

Practical Adoption of Open Source System Verilog Tools

CHIPS Alliance Fall Workshop, 2021-10-12

Michael Gielda, mgielda@antmicro.com





Enabling Faster Innovation

Scalable Hybrid Computation

Broadening the Outreach

Bridging Existing Methodologies

Breaking Down Complexity

Enabling Collaboration



Enabling Faster Innovation

Scalable Hybrid Computation

Broadening the Outreach

Bridging Existing Methodologies

Breaking Down Complexity

Enabling Collaboration

WHY DO WE NEED SYSTEMVERILOG SUPPORT IN OPEN SOURCE TOOLS?

- Proprietary licensing of existing tools makes it hard to build scalable, reproducible CIs
 - Especially publicly accessible CIs in multi-org projects - OpenTitan, CHIPS Alliance
- Number of open source cores and lots of pre-existing IP implemented in SystemVerilog, e.g.
 - [SweRV](#)
 - [Ibex](#)
 - [BlackParrot](#)
 - [Core-V](#)
- Open source tools will help building a collaborative ecosystem around ASIC and FPGA design



HOW TO BUILD AN OPEN SOURCE SYSTEMVERILOG ECOSYSTEM

- Identify missing functionalities and features
- Reuse existing solutions
 - There are many existing projects which can be improved
- Create well documented and transparent projects
 - Include automated tests and status reporting in projects
- Cooperate with others
 - Gather information on what is needed
- **Provide incremental value**



CHIPS ALLIANCE & HDLS

- CHIPS wants to encourage all sorts of open source ASIC development - and acknowledges that the future ecosystem will be pluralistic
- workgroups for both Chisel and SystemVerilog peacefully coexist and collaborate within our Technical Steering Committee
- See <https://chipsalliance.org/workgroups/> for list of workgroups, <https://github.com/chipsalliance/tsc> for current TSC composition

The logo for CHISEL, rendered in a bold, blue, sans-serif font. The letters 'I' and 'E' have small white circles at their top and bottom ends, resembling pins or electrical connections.The logo for SystemVerilog, featuring the text "SystemVerilog" in a purple serif font. The text is centered within a blue and green swoosh that curves around it from the bottom and sides.

SO, WHAT DO WE MEAN BY PRACTICAL ADOPTION?



LAY OF THE LAND: SV-TESTS

- Test suite to determine the SystemVerilog support level in various open source tools
- Pinpoint all the supported/missing SV features
- Report: symbiflow.github.io/sv-tests
- Three types of tests:
 - SystemVerilog features
 - Existing third party test suites
 - Selected open source IP cores (SweRV, Ibex etc.)

	icarus	moore	moore_parse	odin	slang	surelog	sv2v	zachs	sv_parser	tree_sitter_verilog	verible	verilator	yosys
Ariane RISC-V core	ariane	0/1	0/1	0/1	0/1	0/1	0/1	1/1	1/1	0/1	0/1	0/1	0/1
Tests imported from Basejump STL	basejump	0/302	32/302	164/302	0/302	97/302	295/302	292/302	296/302	0/302	295/302	98/302	6/302
BlackParrot RISC-V core	black-parrot	0/4	0/4	0/4	0/4	0/4	0/4	4/4	0/4	0/4	0/4	0/4	0/4
	earlgrey	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
FX68k m68k core	fd68k	0/1	0/1	1/1	0/1	1/1	1/1	1/1	1/1	0/1	1/1	1/1	0/1
Tests imported from hdlconv	hdlconv	0/306	70/306	127/306	0/306	09/306	283/306	87/306	305/306	191/306	261/306	67/306	44/306
Ibex RISC-V core	ibex	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	0/1	1/1	1/1	0/1
Tests imported from invest	invest	0/2173	451/1902	1474/1902	298/1902	1354/1902	1868/2173	1484/1902	1899/1902	347/1902	1873/1902	1501/2173	450/2173
RSD RISC-V core	rsd	0/1	0/1	0/1	0/1	0/1	0/1	0/1	1/1	0/1	0/1	0/1	0/1
Various sanity checks	sanity	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
SCR1 RISC-V core	scr1	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	0/1	0/1	0/1	0/1
SweRV RISC-V core	swerv	0/1	0/1	0/1	0/1	0/1	1/1	1/1	0/1	0/1	0/1	1/1	0/1
Taiga RISC-V core	taiga	0/1	0/1	0/1	0/1	0/1	1/1	0/1	1/1	0/1	0/1	0/1	0/1
Tests imported from utd-SystemVerilog	utd-sv	0/295	114/295	292/295	183/295	119/295	294/295	287/295	269/295	272/295	294/295	125/295	286/295
Tests imported from UVM	uvm	0/1	0/1	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	0/1	0/1
Tests imported from Yosys	yosys	0/184	132/184	155/184	72/184	167/184	184/184	172/184	183/184	164/184	176/184	169/184	154/184
White space	5.3	0/7	7/7	7/7	1/7	7/7	7/7	7/7	7/7	0/1	1/1	7/7	7/7
Comments	5.4	0/7	7/7	7/7	0/7	7/7	7/7	7/7	7/7	1/1	1/1	7/7	7/7
Identifiers, keywords, and system names	5.6	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
Escaped identifiers	5.6.1	0/2	2/2	2/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2

Search

```

@config: full swerv core test
cc: 0 (swerv success: 1)
Module: full 0
top: none
topdir: /tmp/verilog/sv-tests/third_party/core/verilog/design/include
top: /tmp/verilog/sv-tests/third_party/core/verilog/design/include
type: partition
mode: partition
files: /tmp/verilog/sv-tests/third_party/core/verilog/design/include
time: 1.44s

```

```

1 // NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE NOTE
2 // This is an automatically generated file by Joseph Roman on Thu Oct 10 13:12:18 PDT 2019
3 //
4 // 0: define HW_DATA_ACCESS_MK000 'XXXXXXXX'
5 //
6 #define HW_DATA_ACCESS_ENABLED 1:00
7 #define HW_DATA_ACCESS_ENABLED 1:00
8 #define HW_DATA_ACCESS_ENABLED 1:00
9 #define HW_DATA_ACCESS_ENABLED 1:00
10 #define HW_DATA_ACCESS_ENABLED 1:00
11 #define HW_DATA_ACCESS_ENABLED 1:00
12 #define HW_DATA_ACCESS_MK000 'XXXXXXXX'

```

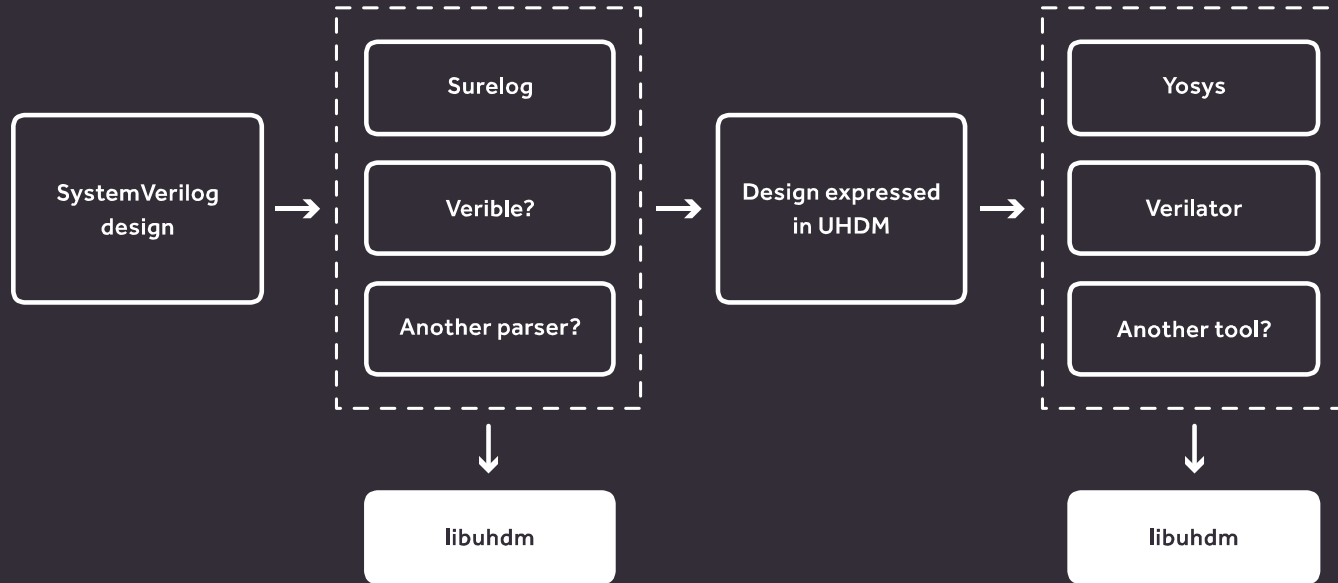


REUSE: SURELOG / UHDM

- How to provide good SV support across the board?
- [Surelog](#): open source SystemVerilog 2017 Pre-processor, Parser, Elaborator and UHDM Compiler
- Universal Hardware Data Model ([UHDM](#)) is used to exchange the information about elaborated SV design between the parser and other tools
- Integrating UHDM into Verilator and Yosys
- Can [parse, synthesize and simulate OpenTitan's lbex core](#) directly from source - <https://github.com/chipsalliance/UHDM-integration-tests>



UHDM/SURELOG + VERILATOR/YOSYS



INCREMENTAL VALUE: VERIBLE

- Open source SV linter/formatter
- Released by Google on [GitHub](#), onboarded into CHIPS
- Actively developed by Google and Antmicro
- Used e.g. in Ibex (OpenTitan) CI
- Very versatile, with lots of practical uses

Files	Code
<pre> ./ rtl/ ibex_alu.sv ibex_compressed_decoder.sv ibex_controller.sv ibex_core.sv ibex_cs_registers.sv ibex_decoder.sv ibex_ex_block.sv ibex_fetch_fifo.sv ibex_id_stage.sv ibex_if_stage.sv ibex_load_store_unit.sv ibex_multdiv_fast.sv ibex_multdiv_slow.sv ibex_pkg.sv </pre>	<pre> 1 // Copyright lowRISC contributors. 2 // Copyright 2018 ETH Zurich and University of Bologna, see also CREDITS.md. 3 // Licensed under the Apache License, Version 2.0, see LICENSE for details. 4 // SPDX-License-Identifier: Apache-2.0 5 6 `ifdef RISC_V_FORMAL 7 `define RVFI 8 `endif 9 10 `include "prim_assert.sv" 11 12 /** 13 * Top level module of the ibex RISC-V core 14 */ 15 module ibex_core #(16 parameter bit PMPEnable = 1'b0, 17 parameter int unsigned PMPGranularity = 0, 18 parameter int unsigned PMPNumRegions = 4, 19 parameter int unsigned MHPMCounterNum = 0, 20 parameter int unsigned MHPMCounterWidth = 40, 21 parameter bit RV32E = 1'b0, 22 parameter ibex_pkg::rv32m_e RV32M = ibex_pkg::RV32MFast, </pre>
<ul style="list-style-type: none"> • Definitions: <ul style="list-style-type: none"> ◦ rtl/ibex_core.sv <ul style="list-style-type: none"> 33: parameter int unsigned DmHaltAddr = 32'h1A110800, • References: <ul style="list-style-type: none"> ◦ rtl/ibex_core.sv <ul style="list-style-type: none"> 401: .DmHaltAddr (DmHaltAddr), 	

VERIBLE - LINTING

- Linter — a static code analysis tool to spot and fix stylistic errors and bugs in SystemVerilog code
- Allows enforcement of rules on a project or company level in various SystemVerilog projects to follow authoritative style guides
- The rules vary from simple ones to more sophisticated and are highly configurable
- E.g. making sure the module name matches the file name, checking variable naming conventions

```
$ verible-verilog-lint --ruleset all core.sv
core.sv:3:11: Interface names must use lower_snake_case
naming convention and end with _if. [Style: interface-
conventions] [interface-name-style]
```

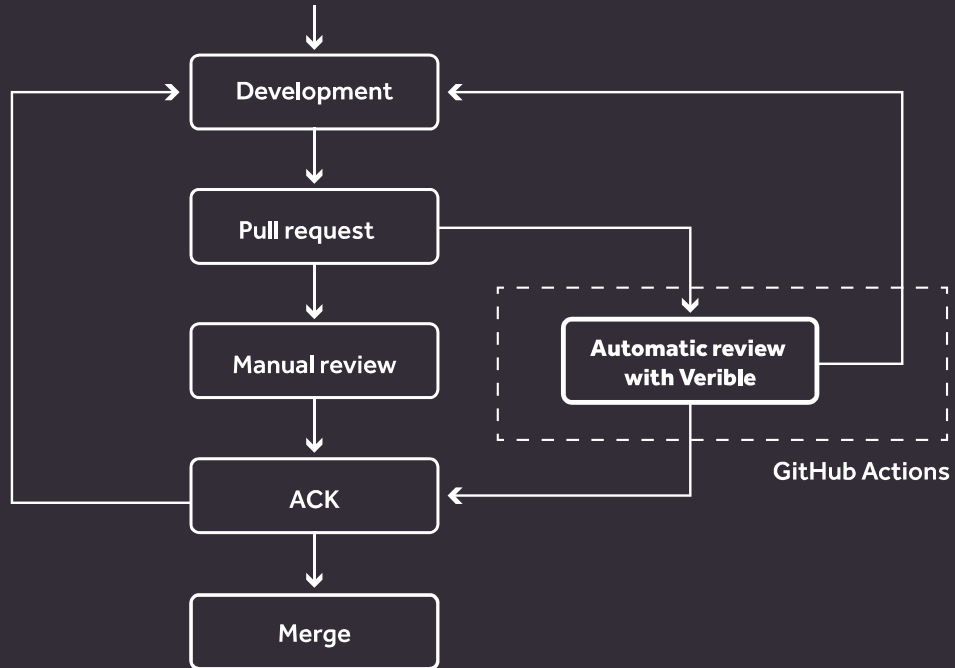
VERIBLE - FORMATTING

- Formatter — a complementary tool for the linter used to automatically detect various formatting issues like improper indentation or alignment
- As opposed to the linter, it only detects and fixes issues that have no lexical impact on the source code
- Linter + formatter can effectively remove all the discussions about styling, preferences and conventions from all pull requests
- Developers can then focus solely on the technical aspects of the proposed changes

```
$ cat sample.sv
typedef struct {
    bit first;
        bit second;
    bit
        third
        ;
    bit fourth;
    bit fifth; bit sixth;
}
foo_t;
```

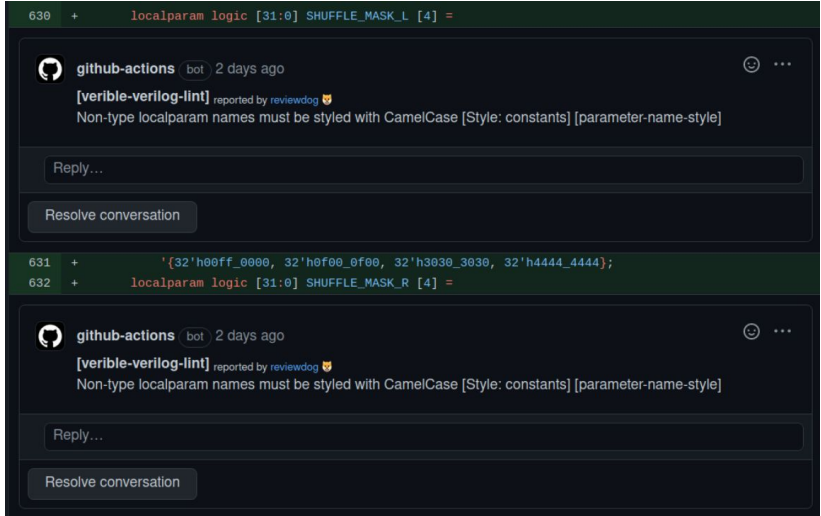
```
$ verible-verilog-format sample.sv
typedef struct {
    bit first;
    bit second;
    bit third;
    bit fourth;
    bit fifth;
    bit sixth;
} foo_t;
```

VERIBLE GITHUB ACTIONS



VERIBLE LINTER GITHUB ACTION

- [chipsalliance/verible-linter-action](https://github.com/chipsalliance/verible-linter-action)
- Any GitHub-hosted open-source or private project can inform committers about the issues detected in their code
- Wide range of applications can be devised, from faster pull requests reviews, to isolating erroneous portions of code
- [Blog note](#)



```
630 + localparam logic [31:0] SHUFFLE_MASK_L [4] =
```

github-actions bot 2 days ago
[verible-verilog-lint] reported by reviewdog
Non-type localparam names must be styled with CamelCase [Style: constants] [parameter-name-style]

Reply...

Resolve conversation

```
631 + '32'h00ff_0000, 32'h0f00_0f00, 32'h3030_3030, 32'h4444_4444};
632 + localparam logic [31:0] SHUFFLE_MASK_R [4] =
```

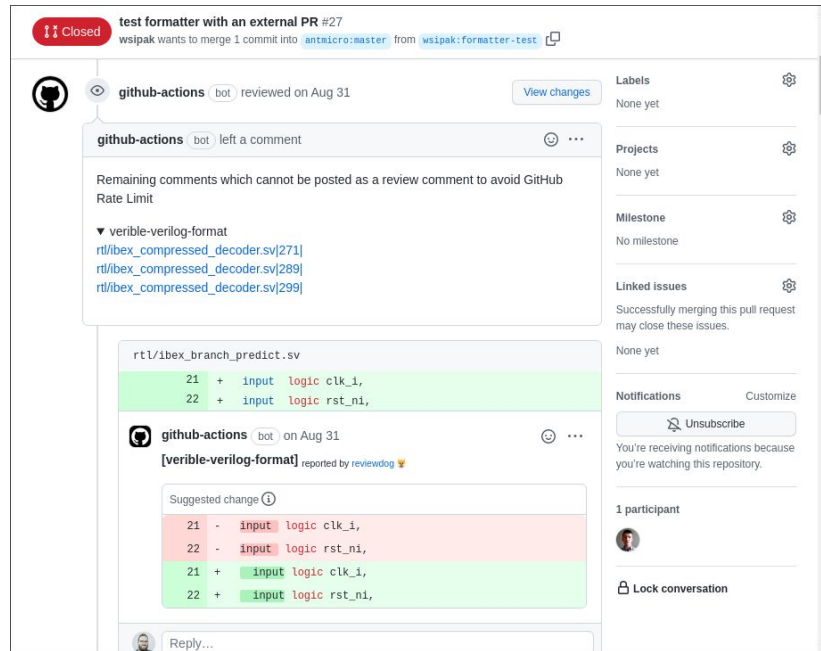
github-actions bot 2 days ago
[verible-verilog-lint] reported by reviewdog
Non-type localparam names must be styled with CamelCase [Style: constants] [parameter-name-style]

Reply...

Resolve conversation

VERIBLE FORMATTER GITHUB ACTION

- [chipsalliance/verible-formatter-action](https://github.com/chipsalliance/verible-formatter-action)
- Same as above, except it automatically generates code suggestions which can just be accepted - instant benefit!



test formatter with an external PR #27
wsipak wants to merge 1 commit into antmicro:master from wsipak:formatter-test

github-actions bot reviewed on Aug 31

github-actions bot left a comment

Remaining comments which cannot be posted as a review comment to avoid GitHub Rate Limit

▼ verible-verilog-format

- rt/ibex_compressed_decoder.sv[271]
- rt/ibex_compressed_decoder.sv[289]
- rt/ibex_compressed_decoder.sv[299]

```
rt/ibex_branch_predict.sv
21 + input logic clk_i,
22 + input logic rst_ni,
```

github-actions bot on Aug 31

[verible-verilog-format] reported by reviewdog

Suggested change

```
21 - input logic clk_i,
22 - input logic rst_ni,
21 + input logic clk_i,
22 + input logic rst_ni,
```


USING KYTHE FOR GENERATING INDEXED CODE DATABASE

- Verible can be used to perform many other tasks, e.g. generating a [Kythe](#) compatible indexing database
- Indexing an SV project simplifies collaboration - navigate through the source code using a web browser
- [Showcase GitHub repository](#)
- [Example index webpage](#)

Files

```

::/
  rtl/
    ibex_alu.sv
    ibex_compressed_decoder
    ibex_controller.sv
    ibex_core.sv
    ibex_cs_registers.sv
    ibex_decoder.sv
    ibex_ex_block.sv
    ibex_fetch_fifo.sv
    ibex_id_stage.sv
    ibex_if_stage.sv
    ibex_load_store_unit.sv
    ibex_multdiv_fast.sv
    ibex_multdiv_slow.sv
    ibex_pkg.sv

```

```

1 // Copyright lowRISC co
2 // Copyright 2018 ETH Z
3 // Licensed under the A
4 // SPDX-License-Identifi
5
6 `ifdef RISCVC_FORMAL
7   `define RVFI
8 `endif
9
10 `include "prim_assert.s
11
12 /**
13  * Top level module of
14  */
15 module ibex_core #(
16   parameter bit
17   parameter int unsig
18   parameter int unsig
19   parameter int unsig
20   parameter int unsig
21   parameter bit
22   parameter ibex_pkg:

```

• Definitions:

- rtl/ibex_core.sv

```
33: parameter int unsigned    DmHaltAddr    = 32'h1A110
```

• References:

- rtl/ibex_core.sv

```
401: .DmHaltAddr    ( DmHaltAddr    ),
```

COOPERATE WITH OTHERS: ADOPTION

- Our main focus in ongoing development was the OpenTitan project
- Currently pushing wider adoption at Google and other OT partners
- Now also used by some of the Core-V users, such as QuickLogic
- Starting collaboration with ZeroASIC
- Also working with the wider Ibex, BlackParrot, Core-V and SWeRV communities

The multi-colored Google logo.The logo for zerlo, consisting of the lowercase letters "z e r | o |" in a monospace font.The logo for QuickLogic, featuring the word "QuickLogic" in a bold sans-serif font with a red and grey swoosh graphic to the left.The logo for opentitan, featuring a pixelated gear icon to the left of the word "opentitan" in a bold sans-serif font.

REUSE / COLLABORATE: UVM IN OPEN SOURCE

- Lots of pre-existing IP and test benches implemented in SystemVerilog
- Lots of developers familiar with UVM
- Open source UVM is necessary long term to combine commercial ecosystem with open source tools and methodologies
- Chip-making companies can benefit from open source while keeping their existing UVM codebase
- Open source tools will enable infinitely scalable, reproducible CIs



OPEN SOURCE UVM

- Ongoing work on extending Verilator with SystemVerilog features required by UVM:
 - Stratified scheduler
 - Randomize methods
 - Class support
- One of already reached milestones is development of [dynamic scheduling in Verilator](#)

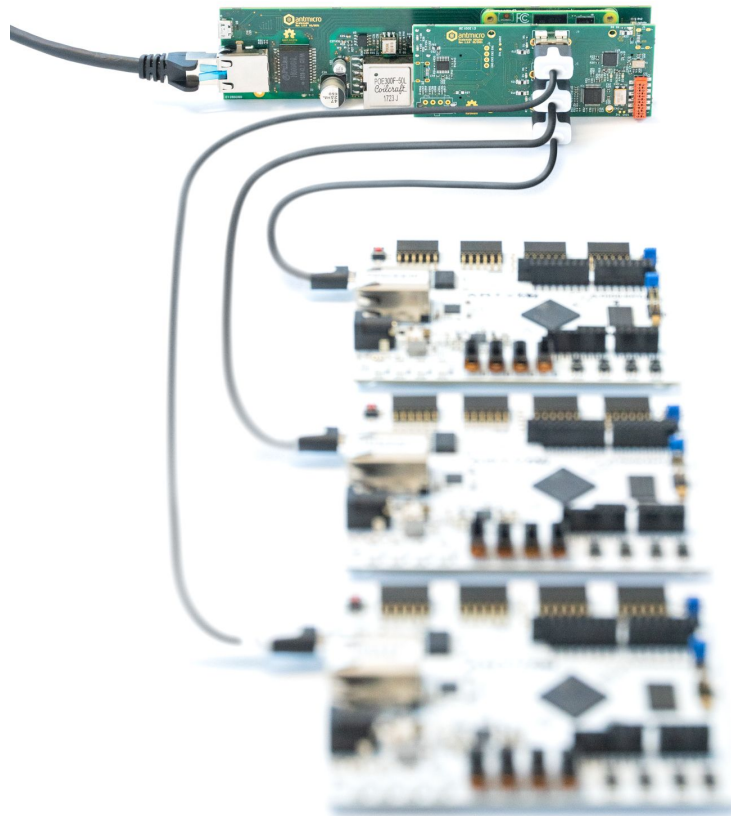


Western Digital.



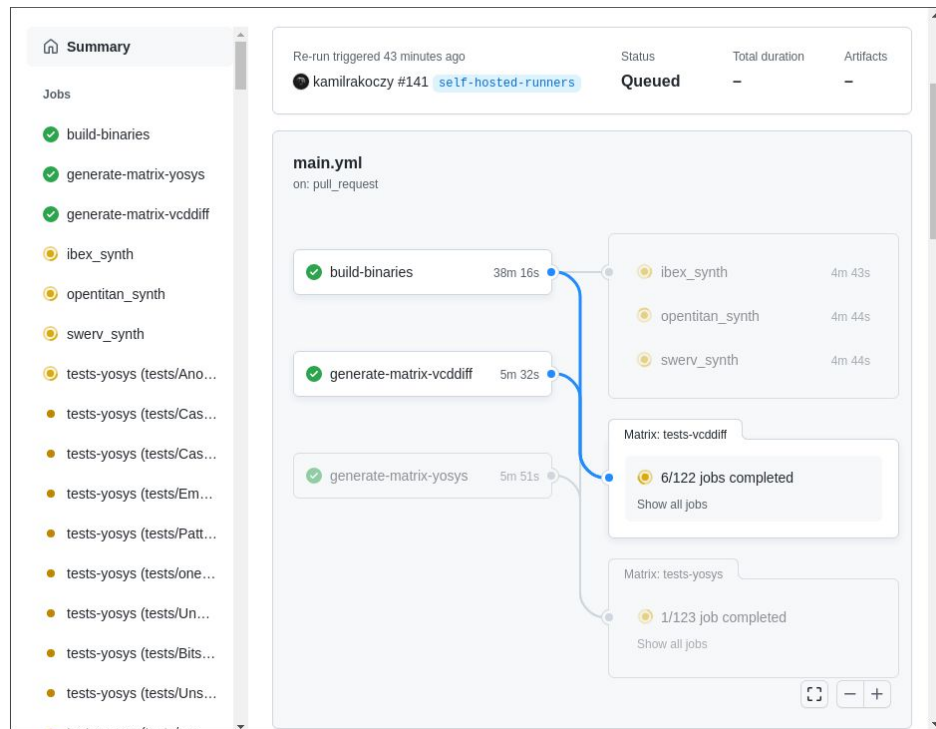
SCALE: GITHUB CUSTOM RUNNERS FOR CLOUD-ASSISTED CI

- Antmicro helps customers scale up ASIC development between teams and companies and into the cloud
- All the tools described here are meant to be used in a cloud context, on GitHub as well as private enterprise installations
- Mix and match open and closed components
- Use [custom, powerful runners](#) with scalable compute and custom peripherals e.g. FPGA boards for testing
- [Read more on our blog](#)



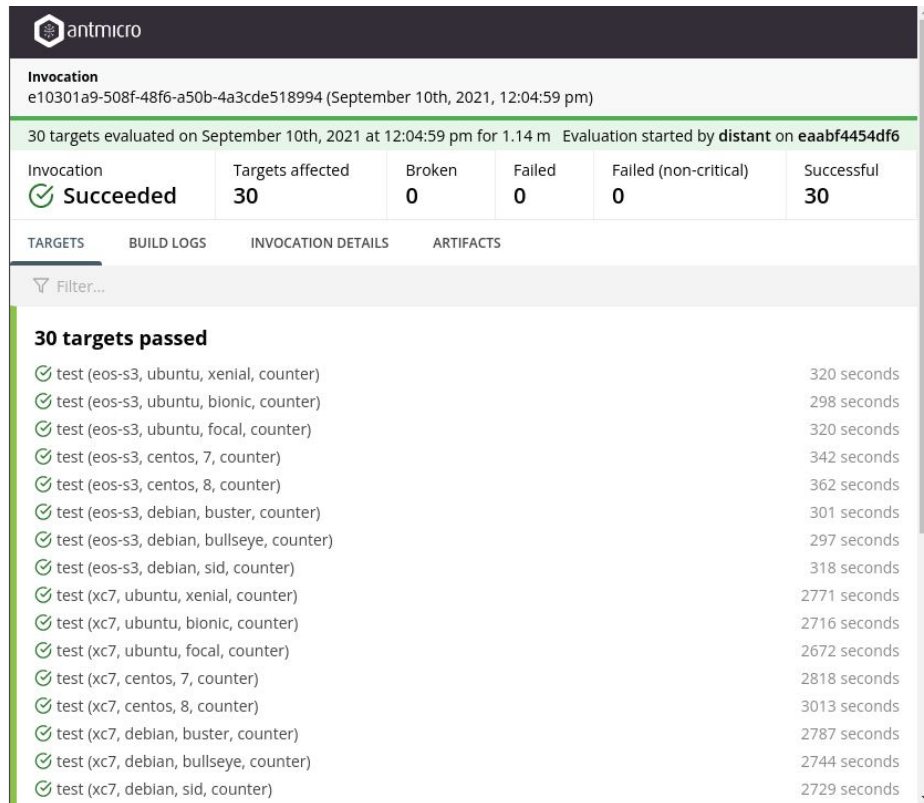
CUSTOM RUNNERS IN ACTION(S) WITH GITHUB

- Used for testing the UHDM integrations
- Gives us ability to do longer runs with more compute resources
- We can generate additional statistics of execution metrics, decrease resource usage and provide more insight
- Easier to find bottlenecks



CUSTOM RUNNERS IN ACTION - DISTANT-BES

- Run private builds, but potentially share (sanitized) build logs/results
- Upload results to own servers
- Store them for as long as needed
- Create more customized dashboards and detailed views of “what went wrong”



antmicro

Invocation
e10301a9-508f-48f6-a50b-4a3cde518994 (September 10th, 2021, 12:04:59 pm)

30 targets evaluated on September 10th, 2021 at 12:04:59 pm for 1.14 m Evaluation started by **distant** on **eaabf4454df6**

Invocation	Targets affected	Broken	Failed	Failed (non-critical)	Successful
✔ Succeeded	30	0	0	0	30

TARGETS BUILD LOGS INVOCATION DETAILS ARTIFACTS

Filter...

30 targets passed

✔ test (eos-s3, ubuntu, xenial, counter)	320 seconds
✔ test (eos-s3, ubuntu, bionic, counter)	298 seconds
✔ test (eos-s3, ubuntu, focal, counter)	320 seconds
✔ test (eos-s3, centos, 7, counter)	342 seconds
✔ test (eos-s3, centos, 8, counter)	362 seconds
✔ test (eos-s3, debian, buster, counter)	301 seconds
✔ test (eos-s3, debian, bullseye, counter)	297 seconds
✔ test (eos-s3, debian, sid, counter)	318 seconds
✔ test (xc7, ubuntu, xenial, counter)	2771 seconds
✔ test (xc7, ubuntu, bionic, counter)	2716 seconds
✔ test (xc7, ubuntu, focal, counter)	2672 seconds
✔ test (xc7, centos, 7, counter)	2818 seconds
✔ test (xc7, centos, 8, counter)	3013 seconds
✔ test (xc7, debian, buster, counter)	2787 seconds
✔ test (xc7, debian, bullseye, counter)	2744 seconds
✔ test (xc7, debian, sid, counter)	2729 seconds

SUMMARY

- CHIPS Alliance is building an open source SystemVerilog tooling ecosystem - join us!
- **Practical use cases are possible already now, both in local and collaborative / cloud development**
- More features and use cases are actively being worked on
- We want to hear about your needs and projects!
- Join our mailing lists:
sv-wg@lists.chipsalliance.org





**WANT TO USE OPEN SOURCE
SYSTEMVERILOG TOOLS?**

reach out to us:

contact@antmicro.com

