Computer Architecture

Courtsy L. N. Bhuyan

What is *Computer Architecture*

Computer Architecture =

Instruction Set Architecture (Opcodes, addressing modes, architected registers, IEEE floating point) +

Organization (High Level aspects of computer's design i.e Memory System,bus structure ,internal CPU where arithmetic,logic ,branching and data transfers are implemented)+

Hardware (Logic designs,Packaging technology,Clock rate,Supply voltage) + …

Ref pp4 Patterson edition 2nd

Instruction Set

1) format

- length, encoding

2) operations

- operations, data types, number & kind of operands

Instruction Set

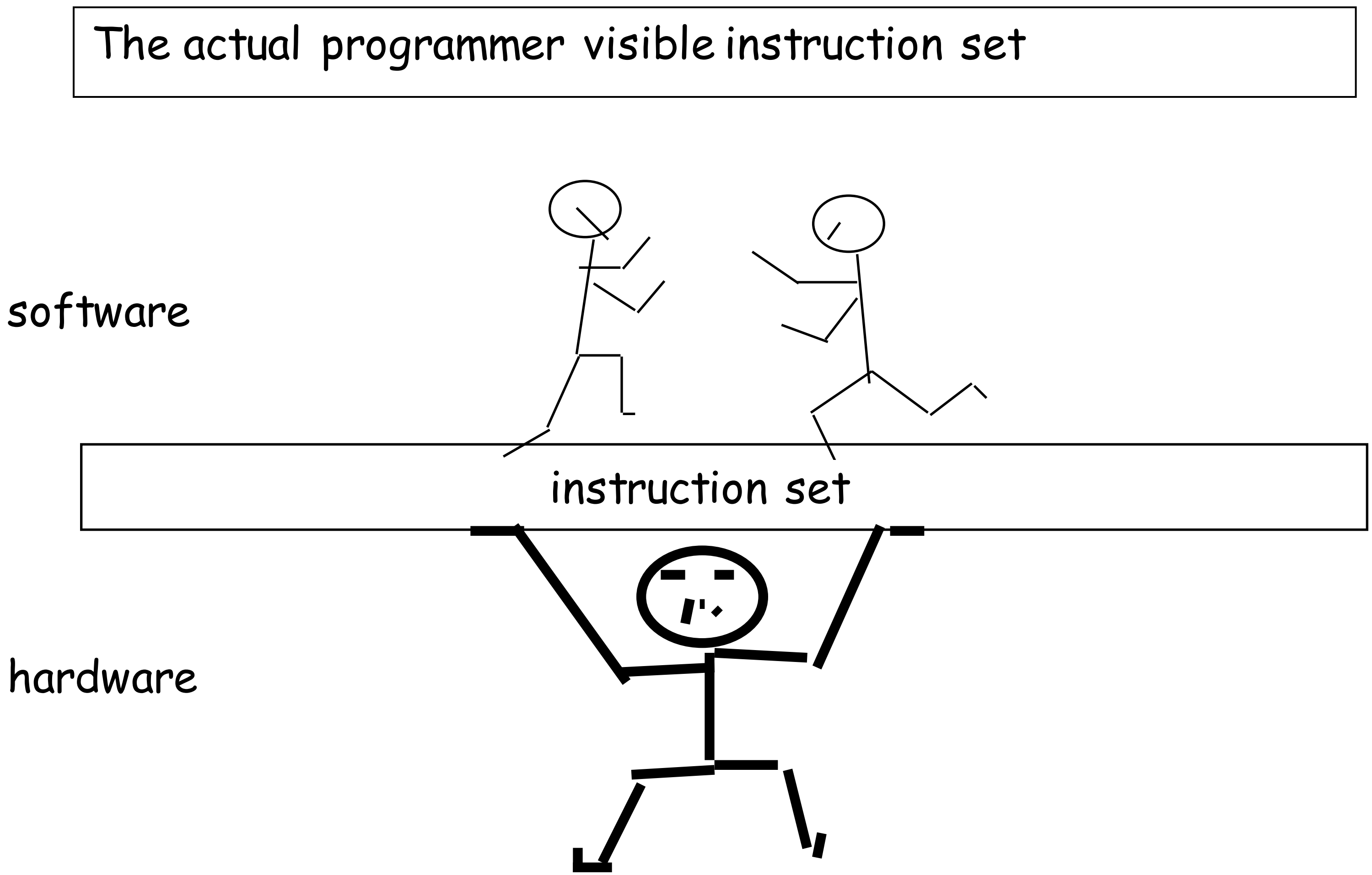
3) storage(Address modes)

- internal: accumulator, stack, general-purpose register
- memory: address size, addressing modes, alignments

4) control

- branch conditions, special support for procedures, predication

The Instruction Set: a Critical Interface



Instruction-Set Processor Design

- Architecture (ISA) programmer/compiler view
  - "functional appearance to its immediate user/system programmer"
  - **Opcodes, addressing modes, architected registers, IEEE floating point**
- Implementation (µarchitecture) processor designer/view
  - "logical structure or organization that performs the architecture"
  - **Pipelining, functional units, caches, physical registers**
- Realization (chip) chip/system designer view
  - "physical structure that embodies the implementation"
  - **Gates, cells, transistors, wires**

Hardware

- Machine specifics:
  - Feature size (10 microns in 1971 to 0.18 microns in 2001)
    - Minimum size of a transistor or a wire in either the x or y dimension
  - Logic designs
  - Packaging technology
  - Clock rate
  - Supply voltage

  …

Relationship Between the Three Aspects

- Processors having identical ISA may be very different in organization.
  - e.g. NEC VR 5432 and NEC VR 4122
- Processors with identical ISA and nearly identical organization are still not nearly identical.
  - e.g. Pentium II and Celeron are nearly identical but differ at clock rates and memory systems

➢ Architecture covers all three aspects.

Why Study Computer Architecture

- Aren’t they fast enough already?
  - Are they?
  - Fast enough to do everything we will EVER want?
    - AI, protein sequencing, graphics
  - Is speed the only goal?
    - Power: heat dissipation + battery life
    - Cost
    - Reliability
    - Etc.

Answer #1: requirements are always changing

Example of Changing Designs

- Having, or not having caches
  - 1970: 10K transistors on a single chip, DRAM faster than logic $\rightarrow$ having a cache is bad
  - 1990: 1M transistors, logic is faster than DRAM $\rightarrow$ having a cache is good
  - 2000: 600M transistors -> multiple level caches and multiple CPUs
  - Will caches ever be a bad idea again?

Example of Changing Designs

- During 1970s, Mainframes and minicomputers dominated the industry
- Performance improved 25% to 30% per year due to improved
  - architectures and technological developments
- Late 1970s –microprocessor came (Improvement in ICTechnology) which led to higher rate of improvement – roughly 35% growth per year in performance

Design Changes contd..

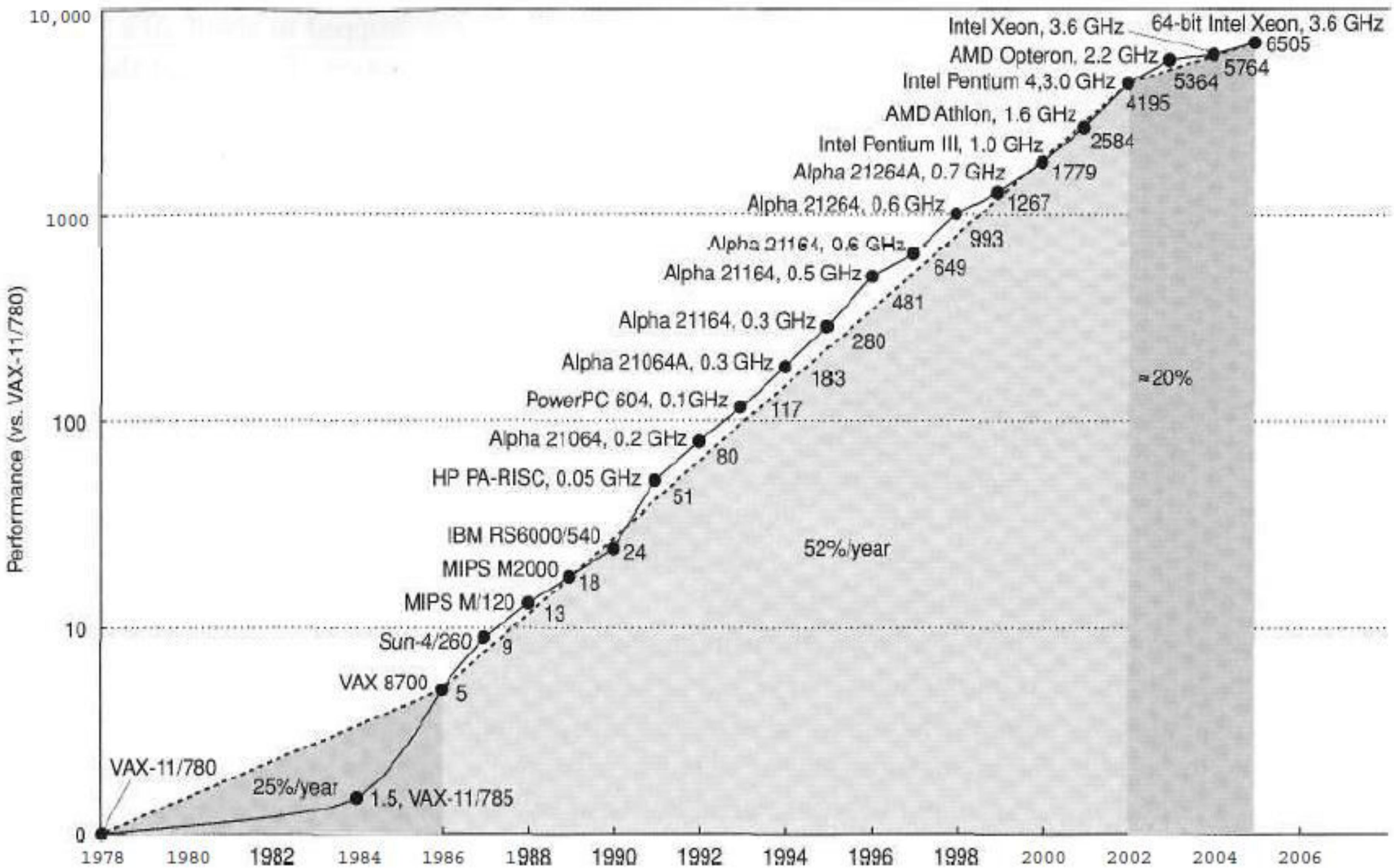
- Microprocessor based computers became Commercially successful due to
  - Mass production of microprocessors
  - Portable object-code creation techniques (cross compilers:-each new architecture required time consuming assembly language programming previously)
  - Creation of standardized vendor-independent operating systems, such as UNIX and its clone, Linux

Design Changes contd..

- In the early 1980s, development of a new set of architectures
  - RISC (Reduced Instruction Set Computer)
    - RISC-based machines focused critical performance techniques
      - Exploitation of instruction level parallelism (pipelining, multiple instruction issue)
      - Use of caches

Performance Growth in Perspective

- Same absolute increase in computing power
  - Big Bang – 2001
  - 2001 – 2003
- 1971 – 2001: performance improved 35,000X!!!
  - What if cars or planes improved at this rate?



Uniprocessor

Since 2002, the limits of power, available instruction-level parallelism, and long memory latency have slowed uniprocessor performance recently, to about 20% per year.

Multiprocessor per Chip

- Instruction Level Parallelism ILP(started on Uni-processor)
  - compiler and hardware used it
- Thread Level Parallelism(TLP) and Data Level Parallelism (DTL)
  - Explicitly parallel
    - Require programmer to write code to gain performance

Applications and Requirements

- Scientific/numerical: weather prediction, molecular modeling
  - Need: large memory, floating-point arithmetic
- Commercial: inventory, payroll, web serving, e-commerce
  - Need: integer arithmetic, high I/O
- Embedded: automobile engines, microwave, PDAs
  - Need: low power, low cost, interrupt driven
- Home computing: multimedia, games, entertainment
  - Need: high data bandwidth, graphics

Classes of Computers

- High performance (supercomputers-going towards cluster of Low cost/power)
  - Supercomputers – Cray T-90
  - Massively parallel computers – Cray T3E
- Balanced cost/performance (high reliability,scalability)
  - Workstations – SPARCstations
  - Servers – SGI Origin, UltraSPARC
  - High-end PCs – Pentium quads
- Low cost/power
  - Low-end PCs, laptops, PDAs – mobile Pentiums