MIAOW - An Open Source GPGPU[*]

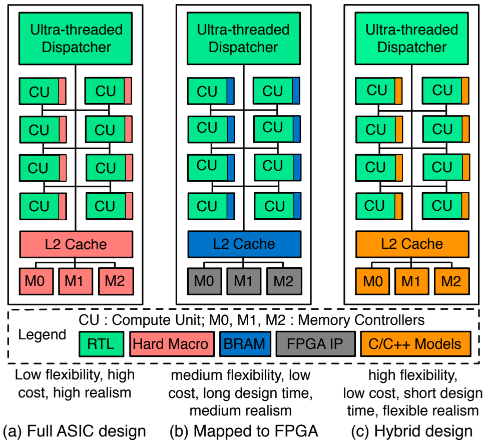
Raghuraman Balasubramanian Vinay Gangadhar Ziliang Guo Chen-Han Ho Cherin Joseph Jaikrishnan Menon
Mario Paulo Drumond Robin Paul Sharath Prasad Pradip Valathol Karu Sankaralingam
University of Wisconsin-Madison (karu@cs.wisc.edu) (www.miaowgpu.org)
Presenter: Karu Sankaralingam, Associate Professor UW-Madison

Abstract: *This paper describes an open-source GPU implementation. Its contributions are relevant both for industry and academia and include – i) It serves as the first publicly disclosed description of GPU microarchitecture, ii) A performance comparison of an academic open-source design to proprietary hardware products, iii) It contributes toward the emerging trend of open source hardware being driven by initiatives like OpenCompute, OpenPOWER, Arduino, and the Maker movement, and iv) Technical contributions on research case studies on GPU microrarchitecture feasible only with an RTL implementation. ASIC synthesis and floorplan of MIAOW are complete. An FPGA prototype is complete for demo. The source code is released at https://github.com/VerticalResearchGroup/miaow.*

Motivation & Overview This project builds on two trends in IT: *big data* and *open hardware*. Big data is transforming various businesses, sciences, and society and from a microprocessor standpoint it needs power-efficient computing. Many have argued that the same *principles* of power-efficient computing will span brawny cores and tiny cores in IoT-like devices. The second trend, which is related but a bit under-appreciated is open hardware and the maker movement. Open hardware has largely focused on system-architecture (Open Compute, OpenPOWER etc.) while the maker movement has focused on non-semiconductor physical objects (3D printing etc.) and boards (Arduino being the most prominent example). Last year, 750,000 people attended 131 maker fairs around the world [1]. A recent New York Times article summarizes some bold moves by Facebook and Tesla motors for open technology development [1] and a recent IEEE Spectrum[4] article argues for open source hardware in the context of Moore's law's "deceleration". The benefits of open source have been established by the open-source software community; this includes better products, fewer bugs, faster product development, and perhaps even better security [2].

GPUs are an important means to provide power-efficient computing compared to conventional processors spanning brawny to tiny cores. This project reports on the development of an (the first and only) open-source GPU called MIAOW that can play a role in this nascent, yet fast growing space of open source hardware. In addition to the "high-level" relevance, this project can be an important part of the OpenPOWER microprocessor hardware initiative in building customizable chips. The project also proposes and describes the details of microarchitecture implementation of a GPU publicly for the first time. This should be of interest to Hotchips audience from both an academic and industry perspective. It gives industry an idea of how academia thinks of GPU microarchitecture details. It also widely disseminates to academia – a detailed microarchitecture implementation of GPU for the first time.

Goals The primary driving goals for MIAOW are: i) *Open source:* The RTL source should be released open source along with a verification tool chain, synthesis scripts etc. ii) *Software-compatible:* It should use standard and widely available software stacks like OpenCL or CUDA compilers to enable execution of various applications and not be tied to in-house compiler technologies and languages. iii) *Realism:* It should be a *realistic* implementation of a GPU, resembling principles and implementation tradeoffs like industry GPUs; iv) *Flexible:* It should be flexible enough to accommodate research studies of various types, the exploration of forward-looking ideas, and form an end-to-end open source tool; v) *ASIC-certifiable:* It should be implementable as an ASIC.


Ultra-threaded Dispatcher
CU
L2 Cache
M0
M1
M2
Legend
CU : Compute Unit; M0, M1, M2 : Memory Controllers
RTL
Hard Macro
BRAM
FPGA IP
C/C++ Models

Low flexibility, high cost, high realism
(a) Full ASIC design

medium flexibility, low cost, long design time, medium realism
(b) Mapped to FPGA

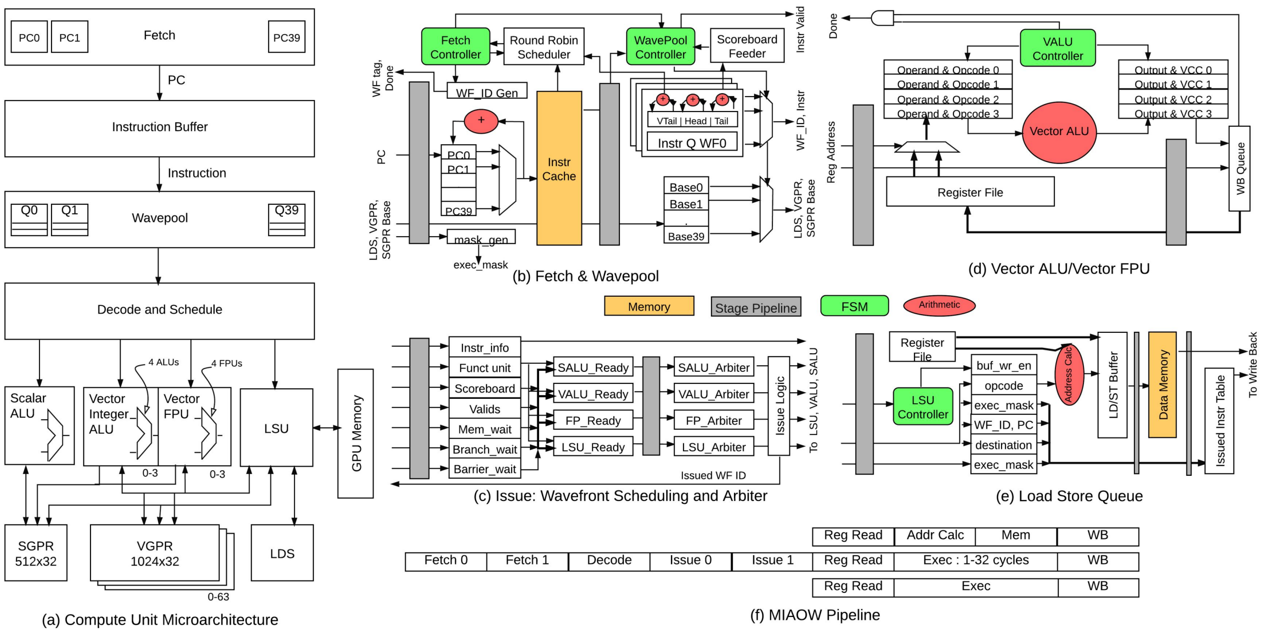
high flexibility, low cost, short design time, flexible realism
(c) Hybrid design

Figure 1: MIAOW instantiations

Hardware Overview We have developed MIAOW as an implementation of a subset of AMD's Southern Islands (SI) ISA [3][1]. Our implementation includes a scalable number of compute units (CU), the ultra-threaded dispatcher (scheduler) that sends work units to the CUs, L2-cache and memory controller. Figure 1 shows three different ways MIAOW can be used, the second and third being most relevant for academics. MIAOW follows established principles of GPU organization and its microarchitecture is close to the state-of-art as sketched in Figure 2. At this

[*] Some authors have current affiliations at Google, Qualcomm, NVIDIA, and EPFL. Work done at UW-Madison

[1] Our public wiki and a to-appear TACO paper discuss many of these details and tradeoffs.



(a) Compute Unit Microarchitecture

(b) Fetch & Wavepool

(c) Issue: Wavefront Scheduling and Arbiter

(d) Vector ALU/Vector FPU

(e) Load Store Queue

(f) MIAOW Pipeline

Figure 2: MIAOW Compute Unit Block Diagram and Design of Submodules

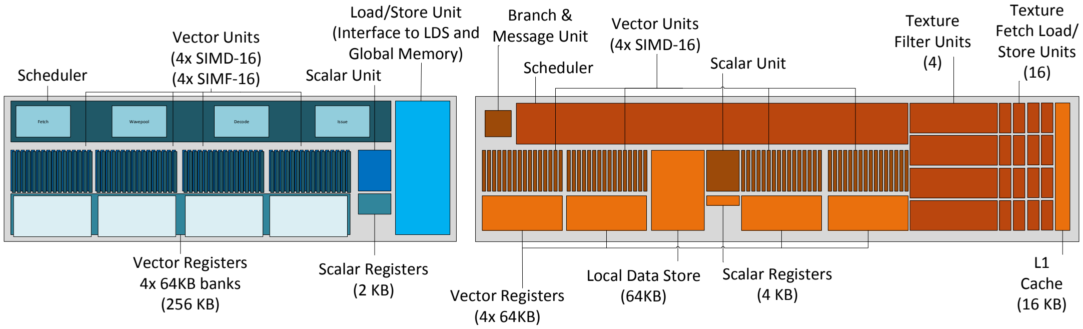

Scheduler
Vector Units (4x SIMD-16) (4x SIMF-16)
Load/Store Unit (Interface to LDS and Global Memory)
Scalar Unit
Fetch
Wavepool
Decode
Issue
Vector Registers 4x 64KB banks (256 KB)
Scalar Registers (2 KB)
Branch & Message Unit
Scheduler
Vector Units (4x SIMD-16)
Scalar Unit
Texture Filter Units (4)
Texture Fetch Load/ Store Units (16)
Vector Registers (4x 64KB)
Local Data Store (64KB)
Scalar Registers (4 KB)
L1 Cache (16 KB)

Figure 3: MIAOW GPU vs Kaveri APU Comparison

point, MIAOW does *not* provide any graphics capability and is focused exclusively on the programmable part of the graphics pipeline and execution engine. An FPGA implementation of MIAOW maps a single-CU design to the Virtex-7 FPGA. An ASIC floorplan is complete, however we *do not* have chip manufactured and will *not* in time for Hotchips. Regardless we feel this presentation will be of interest to the Hotchips audience. We have completed an area, power, and performance analysis of MIAOW. At 32nm process node, MIAOW has an area of 9.31mm^2, runs at 222 MHz and consumes typically 1.1 Watts. On the AMD OpenCL SDK benchmarks, MIAOW has *CPI* of 24% to 93% of the comparable industry GPU. Considering the inexperience of our design team, with inefficient datapath modules, and limited physical design optimizations, these results are good. Figure 3 shows the design comparison of MIAOW and AMD Kaveri APU.

Our talk will present the microarchitecture details, implementation tradeoffs in depth, and discuss design choices made in various intricate places like the ultra-threaded dispatcher, register-file operand collectors etc. Our completed research case studies demonstrate MIAOW's transformative capability: we show how a previously proposed thread-scheduling microarchtiecture technique impacts the timing critical path, we show how transient faults affect GPUs (the first such low-level fault injection study), we have developed a permanent fault-detection implementation using Sampling-DMR re-purposed for GPUs. Our talk with summarize these.

Relevance and Impact MIAOW could be of wide interest to Hotchips audience for its technical contributions in GPU design and for its potential to trigger open source development in the microprocessor design community.

References

[1] For Hardware Makers, sharing their secrets is part of the Business Plan, New York Times March 29, 2015.

[2] Is open source good for security?, `http://www.dwheeler.com/secure-class/Secure-Programs-HOWTO/open-source-security.html`.

[3] Reference guide: Southern islands series instruction set architecture, `http://developer.amd.com/wordpress/media/2012/10/AMD_Southern_Islands_Instruction_Set_Architecture.pdf`, 2012.

[4] A. Huang. The Death of Moores Law Will Spur Innovation, IEEE Spectrum March 31, 2015.