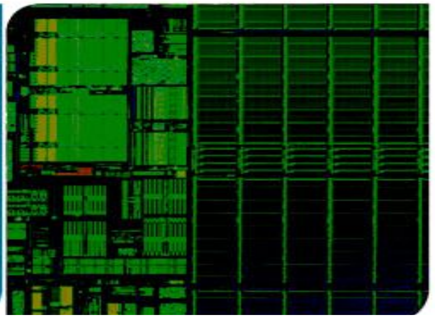
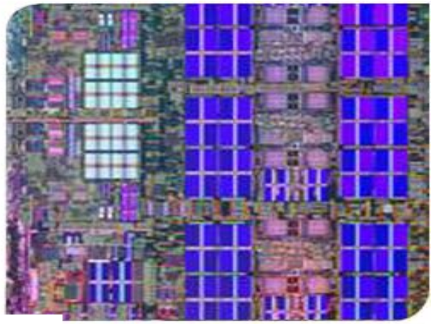
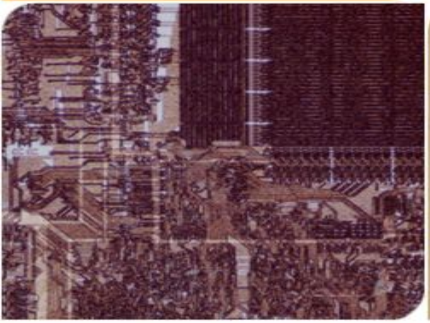




Vortex: A Reconfigurable RISC-V GPGPU Accelerator for Architecture Research

Fares Elsabbagh, Blaise Tine, Apurve Chawda, Will Gulian,
Yaotian Feng, Da Eun Shim, Priyadarshini Roshan,
Ethan Lyons, Lingjun Zhu, Sung Kyu Lim, Hyesoon Kim

Georgia Tech comparch



Abstract

- The emergence of data parallel architectures have enabled new opportunities to address the power limitations and scalability of multi-core processors, allowing new ways to exploit the abundant data parallelism present in emerging big-data and machine learning applications. This transition is getting a significant boost with the advent of RISC-V with its unique modular and extensible ISA, allowing a wide range of low-cost processor designs. In this work, we present Vortex, a full-stack RISC-V GPGPU processor with OpenCL support. The Vortex platform is highly customizable and scalable with a complete open-source compiler, driver, and runtime software stack to enable research in GPU architectures.
- We evaluated this design using 15 nm technology. We also show the preliminary performance and energy numbers of running them with a subset of benchmarks from the Rodinia Benchmark suite.

RISC-V Ecosystem

- RISC-V ISA features
  - Open ISA for accessibility
  - Frozen ISA for compatibility
  - Extensible ISA for customization
  - **Ideal for architecture research!**
- Open-source cores
  - Rocket, Boom, Ariane, Piccolo, etc..
- Open-source compiler
  - LLVM, GCC
- Open-source software
  - Linux, FreeRTOS, QEMU, BSD, etc...

https://www.sifive.com

RISC-V Vector Extension

Current standard ISA supports

- In-order processor
- Out-of-order processor
- Vector processor (in-the-works)

RISC-V Vector ISA extension

- Mixed-width computations
- Fixed-point and f16

Implementations

- Ara[1]: implements early ISA proposal
- HWACHA[2]: use micro-ops and co-processor

https://www.sifive.com

[1] M. A. Cavalcante et al., "Ara: A 1 GHz+ RISC-V vector processor
[2] Y. Lee et al., "A 45nm 1.3ghz 16.7 risc-v processor with vector acc"

RISC-V SIMT Extension

- SIMT Advantages:
  - Scalar-based programming model
    - Easier to program
  - Parallel-first architecture
    - Best efficiency for highly parallel workloads
    - Ideal for graphics rendering
- No ISA proposal in the works
  - Some implementations exists
    - Simty[3]: a microarchitecture only.
- Challenges
  - A flexible minimal ISA extension for SIMT

https://www.sifive.com

[3] S. Collange, “Simty: generalized simt execution on risc-v”

A RISC-V-based GPGPU Accelerator

- Lower-cost implementation
  - Leveraging existing ISA and software stack
  - Today’s FPGAs have enough capacity
    - e.g. Intel Arria10, Stratrix10
- Great for architecture research
  - Design Verification
  - CPU-GPU communication
  - Near memory
  - Hardware virtualization
  - Hardware security
  - Graphics rendering
  - Hardware Specialization: NPU, VPU, etc..

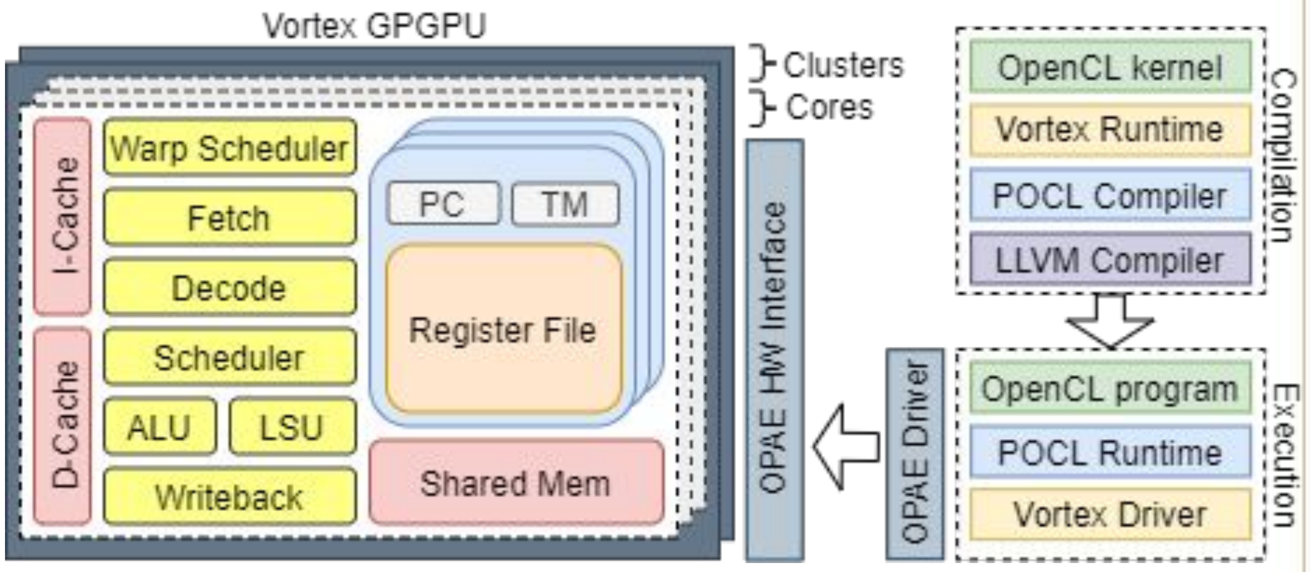
https://www.sifive.com

Vortex GPGPU Systems Architecture

Software Stack
- Supports OpenCL 1.1 API

Driver Stack
- Portable Driver API
  - FPGA, ASE, RTLSim, SimX
- Current Target FPGA:
  - Arria10 Intel Accelerator Card v1.0

Open-source Toolchain
  - POCL: OpenCL Compiler & runtime
  - OPAE: FPGA Driver API
  - Verilator: RTL simulation
  - Yosys: FPGA Synthesis
  - Gem5: CAS Simulation



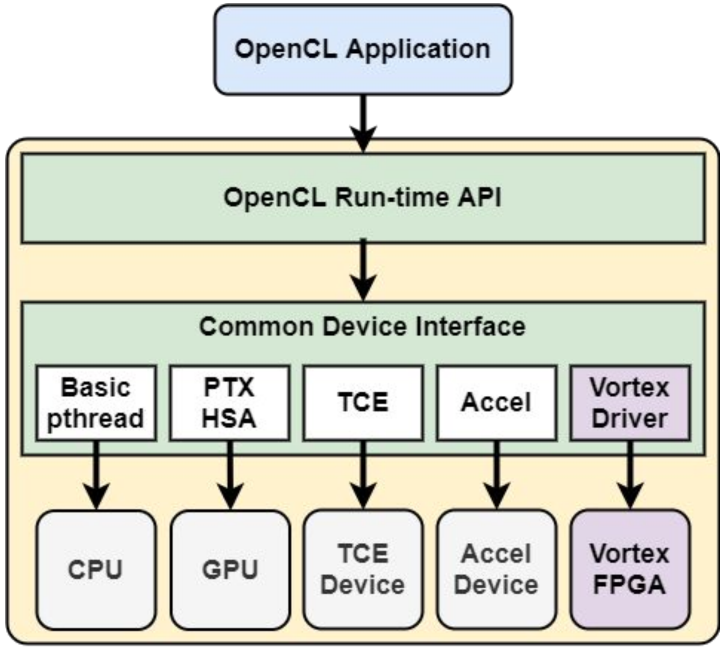
OpenCL Software Stack

OpenCL Runtime

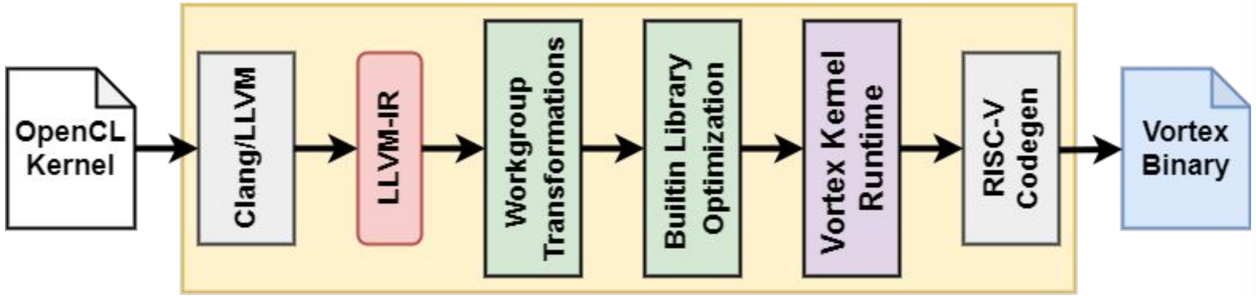
- Use POCL Runtime framework[4]
- Added new device target for Vortex FPGA
- FPGA Driver uses Intel OPAE API[5]

OpenCL Compiler

- Use POCL Compiler framework[4]
- Added Vortex Kernel Runtime Pass
  - Work items => Vortex threads
  - Hardware Warp invocations



POCL Runtime



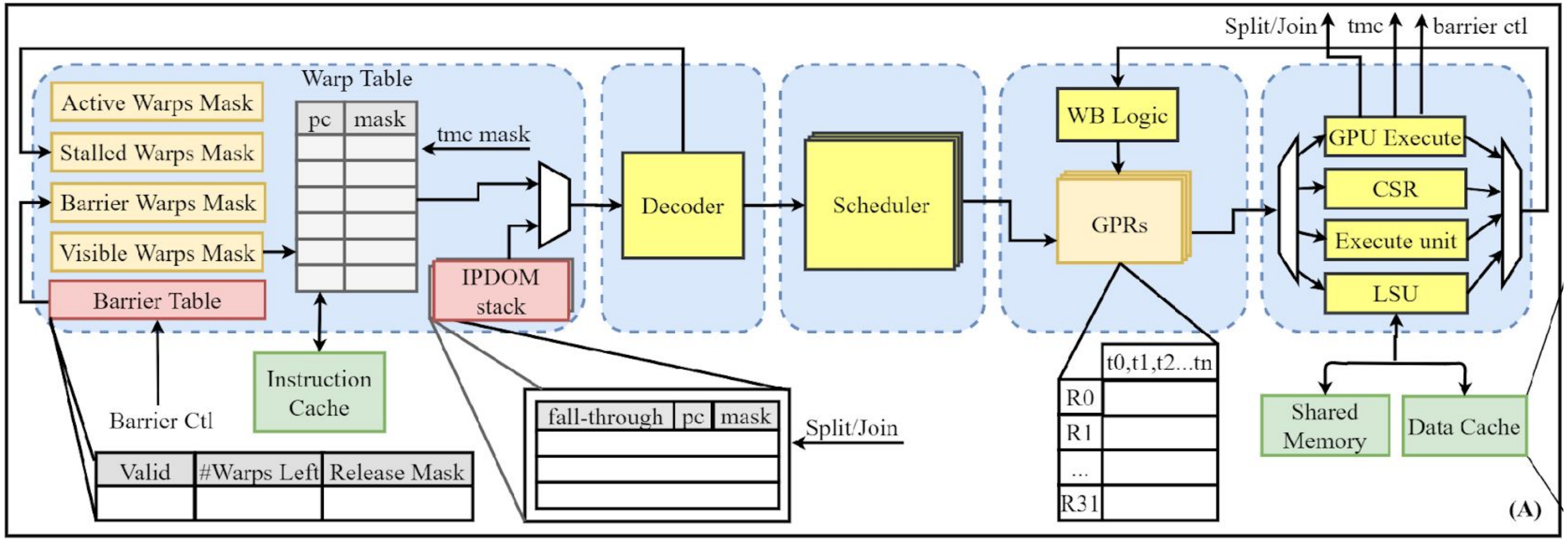
POCL Compiler

[4] Pekka Jääskeläinen et al "pocl: A Performance-Portable OpenCL Implementation"
[5] Intel OPAE « https://01.org/OPAE »

Vortex GPU Microarchitecture

Vortex RISC-V ISA Extension for SIMT Execution

- **wspawn**: warps creation
- **tmc**: threads activation
- **split/join**: control flow divergence
- **bar**: memory barriers



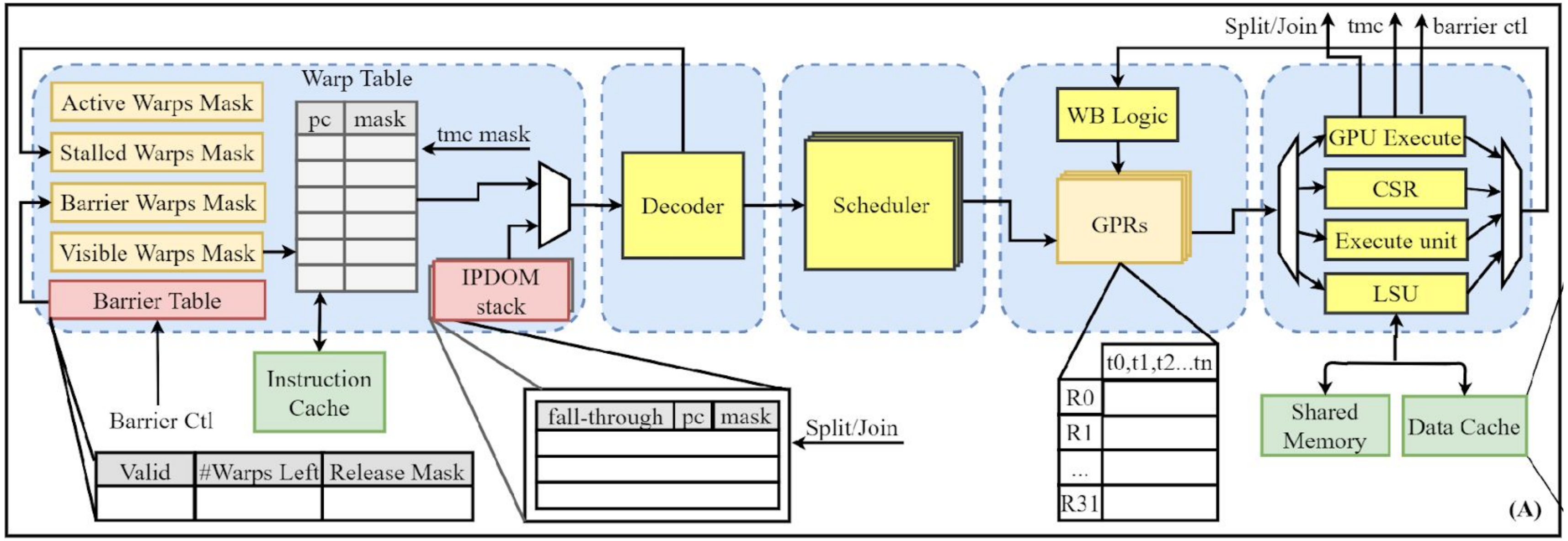
Vortex GPU Microarchitecture

Split/Join Instructions:

- Handles control divergence by keeping track of divergent threads and their PCs in the IPDOM stack

Bar Instruction:

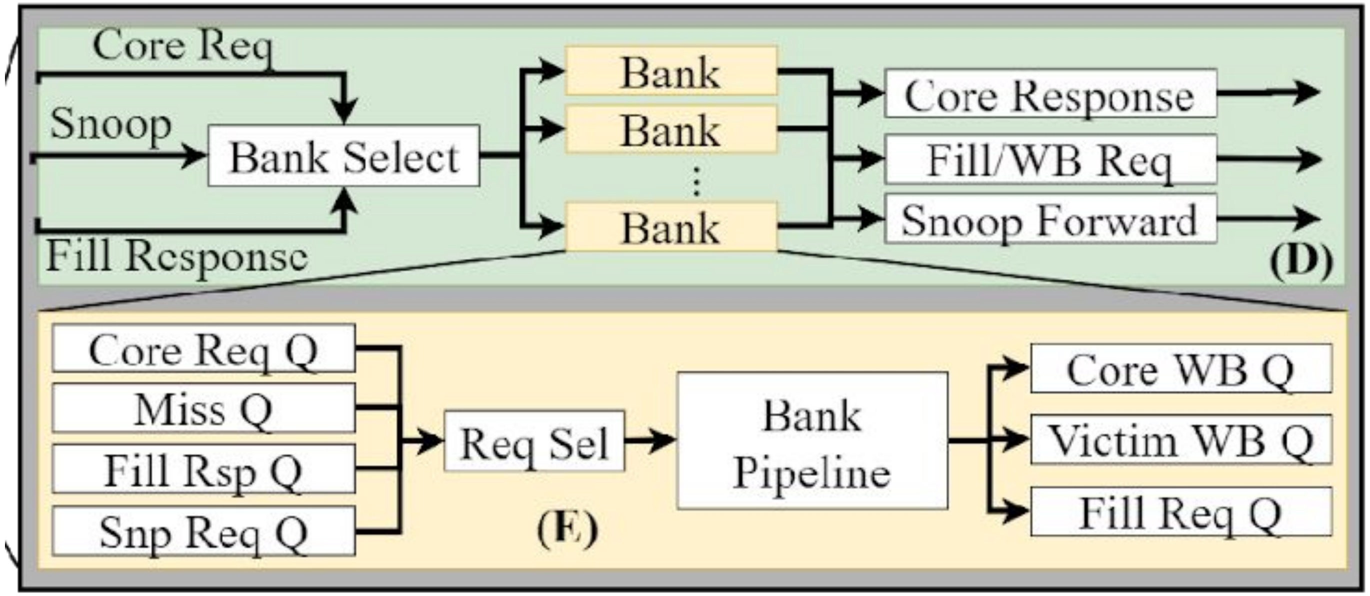
- Handles synchronization by locking warps that execute the bar instruction with a bar_id using the Barrier Table, and releases them once they are synchronized



Vortex GPU Microarchitecture

Vortex's fully configuration cache sub-system

- High-bandwidth with bank parallelism
- Snoop protocol to flush data for CPU access
- Generic Design: Dcache, Icache, Shared Memory, L2, L3

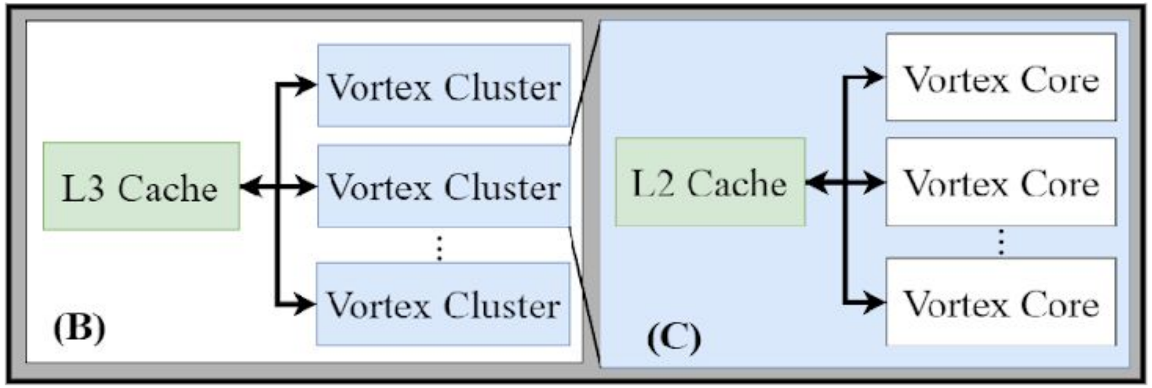


Georgia Tech comparch

Vortex GPU Microarchitecture

Vortex's modular and scalable architecture

- **Threads**: smallest unit of computation
- **Warps**: collection of concurrent hardware threads
- **Cores**: processing element containing multiple warps
- **Clusters**: collection of processing elements



Preliminary Performance Evaluation

Simulation performance

- Good performance scaling with added cores
  - Config: 2-cores, 4-cores, 8-cores
- Use OpenCL Rodinia benchmark

FPGA Synthesis

- Clock frequency ~192 Mhz
- 2 cores to 8 cores configuration
- Static power of 2.2W

Layout synthesis

- Use 15-nm educational cell library
- Main power dissipation from caches, GPRs

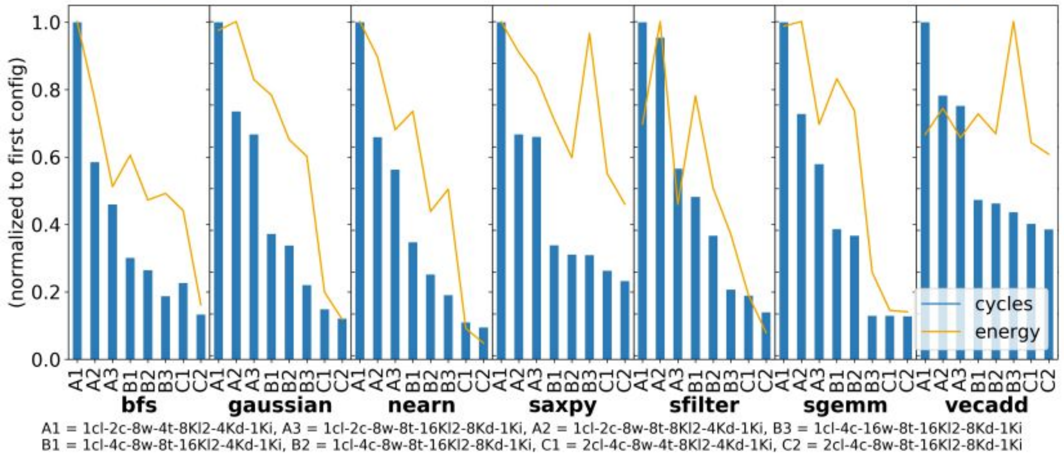
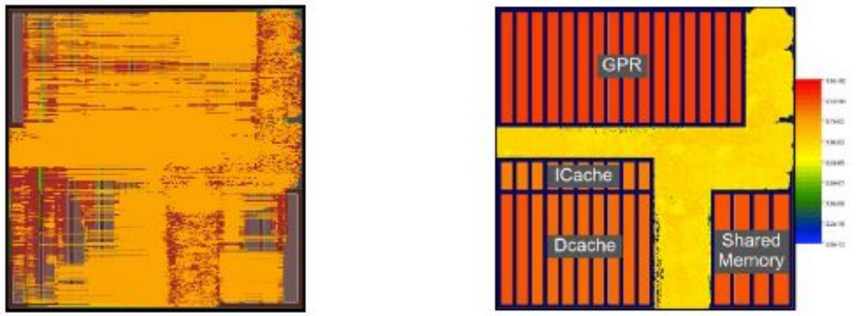


Fig. 4: Vortex v0.1 performance for Rodinia benchmark with normalized cycle and energy utilization on Arria 10 FPGA



(a) GDS Layout (b) Power density distribution

Fig. 3: GDS layouts for a single-core 8-warp 4-thread configuration synthesized @300Mhz produced 46.8mW total power

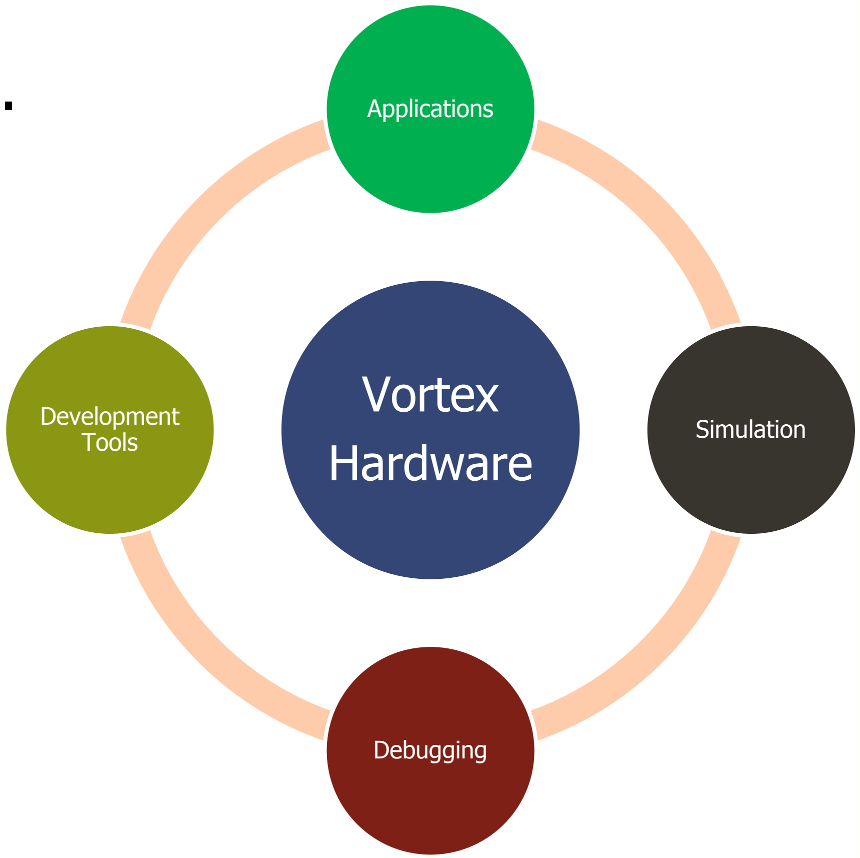
Vortex Accelerator Roadmap

Full-featured GPU Implementation

- Mainstreams GPU APIs
  - CUDA, OpenCL, Vulkan, OpenVision, Tensorflow, etc..
- GPU applications
  - Compute, graphics, vision, ML, etc..
- FPGA ports
  - Altera, Xilinx

Research Focus

- Simulation and debugging features
- Open-source GPU drivers (HSA)
- Open-source development tools
  - LLVM, POCL, Verilator, OPAE, yosys, Gem5



References

- Website: https://vortex.cc.gatech.edu
- Github: https://github.com/vortexgpgpu/vortex

Thank you!