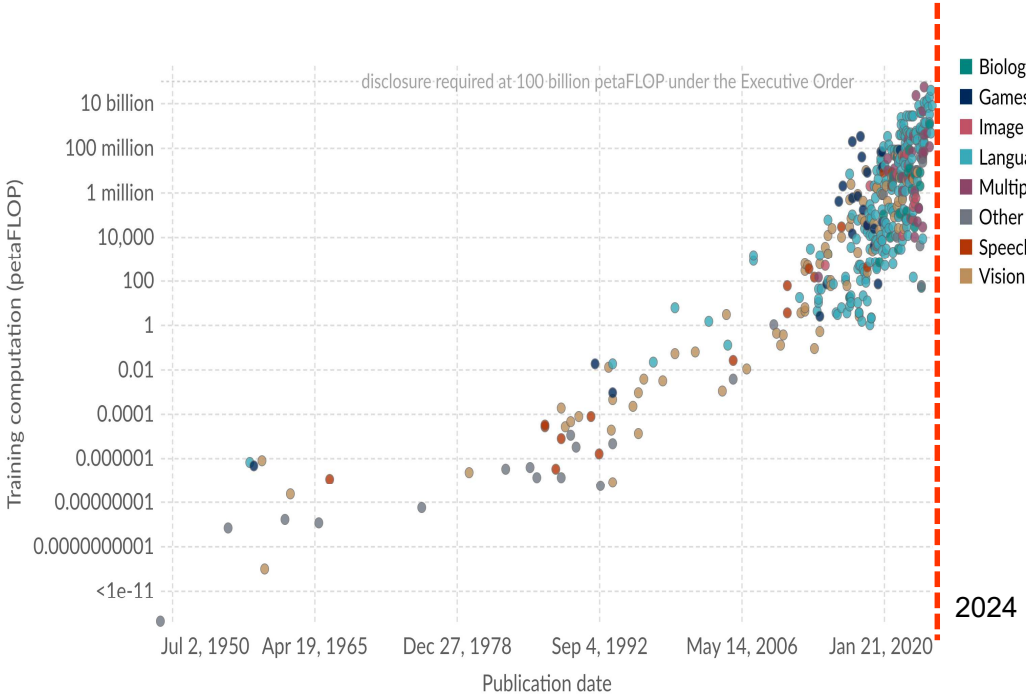
The 42nd IEEE International Conference on Computer Design (ICCD 2024)

Ventus: A High-performance Open-source GPGPU Based on RISC-V and Its Vector Extension

Jingzhou Li, Kexiang Yang, Chufeng Jin, Xudong Liu, Zexia Yang, Fangfei Yu,
Yujie Shi, Mingyuan Ma, Li Kong, Jing Zhou, Hualin Wu, Hu He

Motivation



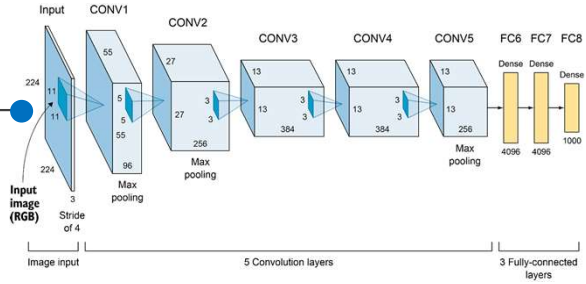
AI computing power demand is growing exponentially

ref: Our World in Data

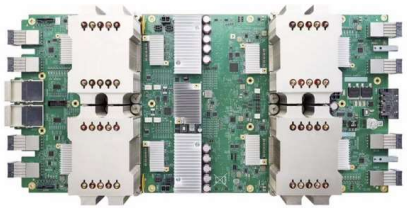
NVIDIA GTX580

2010

2012



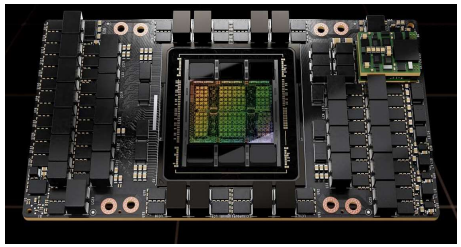
AlexNet



Google TPU V2

2017

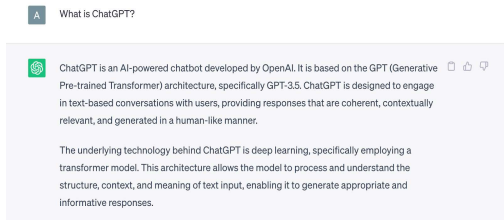
AlphaGo



NVIDIA H100

2022

2023



ChatGPT

Our Contribution: Ventus GPGPU

- Detail a holistic software toolchain from **OpenCL** to our RVV-based GPGPU ISA
- Offer an ISA conversion proposal from a **vector** processor to a high performance **GPGPU**
- Develop a GPGPU design project using **Chisel** agilely for design space exploration

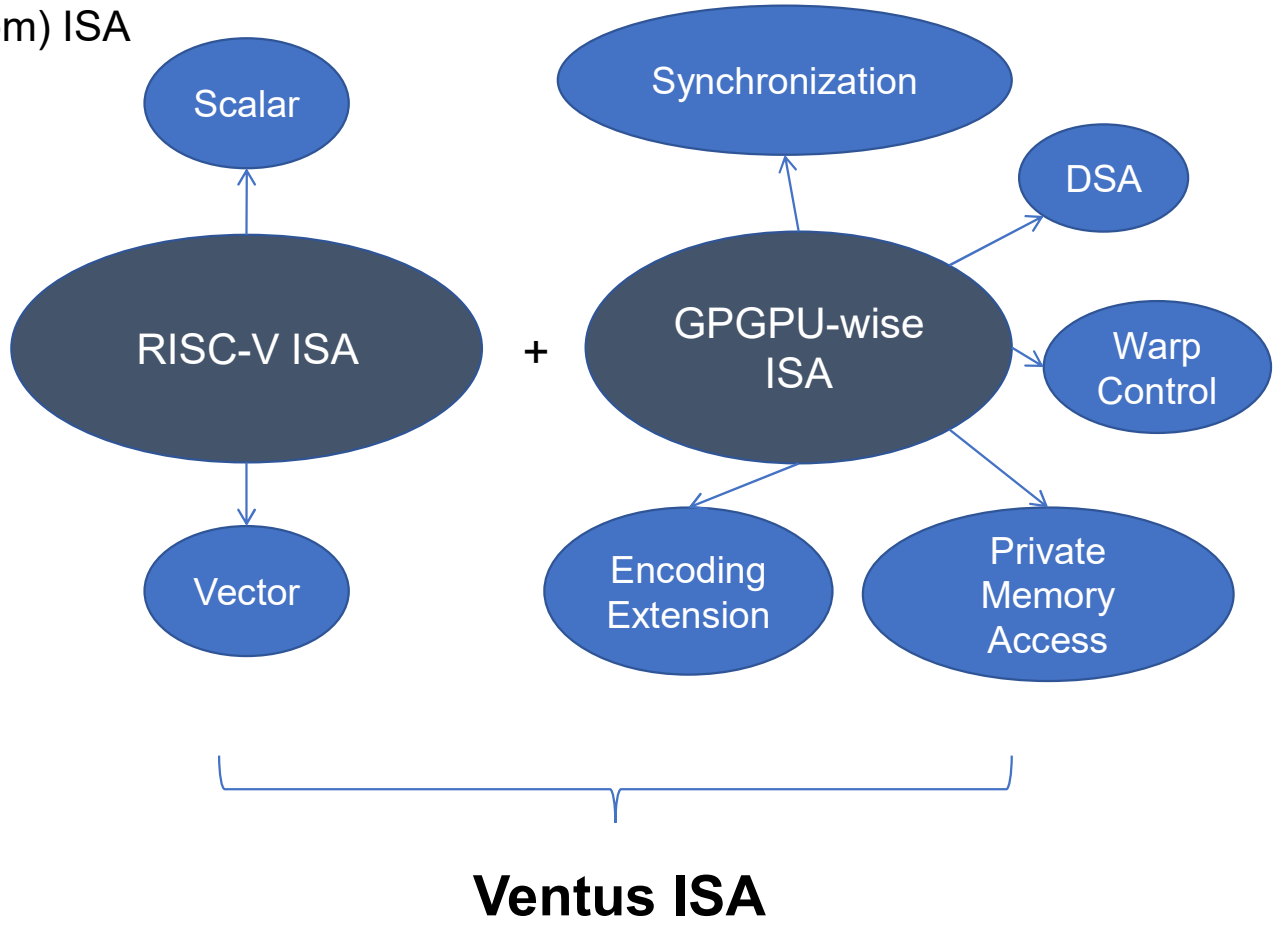
Ventus: ISA Extensions

Tsinghua University

Ventus ISA= RISC-V + GPGPU-wise (custom) ISA

RISC-V:
- Scalar: RV32IMA zfinx zicsr
- Vector: zve32f (i.e., 32 thread x 32-bit)

GPGPU-wise:
- Warp Control
  - kernel response: endprg
  - branch: vbeq, join, setrpc
  - synchronzization: barrier
- Encoding Extension: regext
- Private Memory Access: vlw.v, vsw.v ...
- DSA (tensor)
  - matrix multiply & add: vftta
  - exponential: vfexp
  - reduction / shuffle



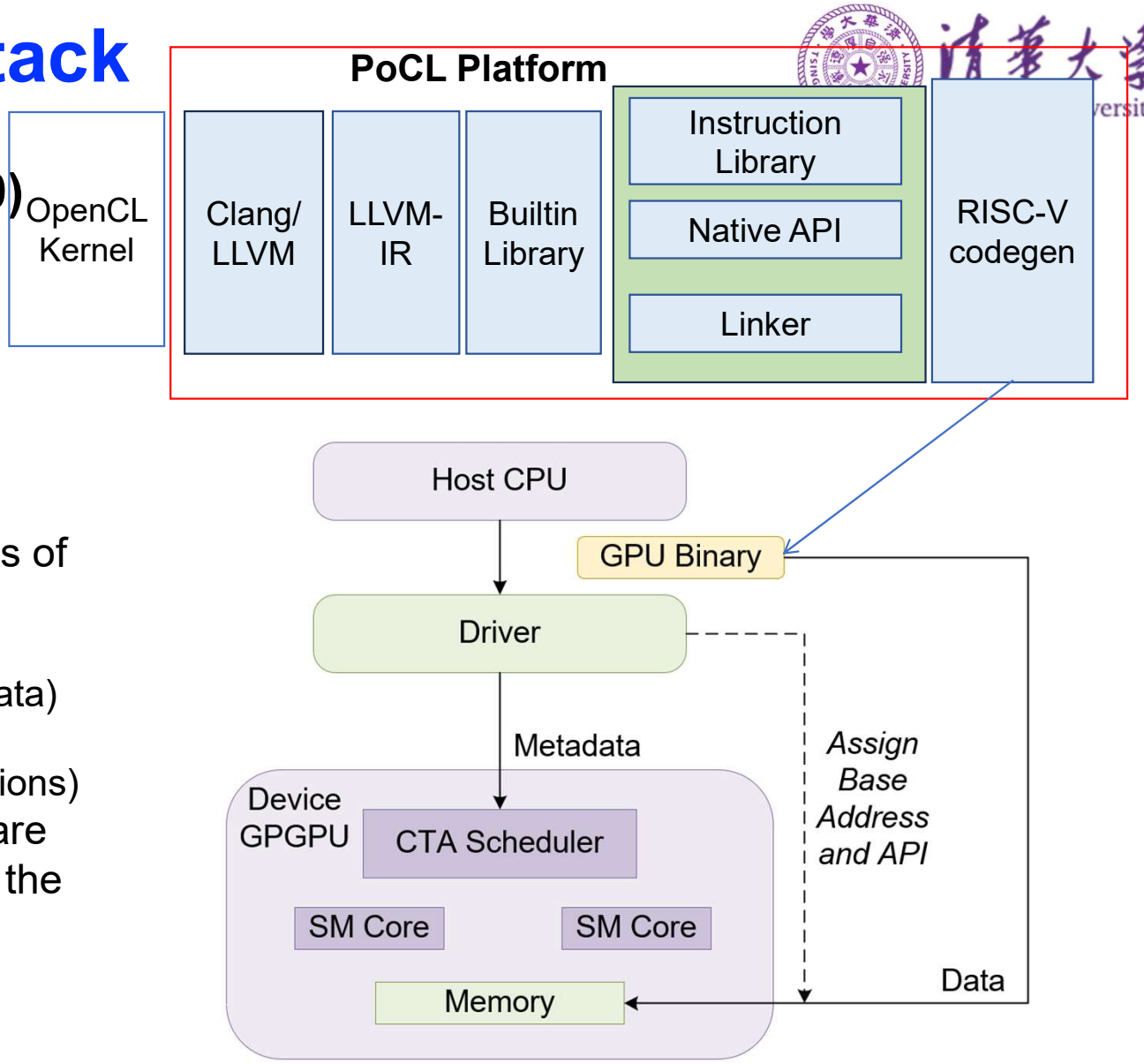
Ventus: Software Stack

Compiler (Support OpenCL 2.0)

- Implemented on PoCL Platform
- LLVM Backend
- OpenCL runtime

Hardware Driver

- OpenCL API support
- Manage different memory regions of Ventus GPGPU devices
  - Warp: LocalMemory (stack)
  - CTA: SharedMemory (shared data)
  - Kernel: GlobalMemory (data), ConstantMemory (data, instructions)
- Analyze program, extract hardware parameters and transfer them to the hardware
  - Thread size, workgroup id...



Ventus: Hardware Architecture

Tsinghua University

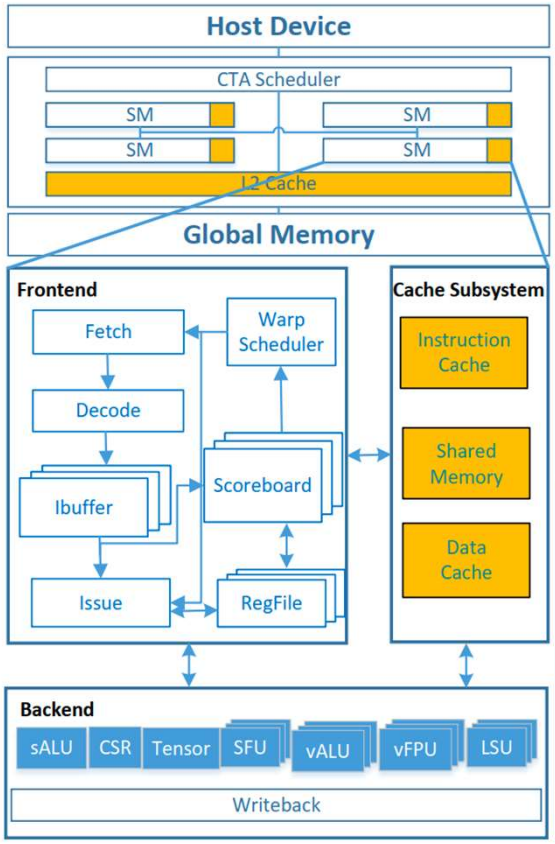
Host Device--> CTA Scheduler--> Streaming Multiprocessor (SM)--> Memory

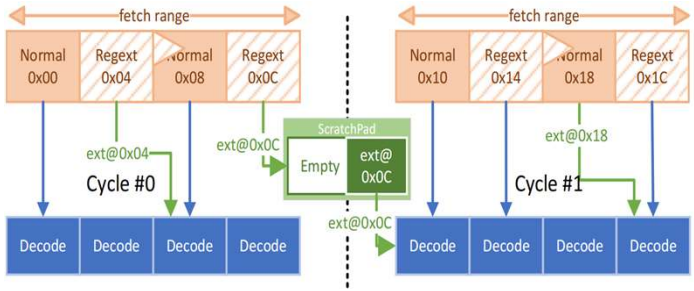
SM:

- A 6 stage SIMD processor
  - Fetch + Decode + Dispatch + Issue + Execute + Writeback
- Scalar & Vector dual issue
- Fine-grain warp scheduling
- Tensor core support

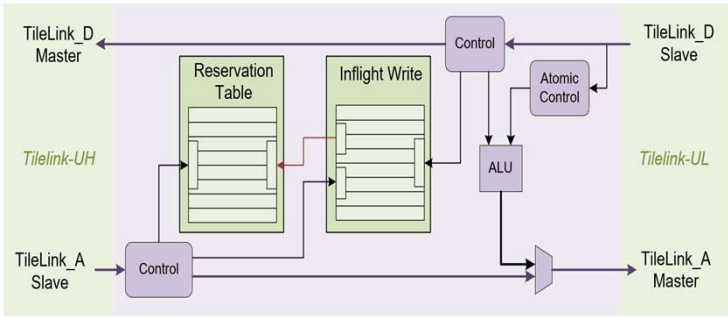
Cache Subsystem:

- L1 instruction & data cache +shared memory
- L2 cache banks shared by SMs
- Non-blocking
- Dedicated atomic unit and coarse-grained coherence mechanism





Regext instruction buffering



Atomic unit between L1 and L2 cache

Evaluation

Software stack：

- 83.9% executed instruction count reduction

Ins_count= committed_ins

Hardware Performance:

- 87.4% Cycle Per Instruction (CPI) reduction

CPI =total cycles/(committed_ins* active_thread_num)

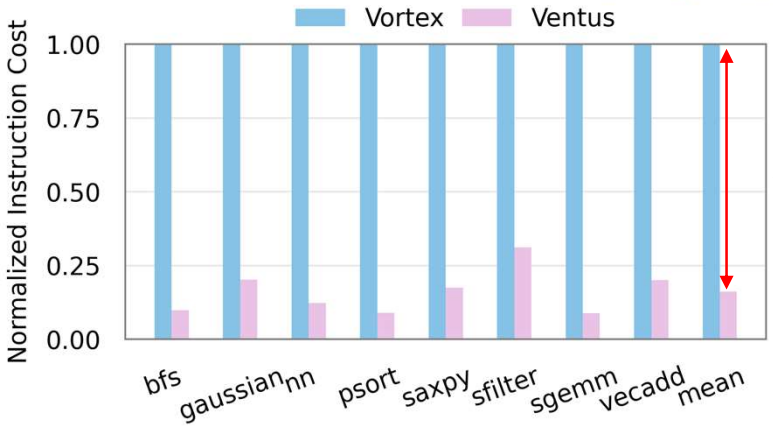
Synthesis Result:

- 1.2 GHz , TSMC 12nm 6 track DC
- 76.8 (1 SM) + 614.4 (1 Tensor Core) GFlops

Node	TSMC 12nm, 6 track cell
Frequency (DC)	1.2 GHz
Area (DC)	876084 μm^2
Warp & Thread Number	8, 16
Scalar & Vector Register Number	1024, 1024
L1 ICache & DCache	16KB, 2 Ways
Shared Memory	16KB
Cache Policy	Random Replacement, Writeback, Write no allocate
Cache Line Size	64 Bytes

DC Configuration

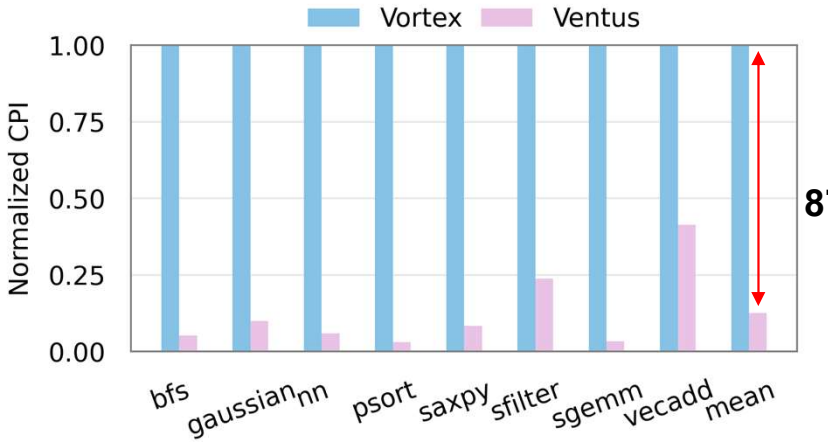
(Tine et al, MICRO 21)



83.9% reduction

Instruction Cost

(Tine et al, MICRO 21)



87.4% reduction

CPI evaluation

Conclusion

清華大學 Tsinghua University

The achievements of Ventus:

- A high-performance, OpenCL compatitive, open-source GPGPU on RISC-V and RVV
- Significant reductions in instruction count and CPI over Vortex (83.9% instruction, 87.4% CPI)
- Agile, parameterized hardware design for design space exploration

Open-source link:
https://github.com/THU-DSP-LAB/ventus-gpgpu

Thanks!

Q&A