ADVANCED COMPUTER ARCHITECTURE

Marco Ferretti

Tel. Ufficio: 0382 985365

E-mail: marco.ferretti@unipv.it

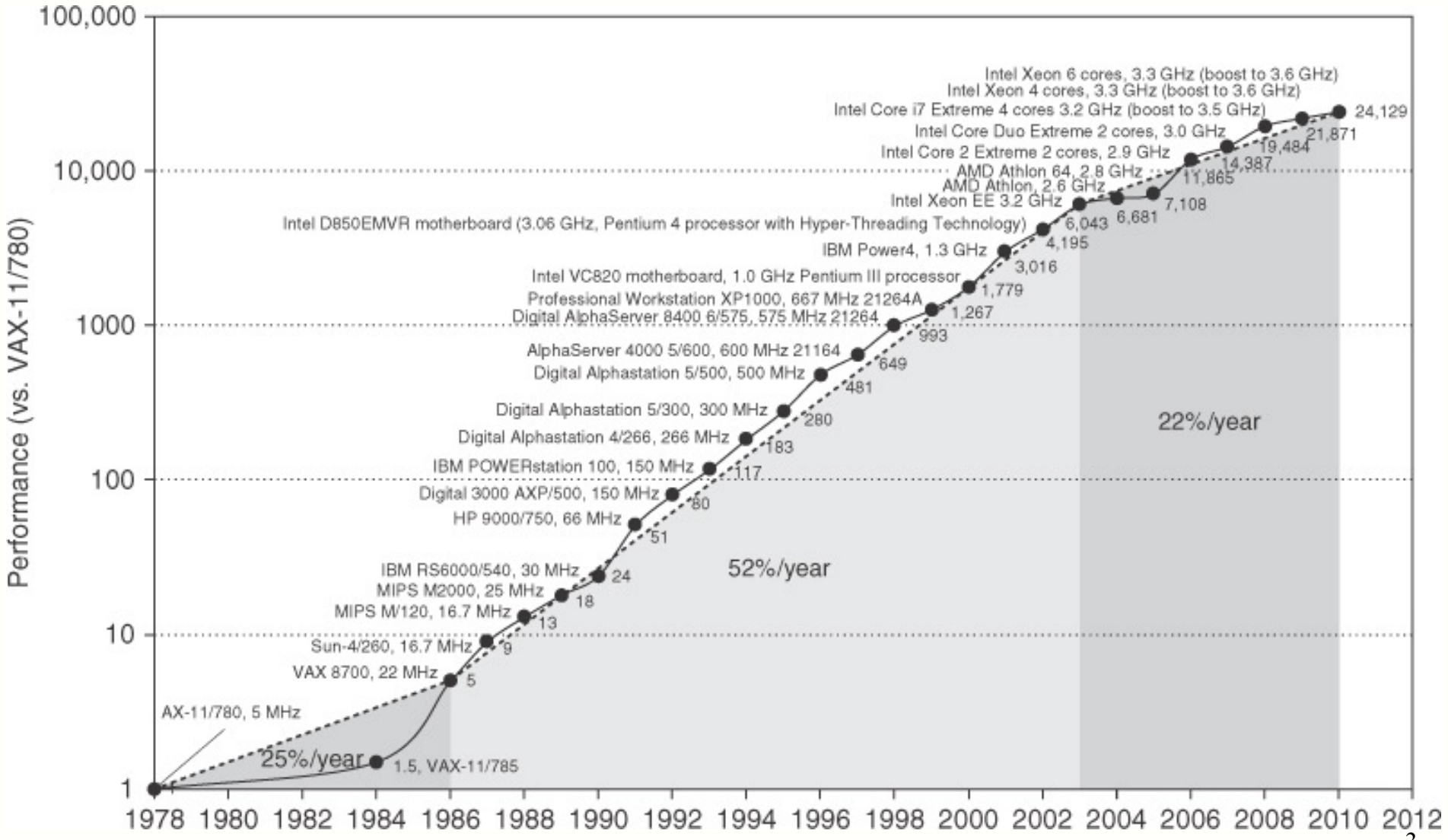
Web: www.unipv.it/mferretti

Course syllabus and motivations

- This course covers the **internal architecture** and operation of modern processors, and the **basics of parallel programming.**
- The increase in performance of processors in the last years is mainly due to three factors:
  - speed-up in instruction execution
    (what is actually the speed of execution, how can it be measured?)
  - increase in the number of instructions executed in parallel
    (given a set of instructions, which are the properties they must satisfy to be executed in parallel)
  - increase in the number of processors that are embedded in the same chip/board and that can be programmed to execute an algorithm

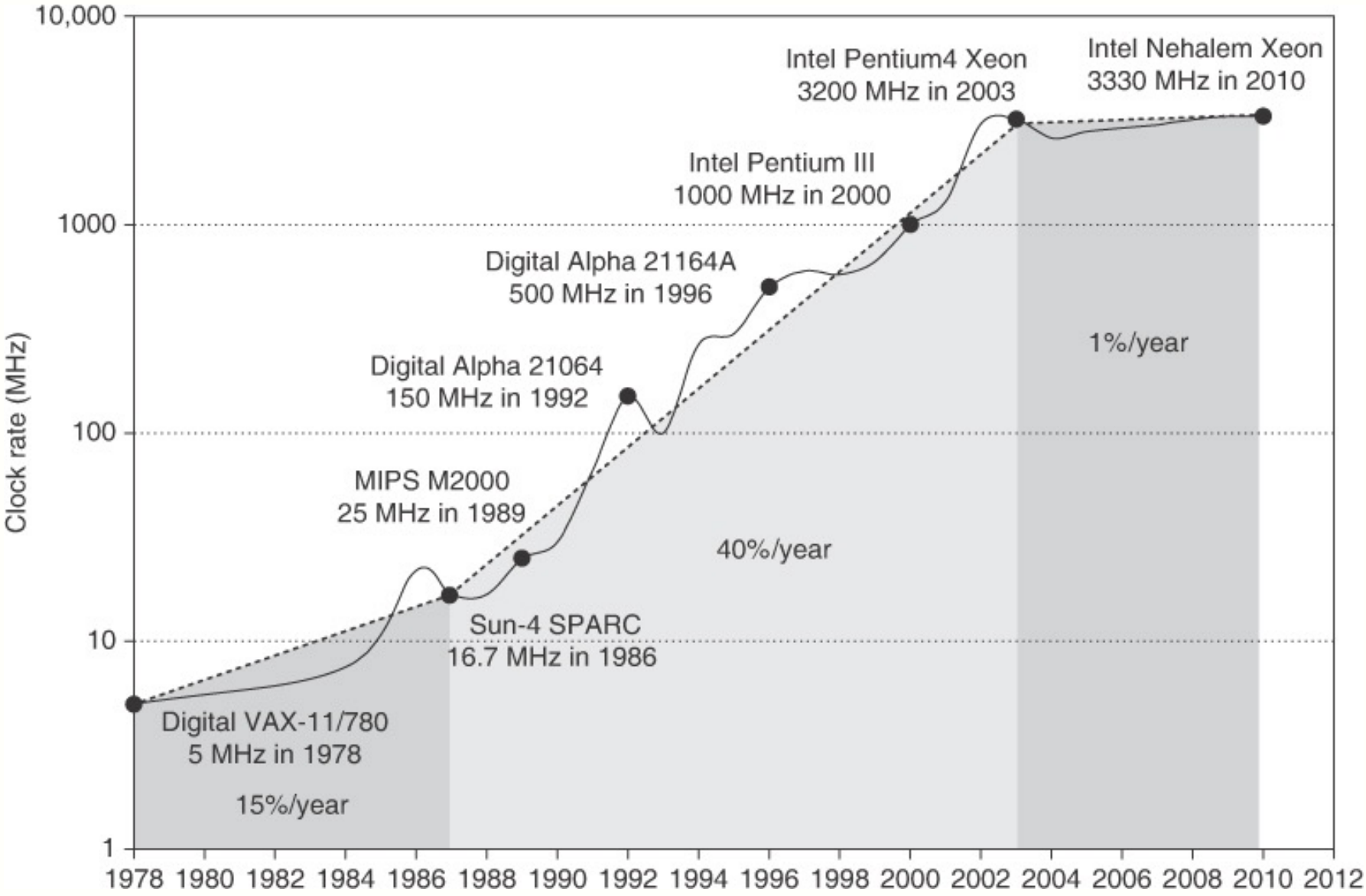
Course syllabus and motivations

- Increase in Cpu performance from 1978 to 2010 (Fig. 1.1 H-P5)



Course syllabus and motivations

- Increase in clock speed 1978 to 2010 (Fig. 1.1 H-P5)



Course syllabus and motivations

- Since 2003 this trend has slowed down consistently (can you guess why?)
- Designers have tried alternative approaches: using multiple execution units. This can be realized in a few ways, also with a mixed approach:
  - Multithreading
  - Multi-core processors, enhanced integration with GPUs
  - Shared-memory multiprocessors systems
  - Multiprocessors with partitioned memory
- (Question: these solutions require a novel approach with respect to a uniprocessor, can you figure out?)

Course syllabus (A)

- **Part I :**
  - Basics in modern architectures
  - PIPELINING basics
  - Instruction Level Parallelism (ILP): the DYNAMIC approach
  - Instruction Level Parallelism: the STATIC approach
  - Fundamentals of CACHING
- **Part II:**
  - Multi-threading
  - Shared memory multiprocessors
  - Partitioned memory multiprocessors
  - Multimedia SIMD extensions, data parallel support, GPGPU.
- **Part III:**
  - parallel programming with MPI (laboratory)
  - cloud computing: short assignement

Course syllabus schedule (B)

- **Part I :**
  - Basics in modern architectures
  - PIPELINING basics
  - Fundamentals of CACHING
- **Part II:**
  - Multi-threading
  - Distributed memory multiprocessors (intro)
- **Part III:**
  - parallel programming with MPI (laboratory)
  - cloud computing: short assignement

Course syllabus schedule (B)

- **Part I :**
  - Instruction Level Parallelism (ILP): the DYNAMIC approach
  - Instruction Level Parallelism: the STATIC approach
- **Part II:**
  - Shared memory multiprocessors
  - Partitioned memory multiprocessors
  - Multimedia SIMD extensions, data parallel support.
  - Tensor flow ISA & processors & GPGPU

Books

- **J. L. Hennessy & D. A. Patterson:** Computer Architecture: A Quantitative Approach, Elsevier - Morgan Kaufmann.
  - 3rd ed. 2003: in the charts as "Hennessy-Patterson"
  - 4th ed. 2007: in the charts as "H-P4"
  - **5° ed. 2011: in the charts as "H-P5"**
- **D. A. Patterson & J. L. Hennessy :** *Computer Organization and Design, The Hardware/Softwae Interface, third edition, 2005, Elsevier.*
  - in the charts as "Patterson-Hennessy"
  - **A. S. Tanenbaum:** *Structured Computer Organization*, 5th ed. Pearson, 2005.

Books – Italian editions

- **J. L. Hennessy & D. A. Patterson:** *Architettura degli elaboratori* Apogeo, 2008 (quarta edizione americana).
  - 4th ed. 2007: in the charts as "H-P4"
- **D. A. Patterson & J. L. Hennessy :** *Struttura e Progetto dei Calcolatori: l'interfaccia Hardware-software, seconda edizione* (terza edizione americana) Zanichelli, 2006.
- **D. A. Patterson & J. L. Hennessy :** *Struttura e Progetto dei Calcolatori: terza edizione* (quarta edizione americana) Zanichelli, 2010.
  - **A. S. Tanenbaum:** *Architettura dei Calcolatori, un approccio strutturale*, 5t ed. Pearson/Prenctice Hall, 2006.

Further material

- Most recent architectures are not properly described in textbooks. The course uses papers, benchmarks and graphical material drawn from web resources, mainly from
- **INTEL**: http://www.intel.com/technology/ITJ/
- **Wikipedia**: en.wikipedia.org
- **Anandtech**: www.anandtech.com
- **Tom's Hardware**: www.tomshardware.com
- **Hardware Upgrade**: www.hwupgrade.it

Lessons charts

- Download charts used during class from the ACA section of the professor's web site
- Use the charts as a trace of the arguments
- Textbook are the course reference material; detailed instructions on which texbooks and which chapters cover the exam will be published in due course on the web site.

Course entry requirements

- Basic knowledge of the “Von-neuman” CPU and of virtual memory
- Basic knowledge of operating systems (processes, thread, mutual exclusion, synchronization)
- Elementary knowledge of assembly language
- Knowledge of the C-language

Testing and exams

- Parallel programming project
  - assigned to groups of 2/3 persons
  - Developed on personal PC & cloud resources
- Written test
  - individual
- Oral test
  - individual and discussion of group project

Cloud Resources

- Google Cloud Platform
- Free credits ($50 each student) will be made available to develop a parallel project
- Instructions on the redemption of these credits will be sent as soon as they are received by the professor
- The cloud deployment is mandatory for the exam, students that fail to redeem in due time the free credits will have to get access to the Google platform on their own

Course web site

www.unipv.it/mferretti

- section on ACA to be enlarged with learning material in English
- Past exams and exercises: texts available also in the Italian section

Course outline

The architecture from the programmer's view point

10000x10000 array, Intel Core 2 Duo @ 2.8 Ghz

```
int sum1(int** m, int n) {
  int i,j,sum=0;
   for (i=0; i<n;i++)
     for (j=0; j<n; j++)
       sum += m[i][j];
  return sum;
}
```

0.4 seconds

```
int sum2(int** m, int n) {
  int i,j,sum=0;
   for (i=0; i<n;i++)
     for (j=0; j<n; j++)
       sum += m[j][i];
  return sum;
}
```

1,7 seconds
(4.2 times slower !!)

Course outline

- Understanding the operation of modern architectures requires a little bit of analysis of their evolution
- All modern processor have roughly the same operation mode, and quite similar internal architecture, the so-called **RISC architecture**
- This denomination is almost forgot, it was very much in use in the 80's, it marked the distinction between the "old" **CISC architectures** and the "new" ones, RISC, that allowed to reach higher performances.

Course outline

- RISC architecture are simpler (and nicer) in concept, which allowed to deploy some fundamentals techniques to speed-up program execution:
  - pipelining (partially overlaps instruction execution)
  - parallel execution of independent instructions
  - branch prediction (knowing in advance which instruction to fetch)
  - speculation (instruction executed before knowing if they belong to the correct path, with possible “undoing”)
  - compile-time instruction re-ordering (static schedulation) to maximize processor utilization.

Course outline

- These techniques are used in all modern architectures, though their actual implementation differ from processor to processor.
- After the major “CISC to RISC” step, performance increase has been due mainly to technologies, rather than to new concepts (To be debated).
- A single core in a dual/quad core Intel CPU is still very similar to an “old” Pentium I

Course outline

- Fabrication improvements however have almost stopped from 2003 onwards, so that uniprocessor performances have had no sensible increase any longer.
- The requirement of more and more computational power has led to alternative approaches.
- The basic idea was coupling two or more processors (“core”) in a multi-core architecture: modern dual and quad-core processors are small shared (and cache) multiprocessor systems.

Course outline

- Two or even more programs can be executed in parallel on the same multi-core processor (provided software can actually exploit available cores)
- Nothing new! this is the architecture of shared-memory multiprocessor systems (that featured private caches, yet)
- And even before multi-core processors, multi-threading tried already to carry out parallele execution of instructions belonging to different threads.

Course outline

- If even more computational power is required, it is unfeasible using processors that share the same main memory.
- It is mandatory using partitioned-memory multiprocessors, even thousands of them, each possibly a multi-core.
- Moder multiprocessors featuring the highest performances can be a synthesis of different architectural solutions: hundreds or thousands of computational units, linked with an ad-hoc network, each computational unit hosting multiple cores that share main memory, each core itself capable of multi-threading.

Course outline

- Old “alternative” architectures, such as vector processors, have almost completely disappeared.
- They left as their heritage multi-media SIMD instructions currently embedded in all processors, and in graphics processors (GPU).
- Currently, the development environments support more and more GPU, even in general-purpose, non-graphicals applications.