CMPE-310

Lecture-01: Introduction

Outline

Introduction

Course description

Systems - overview, characteristics and Metrics

Differentiating processing units with examples

80x86 Evolution

Introduction

Instructor:
Deepak Krishnankutty

Text:
Barry B. Brey, 'The Intel Microprocessors' Eighth Edition, Pearson/Prentice Hall (2009).

Supplementary text:
Muhammad Ali Mazidi and Janice Gillispie Mazidi, 'The 80x86 IBM PC and Compatible Computers (Volumes I&II), Assembly Language, Design, and Interfacing', Third Edition, Prentice Hall (2000).

Frank Vahid and Tony Givargis, 'Embedded System Design', John Wiley (2002).

Raj Kamal, 'Embedded Systems', McGraw Hill (2008).

Lab Text:
Bob Neveln, 'Linux Assembly Language Programming', Prentice Hall PTR.

Website:
swe.umbc.edu/~deepakk1/cmpe310

Course Description

This course covers:

- Assembly language Programming (Intel x86).
- General system design concepts, devices and support chips
- Specifically covers architecture of the Intel microprocessors.
- Hardware configuration and control of :
  - Common microprocessor support chips, e.g. Interrupt controller, DMA controller.
  - Popular I/O devices, e.g. UART, parallel IO, timers.

Prerequisites:

- Experience with the C programming language.
- Some familiarity with Operating Systems, such as Windows.
- Experience with the Linux operating system.

Projects and Labs:

- Assembly Language Programming
- Hardware Project (Board Design and programming)

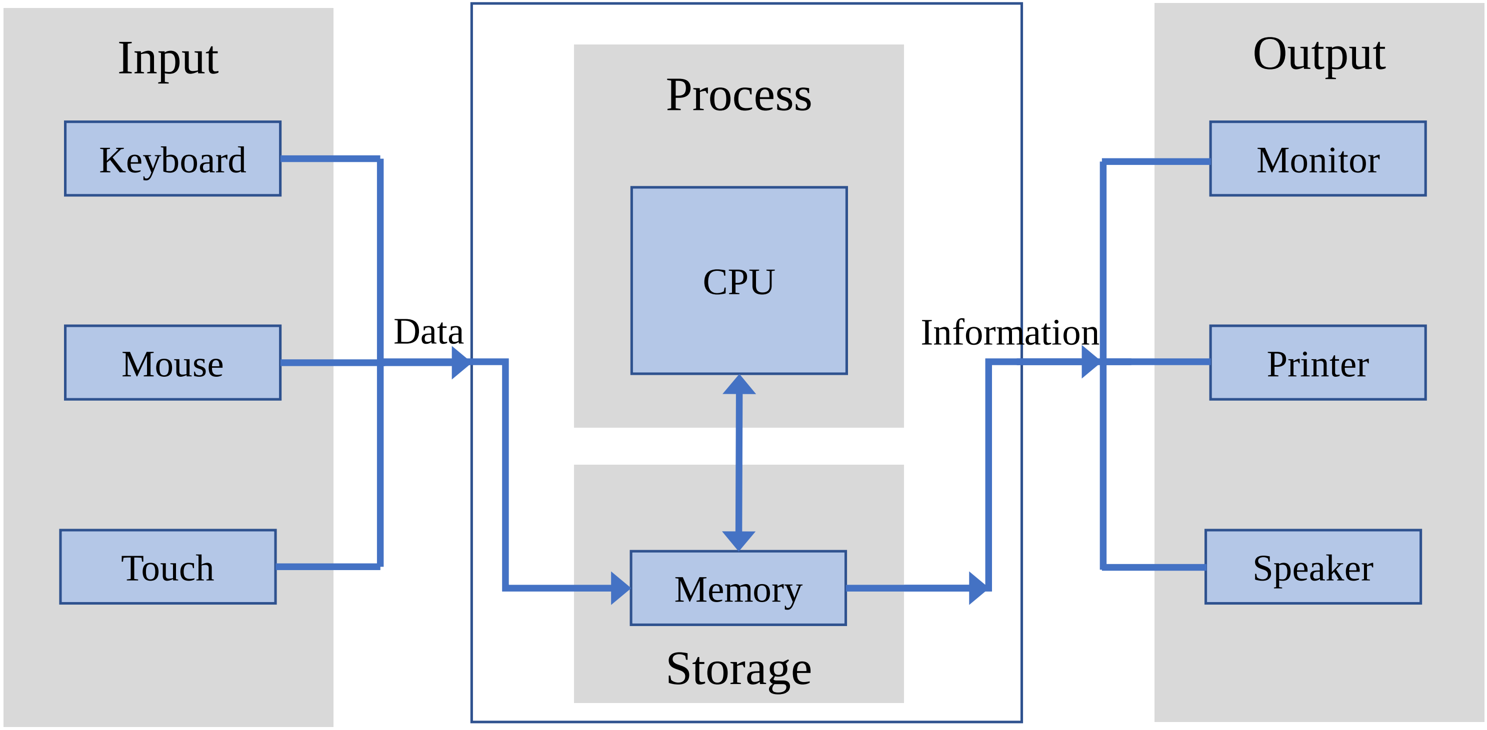
Course Description

Component	Weightage
Midterm	20%
Final	25%
Programming Projects/ Labs/ Homeworks	50%
Class Participation/Quiz/ In-class exercise	5%

Course Description

Component	Weightage
Programming Projects/ Labs/ Homeworks	**50%**
Five Homeworks	35% (7% each)
One final Project	15%

What is a computer?

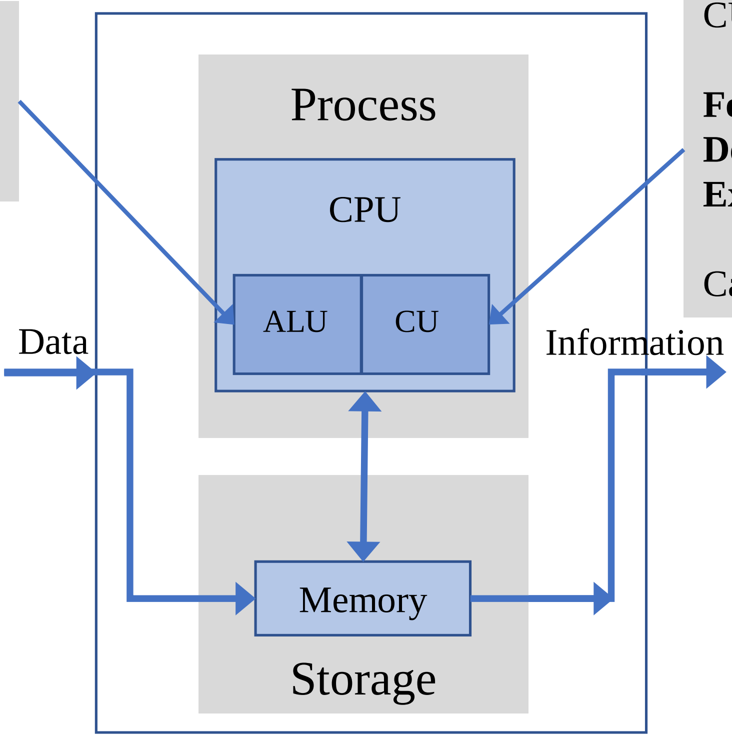


Adapted From Dr Chintan Patel And Avani Dave’s Lecture Notes

What is a computer?

ALU – Arithmetic Logic Unit

Performs Arithmetic and Logic Operations



CU – Control Unit

Fetches Instructions from memory
Decodes And
Executes them

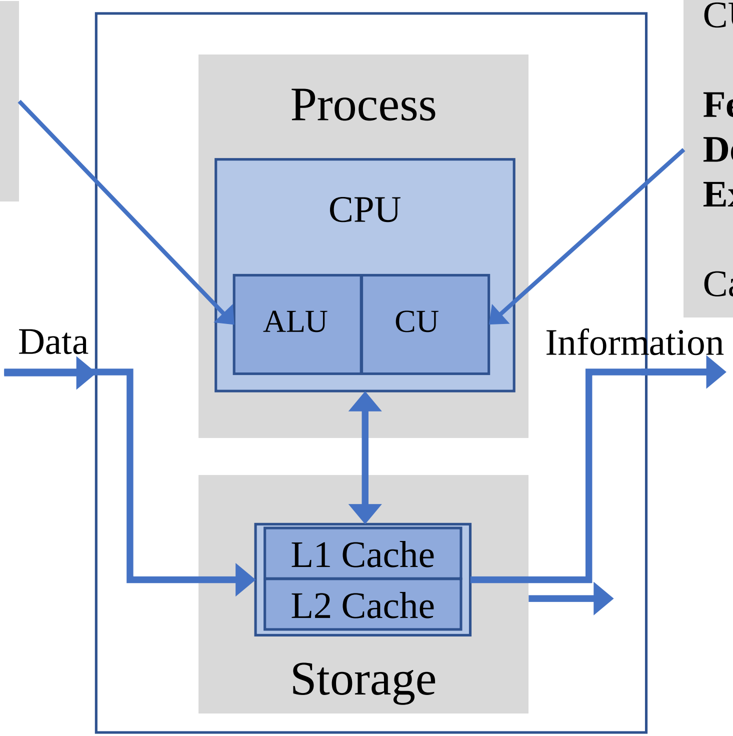
Calls upon ALU when needed

Adapted From Dr Chintan Patel And Avani Dave’s Lecture Notes

What is a computer?

ALU – Arithmetic Logic Unit

Performs Arithmetic and Logic Operations



CU – Control Unit

Fetches Instructions from memory
Decodes And
Executes them

Calls upon ALU when needed

Adapted From Dr Chintan Patel And Avani Dave’s Lecture Notes

Systems Overview

General Purpose Computing Systems

- Personal Computers, laptops, workstations, mainframes and servers

Systems for Dedicated Functions

- Usually embedded within larger electronic devices (referred to as embedded systems)
- Difficult to define exactly as they encompass a wide variety of electronic systems
- Definitions from several authors:
  - Any computing systems other than a general purpose computer.
  - A system consisting of hardware, main application software and an optional real-time operating systems (RTOS).
  - Loosely defined, it is any device that includes a programmable computer but is not itself intended to be a general-purpose computer.
  - Electronic systems that contain a microprocessor or microcontroller, but we do not think of them as computers - the computer is hidden or embedded in the system.
  - It is a system whose principal function is not computational, but which is controlled by a computer embedded within it, ...

  And many more....

System Characteristics

Single-functioned

Usually executes a specific program repeatedly

Exceptions are in cases when a system's program is updated with a newer program

Program can be swapped in and out of the system due to size limitations, depending on the function required at a specific time

Tightly Constrained

Tight constraints for design metrics such as cost, size, performance and power

Reactive and Real Time

Many systems must continually react to changes in system's environment and must compute certain results in real time without delay e.g. car cruise control

Contrast to desktop systems that typically focus on computations with relatively infrequent reactions to input devices (from the computer's perspective).

Delay in computations on desktop systems, while inconvenient to the computer user, typically does not result in a system failure

Systems Design Metrics

- **NRE Cost** (**N**on **R**ecurring **E**ngineering **C**ost)**:** One-time monetary cost of designing the system
- **Unit Cost:** The monetary cost of manufacturing each copy of the system.
- **Size:** The physical space required by the system.
- **Performance:** The execution time of the system.
- **Power:** The amount of power consumed by the system.
  May determine the lifetime of the battery or the cooling requirements.
- **Flexibility:** Ability to change the functionality of the system, software.
- **Time-to-prototype:** The time needed to build a working version of the system.
  May be bigger and more expensive than the final system implementation.
  Used for system verification, debugging and refinements.
- **Time-to-market:** Time required to develop a system to the point than it can be sold.
- **Maintainability:** The ability to modify the system after its initial release, especially by designers who did not originally design the system.
- **Correctness:** Confidence that the system's functionality has been implemented correctly.
- **Safety:** Probability that the system will not cause harm.

Processing Units

- Microprocessor
- Microcontroller
- Single-Purpose Processor
- Digital Signal Processor
- Application Specific Instruction-Set Processors (ASIPs or ASSPs)
- Programmable Logic Devices (PLD)/ Field Programmable Gate Arrays (FPGA)
- Application Specific Integrated Circuits (ASICs)/ System-on-a-chip (SOCs)

Microprocessor

Single VLSI chip that has a CPU and may also have other units (e.g. caches, floating point processing arithmetic unit, pipelining and super-scaling units) that are additionally present and result in faster processing of instructions.

Examples: Intel 8085, Intel x86 processors, Motorola 68HCxxx, Sun Sparc, IBM PowerPC etc.

- System designer need not be concerned about the design of the microprocessor
  - Only needs to understand the architecture related to the programming of the processor’s memory to carry out the required functionality i.e. implement the software.

- Time-to-market and NRE costs are lower when systems are designed with microprocessors as the designer must only write a program. Flexibility is also high.

- Unit cost may be low in small quantities compared with designing a dedicated chip.

- Performance varies by application, unit cost may be high for larger volumes, size and power might be higher due to unnecessary processor hardware.

Microcontroller

A microcontroller is a single chip unit which, though having limited computational capabilities, possesses enhanced input-output capabilities and a number of on-chip functional units.

Examples: Motorola 68HC11xx, HC12xx, HC16xx, Intel 8051, 80251, PIC 16F84,
PIC18, ARM9, ARM7, Atmel AVR etc.

- Particularly suited for use in embedded systems for real-time control applications with on-chip program memory and devices.

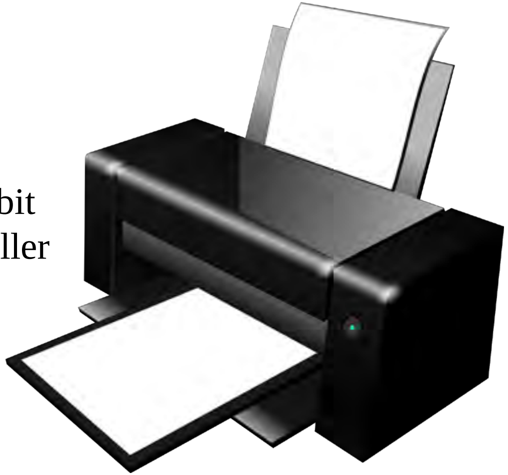
- Common peripherals include serial communication devices, timers, counters, pulse-width modulators, analog-to-digital and digital-to-analog convertors.

- Enables single-chip system implementation and hence smaller and lower-cost products.

Applications of microcontrollers

8051 - 16 bit microcontroller
Source: Wikimedia commons

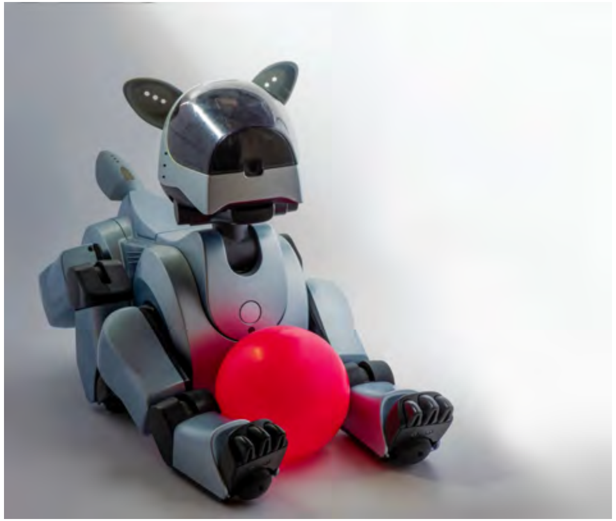
ATM, 32 bit microcontroller



Printer - 16 bit microcontroller

Camera, 32 bit microcontroller

Applications of microprocessors



Robot, 64 bit microprocessor

Source: **Brett Jordan** from **Pexels**

DVD player, 32 bit microprocessor

Source: Wikimedia commons

Smart meter, 32 bit microprocessor

Source: Wikipedia, cc

Single-Purpose Processor

A digital circuit designed to execute exactly one program.

Commonly referred to as coprocessor, accelerator or peripheral.

Examples JPGE codec, Serial-to-Ethernet convertor, etc.

Digital Signal Processor

Essential for systems that require large number of operations on digital signals, which are the digital encoding of analog signals like video and audio.

They carry out common signal processing tasks like signal filtering, transformations or combinations.

Used widely in image processing applications, multimedia, audio, video, HDTV, DSP modem and telecommunication processing systems.

They perform math-intensive operations, including operations like multiply and add or shift and add etc.

Examples: TI TMS320Cxx, Analog Devices SHARC, Motorola 5600xx, etc.

Application Specific Instruction-Set Processors (ASIPs or ASSPs)

A programmable processor optimized for a particular class of applications having common characteristics. Microcontrollers and DSPs can be considered as ASSPs.
ASIPs are available for broad application classes (e.g. graphics processor) as well as very small application classes, some as small as a handful of programs.
Examples: ASSP chip with TCP, UDP, IP, ARP and Ethernet 10/100 MAC logic.

Programmable Logic Devices (PLD)/ Field Programmable Gate Arrays (FPGA)

Contains general purpose logic elements that can be programmed to implement desired functionality, very flexible for implementing custom logic circuits

PLD usually are smaller and contain programmable gates like AND/OR arrays

FPGAs provide lot more functionality and can be used to implement complex designs

FPGAs can have on-chip microprocessors, memory, DSP, communication devices

Examples: Xilinx Virtex, Spartan series FPGAs, Actel, Altera, Lattice, QuickLogic

Application Specific Integrated Circuits (ASICs)/ System-on-a-chip (SOCs)

Custom designed VLSI chips that perform the required function.

Functionality can be integrated using IP (Intellectual property) cores.

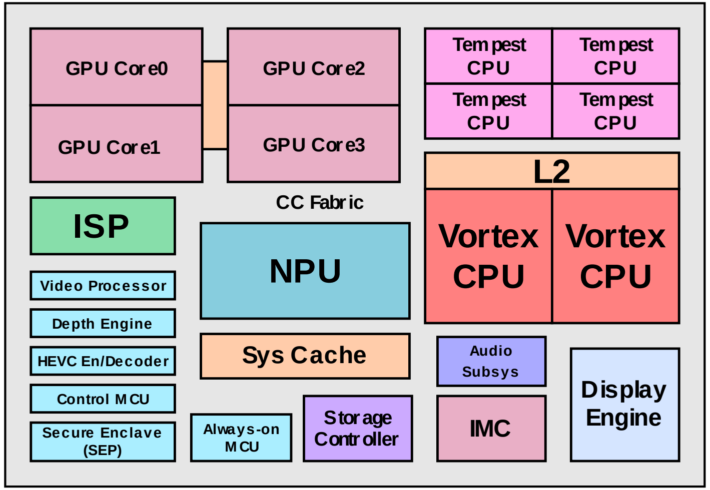
General purpose processors are also available as IP cores and can be integrated on the chip.

Embedded processors are available from ARM, Intel, Texas Instruments and various other vendors.

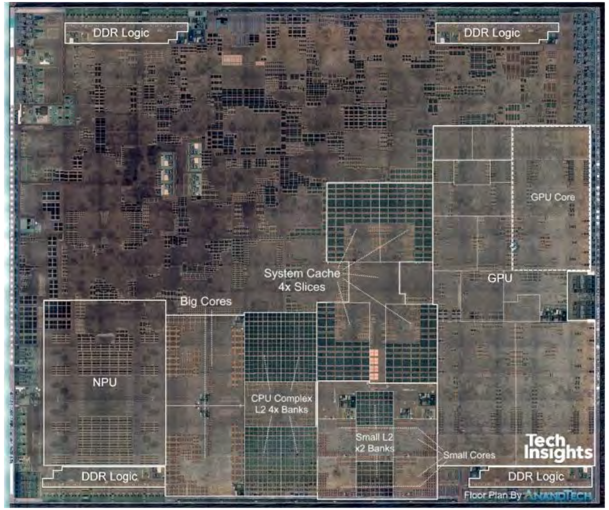
Only feasible for high volume, relatively high cost systems as NRE costs and significant time-to-market.

Examples: Apple A12 Bionic, Qualcomm’s Snapdragon series etc.

A12 SoC



Source: wikichip.org

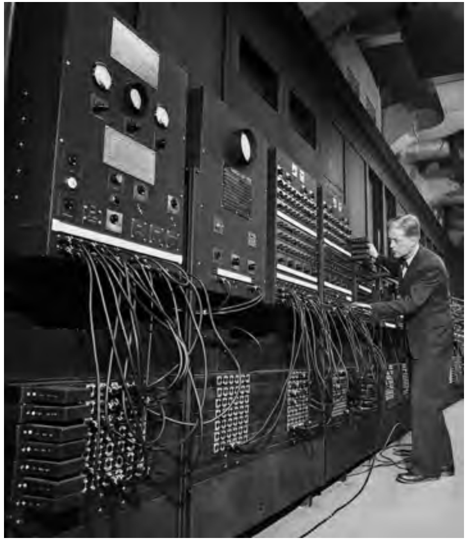


Source: TechInsights with labeling by AnandTech

Current spectrum of systems

Types	Applications
Disposable Computers	Greeting cards, RFID applications
Microcontrollers	Home appliances, video games and smartphones
Personal/Mobile Computers	Desktop or notebook computers, Ipads, Tablets
Servers	Network servers
Mainframe Computers	Batch data processing in a bank

Evolution of Computers



Source: Michael Dunn, Flickr

Source: Wikipedia , cc

1960s

1970s

1980s

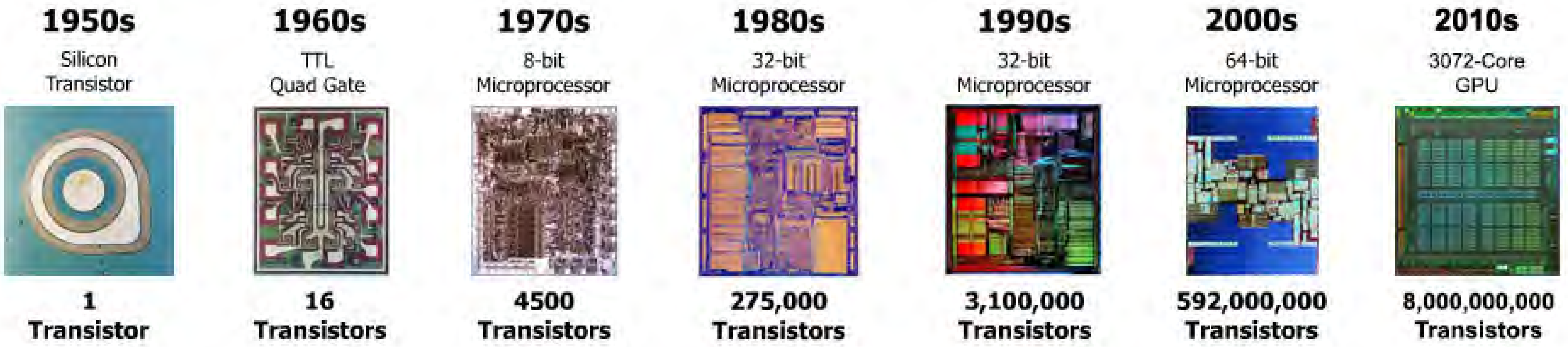
1990s

2000

2010

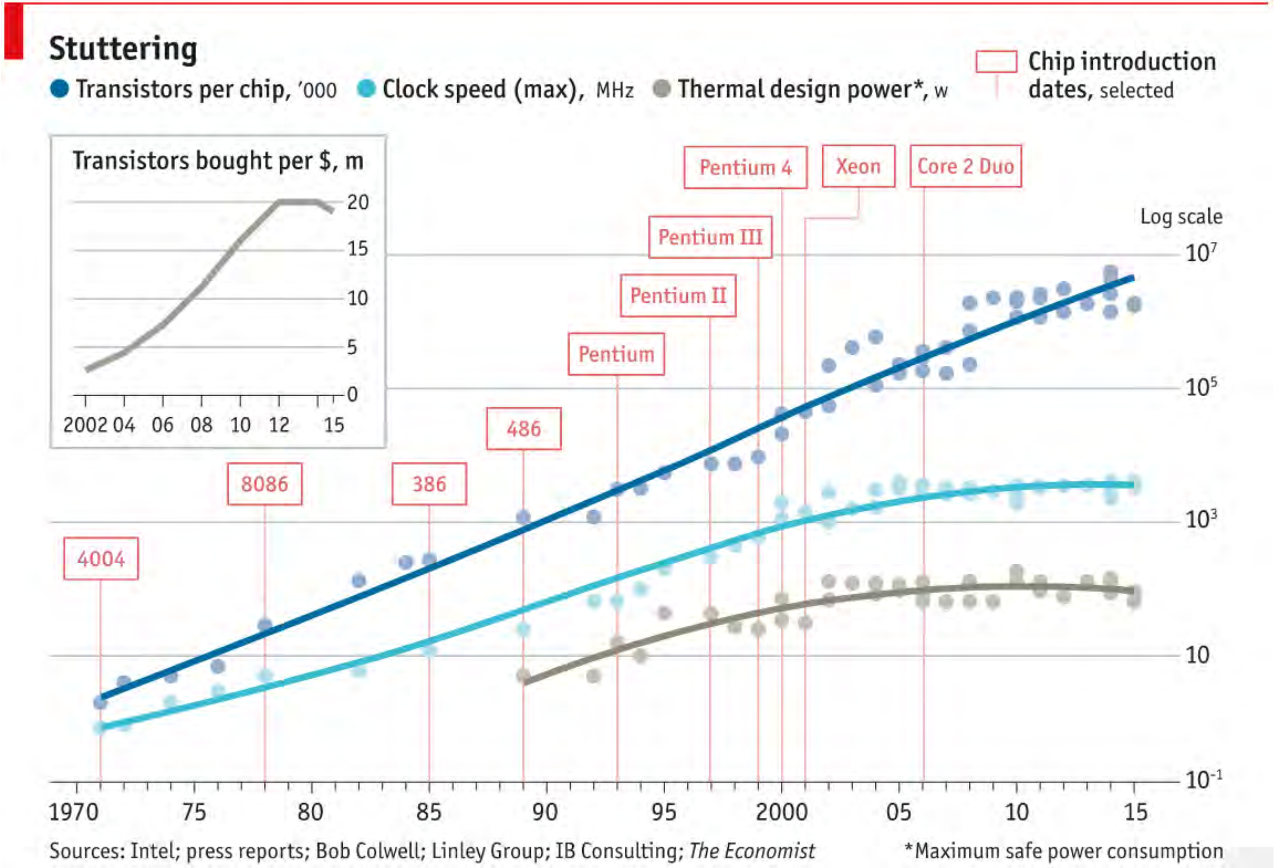
Moore's Law

Transistor density on integrated circuits doubles about every two years. (Gordon Moore 1975)



Source: Computer History Museum - The Silicon Engine

x86 Evolution



Adapted From Dr Chintan Patel And Avani Dave's Lecture Notes

x86 Evolution

Chip	Date	MHz	Transistors	Memory	Notes
4004	Apr-71	0.108	2300	4 KB	First microprocessor on a chip
8008	Apr-72	0.108	3500	16 KB	First 8-bit microprocessor
8080	Apr-74	2	6000	64 KB	First general-purpose CPU on a chip
8086	Jun-78	5–10	29,000	1 MB	First 16-bit CPU on a chip
8088	Jun-79	5–8	29,000	1 MB	Used in IBM PC
80286	Feb-82	8–12	134,000	16 MB	Memory protection present
80386	Oct-85	16–33	275,000	4 GB	First 32-bit CPU
80486	Apr-89	25–100	1.2M	4 GB	Built-in 8-KB cache memory
Pentium	Mar-93	60–233	3.1M	4 GB	Two pipelines; later models had MMX
Pentium Pro	Mar-95	150–200	5.5M	4 GB	Two levels of cache built in
Pentium II	May-97	233–450	7.5M	4 GB	Pentium Pro plus MMX instructions
Pentium III	Feb-99	650–1400	9.5M	4 GB	SSE Instructions for 3D graphics
Pentium 4	Nov-00	1300–3800	42M	4 GB	Hyperthreading; more SSE instructions
Core Duo	Jan-06	1600–3200	152M	2 GB	Dual cores on a single die
Quad Core	Jul-06	1200–3200	410M	64 GB	64-bit quad core architecture
Core i7	Jan-11	1100–3300	1160M	24 GB	Integrated graphics processor

Source: Tanenbaum, Structured Computer Organization, Figure 1-11