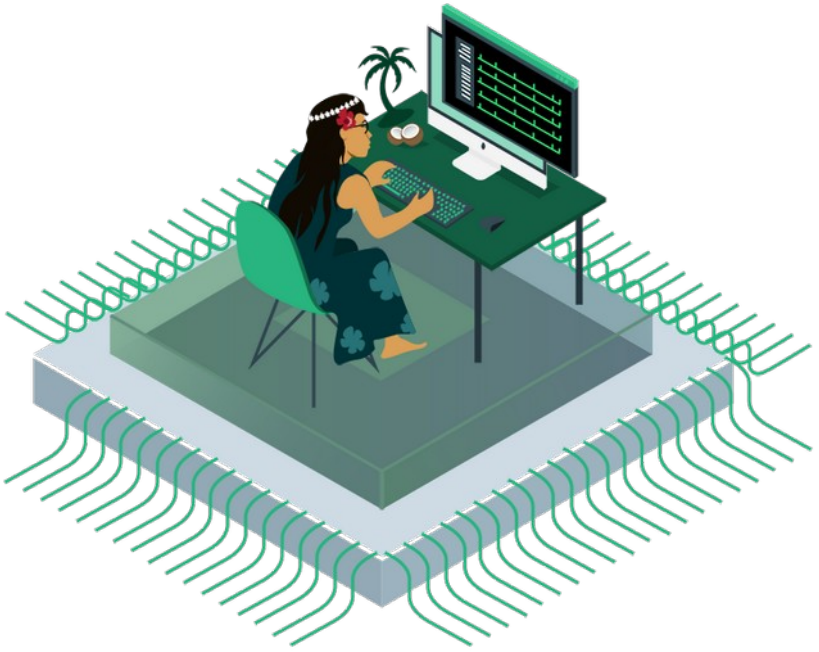
cocotb

How cocotb made Python-powered verification mainstream



And what comes next

Philipp Wagner
ORConf, 2023-09-15, Munich

cocotb makes verification fun.

It's hard work to make that happen.

About Philipp

lowRISC

opentitan

The FOSSi Foundation.

"IBM Z marca el inicio de la nueva era de la protección de datos" by IBM España. is marked with Public Domain Mark 1.0.

cocotb

open source initiative

How to contact me

mail@philipp-wagner.com
@imphil on GitHub
@MrImphil on X/Twitter

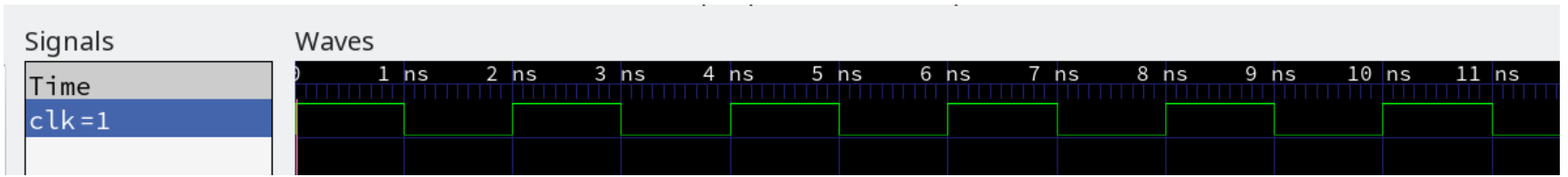
Introducing cocotb

test_fifo.py: Our first “test”

```python
@cocotb.test()
async def test_create_clock(dut):
    clock = dut.clk
    for _ in range(10):
        clock.value = 1
        await Timer(1, units='ns')
        clock.value = 0
        await Timer(1, units='ns')
```

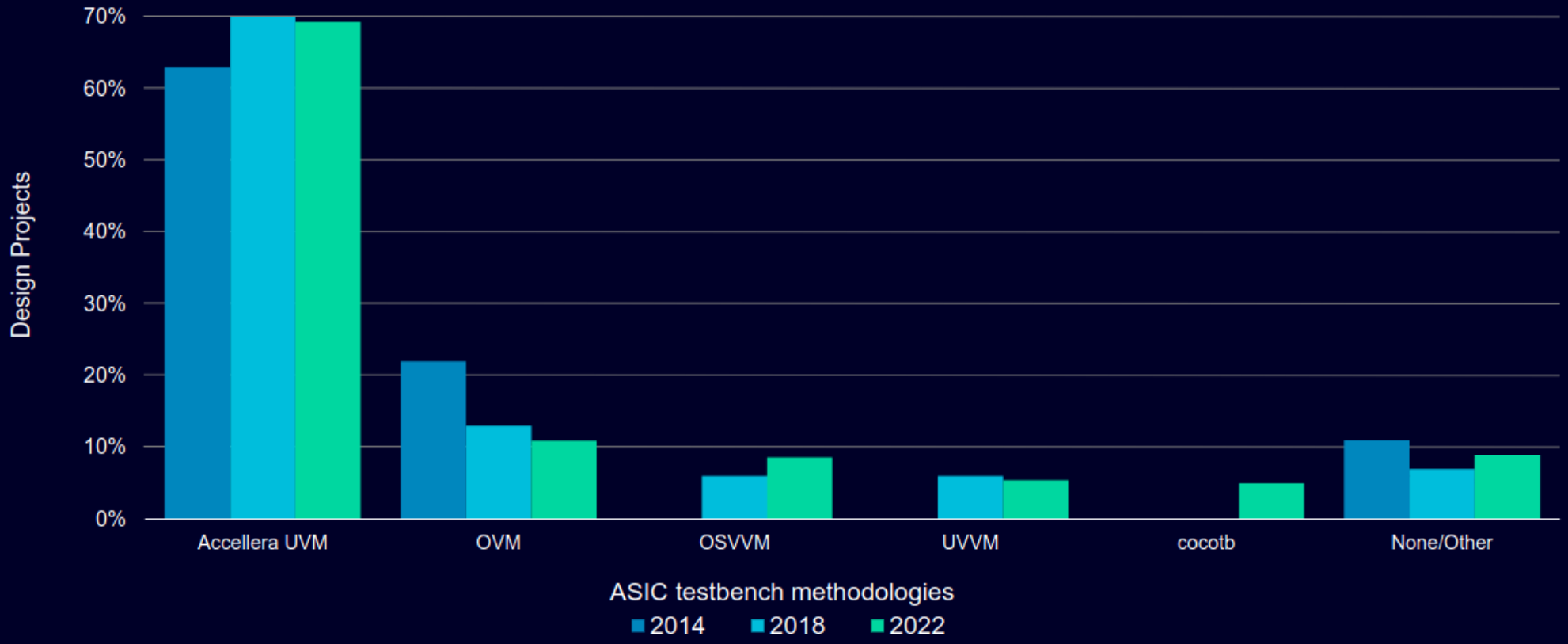
```
❯ SIM=icarus make clean sim WAVES=1
make -f Makefile results.xml
make[1]: Entering directory '/home/philipp/src/cocotb-tutorial/demo_fifo'
mkdir -p sim_build
/home/philipp/swinstall/iverilog/bin/iverilog -o sim_build/sim.vvp -D COCOTB_SIM=1 -s fifo -f sim_build/cmds.f -g2012   /home/phi
MODULE=test_fifo TESTCASE= TOPLEVEL=fifo TOPLEVEL_LANG=verilog \
         /home/philipp/swinstall/iverilog/bin/vvp -M /home/philipp/src/cocotb-tutorial/.direnv/python-3.8.12/lib64/python3.8/site
.vvp
     -.--ns INFO     cocotb.gpi                         ..mbed/gpi_embed.cpp:109  in set_program_name_in_venv        Using Python
-tutorial/.direnv/python-3.8.12/bin/python
     -.--ns INFO     cocotb.gpi                         ../gpi/GpiCommon.cpp:99   in gpi_print_registered_impl       VPI register
     0.00ns INFO     Running on Icarus Verilog version 11.0 (stable)
     0.00ns INFO     Running tests with cocotb v1.6.0 from /home/philipp/src/cocotb-tutorial/.direnv/python-3.8.12/lib64/python3.
     0.00ns INFO     Seeding Python random module with 1638222964
     0.00ns INFO     Found test test_fifo.test_create_clock
     0.00ns INFO     Found test test_fifo.test_fifo_manual
     0.00ns INFO     running test_create_clock (1/2)
VCD info: dumpfile dump.vcd opened for output.
    20.00ns INFO     test_create_clock passed
    20.00ns INFO     running test_fifo_manual (2/2)
    24.00ns INFO     test_fifo_manual failed
                     Traceback (most recent call last):
                       File "/home/philipp/src/cocotb-tutorial/demo_fifo/test_fifo.py", line 61, in test_fifo_manual
                         assert dut.dout.value.integer == expected_output
                     AssertionError: assert 101 == 100
                      +  where 101 = 01100101.integer
                      +    where 01100101 = ModifiableObject(fifo.dout).value
                      +      where ModifiableObject(fifo.dout) = HierarchyObject(fifo).dout
    24.00ns INFO     ****************************************************************************************
                     ** TEST                          STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
                     ****************************************************************************************
                     ** test_fifo.test_create_clock    PASS          20.00           0.00       8490.06  **
                     ** test_fifo.test_fifo_manual     FAIL           4.00           0.00       2185.93  **
                     ****************************************************************************************
                     ** TESTS=2 PASS=1 FAIL=1 SKIP=0                24.00           0.01       1776.96  **
                     ****************************************************************************************

make[1]: Leaving directory '/home/philipp/src/cocotb-tutorial/demo_fifo'
```

Waves in GTKWave



cocotb is successful

ASIC testbench methodologies



Source: Wilson Research Group and Siemens EDA, 2022 Functional Verification Study

* Multiple replies possible

SIEMENS

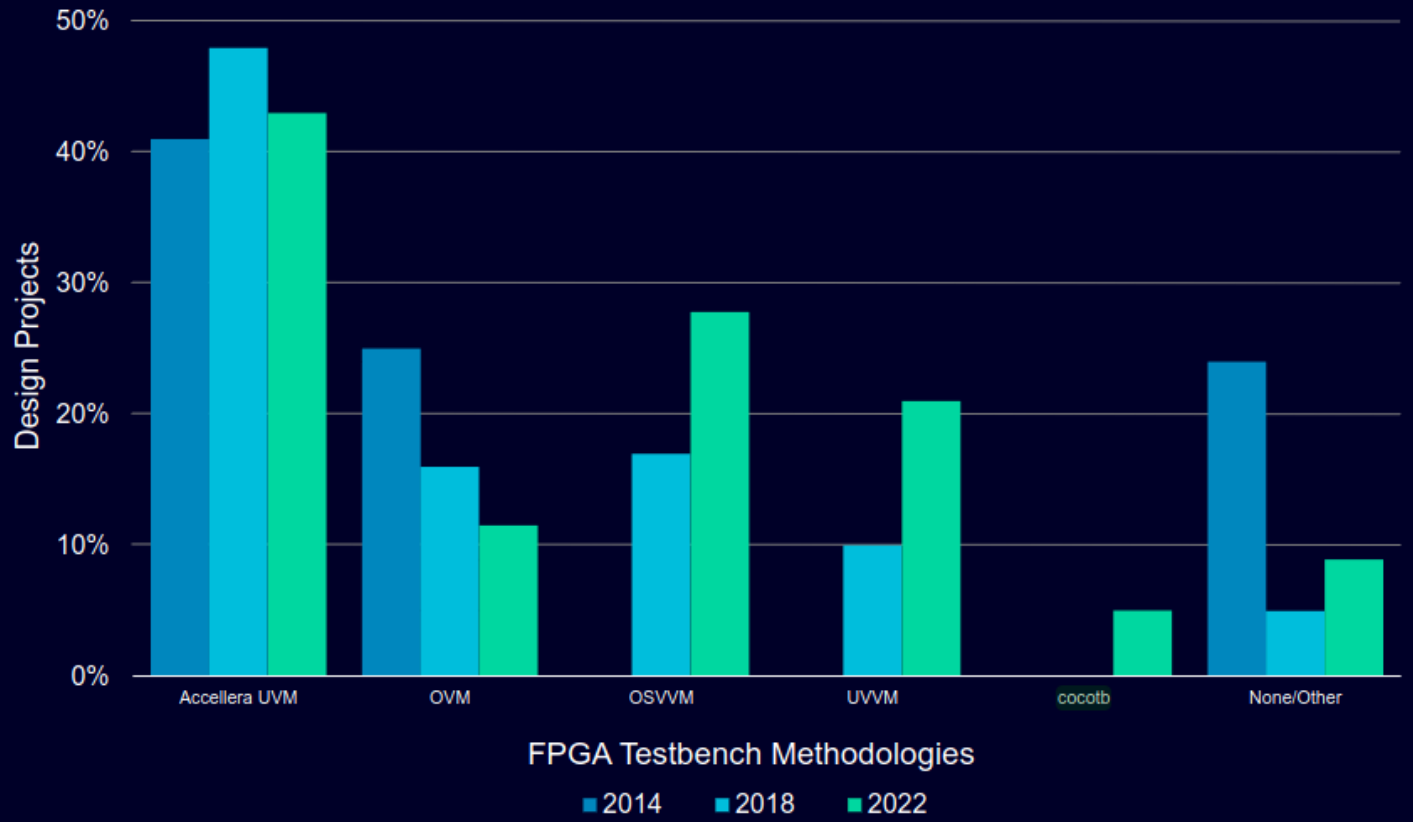
FPGA testbench methodologies

70%

UVM users are doing constrained-random

35%

OSVVM and UVM user are doing constrained-random



Source: Wilson Research Group and Siemens EDA, 2022 Functional Verification Study

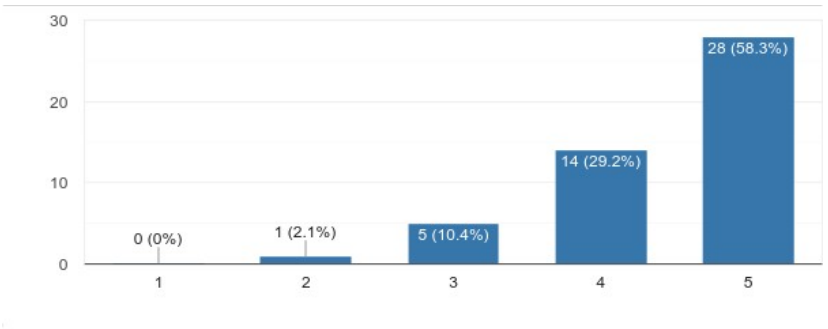
* Multiple replies possible

SIEMENS

Users love cocotb

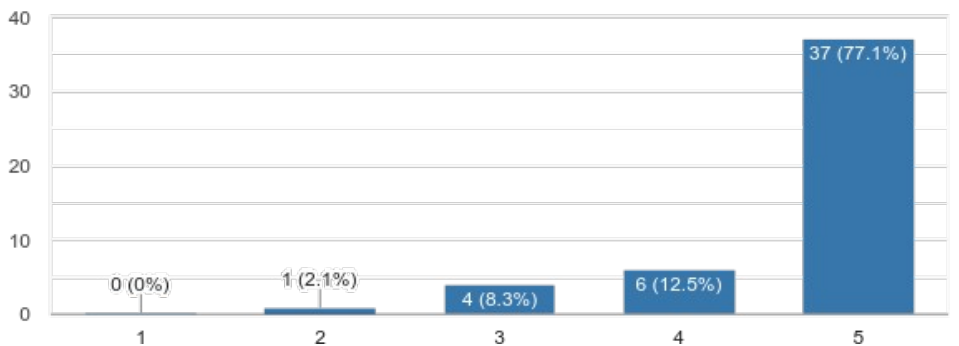
How much do you enjoy using cocotb?

48 responses



How likely are you to recommend cocotb to others?

48 responses



Why Python for verification?

- **Productive** to write, easy to read
- **Easy to interface** with
- Huge existing **ecosystem**
- **Popular language**: easy to find engineers

Year	Winner
2021	Python
2020	Python
2019	C
2018	Python
2017	C
2016	Go
2015	Java
2014	JavaScript
2013	Transact-SQL
2012	Objective-C
2011	Objective-C
2010	Python
2009	Go
2008	C
2007	Python
2006	Ruby
2005	Java
2004	PHP
2003	C++

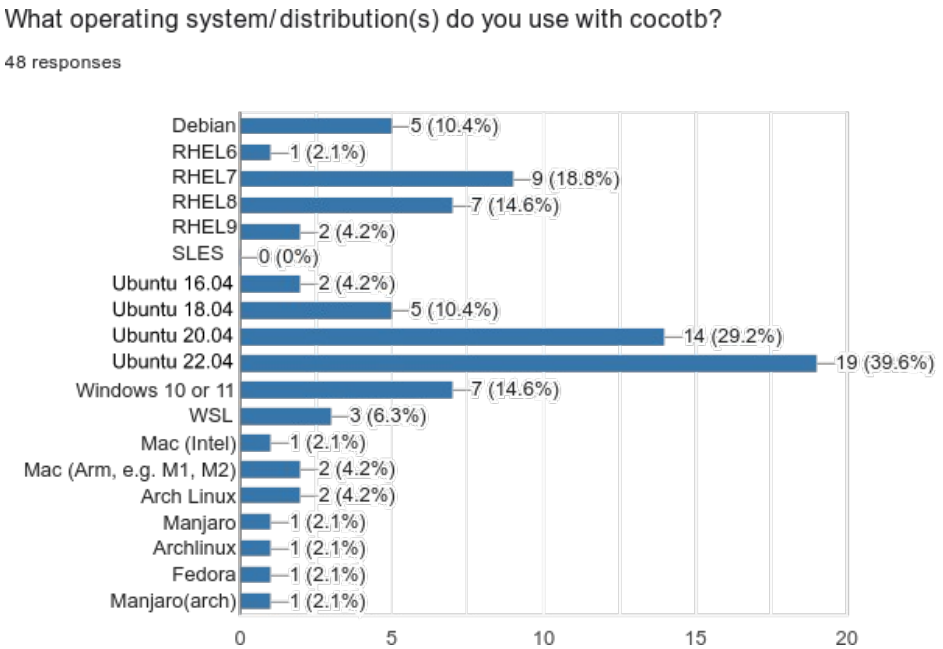
TIOBE Index Programming Language Hall of Fame (https://www.tiobe.com/tiobe-index/)

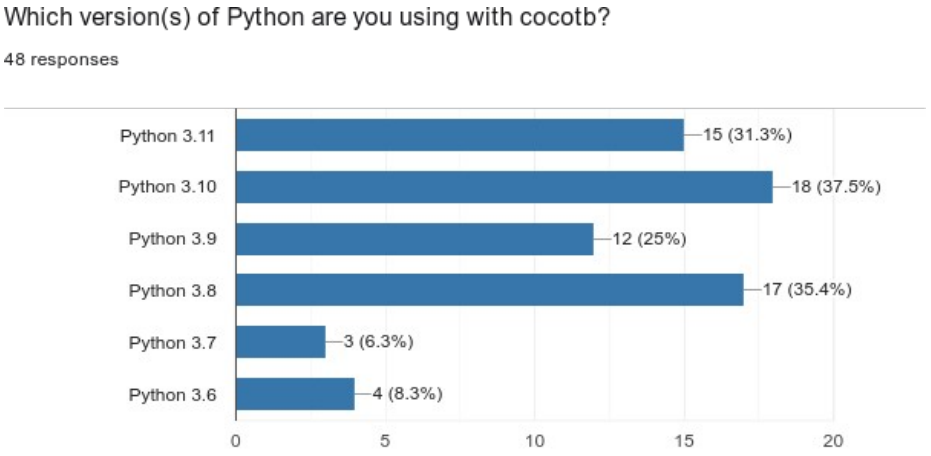
Verification is software.

With cocotb, you write verification code like software.
In Python.

Dealing with success

Serving a diverse user base





Strengthen your foundations.

Onboarding: Dead-simple installation

- From: Users compile on demand
- over: compile at installation
- to: users compile not at all (250 binaries!)

Manual testing to continuous integration

- cocotb supports 11 simulators, most of which are proprietary
- Testing!

Be explicit about what you want to support

Deprecations and Removals

- Setting values on indexed handles using the `handle[i] = value` syntax is deprecated. Instead use the `handle[i].value = value` syntax. (#2490)
- Setting values on handles using the `dut.handle = value` syntax is deprecated. Instead use the `handle.value = value` syntax. (#2490)
- Setting values on handles using the `signal <= newval` syntax is deprecated. Instead, use the `signal.value = newval` syntax. (#2681)
- `cocotb.utils.hexdump()` is deprecated; use `scapy.utils.hexdump()` instead. (#2691)
- `cocotb.utils.hexdiffs()` is deprecated; use `scapy.utils.hexdiff()` instead. (#2691)
- Passing `None` to `cocotb.utils.get_sim_time()` is deprecated; use `'step'` as the time unit instead. (#2691)
- The `stdout` and `stderr` attributes on `cocotb.result.TestComplete` and subclasses are deprecated. (#2692)
- `cocotb.result.TestFailure` is deprecated, use an `assert` statement instead. (#2692)

Going forward

Design access

```
diff --git a/cocotb/handle.py b/cocotb/handle.py
index 740c20f3..3c93cf0d 100755
--- a/cocotb/handle.py
+++ b/cocotb/handle.py
@@ -903,7 +903,7 @@ class ModifiableObject(NonConstantObject):
         return value._as_gpi_args_for(self)

     @NonConstantObject.value.getter
-    def value(self) -> BinaryValue:
+    def value(self) -> LogicArray:
         binstr = self._handle.get_signal_val_binstr()
         # Skip BinaryValue.assign() as we know we are using a binstr
         result = BinaryValue(n_bits=len(binstr))
```

Design access

- Replace `BinaryValue` with `LogicArray`
- Make more type information available at coding time

Scheduler + coroutines

- Align with other Python async frameworks
  `fork()` → `start_soon()` or `start()`
- Error propagation in coroutines

Build and run your simulation

- **Makefiles**
- cocotb-test

Makefile

```
PWD             := $(shell pwd)

VERILOG_SOURCES := $(PWD)/fifo.sv
TOPLEVEL_LANG   := verilog

TOPLEVEL        := fifo
MODULE          := test_fifo

include $(shell cocotb-config --makefiles)/Makefile.inc
```

Tip: Find a list of all supported make variables at https://docs.cocotb.org/en/stable/building.html#make-system.

There’s more than one way to run a simulation

Python test runner

The build description

```python
def test_simple_dff_runner():
    proj_path = Path(__file__).resolve().parent

    verilog_sources = [proj_path / "dff.sv"]

    runner = get_runner("icarus")
    runner.build(
        verilog_sources=verilog_sources,
        hdl_toplevel="dff",
        always=True,
    )

    runner.test(hdl_toplevel="dff", test_module="test_dff,")
```

Run

```
SIM=questa HDL_TOPLEVEL_LANG=vhdl pytest examples/simple_dff/test_dff.py
```

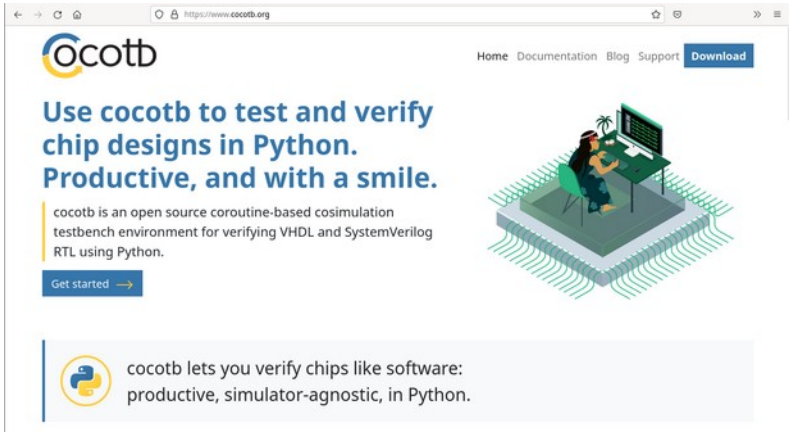
Let’s shape the future together

cocotb unconference

- Sunday, 9:00 am – 1:00 pm @ KHG TUM
- It’s an unconference!
- Bring your discussion topics.

Remember.

- cocotb is productive, and fun.
- Appreciate maintainers.
- Discuss *your* ideas tomorrow!



www.cocotb.org
https://twitter.com/cocotbnews

Our sponsors

ALDEC THE DESIGN VERIFICATION COMPANY

Technology Partner Digital Industries Software SIEMENS

cadence

The FOSSi Foundation.

Be a sponsor! Talk to philipp@fossi-foundation.org