**Ventus: an RVV-based General Purpose GPU Design and Implementation**

Kexiang Yang1,2, Hualin Wu3, Jingzhou Li1,2, Chufeng Jin1,2, Yujie Shi1,2, Xudong Liu1,2, Zexia Yang1,2, Fangfei Yu1,2, Mingyuan Ma1,2, Sipeng Hu4, Tianwei Gong4, Hu He1,2

1Tsinghua University, 2International Innovation Center of Tsinghua University, Shanghai, 3Terapines Ltd, 4Beijing Information Science and Technology University

---

**What is Ventus?**
- Open-sourced RVV-based GPGPU
- An implementation of Chisel HDL, driver and compiler
- OpenCL compatibility
- RISC-V compatibility with 256 registers available

**ISA: RV32IMA_ZFinx_Zicsr_V**
- Vector instructions for per-thread operation, elen=32 bit, vlen=32*elen
- Scalar instructions for common data
- Custom instructions:
  - VBranch/Join to control thread divergence
  - EndProgram and Barrier to control warps
  - RegisterExtension to extend register index
- Registers: 64 sGPRs, 256 vGPRs
- Memory space definition and access methods
- Custom CSRs and metadata to launch workgroup and implement workitem functions

**Software Stack**
- Ventus-LLVM: compiler based on LLVM for Ventus ISA and library
- PoCL: OpenCL platform implementation
- Ventus-driver: KMD implementation
- Ventus-gpgpu-isa-simulator: ISS based on Spike

**Microarchitecture**
- Multi-level task allocation is implemented by driver and CTA-scheduler
- SM works as an RVV processor supporting warp scheduling
- 4-bank register files can be allocated according to usage
- Tensor Core supports custom tensor operations

---

**Evaluation & Conclusion**
- A complete implementation of GPGPU based on RVV
- Chisel HDL configurable in num of warps, threads, SMs, lanes...
- A 16SM-16warp-16lane version with Tensor Core occupies 65% of the area of 4 VU19P FPGAs
- Open-sourced at [https://github.com/THU-DSP-LAB/ventus-gpgpu](https://github.com/THU-DSP-LAB/ventus-gpgpu)