C2115 Practical Introduction to Supercomputing

3rd Lesson

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

National Centre for Biomolecular Research, Faculty of Science
Masaryk University, Kotlářská 2, CZ-61137 Brno

esf european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitivness
UNIVERSITAS MASARYKIANA BRUNENSIS

INVESTMENTS IN EDUCATION DEVELOPMENT

CZ.1.07/2.2.00/15.0233

Contents

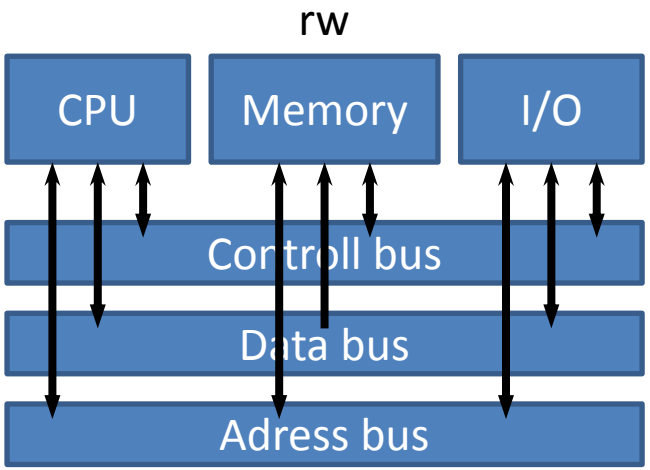
- **Computer Architecture**
  CPU, memory, graphics system, disks, network, pheripherials
- **Numerical values representation in digital devices**
  whole numbers, real numbers
- **From problem to result**
  algorithm, source codes, compilation, program running, programming languages

Computer architecture

- CPU, memory, graphics system, disks, network, pheripherials

Overview

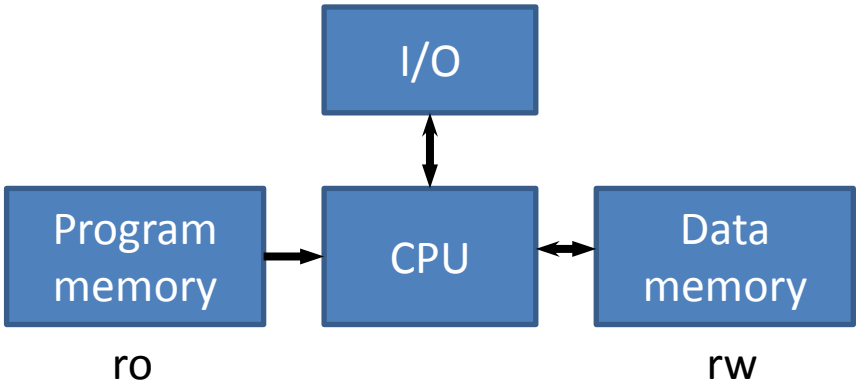
1945 von Neumann architecture



- program may change itself
- program and data can not be loaded simultaneously

John von Neumann, hungarian mathematician, worked in US

1944 Hardward architecture

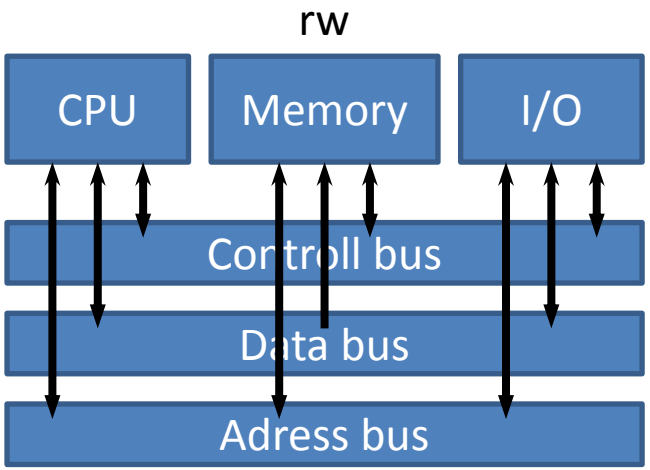


- program may not change itself
- program and data may be loaded simultaneously

Harvard Mark I - completed computer from relay, 24 bit instructions

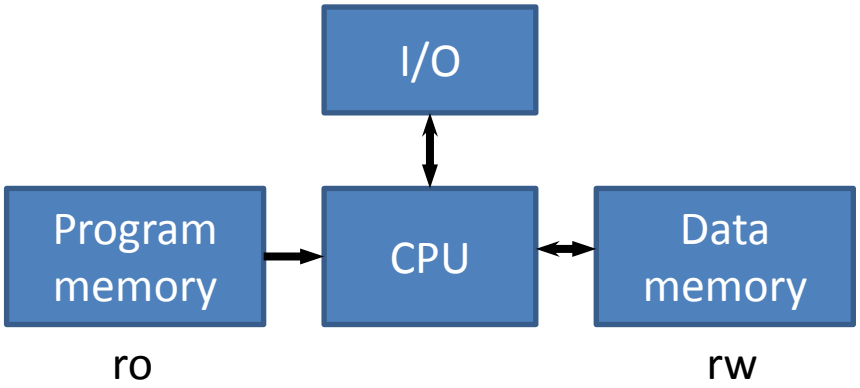
Overview

1945 von Neumann architecture



- program may change itself
- program and data can not be loaded simultaneously

1944 Hardward architecture



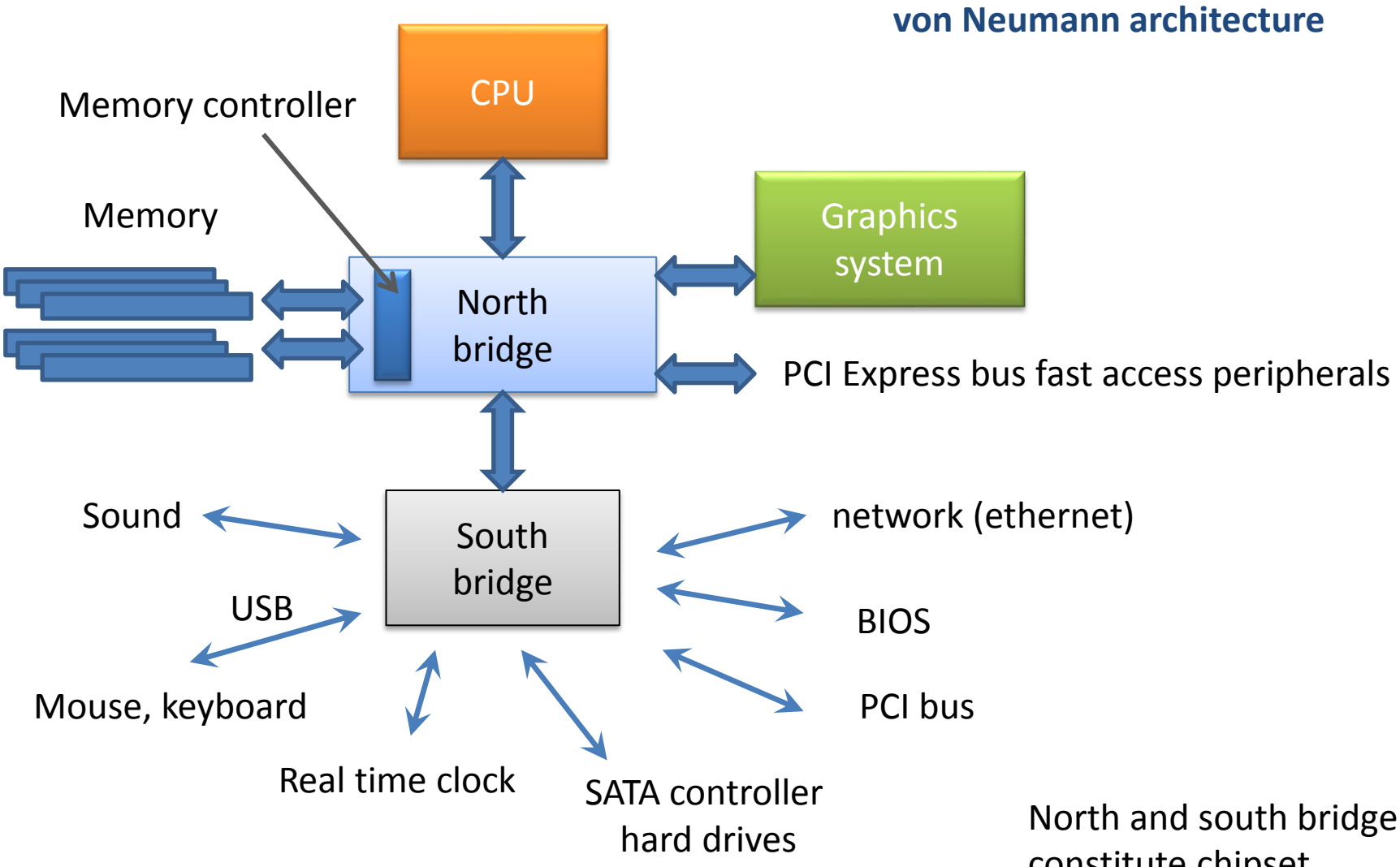
- program may not change itself
- program and data may be loaded simultaneously

Current computers combine both models.

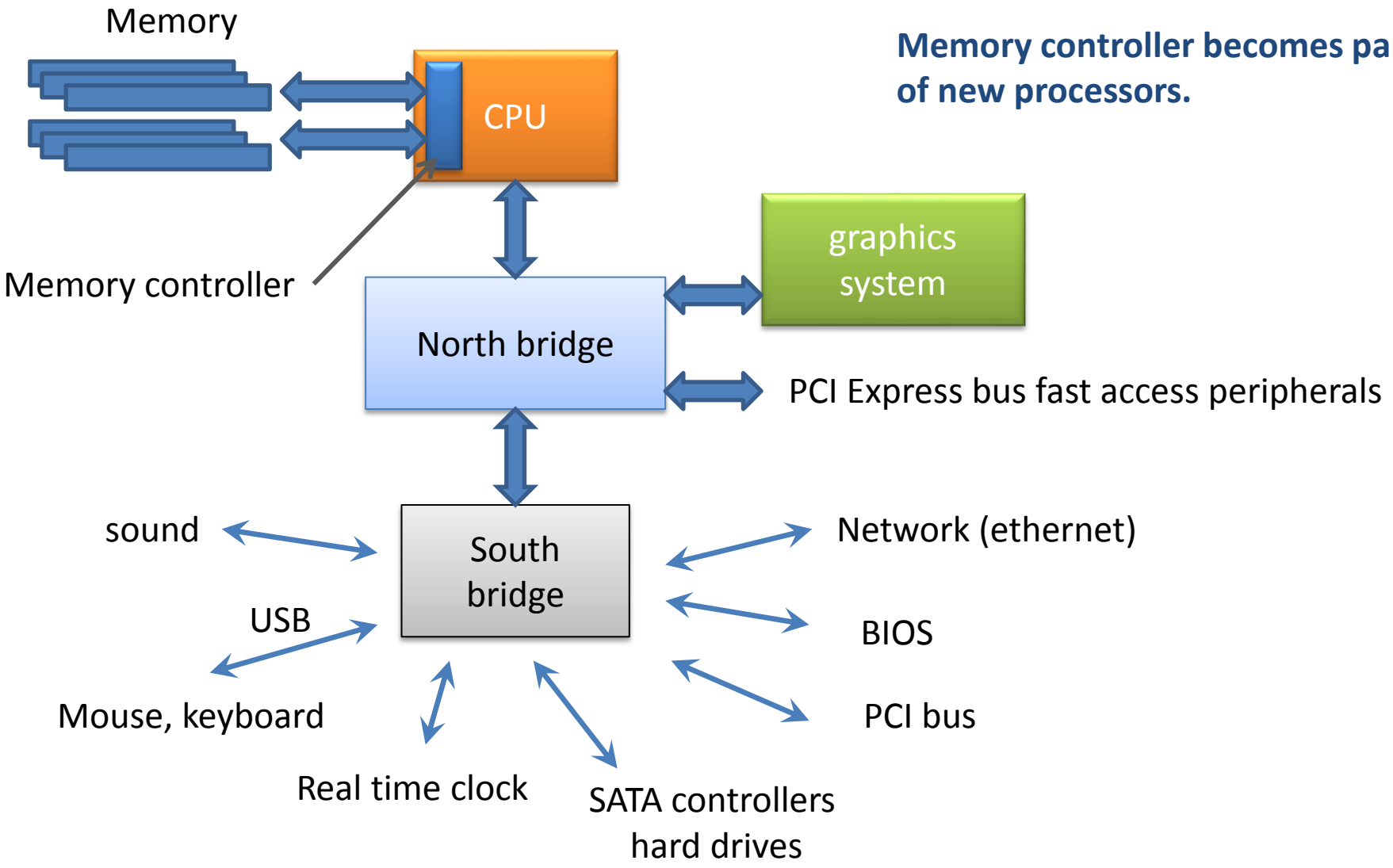
John von Neumann, hungarian mathematician, worked in US

Harvard Mark I - completed computer from relay, 24 bit instructions

Typical computer scheme



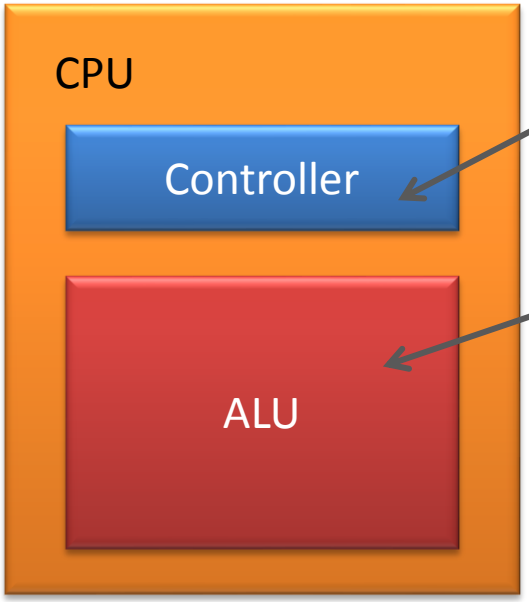
Typical computer scheme, II



CPU

Processor or **CPU** - **Central Processing Unit** is main part of computer; it is complicated sequence circuite that processes **machine code** from operating memory. Machine code consists from machine instructions of computer programs in memory.

www.wikipedia.org



Loads machine instructions and data and prepares their processing in ALU

ALU (arithmetic and logic unit), processes arithmetic and logic operations

Sequence processing of machine instructions is controlled by internal clock tick

Numerical values representation

- Whole numbers
- Real numbers

Whole numbers

Smallest data unit in digital technique is one **bit**. Words are made from bits. Smallest word is **byte**, that contains 8 bits.

One byte may contain numbers from 0 to 255.

128	64	32	16	8	4	2	1	
0	1	0	1	0	1	1	1	= 87

Whole numbers with sign may be expressed too. In such a case one bit is dedicated to sign, remaining bits for number. There are multiple possible implementations. Intel architecture uses **two's complement**, that leads to range from -128 to 127.

128	64	32	16	8	4	2	1		
0	1	1	1	1	1	1	1	=	127
0	1	0	1	0	1	1	1	=	87
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	0	0	=	0
1	1	1	1	1	1	1	1	=	-1
1	0	1	0	1	0	0	0	=	-87
1	0	0	0	0	0	0	0	=	-128

Whole numbers, II

Whole numbers with larger range may be expressed by larger words typically consisting of four bytes (32 bit word) or eight bytes (64 bit word).

32 bit whole number unsigned:	0 to 4.294.967.295
32 bit whole number signed :	−2.147.483.648 to 2.147.483.647
64 bit whole number unsigned :	0 to 18.446.744.073.709.551.615
64 bit whole number signed :	−9.223.372.036.854.775.808 to 9.223.372.036.854.775.807

Working with numbers is limited by range, it is **not possible to express any large number**, **avoid overflow** of the value.

Real numbers

Real numbers are expressed in **floating point** format:

$$X = (-1)^s . (1+Q) . 2^E$$

exponent

mantissa

$$Q = m_1 \frac{1}{2^1} + m_2 \frac{1}{2^2} + m_3 \frac{1}{2^3} + m_4 \frac{1}{2^4} \ldots$$

m_1, m_2, m_3 are mantissa bits

In digital technology real numbers are usually expressed according to standard **IEEE 754**.

type	width	mantissa	exponent
single precision	32	23	8
double precision	64	52	11

Real numbers, II

type	range	Precision
single precision	$\pm 1{,}18\times10^{-38}$ to $\pm 3{,}4\times10^{38}$	Approximately 7 decimal places
double precision	$\pm 2{,}23\times10^{-308}$ to $\pm 1{,}80\times10^{308}$	Approximately 15 decimal places

Special combination of mantissa and exponent **special values** may be expressed:

0	positive zero
-0	negative zero
NaN	not a number, e. g. division by zero
+Inf	positive infinity (number too large to express)
-Inf	negative infinity (negative number too large to express)

Working with real numbers is limited by spread of **rounding error**. Logical compating operators **equal to** and **not equal to are not appropriate** to be used with real numbers (except to comparing with zero).

Exercise LII.1

1. Variable of **signed char** type(byte with sign) contains number 127. What will be walue if we increment value by one?
2. What will be result of sum of two real numbers in double precission with values:

   $0{,}1346978.10^{-12}$

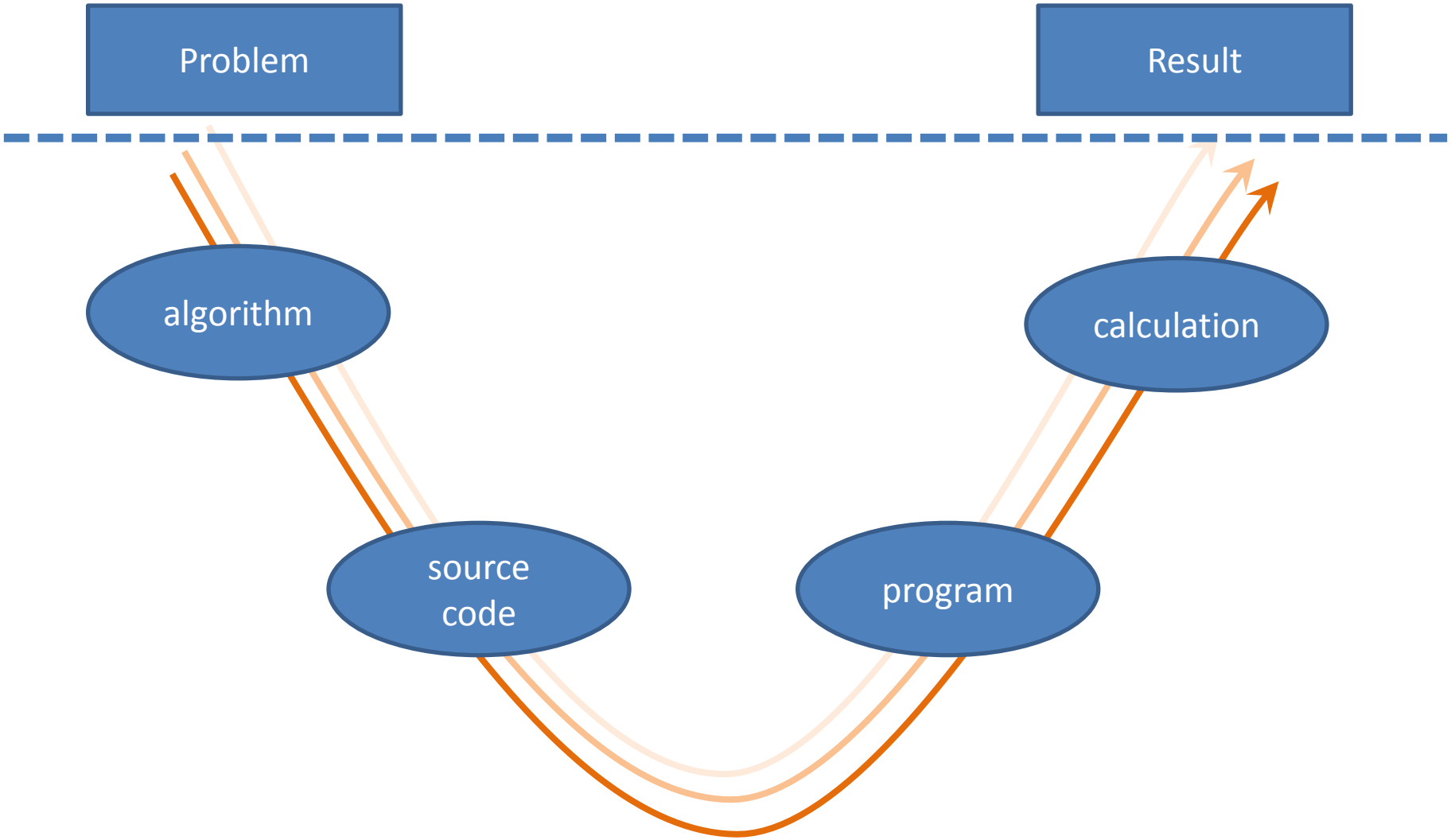
   $1{,}2312657.10^{6}$

From problem to result

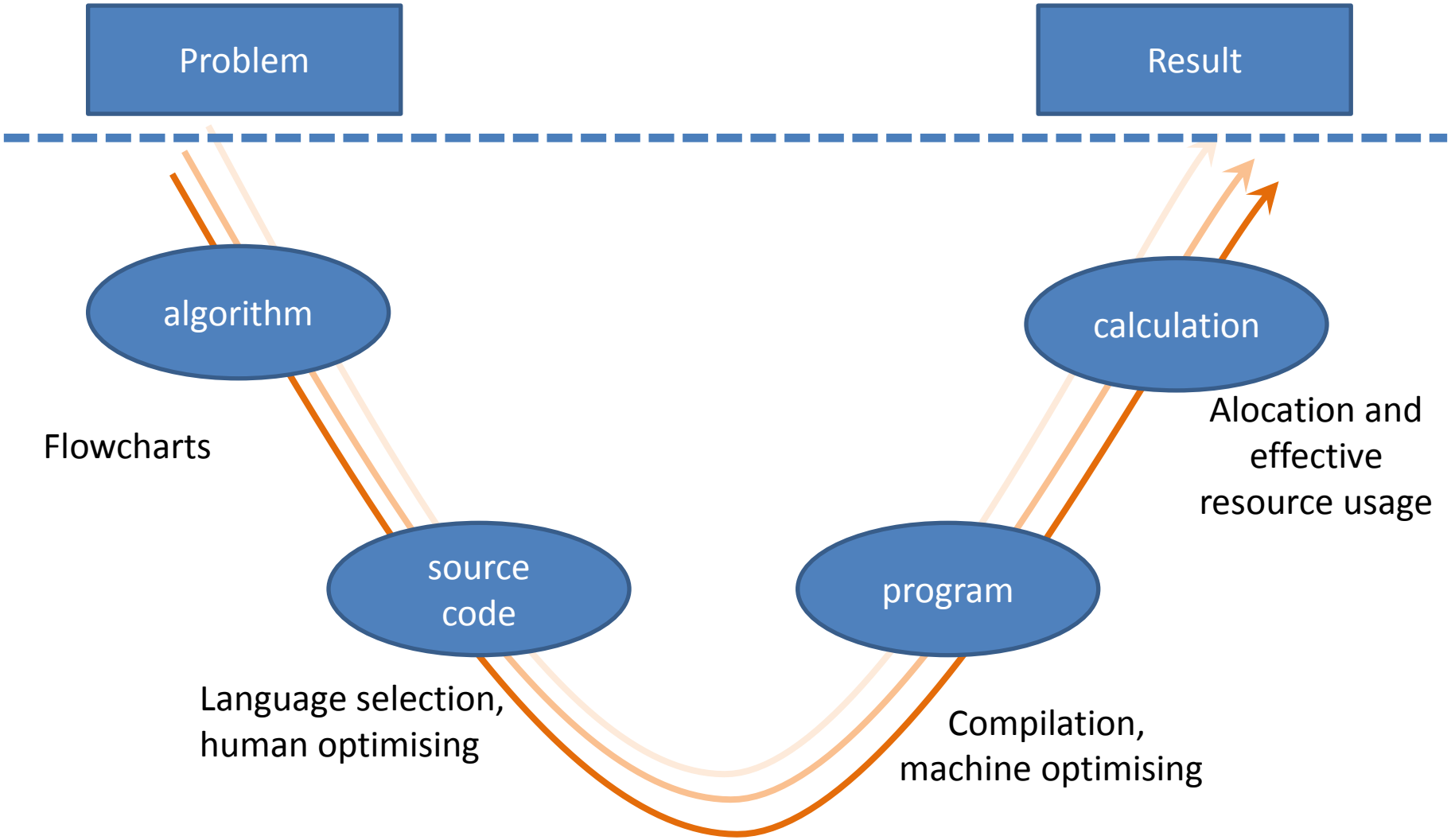
- Algorithm
- Source codes, compilation
- Running program
- Programming languages

From problem to result ...

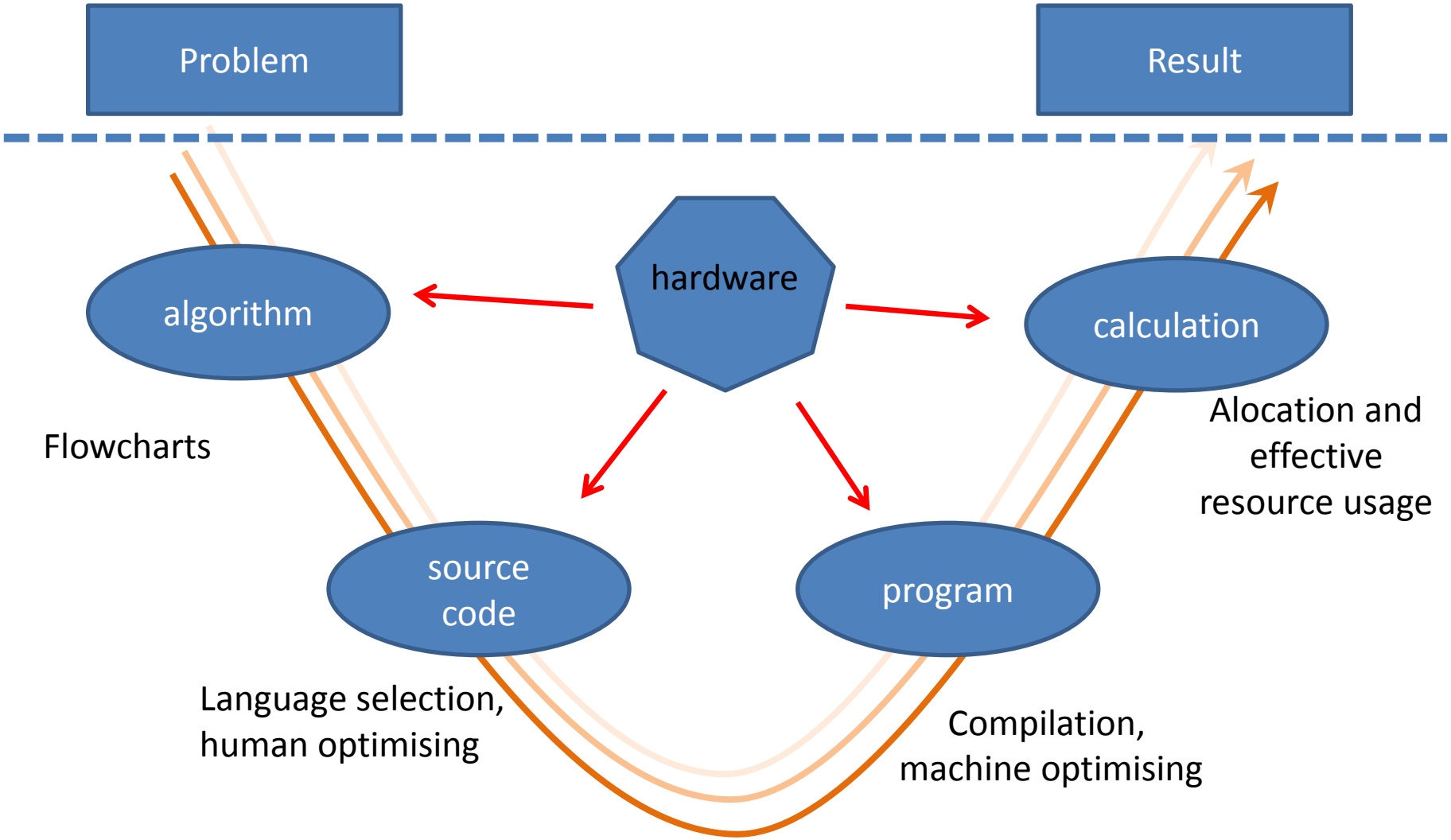
From problem to result ...



From problem to result ...

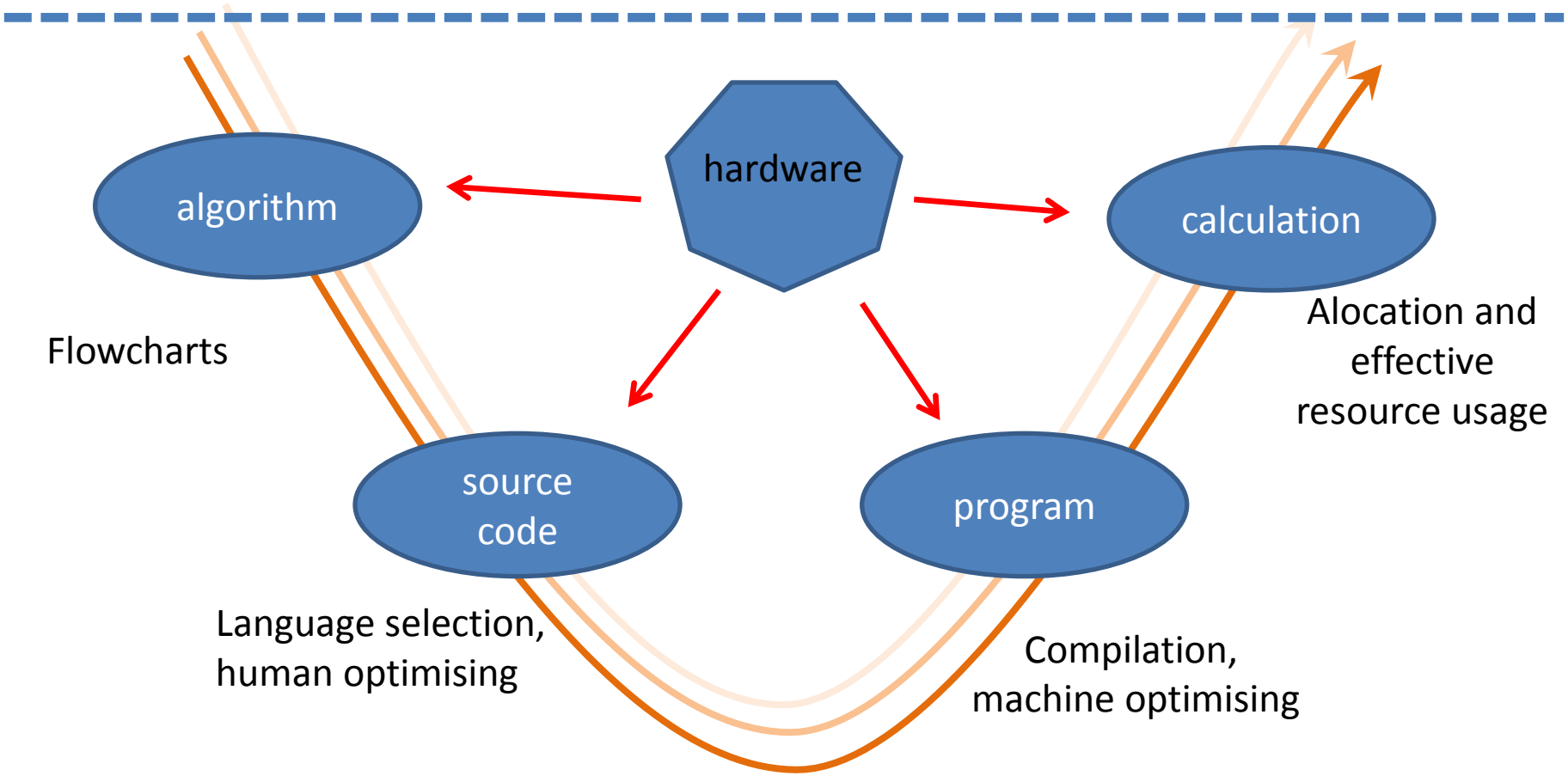


From problem to result ...



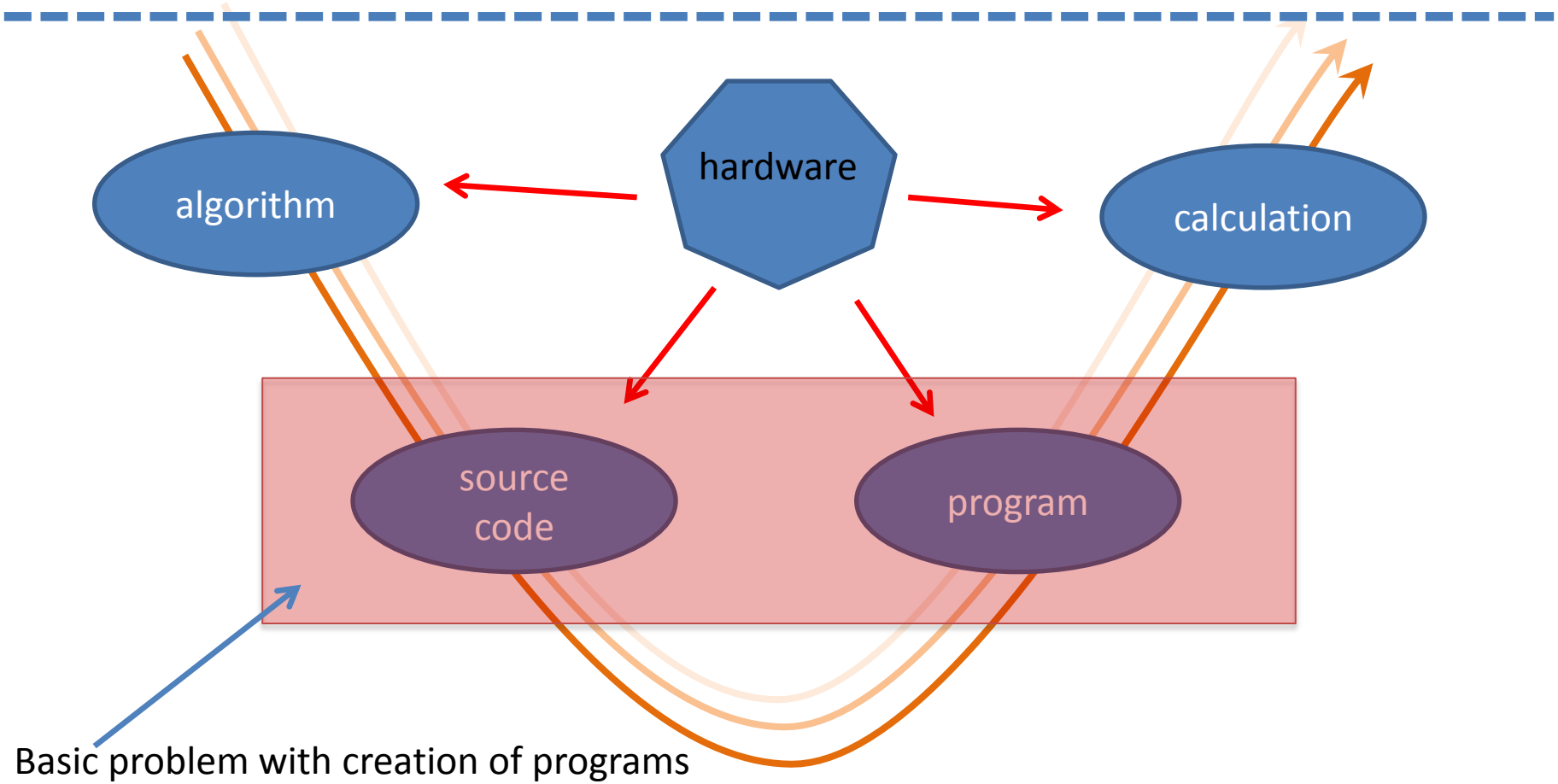
From problem to result ...

Solving problems using computers (supercomputers) it is necessary to **evaluate** number of **aspects** including used hardware and architecture.



Lecture focus ...

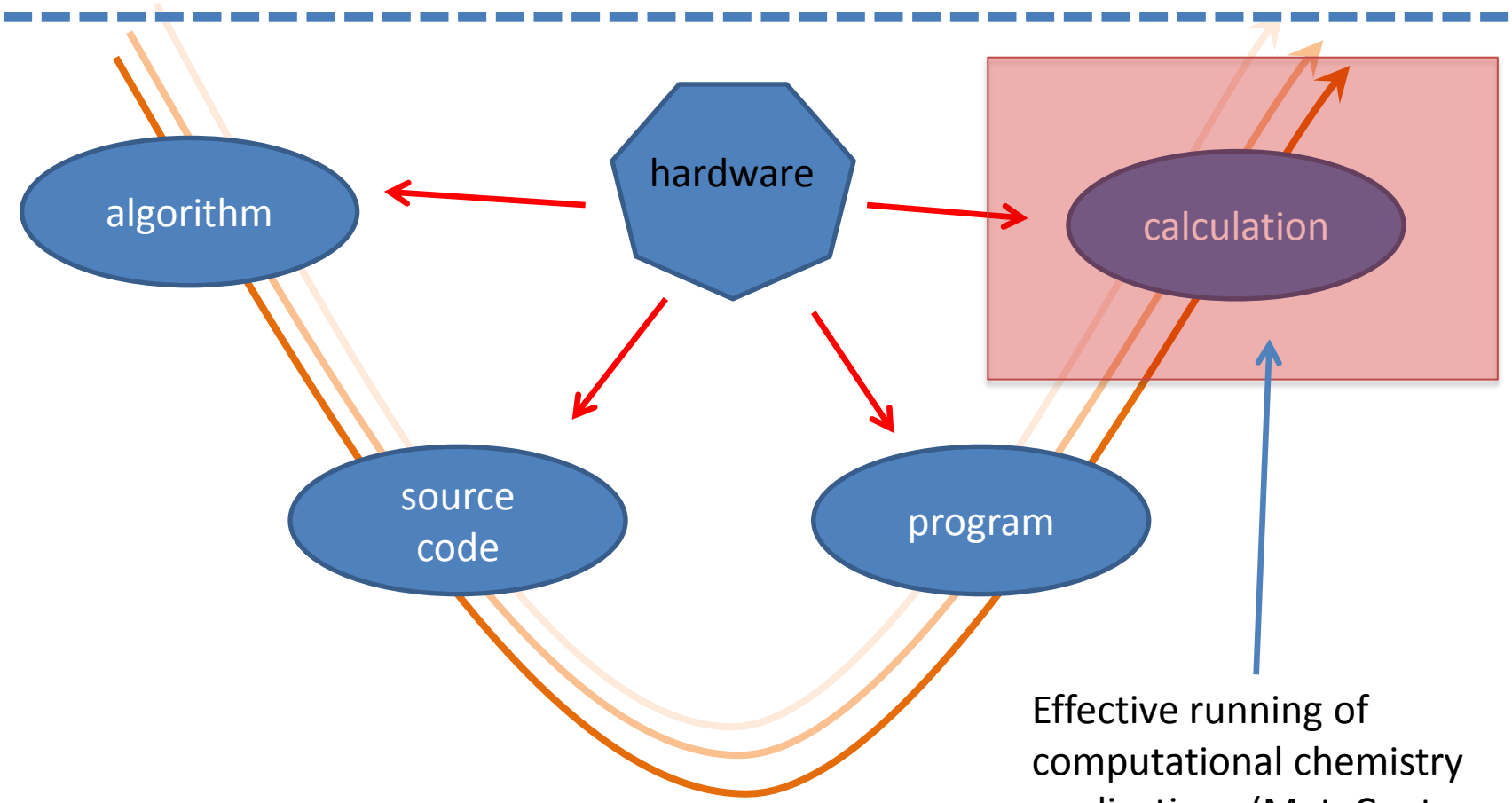
Solving problems using computers (supercomputers) it is necessary to **evaluate** number of **aspects** including used hardware and architecture.



Basic problem with creation of programs adreesing complex problems - parallelisation

Lecture focus ...

Solving problems using computers (supercomputers) it is necessary to **evaluate** number of **aspects** including used hardware and architecture.



Programs *vs.* Scripts

Program is machine instruction file processed directly by processor. It is created by procedure called **compilation** from source code.

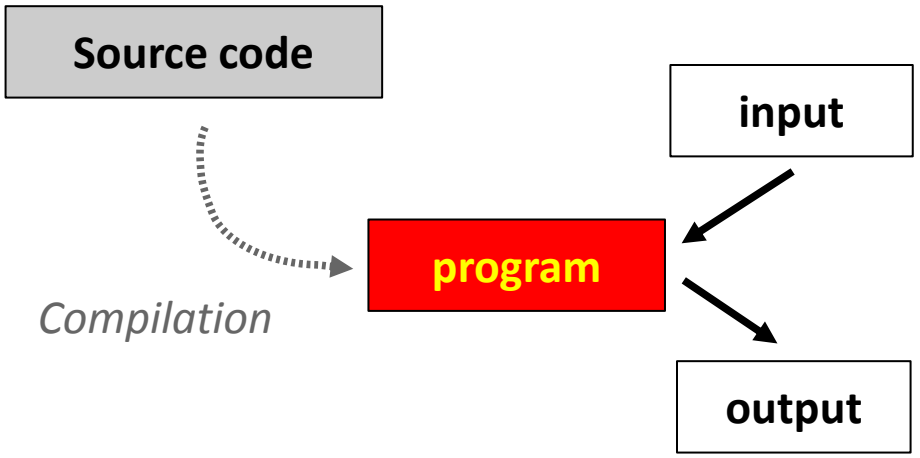
Compiled languages:

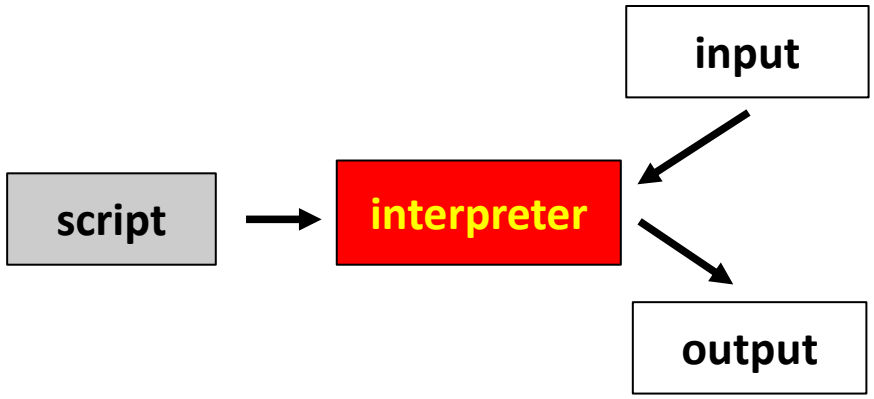
C/C++
Fortran

Script is text file containing commands and special constructions, these are processed by interpreter of scripting language.

Skripting languages:

bash
gnuplot
awk
JavaScript
PHP





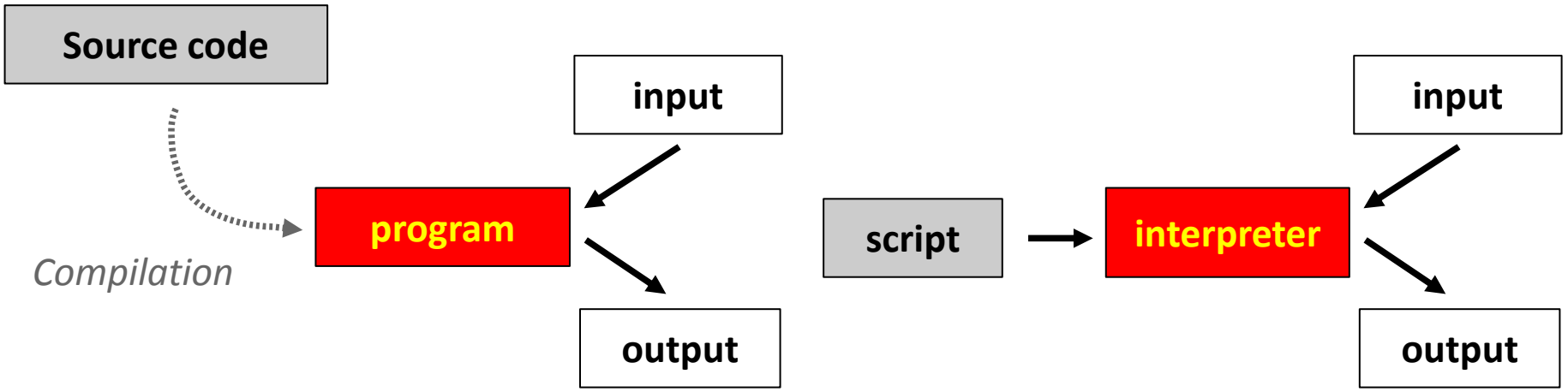
Programs *vs.* Scripts, ...

- Easy optimization
- Fast processing

- Recompilation needed
- Self run code not available

- No recompilation
- Program can generate and run self running code

- Poor optimization
- Slower processing



Programs *vs* Scripts

Programs dedicated to **demanding scientific-technical calculations** are allways written in c**ompiled programming languages.** These include:

- Fortran
- C/C++

Scripting languages are not used in such calculations at all, or **only in supporting parts** of calculation, that are not demanding.

Exercise LII.2

1. Determine rate of programs in languages Fortran, C/C++ and others, that are listed on page:

   http://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid_state_physics_software

   Plot result in sector graph.