TOGETHER FOR RISC-V TECHNOLOGY
& APPLICATIONS

RISC-V Summit Europe 2024

TRISTAN: Free Access Training on EDA tooling for RISC-V

09:00-11:30, Friday, 28.06.2024, Munich

TRISTAN has received funding from Chips Joint Undertaking (CHIPS-JU) under grant agreement nr. 101095947.
CHIPS JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia,
Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Verification

Mario Rodriguez
mario@openhwgroup.org

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Agenda

- **Introduction**
- **Explaining cve2**
- **Explaining core-v-verif**
- **Explaining cve2 Testbench**
- **Running cve2**

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

What is TRISTAN?

"**T**ogether for **RIS**c-V **T**echnology and **A**pplicatio**N**s"

It's a KDT-JU (Key-Digital Technology Joint Undertaking) program under the Horizon-Europe calls.

It started in December 2022 and it will last 36 months (3 years).

It aims to reinforce the EU's strategic autonomy in the electronic components and systems sector.

TRISTAN will expand, mature and industrialize the **European RISC-V ecosystem** so that it is able to compete with existing commercial/proprietary alternatives.

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Open-Source and High-Quality

TRISTAN believes in high-quality open-source hardware!

- Many deliverables are in fact based on open-source high TRL IPs

OpenHW Group is the organization that many TRISTAN's partners trust to **maintain** and **support high-quality** (RTL code, testbenches, and documentation) open-source hardware IPs for the l**ong term**.

The TRISTAN project will contribute to many new and improved OpenHW Group IPs that will be accessible to **EVERYONE**.

Unified Access Page

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

What is OpenHW Group ?

- OpenHW Group is a not-for-profit, global organization registered in Canada and Europe. The OpenHW ecosystem is driven by members (corporate & academic) and individual contributors where HW and SW designers collaborate in developing open-source cores, related IP, tools and SW such as the CORE-V Family of open-source RISC-V processors
  - International footprint with developers in North America, Europe and Asia
  - Providing an infrastructure for hosting high quality open-source HW developments in line with industry best practices
  - Strong support from industry, academia and individual contributors worldwide

and

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Agenda

- Introduction
- Explaining cve2
- Explaining core-v-verif
- Explaining cve2 Testbench
- Running cve2

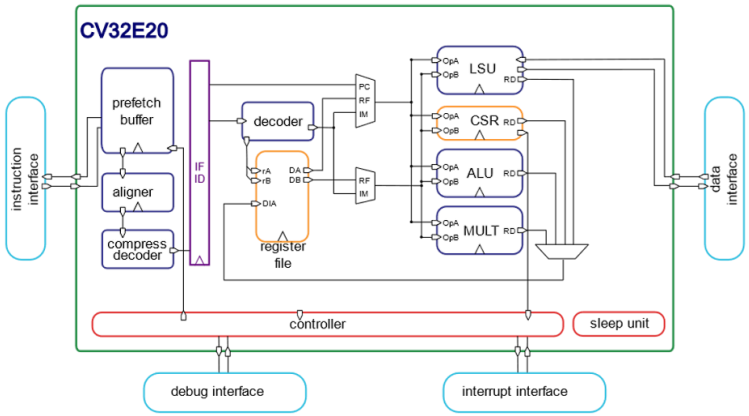
TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

CV32E20 – PL / TRL 4

- 2-stage, in-order, single-issue
- RV32{I,E}[M]CZicount_Zicsr_Zifencei[_Zce]
- M-mode, CLINT, OBI

- Low area core
  - Optimized power and area for control-oriented applications
  - Starting point lowRISC Ibex (which started from ETH zero-riscy)
    - Clean-up parameters
    - Aligning IP interface with CV32E40* cores

- Project goal: industrial grade (TRL 5)



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Agenda

- Introduction
- Explaining cve2
- Explaining core-v-verif
- Explaining cve2 Testbench
- Running cve2

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

What is CORE-V-VERIF?

- It is a GitHub repository: https://github.com/openhwgroup/core-v-verif.

- Central location for:
  - Testbenches for many OpenHW Group's cores.
    - Implemented in System Verilog
    - Universal Verification Methodology (UVM).
  - Verification Documentation.

- CORE-V-VERIF provides:
  - Infrastructure and methodology.
    - Execution framework.
    - Documentation.
  - Components for implementing the testbenches.

- Supports the following cores:
  - CV32E40P (v1 and v2)
  - CV32E40X
  - CV32E40S
  - CVA6
  - CVE20

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

CORE-V-VERIF Hierarchy

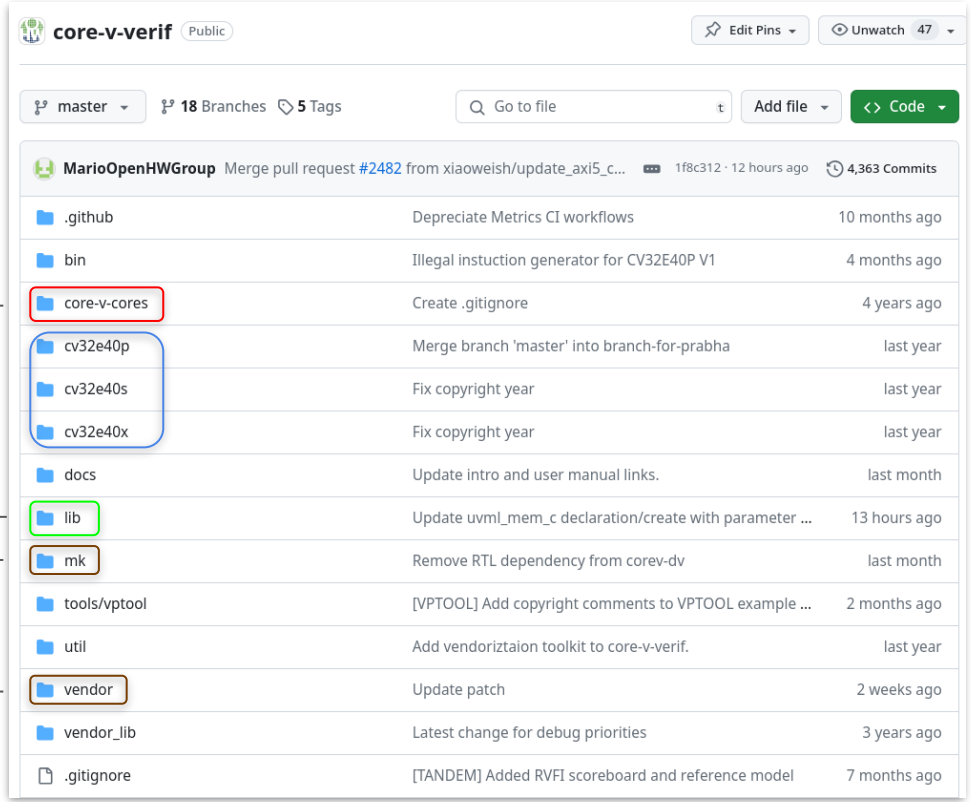
RTL

Core Specific Files

Common Verification Files

Makefile Infrastructure

Vendorized Files



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

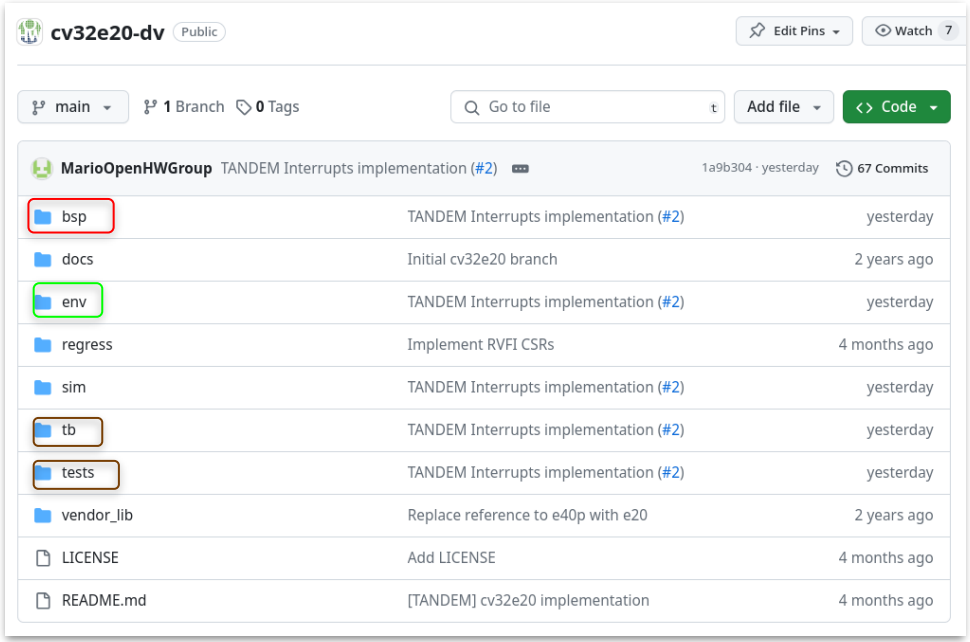
CORE-V-VERIF Core Hierarchy

Board Support Package

UVM environment Files

Testbench Verification Files

UVM tests and Programs



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

CORE-V-VERIF Testbenches

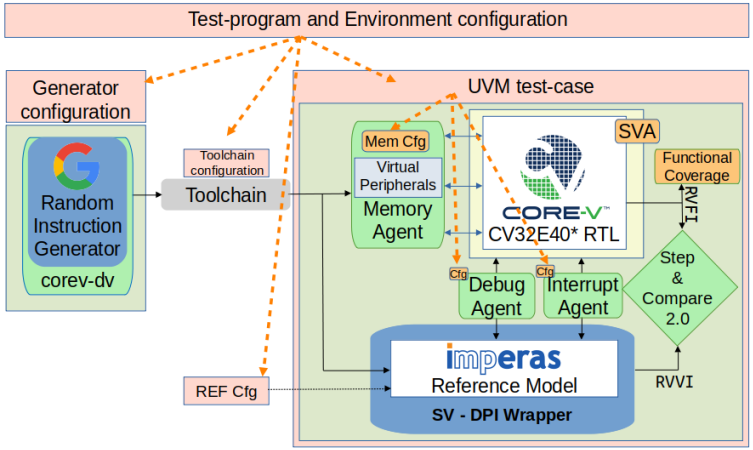
- <core>/tb/uvmt:
  - UVM based
  - Can run with a subset of tools: closed and open source

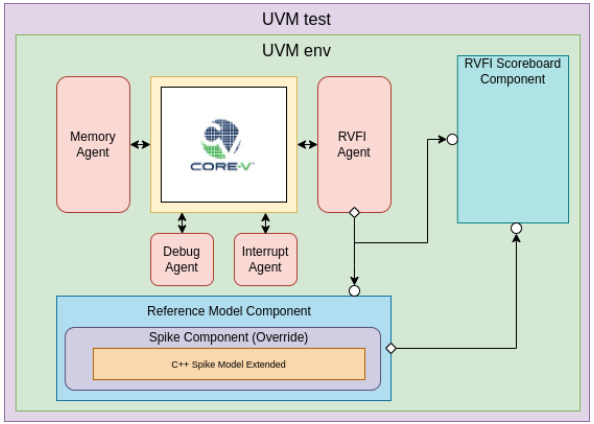
- <core>/tb/core:
  - Pure verilog testbench
  - Can run with a subset of tools: closed and open source
  - Not present in all the cores

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

CORE-V-VERIF UVM Testbench

- The CORE-V-VERIF:
  - Can have on-the-fly comparison between RTL and Reference Model.
  - Built-in support for managing consistency between tools.
  - Dedicated Agents to provide support for Interruptions and Debug.

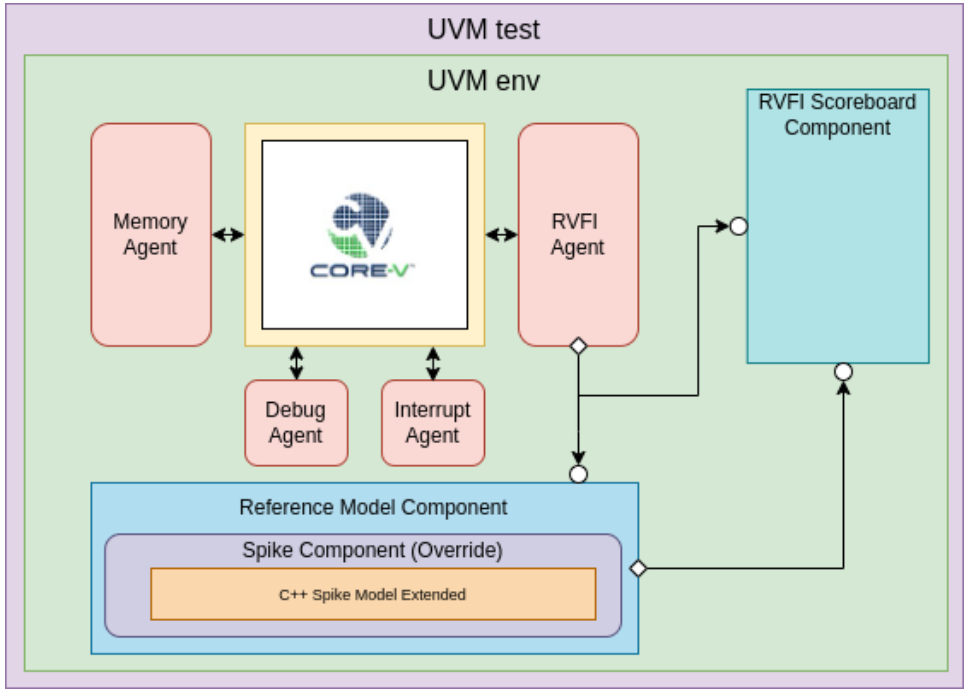




TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

UVM Tandem Architecture

- Typical UVM structure
  - RVFI monitor sends to:
    - Reference Model
    - Scoreboard
  - Reference Model sends to:
    - Scoreboard
- Generic Reference Model
  - Derived Classes: Spike
- Call to common SystemVerilog code



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

UVM Tandem (C++)

- RVFI Implementation C++:
  - Restriction to use uint64_t
    - Verilator issues with passing structs through DPI
  - Used by the simulation and the core
  - Mirror seq item in SV
  - Mirror struct in SV

```
typedef struct  {
    uint64_t              nret_id;
    uint64_t              cycle_cnt;
    uint64_t              order;
    uint64_t              insn;
    uint64_t              trap;
    uint64_t              cause;
    uint64_t              halt;
    uint64_t              intr;
    uint64_t              mode;
    uint64_t              ixl;
    uint64_t              dbg;
    uint64_t              dbg_mode;
    uint64_t              nmip;

    uint64_t              insn_interrupt;
    uint64_t              insn_interrupt_id;
    uint64_t              insn_bus_fault;
    uint64_t              insn_nmi_store_fault;
    uint64_t              insn_nmi_load_fault;

    uint64_t              pc_rdata;
    uint64_t              pc_wdata;

    uint64_t              rs1_addr;
    uint64_t              rs1_rdata;

    uint64_t              rs2_addr;
    uint64_t              rs2_rdata;

    uint64_t              rs3_addr;
    uint64_t              rs3_rdata;

    uint64_t              rd1_addr;
    uint64_t              rd1_wdata;

    uint64_t              rd2_addr;
    uint64_t              rd2_wdata;

    uint64_t              mem_addr;
    uint64_t              mem_rdata;
    uint64_t              mem_rmask;
    uint64_t              mem_wdata;
    uint64_t              mem_wmask;

} st_rvfi;
```

Source: Types.h

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Base Spike and Modifications

- C++ to SV Interface:
  - Ability to setup and step spike
    - *spike_create(...)*
    - *spike_set_params_x(uint64_t, bool, ...)*
    - *spike_step(...)*

```
class Params {
    public:

    unordered_map<string, unordered_map<string, Param>> v;

    any operator[](string str) {
        return this->get(str).a;
    }
```

Source:
Param.h

```
typedef struct {
    string base;
    string name;
    any a;
    string description;
} Param;
```

Source:
Param.h

```
39  extern "C" void spike_set_default_params(const char* profile)
53  {...}
54
55  extern "C" void spike_set_param_uint64_t(const char* base, const char* name, uint64_t value) { params.set(base, name, value); }
56  extern "C" void spike_set_param_str(const char* base, const char* name, const char* value) { params.set(base, name, std::string(value)); }
57
58  extern "C" void spike_create(const char* filename)
105 {...}
```

```
131 extern "C" void spike_step_struct(st_rvfi& reference, st_rvfi& spike)
137 {...}
138
139 extern "C" void spike_step_svOpenArray(svOpenArrayHandle reference, svOpenArrayHandle spike)
146 {...}
```

Source: riscv_dpi.cc

European Commission
KDT JU
Key Digital Technologies Joint Undertaking

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Base Spike and Modifications

- C++ to SV Interface:
  - Ability to parse a binary and get the symbols
    - *read_elf*(...)
    - *read_symbol*(...)
    - *read_section*(...)
  - Part of the code was imported from CVA6 repository

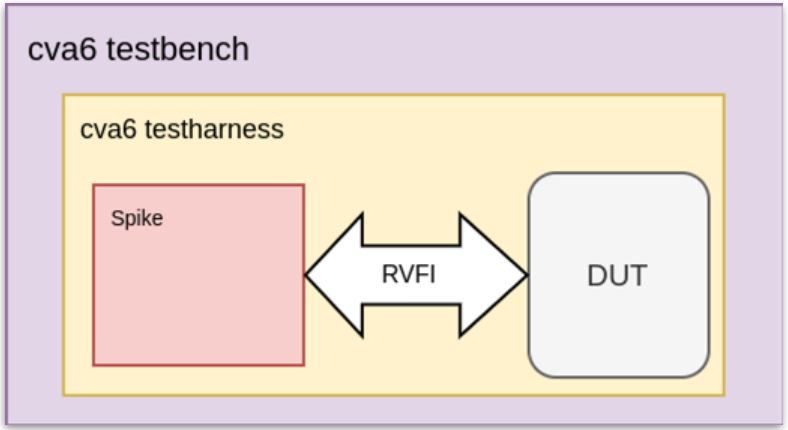
```
65  > extern "C" char get_section (long long* address, long long* len) {
74    ...
75
76  > extern "C" void read_section_void (long long address, void* buffer, uint64_t size = 0) {
86    ...
87
88  > extern "C" void read_section_sv (long long address, const svOpenArrayHandle buffer) {
99    ...
100
101 > extern "C" char read_symbol (const char* symbol_name, long long* address) {
109   ...
110
111 > extern "C" void read_elf(const char* filename) {
121   ...
```

Source: fesvr_dpi.cc

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Tandem Architecture: CVA6

- Spike SystemVerilog module
  - RVFI connected directly from the core
- It is not a class but it could be one
- Call to common SystemVerilog code



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Common SV Code

- There are some pure SV functions that implement common functionality
  - *rvfi_spike_initialize:* Function that initializes Spike Simulation object.
  - *rvfi_spike_step:* Function that steps Spike one instruction
  - *rvfi_compare:* Function that compares two RVFI structs
- In order to do this:
  - Dependency with UVM had to be broken
    - Implementation of sequence item to struct

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Other ISS implementation

- If another ISS is wanted we need the following:
  - Implement on the ISS the DPI interface to:
    - x_initialize, x_configure and x_step
  - Implement *uvmc_rvfi_x* (ovpsim, sail, …) for UVM
    - Override *uvmc_rvfi_reference_model*
  - Implement SV module to use it without UVM
    - At testharness level

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

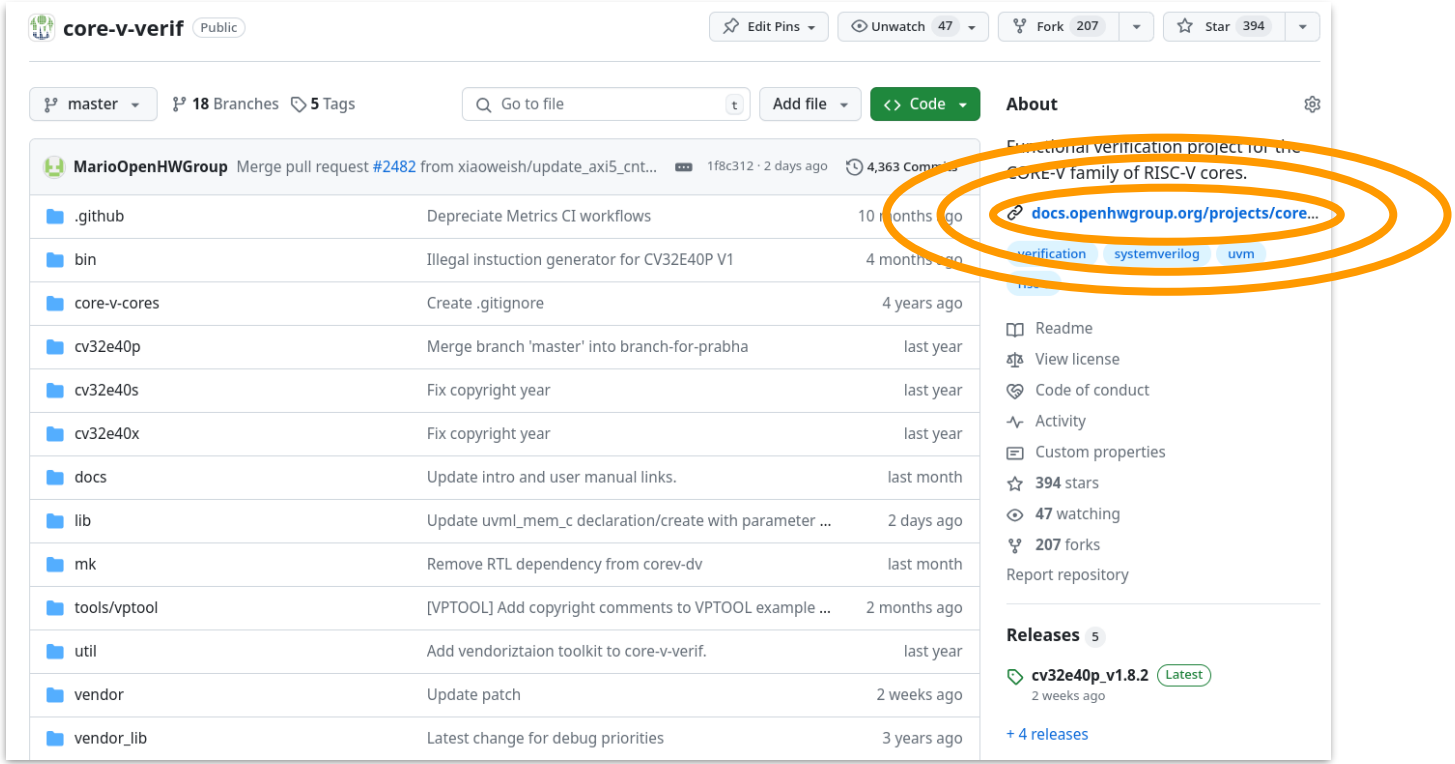
CORE-V-VERIF with another core ?

- Requirements:
  - RTL interfaces implementation.
    - RVFI
  - Testbench implementation using the provided components.

- Already implemented components:
  - Memory Agents (Functionality + Coverage)
    - OBI, AXI
  - CV-X-IF
  - Interrupts, Debug
  - Functional coverage:
    - ISA: ISACOV model

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

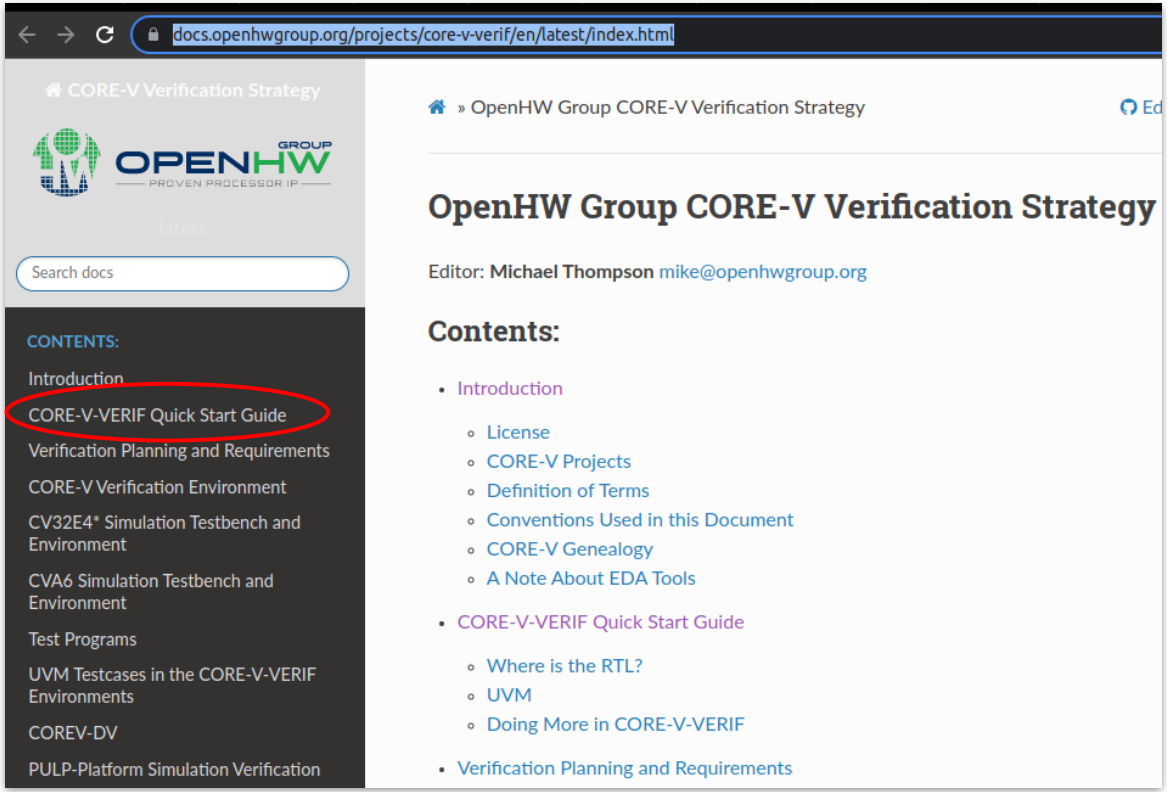
CORE-V-VERIF - More Information ?



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Now What?!?

- CORE-V-VERIF documentation can be found in two places:
  - ReadTheDocs
    - Try the **Quick Start Guide** first.
  - “In-place” README:
    - We attempt to have location-specific information in each directory.



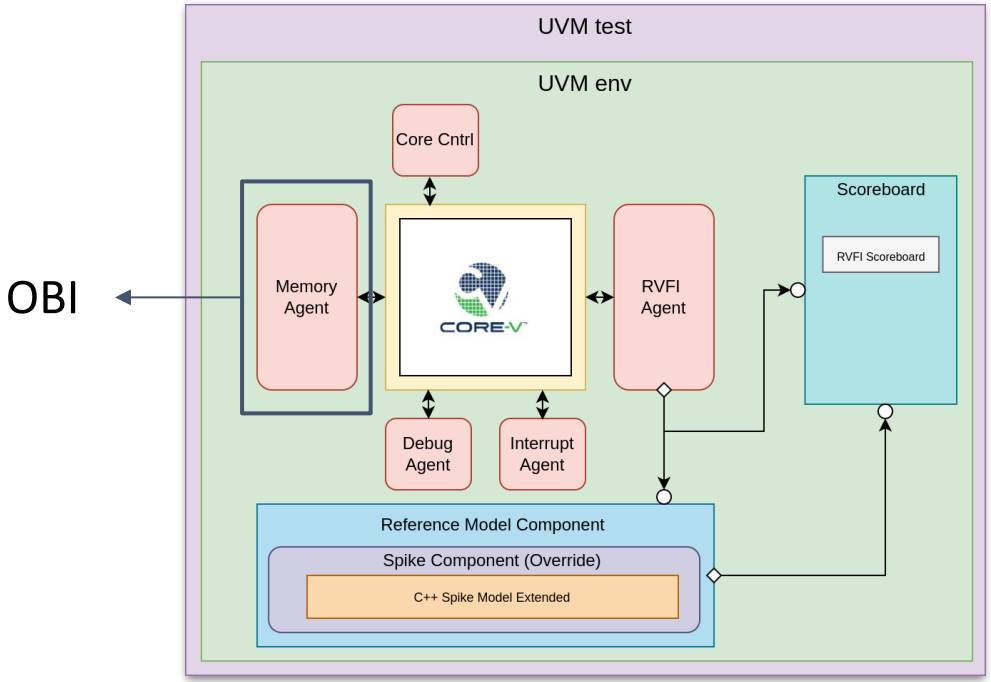
TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union’s Horizon Europe’s research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

Agenda

- Introduction
- Explaining CVE2 Core
- Explaining CORE-V-VERIF
- Explaining CVE2 Testbench
- Running CVE2

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

CVE2 Testbench



TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Testbench

Configuration:

- Extends from core_cntrl_cfg
- Defines memory mapped verification registers

main cv32e20-dv / env / uvme / uvme_cv32e20_cfg.sv

Code Blame 330 lines (263 loc) · 12.3 KB

```
91  constraint cve2_riscv_cons {
92        xlen == uvma_core_cntrl_pkg::MXL_32;
93        ilen == 32;
94
95        mhartid                == 0;
96        marchid                == 'd35;
97        mvendorid              == 'h602;
98
99        ext_i_supported        == 1;
100       ext_a_supported        == 0;
101       ext_m_supported        == 1;
102       ext_c_supported        == 1;
103       ext_p_supported        == 0;
104       ext_v_supported        == 0;
105       ext_f_supported        == 0;
106       ext_d_supported        == 0;
107
```

```
108       ext_zba_supported      == 0;
109       ext_zbb_supported      == 0;
110       ext_zbc_supported      == 0;
111
112       ext_zbe_supported      == 0;
113       ext_zbf_supported      == 0;
114       ext_zbm_supported      == 0;
115       ext_zbp_supported      == 0;
116       ext_zbr_supported      == 0;
117       ext_zbs_supported      == 1;
118       ext_zbt_supported      == 0;
119       ext_zifencei_supported == 1;
120       ext_zicsr_supported    == 1;
121       ext_zcb_supported      == 0;
122
123       ext_cv32a60x_supported == 0;
124       mode_s_supported       == 0;
125       mode_u_supported       == 1;
126
127       pmp_supported          == 0;
128       debug_supported        == 1;
129
130       unaligned_access_supported     == 0;
131       unaligned_access_amo_supported == 0;
```

European Commission
KDT JU Key Digital Technologies Joint Undertaking

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Testbench

Configuration:

- Part of the tandem is configured by the UVM but not all.

```
98          if (core_name == "cve2") begin
99              void'(spike_set_param_uint64_t(base, "mstatus_override_mask", 64'hFFFFFFFF));
100             void'(spike_set_param_uint64_t(base, "mstatus_override_value", 64'h1800));
101             void'(spike_set_param_uint64_t(base, "tdata1_override_mask", 64'hFFFFFFFF));
102             void'(spike_set_param_uint64_t(base, "tdata1_override_value", 64'h28001048));
103             void'(spike_set_param_bool(base, "tdata1_we", 1'h0));
104             void'(spike_set_param_bool(base, "tdata1_we_enable", 1'h1));
105             void'(spike_set_param_bool(base, "non_standard_interrupts", 1'h1));
106             void'(spike_set_param_bool(base, "tinfo_presence", 1'h0));
107             void'(spike_set_param_uint64_t(base, "trigger_count", 64'h0001));
108         end
```

European Commission
KDT JU
Key Digital Technologies Joint Undertaking

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Testbench: Tracer

- RTL Tracer was already in place but the CSRs tracing was not implemented.

main cve2 / rtl / cve2_cs_registers.sv

Code Blame 1554 lines (1358 loc) · 57.7 KB

```
1510
1511    `define RVFI_CONNECT(CSR_ADDR, CSR_NAME, CSR_RDATA, CSR_WDATA, CSR_RMASK, CSR_WMASK) \
1512        bit [63:0] rvfi_``CSR_NAME``_csr_rdata;\
1513        bit [63:0] rvfi_``CSR_NAME``_csr_wdata;\
1514        bit [63:0] rvfi_``CSR_NAME``_csr_rmask;\
1515        bit [63:0] rvfi_``CSR_NAME``_csr_wmask;\
1516        wire [63:0] rvfi_``CSR_NAME``_csr_wmask_q; \
1517        wire [63:0] rvfi_``CSR_NAME``_csr_rmask_q; \
1518        assign rvfi_csr_if.rvfi_named_csr_rdata[CSR_ADDR] = (!rvfi_csr_bypass) ? rvfi_``CSR_NAME``_csr_rdata : ``CSR_RDATA``; \
1519        assign rvfi_csr_if.rvfi_named_csr_wdata[CSR_ADDR] = (!rvfi_csr_bypass) ? rvfi_``CSR_NAME``_csr_wdata : ``CSR_WDATA``; \
1520        assign rvfi_csr_if.rvfi_named_csr_rmask[CSR_ADDR] = (!rvfi_csr_bypass) ? rvfi_``CSR_NAME``_csr_rmask : rvfi_``CSR_NAME``_csr_rmask_q; \
1521        assign rvfi_csr_if.rvfi_named_csr_wmask[CSR_ADDR] = (!rvfi_csr_bypass) ? rvfi_``CSR_NAME``_csr_wmask : rvfi_``CSR_NAME``_csr_wmask_q; \
1522        assign rvfi_``CSR_NAME``_csr_rmask_q =  ((~csr_wr & csr_op_en_i & ~illegal_csr_insn_o & (csr_addr_i == CSR_ADDR)) CSR_RMASK) ? -1 : 0; \
1523        assign rvfi_``CSR_NAME``_csr_wmask_q =   ((csr_wr & csr_op_en_i & ~illegal_csr_insn_o & (csr_addr_i == CSR_ADDR)) CSR_WMASK) ? -1 : 0; \
1524        always @(posedge clknrst_if.clk) begin \
1525            rvfi_``CSR_NAME``_csr_rdata = ``CSR_RDATA``; \
1526            rvfi_``CSR_NAME``_csr_wdata = ``CSR_WDATA``; \
1527            rvfi_``CSR_NAME``_csr_rmask = (rvfi_``CSR_NAME``_csr_rmask_q); \
1528            rvfi_``CSR_NAME``_csr_wmask = (rvfi_``CSR_NAME``_csr_wmask_q); \
1529        end
1530
1531    `RVFI_CONNECT( CSR_MSTATUS,          mstatus                 , mstatus_extended_read  , mstatus_extended_write , , || mstatus_en)
1532    `RVFI_CONNECT( CSR_MIE,              mie                     , mie_extended_read      , mie_extended_write     , , || mie_en    )
1533    `RVFI_CONNECT( CSR_MIP,              mip                     , mip_extended_read      , mip_extended_read      , , )
1534    `RVFI_CONNECT( CSR_MISA,             misa                    , MISA_VALUE             , MISA_VALUE             , ,              )
1535    `RVFI_CONNECT( CSR_MTVEC,            mtvec                   , mtvec_q                , mtvec_d                , , || mtvec_en  )
1536    `RVFI_CONNECT( CSR_MEPC,             mepc                    , mepc_q                 , mepc_d                 , , || mepc_en   )
1537    `RVFI_CONNECT( CSR_MCAUSE,           mcause                  , mcause_extended_read   , mcause_extended_write  , , || mcause_en )
1538    `RVFI_CONNECT( CSR_MTVAL,            mtval                   , mtval_q                , mtval_d                , , || mtval_en  )
1539    `RVFI_CONNECT( CSR_MSTATUSH,         mstatush                , 'h0                    , csr_wdata_int          , , )
1540    `RVFI_CONNECT( CSR_DCSR,             dcsr                    , dcsr_q                 , dcsr_d                 , , || dcsr_en)
1541    `RVFI_CONNECT( CSR_DPC,              dpc                     , depc_q                 , depc_d                 , , || depc_en)
1542    `RVFI_CONNECT( CSR_DSCRATCH0,        dscratch0               , dscratch0_q            , csr_wdata_int          , , || dscratch0_en)
1543    `RVFI_CONNECT( CSR_DSCRATCH1,        dscratch1               , dscratch1_q            , csr_wdata_int          , , || dscratch1_en)
1544    `RVFI_CONNECT( CSR_MSCRATCH,         mscratch                , mscratch_q             , csr_wdata_int          , , || mscratch_en)
1545
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Testbench: Interrupts

- When an interrupt arrives, the RVFI tracer gets the information and notifies the ISS.

master — core-v-verif / vendor / riscv / riscv-isa-sim / riscv / Proc.cc

Code | Blame | 570 lines (466 loc) · 20.3 KB

```
20  st_rvfi Processor::step(size_t n, st_rvfi reference_) {
42
43      if (inject_interrupt && interrupts_injection && !this->taken_trap) {
44          // We need to ensure this is an interrupt to inject mip
45          uint64_t mcause = reference->csr_rdata[INDEX_CSR(CSR_MCAUSE)];
46          if (mcause >> 31) {
47              uint64_t mip = this->mcause_to_mip(mcause);
48              this->get_state()->mip->backdoor_write(mip);
49              this->step_rvfi->intr = 0b101; // Interrupt
50              this->step_rvfi->intr |= ((mcause & 0x3FF) << 3);
51              uint64_t nmi_mcause = (this->params[base + "nmi_mcause"]).a_uint64_t;
52              if (nmi_mcause == (mcause & 0x3FF))
53                  this->nmi_inject = true;
54          }
55      }
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

CVE2 Testbench: Debug

- When an external debug request arrives, the RVFI tracer gets the information and notifies the ISS.

master core-v-verif / vendor / riscv / riscv-isa-sim / riscv / Proc.cc

Code Blame 570 lines (466 loc) · 20.3 KB

```
20  st_rvfi Processor::step(size_t n, st_rvfi reference_) {
57      if (reference->dbg && !this->get_state()->debug_mode && debug_injection && !this->halted()) {
58          uint64_t cause = reference->dbg;
59          if (cause) {
60              enter_debug_mode(cause);
61              rvfi.dbg = cause;
62              rvfi.trap = 0b101;
63              rvfi.trap |= (cause << 9);
64          }
65      }
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

Agenda

- Introduction
- Explaining cve2
- Explaining core-v-verif
- Explaining cve2 Testbench
- Running cve2

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Cloning the repos

- To run cve2 we need:
  - CORE-V-VERIF as a TOP
  - VERIFICATION: Subfolder with verification specific repository
  - RTL

```
drwxr-xr-x 17 4.0K Jun 26 15:56 .
drwxr-xr-x  3 4.0K Jun 26 15:53 ..
drwxr-xr-x  5 4.0K Jun 26 15:53 bin
drwxr-xr-x  2 4.0K Jun 26 15:53 core-v-cores
drwxr-xr-x 11 4.0K Jun 26 15:56 cv32e20
drwxr-xr-x 10 4.0K Jun 26 15:53 cv32e40p
drwxr-xr-x 10 4.0K Jun 26 15:53 cv32e40s
drwxr-xr-x 10 4.0K Jun 26 15:53 cv32e40x
drwxr-xr-x  4 4.0K Jun 26 15:53 docs
drwxr-xr-x  8 4.0K Jun 26 15:53 .git
drwxr-xr-x  4 4.0K Jun 26 15:53 .github
drwxr-xr-x 10 4.0K Jun 26 15:53 lib
drwxr-xr-x  4 4.0K Jun 26 15:53 mk
drwxr-xr-x  3 4.0K Jun 26 15:53 tools
drwxr-xr-x  2 4.0K Jun 26 15:53 util
drwxr-xr-x  4 4.0K Jun 26 15:53 vendor
drwxr-xr-x  4 4.0K Jun 26 15:53 vendor_lib
-rw-r--r--  1  720 Jun 26 15:53 ACKNOWLEDGEMENTS.md
-rw-r--r--  1 3.3K Jun 26 15:53 CODE_OF_CONDUCT.md
-rw-r--r--  1 3.8K Jun 26 15:53 CONTRIBUTING.md
-rw-r--r--  1 6.3K Jun 26 15:53 GitCheats.md
-rw-r--r--  1  910 Jun 26 15:53 .gitignore
-rw-r--r--  1 2.8K Jun 26 15:53 LICENSE.md
-rw-r--r--  1 4.3K Jun 26 15:53 MergeTest.md
-rw-r--r--  1  27K Jun 26 15:53 .metrics.json
-rw-r--r--  1 2.1K Jun 26 15:53 NEWS_ARCHIVE.md
-rw-r--r--  1 4.1K Jun 26 15:53 README.md
-rw-r--r--  1  487 Jun 26 15:53 .readthedocs.yaml
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

Cloning the repos

```
[mario@t14 ~/git/risc-v-summit-europe2024]$ git clone git@github.com:openhwgroup/core-v-verif.git --recurse-submodules
Cloning into 'core-v-verif'...
remote: Enumerating objects: 58269, done.
remote: Counting objects: 100% (6587/6587), done.
remote: Compressing objects: 100% (2541/2541), done.
remote: Total 58269 (delta 4248), reused 5760 (delta 3911), pack-reused 51682
Receiving objects: 100% (58269/58269), 114.96 MiB | 11.60 MiB/s, done.
Resolving deltas: 100% (39315/39315), done.
```

```
[mario@t14 ~/git/risc-v-summit-europe2024]$ git clone git@github.com:openhwgroup/cv32e20-dv.git core-v-verif/cv32e20/
Cloning into 'core-v-verif/cv32e20'...
remote: Enumerating objects: 917, done.
remote: Counting objects: 100% (917/917), done.
remote: Compressing objects: 100% (440/440), done.
remote: Total 917 (delta 464), reused 883 (delta 443), pack-reused 0
Receiving objects: 100% (917/917), 873.60 KiB | 1.88 MiB/s, done.
Resolving deltas: 100% (464/464), done.
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

Simulation

- **Simulation process is heavily transparent (everything managed by Makefiles)**
  - **Step 1:** clone the RTL.
  - **Step 2.1:** compile C++ libraries needed for simulation
  - **Step 2.2:** compile DUT + testbench + test-program
    - Generated using corev-dv.
    - Manually written test-programs skip the generation part.
  - **Step Three**: run the simulation.

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: External Repos

main cv32e20-dv / sim / ExternalRepos.mk

MarioOpenHWGroup TANDEM Interrupts implementation (#2)

Code Blame 34 lines (28 loc) · 1.3 KB

```
1  ################################################################################
2  # Variables to generate the command to clone external repositories.
3  # For each repo there are a set of variables:
4  #      *_REPO:   URL to the repository (note, not all are in GitHub).
5  #      *_BRANCH: Name of the branch you wish to clone;
6  #                Set to 'master' to pull the master branch.
7  #      *_HASH:   Value of the specific hash you wish to clone;
8  #                Set to 'head' to pull the head of the branch you want.
9  #
10
11 export SHELL = /bin/bash
12
13 CV_CORE_REPO   ?= https://github.com/openhwgroup/cve2
14 CV_CORE_BRANCH ?= main
15 CV_CORE_HASH   ?= 370793f52488d1022d0554d194ad24f125156acc
16 CV_CORE_TAG    ?= none
17
18 RISCVDV_REPO    ?= https://github.com/google/riscv-dv
19 RISCVDV_BRANCH  ?= master
20 RISCVDV_HASH    ?= 0b625258549e733082c12e5dc749f05aefb07d5a
21
22 EMBENCH_REPO    ?= https://github.com/embench/embench-iot.git
23 EMBENCH_BRANCH  ?= master
24 EMBENCH_HASH    ?= 6934ddd1ff445245ee032d4258fdeb9828b72af4
25
26 COMPLIANCE_REPO   ?= https://github.com/riscv/riscv-compliance
27 COMPLIANCE_BRANCH ?= master
28 # 2020-08-19
29 COMPLIANCE_HASH   ?= c21a2e86afa3f7d4292a2dd26b759f3f29cde497
30
31 # SVLIB
32 SVLIB_REPO       ?= https://bitbucket.org/verilab/svlib/src/master/svlib
33 SVLIB_BRANCH     ?= master
34 SVLIB_HASH       ?= c25509a7e54a880fe8f58f3daa2f891d6ecf6428
```

European Commission
KDT JU Key Digital Technologies Joint Undertaking

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation

- **Step One:** Two options to clone RTL:
  - Manually
  - Let scripts clone a specific hash

```
[mario@t14 ~/git/risc-v-summit-europe2024/core-v-verif]$ git clone git@github.com:openhwgroup/cve2.git core-v-cores/cv32e20/
Cloning into 'core-v-cores/cv32e20'...
remote: Enumerating objects: 20215, done.
remote: Counting objects: 100% (4868/4868), done.
remote: Compressing objects: 100% (647/647), done.
remote: Total 20215 (delta 4494), reused 4228 (delta 4220), pack-reused 15347
Receiving objects: 100% (20215/20215), 10.77 MiB | 6.24 MiB/s, done.
Resolving deltas: 100% (14436/14436), done.
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947. The KDT JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey

Simulation: Requirements

- **RISC-V Toolchain:**
  - **RISCV environment variable**
- **MARCH**
  - **CV_SW_MARCH environment variable**

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: Makefiles

- **Step Two: Compile**
- **Step Three: Execute**
  - Logs path: **cv32e20/sim/uvmt/$(simulator)_results/**

Command : **CV_SW_MARCH=rv32gc_zicsr make -C cv32e20/sim/uvmt/ test TEST=hello-world SIMULATOR=vsim **
USE_ISS=yes SPIKE_TANDEM=1 UVM_VERBOSITY=UVM_LOW

```
mario@av /home/mario/git/risc-v-summit-europe2024/core-v-verif $ CV_SW_MARCH=rv32imc_zicsr NUM_JOBS=12 make -C cv32e20/sim/uvmt/ test TEST=hello-world SIMULATOR=vsim USE_ISS=yes UVM_VERBOSITY=UVM_LOW WAVES=yes
make: Entering directory '/home/mario/git/risc-v-summit-europe2024/core-v-verif/cv32e20/sim/uvmt'
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:291: CV_SW_CC not defined in either the shell environment, test.yaml or cfg.yaml
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:303: CV_SW_CFLAGS not defined in either the shell environment, test.yaml or cfg.yaml
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:318: RISCV set to /home/mario/local/corev-openhw-gcc-centos7-20230310/
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:319: RISCV_PREFIX set to riscv32-corev-elf-
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:320: RISCV_EXE_PREFIX set to /home/mario/local/corev-openhw-gcc-centos7-20230310//bin/riscv32-corev-elf-
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:321: RISCV_MARCH set to rv32imc_zicsr
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:322: RISCV_CC set to gcc
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:323: RISCV_CFLAGS set to
mkdir -p /home/mario/git/risc-v-summit-europe2024/core-v-verif/cv32e20/sim/uvmt/vsim_results/default/hello-world/0/test_program
make bsp
make[1]: Entering directory '/home/mario/git/risc-v-summit-europe2024/core-v-verif/cv32e20/sim/uvmt'
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:291: CV_SW_CC not defined in either the shell environment, test.yaml or cfg.yaml
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:303: CV_SW_CFLAGS not defined in either the shell environment, test.yaml or cfg.yaml
/home/mario/git/risc-v-summit-europe2024/core-v-verif/mk/Common.mk:318: RISCV set to /home/mario/local/corev-openhw-gcc-centos7-20230310/
```

DISCLAIMER: If you run two simulations of the same tests the results will be overwritten

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: Results

- **Results**
  - Write ***tohost*** method:
    - EXIT_CODE<< 1 | 1

```
# UVM_INFO @ 16.800 ns : uvmt_cv32e20_base_test.sv(285) uvm_test_top [BASE TEST] set load_instr_mem
# uvmt_cv32e20_tb.dut_wrap.cv32e20_top_i.u_cve2_tracer.printbuffer_dumpline: Writing execution trace to trace_core_00000000.log
#
# HELLO WORLD!!!
# This is the OpenHW Group CV32E20 CORE-V processor core.
# CV32E20 is a RISC-V ISA compliant core with the following attributes:
#       mvendorid = 0x602
#       marchid   = 0x23
#       mimpid    = 0x0
#       misa      = 0x40101104
#       XLEN is 32-bits
#       Supported Instructions Extensions: MIC
#       This machine supports USER mode.
#
# UVM_INFO @ 166471.800 ns : uvme_cv32e20_vp_status_flags_seq.sv(95) uvm_test_top.env.obi_memory_data_agent.sequencer@@data_slv_seq.vp_status_flags [VP_VSEQ] virtual peripheral: TEST PASSED WITH CODE 00000000
# UVM_INFO @ 166511.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: hello-world.yaml
# UVM_INFO @ 166511.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: hello-world.yaml
# UVM_INFO @ 167471.800 ns : uvmt_cv32e20_firmware_test.sv(127) uvm_test_top [TEST] Finished RUN: exit status is 0
# UVM_INFO @ 167471.800 ns : uvm_objection.svh(1270) reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
# UVM_INFO @ 167471.800 ns : uvmt_cv32e20_base_test.sv(336) uvm_test_top [END_OF_TEST] DUT WRAPPER virtual peripheral signaled exit_value=0.
# UVM_INFO @ 167471.800 ns : uvm_report_server.svh(847) reporter [UVM/REPORT/SERVER]
# --- UVM Report Summary ---
#
# Quit count :     0 of     5
# ** Report counts by severity
# UVM_INFO :   70
# UVM_WARNING :    0
# UVM_ERROR :    0
# UVM_FATAL :    0
# ** Report counts by id
# [BASE TEST]     4
# [CLKNRST]     1
# [CV32E20CORECTRLAGT]     1
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: Results

- **Results**
  - Write ***tohost*** method:
    - EXIT_CODE<< 1 | 1

```
# --- UVM Report Summary ---
#
# Quit count :     0 of     5
# ** Report counts by severity
# UVM_INFO :   70
# UVM_WARNING :    0
# UVM_ERROR :    0
# UVM_FATAL :    0
# ** Report counts by id
# [BASE TEST]     4
# [CLKNRST]     1
# [CV32E20CORECTRLAGT]     1
# [CV32E20VPSEQ]     2
# [DEBUGCOVG]     1
# [END_OF_TEST]     1
# [FETCHTOGGLE]     1
# [INTERRUPTCOVG]     1
# [MEMREADMEMH]     1
# [OBIVPSEQ]    34
# [OBI_MEMORY_MON]     4
# [RNTST]     1
# [RST_VSEQ]     3
# [TEST]     3
# [TEST_CFG]     1
# [TEST_DONE]     1
# [UVM/RELNOTES]     1
# [VP_VSEQ]     1
# [cv32e20_env]     2
# [firmware_test]     1
# [rv32isa_covg]     1
# [rvfi_scoreboard_utils]     2
# [spike_tandem]     2
#
# ** Note: $finish    : /onespin/questasim_2023.1.2/questasim/verilog_src/uvm-1.2/src/base/uvm_root.svh(517)
#    Time: 167471800 ps  Iteration: 69  Instance: /uvmt_cv32e20_tb
#
# uvmt_cv32e20_tb.end_of_test: *** Test Summary ***
#
#     PPPPPPP    AAAAAA    SSSSSS    SSSSSS   EEEEEEEE  DDDDDDD
#     PP    PP  AA    AA  SS    SS  SS    SS  EE        DD    DD
#     PP    PP  AA    AA  SS        SS        EE        DD    DD
#     PPPPPPP   AAAAAAAA  SSSSSS    SSSSSS   EEEEE     DD    DD
#     PP        AA    AA        SS        SS  EE        DD    DD
#     PP        AA    AA  SS    SS  SS    SS  EE        DD    DD
#     PP        AA    AA   SSSSSS    SSSSSS   EEEEEEEE  DDDDDDD
#     ------------------------------------------------------------
#                         SIMULATION PASSED
#     ------------------------------------------------------------
# End time: 10:54:51 on Jun 27,2024, Elapsed time: 0:00:21
# Errors: 0, Warnings: 25
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: CI CHECK

Command: **CV_SW_MARCH=rv32gc_zicsr NUM_JOBS=12 ./bin/ci_check --core cv32e20 -s vsim --iss 1**

File: **cv32e20/regress/cv32e20_ci_check.yaml**

```yaml
# YAML file to specify the ci_check regression testlist.
name: cv32e20_ci_check
description: Commit sanity for the cv32e20

builds:
  corev-dv:
    cmd: make comp_corev-dv
    dir: cv32e20/sim/uvmt

  uvmt_cv32e20:
    cmd: make comp
    dir: cv32e20/sim/uvmt

tests:
  hello-world:
    build: uvmt_cv32e20
    description: UVM Hello World Test
    dir: cv32e20/sim/uvmt
    cmd: make hello-world
    cmd: make test COREV=YES TEST=hello-world

  interrupt_test:
    build: uvmt_cv32e20
    description: Interrupt directed test
    dir: cv32e20/sim/uvmt
    cmd: make test COREV=YES TEST=interrupt_test

  corev_rand_interrupt:
    build: uvmt_cv32e20
    description: Interrupt random test
    dir: cv32e20/sim/uvmt
    cmd: make gen_corev-dv test COREV=YES TEST=corev_rand_interrupt
    num: 2

  illegal:
    build: uvmt_cv32e20
    dir: cv32e20/sim/uvmt
    cmd: make test COREV=YES TEST=illegal

  debug_test:
    build: uvmt_cv32e20
    dir: cv32e20/sim/uvmt
    cmd: make test COREV=YES TEST=debug_test

  csr_instructions:
    build: uvmt_cv32e20
    description: CSR Instruction Test
    dir: cv32e20/sim/uvmt
    cmd: make test COREV=YES TEST=csr_instructions

  riscv_arithmetic_basic_test_0:
    build: uvmt_cv32e20
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: CI CHECK

Command: **CV_SW_MARCH=rv32gc_zicsr NUM_JOBS=12 ./bin/ci_check --core cv32e20 -s vsim --iss 1**

```
CI Check results:
                vsim-interrupt_test.log : ABORTED
    vsim-corev_rand_jump_stress_test.log :  PASSED
              vsim-csr_instructions.log :  PASSED
                    vsim-illegal.log :  PASSED
                  vsim-debug_test.log :  PASSED
          vsim-corev_rand_instr_test.log :  PASSED
                  vsim-hello-world.log :  PASSED
vsim-corev_rand_arithmetic_base_test.log :  PASSED
  vsim-riscv_arithmetic_basic_test_0.log :  PASSED
          vsim-corev_rand_interrupt.log :  PASSED
          vsim-corev_rand_interrupt.log :  PASSED
```

European Commission
KDT JU
Key Digital Technologies Joint Undertaking

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: Example mismatch

```
257502.000 ns | RVFI |        0 |        0 | 80003fc8 | 00000141 | M | x0  | 00000000 | x0  | 00000000 | x2  | 00000000803ffdc0 | c.addi  sp, 16
257522.000 ns | RVFI |        0 |        0 | 80003fca | 00008082 | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | ret
257532.000 ns | RVFI |        0 |        0 | 80003c6c | 1aa05a63 | M | x0  | 00000000 | x10 | 00000000 | x0  | 0000000000000000 | bge     zero, a0, pc + 436
257552.000 ns | RVFI |        0 |        0 | 80003c70 | 008b2783 | M | x22 | 00000000 | x8  | 00000000 | x15 | 0000000000000001 | lw      a5, 8(s6)
257562.000 ns | RVFI |        0 |        0 | 80003c74 | 000099aa | M | x1  | 00000000 | x0  | 00000000 | x19 | 00000000800101b9 | c.add   s3, a0
257572.000 ns | RVFI |        0 |        0 | 80003c76 | 40a90933 | M | x18 | 00000000 | x10 | 00000000 | x18 | 0000000000000000 | sub     s2, s2, a0
257582.000 ns | RVFI |        0 |        0 | 80003c7a | 00008f89 | M | x1  | 00000000 | x0  | 00000000 | x15 | 0000000000000000 | c.sub   a5, a0
257602.000 ns | RVFI |        0 |        0 | 80003c7c | 00fb2423 | M | x22 | 00000000 | x15 | 00000000 | x0  | 0000000000000000 | sw      a5, 8(s6)
257622.000 ns | RVFI |        0 |        0 | 80003c80 | 16078d63 | M | x15 | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | beq     a5, zero, pc + 378
257642.000 ns | RVFI |        0 |        0 | 80003dfa | 00004501 | M | x0  | 00000000 | x0  | 00000000 | x10 | 0000000000000000 | c.li    a0, 0
257662.000 ns | RVFI |        0 |        0 | 80003dfc | 000050b2 | M | x0  | 00000000 | x0  | 00000000 | x1  | 00000000800034b2 | c.lwsp  ra, 44(sp)
257682.000 ns | RVFI |        0 |        0 | 80003dfe | 00005422 | M | x0  | 00000000 | x0  | 00000000 | x8  | 0000000080010f70 | c.lwsp  s0, 40(sp)
257702.000 ns | RVFI |        0 |        0 | 80003e00 | 00005492 | M | x0  | 00000000 | x0  | 00000000 | x9  | 0000000080010558 | c.lwsp  s1, 36(sp)
257722.000 ns | RVFI |        0 |        0 | 80003e02 | 00005902 | M | x0  | 00000000 | x0  | 00000000 | x18 | 0000000000000000 | c.lwsp  s2, 32(sp)
257742.000 ns | RVFI |        0 |        0 | 80003e04 | 000049f2 | M | x0  | 00000000 | x0  | 00000000 | x19 | 00000000ffff0888 | c.lwsp  s3, 28(sp)
257762.000 ns | RVFI |        0 |        0 | 80003e06 | 00004a62 | M | x0  | 00000000 | x0  | 00000000 | x20 | 0000000000000001 | c.lwsp  s4, 24(sp)
257782.000 ns | RVFI |        0 |        0 | 80003e08 | 00004ad2 | M | x0  | 00000000 | x0  | 00000000 | x21 | 0000000080010000 | c.lwsp  s5, 20(sp)
257802.000 ns | RVFI |        0 |        0 | 80003e0a | 00004b42 | M | x0  | 00000000 | x0  | 00000000 | x22 | 0000000000000002 | c.lwsp  s6, 16(sp)
257822.000 ns | RVFI |        0 |        0 | 80003e0c | 00004bb2 | M | x0  | 00000000 | x0  | 00000000 | x23 | 0000000000000000 | c.lwsp  s7, 12(sp)
257842.000 ns | RVFI |        0 |        0 | 80003e0e | 00004c22 | M | x0  | 00000000 | x0  | 00000000 | x24 | 0000000000000000 | c.lwsp  s8, 8(sp)
257862.000 ns | RVFI |        0 |        0 | 80003e10 | 00004c92 | M | x0  | 00000000 | x0  | 00000000 | x25 | 0000000000000000 | c.lwsp  s9, 4(sp)
257872.000 ns | RVFI |        0 |        0 | 80003e12 | 00006145 | M | x0  | 00000000 | x0  | 00000000 | x2  | 00000000803ffdf0 | c.addi16sp sp, 48
257892.000 ns | RVFI |        0 |        0 | 80003e14 | 00008082 | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | ret
257912.000 ns | RVFI |        0 |        0 | 800034b2 | 000050f2 | M | x0  | 00000000 | x0  | 00000000 | x1  | 00000008000ff00a | c.lwsp  ra, 60(sp)
257932.000 ns | RVFI |        0 |        0 | 800034b4 | 00005462 | M | x0  | 00000000 | x0  | 00000000 | x8  | 0000000080011000 | c.lwsp  s0, 56(sp)
257942.000 ns | RVFI |        0 |        0 | 800034b6 | 00153513 | M | x10 | 00000000 | x1  | 00000000 | x10 | 0000000000000001 | seqz    a0, a0
257952.000 ns | RVFI |        0 |        0 | 800034ba | 40a00533 | M | x0  | 00000000 | x10 | 00000000 | x10 | 00000000ffffffff | sub     a0, zero, a0
257962.000 ns | RVFI |        0 |        0 | 800034be | 0000892d | M | x1  | 00000000 | x0  | 00000000 | x10 | 000000000000000b | c.andi  a0, 11
257982.000 ns | RVFI |        0 |        0 | 800034c0 | 000054d2 | M | x0  | 00000000 | x0  | 00000000 | x9  | 0000000080011000 | c.lwsp  s1, 52(sp)
257992.000 ns | RVFI |        0 |        0 | 800034c2 | 0000157d | M | x0  | 00000000 | x0  | 00000000 | x10 | 000000000000000a | c.addi  a0, -1
258002.000 ns | RVFI |        0 |        0 | 800034c4 | 00006121 | M | x0  | 00000000 | x0  | 00000000 | x2  | 00000000803ffe30 | c.addi16sp sp, 64
258022.000 ns | RVFI |        0 |        0 | 800034c6 | 00008082 | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | ret
258042.000 ns | RVFI |        0 |        0 | 8000ff0a | 0000a801 | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | c.j     pc + 16
258062.000 ns | RVFI |        0 |        0 | 8000ff1a | 341022f3 | M | x0  | 00000000 | x1  | 00000000 | x5  | 0000000080000762 | csrrs   t0, mepc, zero
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

Simulation: Example mismatch

```
uvmt_cv32e20_tb.dut_wrap.cv32e20_top_i.u_cve2_top.u_cve2_core.id_stage_i.controller_i @ 1003207.000 ns: Illegal instruction (hart 0) at PC 0x80000884: 0x30200073
UVM_INFO @ 1003211.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: interrupt_test.yaml
UVM_INFO @ 1003211.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: interrupt_test.yaml
UVM_INFO @ 1003211.800 ns : uvmc_rvfi_scoreboard_utils.sv(198) reporter [spike_tandem]
TRAP Mismatch Mismatch [REF]: 0x1              [CORE]: 0x0
UVM_INFO @ 1003211.800 ns : uvmc_rvfi_scoreboard_utils.sv(199) reporter [spike_tandem] 1003212.000 ns | RVFI |       0 |      1 | 80000880 | 30200073 | M | x0  | 00000000 | x2  | 00000000 | x0  | 0000000000000000 | mret
UVM_ERROR @ 1003211.800 ns : uvmc_rvfi_scoreboard_utils.sv(200) reporter [spike_tandem] 1003212.000 ns | RVFI |       0 |      0 | 80000880 | 14010113 | U | x2  | 803fffb0 | x0  | 00000000 | x2  | 00000000804000f0 | addi    sp, sp, 320 <- CORE

UVM_INFO @ 1003221.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: interrupt_test.yaml
UVM_INFO @ 1003221.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: interrupt_test.yaml
UVM_INFO @ 1003221.800 ns : uvmc_rvfi_scoreboard_utils.sv(198) reporter [spike_tandem]
TRAP Mismatch Mismatch [REF]: 0x0              [CORE]: 0x1
CSR 341  Mismatch   [REF]: 0x80000880        [CORE]: 0x80000884
CSR 342  Mismatch   [REF]: 0x1              [CORE]: 0x2
UVM_INFO @ 1003221.800 ns : uvmc_rvfi_scoreboard_utils.sv(199) reporter [spike_tandem] 1003222.000 ns | RVFI |       3 |      0 | 80000100 | 5b10f06f | M | x1  | 00000000 | x17 | 00000000 | x0  | 0000000080000104 | j       pc + 0xfdb0
UVM_ERROR @ 1003221.800 ns : uvmc_rvfi_scoreboard_utils.sv(200) reporter [spike_tandem] 1003222.000 ns | RVFI |       0 |      1 | 80000884 | 30200073 | U | x0  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | mret <- CORE

UVM_INFO @ 1003261.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: interrupt_test.yaml
UVM_INFO @ 1003261.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: interrupt_test.yaml
UVM_INFO @ 1003261.800 ns : uvmc_rvfi_scoreboard_utils.sv(198) reporter [spike_tandem]
INSN Mismatch    [REF]: 0x7139             [CORE]: 0x5b10f06f
RD ADDR Mismatch [REF]: 0x2                [CORE]: 0x0
RD VAL Mismatch [REF]: 0x803fff70         [CORE]: 0x0
PC Mismatch      [REF]: 0x8000feb0         [CORE]: 0x80000100
UVM_INFO @ 1003261.800 ns : uvmc_rvfi_scoreboard_utils.sv(199) reporter [spike_tandem] 1003262.000 ns | RVFI |       0 |      0 | 8000feb0 | 00007139 | M | x0  | 00000000 | x0  | 00000000 | x2  | 00000000803fff70 | c.addi16sp sp, -64
UVM_ERROR @ 1003261.800 ns : uvmc_rvfi_scoreboard_utils.sv(200) reporter [spike_tandem] 1003262.000 ns | RVFI |      21 |      0 | 80000100 | 5b10f06f | M | x0  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | j       pc + 0xfdb0 <- CORE

UVM_INFO @ 1003271.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: interrupt_test.yaml
UVM_INFO @ 1003271.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: interrupt_test.yaml
UVM_INFO @ 1003271.800 ns : uvmc_rvfi_scoreboard_utils.sv(198) reporter [spike_tandem]
INSN Mismatch    [REF]: 0xc006             [CORE]: 0x7139
RD ADDR Mismatch [REF]: 0x0                [CORE]: 0x2
RD VAL Mismatch [REF]: 0x0                [CORE]: 0x804000b0
PC Mismatch      [REF]: 0x8000feb2         [CORE]: 0x8000feb0
UVM_INFO @ 1003271.800 ns : uvmc_rvfi_scoreboard_utils.sv(199) reporter [spike_tandem] 1003272.000 ns | RVFI |       0 |      0 | 8000feb2 | 0000c006 | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | c.swsp  ra, 0(sp)
UVM_ERROR @ 1003271.800 ns : uvmc_rvfi_scoreboard_utils.sv(200) reporter [spike_tandem] 1003272.000 ns | RVFI |       0 |      0 | 8000feb0 | 00007139 | M | x2  | 804000f0 | x0  | 00000000 | x2  | 00000000804000b0 | c.addi16sp sp, -64 <- CORE

UVM_INFO @ 1003291.800 ns : uvmc_rvfi_scoreboard_utils.sv(73) reporter [rvfi_scoreboard_utils] Opening file for YAML report: interrupt_test.yaml
UVM_INFO @ 1003291.800 ns : uvmc_rvfi_scoreboard_utils.sv(95) reporter [rvfi_scoreboard_utils] Generated YAML report: interrupt_test.yaml
UVM_INFO @ 1003291.800 ns : uvmc_rvfi_scoreboard_utils.sv(198) reporter [spike_tandem]
INSN Mismatch    [REF]: 0xc22a             [CORE]: 0xc006
PC Mismatch      [REF]: 0x8000feb4         [CORE]: 0x8000feb2
UVM_INFO @ 1003291.800 ns : uvmc_rvfi_scoreboard_utils.sv(199) reporter [spike_tandem] 1003292.000 ns | RVFI |       0 |      0 | 8000feb4 | 0000c22a | M | x1  | 00000000 | x0  | 00000000 | x0  | 0000000000000000 | c.swsp  a0, 4(sp)
UVM_ERROR @ 1003291.800 ns : uvmc_rvfi_scoreboard_utils.sv(200) reporter [spike_tandem] 1003292.000 ns | RVFI |       0 |      0 | 8000feb2 | 0000c006 | M | x2  | 804000b0 | x1  | 80002708 | x0  | 0000000000000000 | c.swsp  ra, 0(sp) <- CORE

UVM_INFO @ 1003291.800 ns : uvm_report_server.svh(847) reporter [UVM/REPORT/SERVER]
--- UVM Report Summary ---
```

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

GROUP
OPENHW™
PROVEN PROCESSOR IP

Thank You!

TRISTAN has received funding from the Key Digital Technologies Joint Undertaking (KDT JU) under grant agreement nr. 101095947.

TRISTAN
RISC-V

TOGETHER FOR
RISC-V
TECHNOLOGY
& APPLICATIONS

TRISTAN webpage

TRISTAN LinkedIn

TRISTAN Unified Access Page

TRISTAN has received funding from Chips Joint Undertaking (CHIPS-JU) under grant agreement nr. 101095947.

CHIPS JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Es
Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Tu