GRVI Phalanx: A Massively Parallel RISC-V FPGA Accelerator Framework ⇨ A 1680-core, 26 MB Parallel Processor Overlay for Xilinx UltraScale+ VU9P

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

***Abstract*—GRVI is an austere 32b RISC-V soft processor. Phalanx is a parallel processor and accelerator array overlay framework. Groups of GRVI PEs and accelerator cores form shared memory compute clusters. Clusters and memory and I/O controllers communicate by message passing over an FPGA-optimal Hoplite torus NoC. This paper summarizes recent work to host a GRVI Phalanx on an XCVU9P FPGA as found in Amazon AWS EC2 F1 instances. The design has an array of 30×7=210 clusters, each with eight 32b GRVI cores, 128 KB of shared RAM, and a 300b/link Hoplite router. At 250 MHz, power is 31-40 W, max throughput is 0.4 TIPS, local memory bandwidth is 2.5 TB/s, and NoC bisection bandwidth is 0.9 Tb/s.**

I. Introduction

Complementing server CPUs, FPGA accelerators promise higher throughput, lower latency, and lower energy [1]. But it is challenging to move working software into an accelerator, and to maintain it as the code evolves. RTL, High Level Synthesis, and even OpenCL-to-FPGA tools have serious shortcomings, such as high porting effort and multi-hour builds. Another challenge is system design and timing closure of a complex SOC comprising many cores and fast I/O and DRAM interfaces. How do you interconnect so many cores across the die at full bandwidth? Few organizations can assemble all the skillsets necessary to successfully develop a new FPGA accelerator.

GRVI Phalanx [2] is a parallel processor *overlay* framework that aspires to simplify accelerator development. It supports a software-first, software-mostly approach. A multithreaded C++ workload is recompiled and run on the soft processors in the overlay. Then, as necessary, custom hardware (new instructions, accelerator cores, memories) are introduced to speed up inner loops. Most design iterations are just recompiles, and accelerator development feels more like software performance engineering.

II. The GRVI RISC-V Core

Actual *acceleration* of a software-mostly workload requires an FPGA-efficient soft processor that runs mainstream open source software. RISC-V [3] is a good ISA choice. It is an open specification; it is modern, extensible, layered (pay as you go), and it has "critical mass" of (specs, tests, compilers, tools, simulators, libraries, Linux, and processor and interface IP). Its base 32-bit integer RISC ISA, RV32I, is sufficiently clean and regular to afford a compact FPGA implementation.

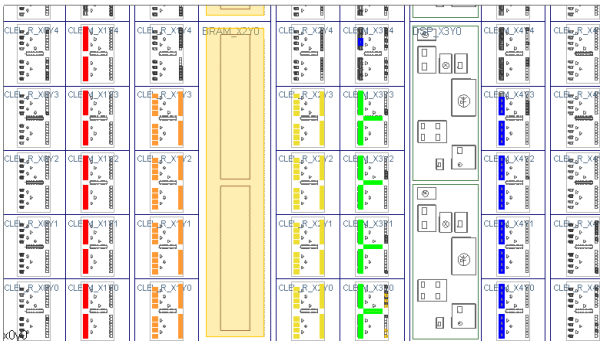

Fig. 1: GRVI datapath RPM

GRVI is a simple scalar core that implements *user-mode* RV32I, sans CSRs, plus MUL/H* and LR/SC. It has been carefully optimized for performance÷area in a Xilinx FPGA. It has a three stage pipeline (fetch; decode; execute), with two cycle loads and three cycle jumps/taken branches. The datapath (Fig. 1) has a 2R/1W register file; two pairs of operand multiplexers and registers with result forwarding; an ALU; a comparator for conditional branches and SLT*; a PC unit for I-fetch, jumps, and branches; and a result multiplexer selecting from ALU, return address, load data, and multiply/shift/custom-function-units. Datapath LUTs are technology mapped and floorplanned into a relationally placed macro (RPM). The ALU, PC unit, and comparator use "carry logic". Each LUT in the synthesized control unit was scrutinized.

To maximize cores/die and hence memory parallelism to the thousands of block RAMs, GRVI factors out inessential logic. Multiply-shift and load/store byte-align sign-extension logic is shared by core pairs in a cluster, halving their amortized cost.

GRVI is small and fast. The datapath uses 250 LUTs; the core overall is 320 LUTs. Configured with BRAM memories, single core Fmax is 375 MHz, or about 0.7 MIPS/LUT.

III. GRVI Clusters

Modern FPGAs are *vast*. F1's VU9Ps provide 1.2M 6-LUTs, 960 32 KB URAMs (30 MB total), 2160 4 KB BRAMs, and 6840 DSPs. What arrangement of FPGA resources into PEs and "uncore" yields an efficient, programmable parallel machine?

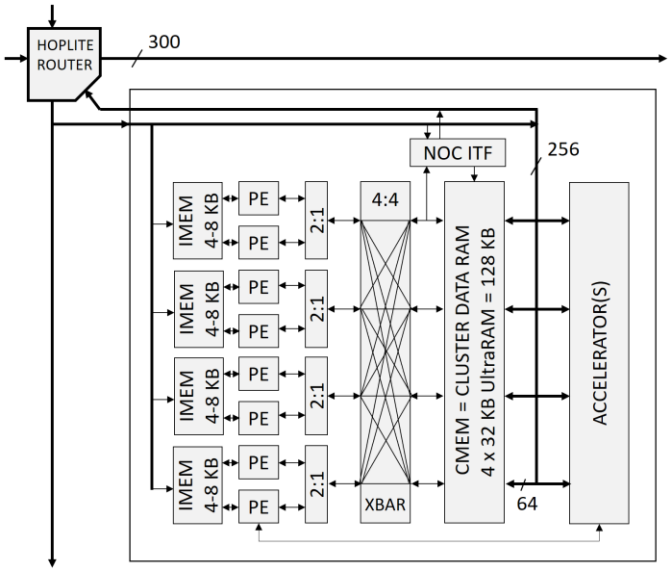

Fig. 2: Cluster: 8 PEs, accelerators, 128KB CMEM, router

Critically, GRVI Phalanx eschews caches, trading off some programming convenience for throughput, scalability, and energy efficiency. Instead the FPGA is divided into many shared memory cluster tiles each with eight PEs, 128 KB of shared cluster memory (CMEM), optional accelerator(s), and a 300b Hoplite router (Fig. 2). Four/eight 4 KB BRAMs form kernel instruction memories (IMEMs) shared by PE pairs. Four 32 KB URAMs form an 8-port 128 KB shared CMEM, with four 32b

PE ports, and four 64b accelerator/NoC ports. Per-bank RISC-V LR/SC reservations provide atomic CMEM accesses.

Accelerator cores communicate with PEs via CMEM, or via PE operand registers and result multiplexers.

NoC messages can be sent or retired at 32 B/cycle/cluster, until the NoC saturates. To send a message, a PE stores a send-message command to the cluster's NoC interface. The latter reads 32 bytes from the source address in CMEM and sends it, via the NoC, to the destination address in some other cluster. Accelerator cores can directly send or receive 32 B messages.

A cluster may be configured with more/fewer PEs, memory, or accelerator cores, to right-size resources to the workload. The area of a PE plus its share of the eight PE cluster is ~480 LUTs.

IV. Hoplite router and NoC

GRVI Phalanx is enabled by Hoplite, a directional deflection torus NoC. A Hoplite router is simple, frugal, wide, and fast. Hoplite rejects textbook virtual channel flit-router orthodoxy, which maps poorly to FPGAs. Instead, Hoplite's bufferless 3x2 router switch is *FPGA-optimal*, with one LUT+FF delay and using just one dual-output 6-LUT per bit of link width. This achieves 100x better area-delay product than prior work. Hoplite has configurable link pipelining, routing, and multicast. [4]

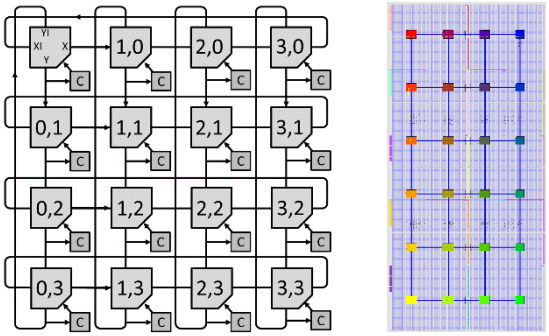

Fig. 3: 4×4 Hoplite NoC; 4×6×256b NoC has 100 Gb/s links

V. A 1680 core, 26 MB GRVI Phalanx

Fig. 4 shows an XCVU9P configured with 30×7=210 GRVI clusters, in all, 1680 cores and 26 MB of CMEM. The blue grid lines are the Hoplite NoC links. Presently it runs in a Xilinx VCU118 ES1 evaluation kit. With 1680 cores operating at 250 MHz, it has a peak (cannot exceed) throughput of 420,000 MIPS; peak shared memory bandwidth into CMEMs of 2.5 TB/s, and NoC bisection bandwidth of 900 Gb/s. (Pending work will double-clock CMEM/accelerators, up to 4.2 TB/s.) 1320 BRAMs and 6000 DSPs are free for use by accelerator cores.

Running a message-passing matrix multiply test workload over all cores, power, measured with INA226, is 31-40 W (varies with phase of computation) ≈ 24 mW/PE, at 44 C.

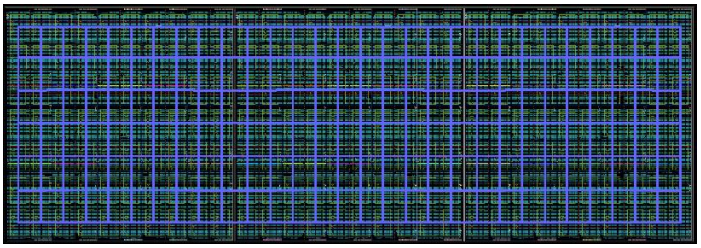

Fig. 4: 30×7=210 cluster GRVI Phalanx: 1680 32b RISC-V cores and 26 MB of cluster shared memory in an XCVU9P

VI. Accelerated parallel programming models & tools

GRVI Phalanx runs multithreaded message passing C++ workloads built with GCC for RISC-V. A post-link config file specifies which kernels/data are loaded into which clusters. Care is required to fit kernel+runtime in a 4-8 KB IMEM. Using 1024 multicast messages, a new kernel can be sent/loaded in ~4 µs.

Longer term, the plan is to run parallel programming models which fit Phalanx's mold: SPMD or MIMD code with small kernels, local shared memory, and global message passing (but no caches). These include OpenCL, streaming over process networks, and 'Gatling gun' parallel packet processing.

Custom function units, accelerators, and memories, written in RTL/HLS/OpenCL, may be coupled to PEs, CMEMs, or routers directly or via AXI-Stream or AXI4 bridges.

VII. Looking ahead: scaling up and down

Amazon AWS F1 instances [5] with one/eight VU9P FPGAs and up to 1.5 TB DRAM (Fig. 5), will democratize access to on-demand cloud-scale FPGA application acceleration appliances.

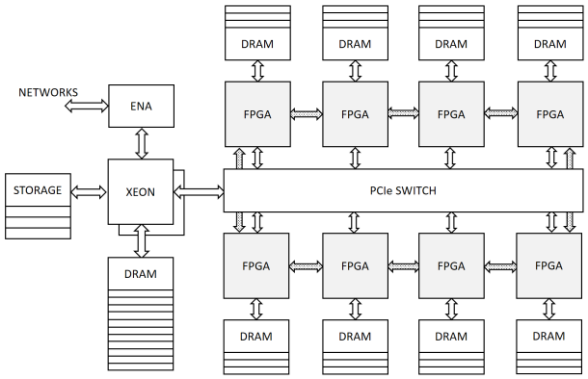

Fig. 5: AWS F1.16xlarge instance: 8 VU9Ps, 1.5 TB RAM

To make this vast fabric accessible to developers, work-in-progress includes: 1) add 32-64 byte/cycle/channel DRAM accesses via Hoplite-RDMA-AXI4-DRAM bridges; 2) widen GRVI to 64 bits to address >4 GB DRAM; and 3) add inter-FPGA (inter-NoC) remote message routing across PCIe and the inter-FPGA ring links. In all, an F1.16xlarge might host 10,000 GRVIs or 5,000 GRVI64s, plus accelerator cores.

At the other extreme, an 80-core GRVI Phalanx fits in the programmable logic region of a Zynq 7Z020 (e.g. $65 PYNQ-Z1 [6]) and will soon be made available as an educational kit.

GRVI Phalanx's frugal design esthetic, careful technology mapping, tiled overlay architecture, and leverage of RISC-V ecosystem assets, demonstrates how software developers might more easily tap the full spatial parallelism of modern FPGAs.

References

[1] A. Putnam, et al, A reconfigurable fabric for accelerating large-scale datacenter services, in 41st Int'l Symp. on Comp. Architecture (ISCA), June 2014.

[2] J. Gray. GRVI-Phalanx: A Massively Parallel RISC-V FPGA Accelerator Accelerator. In Proc. 24th IEEE Symposium on Field-Programmable Custom Computing Machines, May 2016.

[3] K. Asanović, D. Patterson, Instruction sets should be free: the case for RISC-V. Technical Report No. UCB/EECS-2014-146, August 2014.

[4] N. Kapre and J. Gray, Hoplite: Building Austere Overlay NoCs for FPGAs, 25th Int'l Conf. on Field-Programmable Logic and Applications, Sept. 2015.

[5] https://aws.amazon.com/ec2/instance-types/f1/

[6] http://www.pynq.io/board.html