

# **CS425**

# **Computer Systems Architecture**

**Fall 2024**

**Introduction**

# Outline

- Logistics
- CPU Evolution
- Course goal (what is Computer Architecture?)

# Course Information

- Elective course in Hardware and Computer Systems (E4)
  - 6 ECTS
  - **Prerequisite:** CS225 Computer Organization
- **Instructor:**
  - Vassilis Papaefstathiou ([papaef@csd.uoc.gr](mailto:papaef@csd.uoc.gr))
- **Teaching Assistants:**
  - Mr. Sotiris Totomis ([sototo@csd.uoc.gr](mailto:sototo@csd.uoc.gr))
- **Lectures:**
  - Monday 14:15 – 16:00 (H.204)
  - Wednesday 14:15 – 16:00 (H.204)
  - Friday 14:15 – 16:00 (H.204) backup slot when needed
- **Website:**
  - <http://www.csd.uoc.gr/~hy425>
- **Mailing List:**
  - [hy425-list@csd.uoc.gr](mailto:hy425-list@csd.uoc.gr) (subscribe with majordomo)

# Grading

- **Homeworks & Programming Assignments: 35%**
  - Mandatory
  - Average Grade > 4.5
- **Midterm Exam: 20% (mandatory)**
- **Final Exam: 45% (grade > 4.5)**

# Course Textbooks

- Hennessy and Patterson, Computer Architecture: A Quantitative Approach, 6th Edition/2020. Available in Greek (Klidarithmos Publishers, translation by D. Gizopoulos). ISBN 978-960-645-095-2.
- William Stallings, Computer Organization and Architecture: Designing for Performance, 11th Edition/2020. Available in Greek (Tziolas Publishers, translation by M. Roumeliotis). ISBN 978-960-418-892-5.

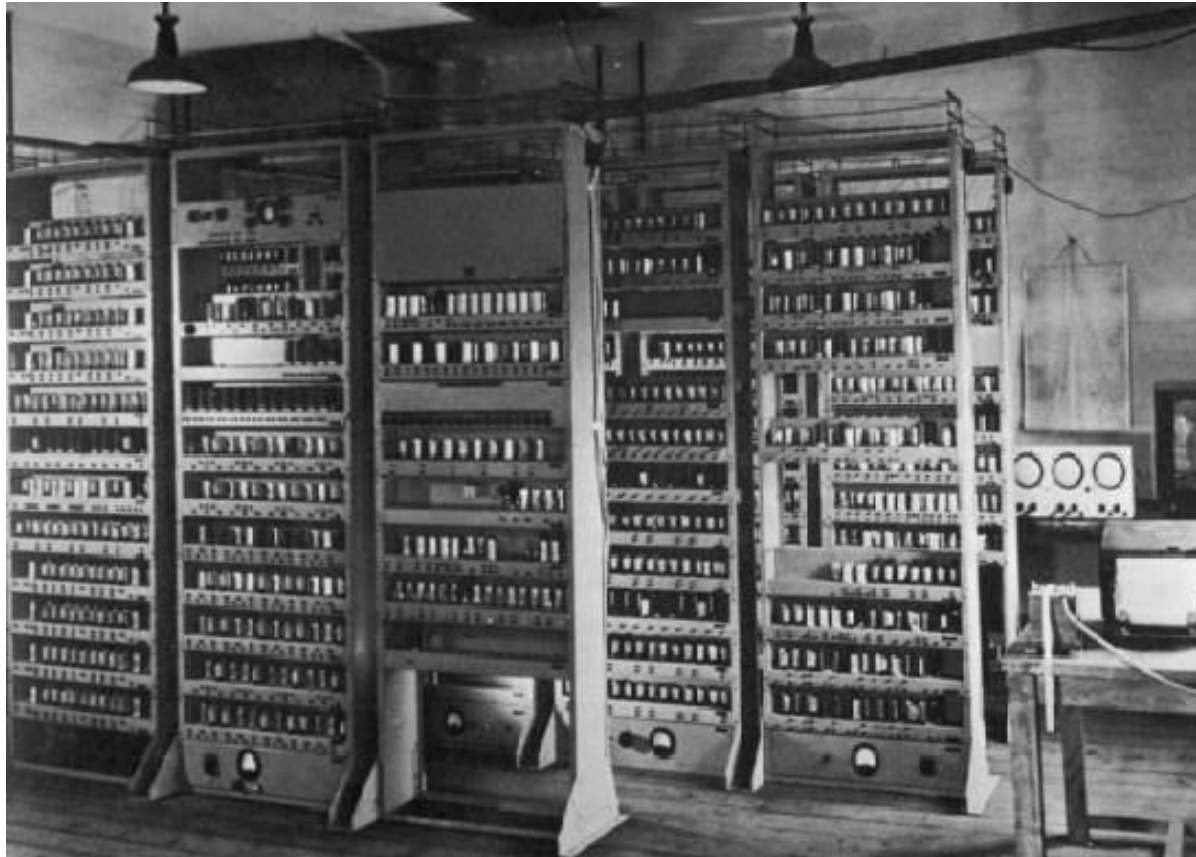


# Tentative Schedule

- Fundamentals, metrics, pipelining (1.5 weeks)
- Instruction Level Parallelism (2 weeks)
- Branch prediction (1 week)
- Multiple issue, VLIW, vector, GPUs, multithreading (3 weeks)
- Memory hierarchy, caches and optimizations (2.5 weeks)
- Multicore processors, cache coherence (2 weeks)
- Main memory technologies (1 week)

# History in Computer Devices

- EDSAC, University of Cambridge, UK, 1949-1958 (mercury-based memory, logic, punched tape, teleprinter, EDSAC2 1965)



# Computing Systems Today

- The world is a large parallel system
  - Microprocessors everywhere
  - Vast infrastructure behind them



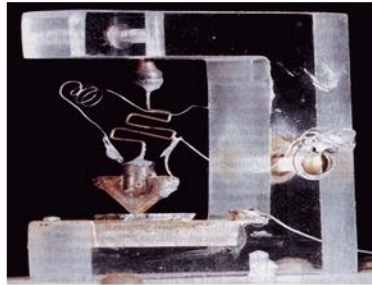
- '70: microproc. & supercomputers
- '80: compilers, OS, RISC, x86
- '90: Internet, WWW, PDA
- '00: mobile, cell phones, embedded cpus
- '10: internet of things (IoT)
- '20: Machine Learning (ML) & AI



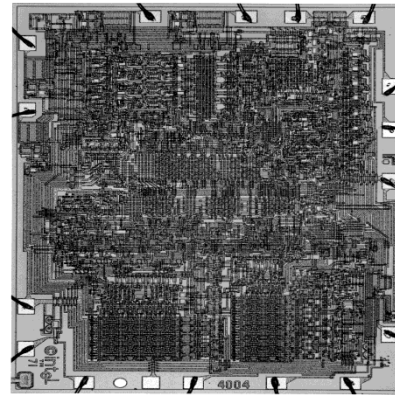
# Improvement in Computer

- Radical progress in computers due to:
  - Technological improvements (next few slides)
    - steady
  - Better computer architectures (course focus)
    - less consistent

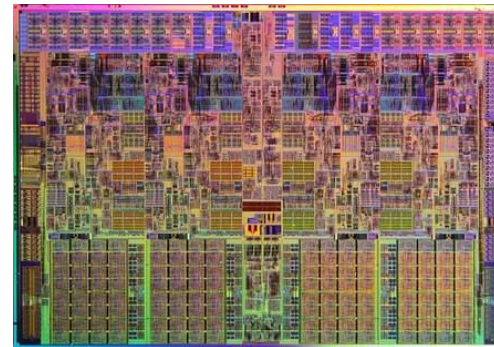
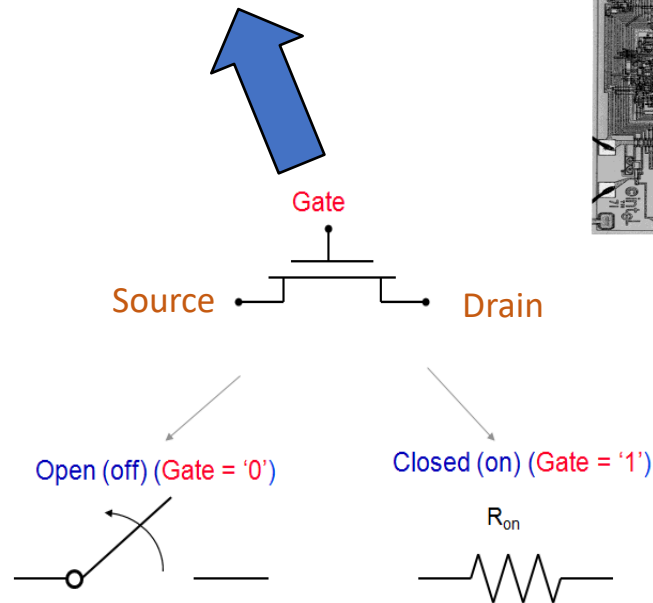
# Technology: Transistor Revolution



Bell Labs, 1948  
First Transistor



Intel 4004, 1971 (Moore, Noyce Intel 1968)  
4-bit  
2,300 transistors  
740KHz operation  
10 $\mu$ m (=10000nm) PMOS technology

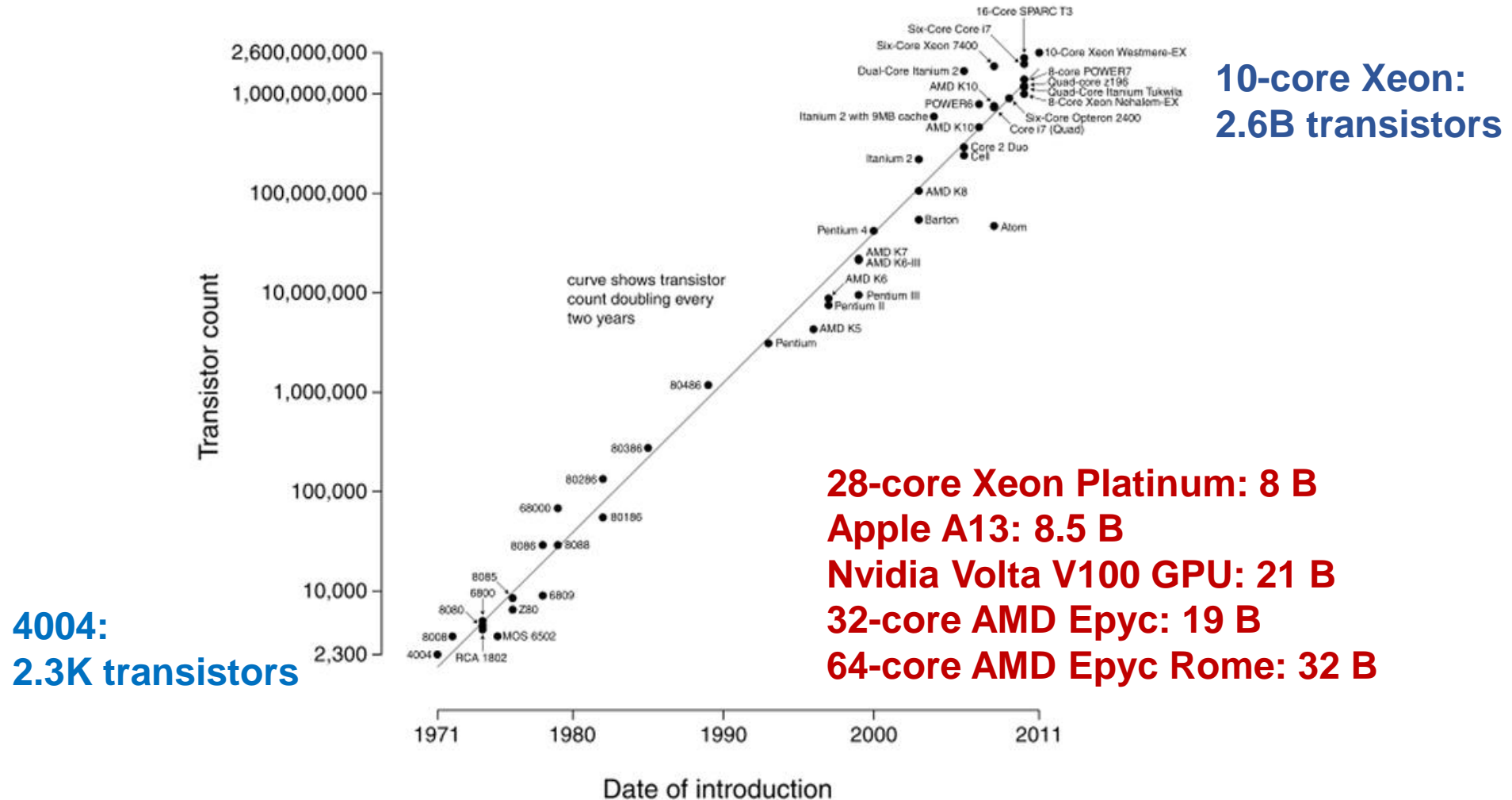


Intel Core i7, 2011  
64-bit  
2,600,000,000 transistors  
3.4GHz  
32nm

# Technology: Moore's Law

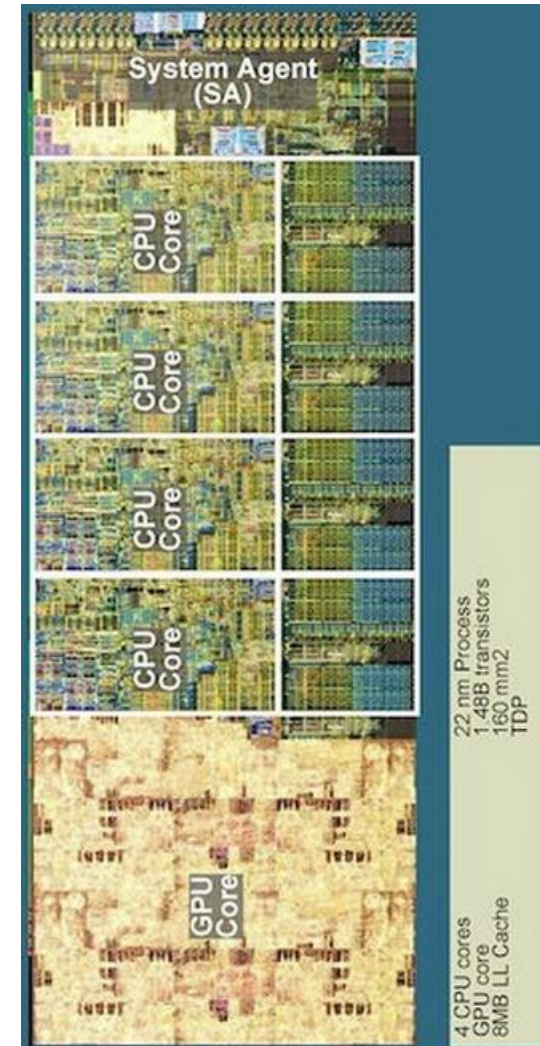
- In 1965, Gordon Moore predicted that the number of transistors that can be integrated on a die would double every 18 months (i.e., grow exponentially with time)
- He made a prediction that semiconductor technology will double its effectiveness every 18 months
- In practice a new technology is introduced every ~two years, with feature sizes of circuit layout 70% of the previous technology

# Technology: Transistor Count



# Technology constantly on the move

- Number of transistors is not the limiting factor
  - Currently ~70+ billion transistors/chip
  - Problems: power, heat, latency
- 3-dimensional chip technology?
  - Sandwiches of silicon (Package on Package)
  - “Through-silicon Vias” TSVs for communication
  - FinFET
- On-chip optical connections?
  - Power savings for large packets
- Intel Core i7 (“Ivy Bridge”)
  - 4 cores + GPU
  - 22 nm, tri-gate (“3D”) transistors
  - 1.4B Transistors
  - Shared L3 Cache - 8MB
  - L2 Cache - 1MB (256K x 4) , L1 – 64KB/core

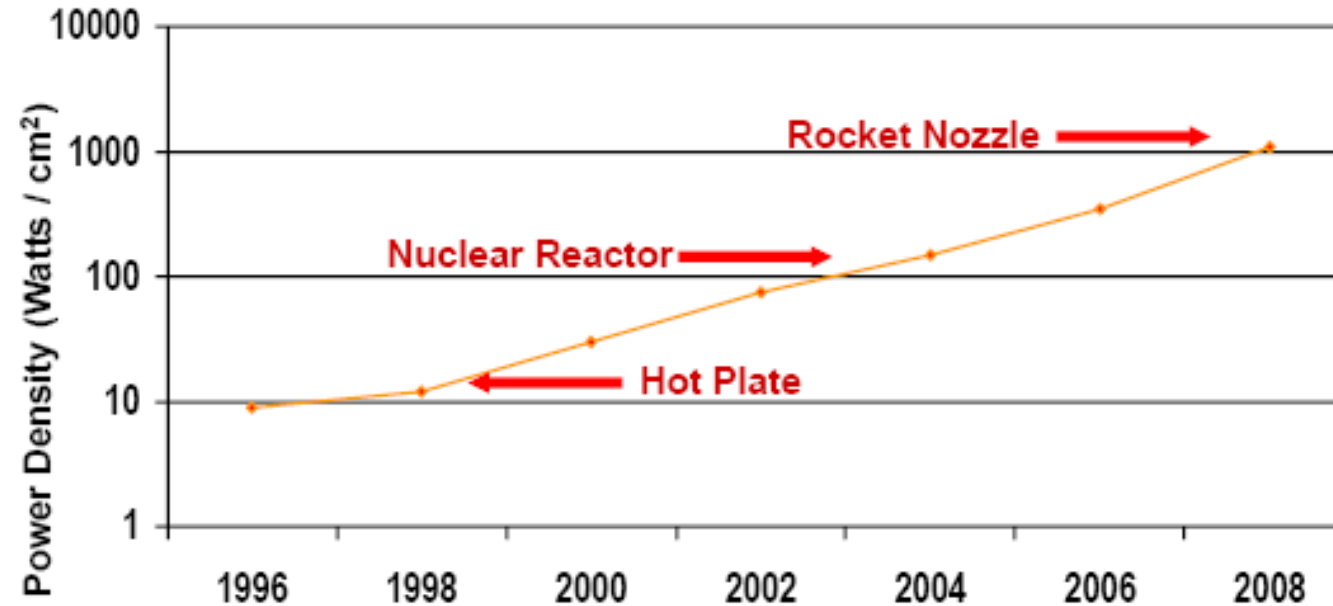


# Transistor size trends and questions

- Feature sizes, higher performance?
  - Transistor size went down from 10 micros to 3 nanometers
  - Quadratic increase in density, linear drop in feature size
  - Linear increase in transistor performance
- Where is the catch?
  - Smaller voltage reduction to maintain safe operation
  - Higher resistance and capacitance per unit of length
  - Shorter wires but with higher resistance/capacitance
  - Wire delays improving poorly compared to transistors

# Limiting Force: Power Density

## Moore's Law Extrapolation: Power Density for Leading Edge Microprocessors

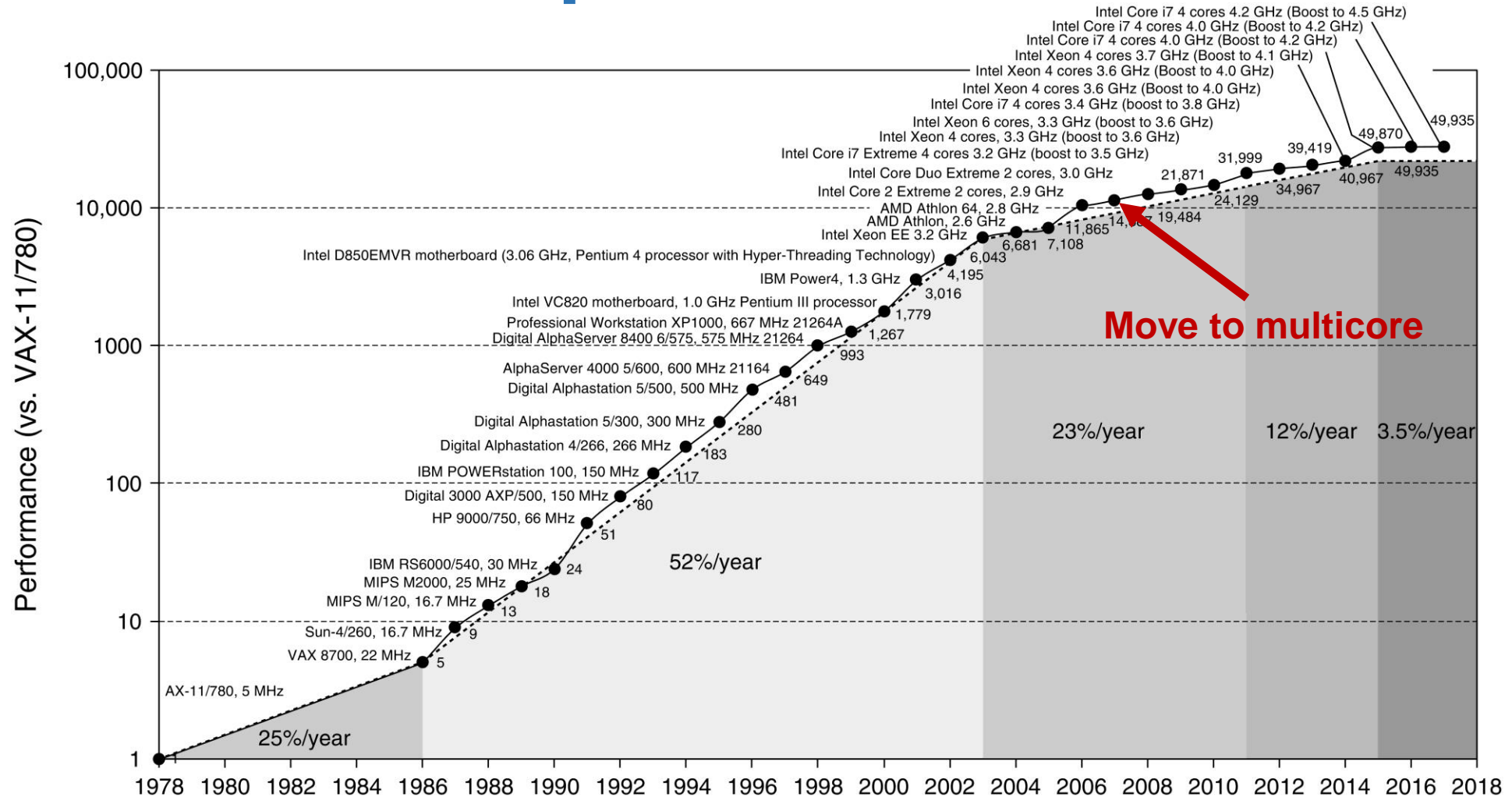


Microprocessor Size  $\approx 2 \text{ cm}^2$

Power Density Becomes Too High to Cool Chips Inexpensively

Source: Shekhar Borkar, Intel Corp

# Crossroads: Uniprocessor Performance

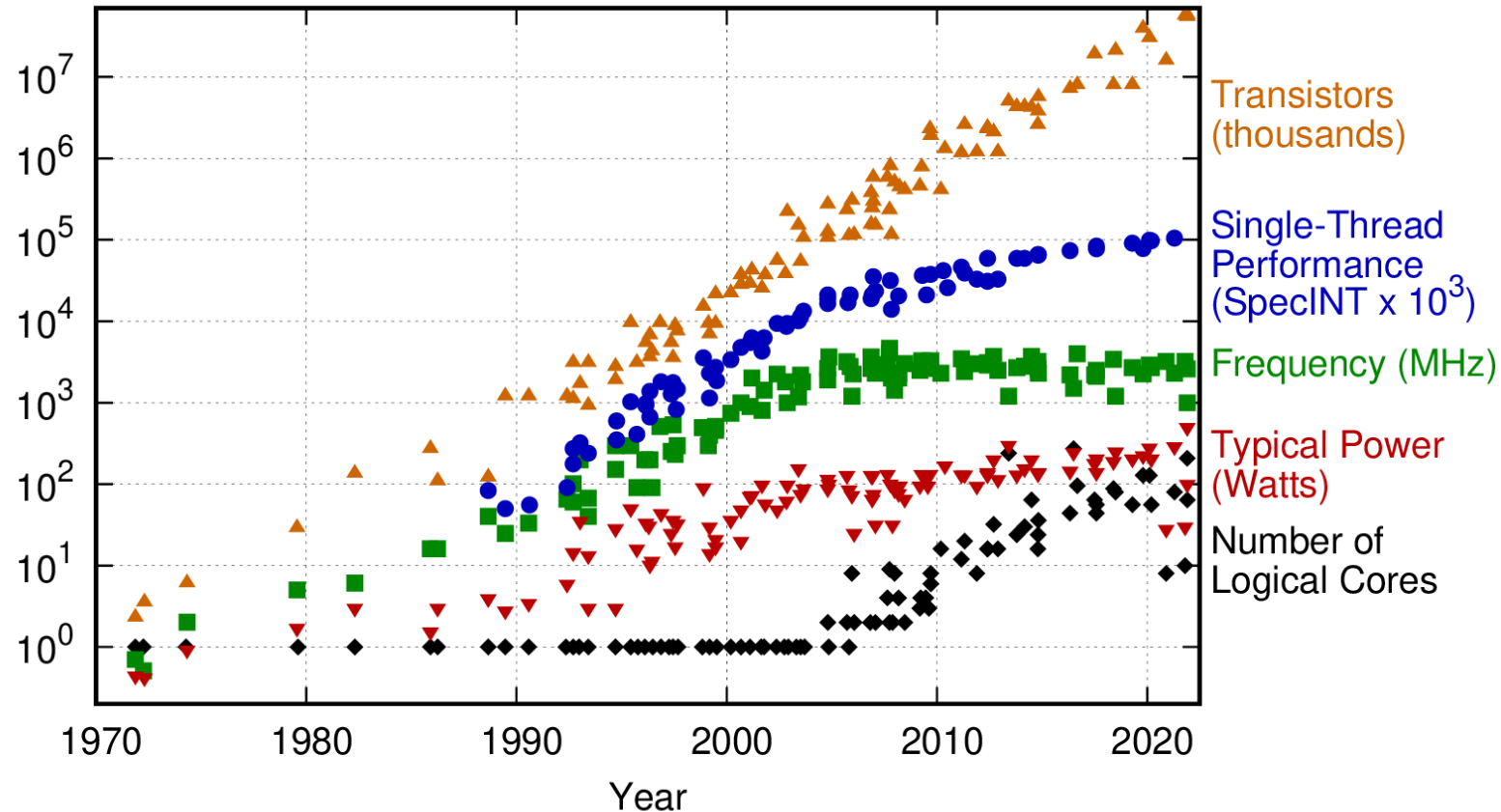


**Constrained by power, instruction level parallelism, memory latency**



# Trends – All in one

50 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2021 by K. Rupp

<https://github.com/karlrupp/microprocessor-trend-data>

# The End of the Uniprocessor Era

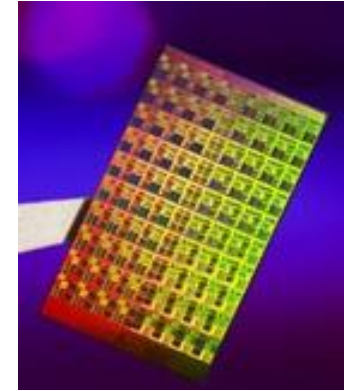
- **Power wall:** power expensive, transistors free
  - can put more on chip than can afford to turn on
- **ILP wall:** law of diminishing returns on more HW for ILP
- **Memory wall:** Memory slow, multiplies fast
  - 200 clock cycles to DRAM memory vs. 4 clocks for multiply
- Power Wall + ILP Wall + Memory Wall = Brick Wall
  - Uniprocessor performance now 2X every 5(?) years

***Single biggest change in the history of computing systems***

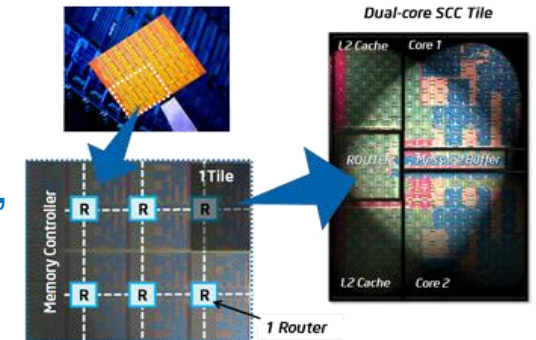
# Many Core Chips: The future is here

- “Many Core” refers to many processors/chip
  - 64 or 128
- How to program these?
  - Use 2 CPUs for video/audio
  - Use 1 for word processor, 1 for browser
  - 76 for virus checking???
- Something new is clearly needed here...

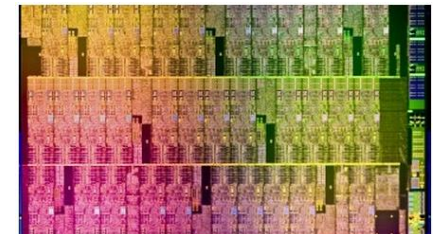
Intel 80-core multicore chip,  
2007, 65nm – 100M transistors



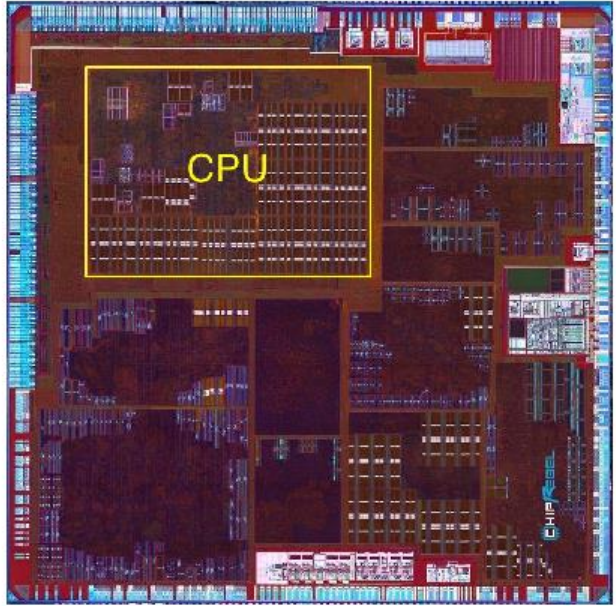
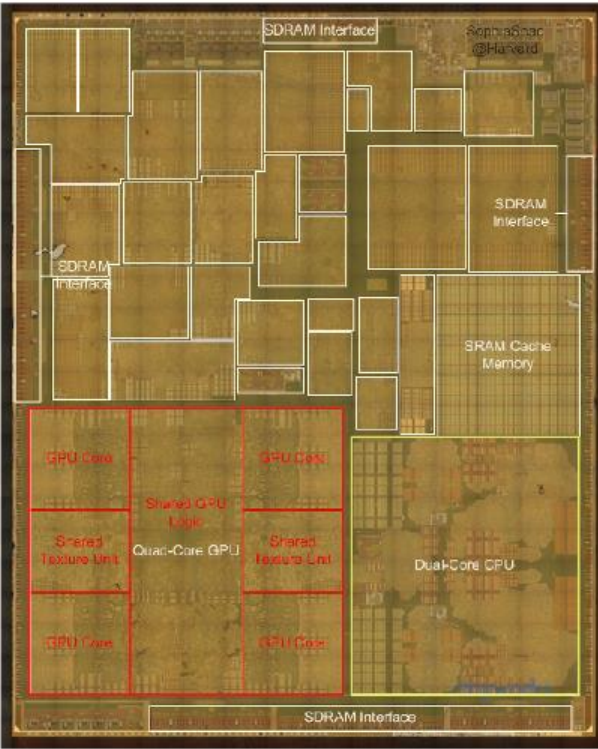
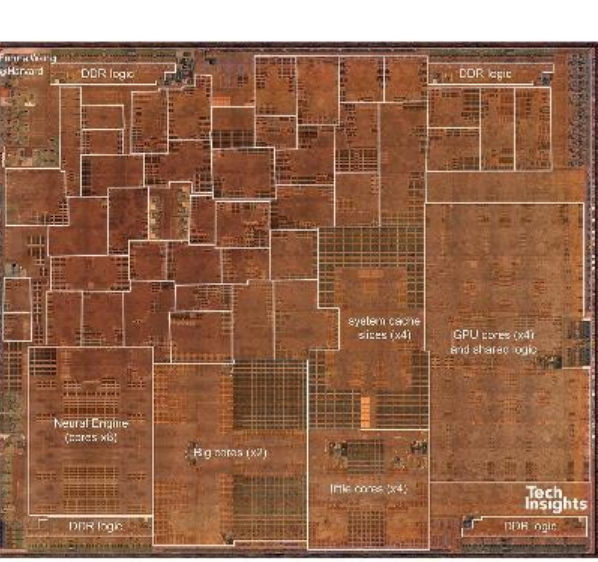
Intel Single-Chip Cloud  
Computer (SCC), 48-cores,  
2010, 4 memory controllers,  
24-router mesh



Intel Many Integrated Core  
Architecture (MIC), 50-cores,  
2012, 22nm, commercial

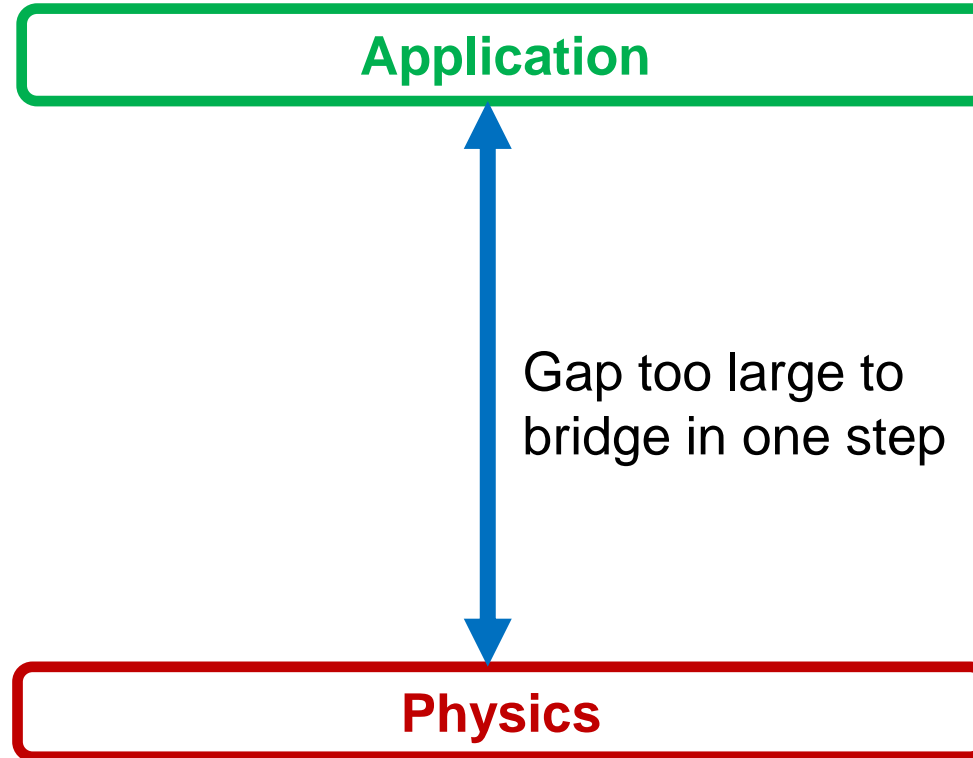


# Accelerator Level Parallelism & Dark Silicon

		
<p><b>2010 Apple A4</b> 65 nm Samsung 53 mm<sup>2</sup> 4 accelerators</p>	<p><b>2014 Apple A8</b> 20 nm TSMC 89 mm<sup>2</sup> 28 accelerators</p>	<p><b>2019 Apple A12</b> 7 nm TSMC 83 mm<sup>2</sup> 42 accelerators</p>

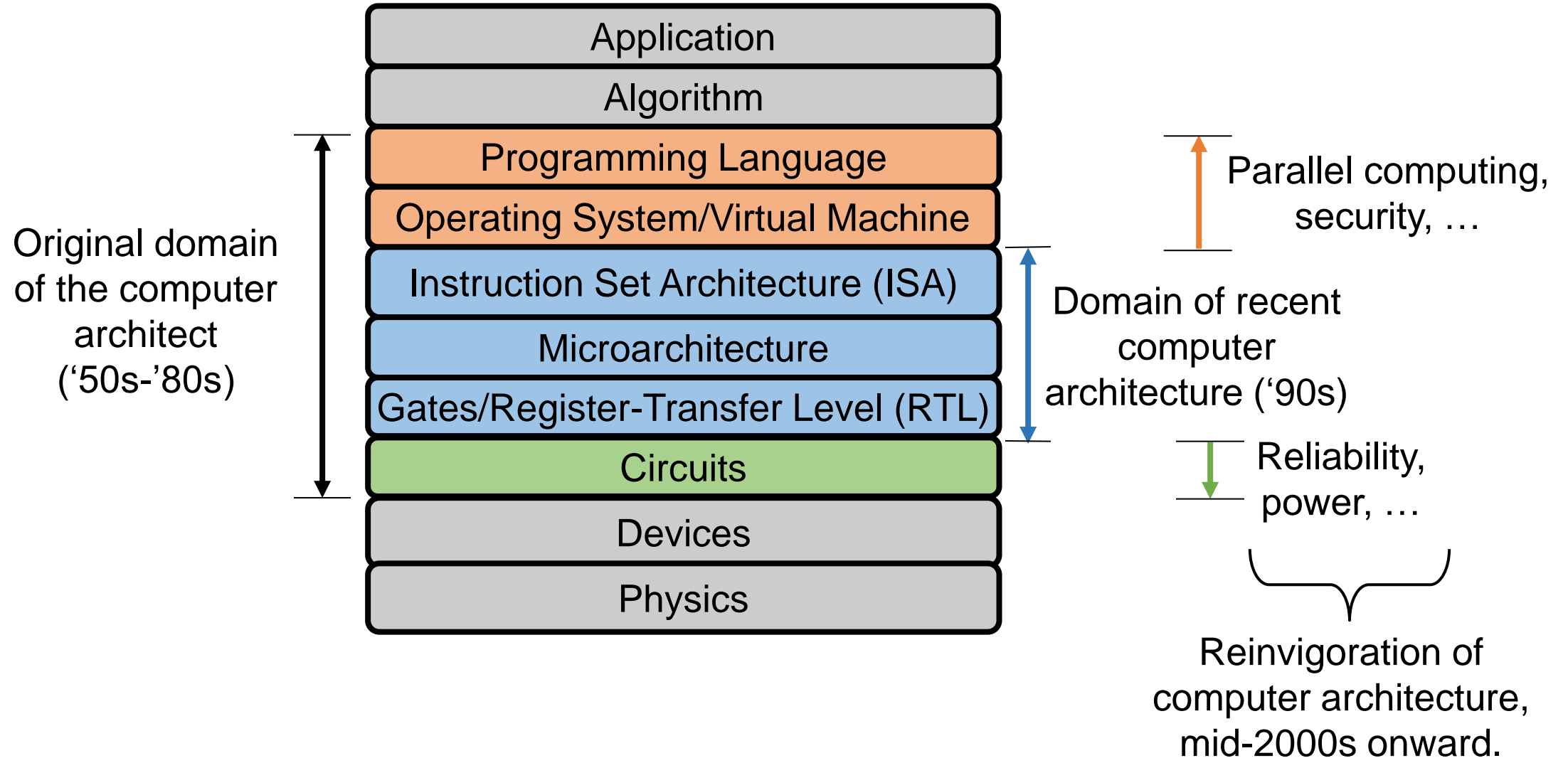
<https://cacm.acm.org/opinion/accelerator-level-parallelism/>

# What is Computer Architecture



- In its broadest definition, computer architecture is the *design of the abstraction layers* that allow us to implement information processing applications efficiently using available manufacturing technologies.

# Abstraction Layers in Modern Systems



# Computer Architecture is an Integrated Approach

- What really matters is the functioning of the complete system
  - hardware, runtime system, compiler, operating system, and application
  - In networking, this is called the “**End to End argument**”
- Computer architecture is not just about transistors, individual instructions, or particular implementations
  - E.g., Original RISC projects replaced complex instructions with a compiler + simple instructions
- It is very important to think across all hardware/software boundaries
  - New technology  $\Rightarrow$  New Capabilities  $\Rightarrow$  New Architectures  $\Rightarrow$  New Tradeoffs
  - **Delicate balance between backward compatibility and efficiency**

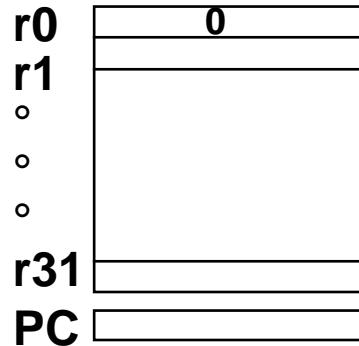
# Defining Computer Architecture (ISA)

## Instruction Set Architecture

- ISAs converged to a common RISC paradigm
  - CISC ISAs implemented on RISC pipelines
- Load-store architectures, general-purpose registers
- Aligned memory addressing, simple addressing modes
- Byte, word, double-word, quad-word operands
- Arithmetic, logic, control operations
- Fixed-length encoding



# Example: MIPS R3000



## Programmable storage

$2^{32}$  x bytes

31 x 32-bit GPRs (R0=0)

32 x 32-bit FP regs (paired DP)

PC

Data types ?

Format ?

Addressing Modes?

## Arithmetic logical

Add, AddU, Sub, SubU, And, Or, Xor, Nor, SLT, SLTU,  
AddI, AddIU, SLTI, SLTIU, AndI, OrI, XorI, LUI  
SLL, SRL, SRA, SLLV, SRLV, SRAV  
MUL, DIV

## Memory Access

LB, LBU, LH, LHU, LW, LWL, LWR  
SB, SH, SW, SWL, SWR

**32-bit instructions on word boundary**

## Control

J, JAL, JR, JALR  
BEq, BNE, BLEZ, BGTZ, BLTZ, BGEZ, BLTZAL, BGEZAL

# ISA vs Computer Architecture

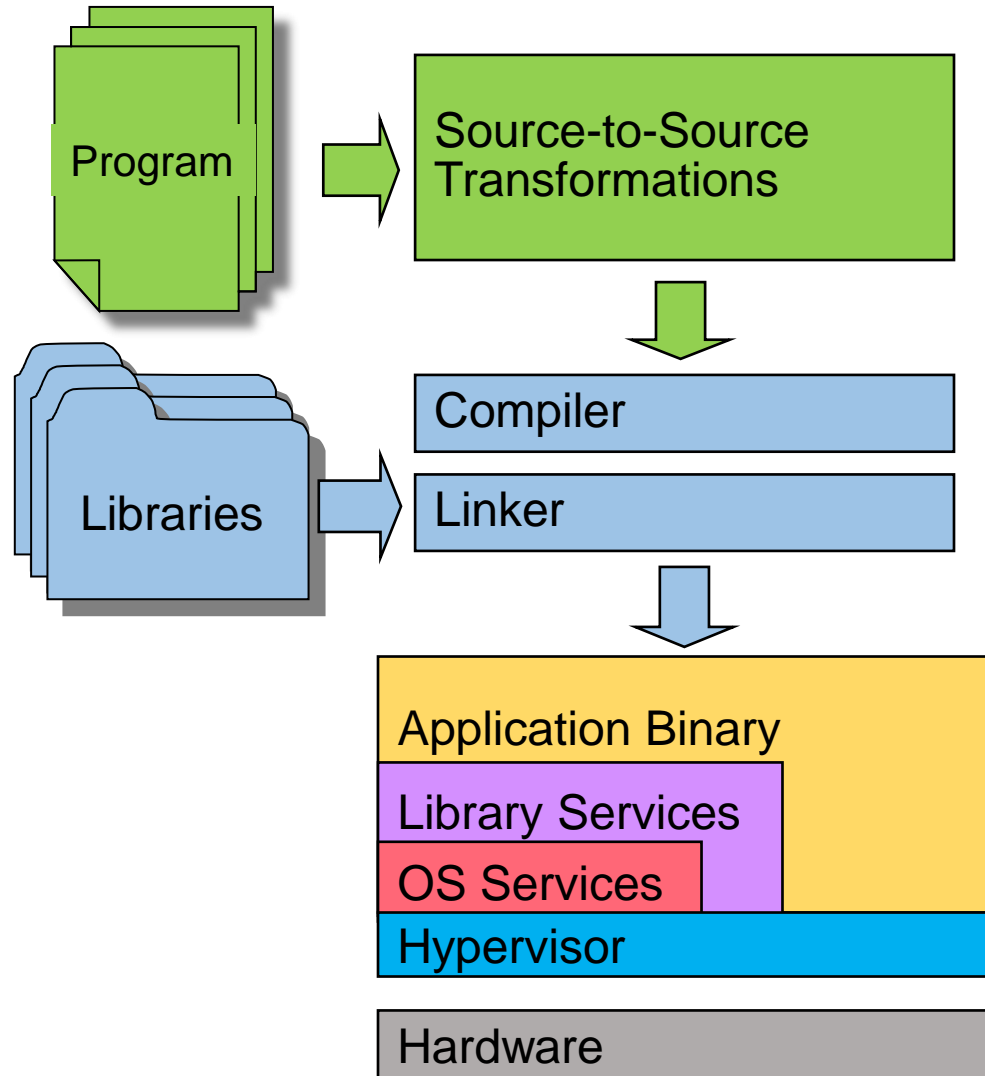
- Old definition of computer architecture == instruction set design
  - Other aspects of computer design called implementation
  - Suggests that implementation is uninteresting or less challenging
- Computer architecture >> ISA
- Architect's job much more than instruction set design; technical hurdles today **more** challenging than those in instruction set design

# Defining Computer Architecture

Architecture = ISA (+prog. lang.) + Organization + Hardware

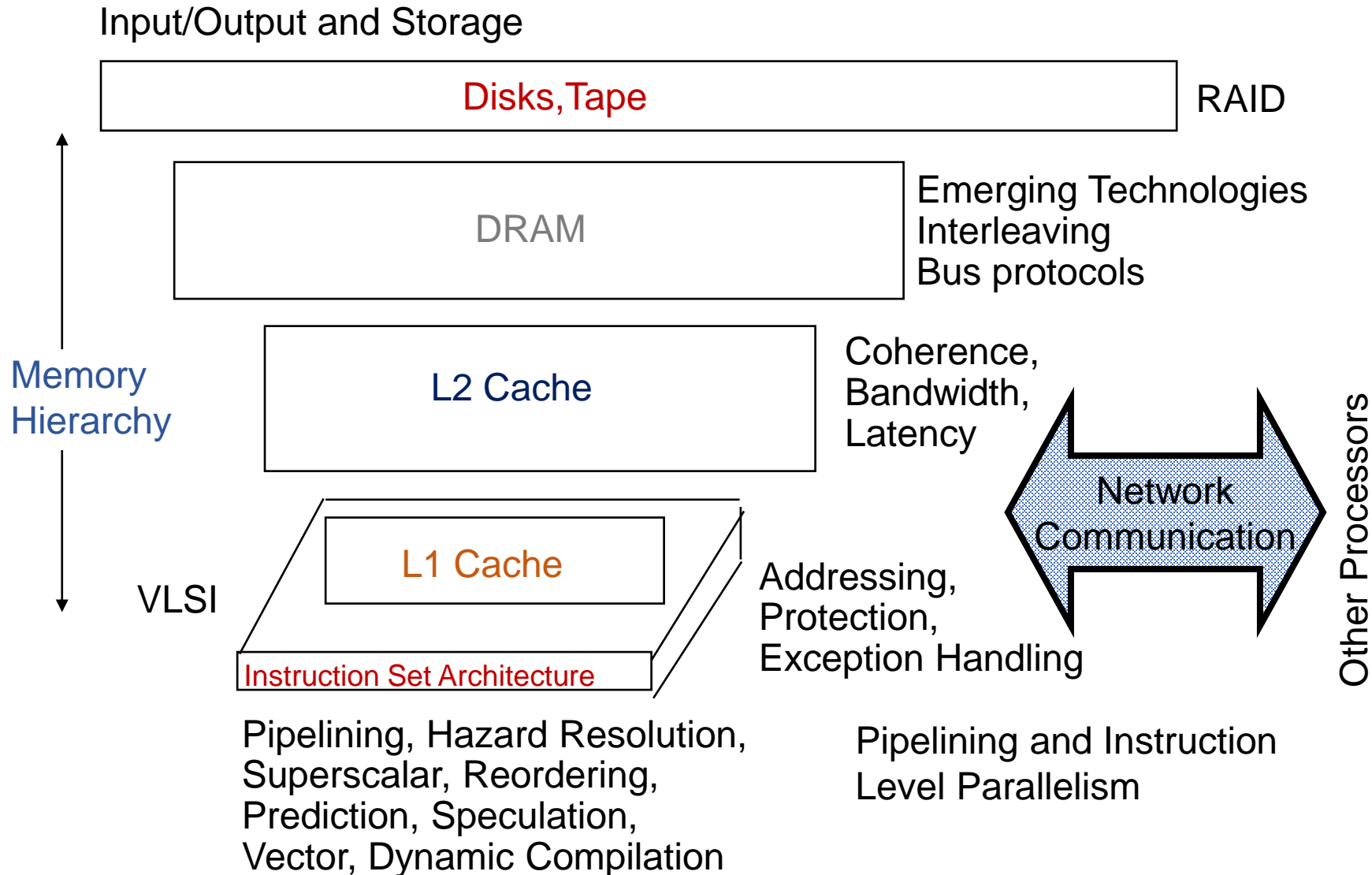
- Processor Architecture
  - Pipelining, hazards, ILP, HW/SW interface
- Memory hierarchies
- Interconnects
- I/O systems
- Hardware technology used (e.g. component size)
- Computer architecture focuses on **organization** and **quantitative principles** of design

# Execution is not just about HW and ISA



- The VAX fallacy
  - Produce one instruction for every high-level concept
  - Absurdity: Polynomial Multiply
    - Single hardware instruction
    - But Why? Is this really faster???
- RISC Philosophy
  - Full System Design
  - Hardware mechanisms viewed in *context* of complete system
  - Cross-boundary optimization
- Modern programmer does not see assembly language
  - Many do not even see “low-level” languages like “C”.

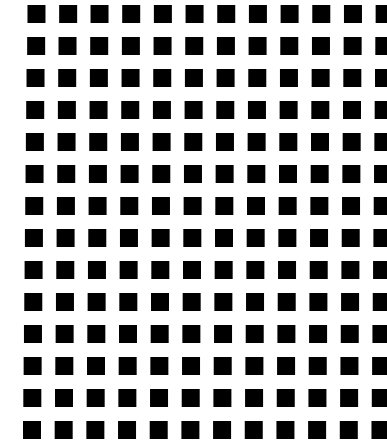
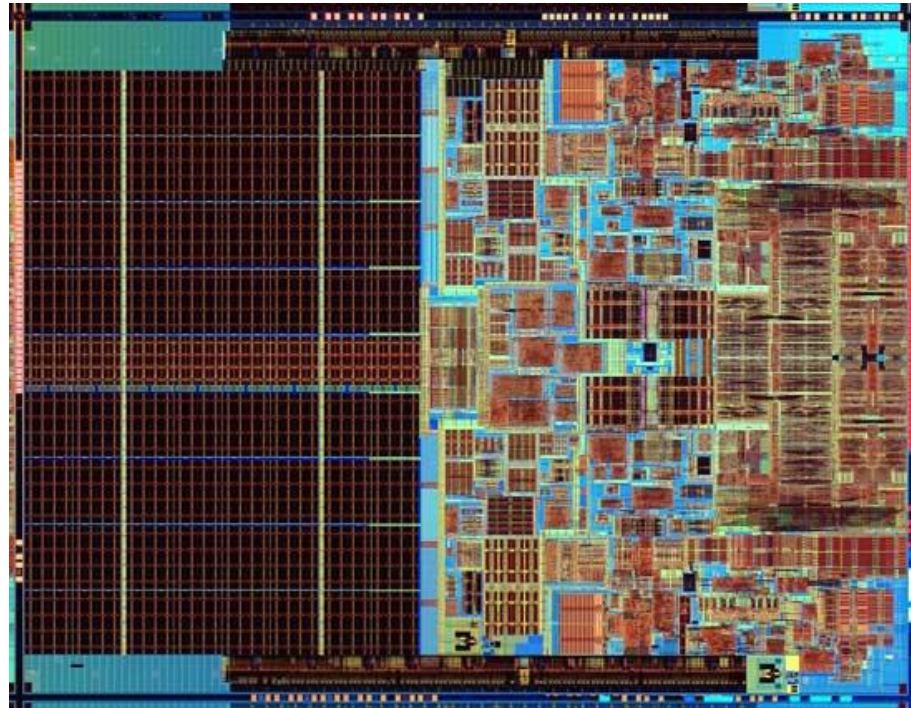
# Computer Architecture Topics



# Executive Summary

The processor  
you built in  
HY225

What you'll  
understand  
after taking  
HY425



Also, the  
technology behind  
multi-core  
processors