RISC-V based GPGPU on FPGA: A Competitive Approach for Scientific Computing ?

E. Guthmuller & J. Fereyre
CEA List
eric.guthmuller@cea.fr

Motivation

GPUs have enabled supercomputers to exceed exaFLOP performance.
But AI is driving GPU architecture evolutions, needing only low precision computing.

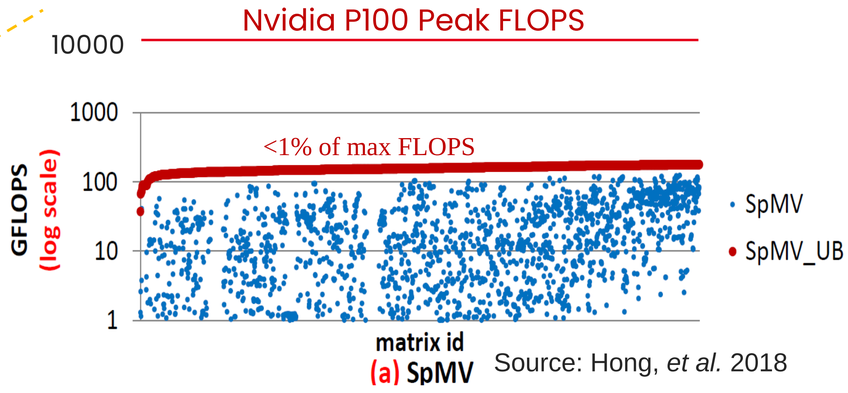
⇒ **How long before 64b support in GPUs is dropped or emulated?**

Typical scientific computing kernel needs 64b floating point support but exhibits low arithmetic intensity and performance is limited by memory throughput.
FPGAs now provide hardened arithmetic units, Network-on-Chips (NoC) and memory controllers, including High Bandwidth Memory (HBM).
While dedicated architectures or CGRAs may better exploit FPGA fine-grained architecture, GPGPUs are easier to program and already exploited in existing code.

⇒ **Would it be possible to implement a GPGPU on FPGA maximizing HBM throughput, and thus being competitive with ASIC GPUs?**

```
function solve_CG(A::Matrix{Float64},
    b::Vector{Float64},
    tol::Float64=1e-12; max_iter::Int=Int(1e5)
    )
n = length(b)
x = zeros(n)  # Initial guess
r = b - A * x  # Initial residual
p = copy(r)  # Initial search direction
r_norm = norm(r)
for iter in 1:max_iter
    # matrix-vector multiplication
    Ap = A * p
    # vector-vector operations
    alpha = dot(r, r) / dot(p, Ap)
     x += alpha * p
     r -= alpha * Ap
    new_r_norm = norm(r)
    if new_r_norm < tol
        return x, iter
    end
    beta = (new_r_norm / r_norm)^2
    p = r + beta * p
    r_norm = new_r_norm
end
return x, max_iter
end
```

Fig 1. Example of a typical HPC kernel: Congugate Gradient (CG) iterative solver.



Source: Hong, *et al.* 2018

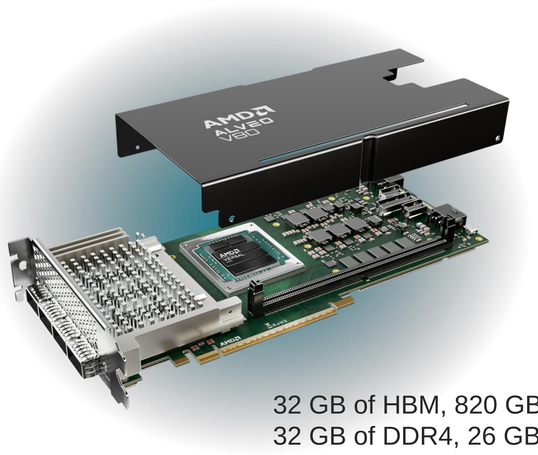
Fig 2. SpMV throughput on Nvidia P100

Kernel	Operation	Max GFLOPS
DOT	$\propto= x \cdot y$	205
AXPY	$y = \propto x + y$	68
SCAL	$x = \propto x$	102
SpMV	$y = Ax$	135

Fig 3. Roofline model of linear algebra kernels for 820 GB/s memory

Platform & Architecture

☐ Targeted Platform: AMD Alveo V80



32 GB of HBM, 820 GB/s
32 GB of DDR4, 26 GB/s
PCIe 5.0 16x, 128 GB/s

128 GB/s VNoC
256 GB/s HNoC

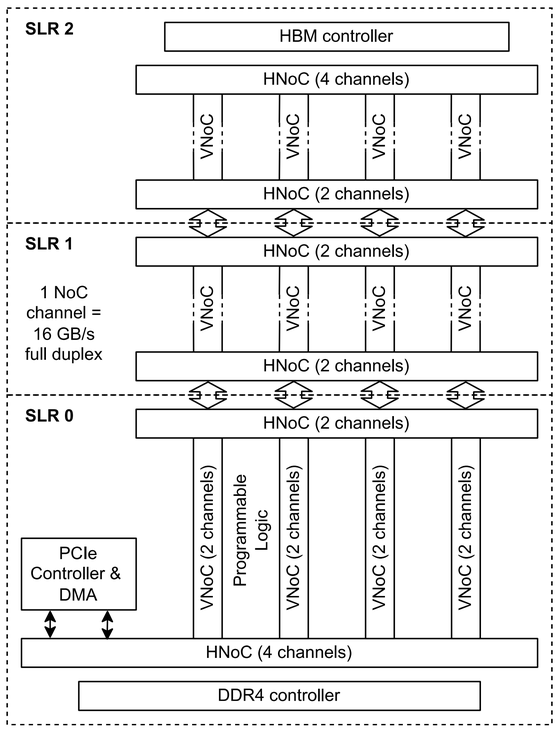


Fig 4. AMD Alveo V80 main features and organization

☐ Vortex FPGA Implementation

Vortex is an opensource RISC-V based GPGPU (https://github.com/vortexgpgpu/vortex).

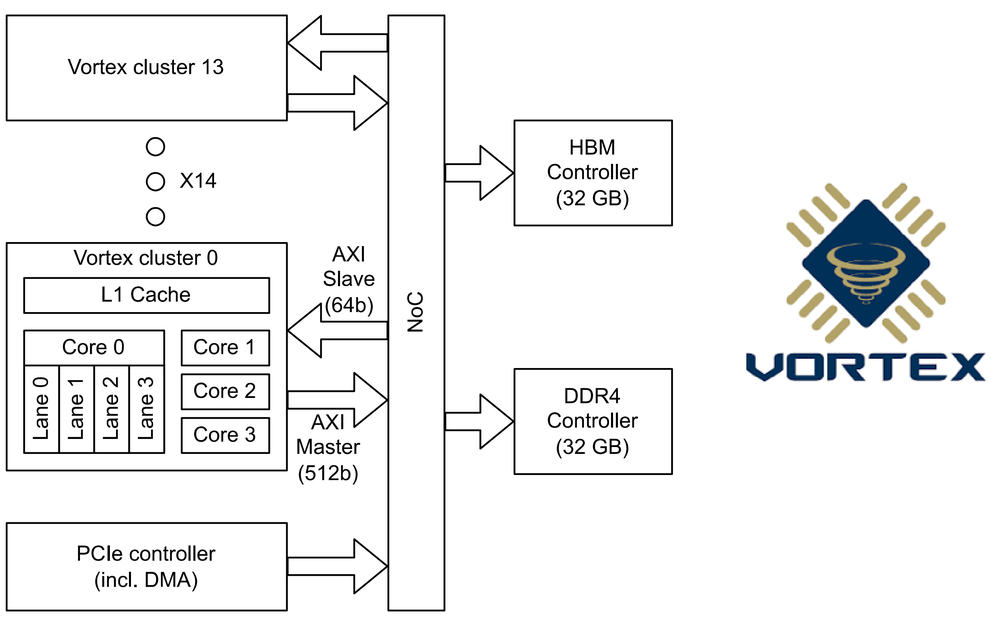
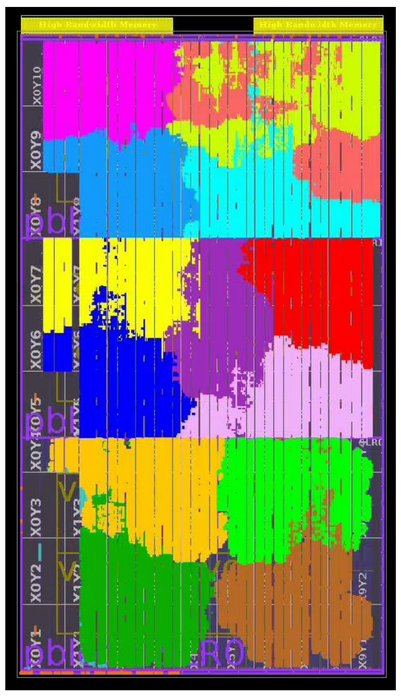


Fig 5. Vortex architecture with 1 to 14 clusters connected to NoC

Early Results



- Mapping up to 56 Vortex cores on Alveo V80
- Up to 224 FMA lanes (Single Precision)
- 4 wavefronts per core
- No SLR crossing
- Max frequency stable at ~300 MHz even with high utilization

- OpenCL driver developed over AMD PCIe QDMA driver

Fig 6. Post-route FPGA floorplan with 14 clusters (colored)

Config	LUTs	FFs	RAM small	RAM big	DSP	Freq	Peak FP32
4 cores	5%	3%	4%	0%	<1%	300 MHz	10 GFLOPS
56 cores	70%	39%	39%	0%	4%	282 MHz	126 GFLOPS
AMD Versal HBM XCV80	2.5 M	-	132 Mb	541 Mb	10.8 K	~800 MHz	17.5 TFLOPS

Fig 7. Implementation results

Future works

- HW support for double precision (FP64) operations
- HPC benchmarks: Linpack and HPCG
- Optimized memory subsystem to exploit HBM bandwidth

E. Guthmuller[1], J. Fereyre[1]
1 **Univ. Grenoble Alpes, CEA, LIST, 38000 Grenoble**