“If you were plowing a field, which would you rather use: two strong oxen or 1024 chickens?“ – Seymour Cray

GRVI Phalanx Update:
Plowing the Cloud with Thousands of RISC-V Chickens

Jan Gray | jan@fpga.org | http://fpga.org
FPGA Datacenter Accelerators Are Here!

- FPGAs as cloud accelerators
  - Massively parallel, customized, 25/100G networking
  - Genomics, video, analytics, machine learning
  - Catapult v2, Amazon F1, Baidu, Alibaba, Huawei

- Design productivity challenge
  - Software: multithread C++ app → spatial accelerator
  - Hardware: tape-out a complex SoC daily?
GRVI Phalanx Accelerator Kit

• A parallel processor overlay for software-first accelerators:
  – Recompile and run on 100s of RISC-V cores
  = More 5 second recompiles, fewer 5 hour PARs
  + Add custom FUs and accelerators, as needed

• **GRVI**: FPGA-efficient RISC-V processing element

• **Phalanx**: fabric of clusters of PEs, memories, IOs

• **Hoplite**: 2D torus network on chip
GRVI: Austere RISC-V Processing Element

- Simpler PEs $\rightarrow$ more PEs $\rightarrow$ more parallelism!
- GRVI: RV32I - CSRs - exceptions + $\text{mul}^* + \text{lr/sc}$

$\approx$320 LUTs @ 375 MHz
GRVI Cluster: 8 PEs, Shared RAM, More

* per pair: lb sb lh sh sll sr* mul*
‡ per bank, per PE: lr sc

~3500 LUTs
Composition: Message Passing On a NoC

• Hoplite: FPGA-optimal 2D torus NoC router
  – Single flit, unidir rings, deflection routing, multicast

• PGAS: dram:40 -or- \{ dest:20; local: 20 \}

256b @ 400 MHz = 100 Gb/s links
10×5 Clusters × 8 GRVI PEs
= 400 GRVI Phalanx (KU040)
11/30/16: Amazon AWS EC2 F1!

F1 Instances
New Instance Family With Customizable Field Programmable Gate Arrays
Run Your Custom Logic On EC2

Preview Available Today
Amazon F1.2xlarge Instance

Diagram:

- XEON
- NVMe
- ENA
- VU9P FPGA
- DRAM 122 GB
- DRAM 64 GB
Amazon F1.16xlarge Instance

- XEON
- DRAM
- NVMe
- VU9P
- PCIe SWITCH
- ENA
F1’s UltraScale+ VU9P FPGAs

- **1.2M** 6-LUTs
- **2K** 4 KB BRAMs = 8 MB
- **1K** 32 KB URAMs = 30 MB
- **7K** DSPs
1680 RISC-Vs, 26 MB CMEM (VU9P, 12/2016)

- 30x7 x { 8 PEs, 128 KB, router }
- First kilocore RISC-V
- (Then) most 32b RISCs/chip
### 1680 Core GRVI Phalanx Statistics

<table>
<thead>
<tr>
<th>Resource</th>
<th>Use</th>
<th>Util. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical nets</td>
<td>3.2 M</td>
<td>-</td>
</tr>
<tr>
<td>Routable nets</td>
<td>1.8 M</td>
<td>-</td>
</tr>
<tr>
<td>CLB LUTs</td>
<td>795 K</td>
<td>67.2%</td>
</tr>
<tr>
<td>CLB registers</td>
<td>744 K</td>
<td>31.5%</td>
</tr>
<tr>
<td>BRAM</td>
<td>840</td>
<td>38.9%</td>
</tr>
<tr>
<td>URAM</td>
<td>840</td>
<td>87.5%</td>
</tr>
<tr>
<td>DSP</td>
<td>840</td>
<td>12.3%</td>
</tr>
</tbody>
</table>

- **Frequency**: 250 MHz
- **Peak MIPS**: 420 GIPS
- **CMEM 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**: 44°C

**Vivado**: 2016.4 / ES1

- **Max RAM use**: ~32 GB
- **Flat build time**: 11 hours
- **Tools bugs**: 0

> 1000 BRAMs + 6000 DSPs available for accelerators
Towards a GRVI Phalanx SDK

• Bridge Phalanx and AXI4 system interfaces
  – Message passing bridge to host CPUs (x86 or ARM)
  – R/W req/resp RDMA bridge to DRAM
• Two SDK hardware targets
  – 8-80-core PYNQ (Z7020) ($65 edu)
  – 800-10K-core AWS F1 ($-$$/hr)
PYNQ-Z1 Demo Video

GRVI Phalanx on Pynq-Z1
88 GRVI RISC-V cores + 352 KB SRAM
+ 32 KB Frame Buffer
(work in progress)

Jan Gray
jan@fpga.org
http://fpga.org/grvi-phalanx
GRVI Phalanx on AWS F1
GRVI Phalanx on AWS F1

0 DDR
884 3 DDR
1240 1 DDR
9920 (F1.16XL)

15x10 NoC
34x5 NoC
Programming Models

• Small kernels, local/DRAM shared mem, messages
• Current: bare metal multithread C++, messages
• W.I.P: OpenCL
  – Leverage Xilinx SDAccel’s “RTL kernel” stack and shell
  – Host: OpenCL to stage buffers and invoke kernels
  – Phalanx: flat data parallel kernels, work item indexed, read/compute/writeback to DRAM (for starters)
• Later
  – True OpenCL kernels, pipes, KPNs, P4, ...?
  – Accelerated with custom RTL FUs, HLS C accelerator cores
Current Work

• Complete first F1.2XL and F1.16XL ports
  – Add SDAccel shell integration

• GRVI Phalanx SDK for F1 and PYNQ
  – F1: AMI+AFI in AWS Marketplace
  – PYNQ: Jupyter Python notebooks, bitstreams
  – Open source tools, specs, libs, tests, examples
In Summary

• Make it easier to access FPGA spatial parallelism
  – Value proposition unproven, awaits workloads
• Competitive perf demands a frugal design
• SDK coming: 8-80-800-10,000 core designs
• Enabled by the excellent RISC-V ecosystem

Thank you!