Verifying RISC-V Vector and Bit Manipulation Extensions using STING Design Verification Tool

Shubhodeep Roy Choudhury, Shajid T, Jevin John, George S
Agenda

- Vector and bit manipulation support – Where we are?
- Overview of STING
- Verification strategy
- Results/Bugs Found
- Future work
Status of Vector and Bit Manipulation Support

- Vector specification version 0.8-draft-20190906
- Bit manipulation specification version 0.9.2
- Support for all the 650+ vector and 100 bit manipulation instructions added to STING
- Work in progress
Overview of STING

- Highly configurable and flexible bare metal program
- Intelligent and architecturally correct instruction sequences complete with results checking
- Supports constrained random, directed and, graph based test generation methodologies
- Portable stimulus – simulations, emulation, FPGA or silicon
- MP and SoC ready
Stimulus Development Framework - Snippets

- Programming framework for RISC-V test development
- High level language constructs mixed with inline assembly
- Allows reservation of registers, memory and hardware threads
- Enables users to install interrupt and exception handlers
- Embedded into the instruction stream after resolving the resource allocation constraints
- Easily integrate existing assembly and C/C++ tests
Rendering of Snippets

- Rendering is the process of embedding the snippet into the instruction stream.

- Rendering of test intent along with different stimulus patterns leads to increased cross product coverage.
Verification Strategy

- Test plan based verification
- Pattern based tests – Fixed and Random
- Constrained random and graph based test generation
- Snippets for directed scenarios
- Cross product with PMP, virtual memory, privilege mode of execution etc.
Fixed Pattern Directed Tests

- Define data sets of fixed input and output patterns for every instruction
- Cover the corner case and interesting values
- Useful for basic testing of ISA implementation
Fixed Pattern Directed Tests (contd)

- Once data sets are ready, snippets are developed to program the input patterns and execute the instruction.

- Results from the execution checkpointed and compared with the expected output.

- Error reported in case a failure is detected during comparison.
Fixed Pattern Directed Tests (contd)

- In case of vectors, the data sets are tested with different configuration settings, which include
  - Supported SEW values
  - Supported LMUL values
  - Different combinations of VSTART and VL
  - Different combinations of mask register (V0)
  - Different combinations of VLEN, ELEN, SLEN and SELEN
Random Pattern Directed Tests

- Fixed pattern tests not adequate to test all input/source values

- Limitation addressed by random pattern tests, where randomly chosen values are programmed into the source register

- The expected output is simulated and compared with the result obtained from the DUT
Random Pattern Directed Tests (contd)

1. Generation of random input
2. Instruction execution
3. Checkpointing of output

Simulated result

Comparison with DUT output checkpointed at an earlier point
Random Tests with Checking

- Pattern based tests are great spot checks but tend to add lot of noise in random tests
- Random tests for increased coverage and interaction with other elements
- Correctness of test execution using cosim checking, multi pass results checking etc.
Random Tests with Checking (contd)

TEST – interleaved_vsb_v

TEST – more_vwmul_v

TEST – only_vlse4b_v
Miscellaneous Tests

- Snippets developed for a large number of directed scenarios to check compliance with the specification
  - Vector register overlap for widening/narrowing instructions and LMUL > 1
  - Verifying if current SEW or LMUL is legal for the executing instruction

- Cross products with existing configurations for PMA, PMP, virtual memory, privileged mode of execution

- Configurations to generate different biased random sequences involving vector and bit manipulation instructions
Results/Bugs Found

- Register overlap check failure for LMUL > 1
- Unexpected load/store address misaligned exceptions on randomizing SEW
- Incorrect decodes of several vector and bit-manipulation instructions
- No trap exception on executing vector atomics with unsupported SEW setting
- Unexpected VL setting at the end of vector fault-only-first load instructions
- Violation of rules in setting VL due to randomization of SEW and AVL
- Incorrect results with vector arithmetic operations
- Irregularities in sign extension of result of vector atomic operations
- Unexpected output of saturating arithmetic instructions on randomization of vector rounding mode (VXRM)
- VXRM/VXSAT not part of FCSR
Future Work

- Add support for the changes introduced by newer revisions of specification
- Improved LMUL randomization
- Handle different random combinations of striping length (SLEN)
- Enhance STING coverage manager for vector and bit manipulation instructions
- Tests for Vector EDIV extension