Computational Physics – PH 354

Manish Jain

Email: mjain@iisc.ac.in
Office: F0-08

Logistics

- TA: Indrajit Maiti (indrajit@iisc.ac.in)
- Please send an email at this address: ph354.cp@gmail.com to register for getting homeworks via email.
- Course website: http://www.physics.iisc.ernet.in/ mjain/pages/teaching.html
- Homeworks will also be posted there periodically. Please check regularly.
- 1st Homework already posted.

Grading scheme

- No Exams!!!!!
- Homeworks and projects based (50% and 50% tentatively)
- Project will be decided by you in consultation with your Masters/PhD/Bachelors advisor – subject to my approval as well.
- Stand alone project – which you have to get working and submit a working code and report. Also, make a presentation on it at the end of the course.
- Projects can start right away.

Basics of Computational Physics

- What is Computational Physics?
- Basic Computer Hardware
- Software: Programming Languages

What is Computational Physics?

"Computational physics is a synthesis of theoretical analysis, numerical algorithms and computer programming"

— P. L. Deries Am. J. Phys., **64**, 364 (1996)

What is Computational Physics?

"Computational physics is a synthesis of theoretical analysis, numerical algorithms and computer programming"

— P. L. Deries Am. J. Phys., **64**, 364 (1996)

"The purpose of computing is insight, not numbers."

— Richard Hamming

What is Computational Physics?

Computational physics is a tool for solving complex numerical problems in physics.

What does Computational Physics do?

- Atomic Physics studies atoms.

What does Computational Physics do?

- Atomic Physics studies atoms.
- Nuclear Physics studies nuclei.

What does Computational Physics do?

- Atomic Physics studies atoms.
- Nuclear Physics studies nuclei.
- Condensed Matter Physics studies condensed systems (solids).

What does Computational Physics do?

- Atomic Physics studies atoms.
- Nuclear Physics studies nuclei.
- Condensed Matter Physics studies condensed systems (solids).
- Computational Physics *does not* study computers.

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?
- We use equations to create models to describe nature.

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?
- We use equations to create models to describe nature.
- Exact (analytic) solutions of these models is *very rare* unless the model is very simple.

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?
- We use equations to create models to describe nature.
- Exact (analytic) solutions of these models is *very rare* unless the model is very simple.
- Computational Physics is needed whenever:
  - We cannot solve problems analytically!

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?
- We use equations to create models to describe nature.
- Exact (analytic) solutions of these models is *very rare* unless the model is very simple.
- Computational Physics is needed whenever:
  - We cannot solve problems analytically!
  - We have too much data to process!

Why do we need Computational Physics?

- In physics, we try to address the question – how does nature work?
- We use equations to create models to describe nature.
- Exact (analytic) solutions of these models is *very rare* unless the model is very simple.
- Computational Physics is needed whenever:
  - We cannot solve problems analytically!
  - We have too much data to process!

Many, if not most, problems in physics could never be solved without computers!

Computational physics in contemporary physics

- Numerical calculations: Solutions of well defined mathematical problems to produce numerical solutions. Examples: Systems of differential equations, integration, system of linear equations etc.

Computational physics in contemporary physics

- Numerical calculations: Solutions of well defined mathematical problems to produce numerical solutions. Examples: Systems of differential equations, integration, system of linear equations etc.
- Visualization and animation: Presentation of data in a form that is easily understandable: 2D and 3D plots Examples: Animation, using colors and textures.

Computational physics in contemporary physics

- Numerical calculations: Solutions of well defined mathematical problems to produce numerical solutions. Examples: Systems of differential equations, integration, system of linear equations etc.
- Visualization and animation: Presentation of data in a form that is easily understandable: 2D and 3D plots Examples: Animation, using colors and textures.
- Computer simulation: Testing models of nature Examples: Weather forecasting.

Computational physics in contemporary physics

- Numerical calculations: Solutions of well defined mathematical problems to produce numerical solutions.
  Examples: Systems of differential equations, integration, system of linear equations etc.
- Visualization and animation: Presentation of data in a form that is easily understandable: 2D and 3D plots
  Examples: Animation, using colors and textures.
- Computer simulation: Testing models of nature
  Examples: Weather forecasting.
- Data collection and analysis: Processing and analyzing data (small/large amounts of it).
  Examples: Data from telescopic arrays

Computational physics in contemporary physics

- Numerical calculations: Solutions of well defined mathematical problems to produce numerical solutions.
  Examples: Systems of differential equations, integration, system of linear equations etc.
- Visualization and animation: Presentation of data in a form that is easily understandable: 2D and 3D plots
  Examples: Animation, using colors and textures.
- Computer simulation: Testing models of nature
  Examples: Weather forecasting.
- Data collection and analysis: Processing and analyzing data (small/large amounts of it).
  Examples: Data from telescopic arrays
- Symbolic manipulation: Computer Algebra Systems.
  Examples: Mathematica, SymPy etc.

Computer Simulations (examples)

- Electronic structure of materials

Computer Simulations (examples)

- Electronic structure of materials
- Molecular dynamics

Computer Simulations (examples)

- Electronic structure of materials
- Molecular dynamics
- Weather forecast

Computer Simulations (examples)

- Electronic structure of materials
- Molecular dynamics
- Weather forecast
- N-body simulations (cosmology) – Millennium Simulation

Computer Simulations (examples)

- Electronic structure of materials
- Molecular dynamics
- Weather forecast
- N-body simulations (cosmology) – Millennium Simulation
- ...

Computer Hardware

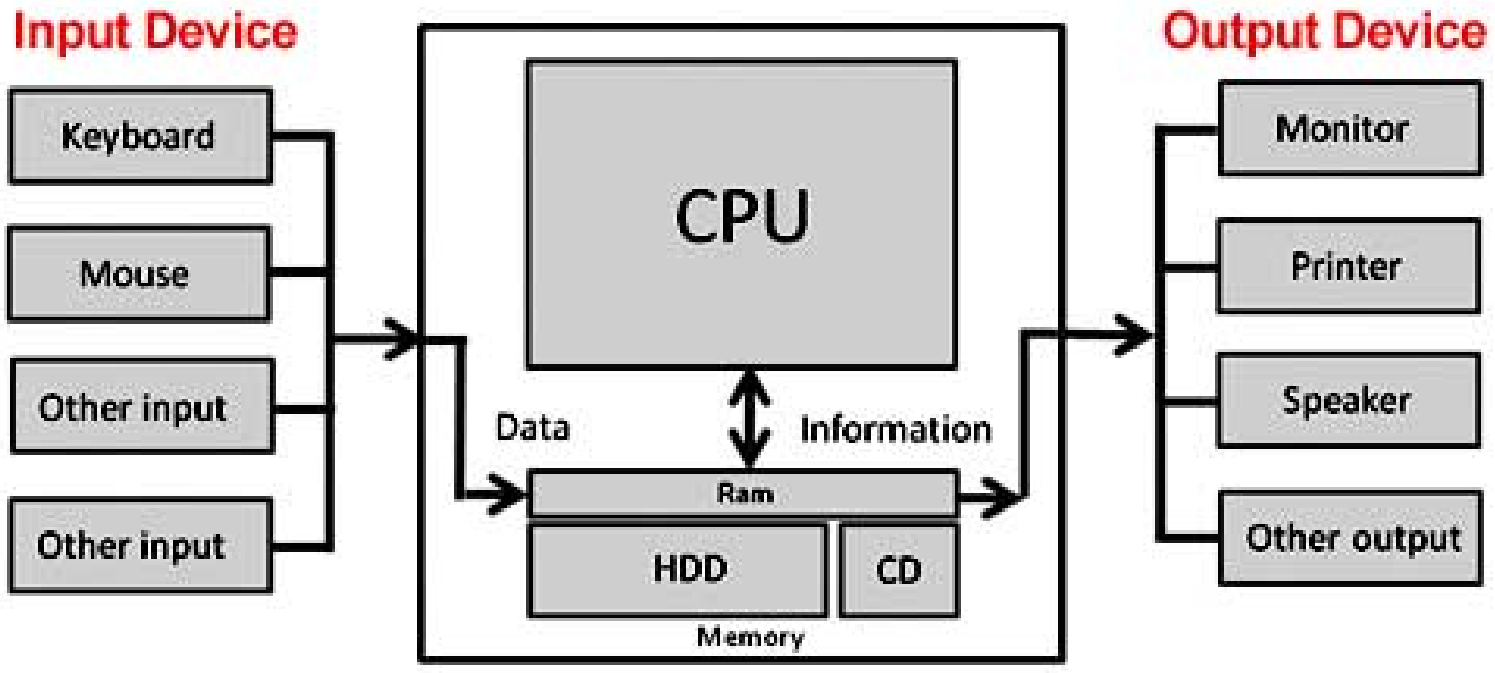
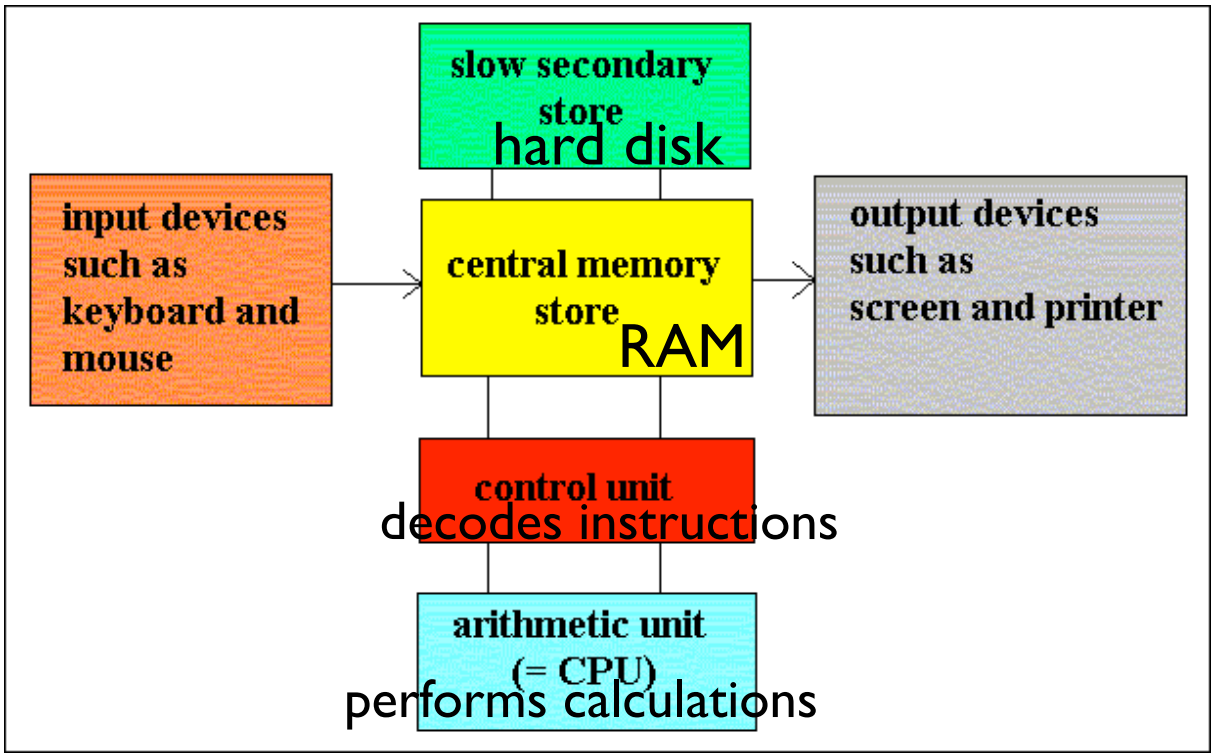


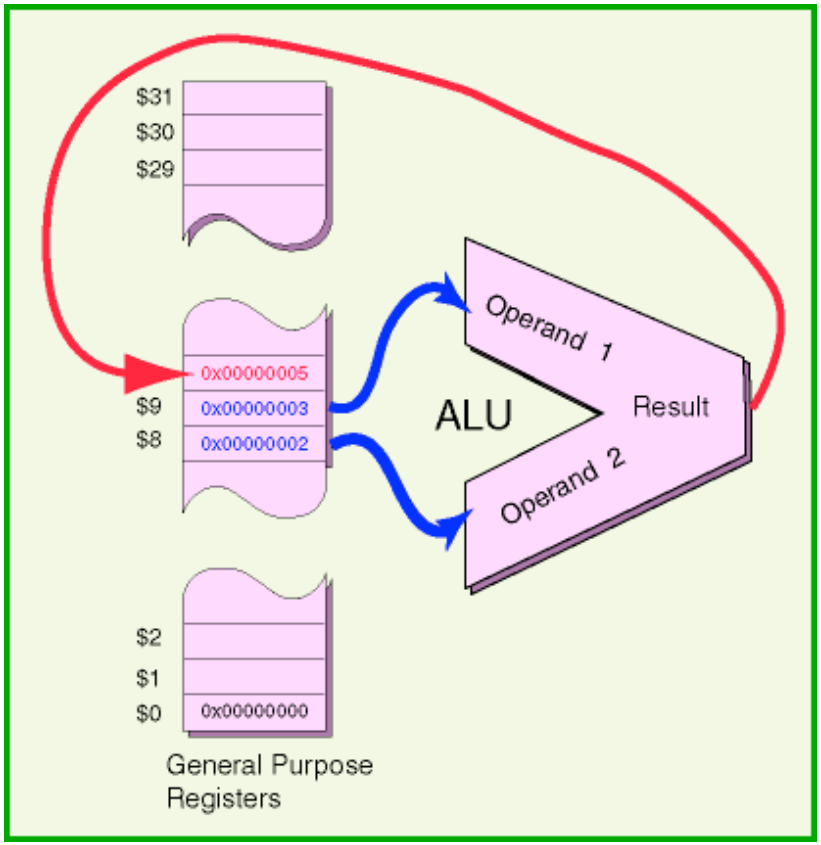
Image:http://tipsboss.com/basic-computer-tips-for-beginners/

Computer Architecture



both instructions & data sent by input devices to memory
loaded from memory to CPU registers
Instruction Set Architecture (ISA): machine language instruction set, word size, registers

ALU



bitwise logic ops.
AND, OR, NOT,XOR

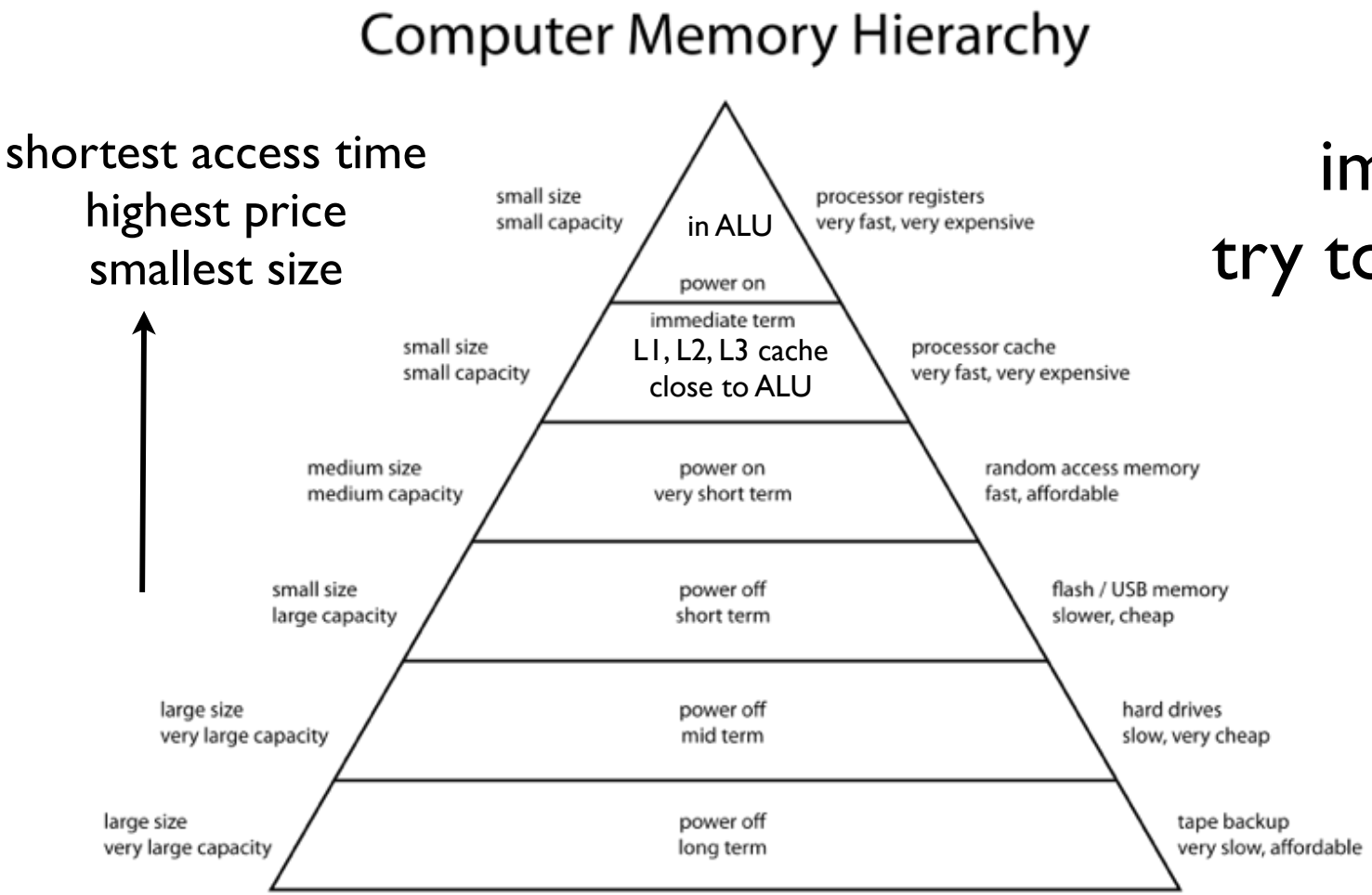
integer arithmetic ops.
add,subtract,multiply,divide

bit shifting (* or / by 2^n)

FPU: floating point unit

+,-,* fast / slow and so are exp, cos, & other transcendental fns.
commonly used function are coded in machine language

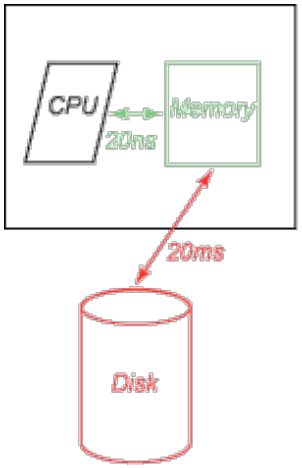
Hierarchical Memory



implication:
try to reuse cache

Cache Utilization

data stored in memory as a 1-D array



column major in fortran

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix} = [\,1\ 6\ 2\ 5\ 3\ 4\,]$$

row major in C!

```
do j = 1, jmax
do i = 1, imax
  a(i,j) = float(i-j)/float(i+j)
enddo
enddo
```

faster than

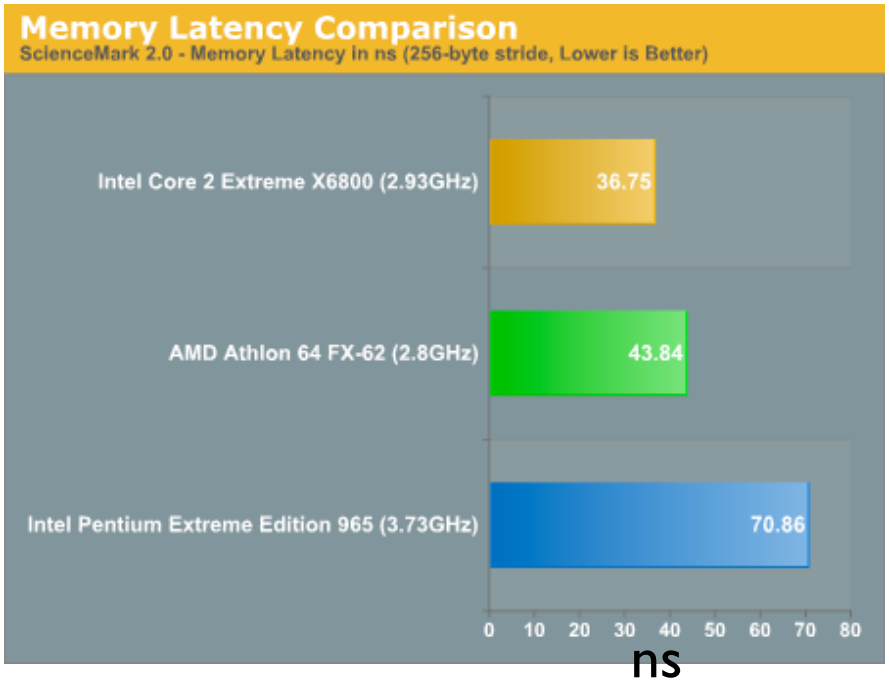
```
do i = 1, imax
do j = 1, jmax
  a(i,j) = float(i-j)/float(i+j)
enddo
enddo
```

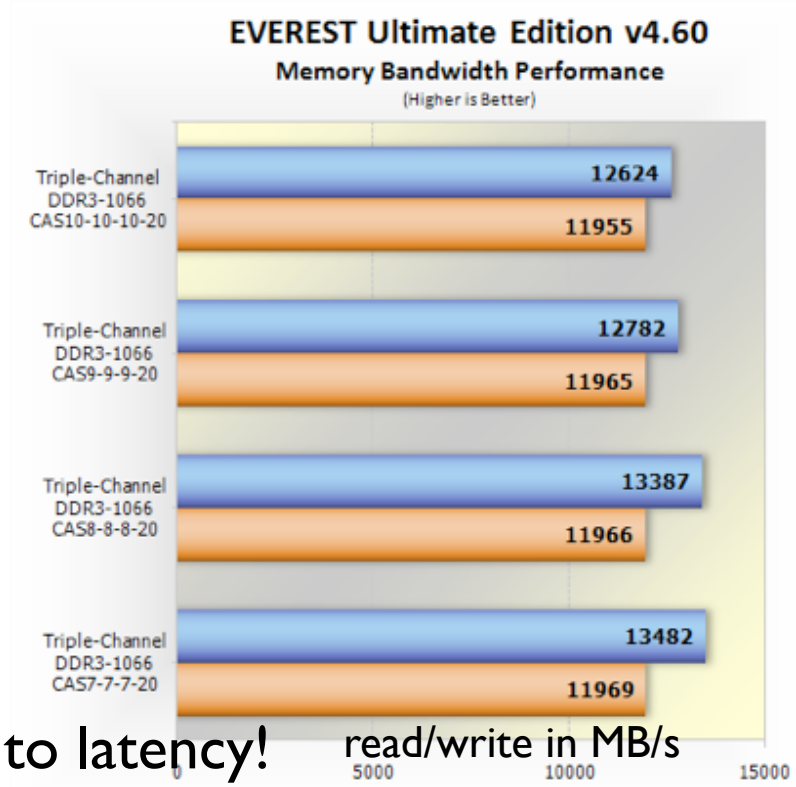
sometimes compilers do these optimizations (-O3)

Latency & Bandwidth

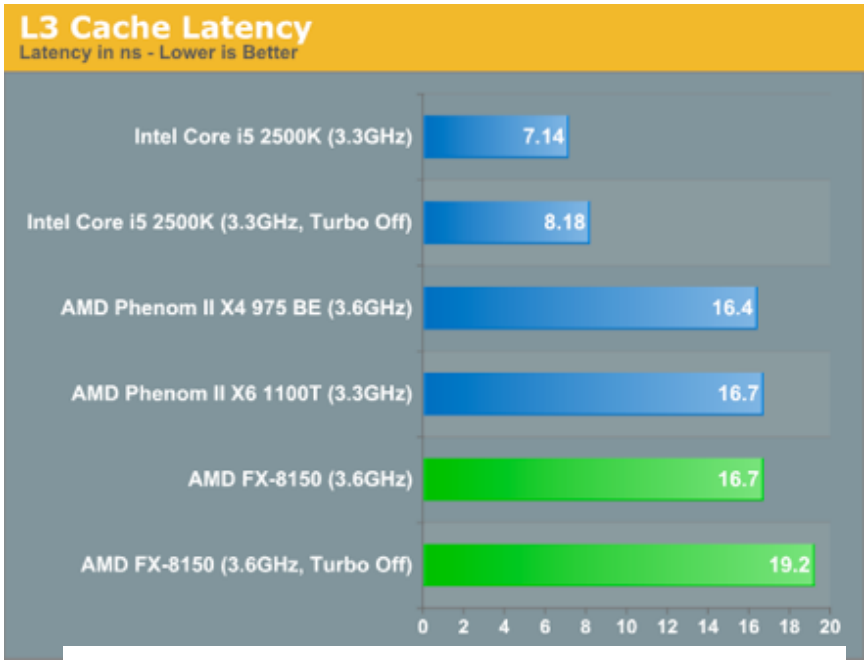
minimum time to do an action
(access time)

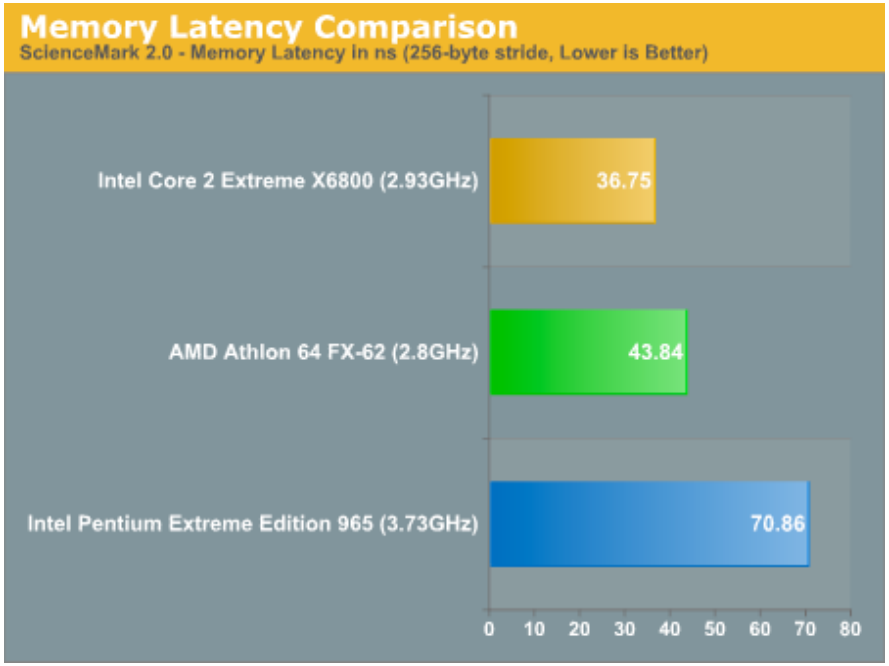
rate of action once action is initialized

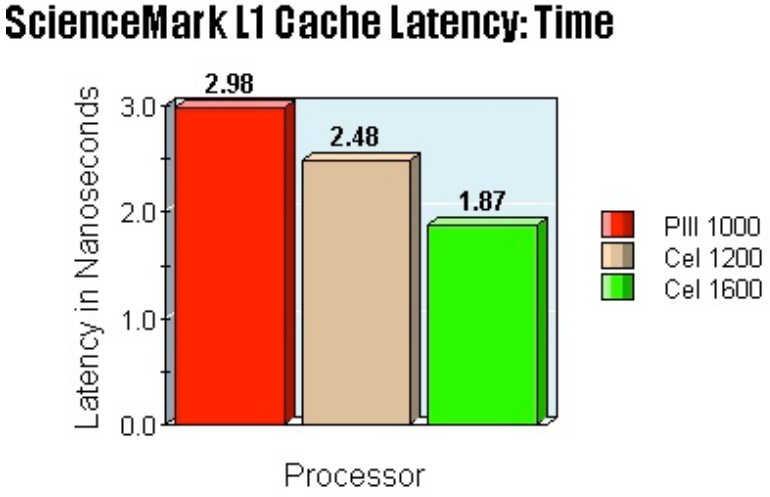




better to minimize memory access due to latency!



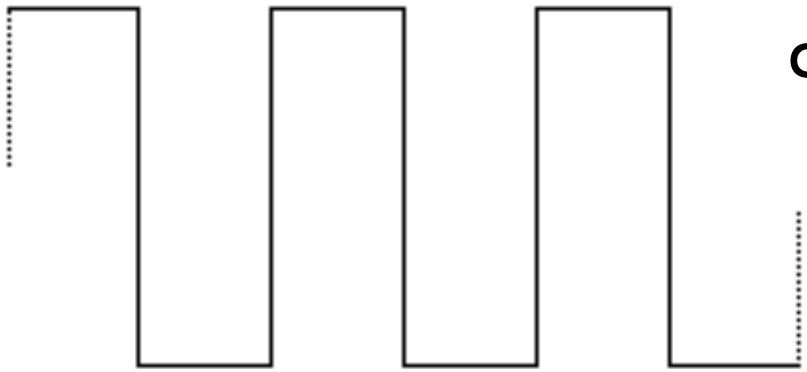




Nehalem processor:
L1~64 kB
L2~2 MB
L3~30 MB

L1 cache ~ 5 times faster than L3 cache ~5 times faster than RAM!

Clock Rate



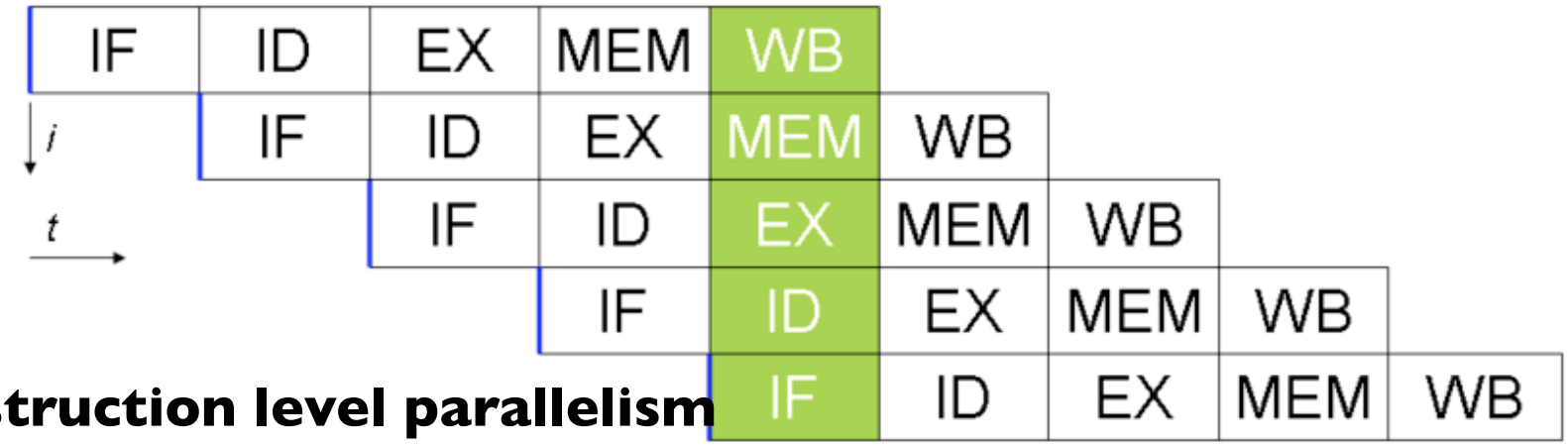
clock coordinates
different actions

modern CPUs upto 4 FLOPs per cycle:
2.4 GHz => 4x2.4 10^9 ~10^{10} FLOPs/cycle/core (10 GF)
if the cluster has 80 cores => 800 GF machine

this is not the only parameter! since data access is more time-consuming (40 ns) than FLOPs (0.1 ns); having larger RAM/ cache/interconnects more important than just clock speed

Architecture level Parallelism

bit level parallelism: 4 bit ... 32 to 64 bit word-size (=register size);
more bits processed/cycle



five-stage pipeline in a RISC (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back)

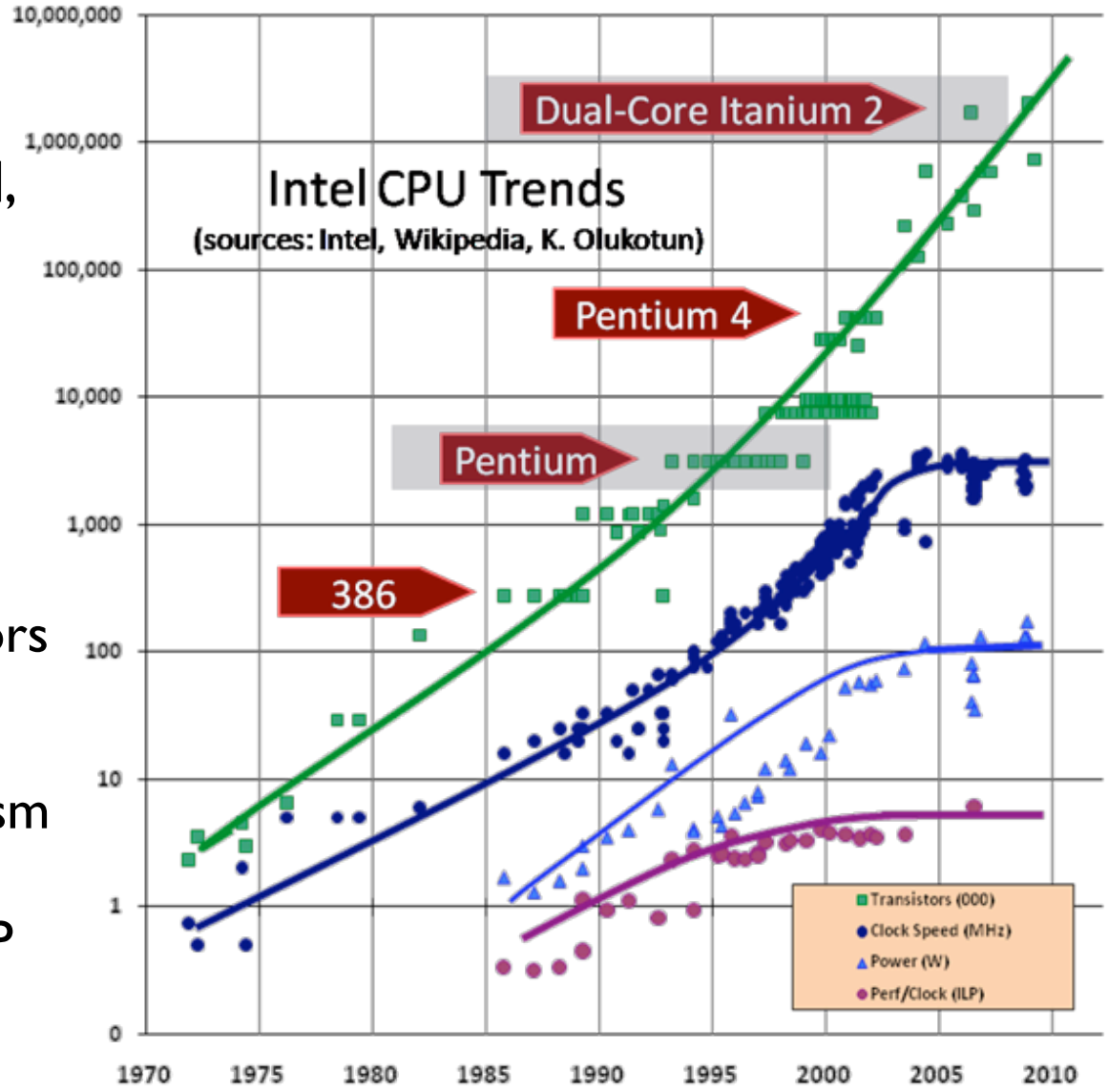
Moore's Law

saturation of clock-speed, power efficiency, ILP

=> paradigm shift

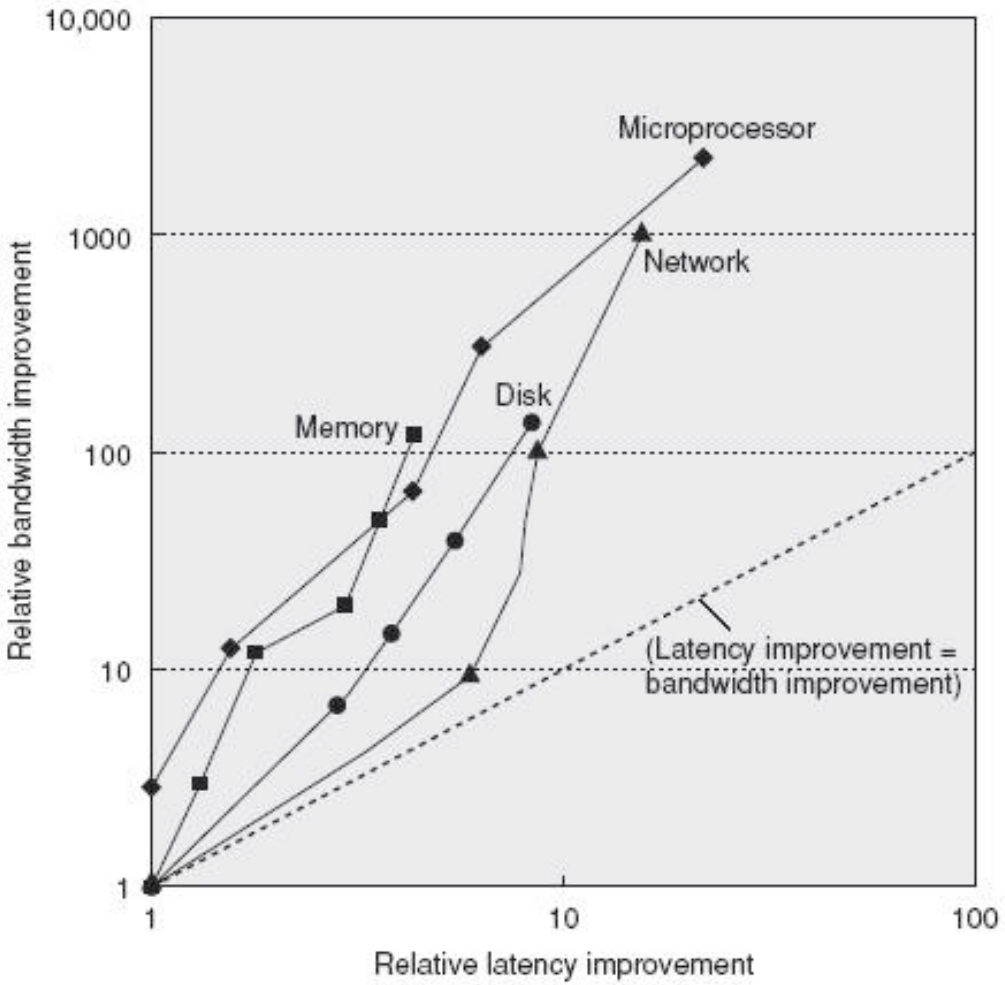
multiple processors/chip rather than faster processors

thread/data level parallelism requires programming (MPI,openMP) unlike ILP



improvements governed by technology

architecture, compiler, programs reflect this



power issues!

chips becoming smaller and smaller

$$P = 1/2\ C V^2 f$$

higher frequency => more power consumption & heating

can't be air cooled!

reduce operating voltage (transistor errors), frequency (speed reduction)

software closely tied to hardware

esp. with parallel systems

source code: high level language (fortran, c, c++)

compiler ↓ (also optimizes the code, e.g., -O2, -O3 flags)

object code & executable (lower level assembly/machine code)

interpreted languages (e.g., python, perl,MATLAB, Mathematica, IDL scripting languages) slower but handy/easier

important to remember architecture to attain maximum performance

Programming Languages

The basic ideas behind computational physics are language independent!

What language to chose?

- Choice depends on the problem:

What language to chose?

- Choice depends on the problem:
  - Numerical simulation

What language to chose?

- Choice depends on the problem:
  - Numerical simulation
  - System programming

What language to chose?

- Choice depends on the problem:
  - Numerical simulation
  - System programming
  - Web programming

What language to chose?

- Choice depends on the problem:
  - Numerical simulation
  - System programming
  - Web programming
- Available libraries and computer codes.

What language to chose?

- Choice depends on the problem:
  - Numerical simulation
  - System programming
  - Web programming
- Available libraries and computer codes.
- Experience.

What language to chose?

For effective and efficient work you need to:

- Choose a language that is right for you *now* and in the nearest future.
  You may need to know/learn more than one language!

What language to chose?

For effective and efficient work you need to:

- Choose a language that is right for you *now* and in the nearest future.
  You may need to know/learn more than one language!
- Have a good book to learn the language from.

What language to chose?

For effective and efficient work you need to:

- Choose a language that is right for you *now* and in the nearest future.
  You may need to know/learn more than one language!
- Have a good book to learn the language from.
- Have a compiler that fits your demands and budget!

What language to chose?

For effective and efficient work you need to:

- Choose a language that is right for you *now* and in the nearest future.
  You may need to know/learn more than one language!
- Have a good book to learn the language from.
- Have a compiler that fits your demands and budget!
- Practice! Practice!! Practice!!!