

fifo_ctrl.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/29

INFORMATION

Brief

Control block for fifo operations, emulates xilinx fifo.

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.

fifo_ctrl

```
module fifo_ctrl #(
    parameter FIFO_DEPTH
        =
        256,
    parameter BYTE_WIDTH
        =
        1,
    parameter ADDR_WIDTH
        =
        1,
    parameter
```

```

COUNT_WIDTH
=
1,
parameter
GREY_CODE
=
1,
parameter
COUNT_DELAY
=
1,
parameter
COUNT_ENA
=
1,
parameter
ACK_ENA
=
0,
parameter
FWFT
=
0
)

(
input
rd_clk,
input
rd_rstn,
input
rd_en,
output
[ADDR_WIDTH-1:0]
rd_addr,
output
rd_valid,
output
rd_mem_en,
output
rd_empty,
input
wr_clk,
input
wr_rstn,
input
wr_en,
output
[ADDR_WIDTH-1:0]
wr_addr,
output
wr_ack,
output
wr_mem_en,
output
wr_full,
input
data_count_clk,
input
data_count_rstn,
output
[COUNT_WIDTH:0]
data_count
)

```

Control block for fifo operations, emulates xilinx fifo.

Parameters

FIFO_DEPTH parameter	Depth of the fifo, must be a power of two number(divisible aka 256 = 2^8). Any non-power of two will be rounded up to the next closest.
BYTE_WIDTH parameter	How many bytes wide the data in/out will be.
ADDR_WIDTH parameter	Width of the RAM address bus to write data to.
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).
GREY_CODE parameter	RAM address uses grey code instead of linear addressing.
COUNT_DELAY parameter	Delay count by one clock cycle of the data count clock. Set this to 0 to disable (only disable if read/write/data_count are on the same clock domain!).
COUNT_ENA parameter	Enable the count output.
ACK_ENA parameter	Enable ack on write.
FWFT parameter	1 for first word fall through mode. 0 for normal.

Ports

rd_clk input	Clock for read data
rd_rstn input	Negative edge reset for read.
rd_en input	Active high enable of read interface.
rd_addr output [ADDR_WIDTH- 1:0]	Address to read data from in RAM.
rd_valid output [ADDR_WIDTH- 1:0]	Active high output that the data is valid.
rd_mem_en output [ADDR_WIDTH- 1:0]	Active high enable to read from RAM.
rd_empty output [ADDR_WIDTH- 1:0]	Active high output when read is empty.
wr_clk input [ADDR_WIDTH- 1:0]	Clock for write data
wr_rstn input [ADDR_WIDTH- 1:0]	Negative edge reset for write
wr_en input [ADDR_WIDTH- 1:0]	Active high enable of write interface.
wr_addr output [ADDR_WIDTH- 1:0]	Address to write data to in RAM.
wr_ack output [ADDR_WIDTH- 1:0]	Active high when enabled, that data write has been done.
wr_mem_en output [ADDR_WIDTH- 1:0]	Active high enable to write to RAM.
wr_full output [ADDR_WIDTH- 1:0]	Active high output that the FIFO is full.
data_count_clk input [ADDR_WIDTH- 1:0]	Clock for data count
data_count_rstn input [ADDR_WIDTH- 1:0]	Negative edge reset for data count.

data_count Output that indicates the amount of data in the FIFO.
output [COUNT_WIDTH:0]