AMD Zen Architecture

Boru Chen (boruc2)
Runyao Fan (runyaof2)
Matt Hokinson (mkh7)
Jadon Timothy Schuler (jadonts2)

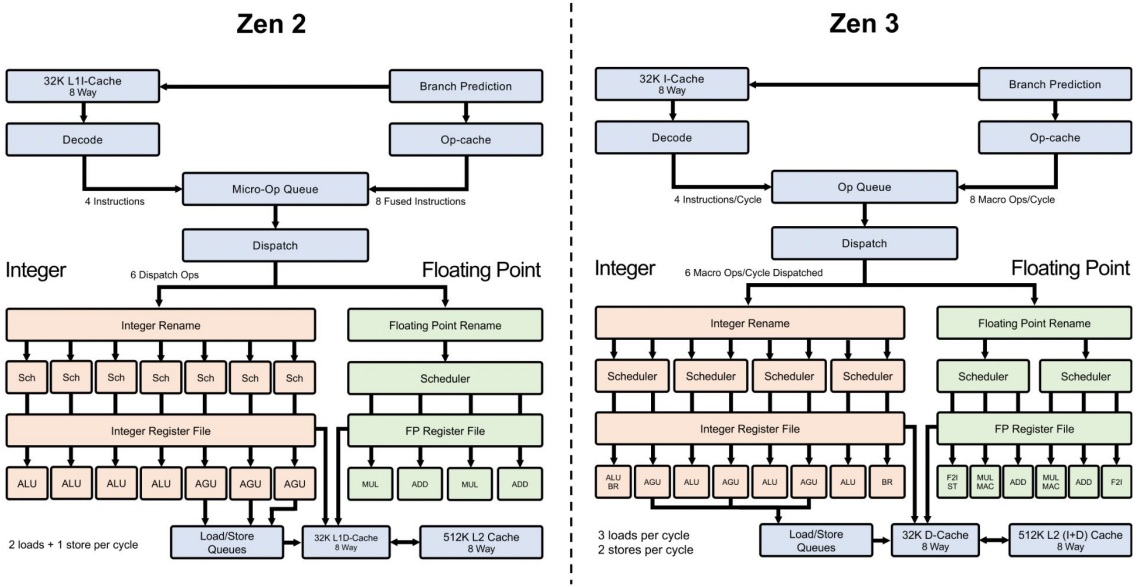
Overview

- Zen microarchitecture
- Memory Hierarchy
- Multicore
- Security

Zen Microarchitecture

Zen Microarchitecture

- AMD64 ISA
- First came out in 2017
- Zen 4: 2022
- Details of Zen 3 readily available



https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9473057

Core Microarchitecture

- Zen 3 reports 19% increase in IPC compared to Zen 2, and Zen 4 reports 13% increase in IPC compared to Zen 3
- Better Branch Prediction
- Greater Parallelism in Integer Execution
- Greater Parallelism in Floating-point Execution

https://www.amd.com/en/processors/ryzen

Core Microarchitecture - Branch Prediction

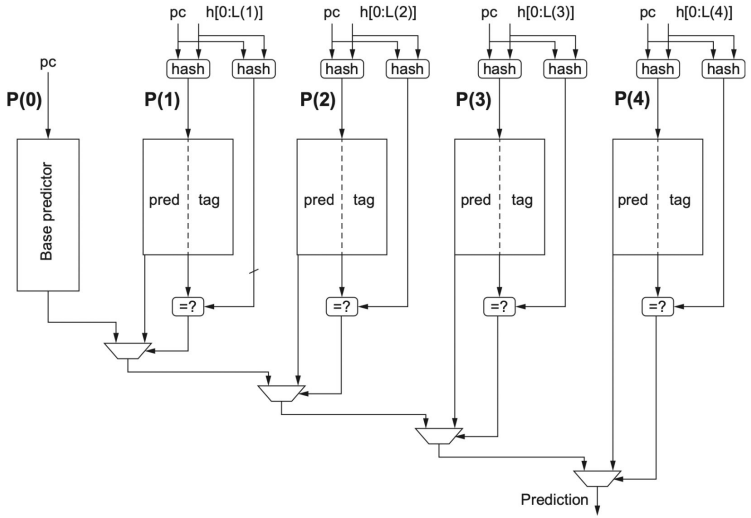


Figure 3.7 A five-component tagged hybrid predictor has five separate prediction tables, indexed by a hash of the branch address and a segment of recent branch history of length 0–4 labeled "h" in this figure. The hash can be as simple as an exclusive-OR, as in gshare. Each predictor is a 2-bit (or possibly 3-bit) predictor. The tags are typically 4–8 bits. The chosen prediction is the one with the longest history where the tags also match.

TAGE (TAgged GEometric predictors):

N predictors are used, where each predictor is indexed using a hash of PC and global branch history of varying length following a geometric series (eg: 0, 2, 4, 8, 16 when N = 5, P(0) does not consider any branch history).

Part of a predictor entry is used for tag match, where a tag is compared with a hashed value of PC and global branch history

Prediction made based on predictor with longest history and tag match.

John L. Hennessy and David Patterson, "*Computer Architecture: A Quantitative Approach*", Elsevier, sixth edition

https://hal.inria.fr/hal-03408381/

Core Microarchitecture - Branch Prediction

Some improvements of Zen 3 architecture (compared to Zen 2):

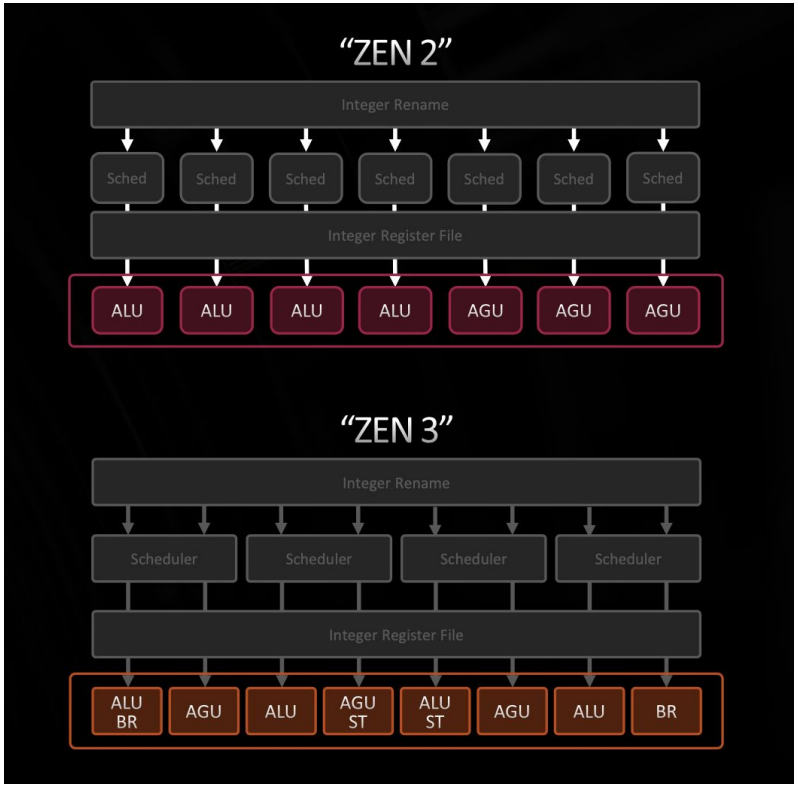
L1 branch target buffer: 512 -> 1024 entries

L2 branch target buffer: 6656 entries

Indirect target array for indirect branches: 768 -> 1536 entries

https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9718180
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9567108
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9473057

Core Microarchitecture - Integer Execution



Improvements of Zen 3 architecture (compared to Zen 2):

Reorder buffer: 224 -> 256 entries

Overall integer execution unit issue width: 7 -> 10

Instead of adding more ALUs which is costly, added new branch and store data capabilities at a lower cost, without extra write port overhead

https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9718180
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9567108
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9473057

Core Microarchitecture - Floating-point Execution

Some improvements of Zen 3 architecture (compared to Zen 2):

Scheduler: 36 -> 64 entries

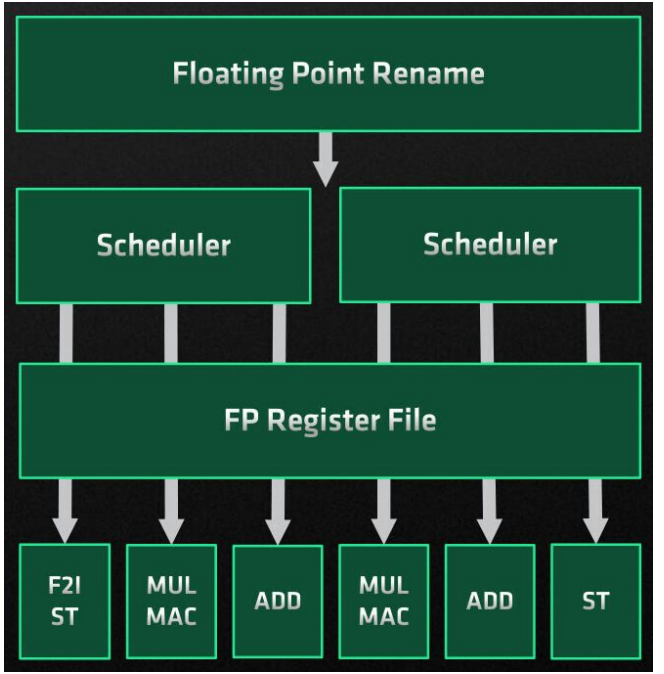
Dedicated Float-to-int and store units

Greater utilization of main functional units (2 add units and 2 mult units) on actual compute instructions

https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9718180
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9567108
https://ieeexplore-ieee-org.proxy2.library.illinois.edu/document/9473057

AVX-512

- New instruction set for 512-bit vector instructions, implemented with multiple 256-bit SIMD execution units
- Able to store only a single ROB entry for each instruction
- However, Zen has a small store queue, and can therefore only handle a single 265-bit store per cycle

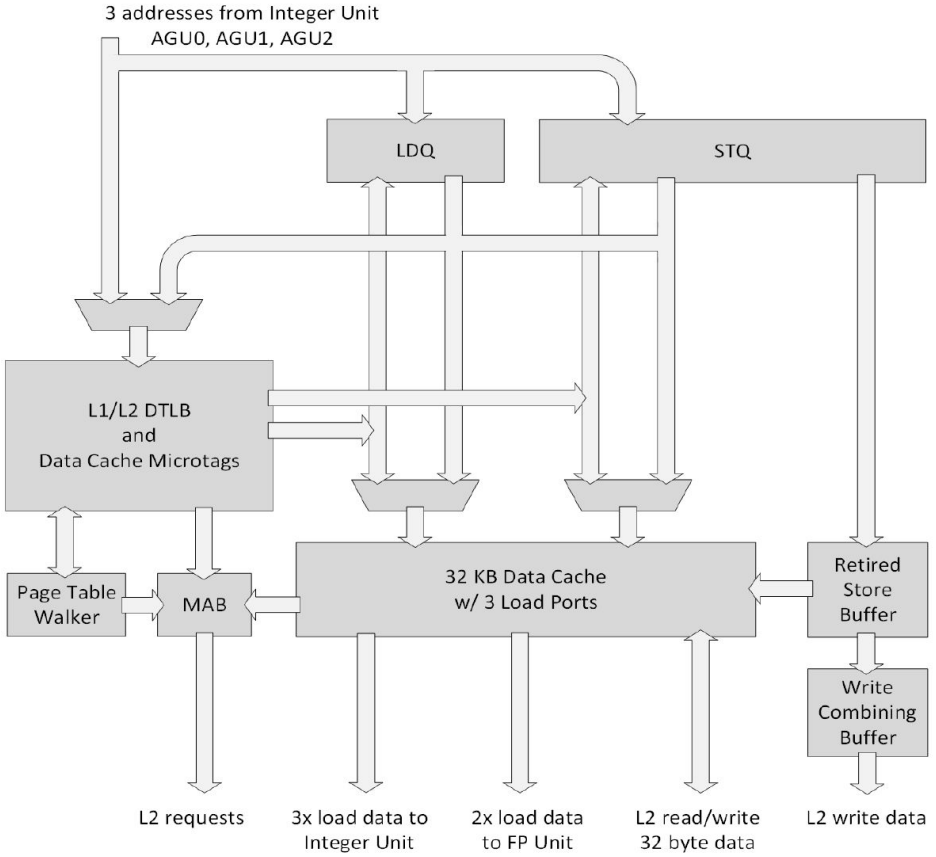


https://www.techpowerup.com/300936/amd-launches-4th-gen-epyc-genoa-zen-4-server-processors-100-performance-uplift-for-50-more-cores#g300936-32
https://www.amd.com/en/campaigns/epyc-9004-architecture

Memory Hierarchy

Load/Store Unit

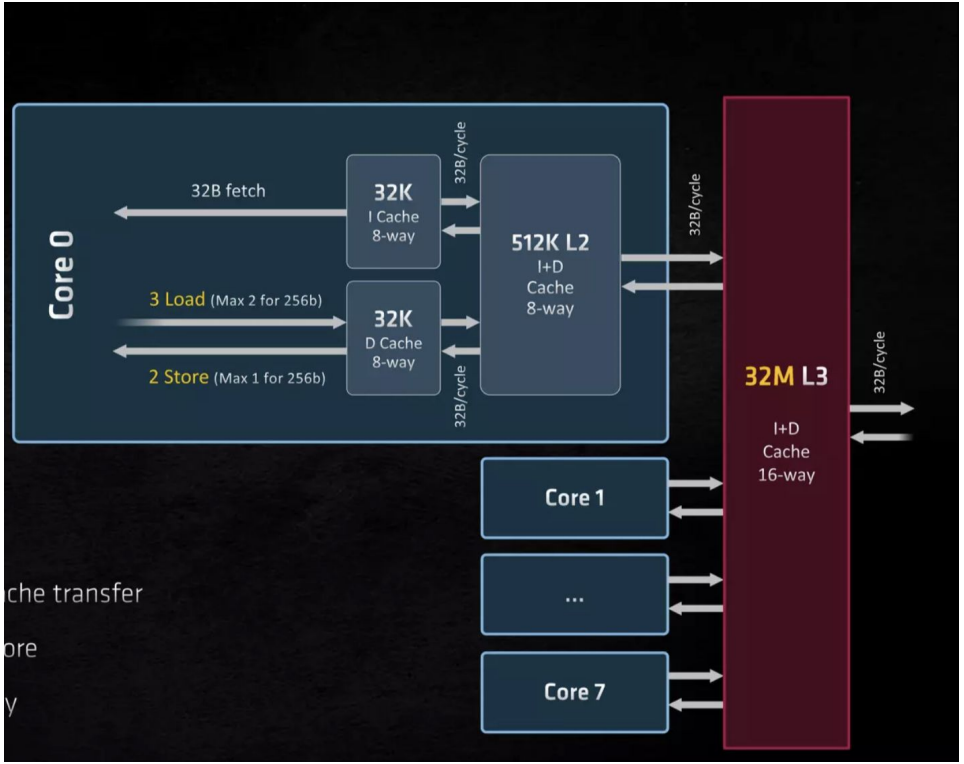
- 3 memory operation per cycle
- 72 out-of-order loads
- 24 outstanding missing with Miss Address Buffer
- Store to load forwarding (when older store contains load's bytes)
- Write combining operations
- Prefetcher
  - L1 stream/L1 stride/L1 region/L2 stream/L2 Up(down)
  - MSR disable prefetcher



56665, Software Optimization Guide for AMD Family 19h Processors (PUB)

Cache

- L1 cache
  - 32KB 8-way L1 D Cache
    - Write back
    - ECC
  - 32KB 8-way L1 I Cache
    - Prefetch with branch predictor
    - Parity
  - **Linear address utag/way-predictor**
- 512KB 8-way L2 Cache (**1MB for Zen 4**)
  - Write back
  - Inclusive
- 32MB L3 Cache
  - Write back
  - **Mostly exclusive (read code, shared by multiple cores)**
  - **Shadow L2 tag**

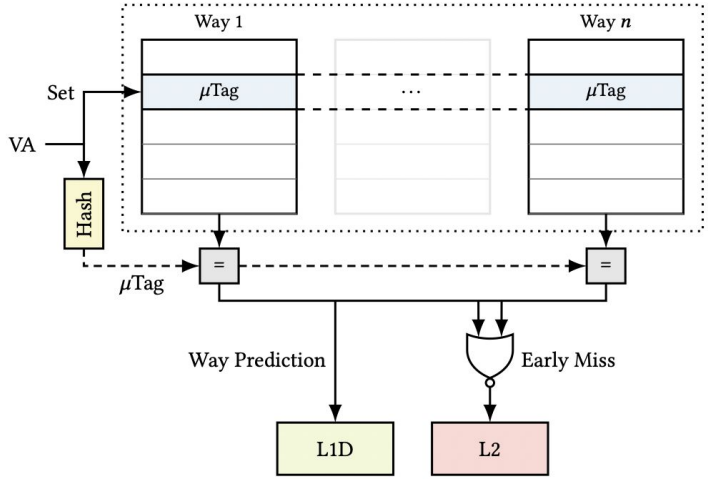


56665, Software Optimization Guide for AMD Family 19h Processors (PUB)

https://www.amd.com/en/technologies/zen-core-3

Linear address utag/way-predictor

- How work?
- Strength:
  - Speculative load data (bypass address translation)
  - Reduce bank conflicts
  - Less power consumption (large physical tag, small utag)
- Weakness:
  - Address alias
    - Continuous missing
  - Security issue
    - Timing side-channel



56665, Software Optimization Guide for AMD Family 19h Processors (PUB)

https://mlq.me/download/takeaway.pdf

TLB

- L1 ITLB / L1 DTLB
  - Fully-associative
  - 64 entries
  - 4Kbyte/2Mbyte/1Gbyte page entries
- L2 ITLB
  - 8-way set associative
  - 512 entries
  - 4Kbyte/2Mbyte page entries
- L2 DTLB
  - 16-way set associative
  - 2048 entries
  - 4Kbyte/2Mbyte page entries
- 6 Page Table Walkers

56665, Software Optimization Guide for AMD Family 19h Processors (PUB)

Multicore

Multicore - Overview **(EPYC 9004)**

Every core supports **Simultaneous Multithreading (SMT)**

- Each core supports **2 hardware threads**
- Hardware threads share a core's L2 cache

Cores are further divided into **Core Complexes (CCX)**

- **8 cores** per CCX
- 16 total concurrent hardware threads
- Shared L3 cache

Placed on a **Core Complex Die (CCD)**

https://www.amd.com/system/files/documents/58015-epyc-9004-tg-architecture-overview.pdf

Multicore - Core Complexes

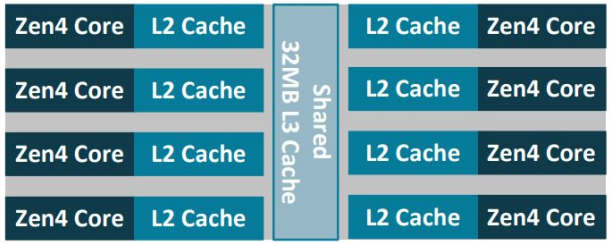


Figure 1-2: Eight Compute Cores sharing an L3 cache within a single Core Complex Die (CCD)

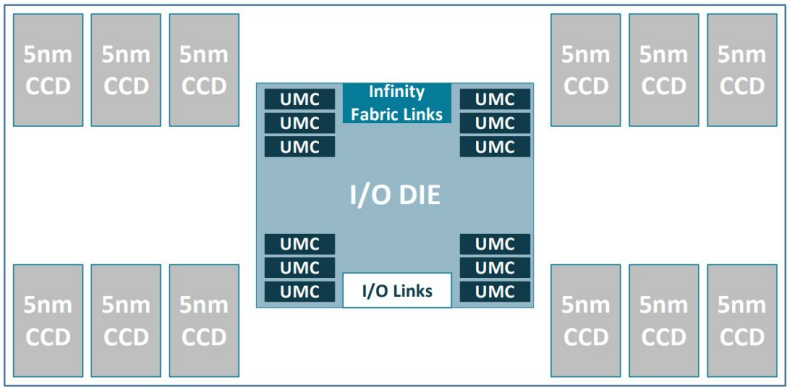


Figure 1-3: AMD EPYC 9004 Series Processor internals interconnect via AMD Infinity Fabric (12 CCD processor shown)

https://www.amd.com/system/files/documents/58015-epyc-9004-tg-architecture-overview.pdf

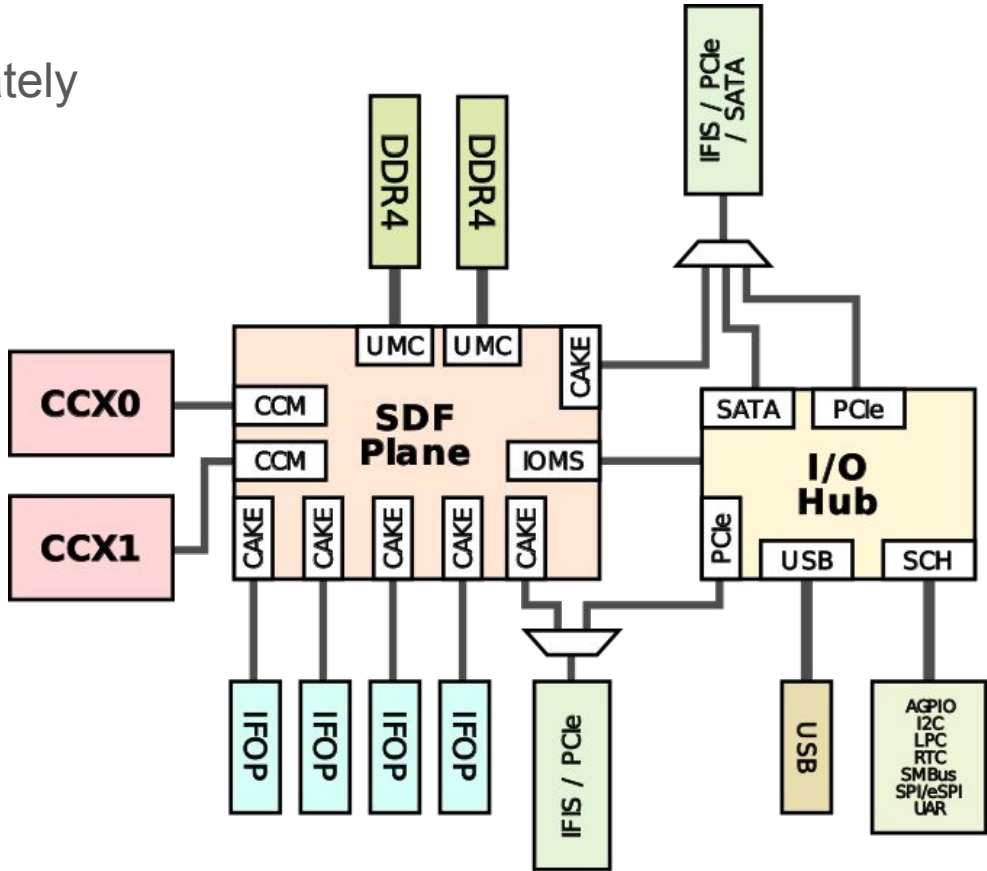
Multicore - Infinity Fabric: The Interconnect

- This is where the magic happens!
- Used between **cores**, **memory**, **CPUs**, and more
- 36 Gb/s between CCX and I/O die
- I/O die offers **12 Infinity Fabric interfaces** for CCXs
    - Each CCX supports up to 2 interfaces (72 Gb/s max bandwidth)
- Some **PCIe** lanes are shared with **Infinity Fabric**, so tradeoff between interprocessor communication and I/O lanes

https://www.amd.com/en/campaigns/epyc-9004-architecture

Multicore - Infinity Fabric Continued

- Not much up-to-date info, unfortunately
- **SDF (Scalable Data Fabric)**
  - Data communication plane
- **Cache Coherent Master (CCM)**
  - Handles coherency
- **Coherent AMD Socket Extender**
  - **CAKE** links dies/chips together
- Key Takeaway:
  - **Topology is configurable!**



https://en.wikichip.org/wiki/amd/infinity_fabric

Multicore - Network Topology

- **NUMA** (Non-Uniform Memory Access), **4 NPS** (Nodes Per Socket) shown

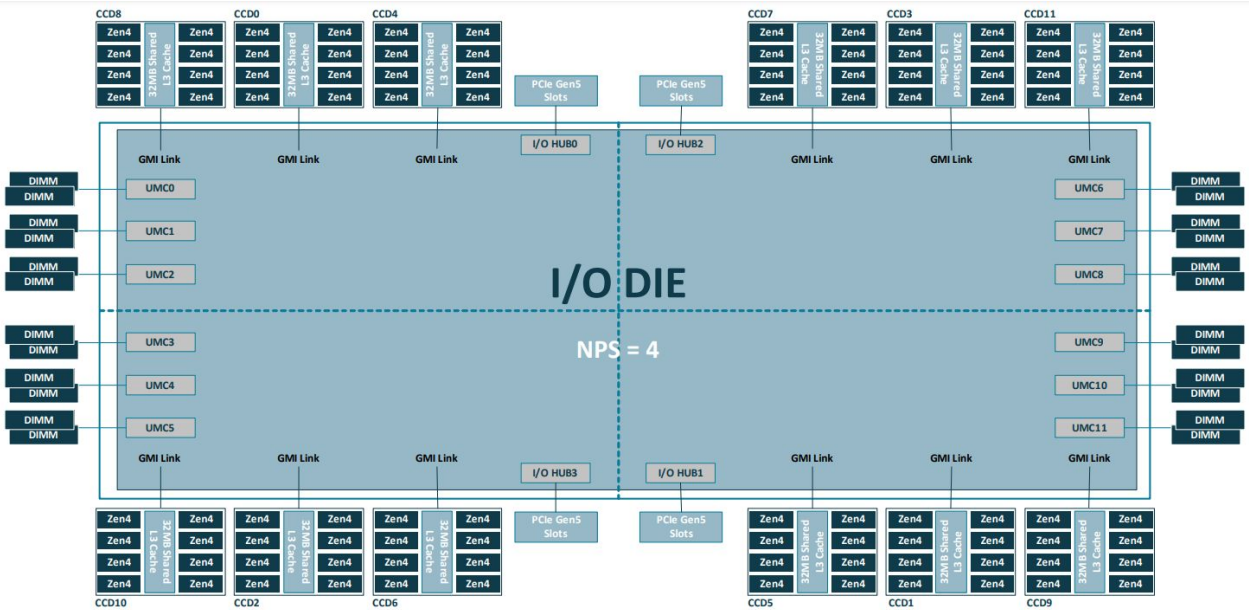


Figure 1-4: The AMD EPYC 9004 System on Chip (SoC) consists of up to 12 CCDs and a central IOD

https://www.amd.com/system/files/documents/58015-epyc-9004-tg-architecture-overview.pdf

Multicore - Network Topology Continued

- NUMA is configurable in BIOS!
  - **NPS=2:** Affinity is based on halves rather than quadrants
  - **NPS=1:** The entire processor is a single NUMA node
  - **LLC as NUMA:** Each CCD is treated as a NUMA node by its L3 cache
- Dual Socket Configuration
  - 2 identical processors connected via **xGMI (external GMI)**
- Up to 64 lanes of **CXL 1.1+** (Compute Express Link) for memory expansion
- Cache coherence
  - **MOESI (Modified, Owned, Exclusive, Shared, Invalid)** protocol used (from **AMD64**)

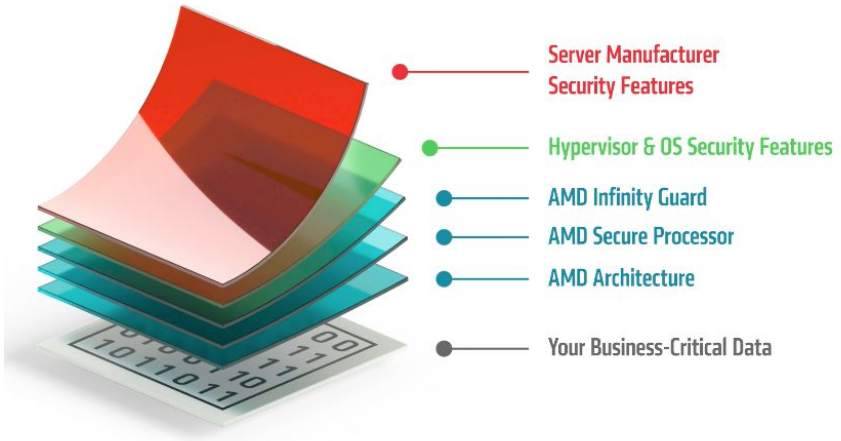
https://www.amd.com/system/files/TechDocs/24592.pdf

https://www.amd.com/system/files/documents/58015-epyc-9004-tg-architecture-overview.pdf

Security

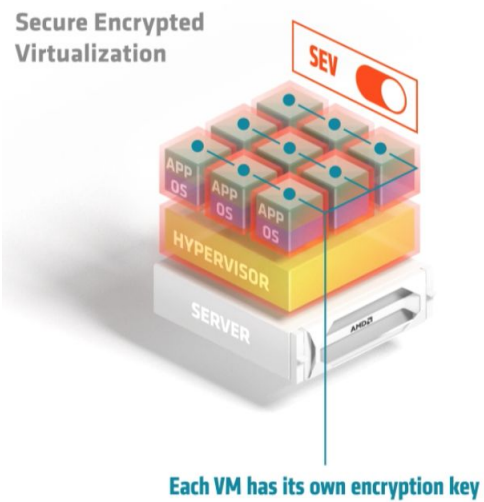
Security

- AMD Infinity Guard
- AMD Memory Guard
- AMD Shadow Stack
- Side Channel Defenses



AMD Infinity Guard

- Part of the AMD Secure Processor build in to AMD Zen Chips
  - Runs on 32-bit microcontroller with it's own secure OS
- Supports an array of features
  - HW Validated Boot
  - AMD Secure Memory Encryption (SME)
  - AMD Secure Encrypted Virtualization (SEV) and Secure Nested Paging (SNP)

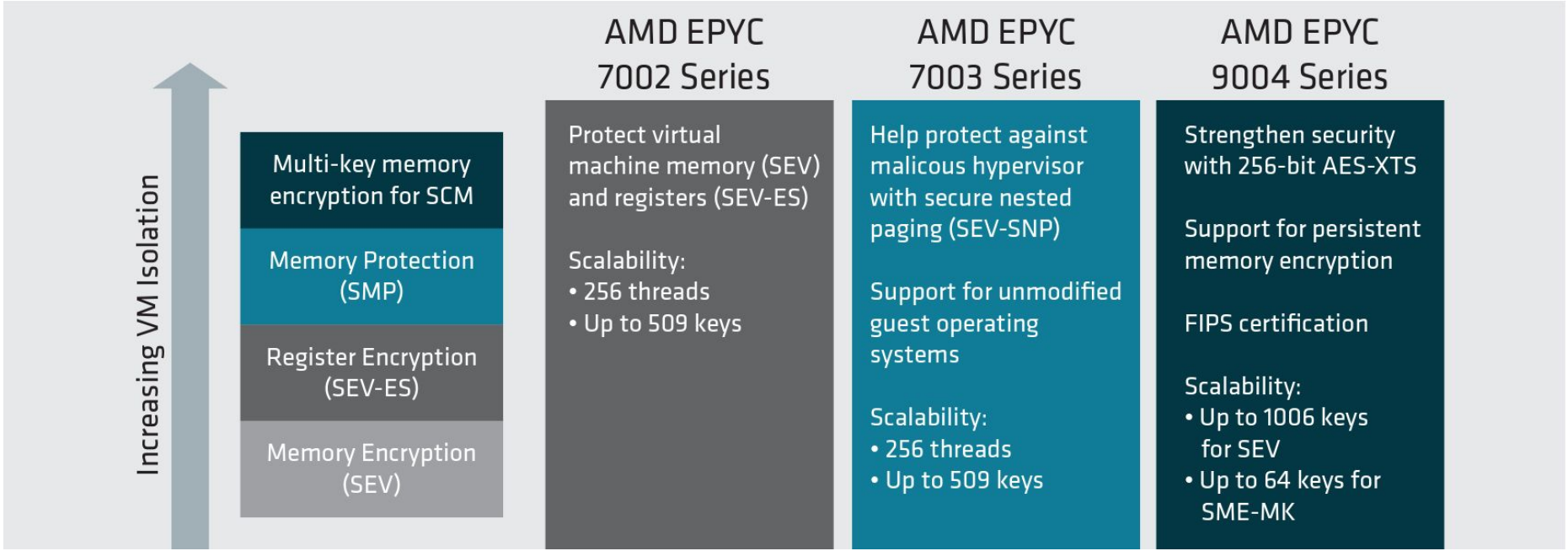


https://www.amd.com/en/technologies/infinity-guard

SEV-SNP

- Idea: Want strict isolation between different virtualized guests
- Gives each VM their own encryption key
- Isolation Focus: Use RMP (Reverse Map table) to ensure integrity/access permissions
- RMP also enforces that there can be no aliasing of physical pages and memory remapping can only be done by trusted entities

https://www.amd.com/en/processors/amd-secure-encrypted-virtualization



Virtualization Defenses

https://www.amd.com/en/campaigns/epyc-9004-architecture

AMD Memory Guard

- Built into the AMD Secure Processor (ASP)
- Threat: Some encryption metadata (AES Keys) stored in the DRAM can be read in cold boot attacks or DRAM snooping
- Memory guard provides fast (HW supported) AES encryption with random keys, stored within the processor itself

https://www.amd.com/system/files/documents/amd-memory-guard-white-paper.pdf

Shadow Stack and Side Channels

- Control Flow Hijacking (ROP)
- Uses a shadow stack to ensure integrity of return addresses

- Spectre
- Optional protection to track source of BTB entries
    - Guest/Thread/Process tagged in the buffer, goal is to avoid malicious/benign interference
    - Can flush BTB so that we don’t use speculative entries installed by others

https://www.amd.com/en/processors/amd-secure-encrypted-virtualization
https://www.amd.com/en/technologies/infinity-guard