2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Reusable Verification Environment for a RISC-V Vector Accelerator

R. Ignacio Genovese, Josue Quiroga, Ivan Diaz, Henrique Yano, Asif Ali, Nehir Sonmez, Oscar Palomar, Victor Jimenez, Mario Rodriguez, Marc Dominguez.

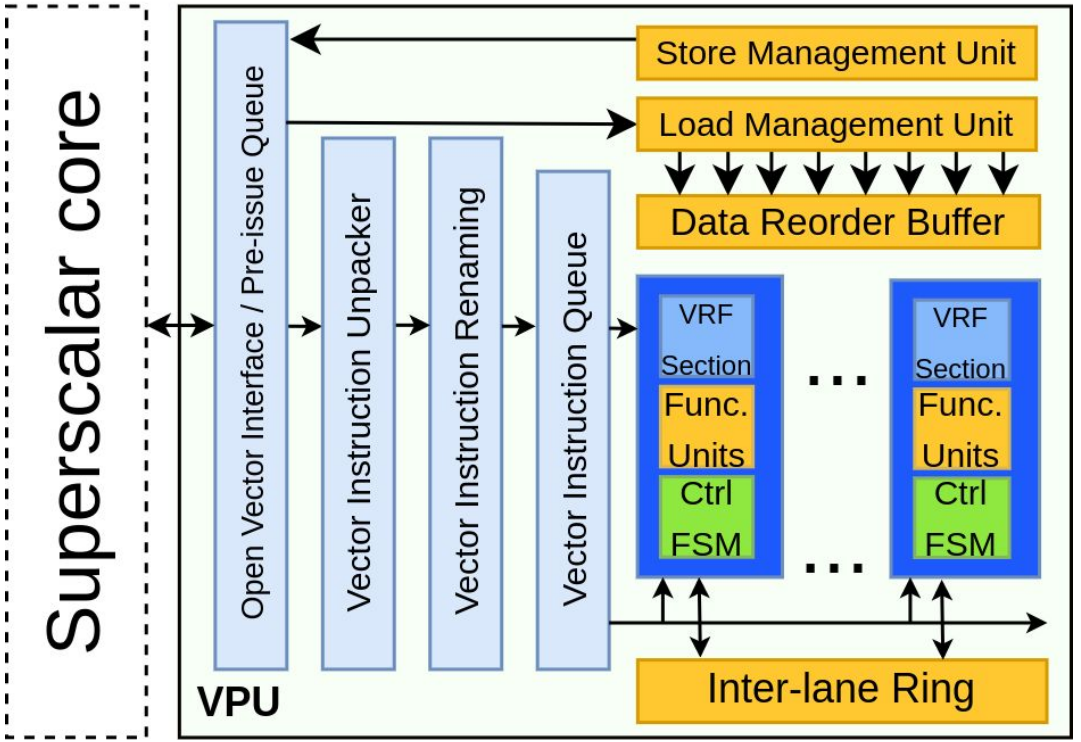
BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

accellera SYSTEMS INITIATIVE

- Free and open Instruction Set Architecture
- Growing rapidly, open collaboration
- HPC, automotive, transportation, cloud, communications, consumer, IoT, etc.
- Modular ISA
  - Vector extension

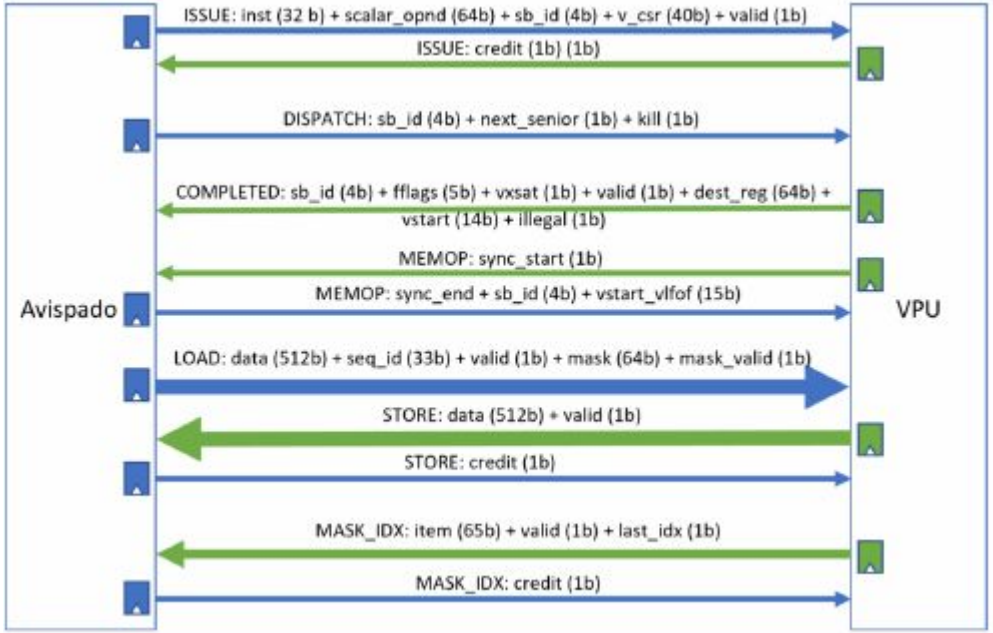
Vector Processing Unit (VPU)

- DUT
- Vector extension 0.7.1
  - Updating to 1.0
- Different “flavors”
  - EPI*
  - eProcessor*



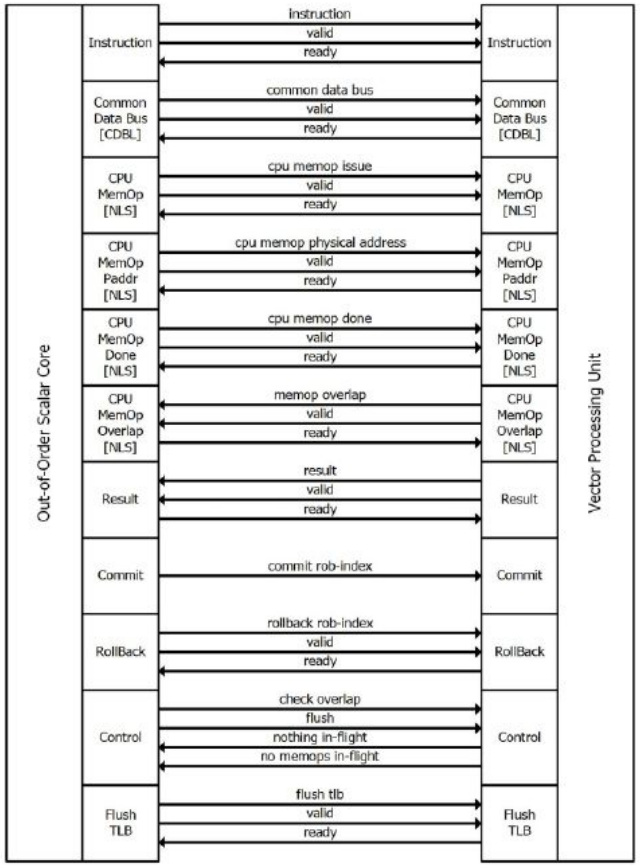
European Processor Initiative

- Core (Avispado) developed by partner (Semidynamics)
- Communicates with the VPU through OVI
- Vector memory accesses are performed by the core



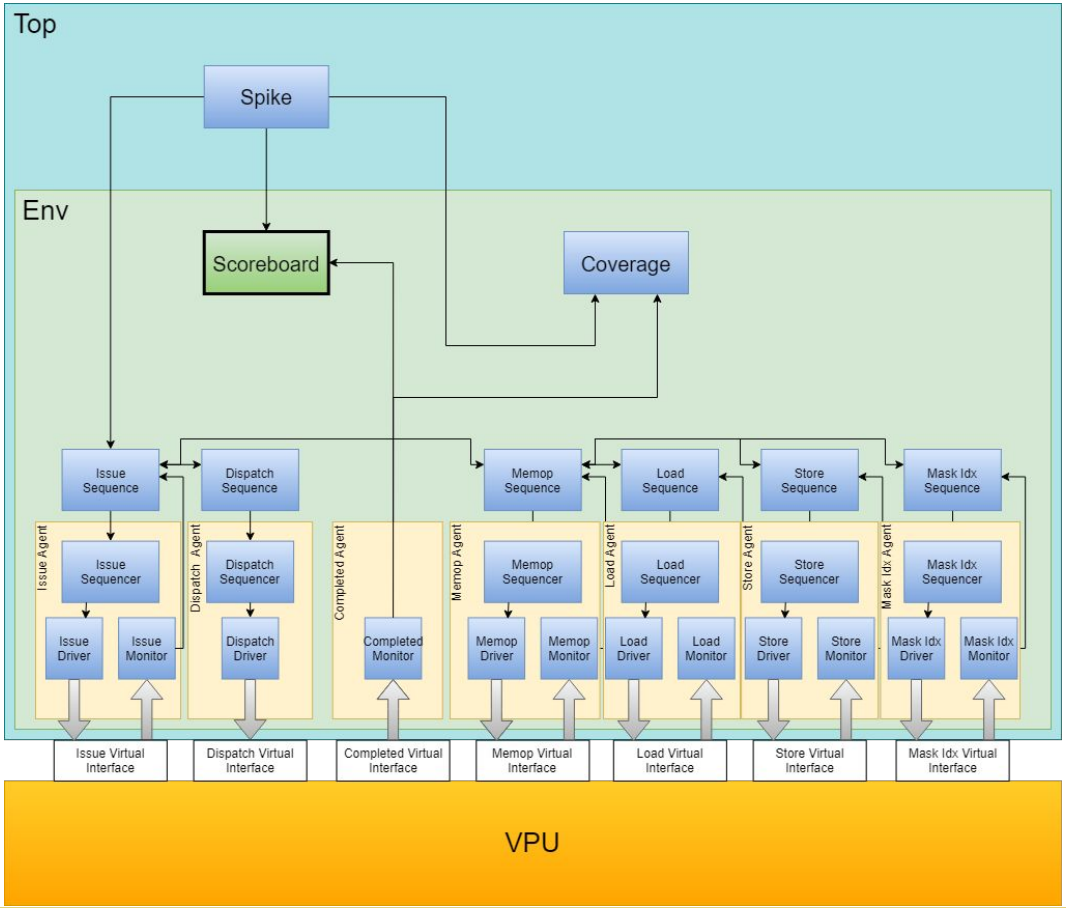
eProcessor

- Different interface for communicating with the core
- VPU access memory directly through AMBA5 CHI interface
- Interface needs signals to solve memory aliasing issues



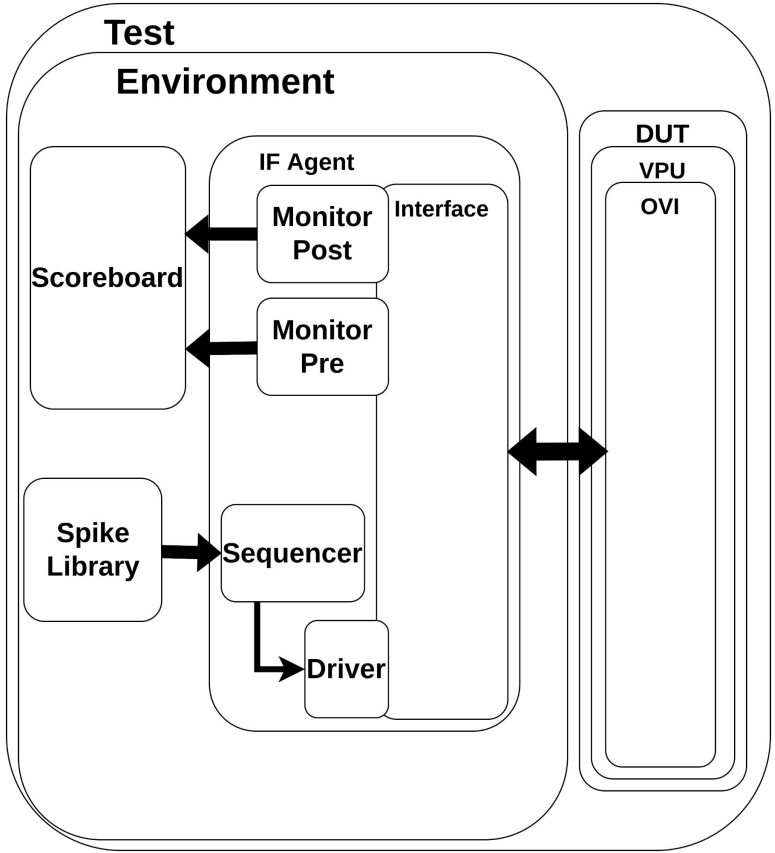
Previous environment

- One agent per each OVI channel
  - Massive interprocess communication
- Difficult to maintain, adapt and evolve

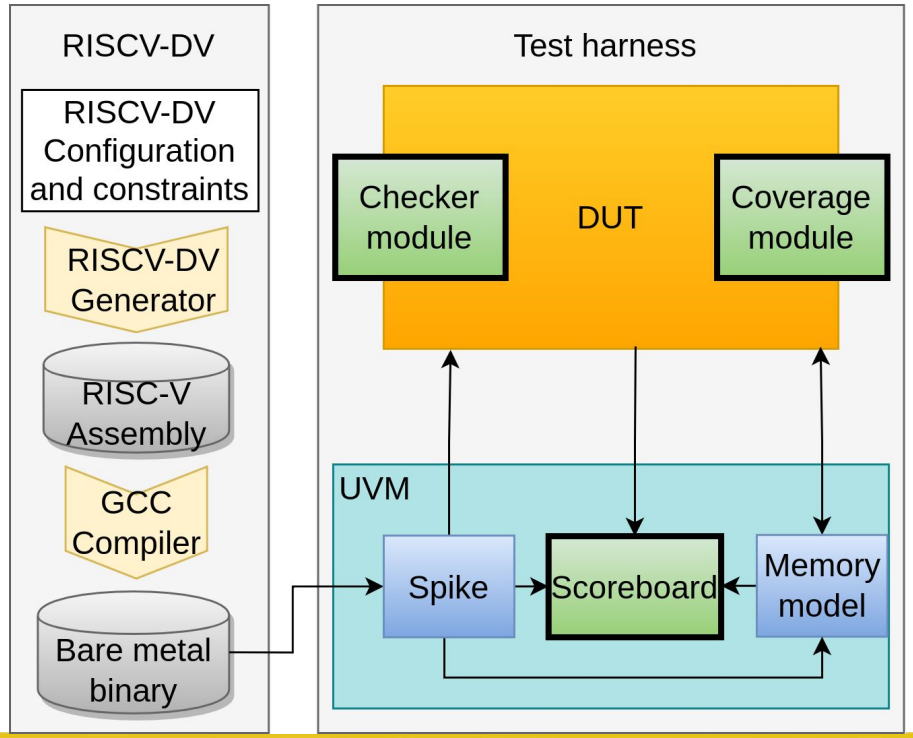


New UVM environment

- Common to all projects using the VPU
- Base interface-agnostic verification environment with common components (vpu-dv)
  - Test generation
  - Reference model usage
  - UVCs
  - Interface abstract class
  - Continuous integration
  - Coverage collection and reporting
- Project-specific environment
  - Interface implementation
  - Modified reference model library
  - Specific configurations



Verification environment flow

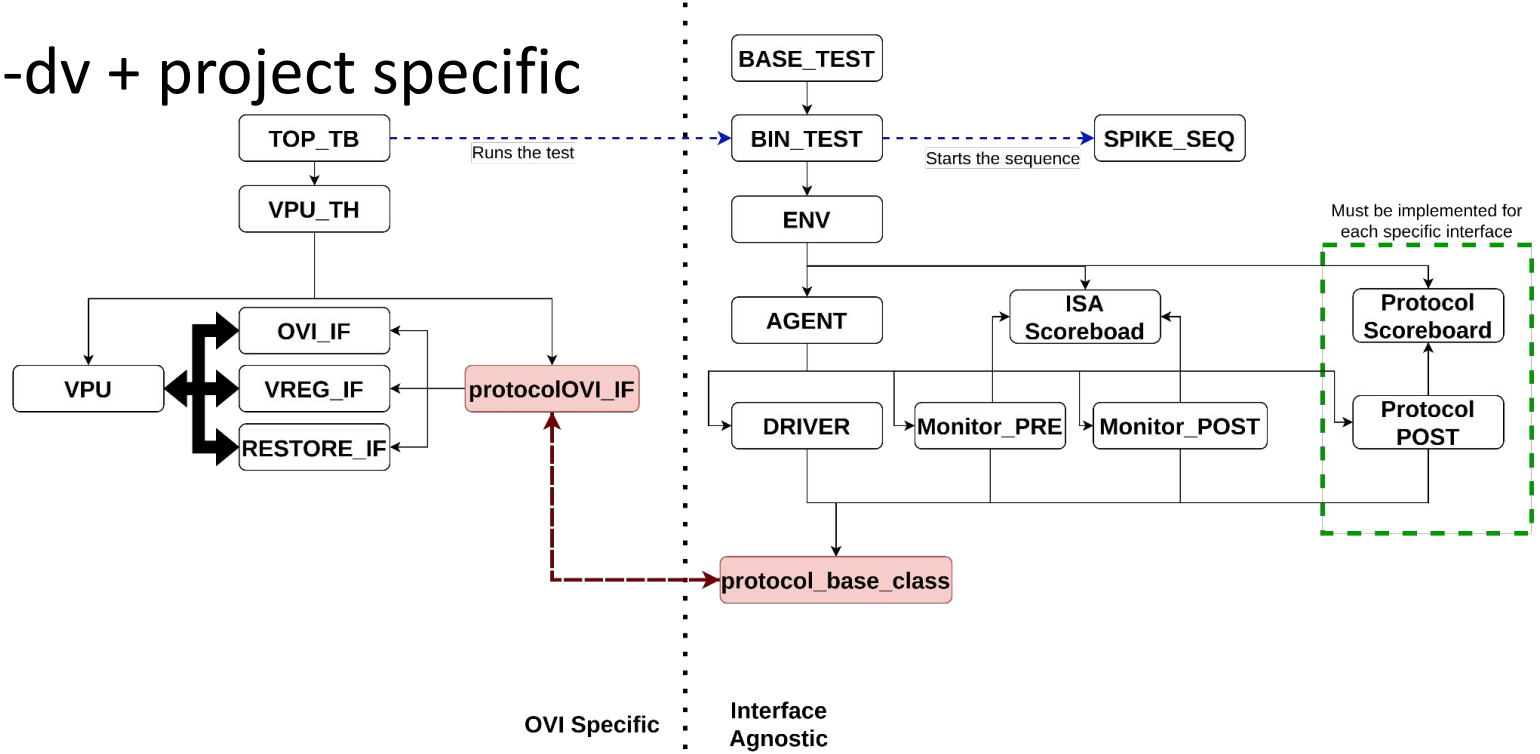


vpu-dv

- Interface base abstract class
  - With methods called by UVCs
  - Implemented in specific project environment

```
1.  pure virtual task do_protocol(); //Runs the specific protocol of the interface to stimulate the DUT
2.  pure virtual task wait_for_clk(int unsigned num_cycles = 1) ; //Waits for as many num_cycles cycles of the interface clock
3.  pure virtual function drive (ins_tx req); //Pushes the instruction inside the transaction into the pending instructions queue
4.  pure virtual function bit new_ins_tx(); //Returns whether or not there are new instructions received from the driver
5.  pure virtual function iss_state_t monitor_pre(); //Returns the first pending instruction received from the driver
6.  pure virtual function bit new_dut_tx(); //Returns whether or not there are new completed instructions
7.  pure virtual function dut_state_t monitor_post(); //Returns the first pending completed instruction's result
8.  pure virtual function bit new_protocol_tx(); //Returns whether or not there are new completed instructions
9.  pure virtual function protocol_instr_t monitor_protocol(); //Returns the first pending completed instruction
10. pure virtual function protocol_instr_t next_infl_instr(); //Returns the first inflight instruction
```

vpu-dv + project specific



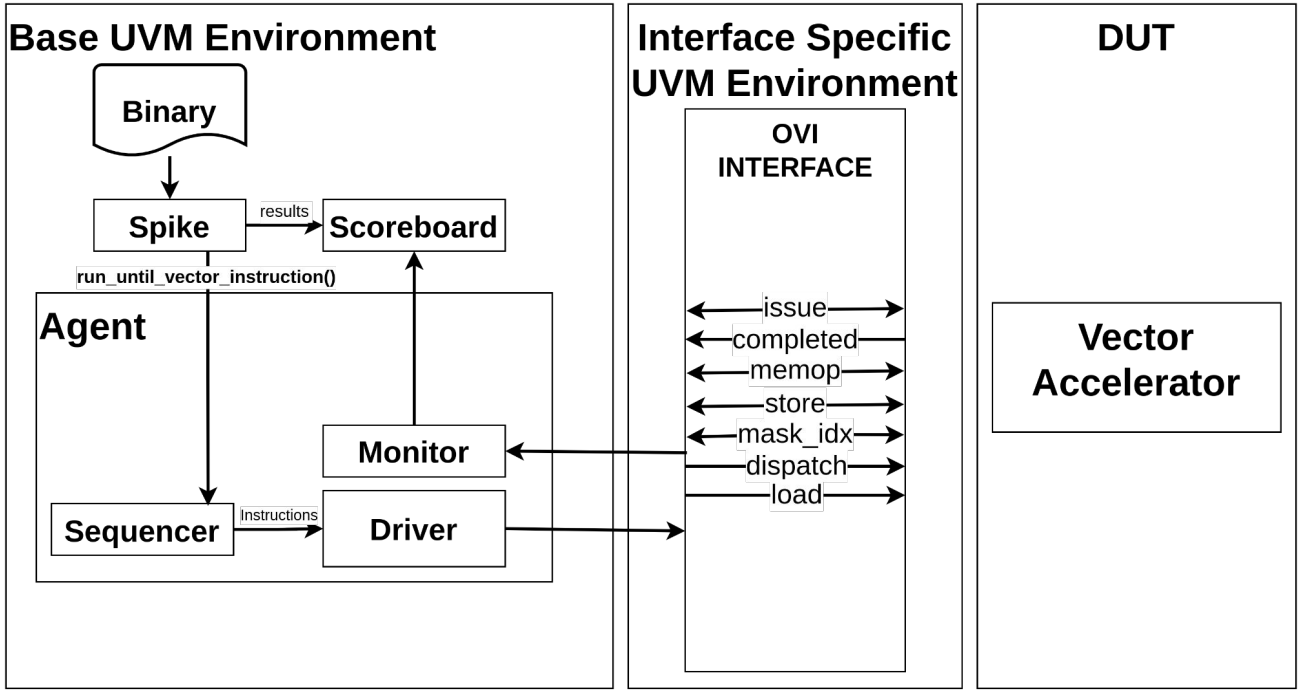
vpu-dv

- Reference model
  - Wrapper abstract class
    - Declares methods to be implemented by ISS.
    - Wrapper class overridden in build phase of uvm_test:
      - set_type_override_by_type(env_pkg::env::get_type(), eprocessor_env::get_type());
  - Derived class
    - Implements methods
- Spike
  - C++ implemented
  - Communication done using DPI.
  - Specific versions for each project, with different configurations

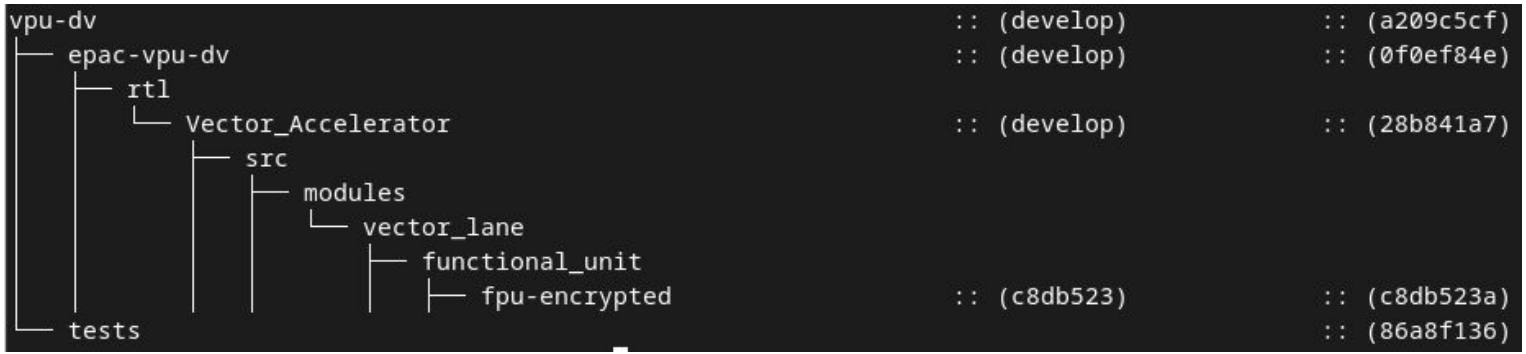
vpu-dv

- Test generation
  - tests repository as submodule
    - ISA tests with different configurations
    - Branch of submodule specified by project environment
  - RISCV-DV random binary generator
    - Modified according to our needs and for different projects.
    - Executed as first step in CI pipelines flow

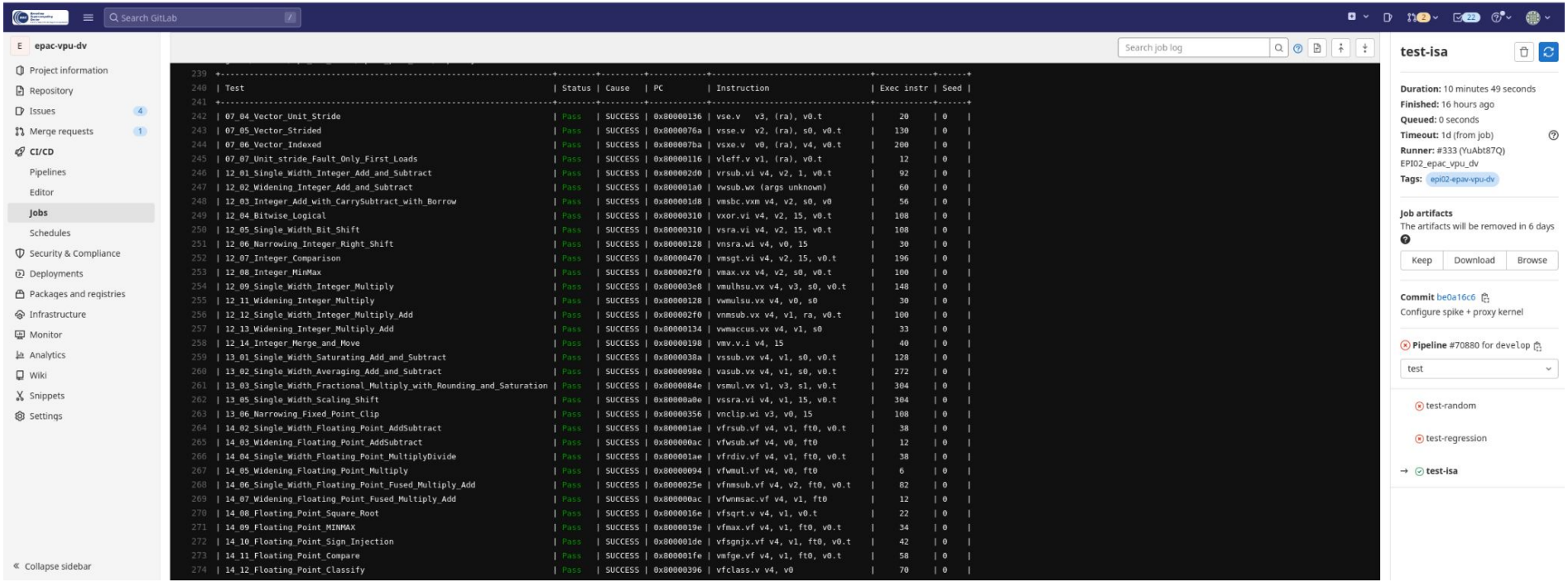
epac-vpu-dv


Base UVM Environment
Binary
Spike
results
Scoreboard
run_until_vector_instruction()
Agent
Monitor
Sequencer
Instructions
Driver
Interface Specific UVM Environment
OVI INTERFACE
issue
completed
memop
store
mask_idx
dispatch
load
DUT
Vector Accelerator

epac-vpu-dv

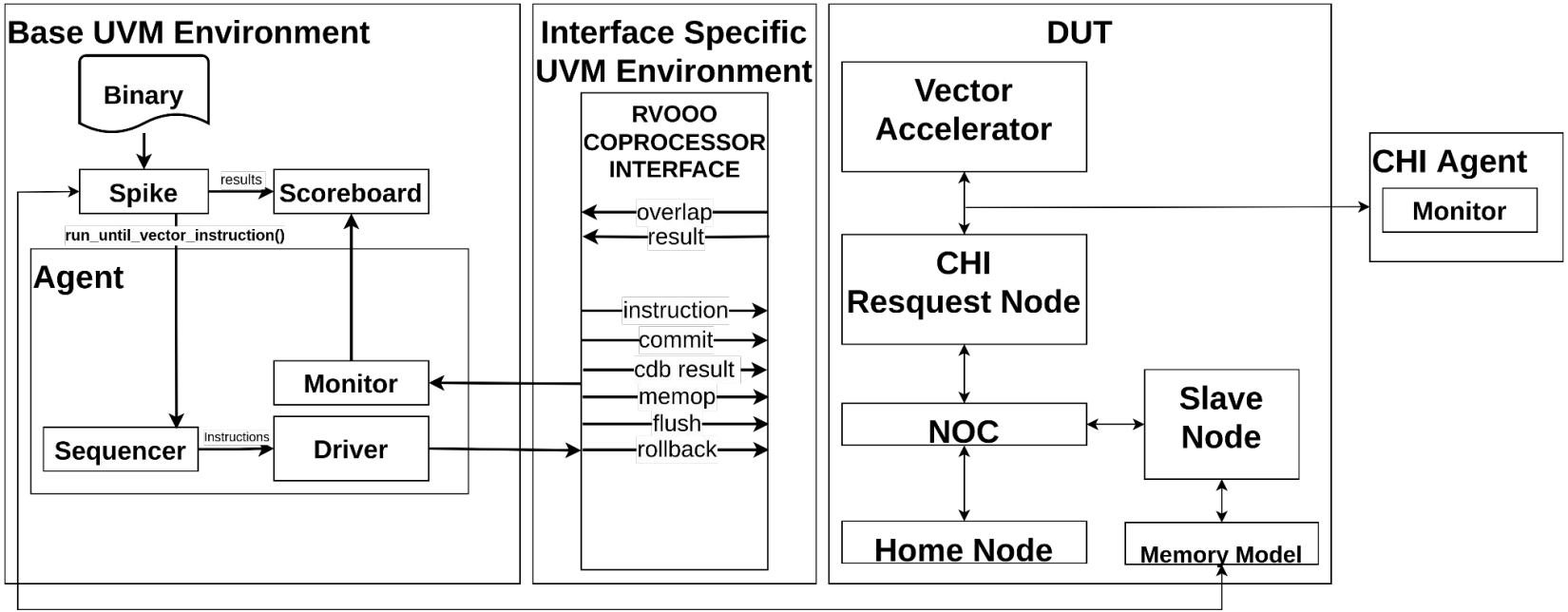

vpu-dv :: (develop) :: (a209c5cf)
epac-vpu-dv :: (develop) :: (0f0ef84e)
rtl
Vector_Accelerator :: (develop) :: (28b841a7)
src
modules
vector_lane
functional_unit
fpu-encrypted :: (c8db523) :: (c8db523a)
tests :: (86a8f136)

epac-vpu-dv



Test	Status	Cause	PC	Instruction	Exec instr	Seed
07_04_Vector_Unit_Stride	Pass	SUCCESS	0x80000136	vse.v v3, (ra), v0.t	20	0
07_05_Vector_Strided	Pass	SUCCESS	0x8000076a	vsse.v v2, (ra), s0, v0.t	130	0
07_06_Vector_Indexed	Pass	SUCCESS	0x800007ba	vsxe.v v0, (ra), v4, v0.t	200	0
07_07_Unit_stride_Fault_Only_First_Loads	Pass	SUCCESS	0x80000116	vleff.v v1, (ra), v0.t	12	0
12_01_Single_Width_Integer_Add_and_Subtract	Pass	SUCCESS	0x800002d0	vrsub.vi v4, v2, 1, v0.t	92	0
12_02_Widening_Integer_Add_and_Subtract	Pass	SUCCESS	0x800001a0	vwsub.wx (args unknown)	60	0
12_03_Integer_Add_with_CarrySubtract_with_Borrow	Pass	SUCCESS	0x800001d8	vmsbc.vxm v4, v2, s0, v0	56	0
12_04_Bitwise_Logical	Pass	SUCCESS	0x80000310	vxor.vi v4, v2, 15, v0.t	108	0
12_05_Single_Width_Bit_Shift	Pass	SUCCESS	0x80000310	vsra.vi v4, v2, 15, v0.t	108	0
12_06_Narrowing_Integer_Right_Shift	Pass	SUCCESS	0x80000128	vnsra.wi v4, v0, 15	30	0
12_07_Integer_Comparison	Pass	SUCCESS	0x80000470	vmsgt.vi v4, v2, 15, v0.t	196	0
12_08_Integer_MinMax	Pass	SUCCESS	0x800002f0	vmax.vx v4, v2, s0, v0.t	100	0
12_09_Single_Width_Integer_Multiply	Pass	SUCCESS	0x800003e8	vmulhsu.vx v4, v3, s0, v0.t	148	0
12_11_Widening_Integer_Multiply	Pass	SUCCESS	0x80000128	vwmulsu.vx v4, v0, s0	30	0
12_12_Single_Width_Integer_Multiply_Add	Pass	SUCCESS	0x800002f0	vnmsub.vx v4, v1, ra, v0.t	100	0
12_13_Widening_Integer_Multiply_Add	Pass	SUCCESS	0x80000134	vwmaccus.vx v4, v1, s0	33	0
12_14_Integer_Merge_and_Move	Pass	SUCCESS	0x80000198	vmv.v.i v4, 15	40	0
13_01_Single_Width_Saturating_Add_and_Subtract	Pass	SUCCESS	0x8000038a	vssub.vx v4, v1, s0, v0.t	128	0
13_02_Single_Width_Averaging_Add_and_Subtract	Pass	SUCCESS	0x8000098e	vasub.vx v4, v1, s0, v0.t	272	0
13_03_Single_Width_Fractional_Multiply_with_Rounding_and_Saturation	Pass	SUCCESS	0x8000084e	vsmul.vx v1, v3, s1, v0.t	304	0
13_05_Single_Width_Scaling_Shift	Pass	SUCCESS	0x80000a8e	vssra.vi v4, v1, 15, v0.t	304	0
13_06_Narrowing_Fixed_Point_Clip	Pass	SUCCESS	0x80000356	vnclip.wi v3, v0, 15	108	0
14_02_Single_Width_Floating_Point_AddSubtract	Pass	SUCCESS	0x800001ae	vfrsub.vf v4, v1, ft0, v0.t	38	0
14_03_Widening_Floating_Point_AddSubtract	Pass	SUCCESS	0x800000ac	vfwsub.wf v4, v0, ft0	12	0
14_04_Single_Width_Floating_Point_MultiplyDivide	Pass	SUCCESS	0x800001ae	vfrdiv.vf v4, v1, ft0, v0.t	38	0
14_05_Widening_Floating_Point_Multiply	Pass	SUCCESS	0x80000094	vfwmul.vf v4, v0, ft0	6	0
14_06_Single_Width_Floating_Point_Fused_Multiply_Add	Pass	SUCCESS	0x8000025e	vfnmsub.vf v4, v2, ft0, v0.t	82	0
14_07_Widening_Floating_Point_Fused_Multiply_Add	Pass	SUCCESS	0x800000ac	vfwnmsac.vf v4, v1, ft0	12	0
14_08_Floating_Point_Square_Root	Pass	SUCCESS	0x8000016e	vfsqrt.v v4, v1, v0.t	22	0
14_09_Floating_Point_MINMAX	Pass	SUCCESS	0x8000019e	vfmax.vf v4, v1, ft0, v0.t	34	0
14_10_Floating_Point_Sign_Injection	Pass	SUCCESS	0x800001de	vfsgnjx.vf v4, v1, ft0, v0.t	42	0
14_11_Floating_Point_Compare	Pass	SUCCESS	0x800001fe	vmfge.vf v4, v1, ft0, v0.t	58	0
14_12_Floating_Point_Classify	Pass	SUCCESS	0x80000396	vfclass.v v4, v0	70	0

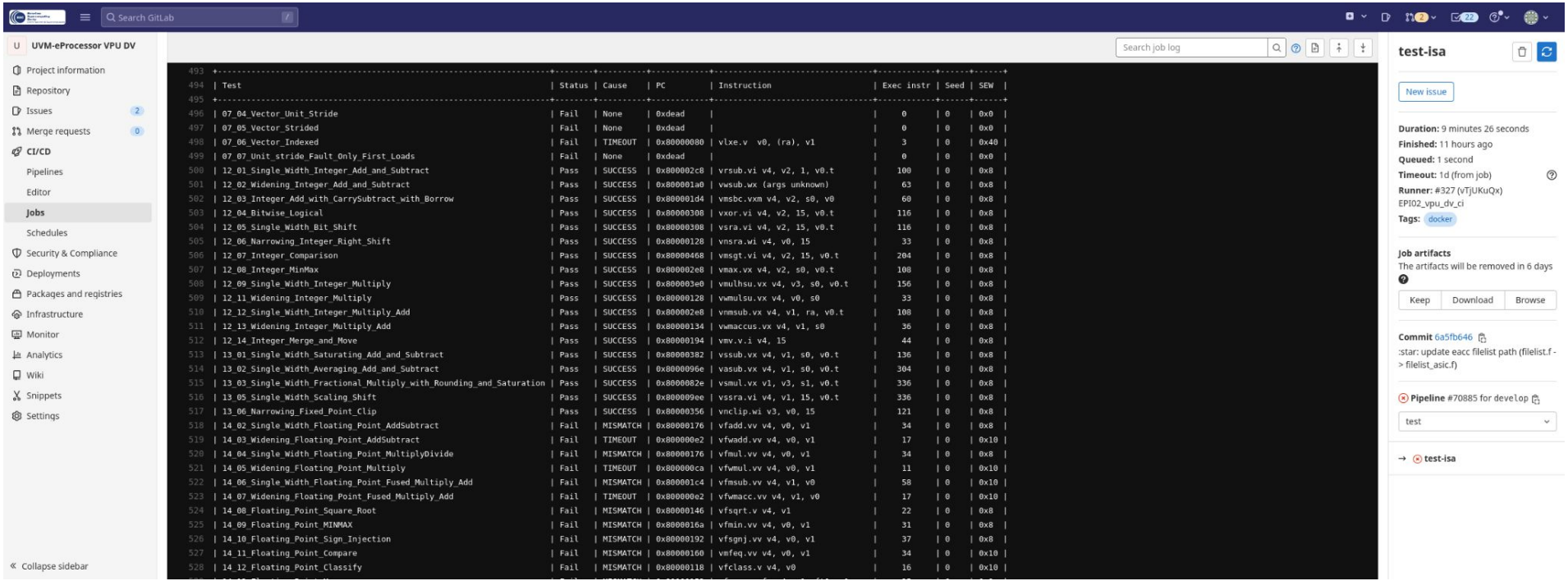
eprocessor-vpu-dv



eprocessor-vpu-dv

```
vpu-dv                                                  :: (develop)        :: (a209c5cf)
├── eprocessor-vpu-dv                                   :: (develop)        :: (473c2506)
│   ├── rtl
│   │   ├── eAccelerator                                :: (develop)        :: (06d011bc)
│   │   │   ├── docs
│   │   │   │   └── specs                               :: (d088cc5)        :: (d088cc5f)
│   │   │   ├── src
│   │   │   │   ├── package
│   │   │   │   │   └── riscv_pkg                                           :: (b2a90d84)
│   │   ├── epi_nocsim                                  :: (dev)            :: (b34c3283)
│   │   │   ├── modules
│   │   │   │   ├── ep-fake-snf                                             :: (57dea70a)
│   │   │   │   ├── ep-rni                                                  :: (15e118a2)
│   │   │   │   ├── l2hn                                                    :: (7f07a642)
│   │   │   │   ├── noc_ep0                                                 :: (29897d7a)
│   │   │   │   ├── ProNoC                                                  :: (7d728b19)
│   │   │   │   └── vpu_chi_interface                                       :: (e0ecafe1)
└── tests                                                                   :: (86a8f136)
```

eprocessor-vpu-dv



Test	Status	Cause	PC	Instruction	Exec instr	Seed	SEW
07_04_Vector_Unit_Stride	Fail	None	0xdead		0	0	0x0
07_05_Vector_Strided	Fail	None	0xdead		0	0	0x0
07_06_Vector_Indexed	Fail	TIMEOUT	0x80000080	vlxe.v v0, (ra), v1	3	0	0x40
07_07_Unit_stride_Fault_Only_First_Loads	Fail	None	0xdead		0	0	0x0
12_01_Single_Width_Integer_Add_and_Subtract	Pass	SUCCESS	0x800002c8	vrsub.vi v4, v2, 1, v0.t	100	0	0x8
12_02_Widening_Integer_Add_and_Subtract	Pass	SUCCESS	0x800001a0	vwsub.wx (args unknown)	63	0	0x8
12_03_Integer_Add_with_CarrySubtract_with_Borrow	Pass	SUCCESS	0x800001d4	vmsbc.vxm v4, v2, s0, v0	60	0	0x8
12_04_Bitwise_Logical	Pass	SUCCESS	0x80000308	vxor.vi v4, v2, 15, v0.t	116	0	0x8
12_05_Single_Width_Bit_Shift	Pass	SUCCESS	0x80000308	vsra.vi v4, v2, 15, v0.t	116	0	0x8
12_06_Narrowing_Integer_Right_Shift	Pass	SUCCESS	0x80000128	vnsra.wi v4, v0, 15	33	0	0x8
12_07_Integer_Comparison	Pass	SUCCESS	0x80000468	vmsgt.vi v4, v2, 15, v0.t	204	0	0x8
12_08_Integer_MinMax	Pass	SUCCESS	0x800002e8	vmax.vx v4, v2, s0, v0.t	108	0	0x8
12_09_Single_Width_Integer_Multiply	Pass	SUCCESS	0x800003e0	vmulhsu.vx v4, v3, s0, v0.t	156	0	0x8
12_11_Widening_Integer_Multiply	Pass	SUCCESS	0x80000128	vwmulsu.vx v4, v0, s0	33	0	0x8
12_12_Single_Width_Integer_Multiply_Add	Pass	SUCCESS	0x800002e8	vnmsub.vx v4, v1, ra, v0.t	108	0	0x8
12_13_Widening_Integer_Multiply_Add	Pass	SUCCESS	0x80000134	vwmaccus.vx v4, v1, s0	36	0	0x8
12_14_Integer_Merge_and_Move	Pass	SUCCESS	0x80000194	vmv.v.i v4, 15	44	0	0x8
13_01_Single_Width_Saturating_Add_and_Subtract	Pass	SUCCESS	0x80000382	vssub.vx v4, v1, s0, v0.t	136	0	0x8
13_02_Single_Width_Averaging_Add_and_Subtract	Pass	SUCCESS	0x8000096e	vasub.vx v4, v1, s0, v0.t	304	0	0x8
13_03_Single_Width_Fractional_Multiply_with_Rounding_and_Saturation	Pass	SUCCESS	0x8000082e	vsmul.vx v1, v3, s1, v0.t	336	0	0x8
13_05_Single_Width_Scaling_Shift	Pass	SUCCESS	0x800009ee	vssra.vi v4, v1, 15, v0.t	336	0	0x8
13_06_Narrowing_Fixed_Point_Clip	Pass	SUCCESS	0x80000356	vnclip.wi v3, v0, 15	121	0	0x8
14_02_Single_Width_Floating_Point_AddSubtract	Fail	MISMATCH	0x80000176	vfadd.vv v4, v0, v1	34	0	0x8
14_03_Widening_Floating_Point_AddSubtract	Fail	TIMEOUT	0x800000e2	vfwadd.vv v4, v0, v1	17	0	0x10
14_04_Single_Width_Floating_Point_MultiplyDivide	Fail	MISMATCH	0x80000176	vfmul.vv v4, v0, v1	34	0	0x8
14_05_Widening_Floating_Point_Multiply	Fail	TIMEOUT	0x800000ca	vfwmul.vv v4, v0, v1	11	0	0x10
14_06_Single_Width_Floating_Point_Fused_Multiply_Add	Fail	MISMATCH	0x800001c4	vfmsub.vv v4, v1, v0	58	0	0x10
14_07_Widening_Floating_Point_Fused_Multiply_Add	Fail	TIMEOUT	0x800000e2	vfwmacc.vv v4, v1, v0	17	0	0x10
14_08_Floating_Point_Square_Root	Fail	MISMATCH	0x80000146	vfsqrt.v v4, v1	22	0	0x8
14_09_Floating_Point_MINMAX	Fail	MISMATCH	0x8000016a	vfmin.vv v4, v0, v1	31	0	0x8
14_10_Floating_Point_Sign_Injection	Fail	MISMATCH	0x80000192	vfsgnj.vv v4, v0, v1	37	0	0x8
14_11_Floating_Point_Compare	Fail	MISMATCH	0x80000160	vmfeq.vv v4, v0, v1	34	0	0x10
14_12_Floating_Point_Classify	Fail	MISMATCH	0x80000118	vfclass.v v4, v0	16	0	0x10

Conclusions

- Developed a verification infrastructure reusable for many projects
  - Currently for EPI SGA1 and eProcessor
  - In development for EPI SGA2 (RVV 1.0)
- Easy to extend and maintain
- Ability to uncover more bugs

Acknowledgments

*This research has received funding from the European High Performance Computing Joint Undertaking (JU) under Framework Partnership Agreement No 800928 (European Processor Initiative) and Specific Grant Agreement No 101036168 (EPI SGA2) and No 956702 (eProcessor) .

The JU receives support from the European Union's Horizon 2020 research and innovation programme and from Croatia, France, Germany,Greece, Italy, Netherlands, Portugal, Spain, Sweden, and Switzerland.

The EPI-SGA2 project,PCI2022-132935_N1618737 is also co-funded by MCIN/AEI /10.13039/501100011033 and by the UE NextGenerationEU/PRTR.

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Questions?

accellera SYSTEMS INITIATIVE