ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences

智能软件研究中心 Intelligent Software Research Center

从零开始的RISC-V模拟器开发

第15讲 QEMU之ACT测试

中国科学院软件研究所
PLCT实验室

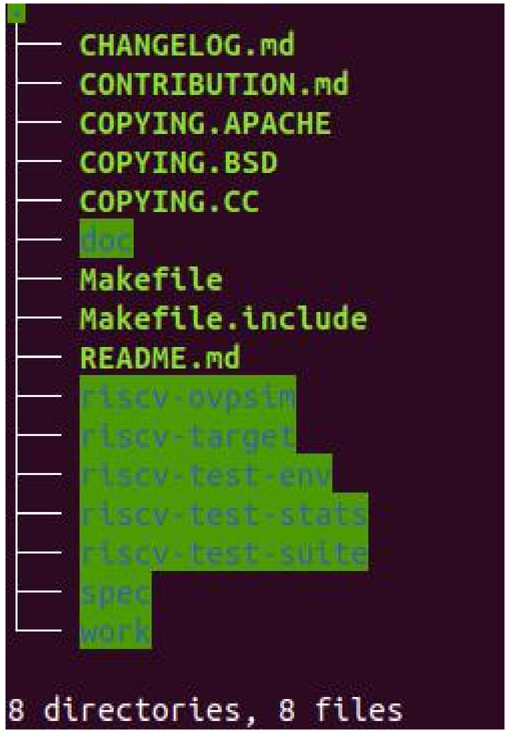
王俊强 wangjunqiang@iscas.ac.cn
李威威 liweiwei@iscas.ac.cn
吴伟 wuwei2016@iscas.ac.cn

ACT测试

- ACT(Archtecture test / Architectural Testing Framework)是一组不断发展的RISC-V测试集，也是一种用于测试RISC-V设备是否正确理解和实现RISC-V规格说明的测试框架
    - 用于辅助保证符合特定RISC-V规格说明的软件能够运行在所有支持该规格说明的设备实现上
    - 该测试集只是一个最低的过滤器，通过该测试集并不意味着设计实现符合RISC-V体系架构，不能代替严格的设计验证
- 通过测试的实现(DUT)可以称为RISC-V架构测试兼容(RISC-V Architecture Test compliant)

github仓库：https://github.com/riscv/riscv-arch-test

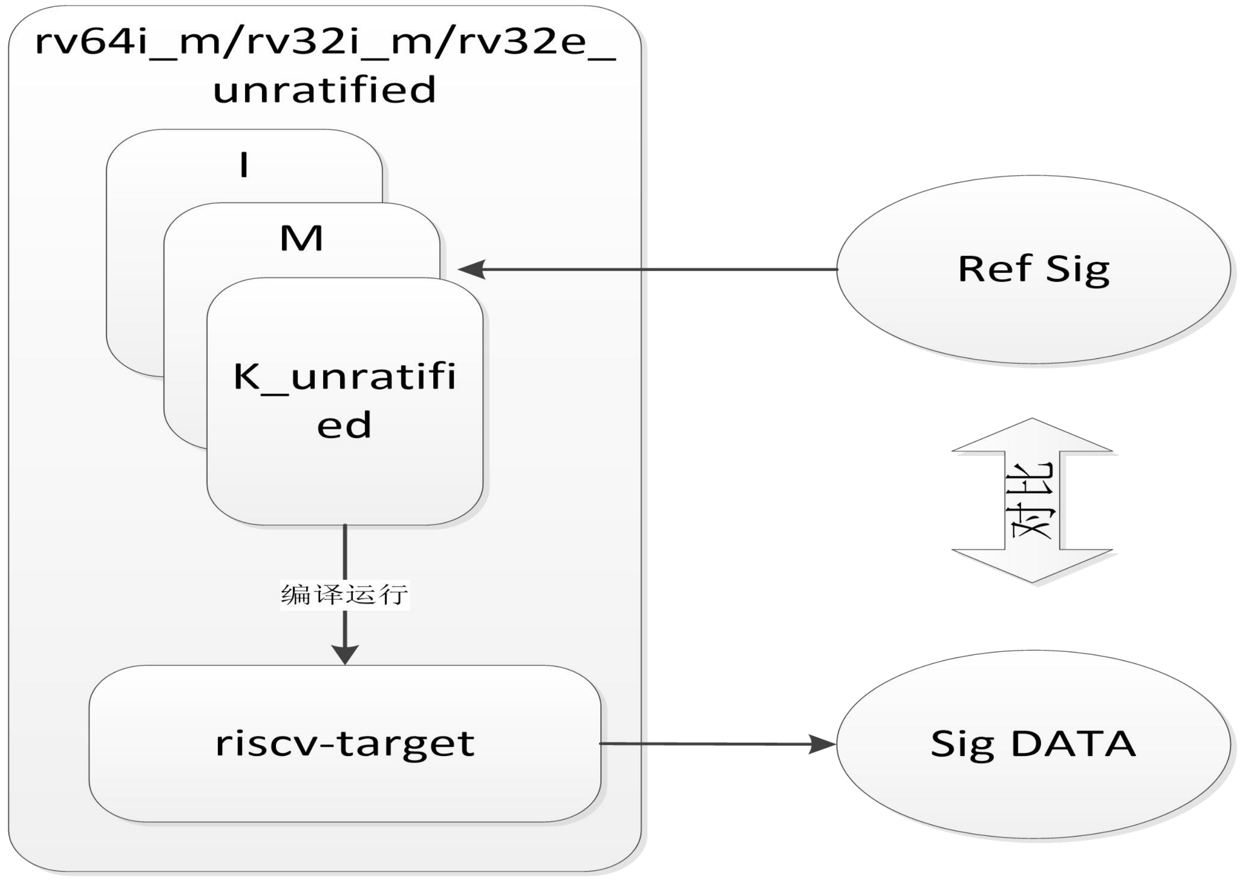
ACT测试代码结构



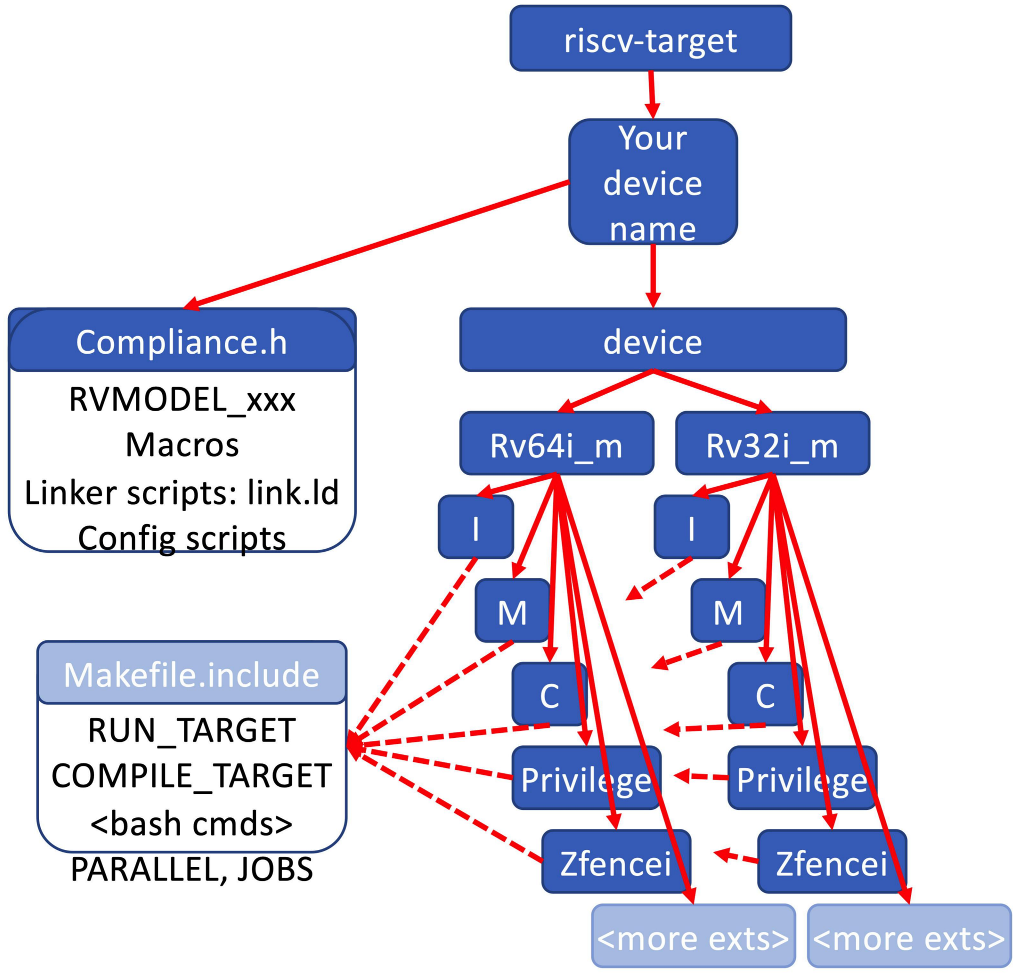
```
.
├── CHANGELOG.md
├── CONTRIBUTION.md
├── COPYING.APACHE
├── COPYING.BSD
├── COPYING.CC
├── doc
├── Makefile
├── Makefile.include
├── README.md
├── riscv-ovpsim
├── riscv-target
├── riscv-test-env
├── riscv-test-stats
├── riscv-test-suite
├── spec
└── work

8 directories, 8 files
```

- doc: 项目相关的文档
- riscv-ovpsim: 包含了移植到Imperas OVP riscvOVPsim模拟器的说明
- riscv-target: 包含了目标相关的文件，例如目标对应的model_test.h头文件以及device目录
- riscv-test-env: 包含了环境相关的文件，例如arch_test.h和encoding.h的链接文件，不同配置下的riscv_test.h文件，以及检查测试结果的verify.sh等
- riscv-test-stats: 包含了测试集的状态报告文件，例如每个测试集的coverage状态以及数据传递状态
- riscv-test-suite: 包含了env目录（包含arch_test.h和encoding.h文件）以及各种测试集的汇编文件以及结果对比文件
- spec: 包含了测试格式规格说明
- work: 编译生成的文件夹，测试案例对应的elf文件以及运行结果都会默认生成到该文件夹中

ACT运行框架

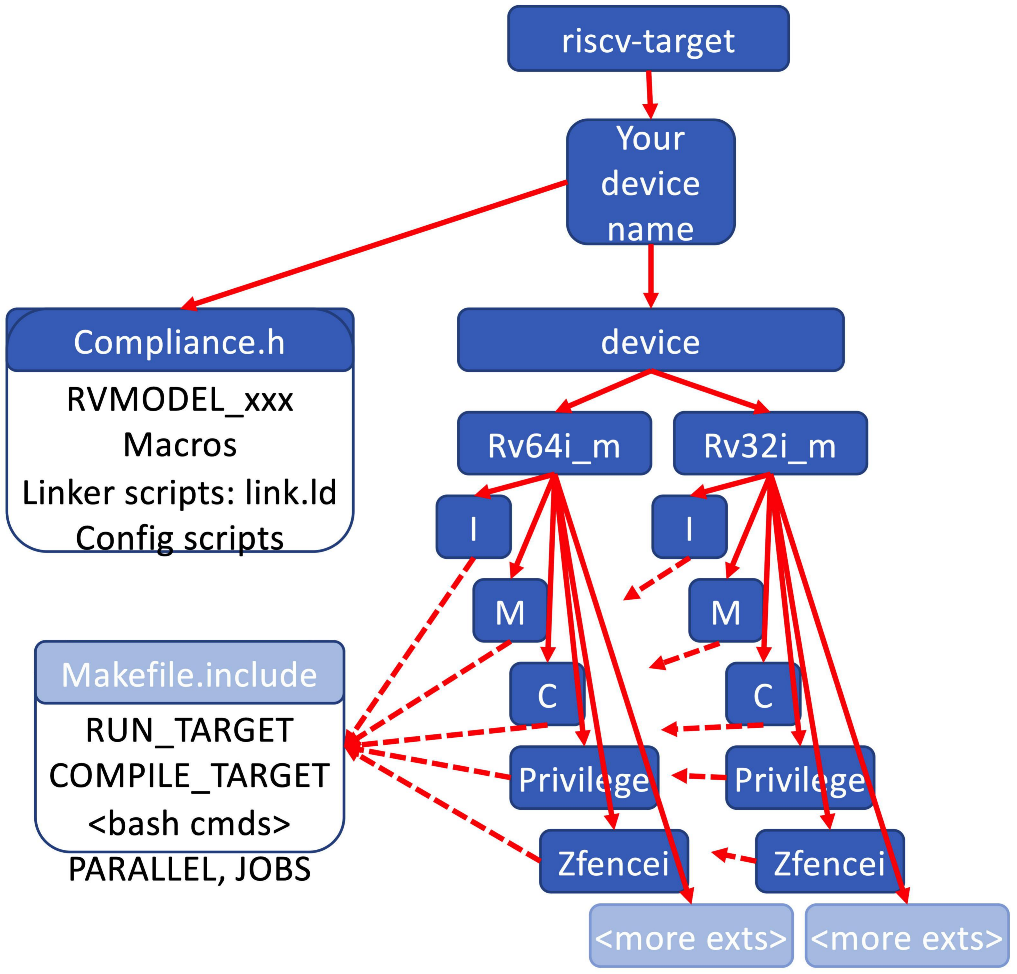

rv64i_m/rv32i_m/rv32e_unratified
I
M
K_unratified
编译运行
riscv-target
Ref Sig
对比
Sig DATA

RISC-V Target



出自https://github.com/riscv/riscv-arch-test/blob/master/doc/file-struct.jpg

RISC-V Target

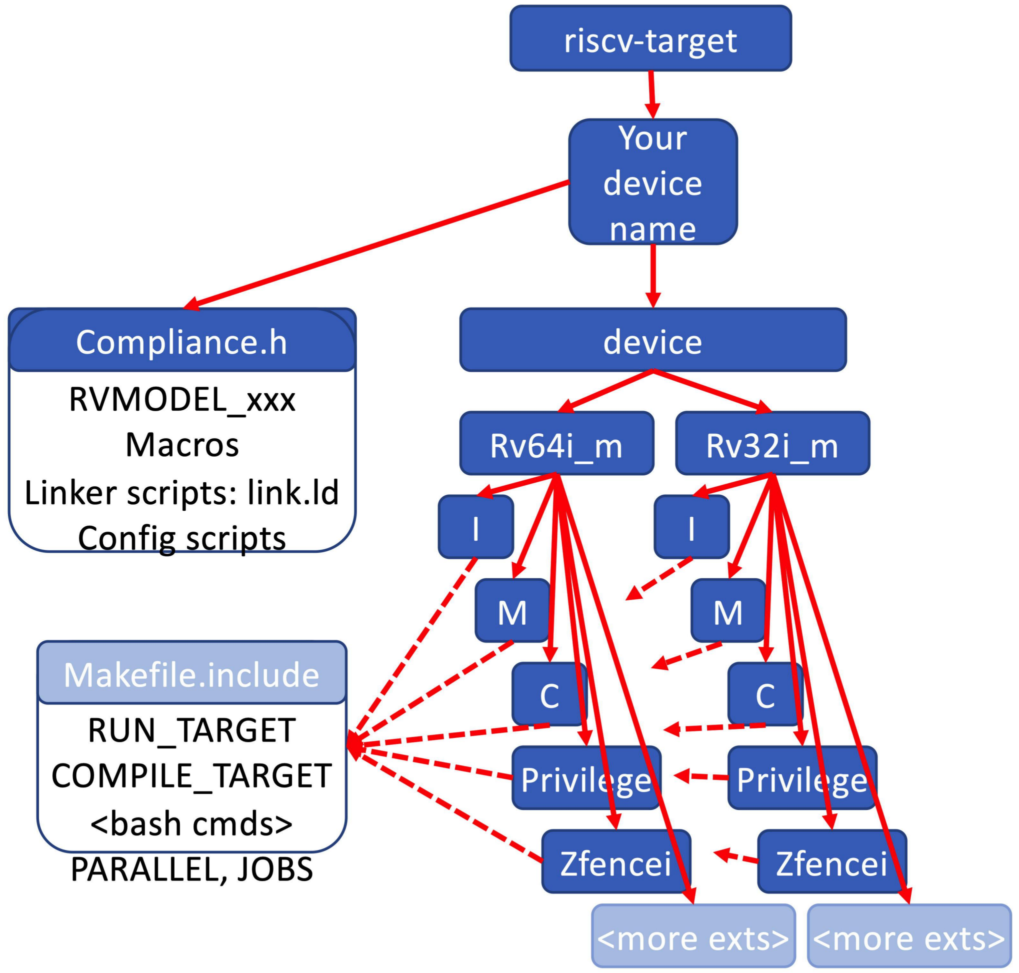


- compliace.h/model_test.h: 包含了用于编译模拟测试案例所需的各种target相关的汇编宏

```
//RV_COMPLIANCE_HALT
#define RVMODEL_HALT \
addi x1, x1, 4; \
li x1, 1; \
write_tohost: \
sw x1, tohost, t5; \
sw zero, tohost + 4, t5; \
self_loop: j self_loop;
```

出自https://github.com/riscv/riscv-arch-test/blob/master/doc/file-struct.jpg

RISC-V Target

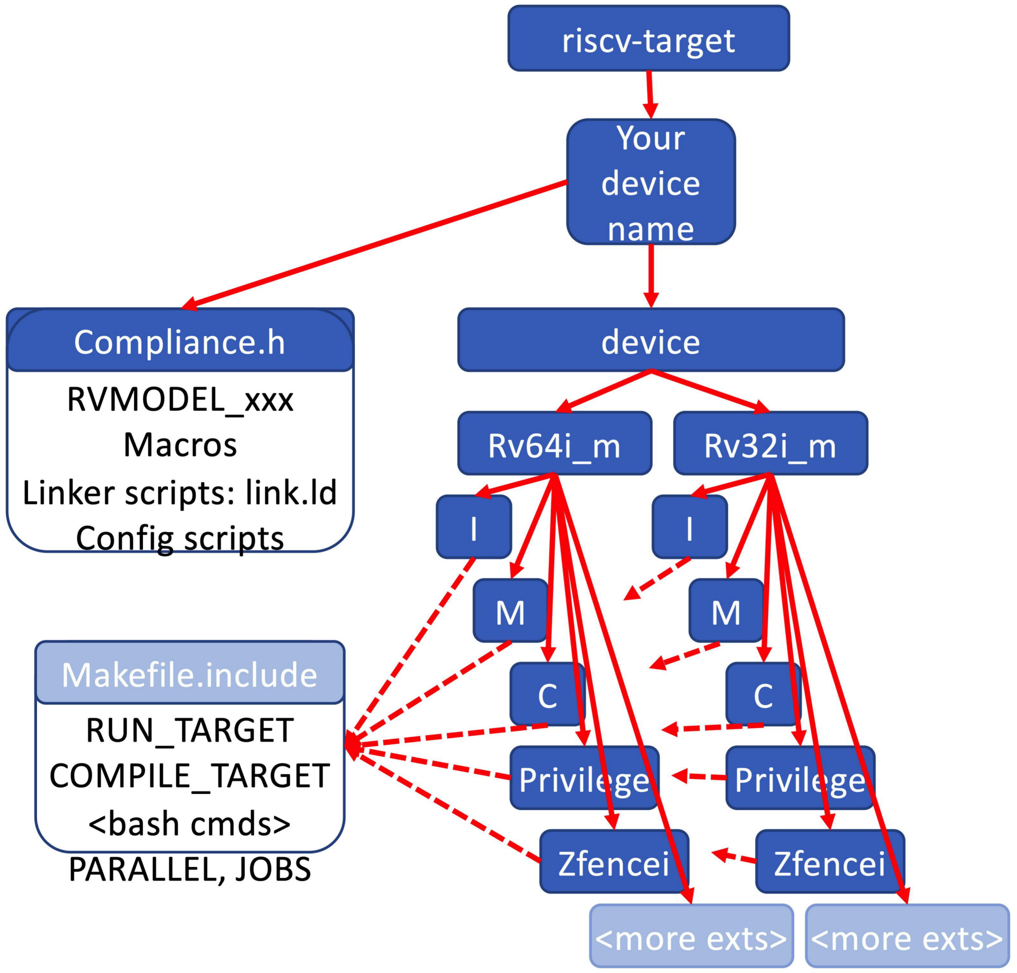


出自https://github.com/riscv/riscv-arch-test/blob/master/doc/file-struct.jpg

- model_test.h: 包含了用于编译模拟测试案例所需的各种target相关的汇编宏
- link.ld: 为目标编译测试案例的linker脚本

```
OUTPUT_ARCH( "riscv" )
ENTRY(rvtest_entry_point)

SECTIONS
{
  . = 0x80000000;
  .text.init : { *(.text.init) }
  . = ALIGN(0x1000);
  .tohost : { *(.tohost) }
  . = ALIGN(0x1000);
  .text : { *(.text) }
  . = ALIGN(0x1000);
  .data : { *(.data) }
  .data.string : { *(.data.string)}
  .bss : { *(.bss) }
  _end = .;
}
```

RISC-V Target

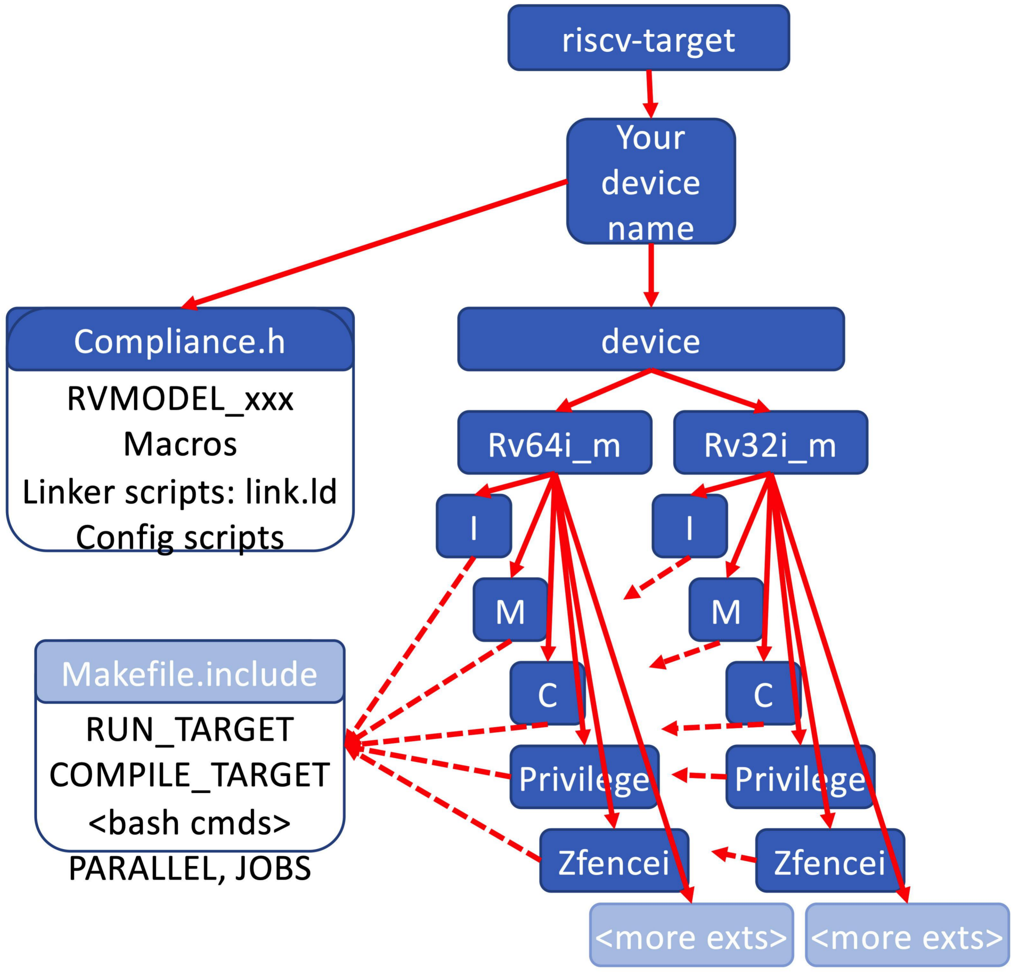


- model_test.h: 包含了用于编译模拟测试案例所需的各种target相关的汇编宏
- link.ld: 为目标编译测试案例的linker脚本
- Makefile.include: 包含了各种环境变量的设置

```
export TARGETDIR ?=
export XLEN ?= 64
export RISCV_TARGET       ?= qemu
export RISCV_DEVICE       ?=
export RISCV_TARGET_FLAGS ?=
export RISCV_ASSERT       ?= 0
JOBS= -j1
```

出自https://github.com/riscv/riscv-arch-test/blob/master/doc/file-struct.jpg

RISC-V Target



- model_test.h: 包含了用于编译模拟测试案例所需的各种target相关的汇编宏
- link.ld: 为目标编译测试案例的linker脚本
- Makefile.include: 包含了各种环境变量的设置
- 子目录中的Makefile.include:包含了在target中编译运行测试集的命令

```
TARGET_SIM   ?= qemu-system-riscv64
TARGET_FLAGS ?= $(RISCV_TARGET_FLAGS)
RISCV_PREFIX   ?= riscv64-unknown-elf-
RISCV_GCC      ?= $(RISCV_PREFIX)gcc
RISCV_OBJDUMP  ?= $(RISCV_PREFIX)objdump
RISCV_GCC_OPTS ?= -g -static -mcmodel=medany -
fvisibility=hidden -nostdlib -nostartfiles $(RVTEST_DEFINES)

COMPILE_TARGET = ...

RUN_TARGET = ...
```

出自https://github.com/riscv/riscv-arch-test/blob/master/doc/file-struct.jpg

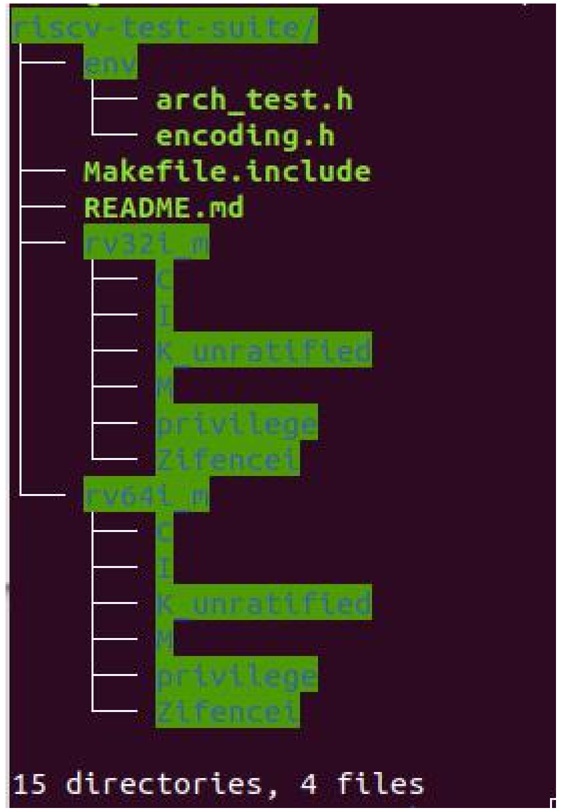
RISC-V Target 编译运行命令

```
COMPILE_CMD = $$(RISCV_GCC) $(1) $$(RISCV_GCC_OPTS) \
                                                    -I$(ROOTDIR)/riscv-test-suite/env/ \
                                                    -I$(TARGETDIR)/$(RISCV_TARGET)/ \
                                                    -T$(TARGETDIR)/$(RISCV_TARGET)/link.ld \
                                                    $$(<) -o $$@
```

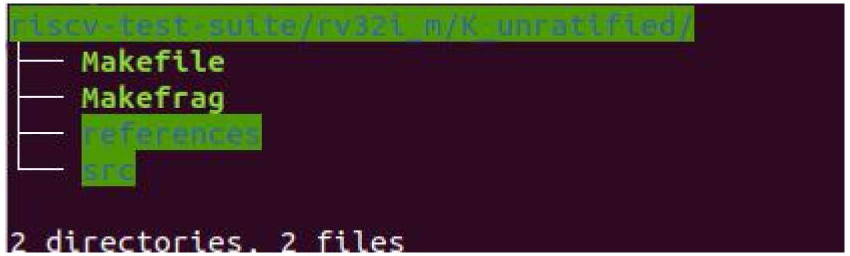
```
RUN_CMD = $(TARGET_SIM) $(TARGET_FLAGS) -bios none -nographic -cpu rv64,x-
k=true,x-b=true\
       -dump-signature-data $(*).signature.output \
       -kernel $<
```

中国科学院软件研究所 Institute of Software Chinese Academy of Sciences | 智能软件研究中心 Intelligent Software Research Center

RISC-V test suite



```
riscv-test-suite/
├── env
│   ├── arch_test.h
│   └── encoding.h
├── Makefile.include
├── README.md
├── rv32i_m
│   ├── C
│   ├── I
│   ├── K_unratified
│   ├── M
│   ├── privilege
│   └── Zifencei
└── rv64i_m
    ├── C
    ├── I
    ├── K_unratified
    ├── M
    ├── privilege
    └── Zifencei

15 directories, 4 files
```



```
riscv-test-suite/rv32i_m/K_unratified/
├── Makefile
├── Makefrag
├── references
└── src

2 directories, 2 files
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
//RV_COMPLIANCE_DATA_BEGIN
#define RVMODEL_DATA_BEGIN \
.align 4; .global begin_signature; begin_signature:

//RV_COMPLIANCE_DATA_END
#define RVMODEL_DATA_END \
.align 4; .global end_signature; end_signature: \
RVMODEL_DATA_SECTION \
```

```
#define RVMODEL_DATA_SECTION \
.pushsection .tohost,"aw",@progbits; \
.align 8; .global tohost; tohost: .dword 0; .size tohost, 8; \
.align 8; .global fromhost; fromhost: .dword 0; .size fromhost, 8; \
.popsection; \
.align 8; .global begin_regstate; begin_regstate: \
.word 128; \
.align 8; .global end_regstate; end_regstate: \
.word 4;
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
.macro RVTEST_CODE_BEGIN
.align UNROLLSZ
.section .text.init;
.globl rvtest_init; \
rvtest_init:
#ifdef rvtest_mtrap_routine
LA(x1, rvtest_trap_prolog );
jalr ra, x1
rvtest_prolog_done:
#endif
LI (x1, (0xFEEDBEADFEEDBEAD & MASK));
LI (x2, (0xFF76DF56FF76DF56 & MASK));
LI (x3, (0x7FBB6FAB7FBB6FAB & MASK));
LI (x4, (0xBFDDB7D5BFDDB7D5 & MASK));
LA (x5, rvtest_code_begin);
LA (x6, rvtest_data_begin);
LI (x7, (0xB7FBB6FAB7FBB6FA & MASK));
LI (x8, (0x5BFDDB7D5BFDDB7D & MASK));
LI (x9, (0xADFEEDBEADFEEDBE & MASK));
…
LI (x30, (0xF76DF56FF76DF56F & MASK));
LI (x31, (0xFBB6FAB7FBB6FAB7 & MASK));
.globl rvtest_code_begin
rvtest_code_begin:
.endm
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
RVTEST_SIGBASE( x1,signature_x1_1)

inst_0:
// rs1 == rs2 != rd, rs1==x6, rs2==x6, rd==x23, rs1_val == 0x0706050403020100
and rs2_val == 0x0f0e0d0c0b0a0908
// opcode: aes64ds ; op1:x6; op2:x6; dest:x23; op1val:0x706050403020100;
op2val:0x706050403020100
TEST_RR_OP(aes64ds, x23, x6, x6, 0x0000000000000000, 0x706050403020100,
0x706050403020100, x1, 0, x5)
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
RVTEST_SIGBASE( x1,signature_x1_1)

inst_0:
// rs1 == rs2 != rd, rs1==x6, rs2==x6, rd==x23, rs1_val == 0x0706050403020100
and rs2_val == 0x0f0e0d0c0b0a0908
// opcode: aes64ds ; op1:x6; op2:x6; dest:x23; op1val:0x706050403020100;
op2val:0x706050403020100
TEST_RR_OP(aes64ds, x23, x6, x6, 0x0000000000000000, 0x706050403020100,
0x706050403020100, x1, 0, x5)
```

```
#define RVTEST_SIGBASE(_R,_TAG) \
LA(_R,_TAG);\
.set offset,0;
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
RVTEST_SIGBASE( x1,signature_x1_1)

inst_0:
// rs1 == rs2 != rd, rs1==x6, rs2==x6, rd==x23, rs1_val == 0x0706050403020100
and rs2_val == 0x0f0e0d0c0b0a0908
// opcode: aes64ds ; op1:x6; op2:x6; dest:x23; op1val:0x706050403020100;
op2val:0x706050403020100
TEST_RR_OP(aes64ds, x23, x6, x6, 0x0000000000000000, 0x706050403020100,
0x706050403020100, x1, 0, x5)
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
RVTEST_SIGBASE( x1,signature_x1_1)

inst_0:
// rs1 == rs2 != rd, rs1==x6, rs2==x6, rd==x23, rs1_val == 0x0706050403020100
and rs2_val == 0x0f0e0d0c0b0a0908
// opcode: aes64ds ; op1:x6; op2:x6; dest:x23; op1val:0x706050403020100;
op2val:0x706050403020100
TEST_RR_OP(aes64ds, x23, x6, x6, 0x0000000000000000, 0x706050403020100,
0x706050403020100, x1, 0, x5)
```

```
#define TEST_RR_OP(inst, destreg, reg1, reg2, correctval, val1,
val2, swreg, offset, testreg) \
TEST_CASE(testreg, destreg, correctval, swreg, offset, \
LI(reg1, MASK_XLEN(val1)); \
LI(reg2, MASK_XLEN(val2)); \
inst destreg, reg1, reg2; \
```

Test 构成

```
#define TEST_CASE(testreg, destreg, correctval, swreg, offset, code... ) \
code; \
RVTEST_SIGUPD(swreg,destreg,offset); \
RVMODEL_IO_ASSERT_GPR_EQ(testreg, destreg, correctval)
```

```
#define RVTEST_SIGUPD(_BR,_R,...)\
.if NARG(__VA_ARGS__) == 1;\
SREG _R,_ARG1(__VA_ARGS__,0)(_BR);\
.set offset,_ARG1(__VA_OPT__(__VA_ARGS__,)0)+REGWIDTH;\
.endif;\
.if NARG(__VA_ARGS__) == 0;\
SREG _R,offset(_BR);\
.set offset,offset+REGWIDTH;\
.endif;
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

```
RVTEST_DATA_BEGIN
.align 4
rvtest_data:
.word 0xbabecafe
RVTEST_DATA_END
```

```
RVTEST_DATA_BEGIN
.align 4
rvtest_data:
.dword 0x08577eb1924770d3
.dword 0x93fdcab87b89296c
.dword 0xd2d6b8777dc59a3a
.dword 0xcf84b683a749f9c5
.dword 0x854a965708ceac39
.dword 0x137a977753e8eb43
。 。 。
.dword 0x807da245d814d575
.dword 0x3d06143769b1dcbf
.dword 0x7f21682208208d09
.dword 0x14b91c79dae98554
.dword 0xc5ec6148c6880007
.dword 0x7213516d6a013380
.dword 0x4652f62dae4839a1
.dword 0x85986adb9e044706
RVTEST_DATA_END
```

Test 构成

```
#include "model_test.h"
#include "arch_test.h"
RVTEST_ISA("RV64IK")

.section .text.init
.globl rvtest_entry_point
rvtest_entry_point:
RVMODEL_BOOT
RVTEST_CODE_BEGIN
…(code)
RVTEST_CODE_END
RVMODEL_HALT

RVTEST_DATA_BEGIN
...(data)
RVTEST_DATA_END

RVMODEL_DATA_BEGIN
… (signature data)
RVMODEL_DATA_END
```

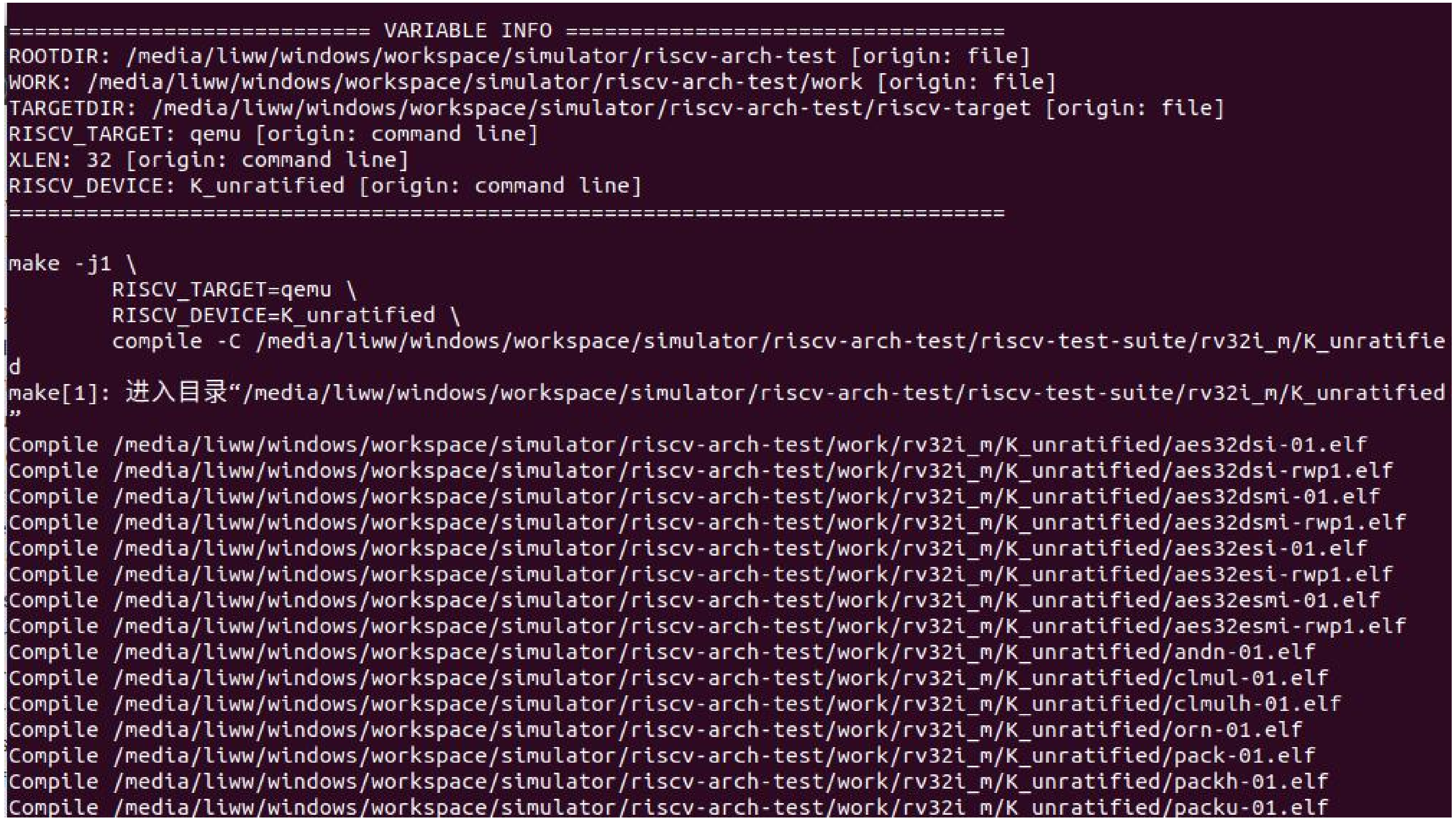
```
signature_x31_1:
    .fill 78*(XLEN/32),4,0xdeadbeef
```

```
signature_x6_0:
    .fill 0*(XLEN/32),4,0xdeadbeef
signature_x6_1:
    .fill 21*(XLEN/32),4,0xdeadbeef
signature_x4_0:
    .fill 256*(XLEN/32),4,0xdeadbeef
signature_x4_1:
    .fill 256*(XLEN/32),4,0xdeadbeef
signature_x4_2:
    .fill 136*(XLEN/32),4,0xdeadbeef
```

运行举例

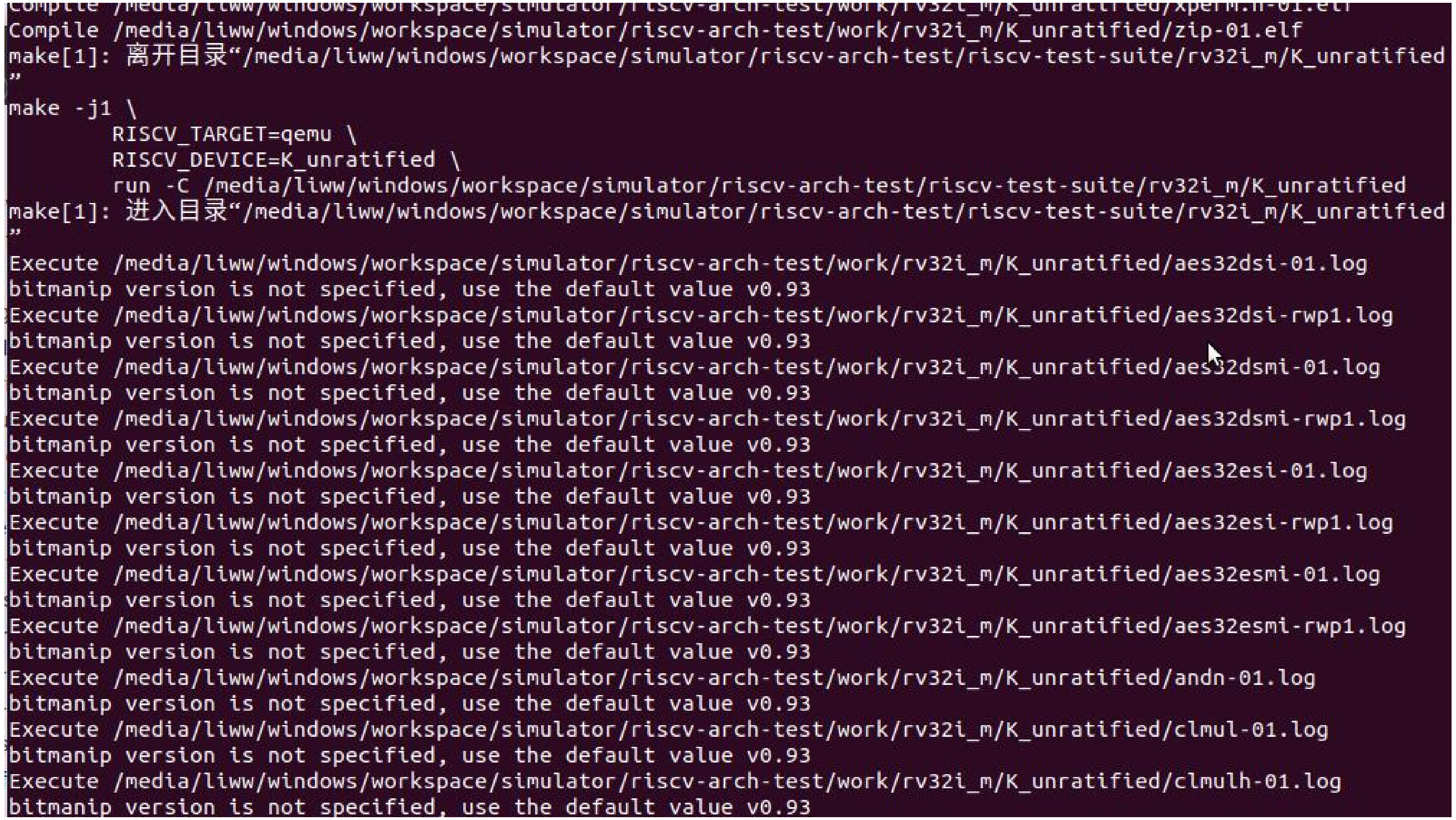
```
make RISCV_TARGET=qemu RISCV_DEVICE=K_unratified
TARGET_SIM=/media/liww/windows/workspace/dragon/plct-qemu/build/qemu-system-riscv32
RISCV_PREFIX=/workspace/riscv/build/bin/riscv64-unknown-elf- compile simulate verify
XLEN=32
```

运行举例



```
============================ VARIABLE INFO ==================================
ROOTDIR: /media/liww/windows/workspace/simulator/riscv-arch-test [origin: file]
WORK: /media/liww/windows/workspace/simulator/riscv-arch-test/work [origin: file]
TARGETDIR: /media/liww/windows/workspace/simulator/riscv-arch-test/riscv-target [origin: file]
RISCV_TARGET: qemu [origin: command line]
XLEN: 32 [origin: command line]
RISCV_DEVICE: K_unratified [origin: command line]
=============================================================================

make -j1 \
        RISCV_TARGET=qemu \
        RISCV_DEVICE=K_unratified \
        compile -C /media/liww/windows/workspace/simulator/riscv-arch-test/riscv-test-suite/rv32i_m/K_unratifie
d
make[1]: 进入目录“/media/liww/windows/workspace/simulator/riscv-arch-test/riscv-test-suite/rv32i_m/K_unratified
”
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsi-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsi-rwp1.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsmi-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsmi-rwp1.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esi-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esi-rwp1.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esmi-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esmi-rwp1.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/andn-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/clmul-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/clmulh-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/orn-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/pack-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/packh-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/packu-01.elf
```

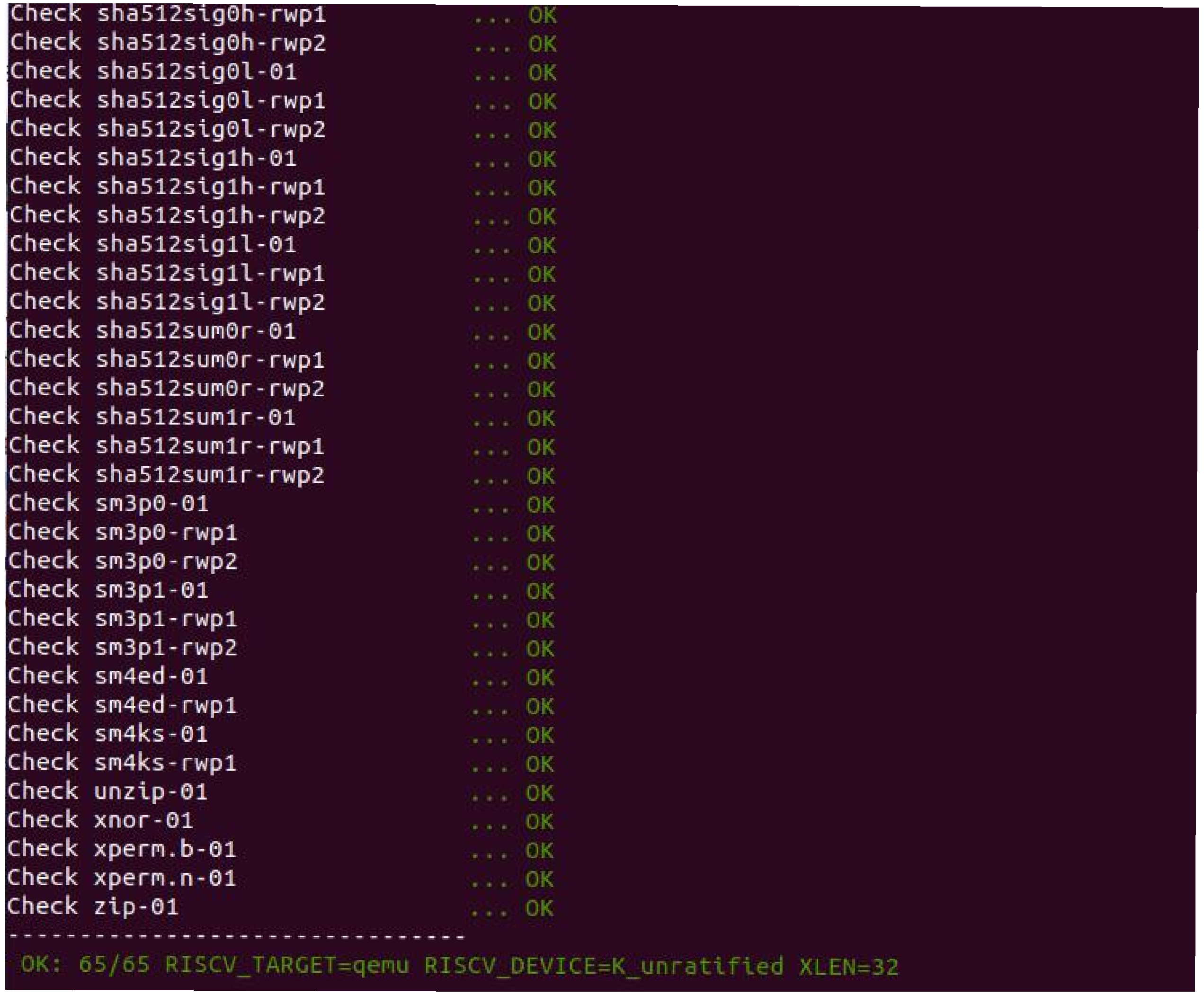
运行举例



```
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/xperm.n-01.elf
Compile /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/zip-01.elf
make[1]: 离开目录“/media/liww/windows/workspace/simulator/riscv-arch-test/riscv-test-suite/rv32i_m/K_unratified
”
make -j1 \
        RISCV_TARGET=qemu \
        RISCV_DEVICE=K_unratified \
        run -C /media/liww/windows/workspace/simulator/riscv-arch-test/riscv-test-suite/rv32i_m/K_unratified
make[1]: 进入目录“/media/liww/windows/workspace/simulator/riscv-arch-test/riscv-test-suite/rv32i_m/K_unratified
”
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsi-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsi-rwp1.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsmi-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32dsmi-rwp1.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esi-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esi-rwp1.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esmi-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/aes32esmi-rwp1.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/andn-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/clmul-01.log
bitmanip version is not specified, use the default value v0.93
Execute /media/liww/windows/workspace/simulator/riscv-arch-test/work/rv32i_m/K_unratified/clmulh-01.log
bitmanip version is not specified, use the default value v0.93
```

ISCAS 中国科学院软件研究所 Institute of Software Chinese Academy of Sciences | 智能软件研究中心 Intelligent Software Research Center

运行举例



```
Check sha512sig0h-rwp1          ... OK
Check sha512sig0h-rwp2          ... OK
Check sha512sig0l-01            ... OK
Check sha512sig0l-rwp1          ... OK
Check sha512sig0l-rwp2          ... OK
Check sha512sig1h-01            ... OK
Check sha512sig1h-rwp1          ... OK
Check sha512sig1h-rwp2          ... OK
Check sha512sig1l-01            ... OK
Check sha512sig1l-rwp1          ... OK
Check sha512sig1l-rwp2          ... OK
Check sha512sum0r-01            ... OK
Check sha512sum0r-rwp1          ... OK
Check sha512sum0r-rwp2          ... OK
Check sha512sum1r-01            ... OK
Check sha512sum1r-rwp1          ... OK
Check sha512sum1r-rwp2          ... OK
Check sm3p0-01                  ... OK
Check sm3p0-rwp1                ... OK
Check sm3p0-rwp2                ... OK
Check sm3p1-01                  ... OK
Check sm3p1-rwp1                ... OK
Check sm3p1-rwp2                ... OK
Check sm4ed-01                  ... OK
Check sm4ed-rwp1                ... OK
Check sm4ks-01                  ... OK
Check sm4ks-rwp1                ... OK
Check unzip-01                  ... OK
Check xnor-01                   ... OK
Check xperm.b-01                ... OK
Check xperm.n-01                ... OK
Check zip-01                    ... OK
--------------------------------
 OK: 65/65 RISCV_TARGET=qemu RISCV_DEVICE=K_unratified XLEN=32
```

Test生成

```
// -----------
// This file was generated by riscv_ctg (https://gitlab.com/incoresemi/riscv-compliance/riscv_ctg)
// version   : 0.4.5
// timestamp : Thu May 20 05:32:05 2021 GMT
// usage     : riscv_ctg \
//               --cgf /scratch/git-repo/github/riscv-ctg/sample_cgfs/dataset.cgf \
//               --cgf /scratch/git-repo/github/riscv-ctg/sample_cgfs/rv64i_k.cgf \
//               --base-isa rv64i \
//               --randomize
// -----------
```

Test生成——dataset.cgf

```
all_regs: &all_regs
   x0: 0
   x1: 0
   x2: 0
   x3: 0
   x4: 0
   x5: 0
   x6: 0
   x7: 0
   x8: 0
   x9: 0
   x10: 0
   x11: 0
   x12: 0
   x13: 0
   x14: 0
   x15: 0
   ...
```

```
rfmt_op_comb: &rfmt_op_comb
  'rs1 == rs2 != rd': 0
  'rs1 == rd != rs2': 0
  'rs2 == rd != rs1': 0
  'rs1 == rs2 == rd': 0
  'rs1 != rs2  and rs1 != rd and rs2 != rd': 0
```

Test生成——rv64i_k.cgf

```
aes64ds:
  config:
   - check ISA:=regex(.*RV64.*I.*K.*)
   - check ISA:=regex(.*RV64.*I.*ZKn.*)
   - check ISA:=regex(.*RV64.*I.*ZKnd.*)
  opcode:
   aes64ds: 0
  rs1:
   <<: *all_regs
  rs2:
   <<: *all_regs
  rd:
   <<: *all_regs
  op_comb:
   <<: *rfmt_op_comb
  val_comb:
   abstract_comb:
    'byte_count(64, ["rs1_val","rs2_val"])': 0
    'uniform_random(20, 100, ["rs1_val","rs2_val"], [64, 64])': 0
```

Test生成——template.yaml

```
aes64ds:
  xlen: [64]
  std_op:
  isa: IK
  formattype: 'rformat'
  rs1_op_data: *all_regs
  rs2_op_data: *all_regs
  rd_op_data: *all_regs
  template: |-

    // $comment
   // opcode: $inst ; op1:$rs1; op2:$rs2; dest:$rd; op1val:$rs1_val; op2val:$rs2_val
   TEST_RR_OP($inst, $rd, $rs1, $rs2, $correctval, $rs1_val, $rs2_val, $swreg, $offset, $testreg)
```

其它测试

- RISU: Random Instruction Sequence generator for Userspace testing
  - 随机指令序列生成器
  - 适用于用户态测试:仅针对linux用户空间可见的部分进行测试

  https://git.linaro.org/people/peter.maydell/risu.git
  https://lore.kernel.org/qemu-devel/20200711161655.2856-1-zhiwei_liu@c-sky.com/
  https://www.bilibili.com/video/BV1FZ4y1M7zG ——石可人
  https://www.bilibili.com/video/BV17A411E7PS?p=3

- Qtest: 是QEMU自身提供的一种针对设备的单元测试框架
  https://www.bilibili.com/video/BV1Kf4y127op —— 陈嘉炜
  https://www.bilibili.com/video/BV1pA411A7rZ —— 胡轩

谢谢

liweiwei@iscas.ac.cn