Parallel and Distributed Computing

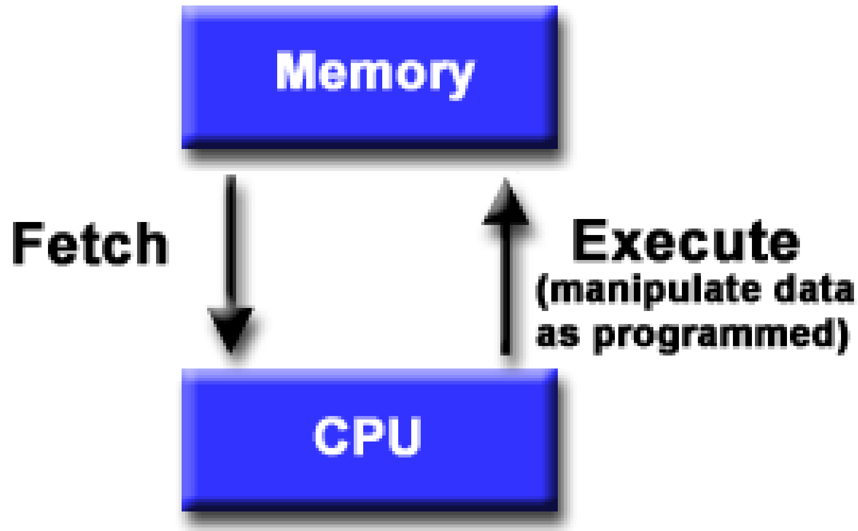
Chapter 1: Introduction to Parallel Computing

Jun Zhang

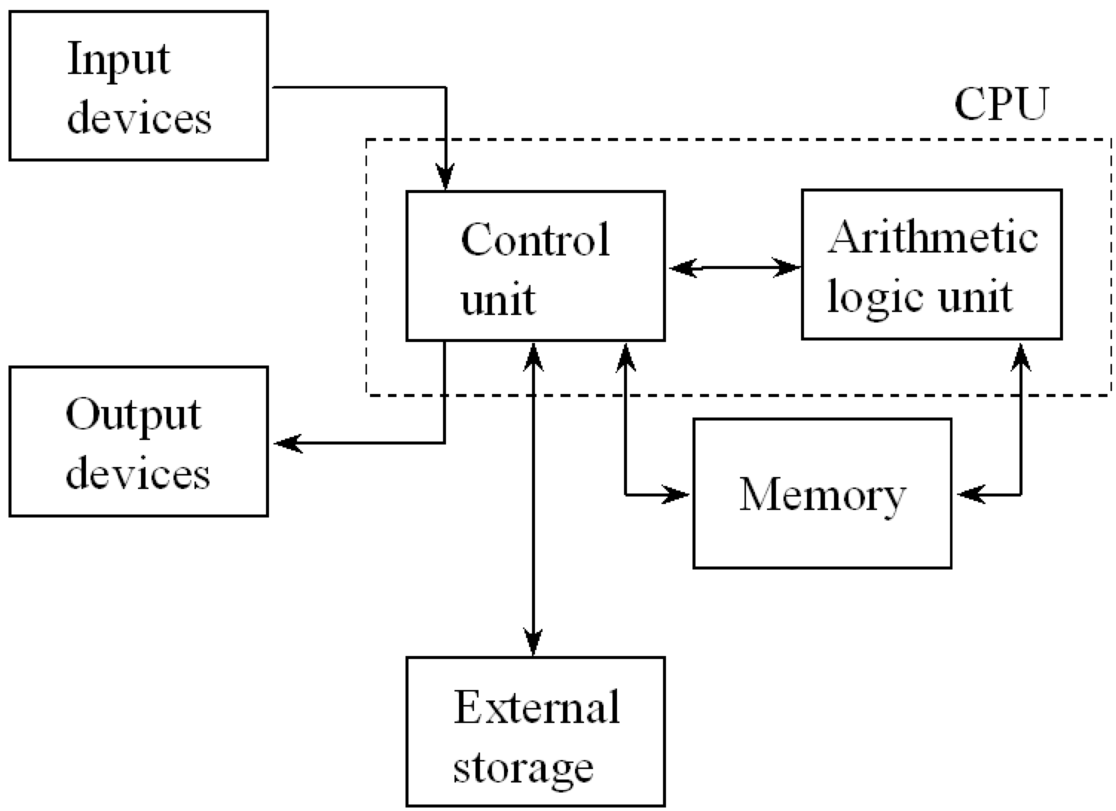
Laboratory for High Performance Computing & Computer Simulation
Department of Computer Science
University of Kentucky
Lexington, KY 40506

1.1a: von Neumann Architecture

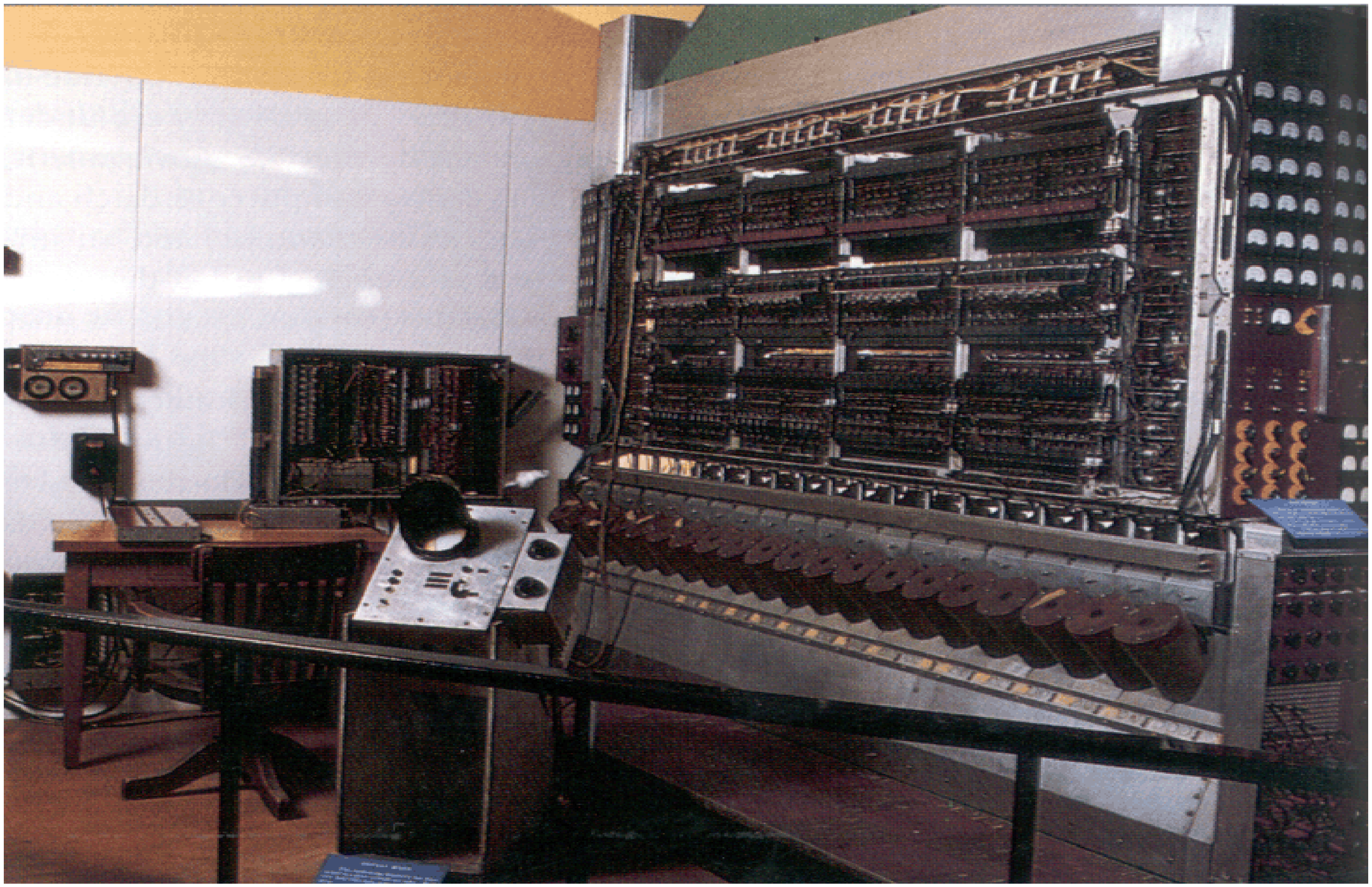
- Common machine model for over 40 years
- Stored program concept
- CPU executes a stored program
- A sequence of read and write operations on the memory
- Order of operations is sequential



1.1b: A More Detailed Architecture based on von Neumann Model



1.1c: Old von Neumann Computer



1.1d: CISC von Neumann Computer

- CISC stands for Complex Instruction Set Computer with a single bus system
- Harvard (RISC) architecture utilizes two buses, a separate data bus and an address bus
- RISC stands for Reduced Instruction Set Computer
- They are SISD machines – Single Instruction Stream on Single Data Stream

1.1e: Personal Computer

1.1f: John von Neumann

- December 28, 1903 – February 8, 1957
- Hungarian mathematician
- Mastered calculus at 8
- Graduate level math at 12
- Got his Ph.D. at 23
- His proposal to his 1st wife, “You and I might be able to have some fun together, seeing as how we both like to drink."

1.2a: Motivations for Parallel Computing

- Fundamental limits on single processor speed
- Disparity between CPU & memory speeds
- Distributed data communications
- Need for very large scale computing platforms

1.2b: Fundamental Limits – Cycle Speed

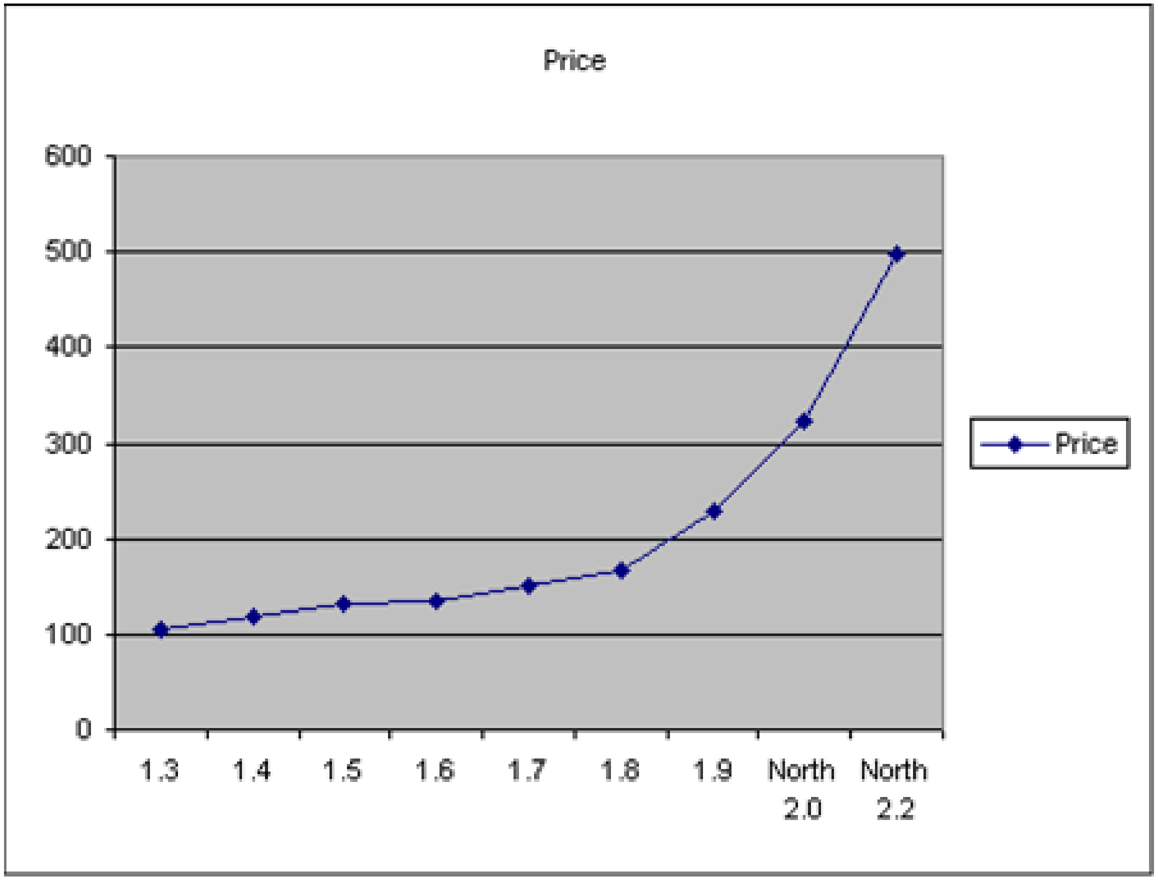
Cray 1:	12ns	1975
Cray 2:	6ns	1986
Cray T-90	2ns	1997
Intel PC	1ns	2000
Today's PC	0.3ns	2006 (P4)

- Speed of light: 30cm in 1ns
- Signal travels about 10 times slower

1.2c: High-End CPU is Expensive

Price for high-end CPU rises sharply

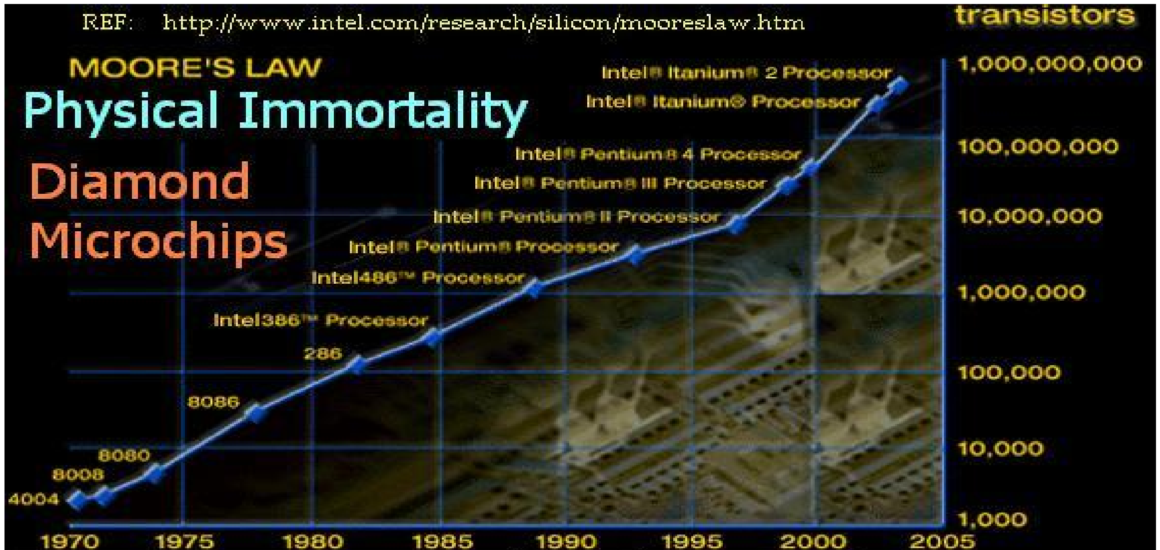
Intel processor Price/performance



1.2d: Moore’s Law

- Moore’s observation in 1965: the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented
- Moore’s revised observation in 1975: the pace slowed down a bit, but data density has doubled approximately every 18 months
- How about the future? (price of computing power falls by half every 18 months?)

1.2e: Moore’s Law – Held for Now

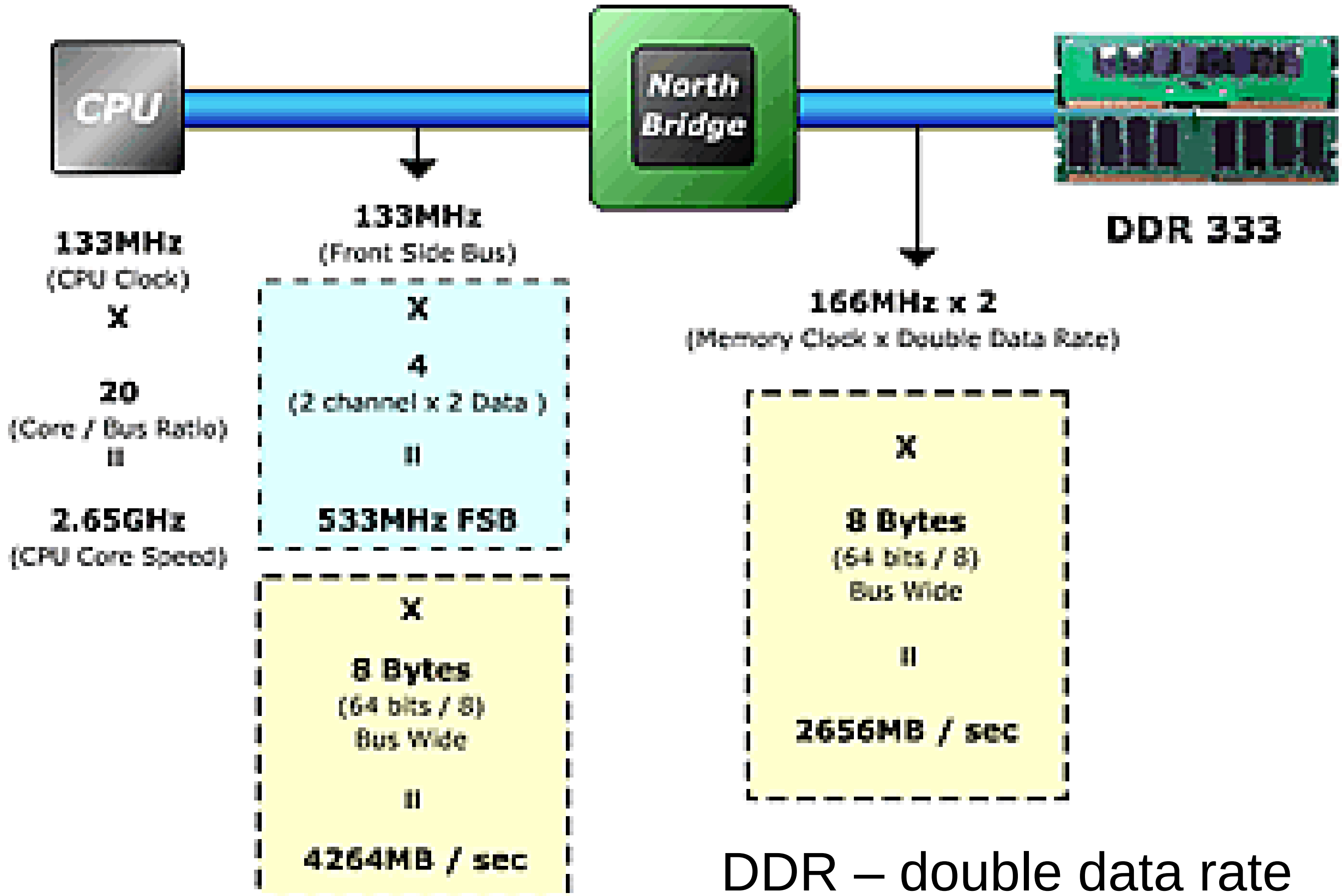


	Year of Introduction	Transistors
4004	1971	2,250
8008	1972	2,500
8080	1974	5,000
8086	1978	29,000
286	1982	120,000
Intel386™ processor	1985	275,000
Intel486™ processor	1989	1,180,000
Intel® Pentium® processor	1993	3,100,000
Intel® Pentium® II processor	1997	7,500,000
Intel® Pentium® III processor	1999	24,000,000
Intel® Pentium® 4 processor	2000	42,000,000
Intel® Itanium® processor	2002	220,000,000
Intel® Itanium® 2 processor	2003	410,000,000

1.3a: CPU and Memory Speeds

- In 20 years, CPU speed (clock rate) has increased by a factor of 1000
- DRAM speed has increased only by a factor of smaller than 4
- How to feed data *faster enough* to keep CPU busy?
- CPU speed: 1-2 ns
- DRAM speed: 50-60 ns
- Cache: 10 ns

1.3b: Memory Access and CPU Speed



DDR – double data rate

1.3c: Possible Solutions

- A hierarchy of successive fast memory devices (multilevel caches)
- Location of data reference
- Efficient programming can be an issue
- Parallel systems may provide

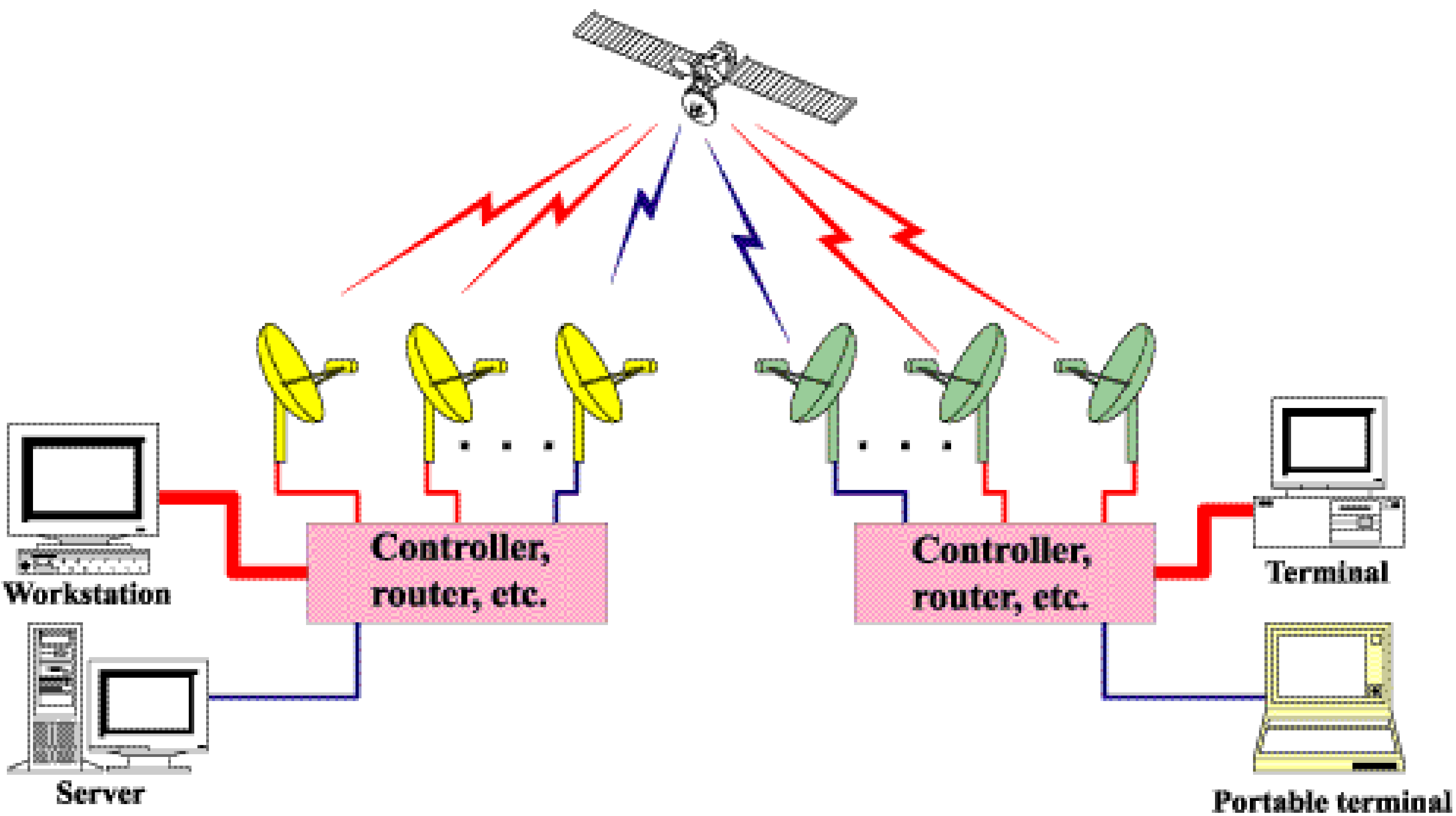
  1.) larger aggregate cache

  2.) higher aggregate bandwidth to the memory system

1.4a: Distributed Data Communications

- Data may be collected and stored at different locations
- It is expensive to bring them to a central location for processing
- Many computing assignments many be inherently parallel
- Privacy issues in data mining and other large scale commercial database manipulations

1.4b: Distributed Data Communications



1.5a: Why Use Parallel Computing

- Save time – wall clock time – many processors work together
- Solve larger problems – larger than one processor's CPU and memory can handle
- Provide concurrency – do multiple things at the same time: online access to databases, search engine

Google's 4,000 PC servers are one of the largest clusters in the world

1.5b: Other Reasons for Parallel Computing

- Taking advantages of non-local resources – using computing resources on a wide area network, or even internet
- Cost savings – using multiple “cheap” computing resources instead of a high-end CPU
- Overcoming memory constraints – for large problems, using memories of multiple computers may overcome the memory constraint obstacle

1.6a: Need for Large Scale Modeling

- Weather forecasting
- Ocean modeling
- Oil reservoir simulations
- Car and airplane manufacture
- Semiconductor simulation
- Pollution tracking
- Large commercial databases
- Aerospace (NASA microgravity modeling)

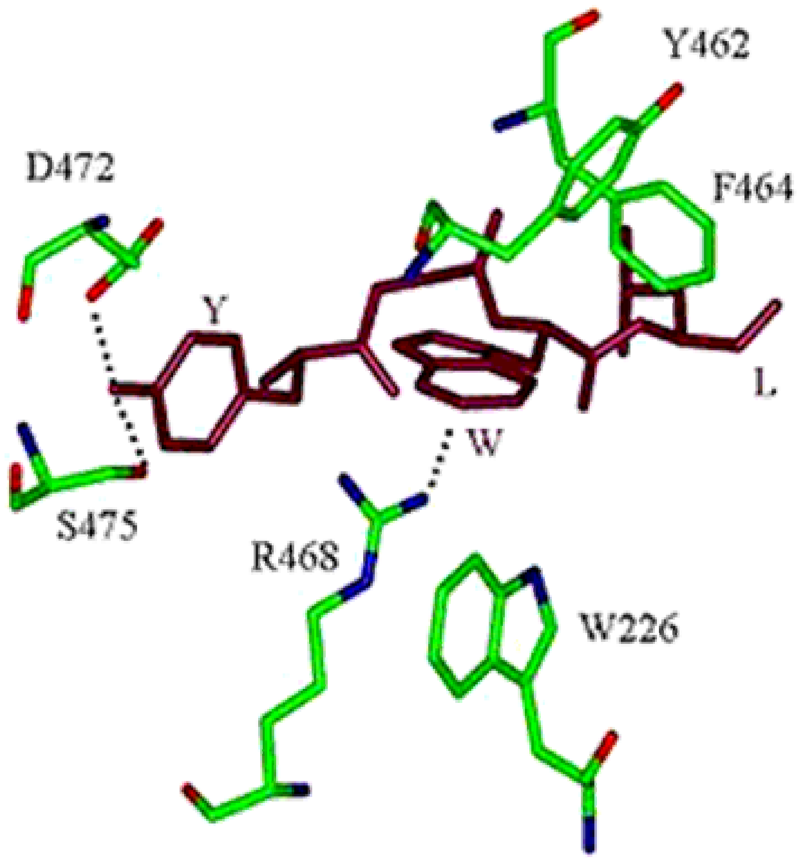
1.6b: Semiconductor Simulation

- Before 1975, an engineer had to make several runs through the fabrication line until a successful device was fabricated
- Device dimensions shrink below 0.1 micro-meter
- A fabrication line costs 1.0 billion dollars to build
- A design must be thoroughly verified before committed to silicon
- A realistic simulation for one process may take days or months to run on a workstation
- Chip price drops quickly after entering the market

1.6c: Drug Design

- Most drugs work by binding to a specific site, called a receptor, on a protein
- A central problem is to find molecules (ligands) with high binding affinity
- Need to accurately and efficiently estimate electrostatic forces in molecular and atomic interactions
- Calculate drug-protein binding energies from quantum mechanics, statistical mechanics and simulation techniques

1.6d: Computing Protein Binding



1.7: Issues in Parallel Computing

- Design of parallel computers
- Design of efficient parallel algorithms
- Methods for evaluating parallel algorithms
- Parallel computer languages
- Parallel programming tools
- Portable parallel programs
- Automatic programming of parallel computers
- Education of parallel computing philosophy