ECE 571 – Advanced Microprocessor-Based Design Lecture 35

Vince Weaver

https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

9 December 2024

Announcements

- Don't forget projects, presentations on (Wed and Fri)
- Final writeup due last day of exams (20th)
- Will try to get homeworks graded soon.
- Reminder: no final
- Don't forget course evals! We're at 10% currently

Project Presentations

- You will have roughly 10 minutes to present. Plan for 8 minutes describing your project and allowing 2 minutes for questions.
- Give a summary of what you did and why. Show any results you obtained. Describe any future work that needs to be done.
  - Intro/Overview – give background on the project
  - Experimental Setup
    - Hardware – some brief details on the hardware you

used

    - Software – the software you used, the operating system, the benchmarks
    - Test Setup – any measurement or tooling, what software (if any) was used for gathering info
  - Results
  - Future Work
    - What you would do if you had more time
  - Demo if applicable (videos/pictures fine too)
- You may present slides using the projector if you want, but that's not strictly necessary.

Question on LCD Types

- Various types, often proprietary, one that's out of patent is 2-(4-alkoxyphenyl)-5-alkylpyrimidine with cyanobiphenyl

Question on why no courses teach GPU design

- Short answer, I don't know why, but it definitely isn't common
- It is a lot harder to design floating point hardware from scratch than integer

Brief Review of GPU concepts

- CPU better for single-thread, low-latency workloads
- GPU better at massively parallel high-bandwidth workloads
- CPUs becoming more GPU like as GPUs becoming more CPU like

NVIDIA GPUs

Tesla	2006	90-40nm
Fermi	2010	40nm/28nm
Kepler	2012	28nm
Maxwell	2014	28nm
Pascal/Volta	2016	16nm/14nm
Turing	2018	12nm
Ampere	2020	8nm/7nm
AdaLovelace/Hopper	2022	TSMC 4N / N4 (5nm/4nm)
Blackwell	2024	TSMC 4N

- GeForce – Gaming
- Quadro (old) / Workstation
- Tesla (old) / DataCenter

Gaming GPUs

- Geforce 40 (40xx)
- Aside, 4090 goes for $\$4000, 4070 for $\$550
- RTX 4090 16,384 FP32 cores (82.6 TFLOPS)
  24G GDDR6X memory, 72M L2 cache, 450W
- RTX 4080
- CUDA Compute 8.9
- TSMC 4N (custom NVIDIA 5nm, easily confused with TSMC N4 which is different)
- Fourth-gen Tensor cores with FP8, FP16, bfloat-16

sparsity acceleration? (sparse arrays? lots of zeros?)

- Third-gen Raytracing

  Opacity Micromap Engine, Displaced Micro-mesh Engine
- Shader Execution Reordering

  When following ray, might end up diverging in behavior and not as vectorable. Allow to re-order so more similar?
- Optical Flow Acceleration (use AI to guess between-frame animations)
- FP64 performance 1/64 that of F32
- Controversy over 4080 naming
- 12VHPWR connector failure (PCIe gen 5 16-pin

connector) melting

600W vs 150W or 75W of previous connectors

Aside on Floating Point Sizes

- FP64 – standard IEEE, what HPL / supercomputer report FLOPS in
- FP32 – GPUs natively use this for graphics, so much faster as this
- FP16, FP8, FP4 – smaller, faster, used in AI
- BFLOAT16 – like FP16 but with larger exponent and smaller mantissa which matches some workloads better
- FP6 – can use FP8 paths but takes up less room in RAM

Tensor Cores

- High value, low precision
- Useful for AI

Ray Tracing Hardware

- Anything other than simple ray-casting requires recursion (Each time you hit an object) as well as random-access to the entire 3d-space
- NVIDIA RTX, RDNA 2/3
- Hybrid raytracing – traditional casting+rasterization used for visibility, raytracing for shadows
- Only continue tracing rays of surface has more than threshold of reflectability

Ray Tracing Hardware

- BVH – bounding volume hierarchy
- Items wrapped in a volume
- Collision detection and ray tracing can be faster, if the point of interest is outside the volume you can ignore it, only do detailed math if it's inside the volume
- Trees?

Brief Blackwell Background

- Named for David Blackwell, mathematician
- Successor to Hopper, 5x performance (at 4 bit fp)
- 20 PFLOPs (at 4 bit fp), 300W? B100?
- Two dies, 10TB/s NVLINK, 8 HBM3e stacks (192GB)
- Tower of them get get over an exaflop (a lot of power, and again 4 bit fp, for AI)

Homework Reading #1

https://resources.nvidia.com/en-us-blackwell-ar

“New Class of Superchip”

- Mixed in with info From “NVIDIA Blackwell Architecture” document
- Chip itself
  - 208 billion transistor, custom TSMC 4NP process
  - 2.5x transistors as Hopper
  - Two reticle-limited dies, 10TB/s chip-chip interconnect
  - Largest GPU ever built
  - 20 petaFLOPS (at what bits floating point?)

- 2nd gen Blackwell Tensor Core (for large-language model LLM)
  - 4-bit floating point (FP4) AI
  - Nemo Framework?
  - Megatron core (intentional pun, pytorch for training large-scale transformers)
- Confidential Computing (security)
  - Older GPUs often results just hand around in RAM and can be read out. Though NVIDIA seems concerned about AI IP (especially in cloud environments)
  - Trusted Execution Environment

- NVLINK – hook up to 576 GPUs together
  - Twice the bandwidth of Hopper
  - Fifth gen has two differential pairs, 50GB/s
  - Blackwell has 18 of these for 1.8TB/s total
  - 14x bandwidth of PCIe Gen5
  - 7Petabytes/hour (18 years of streaming 4k movies)
- Decompression engine
  - 18x faster than x86
  - nvlink directly to ARM grace CPU
- Reliability, Availability and Serviceability (RAS) Engine
  - monitor HW/SW health

  - AI-powered predictive management
- Companion NVIDIA Grace CPU
  - GB200 Grace Blackwell Superchip
  - GB200 1 Grace CPU and 2 Blackwell GPUs
  - Ah, the 20/40 petaFLOPS is FP4 math, only 90 teraFLOPS FP64
  - 114MB L3 Cache
  - HBM memory, up to 384GB
  - LPDDR5x memory up to 480GB
- GB200 NVL72 server – 36 grace CPUs and 72 Blackwell GPUs

  - liquid cooled
  - Don’t say much about power
  - Say 25x lower TCO and energy than H100 air-cooled
- HGX B200 – up to 1000W per GPU

Grace CPU

- 144 Arm neoverse V2 cores

Homework Reading #2

- 30% more transistors despite same process
- Two dies as one GPU. They just ran out of room and had to split it in half (chiplets of a sort)
- Each chip 4 stacks of HBM memory, 8192-bit wide
- B200 is 1000W per module
- Sparse FP4 performance (sparse?)
- FP6? Runs through the FP8 circuitry but takes up less RAM?

Intel Xe-HPG / Arc

- Each render slice contains 4Xe cores, 4 tracing units
- DirectX12 Ultimate compliant
- Ray Tracing, Variable Rate Shading, Mesh Shading
- 16 256-bit XVE vector engines (rasterization)
- 16 1024-bit XMX matrix engines (machine learning) 4-deep systolic array can calc 256 multiply-accumulate ops per clock for INT8 inferencing (?)
- Thread sorting unit for Ray-tracing
- Concurrent INT/FP32 pipelines (NVidia introduced in

Turing)

- Media engine, 8K HDR encode/decode, HEVC, VP9, AV1
  AV1 decoding 50x faster than software
- Clock speed comparisons across GPUs difficult

AMD GPUs

Caribbean Islands			Fiji
Sea Islands			
Volcanic Islands			
Polaris RX400	2016	28/14nm	
Polaris / RX500	2017	14/12nm	
Vega	2017	14/7nm	GCN5
Navi / RX5000	2019	7nm	RDNA
Navi2x / RX6000	2020	7nm	RDNA2
	2022	N5	RDNA3

RDNA3

- RX7900
- Use chiplet design

  Can use 5nm for cores and 6nm for memory
- Perf for watt
- GDDR6 vs GDDR6X because uses less power

AMD RDNA vs CDNA

- RDNA (Radeon DNA) is gaming
- CDNA is compute
- going to merge them?

AMD Instinct

- Found on supercomputers
- MI300A on El Capitan, $27,000, A version is an APU (combined with an AMD CPU)
- Instinct MI300X
  - 304 compute units, 1216 matrix cores, 19456 stream processors
  - 81.7 TFLOPs 64-bit
  - 163.4 TFLOPs fp64 matrix (?)
  - 750W

  - infinity fabric
  - 192GB RAM (HBM3E)
  - CDNA3 – chiplet based
  - TSMC N5/N6
- Limited HBM supply, suppliers sold out?
- CDNA4 in 2025 with 3nm, FP4/FP6

Embedded GPUs

- Videocore on Pis?
- MALI?

Apple M systems

- Apple M1 - M4
- People reverse engineering it to great effect
- Linux version written in Rust?
- Honeykrisp vulkan driver
- Work sponsored by valve to get support for Windows games emulated on ARM/Linux
  - emulate x86
  - emulate DirectX into Vulkan
  - Vulkan driver support

- ◦ 4k page table pages (Apple is 16k), solution is to run in VM image?