2GRVI Phalanx: A Kilocore RISC-V RV64I Processor Cluster Array with HBM2 In a Xilinx VU37P FPGA

Work in Progress Report

Jan Gray | Gray Research LLC | Bellevue, WA | http://fpga.org

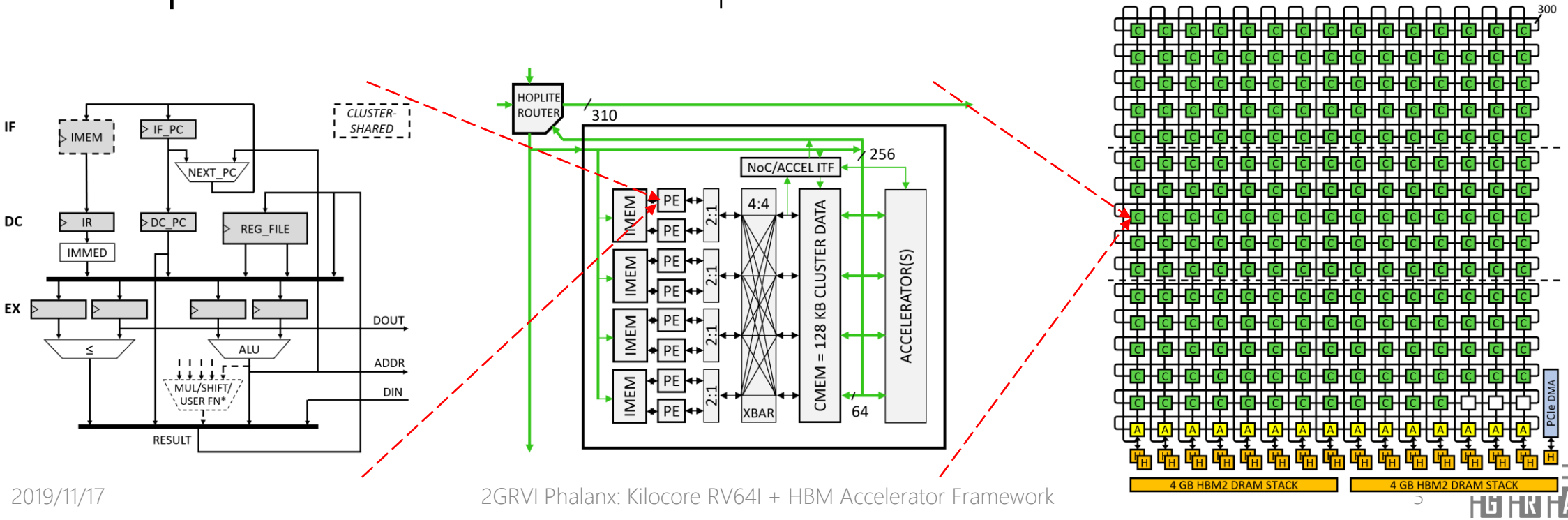
RISC-V

Software-First FPGA Accelerator Design

- *Make it easier* for programmers to exploit spatial fabrics
- Manycore accelerator overlays
  - Run C++ or OpenCL kernels on 100s of soft processors
  - Add custom functions/accelerators/memories to suit
  - More 5 second recompiles, fewer 5 hour place and routes
- Software + overlays = familiar programming experiences, easier ports, rapid iteration, design agility

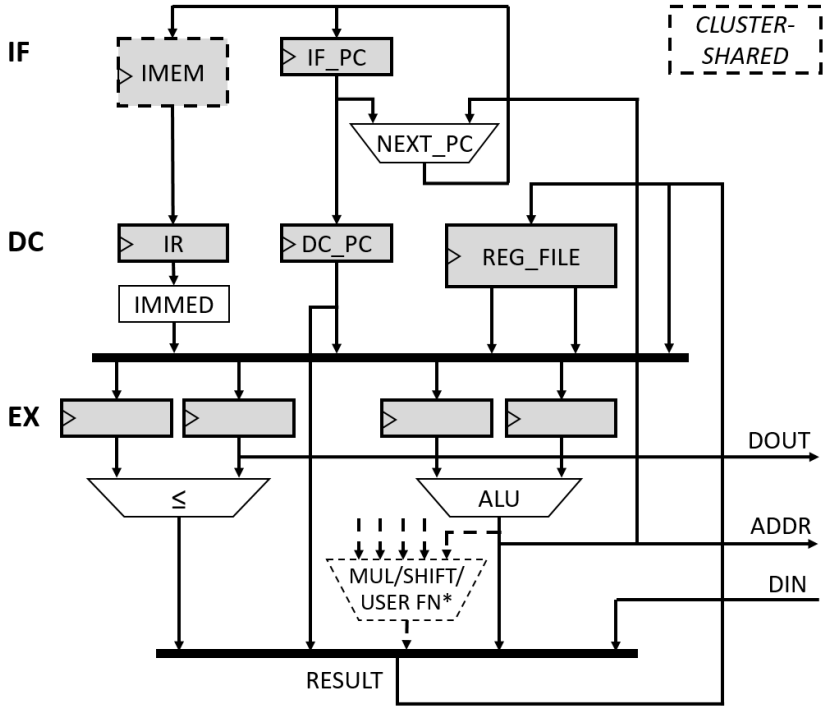
GRVI Phalanx Accelerator Framework

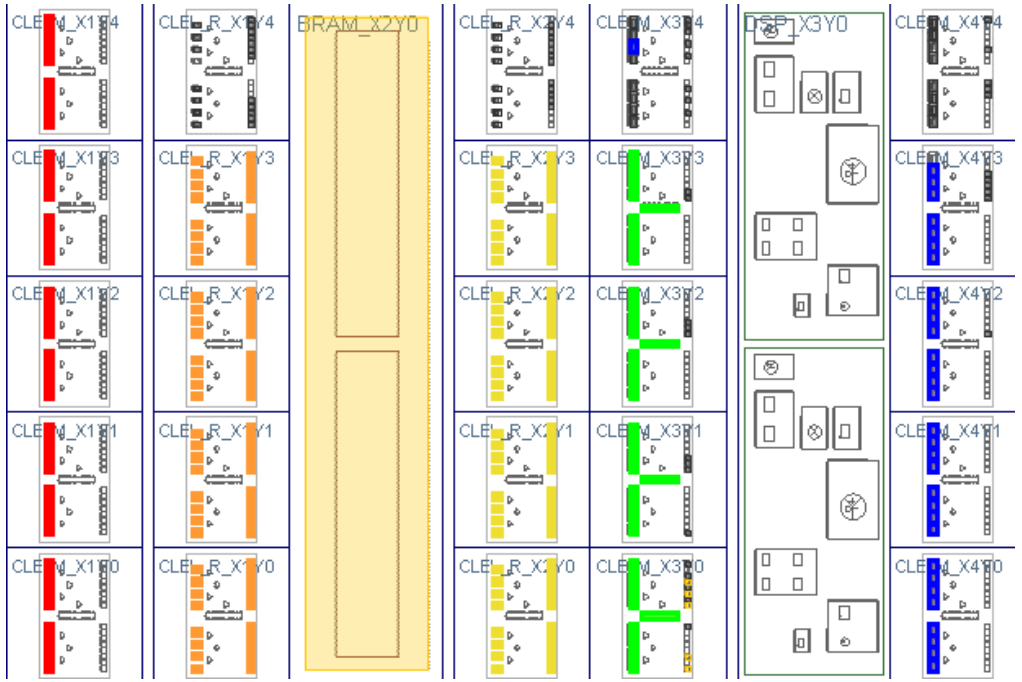
- A processor cluster array overlay
  - **GRVI/2GRVI:** RISC-V processing elements
  - **Phalanx:** fabric of clusters of PEs, memories, accelerators, bridges, IOs
  - **Hoplite:** 2D torus network on chip



GRVI Processing Element

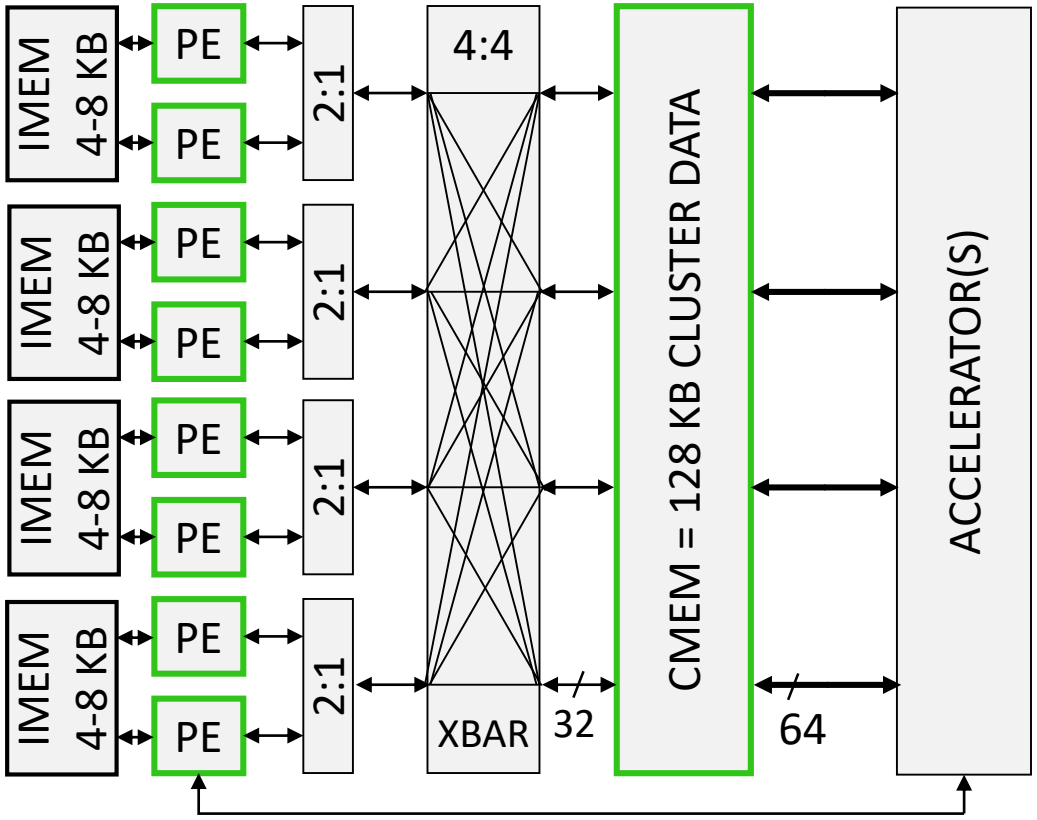
- Simpler PEs → more PEs → greater memory parallelism
- GRVI: austere RISC-V RV32I + mul*/lr/sc

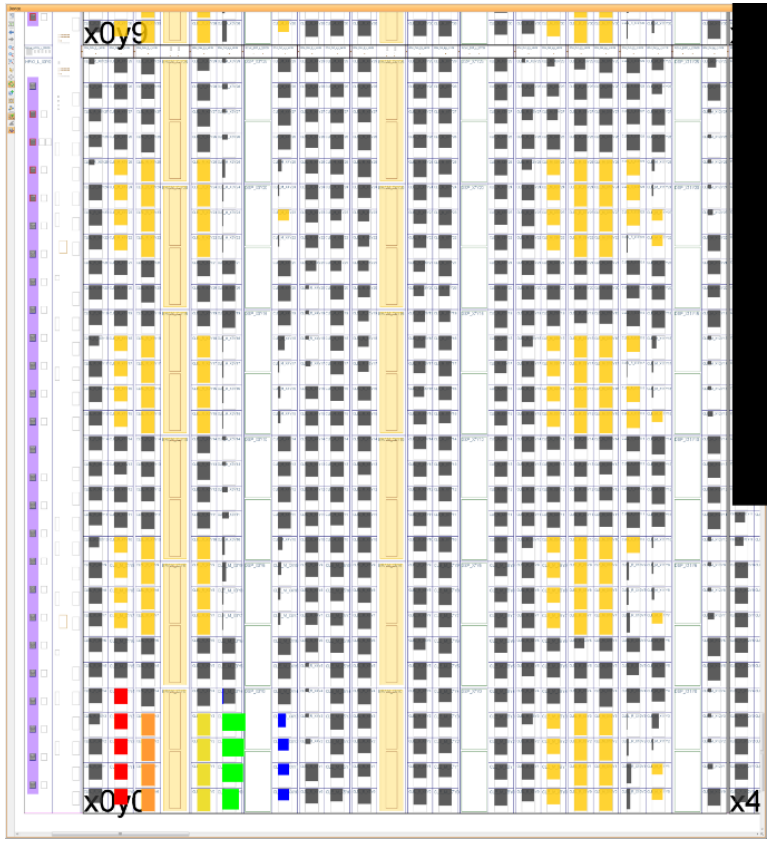




~320 LUTs @ 400 MHz

GRVI Cluster: PEs, Shared Memory, Accelerators

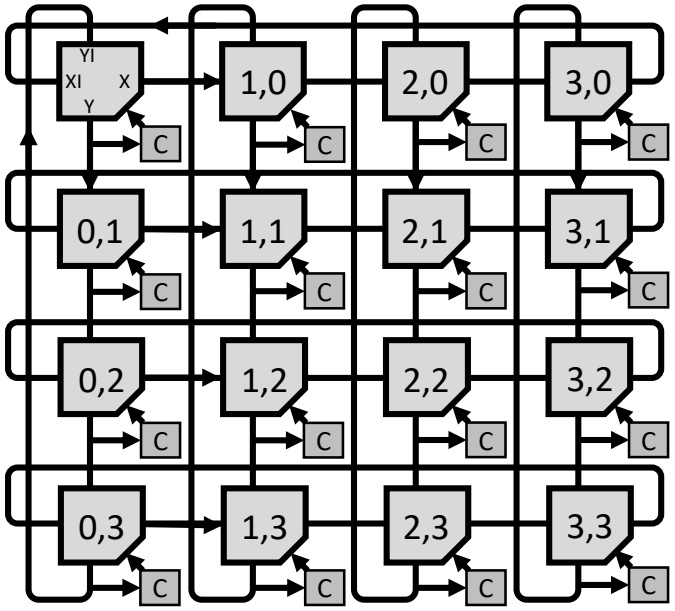


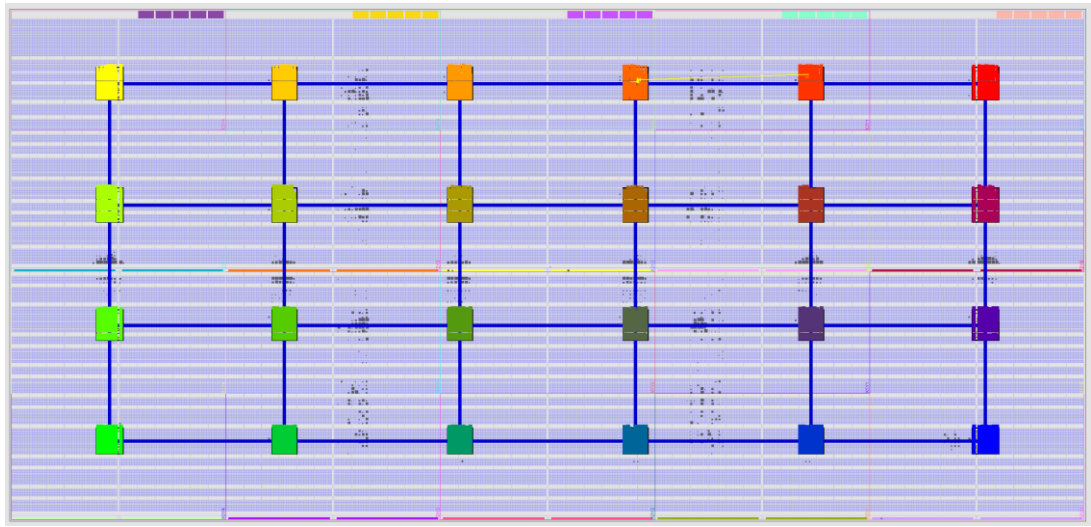


~3500 LUTs

Cluster Composition: Message Passing On a NoC

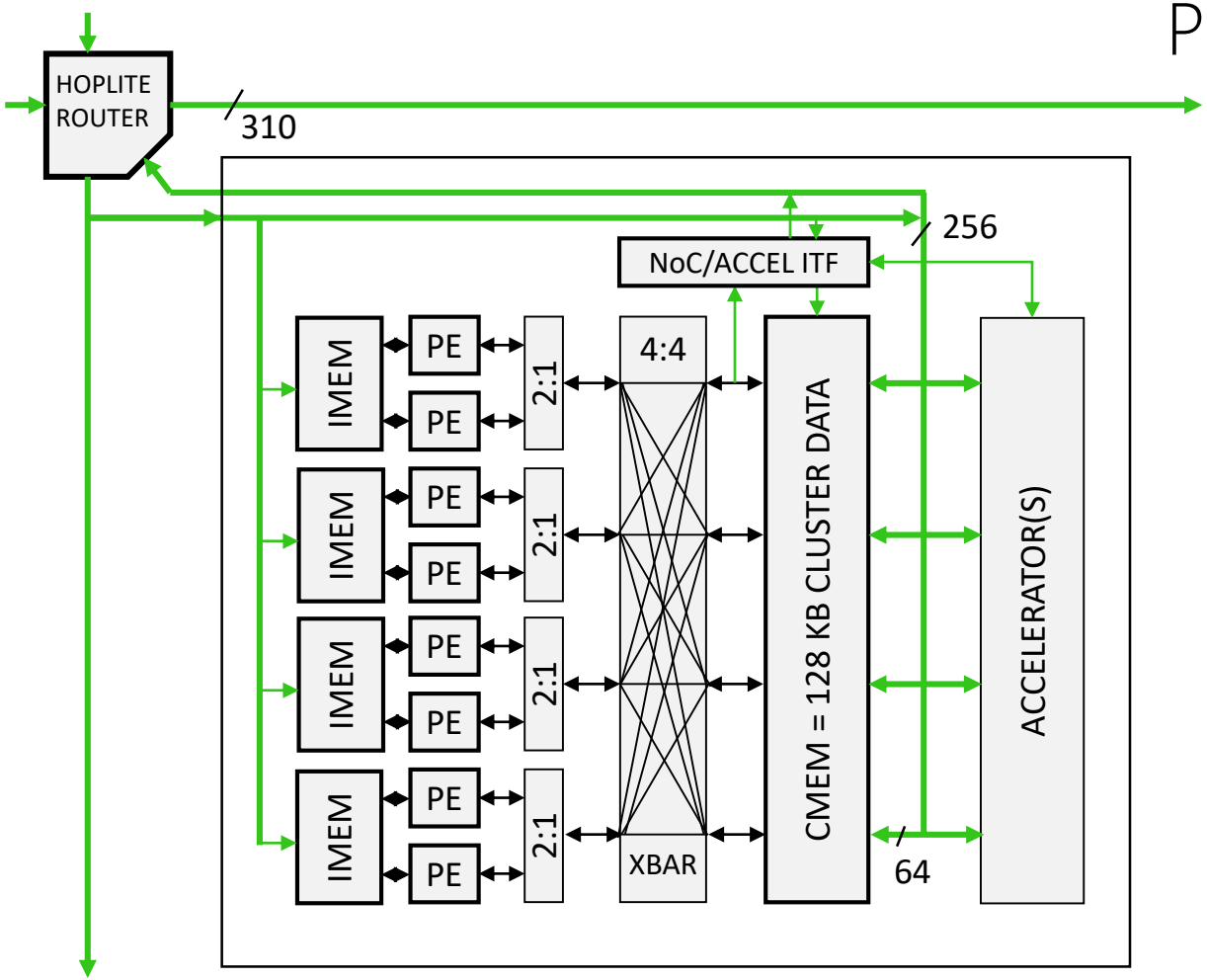
- **Hoplite**: *FPGA-optimal* 2D torus NoC router
  - Single flits, unidirectional rings, deflection routing, multicast; configurable
  - 300b-wide router uses only ~330 LUTs



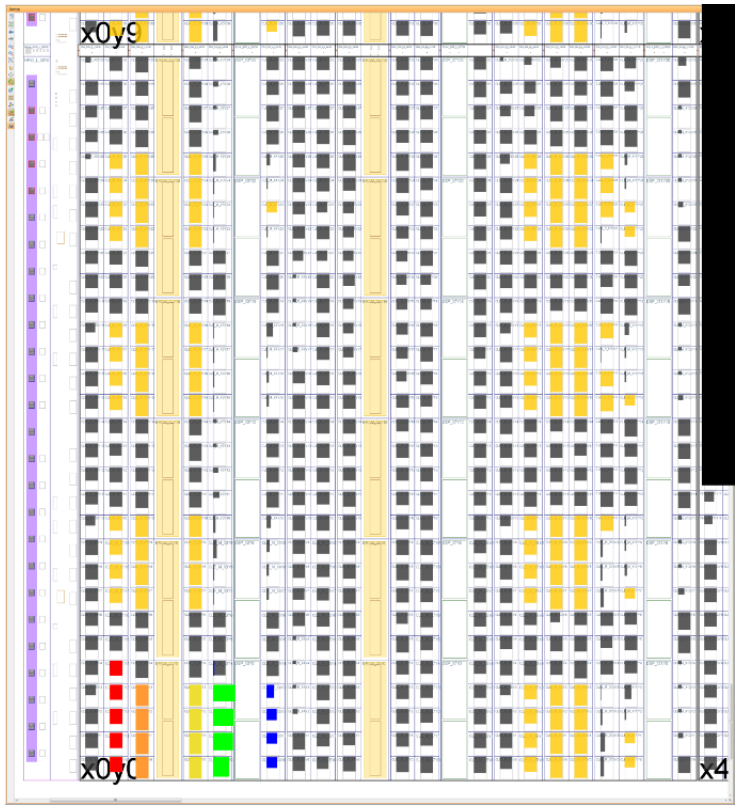


256b @ 400 MHz = 100 Gb/s links

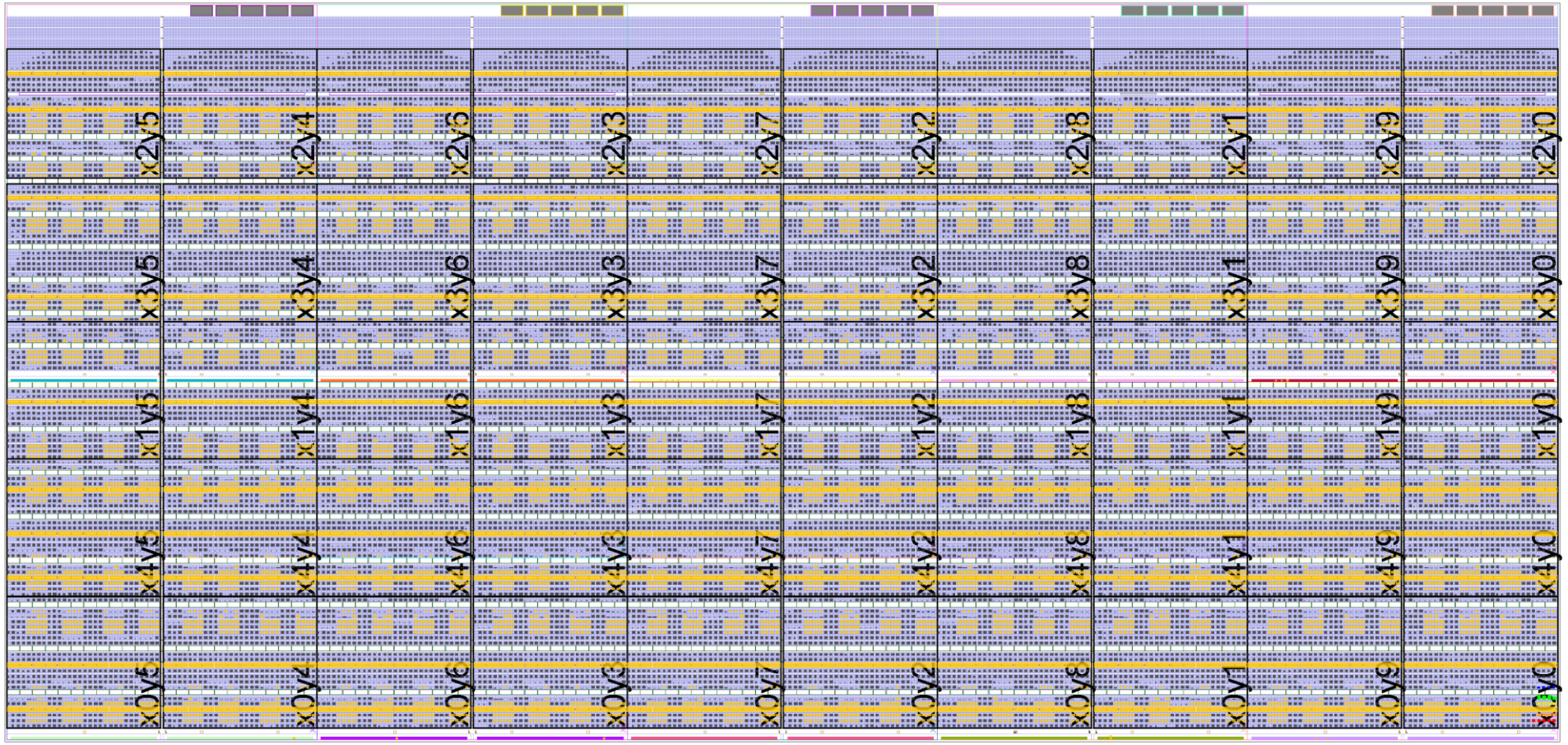
GRVI Cluster: PEs, Memory, Router, Message Passing



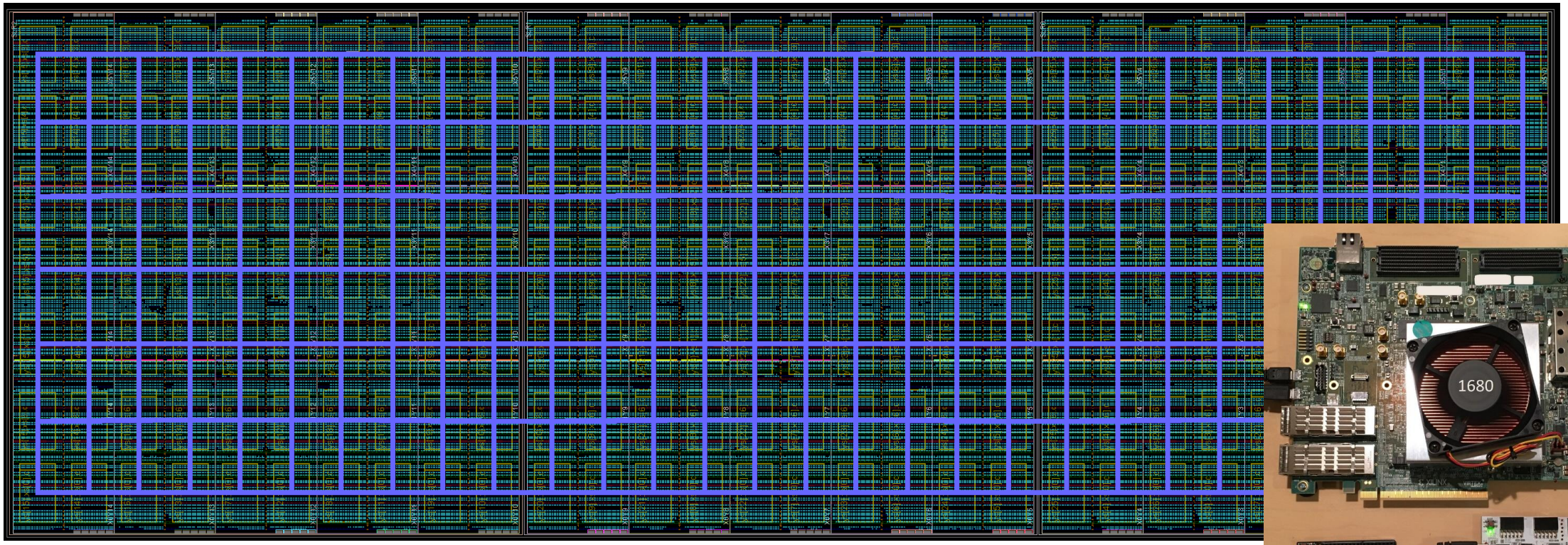
PGAS: { mx:1; my:1; x:4; y:6; addr:20 }
or { dram_addr:40 }



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



30×7 Clusters x 8 PE = 1680 PEs, 26 MB SRAM
(VU9P, 12/2016)



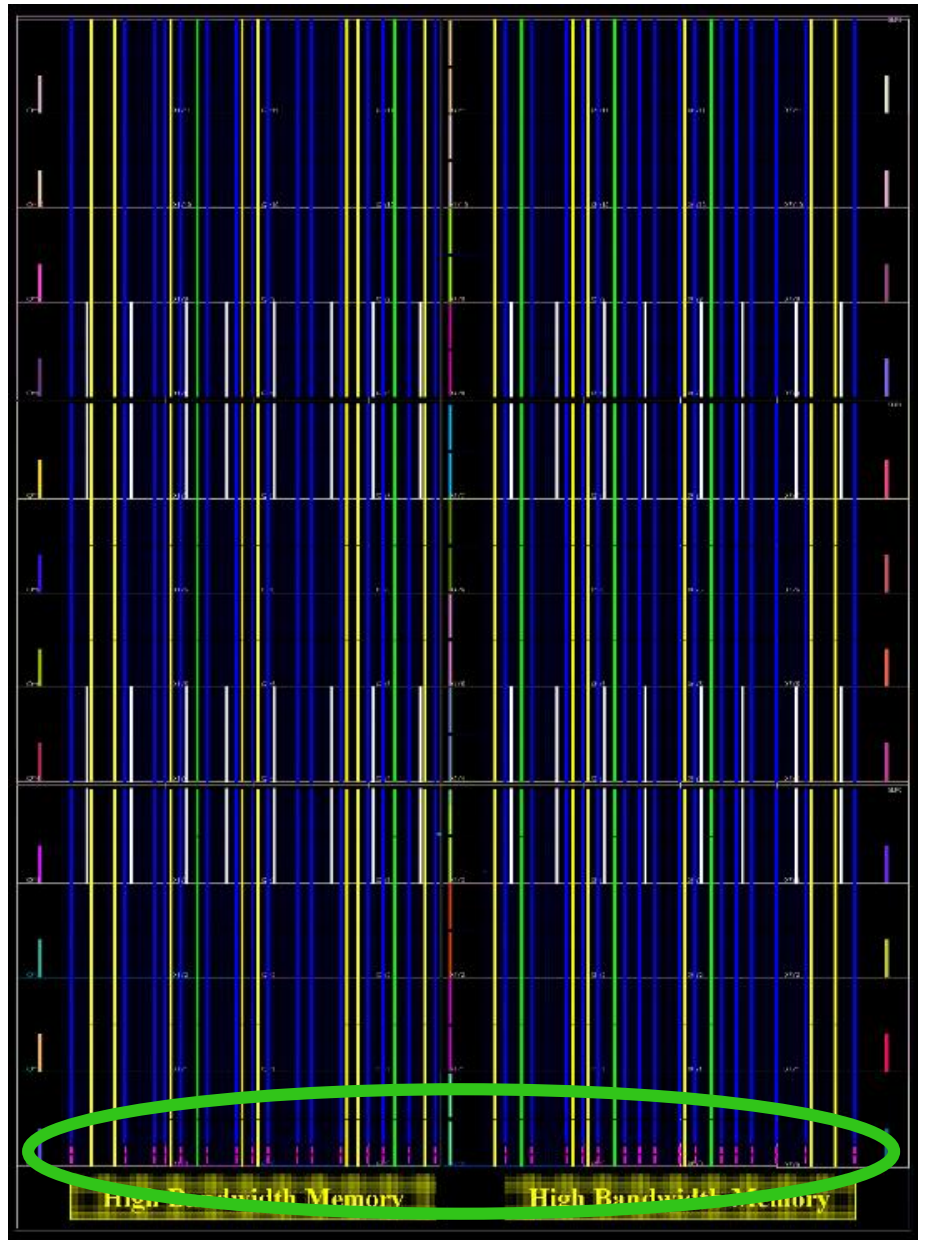
- 400,000 MIPS @ 250 MHz @ 40 W

GRVI Phalanx V1 Shortcomings

- 32b pointers: awkward for big data on AWS F1, OpenCL
- 32b accesses: wastes half of 64b UltraRAMs bandwidth?
- In-order µarch: stall on loads = ~5 cycles
- DDR4 bandwidth << GPU GDDRx/HBM2 bandwidth

UltraScale+ HBM2 FPGAs!

- VU37P w/ two 4 GB HBM2 stacks
- 32 AXI-HBM bridge/controllers
- 32 x 256b x 450 MHz = **460 GB/s**



V2 Redesign for HBM FPGAs

- Latency tolerant 2GRVI RV64I PEs
- 64b cluster datapaths
- 32B/cycle deep pipeline NoC-AXI RDMA bridges
- Double NoC *column rings*

2GRVI – A Simple, Latency Tolerant RV64I PE

- Register file scoreboard: only stall issue on use of a **busy** register
  - Concurrent execution and out of order retirement
  - Example: unrolled block copy – no issue stalls even with 7 cycle memory

```
for (int i = 0; i < N; i += 8) {
    to[i] = from[i];
    to[i+1] = from[i+1];
    to[i+2] = from[i+2];
    to[i+3] = from[i+3];
    to[i+4] = from[i+4];
    to[i+5] = from[i+5];
    to[i+6] = from[i+6];
    to[i+7] = from[i+7];
}
```

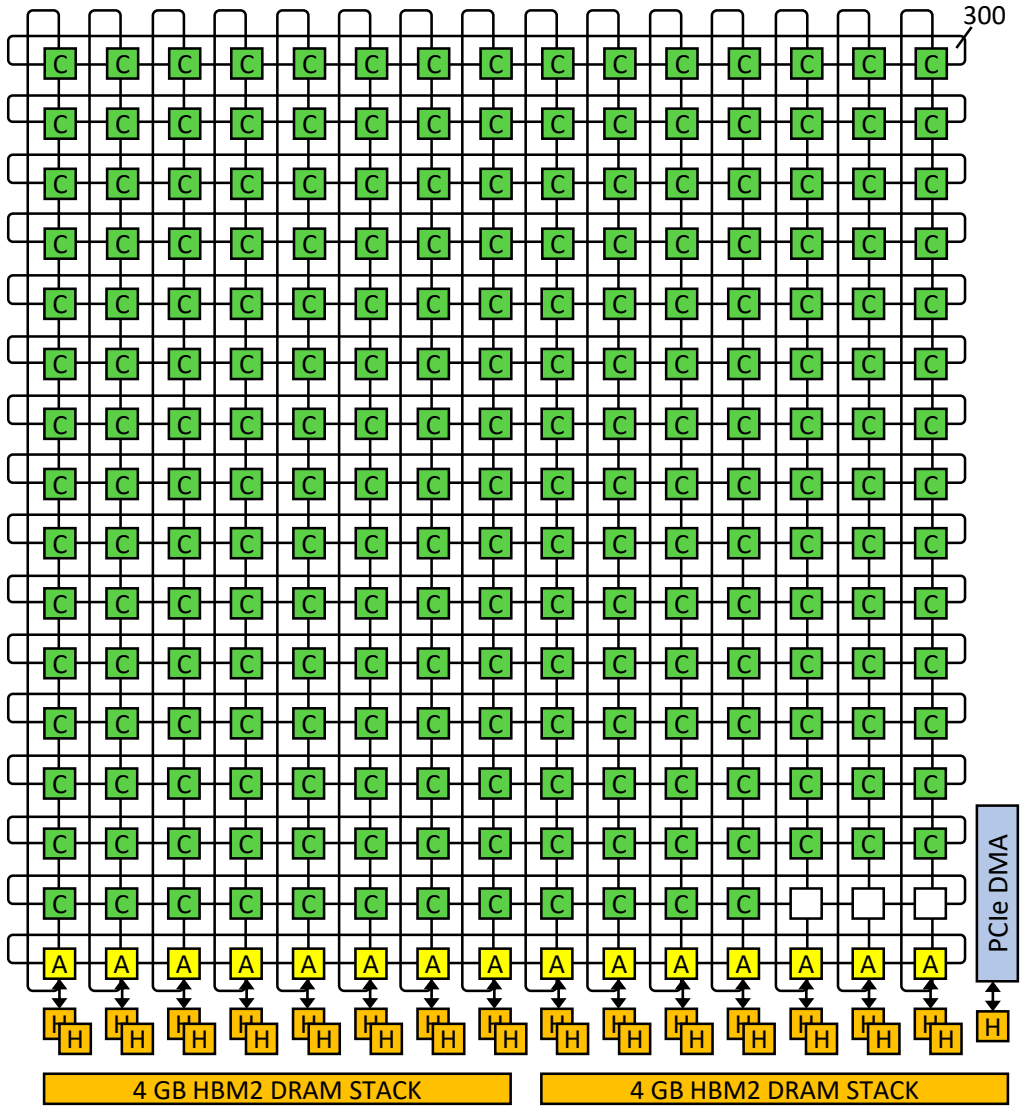
- 400 6-LUTs (sans <<)!

```
time # ex:pc insn              rd                res
3818 0  010c bne a4,t4,00c4   ---- ---------------- 0[720]=00000000000000e4 if_pc=00c4
3820 0 ~0110 (annul)          ---- ---------------- 0[728]=00000000000000e5
3822 0  00c4 ld t3,0(a4)      ---- ---------------- 0[730]=00000000000000e6
3824 0  00c8 ld t1,8(a4)      ---- ---------------- 0[738]=00000000000000e7 t3?
3826 0  00cc ld a7,16(a4)     ---- ---------------- t1? t3?
3828 0  00d0 ld a6,24(a4)     ---- ---------------- t1? a7? t3?
3830 0  00d4 ld a0,32(a4)     ---- ---------------- t1? a6? a7? t3?
3832 0  00d8 ld a1,40(a4)     ---- ---------------- t1? a0? a6? a7? t3?
3834 0  00dc ld a2,48(a4)     ---- ---------------- 0[f40] t1? a0? a1? a6? a7? t3?
3836 0  00e0 ld a3,56(a4)     0:t3=@00000000000000e8 0[f48] t1? a0? a1? a2? a6? a7? t3? t3!
3838 0  00e4 sd t3,0(a5)      0:t1=@00000000000000e9 0[f50] t1? a0? a1? a2? a3? a6? a7? t1!
time # ex:pc insn              rd                res
3840 0  00e8 sd t1,8(a5)      0:a7=@00000000000000ea 0[f58] a0? a1? a2? a3? a6? a7? a7!
3842 0  00ec sd a7,16(a5)     0:a6=@00000000000000eb 0[f60] a0? a1? a2? a3? a6? a6!
3844 0  00f0 sd a6,24(a5)     0:a0=@00000000000000ec 0[f68] a0? a1? a2? a3? a0!
3846 0  00f4 sd a0,32(a5)     0:a1=@00000000000000ed 0[f70] a1? a2? a3? a1!
3848 0  00f8 sd a1,40(a5)     0:a2=@00000000000000ee 0[f78] a2? a3? a2!
3850 0  00fc sd a2,48(a5)     0:a3=@00000000000000ef 0[740]=00000000000000e8 a3? a3!
3852 0  0100 sd a3,56(a5)     ---- ---------------- 0[748]=00000000000000e9
3854 0  0104 addi a4,a4,64    0:a4=+0000000000001f80 0[750]=00000000000000ea
3856 0  0108 addi a5,a5,64    0:a5=+0000000000001780 0[758]=00000000000000eb
3858 0  010c bne a4,t4,00c4   ---- ---------------- 0[760]=00000000000000ec if_pc=00c4
time # ex:pc insn              rd                res
```

GRVI vs. 2GRVI

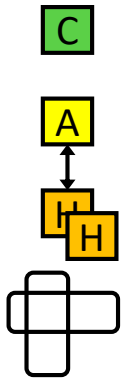
	32b GRVI PE	64b 2GRVI PE
Year	2015 Q4	2019 Q2
ISA	RV32I + mul/lr/sc	RV64I + lr/sc
Area	320 6-LUTs	400 6-LUTs (sans shared <<)
Fmax / congested	400 / 300 MHz	500+ / TBD MHz
Pipeline stages	2 / 3	2 / 3 / 4 (superpipelined)
Out-of-order retire		yes
Cluster, load interval	5 cycles	1 / cycle
Cluster, load-to-use	5 cycles	3-6 cycles
Cluster, Σ RAM BW	4.8 GB/s (300 MHz)	12.8 GB/s (400 MHz)

Phalanx SoC: 15x15-3 Array of Clusters + HBM + PCIe



PE ↔ Cluster RAM ↔ NoC ↔ AXI ↔ HBM

- 32 B write request message;
  32×*n* B burst-read request → n×32 B read responses
- PE sends R/W request message to its NoC-AXI bridge;
  bridge issues request to its AXI-HBM channel(s);
  bridge sends read response messages to dest. address
- 32 B write + 32 B read response per cycle per bridge
- Measured ~130 GB/s write + ~130 GB/s read at 300 MHz



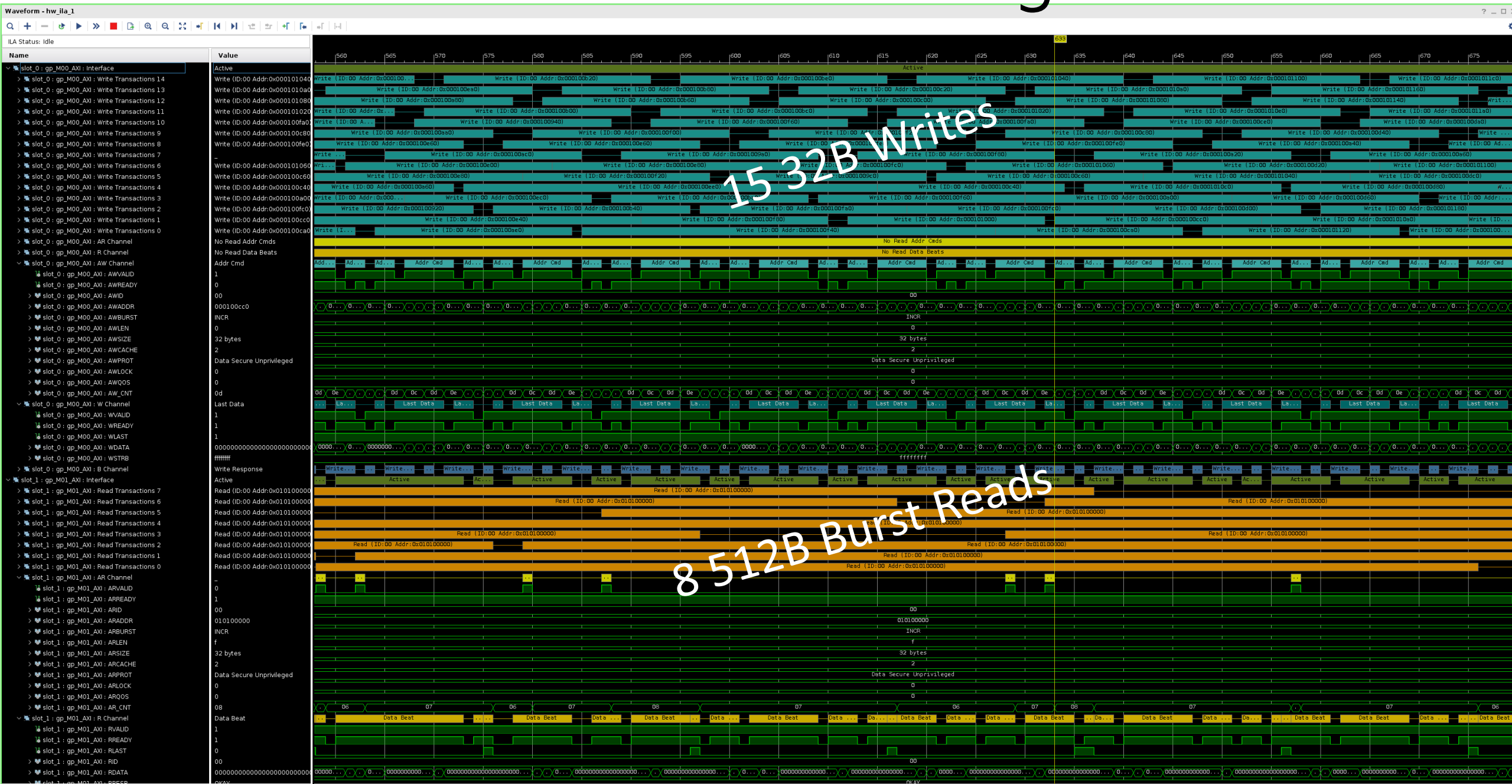
Cluster { 8 GRVI / 6 2GRVI, 4-8 KB IRAM, 128 KB CRAM, Hoplite router }

NoC-AXI RDMA bridge { 2 256b AXI R/W req queues, 2 resp queues }

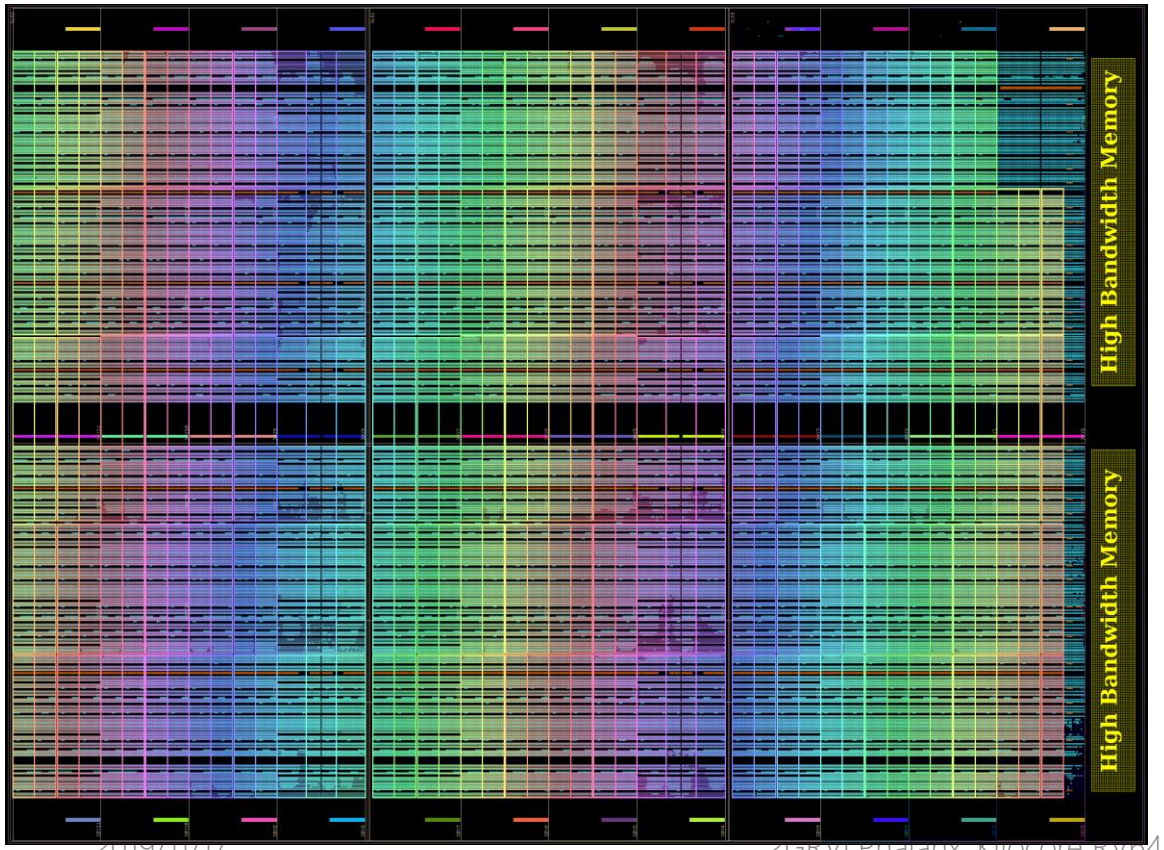
Two AXI-*switch*-MC-HBM2 bridges, each 256b R/W at up to 450 MHz

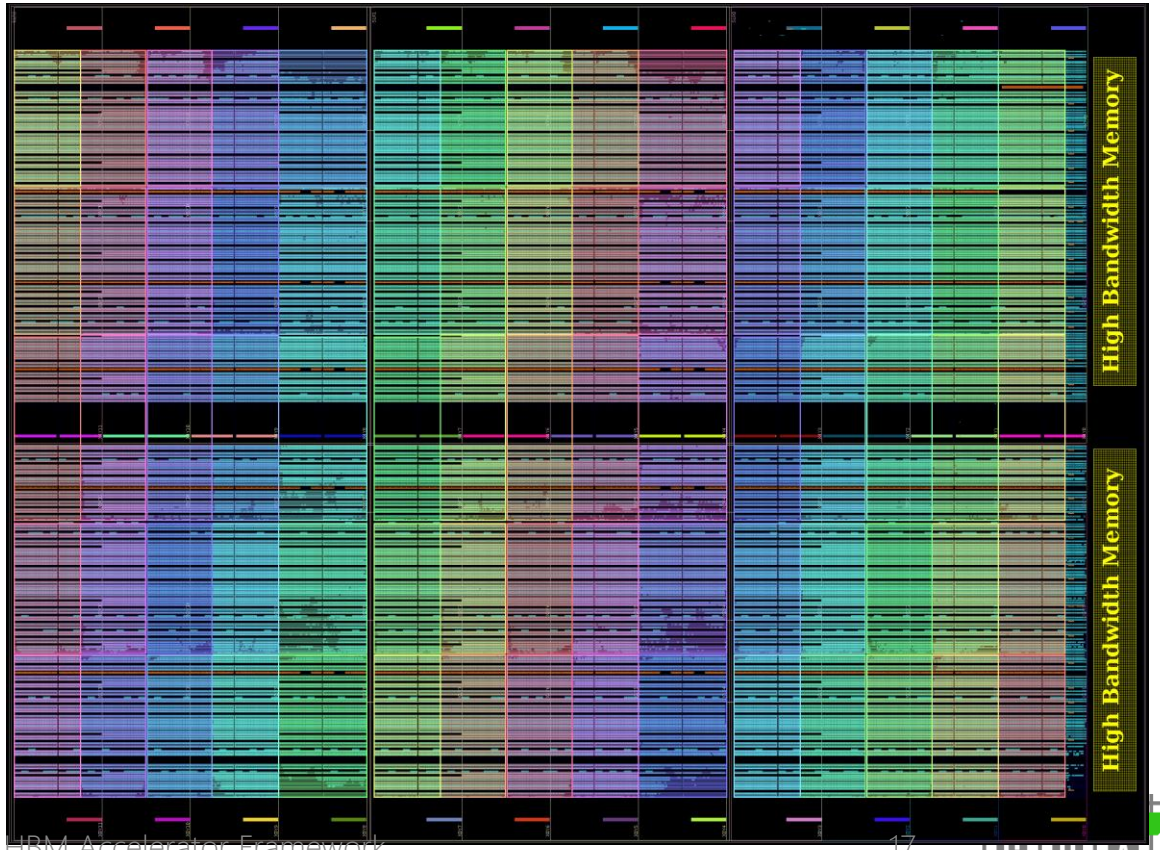
Unidirectional Hoplite NoC X-ring rows and Y-ring columns

NoC-AXI-HBM Transactions in Flight



222 x 8 GRVI PEs = 1776 RV32I PEs
222 x 6 2GRVI PEs = 1332 RV64I PEs





Phalanx-HBM2 Next Steps

- Tune up to 400+ MHz = ~200 GB/s writes + ~200 GB/s reads
- Computational HBM2 – compute at the bridges
  - Scatter/gather, add-to-memory, block zero, copy, hash, reduce, select, regexp, sort, …

Phalanx Parallel Programming Models

- Architecture: array of clusters of PEs, no caches, message passing

- Today: bare metal C/C++ + message passing runtime
- Future
  - Flat data parallel NDRange OpenCL kernels
  - Streaming kernels composed with OpenCL pipes
  - ‘Gatling gun’ parallel packet processing

An OpenCL-like Model and Tools

- Familiar to GPU developers(?)

- Host side: Xilinx SDAccel OpenCL runtime
  - Setup, copy buffers, queue parallel kernel calls, wait, copy results
- FPGA side: GRVI Phalanx ⇔ SDAccel-for-RTL shell
  - Map **work groups** to PE clusters, **work items** to PEs
  - Memory: **global** = HBM; **local** = cluster RAM (static); **private** = thread (auto)
  - Scheduler (PEs at cluster 0): distribute kernels, map work groups to idle clusters
- *Plan. Not yet implemented*

OpenCL "Like"

```
kernel void vector_add(
  global int* g_a,
  global int* g_b,
  global int* g_sum,
  const unsigned n)
{
  local align int a[N], b[N], sum[N];
  int iloc = get_local_id(0) * n;
  int iglb = (get_group_id(0) * get_local_size(0) + get_local_id(0)) * n;
  int size = n * sizeof(int);

  copy(a + iloc, g_a + iglb, size); // from HBM
  copy(b + iloc, g_b + iglb, size);
  barrier(CLK_LOCAL_MEM_FENCE);

  for (int i = 0; i < n; ++i)
    sum[i] = a[i] + b[i];
  barrier(CLK_LOCAL_MEM_FENCE);

  copy(g_sum + iglb, sum + iloc, size); // to HBM
}
```

Take Aways

- (Prior work)
  - Software-first, software-mostly manycore accelerators
  - Die filling, FPGA frugal, clustered, tiled, NoC-interconnected overlays
- Democratizing HBM
  - Xilinx AXI-HBM bridges are easy to use, simplify interconnects, save 100Ks LUTs
  - HBM bandwidth is now accessible to all
- Towards an OpenCL-like SDK, on AWS F1, Azure NP10, Alveo

GRAY

References

- The Past and Future of FPGA Soft Processors (2014)
  http://fpga.org/2014/12/31/the-past-and-future-of-fpga-soft-processors/
- GRVI Phalanx and Hoplite NoC (2016)
  http://fpga.org/grvi-phalanx/
  http://fpga.org/hoplite/
- 2GRVI Phalanx at Hot Chips 31 (2019)
  http://fpga.org/2019/08/19/2grvi-phalanx-at-hot-chips-31-2019/
- Xilinx AXI High Bandwidth Memory Controller v1.0
  https://www.xilinx.com/support/documentation/ip_documentation/hbm/v1_0/pg276-axi-hbm.pdf