Lecture 3

Computer Organization and Assembly Language CS/EE 320

Shahid Masud

Spring 2025

Topics

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

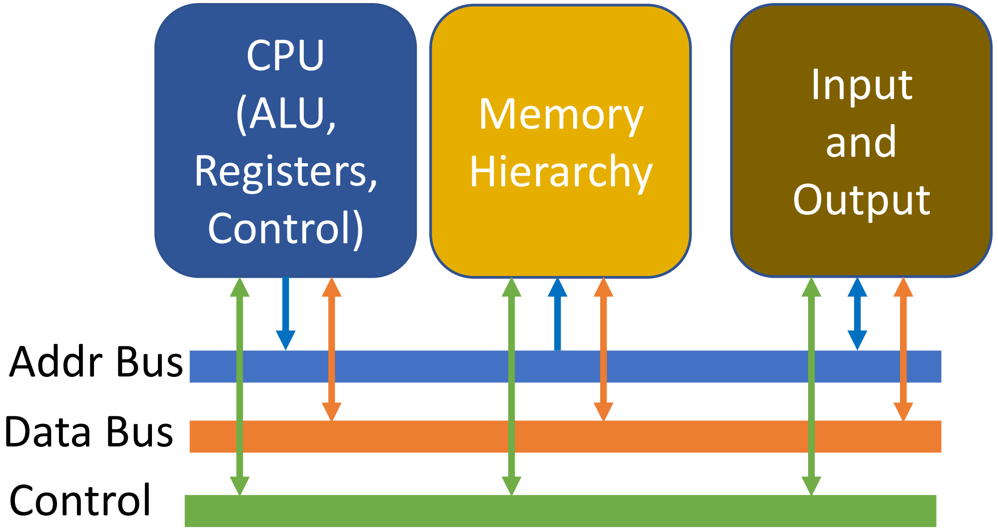
Von-Neumann

- **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



Power Dissipation as Measure of Performance

Power Dissipation in Digital Systems

- $\boldsymbol{Power = \frac{1}{2} C \,.\, V^2 \,.\, f}$
- ½ 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

Low Power at Idle

- Look back at i7 power benchmark
  - 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**

Execution Time as Measure of Performance

Computer Performance - What do we want to measure?

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

Evaluate Relative Performance

- Define Performance = 1/Execution Time
- "Computer X is n time faster than Computer Y"

$$\frac{Performance_X}{Performance_Y} = \frac{Execution\_Time_Y}{Execution\_Time_X}$$

$$= n$$

- Example: time taken to run a program
  - 10s on Computer A, 15s on Computer B
  - Execution Time_B / Execution Time_A = 15s / 10s = 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

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

CPU Time

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

- **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

CPU Time

- **Computer A**: 2GHz clock, 10s CPU time
- Designing **Computer B**
  - Aim for 6s CPU time
  - Causes 1.2 × clock cycles
- How fast must Computer B clock can be?

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

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

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

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

Instruction Count and CPI

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

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

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

- Instruction Count $\mathrm{I_c}$ for a program
  - Determined by program, ISA and compiler
- Average cycles per instruction
  - Determined by CPU hardware
  - If 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?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster…

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

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

…by this much

CPI - Averaging

- If different instruction classes take different numbers of cycles, we use average CPI:

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

Formula for Overall CPI

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

Let $\mathrm{CPI_i}$ be the number of cycles required for instruction type i

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

The overall CPI is as follows:

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

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

τ = 1/f ; cycle time

I_c = instruction count

CPI Example

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

- 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
= 2×1 + 1×2 + 2×3
= 10
  - Avg. CPI = 10/5 = 2.0
- Sequence 2: $I_C = 6$
  - Clock Cycles
= 4×1 + 1×2 + 1×3
= 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

MIPS Formula and Example

- 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:

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

Example of MIPS

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

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	***CPI***	**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

FLOPS – Floating-Point operations per second

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

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

MFLOPS = Millions of Floating Point Operations per second

Benchmarks as Measure of Performance

SPEC Benchmarks

- 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 may contain 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.

Why Benchmarks?

- 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

SPEC CPU Benchmark

- 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

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

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

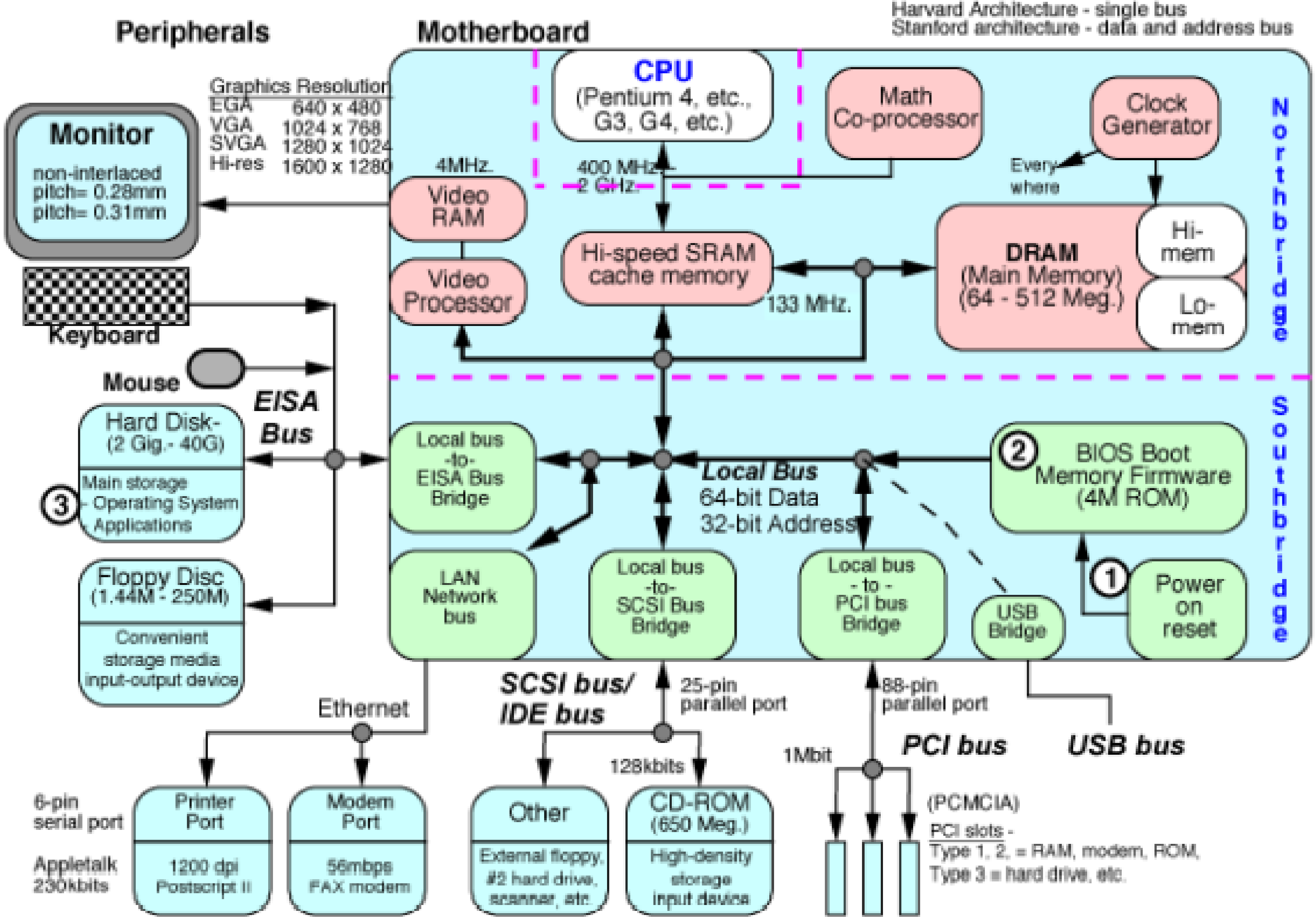
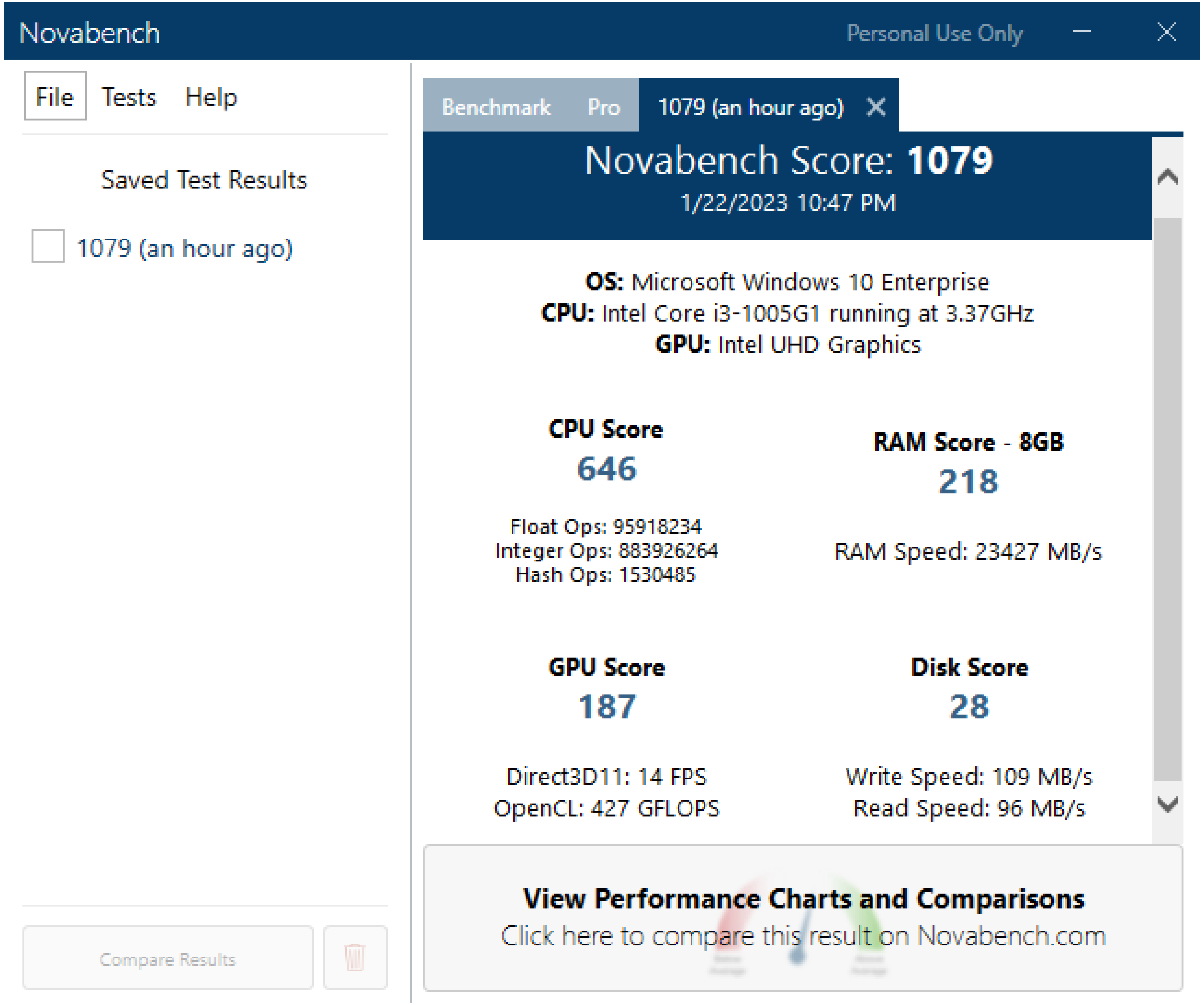


Figure 4. The architecture of personal computers

A survey of ICT: Evolution of architectures, models and layers
August 2016, Conference: International Symposium on Intelligent Systems and Informatics, Serbia
Volume: 14, Imre Petkovic, Ármin Petkovics, Janos Simon


Novabench
Personal Use Only
File
Tests
Help
Saved Test Results
1079 (an hour ago)
Compare Results
Benchmark
Pro
1079 (an hour ago)
Novabench Score: 1079
1/22/2023 10:47 PM
OS: Microsoft Windows 10 Enterprise
CPU: Intel Core i3-1005G1 running at 3.37GHz
GPU: Intel UHD Graphics
CPU Score
646
Float Ops: 95918234
Integer Ops: 883926264
Hash Ops: 1530485
RAM Score - 8GB
218
RAM Speed: 23427 MB/s
GPU Score
187
Direct3D11: 14 FPS
OpenCL: 427 GFLOPS
Disk Score
28
Write Speed: 109 MB/s
Read Speed: 96 MB/s
View Performance Charts and Comparisons
Click here to compare this result on Novabench.com

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

This system's overall rank, and its score relative to its age

System Era	Relative Score	Overall Rank
2018 - 2019	-30%	49th
	Era average: 1546	All time percentile

Novabench - today

Efficiency

CPU Efficiency

Idle to peak efficiency

99%

Peak performance efficiency

13 GFLOPS/Watt

novabench score 1110

1/23/2024 | Microsoft Windows 11 Enterprise | x64

Overview | Online Comparison | Advisor

CPU Score 845

Intel Core i5

Clock Speed	3.9 GHz
Peak workload	
SIMD	963 GFLOPS
Varied workload	
Compression	467 MB/s
Cryptography	21778 MH/s

GPU Score 18

Intel UHD Graphics 630

Workload	
Direct3D 11	4 FPS
Compute	362 GFLOPS
Memory transfer	
On-device	*Test failed*
Host to device	3957 MB/s

Memory Score 183

8GB DDR4

Transfer	
Peak speed	6753 MB/s
Access	
Latency	83 ns

Storage Score 64

TOSHIBA DT01ACA100

Write	
Sequential	155 MB/s
Random	3 MB/s
Read	
Sequential	494 MB/s
Random	60 MB/s

Novabench – latest version

CPU Score 787

Intel Core i5 @ 3.89 GHz

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 Score 181

8GB DDR4

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

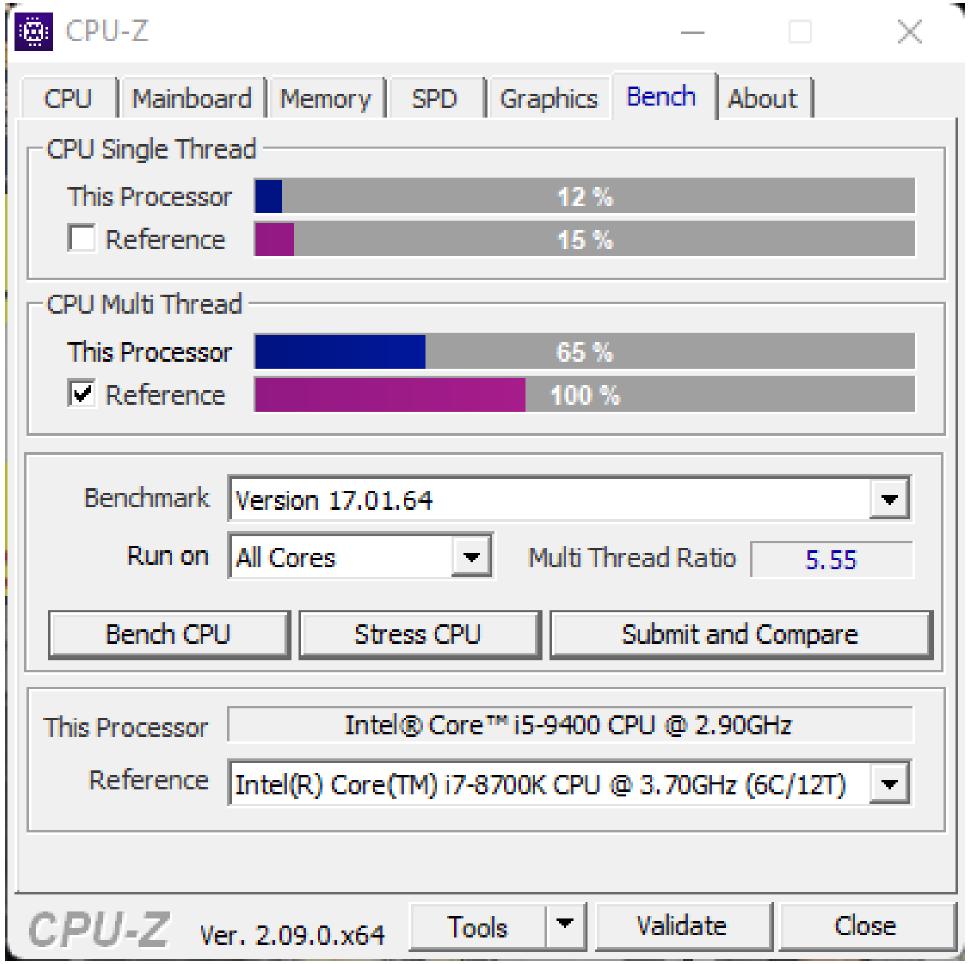
88 ns

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.

Online CPU Benchmarks

- https://cpux.net/cpu-benchmark-online online
- https://cpu-benchmark.org/ online
- https://www.matthew-x83.com/online/cpu-benchmark.php online
- https://www.cpuid.com/softwares/cpu-z.html download
- https://novabench.com/ download
- https://www.userbenchmark.com/Software download

CPU-Z


CPU-Z
CPU
Mainboard
Memory
SPD
Graphics
Bench
About
CPU Single Thread
This Processor
12 %
Reference
15 %
CPU Multi Thread
This Processor
65 %
Reference
100 %
Benchmark
Version 17.01.64
Run on
All Cores
Multi Thread Ratio
5.55
Bench CPU
Stress CPU
Submit and Compare
This Processor
Intel® Core™ i5-9400 CPU @ 2.90GHz
Reference
Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz (6C/12T)
CPU-Z
Ver. 2.09.0.x64
Tools
Validate
Close

Cpux.net

online

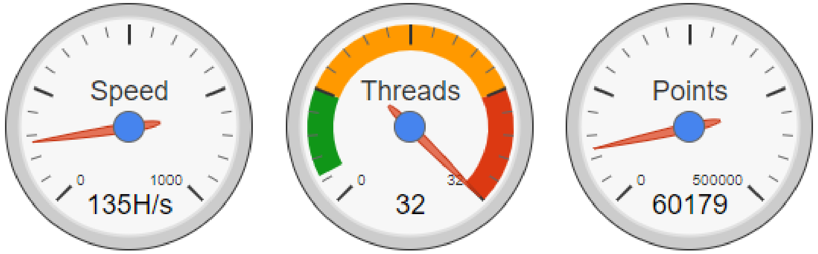
https://cpux.net/cpu-benchmark-online/g2x5o4

Rank	Score	Speed
228868/498813	60179	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 Hash functions per second
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

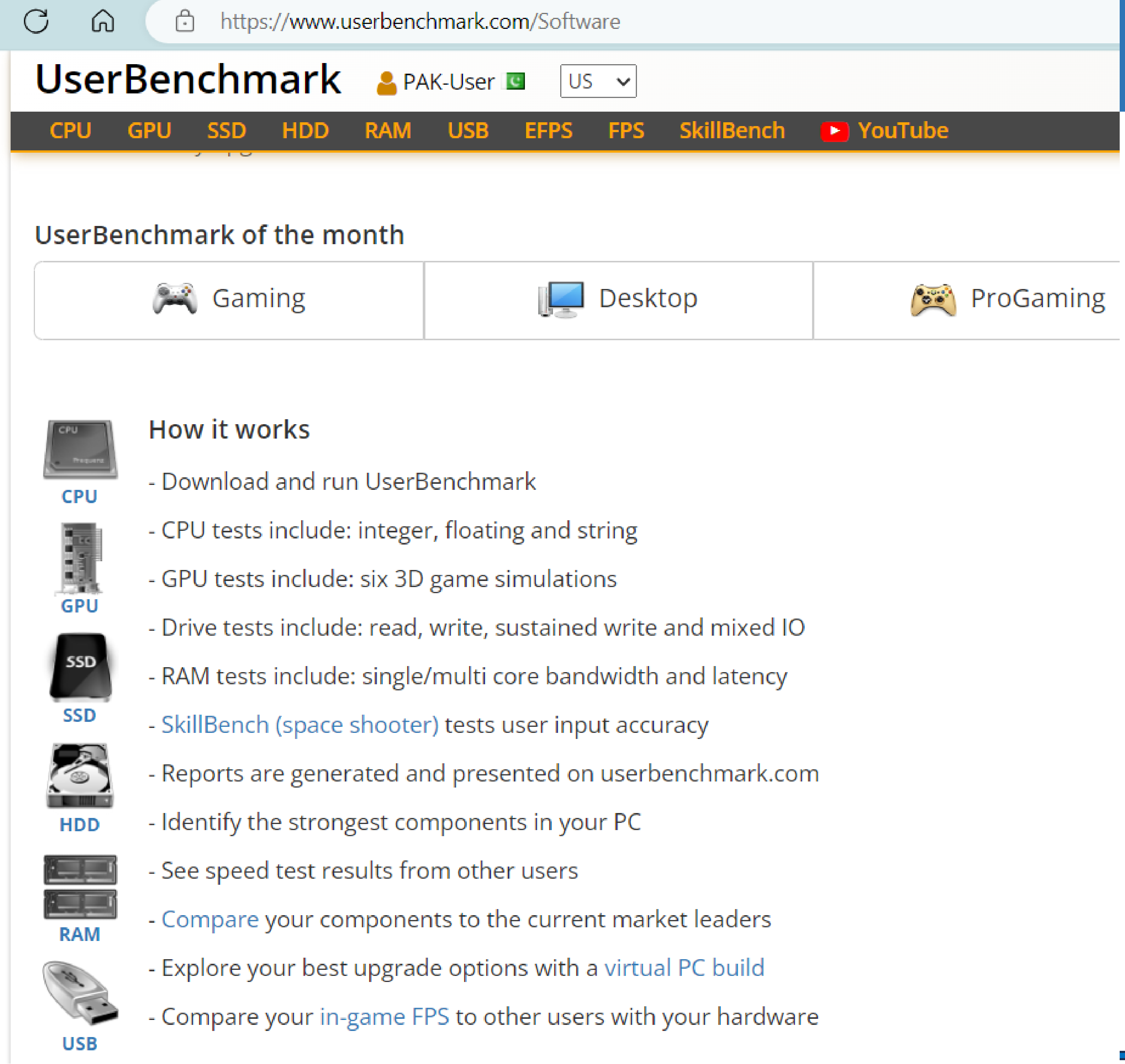


Matthew-x83.com online performance

https://www.matthew-x83.com/online/cpu-benchmark.php

Type of Test - Algorithms	Progress	Single Thread Test	
Simple do-while cycle with a variable increment		55	seconds
Some Mathematical functions - Javascript built-in object Math		19	seconds
Linear Algebra - Matrix multiplication		21	seconds
Multiple LZW Compression/Decompression Operations		43	seconds
Multiple MD5 operations on a 5MB dictionary		19	seconds
Multiple AES Encryption/Decryption operations on 100kb text		13	seconds
	Result:	Elapsed time: 173 Seconds	
Multi Thread Test: Test with 6 (Low Load) or 12 (Middle Load) or 24 (High-end Load) Threads by simultaneously combining the algorithms listed above. Threads are subprocesses that run concurrently (simplified explanation)		Elapsed time: 118 Seconds	
Benchmark Result: 12015			

UserBenchmark



Readings

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