axis_fifo.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/29

INFORMATION

Brief

Wraps the standard FIFO with an axi streaming interface.

License MIT

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_axis_tready,
 output
  [(BUS_WIDTH*8)-1:0]
 m_axis_tdata,
 output
  [BUS_WIDTH-1:0]
 m_axis_tkeep,
 output
 m_axis_tlast,
 output
  [USER_WIDTH-1:0]
 m_axis_tuser,
 output
  [DEST_WIDTH-1:0]
 m_axis_tdest,
 input
 s_axis_aclk,
 input
 s_axis_arstn,
 input
 s_axis_tvalid,
 output
 s_axis_tready,
 input
  [(BUS_WIDTH*8)-1:0]
 s_axis_tdata,
 input
  [BUS_WIDTH-1:0]
 s_axis_tkeep,
 input
 s_axis_tlast,
 input
  [USER_WIDTH-1:0]
 s_axis_tuser,
```

```
 input
  [DEST_WIDTH-1:0]
 s_axis_tdest,
 input
 data_count_aclk,
 input
 data_count_arstn,
 output
  [COUNT_WIDTH:0]
 data_count
)
```

AXIS fifo

Parameters

FIFO_DEPTH 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.
COUNT_WIDTH 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).
BUS_WIDTH parameter	Width of the axis data bus input/output in bytes.
USER_WIDTH parameter	Width of the axis user bus input/output in bits.
DEST_WIDTH parameter	Width of the axis dest bus input/output in bits.
RAM_TYPE parameter	RAM type setting.
PACKET_MODE parameter	Set axis fifo to wait for tlast before allowing a read on master port output.
COUNT_DELAY parameter	Delay count by one clock cycle of the data count clock.
COUNT_ENA parameter	Enable count, set this to 0 to disable (only disable if read/write/data_count are on the same clock domain!).

Ports

m_axis_aclk input	Clock for AXIS
m_axis_arstn input	Negative reset for AXIS
m_axis_tvalid output	When active high the output data is valid
m_axis_tready input	When set active high the output device is ready for data.
m_axis_tdata output [(BUS_WIDTH* 8)- 1:0]	Output data
m_axis_tkeep output [BUS_WIDTH- 1:0]	Output valid byte indicator
m_axis_tlast output [BUS_WIDTH- 1:0]	Indicates last word in stream.
m_axis_tuser output [USER_WIDTH- 1:0]	Output user bus
m_axis_tdest output [DEST_WIDTH- 1:0]	Output destination
s_axis_aclk input [DEST_WIDTH- 1:0]	Clock for AXIS

s_axis_arstn `input [DEST_WIDTH- 1:0]`	Negative reset for AXIS
s_axis_tvalid `input [DEST_WIDTH- 1:0]`	When set active high the input data is valid
s_axis_tready `output [DEST_WIDTH- 1:0]`	When active high the device is ready for input data.
s_axis_tdata `input [(BUS_WIDTH* 8)- 1:0]`	Input data
s_axis_tkeep `input [BUS_WIDTH- 1:0]`	Input valid byte indicator
s_axis_tlast `input [BUS_WIDTH- 1:0]`	Is this the last word in the stream (active high).
s_axis_tuser `input [USER_WIDTH- 1:0]`	Input user bus
s_axis_tdest `input [DEST_WIDTH- 1:0]`	Input desitination
data_count_aclk `input [DEST_WIDTH- 1:0]`	Clock for data count
data_count_arstn `input [DEST_WIDTH- 1:0]`	Negative edge reset for data count.
data_count `output [COUNT_WIDTH:0]`	Output that indicates the amount of data in the FIFO.

INSTANTIATED MODULES

axis_fifo

Generic FIFO that acts like a Xilinx FIFO.

axis_control

Create signals to control FIFO and provide AXIS interace.