

# tm\_stim\_axis.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2022/10/24

---

## INFORMATION

---

### Brief

---

All modules for AXIS test bench top are here. There will be loop of tests the axis core must pass. In these tests is where the end user must alter the checks if the input does not equal the output. As these were designed with a FIFO in mind.

### License MIT

---

Copyright 2022 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.

## slave\_axis\_stimulus

---

```
module slave_axis_stimulus #(
  parameter
    BUS_WIDTH
    =
    1,
  parameter
    USER_WIDTH
    =
    1,
```

```

parameter
DEST_WIDTH
=
1,
parameter
BYTE_SWAP
=
0,
parameter
FILE
=
"test.bin"
) ( input m_axis_aclk, input m_axis_arstn, output reg m_axis_tvalid, input

```

Simulator core to read file data, and output it over master axis dut.

## Parameters

<b>BUS_WIDTH</b> parameter	bus width in bytes for data bus
<b>USER_WIDTH</b> parameter	user width in bits
<b>DEST_WIDTH</b> parameter	dest width in bits
<b>BYTE_SWAP</b> parameter	swap bytes fed to the DUT
<b>FILE</b> parameter	input file name

## Ports

<b>m_axis_aclk</b>	master axis clock
<b>m_axis_arstn</b>	master axis negative reset
<b>m_axis_tvalid</b>	master axis data valid active high
<b>m_axis_tready</b>	master axis, is the input device ready?
<b>m_axis_tdata</b>	master axis data input.
<b>m_axis_tkeep</b>	master axis byte indicator
<b>m_axis_tlast</b>	master axis data is last word
<b>m_axis_tuser</b>	master axis user defined
<b>m_axis_tdest</b>	master axis desitination
<b>eof</b>	end of input file has been reached.

## master\_axis\_stimulus

```

module master_axis_stimulus #(
parameter
BUS_WIDTH
=
1,
parameter
USER_WIDTH
=
1,

```

```

parameter
DEST_WIDTH
=
1,
parameter
RAND_READY
=
0,
parameter
FILE
=
"out.bin"
) ( input s_axis_aclk, input s_axis_arstn, input s_axis_tvalid, output reg

```

Simulator core to write file data, from input over slave axis dut. This module will keep a constant ready to the dut.

## Parameters

<b>BUS_WIDTH</b> parameter	bus width in bytes for data bus
<b>USER_WIDTH</b> parameter	user width in bits
<b>DEST_WIDTH</b> parameter	dest width in bits
<b>RAND_READY</b> parameter	random ready if set anything other than 0
<b>FILE</b> parameter	output file name

## Ports

<b>s_axis_aclk</b>	slave axis clock
<b>s_axis_arstn</b>	slave axis negative reset
<b>s_axis_tvalid</b>	slave data valid
<b>s_axis_tready</b>	slave ready
<b>s_axis_tdata</b>	slave data
<b>s_axis_tkeep</b>	slave keep
<b>s_axis_tlast</b>	slave last word of data
<b>s_axis_tuser</b>	slave user port
<b>s_axis_tdest</b>	slave destination port
<b>eof</b>	end of file will trigger \$finish to end sim