



Table of contents



DSLX

Convert to Hardware IR

Generate RTL



Run the OpenLane flow

Configuration

Synthesis

Preview

Floorplan

Preview

Placement

Preview

Routing

Preview

Sign off

Preview

+ Code

+ Text

Copy to Drive

```
[ ] %%bash -c "cat > test.py; make"
import cocotb
from cocotb.triggers import ClockCycles, RisingEdge
from cocotb.clock import Clock

@cocotb.test()
async def test_test(dut):
    c = Clock(dut.clk, 1, 'ns')
    cocotb.start_soon(c.start())

    # TODO(YOU): update with the 8-input to user_module
    dut.user_module_tb.io_in.value = NaN
    # pipeline stage 0: fetch
    await ClockCycles(dut.clk, 2)
    # TODO(YOU): verify that the input value propagate to stage #0
    assert dut.user_module_tb.p0_io_in == NaN

    # pipeline stage 1: slice
    await ClockCycles(dut.clk, 2)
    # TODO(YOU): update `p1_bit_slice_*` names to match the wire in the generator
    # TODO(YOU): verify that the input value is sliced into 4-bit umul operation
    assert dut.user_module_tb.p1_bit_slice_TODO == NaN
    assert dut.user_module_tb.p1_bit_slice_TODO == NaN

    # pipeline stage 2: multiply
    await ClockCycles(dut.clk, 2)
    # TODO(YOU): verify that the output value match the multiplication result
    assert dut.user_module_tb.out == NaN
```