# Lecture 1 Inf2C - Computer Systems: Course overview & the big picture

Paul Jackson

School of Informatics

University of Edinburgh

# Practicalities

- Lectures:
  - Tue, Fri, AT LT3 @ 3:00 – 3:50pm
- Tutorials
  - Groups were announced yesterday by email.
  - Start at Week 3
- Exam - 75%
  - In December; exact date not available yet.
- Coursework - 25%, min cw mark: 25/100
  - Two assignments
  - Check schedule web page for deadlines
- All material will be on course web-page: http://www.inf.ed.ac.uk/teaching/courses/inf2c-cs

# Student representatives

- We need 2
  - Please volunteer
- Duties
  - Point of contact for suggestions, complaints of general concern
  - Attend a staff-student liaison meeting to discuss the above
  - If you don't tell us something is wrong we assume it's all going fine
    - You might not want that

# Late coursework

- School-wide consistent policy:
  **Normally, you will not be allowed to submit coursework late**
- If you have a *good reason* to submit late, contact the ITO via their Support Form.
  - The ITO will log the report and pass it on to the UG2 Course/Year Organiser (Colin Stirling)
  - Only in exceptional circumstances (*e.g.*, illness that stopped you getting to email), would an extension be granted after a deadline has passed
- See the UG Second Year Guide for details

# Good reason

Something that, in the judgement of the member of staff responsible, would prevent a competent, well-organised, conscientious student from being able to submit on time. E.g.:

- Significant illness
- Serious personal problems
- Interviews/selection procedures, in some circumstances

Non-examples:

- Difficult cluster of deadlines
- Last-minute computer problems, (your own) back up failure, …

# Syllabus Overview

- Hardware:
  - Data representation and operations
  - Processor organisation & design of simple circuits
  - Exceptions and interrupts
  - The memory sub-system
  - Input/Output
- Software:
  - Low-level programming
  - Operating systems basics
  - Introduction to C programming (2 lectures)

# Learning outcomes

- Demonstrate an understanding of binary representation and basic operations on binary data.
- Demonstrate an understanding of key concepts in computer architecture, including: exceptions, interrupts, virtual memory, processes and pipelined execution.
- Sketch the design of a simple processor and explain how it operates.
- Demonstrate knowledge of I/O devices and the means by which they interface to a processor and its memory system.
- Demonstrate an understanding of the design and operation of important combinational and sequential components within a processor, such as adders, registers, and state machines.
- Demonstrate understanding of an execution pipeline, based on the MIPS architecture.

# Course activities

- Coursework
  1. MIPS (Microprocessor without Interlocked Pipeline Stages) assembly programming
  2. Implement the control unit of a simplified MIPS processor in system-C (To be confirmed)
- Drop-in labs:
  - Demonstrators available to provide help
- Tutorials:
  - Weeks 3,5,7,9
- Notes are provided, but you **must** read the book too

# Inf2C Systems – books

- Patterson & Hennessy: Computer Organization and Design, Morgan Kaufmann
  - 4/e *revised printing* available
  - Library has 2/e & 3/e (both still OK) and 4/e ebook

  Worth buying if this is the only course on computer architecture/hardware you will ever take
- Silberschatz, Galvin, Gagne: Operating Systems Concepts, Willey 8/e (library copies 5/e)
  - Only a few sections needed for this course

# Evolution of computers

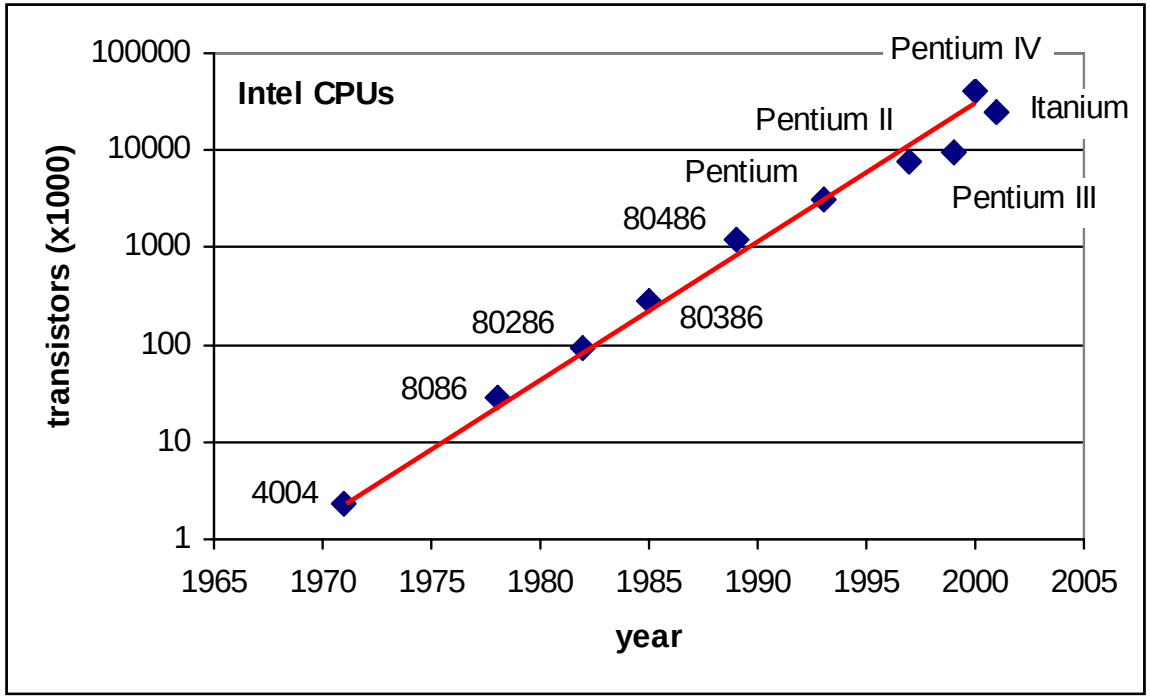
- Early computers had their programs set up by plugging cables and setting switches
- John von Neumann first proposed to store the program in the computer's memory
- All computers since then (~1945) are stored-program machines

# Evolution of computers

- What has changed is the number of transistors (electronic switches) and their speed
- Implementation technology progressed from valves (tubes) to discrete bipolar transistors, MOS transistors, and Integrated Circuits (chips)

- At the same time, the cost per transistor has been dropping

# Moore's Law




- Transistor counts roughly double every 18 to 24 months
  - Intel 4004 (year 1971): 2300 transistors
  - Intel Pentium IV (year 2000): 42,000,000 transistors

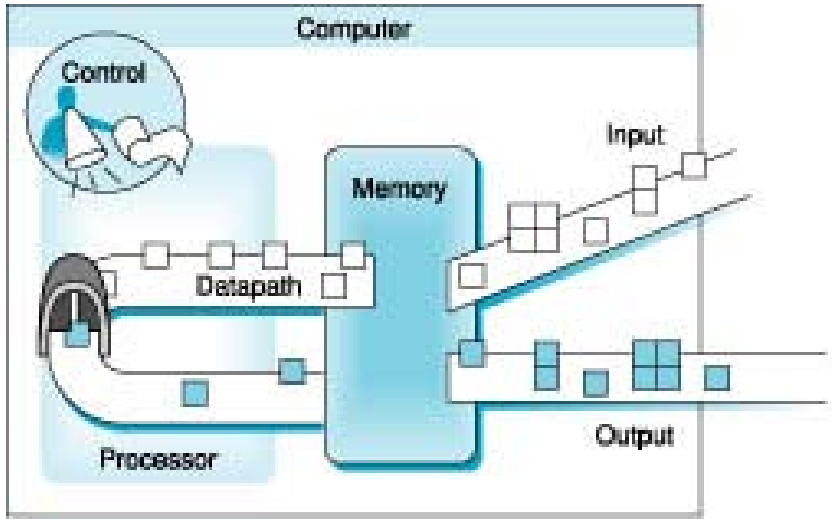
**10,000x increase in 30 years!**

# Computer components

- Data path
  - Performs actual operations on data
- Control path
  - Fetches instructions from program in memory
  - Requests operations on data from data path also in order

} Processor

- Memory
  - Stores data and instructions
- Input/Output
  - Interfaces with other devices for getting/giving data



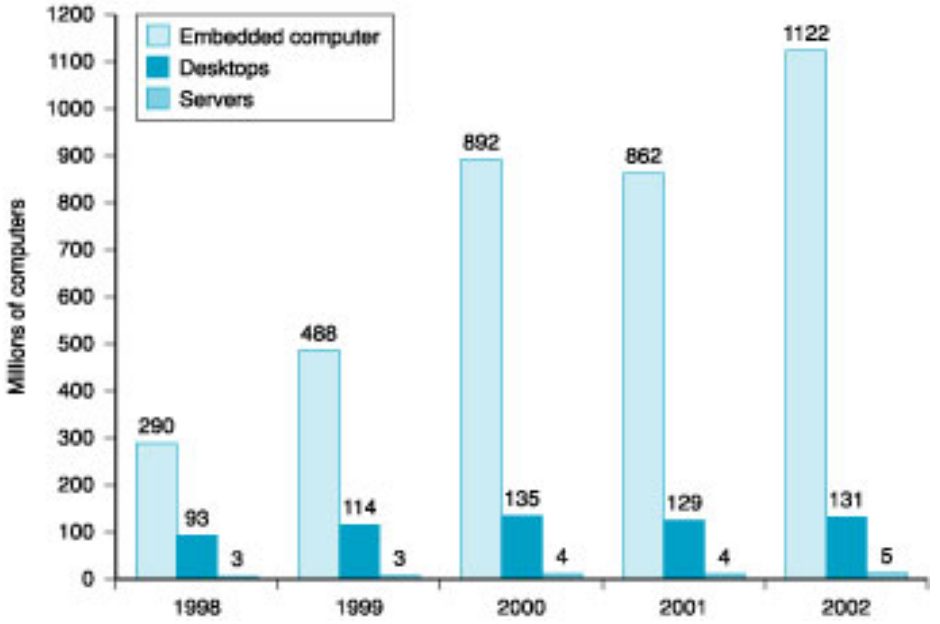

# Types of computer systems

- Servers
  - Fast processor(s), fast I/O
  - Used for either few large tasks (engineering apps), or many small tasks (web server)
  - Multi-user, multi-program
- Desktops
  - The common PC
  - Balance cost, processing power
  - Single/multi-user, multi-program

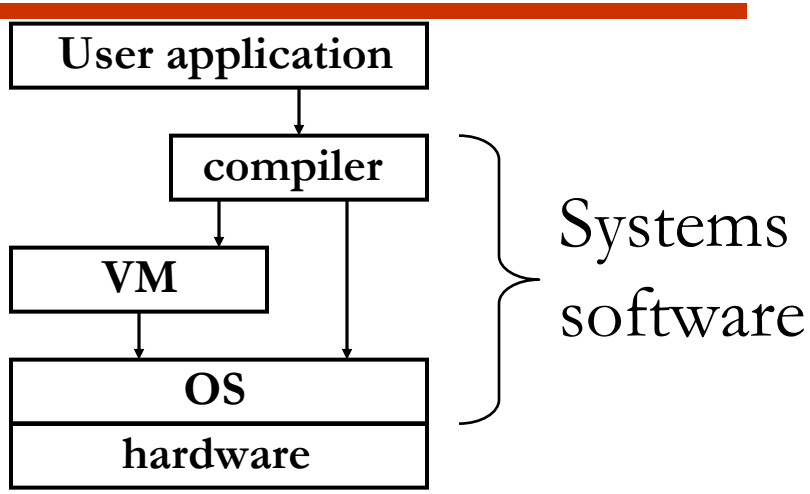
# Types of computer systems - 2

- Embedded:
  - Computing not main purpose of the device
  - Low-cost, low-power (for portable devices)
  - Single user, usually single program, not user programmable
- Which is the largest category?

  Embedded computers



# Modern computer system




- Operating System (OS)
  - Mediates access to hardware resources (CPU, Memory, I/O)
  - Schedules applications
- Compiler
  - Translates **High Level Language** (**HLL**) into **machine language** or **byte code**
- Virtual Machine (VM)
  - Interprets and "executes" byte code