

EE382N-20 Computer Architecture

Parallelism and Locality

Lecture 1

Mattan Erez

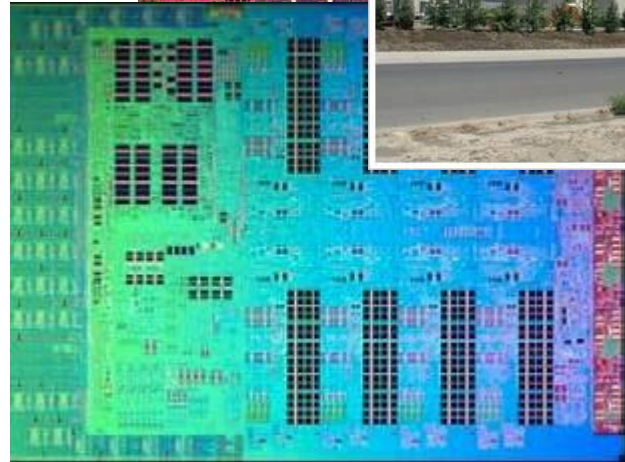
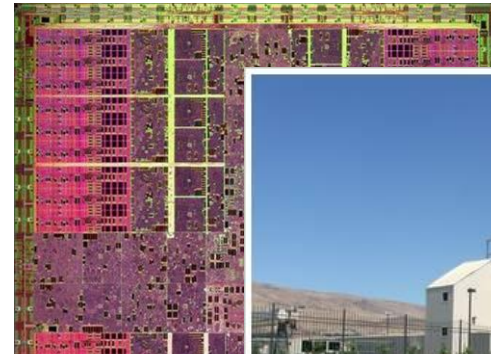
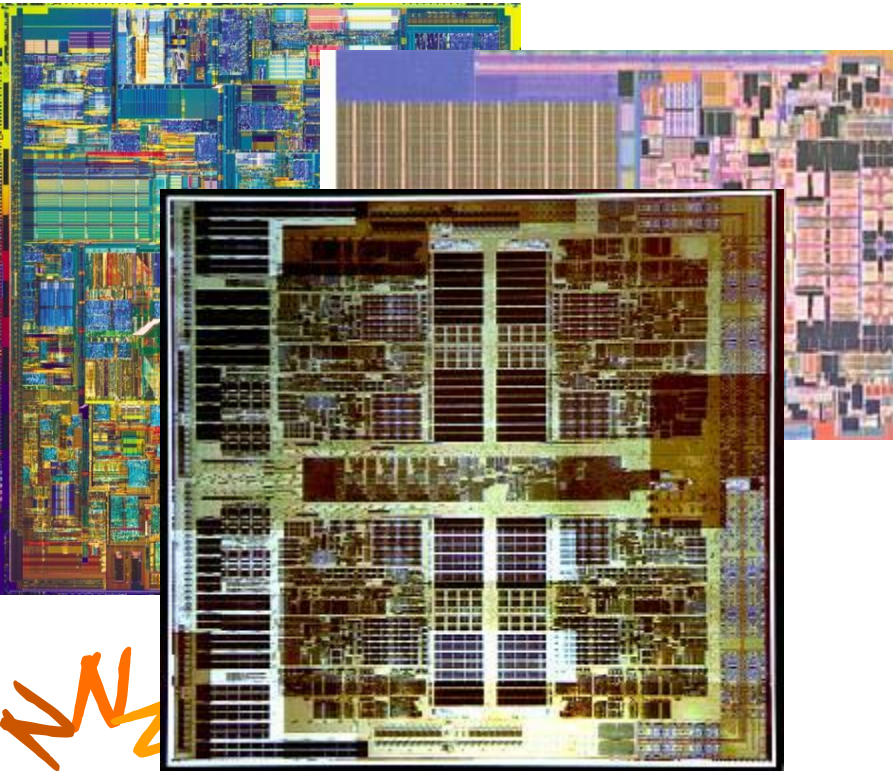


The University of Texas at Austin



What is this class about?

- Computer architecture
- Principles in computer architecture
 - Parallelism
 - Locality
 - Hierarchy



What is this class about?

- Computer architecture
- Principles in computer architecture
 - Parallelism
 - Locality
 - Hierarchy
- Advanced class computer architecture
 - Problems
 - Principles
 - Solutions
- Get some original research started



Outline (for today)

- Quick intro to computer architecture
 - What is it
 - What are the main challenges today
- What are parallelism, locality, and hierarchy
 - Why are they principles
 - How do they address the challenges
- Topics we'll cover in class
- Class procedures and expectations
- Other technicalities

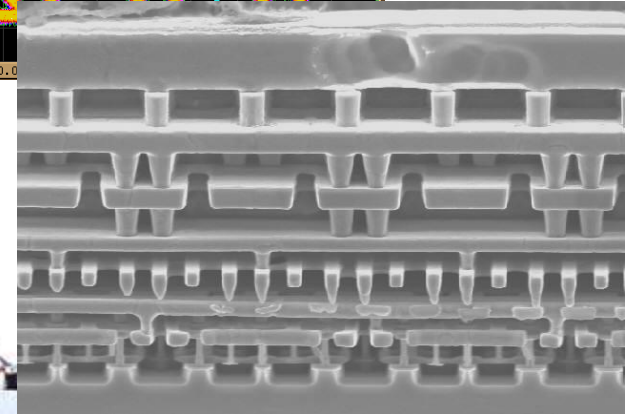
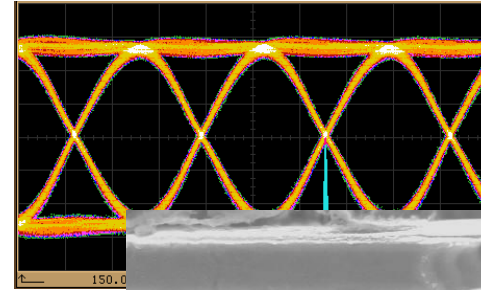


What is Architecture?

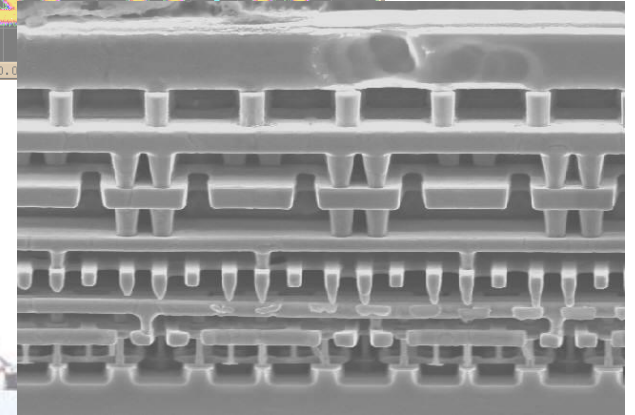
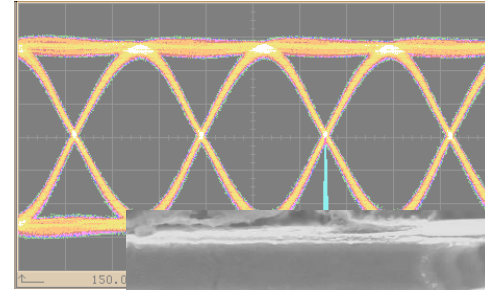
Form follows function Louis Sullivan



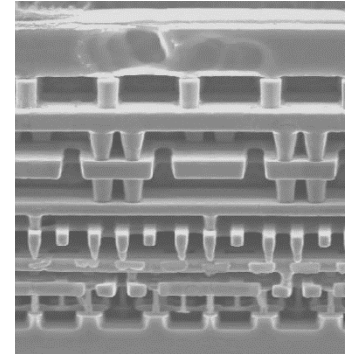
Computer Architects Match Hardware Technology with User Requirements



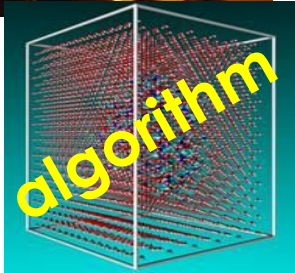
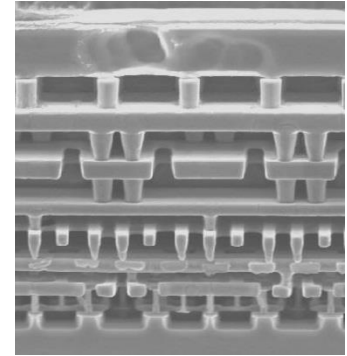
Computer Architects Match Hardware Technology with User Requirements



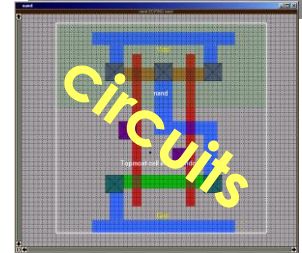
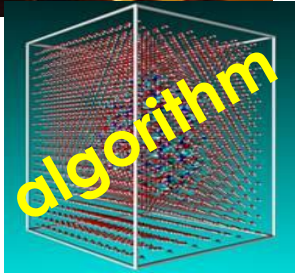
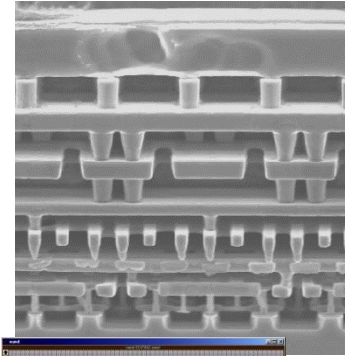
Computer Architects Match Hardware Technology with User Requirements



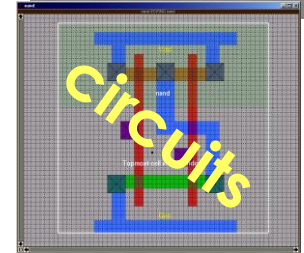
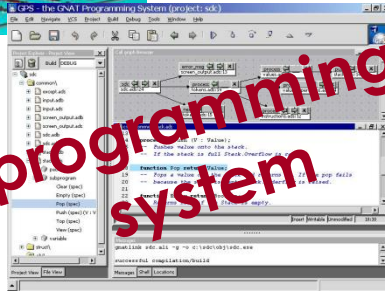
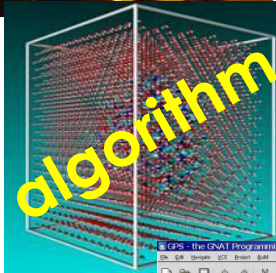
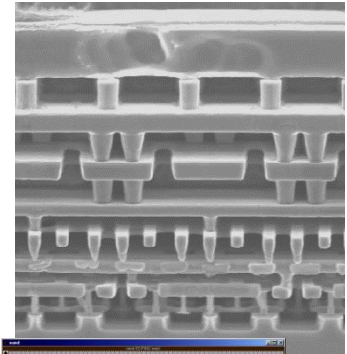
Computer Architects Match Hardware Technology with User Requirements



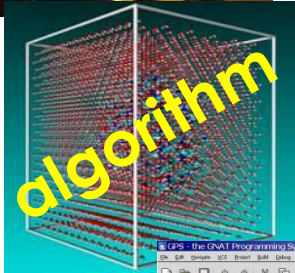
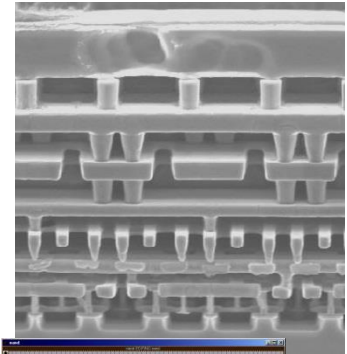
Computer Architects Match Hardware Technology with User Requirements



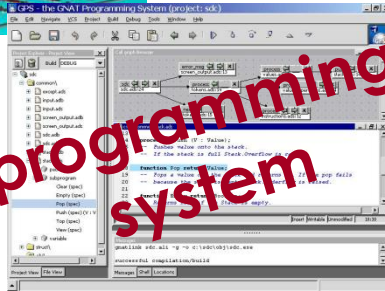
Computer Architects Match Hardware Technology with User Requirements



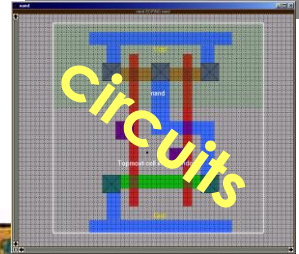
Computer Architects Match Hardware Technology with User Requirements



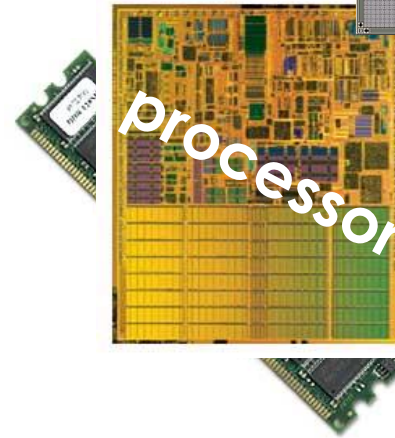
algorithm



programming system



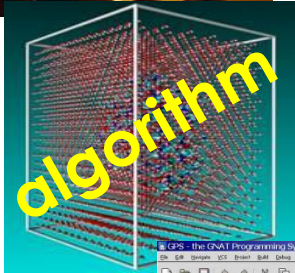
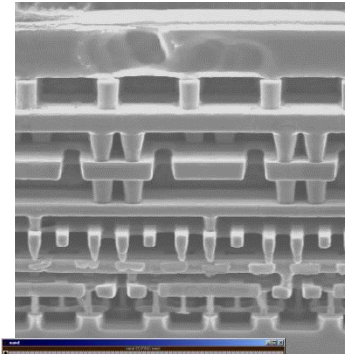
circuits



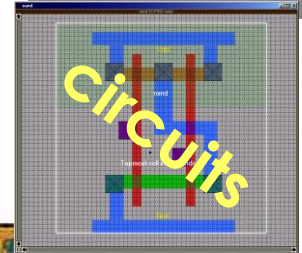
processor



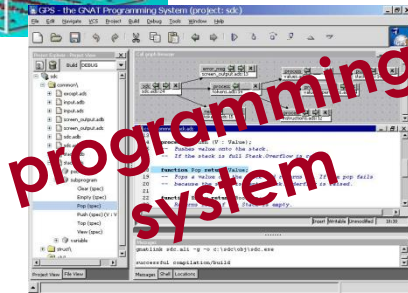
Computer Architects Match Hardware Technology with User Requirements



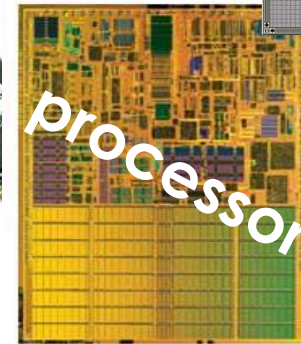
algorithm



circuits



programming system



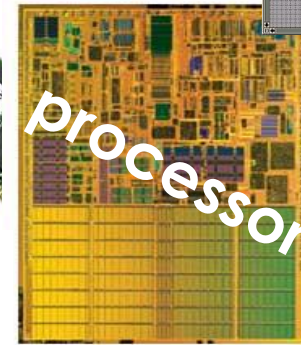
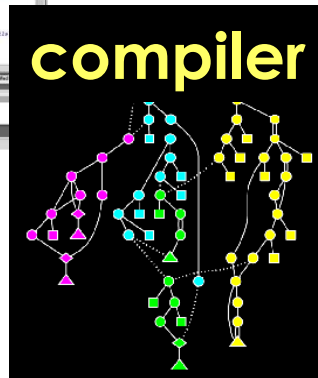
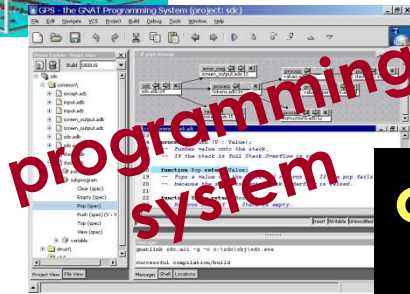
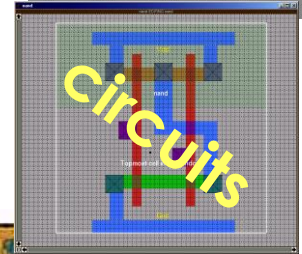
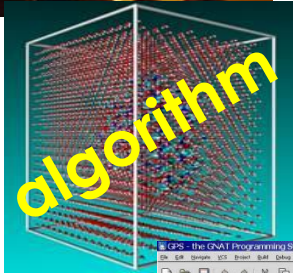
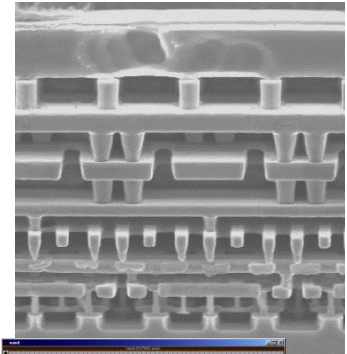
processor



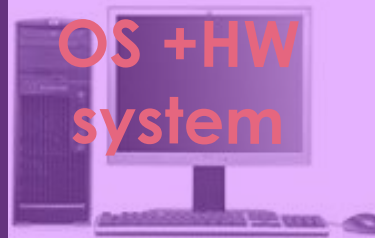
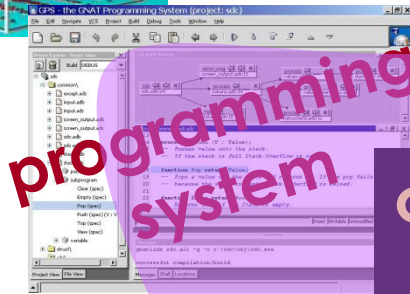
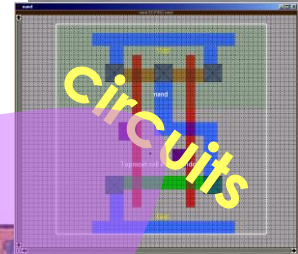
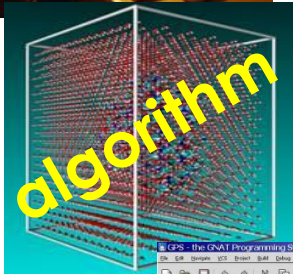
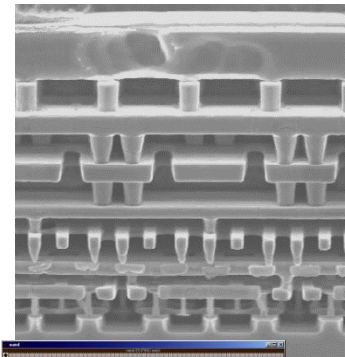
OS + HW system



Computer Architects Match Hardware Technology with User Requirements



Computer Architects Match Hardware Technology with User Requirements



Five “major” Challenges for Computer Architects

“The number of people saying
Moore’s Law is dead
doubles every 18 months”



The Performance Challenge

Wireless communications (3G, UWB, ...)



- Higher data rates
- More complex air interfaces

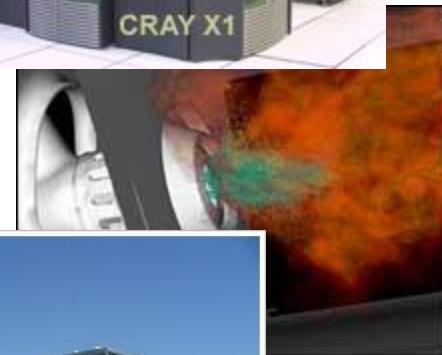
Workstations (Games, CAD)



- Higher resolution
- Realism
- Accuracy

Supercomputers (Scientific simulations)

- Fidelity
- Time scales



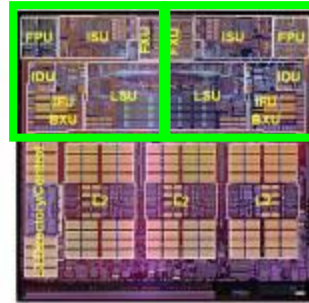
The (Power) Efficiency Challenge

Embedded



- Battery life and heat
- Commodity and volume

Mainstream “CPUs”



- Peak power
- Cooling

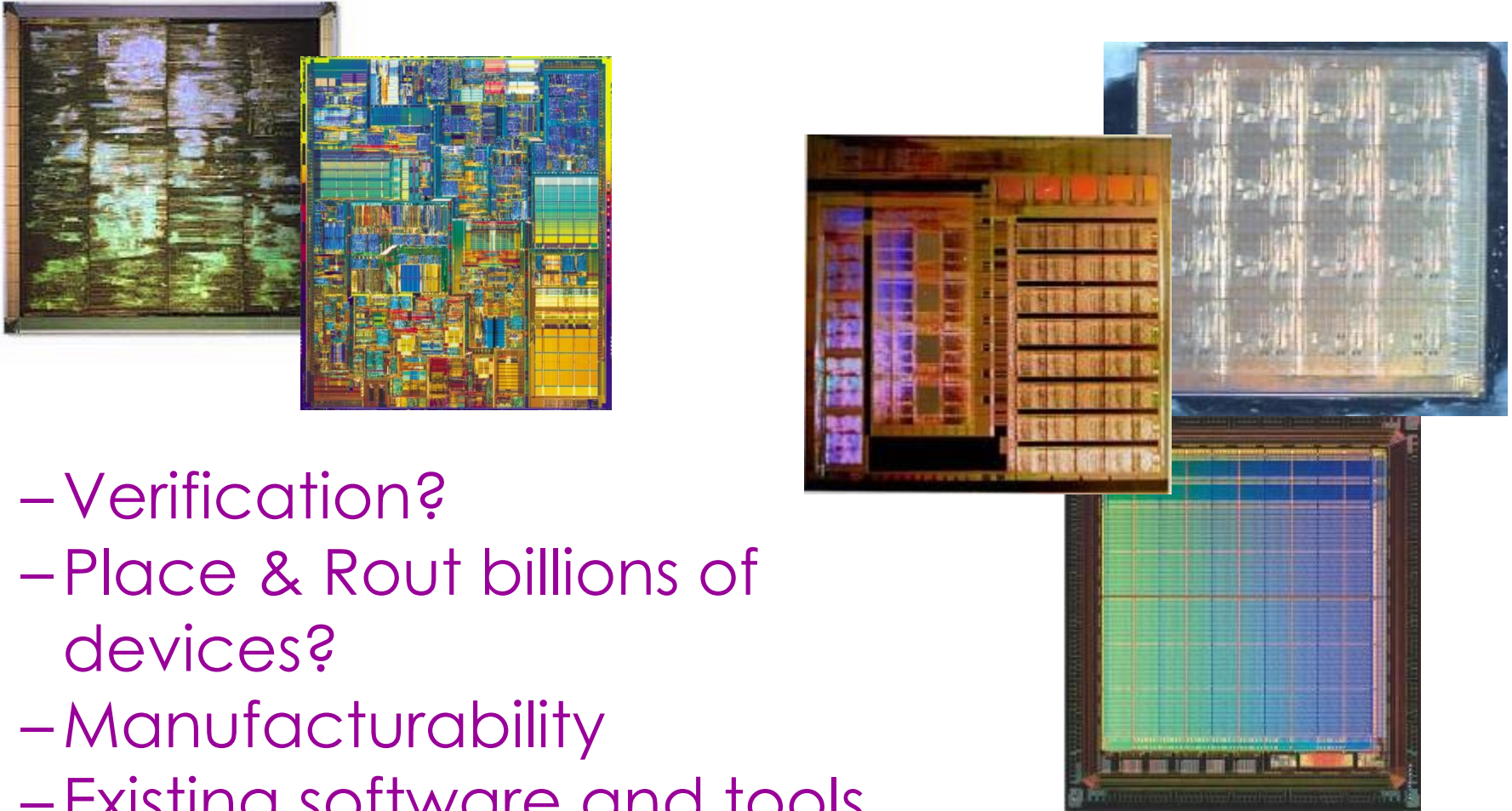
Supercomputers



- Energy bill (10 MW)
- Price/performance



The Practicality Challenge



- Verification?
- Place & Rout billions of devices?
- Manufacturability
- Existing software and tools



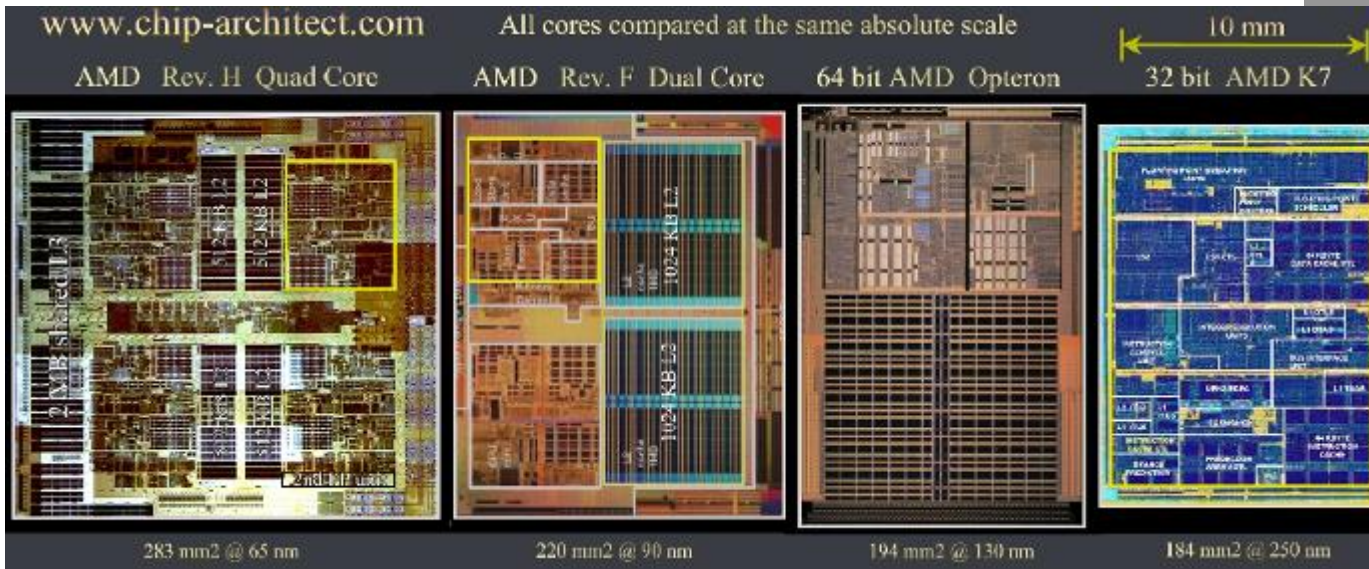
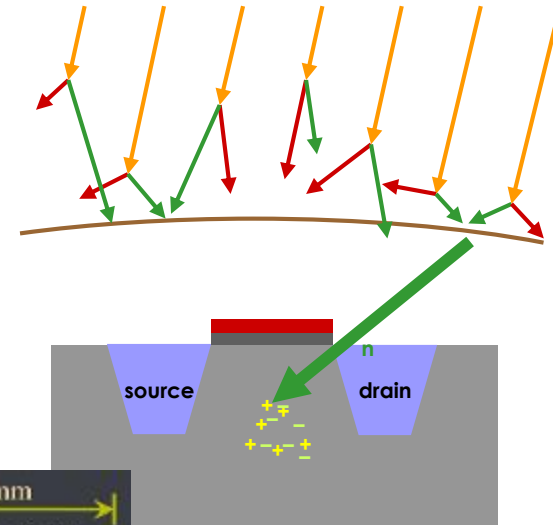
The Programmability Challenge

- Programmability reduces cost, enables adaptation, and improves time-to-market
 - Multiple modes
 - Evolving standards
 - Evolving features, differentiation
 - Design/tooling costs



The Reliability Challenge

- More devices
- Smaller devices
- Greater variability



What can we do?

- Specialize more
 - But still innovate on algorithms
- Compute less and store less
 - Use better algorithms
- **Proportionality: waste less**
 - Different applications and scenarios have different requirements
 - Main ways to save: locality, parallelism, and hierarchy



Overall Approach

Application layer:
numerical methods, DSLs

Portability and tuning layer

**HW-aware
abstractions**



**Automatic
transformations**

Architecture layer:
mechanisms

Physical component layer:
power, bandwidth, performance



But first ...

- To get an A, you will have to put in effort
 - I will be less lenient than historical evidence suggests, especially w.r.t. final project
- Work will be open-ended
 - You'll have to figure out what to do on your own
 - You'll have to figure out how to do it
 - You'll have to decide what is good enough for you
- Lab 1 is already out
 - It will take time
 - It's the easiest because most spelled out
- Work in groups
 - Groups of 3 ideal and expected (see me for rare exceptions)
 - Switch teams for the 3 labs – plenty of great people here



Outline (for today)

- Quick intro to computer architecture
 - What is it
 - What are the main challenges today
- What are parallelism, locality, and hierarchy
 - Why are they principles
 - How do they address the challenges
- Topics we'll cover in class
- Class procedures and expectations
- Other technicalities

Rest of class was on the whiteboard

