CS 2340 - Computer Architecture

10 CPU Performance
Dr. Alice Wang

Please get out your calculators - I will need help!

Housekeeping

- Exam 1 on Tues, Oct 7 *(check your registration?)*
  - You can use a scientific calculator (not graphing)
  - You can bring three cheat sheets total:
    - 2 double sided sheet (4 sides) of cheat sheets are allowed.
    - MIPS Reference “Green card”. You can print this onto 1 double sided sheet (2 sides).
- Optional Exam 1 review will be
  - **Date/Time:** Thursday, Oct 2, 12-1pm
  - **Location:** TI auditorium, ECSS 2.102
    - The review will also be on MS Teams and recorded
- There is no class Oct 2, 6 and 7 - use this time to study!

Syscalls - random number generator

Service	Code in $v0	Arguments	Result
set seed	40	$a0 = i.d. of pseudorandom number generator (any int). $a1 = seed for corresponding pseudorandom number generator.	No values are returned. Sets the seed of the corresponding underlying Java pseudorandom number generator (`java.util.Random`). *See note below table*
random int	41	$a0 = i.d. of pseudorandom number generator (any int).	$a0 contains the next pseudorandom, uniformly distributed int value from this random number generator's sequence. *See note below table*
random int range	42	$a0 = i.d. of pseudorandom number generator (any int). $a1 = upper bound of range of returned values.	$a0 contains pseudorandom, uniformly distributed int value in the range 0 = [int] [upper bound], drawn from this random number generator's sequence. *See note below table*

- Two system calls to create random numbers
  - Random int (41) - returns pseudorandom number in $a0
  - Random int in range (41) - returns pseudorandom number from a range in $a0

Agenda

Last time

- Procedures
- Modular code
- Types: Leaf, Non-leaf, Recursive
- Using `jal, jr`

This time

- CPU Performance: CPU Time
- Terminology: Clock Period, Frequency, IC, CPI, IPC
- Power constrains performance
- Performance Fallacies

What is “Performance”?

1 **a** : the execution of an action

b : something accomplished : DEED, FEAT

2 : the fulfillment of a claim, promise, or request : IMPLEMENTATION

~~3 **a** : the action of representing a character in a play~~

~~**b** : a public presentation or exhibition~~

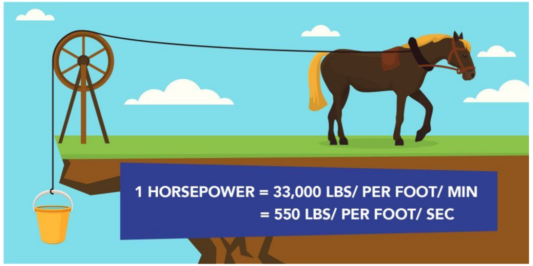
a benefit *performance*

4 **a** : the ability to perform : EFFICIENCY

b : the manner in which a mechanism performs

engine *performance*

What makes a high performance car?



Avg car horsepower = 100-200hp
HP sports cars = 300-500hp

0 to 60 mph time: Dodge Challenger SRT Demon 170 - 1.66sec

Top speed: Koenigsegg Jesko Absolut - 330mph

Quarter-mile time -
Pininfarina Battista - 8.55-seconds

Some metrics focus on time, some on rate or 1/time

History of the CPU Performance Wars (1970–now)

Factors to consider when comparing CPU's:

Performance single-thread and multi-thread benchmarking (Passmark CPU)

Price

Power Consumption

AMD (Team Red):

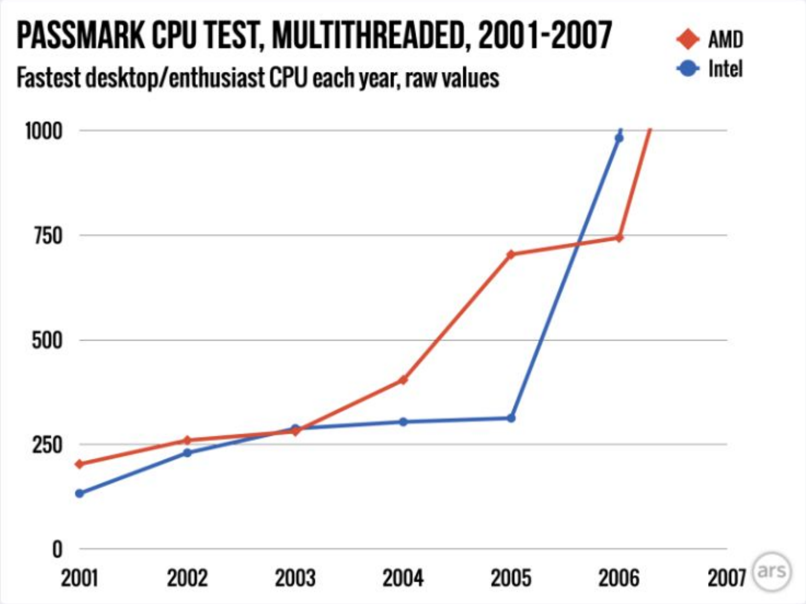
- Offer better price-to-performance ratio
- Pushing innovation with new architectures
- "underdog"

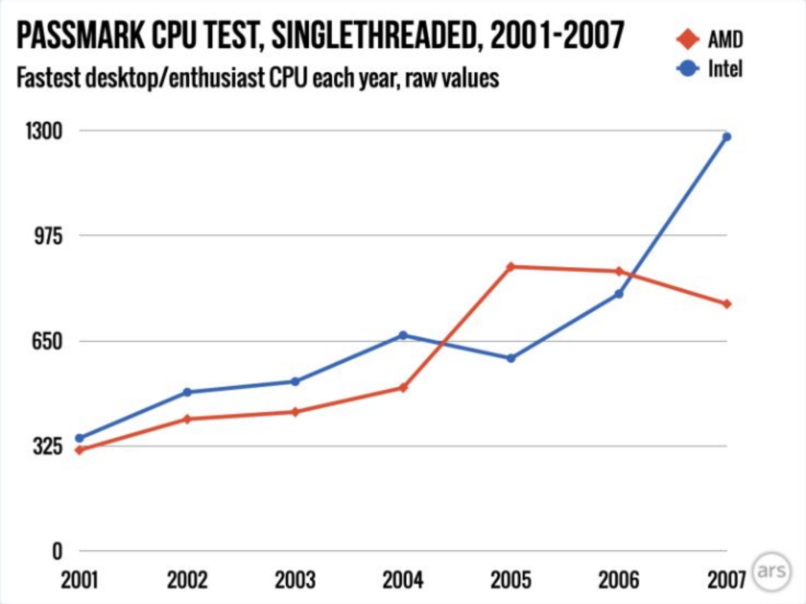
Intel (Team Blue):

- strong single-core performance
- established brand reputation
- higher price points

- 1970 ~ 2000 Intel establishes itself as the market leader, AMD as distant second

CPU performance wars (Intel vs AMD)

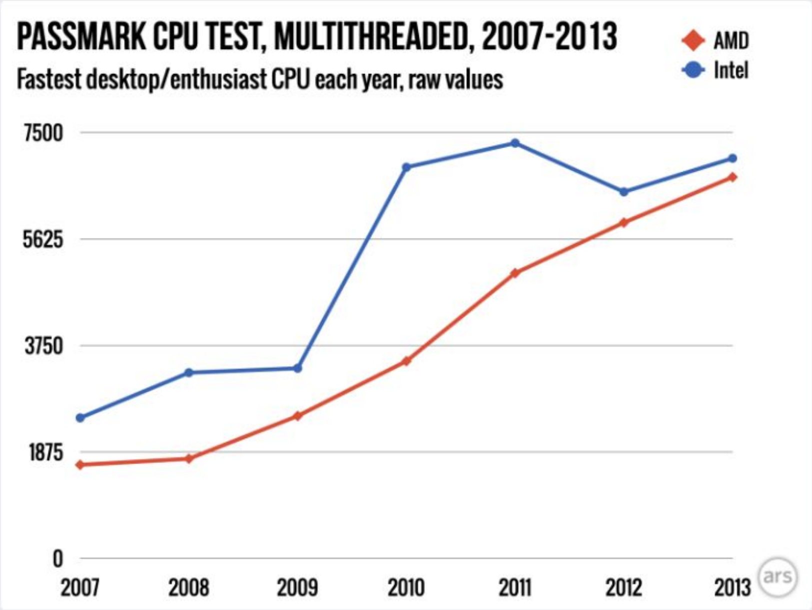


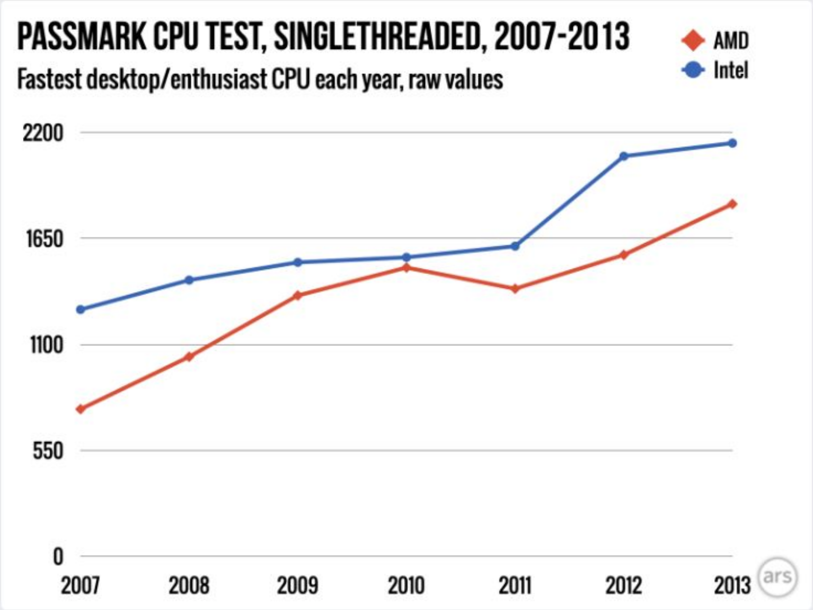


- 2000’s **AMD** first to release 64-bit processors a significant challenge to **Intel's** dominance

https://arstechnica.com

CPU performance wars (Intel vs AMD)

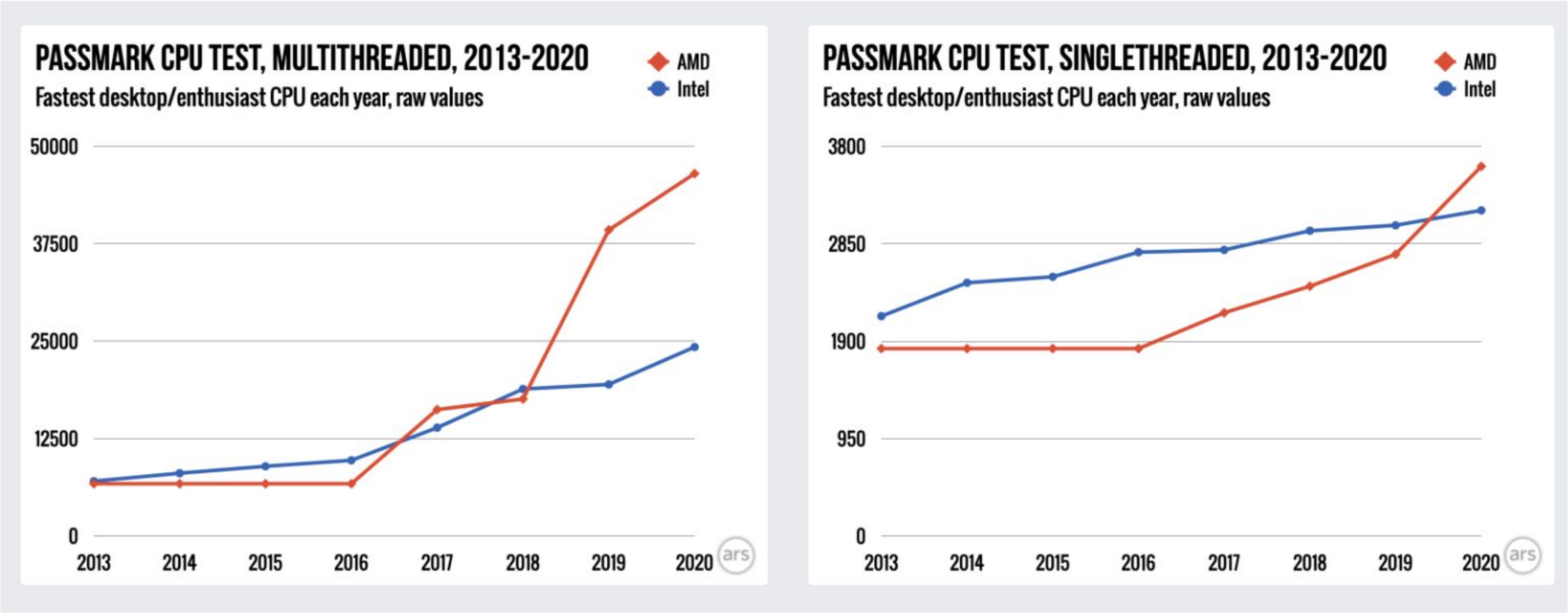




- **Intel** retook the lead starting in 2007 with its Core series

https://arstechnica.com

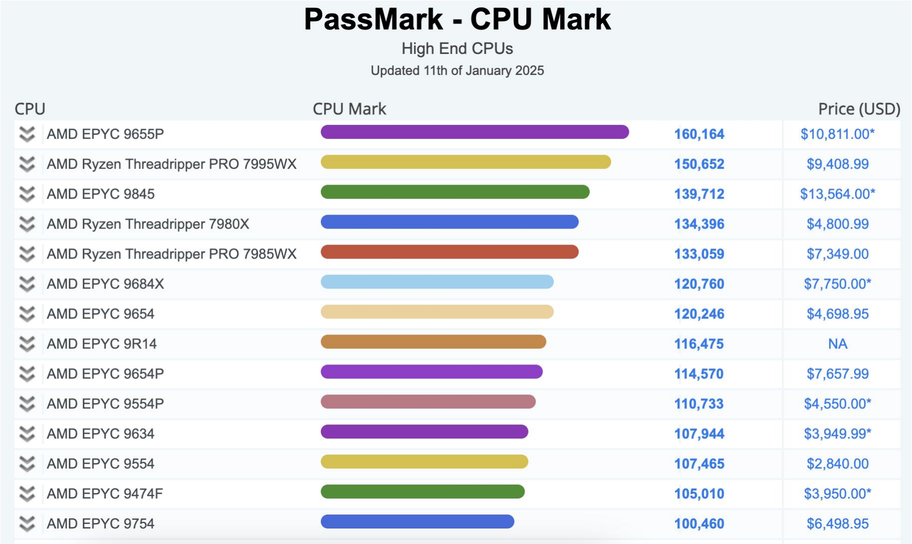
CPU performance wars (Intel vs AMD)



- In 2017, **AMD's** Zen architecture dominated in multi-threaded performance by power/thermal improvements

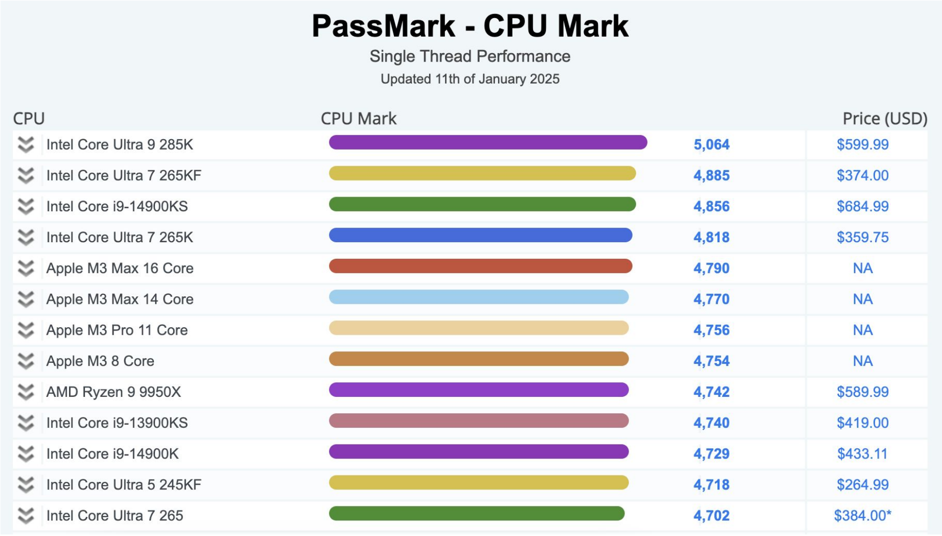
https://arstechnica.com

Current landscape



PassMark - CPU Mark
High End CPUs
Updated 11th of January 2025

CPU	CPU Mark	Price (USD)
AMD EPYC 9655P	160,164	$10,811.00*
AMD Ryzen Threadripper PRO 7995WX	150,652	$9,408.99
AMD EPYC 9845	139,712	$13,564.00*
AMD Ryzen Threadripper 7980X	134,396	$4,800.99
AMD Ryzen Threadripper PRO 7985WX	133,059	$7,349.00
AMD EPYC 9684X	120,760	$7,750.00*
AMD EPYC 9654	120,246	$4,698.95
AMD EPYC 9R14	116,475	NA
AMD EPYC 9654P	114,570	$7,657.99
AMD EPYC 9554P	110,733	$4,550.00*
AMD EPYC 9634	107,944	$3,949.99*
AMD EPYC 9554	107,465	$2,840.00
AMD EPYC 9474F	105,010	$3,950.00*
AMD EPYC 9754	100,460	$6,498.95



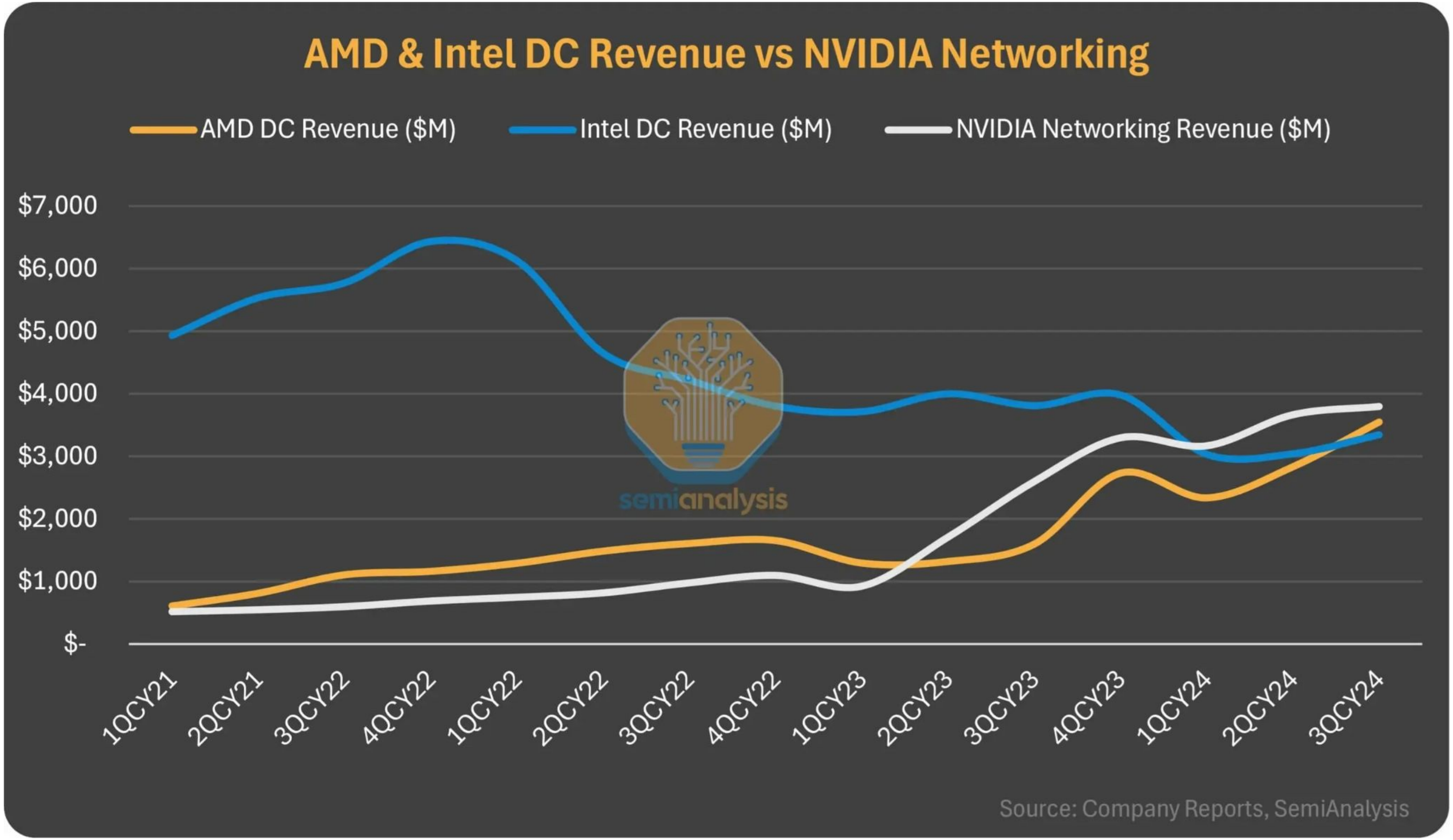
PassMark - CPU Mark
Single Thread Performance
Updated 11th of January 2025

CPU	CPU Mark	Price (USD)
Intel Core Ultra 9 285K	5,064	$599.99
Intel Core Ultra 7 265KF	4,885	$374.00
Intel Core i9-14900KS	4,856	$684.99
Intel Core Ultra 7 265K	4,818	$359.75
Apple M3 Max 16 Core	4,790	NA
Apple M3 Max 14 Core	4,770	NA
Apple M3 Pro 11 Core	4,756	NA
Apple M3 8 Core	4,754	NA
AMD Ryzen 9 9950X	4,742	$589.99
Intel Core i9-13900KS	4,740	$419.00
Intel Core i9-14900K	4,729	$433.11
Intel Core Ultra 5 245KF	4,718	$264.99
Intel Core Ultra 7 265	4,702	$384.00*

- Today: Both companies go back and forth taking the lead on performance
- Note: Apple M3 making an appearance on single-threaded performance!

https://www.cpubenchmark.net/high_end_cpus.html

The true test – market share

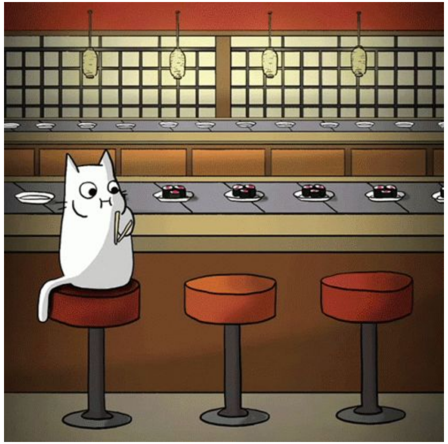


What do we consider good CPU performance?

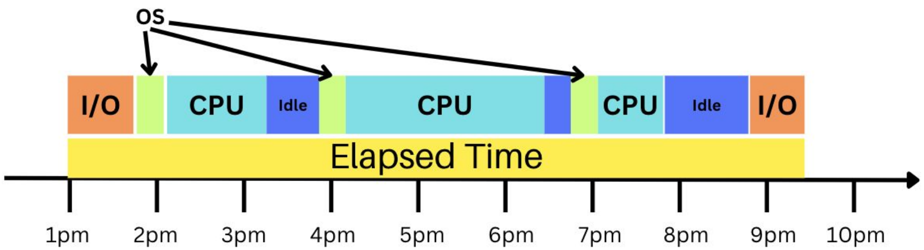
- CPU performance has similar metrics
- **Response time or latency**
  - How long it takes to do a task
- Throughput
  - Rate is 1/time
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour

What do we consider good CPU performance?

- CPU performance has similar metrics
- Response time or latency
  - How long it takes to do a task
- **Throughput**
  - Rate is 1/time
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour



How do we measure CPU Time

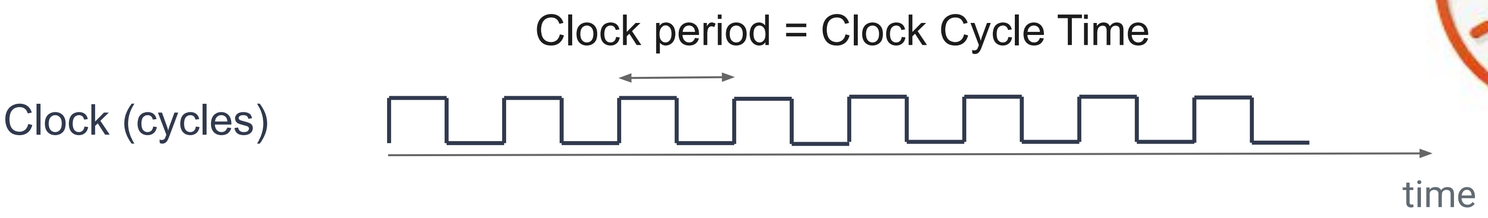


- Elapsed time
  - Total response time, including all aspects
    - Processing, I/O, OS overhead, idle time
- CPU time or Execution time
  - Total time spent processing a given job
  - Made up of **User** and **System** CPU time
    - User: time processor spends in running your application code.
    - System: time the processor spends in running the system functions (e.g. OS)

This class will focus on User CPU time (aka CPU time)

CPU Clocks

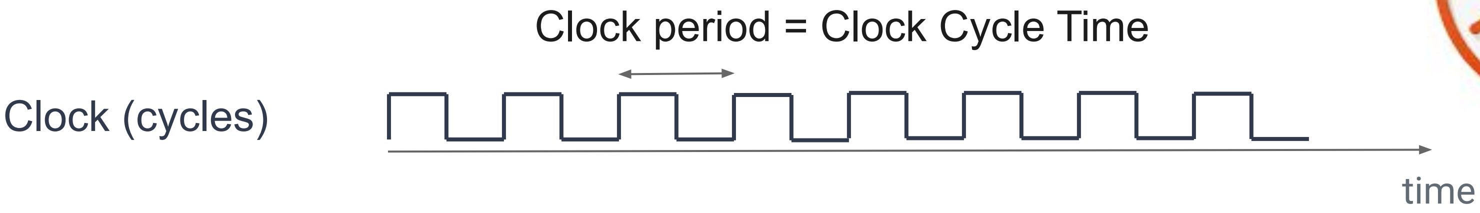
- Digital hardware operation are governed by a constant-rate clock



"Tick-Tock" - 50% of time clock is "1" and 50% of time clock is "0"

CPU Clocks

- Digital hardware operation are governed by a constant-rate clock



- ■ Clock period, Clock Cycle time: duration of 1 clock cycle (unit: sec)
- ■ Clock frequency, Clock rate: cycles per second (unit: Hz or 1/sec)

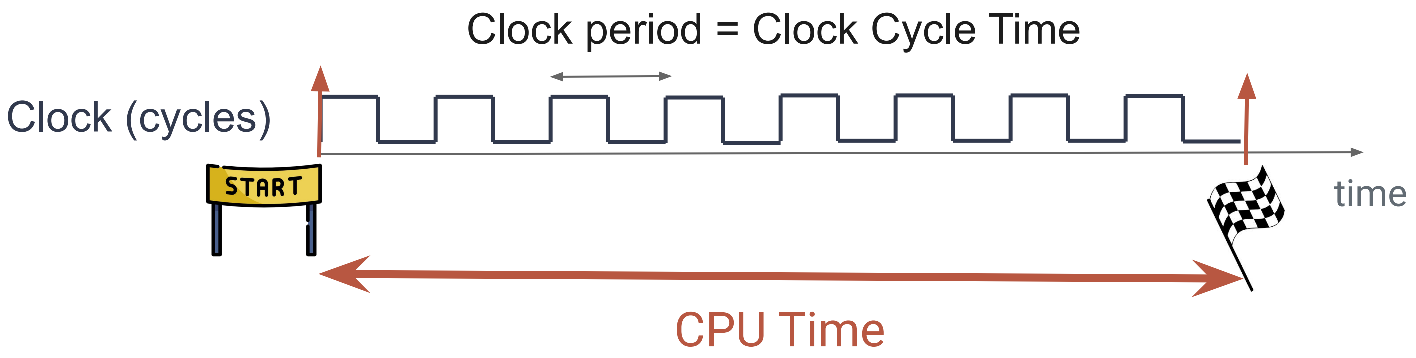
Clock frequency = 1 / Clock period

Powers of 10

Frequencies (Hz)	Time (sec)
• 10^3 = kilo (kHz)	• 10^{-3} = milli (ms)
• 10^6 = Mega (MHz)	• 10^{-6} = micro (μs) or (us)
• 10^9 = Giga (GHz)	• 10^{-9} = nano (ns)
• 10^{12} = Tera (THz)	• 10^{-12} = pico (ps)

- Clock period, Clock Cycle time: (unit: sec)
  - e.g., 250ps = 0.25ns = 250×10^{-12}s
- Clock frequency, Clock rate: (unit: Hz or 1/sec)
  - e.g., 4.0GHz = 4000MHz = 4.0×10^9 Hz

CPU Time



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

# of Clock Cycles to finish the program = __________ **cycles**

If Clock period is 2ns, CPU Time = __________ **ns**

Clock Rate = __________ **Hz or** __________ **MHz**

Improving Performance

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

- Performance is improved by
  - Reducing number of Clock Cycles ⇩
  - Increasing Clock Rate (Clock Frequency) ⇧

Relative Performance

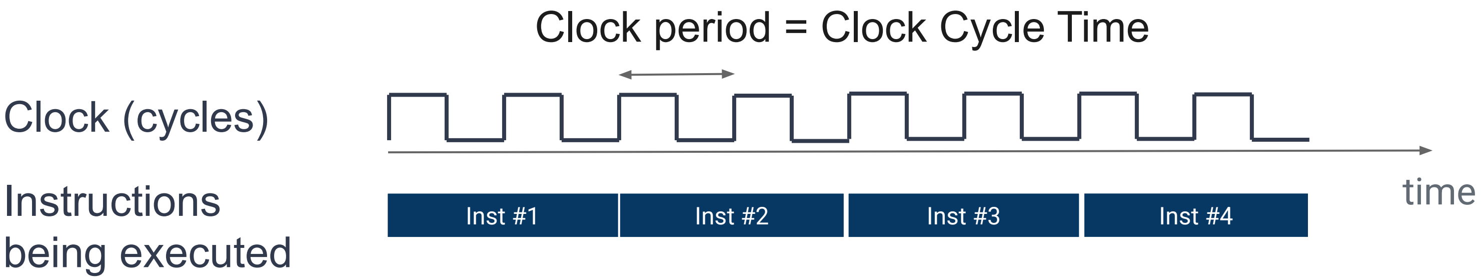
- “X is n time faster than Y”

$$\text{Performance}_X / \text{Performance}_Y$$
$$= \text{CPU time}_Y / \text{CPU time}_X = n$$

- Example for the same program:
  - CPU time is 10s on CPU A, 15s on CPU B
  - Which CPU is faster? How much faster?
  - $\text{CPU Time}_B / \text{CPU Time}_A =$

Instruction Count and Cycles Per Instruction

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

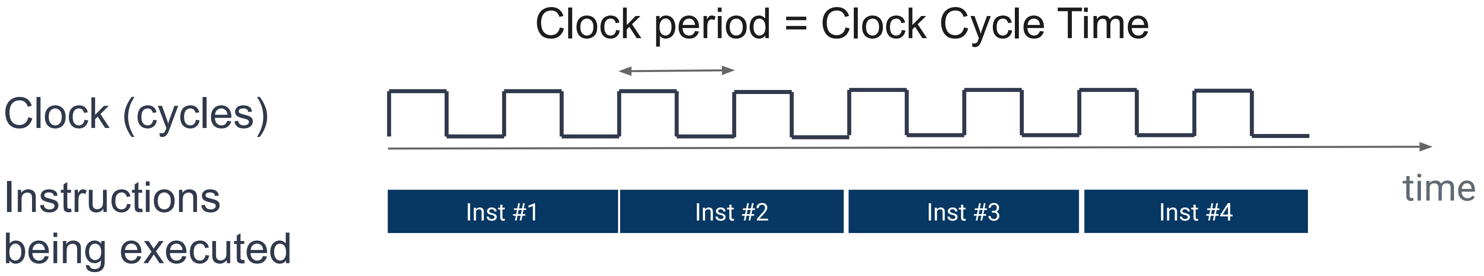


- Instruction Count (IC) for a program
  - Determined by program, ISA and compiler
- Cycles per Instruction (CPI)
  - Determined by CPU hardware

- CPU time = Clock cycles x Clock Cycle Time
  = IC x CPI x Clock Cycle Time

Instruction Count and Cycles Per Instruction

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



<u>For the above example</u>

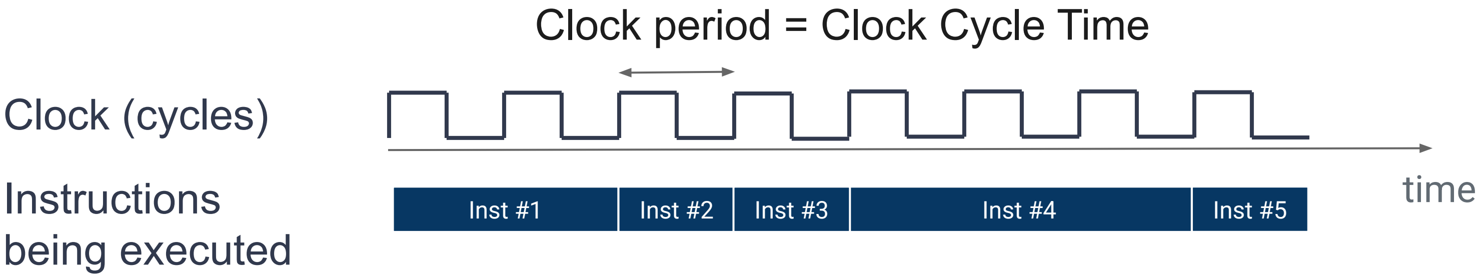
Instruction count (IC) = ______

Cycles per Instruction = ______

Clock cycles = ______

If clock period is 3ns, CPU Time = __________ **ns**

CPI in more detail



- Different instruction classes can take different numbers of cycles

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

$$= 3 \text{ x } 1 + 2 \text{ x } 1 + 1 \text{ x } 3 = 8 \text{ cycles}$$

CPI Example – My Turn

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

Table shows code sequences using instructions in classes A, B, C

Instruction Class	A	B	C
CPI for class	1	2	3
Instruction Count in Sequence 1	2	1	2
Instruction Count in Sequence 2	4	1	1

Sequence 1:

- Total IC = ________
- Clock Cycles = ______
- Avg. CPI = Clock Cycles / IC = ________

CPI Example – Your Turn

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

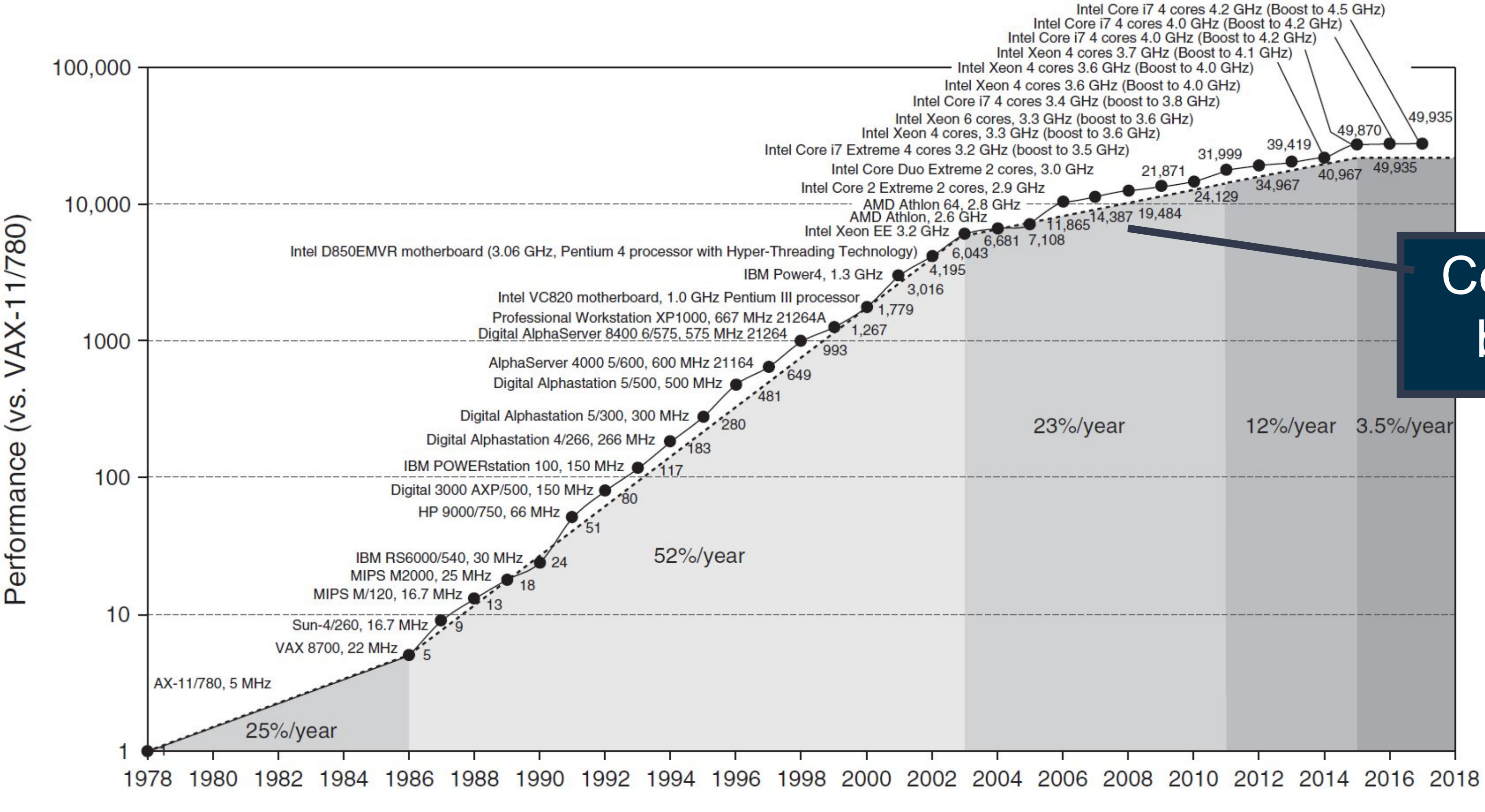
Table shows code sequences using instructions in classes A, B, C

Instruction Class	A	B	C
CPI for class	1	2	3
Instruction Count in Sequence 1	2	1	2
Instruction Count in Sequence 2	4	1	1

Sequence 2:

- Total IC = ________
- Clock Cycles = ______
- Avg. CPI = Clock Cycles / IC = ________

History – Uniprocessor performance



Constrained by power

What is CPU Power Consumption

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

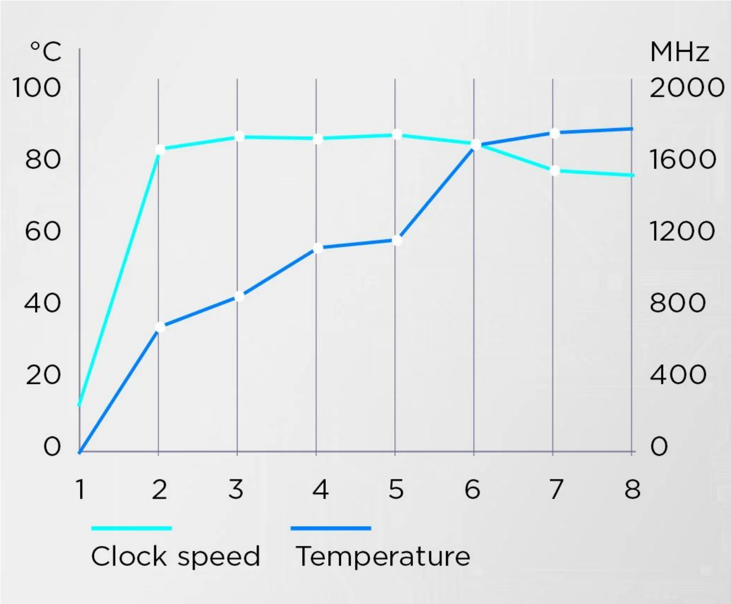
- CPUs require electrical power to operate
- Higher performance CPUs (faster clocks) consume more power
- CPUs with more features (larger capacitance) consume more power

Why does power constrain performance?

- **If a CPU doesn't constrain power it will hit Thermal Limits**
  - More power ⇒ More heat
- **Thermal Design Power (TDP)**
  - Maximum power under typical workload conditions
  - Measured in watts (W)
- If the CPU exceeds the TDP for a long time then it could melt the chip

cgdirector.com

How does power constrain performance?

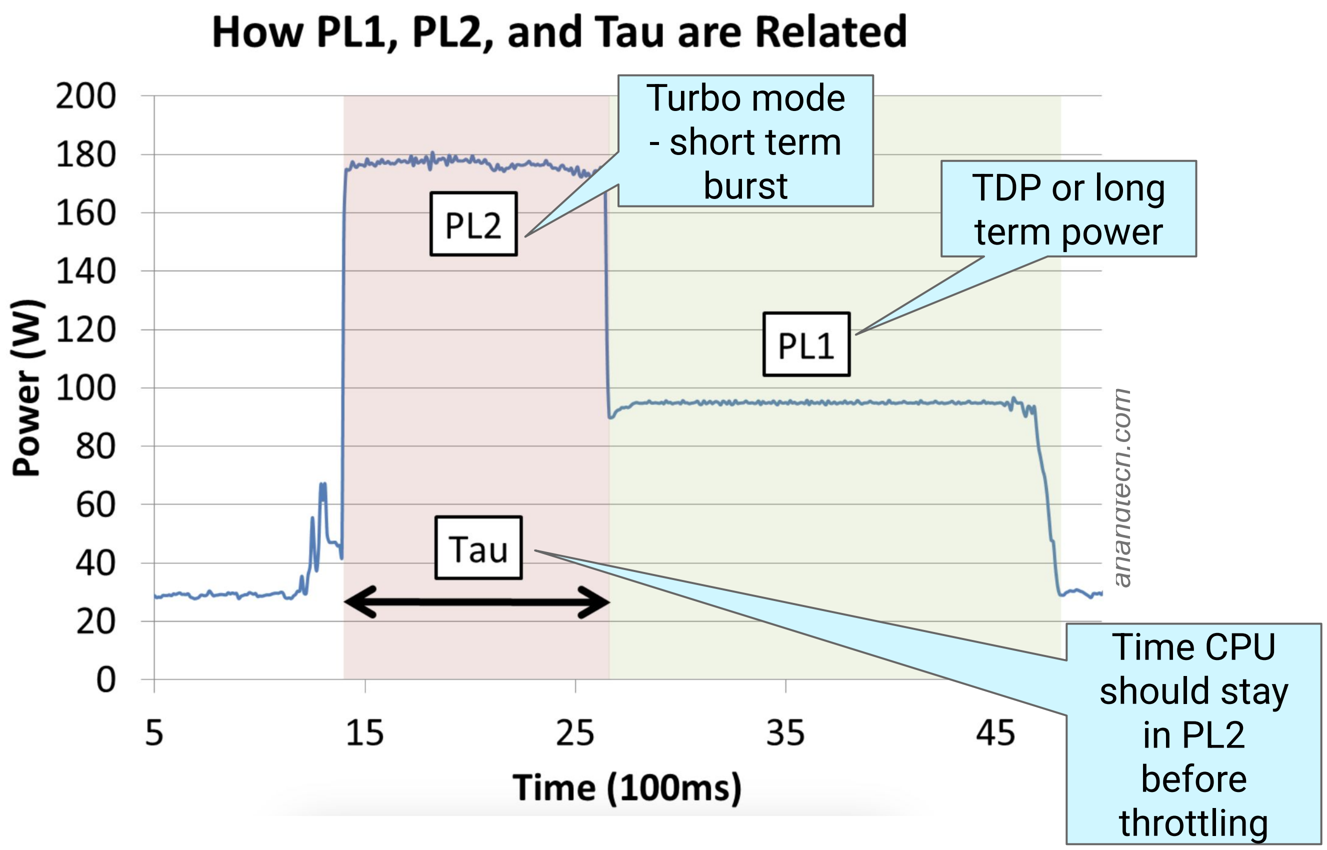


cgdirector.com

- Many CPU's have fans (PC) or liquid cooling (datacenters) to stay below TDP
- Also adds cost ($$)

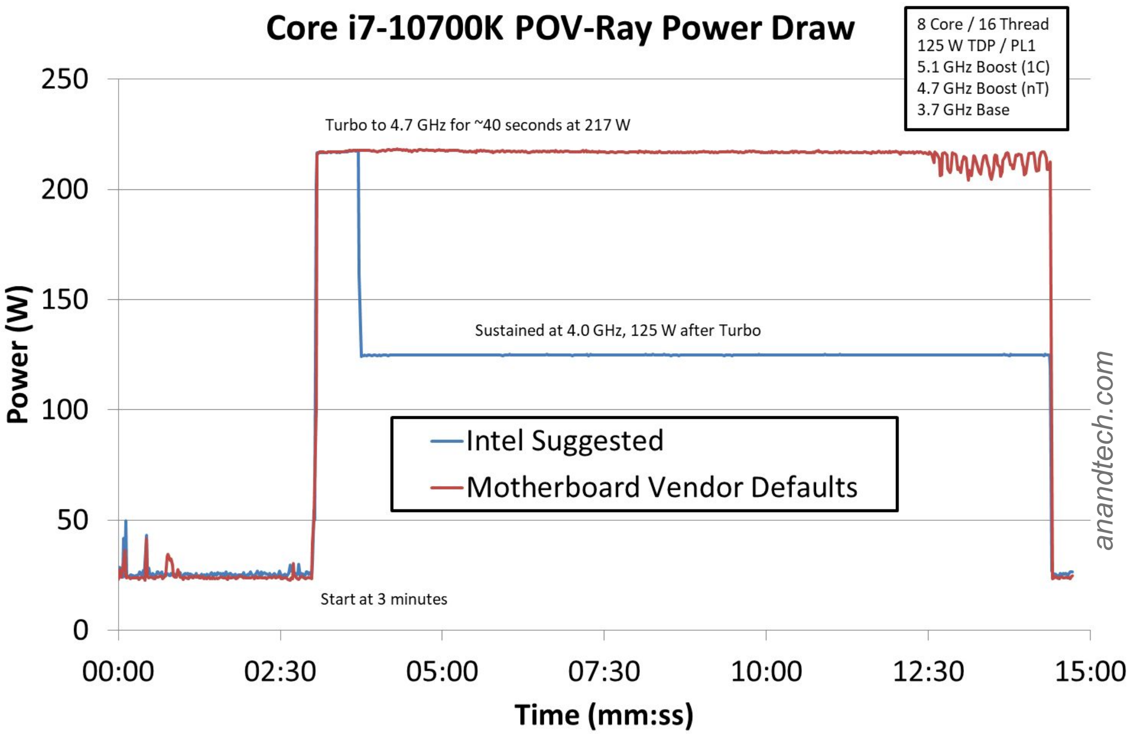
- Performance Throttling
  - Reduce the clock frequency when TDP reached

Performance throttling – Power profile



Real-world power profile

POV-Ray is a CPU benchmark that does 3D rendering



- Some companies don’t strictly follow the manufacturer’s suggestion

Intel 13th and 14th gen troubles

TECH Updated Oct 4, 2024, 6:08 PM CDT

Intel's crashing 13th and 14th Gen Raptor Lake CPUs: all the news and updates

By Sean Hollister, a senior editor and founding member of The Verge who covers gadgets, games, and toys. He spent 15 years editing the likes of CNET, Gizmodo, and Engadget.

0 Comments (0 New)

Many months ago, gamers began experiencing strange crashes with their 13th and 14th Gen Intel Core i9 CPUs — but that was just the tip of the iceberg.

Intel has now extended its warranty by two full years on 24 different 13th Gen and 14th Gen desktop chips, including Core i5, Core i7, and Core i9 models, after discovering that many CPUs based on its Raptor Lake architecture are susceptible to permanent damage. They were being fed too much voltage, and some have irreversibly degraded. Intel has not yet said if laptop chips are failing the same way.

- "Too much voltage" caused by a microcode algorithm error
- Exceeding TDP and causing CPU's to crash

CPU Performance Fallacy #1

	Processor #1	Processor #2
Clock Rate	4GHz	3GHz
CPI	0.9	0.75
Instruction Count	5.0E9	1.0E9

A common fallacy is to consider the CPU with the highest clock rate as having the largest performance. Is it true in this case?

<u>Clock rate #1 vs #2</u>

<u>CPU time #1 vs #2</u>

CPU Performance Fallacy #2

	Processor #1	Processor #2
Clock Rate	4GHz	3GHz
CPI	0.9	0.75
Instruction Count	5.0E9	1.0E9

Another common fallacy is to consider the CPU that executes a larger number of instructions is more powerful. Is this true?

Instruction Count #1 vs #2

CPU time #1 vs #2

CPU Performance Fallacy #3

	Processor #1	Processor #2
Clock Rate	4GHz	3GHz
CPI	0.9	0.75
Instruction Count	5.0E9	1.0E9

A third common fallacy is to use the MIPS (millions of instructions per second) metric to compare the performance. Is this true?
[Hint: MIPS = Instruction count (in millions) / CPU time]

<u>MIPS #1 vs #2</u>

<u>CPU time #1 vs #2</u>

More CPU Performance Examples - My turn

Types of Instructions	CPI	Instruction Count (Millions)
Floating Point	1	50
Integer	1	110
Load/Store	4	80
Branch	2	16

By how much is the CPU time of the program improved if the CPI of Integer and Floating point instructions is reduced by 40% and the CPI of load/store and branch is reduced by 30%?

CPU time (original) = $\sum IC \times CPI \times T_c$ =

CPU time (new) = $\sum IC \times CPI \times T_c$

=

More CPU Performance Examples – Your turn

Types of Instructions	CPI	Instruction Count (Millions)
Floating Point	10	20
Integer	1	200
Load/Store	5	150
Branch	2	15

By how much do we need to reduce the CPI of Load/Store instructions if we want the program to run 25% faster?

CPU time (original) = $\sum IC \times CPI \times T_c$ =

CPU time (new) =

CPI of Load/Store needs to be reduced by ____________

Methods for Tuning Performance

The BIG Picture

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

$$= \quad IC \quad \times \quad CPI \quad \times \quad T_c$$

HW or SW component	This component affects performance by determining		Where is this topic covered?
Algorithms	Number of source-level statements and the number of I/O operations executed	IC, CPI?	Advanced courses
Programming language, compiler and architecture	Number of computer instructions for each source-level statement	IC, CPI	Chapters 2 & 3 Lectures 5 ~ 13
Processor and Memory system	How fast instructions can be executed	IC, CPI	Chapters 4, 5 Lectures 14 ~ 24
I/O system (hardware and operating system)	How fast I/O operations may be executed	IC, CPI, T_c	Chapters 4, 5 Lectures 14 ~ 24

Summary

- CPU time: the best performance measure
  - Not Clock Frequency, Largest # of instructions, MIPS (millions of instructions per second)
- Clock Frequency is inverse of Clock period
  - Frequency and Rate are equivalent
  - Period and Clock cycle time are equivalent
- Clock period, Instruction Counts, Cycles per Instruction all factor into CPU Time
- Power constrains the Maximum Performance

Next lecture

Binary Arithmetic