The Abdus Salam
International Centre
for Theoretical Physics

Modern Computer Architectures

Ivan Girotto – igirotto@ictp.it

Information & Communication Technology Section (ICTS)

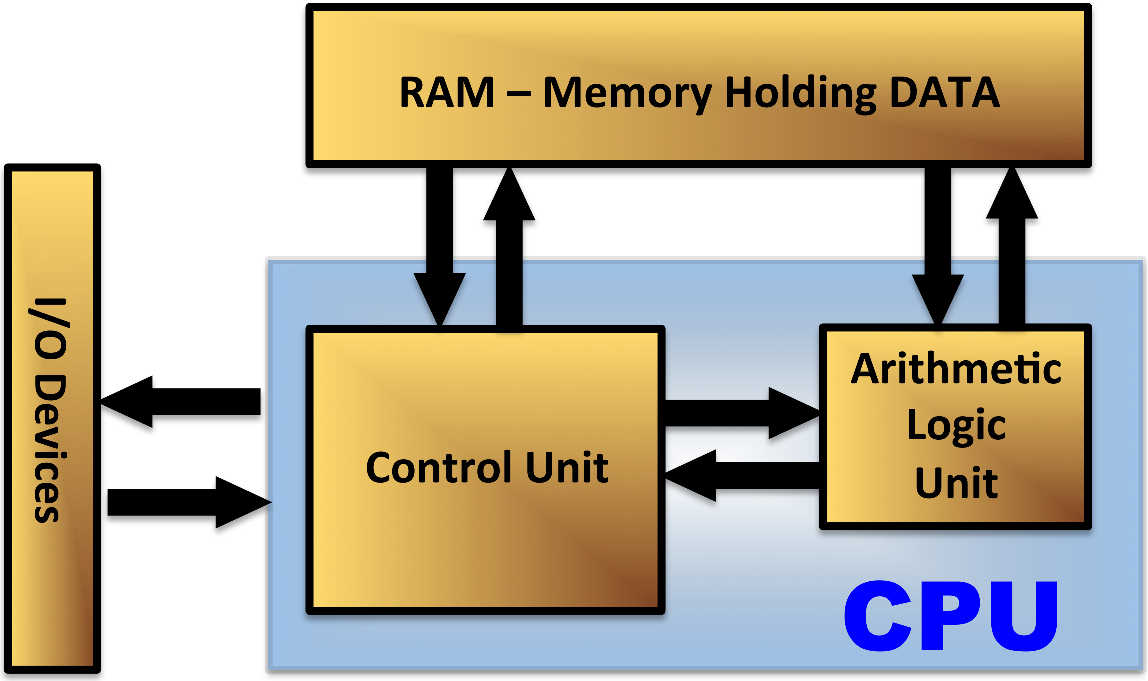
International Centre for Theoretical Physics (ICTP)

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 FLOP/s
- The real performance is in many cases mostly related to the memory bandwidth (Bytes/s) and the exploitation of the parallelism within the CPU

John Von Neumann

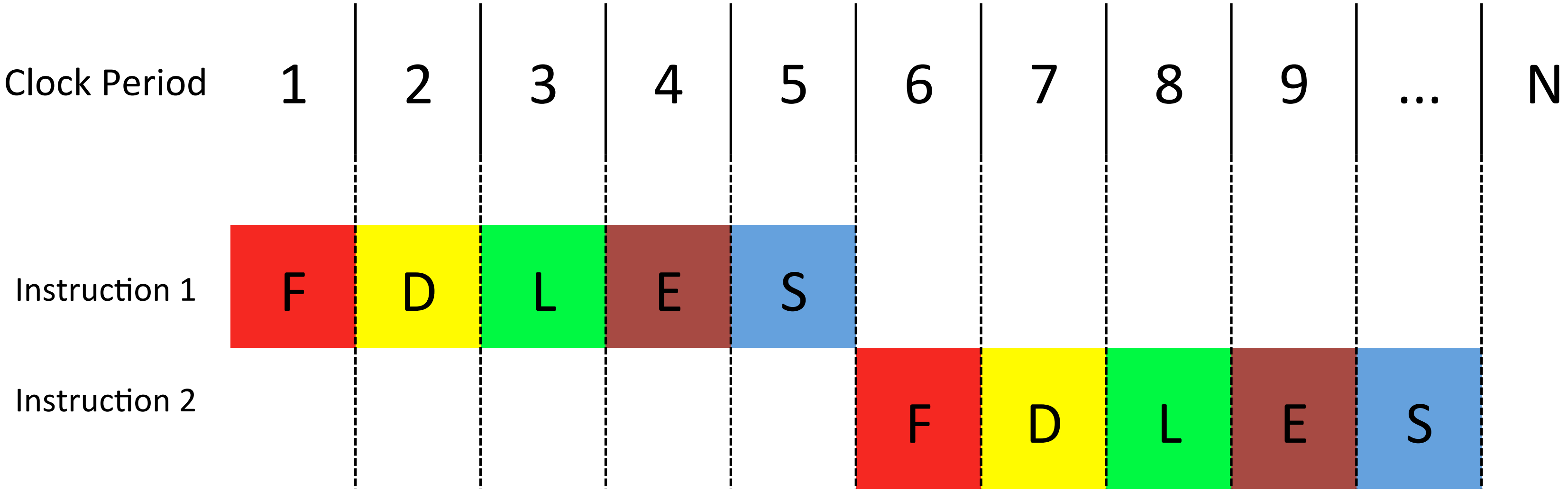
The Classical Model



The Instruction Processing Cycle

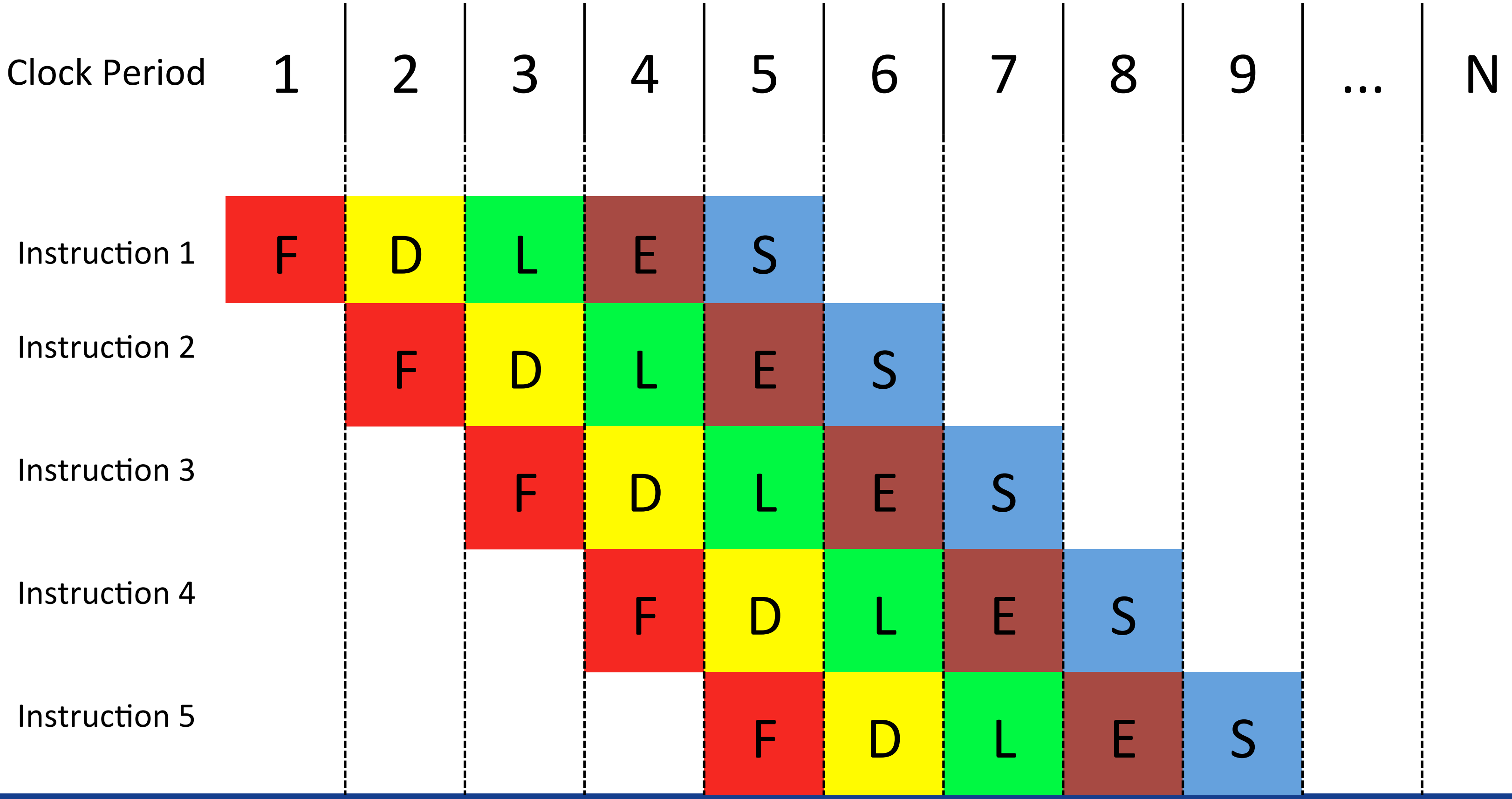
- Fetch: read the next instruction from memory
  - 001000 00001 00010 0000000100001000
- Decode: operands and operation are decoded
  - add, $r1, $r2, 10
- Load: retrieve the data from memory to registers
- Execute: execute the instruction
  - $r1 = 4500 + 10
- Store: store the results

Sequential Processing



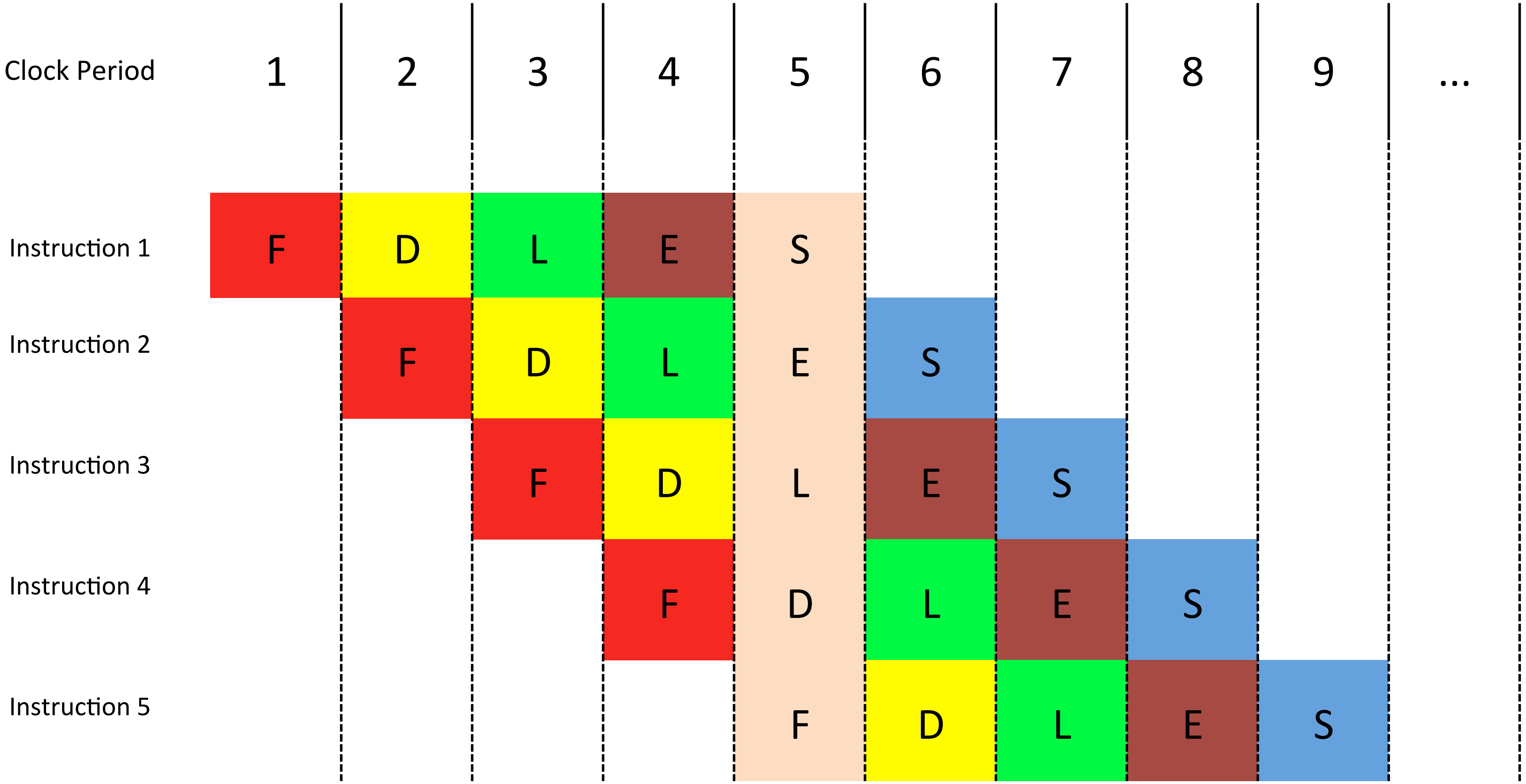
Clock Period	1	2	3	4	5	6	7	8	9	…	N
Instruction 1	F	D	L	E	S						
Instruction 2						F	D	L	E	S	

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				

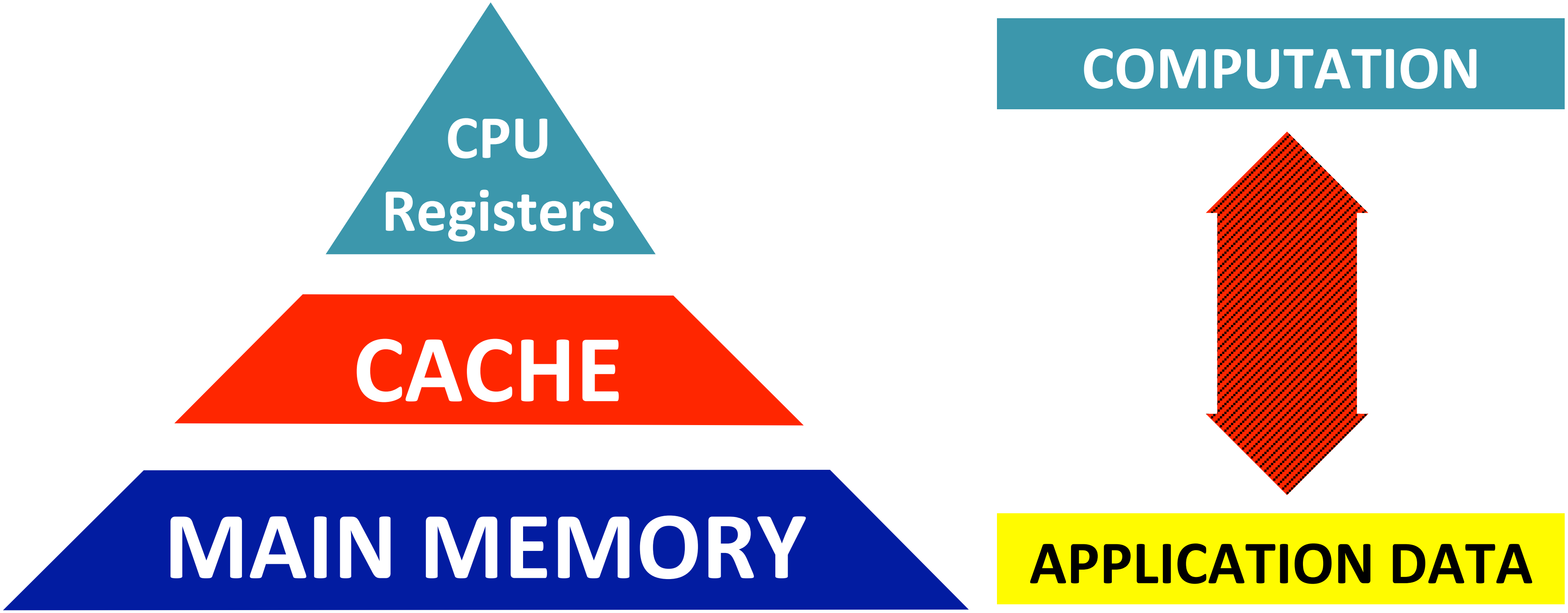
Loops and Pipeline

```
for( i = 0; i < N; i += 1 )
{
   A[i] = s * A[i]
}
```

Time	1	2	3	4	5	6	7	8	9	…	N
Step 1	L1	L2	M	St	B						
Step 2		L1	L2	M	St	B					
Step 3			L1	L2	M	St	B				
Step 4				L1	L2	M	St	B			
Step 5					L1	L2	M	St	B		

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

The CPU Memory Hierarchy



CPU Registers

CACHE

MAIN MEMORY

COMPUTATION

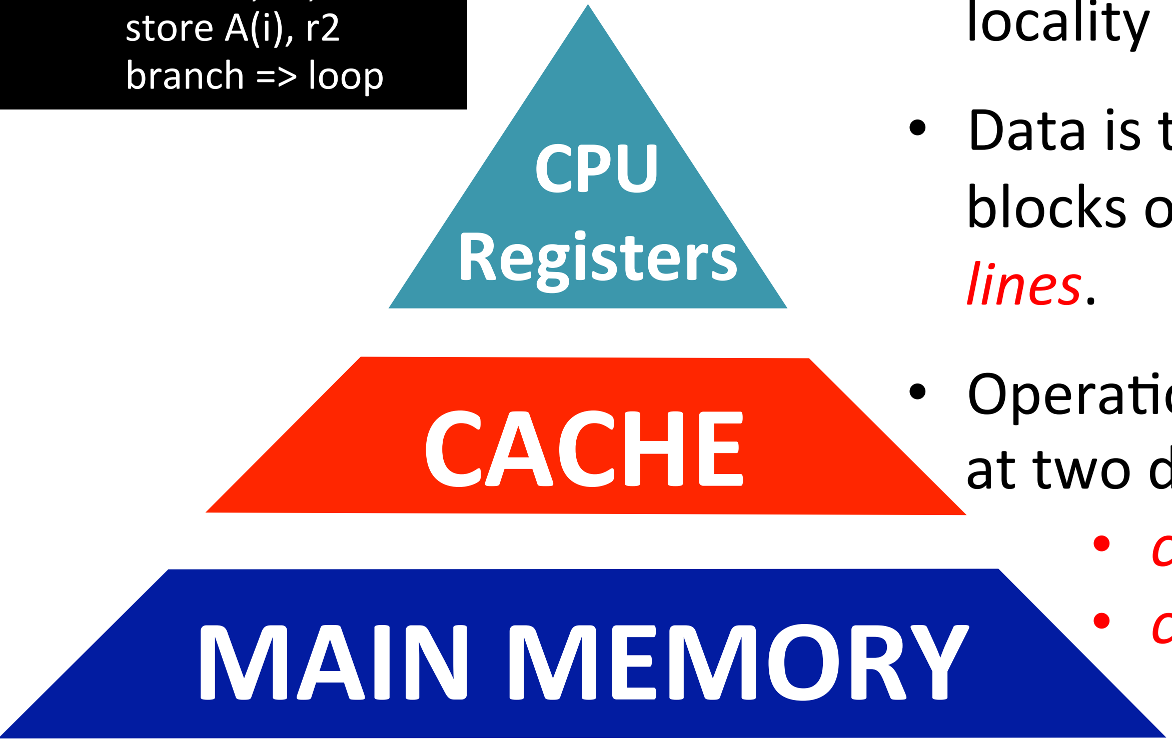
APPLICATION DATA

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

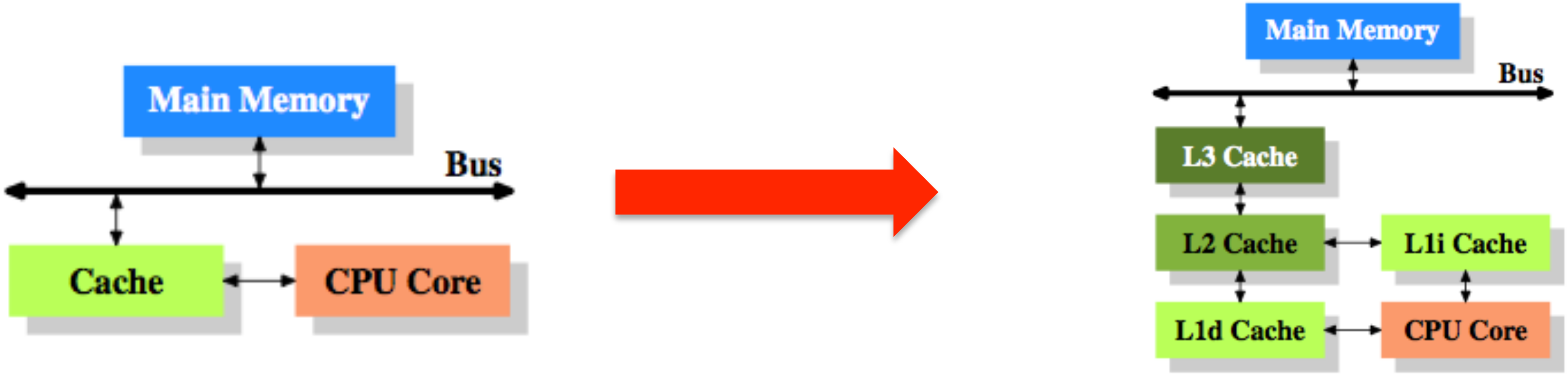
```
Loop: load r1, A(i)
      load r2, s
      mult r3, r2, r1
      store A(i), r2
      branch => loop
```



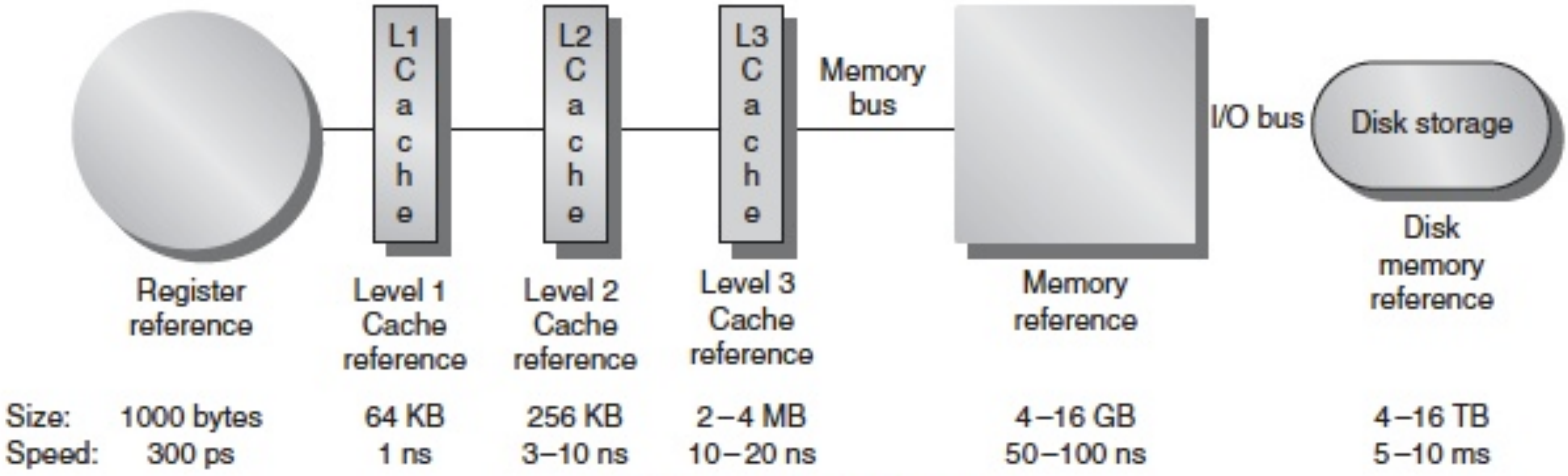
- Designed for temporal/spatial locality
- Data is transferred to cache in blocks of fixed size, called *cache lines*.
- Operation of LOAD/STORE can lead at two different scenario:
  - *cache hit*
  - *cache miss*

Caches

- Fast memory to exploit spatial and temporal locality!



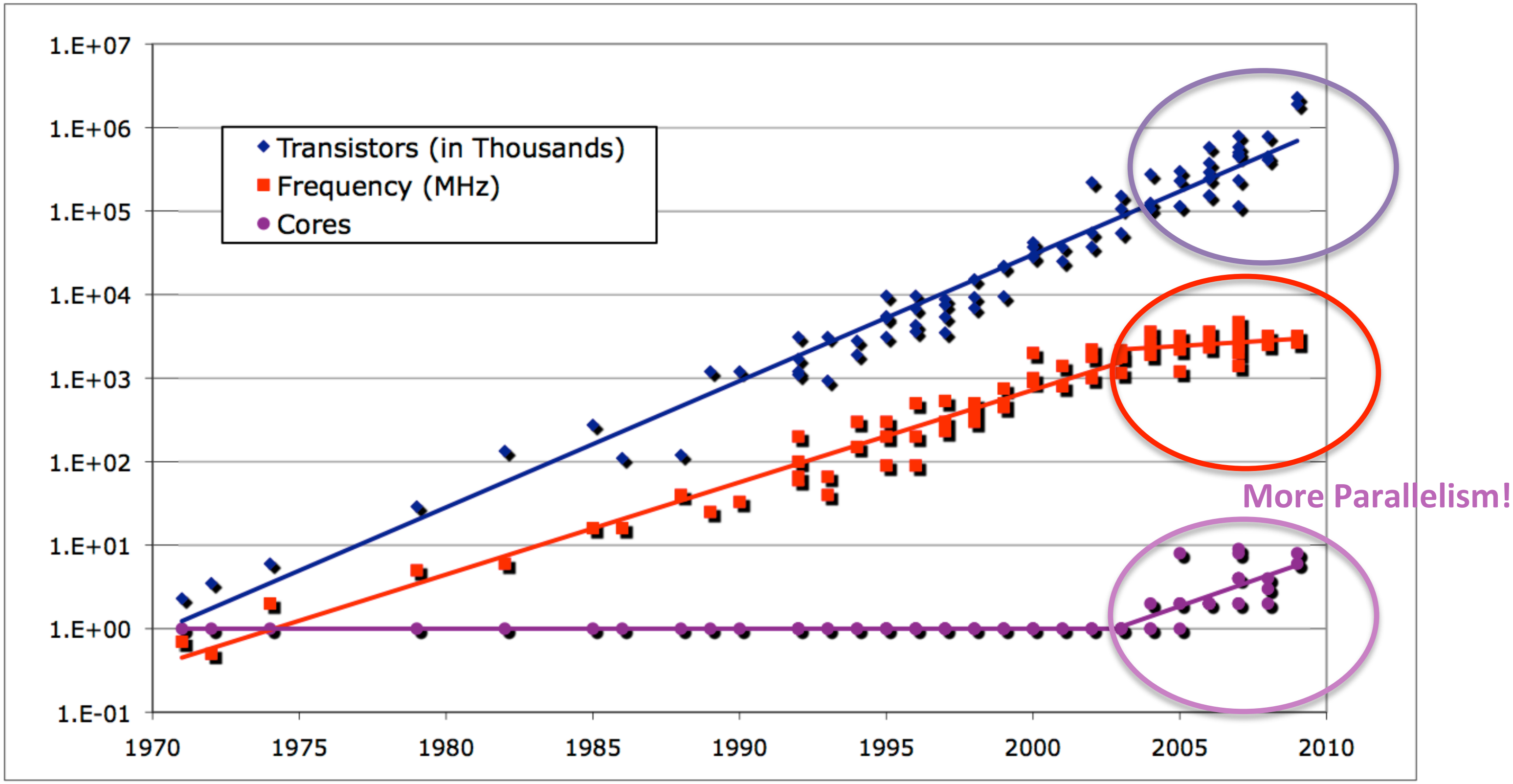
The CPU Memory Hierarchy



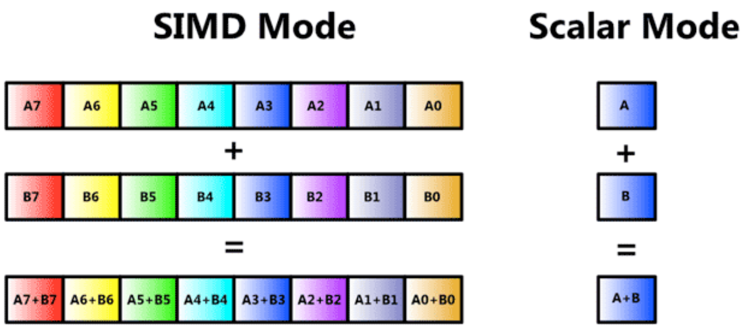
	Register reference	Level 1 Cache reference	Level 2 Cache reference	Level 3 Cache reference	Memory reference	Disk memory reference
Size:	1000 bytes	64 KB	256 KB	2–4 MB	4–16 GB	4–16 TB
Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	5–10 ms

(a) Memory hierarchy for server

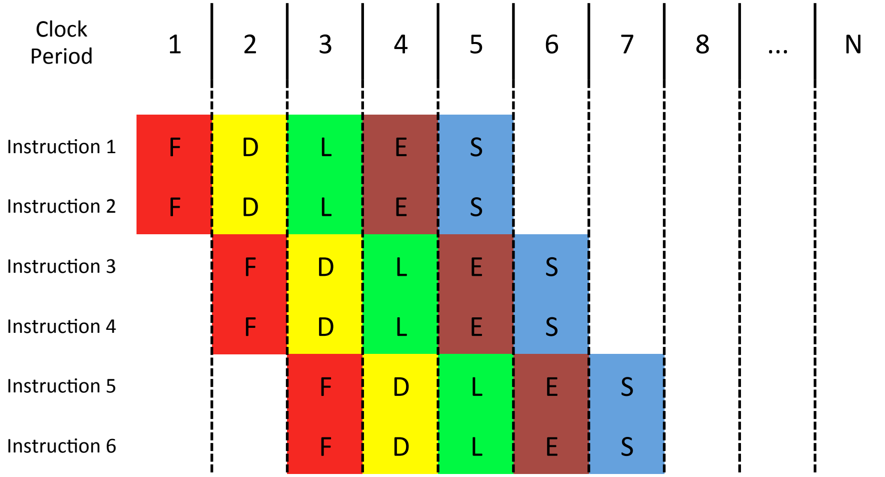
HPC Trend and Moore's Law



To the Extreme - Parallel Inside



Vector Units for processing multiple data in //



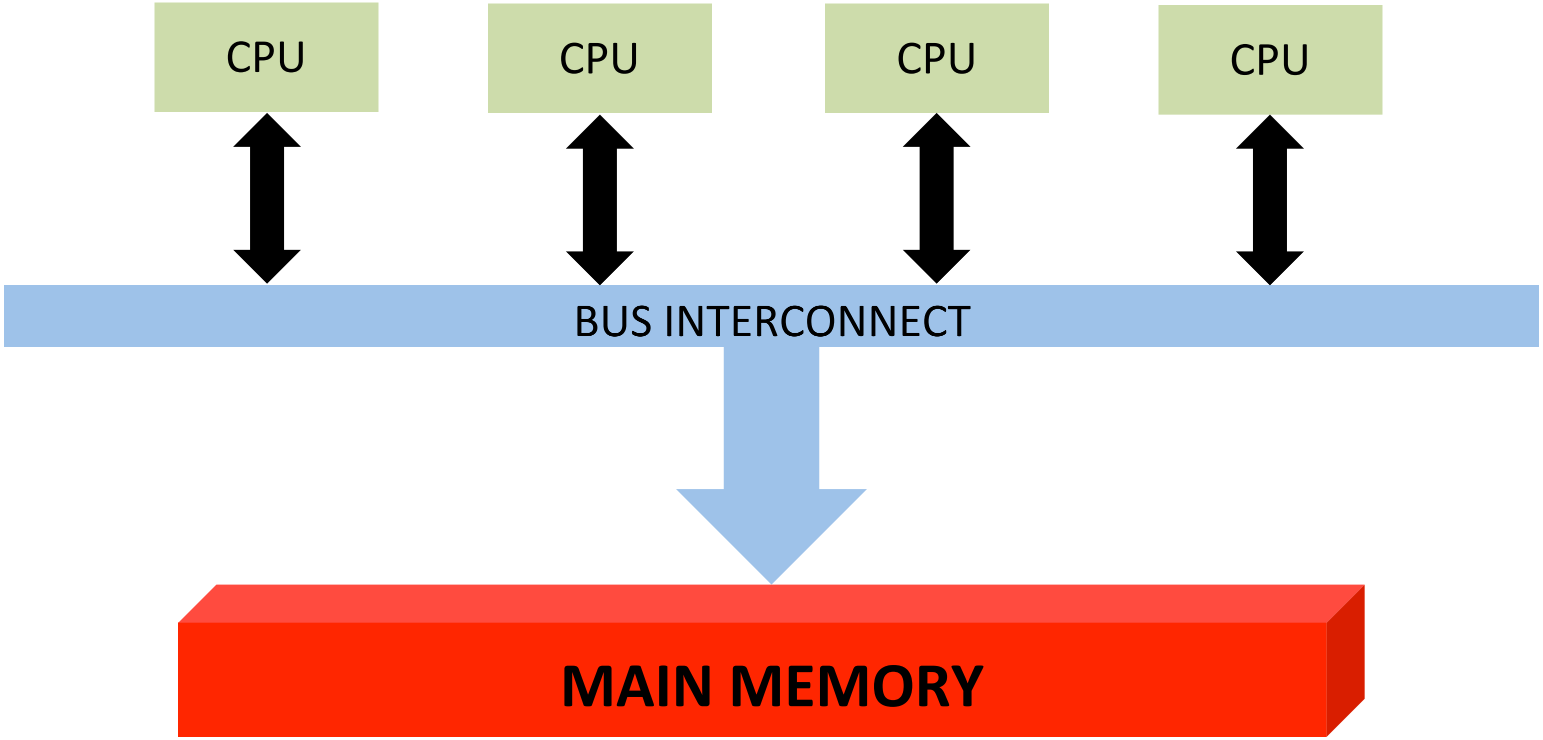
Clock Period	1	2	3	4	5	6	7	8	…	N
Instruction 1	F	D	L	E	S					
Instruction 2	F	D	L	E	S					
Instruction 3		F	D	L	E	S				
Instruction 4		F	D	L	E	S				
Instruction 5			F	D	L	E	S			
Instruction 6			F	D	L	E	S			

Pipelined/Superscalar design: multiple functional units operate concurrently

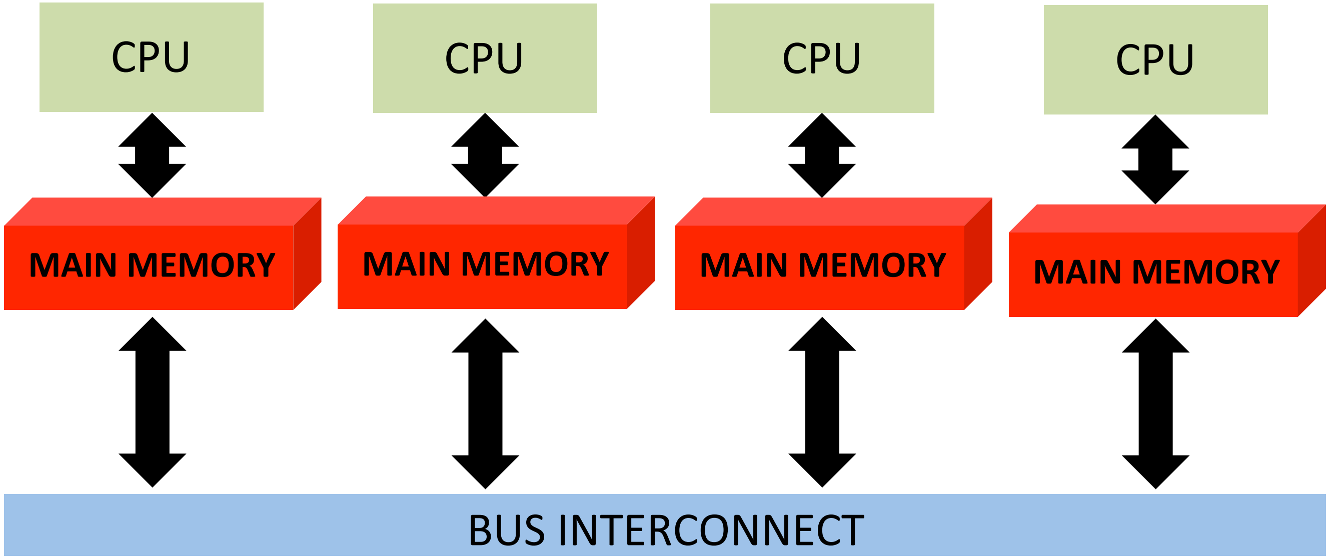
Few basic rules for optimized codes

- Do less work!!
  - Elimination of common sub-expressions
- Avoid expensive operations
  - Reduce your math to cheap operations
  - Avoid branches
- Think as a the compiler works
  - Enhance the compiler

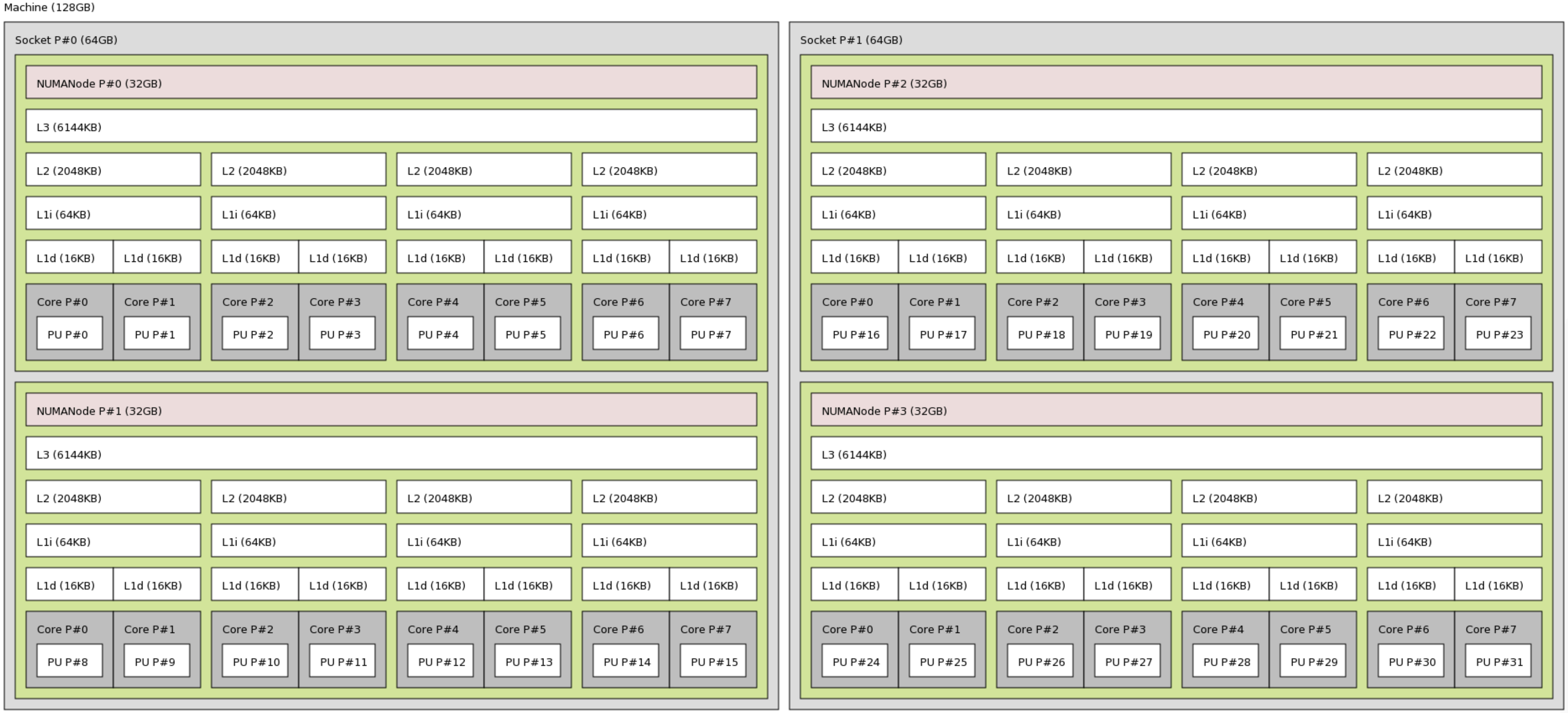
Symmetric Multiprocessors (SMP)



Modern NUMA Multicores



The AMD Opteron 6380 Abu Dhabi 2.5GHz



Machine (128GB)

Socket P#0 (64GB)

NUMANode P#0 (32GB)

L3 (6144KB)

L2 (2048KB) | L2 (2048KB) | L2 (2048KB) | L2 (2048KB)

L1i (64KB) | L1i (64KB) | L1i (64KB) | L1i (64KB)

L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB)

Core P#0 PU P#0 | Core P#1 PU P#1 | Core P#2 PU P#2 | Core P#3 PU P#3 | Core P#4 PU P#4 | Core P#5 PU P#5 | Core P#6 PU P#6 | Core P#7 PU P#7

NUMANode P#1 (32GB)

L3 (6144KB)

L2 (2048KB) | L2 (2048KB) | L2 (2048KB) | L2 (2048KB)

L1i (64KB) | L1i (64KB) | L1i (64KB) | L1i (64KB)

L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB)

Core P#0 PU P#8 | Core P#1 PU P#9 | Core P#2 PU P#10 | Core P#3 PU P#11 | Core P#4 PU P#12 | Core P#5 PU P#13 | Core P#6 PU P#14 | Core P#7 PU P#15

Socket P#1 (64GB)

NUMANode P#2 (32GB)

L3 (6144KB)

L2 (2048KB) | L2 (2048KB) | L2 (2048KB) | L2 (2048KB)

L1i (64KB) | L1i (64KB) | L1i (64KB) | L1i (64KB)

L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB)

Core P#0 PU P#16 | Core P#1 PU P#17 | Core P#2 PU P#18 | Core P#3 PU P#19 | Core P#4 PU P#20 | Core P#5 PU P#21 | Core P#6 PU P#22 | Core P#7 PU P#23

NUMANode P#3 (32GB)

L3 (6144KB)

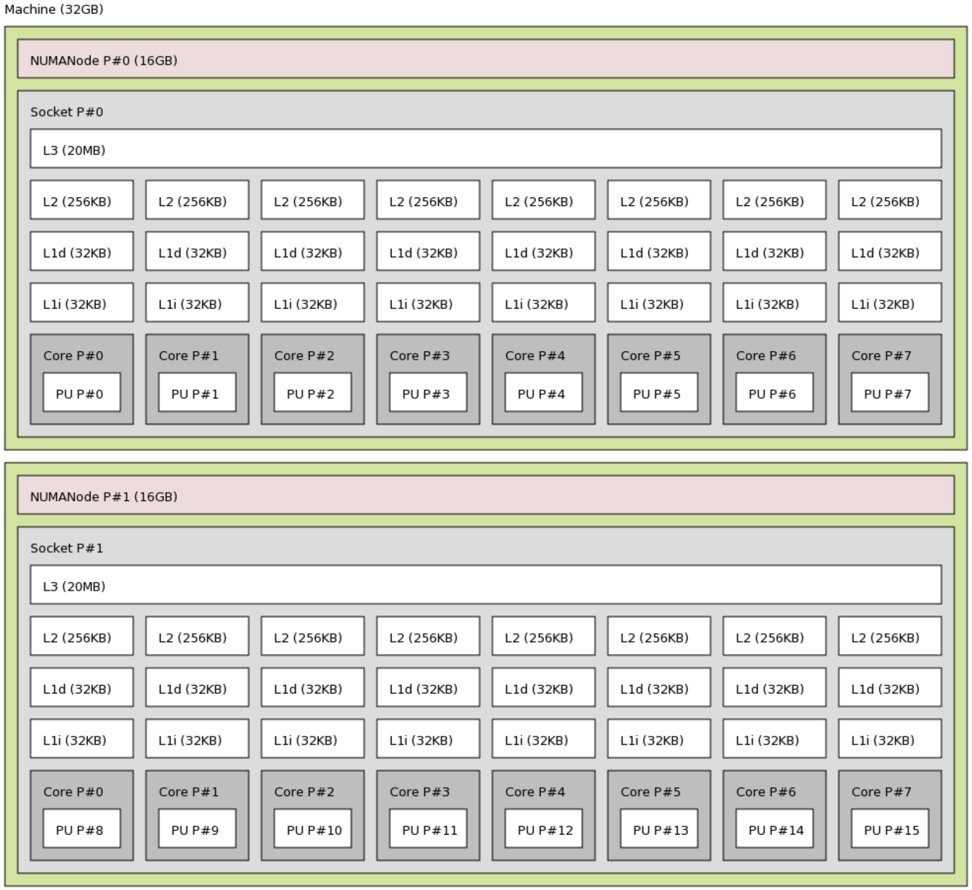
L2 (2048KB) | L2 (2048KB) | L2 (2048KB) | L2 (2048KB)

L1i (64KB) | L1i (64KB) | L1i (64KB) | L1i (64KB)

L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB) | L1d (16KB)

Core P#0 PU P#24 | Core P#1 PU P#25 | Core P#2 PU P#26 | Core P#3 PU P#27 | Core P#4 PU P#28 | Core P#5 PU P#29 | Core P#6 PU P#30 | Core P#7 PU P#31

The Intel Xeon E5-2665 Sandy Bridge-EP 2.4GHz



Machine (32GB)

NUMANode P#0 (16GB)

Socket P#0

L3 (20MB)

L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)
L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)
L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)
Core P#0 PU P#0	Core P#1 PU P#1	Core P#2 PU P#2	Core P#3 PU P#3	Core P#4 PU P#4	Core P#5 PU P#5	Core P#6 PU P#6	Core P#7 PU P#7

NUMANode P#1 (16GB)

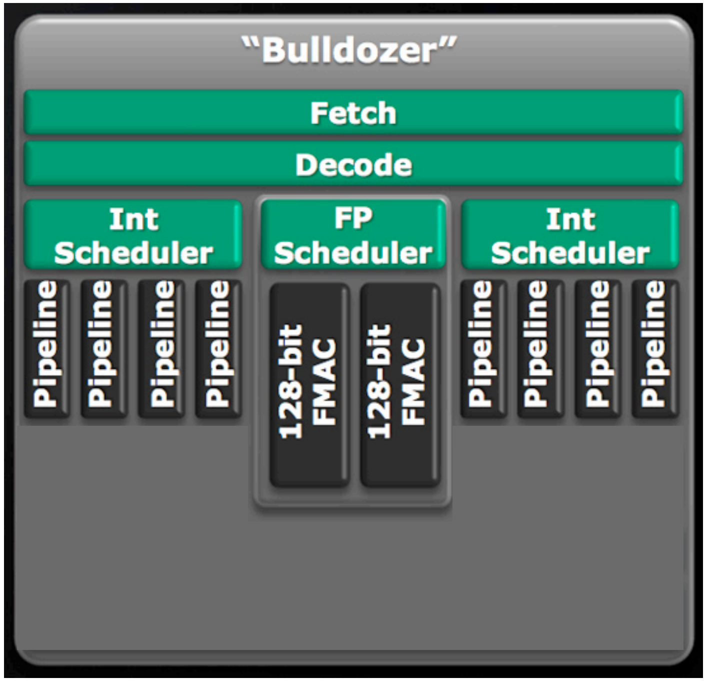
Socket P#1

L3 (20MB)

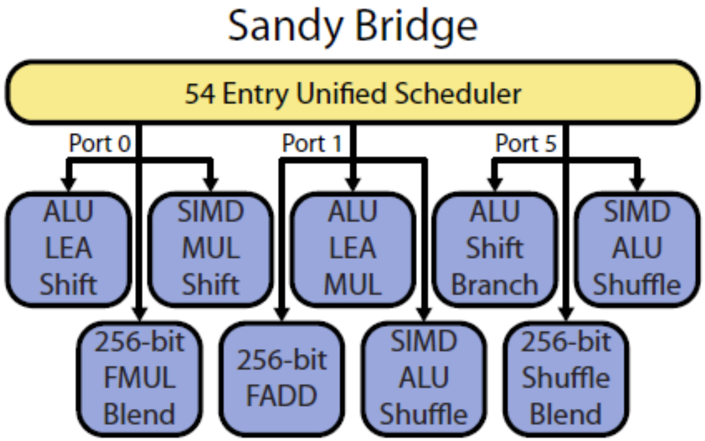
L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)	L2 (256KB)
L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)	L1d (32KB)
L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)	L1i (32KB)
Core P#0 PU P#8	Core P#1 PU P#9	Core P#2 PU P#10	Core P#3 PU P#11	Core P#4 PU P#12	Core P#5 PU P#13	Core P#6 PU P#14	Core P#7 PU P#15

State of the art

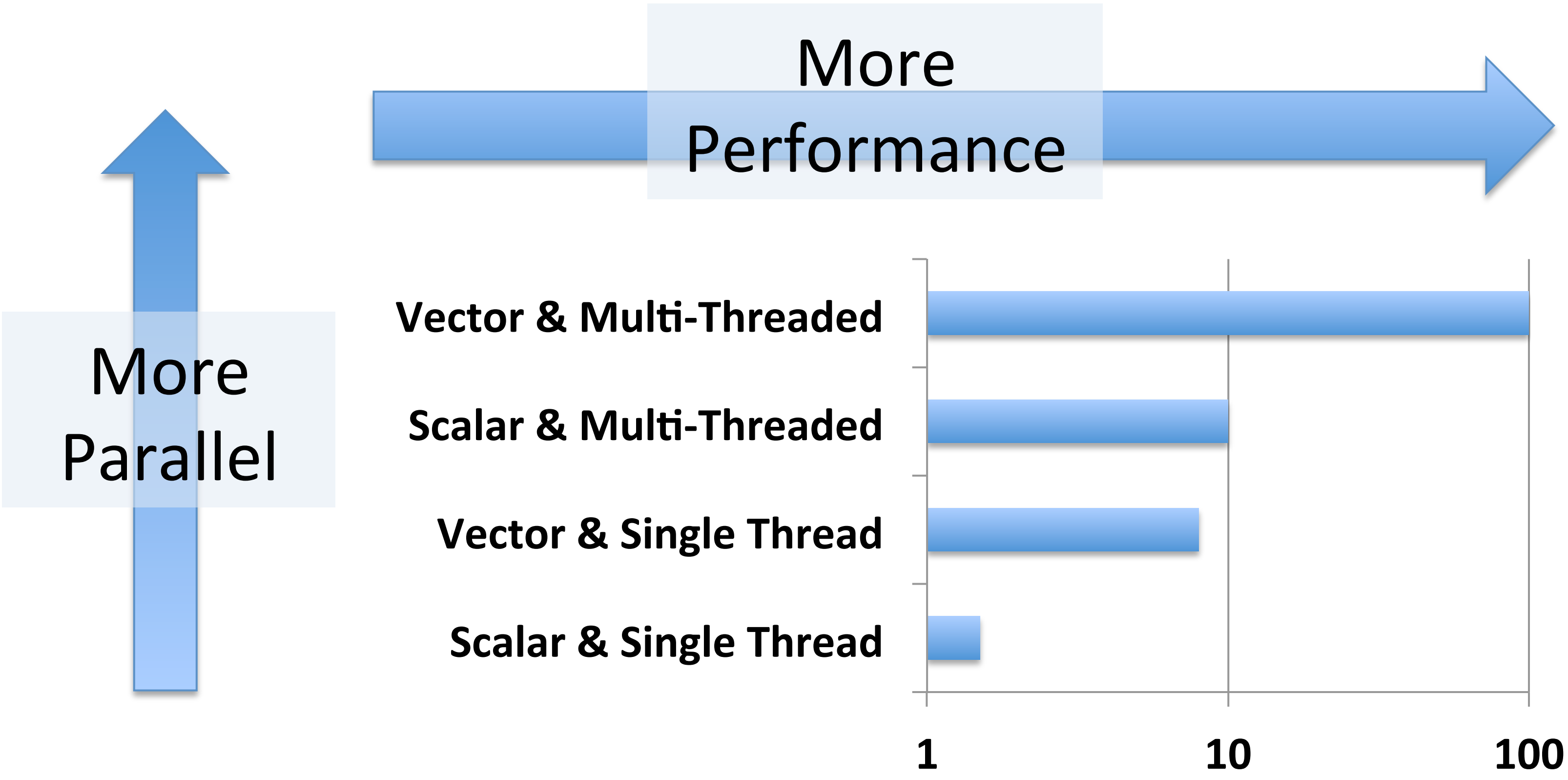
- AMD

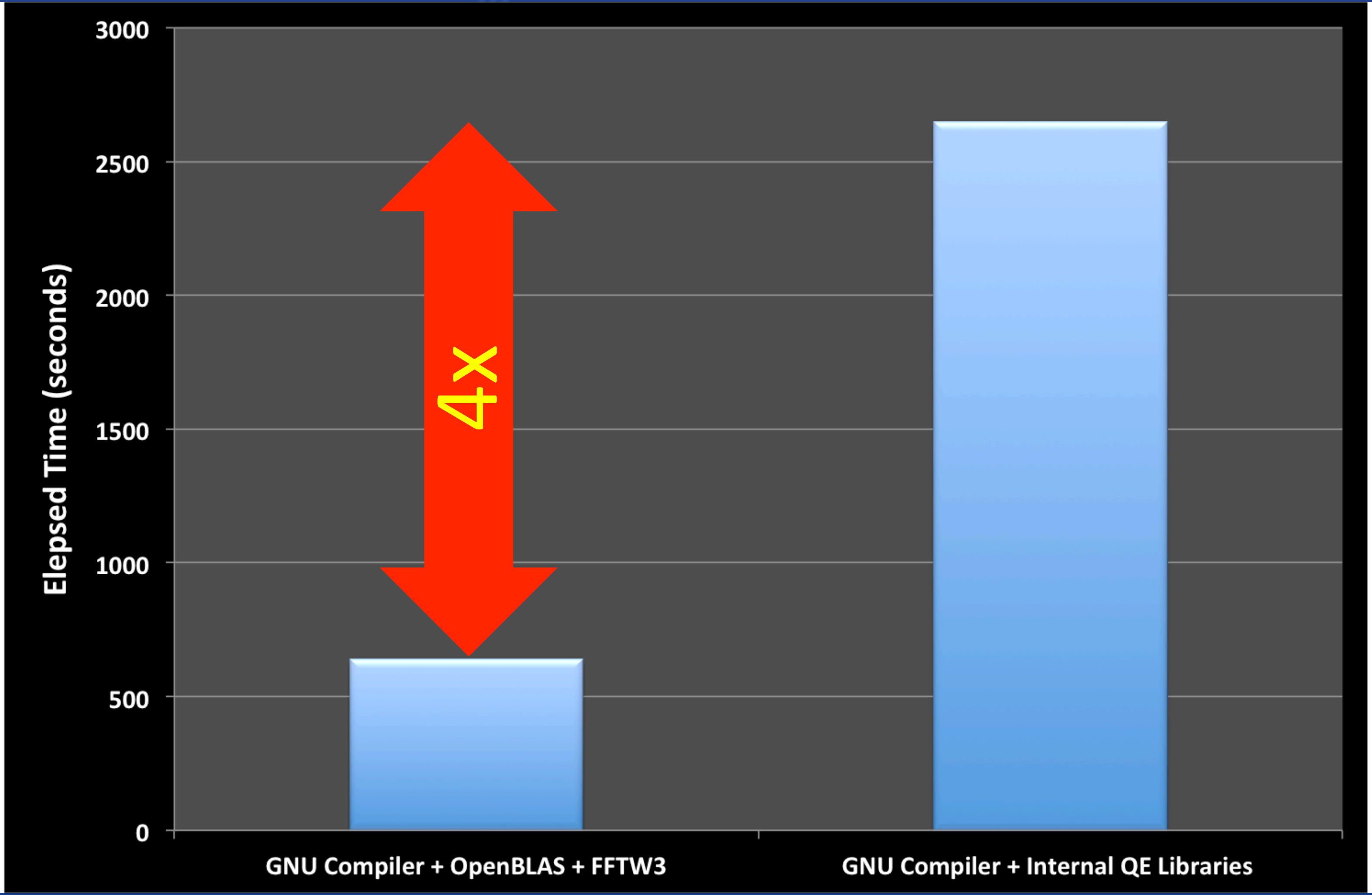


- Intel



Threading and Vectorization




3000
2500
2000
1500
1000
500
0
Elepsed Time (seconds)
4x
GNU Compiler + OpenBLAS + FFTW3
GNU Compiler + Internal QE Libraries