COL216
Computer Architecture

Architecture Space
24th Jan, 2022

What is “Computer Architecture”?

Any relation with architecture (of buildings) ?

Is there an analogy ?

architecture (function, appearance)

vs

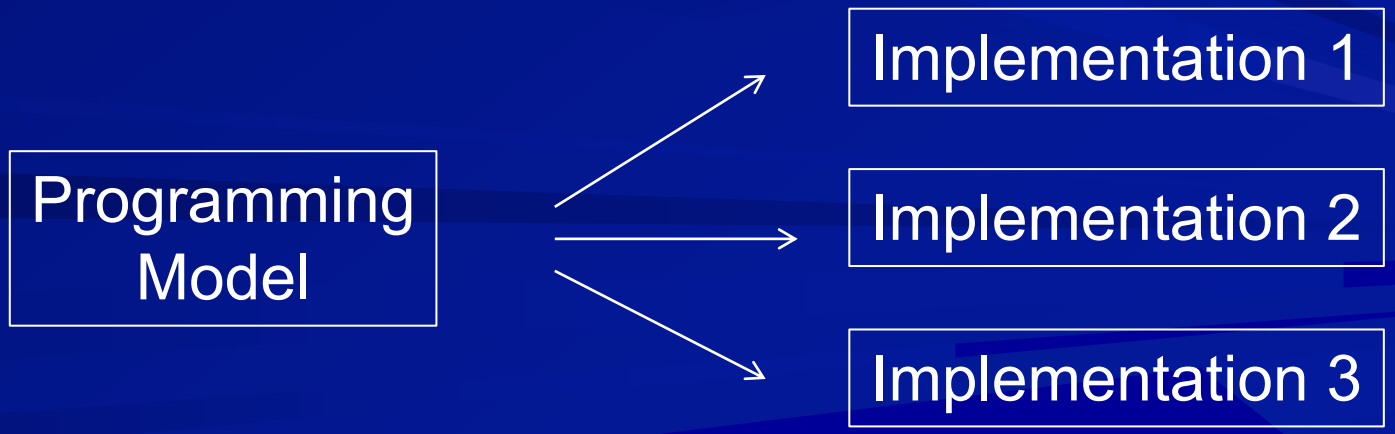
engineering (structure, material)

of

buildings | computers

Architecture of a Computer

- Conceptual design and fundamental operational view of a computer system
- Programming model of a computer, but not a particular implementation

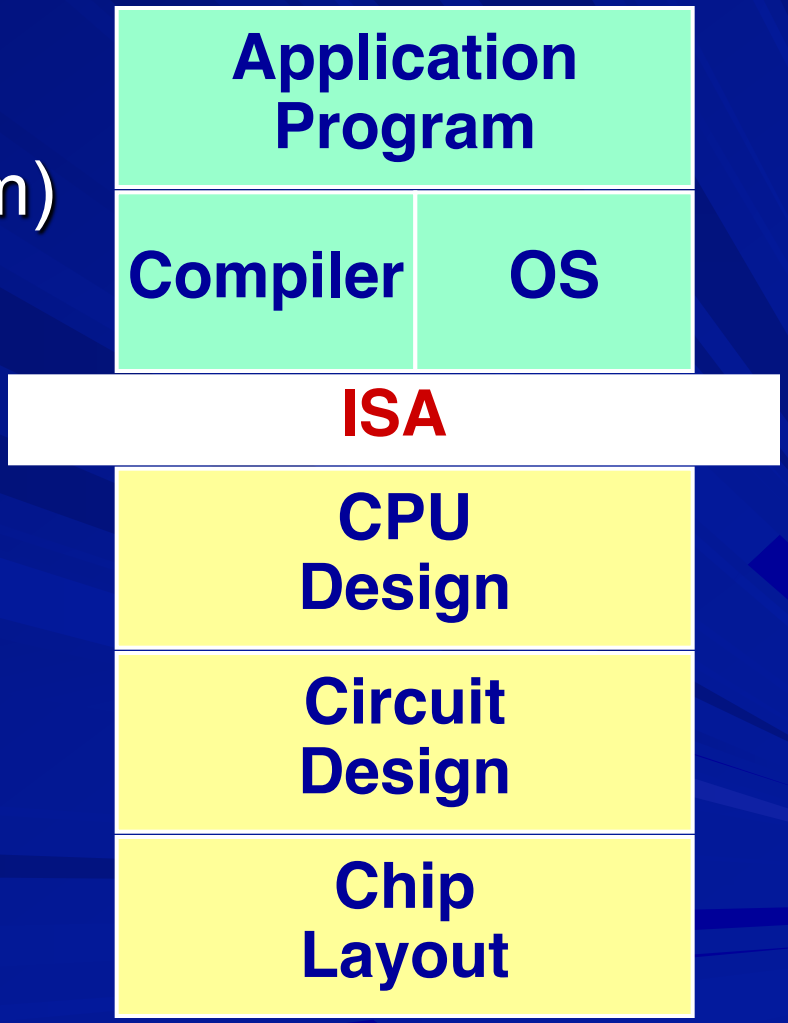


Architecture levels

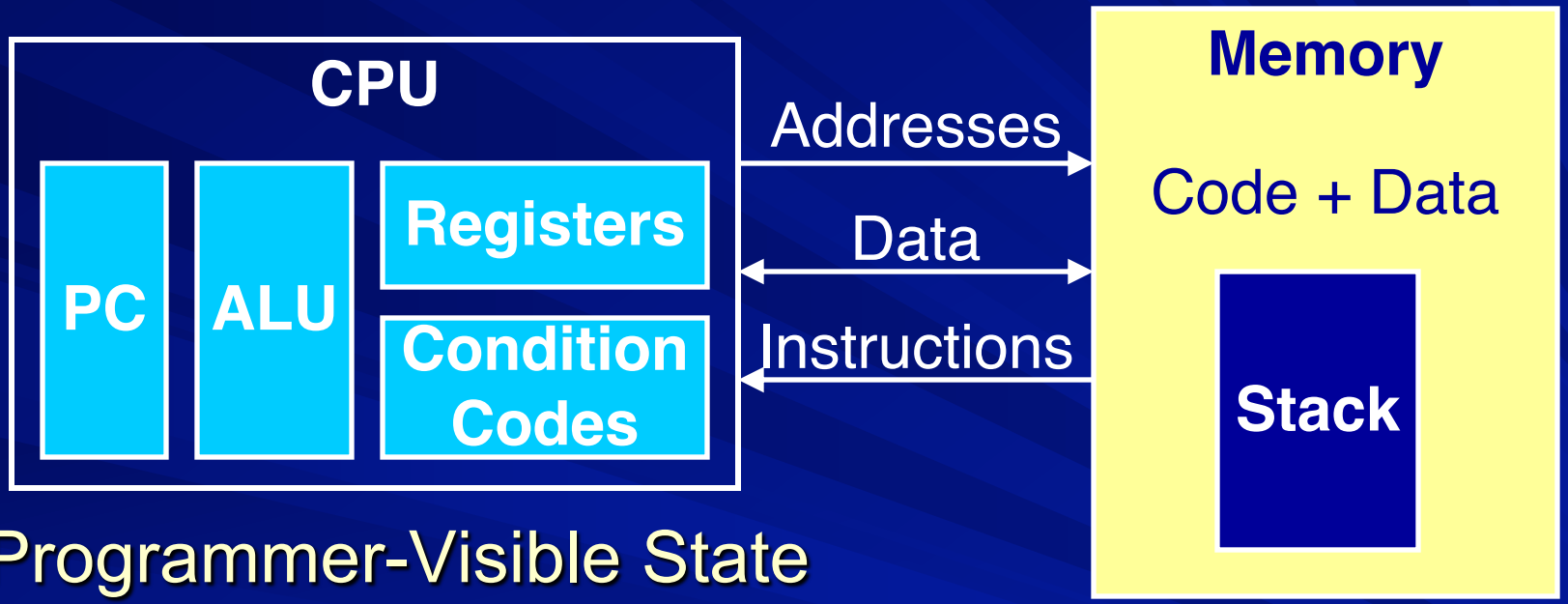
- Instruction set architecture (ISA)
  - Lowest level visible to a programmer
- Micro architecture
  - Fills the gap between instructions and logic modules
- System architecture
  - How processors, memories, buses, etc are put together

Instruction Set Architecture

- Machine Language View
  - Processor state (RF, mem)
  - Instruction set and encoding
- Layer of Abstraction
  - Above: how to program machine - HLL, OS
  - Below: what needs to be built – how to make it run fast



The Abstract Machine



- Programmer-Visible State
  - PC Program Counter
  - Register File
    - heavily used data
  - Condition Codes
  - Memory
    - Byte array
    - Code + data
    - stack

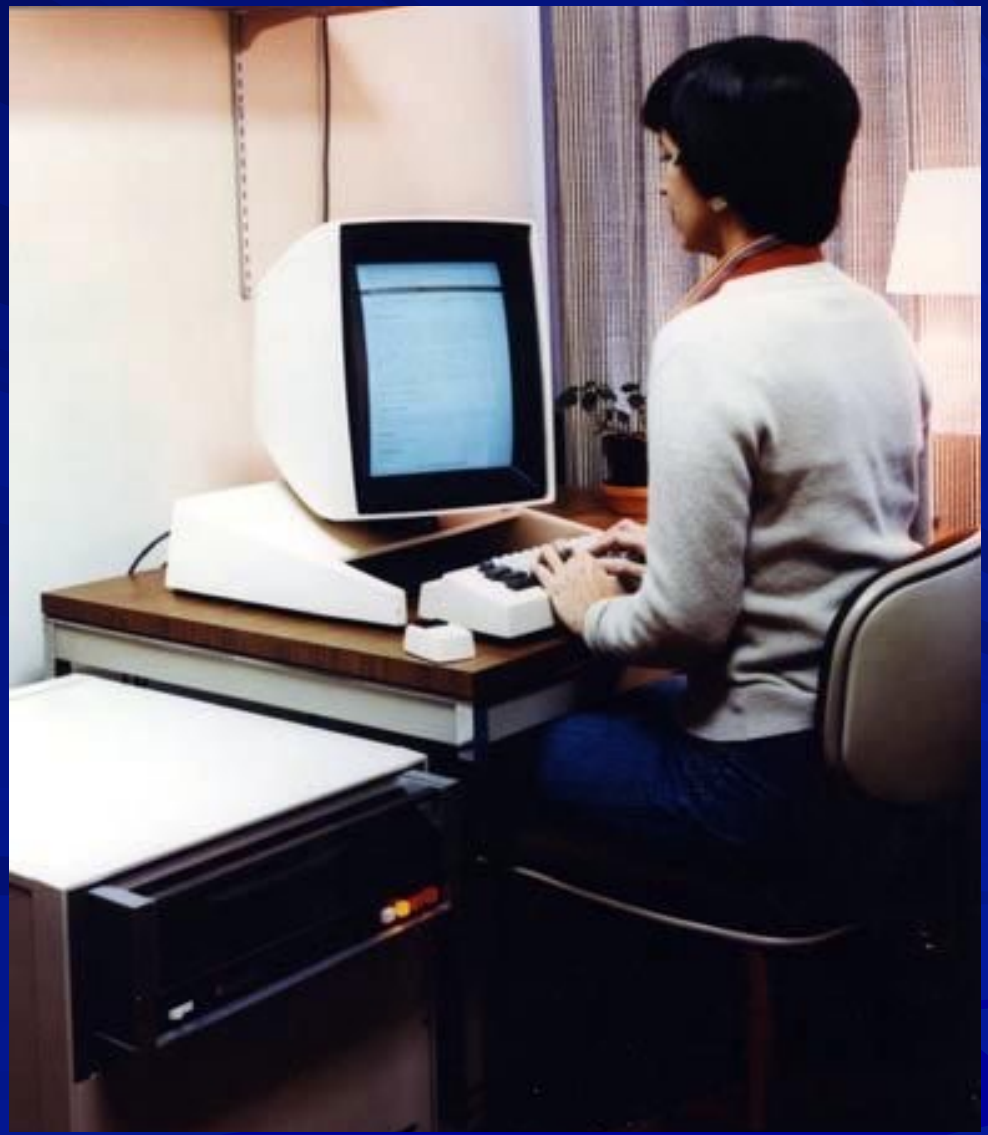
Computers in past

Univac, early 1950s

PDP 1, early 1960s

IBM 360, mid 1960s

Xerox PARC Alto, mid 1970s



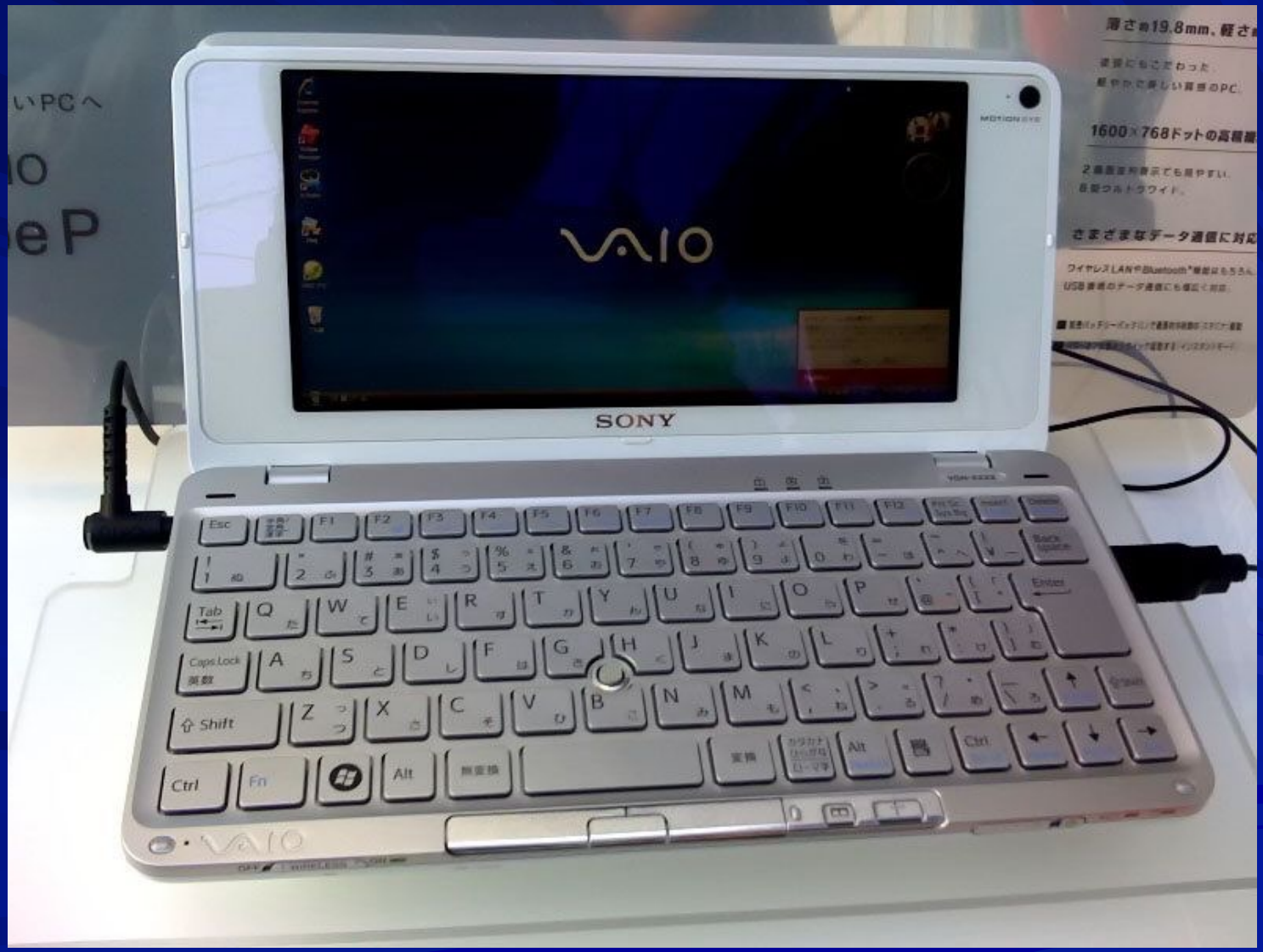
Personal computer

IBM PC, early 1980s

Apple Macintosh, mid 1980s

Apple PowerBook, early 1990s

Sony Vaio Mid 1990s



Earth simulator, 2002

Computers today

- Laptops
- Desktops
- Servers
- Data centres
- Super computers
- Phones, tablets, watches?
- Embedded computers?
- IoT ?

Embedded computers

- Processor as an intelligent electronic component rather than Processor as a computing engine
- Real time operation
- Requires hardware-software co-design
- Highly customized

Difference between processors?

- What is the difference between processors used in desk-tops, lap-tops, mobile phones, washing machines etc.?
  - Performance / speed
  - Power consumption
  - Cost
  - General purpose / special purpose

Studying computer architecture

- How computers work, basic principles
- How to analyze their performance
- How computers are designed and built to meet functional, performance and cost goals
- Advanced concepts related to modern processors (caches, pipelines, etc.)

Why should we study this?

This knowledge will be useful if you need to

- design/build a new computer
  - rare opportunity
- design/build a new version of a computer
- improve software performance
- purchase a computer
- provide a solution with an embedded computer

Is it enough to study ARM ISA?

- YES

It is possible to learn the basic principles through this example architecture

- NO

It is useful to have some idea about the other prominent architectures also

Basic Principles

- How a program is represented and executed in hardware
  - Performing computations
  - Organizing data
    static and dynamic structures
  - Organizing control flow
    conditionals, loops, functions, recursion

What constitutes ISA?

Main features:

- Set of basic/primitive operations
- Storage structure – registers/memory
- How addresses are specified
- How instructions are encoded

Set of operations

Basic/primitive operations only vs more powerful operations

- e.g. “K++ and branch to L if K>N, where K is in memory” or “copy a block of data in memory”

- Goal is to reduce number of instructions executed
- Danger is a slower cycle time and/or a higher CPI

Location of operands – R/M

- R-R both operands in registers
- R-M one operand in register and one in memory
- M-M both operands in memory
- R+M Combines R-R, R-M and M-M

How many operand fields?

- 3 address machine — r1 = r2 + r3
- 2 address machine — r1 = r1 + r2
- 1 address machine — Acc = Acc + x
  Acc is implicit
- 0 address machine — add values on top of stack

Register organizations

- Register-less machine
- Accumulator based machine
- A few special purpose registers
- Several general purpose registers
- Large number of registers / register windows

RISC vs. CISC

Reduced (vs. Complex) Instruction Set Computer

- Uniformity of instructions
- Simple set of operations and addressing modes
- Register based architecture with 3 address instructions
- Virtually all new instruction sets since 1982 have been RISC

RISC and CISC

- Concept introduced by Patterson & Ditzel in 1980
- Followed by development of MIPS at Stanford and Berkeley RISC at UCB
- Virtually all new instruction sets since 1982 have been RISC
- Patterson & Hennessey were given Touring Award in 2018 for their “pioneering work on computer chip design”

"The case for the Reduced Instruction Set Computer"

by Patterson & Ditzel, ACM Sigarch, 1980

- *" .. the next generation of VLSI computers may be more efficiently implemented as RISC's than CISC's."*
- Reasons for increased complexity
- Usage and consequences of CISC instructions
- RISC and VLSI technology

Why processors became complex?

- Trend to provide complex instruction to do common tasks
- Adding instructions facilitated by microprogrammed control approach
- New generations created by adding new instructions, rather than fresh designs
- Code density was considered important
- Attempt to support HLLs

Consequences of CISC approach

- Only a few instructions used by compilers
- Sequence of simpler instructions may be faster than complex instructions
- Advantage of higher code density diminished with advances in memory technology
- Benefit for HLL features questionable
- Lengthened design times, increased design errors

RISC and VLSI technology

- Easier to fit in a chip
- Short design time suitable for rapidly changing technology
- Efficient implementation
- Area saved usable for cache, pipelining etc

Thank you