AMD Zen

Rohin, Vijay, Brandon

Outline

1. History and Overview
2. Datapath Structure
3. Memory Hierarchy
4. Zen 2 Improvements

History and Overview

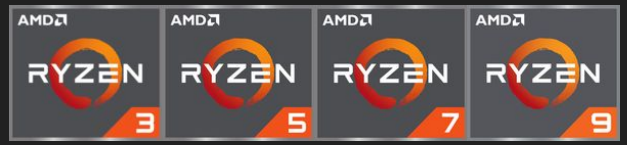
AMD History

- IBM production too large, forced Intel to license their designs to 3rd parties
- AMD fills the gap, produces clones for 15ish years - legal battles ensued
- K5 first in-house x86 chip in 1996
- Added more features like out of order, L2 caches, etc
- Current CPUs are Zen*

tomshardware.com/picturestory/713-amd-cpu-history.html

Zen Brand

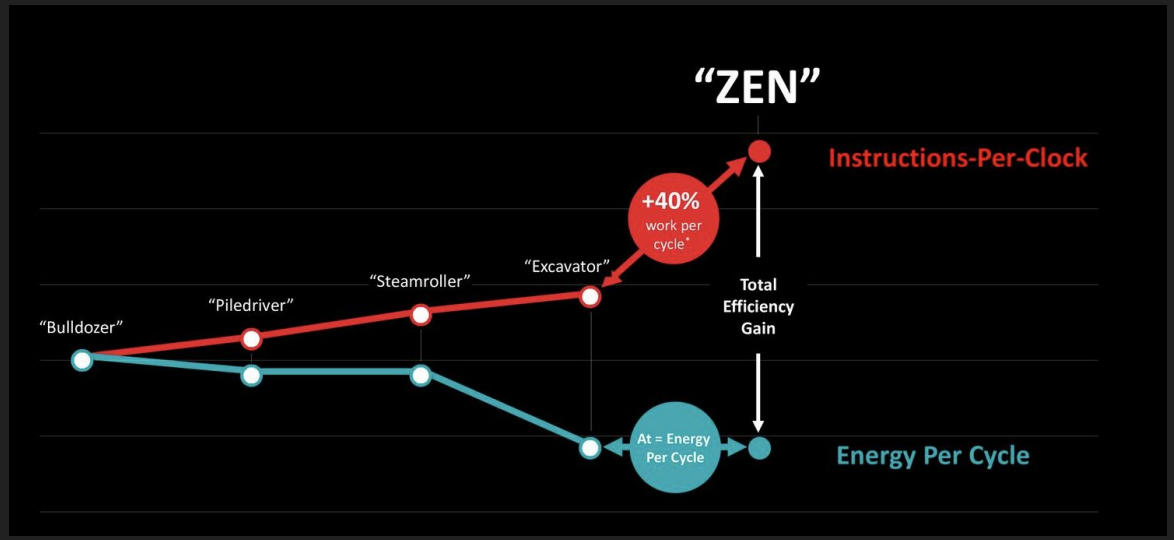
- Performance desktop and mobile computing
  - Athlon
  - Ryzen 3, Ryzen 5, Ryzen 7, Ryzen 9
  - Ryzen Threadripper
- Server
  - EPYC



https://en.wikichip.org/wiki/amd/microarchitectures/zen

Zen History

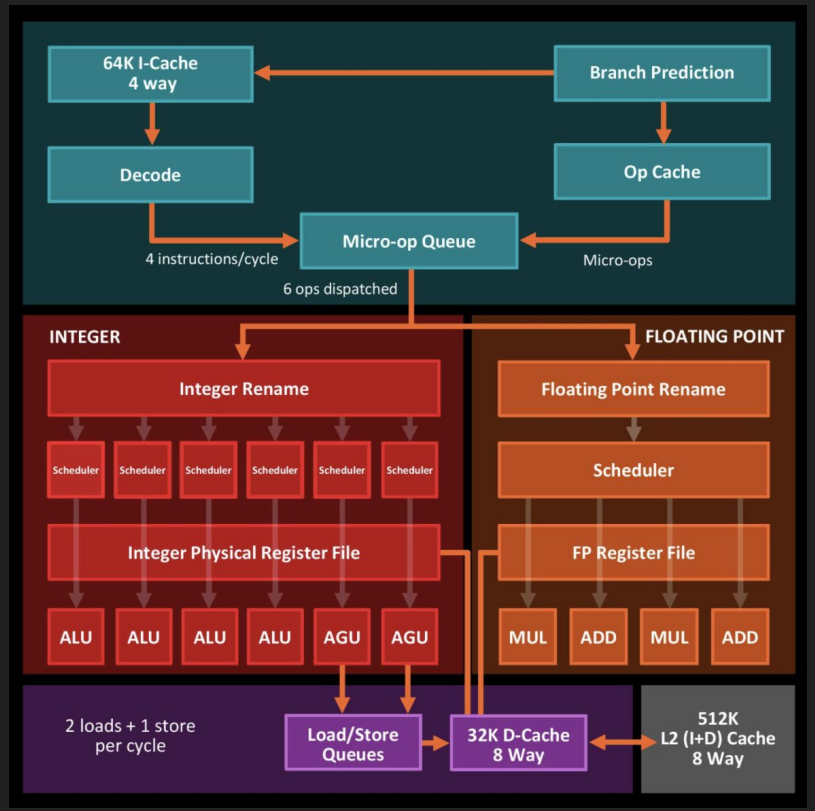
- Aimed to replace two of AMD's older chips
  - Excavator: high performance architecture
  - Puma: low power architecture



https://en.wikichip.org/wiki/amd/microarchitectures/zen#Block_Diagram

Zen Architecture

- Quad-core
- Fetch 4 instructions/cycle
- Op cache 2k instructions
- 168 physical integer registers
- 72 out of order loads
- Large shared L3 cache
- 2 threads per core

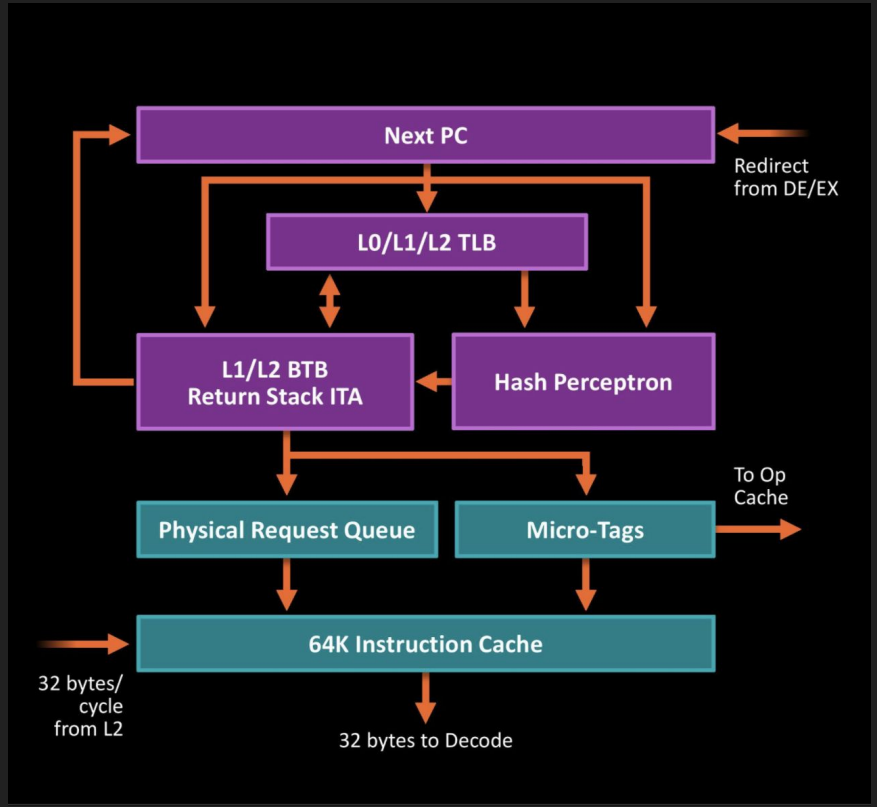


https://www.slideshare.net/AMD/amd-epyc-microprocessor-architecture

Datapath Structure

Fetch

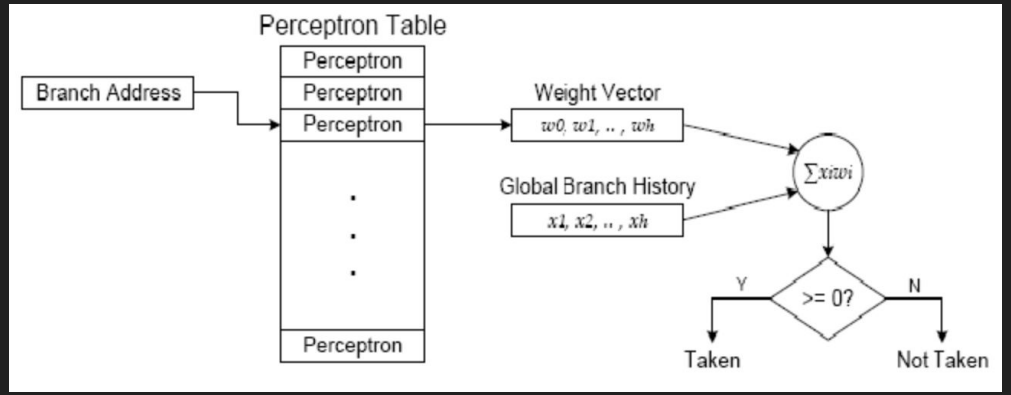
- Decoupled branch predictor
  - Runs ahead of fetches
  - Successful predictions help latency and memory parallelism
  - Mispredictions incur power penalty
- 3 layer TLB
  - L0: 8 entries
  - L1: 64 entries
  - L2: 512 entries



https://www.anandtech.com/show/10591/amd-zen-microarchiture-part-2-extracting-instructionlevel-parallelism/3

Branch Predictor

- Perceptron: simple neural network
- Table of perceptrons, each a vector of weights
- Branch address used to access perceptron table
- Dot product between weight vector and branch history vector



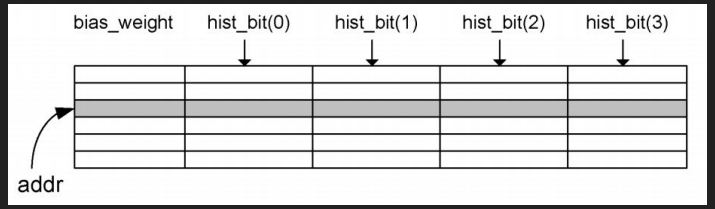
Perceptron Branch Predictor

- ~10% improve prediction rates over *gshare* predictor - (2, 2) correlating predictor
- Can utilize longer branch histories
  - Hardware requirements scale linearly whereas they scale exponentially for other predictors

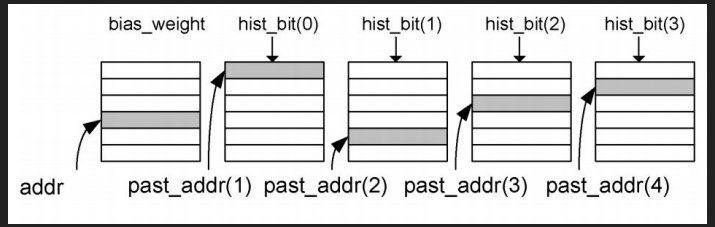
D. A. Jimenez and C. Lin, "Dynamic branch prediction with perceptrons," Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture, Monterrey, Nuevo Leon, Mexico, 2001, pp. 197-206, doi: 10.1109/HPCA.2001.903263.

Hashed Perceptron Branch Predictor

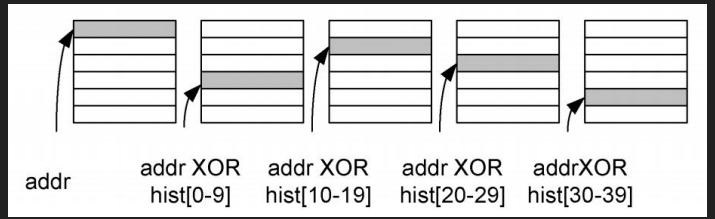
- Combines concepts behind global predictors and path-based predictors
- Hashes segments of histories to access different weight tables
- Sum weights and apply threshold to predict



global



path-based

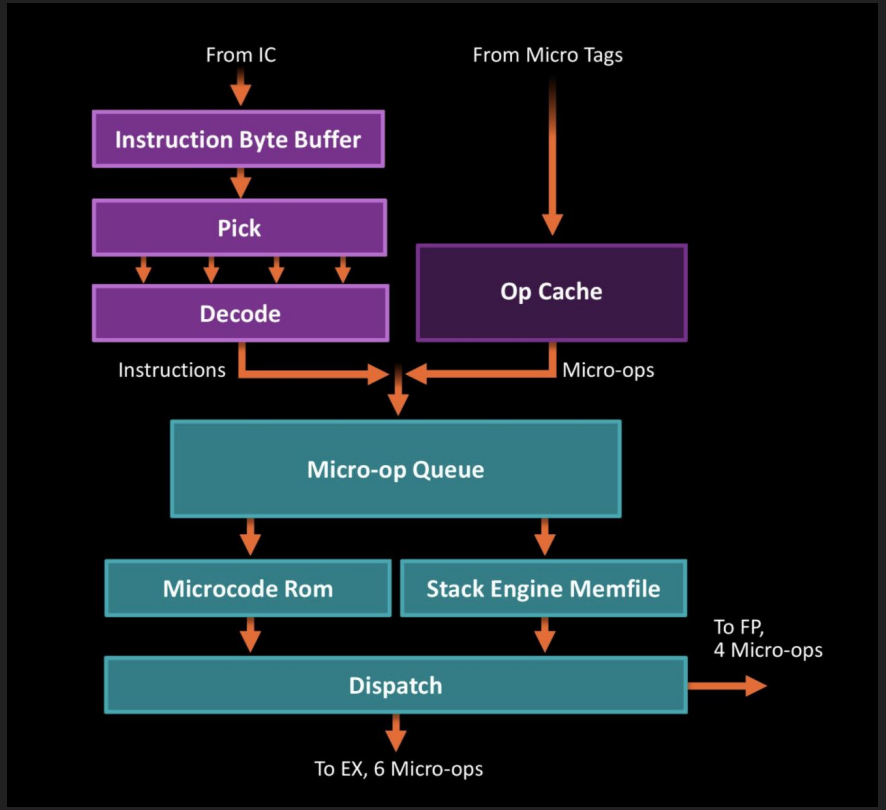


hashed-perceptron

David Tarjan and Kevin Skadron. 2005. Merging path and gshare indexing in perceptron branch prediction. ACM Trans. Archit. Code Optim. 2, 3 (September 2005), 280–300. DOI:https://doi.org/10.1145/1089008.1089011

Decode

- 2k instruction Op cache
- Can handle 4 instructions per clock cycle (including instructions generating two micro-ops)



https://www.anandtech.com/show/10591/amd-zen-microarchiture-part-2-extracting-instructionlevel-parallelism/3

Op cache

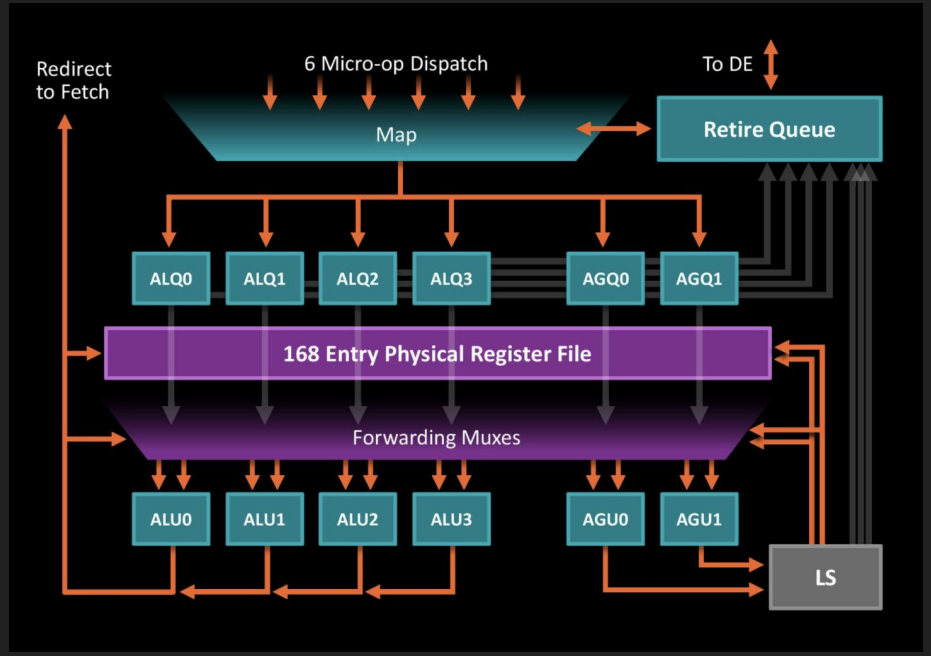
- a.k.a. Micro-op cache or µop cache
- Variable-length complex instructions are often decoded into smaller fixed-length micro-operations (µops)
- If an instruction has been decoded and cached in the Op cache, can save time and power
- Micro-Tags determine whether to access instruction from Op cache or L1
- 2048 µops in Zen 1
- 4096 µops in Zen 2

Execute

- Two separate units for Integer and Floating Points
- Each unit has:
  - Register file with register renaming
  - Schedulers
  - Several execution units
- Both units have access to a Retire Queue
  - Similar to a Reorder Buffer
  - 192 entries for Zen 1
  - 244 entries for Zen 2
  - Retires 8 instructions per cycle

Integer Execute

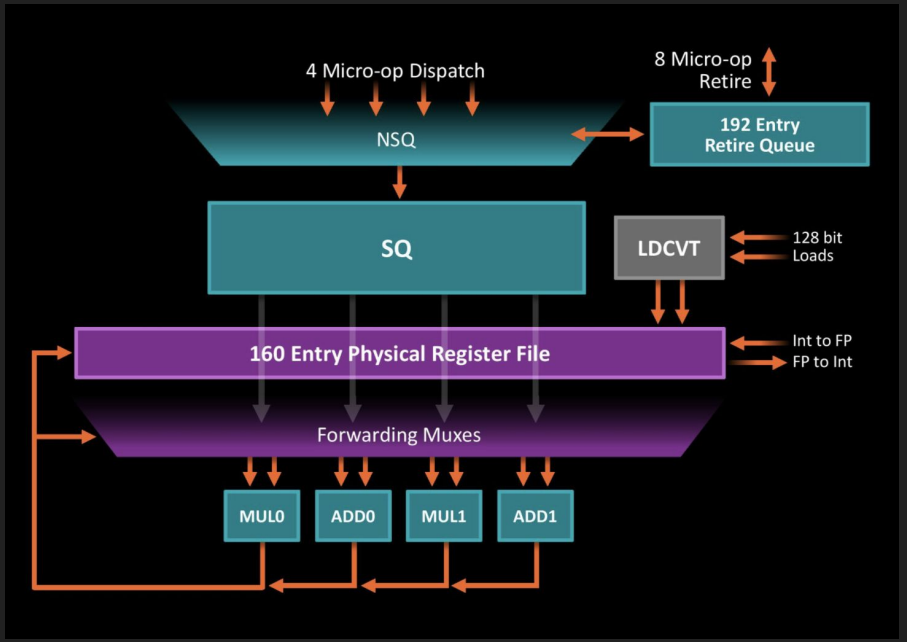
- Register file
  - Zen 1: 168 registers of 64 bits
  - Zen 2: 180 registers of 64 bits
- Scheduling Queues
  - 14 entries each
  - 4 ALQs
  - 2 AGQs
- 6 issue per cycle
  - 4 ALUs
  - 2 AGUs (Address Generation Units)



https://www.anandtech.com/show/10591/amd-zen-microarchiture-part-2-extracting-instructionlevel-parallelism/4

Floating Point Execute

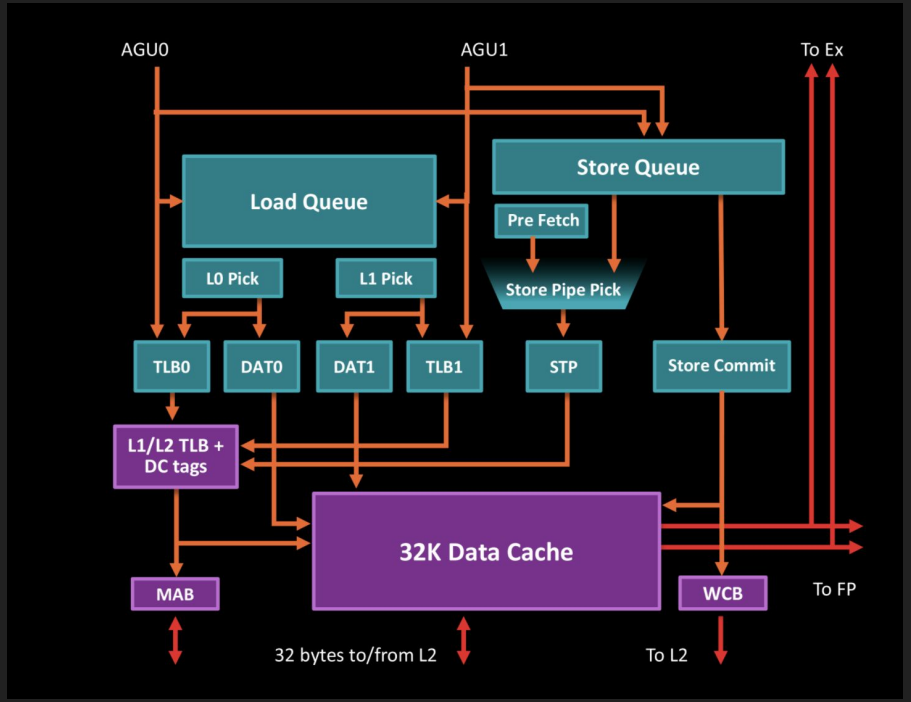
- Floating point register file
  - Zen 1: 160 vector registers of 128 bits
  - Zen 2: 160 vector registers of 256 bits



https://www.anandtech.com/show/10591/amd-zen-microarchiture-part-2-extracting-instructionlevel-parallelism/4

Load/Store

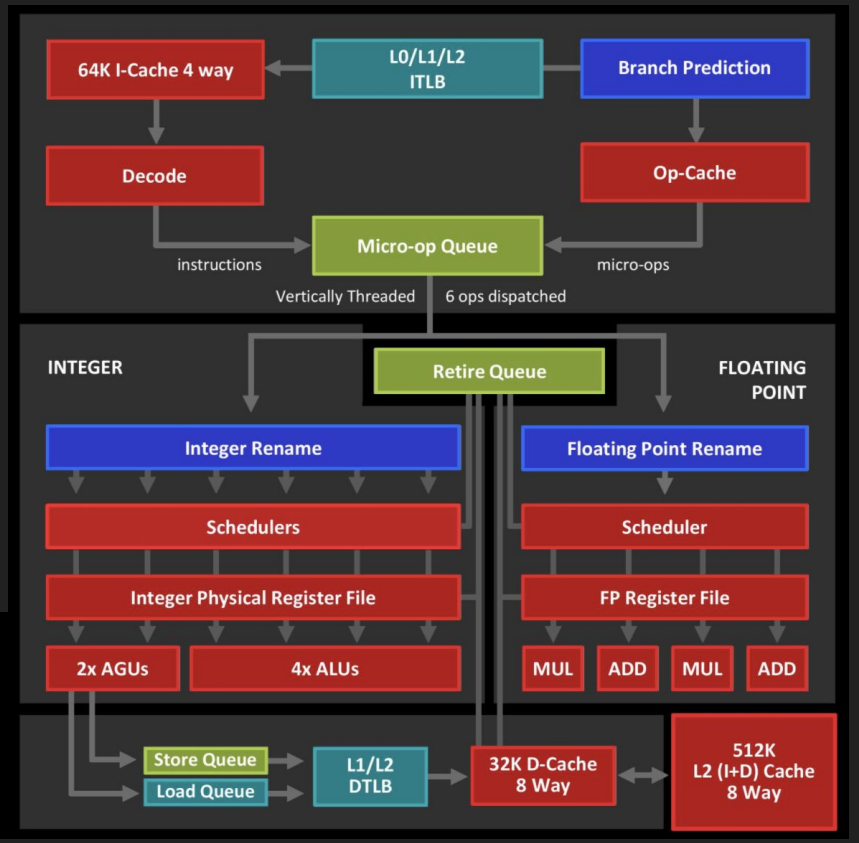
- Load Queue: 44 entries
  - AGU depth of 28
  - = 72 inflight loads at a time
- Store Queue
  - Zen 1: 44 entries
  - Zen 2: 48 entries
- L1 data cache
  - 32K
  - 8 way



https://www.anandtech.com/show/10591/amd-zen-microarchiture-part-2-extracting-instructionlevel-parallelism/4

Zen - SMT

- AMD's first Simultaneous Multithreading
- Allows instructions from multiple threads to be issued in the same cycle
- Queues round-robin
- Tagged means marked ahead of time - UI needs preferential treatment
- Algorithmic uses internal analysis of access patterns, etc



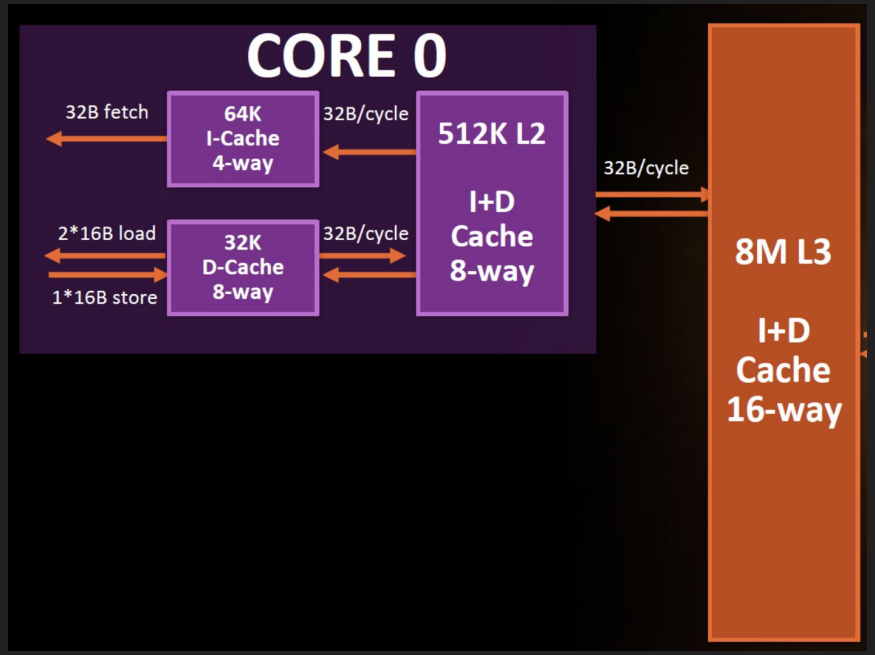
https://www.anandtech.com/show/11170/the-amd-zen-and-ryzen-7-review-a-deep-dive-on-1800x-1700x-and-1700/10

Competitively shared structures
Competitively shared and SMT Tagged
Competitively shared with Algorithmic Priority
Statically Partitioned

Memory Hierarchy

Cache Hierarchy

- L1
  - Write-back
  - Separate I and D caches
  - 8-way set associative
- L2
  - Write-back
  - 2x size of Intel Skylake and Broadwell
  - 8 way set associative
  - = 1.414x chance of cache hit
- L3
  - Zen 1: 2 MB per core
  - Zen 2: 4 MB per core
  - 16 way set associative
  - Acts as a victim cache for L1/L2 victims



https://www.anandtech.com/show/10578/amd-zen-microarchitecture-dual-schedulers-micro-op-cache-memory-hierarchy-revealed/2

Zen 2 Improvements

Improvements

- 2x L3 cache size
- 2x L1 load/store bandwidth
- 2x Op cache size
- +15% IPC
- Hardware mitigations to Spectre vulnerability
- 256-bit datapath as opposed to 128-bit
- Infinity Fabric 2 (I/O) - 2.3x transfer rate per link (25 GT/s, up from ~10.6 GT/s)
- Floating Point Bandwidth - 256 bit (improvement over 128 bit); allows faster physics simulation and faster inline memcpy and memset for all instructions using 256 bit intrinsic registers