CS 758: Advanced Topics in Computer Architecture

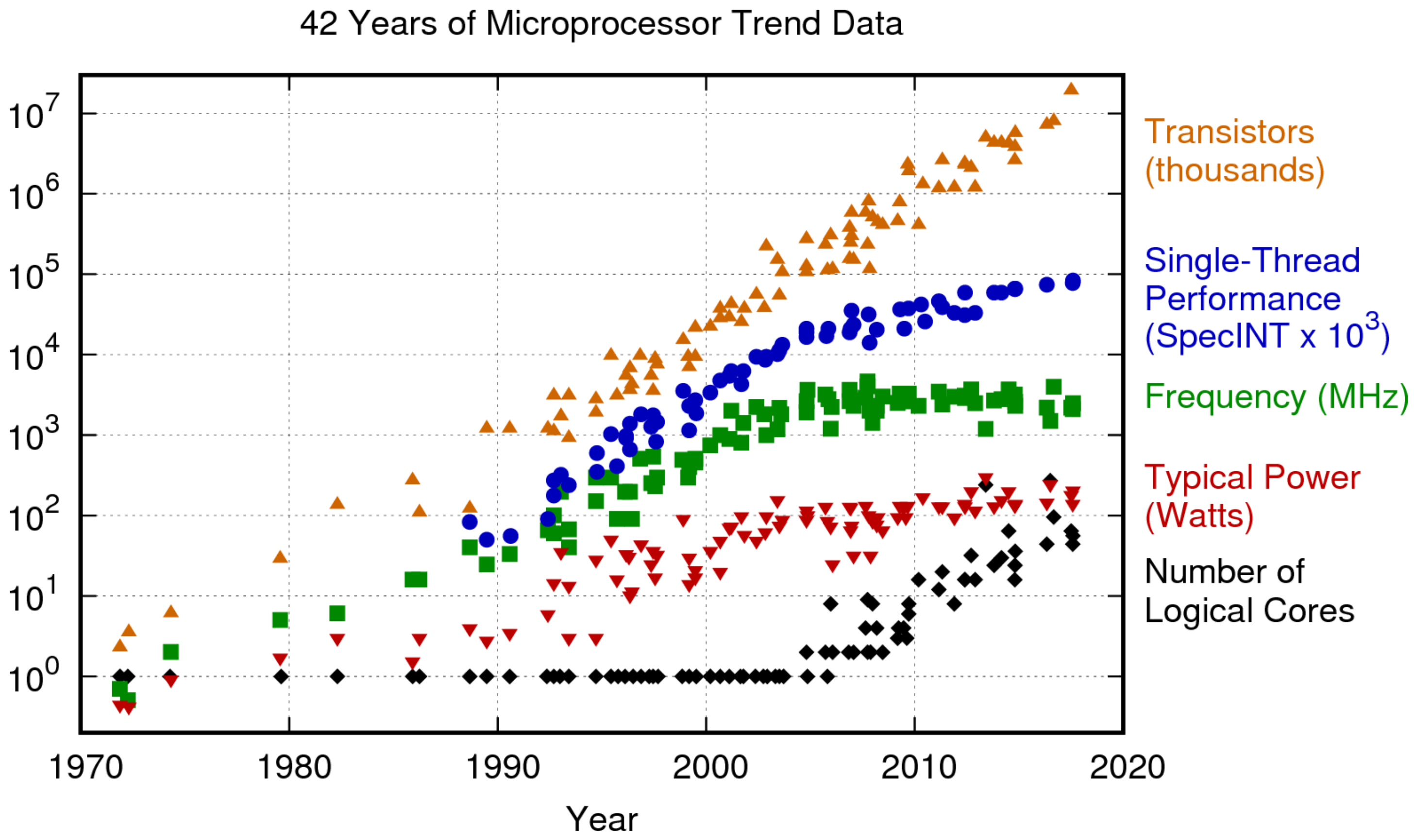
Lecture #18: Accelerator Wall

Professor Matthew D. Sinclair

Announcements

- Midterm #2 on Tuesday at night
  - Exam review on Tuesday in class
- Project lightning talks due following week (12/3 at 9 AM)
- Project presentations 12/10

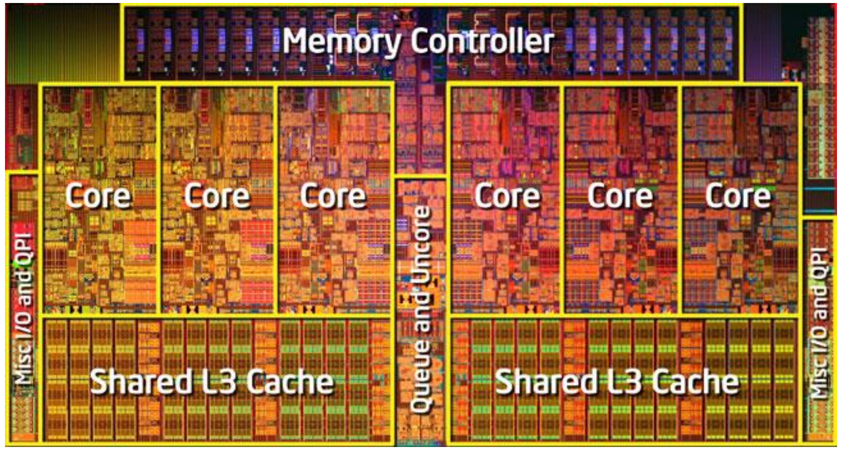
Moore’s Law



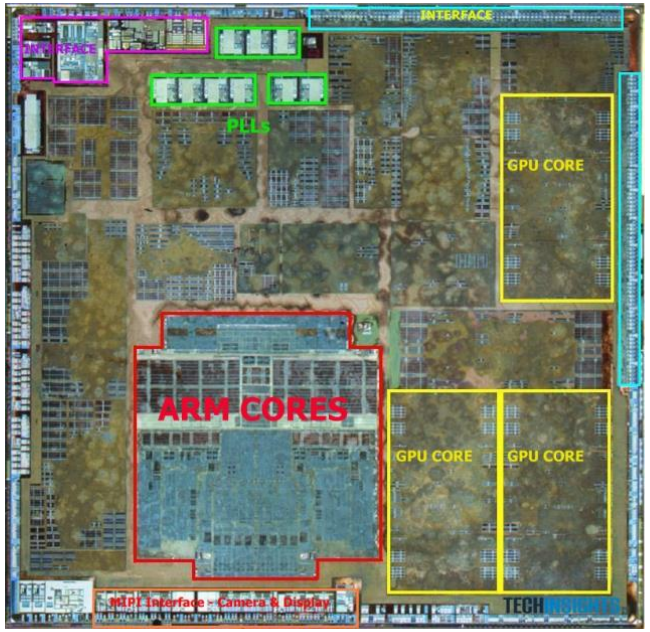
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Silver Bullet for Moore’s Law?

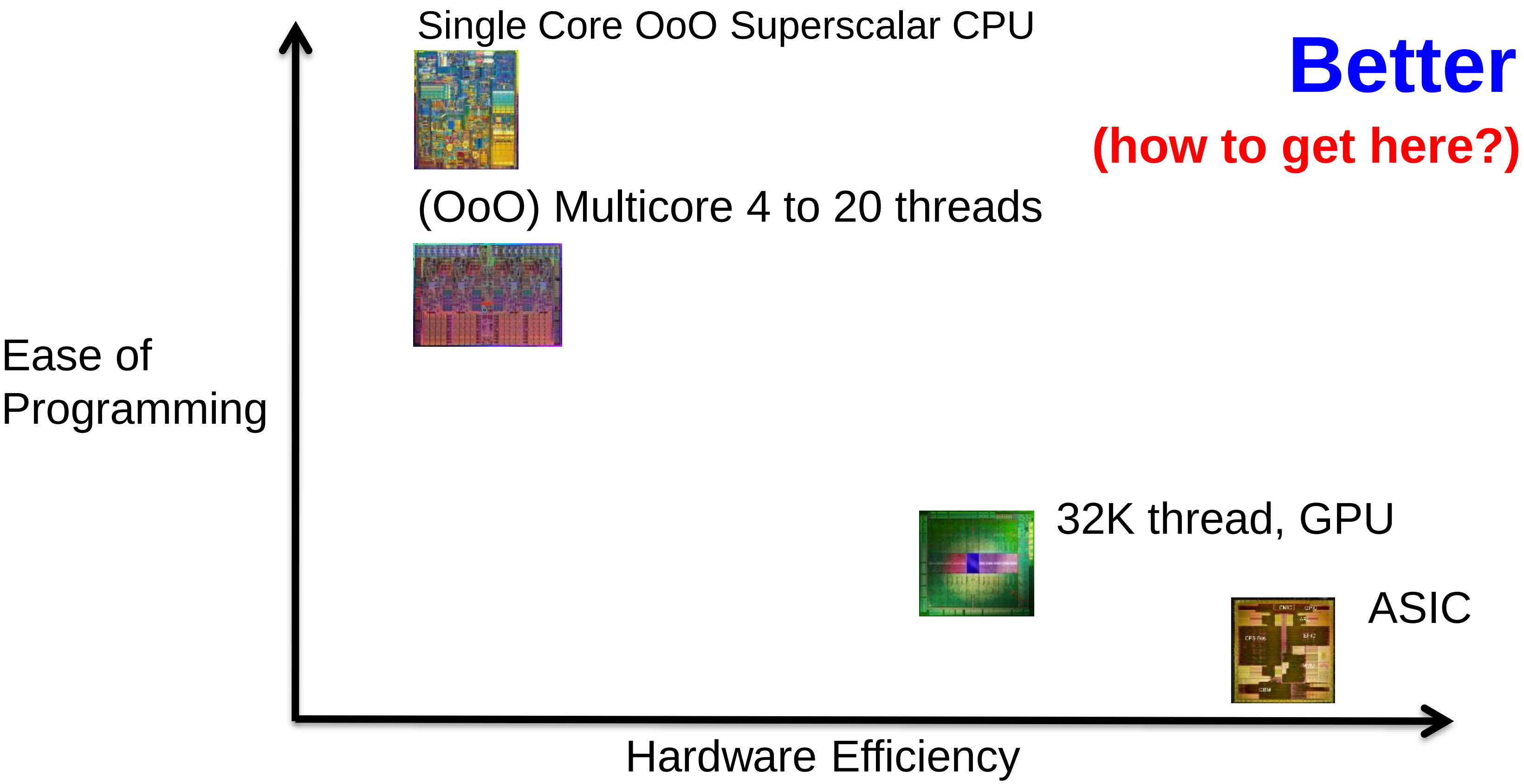
Parallelism



Specialization



Fundamental tradeoff: Programmability vs. Efficiency



Single Core OoO Superscalar CPU

(OoO) Multicore 4 to 20 threads

32K thread, GPU

ASIC

Better

(how to get here?)

Ease of Programming

Hardware Efficiency

Issues

- Moore’s Law is dead/dying
- Heterogeneous computing to the rescue?
- Just keep adding more ASICs?

ASIC approach

- No more/little transistor scaling coming …
- … so use the transistors in 3 easy steps:

1. Identify a killer application (e.g., machine learning)
2. Tape out an ASIC specialized for that application
3. Profit!

How to determine gains from specialization?

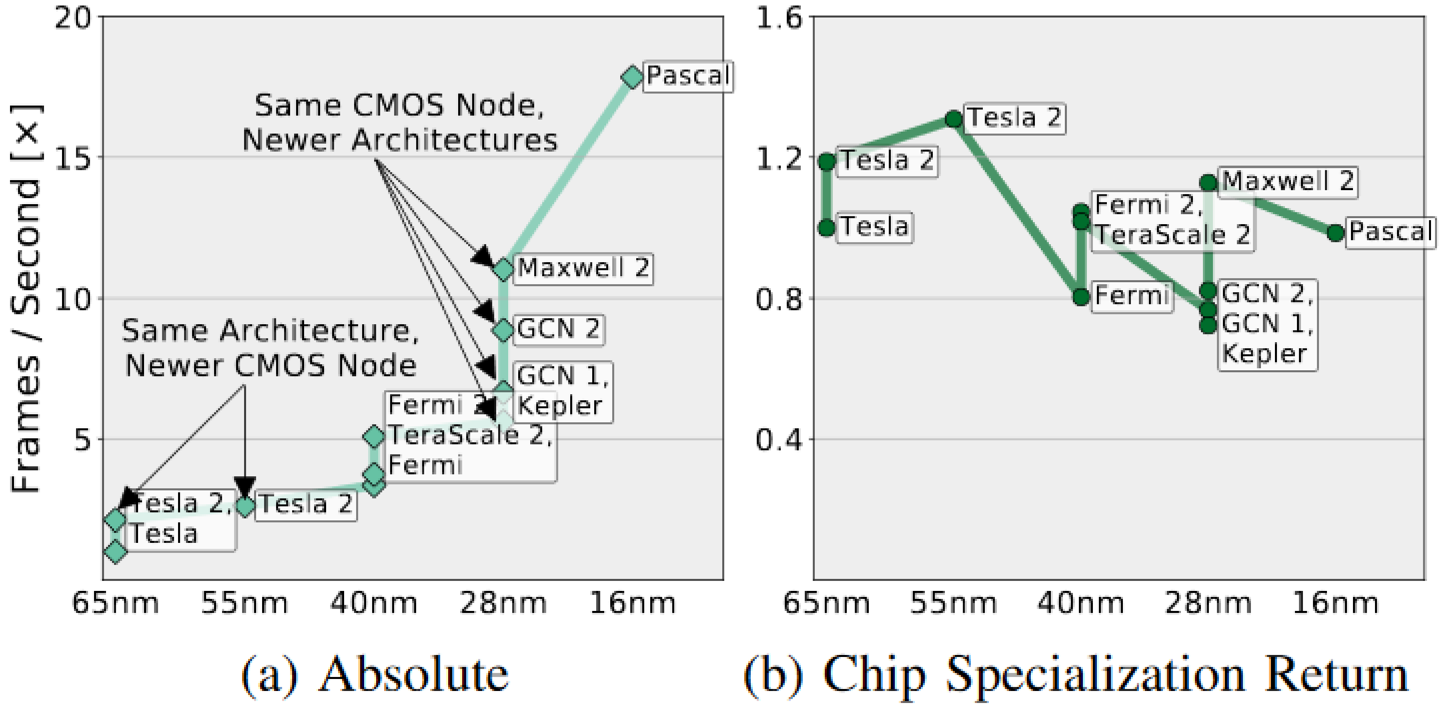
$$Gain = \underbrace{CMOS\ Potential}_{CMOS\text{–}Driven\ Return} \times \underbrace{\left(\frac{Gain}{CMOS\ Potential}\right)}_{Chip\ Specialziation\ Return}$$

This essentially only improves from transistor scaling

This improves with specialization

Once Moore’s Law ends, little to gain from CMOS Potential

Gains from transistor scaling vs. specialization



(a) Absolute

(b) Chip Specialization Return

- Trends:
  - Almost all of the true gains come from transistor scaling
  - Takes some iterating to optimize an architecture for CSR

Results

- Common patterns
  - Computational confinement: fixed and straightforward HW implementation
    - Issue: Amdahl’s Law bottlenecks already sped up
    - Hard to improve further without drastic redesign (easy things done already)
  - Massive parallelism:
    - Issue: eventually dark silicon will come for us all (or at least our chips)
    - Eventually won’t be able to increase the number of compute units further per mm^2
  - Domain maturity
    - Issue: once accelerators converge to optimal design, what else is left?
    - All domains will quickly become mature?

Need to think bigger, bolder → break inherent assumptions

Applications and software are not static