ICTP The Abdus Salam International Centre for Theoretical Physics

Foundation of Modern Computer Architectures for HPC

Ivan Girotto – igirotto@ictp.it

Information & Communication Technology Section (ICTS)

International Centre for Theoretical Physics (ICTP)

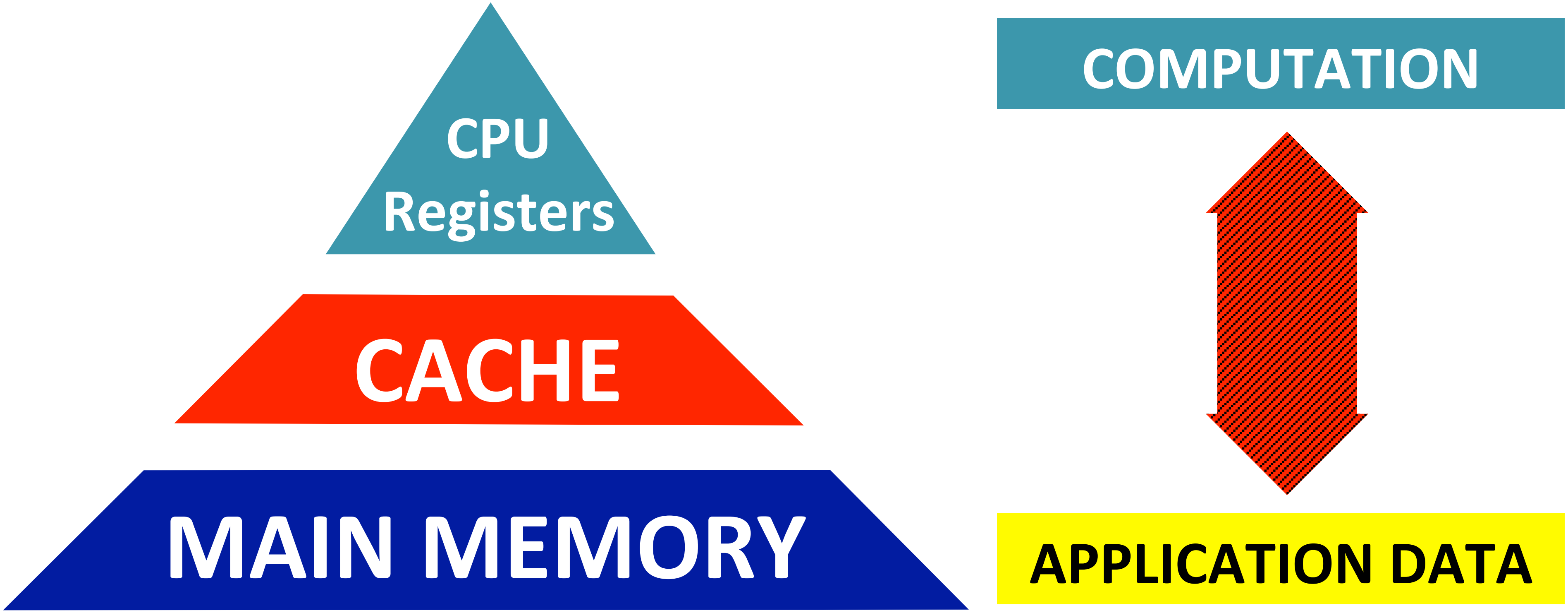
What is High-Performance Computing (HPC)?

- Not a real definition, depends from the prospective:
  - HPC is when I care how fast I get an answer
- Thus HPC can happen on:
  - A workstation, desktop, laptop, smartphone!
  - A supercomputer
  - A Linux Cluster
  - A grid or a cloud
  - Cyberinfrastructure = any combination of the above
- HPC means also High-Productivity Computing

How fast is my CPU?!

- CPU power is measured in FLOPS
  - number of floating point operations x second
  - FLOPS = #cores x clock x $\frac{\text{FLOP}}{\text{cycle}}$
- FLOP/cycle is the number of multiply-add (FMA) performed per cycle
  - architectural limit
  - depend also by the instruction set supported

The CPU Memory Hierarchy



CPU Registers

CACHE

MAIN MEMORY

COMPUTATION

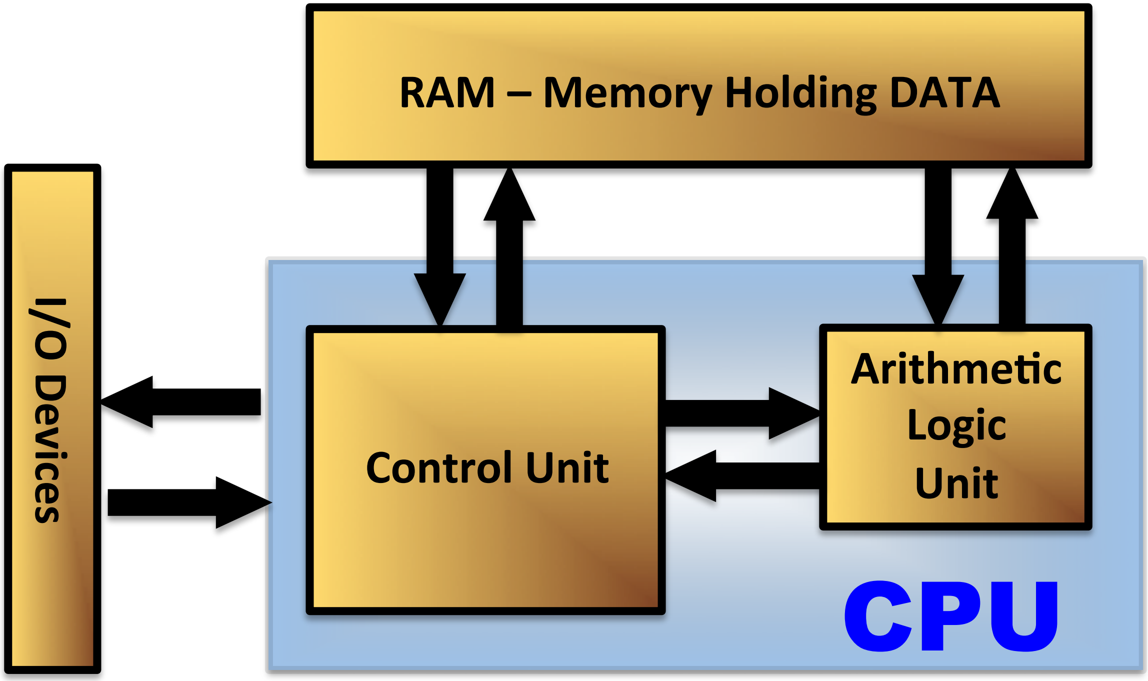
APPLICATION DATA

What Determines Performance?

- How fast is my CPU?
- How fast can I move data around?
- How well can I split work into pieces?

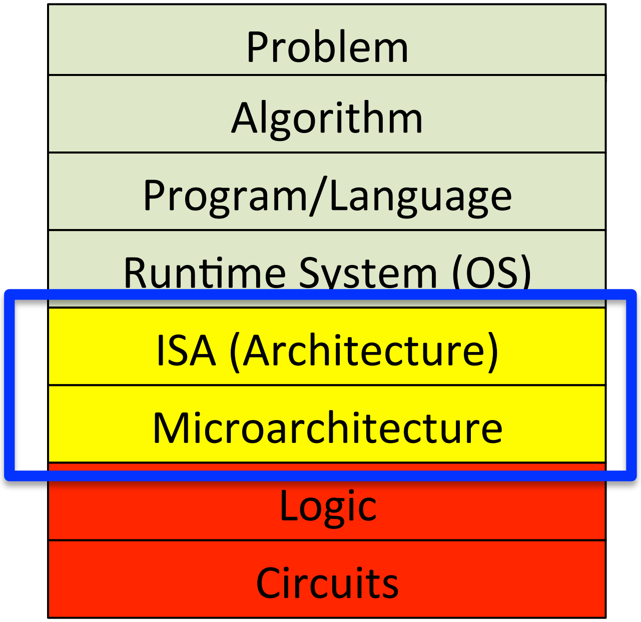
John Von Neumann

The Classical Model



Levels of Transformation

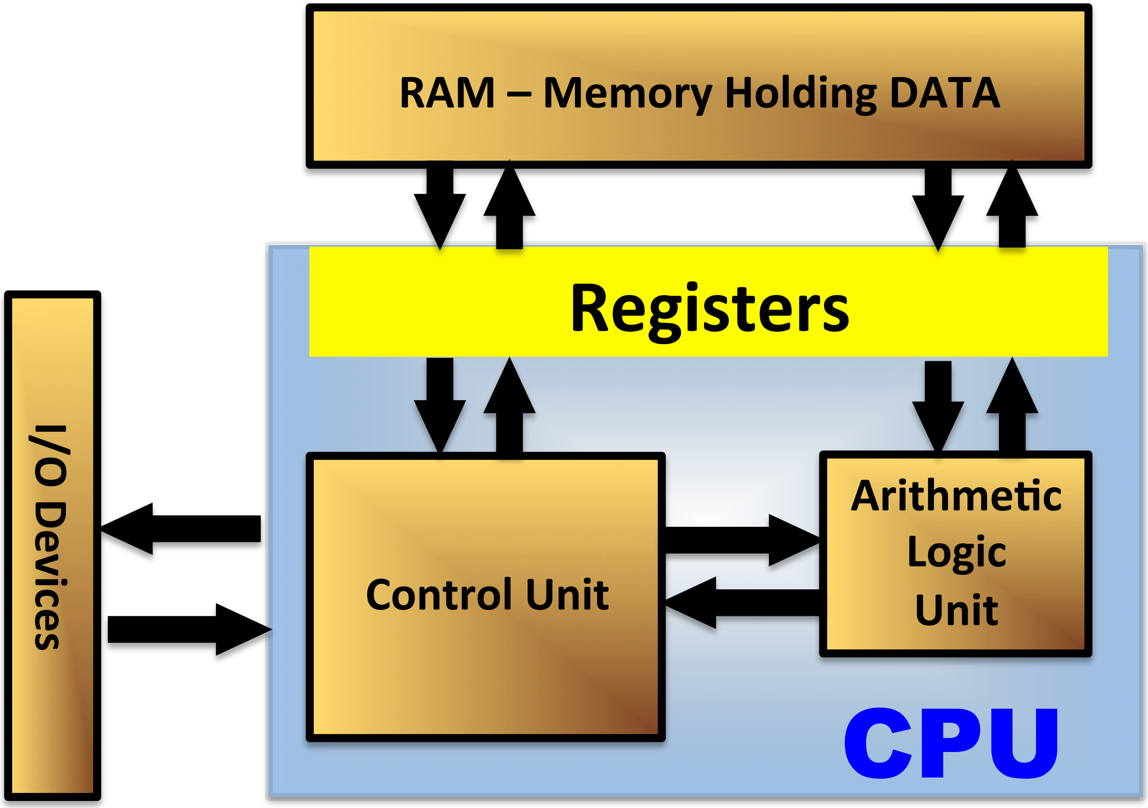
- An algorithm is the sequence of finite steps that describes how to compute a function
- A computer code is a set of instructions written in a syntax defined by the programming language
- The code is finally compiled to become architecture-compatible - “readable” by the CPU.



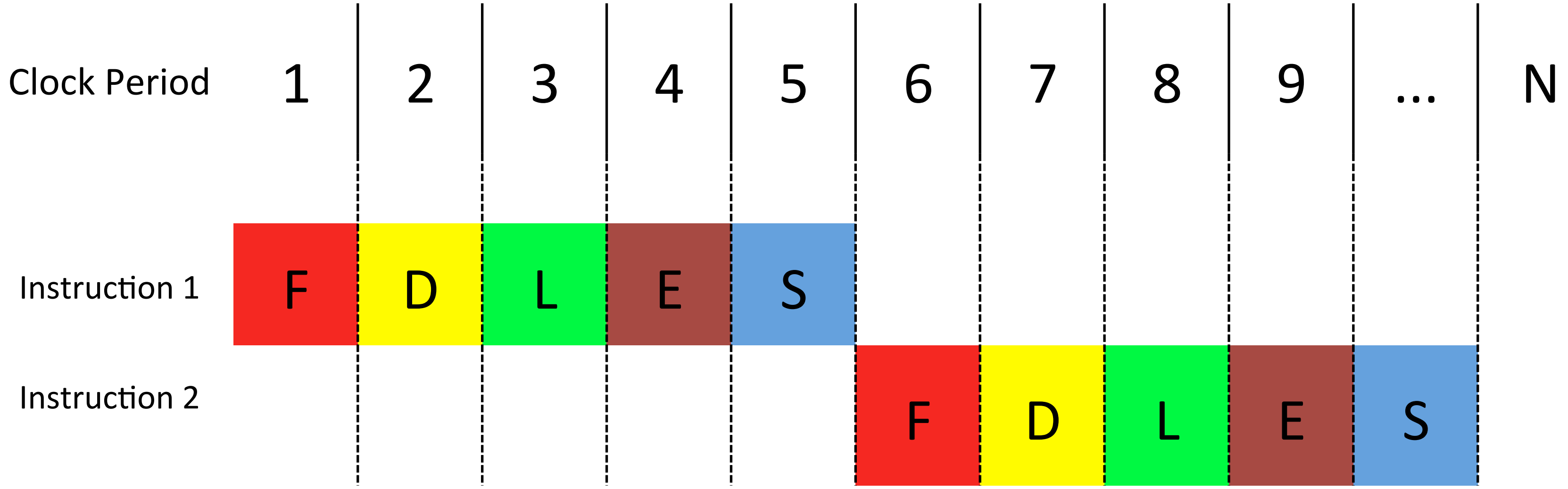
Problem
Algorithm
Program/Language
Runtime System (OS)
ISA (Architecture)
Microarchitecture
Logic
Circuits

John Von Neumann

The Classical Model

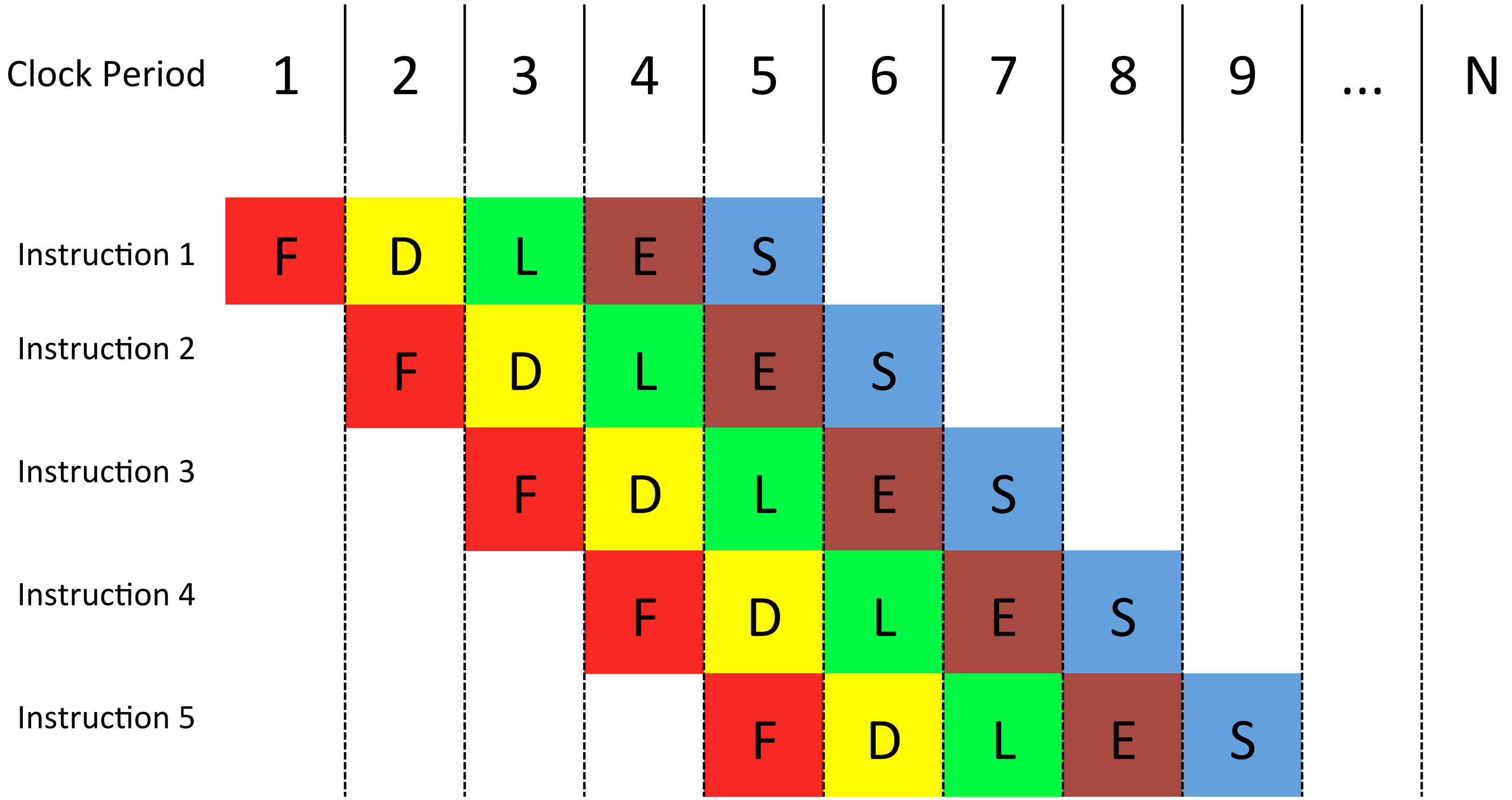


Sequential Processing



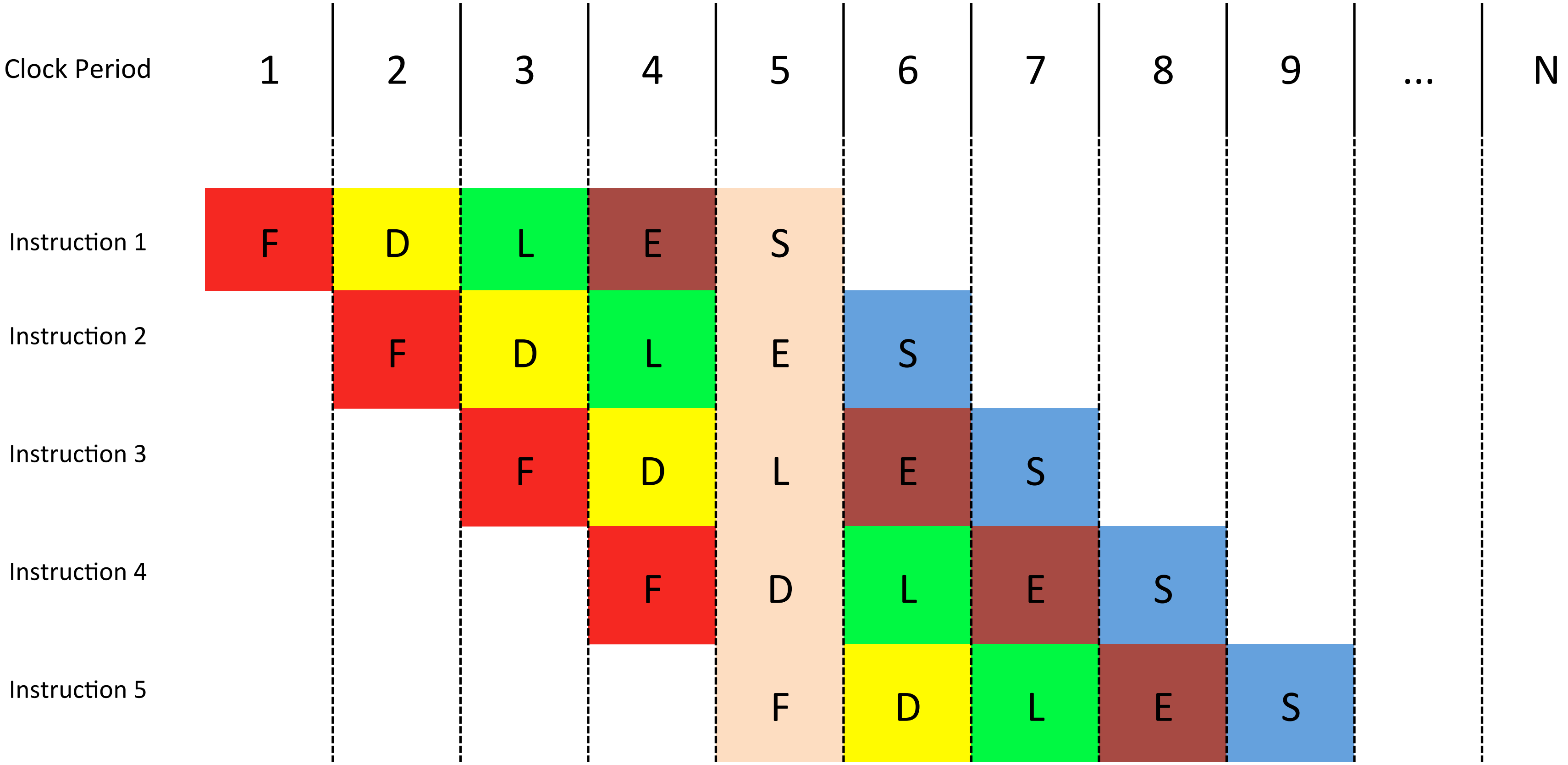
Clock Period	1	2	3	4	5	6	7	8	9	…	N
Instruction 1	F	D	L	E	S						
Instruction 2						F	D	L	E	S	

Pipelining



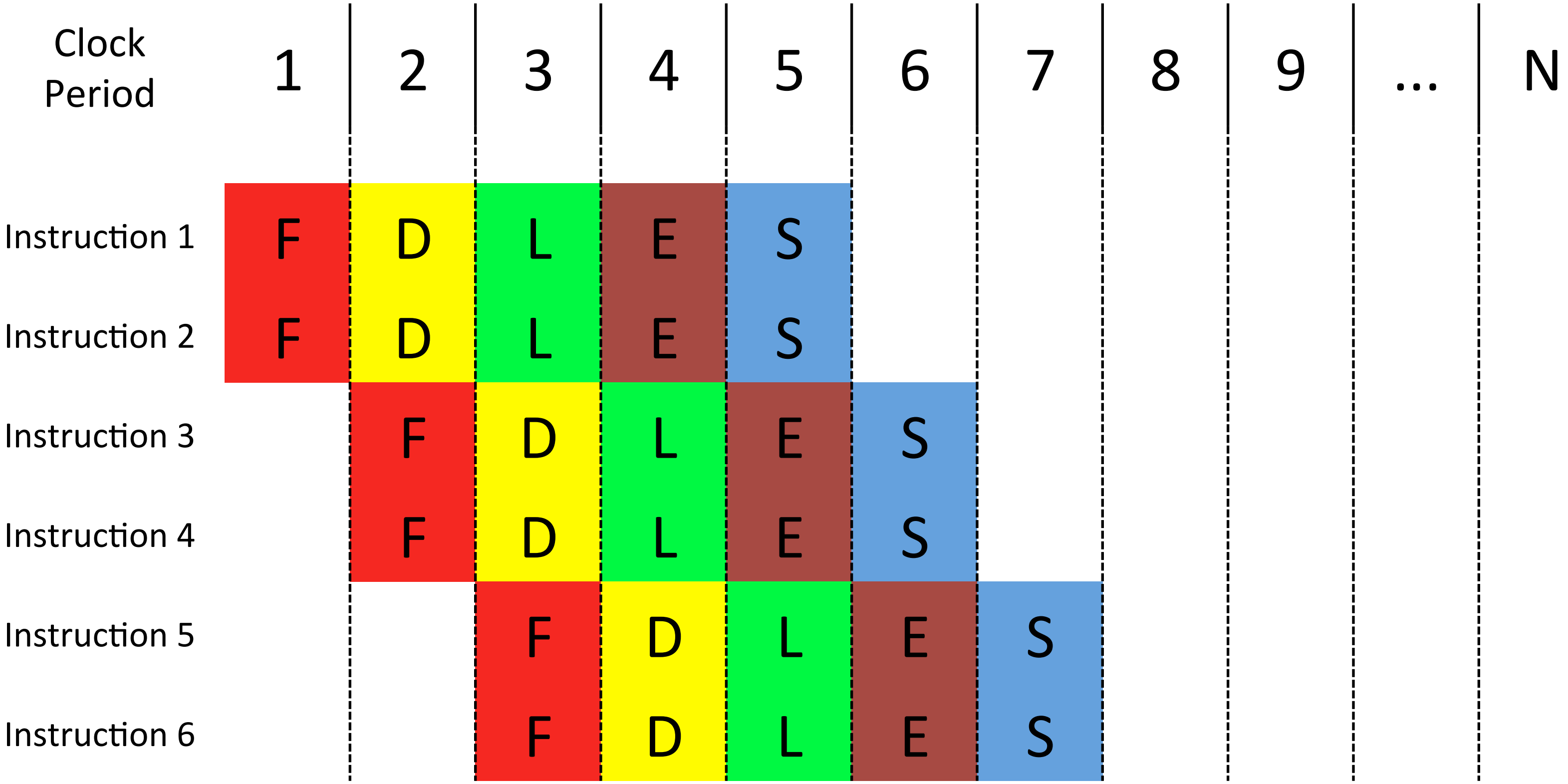
Clock Period	1	2	3	4	5	6	7	8	9	…	N
Instruction 1	F	D	L	E	S						
Instruction 2		F	D	L	E	S					
Instruction 3			F	D	L	E	S				
Instruction 4				F	D	L	E	S			
Instruction 5					F	D	L	E	S		

Pipelining



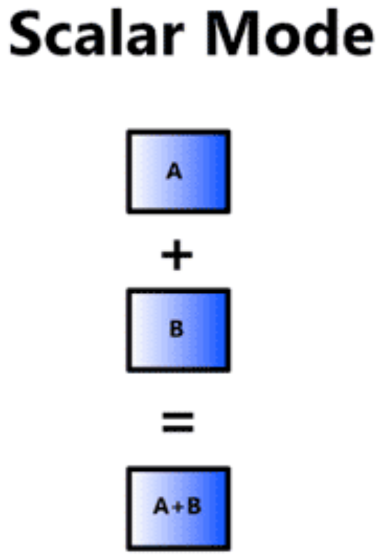
Clock Period	1	2	3	4	5	6	7	8	9	…	N
Instruction 1	F	D	L	E	S						
Instruction 2		F	D	L	E	S					
Instruction 3			F	D	L	E	S				
Instruction 4				F	D	L	E	S			
Instruction 5					F	D	L	E	S		

Superscalaring

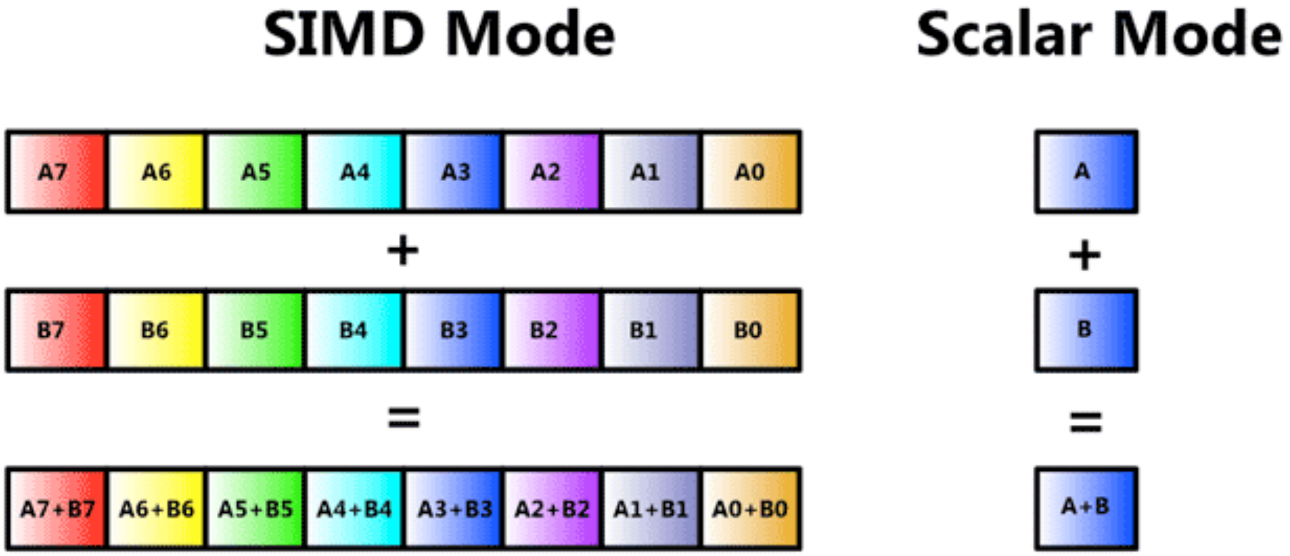


Clock Period	1	2	3	4	5	6	7	8	9	…	N
Instruction 1	F	D	L	E	S						
Instruction 2	F	D	L	E	S						
Instruction 3		F	D	L	E	S					
Instruction 4		F	D	L	E	S					
Instruction 5			F	D	L	E	S				
Instruction 6			F	D	L	E	S				

The Inside Parallelism



The Inside Parallelism



SIMD Mode

A7 A6 A5 A4 A3 A2 A1 A0

+

B7 B6 B5 B4 B3 B2 B1 B0

=

A7+B7 A6+B6 A5+B5 A4+B4 A3+B3 A2+B2 A1+B1 A0+B0

Scalar Mode

A

+

B

=

A+B

Performance Metrics

- When all CPU component work at maximum speed that is called *peak of performance*
  - Tech-spec normally describe the theoretical peak
  - Benchmarks measure the real peak
  - Applications show the real performance value
- CPU performance is measured as:
  - Floating point operations per seconds GFLOP/s
- But the real performance is in many cases mostly related to the memory bandwidth (GBytes/s)

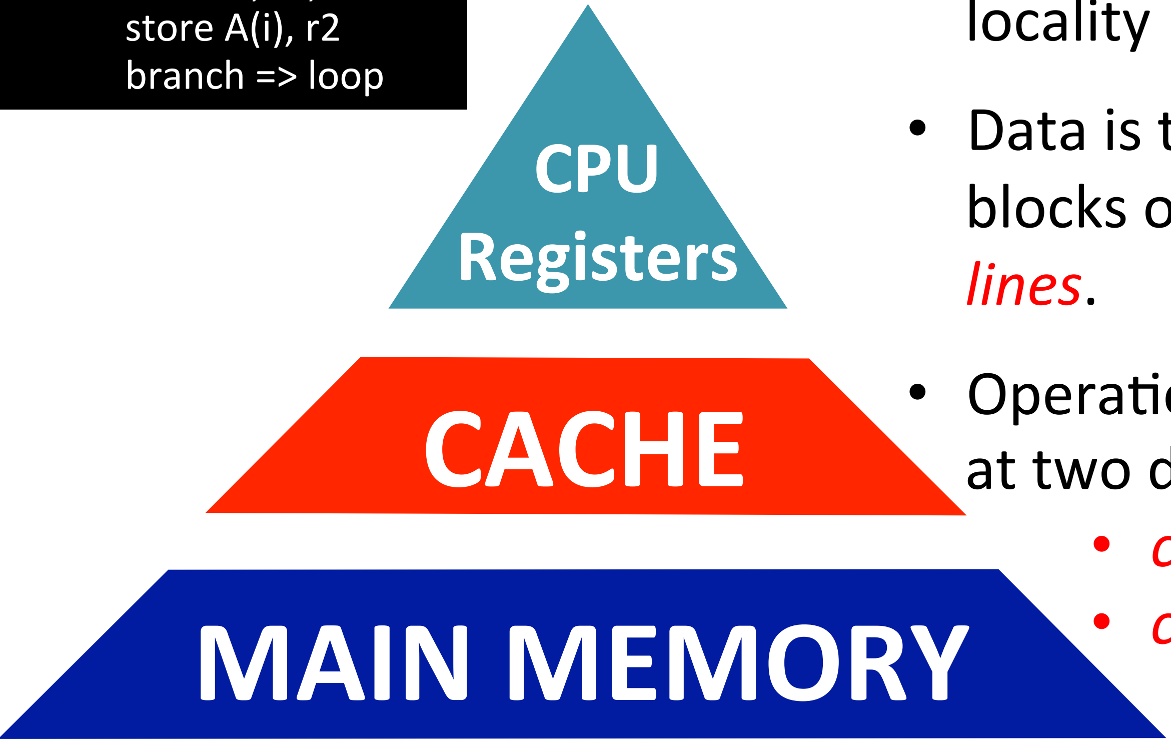
Cache Memory

- Expensive (SRAM) high-speed memory
- Relatively low-capacity in regards to RAM
- Cache Memory are for Instructions (i.e., L1I) and for Data (i.e., L1D)
- Modern CPU are designed with several levels of cache memories

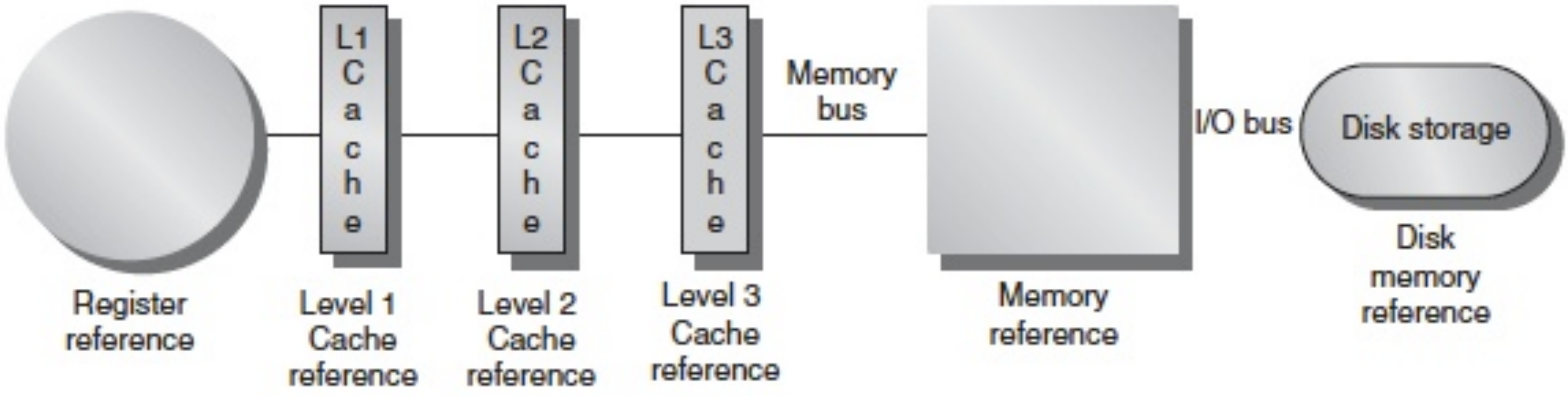
Cache Memory

```
Loop: load r1, A(i)
      load r2, s
      mult r3, r2, r1
      store A(i), r2
      branch => loop
```

- Designed for temporal/spatial locality
- Data is transferred to cache in blocks of fixed size, called *cache lines*.
- Operation of LOAD/STORE can lead at two different scenario:
  - *cache hit*
  - *cache miss*



The CPU Memory Hierarchy



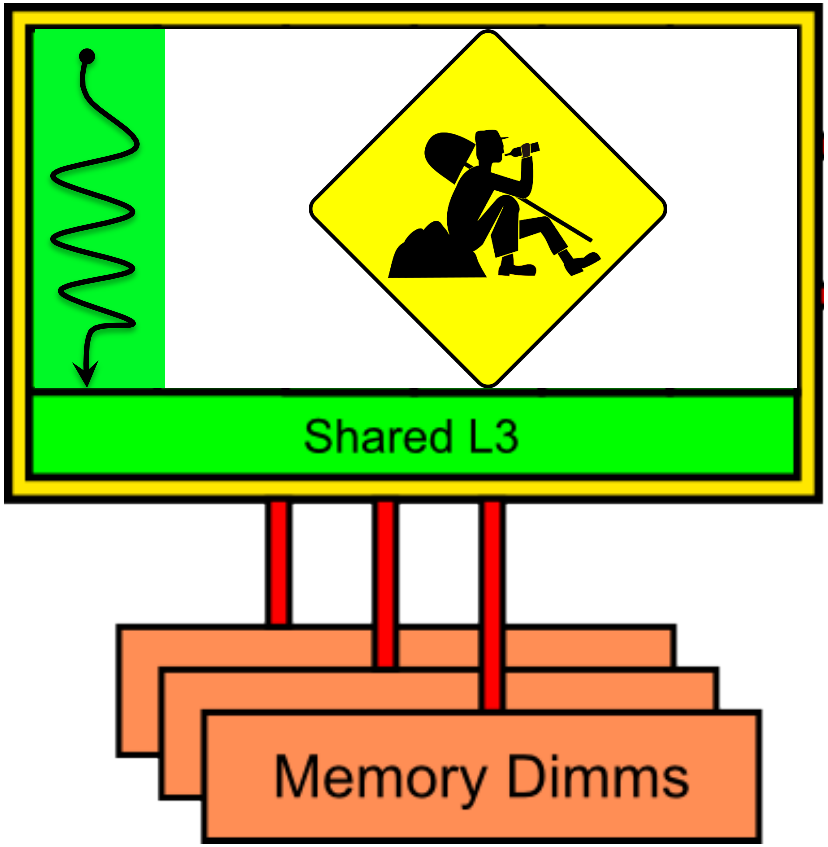
	Register reference	Level 1 Cache reference	Level 2 Cache reference	Level 3 Cache reference	Memory reference	Disk memory reference
Size:	1000 bytes	64 KB	256 KB	2–4 MB	4–16 GB	4–16 TB
Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	5–10 ms

(a) Memory hierarchy for server

Data Memory Access

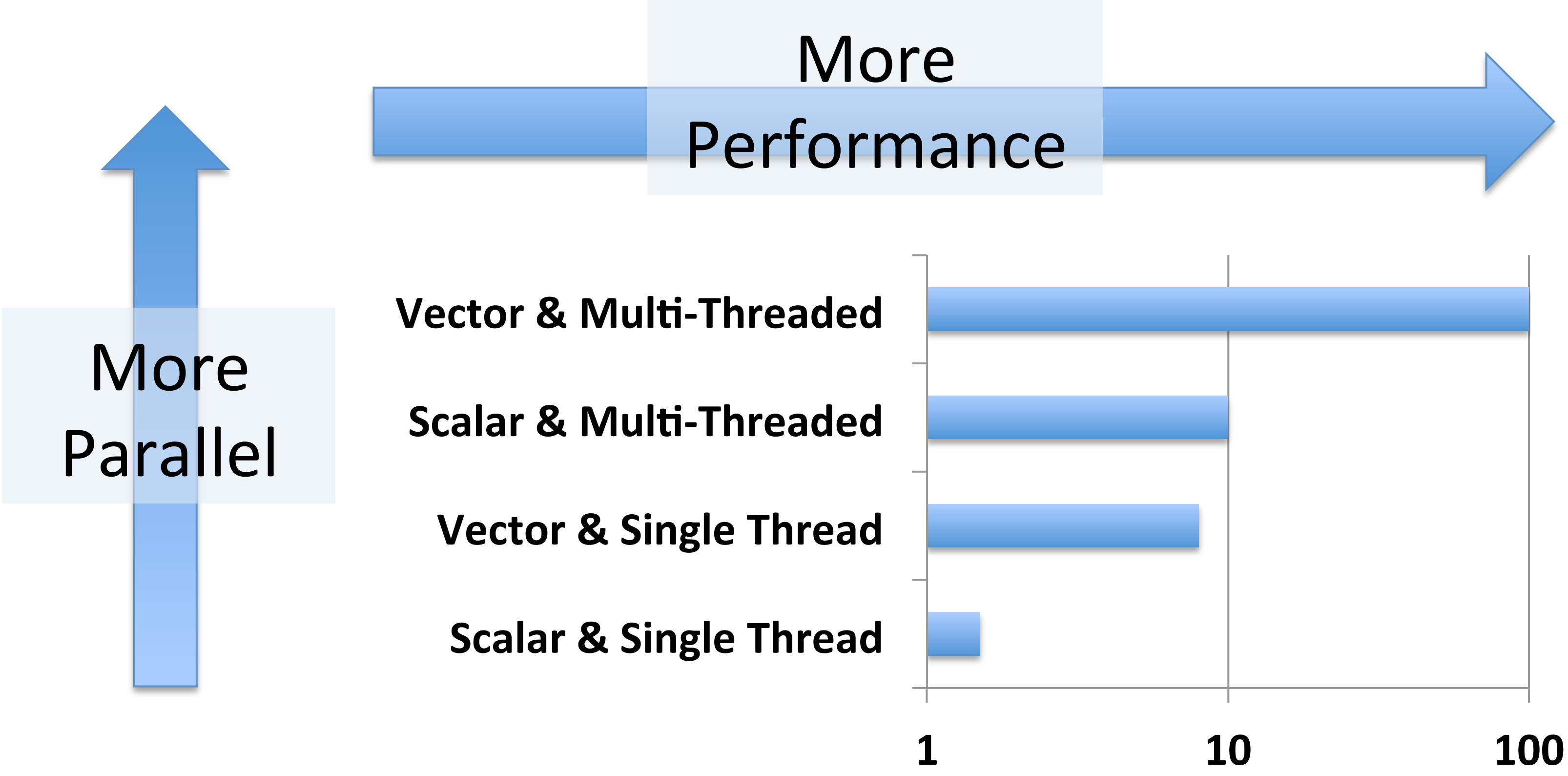
- Data ordering
- Reduce at minimum the data transfers
- Avoid complex data structure within computational intensive kernels
- Define constants and help the compiler
- Exploit the memory hierarchy

Multi-core system Vs Serial Programming



Xeon E5650 hex-core processors (12GB - RAM)

Threading and Vectorization



More Performance

More Parallel