

## Introduction

## Technology Trends

- Building blocks:

	Capacity	Speed
Logic	2x in 3 years	2x in 3 years
DRAM	4x in 3 years	1.4x in 10 years
Disk	2x in 3 years	1.4x in 10 years

- Workstation performance (measured in SpecMarks) improves roughly 50% per year
- Improvement in cost performance estimated at 70% per year

## Processor Perspective

- Putting performance growth in perspective:

	IBM POWER2 Workstation	Cray YMP Supercomputer
Year	1993	1988
MIPS	>200 MIPS	<50 MIPS
Linpack	140 MFLOPS	160 MFLOPS
Cost	\$120,000	\$1M (\$1.6M in 1994\$)
Clock	71.5 MHz	167 MHz
Cache	256 KB	0.25 KB
Memory	512 MB	256 MB

- 1988 supercomputer in 1993 server!

## Hardware Technology

	1980	1990	2000
Memory	64 K	4 M	256 M - 1 G
Speed	1-2	20-40	400-1000 MIPS
Disks	40 M	1 G	20 G
Floppies	.256 M	1.5 M	0.5-2GB
LAN	2-10 Mbits	10 (100) Mb	620Mb (ATM), 1Gb Ethernet
Busses	2-20 Mbytes	40-400	1GB

## Software Technology

	1980	1990	2000
Languages	C, FORTRAN	C++, HPF	Java??
Operating system	proprietary	+DUM	+DUM+NT
User I/F	glass Teletype	WIMP	stylus, voice, audio, video?
Comp styles	T/S, PC	Client/Server	agents, mobile
New things	PC and WA	parallel proc.	net appliances
Capabilities	WP, SS	WP, SS, mail	video, ??

- DUM = DOS, n-Unix's, MAC
- WIMP = Windows, Icons, Mouse, Pull-down menus
- Agents = software robots that work on information

## Computing 2001

- Continue quadrupling memory every 3 years
  - 1K chip in 72 becomes 1 gigabit chip (128 Mbyte) in 2002
- On-line 12-25 gigabytes; \$10 1-Gbyte floppies and CDs
- Processor performance improves at 60% per year; parallelism  $\approx$  100 fold
  - ...
- Radio links for untethered computing

## Application Opportunities

- Integration of video, communication, and computing; desktop video conferencing
- Complex distributed multimedia applications (e.g., VR environments for training)
- Large commercial transactions processing systems over the Internet
- Computational science and engineering applications
- Computer commoditization and eventual disappearance: ubiquitous computing
- ....

## Where Has This Performance Improvement Come From?

- Technology?
- Organization?
- Instruction set architecture?
- Software?
- Some combination of all of the above?

## Advances in Computer Architecture

- Bit level parallelism
- Pipelining
- Caches
- Instruction level parallelism
- Out-of-order executions
- Speculation

## What is Ahead?

- Greater instruction level parallelism?
- Bigger caches?
- Multiple processors per chip?
- Complete systems on a chip?
- High performance LAN, interface, and interconnect
- .... and software innovations!!

## What Is Computer Architecture?

- **Definitions:**
  - Computer architecture defines the attributes of the system as seen by the programmer (i.e., the conceptual structure/organization and functional behavior of the system, which is distinct from logic design and physical implementation)
  - Abstract view of the hardware
- Given the requirements of a computer system, the job of the computer architect is to provide a design that will meet the requirements
- High-level requirements typically include low cost and high performance
- Analogy: An Architect (as contrasted to a plumber)

## Architecture Versus Logic Design

### City metaphor

- Logic design: designing and constructing a building
- Architecture: planning and running a city
  - residential areas
  - industrial areas
  - commercial areas
  - transportation
  - utilities (water, power, ...)

## Architecture, Implementation and Realization

- **Architecture:** functionality of the machine, or interface to software
  - instructions; operand types; logical memory organization ...
- **Implementation:** the logical hardware features that support the architecture
  - serial adder versus parallel adder
  - array multiplier versus shift and add multiplier
- **Realization:**
  - technology (ECL, NMOS, CMOS, ...)
  - packaging (chips, chip carriers, boards, ...)

## Why Computer Architecture? - A Traditional View

- **Uniform architectures define computer families**
  - Machines sharing same architecture
  - but having different implementations
  - and providing the user with a wide range of cost/performance choices
- Major advantage: **software compatibility**
  - The software investment of the client is preserved as new models are introduced
  - The client can upgrade to more powerful models with relative ease

## Why Computer Architecture? - Today's View

- Goals:
  - Maximize the overall performance of the system
  - Bridge the performance gap between the slowest and the fastest component in a computer system
- Architecture design is an iterative process:
  - Searching the space of possible designs
  - Evaluate the performance of the design choice
  - Identify bottlenecks, redesign, and repeat the process
- Applicable at all levels of computer systems