FCVT support for ACT through RISCOF

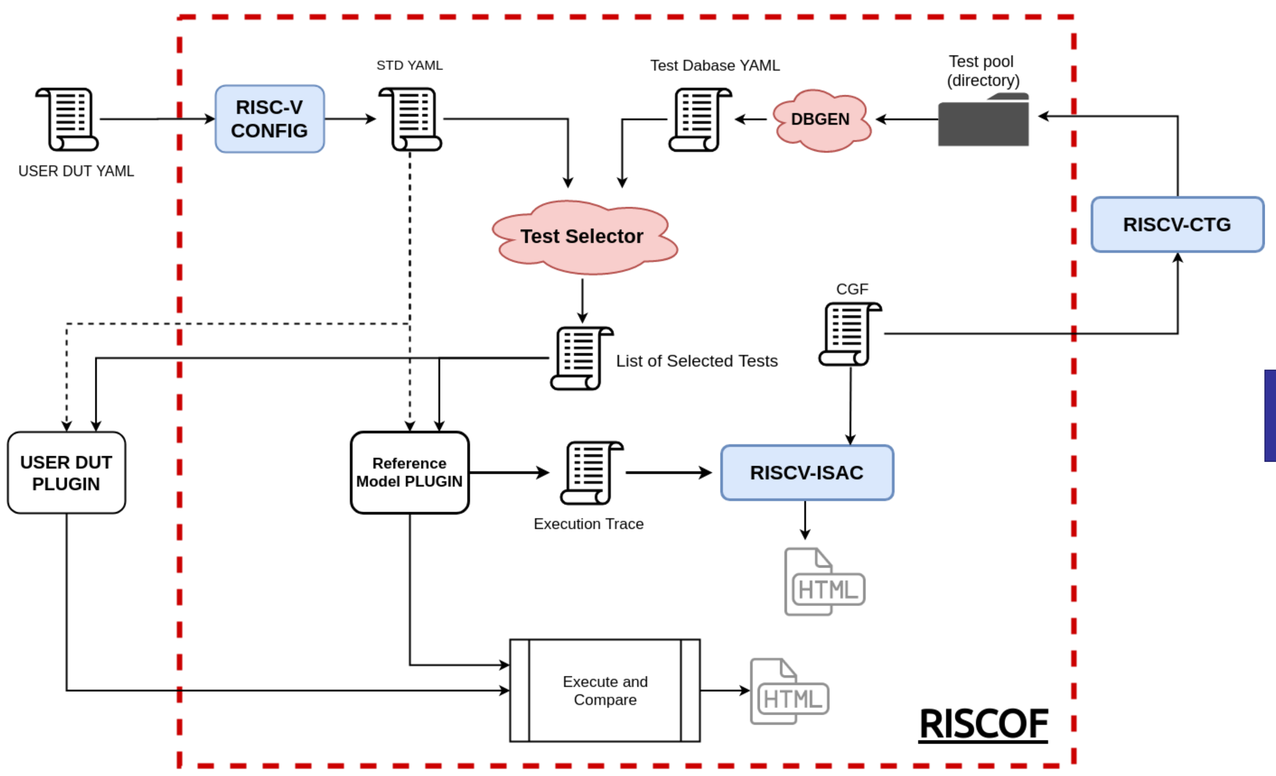
Xuchang Zhu[1], Yunxiang Luo[1]
Programming Language and Compiler Technology Lab, Institute of Software, Chinese Academy of Sciences (ISCAS)
Email: {zhuxuchang,luoyunxiang}@iscas.ac.cn, Tele.: (0086) 18600113892

Motivation

- As the number of RISC-V processor models continues to increase, verifying whether a RISC-V processor complies with the ISA specification has become an important issue.
- As the official testing tool used for ACT testing, RISCOF can leverage the Sail-RISCV model as a reference to check whether the tested model conforms to the specification.
- However, the ACT test repository used by RISCOF lacks support for many test instructions and extensions, including several test cases for the zfh extension, such as `fcvt.d.h`. Therefore, we will add new test instruction support to RISCOF to address this issue.

RISCOF

- RISCOF relies on multiple testing tools for test development. These include RISCV-CTG for generating test cases and RISCV-ISAC for coverage testing.



RISC-V CTG Support

- RISCV-CTG is the RISC-V based Compatibility Test Generator. This tool is used to generate tests used in the official RISC-V Architectural Test Suite and the RISC-V architectural test framework RISCOF.
- To support new test instructions, we need to add YAML nodes for the test instructions in RISC-V CTG to define them and write corresponding CGF files for the instructions.

```yaml
fcvt.d.h:
  sig:
    stride: 2
    sz: 'SIGALIGN'
  val:
    stride: 1
    sz: 'FLEN/8'
    val_template: "'.word $val;'"
    load_instr: "lw"
  xlen: [32,64]
  isa:
    - IFD_Zicsr_Zfh
  flen: [16,32,64]
```

```yaml
fcvt.d.h_b1:
    config:
      - check ISA:=regex(.*I.*F.*D.*Zfh.*)
    mnemonics:
      fcvt.d.h: 0
    rs1:
      <<: *all_fregs
    rd:
      <<: *all_fregs
    op_comb:
      <<: *ifmt_op_comb
    val_comb:
      abstract_comb:
        'ibm_b1(flen,16, "fcvt.d.h", 1)': 0
```

RISC-V ISAC Support

- RISCV-ISAC is primarily split into 2 major parts: the front-end parser and the backedn coverage analyser. This split enables RISCV-ISAC to support parsing of multiple different execution log formats and provide the same level of coverage and QA support.
- we need add the relevant instruction checks to the decoder in ISAC to support the coverage detection of the new instructions.

```python
#fcvt.d.h fcvt.s.h fcvt.h.d fcvt.h.s
if funct7 == 0b100001:
    if rs2[0] == 0b10:
        instrObj.instr_name = 'fcvt.d.h'
        return instrObj
elif funct7 == 0b100000:
    if rs2[0] == 0b10:
        instrObj.instr_name = 'fcvt.s.h'
        return instrObj
elif funct7 == 0b100010:
    if rs2[0] == 0b1:
        instrObj.instr_name = 'fcvt.h.d'
        return instrObj
elif rs2[0] == 0:
    instrObj.instr_name = 'fcvt.h.s'
    return instrObj
```

Result and Conculution

- Afterward, we will run RISCOF with the generated test cases, and the results will show that the test outcomes are accurate.
- By adding support for test instructions to ACT via RISCOF, we can provide greater flexibility for testing, further advancing the comprehensiveness and accuracy of ACT testing. Additionally, RISCOF offers significant editability, allowing modifications to address various scenarios that require ISA specification validation.

Test Stats

Test Name	Mem Footprint (Bytes)	Code size (Bytes)	Data size (Bytes)	Sign size (Bytes)	Covergroups
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/div-01.S	23990	12748	4096	2368	{'div'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/divu-01.S	27256	15464	4096	2908	{'divu'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/mul-01.S	23868	12644	4096	2352	{'mul'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/mulh-01.S	23914	12708	4096	2348	{'mulh'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/mulhsu-01.S	25352	13884	4096	2600	{'mulhsu'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/mulhu-01.S	27204	15428	4096	2904	{'mulhu'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/rem-01.S	23908	12668	4096	2356	{'rem'}
/home/user/Work/Tests/riscv-test-suite/rv32i_m/M/src/remu-01.S	27184	15408	4096	2904	{'remu'}

Coverage Report (Total Coverpoints: 6328)

Show all details / Hide all details

Coverage Label	(Covered-points)/(Total-points)	Percentage
div	731/735 (hide details)	99.46%
divu	869/870 (show details)	99.89%
mul	730/730 (show details)	100.00%
mulh	730/730 (show details)	100.00%
mulhsu	792/792 (show details)	100.00%

```
config:
  - check ISA:=regex(.*I.*M.*)
mnemonics:
    div: 5
    coverage: 1/1
op_comb:
    rd == "x0" != rs1: 0
    rs1 != rs2  and rs1 != rd and rs2 != rd: 586
    rs1 == "x0" != rd: 0
    rs1 == rd != rs2: 1
    rs1 == rd != rs2 and rd != "x0": 0
    rs1 == rd != rs2 and rd == "x0": 1
    rs1 == rs2 != rd: 1
    rs1 == rs2 == rd: 1
    rs2 == rd != rs1: 1
    coverage: 6/9
rd:
```