

CoE3DR4

Computer Organization

Chapter 1



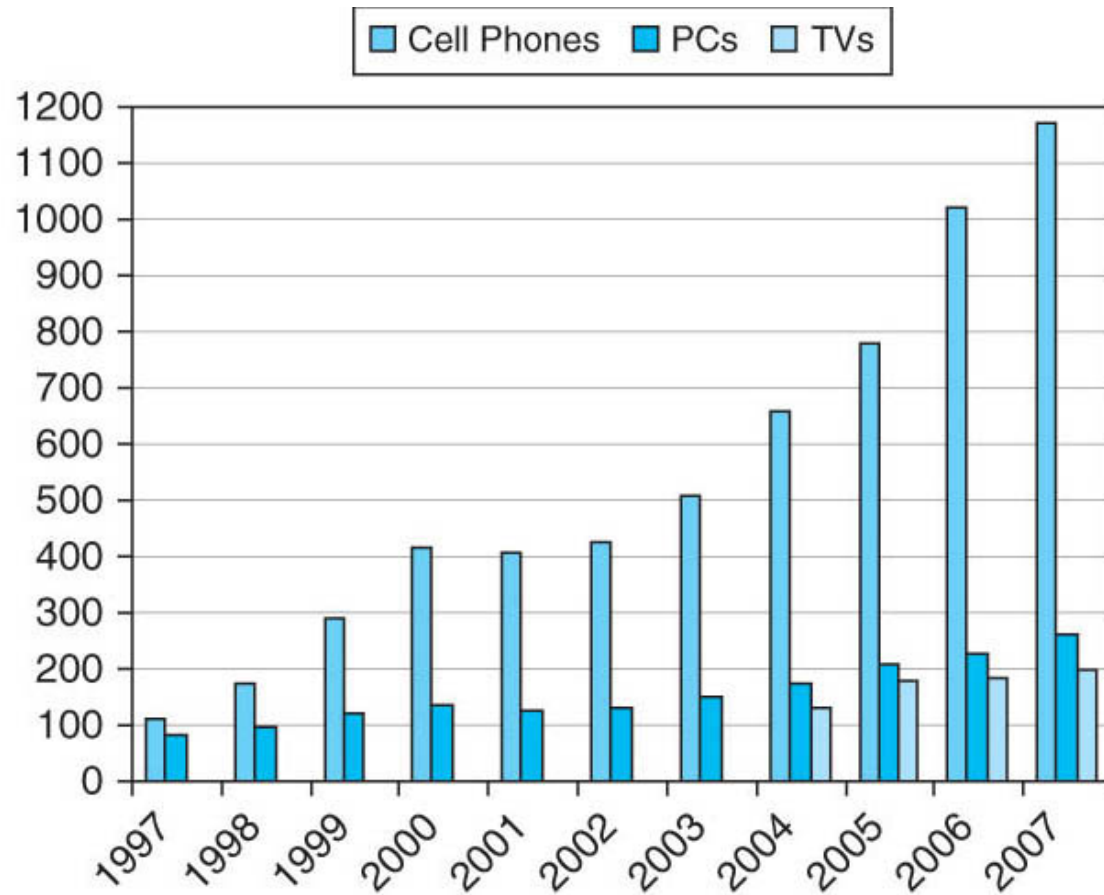
Classes of computers

- A common set of hardware technologies is used in all computers
- Different applications have different design requirements and employ the core hardware technology in different ways
- Computers are used in three different classes of applications:
 1. Desktop computers: designed for use by an individual, usually incorporating a graphic display, keyboard, and mouse.
 - good performance, low cost

Classes of computers

2. Servers: computers used for running larger programs for multiple users often simultaneously and typically accessed only via a network
 - greater emphasis on expandability and dependability,
4. Embedded computers: a computer inside another device used for running one predetermined application or collection of software
 - Examples: processors in washing machine and car
 - Minimum performance with limitations on cost or power

Number of computers sold



Below your program

- A typical application (e.g., a database system) may consist of hundreds of thousands of lines of code and may rely on software libraries
- Hardware in a computer can only execute extremely simple low-level instructions
- Several layers of software (system software) interpret or translate high-level operations into simple computer instructions
- Two types of system software:
 - Operating system
 - Compiler

Below your program

- Operating system: interfaces between a user's program and the hardware and provides a variety of services and supervisory functions such as:
 - handling input and output operations
 - allocating storage and memory
 - providing for sharing the computer among multiple applications using it simultaneously
- Compiler: translates a program written in a high-level language such as C or Java into instruction that the hardware can execute

Below your program

- Instructions are simply binary numbers the computer understands to perform a job
- Binary numbers are used for both instructions and data
- Assembler: a program that translates a symbolic version of instructions into the binary version
- Instead of issuing a command like
3E 01 02
to add two numbers (01 and 02 are locations where these numbers are kept), a programmer could say something like
ADD A, B
- Assembler will create the binary equivalent from the human-like language

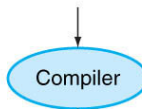
Below your program

- Next step in the evolution ladder: compiler
 - Translate a more English-like (or mathematical notation) language to assembly
 - Allows for increase in productivity of a programmer
 - High-level languages made the programs somewhat machine independent (portable)

Below your program

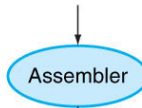
High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly
language
program
(for MIPS)

```
swap:
  muli $2, $5,4
  add $2, $4,$2
  lw $15, 0($2)
  lw $16, 4($2)
  sw $16, 0($2)
  sw $15, 4($2)
  jr $31
```

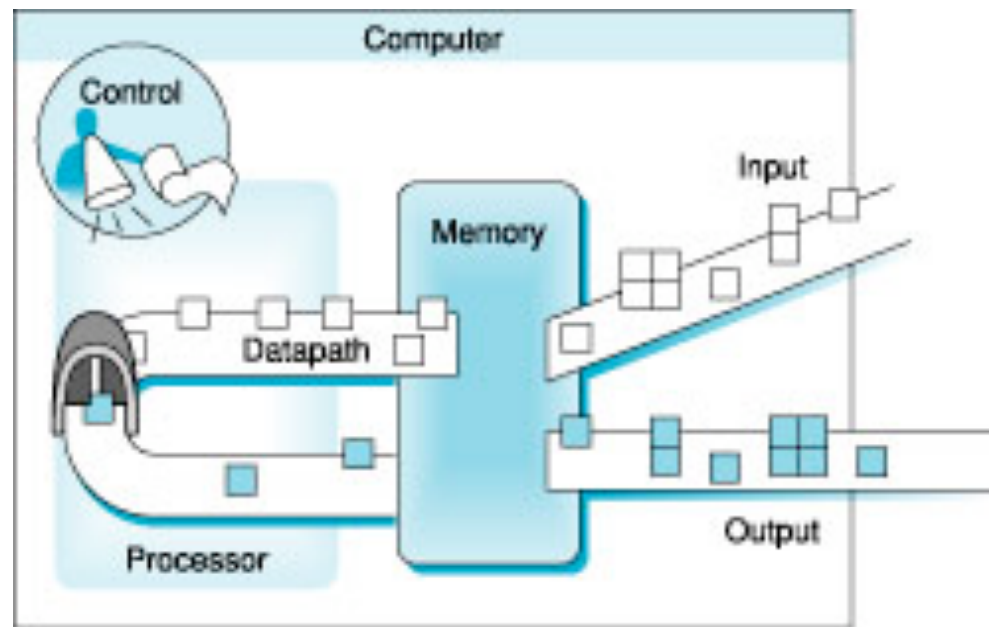


Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
10001100111100100000000000000100
101011001111001000000000000000000
10101100011000100000000000000100
0000001111100000000000000001000
```

Under the covers

- Task of a computer: inputting, outputting, processing and storing data
- The five classic components of a computer are input, output, memory, datapath and control
- Datapath and control are sometimes combined and called the processor



Input devices

- Input devices: keyboard, mouse
- Mouse: A basic pointing device
 - Can be used to determine the position in two dimensions
 - Mechanical vs. Optical mouse
- Mechanical mouse: a ball that when rolled across a surface causes an x and y counter to be incremented
- Optical mouse: an LED to provide lighting, a tiny black and white camera and a simple optical processor
- LED illuminates the surface underneath mouse, camera takes 1500 pictures a second, successive pictures are compared by the optical processor to determine if the mouse has moved

Monitor

- Raster cathode ray tube (CRT): an electron beam scanned across a screen
- Refresh rate of between 30 to 75 time per second
- Image composed of pixels (picture elements) which can be represented by a matrix of bits or bitmap
- Simplest display: 1 bit per pixel (bpp)
- Gray scale display: 256 gray scale values per pixel, or 8 bpp
- Color display: 8 bpp for each of the three primary colors (red, green, blue), giving 24 bpp per pixel
 - A total of 2^{24} colors, or 16 million different colors

Monitor

- Liquid crystal display (LCD)
- Thin, low power display
- Rod shaped molecules in a liquid that bend the light entering the display, possibly from behind the display
- Rods straighten out when a current is applied and no longer bend the light
- Liquid crystal material is between two screens polarized at 90 degrees, the light cannot pass through unless it is bent
- Computer hardware support of the monitor: Bit map stored in a frame buffer or raster refresh buffer
 - The bit pattern per pixel is read out to the display at refresh rate

Opening the box

- Motherboard: A plastic board containing packages of chips, including processors, cache, memory, and connectors for I/O devices such as networks and disks
- Processor or Central Processing Unit (CPU): active part of the motherboard
 - Add numbers, test numbers, signals I/O devices to activate and so on
- Processor comprises two main components: datapath and control
- Datapath: the component of the processor that performs arithmetic operations
- Control: the component of the processor that commands the datapath, memory and I/O devices according to the instructions of the program

Opening the box

- Memory keeps all program code and data while the program undergoes execution
 - Read-only memory (ROM)
 - Random Access Memory (RAM): memory access take the same amount of time no matter what portions of the memory is read
- Hierarchical memories
 - DRAM: Dynamic random access memory
 - Cache: Small fast memory to act as a buffer for DRAM
 - Cash is build using a different memory technology, static random access memory (SRAM)

Opening the box

- Abstraction
 - Hiding lower level details in order to facilitate design of sophisticated systems
- Instruction set architecture (ISA) also called architecture: an abstract interface between the hardware and the lowest level software of a machine
- ISA consists of all the information necessary to write a machine language program that will run correctly including instructions, registers, memory access, I/O, and so on.

A safe place for data

- Memory on motherboard is volatile (retains data only if powered)
 - Sometimes called primary or main memory
 - DRAM: common primary memory
- Memory that stores programs between runs is called secondary memory
- Magnetic disks: common secondary memory
- Organized as a collection of platters, rotating on a spindle at constant speed
- Platters are covered with magnetic recording material
- Movable arm with read/write head
- About 5-20 ms for data access compared to 50-70 ns for DRAMs

Removable storage technologies

- Optical disks including CDs and DVDs
 - discussed in the next slide
- Magnetic tapes
 - slow serial access
 - used mainly for backups
 - replaced recently by duplicate hard drives
- Flash-base removable memory cards
 - nonvolatile semiconductor memory
 - typically connected by a USB connection
- Floppy drives and Zip drives
 - a version of magnetic disk technology with removable disks

CD and DVD

- In a CD data is recorded in a spiral fashion with individual bits being recorded by burning small pits into the disk surface
- Disk is read by shining a laser at the CD surface and examining the reflected light to see if there is a pit
- DVDs use the same approach
- DVDs: smaller pits, multilayer
- Rewritable CDs and DVDs use a different recording surface that has a crystalline reflective material
- Pits are formed that are not reflective in a manner similar to write-once CD or DVD
- To erase the CD or DVD, the surface is heated and cooled slowly to restore the surface to its crystalline structure

Communications

- Advantages of networked computers:
 - Information exchange between computers at high speed
 - Sharing of resources such as disks and I/O devices
 - Accessing machines remotely
- Ethernet
 - High speed network, typically 10 Mb per sec
 - Limited to one kilometer
 - Good for local area networks (LAN)
- WAN
 - Backbone of the internet
 - Based on optical fibers
 - May not be as fast as Ethernet
- Wireless networks
 - Are becoming more common
 - 802.11 standard allow for transmission rates from 1 to less than 100 Mbps

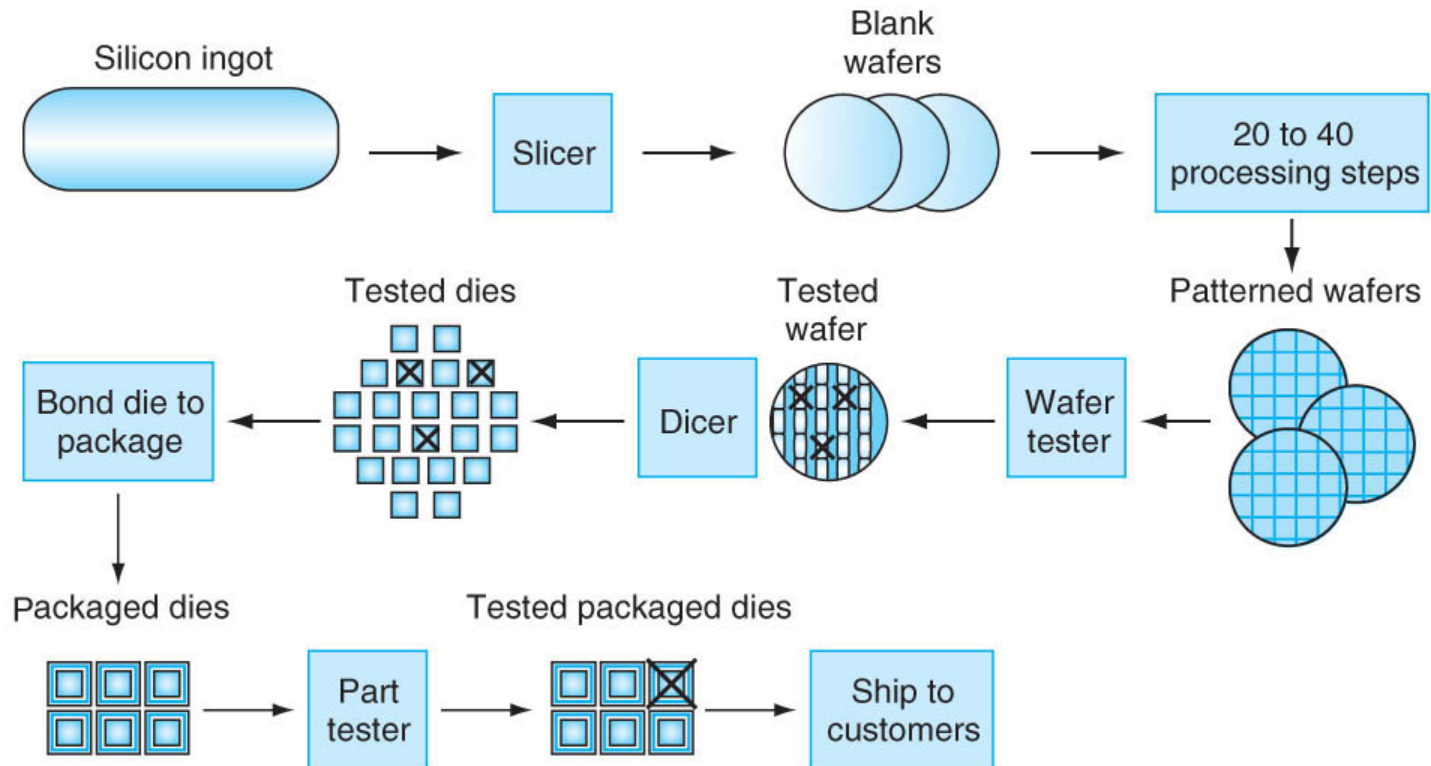
Integrated Circuits

- Transistor: a simple on/off switch
- An IC combines hundreds of transistors on a single chip
- VLSI (Very Large Scale IC) : a device containing hundreds of thousands to millions of transistors
- Silicon
 - Basic building block of ICs
 - Known as semiconductor because it does not conduct electricity very well
 - Chip building requires adding materials to silicon so that it can
 - conduct electricity very well
 - insulate electricity very well
 - conduct or insulate like a switch

Integrated Circuits

- VLSI manufacturing
 - Start with an ingot of silicon crystal, 6-12" in diameter and 12-24" long
 - Ingot is sliced into wafers no more than 0.1" thick
 - Wafers are processed by chemicals to create conductor, insulator, and switch regions
 - Process may result into imperfections, and so, a number of dies are created on each wafer
 - Imperfect dies are discarded, with good dies bonded to wires or I/O pins in a package, called chip
 - Good dies are connected to input/output pins of a package (bonding)

Integrated Circuits



Performance

- Hardware performance is key to the effectiveness of an entire system, including hardware and software
- Performance measurement is one of the most important and difficult problems in computers
- Accurately measuring and comparing performance of different computers is critical to purchasers and designers
- Understanding how to determine the performance impact of hardware factors is crucial to understanding the motivation behind the design of particular aspect of computer

Defining performance

- Different aspects of performance may require different performance metrics
- Response time (Execution time): total time required for the computer to complete a task
 - Response time is dependent on computer load, I/O wait, and OS overhead
- Throughput: the total amount of work done by the computer in a given time
- We will be primarily interested in response time as performance measure
- Performance = 1/Execution time
- For two machines, performance and execution time obey the relation:
Performance A/Performance B=Execution time B/Execution time A

Measuring performance

- A processor may work on several programs simultaneously trying to optimize throughput
- We have to distinguish between the elapsed time and the time processor is working on our behalf
- CPU time (or CPU execution time): the time CPU spends computing for a particular task
- Clock cycles
 - Constant time interval for the clock within the system
 - Dictates how fast a CPU can execute each instruction
- Clock rate
 - Inverse of clock cycle
 - 500 MHz
 - Clock cycle for 500 MHz is 2ns

CPU Performance

- CPU execution time is given by the product of CPU clock cycles for program and clock cycle time
- It can also be measured by $(\text{CPU clock cycles for program}) / \text{Clock rate}$
- Improving performance
 - Current system
 - Execution time: 10 sec
 - Clock: 2 GHz
 - New system
 - Execution time: 6 sec
 - Clock: ?
 - Number of clock cycles: 1.2 times current system

CPU Performance

- Compute the number of clock cycles for current system

CPU time = CPU clock cycles for program/Clock rate

10sec = CPU clock cycles for program/ 2×10^9 cps

CPU clock cycles for program = 20×10^9

- Compute the clock speed for new system

CPU time = CPU clock cycles for program/Clock rate

6sec = $1.2 \times 20 \times 10^9$ /Clock rate

Clock rate = 4 GHz

CPU Performance

- Clock cycles per instruction, or CPI
 - Average number of cycles for all instructions for the program being executed
- CPU clock cycles is given by the product of number of instructions and CPI
- CPI provides one way of comparing two different implementations of the same instruction set architecture (ISA)
- Two implementations of the same ISA: A and B
 - A: clock cycle time 250 ps and CPI 2.0 for some program
 - B: clock cycle time 500 ps and CPI 1.2 for same program
- Identify faster machine?

CPU Performance

- Let total clock cycles for the program on respective machines be c_a and c_b , and number of instructions be I
 - $c_a = I \times 2.0$
 - $c_b = I \times 1.2$
- CPU time = CPU clock cycles \times Clock cycle time
- CPU time a = $I \times 2.0 \times 250 = 500I$ ps
- CPU time b = $I \times 1.2 \times 500 = 600I$ ps
- Machine A is faster; since performance is inversely proportional to time
- The performance gain is given by:
CPU performance A/ CPU performance B = CPU time B/ CPU time A = $600I/500I = 1.2$

CPU Performance

- Basic performance equation:
- CPU time = Instruction count x CPI x Clock cycle time
or
- CPU time = Instruction count x CPI /Clock rate

CPU Performance

- Measuring the performance factors
 - Measure CPU time by actually running the program
 - Clock cycle time is usually available as part of documentation
 - Instruction count and CPI are more difficult to obtain
 - Instruction count can be measured by using profiling tools
 - CPI can be obtained by detailed simulation of an implementation or by combining hardware counters and simulation
 - You may be able to compute CPU clock cycles by looking at different types of instructions and using their individual clock cycle counts

$$\text{CPU clock cycles} = \sum_{i=1}^n CPI_i \times C_i$$

- C_i is the number of instructions of class i
- CPI_i is the average number of cycles per instruction for class i
- n is the number of instruction classes

CPU Performance

- Comparing code segments
- Instruction classes

Instruction class	CPI
A	1
B	2
C	3

- Instruction count for different code sequences

Code	A	B	C
C1	2	1	2
C2	4	1	1

- Find out the number of instructions for each code sequence, the faster code sequence, and CPI for each code sequence

CPU Performance

- Number of instructions in sequence C1 = $2 + 1 + 2 = 5$
- Number of instructions in sequence C2 = $4 + 1 + 1 = 6$
- Obviously, sequence c1 executes fewer instructions
- CPU clock cycles₁ = $(2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10$
- CPU clock cycles₂ = $(4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9$
- Code sequence c2 is faster
- $\text{CPI} = \text{CPU clock cycles} / \text{Instruction count}$
- $\text{CPI}_1 = 10 / 5 = 2$
- $\text{CPI}_2 = 9 / 6 = 1.5$

Evaluating performance

- Workload: a set of programs run on a computer that is either the actual collection of applications run by a user or is constructed from real programs to approximate such mix
- To evaluate two computers, a user would simply compare the execution time of the workload on the two computers
- Most users are not in this situation
- The alternative is to evaluate the computer using a set of benchmarks
- Benchmark: Programs specifically chosen to measure performance

Evaluating performance

- Different classes and applications of computers will require different benchmarks
- For desktop computers the most common benchmarks are either measures of CPU performance or benchmarks focusing on a specific task such as graphic performance
 - Example: SPEC CPU benchmark
- For servers, benchmark depends on the nature of the intended application
- Scientific servers: CPU oriented benchmarks
- Other servers: benchmarks of Web serving, file serving and databases
 - Example: SPECweb99

Evaluating performance

- How to summarize a set of benchmarks?

	Computer A	Computer B
Program 1	1	10
Program 2	1000	100
Total time	1001	110

- Wrong summary can present a confusing picture
- A is 10 times faster than B for program 1
- B is 10 times faster than A for program 2
- Total execution time is a consistent summary measure
- The relative execution times for the same workload is an informative performance summary

Evaluating performance

- Assuming that programs 1 and 2 are executing for the same number of times on computers A and B

Performance B/Performance A=Execution time A/Execution time B
=1001/110=9.1

$$\text{Arithmetic Mean (AM)} = \frac{1}{n} \sum_{i=1}^n \text{Execution_Time}_i$$

$$\text{Weighted Arithmetic Mean (WAM)} = \sum_{i=1}^n w_i \times \text{Execution_Time}_i$$

Where: n is the number of programs executed

w_i is a weighting factor that indicates the frequency of executing program # i

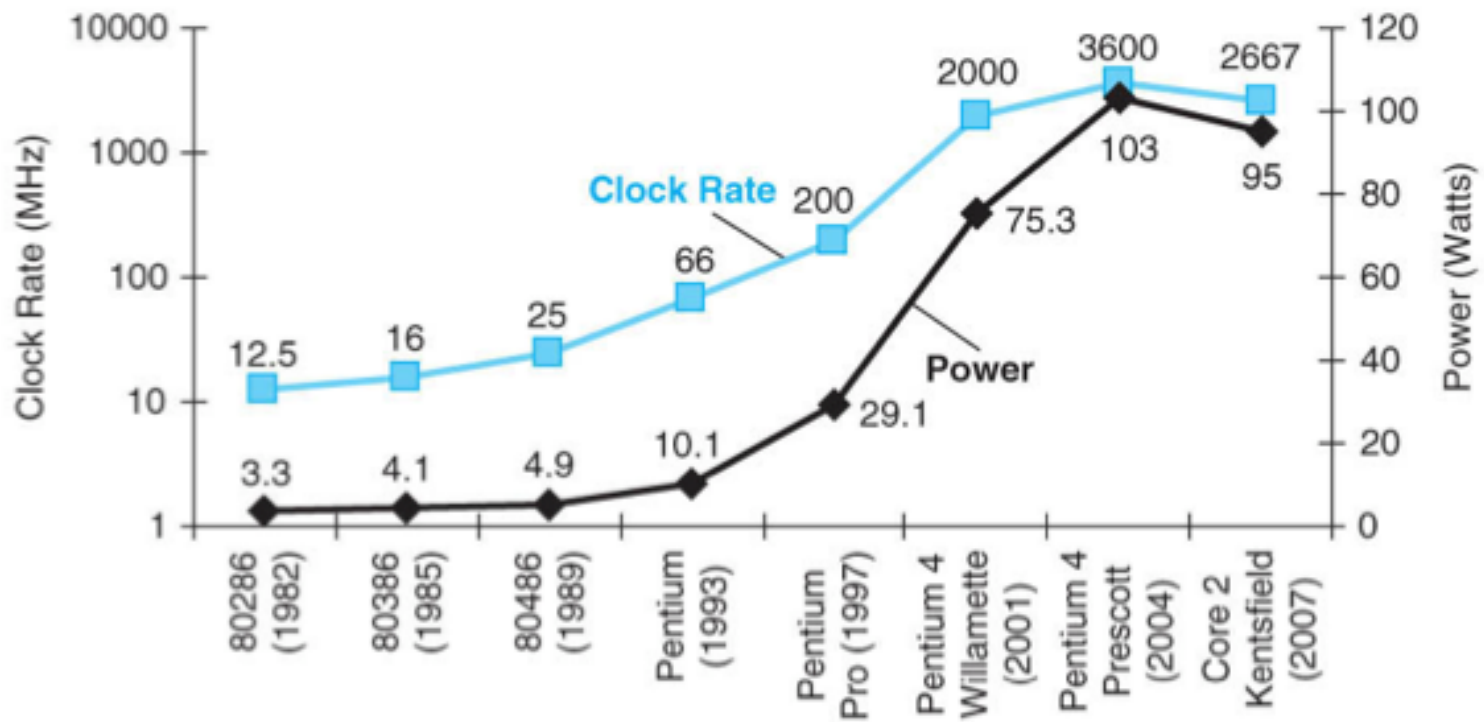
$$\text{with } \sum_{i=1}^n w_i = 1 \quad \text{and} \quad 0 \leq w_i \leq 1$$

Evaluating performance

- Weighted arithmetic means summarize performance while tracking exec. time
- Through the use of weights, a weighted arithmetic mean can adjust for different running times, balancing the contribution of each benchmark in the summary

Power

- Power is a design constraint:
 - Power must be brought in and distributed around the chip which require pins
 - Power is dissipated as heat and must be removed
 - Intel Pentium 4 at 3 GHz burns 82 watts
- What determines the power consumed by an IC?
- Power = Capacitive load x Voltage² x Frequency switched



Multiprocessors

- Power limit has forced a change in the design of microprocessors
- Rather than continuing to decrease the response time of a single program running on a single processor, new microprocessors have multiple processors per chip.
- Programmers have to rewrite their programs to take advantage of multiple processors (parallel hardware)
- Parallel programming is hard:
 - It requires writing performance programs
 - The program must be divided so that each processor has roughly the same amount to do

Multiprocessors

Product	AMD Opteron X4 (Barcelona)	Intel Nehalem	IBM Power 6	Sun Ultra SPARC T2 (Niagara 2)
Cores per chip	4	4	2	8
Clock rate	2.5 GHz	~2.5 GHz?	4.7 GHz	1.4 GHz
Microprocessor power	120 W	~100 W?	~100 W?	94 W