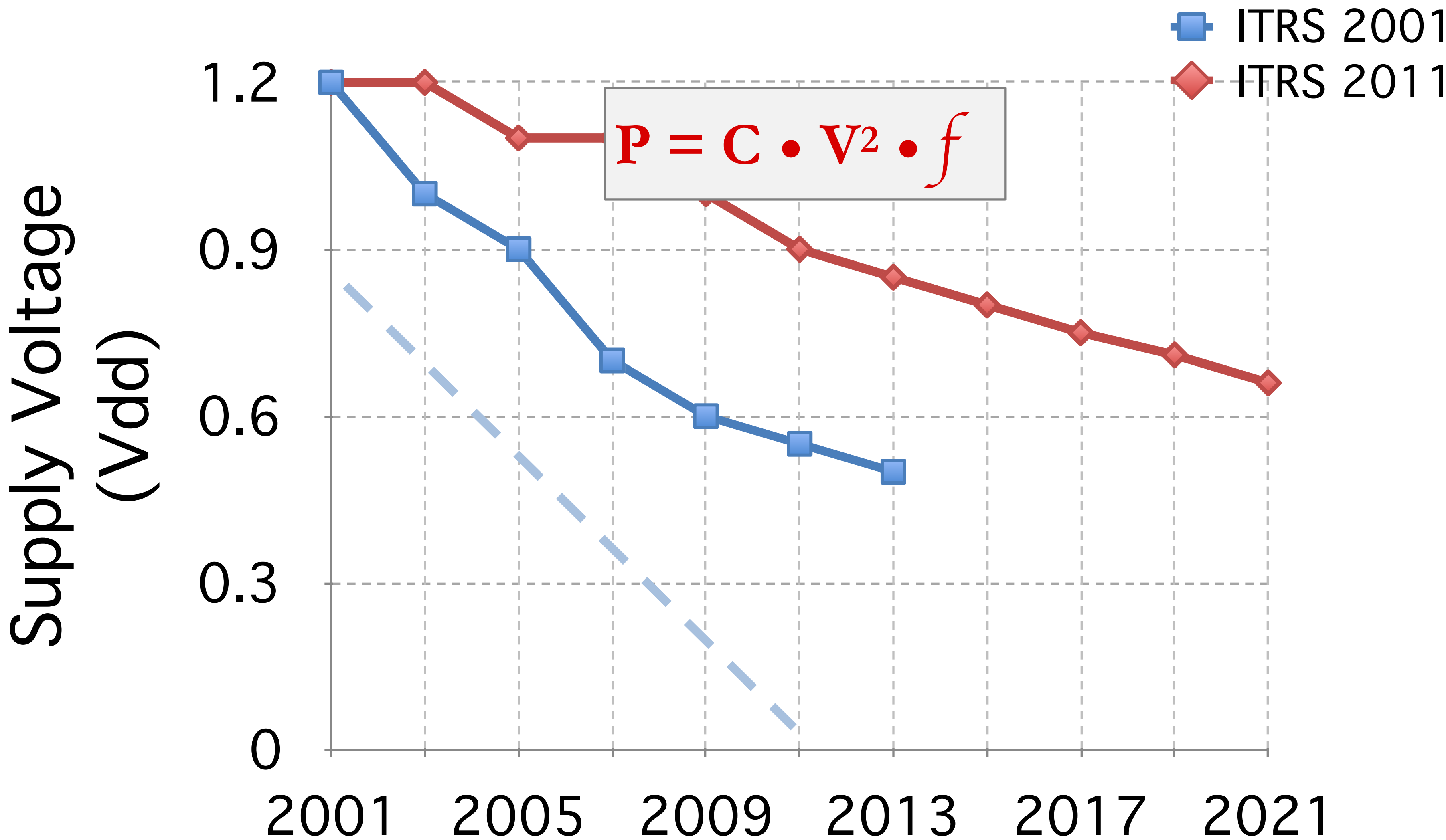
- 6 CPU cores + 3 GPU cores + “Neural Engine”
- 4.3B transistors, $89mm^2$
- ~2W @ ~2.3GHz




ARM Cortex-M3

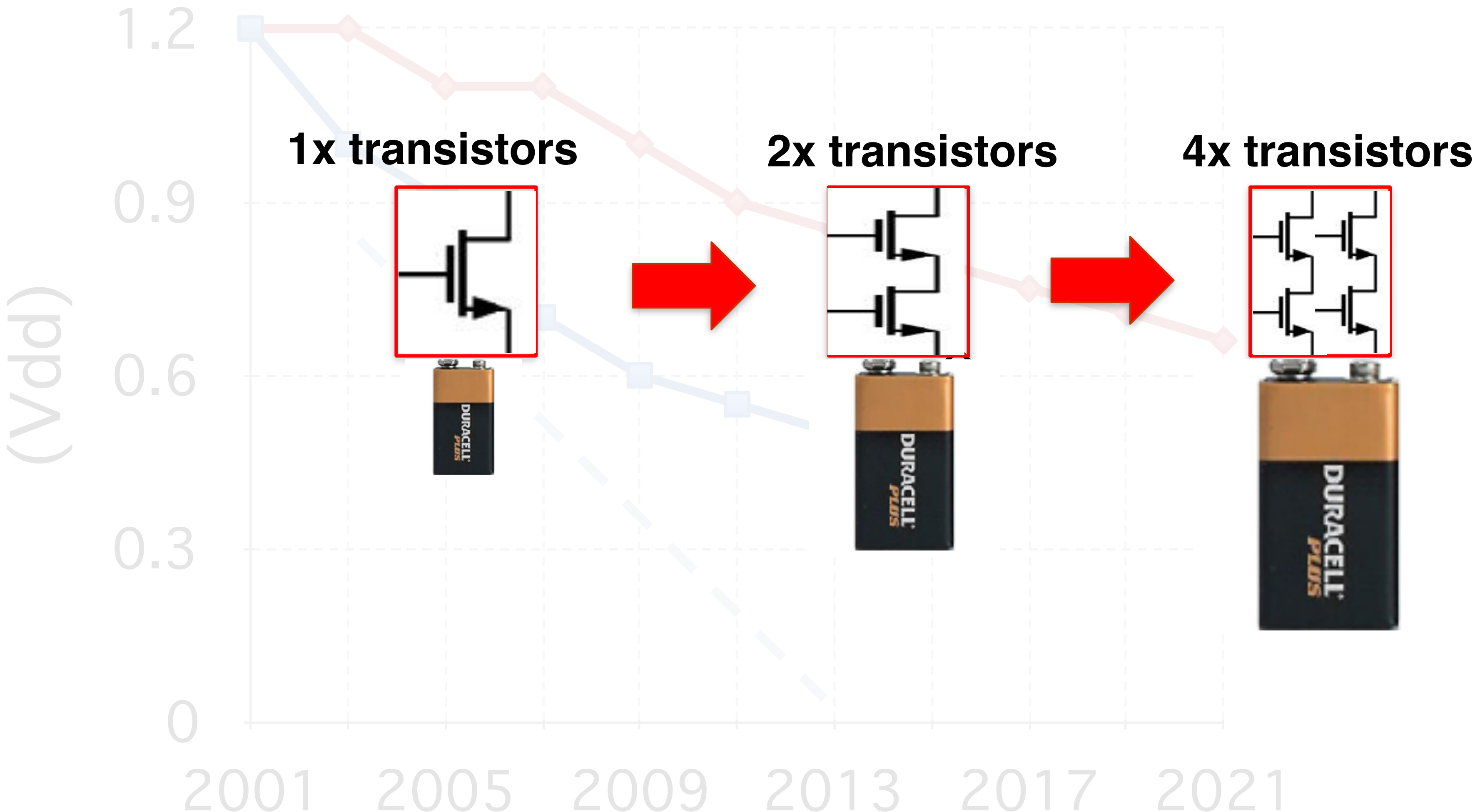
- 3-stage pipeline
- <1mm2
- 4mW @ 100MHz

# Power wall or the end of “free” energy




$$P = C \bullet V^2 \bullet f$$

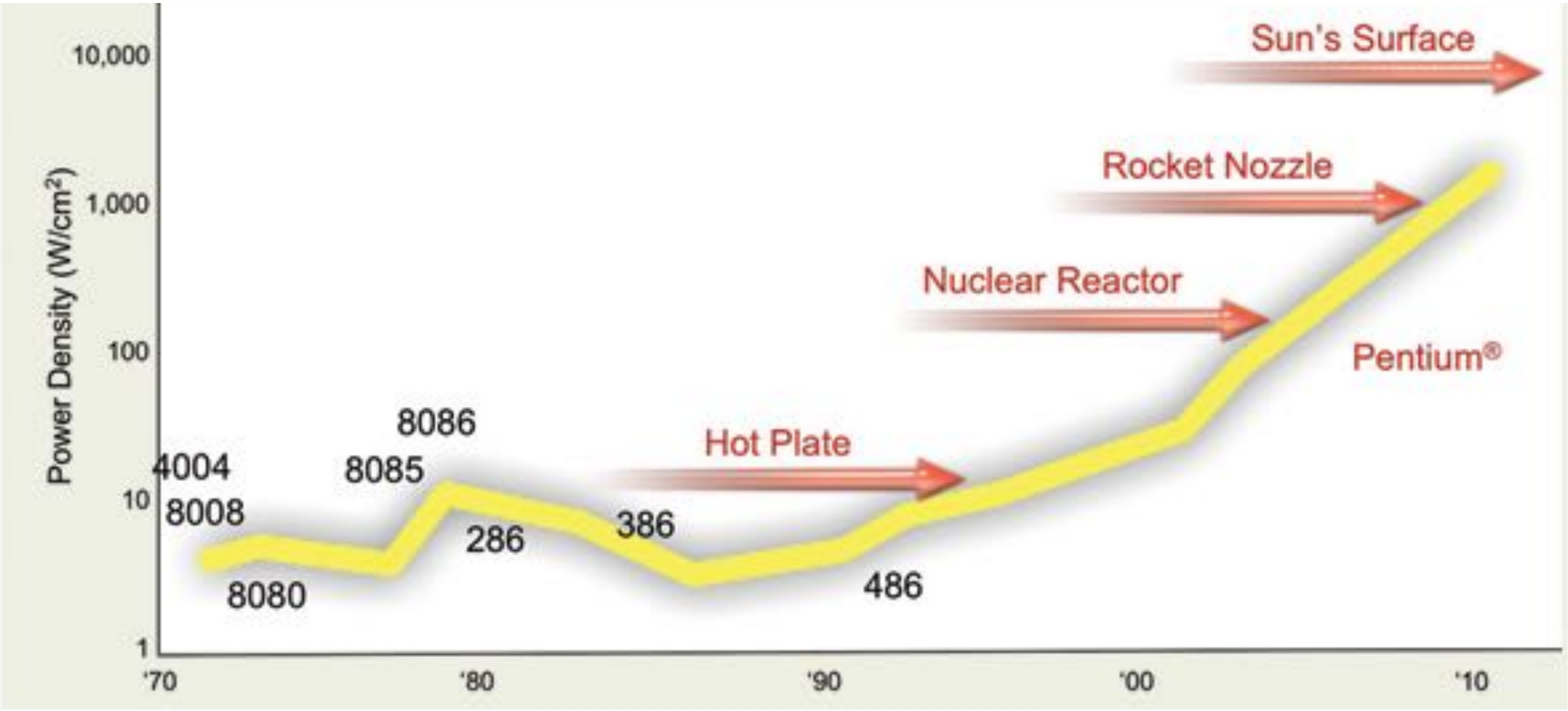
# Power wall or the end of "free" energy



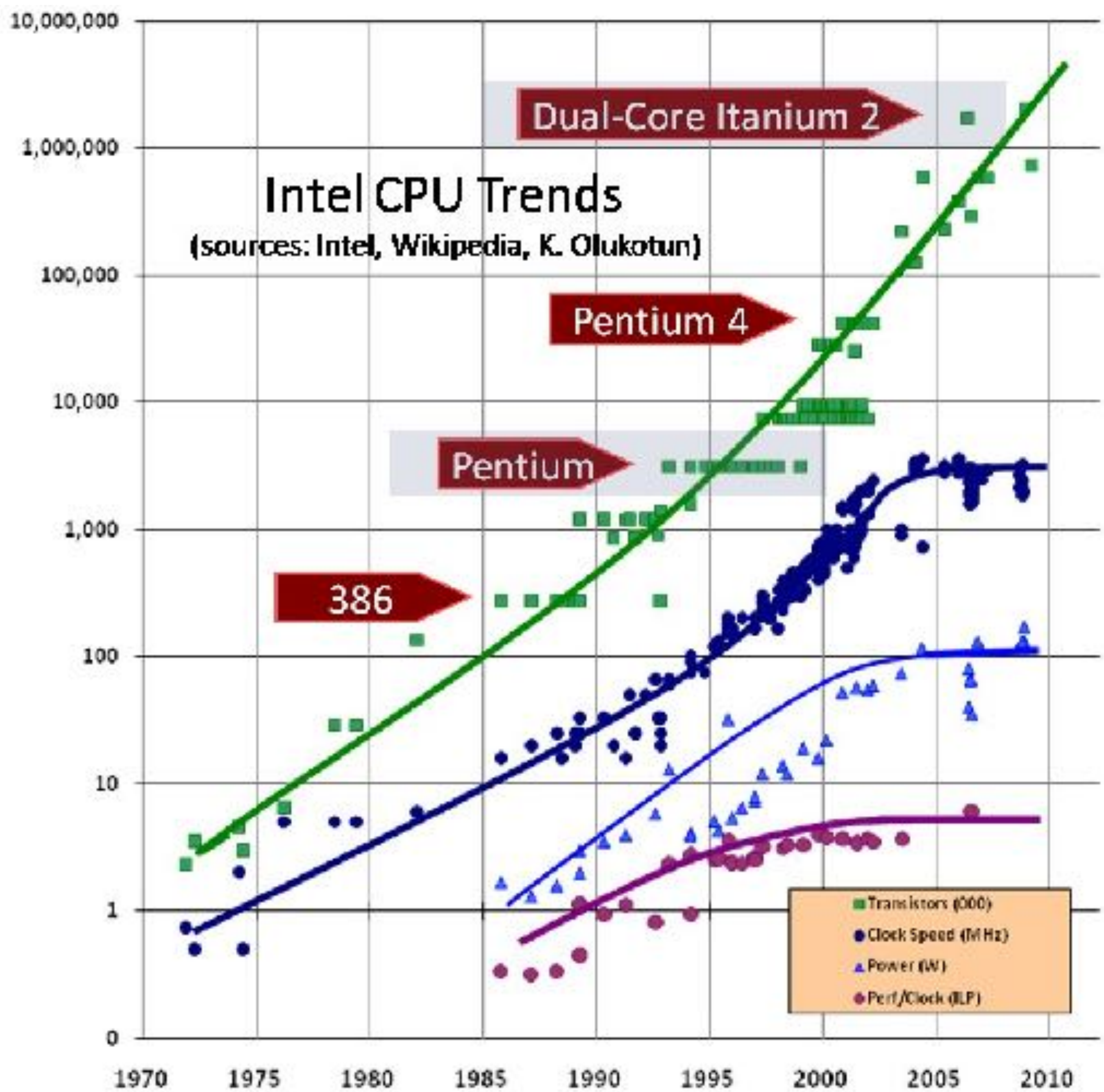
# Heat Dissipation Challenge

Extreme density of transistors and high switching frequency leads to high power density

- This manifests itself as extreme concentration of heat
- Conventional cooling solutions may be insufficient




Pat Gelsinger, Intel Developer Forum, 2004


Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)
Dual-Core Itanium 2
Pentium 4
Pentium
386
Transistors (000)
Clock Speed (MHz)
Power (W)
Perf/Clock (ILP)
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
1970
1975
1980
1985
1990
1995
2000
2005
2010

# Today: the era of multi-cores

- Frequency of processors have stopped scaling since ~2004

- Why?
  - Power wall: Breakdown of Dennard scaling.

- Parallel architectures have arrived
  - Performance through parallelism
  - Software must expose parallelism
  - CA should enable this

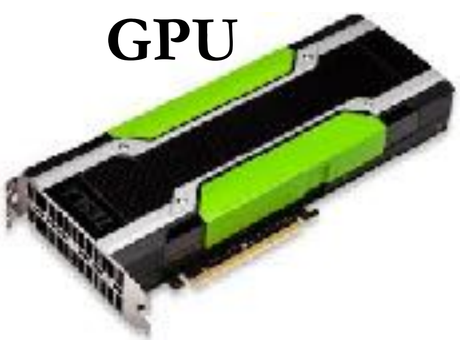
# Today: The age of Specialization

- Moore's law slowing down and predicted to end around 2021!

- Specialized domain-specific processors

**Vision Proc.**

**GPU**

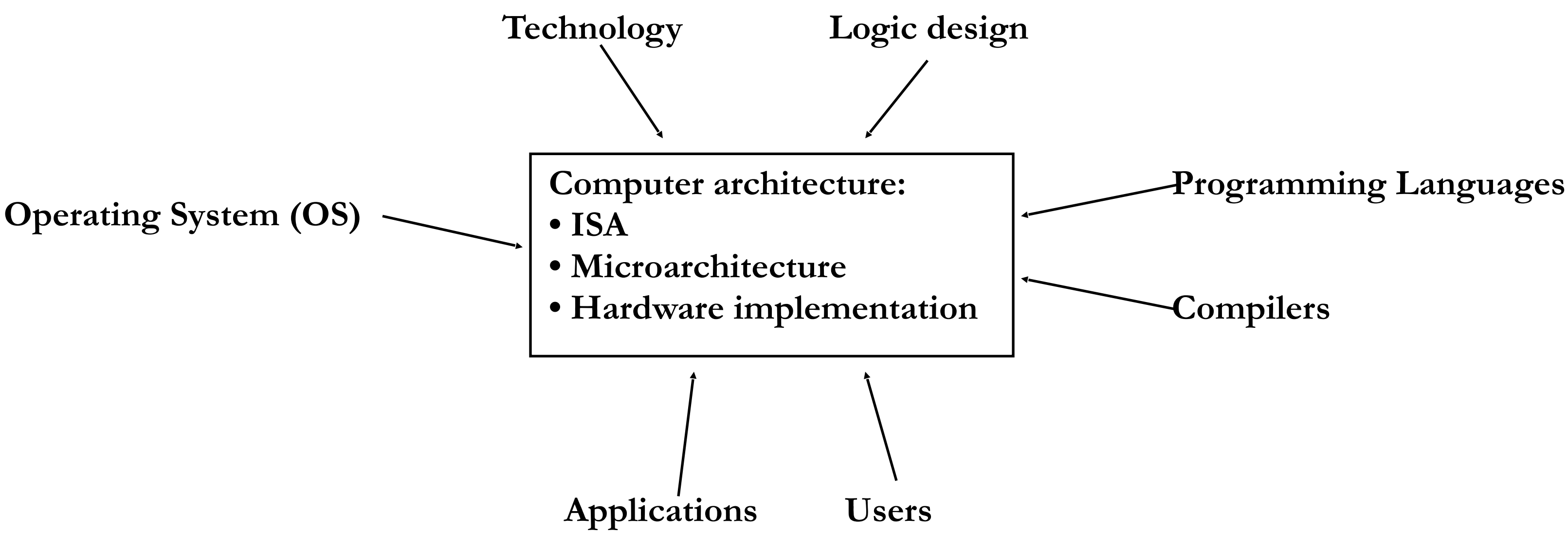


# Current trends in Computer Architecture

- Very complex processor designs
- Parallelism at the chip level
- Power-conscious designs
- Specialisation: domain-specific processors
- Open-source hardware: RISC-V
- Security: (E.g. Read about Meltdown/Spectre!)

# Summary: Computer Architecture

- What is CA?




- Objectives of CA:
  - High-performance
  - Low cost
  - Reliability
  - Low power consumption

# Why study CA?

- Technology is always changing
  - CA should adapt
  - E.g., 10B vs 10K transistor budget
- Requirements are always changing
  - Speed, power, cost, reliability etc.
  - E.g., emergence of smartphones
- Understand computer performance
  - Essential for development of high-performance and/or energy-efficient software
- Get a (design or research) hardware job
  - ARM, Intel, IBM etc.