AMD

The EPYC™ CPU and INSTINCT™ MI250X GPUs in Frontier

Frontier Training Workshop

February 2023

AMD Public

Frontier at a Glance

- Achieved 1.102 EF on HPL: First to Exascale!

Initial Performance Results

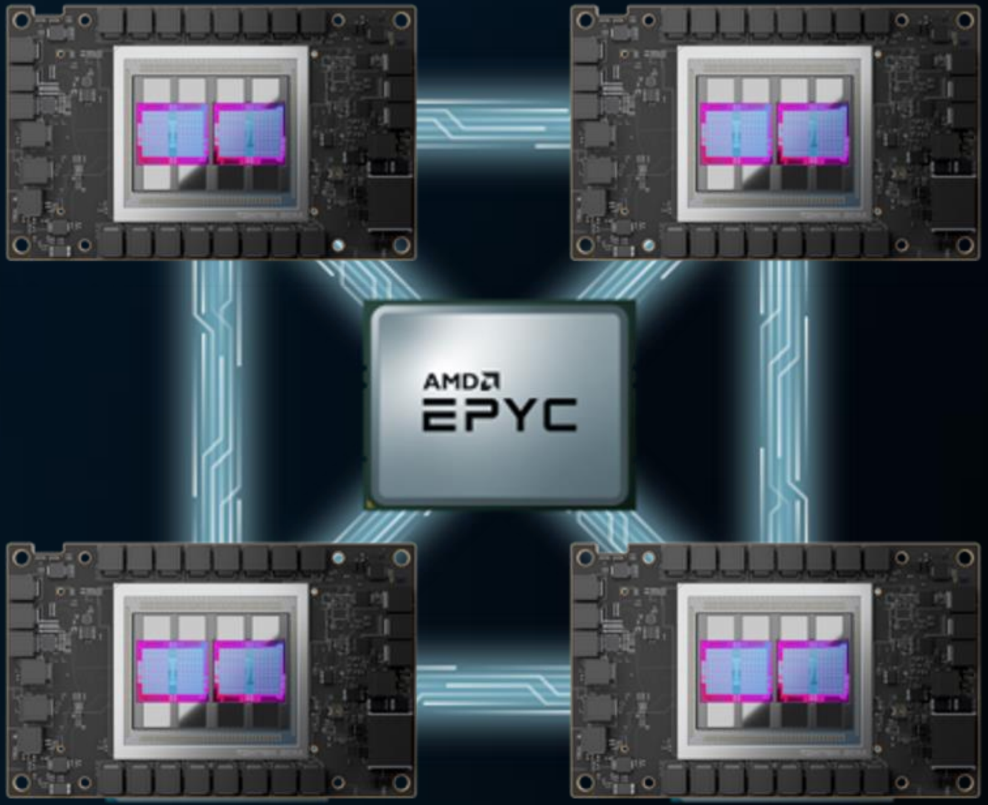
- HPL-AI/HPL-MxP exceeded 7.942 EF reduced-precision
- SNAPSHOT: first exaflop graph AI application
- ACM Gordon Bell Prize at SC22: WarpX

CRAY
OAK RIDGE National Laboratory
U.S. DEPARTMENT OF ENERGY
Hewlett Packard Enterprise
AMD
FRONTIER

ACM GORDON BELL PRIZE
presented by John West (ACM)
Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-In-Cell Simulations on Exascale-Class Supercomputers
University of Paris-Saclay, Lawrence Berkeley National Laboratory, ARM,
RIKEN

Frontier Node at a Glance

- 1x Optimized 3rd Gen AMD EPYC™ CPU (64 core)
- 4x AMD Instinct™ MI250X accelerators
    - Direct Attached to the NIC
- Coherent connectivity
    - Via AMD Infinity Fabric™ interconnect
    - Tightly integrated
    - Unified memory space



	EPYC™ CPU	4x MI250X GPUs	Ratio
Memory Bandwidth	200 GB/s	4 x 3.2 TB/s = 12.8 TB/s	64x
Compute Bandwidth	2 TFLOPs	4 x 53 TFLOPs = 212 TFLOPs	106x

<1% of the FLOPs on Frontier are from the CPUs!

3rd Gen AMD EPYC™ PROCESSORS AT A GLANCE

COMPUTE

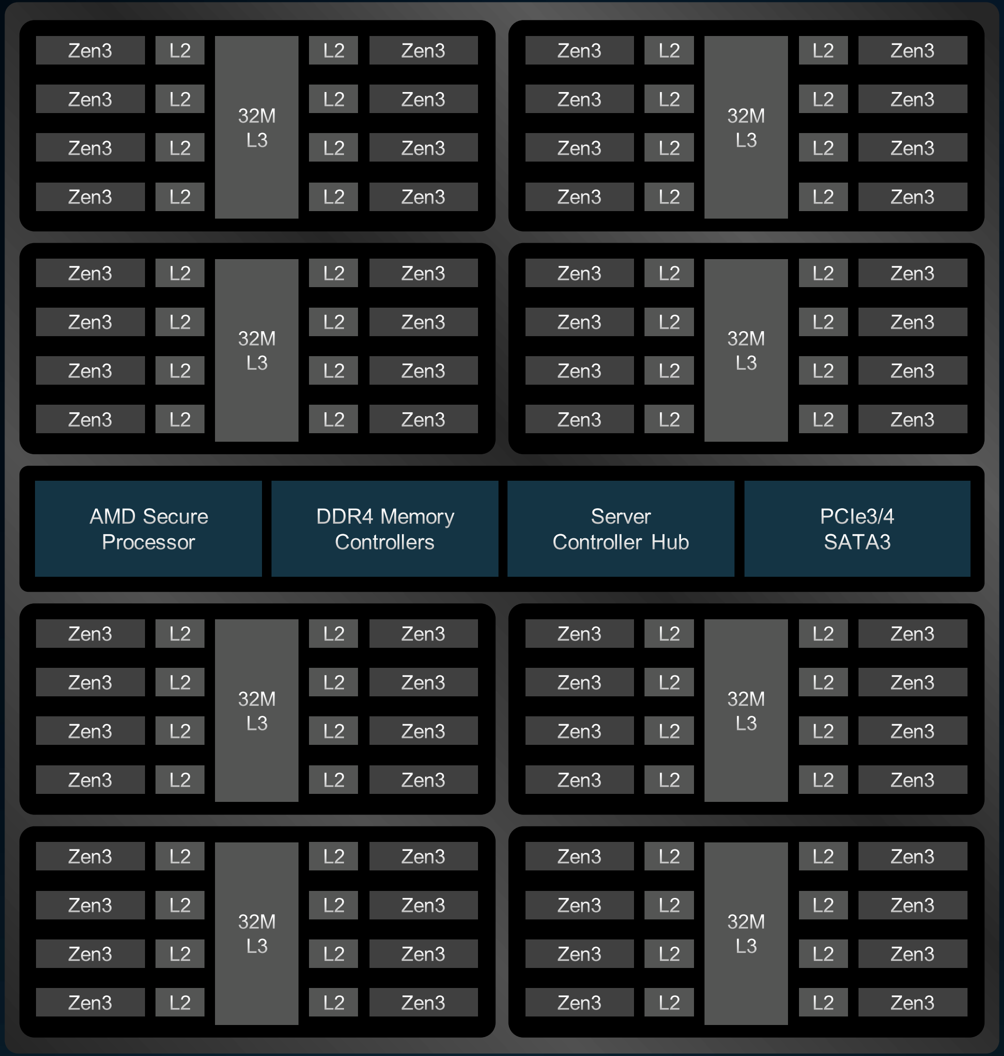
- AMD "Zen3" x86 cores (64 core / 128 threads)
- Up to 32MB L3 cache / core, shared by each chiplet
- Flatter NUMA domain, reduced latency w/ smaller system diameter
- TDP range: 120W-280W

MEMORY

- 8 channel DDR4 with ECC up to 3200 MHz
  Option for 6 channel Memory Interleaving[1]
- RDIMM, LRDIMM, 3DS, NVDIMM-N
- 2 DIMMs/channel capacity of 4TB/socket (256GB DIMMs)

PERFORMANCE

- +Increased socket performance, single threaded performance, performance per core*
- Infinity Fabric™ Gen 2 (xGMI-2)

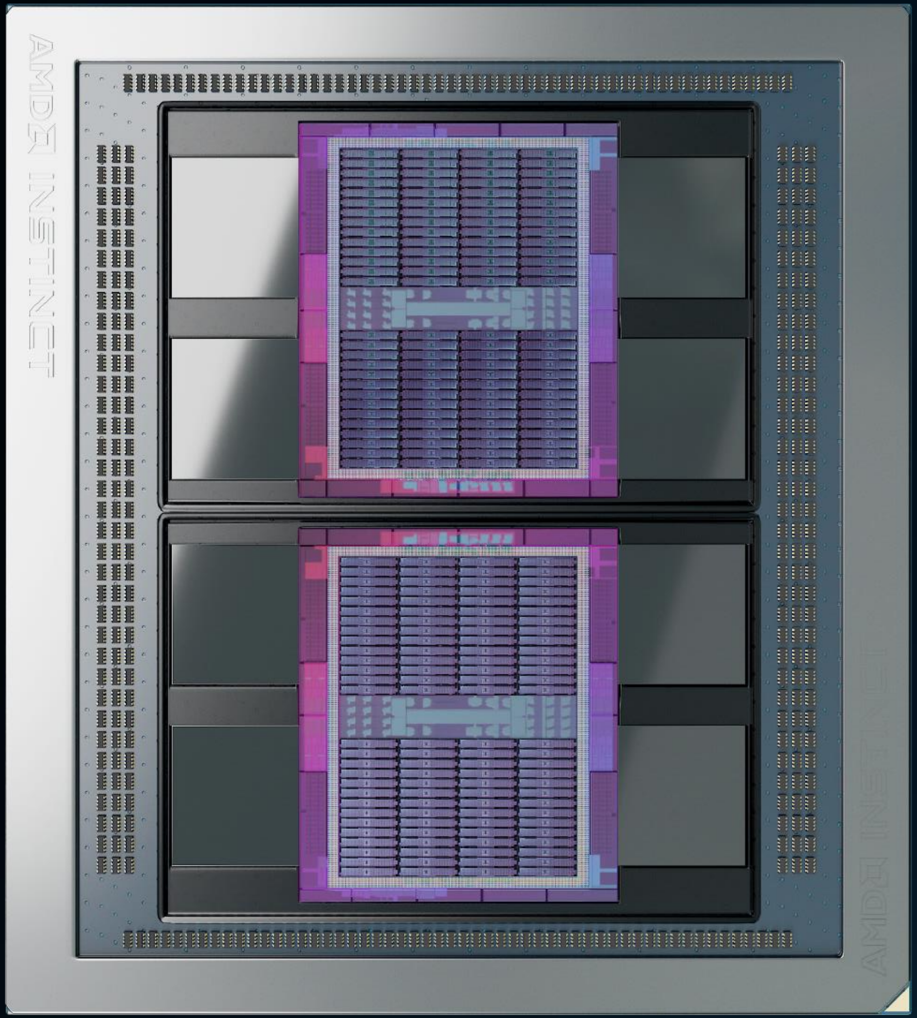


INTEGRATED I/O – NO CHIPSET

- 128 lanes PCIe™ Gen3/4
  - Used for PCIe, SATA, and Coherent Interconnect
  - Up to 32 SATA or NVMe™ direct connect devices
  - 162 lane option (2P config)
- Server Controller Hub
  (USB, UART, SPI, LPC, I2C, etc.)

SECURITY

- Dedicated Security Subsystem
- Secure Boot, Hardware Root-of-Trust
- SME (Secure Memory Encryption)
- SEV-ES (Secure Encrypted Virtualization & Register Encryption
- SNP (Secure Nested Paging)



AMD INSTINCT™ MI250X

WORLD'S MOST ADVANCED DATA CENTER ACCELERATOR

58B
Transistors in 6nm

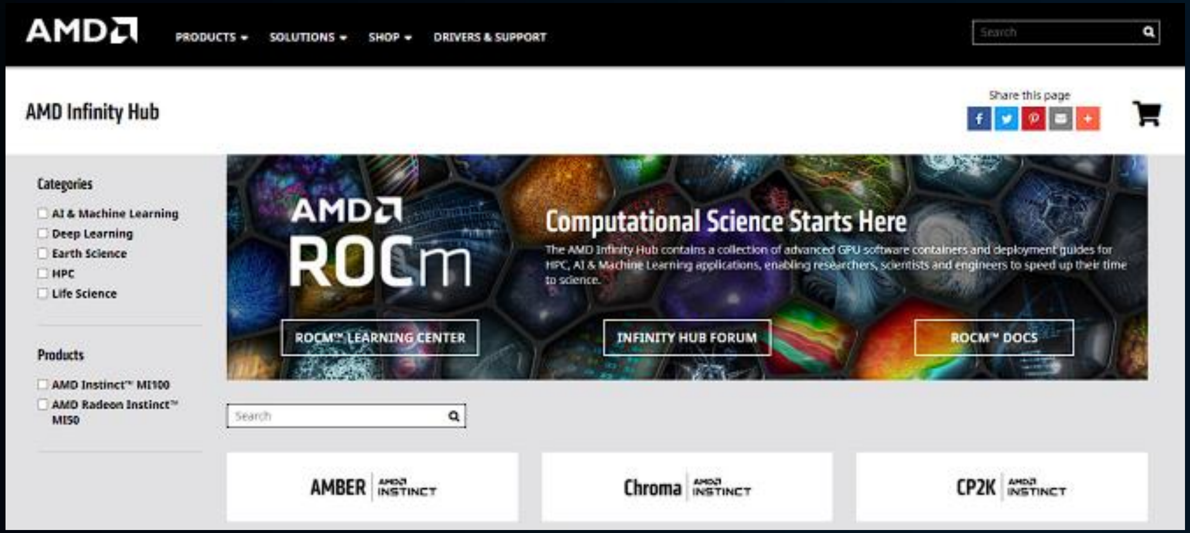
220
Compute Units

880
2nd Gen Matrix Cores

128
GB HBM2E @ 3.2 TB/s

CDNA2 White Paper

https://www.amd.com/system/files/documents/amd-cdna2-white-paper.pdf



AMD Infinity Hub

Categories

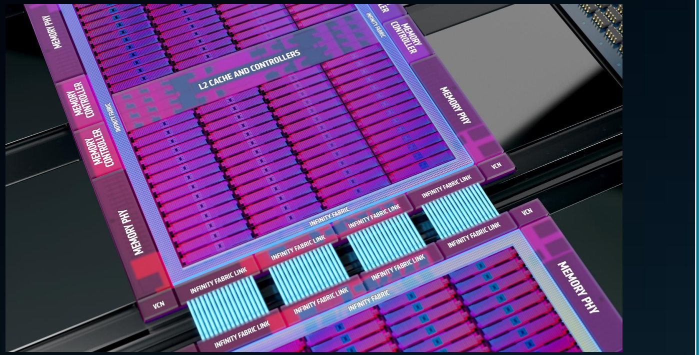
- □ AI & Machine Learning
- □ Deep Learning
- □ Earth Science
- □ HPC
- □ Life Science

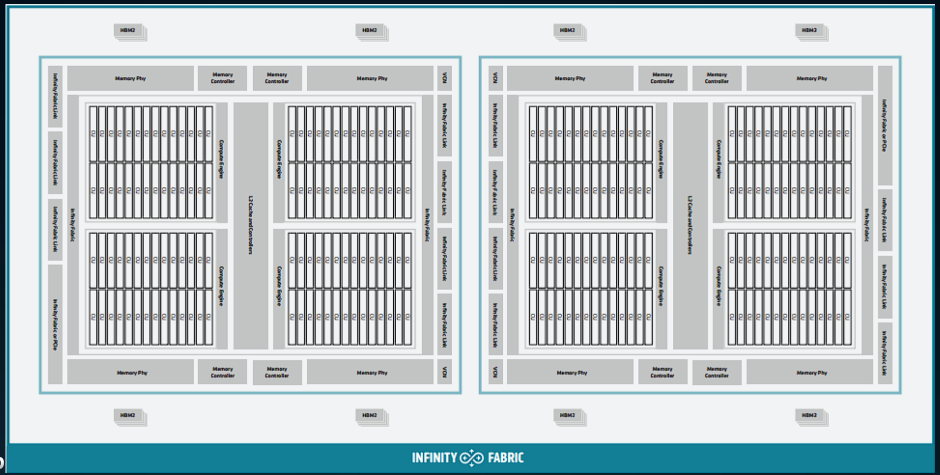
Products

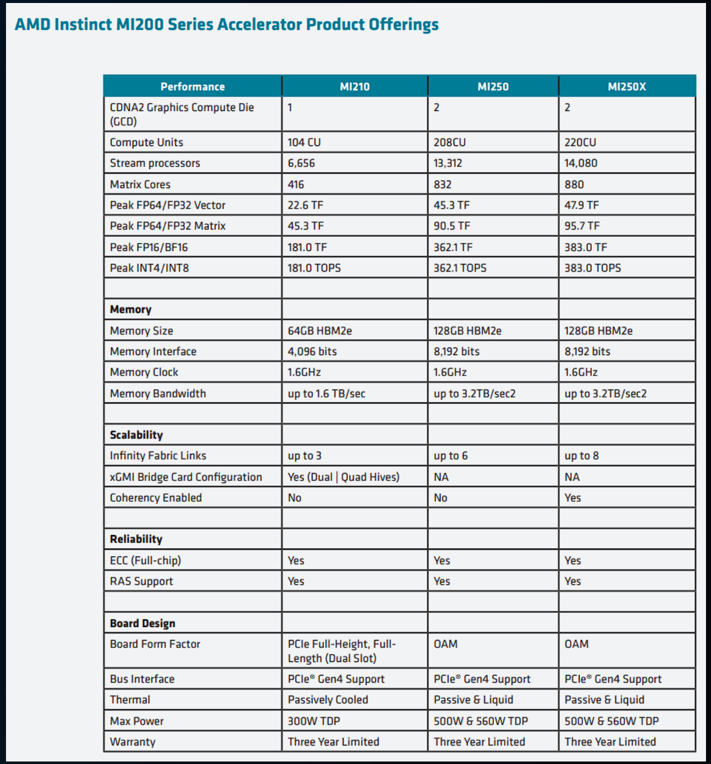
- □ AMD Instinct™ MI100
- □ AMD Radeon Instinct™ MI50

Computational Science Starts Here

The AMD Infinity Hub contains a collection of advanced GPU software containers and deployment guides for HPC, AI & Machine Learning applications, enabling researchers, scientists and engineers to speed up their time to science.

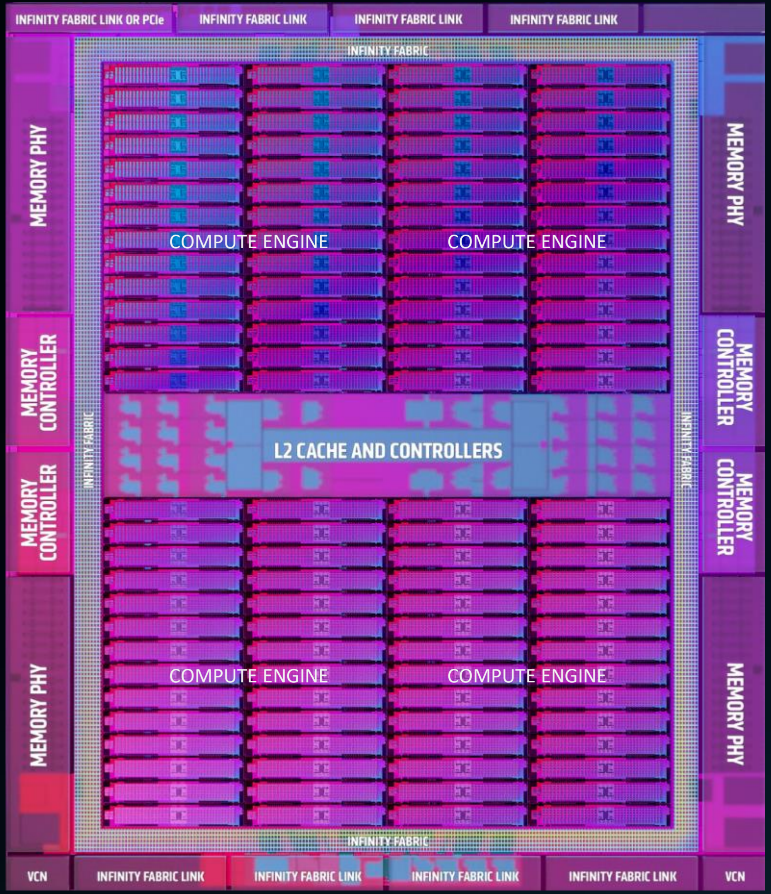






AMD Instinct MI200 Series Accelerator Product Offerings

Performance	MI210	MI250	MI250X
CDNA2 Graphics Compute Die (GCD)	1	2	2
Compute Units	104 CU	208CU	220CU
Stream processors	6,656	13,312	14,080
Matrix Cores	416	832	880
Peak FP64/FP32 Vector	22.6 TF	45.3 TF	47.9 TF
Peak FP64/FP32 Matrix	45.3 TF	90.5 TF	95.7 TF
Peak FP16/BF16	181.0 TF	362.1 TF	383.0 TF
Peak INT4/INT8	181.0 TOPS	362.1 TOPS	383.0 TOPS
Memory			
Memory Size	64GB HBM2e	128GB HBM2e	128GB HBM2e
Memory Interface	4,096 bits	8,192 bits	8,192 bits
Memory Clock	1.6GHz	1.6GHz	1.6GHz
Memory Bandwidth	up to 1.6 TB/sec	up to 3.2TB/sec2	up to 3.2TB/sec2
Scalability			
Infinity Fabric Links	up to 3	up to 6	up to 8
xGMI Bridge Card Configuration	Yes (Dual | Quad Hives)	NA	NA
Coherency Enabled	No	No	Yes
Reliability			
ECC (Full-chip)	Yes	Yes	Yes
RAS Support	Yes	Yes	Yes
Board Design			
Board Form Factor	PCIe Full-Height, Full-Length (Dual Slot)	OAM	OAM
Bus Interface	PCIe® Gen4 Support	PCIe® Gen4 Support	PCIe® Gen4 Support
Thermal	Passively Cooled	Passive & Liquid	Passive & Liquid
Max Power	300W TDP	500W & 560W TDP	500W & 560W TDP
Warranty	Three Year Limited	Three Year Limited	Three Year Limited



2ND GENERATION CDNA ARCHITECTURE

BUILT FOR HPC & AI

TSMC 6NM TECHNOLOGY

UP TO 110 CU PER GRAPHICS CORE DIE

4 MATRIX CORES PER COMPUTE UNIT

MATRIX CORES ENHANCED FOR HPC

8 INFINITY FABRIC LINKS PER DIE

SPECIAL FP32 OPS FOR DOUBLE THROUGHPUT

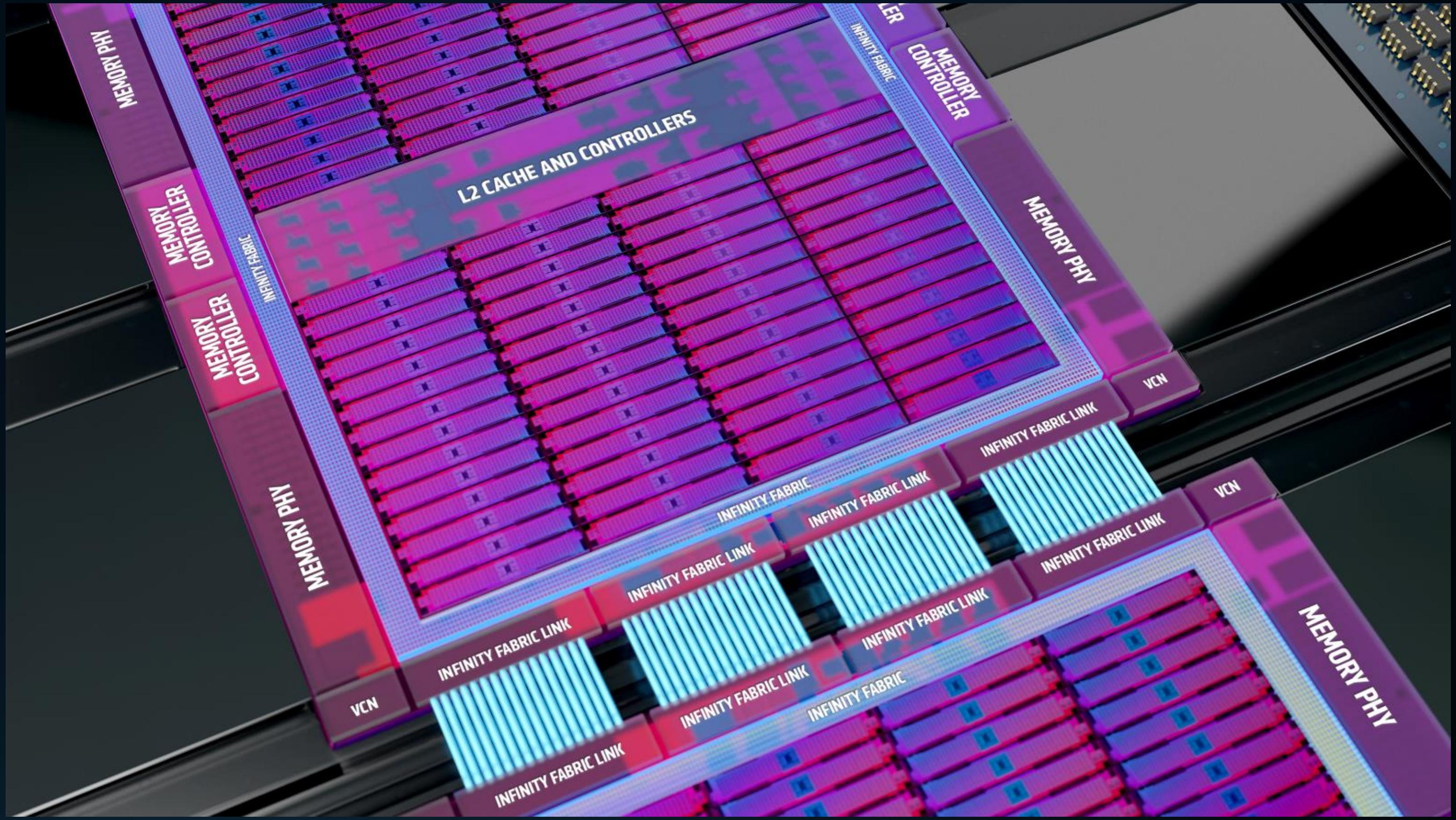
MULTI-CHIP DESIGN

TWO GPU DIES IN PACKAGE TO MAXIMIZE COMPUTE & DATA THROUGHPUT

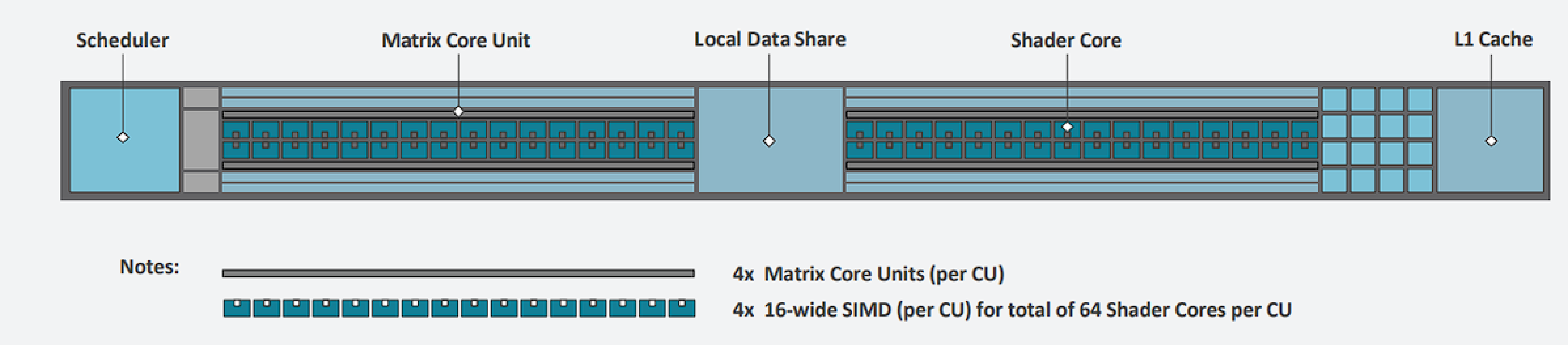
INFINITY FABRIC FOR CROSS-DIE CONNECTIVITY

4 LINKS RUNNING AT 25GBPS

400GB/S OF BI-DIRECTIONALBANDWIDTH



Inside a Compute Unit (CU)

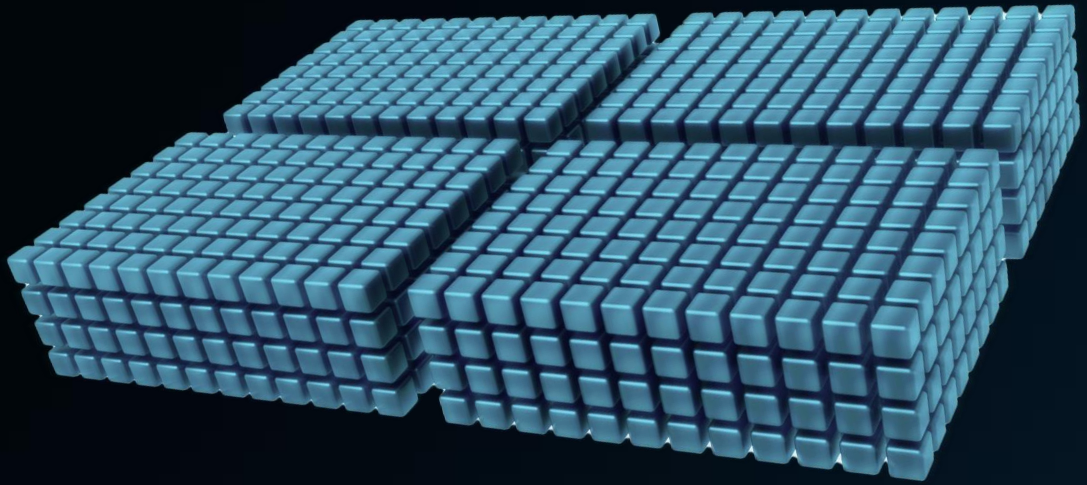

Scheduler
Matrix Core Unit
Local Data Share
Shader Core
L1 Cache
Notes:
4x Matrix Core Units (per CU)
4x 16-wide SIMD (per CU) for total of 64 Shader Cores per CU

- Scheduler
  - Buffer for up to 40 wavefronts – 2560 work-items (parallel threads)
  - At each clock, waves on 1 SIMD unit are considered for execution (via round robin)
- 4x Matrix Core Units per CU
  - 110 CUs per GCD, 880 Matrix Cores per GCD
- 64 KB Local Data Share (LDS, or shared memory)
- 4x SIMD Vector units (each 16 lanes wide): 64 Shader Cores per CU
  - Each 16 lane SIMD unit supports half, single, and double precisions

2nd GENERATION MATRIX CORES

OPTIMIZED COMPUTE UNITS FOR MATRIX OPERATIONS

DOUBLE PRECISON (FP64)
MATRIX CORE THROUGHPUT
REPRESENTATION

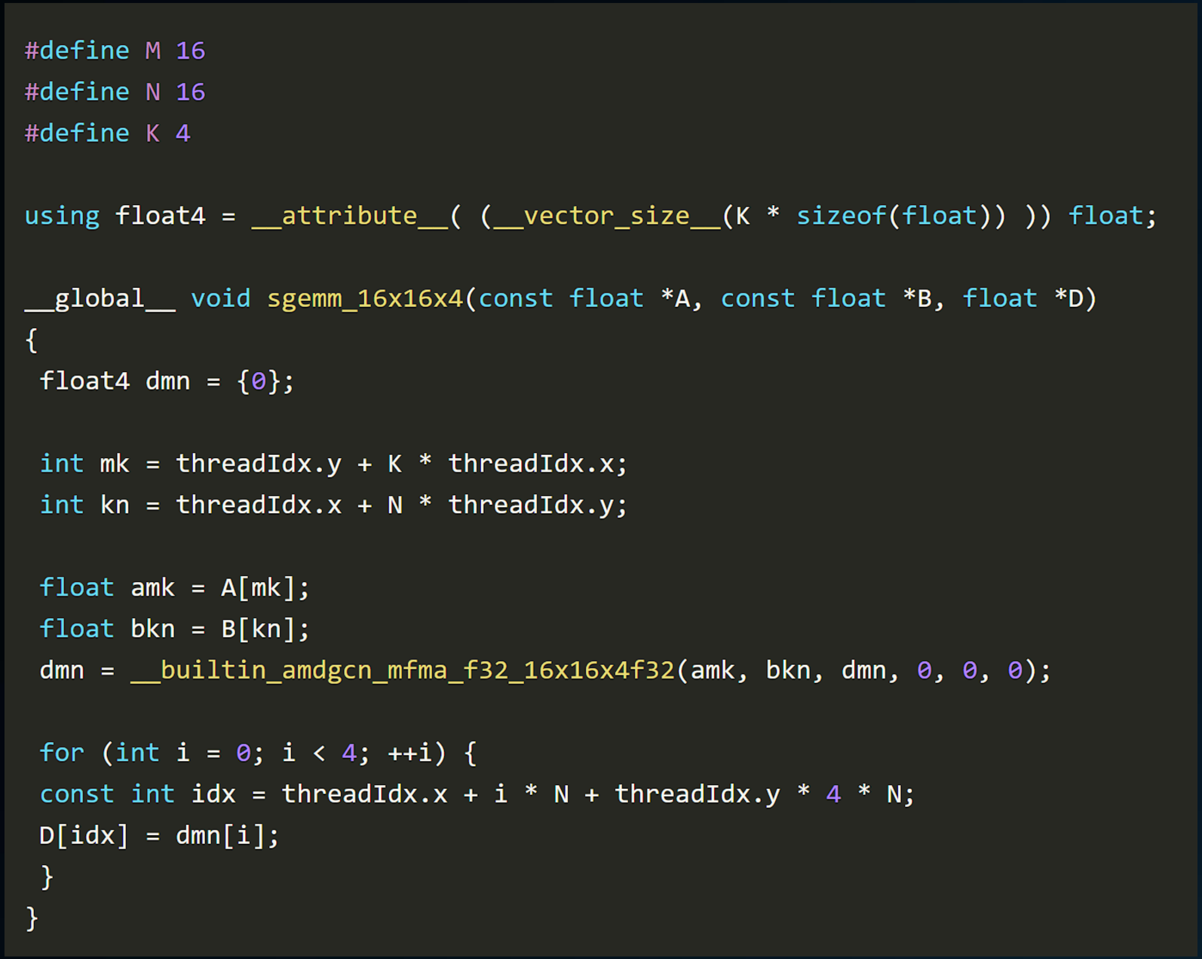


MI100 MATRIX CORES OPS/CLOCK/COMPUTE UNIT	MI250X MATRIX CORES OPS/CLOCK/COMPUTE UNIT
No FP64 Matrix Core	256 FP64
256 FP32	256 FP32
1024 FP16	1024 FP16
512 BF16	1024 BF16
512 INT8	1024 INT8

https://developer.amd.com/wp-content/resources/CDNA2_Shader_ISA_18November2021.pdf

2nd GENERATION MATRIX CORES

OPTIMIZED COMPUTE UNITS FOR MATRIX OPERATIONS



```
#define M 16
#define N 16
#define K 4

using float4 = __attribute__( (__vector_size__(K * sizeof(float)) )) float;

__global__ void sgemm_16x16x4(const float *A, const float *B, float *D)
{
 float4 dmn = {0};

 int mk = threadIdx.y + K * threadIdx.x;
 int kn = threadIdx.x + N * threadIdx.y;

 float amk = A[mk];
 float bkn = B[kn];
 dmn = __builtin_amdgcn_mfma_f32_16x16x4f32(amk, bkn, dmn, 0, 0, 0);

 for (int i = 0; i < 4; ++i) {
 const int idx = threadIdx.x + i * N + threadIdx.y * 4 * N;
 D[idx] = dmn[i];
 }
}
```

- Current support for using MFMA instructions:
    - AMD libraries: rocBLAS
    - AMD's rocWMMA library
    - LLVM builtin compiler intrinsic functions
    - Inline assembly

https://gpuopen.com/learn/amd-lab-notes/amd-lab-notes-matrix-cores-readme/

https://github.com/ROCmSoftwarePlatform/rocWMMA

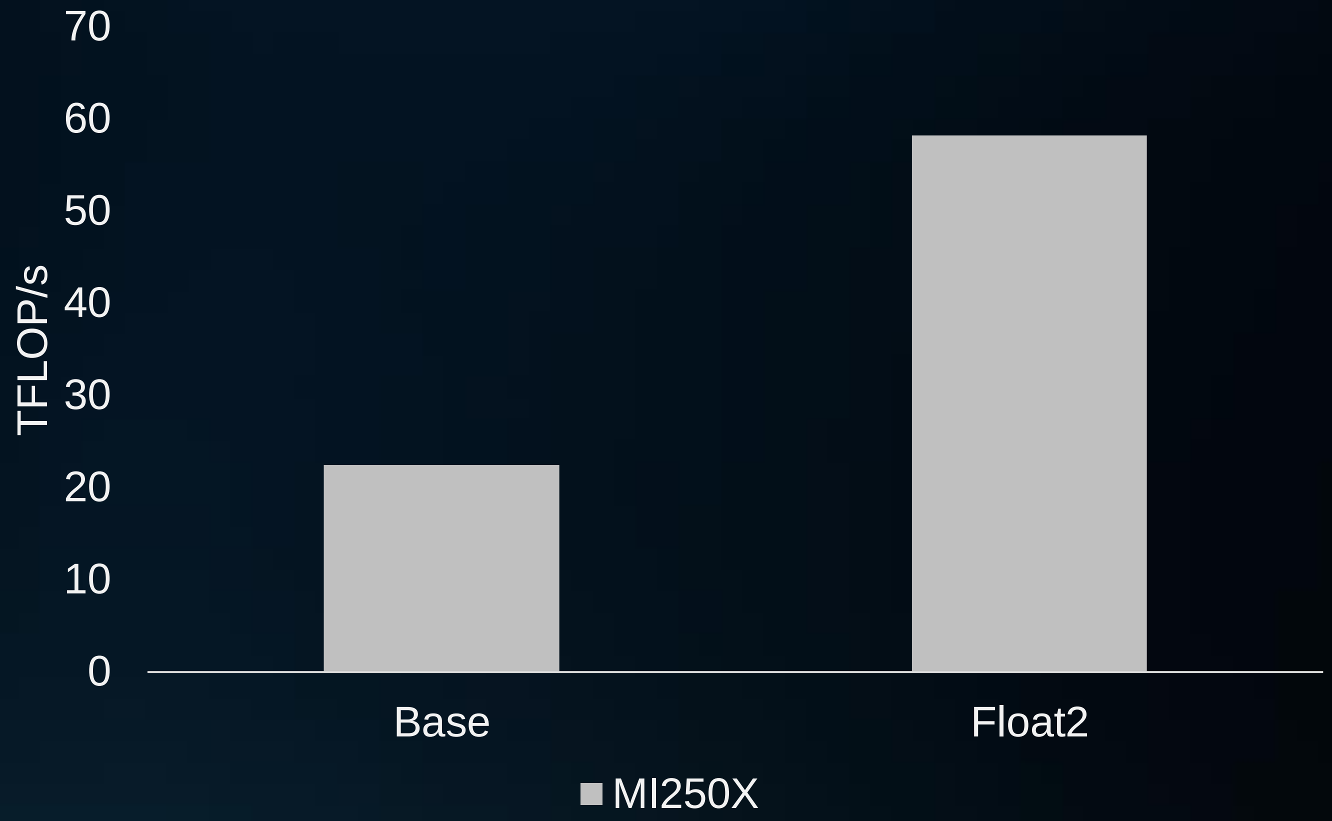
NEW IN AMD INSTINCT MI250X

PACKED FP32

FP64 PATH USED TO EXECUTE TWO COMPONENT VECTOR INSTRUCTIONS ON FP32

DOUBLES FP32 THROUGHPUT PER CLOCK PER COMPUTE UNIT

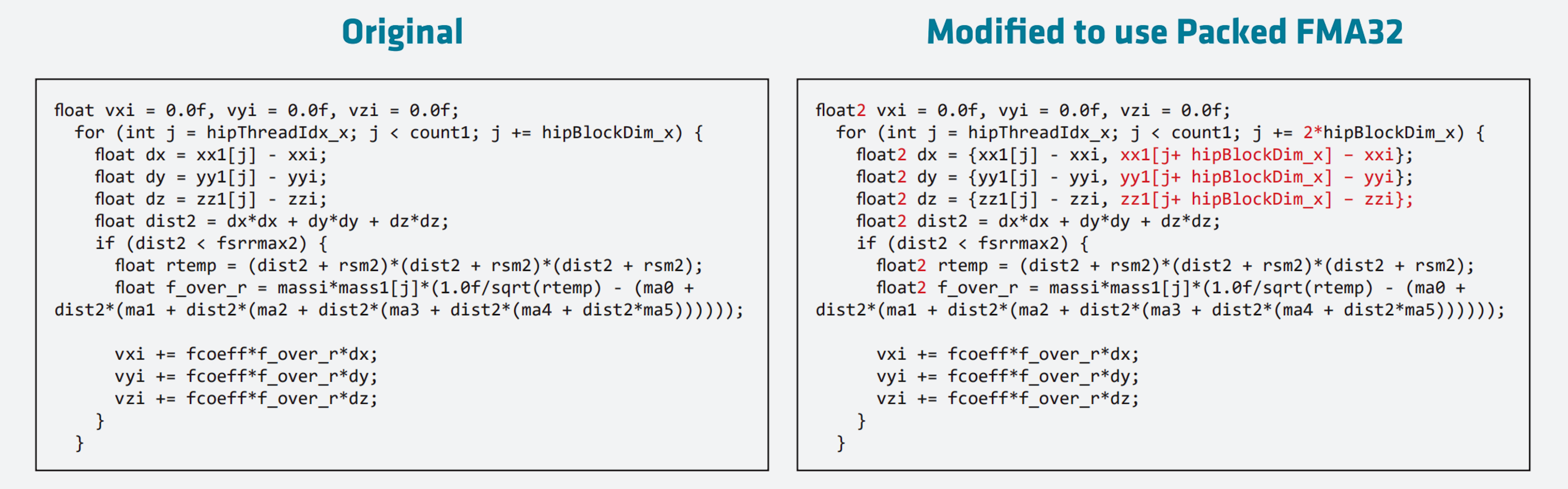
pk_FMA, pk_ADD, pk_MUL, pk_MOV operations



https://www.amd.com/en/technologies/infinity-hub/mini-hacc

NEW IN AMD INSTINCT MI250X

Refactoring code to emit PACKED FP32 instructions



Original

```
float vxi = 0.0f, vyi = 0.0f, vzi = 0.0f;
  for (int j = hipThreadIdx_x; j < count1; j += hipBlockDim_x) {
    float dx = xx1[j] - xxi;
    float dy = yy1[j] - yyi;
    float dz = zz1[j] - zzi;
    float dist2 = dx*dx + dy*dy + dz*dz;
    if (dist2 < fsrrmax2) {
      float rtemp = (dist2 + rsm2)*(dist2 + rsm2)*(dist2 + rsm2);
      float f_over_r = massi*mass1[j]*(1.0f/sqrt(rtemp) - (ma0 +
dist2*(ma1 + dist2*(ma2 + dist2*(ma3 + dist2*(ma4 + dist2*ma5))))));

      vxi += fcoeff*f_over_r*dx;
      vyi += fcoeff*f_over_r*dy;
      vzi += fcoeff*f_over_r*dz;
    }
  }
```

Modified to use Packed FMA32

```
float2 vxi = 0.0f, vyi = 0.0f, vzi = 0.0f;
  for (int j = hipThreadIdx_x; j < count1; j += 2*hipBlockDim_x) {
    float2 dx = {xx1[j] - xxi, xx1[j+ hipBlockDim_x] – xxi};
    float2 dy = {yy1[j] - yyi, yy1[j+ hipBlockDim_x] – yyi};
    float2 dz = {zz1[j] - zzi, zz1[j+ hipBlockDim_x] – zzi};
    float2 dist2 = dx*dx + dy*dy + dz*dz;
    if (dist2 < fsrrmax2) {
      float2 rtemp = (dist2 + rsm2)*(dist2 + rsm2)*(dist2 + rsm2);
      float2 f_over_r = massi*mass1[j]*(1.0f/sqrt(rtemp) - (ma0 +
dist2*(ma1 + dist2*(ma2 + dist2*(ma3 + dist2*(ma4 + dist2*ma5))))));

      vxi += fcoeff*f_over_r*dx;
      vyi += fcoeff*f_over_r*dy;
      vzi += fcoeff*f_over_r*dz;
    }
  }
```

https://www.amd.com/en/technologies/infinity-hub/mini-hacc

Conclusions and Developer Guidance

- Move work to the GPU
- Launch network messages from GPU-resident buffers
- Use vendor provided libraries whenever possible, particularly for dense linear algebra

Q&A

Disclaimer:

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.” AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD