GRVI Phalanx Update:

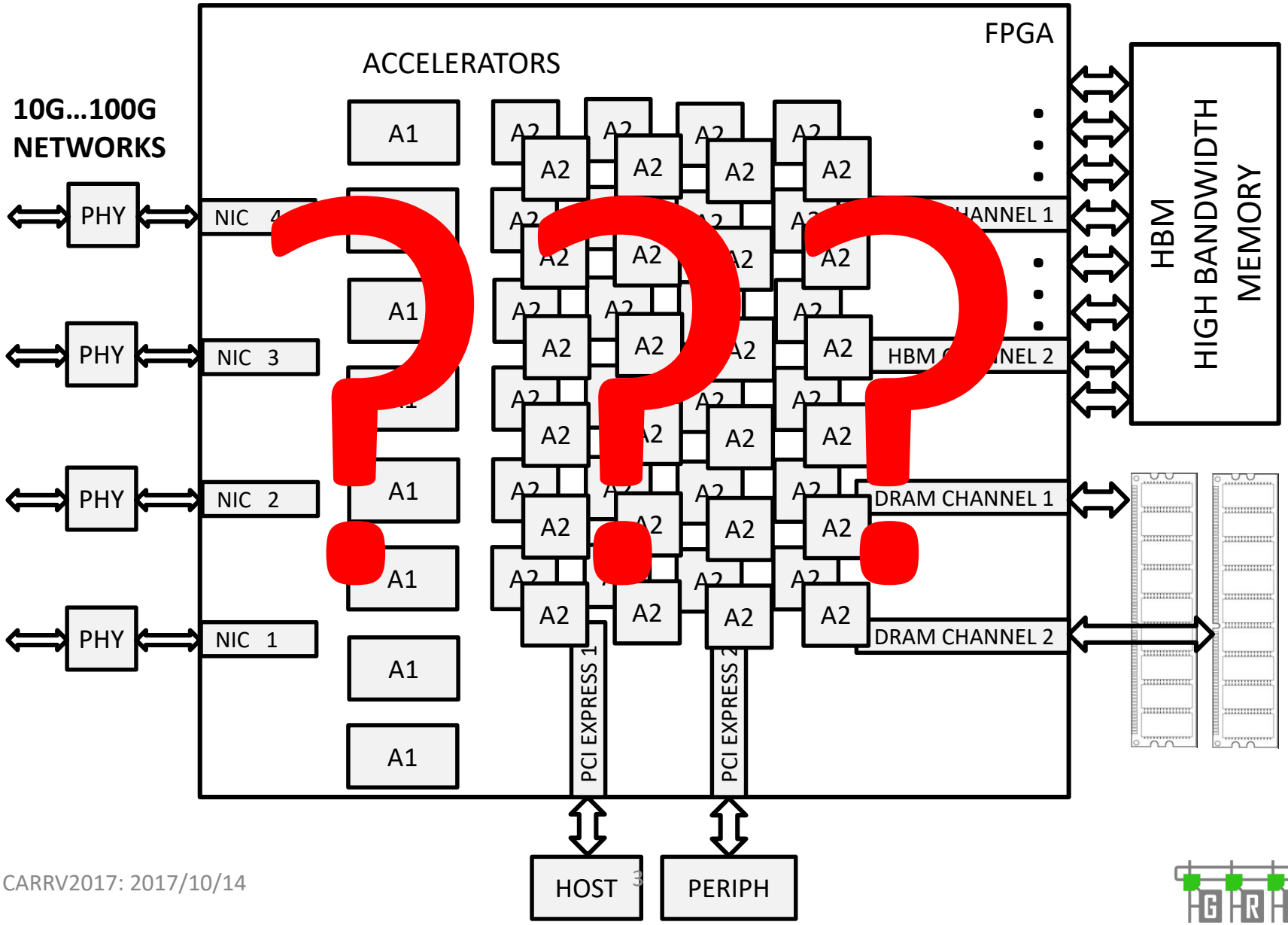
A Massively Parallel RISC-V FPGA Accelerator Framework

Jan Gray | jan@fpga.org | http://fpga.org

FPGA Datacenter Accelerators Are *Almost* Mainstream

- Catapult v2. Intel += Altera. OpenPOWER CAPI. *AWS F1. Baidu. Alibaba. Huawei …*
- FPGAs as computers
  - Massively parallel, customized, connected, versatile
  - High throughput, low latency, low energy
- Great, except for two challenges
  - Software: C++ workload → ??? → FPGA accelerator?
  - Hardware: “tape out” a complex SoC daily?

Hardware Challenge



GRVI Phalanx: FPGA Accelerator Framework

- For *software-first* accelerators:
  - Run parallel software on 100s of soft processors
  - Add custom logic as needed
  - = More 5 second recompiles, fewer 5 hour PARs
- **GRVI:** FPGA-efficient RISC-V RV32I soft CPU
- **Phalanx:** processor/accelerator fabric
  - Many clusters of PEs, RAMs, accelerators, I/O
  - Message passing in a PGAS across a …
- **Hoplite NoC:** FPGA-optimal fast/wide 2D torus

Why RISC-V?

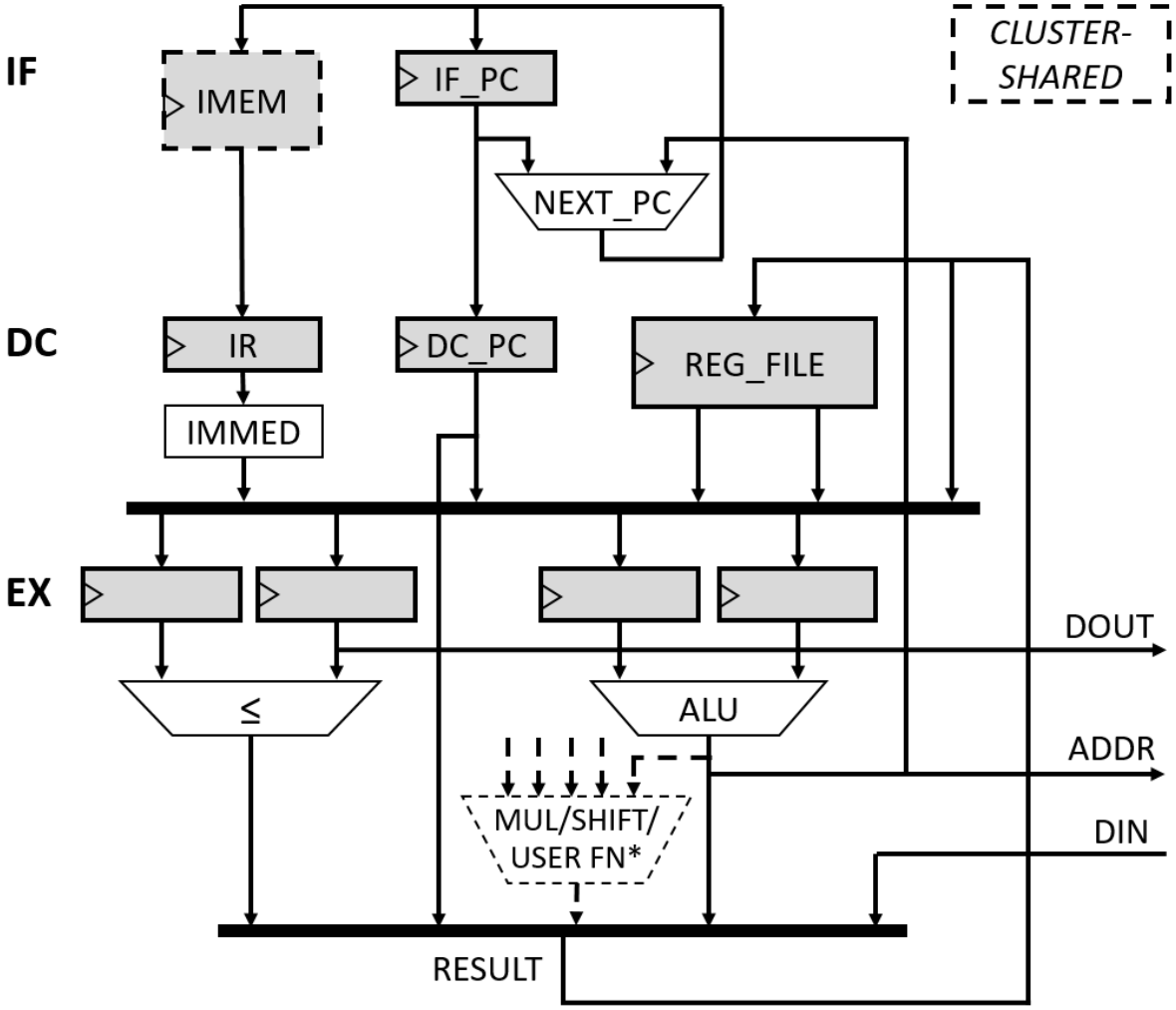
- Open ISA, welcomes innovation
- Comprehensive infrastructure and ecosystem
  - **Specs**, **tests**, simulators, cores, **compilers**, **libs**, FOSS
- As with LLVM, research will accrue to RISC-V

- *Its **simple** ISA allows an efficient FPGA soft CPU*

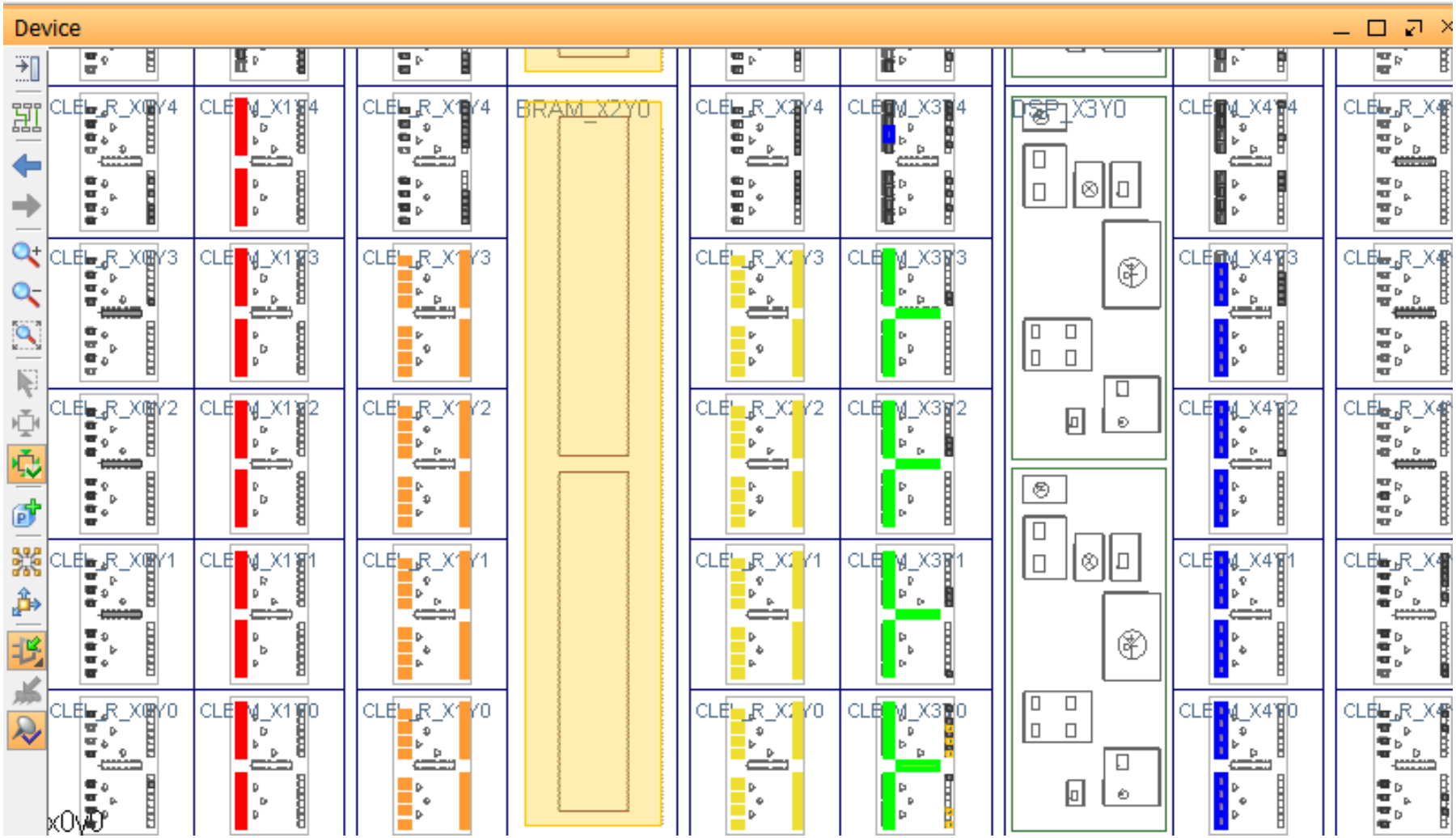
GRVI: Austere RISC-V Processing Element

- Simpler, smaller processors → more processors → more task and memory parallelism
- GRVI core
  - RV32I, *minus* CSRs, exceptions, *plus* `mul*`, `lr/sc`
  - 3 stage pipeline (fetch, decode, execute)
  - 2 cycle loads; 3 cycle taken branches/jumps
  - **Typically 320 LUTs** @ 375 MHz ≈ **0.7 MIPS/LUT**

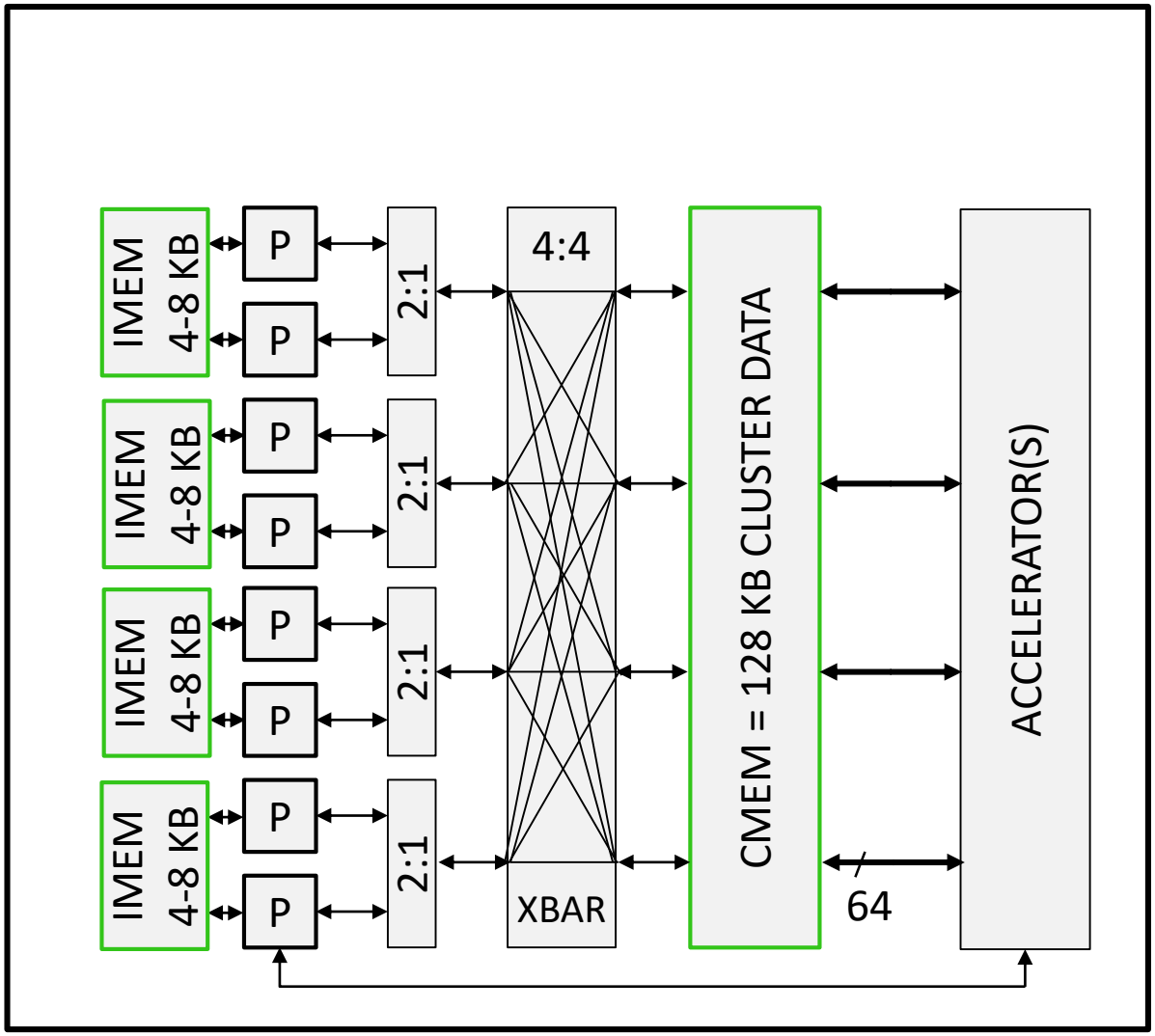
GRVI RV32I Microarchitecture



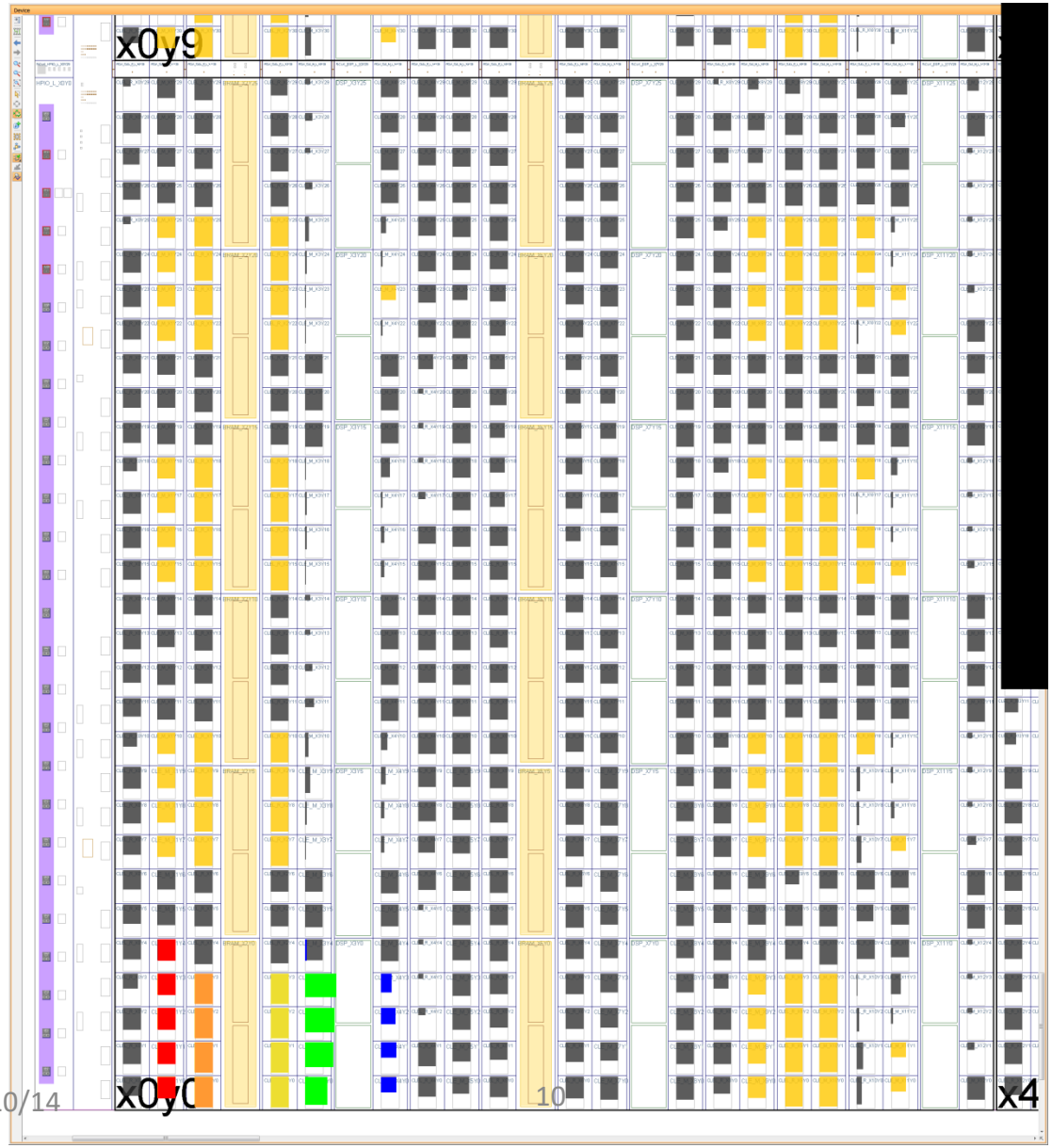
GRVI RV32I Datapath: ~250 LUTs



GRVI Cluster:
0-8 PEs + 32-256 KB Shared Memory



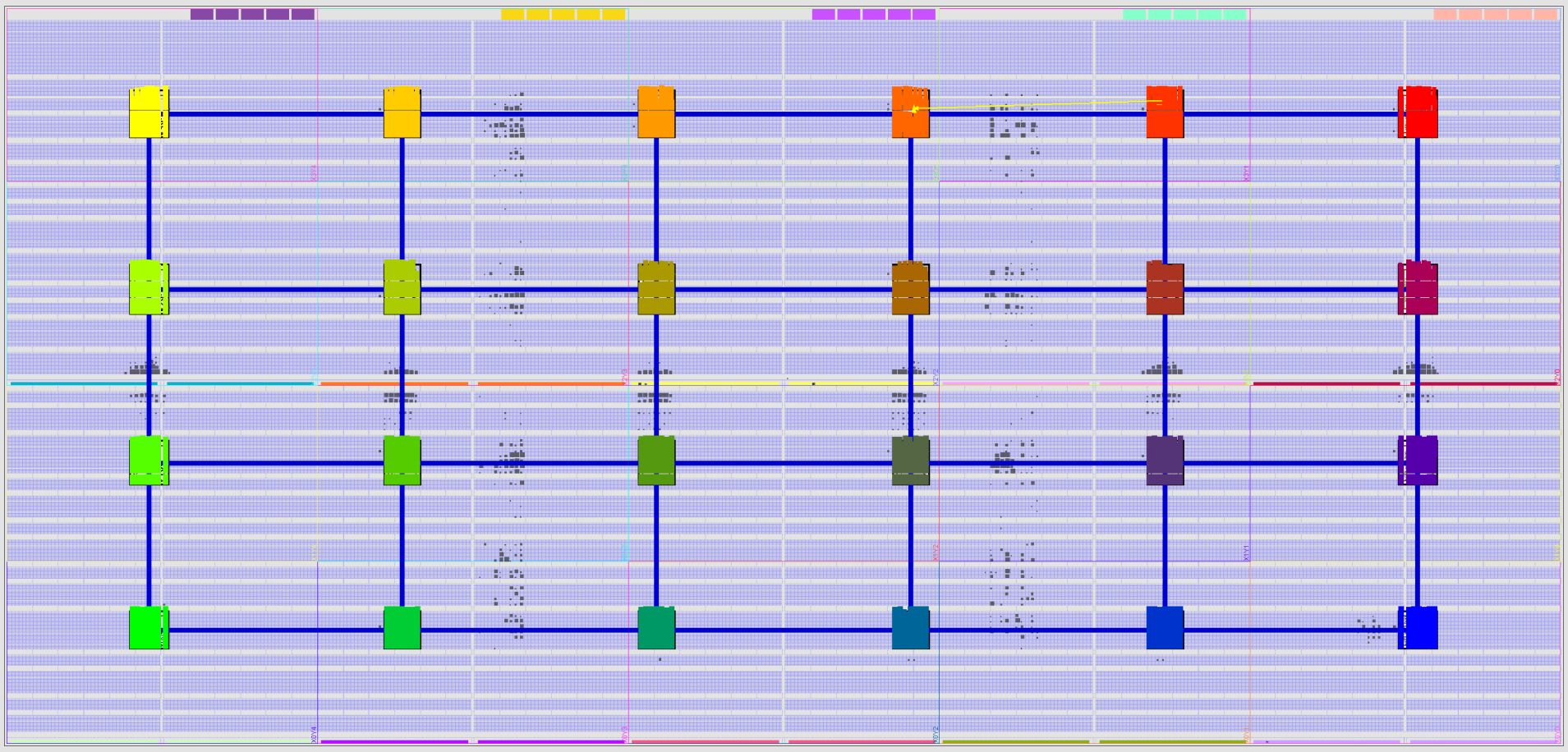
GRVI Cluster Tile: ~3500 LUTs



Composing Clusters with Message Passing on a Hoplite NoC

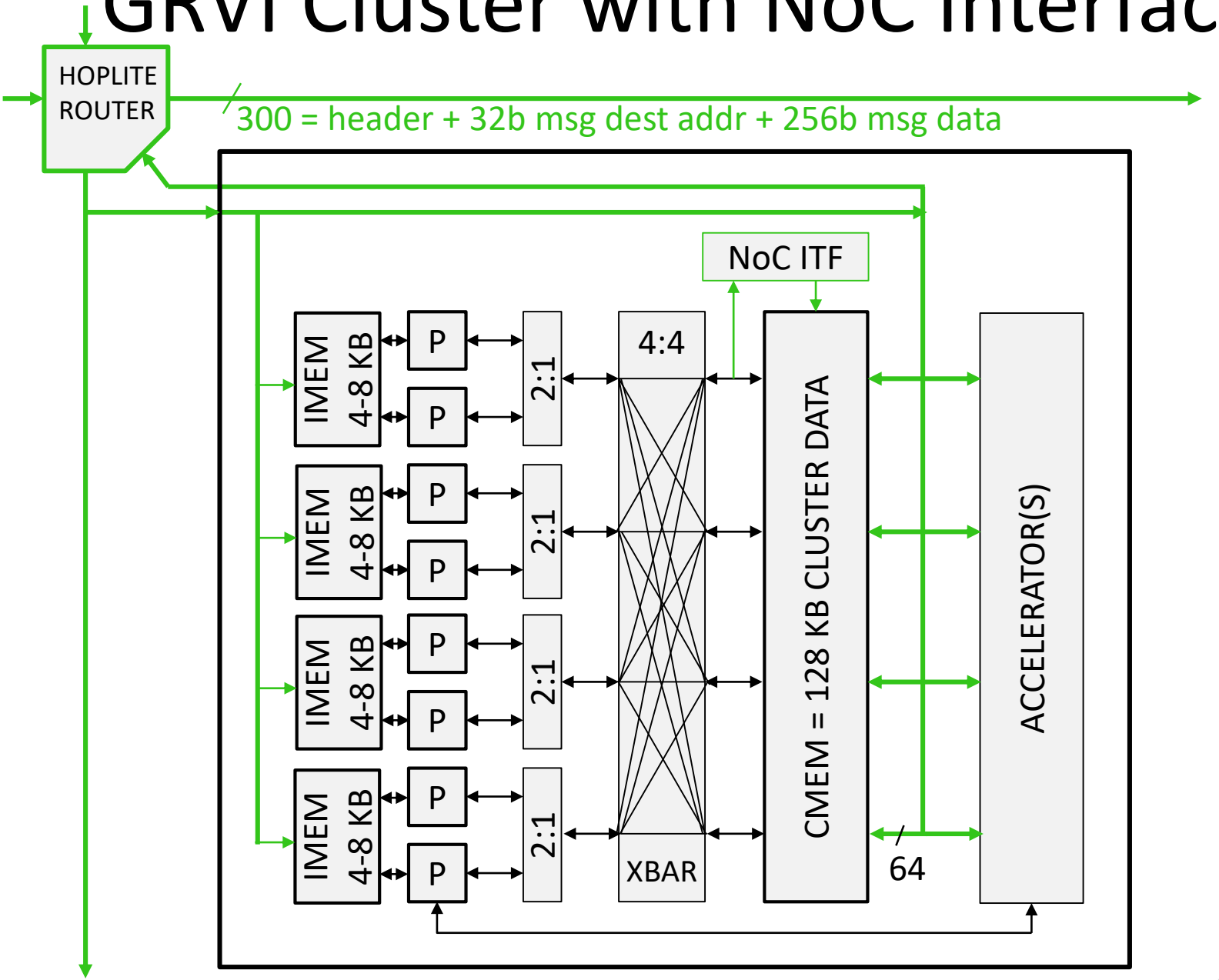
- Hoplite: rethink FPGA NoC router architecture
  - No segmentation/flits, VCs, buffering, credits
  - Unidirectional rings
  - Deflecting dimension order routing of whole msgs
  - Simple; frugal; wide; fast: 1-400 Gbps/link
- **1%** area×delay of FPGA-tuned VC flit routers

Example Hoplite NoC

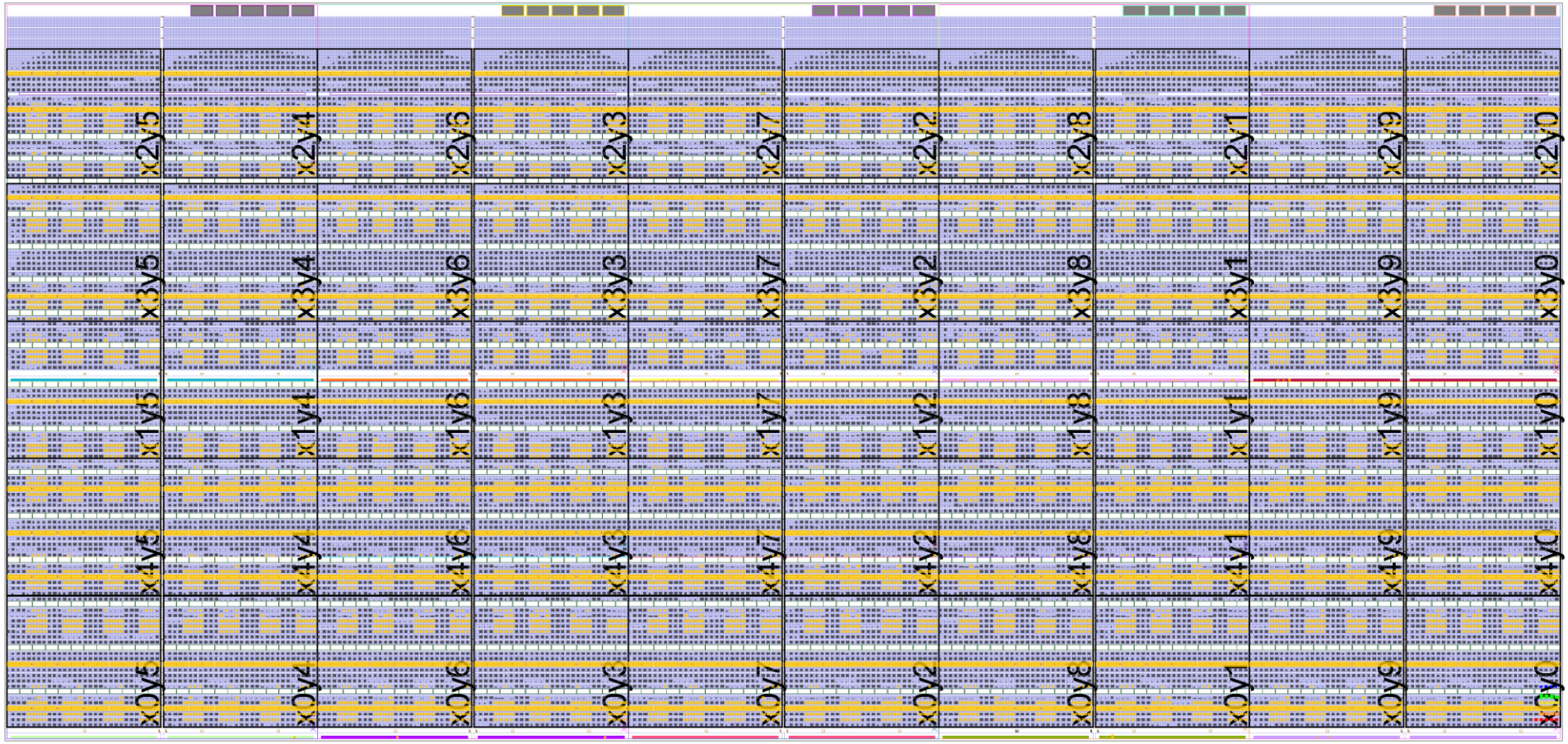


256b links @ 400 MHz = 100 Gb/s links; <3% of FPGA

GRVI Cluster with NoC Interfaces



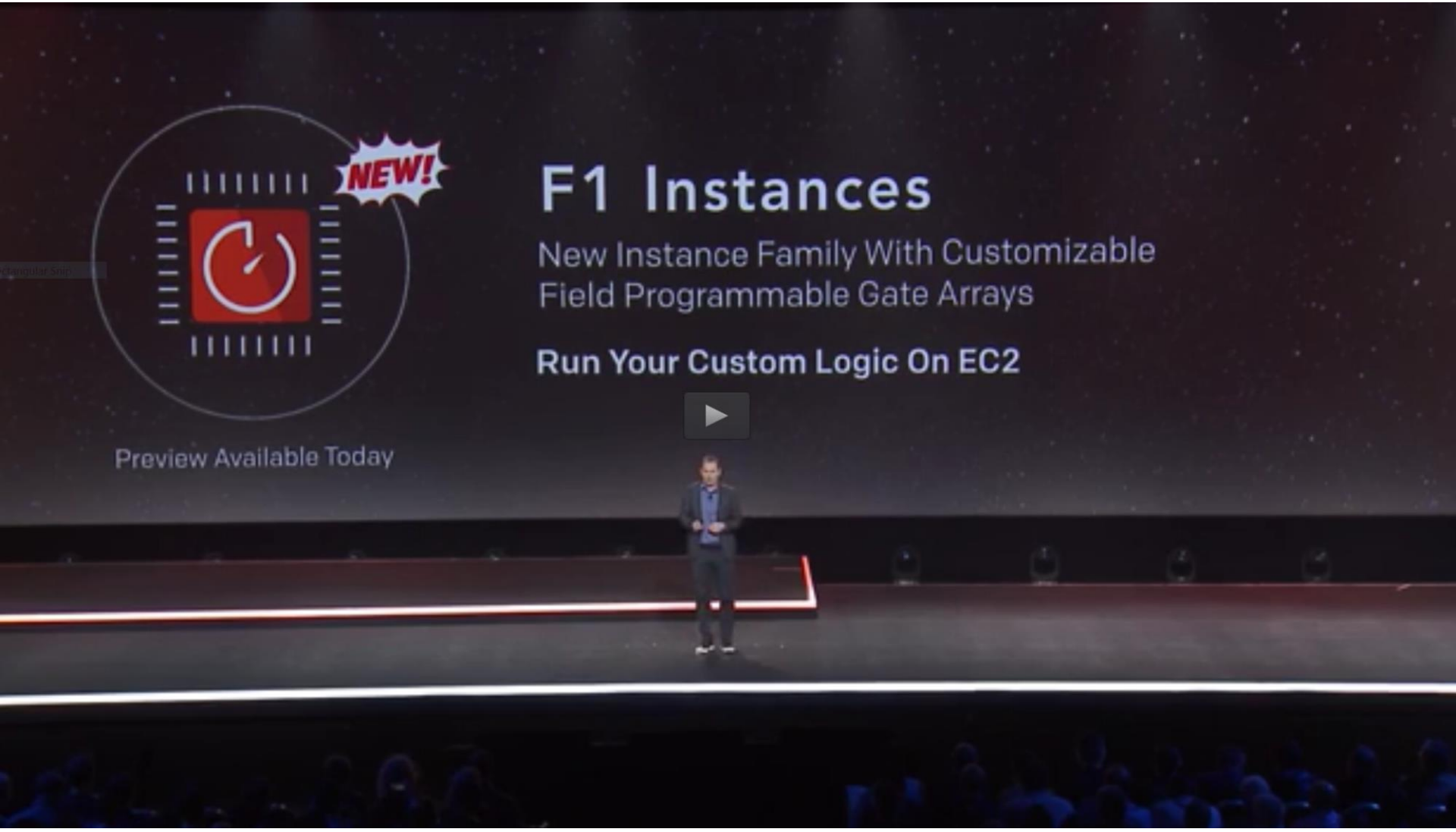
10×5 Clusters × 8 GRVI PEs = 400 GRVI Phalanx (KU040, 12/2015)



Parallel Programming Models?

- *Small* kernels, local or PGAS shared memory, message passing, memcpy/RDMA DRAM
- Current: multithreaded C++ w/ message passing
  - Uses GCC for RISC-V RV32IMA. ***Thank you!***
- Future: OpenCL, KPNs, P4, …
  - Accelerated with custom FUs, AXI cores, RAMs

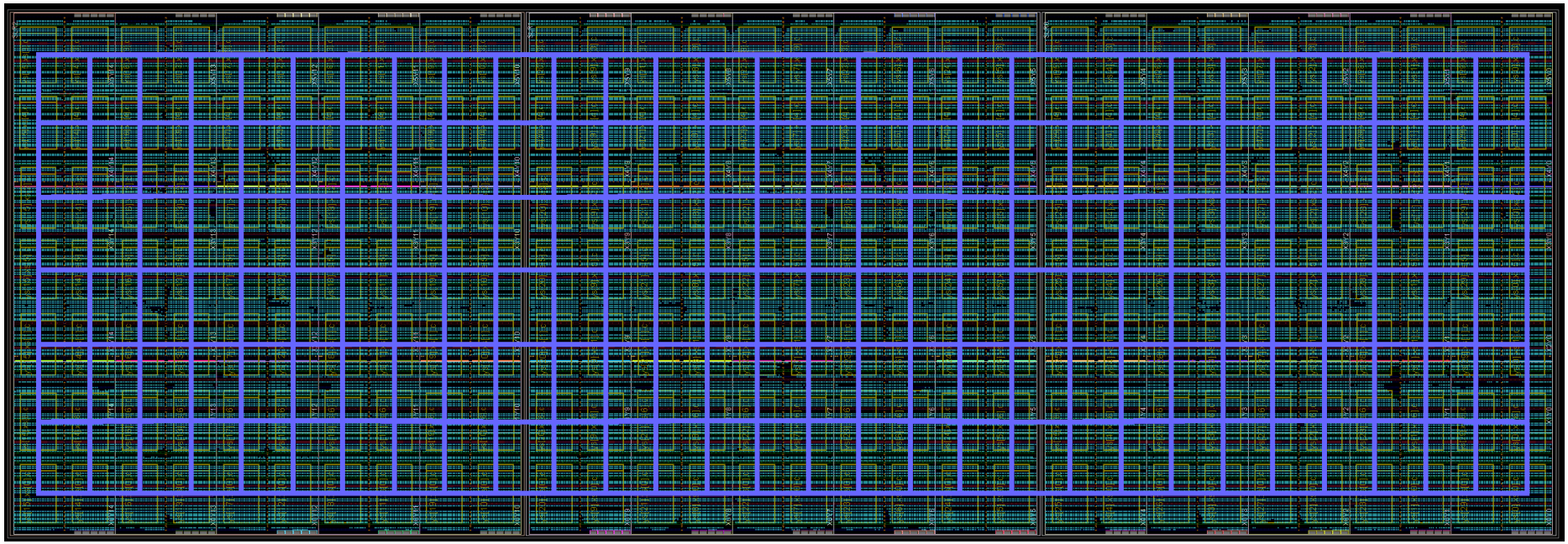
11/30/16: Amazon AWS EC2 F1!



F1's UltraScale+ XCVU9P FPGAs

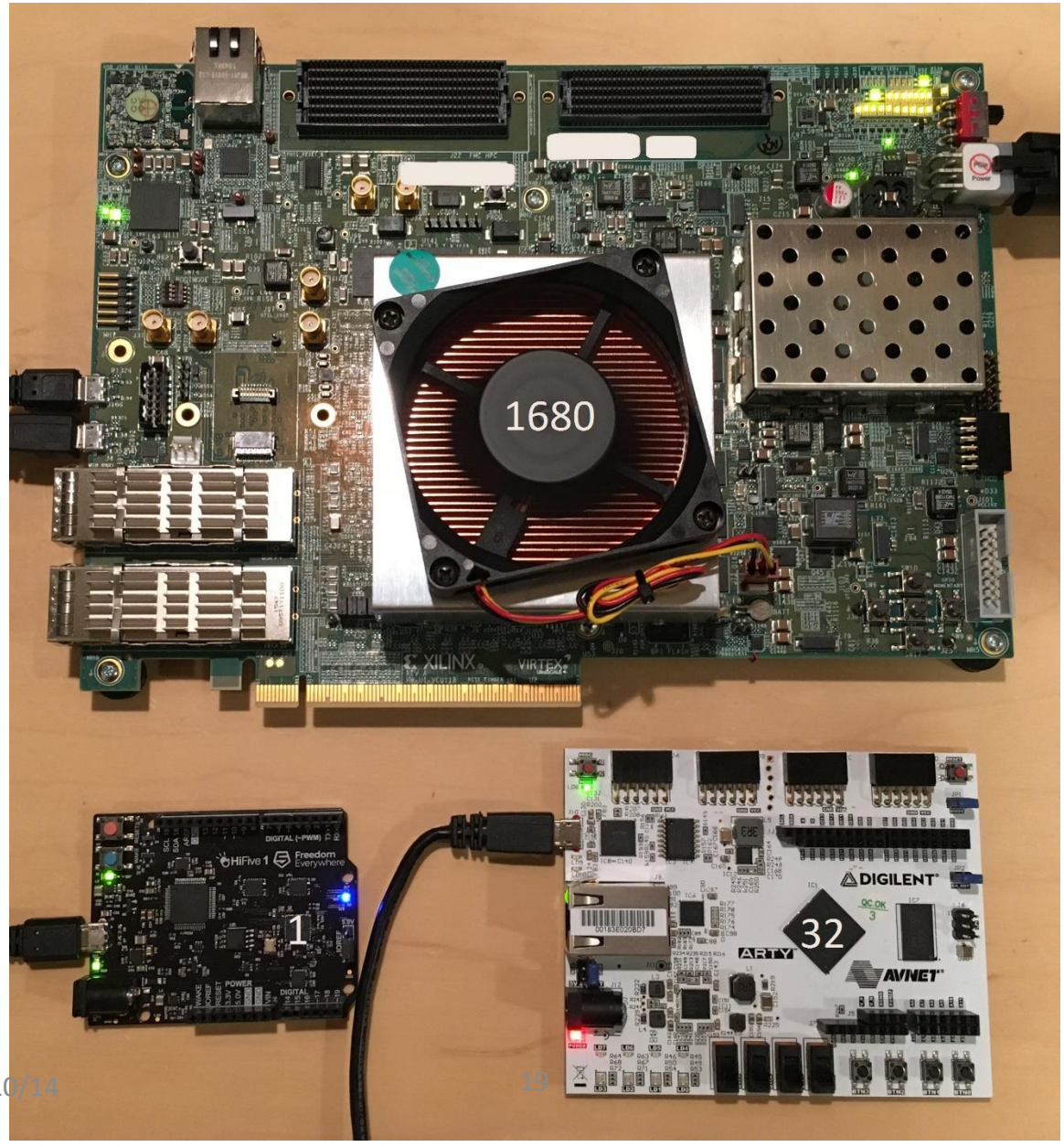
- **1.2 M** 6-LUTs
- **2160** 36 Kb BRAMs (8 MB)
- **960** 288 Kb URAMs (30 MB)
- **6840** DSPs

1680 RISC-Vs, 26 MB CMEM (VU9P, 12/2016)



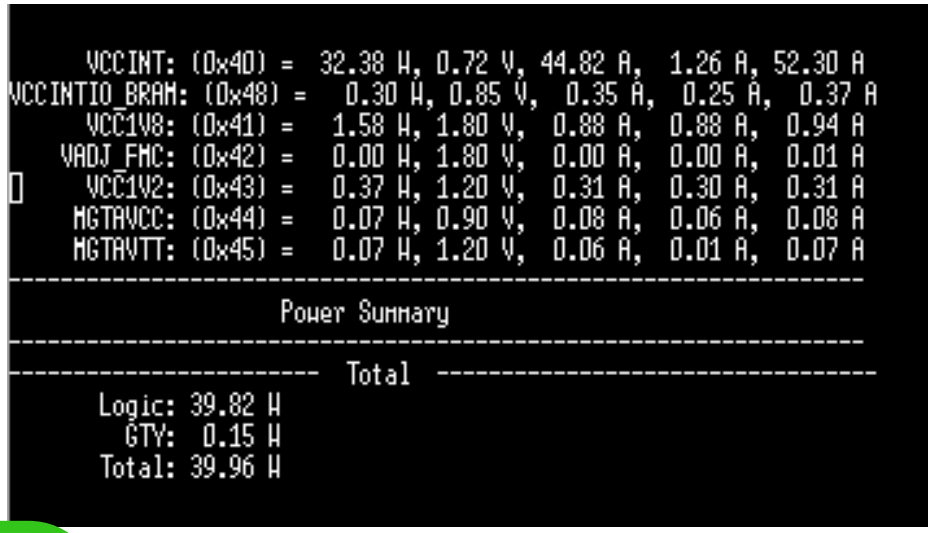
- 30×7 clusters of { 8 GRVI, 128 KB CMEM, router }
- First kilocore RISC-V, and the most 32b RISC cores on a chip in any technology

1, 32, 1680 RISC-Vs



1680 Core GRVI Phalanx Statistics

Resource	Use	Util. %
Logical nets	3.2 M	-
Routable nets	1.8 M	-
CLB LUTs	**795 K**	**67.2%**
CLB registers	744 K	31.5%
BRAM	840	38.9%
URAM	**840**	**87.5%**
DSP	840	12.3%

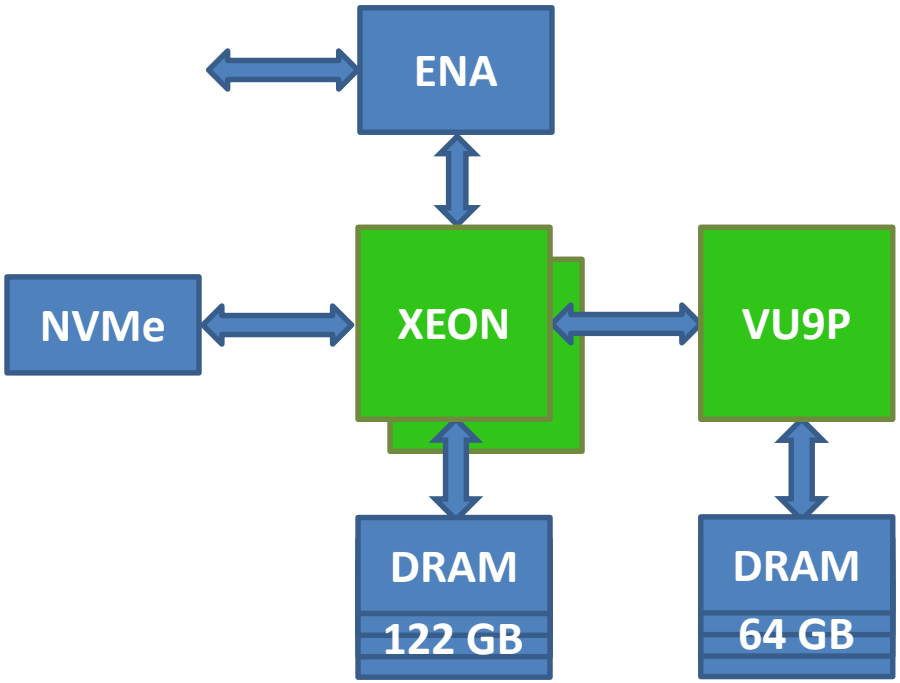


Frequency	250 MHz
Peak MIPS	420 GIPS
CRAM Bandwidth	2.5 TB/s
NoC Bisection BW	900 Gb/s
Power (INA226)	31-40 W
Power/Core	18-24 mW/core
MAX VCU118 Temp	44C

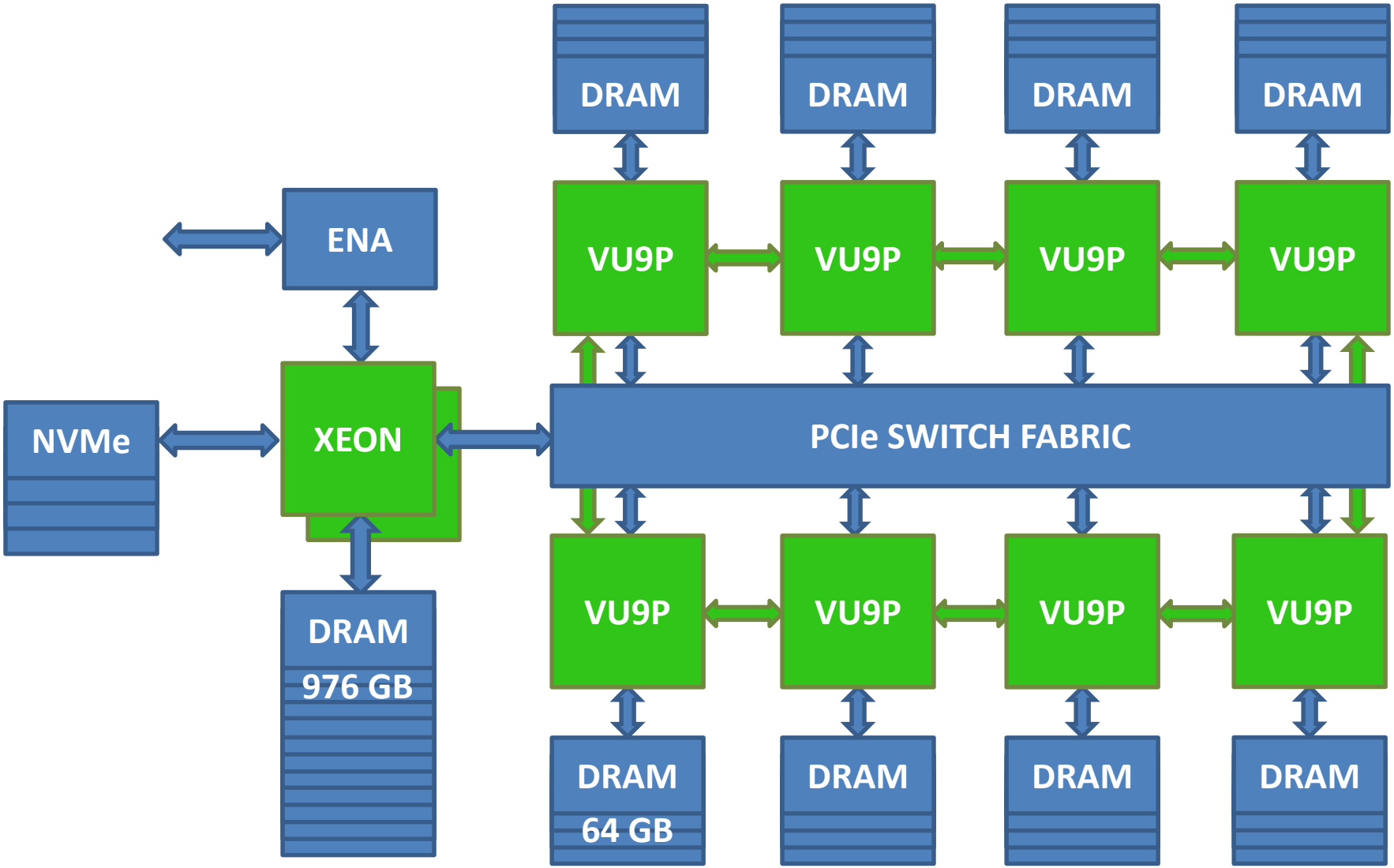
Vivado	2016.4 / ES1
Max RAM use	~32 GB
Flat build time	11 hours
Tools bugs	0

>1000 BRAMs + 6000 DSPs available for accelerators

Amazon F1.2xlarge Instance

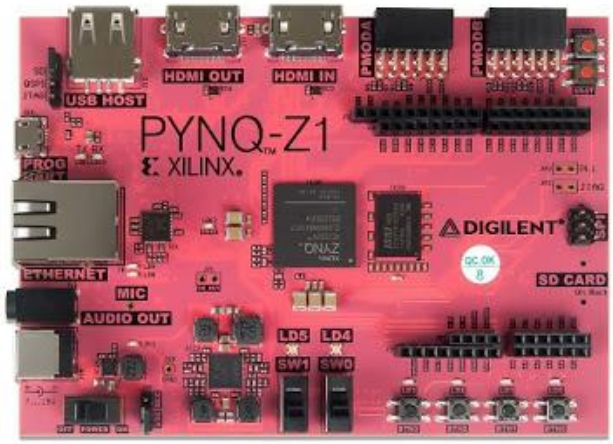


Amazon F1.16xlarge Instance

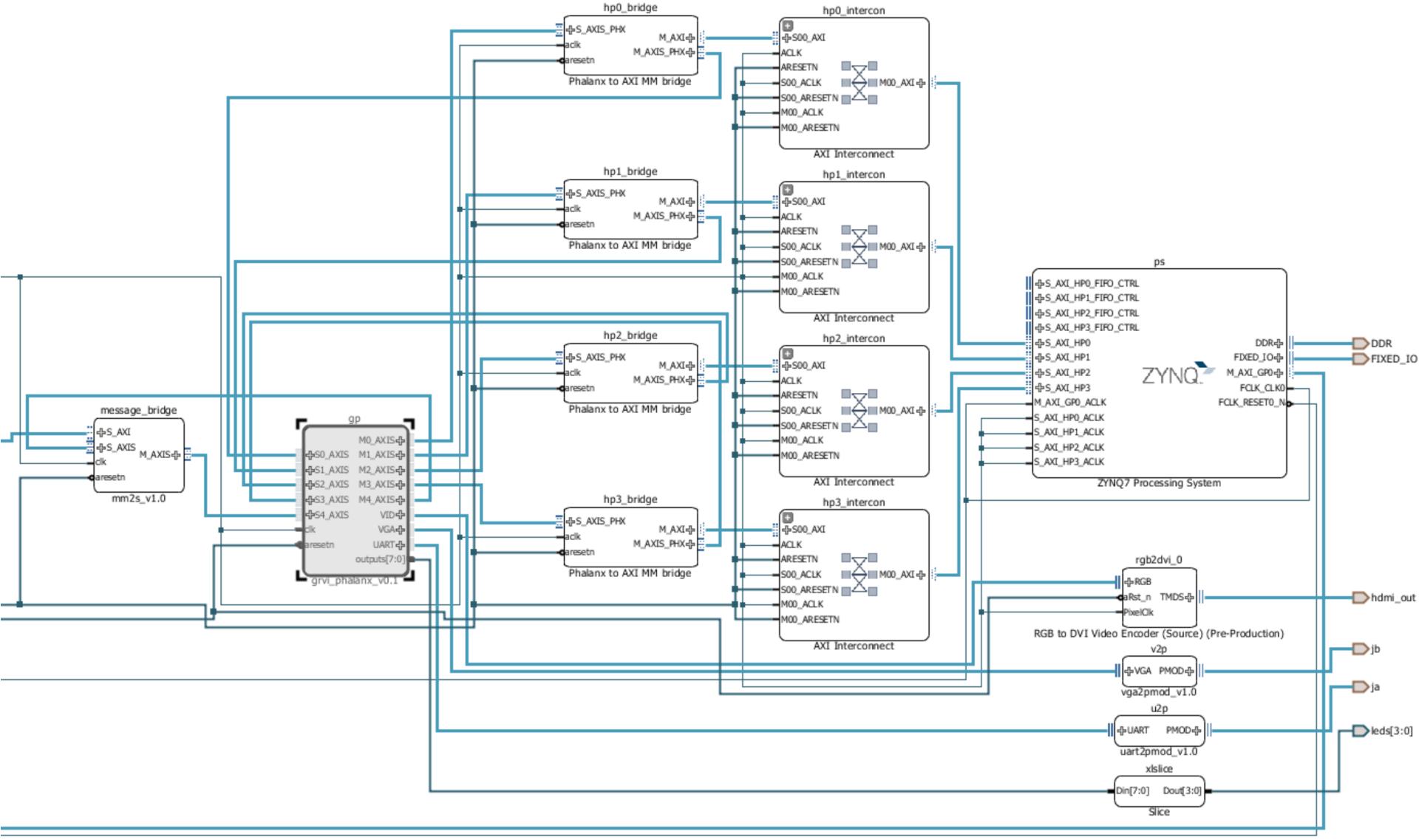


Recent Work

- Bridge Phalanx and AXI4 system interfaces
  - Message passing with host CPUs (x86 or ARM)
  - DRAM channel RDMA request/response messaging
- "SDK" hardware targets
  - 1000-core AWS F1 (<$2/hr)
  - 80-core PYNQ (Z7020) ($65 edu)

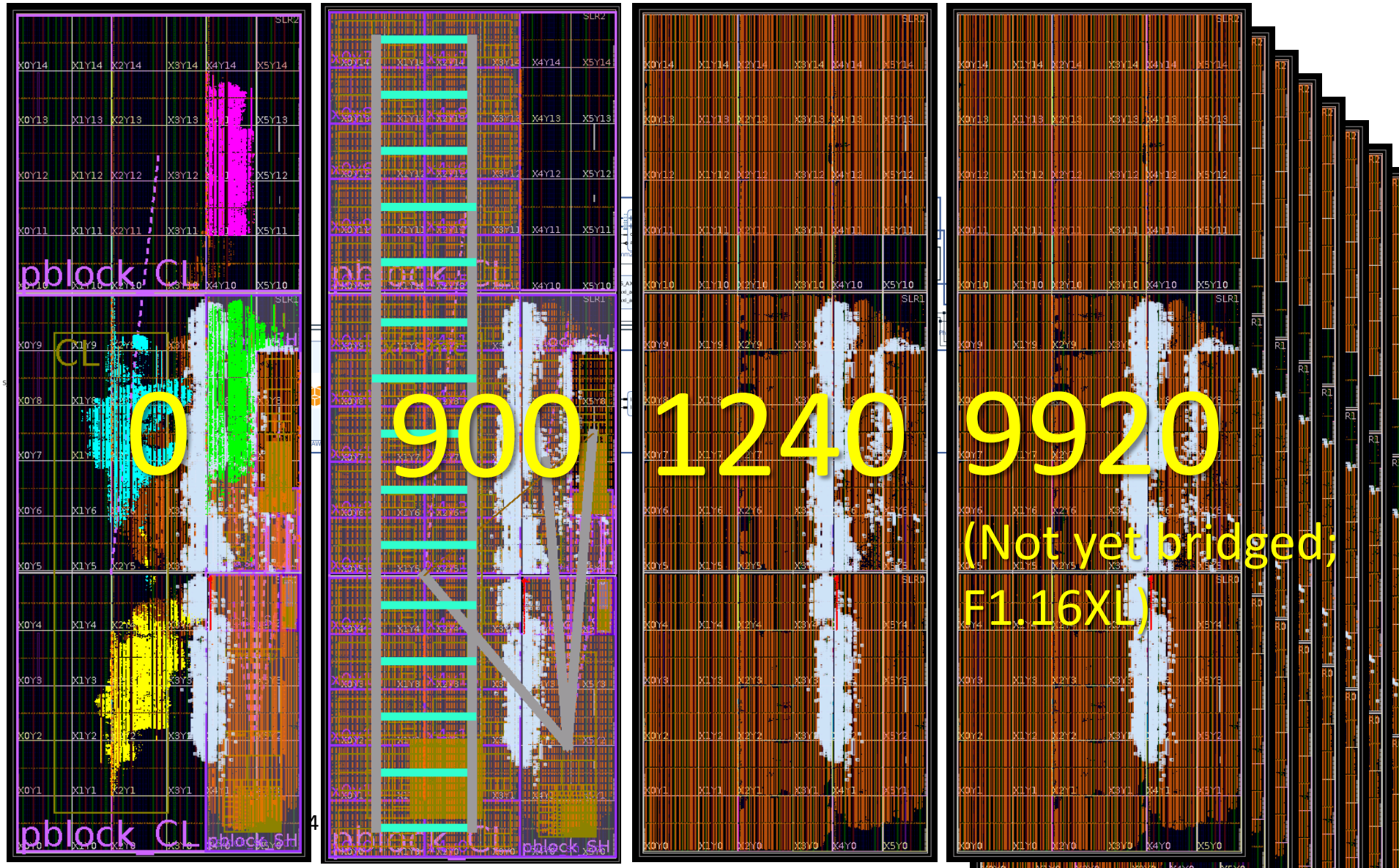


GRVI Phalanx on Zynq with AXI Bridges



PYNQ-Z1 Demo: Parallel Burst DRAM Readback Test: 80 Cores × 2^{28} × 256 B

GRVI Phalanx on AWS F1 (WIP)



4Q17 Work in Progress

- Complete initial F1.2XL and F1.16XL ports
- GRVI Phalanx SDK
  - Specs, libraries, examples, tests
  - As PYNQ Jupyter Python notebooks, bitstreams
  - AMI+AFI in AWS Marketplace
- Full instrumentation – event counters; tracing
- Evaluate porting effort & perf on workloads TBD

In Conclusion

- Enable programmers to access massive reconfigurable / memory parallelism
- Frugal design enables competitive performance
- Value proposition unproven, awaits workloads
- SDK coming soon, enabling parallel RISC-V research and teaching on 80-8,000 core systems

Thank you