fifo.v

AUTHORS

JAY CONVERTINO

DATES

2021/06/29

INFORMATION

Brief

Wrapper to tie together fifo_ctrl, fifo_mem, and fifo_pipe. Emulates Xilinx FIFO core.

License MIT

fifo

```
module fifo #(
parameter
FIFO_DEPTH
 =
256,
parameter
BYTE_WIDTH
 =
1,
parameter
COUNT_WIDTH
 =
8,
parameter
```

```
FWFT
 =
0,
parameter
RD_SYNC_DEPTH
 =
0,
parameter
WR_SYNC_DEPTH
 =
0,
parameter
DC_SYNC_DEPTH
 =
0,
parameter
COUNT_DELAY
 =
1,
parameter
COUNT_ENA
 =
1,
parameter
DATA_ZERO
 =
0,
parameter
ACK_ENA
 =
0,
parameter
RAM_TYPE
 =
"block"
)
                                                                (
input
rd_clk,
input
rd_rstn,
input
rd_en,
output
rd_valid,
output
 [(BYTE_WIDTH*8)-1:0]
rd_data,
output
rd_empty,
input
wr_clk,
input
wr_rstn,
input
wr_en,
output
wr_ack,
input
 [(BYTE_WIDTH*8)-1:0]
wr_data,
output
wr_full,
input
data_count_clk,
input
```

```
 data_count_rstn,
 output
  [COUNT_WIDTH:0]
 data_count
)
```

Wrapper to tie together fifo_ctrl, fifo_mem, and fifo_pipe.

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.
BYTE_WIDTH parameter	How many bytes wide the data in/out will be.
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).
FWFT parameter	1 for first word fall through mode. 0 for normal.
RD_SYNC_DEPTH parameter	Add in pipelining to read path. Defaults to 0.
WR_SYNC_DEPTH parameter	Add in pipelining to write path. Defaults to 0.
DC_SYNC_DEPTH parameter	Add in pipelining to data count path. Defaults to 0.
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.
DATA_ZERO parameter	Zero out data output when enabled.
ACK_ENA parameter	Enable an ack when data is requested.
RAM_TYPE parameter	Set the RAM type of the fifo.

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_valid output	Active high output that the data is valid.
rd_data output [(BYTE_WIDTH* 8)- 1:0]	Output data
rd_empty output [(BYTE_WIDTH* 8)- 1:0]	Active high output when read is empty.
wr_clk input [(BYTE_WIDTH* 8)- 1:0]	Clock for write data
wr_rstn input [(BYTE_WIDTH* 8)- 1:0]	Negative edge reset for write
wr_en input [(BYTE_WIDTH* 8)- 1:0]	Active high enable of write interface.
wr_ack	Active high when enabled, that data write has been done.

output [(BYTE_WIDTH* 8)- 1:0]

wr_data input [(BYTE_WIDTH* 8)- 1:0]	Input data
wr_full output [(BYTE_WIDTH* 8)- 1:0]	Active high output that the FIFO is full.
data_count_clk input [(BYTE_WIDTH* 8)- 1:0]	Clock for data count
data_count_rstn input [(BYTE_WIDTH* 8)- 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

pipe

Pipe for data sync/clock issues.

control

Block RAM control, so it will act like a FIFO.

inst_dc_block_ram

Block RAM