FPGA-Accelerated Simulation of a RISC-V Vector Architecture

Kathryn Chapman[1,2], Maya Gokhale[1], Hiren Patel[3], Jenq-Kuen Lee[2]

[1]Lawrence Livermore National Laboratory, [2]National Tsinghua University (Taiwan), [3]University of Waterloo

Abstract

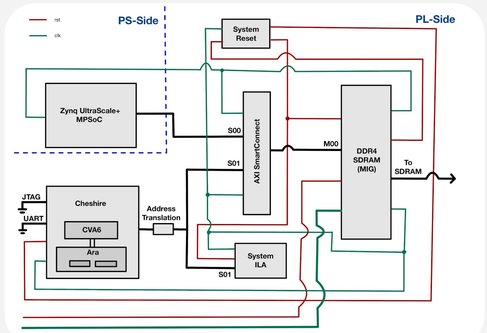
- AI and high-performance computing have driven a surge in RISC-V SoC accelerator architectures
  - The CPU acts as a control processor routing data to specialized IP blocks
  - Designing these SoCs requires software simulation for internal visibility and FPGA mapping for fast workload testing
- This work maps the Ara RISC-V vector processor[1] to a Zynq FPGA with ARM CPU and programmable logic. It uses the ARM to automate benchmark setup, execution, and result collection.

Introduction

- The ARA2 vector unit acts as a coprocessor for the CVA6 RISC-V scalar core
  - Together with peripherals called **Cheshire**
  - Allows us to run RVV 1.0 instructions
- We configured for the ZCU102, a SoC FPGA, containing both:
  - **Processing System (PS):** ARM processors, dedicated DRAM, I/O peripherals
  - **Programmable Logic (PL):** softcore processor & custom logic (CVA6+Ara), custom logic, separate dedicated DRAM
- Built heterogeneous system to allow for fast, flexible, autonomous testing of Ara under an ARM-controlled methodology
  - Offloads non-benchmarking workload to ARM core

Hardware Design

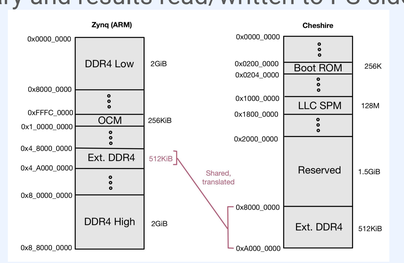
- ARM A53 SoC on PS-side and Cheshire share the same PL-side DRAM address space
  - ARM also has access to its PS-side DRAM
  - Address translation for Cheshire to avoid memory address overlap



ARM PS-side and Cheshire + Memory PL-side

Shared Memory

- Mailbox System
- Break the shared DRAM into regions for:
  - Polling (ARM & Cheshire)
  - RISC-V binary (ARM write, Cheshire read)
  - Benchmark array initialization (ARM)
    - ARM-side running 1.2GHz vs. Cheshire's 50MHz and 10MHz max JTAG clock
  - Results (Cheshire write, ARM read)
- Binary and results read/written to PS-side SD



Memory Map for Cheshire Address Translation

Benchmarks

- TSVC is a benchmark suite to evaluate/test the autovectorization performance of compilers
  - tests over a large number of vectorizable loops
    - high coverage to simulate the vector units
    - use of DRAM allows for large data sets (over L2 SRAM)
- RISC-V Binary is generated from C, as opposed to the official published Ara benchmarks that use assembly/intrinsics
  - Allows for evaluation of compiler's autovectorization

Test Suite Array Lengths

Size	Array LEN
Tiny	15360
Small	30720
Medium	256000
Large	1024000
Huge	4096000

Can Customize:
- array length
- times loop will repeat (ntimes)
- data format (float, double, signed and unsigned int 8,16,32,64)
- tests run consecutively (which ones, how many)

Categories of Compiler Vectorization Tests

Category	Benchmark
Linear Dependence	s000
Induction Variable	s121
Global Data Flow	s131
Control Flow	s162
Symbolics	s171
Statement Reordering**	s211
Node Splitting	s243
Expansion	s251
Cross Thresholds	s1281
Reductons	s311
Packing**	s341

**not vectorizable

Results

Large (float)

Benchmark	ntimes = 1		ntimes = 20		instret/cycle	
	cycles	instret	cycles	instret	ntimes = 1	ntimes = 20
s000	4602108	384032	91872078	7680303	0.0834	0.0836
s121	9051565	528050	180967276	10560563	0.0583	0.0584
s131	15085430	880068	301612030	17600923	0.0583	0.0584
s162	3003936	208029	59921651	4160227	0.0693	0.0694
s171	2773605	176024	45069471	3520183	0.0635	0.0781
s211	60958452	17407991	1224656932	348159484	0.2856	0.2843
s243	2242779	128025	44853093	2560183	0.0571	0.0571
s251	11923698	896068	238469818	17920752	0.0752	0.0751
s1281	20493834	1600097	410037318	32001009	0.0781	0.0780
s311	90764840	1760088	1815294140	35201418	0.0194	0.0194
s341	19800037	9216021	399364129	184320141	0.4655	0.4615

Discussion

- FPGA simulation enables evaluation of much more realistic workloads than possible with software RTL simulation
- ARM manages benchmark setup, execution, and result collection, enabling automated evaluation of the Ara vector processor

Potential Additional Use Cases

- ARM core(s) can be leveraged for
  - high-speed data generation
  - Inspecting execution results and dynamically select next workloads

Future Work

- Memory hierarchy optimization for improved DRAM latency & performance

Acknowledgment This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 under LDRD Project 25-ERD-016

1 M. Perotti, M. Cavalcante, R. Andri, L. Cavigelli and L. Benini, "Ara2: Exploring Single- and Multi-Core Vector Processing With an Efficient RVV 1.0 Compliant Open-Source Processor," in IEEE Transactions on Computers, vol. 73, no. 7, pp. 1822-1836, July 2024.