

ft245_sync_to_axis.v

AUTHORS

JAY CONVERTINO

DATES

2022/08/09

INFORMATION

Brief

Converter FT245 sync FIFO interface to AXIS.

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.

ft245_sync_to_axis

```
module ft245_sync_to_axis #(
    parameter
    BUS_WIDTH
    =
1
) ( input rstn, input ft245_dclk, inout [BUS_WIDTH-1:0] ft245_ben, inout [
```

Converter FT245 sync FIFO interface to AXIS.

Parameters

BUS_WIDTH Width of the FT245 and AXIS bus.
parameter

Ports

rstn Negative reset

ft245_dclk Input clock from FIFO.

ft245_ben Byte enable used in FT60x, similar to AXIS tkeep in 1 is a valid byte for each bit.

ft245_data FIFO data bus

ft245_rdn Enable read on active low

ft245_wrn Enable write on active low

ft245_siwun Send Immediate / Wakeup for USB suspend. Active low.

ft245_txen When low, write data to the fifo.

ft245_rxfn When low, read data from the fifo.

ft245_oen Output enable active low

ft245_rstn Negative Reset

ft245_wakeupn Sleep ft245 active low

s_axis_tdata Input axis data

s_axis_tkeep Input axis data bytes that are valid. Each bit equals one byte.

s_axis_tvalid Input axis data is valid when active high.

s_axis_tready Input data bus is ready when signal is active high.

m_axis_tdata Output axis data

m_axis_tkeep Output what axis data bytes are valid. Each bit equals one byte.

m_axis_tvalid Output is active high when axis data is valid.

m_axis_tready Output data bus is told that the receive device is ready. The device is ready if it asserts this signal active high.

DATA STORE REGISTERS

Register data based upon ft245 clocks.

r_oen

reg r_oen

output enable registers

rr_oen

reg rr_oen

output enable registers registers

rrr_oen

```
reg rrr_oen
```

output enable registers registers registers

r_m_axis_tdata

```
reg [(  
  BUS_WIDTH*8  
)-1:0] r_m_axis_tdata
```

master axis register to hold tdata for tready not ready at end condition

r_m_axis_tkeep

```
reg [BUS_WIDTH-1:0] r_m_axis_tkeep
```

master axis register to hold tkeep for tready not ready at end condition

r_m_axis_tvalid

```
reg r_m_axis_tvalid
```

master axis register to hold tvalid for tready not ready at end condition

ASSIGNMENTS

How various combinations of logic are created and data dealt with.

ft245_data

```
assign ft245_data = (  
  s_axis_tdata  
  : '  
  bz  
)  
rr_oen & r_oen ?
```

combinatorial signals to convert registers and axis to and from ft245. tristate ft245 based on output enable state

ft245_ben

```
assign ft245_ben = (  
  s_axis_tkeep  
  : '  
  bz  
)  
rr_oen & r_oen ?
```

tristate ft245 based on output enable state

ft245_wrn

```
assign ft245_wrn = ft245_txen | ~ft245_rxfn | ~s_axis_tvalid | ~rr_oen
```

only allow write if there is space, nothing available to read, valid data available, and output enable is timed correctly.

ft245_oen

```
assign ft245_oen = rr_oen
```

output enable

ft245_rdn

```
assign ft245_rdn = ~m_axis_tready | rrr_oen | rr_oen & r_oen
```

only ready when output enable is correctly timed and we are ready for data ft245 will output data as soon as oen is applied (FWFT).

ft245_wakeupn

```
assign ft245_wakeupn = 1'b0
```

always keep it awake

ft245_siwn

```
assign ft245_siwn = 1'b0
```

always keep it awake

ft245_rstn

```
assign ft245_rstn = rstn
```

apply system reset to ft245

s_axis_tready

```
assign s_axis_tready = (  
ft245_txen &
```

```
ft245_rxfn  
) & rr_oen
```

convert ft245 to ready. only ready when write buffer is available, nothing is incoming, and output enable is set correctly.

m_axis_tdata

```
assign m_axis_tdata = (  
    r_m_axis_tdata  
    :  
    ft245_data  
)
```

rr_oen | r_oen ?

output ft245 to master axis. at end, output registers incase next core was not ready.

m_axis_tkeep

```
assign m_axis_tkeep = (  
    r_m_axis_tkeep  
    :  
    ft245_ben  
)
```

rr_oen | r_oen ?

output ft245 to master axis. at end, output registers incase next core was not ready.

m_axis_tvalid

```
assign m_axis_tvalid = (  
    r_m_axis_tvalid  
    : ~  
    rrr_oen | ft245_rxfn  
)
```

rr_oen | r_oen ?
(

data is only valid in the correct output enable register state and is no longer valid if rxfn indicates the receive exhausted.