

# tb\_cocotb.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/01/27

---

## INFORMATION

---

### Brief

---

Test bench wrapper for cocotb

### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## tb\_cocotb

---

```
module tb_cocotb #(
  parameter
    CLOCK_SPEED
    =
    20000000,
  parameter
    DELAY
    =
    0
) ( input clk, input rstn, input start0, input clr, input hold, input [31:0]
```

Mod rate enable generator test bench

Parameters

<b>CLOCK_SPEED</b> parameter	This is the aclk frequency in Hz
<b>DELAY</b> parameter	Delay the enable by a number of clock ticks

Ports

<b>clk</b>	Clock used for enable generation
<b>rstn</b>	Negative reset for anything clocked on clk
<b>start0</b>	Start counter at rate if set. Otherwise set to $CLOCK\_SPEED/2+rate$ (midpoint).
<b>clr</b>	Clear counter to initial values.
<b>hold</b>	hold enable low and pause + reset count till hold removed (low).
<b>rate</b>	rate that enable pulse will be generated, must be less then the clock rate.
<b>ena</b>	positive enable that is pulsed high at enable rate.

INSTANTIATED MODULES

---

dut

```
mod_clock_ena_gen #(
    CLOCK_SPEED(CLOCK_SPEED),
    DELAY(DELAY)
) dut ( .clk(clk), .rstn(rstn), .start0(start0), .clr(clr), .hold(hold), .rate(rate) )
```

Device under test, mod\_clock\_ena\_gen