Performance Measurement

DSC 315: Computer Organization & Operating Systems

Dr. Laltu Sardar

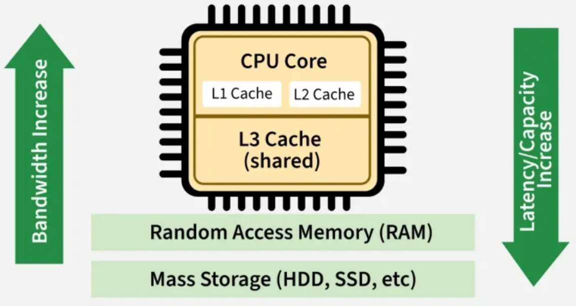
School of Data Science,
Indian Institute of Science Education and Research Thiruvananthapuram (IISER TVM)

7th January, 2026

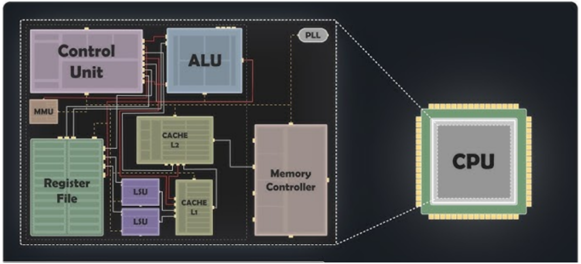
Quick Recap

Clock, Instructions, and Levels of Programming

A processor

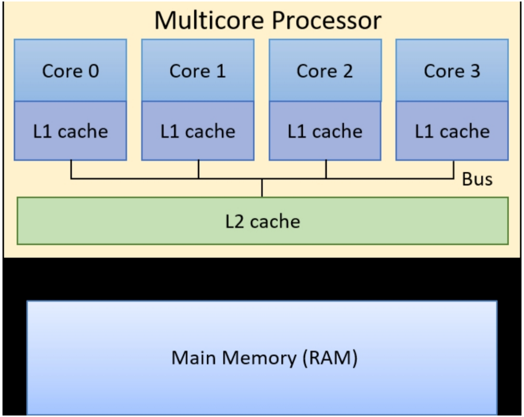


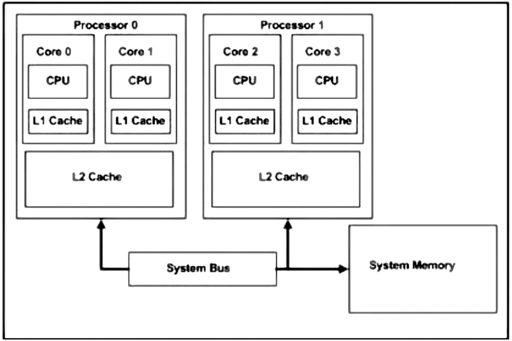
Bandwidth Increase
CPU Core
L1 Cache
L2 Cache
L3 Cache (shared)
Random Access Memory (RAM)
Mass Storage (HDD, SSD, etc)
Latency/Capacity Increase



Control Unit
ALU
PLL
MMU
CACHE L2
Register File
LSU
LSU
CACHE L1
Memory Controller
CPU

Multi-core Micro-processor





1. Here, each core is a processor
2. In some microprocessor chip, cores shares L2

Memory Hierarchy

Registers

(Fastest, Smallest, Highest cost per bit)

↓

Cache Memory (L1, L2, L3)

(Very fast, Small size)

↓

Main Memory (RAM)

(Moderate speed, Larger size)

↓

Secondary Storage (SSD, HDD)

(Slowest, Very large, Persistent)

Hierarchy of Abstraction

High-Level Code
(C, C++, Python)

$\downarrow$

Assembly Code

$\downarrow$

Machine Instructions

$\downarrow$

Hardware

1. Some Language directly generate instructions, some have different levels.

Example Codes

High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Assembly language program (for MIPS)

```
swap:
      multi $2, $5,4
      add   $2, $4,$2
      lw    $15, 0($2)
      lw    $16, 4($2)
      sw    $16, 0($2)
      sw    $15, 4($2)
      jr    $31
```

Binary machine language program (for MIPS)

```
00000000101000100000000100011000
0000000010000010000100000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

Machine understand Instructions only

Clock Cycle

- The clock signal is analogous to a heartbeat.
- Quartz Crystal Oscillators == Heart
- It sends electrical signals
- Different path requires different time delay
- Clock cycle time is determined by the critical path delay.
- Clock tick== clock cycle

Clock speed/ clock rate= # cycles per seconds
Instruction execution is the smallest unit of work.
Each instruction requires one or more clock cycles.

Performance Question

- How do we measure processor performance?
- Execution time depends on:
    - Number of instructions
    - Number of cycles per instruction
    - Clock cycle time

Same Code, Different Performance

- The same program may run differently on different processors.
- Reasons include:
  - Different instruction sets
  - Different cycles per instruction
  - Different clock speeds

Performance Measurement

Measure Performance: Execution Time

- Objective is to measure performance of a processor.
- Performance is typically measured using **execution time**.
- Execution time is the **total time taken to run a program**.
- Lower execution time implies better performance.
- Execution time depends on both software and hardware.

Measure Performance: Execution Time

- Objective is to measure performance of a processor.
- Performance is typically measured using **execution time**.
- Execution time is the **total time taken to run a program**.
- Lower execution time implies better performance.
- Execution time depends on both software and hardware.

Comparing Two Processors

- Consider two processors P1 and P2.
- The processor that completes execution earlier is faster, **better performance**.
- Same program may behave differently on different processors.

Instruction Count

- Different programs have different instruction counts.
- Same program may generate different instruction counts on different instruction set architectures.
- Instruction count alone does not determine performance.

Cycles per Instruction

- Instructions may take different number of clock cycles.
- Simple instructions usually take fewer cycles.
- Complex instructions may take more cycles.

Sometimes average CPI considered as CPI, for a instruction set.

Performance Dependency

Clock Cycle Time

- Clock cycle time is decided by hardware design.
- Smaller clock cycle time means faster clock.
- Clock cycle time directly affects execution time.

Three Key Factors

Execution time depends on:

1. Number of instructions
2. Cycles per instruction (CPI)
3. Clock cycle time

Execution Time Formula

- Total execution time is given by:

$$\sum_i (I_i \times \text{CPI}_i \times t)$$

- Where:
  - I_i is the number of instructions of type i
  - CPI_i is cycles per instruction of type i
  - t is Clock Cycle Time

Average CPI

- Average CPI is defined as:

$$\text{CPI}_{avg} = \frac{\sum_i (I_i \times \text{CPI}_i)}{\sum_i I_i}$$

- Used to simplify performance analysis.

Simplified Execution Time

- Execution Time =

$$\text{Instruction Count} \times \text{CPI}_{avg} \times \text{Clock Cycle Time}$$

- All three terms must be considered together.

Latency and Throughput

performance

Execution time

- Performance is measured using execution time.
- Lower execution time implies better performance.

Latency

- Latency is the execution time for a single program.
- It measures how long a task takes from start to finish.
- Also called response time.

Throughput

- Throughput is the amount of tasks completed per unit time.
- It measures total work done in a given period.
- Higher throughput implies better system performance.

Latency vs Throughput

- Latency focuses on individual task completion time.
- Throughput focuses on overall system productivity.
- A system can have good throughput but poor latency.

Example

- Dell processor vs Ryzen processor.
- One processor may finish a task faster.
- Another processor may complete more tasks per unit time.

Performance and Clock Cycle

- Performance depends on clock cycle time.
- Smaller clock cycle time usually improves performance.

CPU Execution Time

Elapsed Time = system time + user time + I/O time

In practical, CPU time== CPU execution time − I/O time.

- CPU execution time is the time CPU spends executing instructions.
- It excludes I/O waiting time.
- Also called user CPU time.

For Performance, CPU time = user CPU time

MIPS Metric

- MIPS means Millions of Instructions Per Second.
- Given by:

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}$$

- MIPS can be misleading across different architectures.

A problem set

Problem 1

A program runs in 10 seconds on Computer A with a 2 GHz clock. A new Computer B must run the same program in 6 seconds. Due to design changes, Computer B requires $1.2\times$ the number of clock cycles compared to Computer A.

Question: What clock rate should Computer B target?

Target Clock Rate for Computer B

Given:

- Program runtime on Computer A: 10 seconds
- Clock rate of Computer A: 2 GHz
- Desired runtime on Computer B: 6 seconds
- Computer B requires $1.2\times$ the number of clock cycles compared to A

Step 1: Clock cycles on Computer A

$$\text{Cycles}_A = 10 \times 2 \times 10^9 = 20 \times 10^9$$

Step 2: Clock cycles on Computer B $\text{Cycles}_B = 1.2 \times \text{Cycles}_A = 24 \times 10^9$

Step 3: Required clock rate for Computer B

$$\text{Clock Rate}_B = \frac{24 \times 10^9}{6} = 4 \times 10^9 \text{ Hz}$$

Answer: Target clock rate for Computer B is 4 GHz

Problem Statement 2

A compiler designer must choose between two code sequences for a given computer. The CPI for each instruction class is as follows:

- Class A: CPI = 1
- Class B: CPI = 2
- Class C: CPI = 3

For a particular high-level language statement, the instruction counts are:

- Code Sequence 1: A = 2, B = 1, C = 2
- Code Sequence 2: A = 4, B = 1, C = 1

Questions:

- Which code sequence executes the most instructions?
- Which code sequence is faster?
- What is the CPI of each code sequence?

Benchmark

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Perl interpreter	perlbench	2684	0.42	0.556	627	1774	2.83
GNU C compiler	gcc	2322	0.67	0.556	863	3976	4.61
Route planning	mcf	1786	1.22	0.556	1215	4721	3.89
Discrete Event simulation - computer network	omnetpp	1107	0.82	0.556	507	1630	3.21
XML to HTML conversion via XSLT	xalancbmk	1314	0.75	0.556	549	1417	2.58
Video compression	x264	4488	0.32	0.556	813	1763	2.17
Artificial Intelligence: alpha-beta tree search (Chess)	deepsjeng	2216	0.57	0.556	698	1432	2.05
Artificial Intelligence: Monte Carlo tree search (Go)	leela	2236	0.79	0.556	987	1703	1.73
Artificial Intelligence: recursive solution generator (Sudoku)	exchange2	6683	0.46	0.556	1718	2939	1.71
General data compression	xz	8533	1.32	0.556	6290	6182	0.98
Geometric mean	–	–	–	–	–	–	2.36

SPEC Reference Time and Performance Metrics

Reference Time

- A fixed execution time provided by the SPEC organization for each benchmark.
- Measured on a standardized reference machine.
- Used as a baseline to enable fair performance comparisons across different systems.
- Independent of the system being evaluated.

SPECratio

- A normalized performance metric defined as:

$$\text{SPECratio} = \frac{\text{Reference Time}}{\text{Execution Time}}$$

- Indicates how much faster or slower the tested system is compared to the reference.
- Higher values indicate better performance.
- A value of 1 means performance equal to the reference system.

SPEC -> System Performance Evaluation Cooperative

Geometric Mean

- The overall SPECspeed score is the geometric mean of all individual SPECratios.
- Computed as:

$$\left(\prod_{i=1}^{N} \text{SPECratio}_i\right)^{1/N}$$

- Used because performance ratios combine multiplicatively.
- Prevents any single benchmark from dominating the final score.

Dr. Laltu Sardar, Assistant Professor, IISER Thiruvananthapuram
`laltu.sardar@iisertvm.ac.in`, `laltu.sardar.crypto@gmail.com`
Course webpage: https://laltu-sardar.github.io/courses/corgos_2026.html.