

## CV32E40P Verification

Name: shiva karthik

### 1. Tool Versions

- GCC (RISC-V): riscv32-unknown-elf-gcc 15.2.0
- Verilator: 5.044
- Python: 3.10.12
- GTKWave: 3.3.104
- Git: 2.34.1

### 2. RTL Source Explanation

The RTL source code for the CV32E40P processor is not included directly in the core-v-verif repository because the project follows a modular and reusable verification methodology. The core-v-verif repository is designed to provide a common verification framework and testbench infrastructure that can be reused for multiple CORE-V processor implementations. Keeping the RTL in a separate repository allows independent development, version control, and maintenance of the processor design. During simulation, the testbench gains access to the RTL by automatically cloning the official CV32E40P repository from GitHub using scripts defined in the Makefile. The required version of the RTL is checked out and integrated into the simulation environment through file lists and build scripts, enabling the testbench to compile and verify the processor without permanently storing the RTL source code in the verification repository.

### 3. Friction Point and Resolution

During the setup and validation of the CORE-V-VERIF environment, several technical challenges were encountered. Initially, only the riscv64-unknown-elf-gcc toolchain was available, while the CV32E40P verification environment required the riscv32-unknown-elf-gcc compiler. This mismatch caused compilation failures and required rebuilding the RISC-V GNU toolchain with Newlib support for the RV32 architecture. While building Newlib, permission errors occurred when installing to /opt/riscv32, which were resolved by configuring proper installation paths and environment variables. Additionally, an outdated version of Verilator initially prevented proper waveform generation and tracing, which was fixed by upgrading to Verilator version 5.044. Another major issue occurred when attempting to run the custom mscratch\_test, where the simulation continued to execute the default hello-world program instead of the intended test. This problem was traced to missing or incorrectly configured test.yaml files and environment variables, causing the build system to fall back to the default test configuration. After properly defining the test metadata and toolchain variables, the custom test was successfully compiled and executed.

## 4. Waveform Generation

Waveforms were generated using WAVES=1 flag. The generated .vcd file was viewed using GTKWave to analyze instruction fetch behavior.

## 5. mscratch Test Description

A custom program was written to write and read 32-bit values to the mscratch CSR and verify correctness.

Code mscratch\_test.c

```
#include <stdint.h>
#include <stdio.h>

static inline void write_csr(uint32_t val) {
    asm volatile ("csrw mscratch, %0" :: "r"(val));
}

static inline uint32_t read_csr(void) {
    uint32_t val;
    asm volatile ("csrr %0, mscratch" : "=r"(val));
    return val;
}

int main() {
    uint32_t patterns[] = {
        0x00000000,
        0xFFFFFFFF,
        0xA5A5A5A5,
        0x5A5A5A5A
    };

    printf("Starting mscratch test...\n");

    for (int i = 0; i < 4; i++) {
        write_csr(patterns[i]);
        uint32_t r = read_csr();

        printf("Write: 0x%08X Read: 0x%08X\n",
            patterns[i], r);

        if (r != patterns[i]) {
            printf("FAIL at index %d\n", i);
            while (1); // stop here
        }
    }

    printf("MSCRATCH TEST PASSED\n");

    return 0;
}
```

test.yaml

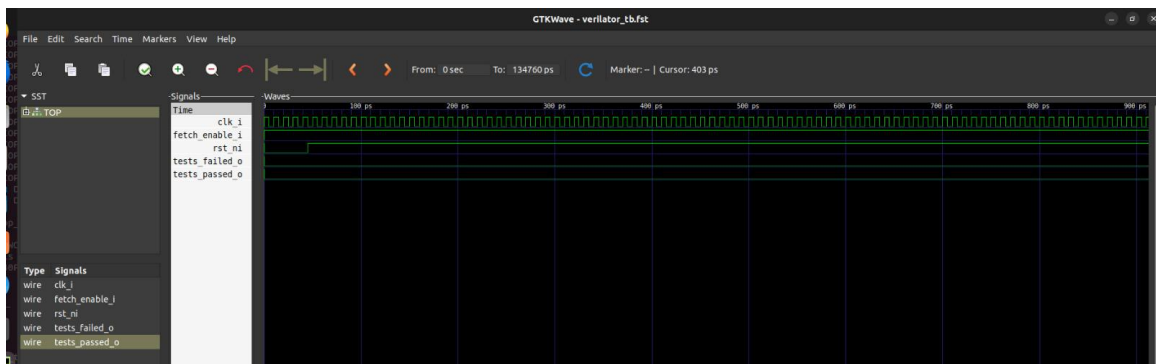
```
name: mscratch_test
description: Test full 32-bit read/write of mscratch CSR
uvm_test: corev_direct_test
program: mscratch_test.c
build:
  gcc_opts: "-march=rv32imc_zicsr -mabi=ilp32"
sim:
  timeout: 200000
```

## 6. Validation Evidence

### HELLO WORLD TERMINAL OUTPUT

```
Activities Terminal Feb 7 22:55
karthik07@karthik07-VirtualBox: ~/Desktop/core-v-verif/cv32e40p/sim/core
$- \
-M no-aliases \
-M numeric \
-I \
/home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/hello-world/hello-world.elf | /home/karthik07/Desktop/core-v-verif/bin/objdump2l1tb -> /home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/hello-world/hello-world.ltb
* Running with Verilator: logfile in simulation_results/hello-world/hello-world.log
mkdir -p simulation_results/hello-world/0/test_program
simulation_results/hello-world/verilator_executable \
  *firmware.../tests/programs/custom/hello-world/hello-world.hex" \
  | tee simulation_results/hello-world/0/test_program/hello-world.log
ScopesDump:
SCOPE 0x5b07aadc230: TOP.tb_top_verilator
SCOPE 0x5b07aadc3a0: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.cs_registers_i
SCOPE 0x5b07aadc390: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.cs_registers_i.gen_no_pulp_secure_write_logic
SCOPE 0x5b07aadc3a0: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.ex_stage_i.alu_i
SCOPE 0x5b07aadc390: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i
SCOPE 0x5b07aadc370: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.controller_i
SCOPE 0x5b07aadc360: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.controller_i.blk_decode_level1
SCOPE 0x5b07aadc360: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.decoder_i
SCOPE 0x5b07aadc350: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.tweedate_a_mux
SCOPE 0x5b07aadc340: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.jump_target_mux
SCOPE 0x5b07aadc330: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i
SCOPE 0x5b07aadc320: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.EXC_PC_MUX
SCOPE 0x5b07aadc310: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.compressed_decoder_i
SCOPE 0x5b07aadc300: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.prefetch_buffer_i.fifo_i
SCOPE 0x5b07aadd000: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.ram_i
DPI-EXPORT 0x5b079313d640: read_byte
DPI-EXPORT 0x5b079313d650: write_byte
[tb_top_verilator] finished dumping memory
HELLO WORLD!!!
This is the OpenHW Group CV32E40P CORE-V processor core.
CV32E40P is a RISC-V ISA compliant core with the following attributes:
mvendorid = 0x652
marchid = 0x4
mimpid = 0x0
misa = 0x40001104
XLEN is 32-bits
Supported Instructions Extensions: MIC
TOP.tb_top_verilator @ 134700: EXIT SUCCESS
- /home/karthik07/Desktop/core-v-verif/cv32e40p/tb/core/tb_top_verilator.sv:83: Verilog $finish
nake(1): leaving directory '/home/karthik07/Desktop/core-v-verif/cv32e40p/sim/core'
karthik07@karthik07-VirtualBox: ~/Desktop/core-v-verif/cv32e40p/sim/core$
```

### WAVEFORM VIEWED



## Mscratch 32 bit read and write terminal output

```
Activities Terminal Feb 8 22:11
karthik07@karthik07-VirtualBox: ~/Desktop/core-v-verif/cv32e40p/sim/core

-H numeric \
-S \
/home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/mscratch_test/mscratch_test.elf > /home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/mscratch_test/mscratch_test.elf
tobjdump
/opt/riscv32/bin/riscv32-unknown-elf-objdump \
-d \
-S \
-H no-allases \
-H numeric \
-l \
/home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/mscratch_test/mscratch_test.elf | /home/karthik07/Desktop/core-v-verif/cv32e40p/tests/programs/custom/mscratch_test/mscratch_test.elf
cv32e40p/tests/programs/custom/mscratch_test/mscratch_test.itb
*****
* Running with Verilator: logfile in simulation_results/mscratch_test/mscratch_test.log
*****
mkdir -p simulation_results/mscratch_test/0/test_program
simulation_results/mscratch_test/0/test_program
simulation_results/mscratch_test/0/test_program/mscratch_test.log
"firmware-.../tests/programs/custom/mscratch_test/mscratch_test.hex" \
| tee simulation_results/mscratch_test/0/test_program/mscratch_test.log

scopesDump:
SCOPE 0x630823db7230: TOP.tb_top_verilator
SCOPE 0x630823db72aa8: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.cs_registers_i
SCOPE 0x630823db7490: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.gen_no_pulp_secure_write_logic
SCOPE 0x630823db75a0: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.ex_stage_i.alu_i
SCOPE 0x630823db7698: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i
SCOPE 0x630823db7770: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.controller_i
SCOPE 0x630823db7860: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.controller_i.blk_decode_level1
SCOPE 0x630823db7968: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.decoder_i
SCOPE 0x630823db7a50: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.immediate_a_mux
SCOPE 0x630823db7b40: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.id_stage_i.jump_target_mux
SCOPE 0x630823db7c38: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i
SCOPE 0x630823db7d10: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.EXC_PC_MUX
SCOPE 0x630823db7e00: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.compressed_decoder_i
SCOPE 0x630823db7f00: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.cv32e40p_core_i.if_stage_i.prefetch_buffer_i.fifo_i
SCOPE 0x630823db8000: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.ram_i
SCOPE 0x630823db8080: TOP.tb_top_verilator.cv32e40p.tb_wrapper_i.ram_i.dp_ram_i
DPI-EXPORT 0x630805073e10: read_byte
DPI-EXPORT 0x630805073e10: write_byte

[tb_top_verilator] finished dumping memory
Starting mscratch test...
Write: 0x00000000 Read: 0x00000000
Write: 0xffffffff Read: 0xffffffff
Write: 0x5A5A5A5A Read: 0x5A5A5A5A
Write: 0x5A5A5A5A Read: 0x5A5A5A5A
MSCRATCH TEST PASSED
TOP.tb_top_verilator # 129430: EXIT SUCCESS
- /home/karthik07/Desktop/core-v-verif/cv32e40p/tb/core/tb_top_verilator.sv:83: Verilog $finish
karthik07@karthik07-VirtualBox: ~/Desktop/core-v-verif/cv32e40p/sim/core $
```