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
**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

<table>
<thead>
<tr>
<th></th>
<th>EPYC™ CPU</th>
<th>4x MI250X GPUs</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Bandwidth</td>
<td>200 GB/s</td>
<td>4 x 3.2 TB/s = 12.8 TB/s</td>
<td>64x</td>
</tr>
<tr>
<td>Compute Bandwidth</td>
<td>2 TFLOPs</td>
<td>4 x 53 TFLOPs = 212 TFLOPs</td>
<td>106x</td>
</tr>
</tbody>
</table>

<1% of the FLOPs on Frontier are from the CPUs!
3rd Gen AMD EPYC™ PROCESSORS AT A GLANCE

COMPUTE
- "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
- 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**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transistors in 6nm</td>
<td>58B</td>
</tr>
<tr>
<td>Compute Units</td>
<td>220</td>
</tr>
<tr>
<td>2nd Gen Matrix Cores</td>
<td>880</td>
</tr>
<tr>
<td>GB HBM2E @ 3.2 TB/s</td>
<td>128</td>
</tr>
</tbody>
</table>
CDNA2 White Paper

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-DIRECTIONAL BANDWIDTH
Inside a Compute Unit (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 Precision (FP64) Matrix Core Throughput Representation**

## MI100 Matrix Cores

<table>
<thead>
<tr>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>No FP64 Matrix Core</td>
<td></td>
</tr>
<tr>
<td>256 FP32</td>
<td></td>
</tr>
<tr>
<td>1024 FP16</td>
<td></td>
</tr>
<tr>
<td>512 BF16</td>
<td></td>
</tr>
<tr>
<td>512 INT8</td>
<td></td>
</tr>
</tbody>
</table>

## MI250X Matrix Cores

<table>
<thead>
<tr>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 FP64</td>
<td></td>
</tr>
<tr>
<td>256 FP32</td>
<td></td>
</tr>
<tr>
<td>1024 FP16</td>
<td></td>
</tr>
<tr>
<td>1024 BF16</td>
<td></td>
</tr>
<tr>
<td>1024 INT8</td>
<td></td>
</tr>
</tbody>
</table>

[Source](https://developer.amd.com/wp-content/resources/CDNA2_ShaderISA_18November2021.pdf)
2nd GENERATION MATRIX CORES
OPTIMIZED COMPUTE UNITS FOR MATRIX OPERATIONS

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

```c
#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(ambient, 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];
    }
}
```

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_MOVER operations

NEW IN AMD INSTINCT MI250X
Refactoring code to emit PACKED FP32 instructions

Original

```c
float vx1 = 0.0f, vy1 = 0.0f, vz1 = 0.0f;
for (int j = hipThreadIdx.x; j < count1; j += hipBlockDim_x) {
    float dx = xx1[j] - xx1;
    float dy = yy1[j] - yy1;
    float dz = zz1[j] - zz1;
    float dist2 = dx*dx + dy*dy + dz*dz;
    if (dist2 < fSrrMax2) {
        float rtemp = (dist2 + rsm2)*(dist2 + rsm2)*(dist2 + rsm2);
        float f_over_r = mass1[mass1[j]]*(1.0f/sqrt(rtemp) - (ma0 +
            dist2*(ma1 + dist2*(ma3 + dist2*(ma4 + dist2*ma5)))));
        vx1 += fcoeff*f_over_r*dx;
        vy1 += fcoeff*f_over_r*dy;
        vz1 += fcoeff*f_over_r*dz;
    }
}
```

Modified to use Packed FMA32

```c
float2 vx1 = 0.0f, vy1 = 0.0f, vz1 = 0.0f;
for (int j = hipThreadIdx.x; j < count1; j += 2*hipBlockDim_x) {
    float2 dx = {xx1[j] - xx1, xx1[j+ hipBlockDim_x] - xx1};
    float2 dy = {yy1[j] - yy1, yy1[j+ hipBlockDim_x] - yy1};
    float2 dz = {zz1[j] - zz1, zz1[j+ hipBlockDim_x] - zz1};
    float2 dist2 = dx*dx + dy*dy + dz*dz;
    if (dist2 < fSrrMax2) {
        float2 rtemp = (dist2 + rsm2)*(dist2 + rsm2)*(dist2 + rsm2);
        float2 f_over_r = mass1[mass1[j]]*(1.0f/sqrt(rtemp) - (ma0 +
            dist2*(ma1 + dist2*(ma3 + dist2*(ma4 + dist2*ma5)))));
        vx1 += fcoeff*f_over_r*dx;
        vy1 += fcoeff*f_over_r*dy;
        vz1 += fcoeff*f_over_r*dz;
    }
}
```

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
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.