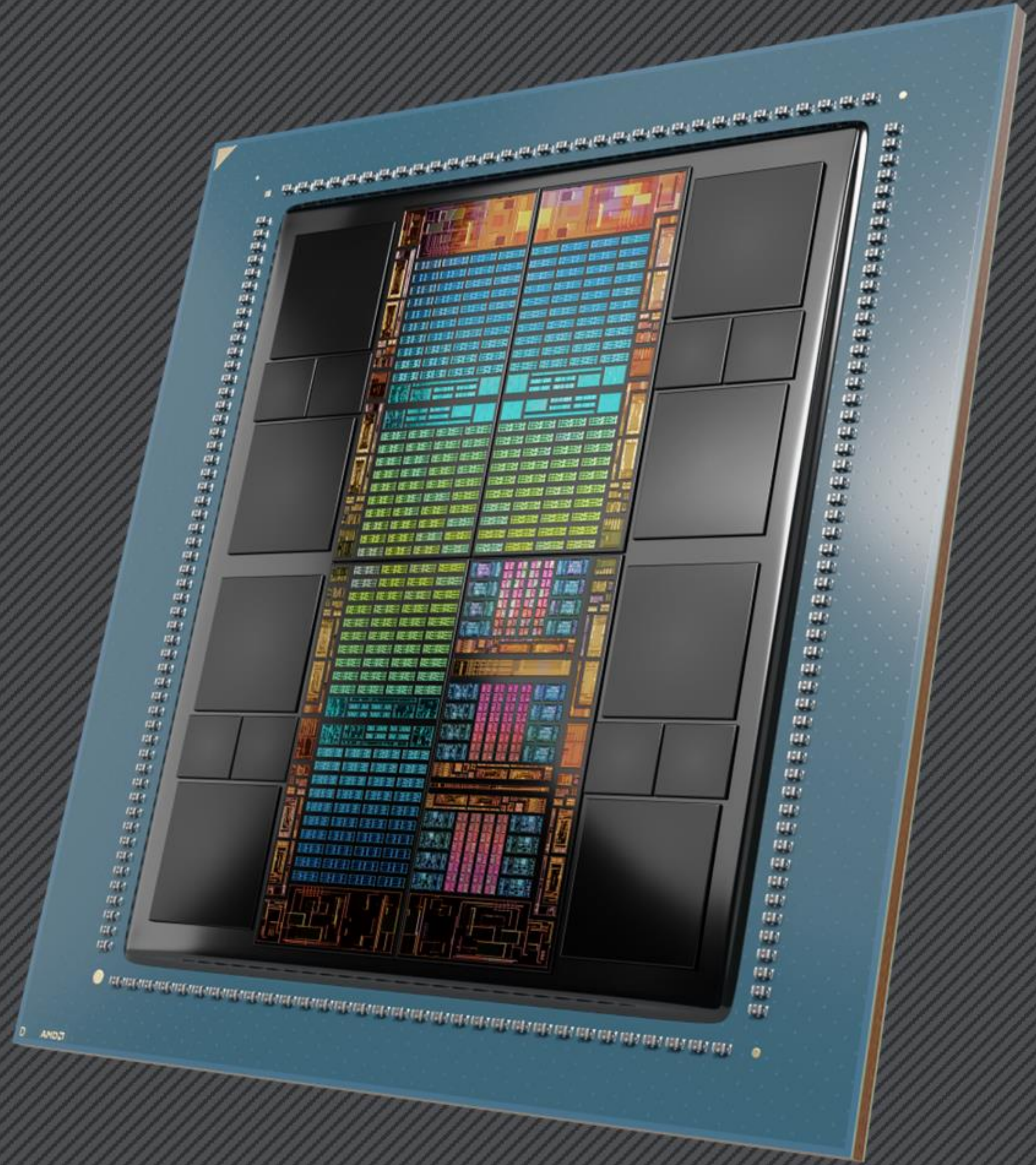
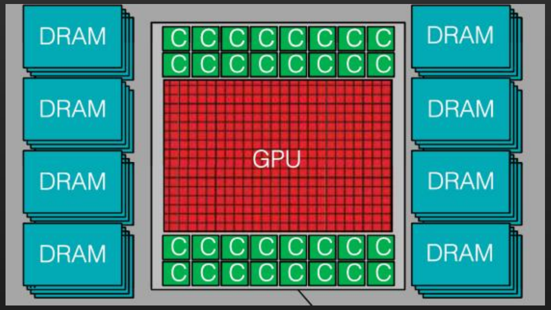
AMD Instinct^{TM} MI300A APU Architecture, Shader ISA Extension, Profiling and Instrumentation Support in ROCm

Timour Paltashev
Vladimir Indic
Nursultan Kabylkas

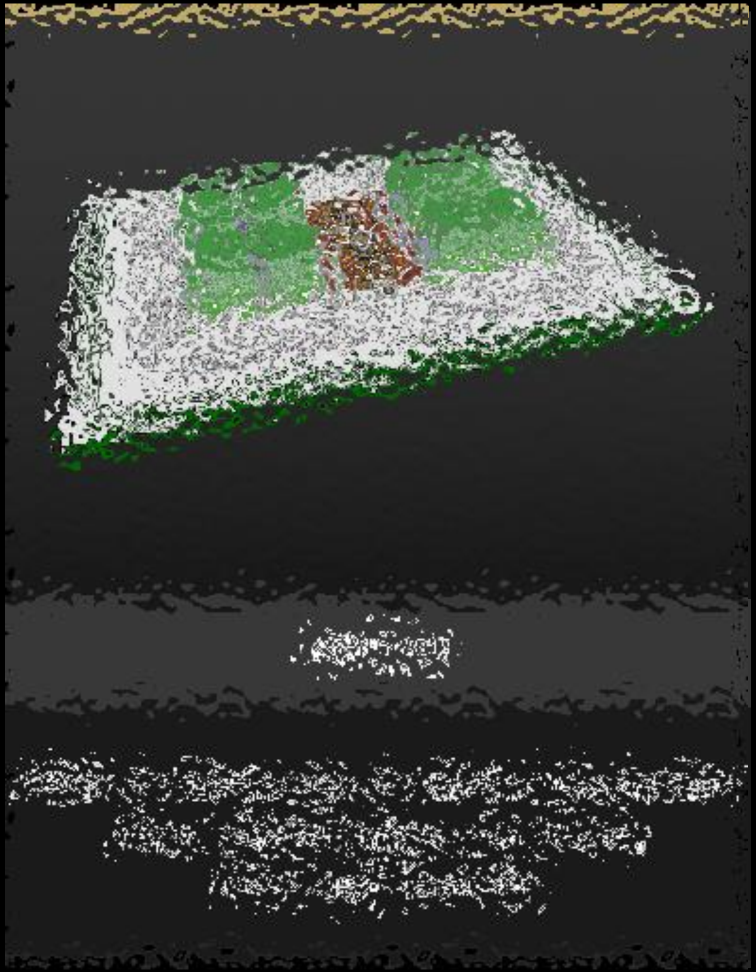


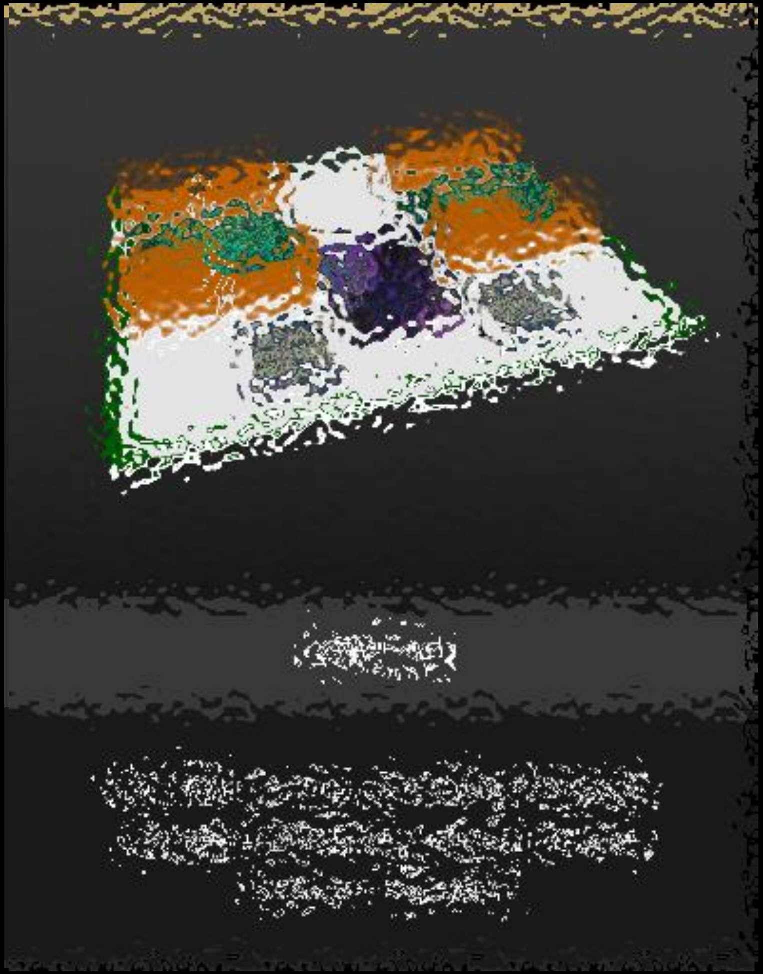
Exascale/HPC APU Vision



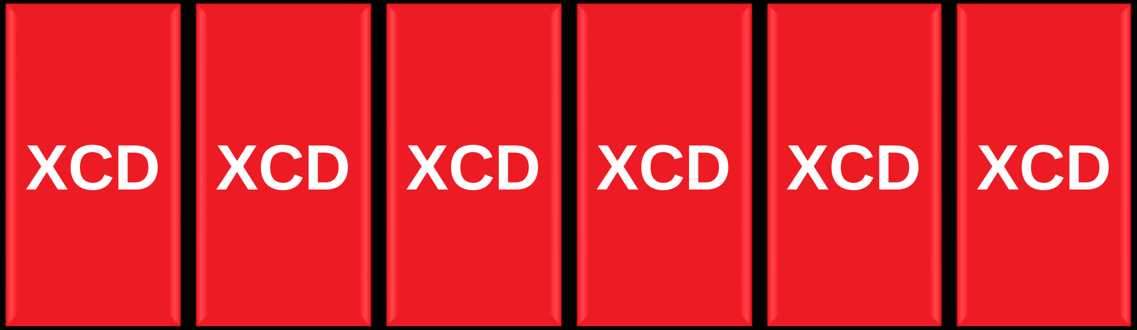
APU

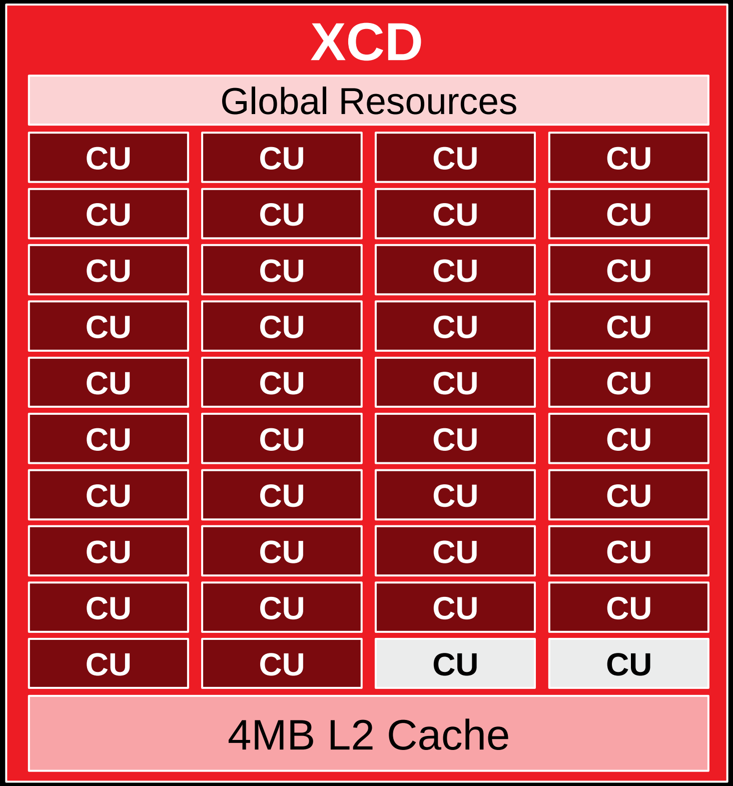
Co-packaged CPU, GPU, Memory
Unified/shared memory space
Zero-copy and programmability





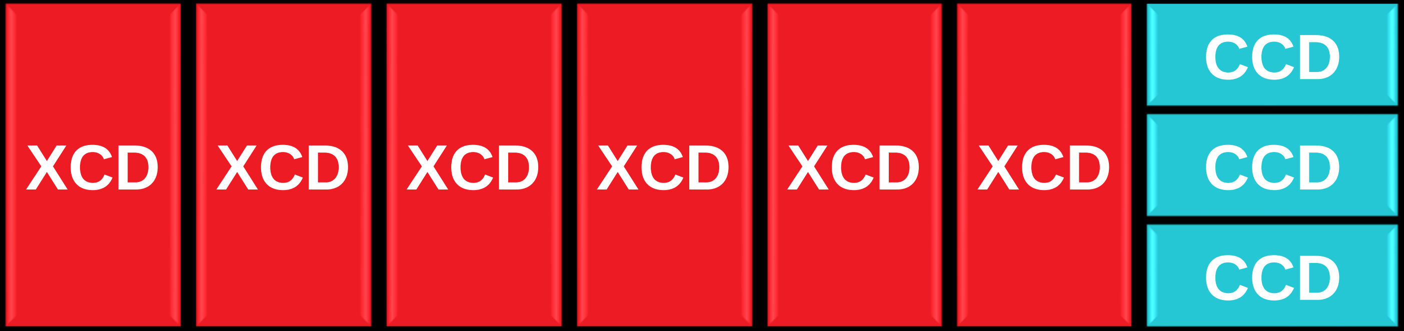
AMD Instinct™ MI300A Accelerated Processing Unit

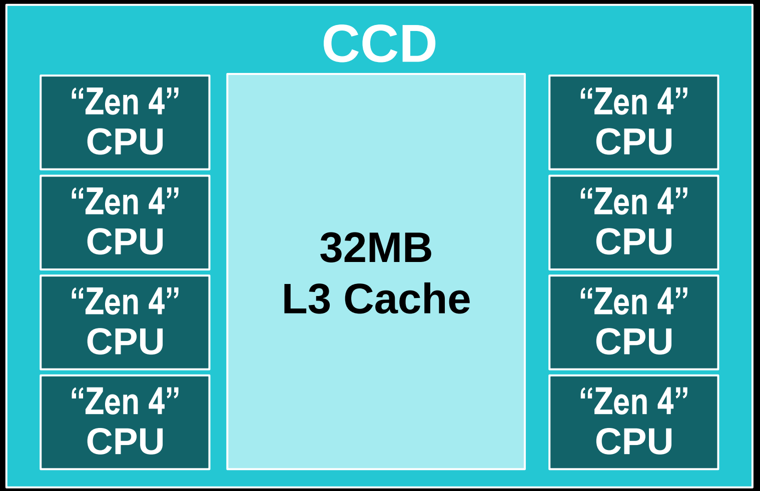




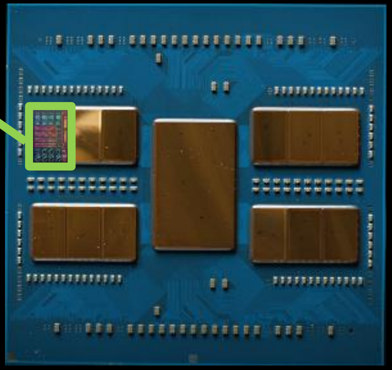
- XCD – Accelerator Complex Die
- 38 CUs per XCD, 228 total AMD CDNA™ 3 architecture

AMD Instinct™ MI300A Accelerated Processing Unit

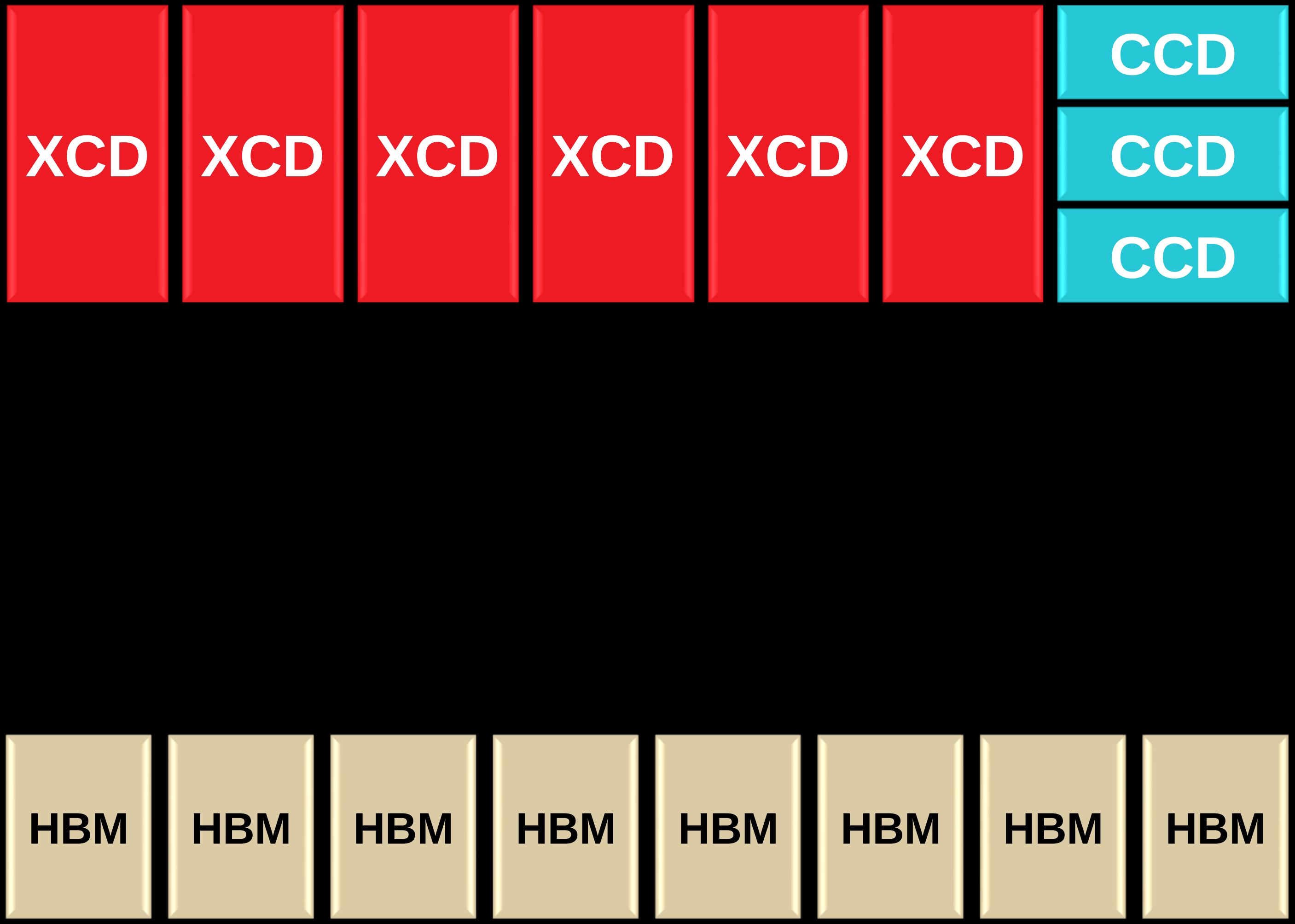




- CCD – CPU Complex Die
- 8 "Zen 4" cores per CCD, 24 total
- Leverage CCD from EPYC



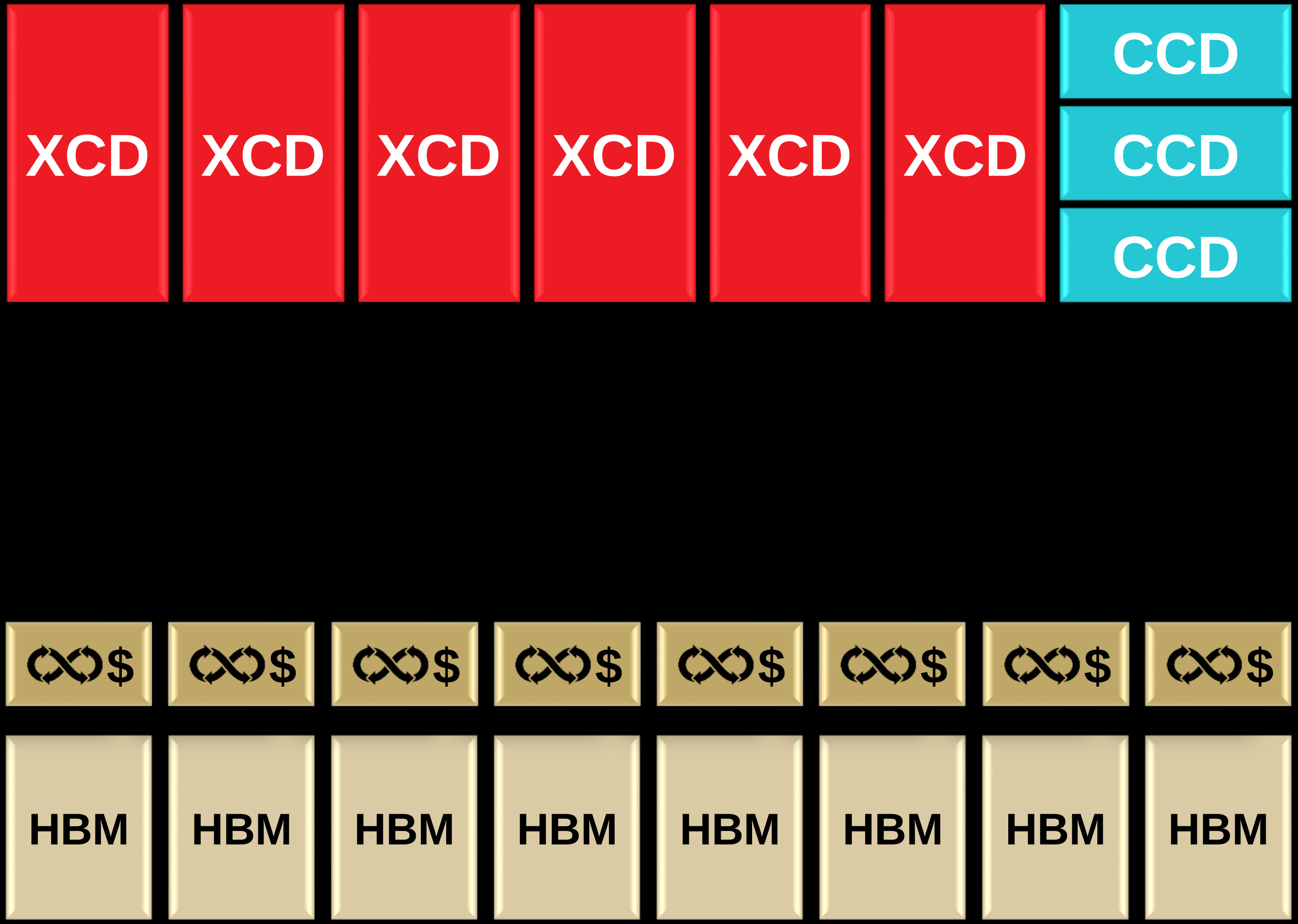
AMD Instinct™ MI300A Accelerated Processing Unit

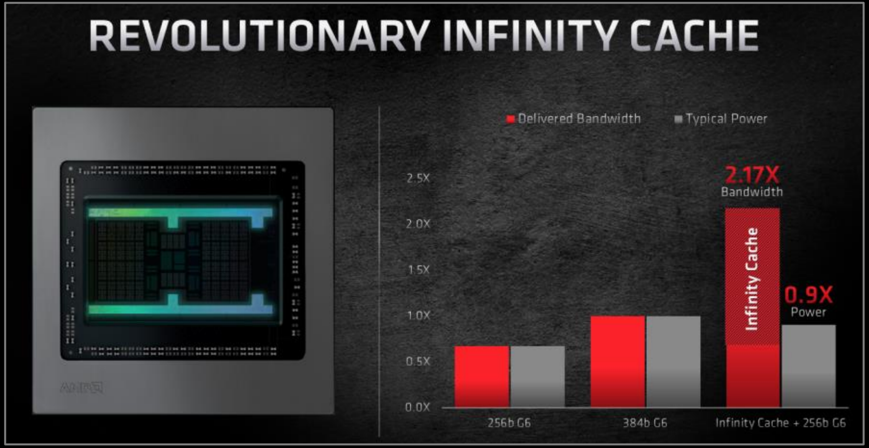


- HBM – High Bandwidth Memory
- HBM gen 3, 16GB (128 GB total)
- 665 GB/s/stack (5.3 TB/s total)
- 128 total memory channels

AMD
together we advance_

AMD Instinct™ MI300A Accelerated Processing Unit

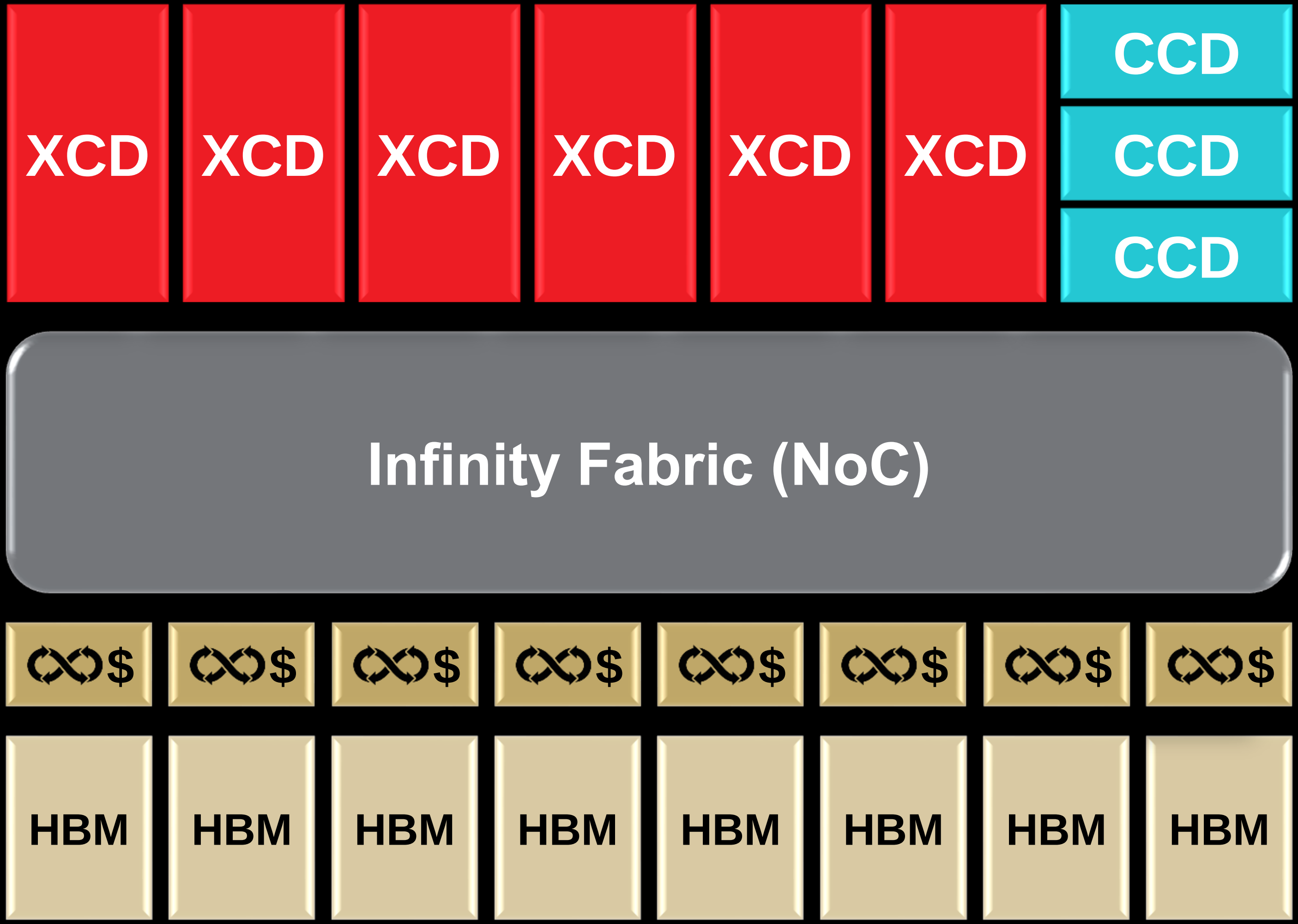


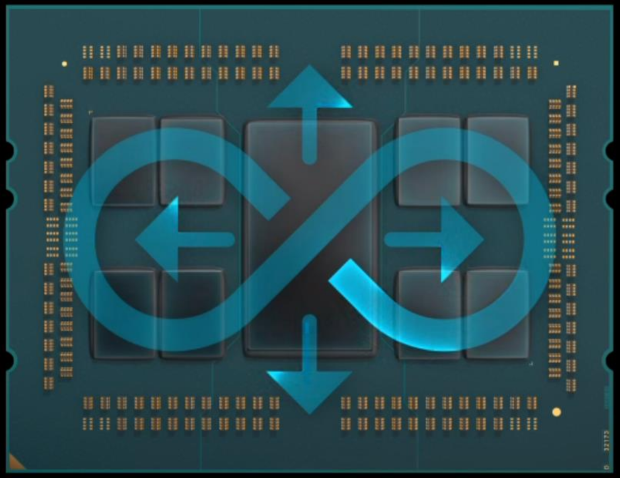


- Memory-side Infinity Cache
- 2MB/channel (256 MB total)
- BW amplification (up to 17 TB/s)

AMD
together we advance_

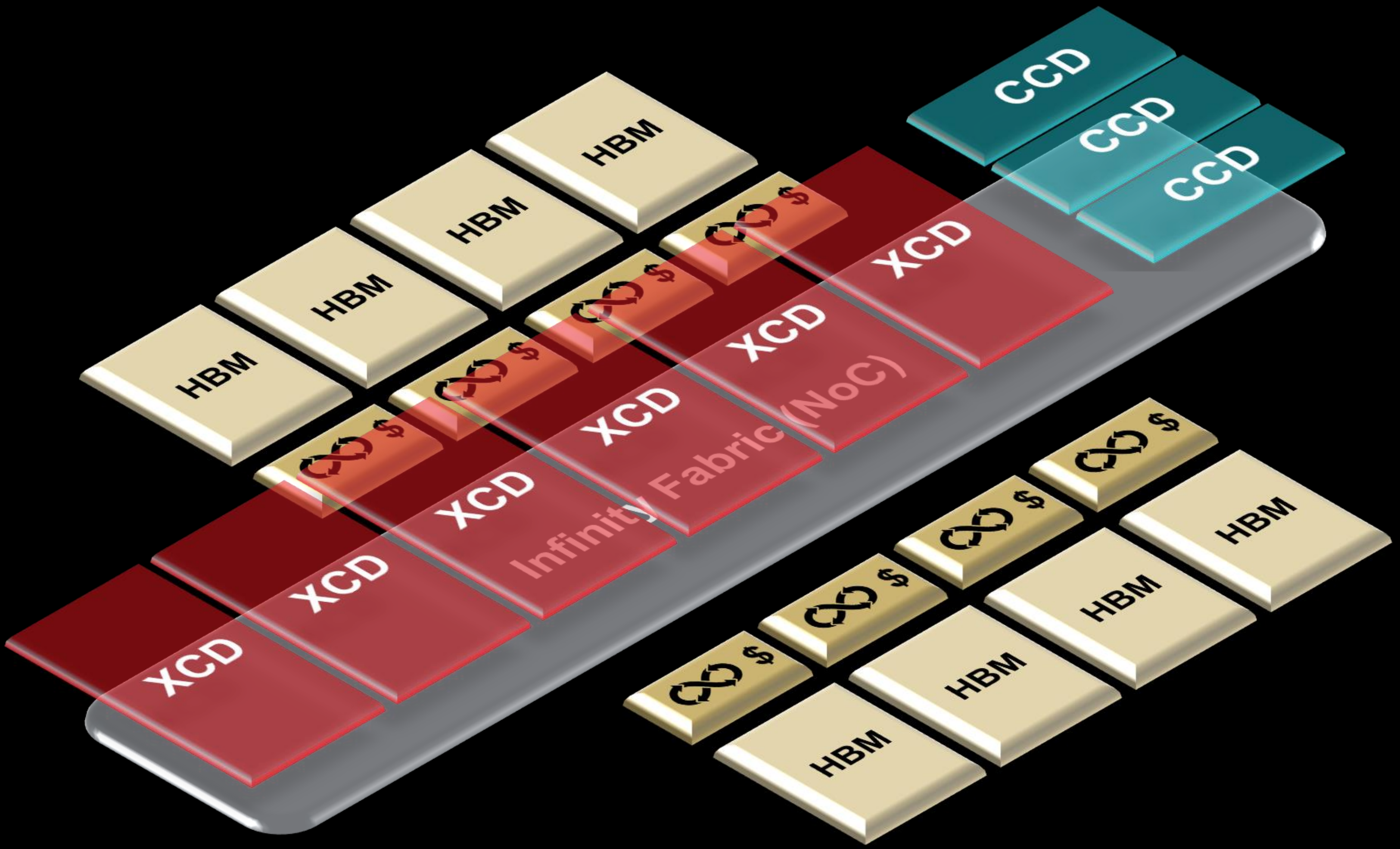
AMD Instinct™ MI300A Accelerated Processing Unit



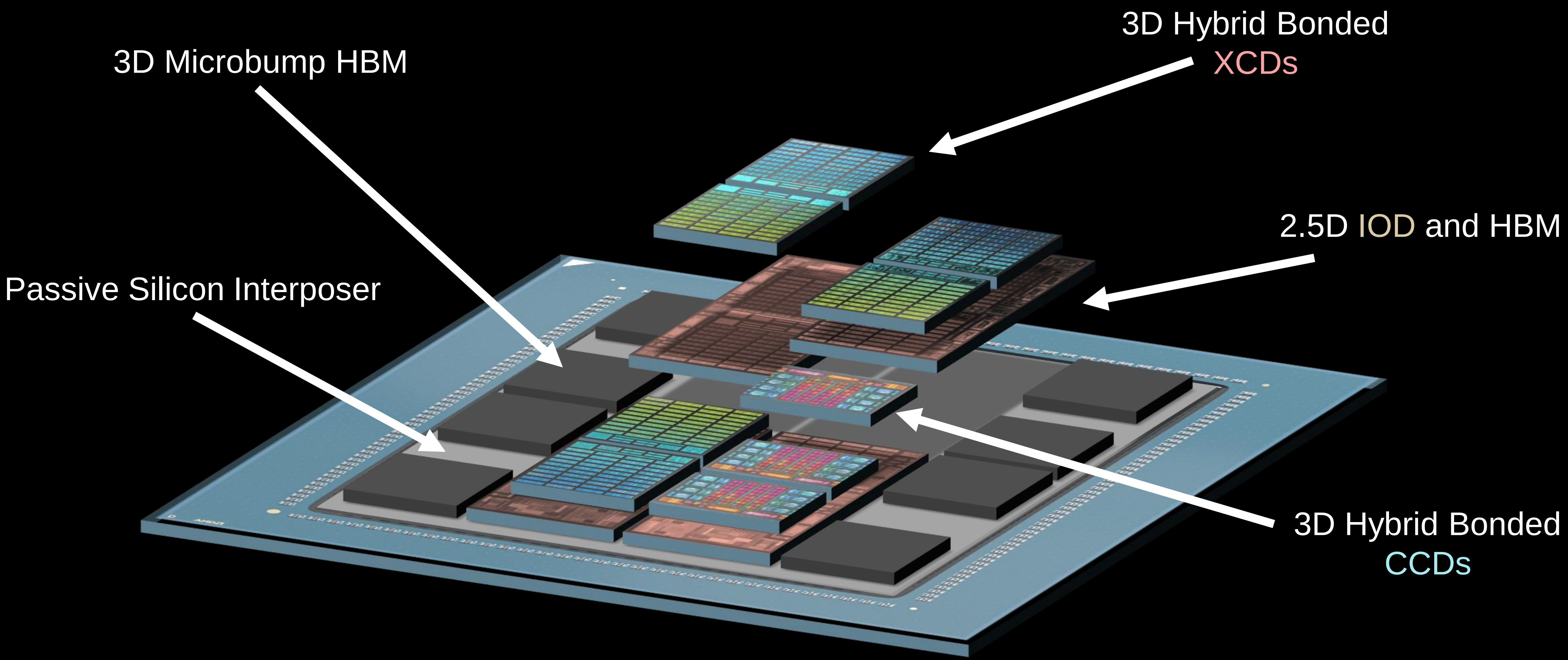


- IF – Infinity Fabric
- Fully-coherent fabric (CPU+GPU)
- Provides I/O connectivity
  - Four x16 links to other MI300A (IF)
  - Four x16 links IF or PCIe® gen5
  - Each link at 64 GB/s/dir

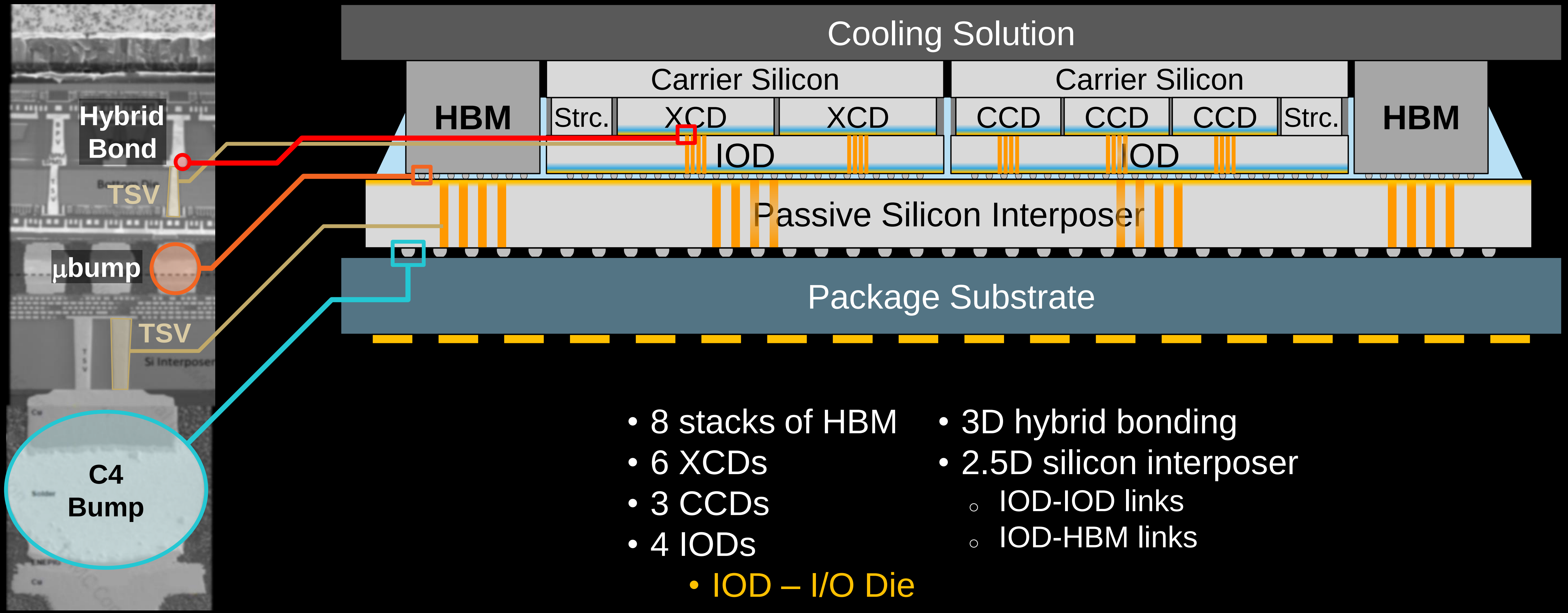
AMD Instinct™ MI300A Accelerated Processing Unit



AMD Instinct™ MI300A Accelerated Processing Unit

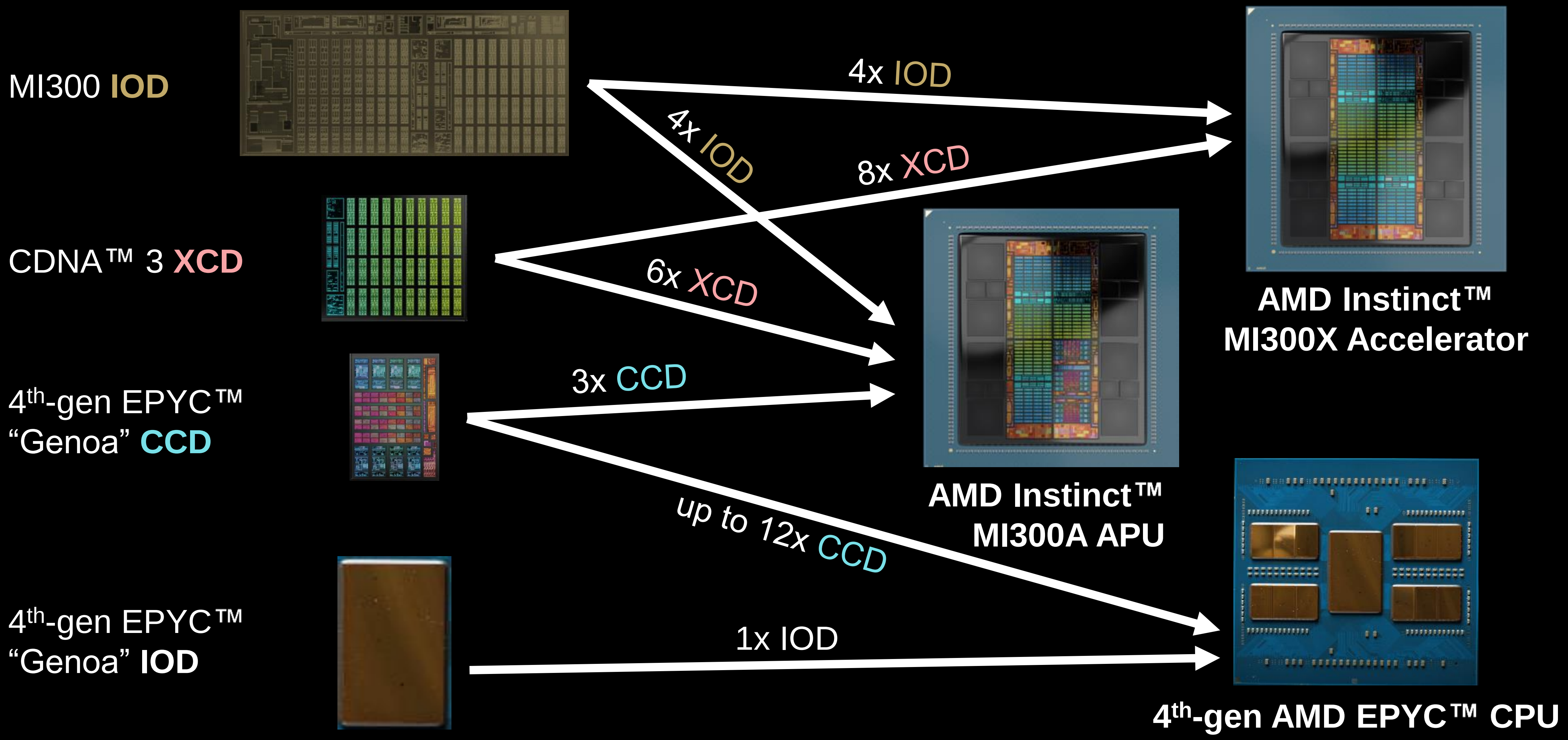


Advanced Heterogeneous Integration Packaging

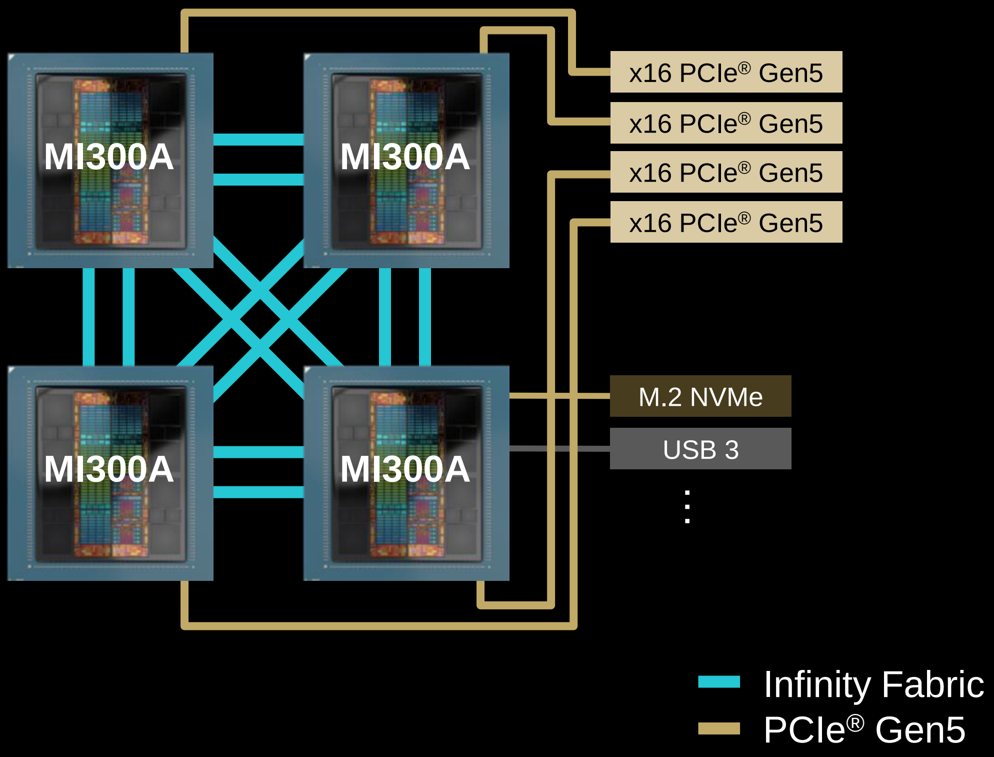


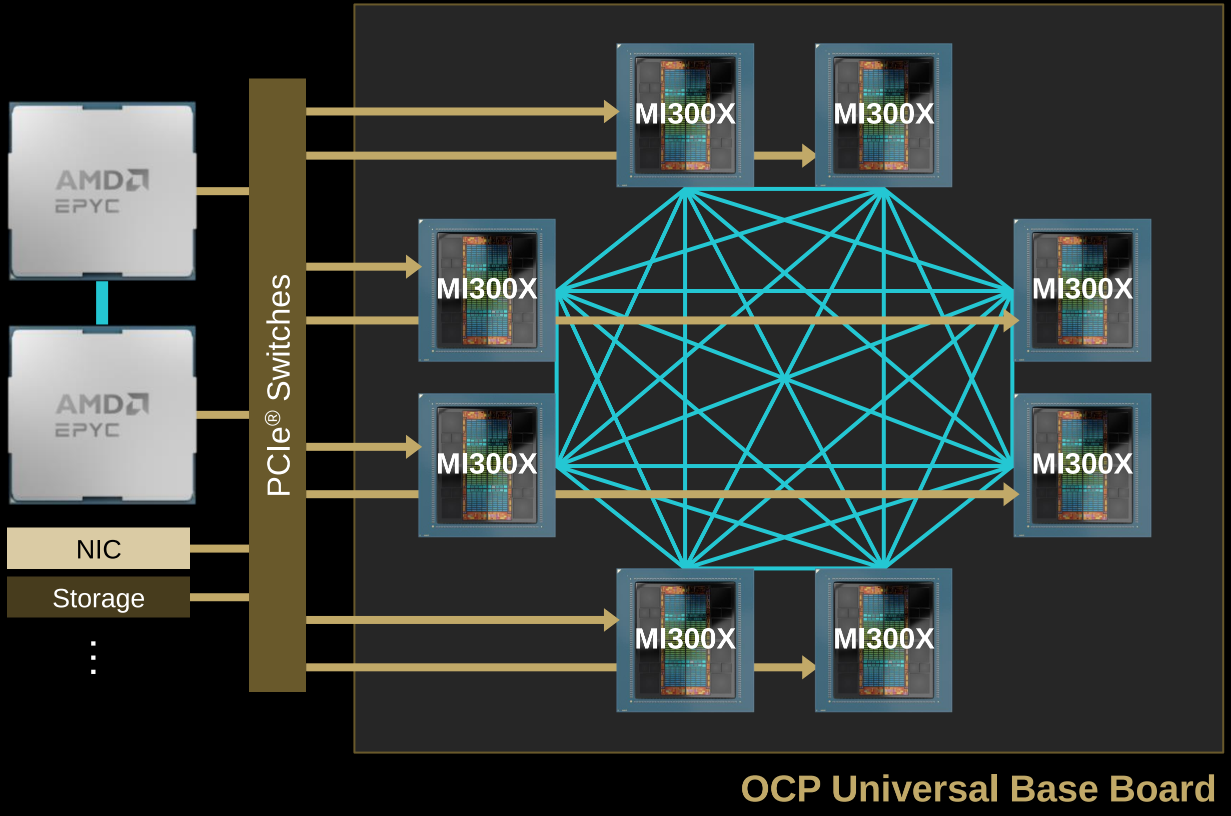
- 8 stacks of HBM
- 6 XCDs
- 3 CCDs
- 4 IODs
  - IOD – I/O Die
- 3D hybrid bonding
- 2.5D silicon interposer
  - IOD-IOD links
  - IOD-HBM links

Modular Design/Chiplet Reuse



Modular I/O Provides Scale-Up Flexibility





AMD
together we advance_

Generational Uplift

		AMD Instinct MI250X (Up to)	AMD Instinct MI300A (Up to)	AMD Instinct MI300X (Up to)
Hardware Specs	Memory Capacity	128 GB HBM2e	128 GB HBM3	192 GB HBM3
	Memory Bandwidth (Peak Theoretical)	~3.2 TB/s	~5.3 TB/s	~5.3 TB/s
	Scale-Out (Back-end) Network Bandwidth	200 Gb/s Ethernet/IB	400 Gb/s Ethernet/IB	400 Gb/s Ethernet/IB
	Max TDP/TBP	560 W	760 W	760 W
	Heterogeneous Integration	2D IF, 2.5D EFB, 3D µbump HBM	2.5D passive silicon interposer, 3D HB active interposer and multiple chiplets, 3D µbump HBM	
HPC Peak Perf. (Peak)	FP64 Vector (TFLOPS)	47.9	61.3	81.7
	FP32 Vector (TFLOPS)	47.9	122.6	163.4
	FP64 Matrix (TFLOPS)	95.7	122.6	163.4
	FP32 Matrix (TFLOPS)	95.7	122.6	163.4
AI Peak Perf. (Peak)	TF32* [TF32 Sparsity] (Matrix)	Not Supported	490.3 [980.6]	653.7 [1307.4]
	FP16 [FP16 Sparsity] (TFLOPS)	383.0 [Not Supported]	980.6 [1961.2]	1307.4 [2614.9]
	BFLOAT16 [BF16 Sparsity] (TFLOPS)	383.0 [Not Supported]	980.6 [1961.2]	1307.4 [2614.9]
	FP8* [FP8 Sparsity] (TFLOPS)	Not Supported	1961.2 [3922.3]	2614.9 [5229.8]
	INT8 [INT8 Sparsity] (TOPS)	383.0 [Not Supported]	1961.2 [3922.3]	2614.9 [5229.8]

together we advance_

MI300A Reference for Details:

- Smith, A., Loh, G.H., Schulte, M.J., Ignatowski, M., Naffziger, S., Mantor, M., Kalyanasundharam, M.F., Alla, V., Malaya, N., Greathouse, J.L., Chapman, E., & Swaminathan, R. (2024).
- Realizing the AMD Exascale Heterogeneous Processor Vision : Industry Product. 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), 876-889.
- https://drive.google.com/file/d/1J9tLeVbFRtarzIzkVrULiEBsRsEMNmEO

Progress in Shader ISA: CDNA^{TM} 3 in MI300A/X (1)

- CDNA^{TM} 3 instruction set continues to reduce support for graphics-related features and add new computing capabilities:
  - For example, for the CDNA^{TM} 3 generation all of the image and sampling instructions have been removed
  - At the same time, several new memory and compute-related features have been added
- Major enhancement is the improvement of matrix operations with introduction new data formats to the group of matrix fused-multiply-add (MFMA)
- It is a pivotal shift towards supporting low-precision multiplication, operations commonly found in neural network models
  - These instructions, while operating on single-precision floating-point numbers, can substantially lower both latency and energy consumption
- See for details "AMD Instinct^{TM} MI300" Instruction Set Architecture Reference Guide" available online (posted July 15, 2024)

Progress in Shader ISA: CDNA™ 3 in MI300A/X (2)

- For high performance AI applications, the matrix cores now extend support to 8-bit floating-point numbers
  - Including BF8 (with a 5-bit exponent and 2-bit mantissa) and FP8 (featuring a 4-bit exponent and 3-bit mantissa)
- CDNA™ 3 ISA also introduces support for **sparse matrix operations** through the innovative SMFMA instructions, broadening the scope for efficiency and performance in specialized computational tasks

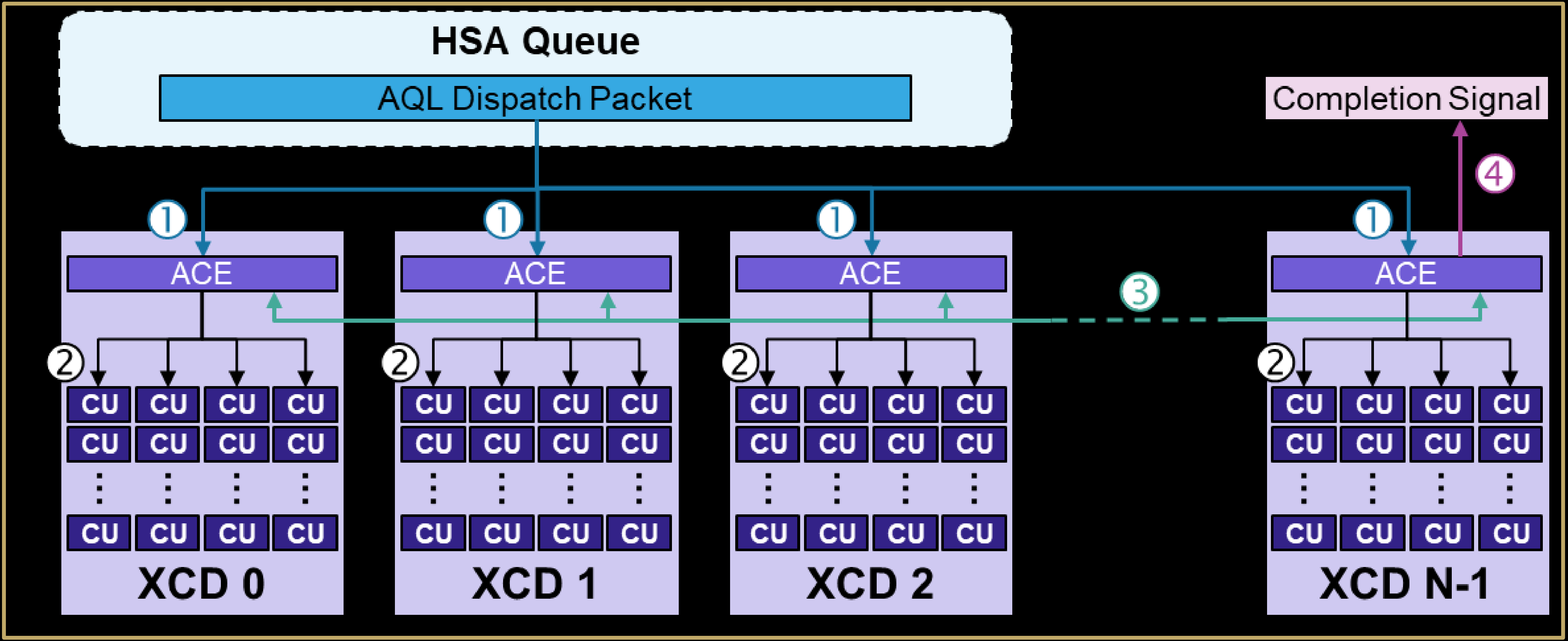
Peak operations-per-clock-per-CU rates for the CDNA™ 2 CUs in MI250X versus the CDNA™ 3 CUs in MI300A

	Vector		**Matrix**						
	FP64	FP32	FP64	FP32	TF32	FP16	BF16	FP/BF8	INT8
CDNA 2	128	128	256	256	n/a	1024	1024	n/a	1024
CDNA 3	**128**	**256**	**256**	**256**	**1024**	**2048**	**2048**	**4096**	**4096**

Challenges in Performance Profiling with ROCProfiler-SDK

- API Tracing – combining data from multiple XCDs into a single trace
- Counter profiling – redesigning the API to support multiple dimensions (XCDs)
- PC sampling – MI300 is the first AMD GPU with hardware support for PC sampling
  - Still Work-in-Progress

HSA AQL Queues and Cooperative Protocol for MI300



Impact on ROCProfiler: AQL Packet Queues

- AQL packet queue in MI300A/X vs. PM4 in previous product
- Packets submitted to these queues follow HSA's Architected Queueing Language (AQL) format
  - PM4 packet formats describe what values to put into which hardware registers to cause a particular GPU to launch a kernel
  - In contrast, AQL packets describe a higher-level goal such as "launch kernel X with Y workgroups, each with Z threads"
- Even though physically there are 6 GPU cores on MI300A, logically only one GPU is exposed to HSA runtime
- It is achieved via cooperative protocol used by ACEs parsing AQL queues
  - ACE – Asynchronous Compute Engine

Impact on ROCProfiler: Cooperative Protocol

- Cooperative protocol allows the partition to take a single kernel dispatch from a single software-visible queue and spread the workgroups across the XCDs
  - It makes the partition appear as a single logical GPU despite its multi-chiplet nature
    - This is a natural pairing to MI300A's unified memory, where the entire HBM memory space is visible to a single process
- Cooperative protocol only works for AQL packets (and very few PM4 packets)
  - ROCProfiler (PMC mode) uses PM4 packets to collect counters from all 6 XCCs directly
  - XCD => XCC – Accelerator Complex Core (similar to GPU core definition)

Impact on ROCProfiler: XCC Counter Support (1)

- XCC counters are the counters located on XCD chiplet
- Three ways to access XCC-based counters:
  - Local access (within the same XCC)
  - Intra-IOD remote access (within the same IOD)
  - Inter-IOD remote access (across IOD access using SMN address)
    - SMN - System Management Network

- ROCProfiler uses local access because all 6 XCCs see the same queue
  - Special AQL PRED_EXEC packet is inserted to tell ACE on different XCCs if the upcoming PM4 packets are to be processed on that XCC

Impact on ROCProfiler: XCC Counter Support (2)

- XCC-based counters report could be generated in two modes:
- Accumulative mode:
  - We handle all XCC-based counters the same ways as how we handle SE-based counters (SE=Shader Engine)
  - The counter values will be accumulated across all XCCs
- Individual mode:
  - No accumulation across XCCs
  - 6 sets of counters will be reported as if they are from 6 different physical GPUs cores (matching XCD chiplets)

Impact on ROCProfiler: IOD Counter Support

- IOD counters are the counters generated on IOD die [such as UMC counters, DF (data fabric) counters, etc.]
- Access IOD-based counters in two ways:
  - Local access (Using XCC ACE to access local IOD)
  - Remote access (across IOD access using SMN address)
    - SMN - System Management Network
- ROCProfiler always uses remote access with assigned one master-XCC to access all IOD-based counters across other IOD dies
  - In APUs like MI300A, IOD0 counters cannot be accessed locally by XCC ACE (there is no XCC on IOD0 which has only compute CCDs)

PC Sampling – Informal Definition

- In CPU world, PC sampling works by taking periodic call stack snapshots of the applications to identify which functions are currently executing
  - Over time, a pattern emerges as certain functions being executed more frequently than others and consumed more resources
- In GPU, PC sampling periodically choose an active wave (in a round robin manner), snapshot its state
  - At least program counter (PC) and the current timestamp
  - Other call stack status information if available
- The process takes place on every compute unit simultaneously
  - Which makes it device-wide across all shader engines
- Generates a histogram of samples that statistically approximates kernel execution

PC Sampling on MI2xx (1)

- To better understand PC sampling on MI300, let’s step back to the MI2xx
- The first beta implementation delivered in ROCm 6.2 (08/02/2024)
- Host-Trap Method used
  - KFD (amdgpu driver) periodically sends trap commands over PCIe
    - Sampling period in micro-seconds
- A trapped wave executes a handler code to snapshot its state
  - PC, exec_mask, CU id, timestamp, wave_id, workgroup_id_{x, y, z}, correlation ID (more information in next slides)

PC Sampling on MI2xx (2)

- Multiple processes can use PC sampling on the same device simultaneously
  - Nvidia CUPTI supports only one process per GPU
- Each sample is tagged with correlation ID (internal + external)
  - Distinguishing between multiple instances of the same kernel
    - Based on request from HPCToolkit folks to attribute kernel costs depending on the launching (invocation) context
- Limitations:
  - Limited by the PCIe throughput
  - No information about whether the wave is issuing a command or stalling

PC Sample Generated by The Host-Trap Method

```
typedef struct
{
    uint64_t                           size;   // Size of this struct
    rocprofiler_pc_sampling_header_v1_t flags; // what fields of this struct are relevant
    uint8_t                            chiplet;   // chiplet ID
    uint8_t                            wave_id;   // wave identifier within the workgroup
~~    uint8_t                            wave_issued : 1;~~
    uint8_t                            reserved    : 7;   // reserved 7 bits, must be zero
    uint32_t                           hw_id;   // compute unit identifier
    uint64_t                           pc;      // sampled PC
    uint64_t                           exec_mask;     // active SIMD lanes
    rocprofiler_dim3_t                 workgroup_id; // wave coordinates within the
                                                     // workgroup
~~    uint32_t                           wave_count;   // number of active waves at the CU~~
    uint64_t                           timestamp;     // timestamp when sample is generated
    rocprofiler_correlation_id_t       correlation_id; // internal + external CID
~~    rocprofiler_pc_sampling_snapshot_v1_t snapshot;   // whether wave is issuing,~~
~~                                                      // if not, what's the stall reason~~
    uint32_t reserved2;  // for future use
} rocprofiler_pc_sampling_record_t

// More information available at https://github.com/ROCm/rocprofiler-sdk/blob/amd-staging/source/include/rocprofiler-sdk/pc_sampling.h
```

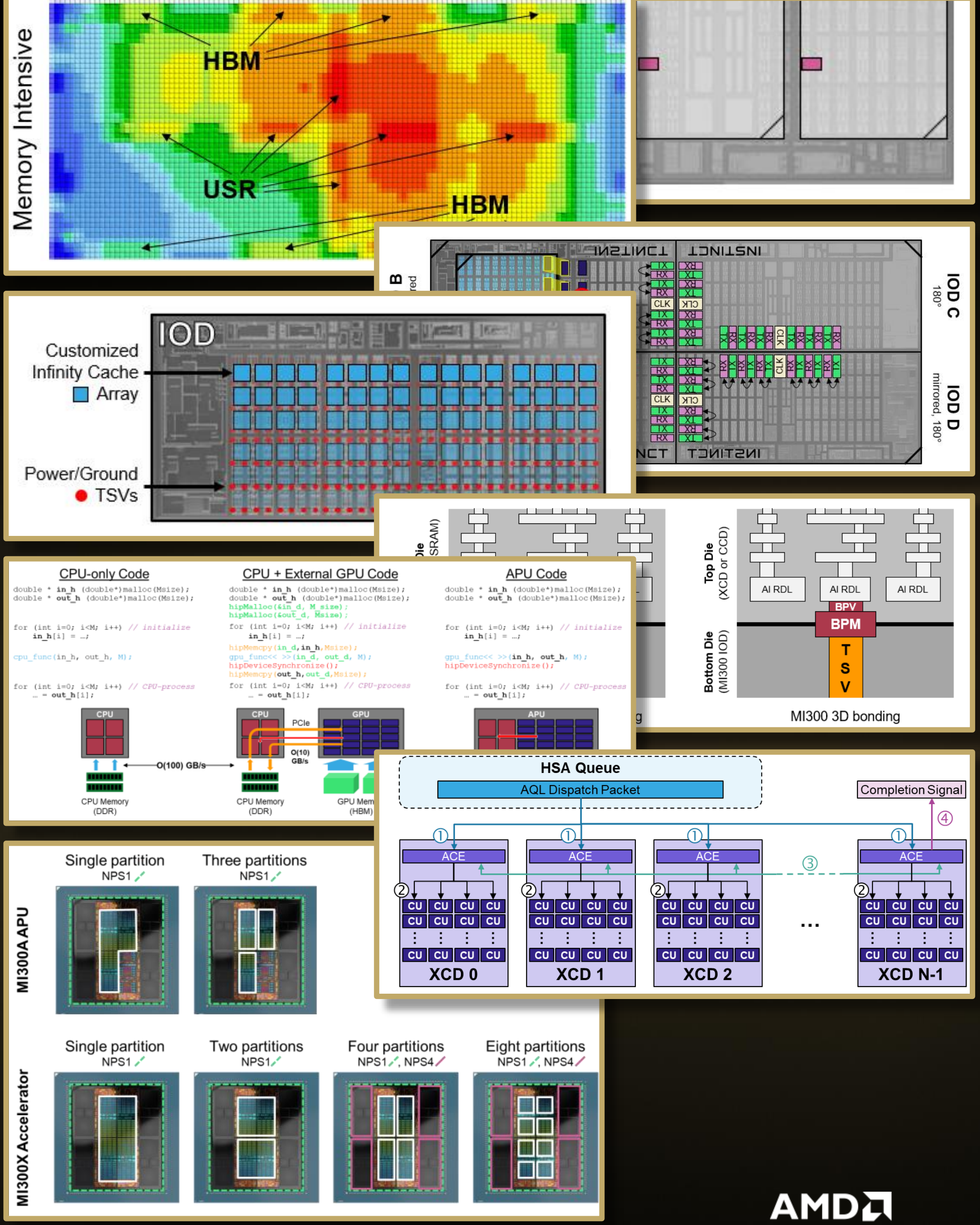
Stochastic PC sampling Method on MI300

- To improve on the MI200 PC sampling mechanism, we designed a dedicated HW component on each CU responsible to periodically trap waves
  - Sampling period: 256 – 2^20 cycles
- Information about whether a wave is issuing command
  - If not, what’s the stall reason
  - Delivered via dedicated GPU register with appropriate encoding

```
struct{
    . . .
    uint8_t                             wave_issued : 1;
    uint32_t                            wave_count;  //  number of active waves at the CU
    rocprofiler_pc_sampling_snapshot_v1_t snapshot {
        uint32_t dual_issue_valu    : 1;
        uint32_t inst_type          : 4;
        uint32_t reason_not_issued  : 7;
        uint32_t arb_state_issue    : 10;
        uint32_t arb_state_stall    : 10;
    }
    . . .
} rocprofiler_pc_sampling_record_t
```

Summary

- HPC and AI domains continue to scale up a demand for performance, perf-per-Watt, and perf-per-mm^3
- It leads to significant co-design efforts between architecture, advanced packaging, physical design, software, and many other teams in AMD
  - Also, between AMD, DOE, TSMC, and other partners
- Complex multi-die heterogeneous architecture and programming models require significant efforts on instrumentation tools development
- Special hardware features should be added on architecture level to support telemetry and system monitoring with minimal performance overhead



CPU-only Code
```
double * in_h (double*)malloc(Msize);
double * out_h (double*)malloc(Msize);

for (int i=0; i<M; i++) // initialize
    in_h[i] = …;

cpu_func(in_h, out_h, M);

for (int i=0; i<M; i++) // CPU-process
    … = out_h[i];
```

CPU + External GPU Code
```
double * in_h (double*)malloc(Msize);
double * out_h (double*)malloc(Msize);
hipMalloc(&in_d, M_size);
hipMalloc(&out_d, Msize);
for (int i=0; i<M; i++) // initialize
    in_h[i] = …;
hipMemcpy(in_d,in_h,Msize);
gpu_func<< >>(in_d, out_d, M);
hipDeviceSynchronize();
hipMemcpy(out_h,out_d,Msize);
for (int i=0; i<M; i++) // CPU-process
    … = out_h[i];
```

APU Code
```
double * in_h (double*)malloc(Msize);
double * out_h (double*)malloc(Msize);

for (int i=0; i<M; i++) // initialize
    in_h[i] = …;

gpu_func<< >>(in_h, out_h, M);
hipDeviceSynchronize();

for (int i=0; i<M; i++) // CPU-process
    … = out_h[i];
```

Copyright and Disclaimer

Thanks a lot for for your attention!

Contacts:

Timour.Paltashev@amd.com – Any problem

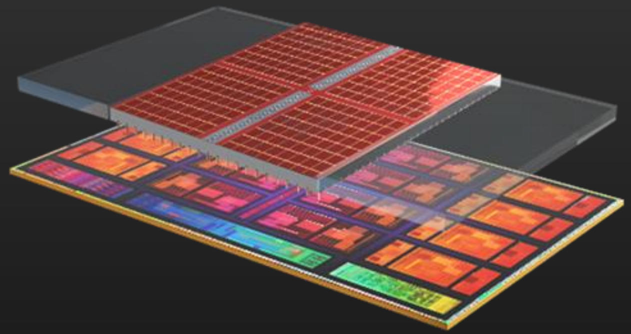
Vladimir.Indic@amd.com – AMD GPU PC Sampling

Nursultan.Kabylkas@amd.com – AMD GPU XML ISA Spec and API

AMD

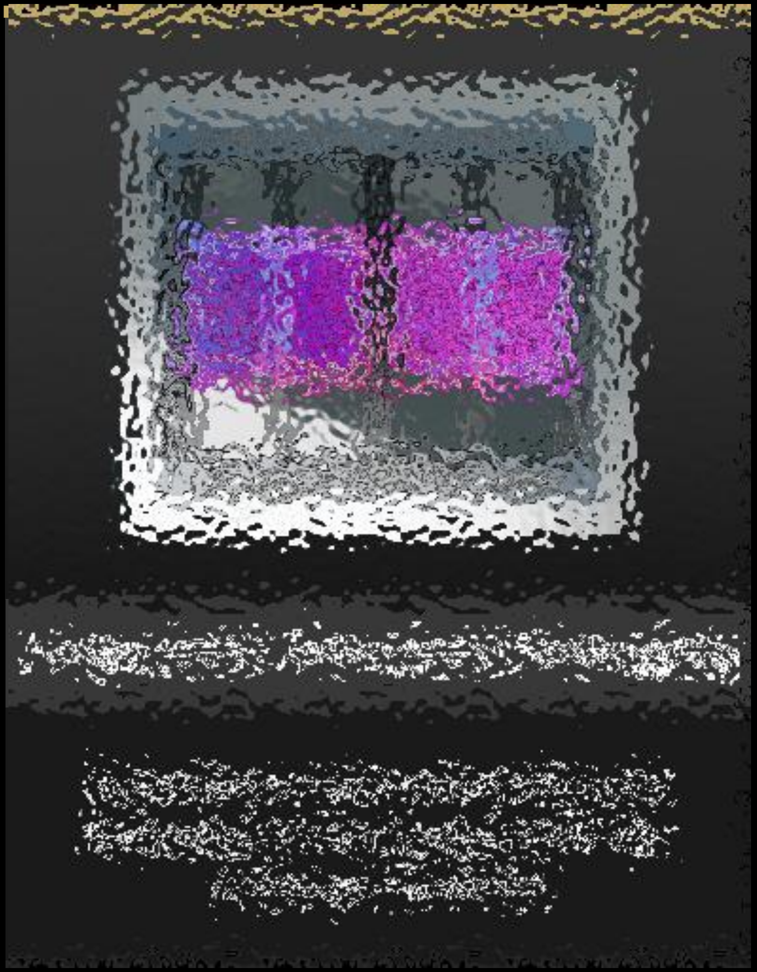
Backup slides

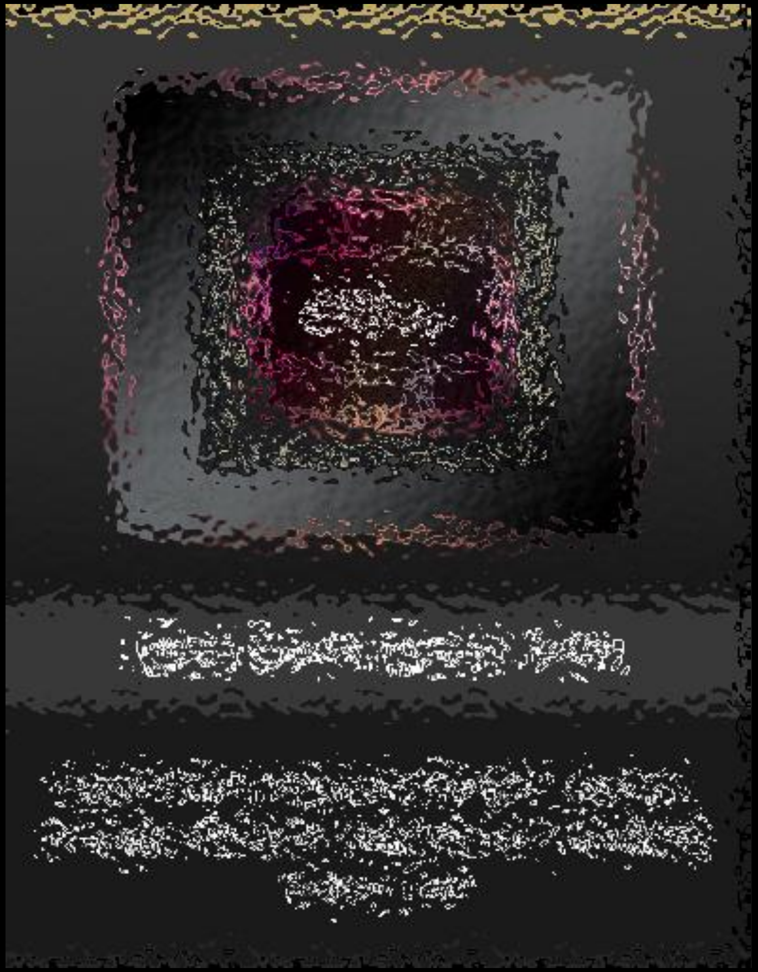
Key Changes Between Exascale Generations



Hybrid Bonding 3D

Leverage learnings with multiple generations of V-Cache™ implementations in EPYC™ and Ryzen™ products





AMD together we advance_

Impact on ROCProfiler: MMIO vs. PM4

- ROCProfiler in PMC mode uses PM4 to access counters from user mode queue
  - While most of other tools via use MMIO KFD to access registers/counters
- When accessing counters using MMIO, the counter address is from the view of the host (CPU)
- When accessing counters using PM4, the counter address is from the local view of GPU
  - Register spec lists different addresses for different views