Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Designing open-source RISCV hardware with open-source software

Adrian Cristal, Osman Unsal,
Cristobal Ramirez

29/06/2020

Workshop on RISC-V and OpenPOWER @ ICS

Designing open-source RISCV hardware with open-source software

Focusing on early architecture system exploration of hardware design for Vector Processing Unit (VPU)

Will discuss:

Architecture Simulator for VPU
- Gem5
- Spike+Sparta

RISCV VALib Benchmark Suite

- Gem5 Simulator

- Background
  - SIMD
    - Multimedia extensions
    - Vector Architectures

- Gem5 Vector Architecture Base Model

- RISCV VALib Benchmark Suite and Evaluation

RISCV VALib Suite

- **Gem5 Simulator**

- Background
  - SIMD
    - Multimedia extensions
    - Vector Architectures

- Gem5 Vector Architecture Base Model

- RISCV VALib Benchmark Suite and Evaluation

RISCV VALib Suite

Why Gem5 ?

- System-on-chip (SoC) systems are complex and expensive to prototype
  - Simulation is a cost-effective way to evaluate new ideas
- gem5 models various system components, such as different CPU models, interconnection topologies, memory subsystems, etc.

Gem5

- Syscall emulation and Full-system emulation
- Multiple ISAs support
  - Alpha
  - ARM
  - SPARC
  - MIPS
  - POWER
  - RISC-V
  - X86

- Timing model for the full cache hierarchy with support for custom coherence protocols
- Multiple CPU models

  Atomic

  In-order

  Out-of-order

Gem5 vector support

- **Intel AVX**
  - 256-bit (4 words – 64-bit)
  - 512-bit(8 words – 64-bit)
- **ARM SVE**
  - MVL support :
    - from 128-bit (2 words – 64-bit)
    - up to 2048-bit (32 words – 64-bit)

- **RISCV-V V-extension – This work**
  - MVL support :
    - from 128-bit (2 words – 64-bit)
    - up to 16 384 (256 words – 64-bit)

    However it can be bigger !!

- Gem5 Simulator

- **Background**
  - SIMD
    - Multimedia extensions
    - Vector Architectures

- Gem5 Vector Architecture Base Model

- RISCV VALib Benchmark Suite and Evaluation

RISCV VALib Suite

Single Instruction Multiple Data (SIMD)

SIMD computers exploit data-level parallelism by applying the same operations to multiple items of data in parallel.

SIMD classification:

Multimedia Extensions

- SSE – x86
- AVX – x86
- MSA – MIPS
- NEON - ARM

Current Vector Architectures

- SVE – ARM – 1st Position on Top500 (Fujitsu Fugaku)
- Vector extensions for NEC (Aurora)
- RISC-V

GPU

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

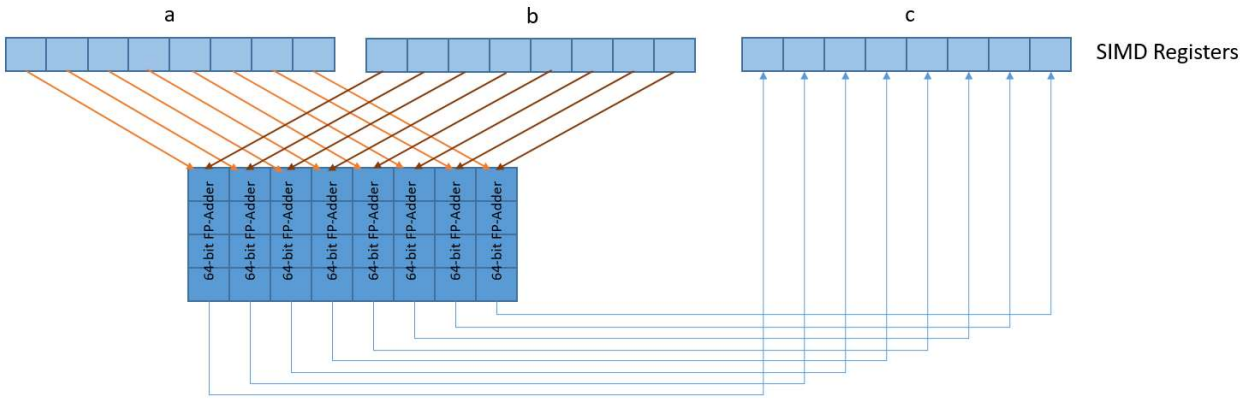
Seymour Cray, Father of the Supercomputer
(arguing for two powerful vector processors versus many simple processors)

- Currently in V0.8, will soon introduce V1.0
- Current specification includes a rich set of classical vector instructions such as scatter, gather, reductions, iota, …
- As well as vector operations that are specific to RISCV: widening, narrowing…
- Completely open specification
- Specification primary focus was support for embedded systems, IoT, edge computing
- Initial implementations from Alibaba, Andes technology, … are in this market segment

512-bit SIMD extension (8 x 64-bit elements)

Example – Serial

```
double *a, *b, *c;
size = 85
for(i=0; i< Size ; i++)
{
        c[i] = a[i]+b[i];
}
```



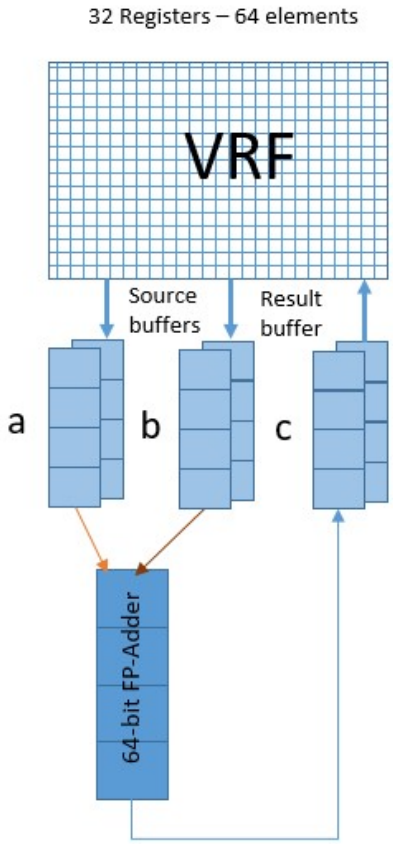
Example SIMD AVX-512

```
SIMD_WIDTH=8
double *a, *b, *c; // cache line aligned
size = 85
for(i=0; i<size-5 ; i=i+SIMD_WIDTH)
{
        va = _mm512_vload_pd(&a[i]);
        vb = _mm512_vload_pd (&b[i]);
        vc = _mm512_vadd_pd (va,vb);
        _mm512_vstore_pd (vc, &c[i]);
}
//loop tail
for( ; i< size ; i++)
{
        c[i] = a[i]+b[i];
}
```

Vector Architecture

MVL = 64 elements (64-bit/element)

1 Lane



Example – Serial

```
double *a, *b, *c;
size=85
for(i=0; i<85 ; i++)
{
        c[i] = a[i]+b[i];
}
```

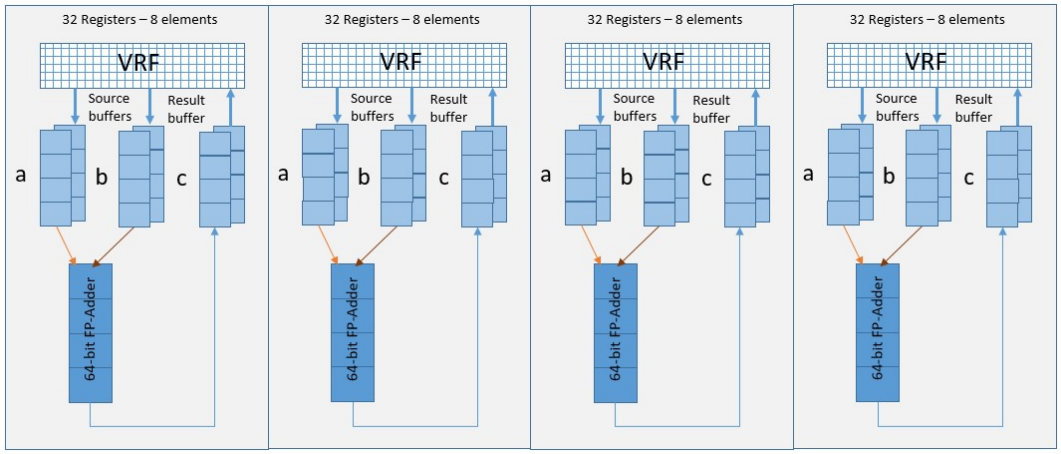
Example Vector , MVL = 64 , VLA Code

```
double *a, *b, *c;
size=85
gvl = vsetvl (size, 64-bit) ; // gvl = 64
for(i=0; i<size ;) {
        va = vload(&a[i]);
        vb = vload(&b[i]);
        vc = vadd(va,vb);
        vstore(vc, &c[i]);
        i=i+ gvl
        gvl = vsetvl (size-i, 64-bit) ;
}
```

Multilane Vector Architecture

MVL = 64 elements (64-bit/element)

4 Lane



Example – Serial

```
double *a, *b, *c;
size=85
for(i=0; i<85 ; i++)
{
        c[i] = a[i]+b[i];
}
```

Example Vector , MVL = 64 , VLA Code

```
double *a, *b, *c;
size=85
gvl = vsetvl (size, 64-bit) ; // gvl = 64
for(i=0; i<size ;) {
        va = vload(&a[i]);
        vb = vload(&b[i]);
        vc = vadd(va,vb);
        vstore(vc, &c[i]);
        i=i+ gvl
        gvl = vsetvl (size-i, 64-bit) ;
}
```

- Gem5 Simulator

- Background
  - SIMD
    - Multimedia extensions
    - Vector Architectures

- **Gem5 Vector Architecture Base Model**

- RISCV VALib Benchmark Suite and Evaluation

RISCV VALib Suite

Parameterizable Vector Architecture Model

Renaming Unit
- Number of physical registers

Issue Queues (Arithmetic & Memory)
- Number of entries
- In-order or Out-of-order Issue

Reorder Buffer (ROB)
- Number of entries

Register File
- Number of R/W-ports
- Size (related to the MVL and Number of Physical Registers)
- MVL

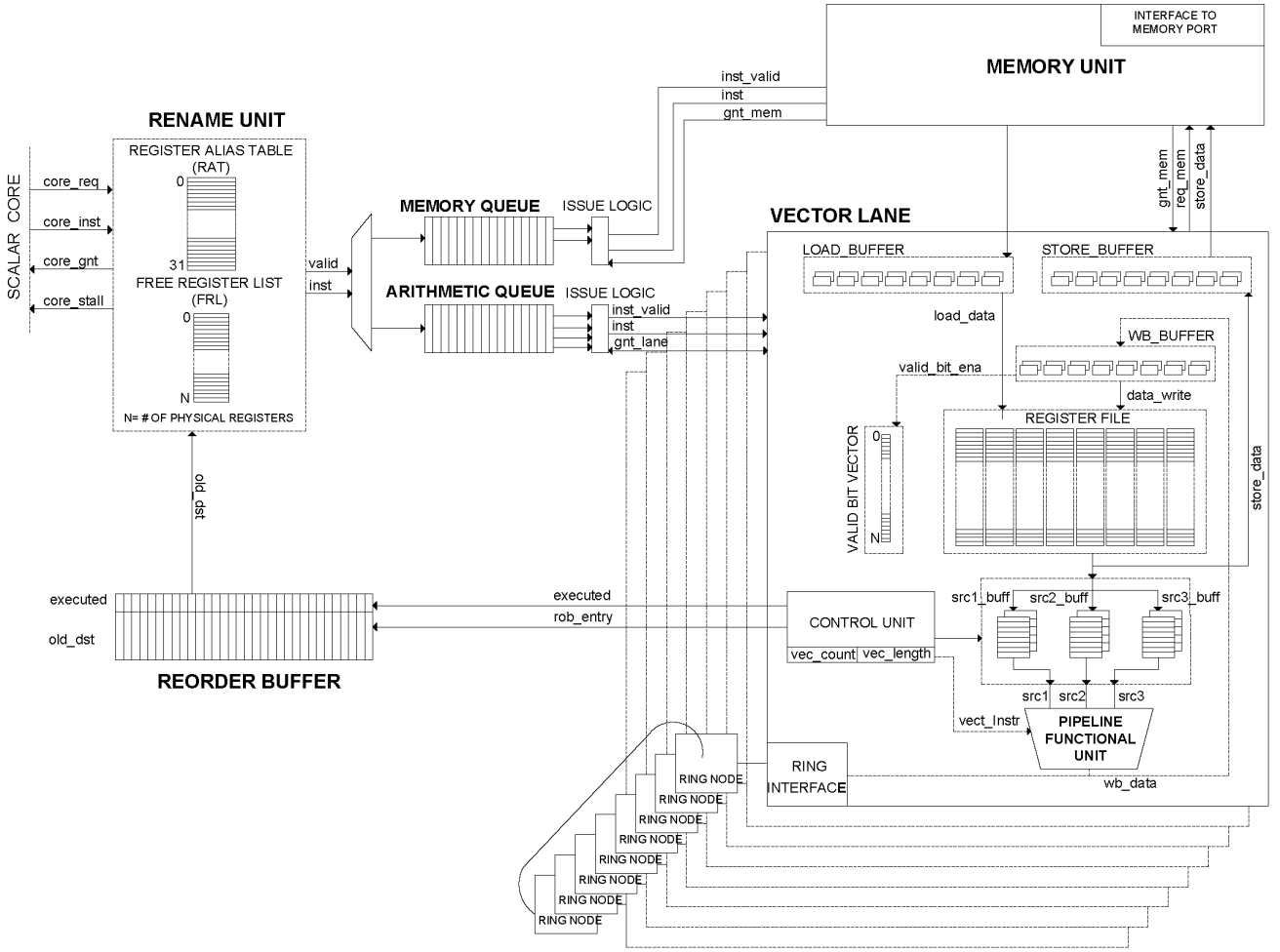
VPU
- Number of Lanes
- Functional units latency

VMU
- Memory ports
- Connect to L1, L2 or Main memory

Lane Interconnection
- Interconnection Topology
  - Ring
  - Crossbar

Caches
- Latency
- Size, etc.

Frequency
- default: 1Ghz



- Future Plans:

- Integrating vector architecture model with Spike+Sparta simulator platform

- Modeling tightly-coupled vector architectures

- Gem5 Simulator

- Background
  - SIMD
    - Multimedia extensions
    - Vector Architectures

- Gem5 Vector Architecture Base Model

- **RISCV VALib Benchmark Suite and Evaluation**

RISCV VALib Suite

- BSC Microsoft Research Center Vectorized Library Suite (2014)

- Vectorized desktop and mobile applications:
  - Sphinx3 – Speech recognition
  - Festival – Speech synthesis
  - SPEC2000/CSU – Face Recognition
  - San Diego – Computer Vision
  - C-Store – Column-Store DBMS
  - NU-MineBench – Data Mining

Milan Stanic, Oscar Palomar, Ivan Ratkovic, Milovan Duric, Osman S. Unsal, Adrián Cristal:
VALib and SimpleVector: Tools for Rapid Initial Research on Vector Architectures.
Conf. Computing Frontiers 2014

Application	Application Domain	Algorithmical Model	DLP Pattern	Benchmark Suite
Blackscholes	Financial Analysis	Dense Linear Algebra	Regular	PARSEC/PARVEC
Canneal	Engineering	Unstructured Grids	Irregular	PARSEC/PARVEC
Heat Diffusion	Heat transfer	Dense Linear Algebra	Regular	-
Particle Filter	Medical Imaging	Structured Grids	Mix	Rodinia
Path Finder	Grid Traversal	Dynamic Programming	Regular	Rodinia
Streamcluster	Data Mining	Dense Linear Algebra	Mix	PARSEC/PARVEC
Swaptions	Financial Analysis	MapReduce	Regular	PARSEC/PARVEC

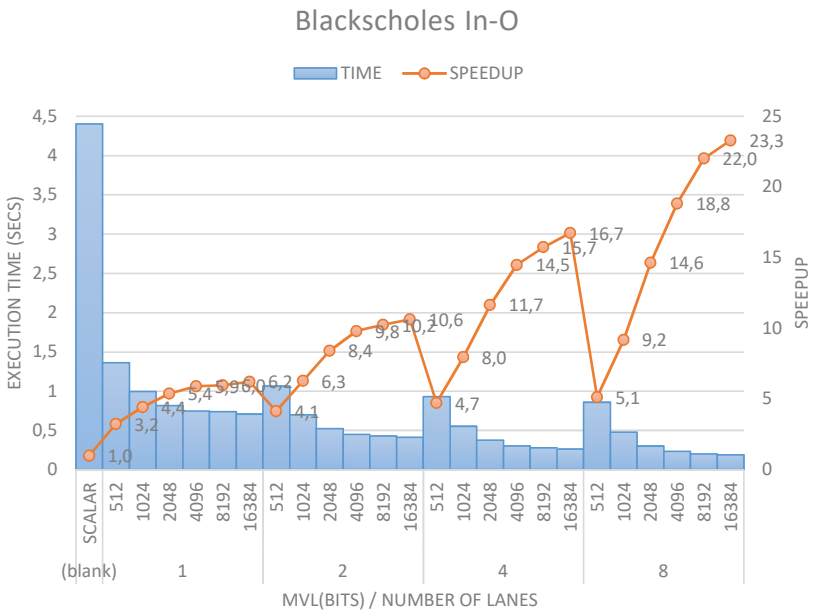
Application		Black-Scholes	Canneal	Heat Diffusion	Particle Filter	Path Finder	Stream cluster	Swaptions
Vector Length	Short	✓	✓	✓	✓	✓	✓	✓
	Medium	✓		✓	✓	✓	✓	✓
	Large	✓		✓	✓	✓		✓
Memory Unit	Unit-stride	✓		✓	✓	✓	✓	✓
	Indexed		✓					
Vector Lane	Arithmetic	✓	✓	✓	✓	✓	✓	✓
	Mask	✓	✓		✓			✓
Interconnection Network	Slides			✓		✓		
	Reductions		✓	✓			✓	✓
Intensive Communication with the Scalar core			✓		✓		✓	

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Blackscholes application calculates the prices for a portfolio of European options analytically with the Black-Scholes partial differential equation (PDE).
It is fully vectorizable, most of the time it runs with the maximum vector length

Blackscholes
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



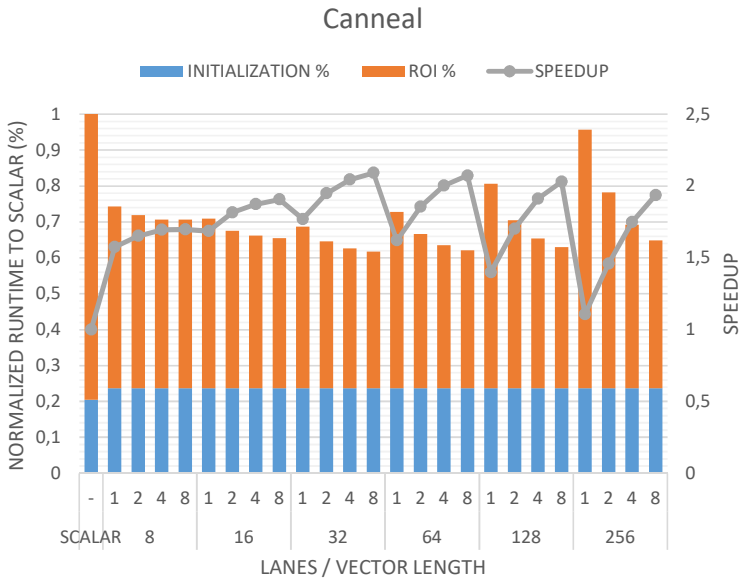
Application		Black-Scholes
Vector Length	Short	✓
	Medium	✓
	Large	✓
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	✓
Interconnection Network	Slides	
	Reductions	
Intensive Communication with the Scalar core		

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Canneal application is focused on minimizing the routing cost of a chip design using cache-aware simulated annealing.
It required a modification in the code to vectorize. The maximum vector length was 44.

Canneal
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



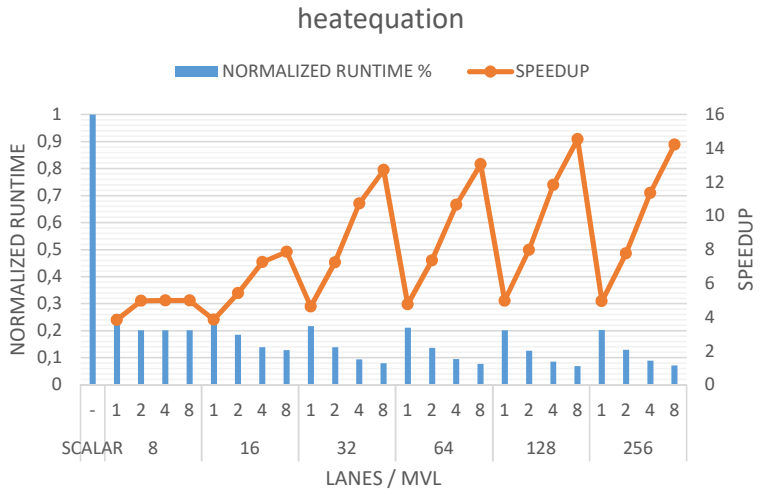
Application		Canneal
Vector Length	Short	✓
	Medium	
	Large	
Memory Unit	Unit-stride	
	Indexed	✓
Vector Lane	Arithmetic	✓
	Mask	✓
Interconnection Network	Slides	
	Reductions	✓
Intensive Communication with the Scalar core		✓

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Heat Diffusion application simulates the diffusion of heat in a solid body using the heat equation. Highly vectorizable code.

Heatequation
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



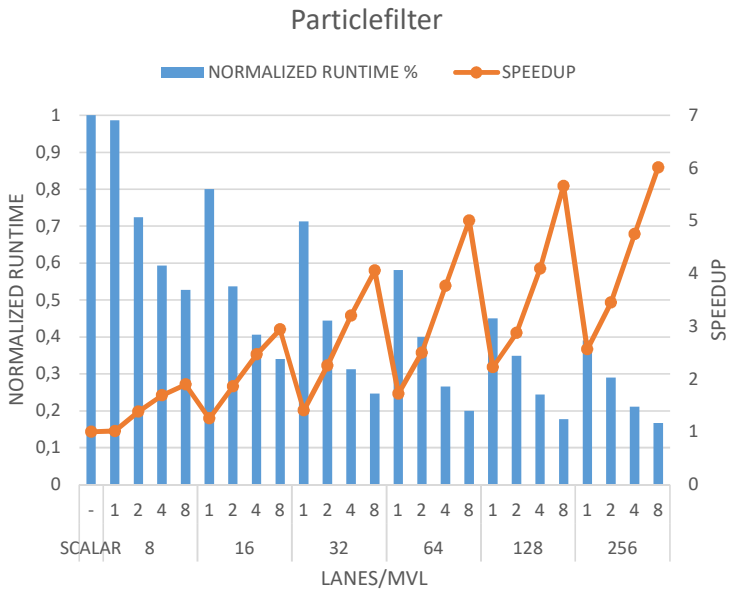
Application		Heat Diffusion
Vector Length	Short	✓
	Medium	✓
	Large	✓
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	
Interconnection Network	Slides	✓
	Reductions	✓
Intensive Communication with the Scalar core		

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Particle Filter (PF) application is a statistical estimator of the location of a target object given noisy measurements of the state. It uses mask related operations like vfirst or vpopc

Particlefilter
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



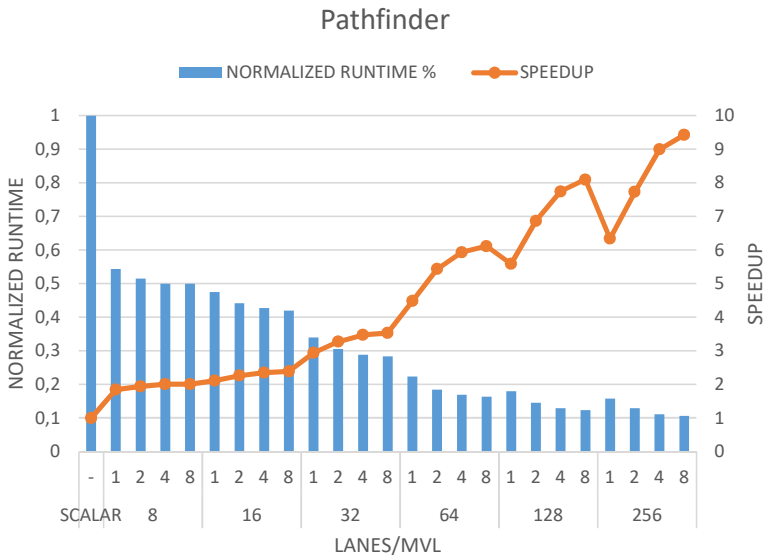
Application		Particle Filter
Vector Length	Short	✓
	Medium	✓
	Large	✓
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	✓
Interconnection Network	Slides	
	Reductions	
Intensive Communication with the Scalar core		✓

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Pathfinder application uses the ghost zone optimization technique to find the shortest paths on a 2-D grid.
Vector element manipulation instructions consumes 26% of the total vector instructions, meaning that the lane interconnection can be stressed by this application

Pathfinder
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



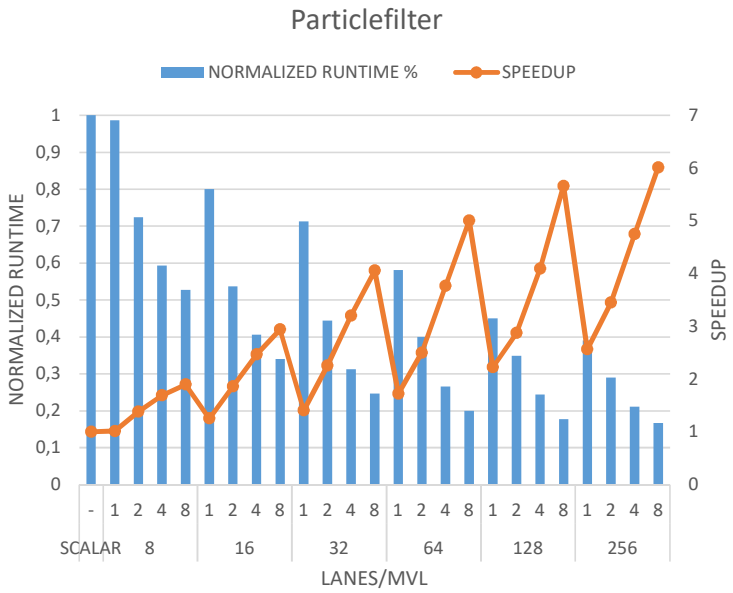
Application		Path Finder
Vector Length	Short	✓
	Medium	✓
	Large	✓
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	
Interconnection Network	Slides	✓
	Reductions	
Intensive Communication with the Scalar core		

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The Streamcluster application solves an online clustering problem.
It is memory bound application

Steamcluster
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



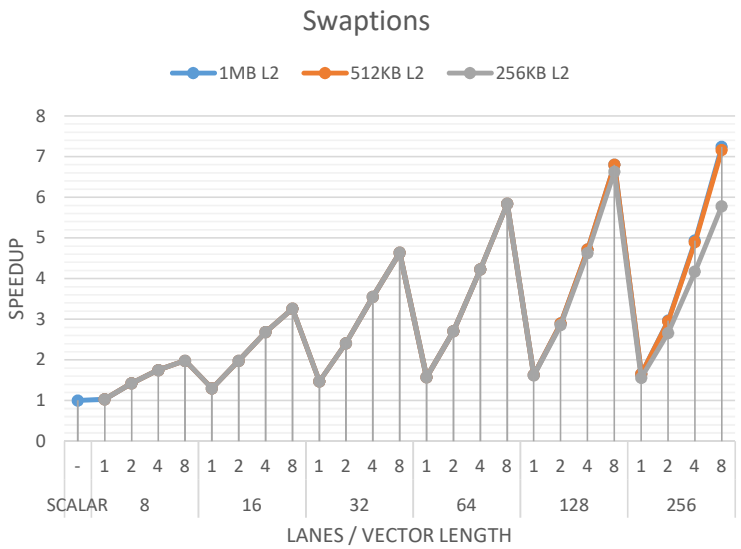
Application		Stream cluster
Vector Length	Short	✓
	Medium	✓
	Large	
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	
Interconnection Network	Slides	
	Reductions	✓
Intensive Communication with the Scalar core		✓

RISC-V Vectorized Benchmark Suite

- Blackscholes
- Canneal
- Heatequation (Jacobi solver)
- Particlefilter
- Pathfinder
- Streamcluster
- Swaptions

The swaptions application uses the Heath-Jarrow-Morton (HJM) framework to price a portfolio of swaptions based on Monte Carlo simulation to compute the prices. Interesting behavior, as larger the vector length, it needs to use large blocks, in consequence, needs more cache.

Swapoptions
Parameters:
-In order Issue logic
-40 physical registers
-MVL from 512-bit up to 16384-bit
-Lanes from 1 up to 8
-Register file - 1 R/W port



Application		Swaptions
Vector Length	Short	✓
	Medium	✓
	Large	✓
Memory Unit	Unit-stride	✓
	Indexed	
Vector Lane	Arithmetic	✓
	Mask	✓
Interconnection Network	Slides	
	Reductions	✓
Intensive Communication with the Scalar core		

Thank you

Questions/comments: adrian.cristal@bsc.es