COSC 511: Computer Architecture
The Processor

Week 4

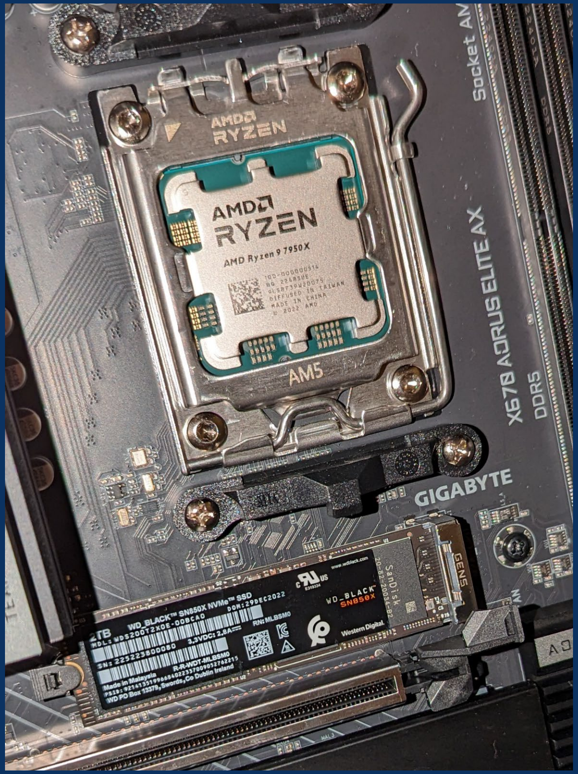
← ⁘ → ×

Last Week

- Arithmetic for Computers
  - Tricks computers use for doing math
    - Twos compliment to use addition to do subtraction
    - Use of bit shifting to do multiplication
    - Slow and fast division
  - Overflow
    - What it is, why it happens, how to manage it.
    - 2038 Problem
  - Using binary to represent fractional values in memory
  - Binary data in memory has no inherent meaning
    - Malicious tampering of memory contents can allow for code injection.

The Processor

- Processor/CPU – Central Processing Unit
  - “Brain” of the computer
  - The CPU is responsible for performing calculations and logical operations
    - CPU uses registers for storing the data that it is manipulating
  - CPU speed is measured in Hertz (Hz)
    - Modern CPUs are so fast that we use gigahertz (GHz) to refer to their speed
    - AMD Ryzen 7950X (released Sep. 27, 2022)
      - Base Clock: 4.5GHz, Boost Clock: 5.7GHz



The Processor

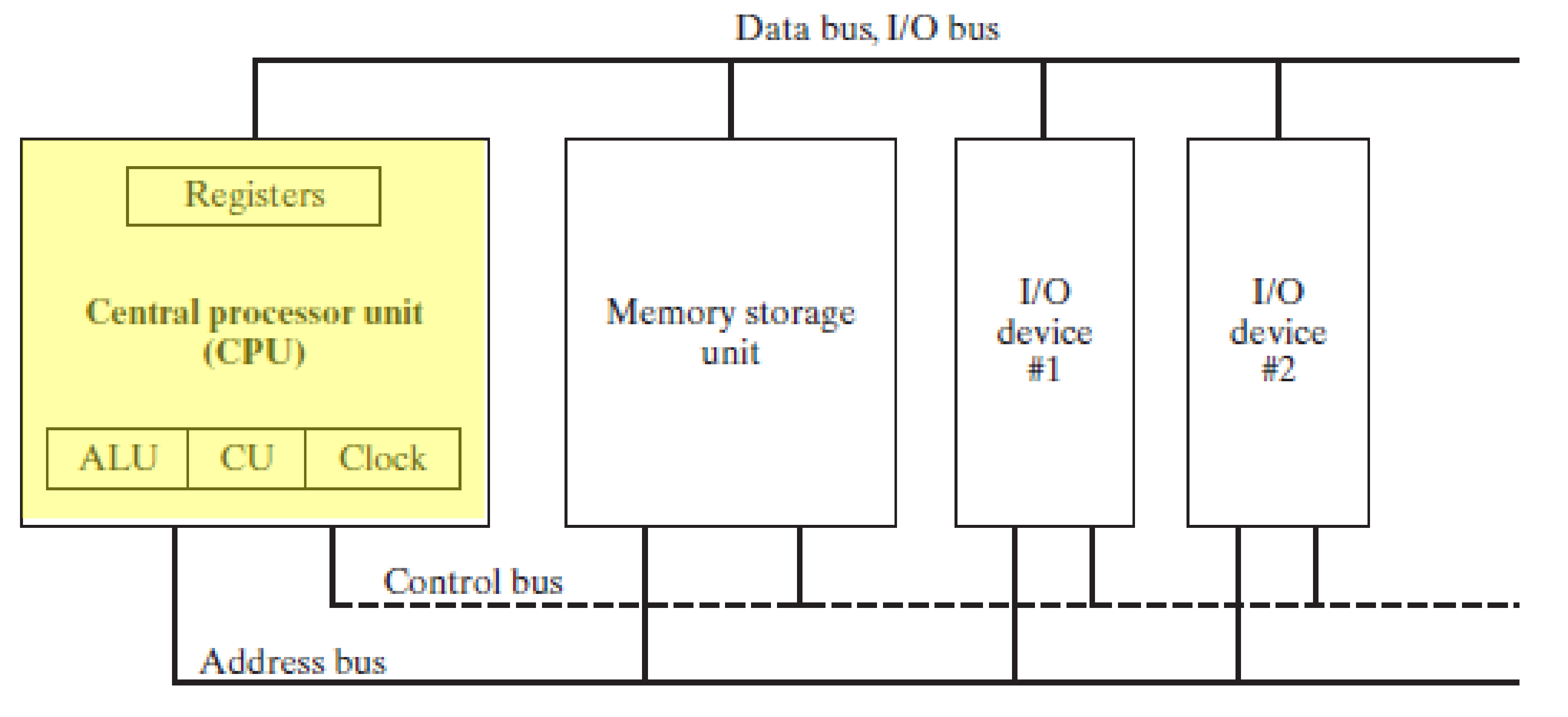
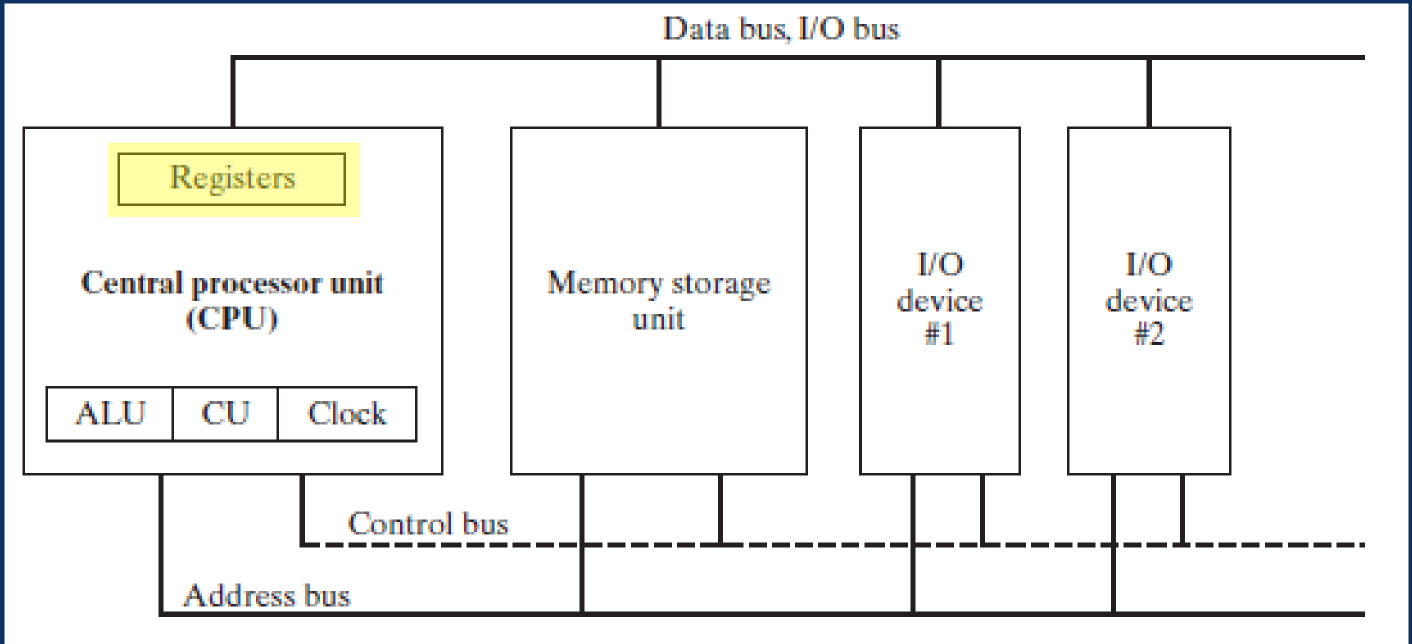


Diagram showing components of a computer, with CPU highlighted.

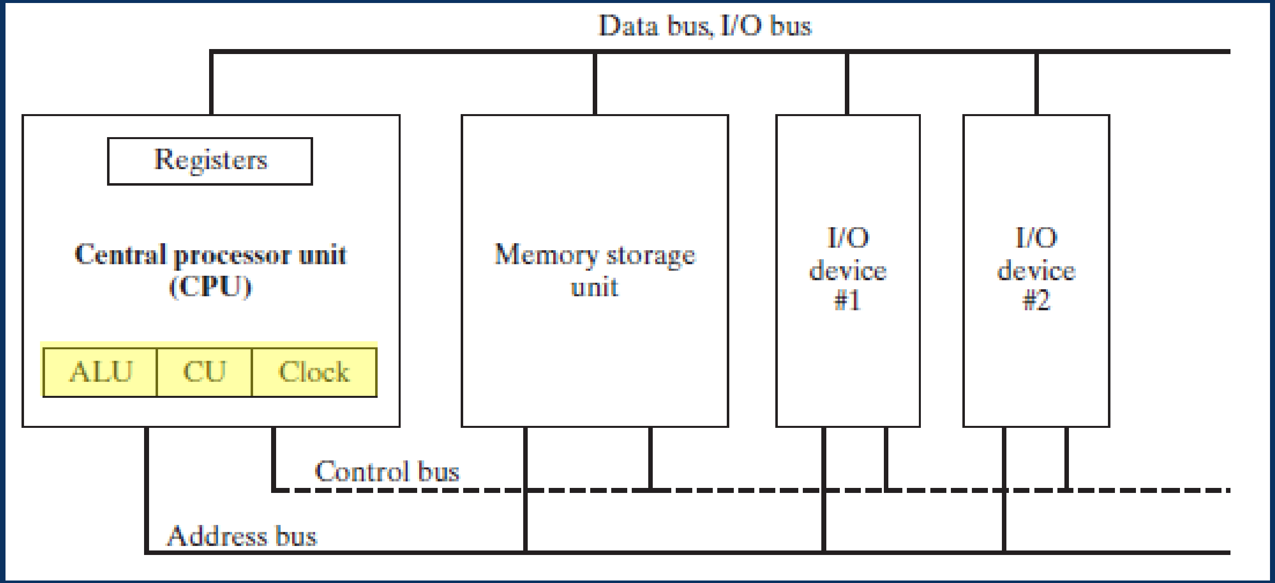
The Processor

- Registers
  - Used for storing frequently accessed data
  - Used for performing mathematical and logical operations
  - Very fast! Faster than retrieving data from memory.
  - Extremely limited storage space
  - Expensive
  - Very few



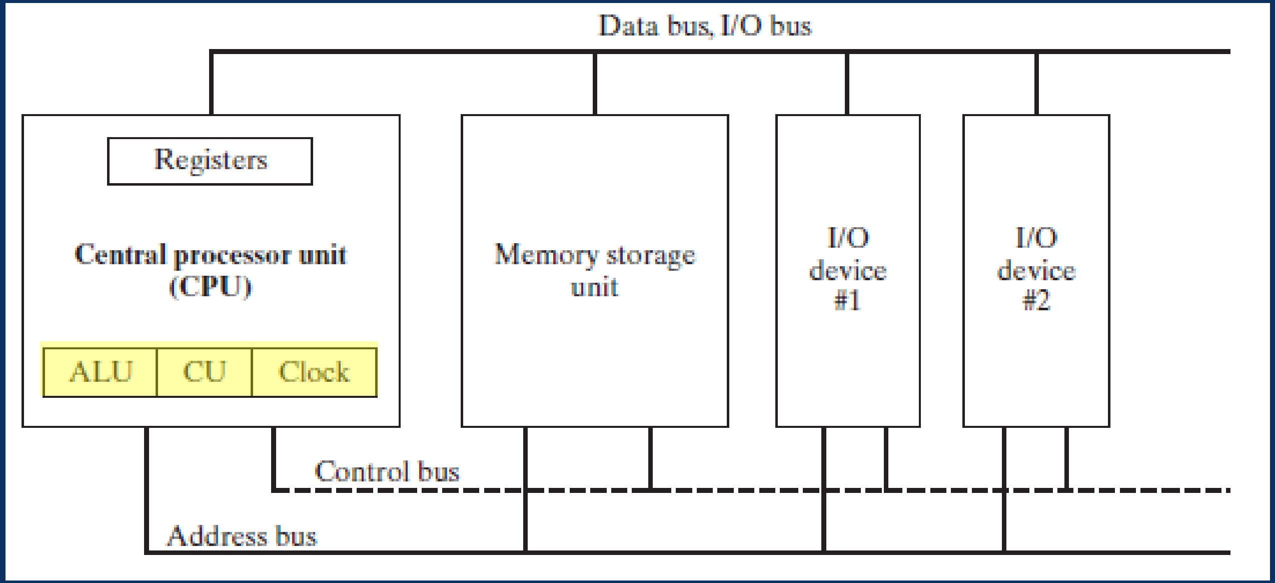
The Processor

- Clock
  - Synchronizes CPU operations with other components
  - Does not refer to “wall clock” time!
  - Each instruction takes at least one clock cycle
  - Measured in Hertz
    - 1 GHz means 1 billion clock cycles per second



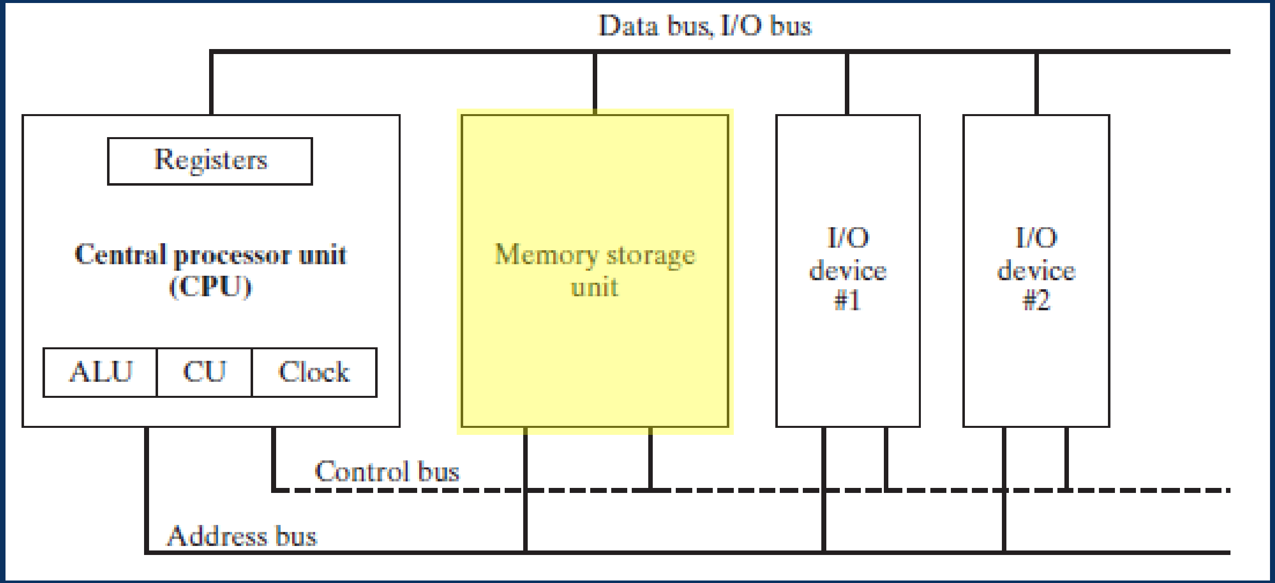
The Processor

- ALU – Arithmetic Logic Unit
  - Handles arithmetic
  - Handles logical calculations
- CU – Control Unit
  - Coordinates sequencing of steps to execute machine instructions



The Processor

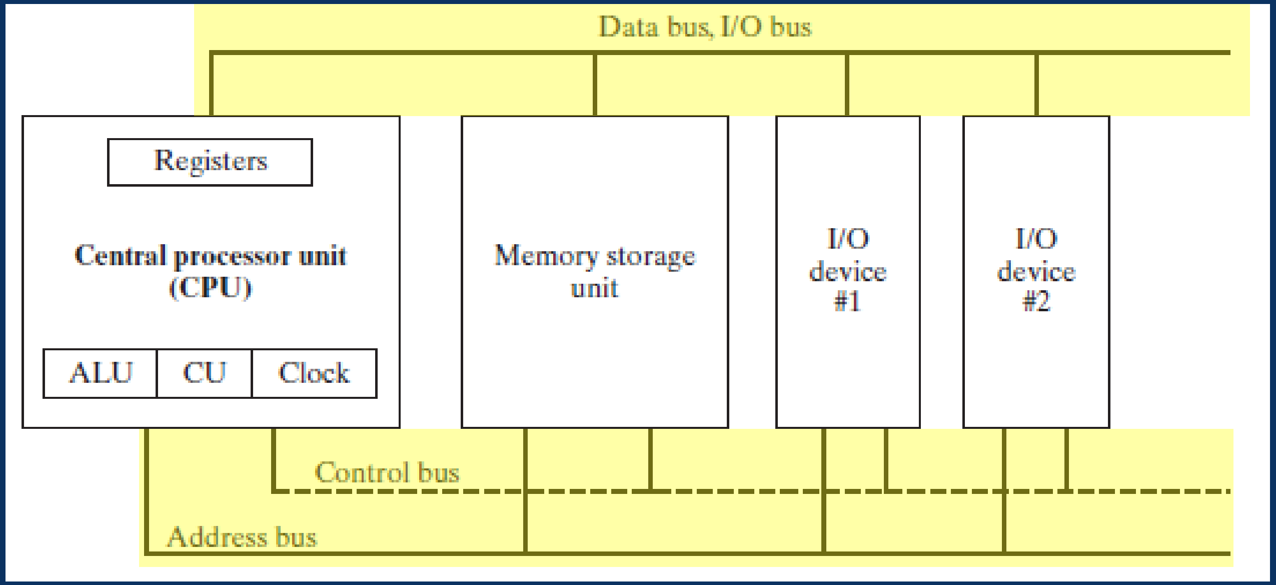
- Memory Storage Unit
  - Holds instructions and data that is relevant to running programs.
  - Handles transfer of data to and from RAM (Random Access Memory)



The Processor

- Bus
  - Generic term for components that transport data from one part of computer to another.

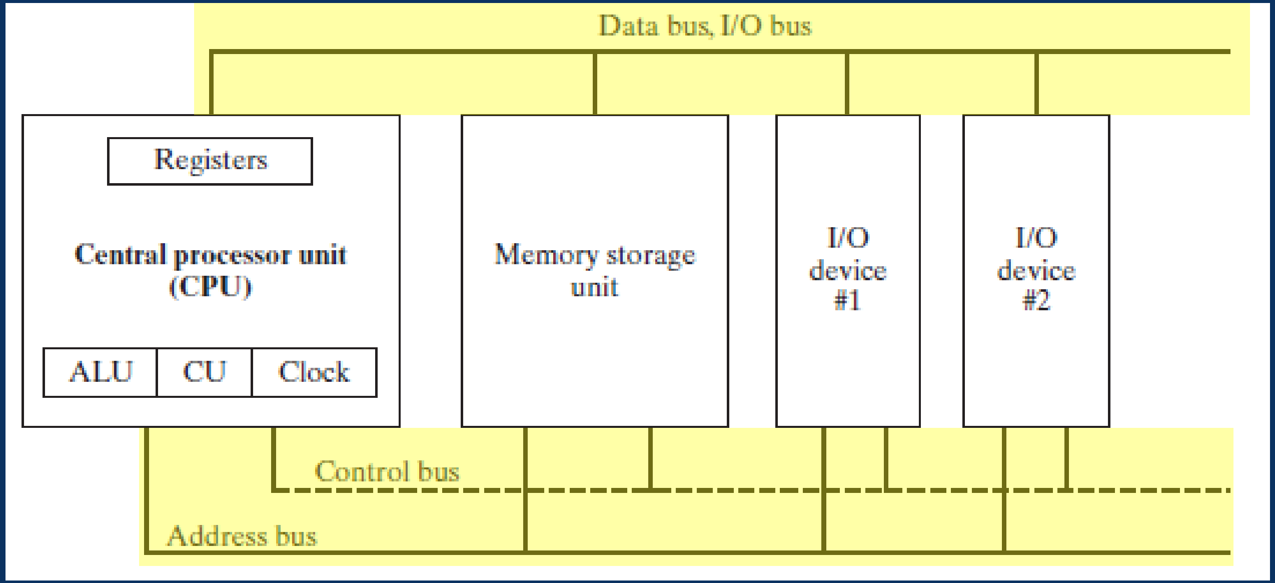
- Data Bus
  - Transfers instructions and data between CPU and memory



The Processor

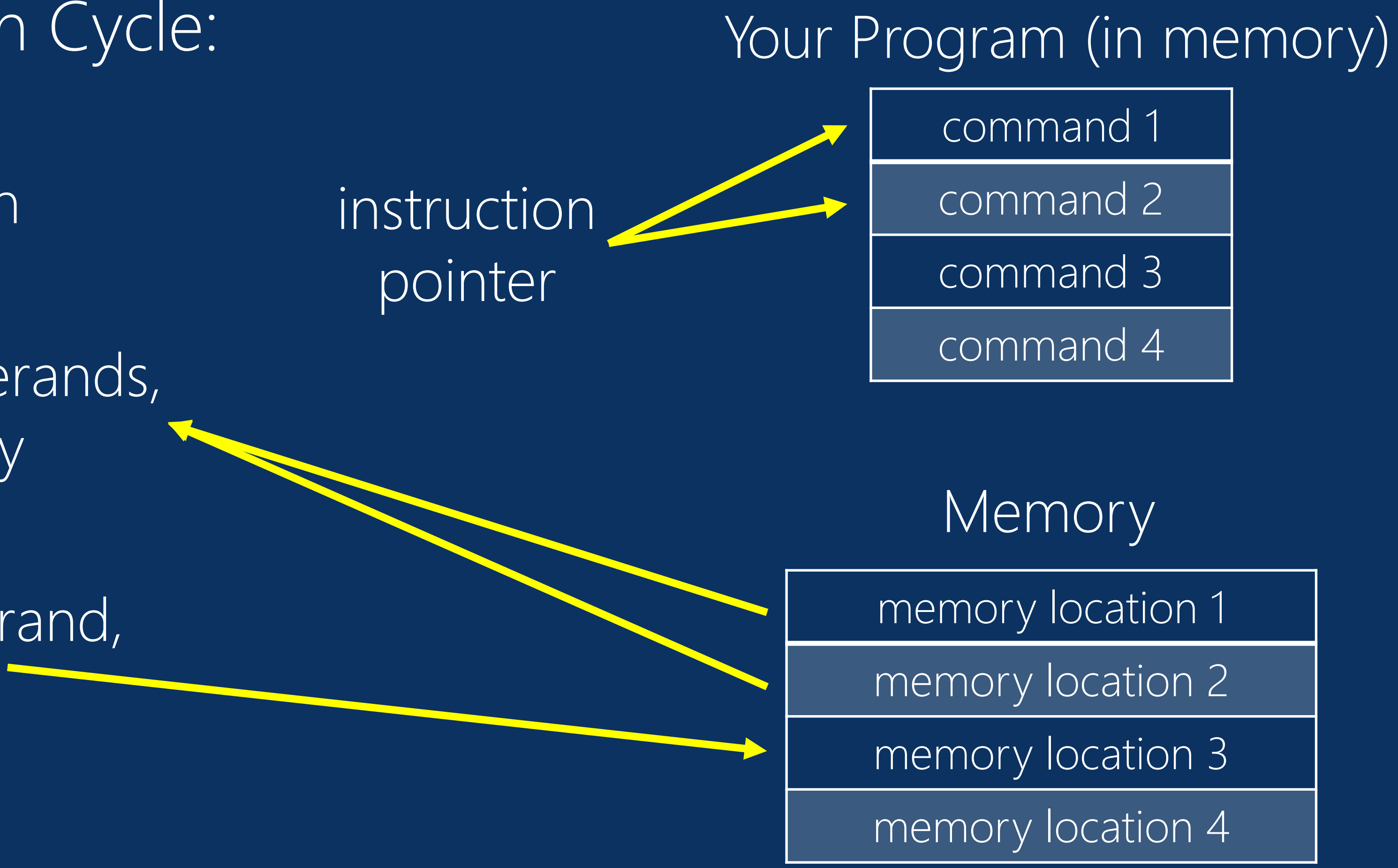
- Control Bus
  - Synchronizes functionality of CPU and other hardware

- Address Bus
  - Holds the addresses of instructions and data during transfer



The Processor

- CPU Instruction Execution Cycle: Fetch, Decode, Execute
  1. Fetch the next instruction
  2. Decode the instruction
  3. If the instruction has operands, fetch them from memory
  4. Execute the instruction
  5. If there is an output operand, store the value there



Your Program (in memory)

instruction pointer

command 1

command 2

command 3

command 4

Memory

memory location 1

memory location 2

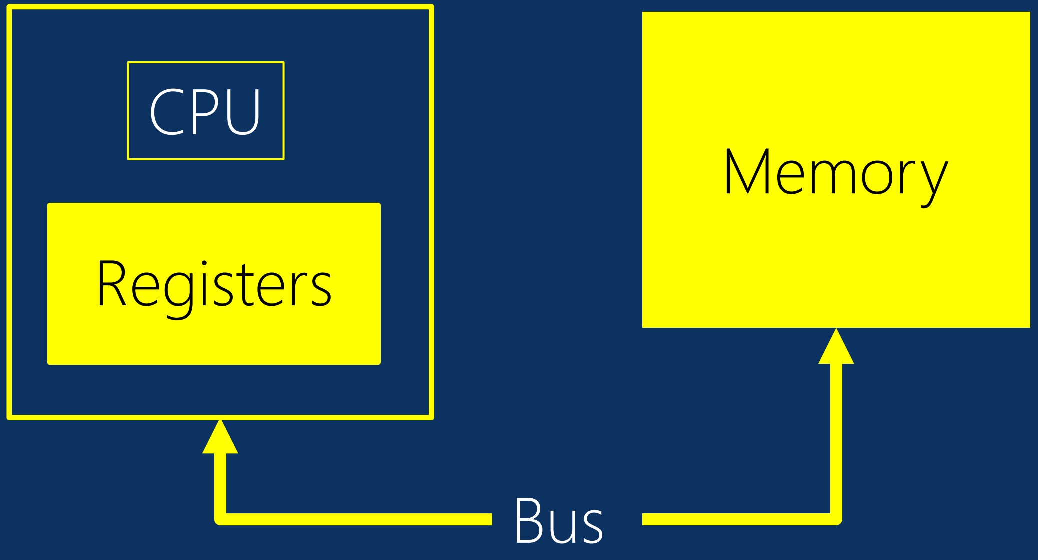
memory location 3

memory location 4

The Processor

- Computers read from memory a lot slower than they access internal registers

  1. Place the address of the value you want on the address bus
  2. Change the value of the CPU ‘read’ pin
     - This indicates that the CPU is ready to accept data.
  3. Wait one clock cycle for memory hardware to respond
  4. Copy retrieved data into destination operand



The Processor

- Using registers is helpful, but we have a problem.
  - CPUs will frequently access lots of different data stored in memory.
  - Registers have very little storage space.
  - CPUs don’t have a lot of them.
  - Having lots of data to access, but few registers, means we must reach out to memory very often.
  - This works, but memory is slow.
  - Our CPU has a speed bottleneck caused by frequent RAM access.

- How do we fix this?
  - Cache!

The Processor

- Memory Cache
  - Created to avoid this speed bottleneck.
  - A CPU is likely to access the same memory and instructions repeatedly.

- When the CPU needs data or an instruction, it checks its cache first.
  - Cache Hit: Data is found in the cache.
  - Cache Miss: Data is not found in the cache, so CPU must get it from memory.
    - When a cache miss happens, the data is copied from memory into the cache.
    - Unless the cached data is removed to make room for something else to cache, future requests of the data will be retrieved from the cache.

The Processor

- Benefits of Memory Cache
  - Reduces impact of performance bottleneck caused by direct RAM access
  - More storage space than registers but still less than RAM
  - Cheaper than registers, but still more expensive than RAM
  - Slower than registers, but still faster than RAM

- Problems with Memory Cache
  - Added complexity to managing data propagation
    - When data is to be removed from cache, it must first be copied to RAM.

← ⁘ → ✕

The Processor

- Most CPUs have at least two cache levels
  - L1/Primary cache
    - L1 cache is the fastest and stored directly on the CPU
  - L2/Secondary cache
    - L2 cache is built into the CPU package, but connected to the actual CPU by a high-speed data bus
    - Since the connection is not direct, L2 cache is a little slower
- Modern CPUs have an L3 cache as well
  - L3 cache is slower than L2, but still faster than RAM
  - First introduced in the 80s by IBM
  - First consumer-grade CPU to have L3 cache
    - Pentium 4 Extreme Edition, launched in 2003, 2MB

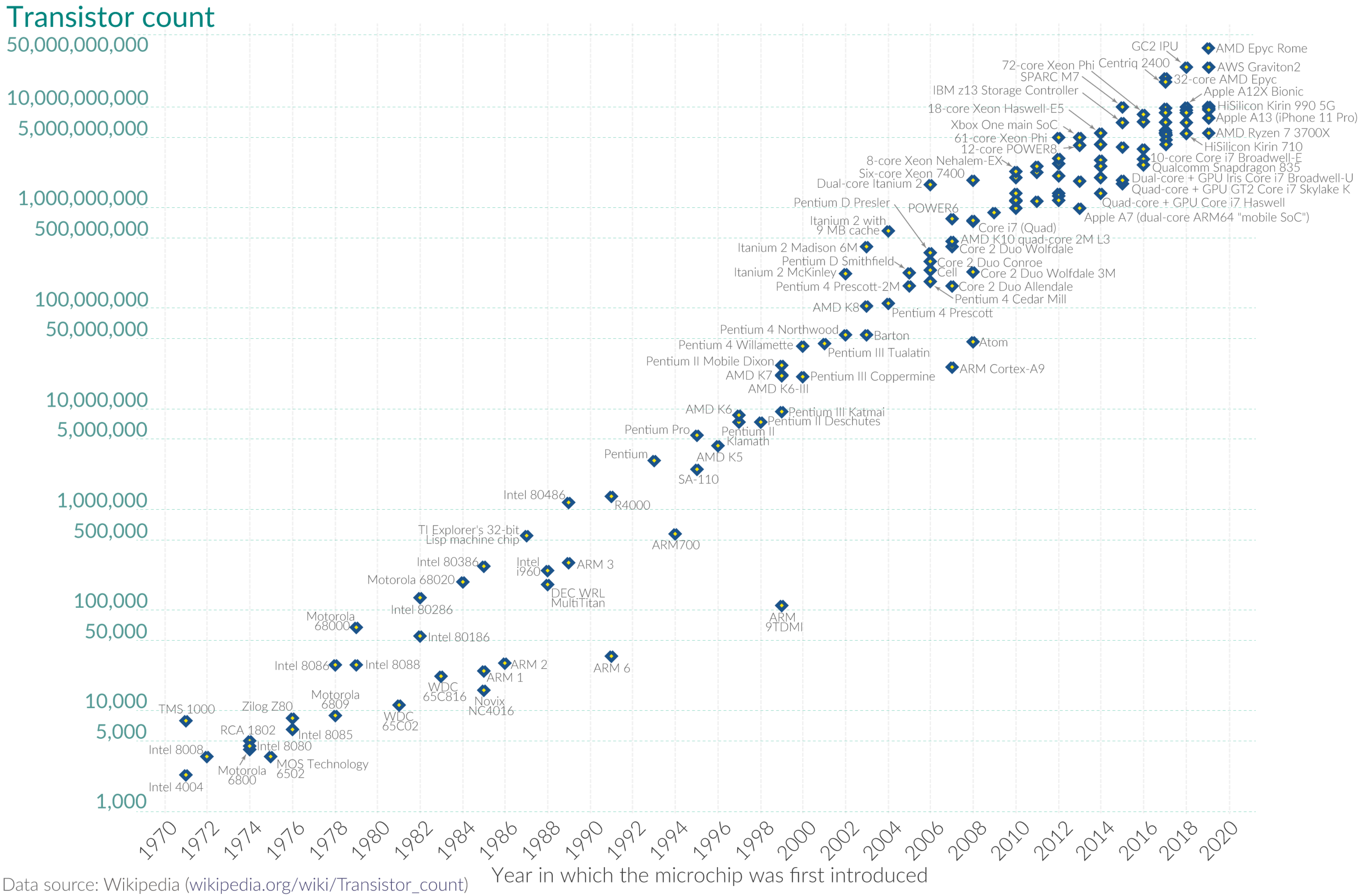
The Processor

- AMD Ryzen 7950X Cache
  - L1: 64K per core
  - L2: 1MB per core
  - L3: 64MB total

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

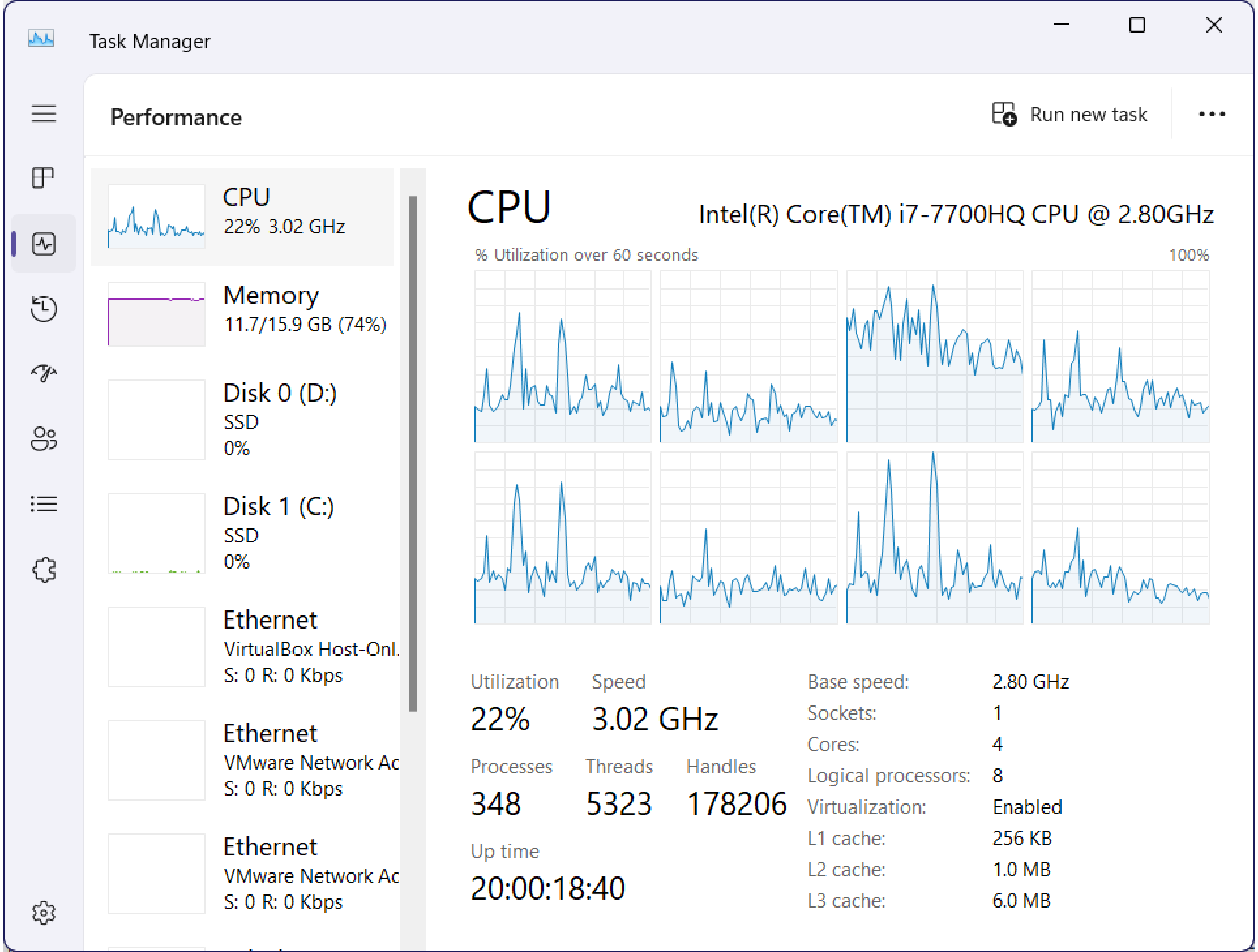
The Processor

- The pace at which CPU clock speed increases has slowed down!
  - Q: So how are we still getting faster CPUs?
  - A: We aren’t. We’re building CPUs that are essentially multiple CPUs in one.
    - We call these “cores.”
- AMD Ryzen 7950X has 16 cores.
  - That means a computer with this CPU actually has 16 physical CPUs.

The Processor

- To make things even faster, CPUs implement SMT.
  - SMT – Simultaneous Multithreading
  - Instead of performing the CPU instruction cycle one step at a time, multiple stages of the process are happening with different instructions at the same time.
  - Intel refers to SMT as Hyperthreading.

- AMD Ryzen 7950X has 16 cores and 32 threads.
  - This means that each core can handle the workload of 2 cores that do not use SMT.



CPU
22% 3.02 GHz

Memory
11.7/15.9 GB (74%)

Disk 0 (D:)
SSD
0%

Disk 1 (C:)
SSD
0%

Ethernet
VirtualBox Host-Onl
S: 0 R: 0 Kbps

Ethernet
VMware Network Ac
S: 0 R: 0 Kbps

Ethernet
VMware Network Ac
S: 0 R: 0 Kbps

CPU Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz

Utilization
22%

Speed
3.02 GHz

Processes
348

Threads
5323

Handles
178206

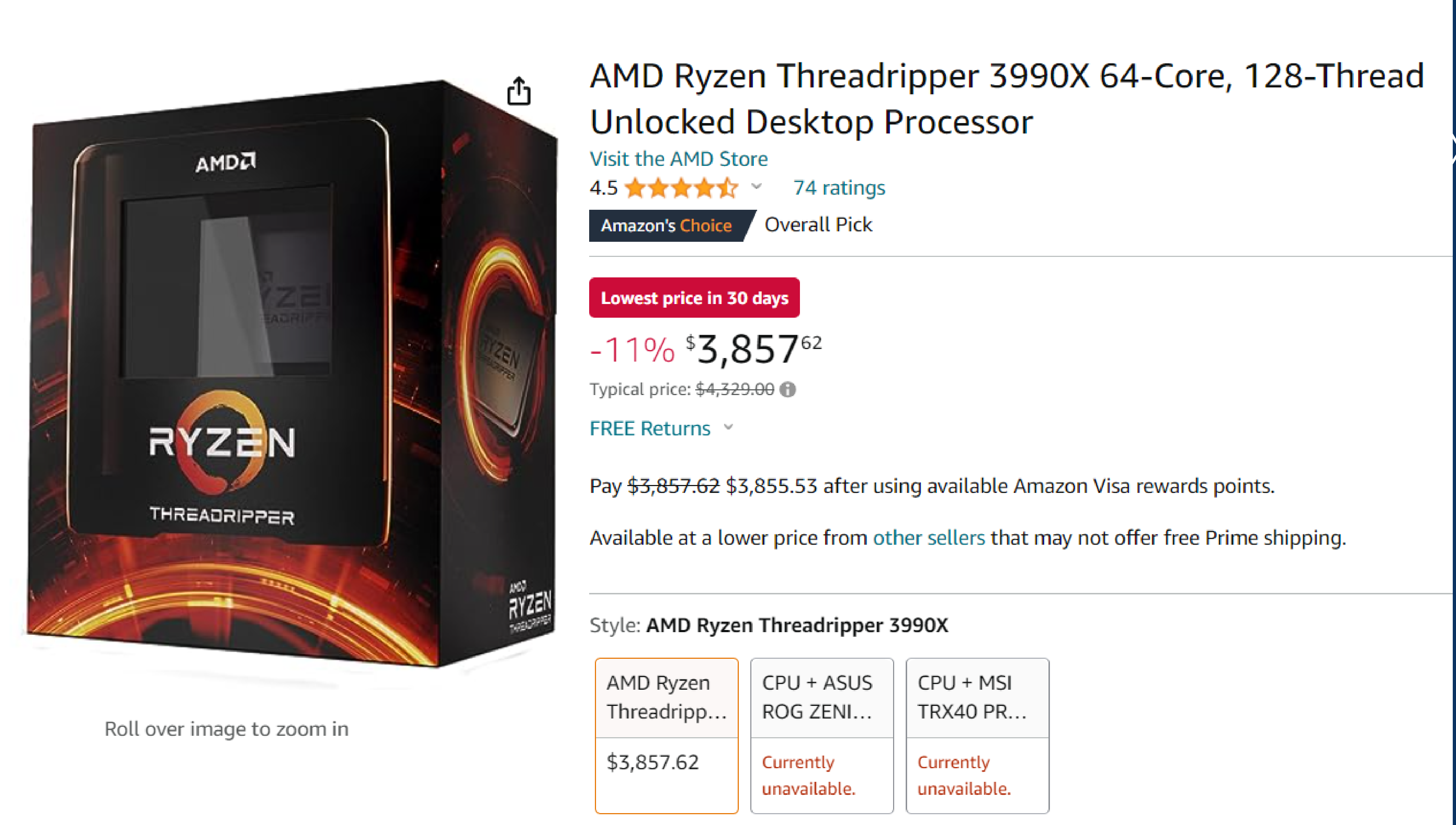
Up time
20:00:18:40

Base speed:	2.80 GHz
Sockets:	1
Cores:	4
Logical processors:	8
Virtualization:	Enabled
L1 cache:	256 KB
L2 cache:	1.0 MB
L3 cache:	6.0 MB

The Processor

- AMD Ryzen 7950X Cache
  - L1: 64K per core
    - 64K × 16 cores = 1,024K = 1MB total
  - L2: 1MB per core
    - 1MB × 16 cores = 16MB total
  - L3: 64MB total

The Processor



AMD Ryzen Threadripper 3990X 64-Core, 128-Thread Unlocked Desktop Processor

Visit the AMD Store

4.5 ★★★★☆ 74 ratings

Amazon's Choice Overall Pick

Lowest price in 30 days

-11% $3,857.62

Typical price: ~~$4,329.00~~

FREE Returns

Pay ~~$3,857.62~~ $3,855.53 after using available Amazon Visa rewards points.

Available at a lower price from other sellers that may not offer free Prime shipping.

Style: **AMD Ryzen Threadripper 3990X**

AMD Ryzen Threadripp…	CPU + ASUS ROG ZENI…	CPU + MSI TRX40 PR…
$3,857.62	Currently unavailable.	Currently unavailable.

The Processor

- Q: Which CPU is better?
  - A: It depends.
- For single-threaded workloads, higher clock speed is better.
- For multi-threaded workloads, it is better to trade clock speed for more cores.

- For the average user, more cores is better.
  - Even if a given application you use is single-threaded, that application is running alongside other applications.

← ⁘ → ✕

The Processor

- Problems with multi-core CPUs
  - More cores results in greater power consumption.
  - Increased complexity of hardware design.
  - Added overhead for ensuring integrity of data.
    - Two cores reading the same data in parallel is fine, but what happens if one overwrites it?
  - Greater complexity in data propagation.
    - L1 cache of one core updates a value also stored in L1 cache of another core.
  - Performance bottleneck if one core is waiting for another core to finish a task.

← ⁘ → ✕

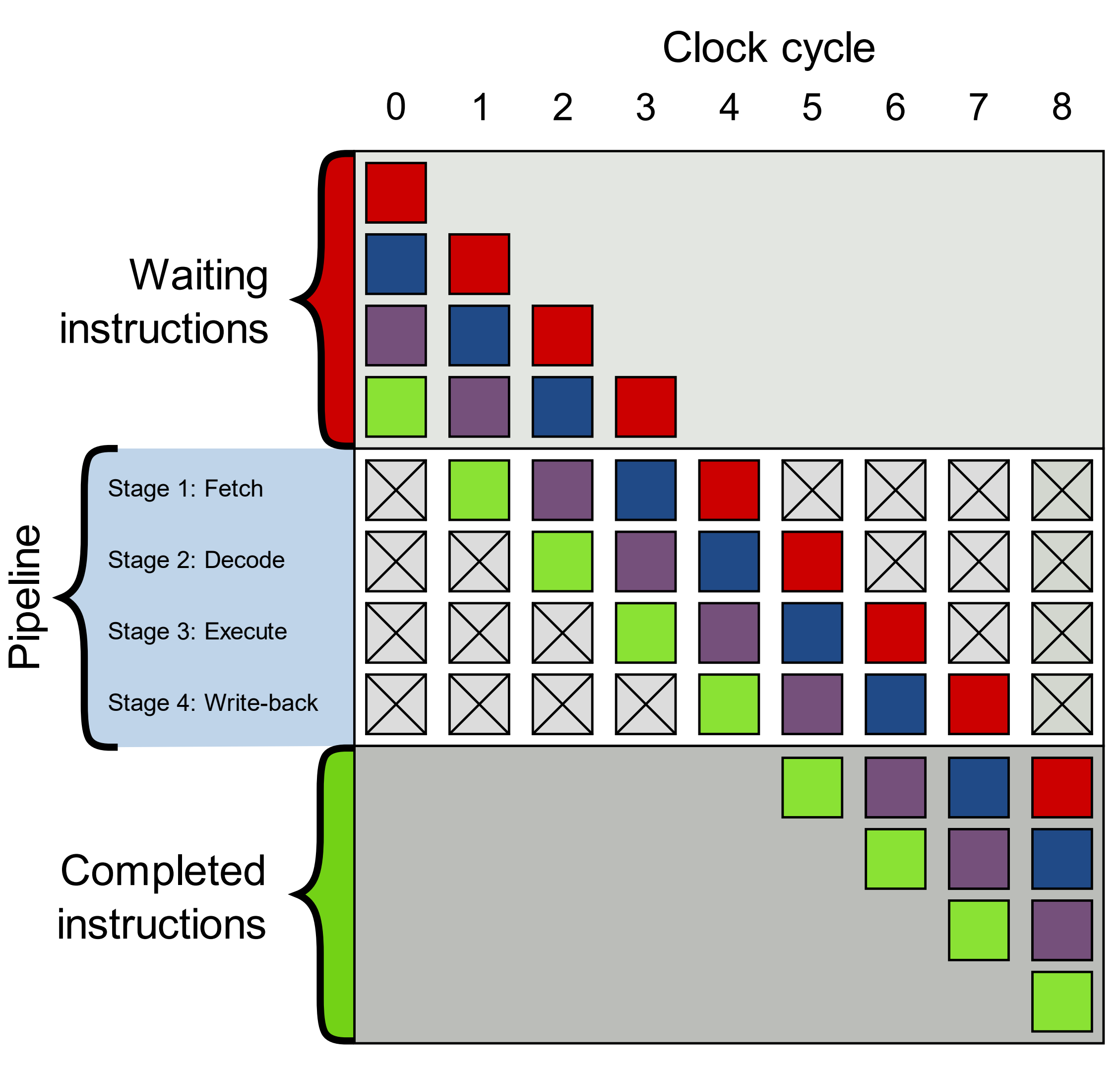
The Processor

- Another trick to make CPUs faster: Branch Prediction
- Branch Prediction
  - A feature of many CPUs whereby a prediction is made about which code path will be taken when a branch occurs in code (ex: if/else statement)
  - Without branch prediction, the proper code path is jumped to only after the conditional is evaluated
  - With branch prediction, a guess is made about which code path to take before the conditional is evaluated
  - The path that is guessed is “speculatively executed”
    - If the guess is correct, the execution is completed faster!
    - If the guess is wrong, the result of the execution is reverted, and the correct path is taken.

The Processor

- Another trick to make CPUs faster: Branch Prediction
- Branch Prediction
  - If the guess is wrong, the time wasted is equal to the number of cycles needed for a single instruction to work its way through the execution pipeline.
  - If no branch prediction is used, this time would always be wasted.
  - There are many different approaches used for doing branch prediction.
    - https://en.wikipedia.org/wiki/Branch_predictor

The Processor



The Processor

- Storytime with David: Predictive Branching Causes Problems
  - January 2018: Security researchers announce the discovery of a security vulnerability present in how CPUs implement predictive branching.
    - They nickname it Spectre
  - Problem: Branch misprediction could leave observable side effects that could reveal private data to attackers.
    - Memory accesses performed by speculative execution that relied on private data could leave that data in the CPU cache.
  - An attacker could use the cache as a side channel through which to perform a timing attack.

The Processor

- Storytime with David: Predictive Branching Causes Problems
  - An attacker could use the cache as a side channel through which to perform a timing attack.
    - Timing Attack – An attack that relies on how long a computer takes to perform certain tasks.
      - Example: Brute forcing a password by determining how long it takes for a password check to fail.
    - Side Channel Attack – An attack that relies on analyzing the effects caused by performing a certain task.
      - Example: Capturing the brightness of an LED on an electronic device to extract an encryption key.

The Processor

- Storytime with David: Predictive Branching Causes Problems
  - All CPUs made before 2019 that support branch prediction are vulnerable to this.
  - Because Spectre is a CPU flaw, it's harder to fix than a software-based security issue.
    - Spectre Mitigation
  - Depending on the CPU, patches released for Spectre caused performance degradation of up to 14%.
    - Because the patches reduced the effectiveness of branch prediction, more incorrect guesses were made.
  - http://spectreattack.com

The Processor

- Conclusion
  - Generic architecture of CPUs
  - The CPU execution cycle
  - Improving performance by pipelining the CPU execution cycle
  - Creating “faster” CPUs by adding more cores
  - Understanding tradeoffs between higher clock speeds and more cores
  - Simultaneous Multithreading
  - Branch Prediction
  - Spectre

Next Week: The CPU, Part 2

- Loading and Executing a Program
- CPU Modes of Operation
- Registers
  - Basic Program Execution Registers
  - General Purpose Registers
  - Segment Registers
  - Flag Registers
- Flags
  - Control Flags
  - Status Flags
- And more…

← ⁘ → ✕