

# axis\_fifo.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2021/06/29

---

## INFORMATION

---

### Brief

---

Wraps the standard FIFO with an axi streaming interface.

### License MIT

---

Copyright 2021 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.

## axis\_fifo

---

```
module axis_fifo #(
    parameter
    FIFO_DEPTH
    =
    256,
    parameter
    COUNT_WIDTH
    =
    8,
    parameter
    BUS_WIDTH
```

```

    =
    1,
    parameter
    USER_WIDTH
    =
    1,
    parameter
    DEST_WIDTH
    =
    1,
    parameter
    RAM_TYPE
    =
    "block",
    parameter
    PACKET_MODE
    =
    0,
    parameter
    COUNT_DELAY
    =
    1,
    parameter
    COUNT_ENA
    =
    1
) ( input m_axis_aclk, input m_axis_arstn, output m_axis_tvalid, input m_axi

```

AXIS fifo

## Parameters

<b>FIFO_DEPTH</b> parameter	Depth of the fifo, must be a power of two number(divisable aka $256 = 2^8$ ). Any non-power of two will be rounded up to the next closest.
<b>COUNT_WIDTH</b> parameter	Data count output width in bits. Should be the same power of two as fifo depth(256 for fifo depth... this should be 8).
<b>BUS_WIDTH</b> parameter	Width of the axis data bus input/output in bytes.
<b>USER_WIDTH</b> parameter	Width of the axis user bus input/output in bits.
<b>DEST_WIDTH</b> parameter	Width of the axis dest bus input/output in bits.
<b>RAM_TYPE</b> parameter	RAM type setting.
<b>PACKET_MODE</b> parameter	Set axis fifo to wait for tlast before allowing a read on master port output.
<b>COUNT_DELAY</b> parameter	Delay count by one clock cycle of the data count clock.
<b>COUNT_ENA</b> parameter	Enable count, set this to 0 to disable (only disable if read/write/data_count are on the same clock domain!).

## Ports

<b>m_axis_aclk</b>	Clock for AXIS
<b>m_axis_arstn</b>	Negative reset for AXIS
<b>m_axis_tvalid</b>	When active high the output data is valid
<b>m_axis_tready</b>	When set active high the output device is ready for data.

<b>m_axis_tdata</b>	Output data
<b>m_axis_tkeep</b>	Output valid byte indicator
<b>m_axis_tlast</b>	Indicates last word in stream.
<b>m_axis_tuser</b>	Output user bus
<b>m_axis_tdest</b>	Output destination
<b>s_axis_aclk</b>	Clock for AXIS
<b>s_axis_arstn</b>	Negative reset for AXIS
<b>s_axis_tvalid</b>	When set active high the input data is valid
<b>s_axis_tready</b>	When active high the device is ready for input data.
<b>s_axis_tdata</b>	Input data
<b>s_axis_tkeep</b>	Input valid byte indicator
<b>s_axis_tlast</b>	Is this the last word in the stream (active high).
<b>s_axis_tuser</b>	Input user bus
<b>s_axis_tdest</b>	Input destination
<b>data_count_aclk</b>	Clock for data count
<b>data_count_arstn</b>	Negative edge reset for data count.
<b>data_count</b>	Output that indicates the amount of data in the FIFO.

## INSTANTIATED MODULES

---

### axis\_fifo

---

```

fifo #(
    FIFO_DEPTH
    c_FIFO_DEPTH),
    BYTE_WIDTH
    c_FIFO_WIDTH),
    COUNT_WIDTH
    COUNT_WIDTH),
    FWFT
    1),
    RD_SYNC_DEPTH
    0),
    WR_SYNC_DEPTH
    0),
    DC_SYNC_DEPTH
    0),

```

```

COUNT_DELAY
COUNT_DELAY),
COUNT_ENA
COUNT_ENA),
DATA_ZERO
1),
ACK_ENA
0),
RAM_TYPE
RAM_TYPE)
) axis_fifo ( .rd_clk (m_axis_aclk), .rd_rstn (m_axis_arstn), .rd_en (s_rd_en),

```

Generic FIFO that acts like a Xilinx FIFO.

## axis\_control

```

axis_fifo_ctrl #(
BUS_WIDTH
BUS_WIDTH),
FIFO_WIDTH
c_FIFO_WIDTH),
USER_WIDTH
USER_WIDTH),
DEST_WIDTH
DEST_WIDTH),
PACKET_MODE(PACKET_MODE)
) axis_control ( .m_axis_aclk (m_axis_aclk), .m_axis_arstn (m_axis_arstn),

```

Create signals to control FIFO and provide AXIS interace.