tb_mod_ena.v

AUTHORS

JAY CONVERTINO

DATES

2025/01/27

INFORMATION

Brief

Test bench for mod clock divide enable generator

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_mod_ena

```
module tb_mod_ena #(
parameter
CLOCK_SPEED
 =
2000000,
parameter
START_AT_ZERO
 =
0,
parameter
ENABLE_RATE
 =
1000,
parameter
```

```
 DELAY
  =
 0
)()
```

mod clock enable test bench

Parameters

CLOCK_SPEED parameter	Clock speed
START_AT_ZERO parameter	Set to 1 to enable start at zero.
DELAY parameter	Set to the number of clock cycles to delay the enable output signal.

INSTANTIATED MODULES

clk_stim

```
clk_stimulus #(
                                                                                .
CLOCKS(1),
                                                                                .
CLOCK_BASE(CLOCK_SPEED),
                                                                                .
CLOCK_INC(1000),
                                                                                .
RESETS(1),
                                                                                .
RESET_BASE(2000),
                                                                                .
RESET_INC(100)
) clk_stim ( .clkv(tb_dut_clk), .rstnv(tb_dut_rstn), .rstv() )
```

Generate a 50/50 duty cycle set of clocks and reset.

dut

```
mod_clock_ena_gen #(
                                                                                .
CLOCK_SPEED(CLOCK_SPEED),
                                                                                .
START_AT_ZERO(START_AT_ZERO),
                                                                                .
DELAY(DELAY)
) dut ( .clk(tb_dut_clk), .rstn(tb_dut_rstn), .hold(1'b0), .rate(ENABLE_RATE
```

Device under test, mod_clock_ena_gen