

# axis\_data\_to\_axis\_string.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2022/09/19

---

## INFORMATION

---

### Brief

---

Parse raw binary data into ASCII string output.

### 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.

## axis\_data\_to\_axis\_string

---

```
module axis_data_to_axis_string #(
    parameter
    DELIMITER
    =
    " , "
    parameter
    TERMINATION
    =
    "\n"
    parameter
    SBUS_WIDTH
```

```

    =
    1,
    parameter
    USER_WIDTH
    =
    4,
    parameter
    DEST_WIDTH
    =
    4,
    parameter
    PREFIX_LEN
    =
    1,
    parameter
    DATA_PREFIX
    =
    "#",
    parameter
    DEST_PREFIX
    =
    "&",
    parameter
    USER_PREFIX
    =
    ""
) ( input aclk, input arstn, input [(SBUS_WIDTH*8)-1:0] s_axis_tdata, input

```

Parse raw binary data into ASCII string output.

## Parameters

<b>DELIMITER</b> parameter	break value between multiple strings
<b>TERMINATION</b> parameter	termination value of full string from serial port, byte only. (\n = 0A \r = 0D).
<b>SBUS_WIDTH</b> parameter	bus width of slave (data) input
<b>USER_WIDTH</b> parameter	user width of slave bus, only in 4 bit nibbles, and at least 4 bits.
<b>DEST_WIDTH</b> parameter	dest width of slave bus, only in 4 bit nibbles, and at least 4 bits.
<b>PREFIX_LEN</b> parameter	length of following prefix strings.
<b>DATA_PREFIX</b> parameter	prefix for data hex strings
<b>DEST_PREFIX</b> parameter	prefix for destination hex strings
<b>USER_PREFIX</b> parameter	prefix for user hex strings

## Ports

<b>aclk</b>	Clock for AXIS
<b>arstn</b>	Negative reset for AXIS
<b>s_axis_tdata</b>	Input data
<b>s_axis_tvalid</b>	When set active high the input data is valid

<b>s_axis_tuser</b>	User data to convert.
<b>s_axis_tdest</b>	Destination data to convert
<b>s_axis_tready</b>	When active high the device is ready for input data.
<b>m_axis_tdata</b>	Output data
<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.

## VARIABLES

---

### s\_axis\_tready

---

```
assign s_axis_tready = (
    arstn                                     (counter == 0) ? 1 &
    :
    0
)
```

ready if count is zero, this is a FWFT so no worries in pumping out data.

### m\_axis\_tdata

---

```
assign m_axis_tdata = char_buffer[STRING_LEN*8-1 -:8]
```

output whatever is in the character buffer.

### m\_axis\_tvalid

---

```
assign m_axis_tvalid = (
    1                                     counter > 0 ?
    :
    0
)
```

Counter greater than 0? Valid output is available.