CS/ECE 5381/7381
Computer Architecture
Spring 2023

Dr. Manikas
Computer Science
Lecture 2: Jan. 19, 2023

Assignments

- Quiz 1 – due Sat., Jan. 21 (11:59 pm)
  - Covers concepts from Module 1 (this week)

Quiz 1 Details

- The quiz is open book and open notes.
- You are allowed 90 minutes to take this quiz.
- You are allowed 2 attempts to take this quiz - your highest score will be kept.
  - Note that some questions (e.g., fill in the blank) will need to be graded manually
- Quiz answers will be made available 24 hours after the quiz due date.

Fundamentals of Quantitative Design and Analysis

(Chapter 1, Hennessy and Patterson)

Note: some course slides adopted from publisher-provided material

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- <u>1.4 Trends in Technology</u>
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Trends in Technology

- Integrated circuit technology (chip)
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year

- DRAM capacity: 25-40%/year (slowing)
  - This is the RAM in your computer
  - 8 GB (2014), 16 GB (2019)

Trends in Technology

- Flash capacity: 50-60%/year
  - 8-10X cheaper/bit than DRAM

- Magnetic disk capacity: recently slowed to 5%/year
  - 8-10X cheaper/bit than Flash
    - Eventually to be replaced by Flash (SSD) as Flash costs decrease
  - 200-300X cheaper/bit than DRAM

Bandwidth and Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 32,000-40,000X improvement for processors
  - 300-1200X improvement for memory and disks

- Latency or response time
  - Time between start and completion of an event
  - 50-90X improvement for processors
  - 6-8X improvement for memory and disks

Transistors and Wires

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .011 microns in 2017
    - 1 micron = 1 micrometer = 10^{-6} meters
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Power and Energy

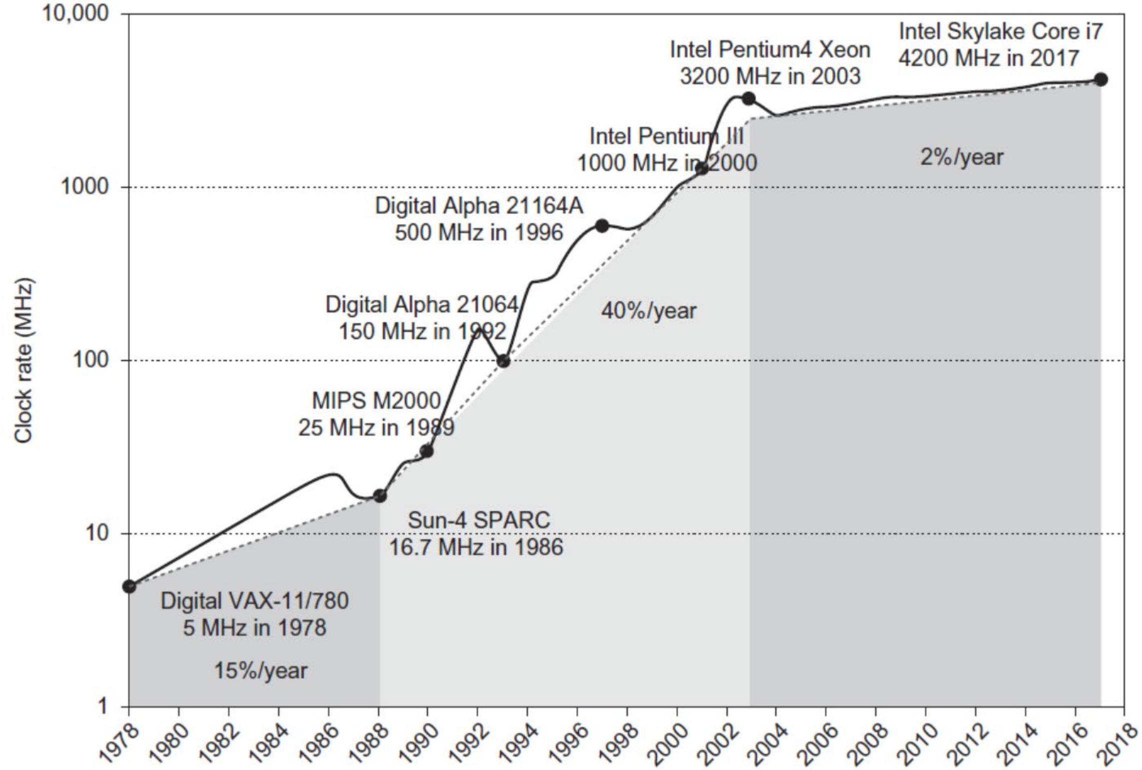
- Problem: Get power in, get power out
- Thermal Design Power (TDP)
  - Characterizes sustained power consumption
  - Used as target for power supply and cooling system
  - Lower than peak power, higher than average power consumption
- Clock rate can be reduced dynamically to limit power consumption
- Energy per task is often a better measurement

Dynamic Energy and Power

- Dynamic energy
  - Transistor switch from 0 -> 1 or 1 -> 0
  - ½ x Capacitive load x Voltage^2
- Dynamic power
  - ½ x Capacitive load x Voltage^2 x Frequency switched
- Reducing clock rate reduces power, not energy

Power

- Intel 80386 (1986) consumed ~ 2 W
- 3.3 GHz Intel Core i7 (2017) consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air



Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Trends in Cost

- Impact of Time
- Impact of Volume
- Cost of an Integrated Circuit
- Cost vs. Price

Impact of Time

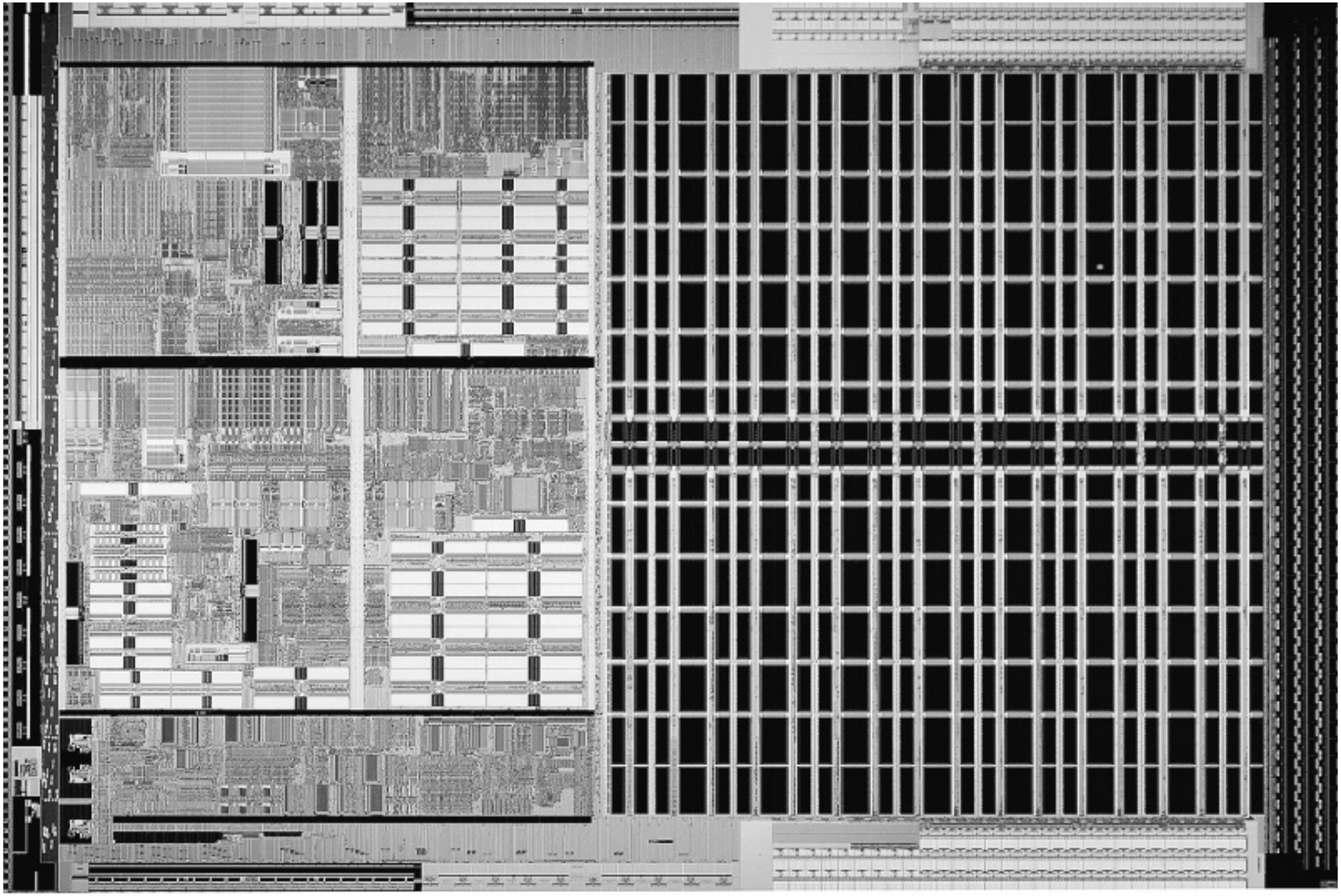
- For a given design, manufacturing costs decrease over time
  - "learning curve" – initial design likely to have defects
  - Later implementations are more reliable

Impact of Volume

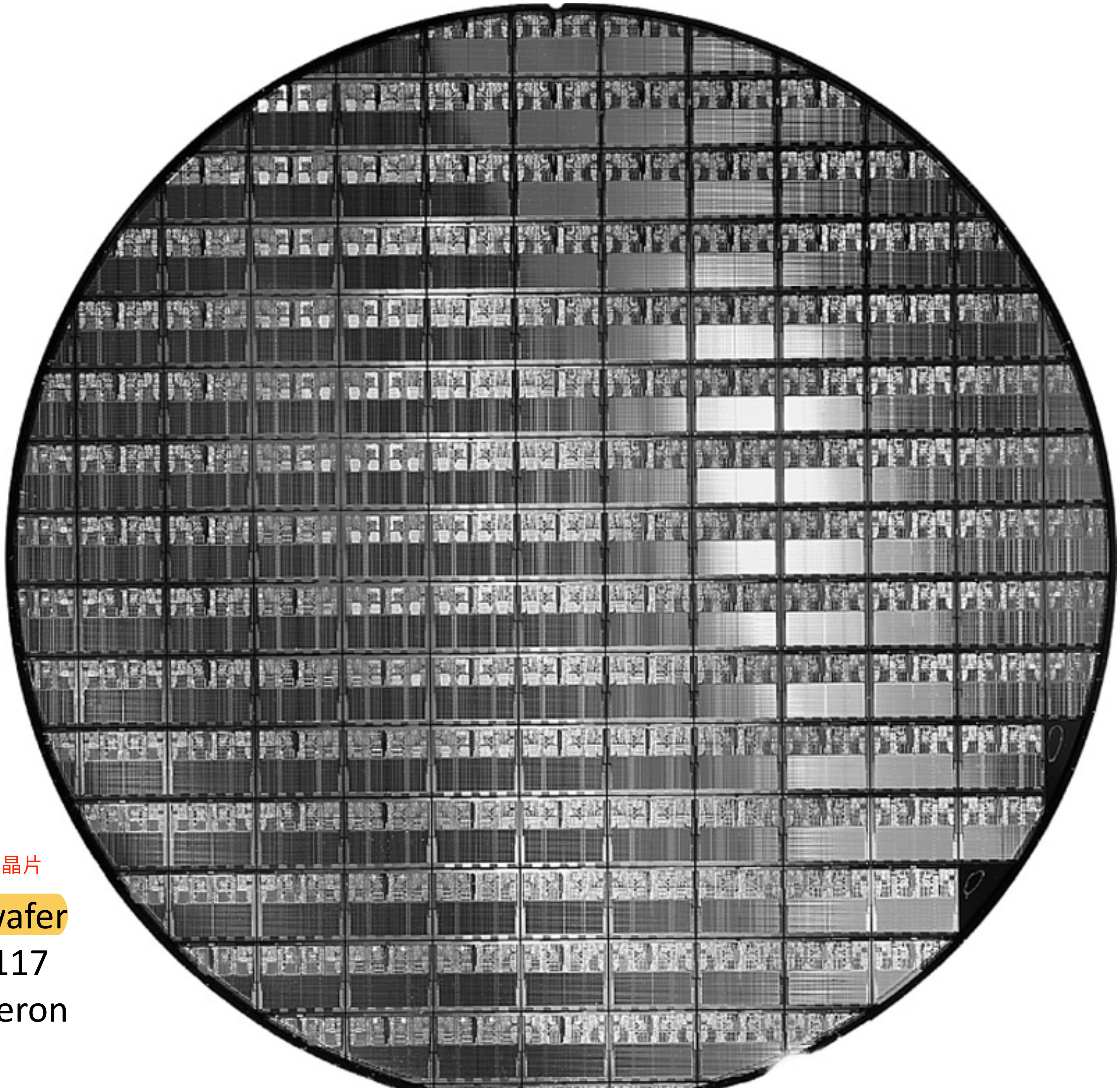
- Chip design and manufacturing cost
  - Cost per part (manufacturing)
  - NRE (Non-Recurring Expense)
    非经常开支
    - One-time cost for design and equipment set-up
    - Usually more expensive than cost per part
- High-volume designs can spread out NRE
  容量
  - Cheaper total cost per part

Cost of an Integrated Circuit (IC)

- Computers contain chips (integrated circuits)
  - MPU, DRAM, etc.
- Thus, IC cost affects computer cost
- A given IC is manufactured as a *die*
- Several dies (IC copies) are manufactured on a *wafer*



Die for an AMD Opteron microprocessor (MPU)



晶片

300mm wafer contains 117 AMD Opteron chips

晶圆(Wafer): 晶圆圆是半导体集成电路的核心材料,是一种圆形的板

晶粒(Die): 很多四边形都聚集在圆形晶圆上。 这些四边形都是集成电子电路的IC芯片

Dies per Wafer

- Area of wafer / area of die

  晶圆面积 晶体面积

- Wafer is round, but die is square
- Estimate number of dies as

$$\frac{Dies}{Wafer} = \frac{\pi\left(Diameter_{wafer}/2\right)^2}{Area_{die}} - \frac{\pi\left(Diameter_{wafer}\right)}{\sqrt{2\left(Area_{die}\right)}}$$

Die Yield

- Fraction of "good" dies on wafer
  - Dies without manufacturing defects

$$Yield_{die} = \frac{Yield_{wafer}}{\left[1 + (defects / unitArea)(area_{die})\right]^N}$$

$$\text{where } N = \text{process - complexity factor}$$

$$\text{Typical contemporary values (for 40 nm process):}$$

$$N = 12$$

$$\text{defects/unit\_area} = 0.04 \text{ defects/cm}^2$$

Effects on IC cost

$$\text{cost}_{\text{die}} = \frac{\text{wafer}_{\text{cost}}}{(\text{dies/wafer})\text{yield}_{\text{die}}}$$

$$\text{cost}_{\text{IC}} = \frac{\text{cost}_{\text{die}} + \text{cost}_{\text{testing\_die}} + \text{cost}_{\text{packaging\_and\_final\_test}}}{\text{yield}_{\text{final\_test}}}$$

Cost vs. Price

- Margin = sales price – manufacturing cost
- Margins cover overhead costs
  - Salaries, benefits, utilities, equipment, maintenance
  - R & D, sales, manufacturing

Example 1.6.1

- New chip with code name “Peruna”
  - Die area is 250 mm^2
  - To be fabricated on wafer with diameter of 300 mm
- Fabrication parameters:
  - Estimated defect rate = 0.03 per cm^2
  - Wafer yield = 100%
  - Process-complexity factor N = 12

Example 1.6.1

a. How many Peruna dies can we fabricate on a wafer?
b. What is the die yield?
c. If we can make a $20 profit per defect-free chip, how much profit can we make for a wafer of Pernuna dies?

没有缺陷

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Dependability

- Module reliability
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF) = MTTF + MTTR
  - Availability = MTTF / MTBF
- Failures in Time (FIT)
  - Rate of failures per *billion* hours
  - MTTF = 10^9/FIT

Example 1.7.1

- Our Peruna chip has the following parameters
  - FIT = 150
  - MTTR = 2 days

a. What is the MTTF of our chip?

b. What is the availability of our chip?

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Measuring, Reporting, and Summarizing Performance

- Performance = “speed” of computer
- User view – how fast does my program run?
  - Execution (response) time: time between start and end of an event
- Web administrator view – how many transactions/hour?
  - Throughput: total amount of work done in a given time

Measuring Execution Time

- User view: “Wall-clock” time
  - E.g., process started at 2:00 pm, ended at 2:10 pm
- Actual execution time: CPU time
  - Time that processor is actually computing
  - Ignores waiting time for I/O, other programs

Benchmarks

- Common set of programs to run on different computers
- SPEC (Standard Performance Evaluation Corporation)
  - Started in 1988 by group of workstation vendors.
  - Non-profit org - used as indep. testing source.
  - Goal: produce benchmarks that measure "real" performance.
  - Becoming the standard for performance measurement.

SPEC Benchmarks

- Benchmark types:
  - Open Systems: benchmarks for PCs, servers
  - High Performance: supercomputers
  - Graphics: high-end graphical workstations (CAD, simulators, games)

Reporting Performance Results

- Experiments should be *reproducible*
  - Report sufficient information such that another researcher can get the same results (assuming he/she follows your exact approach)
- SPEC benchmark reports require detailed info on computer, compiler, program parameters, etc.

Summarizing Performance Results

- We want to compare two computers (A and B)
- Run several different SPEC benchmarks on both computer A and B
- For computer j, SPECRatio for a given benchmark is

$$SPECRatio_j = \frac{Exec\_Time_{ref}}{Exec\_Time_j}$$

$$\text{where ref} = \text{a reference computer (baseline)}$$

$$\text{shorter exec time} \Rightarrow \text{larger SPECRatio}$$

Outline

- 1.1 Introduction
- 1.2 Classes of Computers
- 1.3 Defining Computer Architecture
- 1.4 Trends in Technology
- 1.5 Trends in Power and Energy in Integrated Circuits
- 1.6 Trends in Cost
- 1.7 Dependability
- 1.8 Measuring, Reporting, and Summarizing Performance
- 1.9 Quantitative Principles of Computer Design

Quantitative Principles of Computer Design

1. Parallelism
2. Principle of locality
3. Focus on common case
4. Amdahl’s Law
5. Processor Performance Equation

Take Advantage of Parallelism

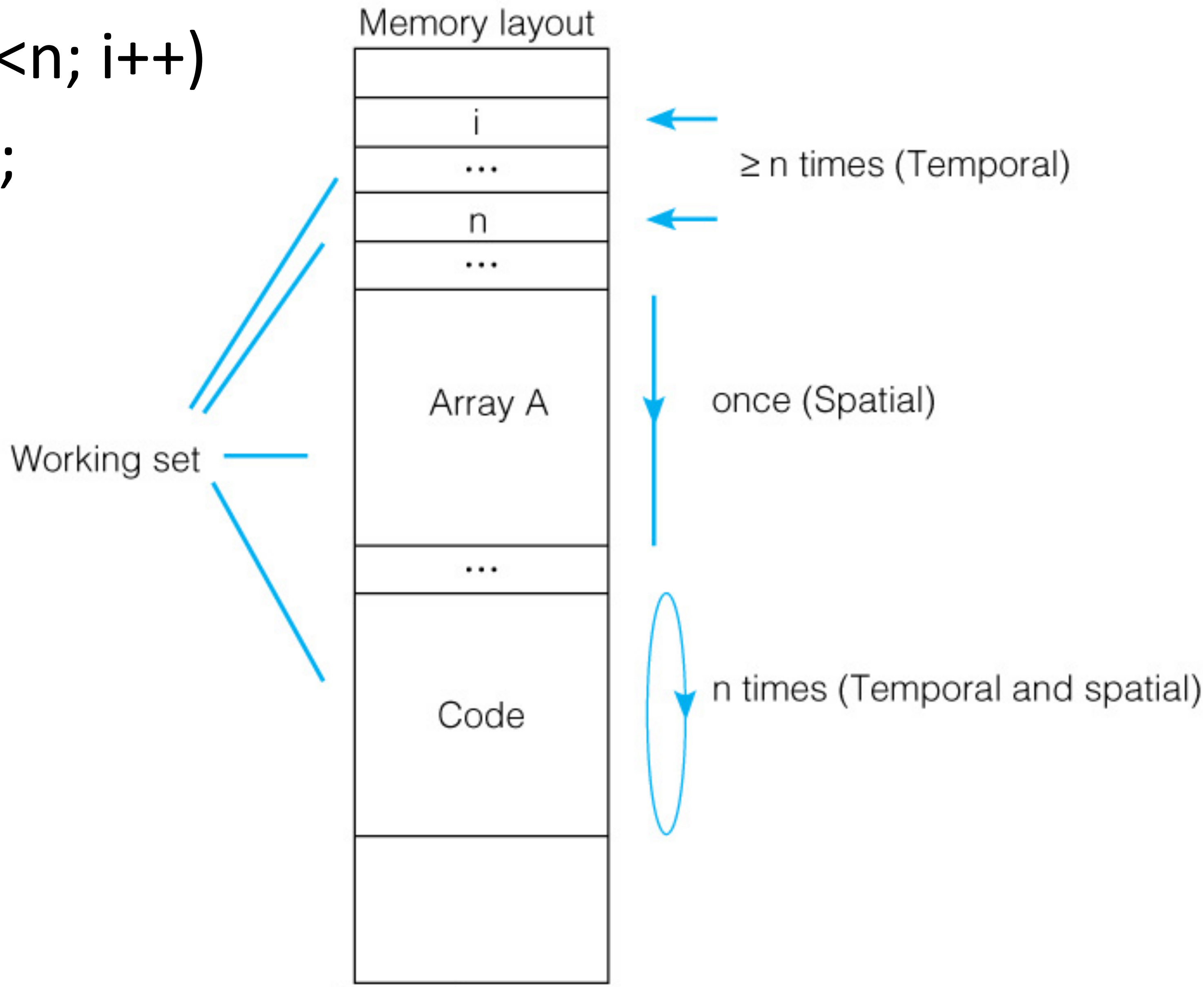
- Parallel operations = faster execution time
- System level – multiple processors
- Processor level – can we do two (or more) tasks at the same time? (pipelining)

Principle of Locality

- Programs tend to reuse recent data and instructions
- **Temporal Locality:** items that have been recently accessed will likely be accessed again soon
- **Spatial Locality:** items that are near each other (memory address) will likely be accessed close together in time

for (i=0; i<n; i++)

A[i] = 0;



Focus on the Common Case

- Most designs have trade-offs: optimizing for one objective will degrade another objective
- For a given design, optimize for main purpose
  - Graphics workstation – optimize mathematical operation speed, at expense of power
  - Laptop – optimize battery life, at expense of operation speed