

Lecture 3

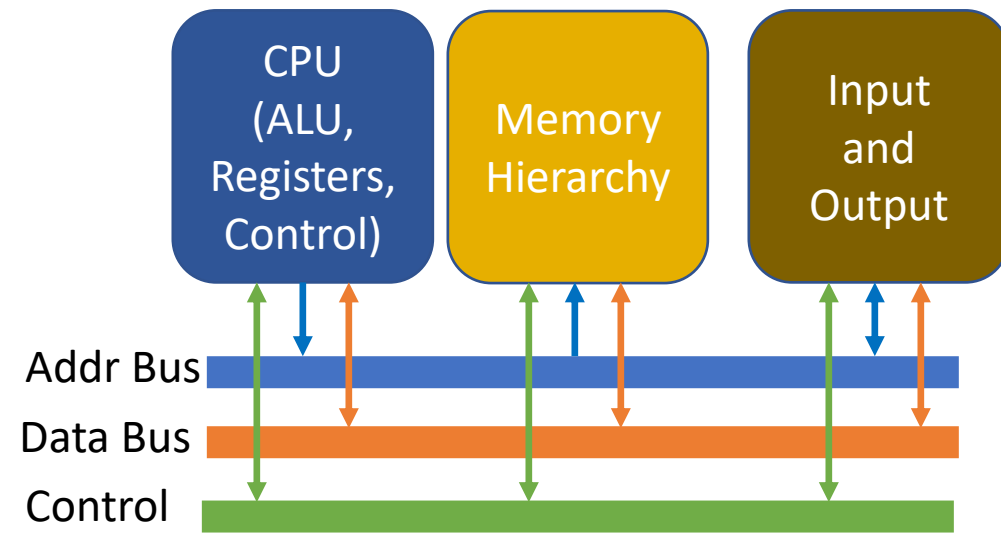
Computer Organization and Assembly Language CS/EE 320

Shahid Masud
Spring 2025

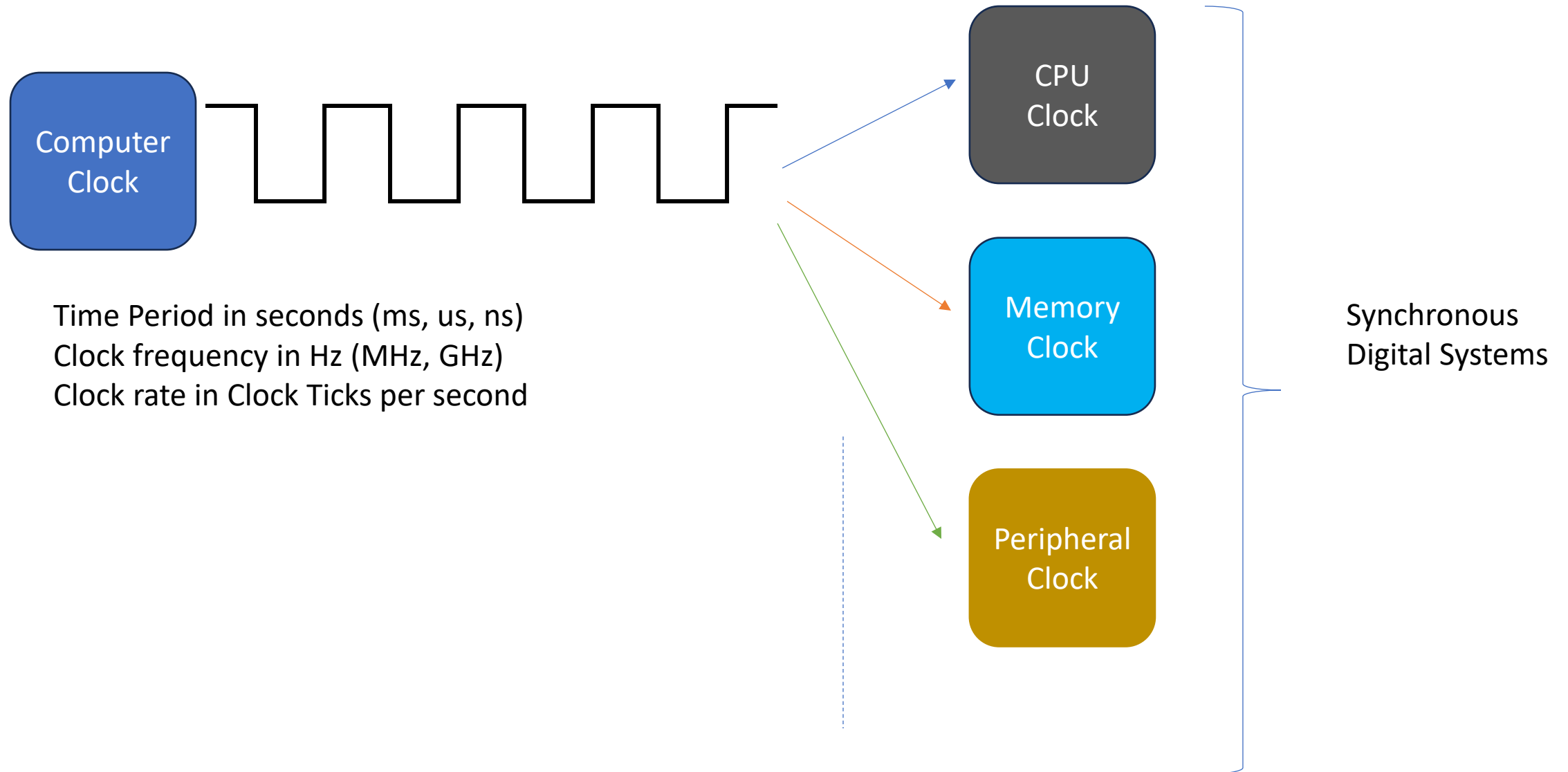
- Computer Performance Measurement Techniques
 - Power Dissipation
 - Execution Time
 - CPI
 - MIPS
 - FLOPS
 - Amdahl's Law (parallel and multicore - next lecture)
 - SPEC Benchmarks

- **Stored Program concept**
- Main memory stores programs and data
- ALU operating on binary data
- Control unit interpreting instructions from memory and executing
- Input and output equipment is operated by control unit
- Program is **sequentially accessed** from memory

Von-Neumann Stored Program Architecture



Clock in Computer



Power Dissipation as Measure of Performance

- ***Power*** = $\frac{1}{2} C \cdot V^2 \cdot f$
- $\frac{1}{2}$ factor is due to average switching of logic circuits, some 0 to 1, others 1 to 0
- ***C*** represents complexity of chip components in terms of overall Capacitance. More components (i.e. transistors) means more capacitance.
- ***V*** is the operating voltage
- ***f*** is the operating frequency

- Look back at i7 power benchmark for PC/Laptop
 - At 100% load: 258W
 - At 50% load: 170W (66%)
 - At 10% load: 121W (47%)
- Google data center
 - Mostly operates at 10% – 50% load
 - At 100% load less than 1% of the time
- **Design processors to make power proportional to load**

} Range



Execution Time as Measure of Performance

- **Execution Time**, for **User Specific Computing**
- **Number of jobs / tasks completed per unit time**, for warehouse or server class computing - **Throughput**

- Define Performance = $1/\text{Execution Time}$
- “Computer X is n time faster than Computer Y”

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution_Time}_Y}{\text{Execution_Time}_X} \\ = n$$

- Example: time taken to run a program
 - 10s on Computer A, 15s on Computer B
 - $\text{Execution Time}_B / \text{Execution Time}_A$
 $= 15\text{s} / 10\text{s} = 1.5$
 - So Computer A is 1.5 times faster than Computer B

Measuring Performance in terms of CPU Execution Time

- Elapsed time
 - Total response time, including all aspects
 - Processing, I/O, OS overhead, idle time
 - Determines system performance
- CPU time
 - Time spent processing a given job
 - Discounts I/O time, other jobs' shares
 - Comprises **user CPU time** and **system CPU time**
 - Different programs are affected differently by CPU and system performance

$$\text{CPU Execution Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock Cycle}}$$

$$\text{CPU Execution Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock Cycle}}$$

- **Performance can be improved by:**
 - Reducing the number of clock cycles
 - Increasing clock rate
 - Hardware designer must find trade off of clock rate against cycle count

- **Computer A:** 2.0 GHz clock, 10s CPU time
- Designing **Computer B:**
 - Aim for **6s CPU time**
 - Causes **1.2 ×** clock cycles
- **How much fast the clock of Computer B should be?**

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6s}$$

$$\begin{aligned}\text{Clock Cycles}_A &= \text{CPU Time}_A \times \text{Clock Rate}_A \\ &= 10s \times 2\text{GHz} = 20 \times 10^9\end{aligned}$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4\text{GHz}$$

CPI as Measure of Performance

CPI = Cycles Per Instruction



- Each microprocessor specifies the number of Clock Cycles or Clock Ticks that specific type of instructions will consume
- CPI is an average number considering occurrence of different types of instructions in a program

$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$

$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- Instruction Count I_c for a program
 - Determined by program, ISA and compiler
- Average cycles per instruction
 - Determined by CPU hardware
 - Different instructions have different CPI
 - Average CPI affected by instruction mix

CPU Time and CPI Example



- **Computer A:** Cycle Time = 250ps, CPI = 2.0
- **Computer B:** Cycle Time = 500ps, CPI = 1.2
- Same ISA
- Which computer is faster, and by how much?

$$\begin{aligned}\text{CPU Time}_A &= \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A \\ &= 1 \times 2.0 \times 250\text{ps} = 1 \times 500\text{ps}\end{aligned}$$

A is faster...

$$\begin{aligned}\text{CPU Time}_B &= \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B \\ &= 1 \times 1.2 \times 500\text{ps} = 1 \times 600\text{ps}\end{aligned}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{1 \times 600\text{ps}}{1 \times 500\text{ps}} = 1.2$$

...by this much

- As different instruction classes take different numbers of cycles, we use **Average CPI**

$$\text{Clock Cycles} = \sum_{i=1}^n (\text{CPI}_i \times \text{Instruction Count}_i)$$

The number of clock cycles varies for different types of instructions such as load, branch, mult etc.

Let CPI_i be the number of cycles required for instruction type i

Let I_i be the number of executed instructions of type i in a given program

The overall CPI is as follows:

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_c}$$

Factors Affecting CPI



The **processor time** T needed to execute a given program can be expressed as:

$$T = I_c \times CPI \times \tau$$

Time to transfer data depends on memory cycle time that can vary significantly viz a viz processor cycle time. Thus:

$$T = I_c \times [p + (m \times k)] \times \tau$$

Where:

p = number of processor cycles needed to decode and execute the instruction

m = number of memory references needed

k = ratio between memory cycle time and processor cycle time

$\tau = 1/f$; cycle time

I_c = instruction count

$$\text{Clock Cycles} = \sum_{i=1}^n (\text{CPI}_i \times \text{Instruction Count}_i)$$

- Alternative compiled code sequences using instructions in classes (or types) A, B, C

Class	A	B	C
CPI for class	1	2	3
I_c in sequence 1	2	1	2
I_c in sequence 2	4	1	1

■ Sequence 1: $I_c = 5$

- Clock Cycles
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$
- Avg. CPI = $10/5 = 2.0$

■ Sequence 2: $I_c = 6$

- Clock Cycles
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$
- Avg. CPI = $9/6 = 1.5$

MIPS as Measure of Performance

MIPS – Million Instructions Per Second

- MIPS is a popular performance measure in RISC processors.
- **Ideal value is to obtain a CPI of 1.0**, million clock cycles would process million instructions
- (Microprocessor with Interlocked Pipelining Stages)



- A common measure of CPU performance is the rate at which instructions are executed
- MIPS is abbreviation for Millions Instructions per second
- How many MIPS can a CPU do is its **MIPS rate**?
- In terms of equations:

$$MIPS\ rate = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

$$\text{MIPS rate} = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6} \quad (2.3)$$

EXAMPLE 2.2 Consider the execution of a program that results in the execution of 2 million instructions on a 400-MHz processor. The program consists of four major types of instructions. The instruction mix and the *CPI* for each instruction type are given below, based on the result of a program trace experiment:

Instruction Type	<i>CPI</i>	Instruction Mix (%)
Arithmetic and logic	1	60
Load/store with cache hit	2	18
Branch	4	12
Memory reference with cache miss	8	10

The average *CPI* when the program is executed on a uniprocessor with the above trace results is $CPI = 0.6 + (2 \times 0.18) + (4 \times 0.12) + (8 \times 0.1) = 2.24$. The corresponding MIPS rate is $(400 \times 10^6) / (2.24 \times 10^6) \approx 178$.

Pitfall: MIPS as a Performance Metric



- MIPS: Millions of Instructions Per Second
 - Doesn't account for
 - Differences in ISAs between computers
 - Differences in complexity between instructions
- CPI varies between programs on a given CPU

MFLOPS as Measure of Performance

- Mega Flops, Giga Flops and Tera Flops are used to measure performance of super computers in Scientific Programming.

$$\text{MFLOPS Rate} = \frac{\text{Number of Executed Floating Point Instructions in a Program}}{\text{Execution Time} \times 10^6}$$

MFLOPS = Millions of Floating Point Operations per second

Benchmarks as Measure of Performance

- Benchmarks are used to evaluate performance of full computer systems. CPI measure is only for microprocessors.
- SPEC benchmarks are standard in Computer Architecture study.
- A benchmark contains several programs in a particular class of computing, i.e. integer calculations, floating-point computations, etc.
- SPEC Rating = Geometric Mean (N^{th} root) of product of performance obtained from running all programs in a particular class in the benchmark.

- Due to differences in instruction sets, the instruction execution rate is not a valid means of comparing performance of different architectures in the form of MIPS or MFLOPS.
- Performance of a program may not be useful in determining how that processor will perform on a very different type of application.
- Benchmarks provide guidance to customers to decide which system to buy and useful support to vendors and designers to design systems that meet benchmark goals
- Characteristics of benchmark programs:
 - Written in high-level languages hence portable across machines
 - Representative of a particular kind of programming domain or paradigm
 - Can be measured easily
 - It has a wide distribution

- Programs used to measure performance
 - Supposedly typical of actual workload
- Standard Performance Evaluation Corp (SPEC)
 - Develops benchmarks for CPU, I/O, Web, ...
- SPEC CPU2006
 - Elapsed time to execute a selection of programs
 - Negligible I/O, so focuses on CPU performance
 - Normalize relative to reference machine
 - Summarizes **geometric mean** of performance ratios

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

Example: SPECspeed 2017 Integer benchmarks on a 1.8 GHz Intel Xeon E5-2650L

<i>Description</i>	<i>Name</i>	<i>Instruction Count x 10⁹</i>	<i>CPI</i>	<i>Clock cycle time (seconds x 10⁻⁹)</i>	<i>Execution Time (seconds)</i>	<i>Reference Time (seconds)</i>	<i>SPECratio</i>
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

Cost of Spec benchmarks



https://www.spec.org/order.html



Purchase Current SPEC Benchmark Suites

SPEC benchmarks are licensed to companies, universities, and organizations around the world; to find out if your organization is a member of SPEC's consortium, please see our [current membership](#). Or to find out if your organization has an existing license for a SPEC product please contact SPEC at info@spec.org.

SPEC provides [special non-profit/educational discounts](#) for organizations with 501C or equivalent standings on selected SPEC benchmark suites. Staff may request proof of qualification. All academic orders are licensed to a professor or full-time staff member.

[Licensing Info](#) [MD5 Checksums](#) [Payment Options](#) [Refund Info](#) [Tax](#)

[SPECapc Benchmark Requirements](#) [Embedded Benchmark Requirements](#)

ADASMark	Purchase (\$7500)
SPECaccel 2023	Purchase (\$2000)
SPEC ACCEL V1.4	Purchase (\$2000)
AutoBench 2.0 (includes AutoBench 1.1)	Purchase (\$2500)
BrowsingBench	Purchase (\$2500)
Chauffeur WDK V2.0.0	Purchase (\$50)
SPEC Cloud IaaS 2018 V1.1	Purchase (\$2000)
CoreMark-Pro (license only)	Purchase (\$2500)
SPEC CPU 2017 V1.1.9	Purchase (\$1000)
DENBench	Purchase (\$2500)
FPMARK	Purchase (\$2500)

The PC / Laptop Computer Organization

- Comprises:**
- CPU
 - Co-Proc / Accelerator
 - Memory Hierarchy
 - Bus Hierarchy
 - Input / Output
 - Peripherals
 - Power

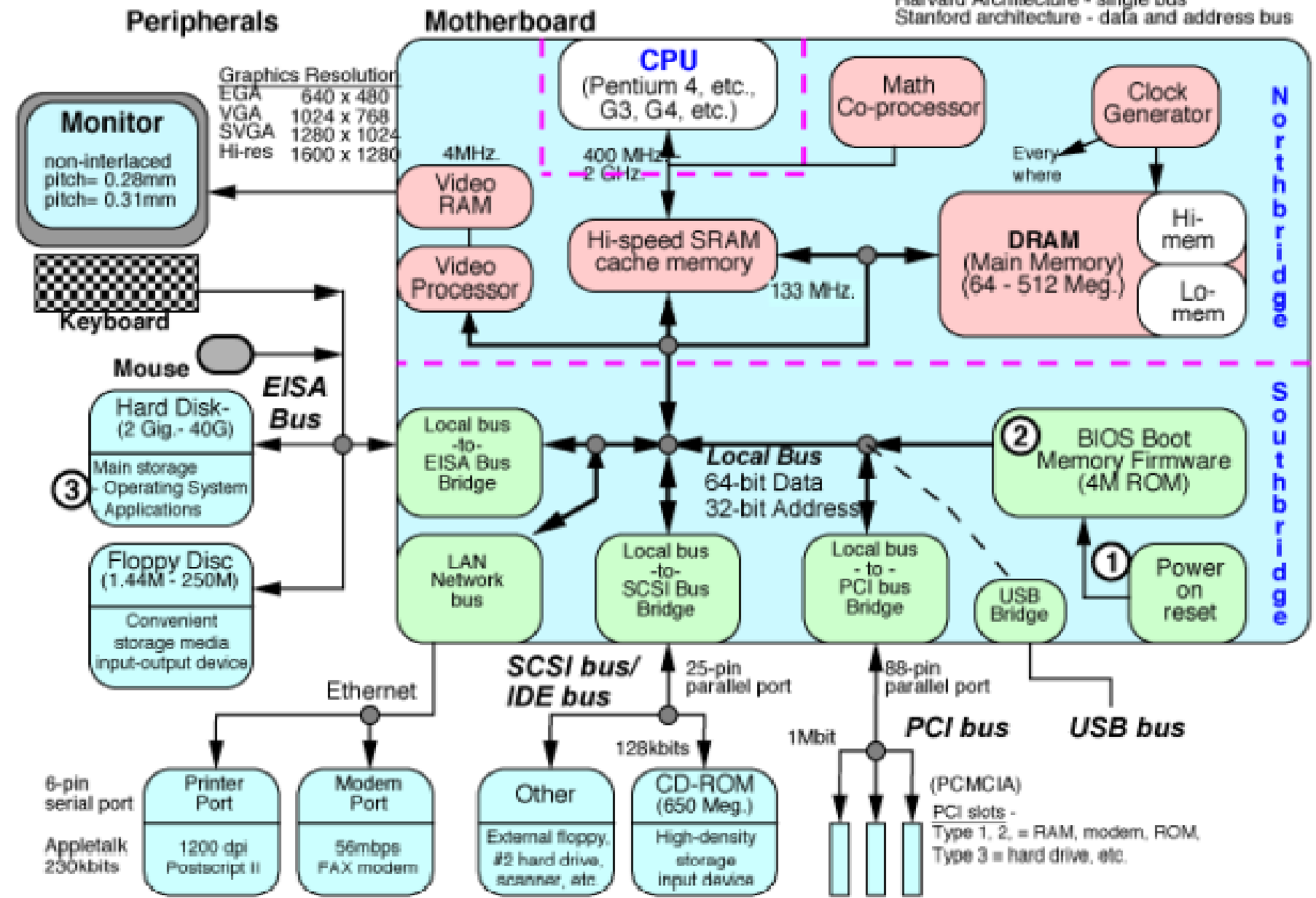
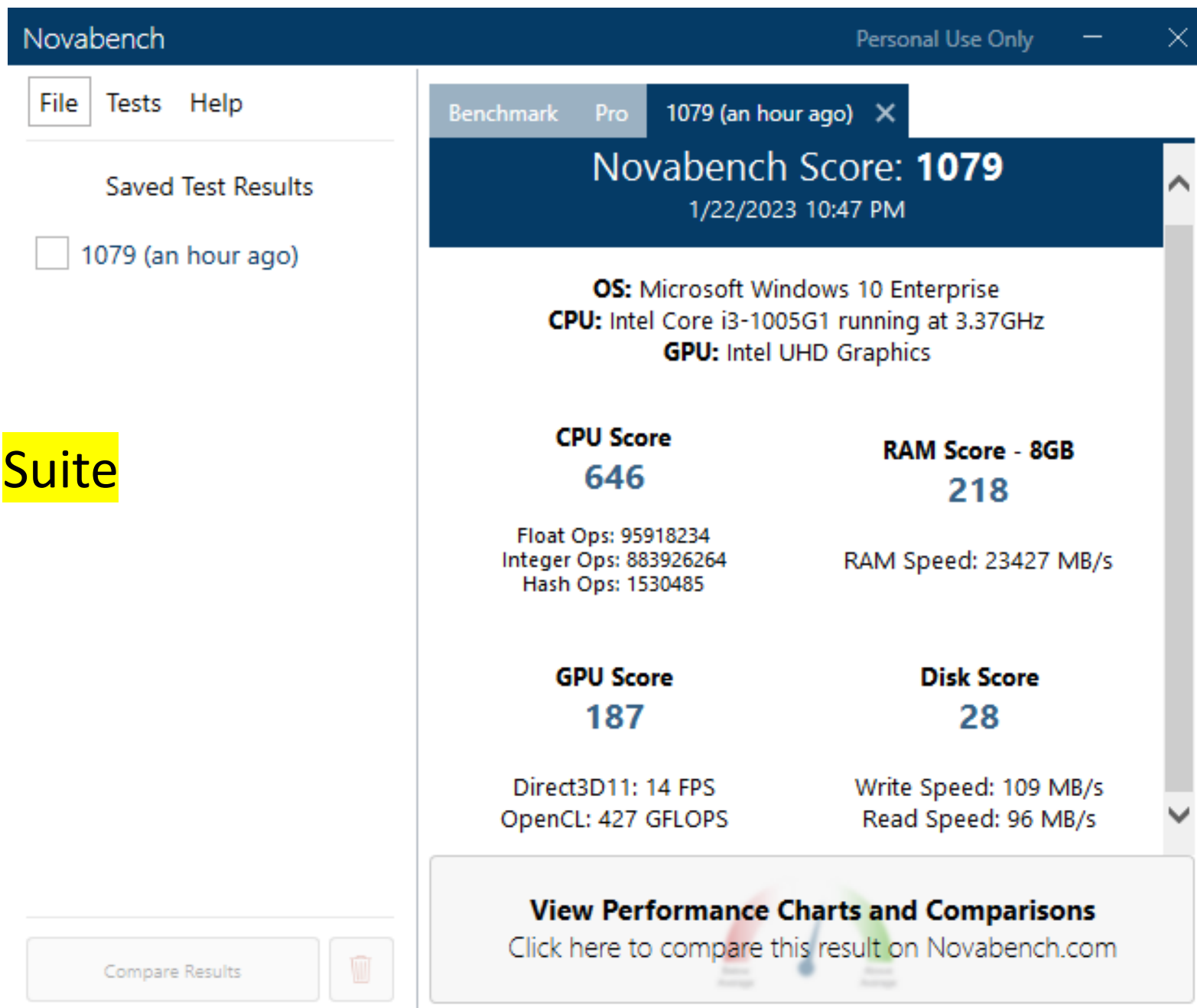


Figure 4. The architecture of personal computers

Novabench Benchmark Suite





Benchmark Result

Jan 22, 2023 at 18:55

Novabench Score **1079**

 Intel Core i3-1005G1 at 3.4 GHz

 Intel UHD Graphics

 8GB DDR4  HDD

 Microsoft Windows 10 Enterprise

CPU Score
646

Floating Ops: 95918234
Integer Ops: 883926264
Hash Ops: 1530485

RAM Score
218

RAM Speed: 23427 MB/s

GPU Score
187

Direct3D 11: 14 FPS
OpenCL: 427 GFLOPS

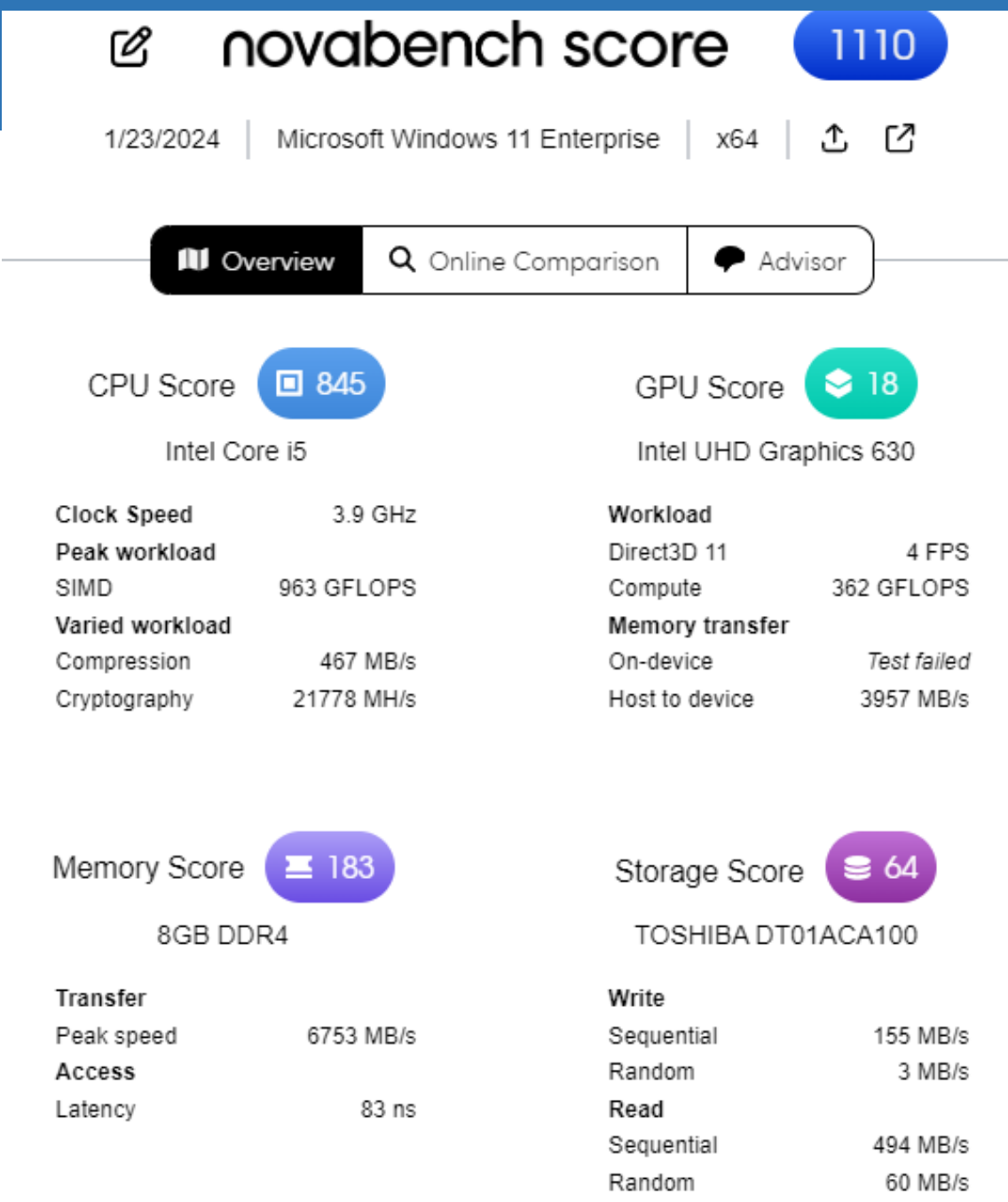
Disk Score
28

Write Speed: 109 MB/s
Read Speed: 96 MB/s

Score Comparison

Overview	Part Performance	Baseline Comparison						
<p>This system's overall rank, and its score relative to its age</p> <table> <tr> <th data-bbox="1345 793 1691 1071">System Era</th><th data-bbox="1691 793 2035 1071">Relative Score</th><th data-bbox="2035 793 2354 1071">Overall Rank</th></tr> <tr> <td data-bbox="1345 893 1691 1071">2018 - 2019</td><td data-bbox="1691 893 2035 1071">-30% Era average: 1546</td><td data-bbox="2035 893 2354 1071">49th All time percentile</td></tr> </table>			System Era	Relative Score	Overall Rank	2018 - 2019	-30% Era average: 1546	49th All time percentile
System Era	Relative Score	Overall Rank						
2018 - 2019	-30% Era average: 1546	49th All time percentile						

Novabench - today



Efficiency



CPU Efficiency

Idle to peak efficiency

99%

Peak performance efficiency

13 GFLOPS/Watt

CPU Score 787

Intel Core i5 @ 3.89 GHz

Memory Score 181

8GB DDR4

Peak Workloads

Tests that measure the CPU's peak performance with low memory access overhead

SIMD

921 GFLOPS

The SIMD benchmark evaluates a processor's ability to handle multiple data elements in a single instruction. SIMD instructions are used to perform high-speed mathematical calculations, such as those used in image processing and data analysis.

Scalar

329 GFLOPS

The scalar CPU benchmark test is a performance evaluation that measures a processor's ability to handle a single data element in each instruction. This type of test is designed to measure a processor's ability to perform basic mathematical operations.

Varied Workloads

Tests that measure the CPU's performance with varied levels of memory access

Memory Transfer

Transfer Speed

6529 MB/s

This test measures the peak rate at which data can be transferred from the main memory to the CPU, which is an important factor in the performance of data-heavy applications.

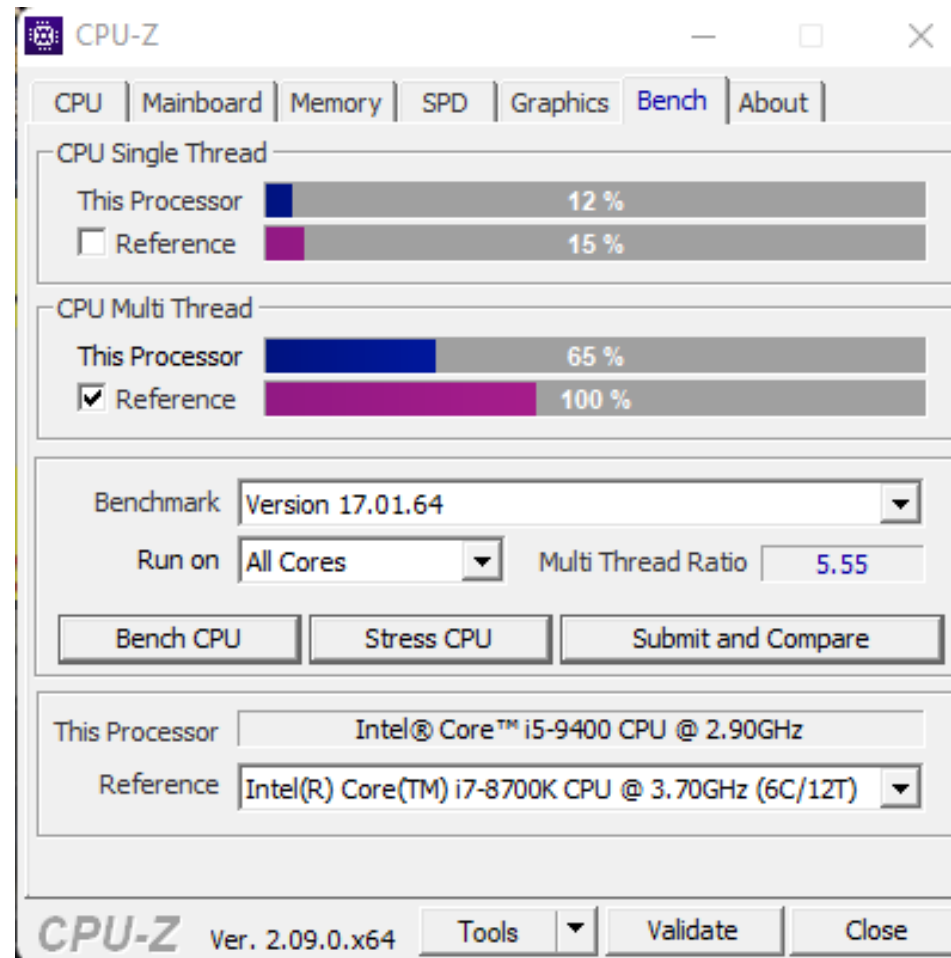
Memory Latency

Access Latency

88 ns

This test measures the average time taken to access random locations in the main memory from the CPU, which is an important factor in the system's overall performance.

- <https://cpux.net/cpu-benchmark-online> online
- <https://cpu-benchmark.org/> online
- <https://www.matthew-x83.com/online/cpu-benchmark.php> online
- <https://www.cpubid.com/software/cpu-z.html> download
- <https://novabench.com/> download
- <https://www.userbenchmark.com/Software> download



**Steel Solutions: The Best Choice for...**

Get high-quality steel solutions for all your...

sponsored by: Steel Solutions

[LEARN MORE](#)

Rank

228868/498813

Score

60179

Speed

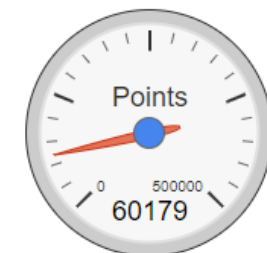
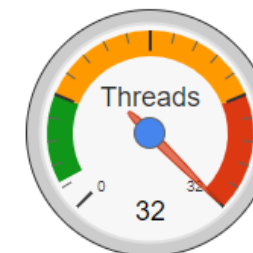
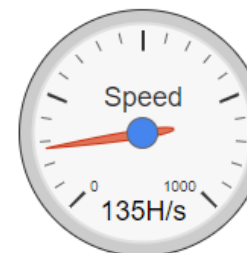
135H/s

Owner message:

office pc

CPU	intel core i5
GPU	ANGLE (Intel, Intel(R) UHD Graphics 630 (0x00003E92) Direct3D11 vs_5_0 ps_5_0, D3D11)
Logical Processors	6
Points	60179
Initial Rank	228868/498813
Current Rank	228868/498813
Speed	~135H/s
Threads	32
Duration	447s
Start Date	23 January 2024, 07:46:35
Finish Date	23 January 2024, 07:54:02
Browser	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
Permalink	https://cpux.net/b/g2x5o4

Hash functions per second






https://www.userbenchmark.com/Software

UserBenchmark PAK-User US







CPU GPU SSD HDD RAM USB EFPS FPS SkillBench YouTube

UserBenchmark of the month

 Gaming  Desktop  ProGaming

How it works

- Download and run UserBenchmark
- CPU tests include: integer, floating and string
- GPU tests include: six 3D game simulations
- Drive tests include: read, write, sustained write and mixed IO
- RAM tests include: single/multi core bandwidth and latency
- [SkillBench \(space shooter\)](#) tests user input accuracy
- Reports are generated and presented on userbenchmark.com
- Identify the strongest components in your PC
- See speed test results from other users
- [Compare](#) your components to the current market leaders
- Explore your best upgrade options with a [virtual PC build](#)
- Compare your [in-game FPS](#) to other users with your hardware

 CPU
 GPU
 SSD
 HDD
 RAM
 USB

- P&H Textbook, Sections 1.6 to the end of chapter 1.