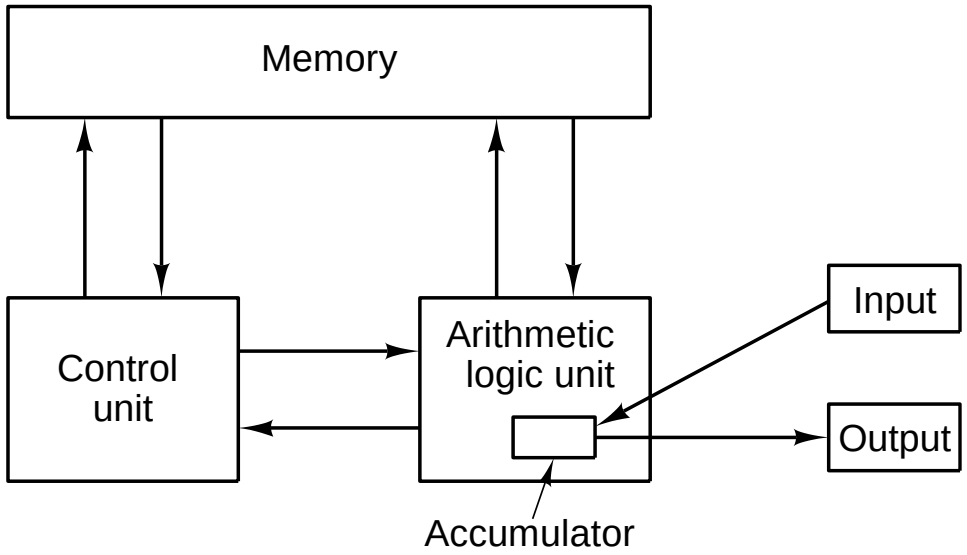
Prof. Dr.-Ing. Jochen Schiller
Computer Systems & Telematics

TI II: Computer Architecture Introduction

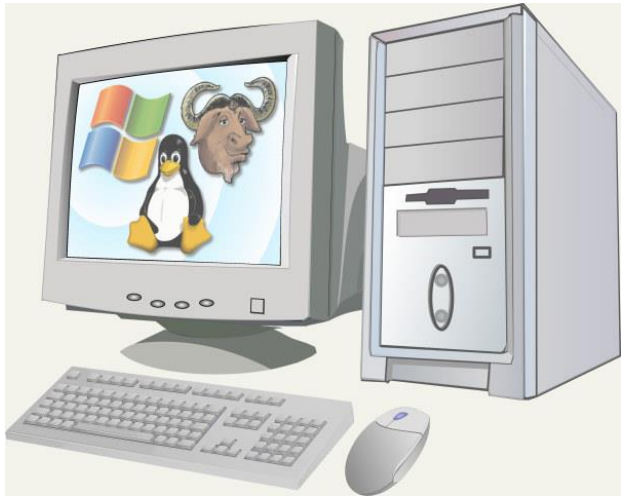


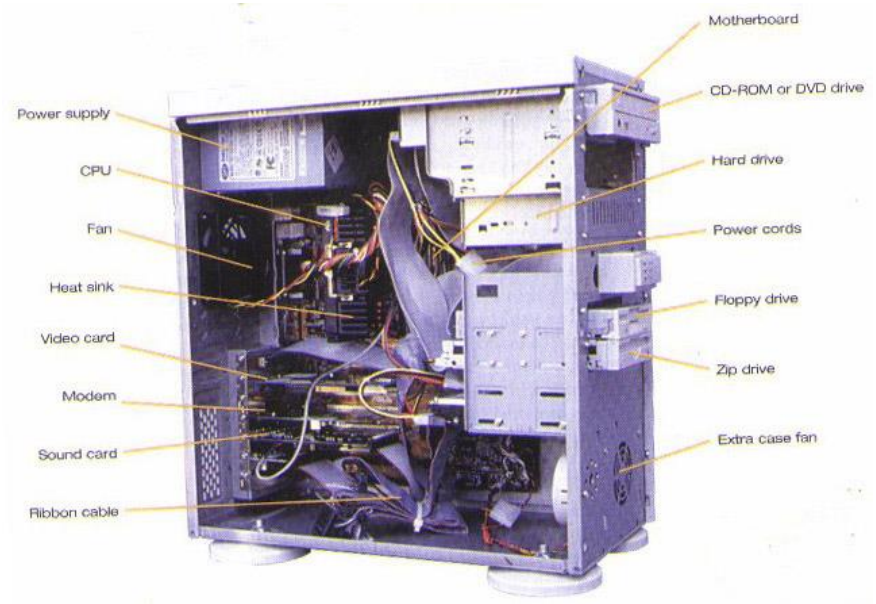
Single Processor Systems
Historical Background
Classification / Taxonomy
Architectural Overview
Examples
The Layered Computer Model

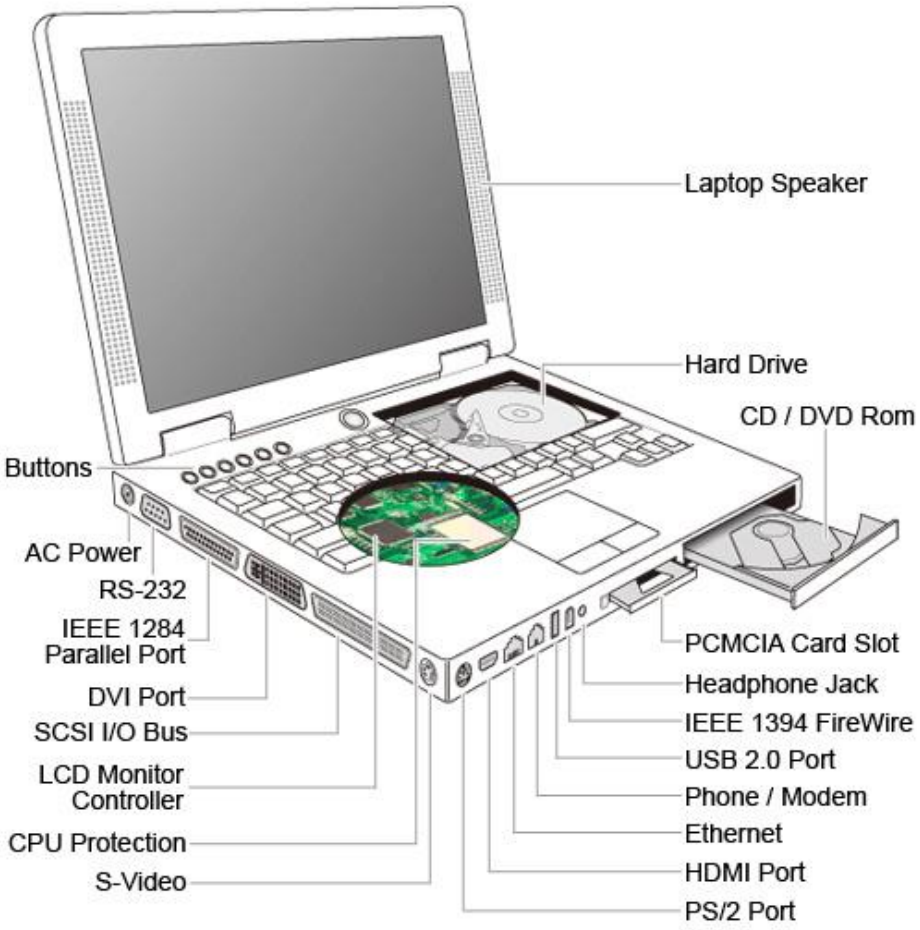
Contents

SINGLE PROCESSOR SYSTEMS

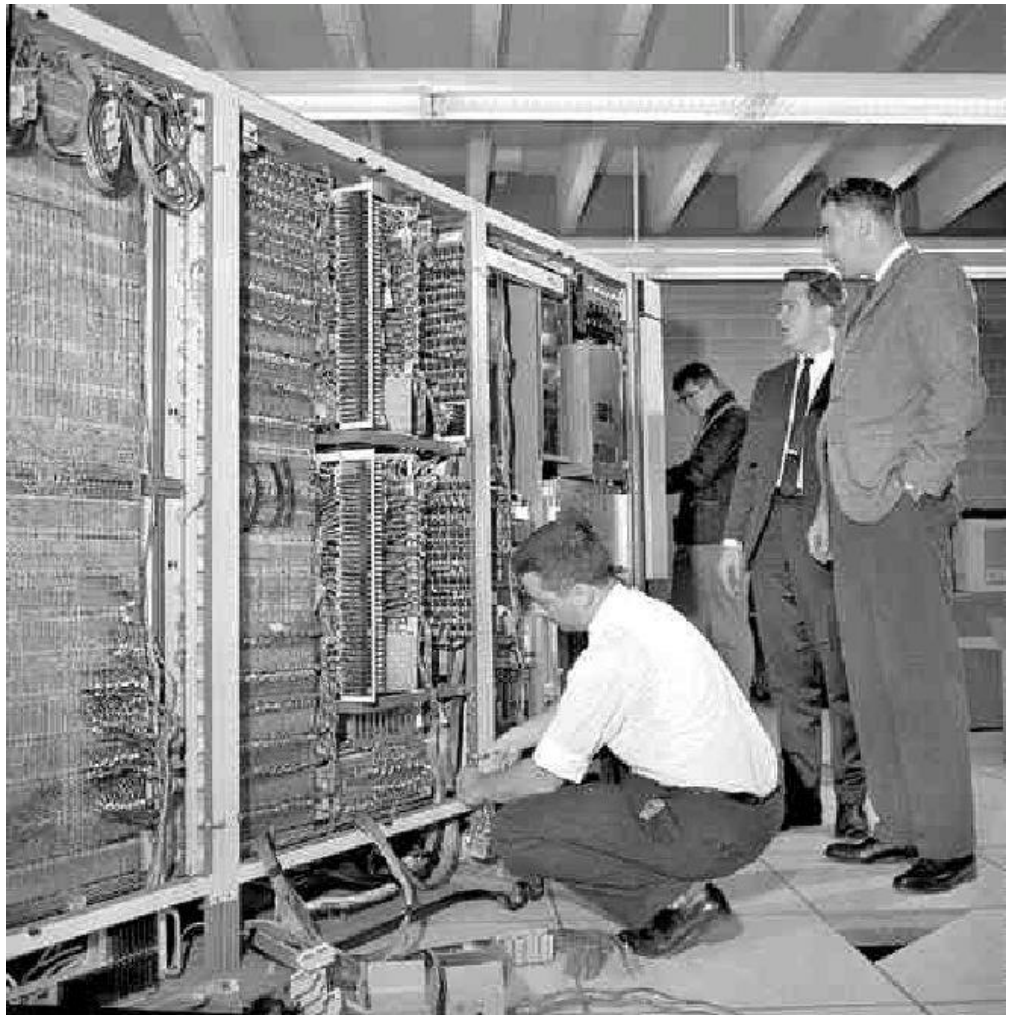
Computer System – the old days …





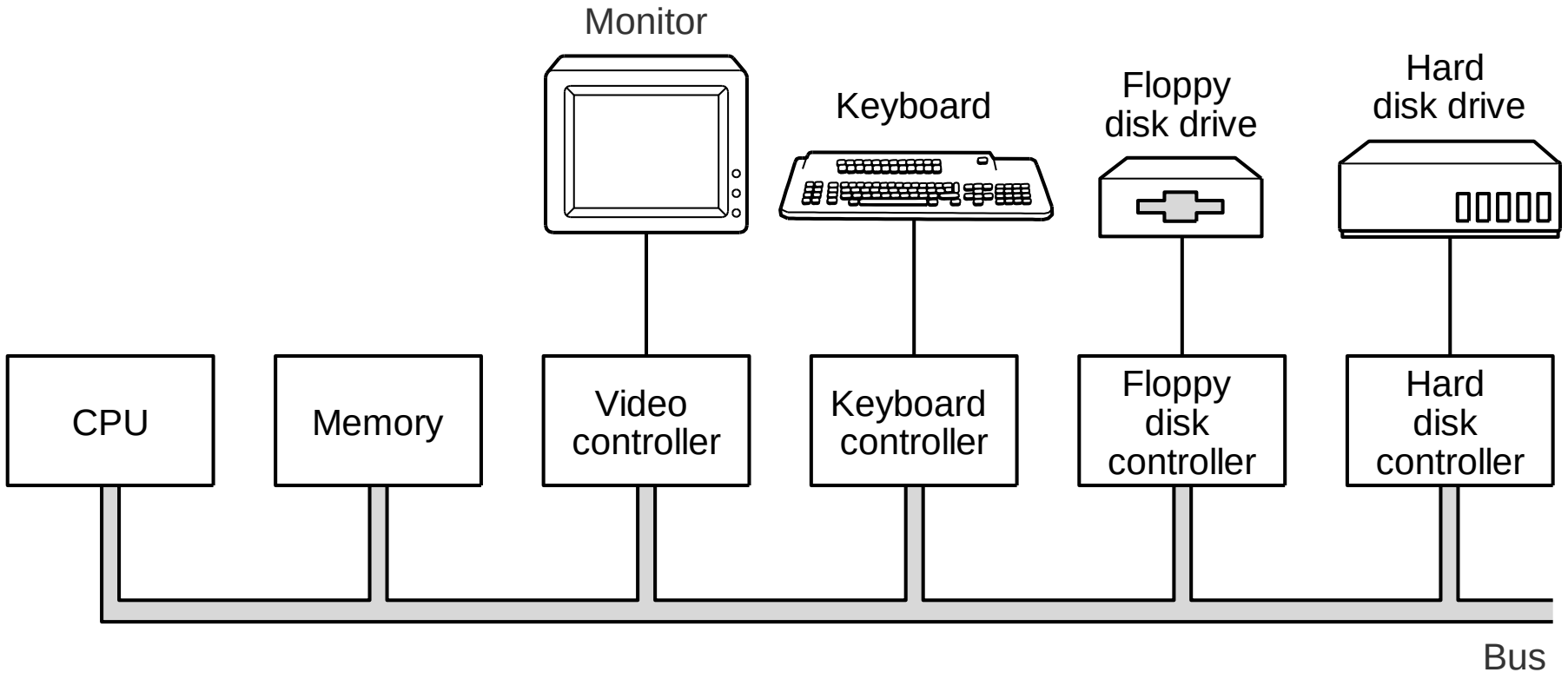


Computer System – very old days

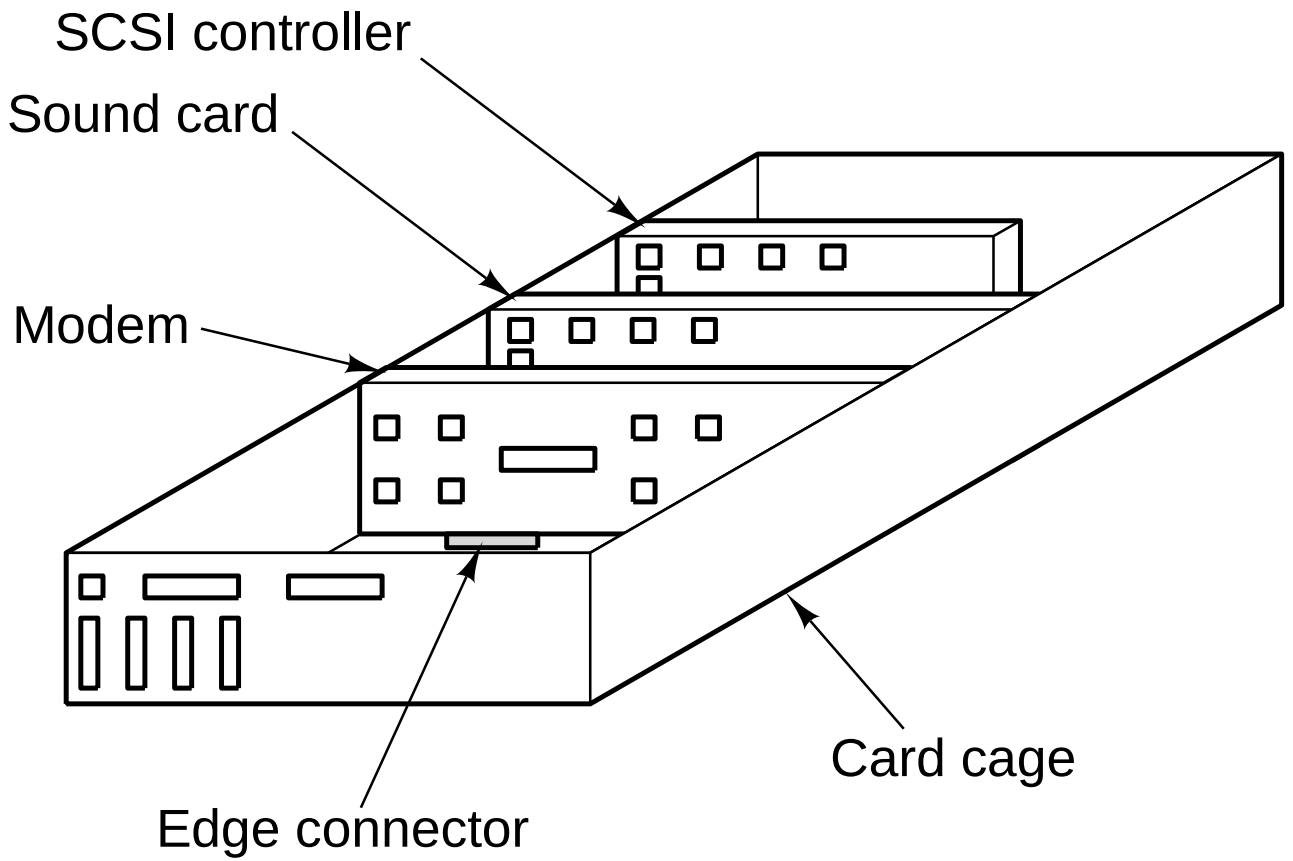


Computer System – not much to see

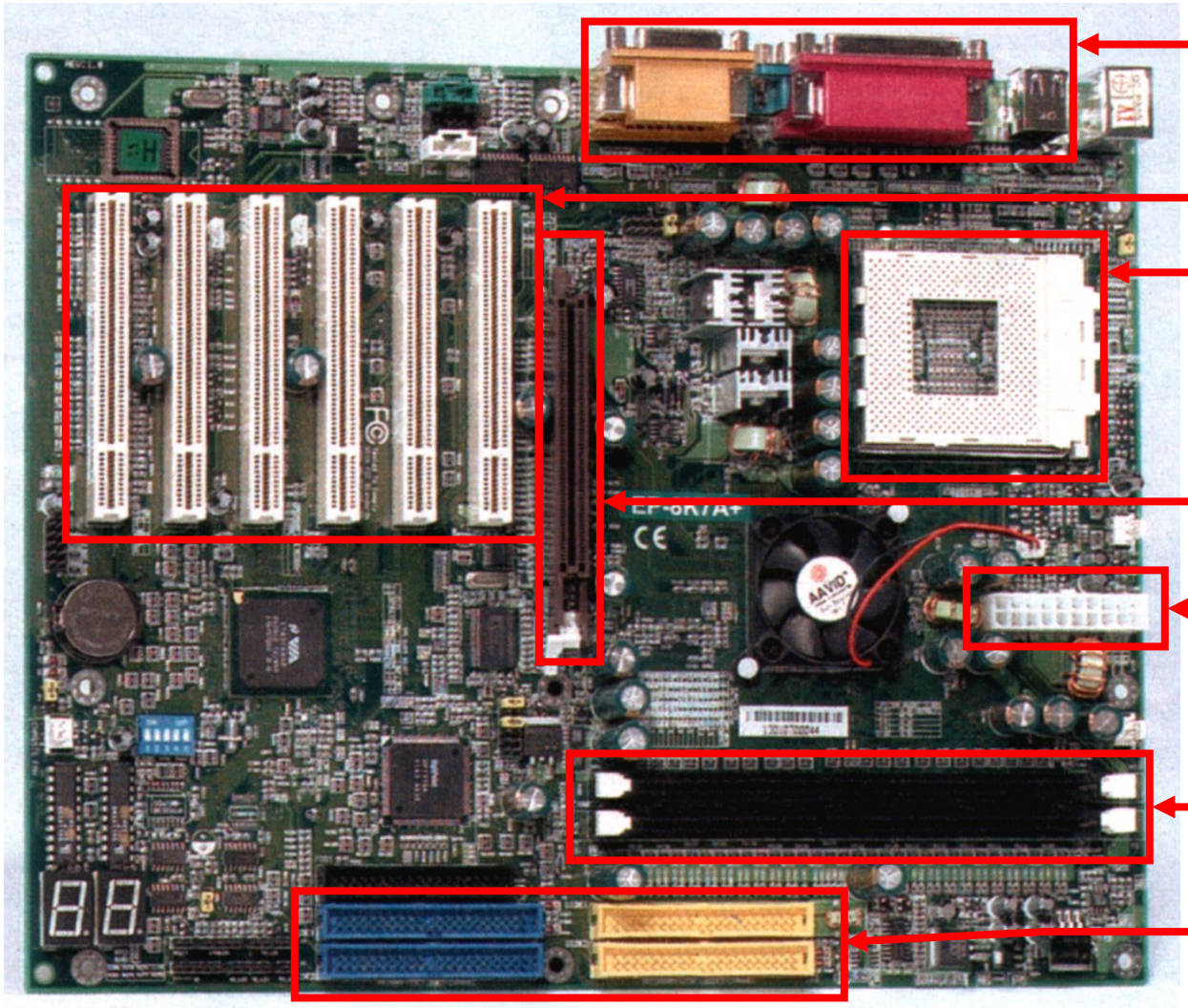
Classical architecture of microcomputers



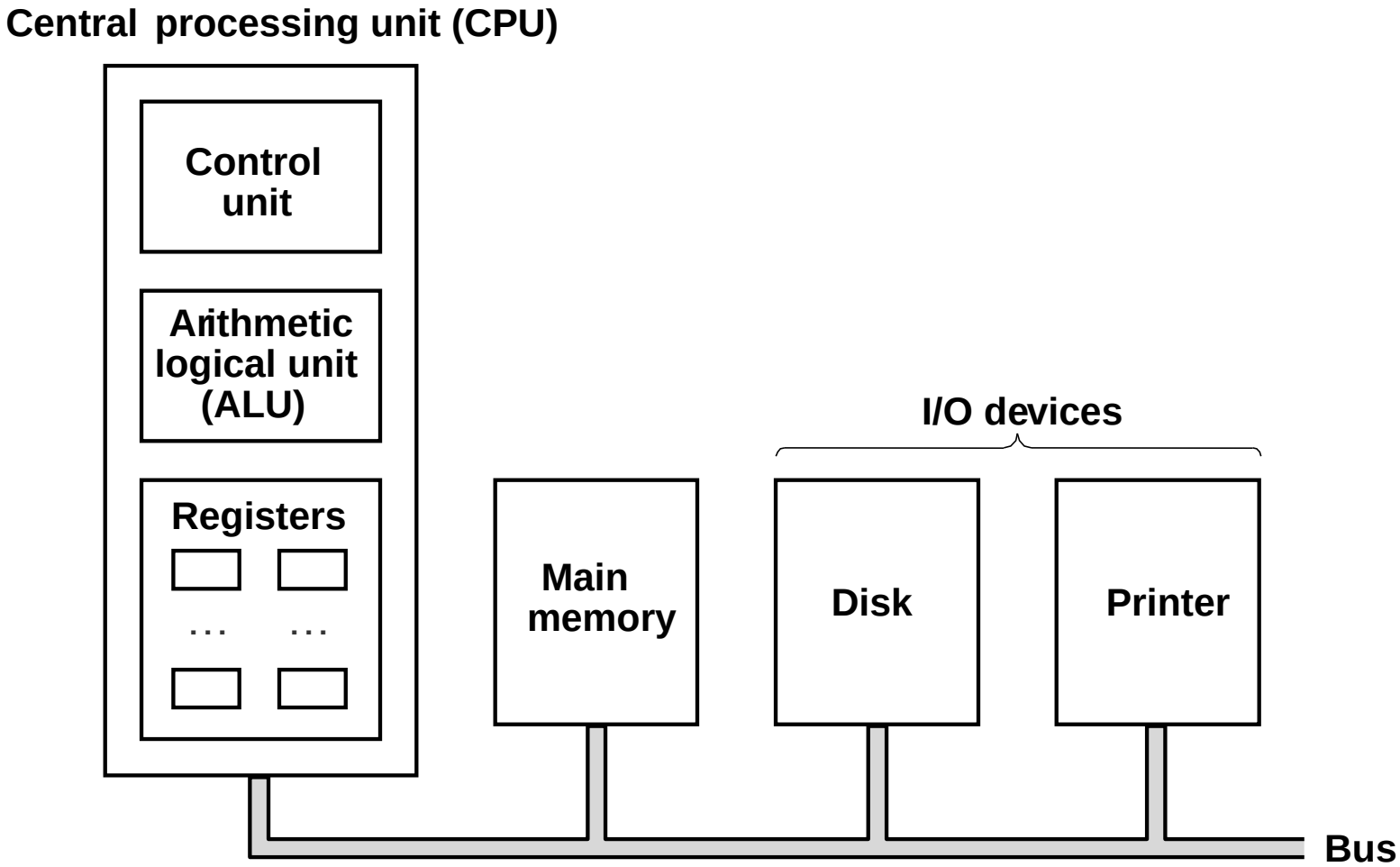
Classical idea of a modular computer system



A complete classical single processor system



Simple single processor system



Do we still have this today? Think of, e.g., mobile phone with main CPU, radio modem, graphics accelerator, GPS, WLAN, … later more about this!

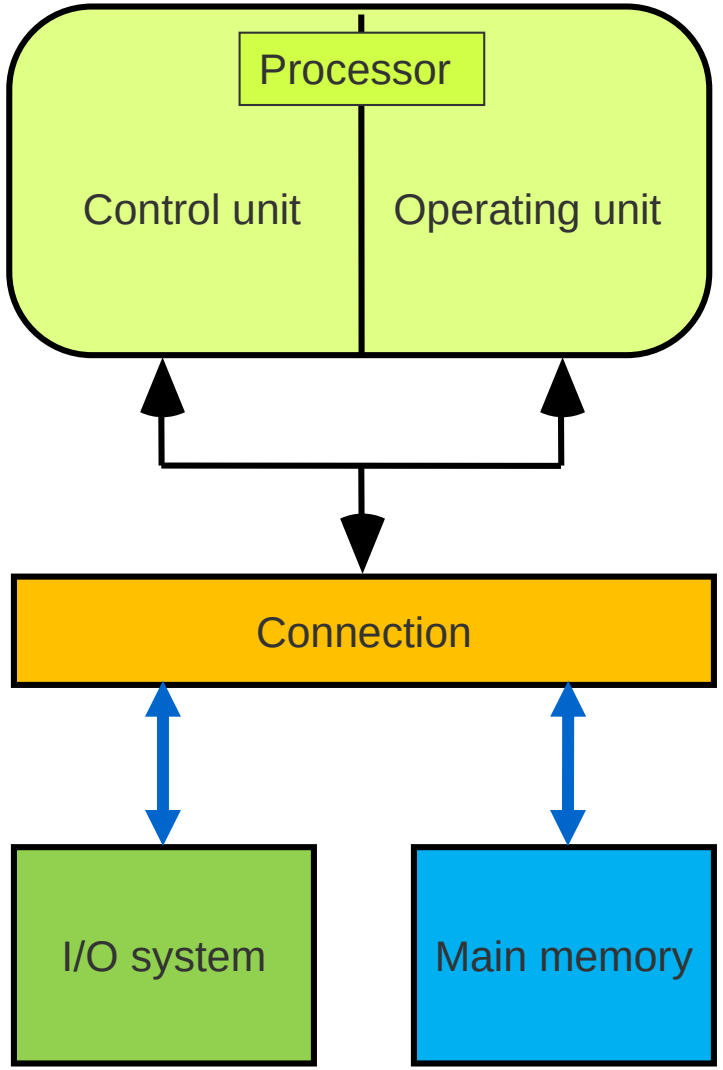
THE VON NEUMANN ARCHITECTURE

The von Neumann architecture

The von Neumann architecture forms the basis of many hardware architectures presented in this course.

The architecture comprises the following main components

- Central processing unit
  - Control unit
  - ALU / Operating unit
- Memory
- Input/Output units
- Interconnection



The von Neumann architecture

Central control of the computer

A computer consists of **several functional units** (central processing unit, memory, input/output unit, connection)

The computer is **not tailored to a single problem** but a general purpose machine. In order to solve a problem a program is stored in the memory (“program controlled universal computer”) – yes, today this sounds so simple…

IMPORTANT

- Instructions (the program) and data (input and output values) are stored in the **same** memory.
- The memory consists of memory cells with a fixed length, each cell can be addressed individually.

The von Neumann architecture

Processor, central unit (CPU: "central processing unit")

- Controls the flow and execution of all instructions

Control unit

- Interprets the CPU instructions
- Generates all control commands for other components

Arithmetic Logical Unit (ALU)

- Executes all instructions (I/O and memory instructions with the help of these units)

Input/Output system

- Interface to the outside world
- Input and output of program and data

Memory

- Storage of data and program as sequence of bits

Interconnection

The von Neumann Architecture

PRINCIPLE OF OPERATION OF A COMPUTER

Principle of Operation of a Computer

At any time the CPU executes only a **single instruction**. This instruction can only manipulate a **single operand**.

- Traditionally, this is called **SISD** (Single Instruction Single Data).

Code and data are stored in the **same memory** without any distinction. There are no memory protection mechanisms – programs can destroy each other, programs can access arbitrary data.

Memory is unstructured and is addressed linearly. Interpretation of memory content depends on the context of the current program only.

Two phase principle of instruction processing:

- During the interpretation phase the content of a memory cell is fetched based on a **program counter**. This content is then interpreted as an instruction (note: this is a pure interpretation!).
- During the execution phase the content of a memory cell is fetched based on the address contained in the instruction. This content is then interpreted as data.

The instruction flow follows a strict sequential order.

Principle of Operation of a Computer Instruction Execution

Example: interpreter(memory, 256);

memory[256] = 80	memory[261] = 7
memory[257] = 0	memory[262] = 20
memory[258] = 5	memory[263] = 2
memory[259] = 80	memory[264] = 0
memory[260] = 1	memory[265] = 1

Byte sequence: 80 0 5 80 1 7 20 2 0 1

Bit sequence: 01010000 00000000 00000101
01010000 00000001 00000111
00010100 00000010 00000000 00000001

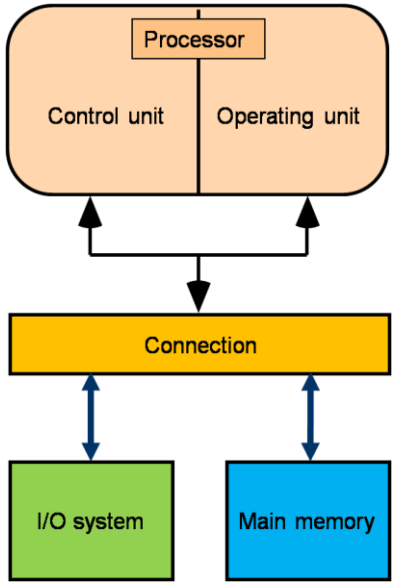
Assembler representation (MMIX Style)

```
LDB   $0,5
LDB   $1,7
ADD   $2,$0,$1
```

High-level programming language representation

Z = X + Y

Pros and Cons of the von Neumann architecture



Advantages

- Principle of minimal hardware requirements
- Principle of minimal memory requirements

Disadvantages

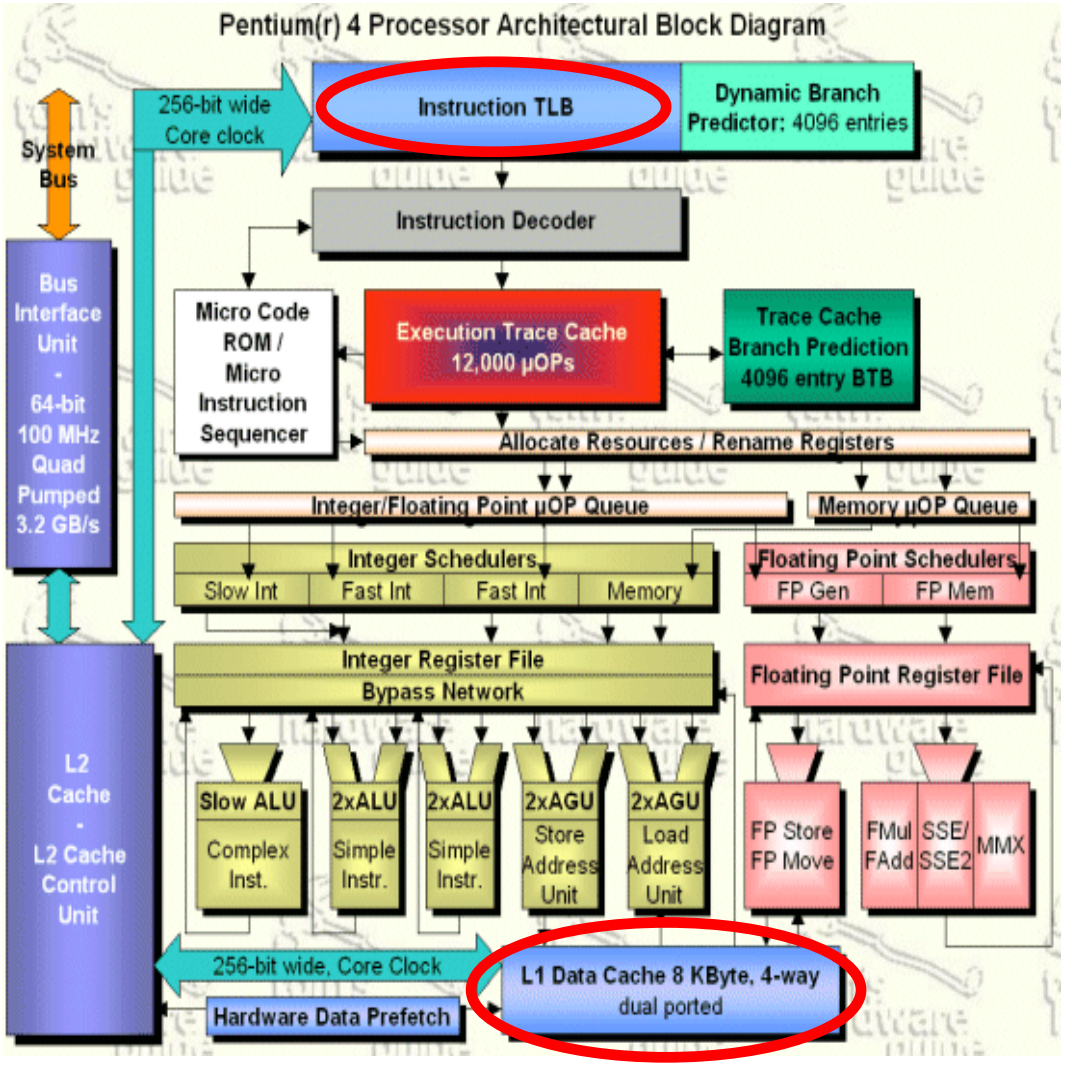
- The main interconnection (memory ↔ CPU) is the central bottleneck: the "**von Neumann bottleneck**"
- Programs have to take into account the sequential data flow across the von Neumann bottleneck → Influences on higher programming languages ("intellectual bottleneck")
- Low structuring of data (a long sequence of bits…)
- The instruction determines the **operand type**
- No memory protection

Look around:
Not everything is von Neumann!

Basic “von Neumann-architecture”: data and program are stored in the same memory

Typical for the PC architecture…

- well, depends on your viewpoint…



Harvard Architecture

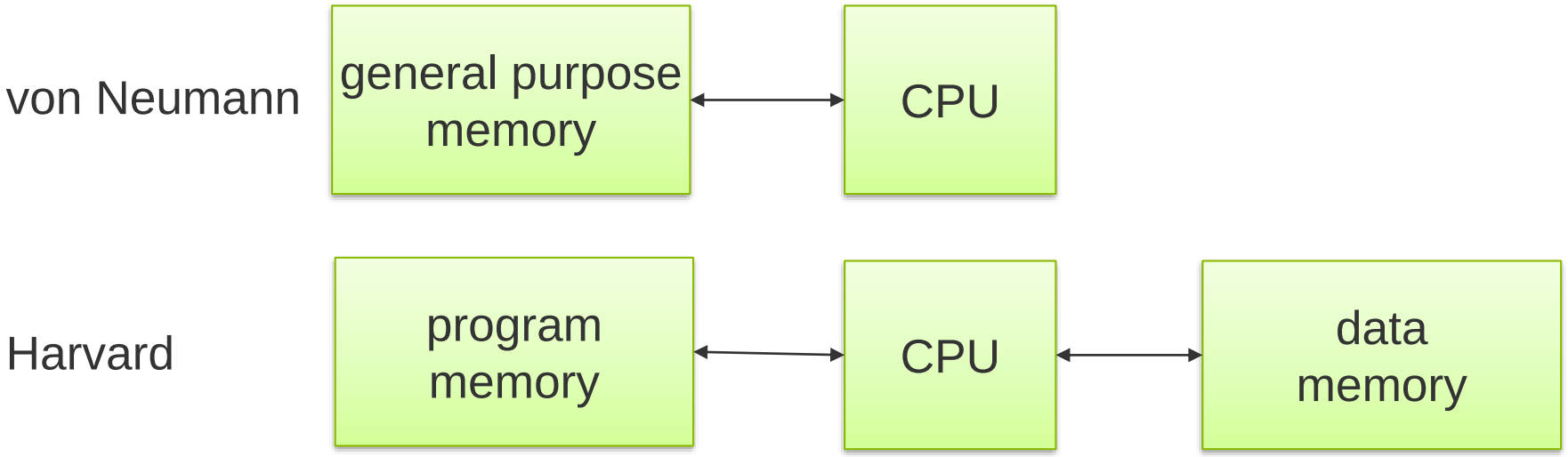
Classical definition of the Harvard architecture

- Separation of data and program memory

Most processors with microarchitecture

- von Neumann from the outside
- Harvard from the inside
- Reason
  - Different time scales and locality when caching data and instructions

High-level comparison



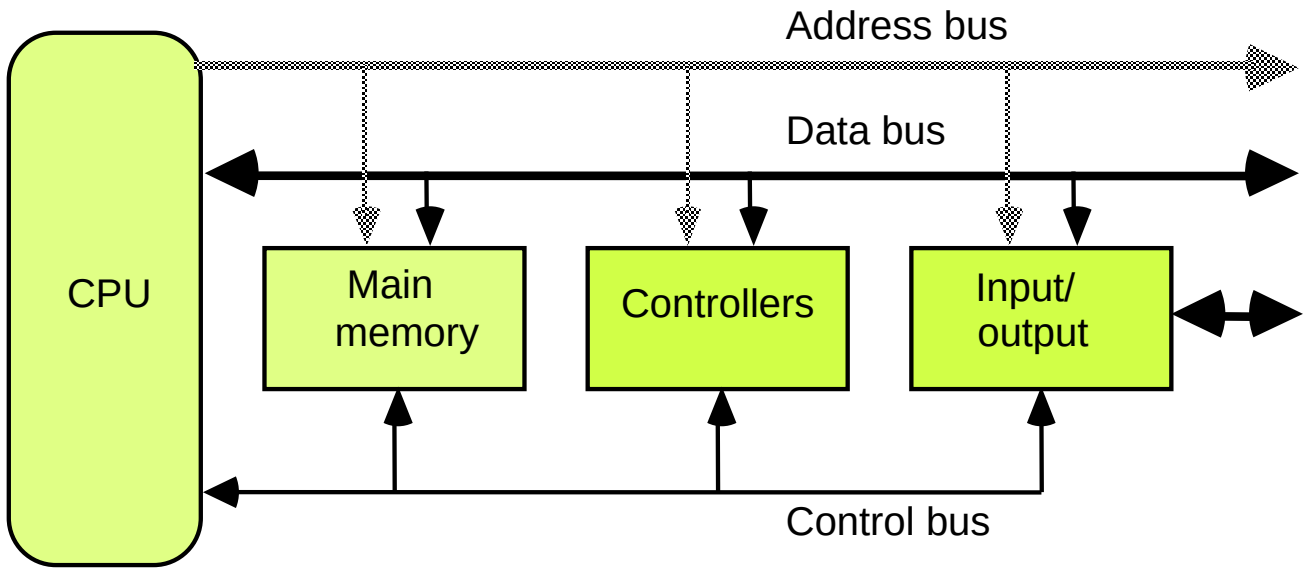
MICRO COMPUTERS

Basic concept of a micro computer

The IBM PC is a modified von Neumann architecture and was introduced by IBM fall 1981.

The interconnection structure was realized by a **bus**.

- The bus connects the CPU with the main memory, several controllers and the input/output system.

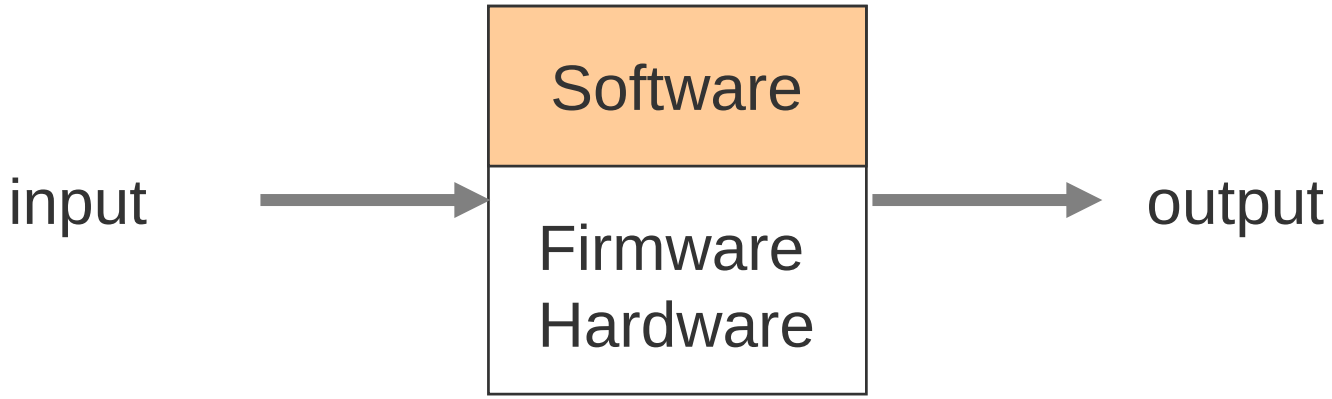


Components of a Computer

Hardware: all mechanical and electronic components

Software: all programms running on the computer

Firmware: micro-programs stored in ROM, somewhere in-between SW and HW



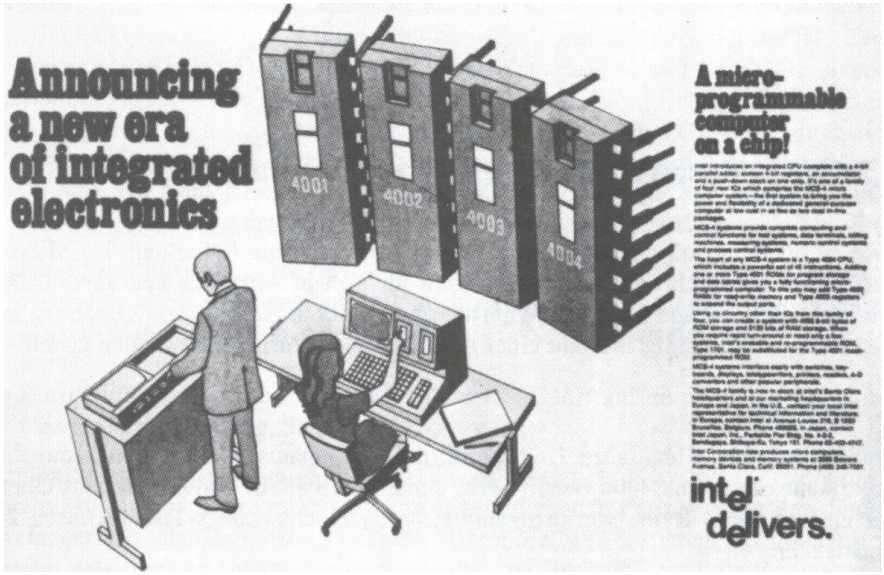
➡Research: Where to place a function – software or hardware?

➡Hardware and Software are logically equivalent!

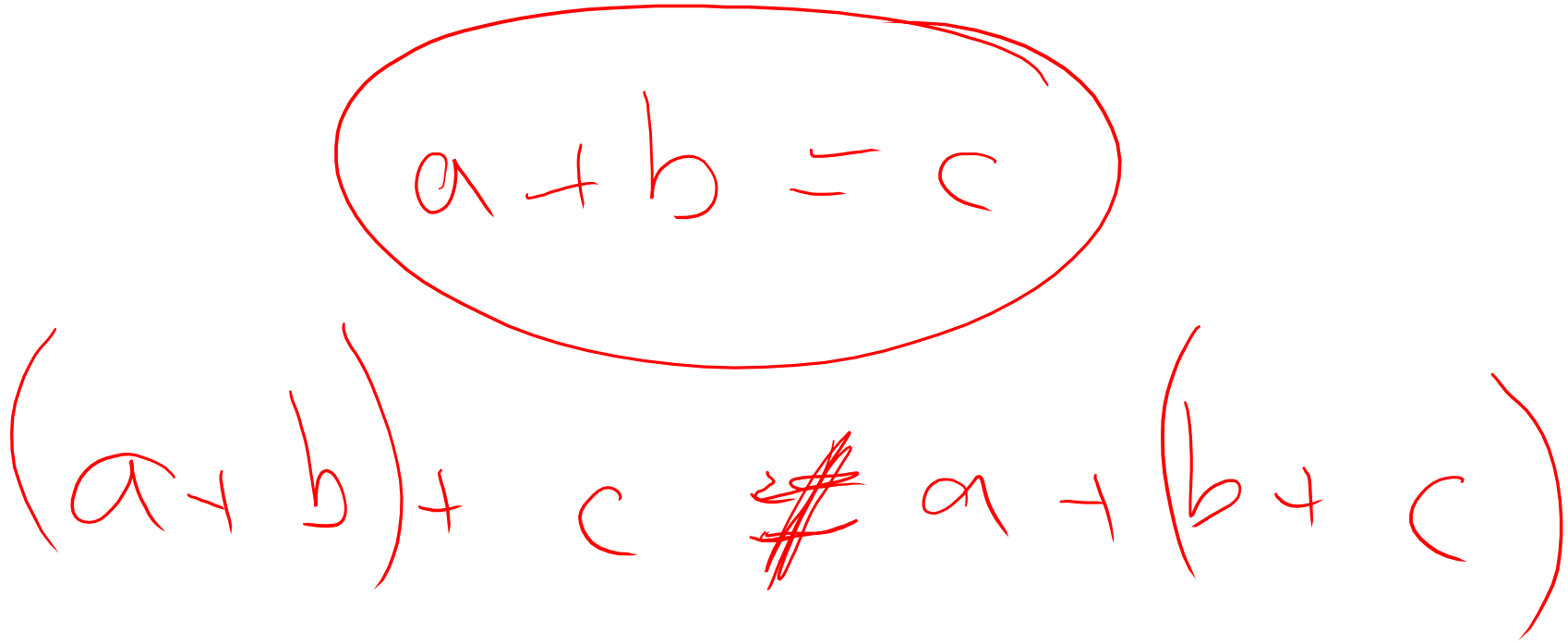
History of Computers

See for example:

- http://www.computerhistory.org/
- http://en.wikipedia.org/wiki/History_of_computing_hardware

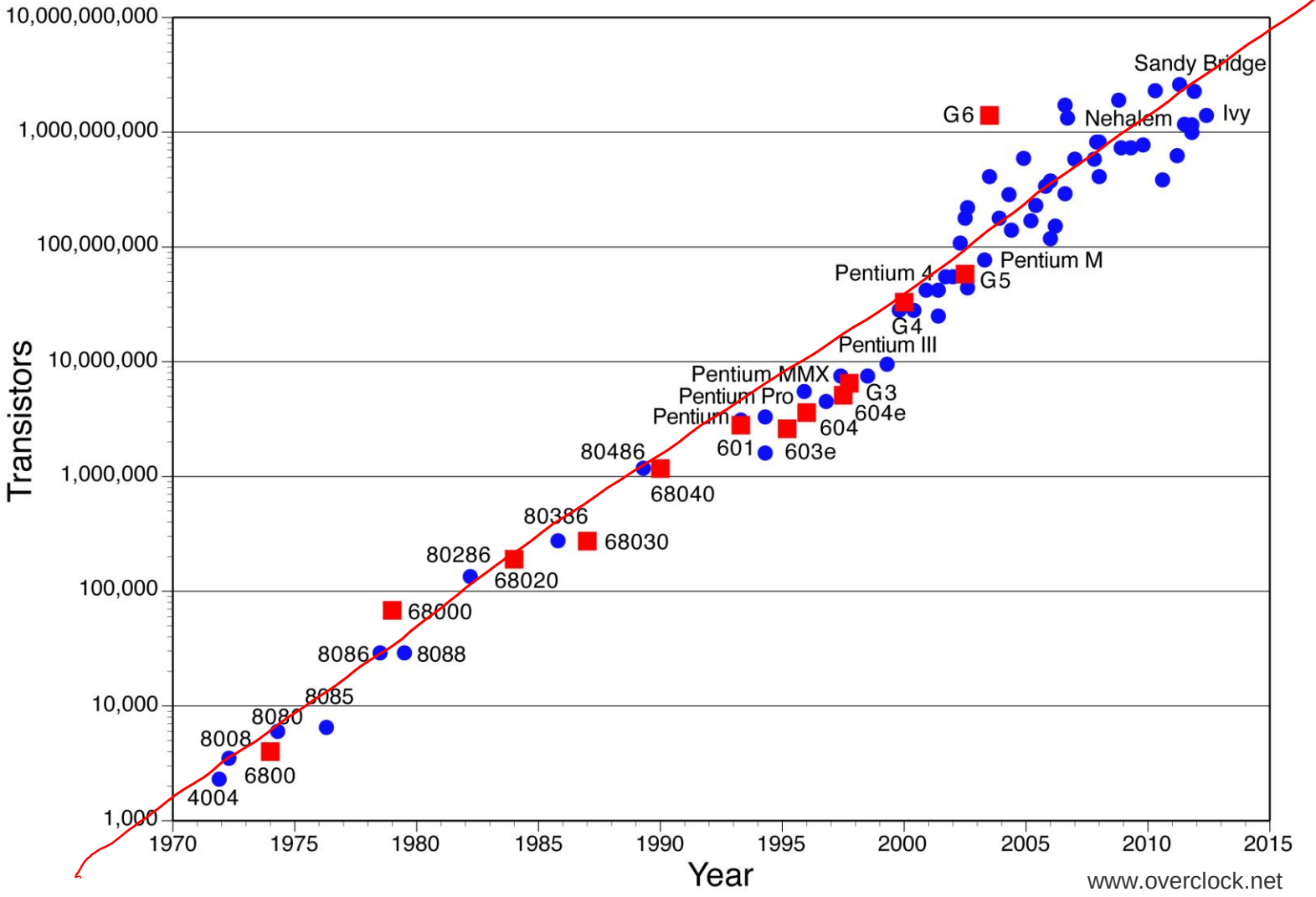


„Computer history museum“ by Dzou - en.wikipedia / CC BY-SA 3.0



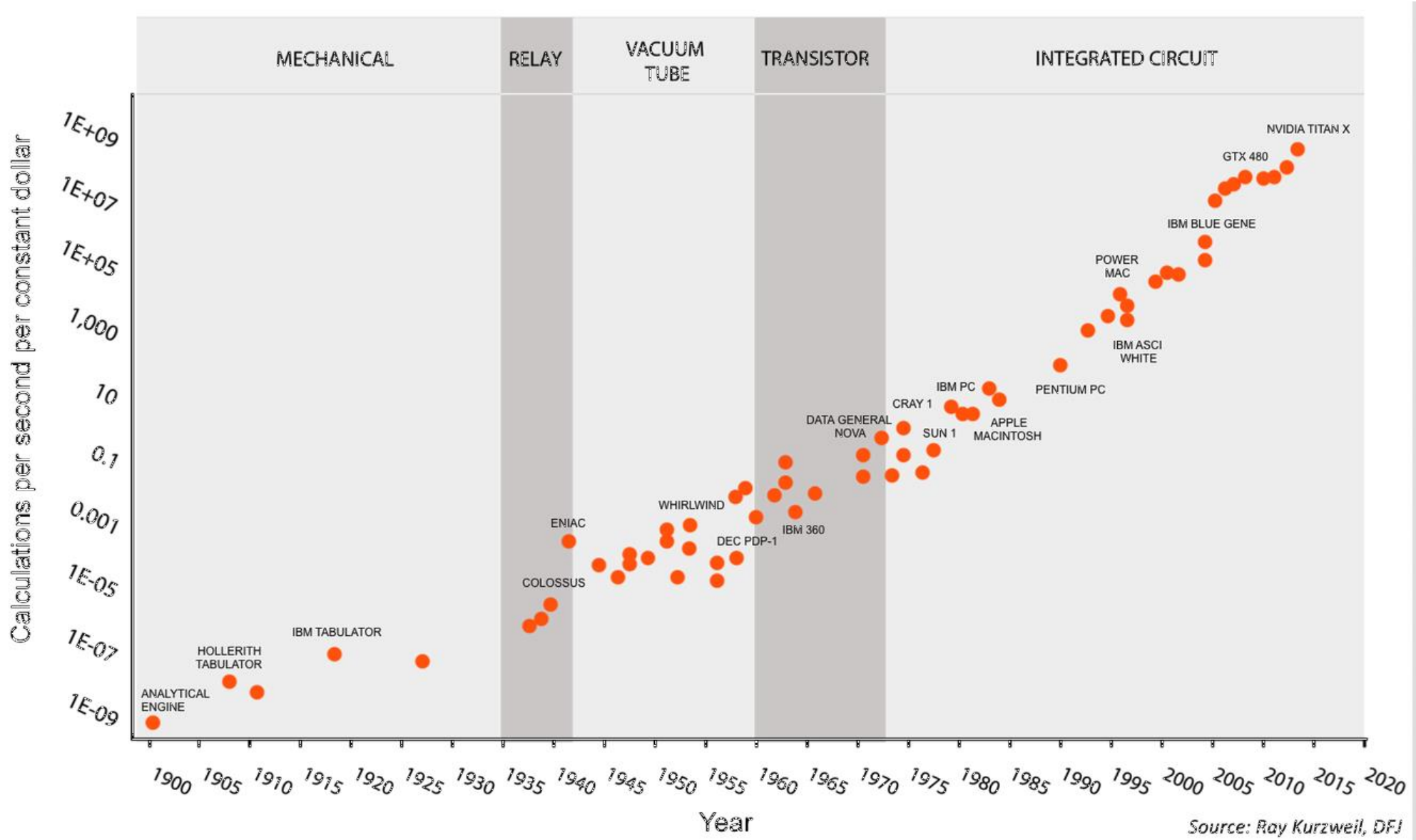
PERFORMANCE OF PROCESSORS

Anwendung vom Mooreschen Gesetz auf CPU-Chips



„Alle 18 Monate verdoppelt sich die verfügbare Rechenleistung“

120 Years of Moore‘s Law



Performance increase, price decrease

Year	Name	Size (cu. ft.)	Power (watts)	Performance (adds/sec)	Memory (KB)	Price	Price-performance vs. UNIVAC	Adjusted price (2003 $)	Adjusted price-performance vs. UNIVAC
1951	UNIVAC I	1,000	125,000	2,000	48	$1,000,000	1	$6,107,600	1
1964	IBM S/360 model 50	60	10,000	500,000	64	$1,000,000	263	$4,792,300	318
1965	PDP-8	8	500	330,000	4	$16,000	10,855	$75,390	13,135
1976	Cray-1	58	60,000	166,000,000	32,000	$4,000,000	21,842	$10,756,800	51,604
1981	IBM PC	1	150	240,000	256	$3,000	42,105	$5,461	154,673
1991	HP 9000/ model 750	2	500	50,000,000	16,384	$7,400	3,556,188	$9,401	16,122,356
1996	Intel PPro PC (200 MHz)	2	500	400,000,000	16,384	$4,400	47,846,890	$4,945	239,078,908
2003	Intel Pentium 4 PC (3.0 GHz)	2	500	6,000,000,000	262,144	$1,600	1,875,000,000	$1,600	11,452,000,000

Source: Patterson, Hennessy, Computer Organization and Design

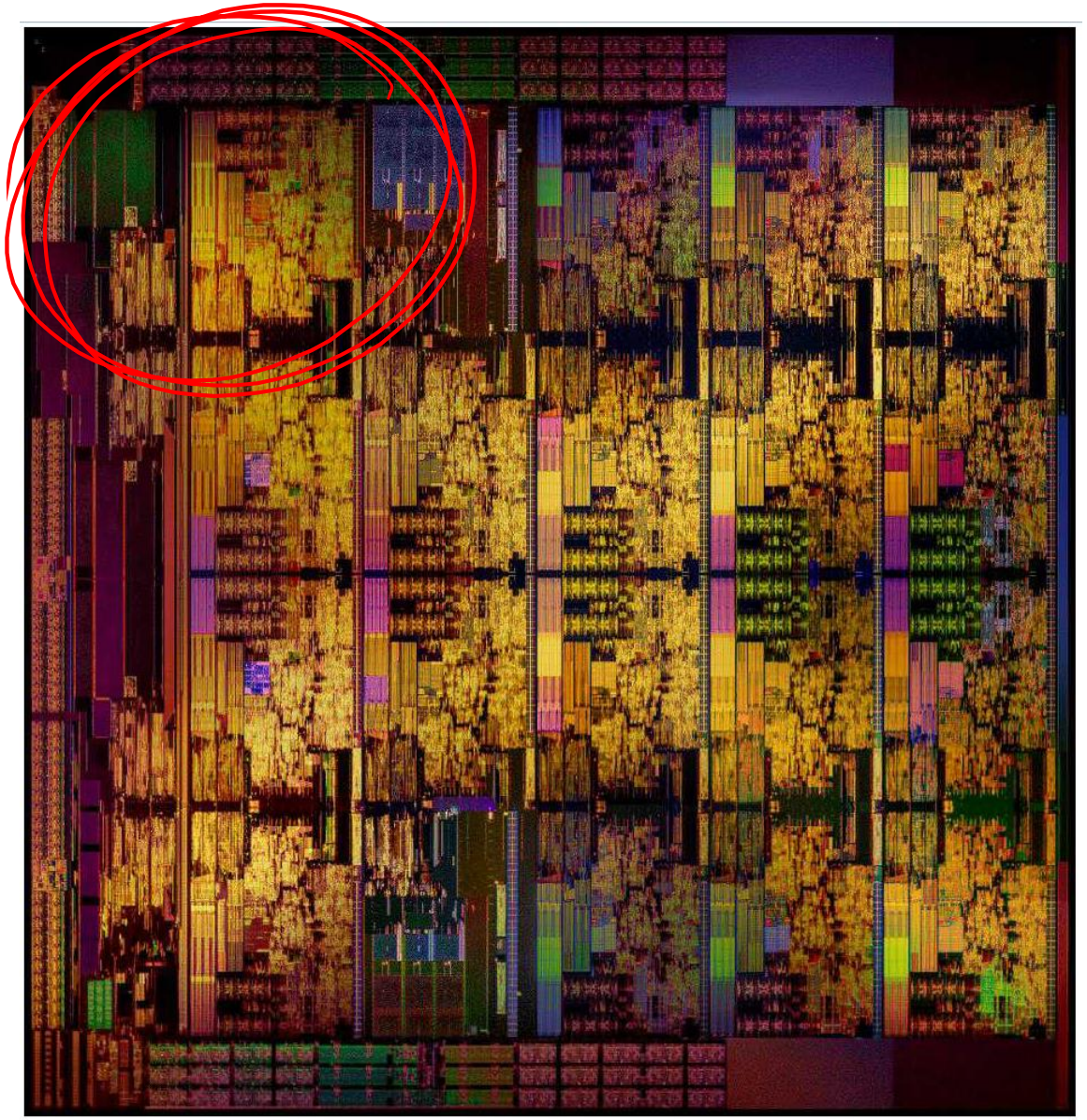
Example Processor: Intel Core i9

Up to 18 cores

Up to $2000

Up to 1.3 Tflop/s dual precision

Up to 165 W TDP



Das Die des Intel Core i9-7900X

Overclocking of i9-7980XE

→ 1000 W!

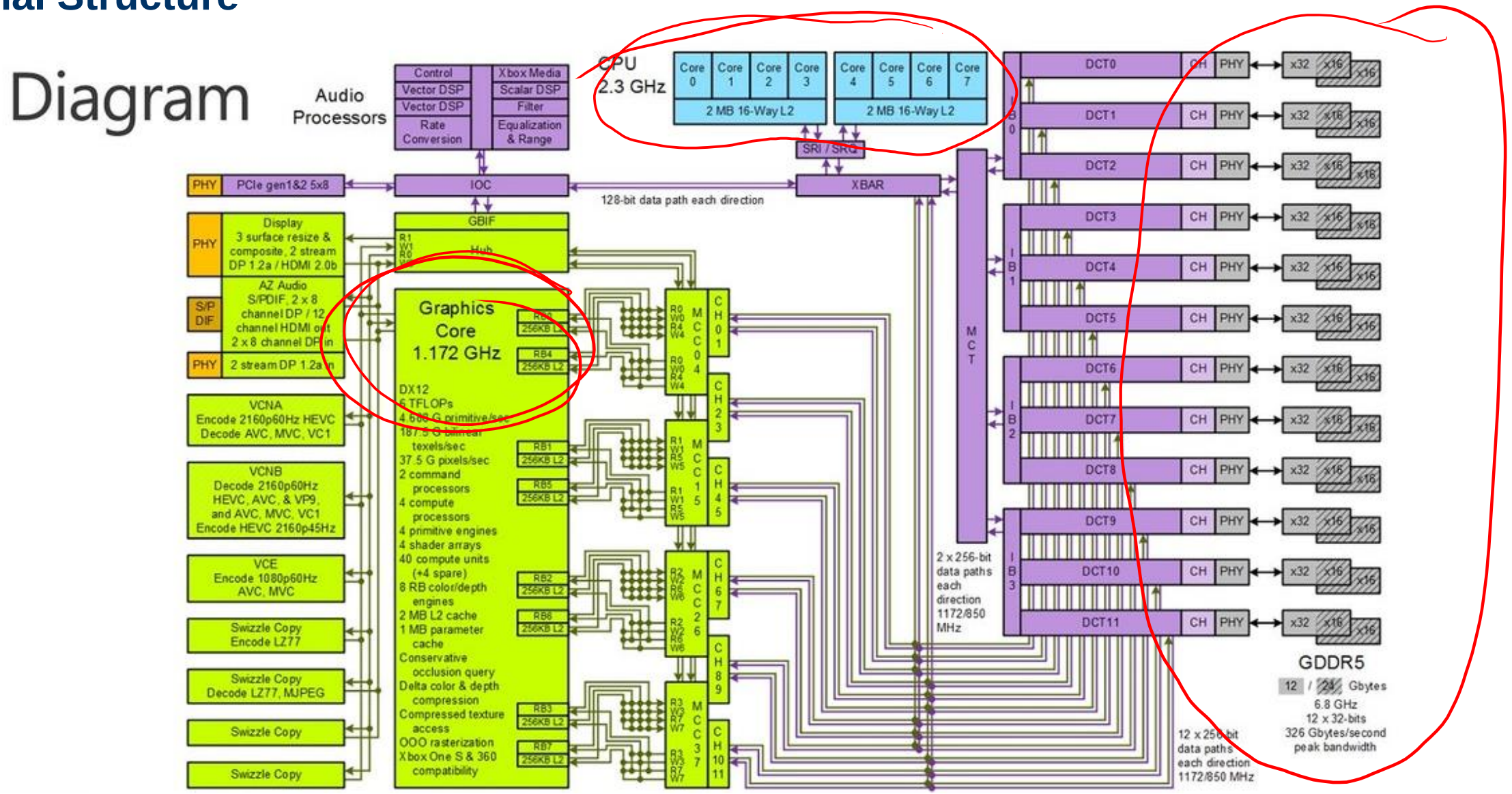
Limits?

Useful?

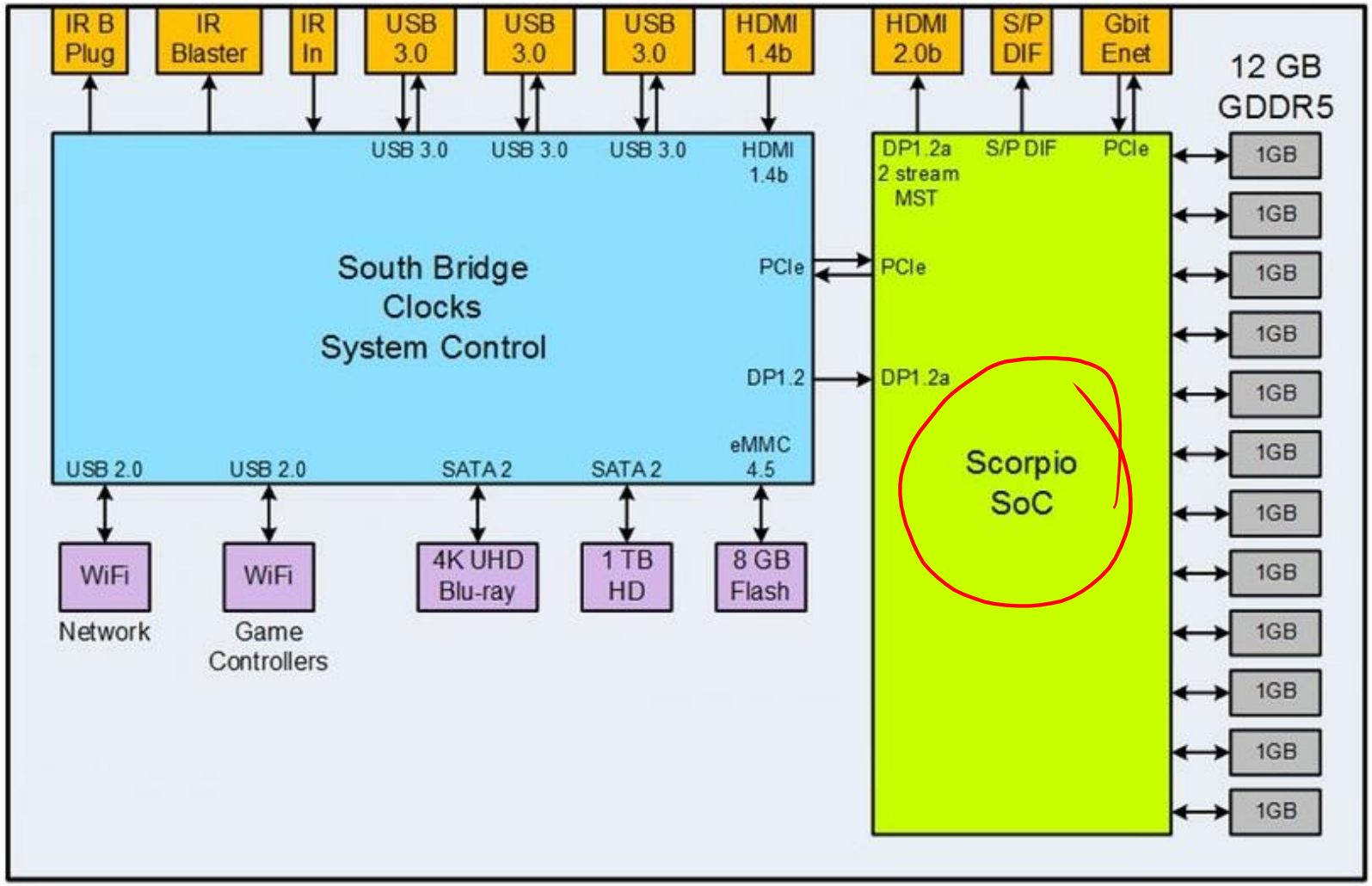
How to go even faster?

Yet another Example: High Performance for 500€

Internal Structure



Many Interfaces



Totally confused? At the end of the course you will understand the basic principles even behind these state-of-the art systems!

WHAT IS COMPUTER ARCHITECTURE?

What is Computer Architecture?

Different opinions exist

- Hardware structure, components, interfaces
- Basic operation principle, applications
- Only external view
- Internal and external view
- …

Computer architecture is NOT (only) standard PC architecture!

- The vast majority of computers are embedded systems, specialized solutions
- One size does NOT fit it all

➡ Should be: Computer Architecture**s**

Amdahl, Blaauw and Brooks 1967

“Computer architecture is defined as the attributes and behavior of a computer as seen by a machine-language programmer. This definition includes the instruction set, instruction formats, operation codes, addressing modes, and all registers and memory locations that may be directly manipulated by a machine language programmer.

Implementation is defined as the actual hardware structure, logic design, and data path organization of a particular embodiment of the architecture.“

Another view: processor architecture

The processor architecture (Instruction Set Architecture) comprises the description of attributes and functions of a system as seen from a machine language programmer's point of view.

The specification of the processor architecture comprises:

- instruction set
- instruction formats
- addressing modes
- interrupt handling
- logical address space
- Register/memory model (as far as a programmer can access it)

The processor architecture does not describe details of the implementation or hardware – all internal operations and components are explicitly excluded.

From the architects of the DEC Alpha microprocessor 1992

„Thus, the architecture is a document that describes the behavior of all possible implementations; an implementation is typically a single computer chip.

The architecture and software written to the architecture are intended to last several decades, while individual implementations will have much shorter lifetimes.

The architecture must therefore carefully describe the behavior that a machine-language programmer sees, but must not describe the means by which a particular implementation achieves that behavior.“

Processor micro architecture

An **implementation** (micro architecture) describes the hardware structure, all data paths, the internal logic etc. of a certain realization of the processor architecture, thus a real microprocessor.

The **micro architecture** defines:

- Number and stages of pipelines
- Usage of super scalar technology
- Number of internal functional units (ALUs)
- Organization of cache memory

The definition of a **processor architecture** (ISA, instruction set architecture) enables the use of programs independent of a certain internal implementation of a microprocessor.

All microprocessors following the same processor architecture specification are called "**binary compatible**" (i.e., the same binaries run on them).

CLASSIFICATION OF COMPUTERS

Classification of Computers: Levels of and techniques for parallelism

How to classify those many different systems, different architectures?

- By hardware? Changes too fast…
- By software? Can run on many systems…
- By size? Price? ???
- Here: by parallelism supported

A parallel program can be seen as a partially ordered set of instructions. The order is given by the dependencies among the instructions. Independent instructions can be executed in parallel.

Levels of parallelism: within a program/a set of programs

Techniques for parallelism: implemented in hardware

Classification of Computers: 5 levels of parallelism

Program level

Process level (or task level)
- Tasks (heavy weighted processes, coarse-grained tasks)

Block level
- Threads, Light-Weighted Processes

Instruction level
- Basic instructions (of the chosen language, of the instruction set)

Sub-operation level
- Basic instructions of a language/an instruction set can be divided into sub-operations/micro operations by a compiler/the processor (e.g., Pentium 4 uses micro instructions internally)

Level of parallelism and granularity

The **granularity** depends on the relation of computation effort over communication or synchronization effort

Typically, program, process, and thread level are considered as **large grained** parallelism

Thus, instruction and sub-operation level may support
fine grained parallelism

Sometimes, the block or thread level is considered as medium grained

PERFORMANCE ASSESSMENT OF COMPUTER SYSTEMS

Performance assessment of computer systems

Needed for:

- Selection of a new computer system
- Tuning of an existing system
- Design of new computer systems

Methods for performance assessment

- Evaluation of hardware parameters
- Run-time measurements of existing programs
- Monitoring during operation of real computer systems
- Model theoretic approaches

Methods for performance assessment

Performance parameters
- Hypothetical maximum performance
- MIPS or MOPS (Millions of Instructions/Operations per Second)
- MFLOPS (Millions of Floating Point Operations per Second, today Tera FLOPS, TFLOPS ~10^{12})
- MLIPS (Millions of Logical Inferences per Second)

Mixes
- Theoretical mix of operations

Benchmark programs
- Number crunching, office applications, …

Monitors / measurement during operation
- Hardware monitor
- Software monitor

Model theoretic approaches
- Analytical methods
- Software simulation

Mixes

Calculation of an assumed average run-time based on the durations of n operations according to the formula:

$$T = \sum_{i=1}^{n} p_i t_i$$

T	average duration
n	number of distinct operations
t_i	duration of operation i
p_i	relative weight of operation i (i.e., relative number of appearances)

The following must hold:

$$\sum_{i=1}^{n} p_i = 1 \quad \text{and} \quad p_i \geq 0$$

Benchmark programs

Sieve of Erathostenes, Ackermann function

Whetstone (1970, typ. Fortran, lots of floating point arithmetic)

Dhrystone (begin of 80s, 53% assignments, 32% control statements, 15% function/procedure calls)

Savage-Benchmark (mathematical standard functions)

Basic Linear Algebra Subprograms (BLAS), core of the LINPACK/LAPACK (Linear Algebra Package) software package

Lawrence Livermore Loops (for vectorizing compilers)

SPEC-Benchmark Suite

SPEC benchmarks

SPEC Standard Performance Evaluation Corporation
- since 1989, many vendors, general purpose applications, focuses on calculation speed and throughput (www.spec.org)

Many benchmark suites exist:
- SPEC CPU2006, CPU2017 (Integer, Floating point)
- SPEC MPI2007, OMP2012, ACCEL (High Performance Computing)
- SPEC jvm2008, jms2007, jEnterprise2010, jbb2015 (Java Client/Server)
- SPEC VIRT_SC 2013 (virtualized servers)
- SPEC Cloud_IaaS 2016
- …

“[…] SPEC is once again seeking to encourage those outside of SPEC to assist us in locating applications that could be used in the next CPU-intensive benchmark suite, currently planned to be SPEC CPU2004.” (http://www.spec.org/cpu2004/ , March 2004)

Integer Component of SPEC CPU2017

SPECrate 2017 Integer	SPECspeed 2017 Integer	Language [1]	KLOC [2]	Application Area
500.perlbench_r	600.perlbench_s	C	362	Perl interpreter
502.gcc_r	602.gcc_s	C	1,304	GNU C compiler
505.mcf_r	605.mcf_s	C	3	Route planning
520.omnetpp_r	620.omnetpp_s	C++	134	Discrete Event simulation - computer network
523.xalancbmk_r	623.xalancbmk_s	C++	520	XML to HTML conversion via XSLT
525.x264_r	625.x264_s	C	96	Video compression
531.deepsjeng_r	631.deepsjeng_s	C++	10	Artificial Intelligence: alpha-beta tree search (Chess)
541.leela_r	641.leela_s	C++	21	Artificial Intelligence: Monte Carlo tree search (Go)
548.exchange2_r	648.exchange2_s	Fortran	1	Artificial Intelligence: recursive solution generator (Sudoku)
557.xz_r	657.xz_s	C	33	General data compression

Floating Point Component of SPEC CPU2017

SPECrate 2017 Floating Point	SPECspeed 2017 Floating Point	Language [1]	KLOC [2]	Application Area
503.bwaves_r	603.bwaves_s	Fortran	1	Explosion modeling
507.cactuBSSN_r	607.cactuBSSN_s	C++, C, Fortran	257	Physics: relativity
508.namd_r		C++	8	Molecular dynamics
510.parest_r		C++	427	Biomedical imaging: optical tomography with finite elements
511.povray_r		C++, C	170	Ray tracing
519.lbm_r	619.lbm_s	C	1	Fluid dynamics
521.wrf_r	621.wrf_s	Fortran, C	991	Weather forecasting
526.blender_r		C++, C	1,577	3D rendering and animation
527.cam4_r	627.cam4_s	Fortran, C	407	Atmosphere modeling
	628.pop2_s	Fortran, C	338	Wide-scale ocean modeling (climate level)
538.imagick_r	638.imagick_s	C	259	Image manipulation
544.nab_r	644.nab_s	C	24	Molecular dynamics
549.fotonik3d_r	649.fotonik3d_s	Fortran	14	Computational Electromagnetics
554.roms_r	654.roms_s	Fortran	210	Regional ocean modeling

[1] For multi-language benchmarks, the first one listed determines library and link options (details)

[2] KLOC = line count (including comments/whitespace) for source files used in a build / 1000

See https://www.spec.org/cpu2017/results/cpu2017.html for results

www.top500.org

Nov. 2014

RANK	SITE
1	National Super Computer Center in Guangzhou China
2	DOE/SC/Oak Ridge National Laboratory United States
3	DOE/NNSA/LLNL United States
4	RIKEN Advanced Institute for Computational Science (AICS) Japan
5	DOE/SC/Argonne National Laboratory United States
6	Swiss National Supercomputing Centre (CSCS) Switzerland
7	Texas Advanced Computing Center/Univ. of Texas United States
8	Forschungszentrum Juelich (FZJ) Germany
9	DOE/NNSA/LLNL United States
10	Government United States

Juni 2015

RANK	SITE
1	National Super Computer Center in Guangzhou China
2	DOE/SC/Oak Ridge National Laboratory United States
3	DOE/NNSA/LLNL United States
4	RIKEN Advanced Institute for Computational Science (AICS) Japan
5	DOE/SC/Argonne National Laboratory United States
6	Swiss National Supercomputing Centre (CSCS) Switzerland
7	King Abdullah University of Science and Technology Saudi Arabia
8	Texas Advanced Computing Center/Univ. of Texas United States
9	Forschungszentrum Juelich (FZJ) Germany
10	DOE/NNSA/LLNL United States

August 2016

Rank	Site
1	National Supercomputing Center in Wuxi China
2	National Super Computer Center in Guangzhou China
3	DOE/SC/Oak Ridge National Laboratory United States
4	DOE/NNSA/LLNL United States
5	RIKEN Advanced Institute for Computational Science (AICS) Japan
6	DOE/SC/Argonne National Laboratory United States
7	DOE/NNSA/LANL/SNL United States
8	Swiss National Supercomputing Centre (CSCS) Switzerland
9	HLRS - Höchstleistungsrechenzentrum Stuttgart Germany
10	King Abdullah University of Science and Technology Saudi Arabia

Juni 2017

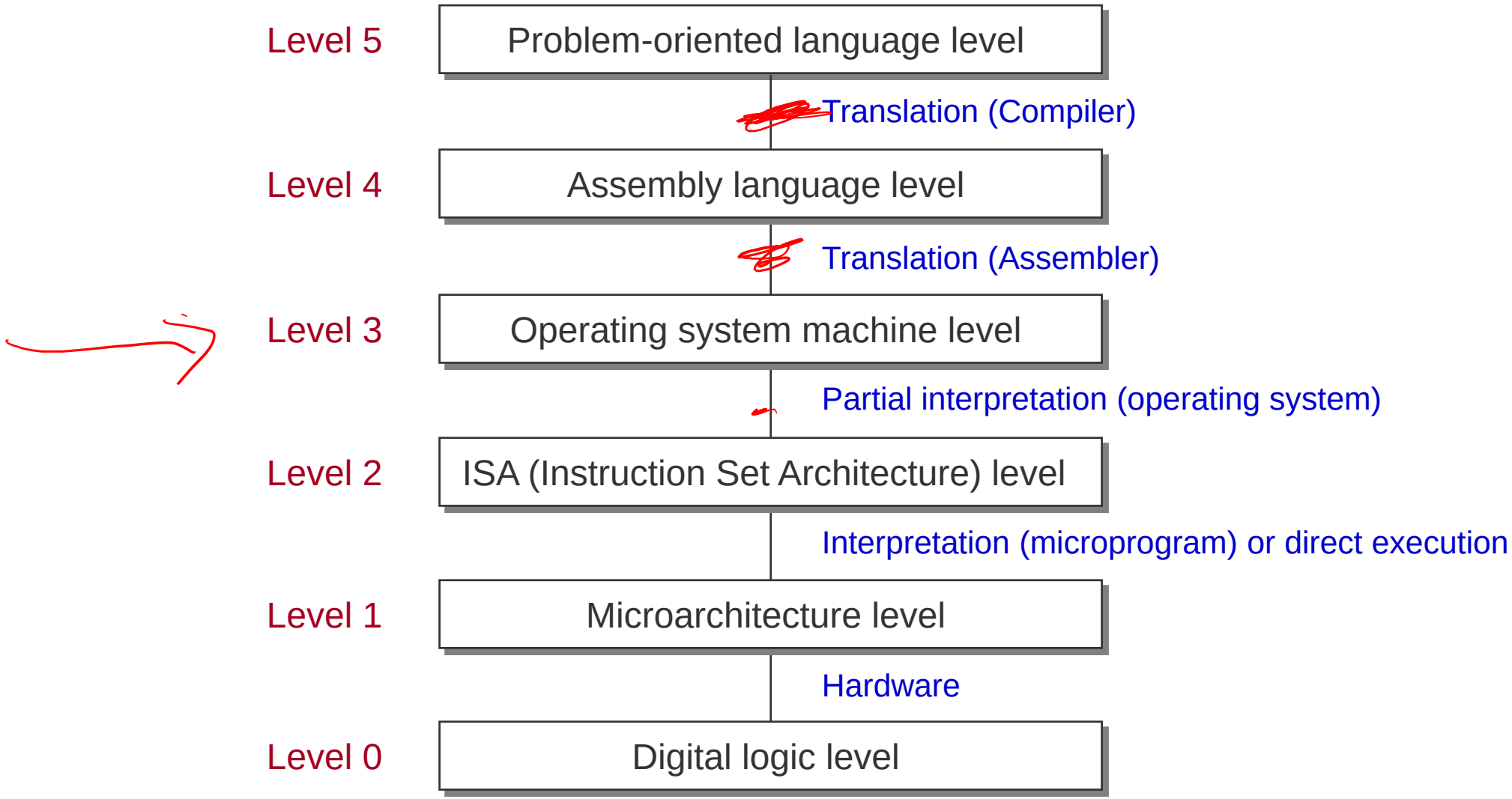
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
4	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
5	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
6	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/SC/LBNL/NERSC United States	622,336	14,014.7	27,880.7	3,939
7	**Oakforest-PACS** - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path , Fujitsu Joint Center for Advanced High Performance Computing Japan	556,104	13,554.6	24,913.5	2,719
8	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	11,280.4	12,660
9	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom , IBM	786,432	8,586.6	10,066.3	3,945

www.top500.org – June 2019

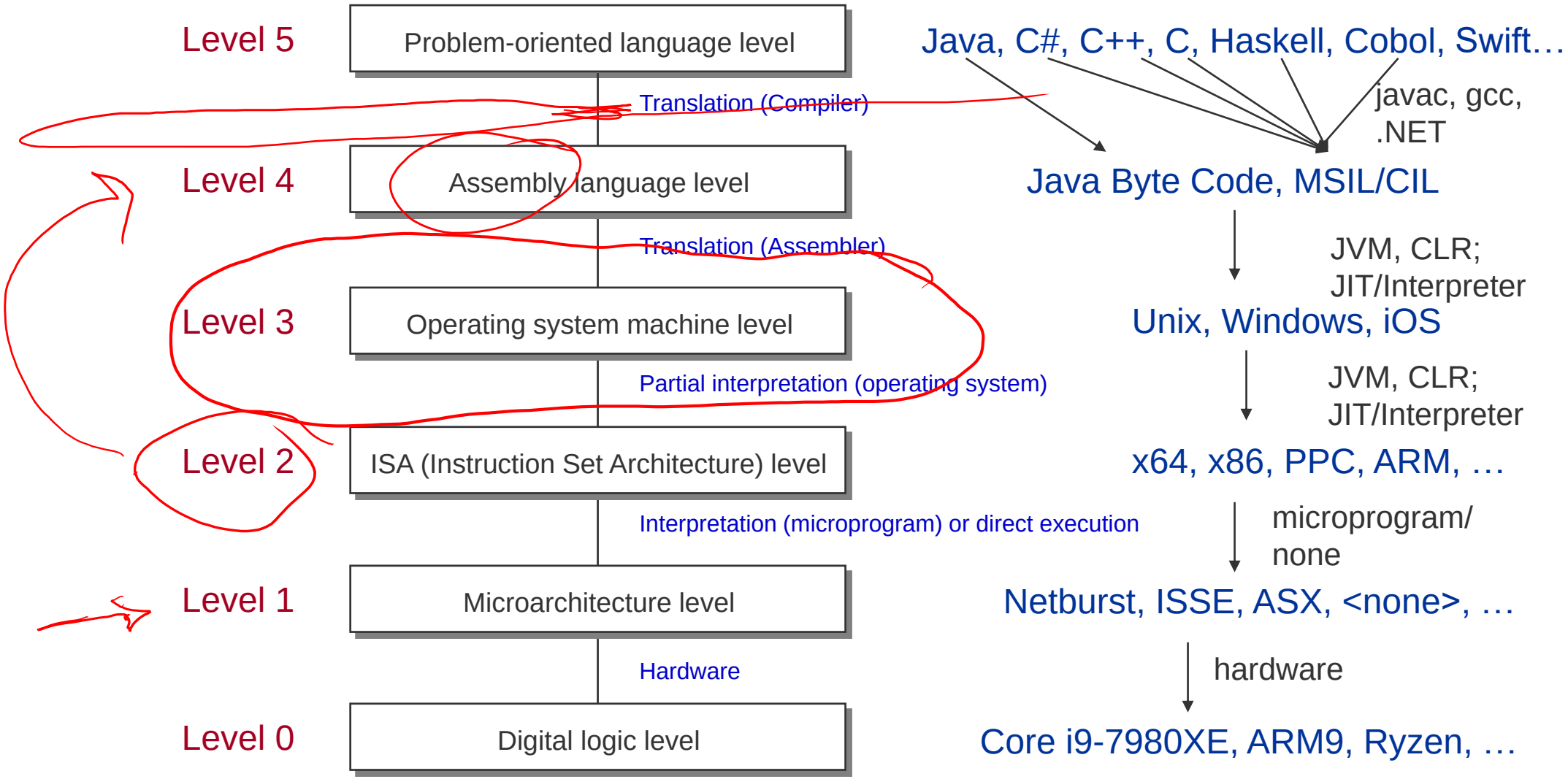
Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	DOE/SC/Oak Ridge National Laboratory United States	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	2,414,592	148,600.0	200,794.9	10,096
2	DOE/NNSA/LLNL United States	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM / NVIDIA / Mellanox	1,572,480	94,640.0	125,712.0	7,438
3	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
4	National Super Computer Center in Guangzhou China	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 NUDT	4,981,760	61,444.5	100,678.7	18,482
5	Texas Advanced Computing Center/Univ. of Texas	**Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz,	448,448	23,516.4	38,745.9	

THE LAYERED COMPUTER MODEL

A Six-level Computer



A Six-level Computer – Examples



Summary

Classical computer architecture
- von Neumann
- Harvard

Universal & special purpose computers

µComputer

Milestones of computer development

A definition of computer architecture

Classification of computers

Performance assessment

Layered computer model