

system_pl_wrapper.v

AUTHORS

JAY CONVERTINO

DATES

2023/11/02

INFORMATION

Brief

System wrapper for pl only for zc702 board.

License MIT

Copyright 2023 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.

system_pl_wrapper

```
module system_pl_wrapper #(
    parameter
    FPGA_TECHNOLOGY
    =
    0,
    parameter
    FPGA_FAMILY
    =
    0,
    parameter
    SPEED_GRADE
```

```

    =
    0,
    parameter
    DEV_PACKAGE
    =
    0,
    parameter
    ADC_INIT_DELAY
    =
    23,
    parameter
    DAC_INIT_DELAY
    =
    0,
    parameter
    DELAY_REFCLK_FREQUENCY
    =
    200
) ( input axi_aclk, input axi_aresetn, input s_axi_awvalid, input [31:0] s_

```

System wrapper for pl only for zc702 board.

Parameters

FPGA_TECHNOLOGY parameter	Type of FPGA, such as Ultrascale, Arria 10. 1 is for 7 series.
FPGA_FAMILY parameter	Sub type of fpga, such as GX, SX, etc. 4 is for zynq.
SPEED_GRADE parameter	Number that corresponds to the ships recommeneded speed. 10 is for -1.
DEV_PACKAGE parameter	Specify a number that is equal to the manufactures package. 14 is for cl.
DELAY_REFCLK_FREQUENCY parameter	Reference clock frequency used for ad_data_in instances
ADC_INIT_DELAY parameter	Initial Delay for the ADC
DAC_INIT_DELAY parameter	Initial Delay for the DAC

Ports

axi_aclk	AXI Lite control bus
axi_aresetn	AXI Lite control bus
s_axi_awvalid	AXI Lite control bus
s_axi_awaddr	AXI Lite control bus
s_axi_awready	AXI Lite control bus
s_axi_awprot	AXI Lite control bus
s_axi_wvalid	AXI Lite control bus
s_axi_wdata	AXI Lite control bus
s_axi_wstrb	AXI Lite control bus
s_axi_wready	AXI Lite control bus
s_axi_bvalid	AXI Lite control bus
s_axi_bresp	AXI Lite control bus
s_axi_bready	AXI Lite control bus

s_axi_arvalid	AXI Lite control bus
s_axi_araddr	AXI Lite control bus
s_axi_arready	AXI Lite control bus
s_axi_arprot	AXI Lite control bus
s_axi_rvalid	AXI Lite control bus
s_axi_rready	AXI Lite control bus
s_axi_rresp	AXI Lite control bus
s_axi_rdata	AXI Lite control bus
adc_dma_irq	fmcomms5 ADC irq
dac_dma_irq	fmcomms5 DAC irq
delay_clk	fmcomms5 delay clock
rx_clk_in_0_p	fmcomms5 0 rx clk
rx_clk_in_0_n	fmcomms5 0 rx clk
rx_frame_in_0_p	fmcomms5 0 rx frame
rx_frame_in_0_n	fmcomms5 0 rx frame
rx_data_in_0_p	fmcomms5 0 rx data
rx_data_in_0_n	fmcomms5 0 rx data
tx_clk_out_0_p	fmcomms5 0 tx clk
tx_clk_out_0_n	fmcomms5 0 tx clk
tx_frame_out_0_p	fmcomms5 0 tx frame
tx_frame_out_0_n	fmcomms5 0 tx frame
tx_data_out_0_p	fmcomms5 0 tx data
tx_data_out_0_n	fmcomms5 0 tx data
txnrx_0	fmcomms5 0 txnrx
enable_0	fmcomms5 0 enable
up_enable_0	fmcomms5 0 enable input
up_txnrx_0	fmcomms5 0 txnrx select input
tdd_sync_0_t	fmcomms5 0 TDD sync i/o
tdd_sync_0_i	fmcomms5 0 TDD sync i/o
tdd_sync_0_o	fmcomms5 0 TDD sync i/o
rx_clk_in_1_p	fmcomms5 1 rx clk
rx_clk_in_1_n	fmcomms5 1 rx clk
rx_frame_in_1_p	fmcomms5 1 rx frame
rx_frame_in_1_n	fmcomms5 1 rx frame
rx_data_in_1_p	fmcomms5 1 rx data
rx_data_in_1_n	fmcomms5 1 rx data
tx_clk_out_1_p	fmcomms5 1 tx clk
tx_clk_out_1_n	fmcomms5 1 tx clk
tx_frame_out_1_p	fmcomms5 1 tx frame
tx_frame_out_1_n	fmcomms5 1 tx frame
tx_data_out_1_p	fmcomms5 1 tx data

tx_data_out_1_n	fmcomms5 1 tx data
txnrx_1	fmcomms5 1 txnrx
enable_1	fmcomms5 1 enable
up_enable_1	fmcomms5 1 enable input
up_txnrx_1	fmcomms5 1 txnrx select input
tdd_sync_1_t	fmcomms5 1 TDD sync i/o
tdd_sync_1_i	fmcomms5 1 TDD sync i/o
tdd_sync_1_o	fmcomms5 1 TDD sync i/o
adc_m_dest_axi_awaddr	fmcomms5 ADC DMA
adc_m_dest_axi_awlen	fmcomms5 ADC DMA
adc_m_dest_axi_awsz	fmcomms5 ADC DMA
adc_m_dest_axi_awburst	fmcomms5 ADC DMA
adc_m_dest_axi_awprot	fmcomms5 ADC DMA
adc_m_dest_axi_awcache	fmcomms5 ADC DMA
adc_m_dest_axi_awvalid	fmcomms5 ADC DMA
adc_m_dest_axi_awready	fmcomms5 ADC DMA
adc_m_dest_axi_wdata	fmcomms5 ADC DMA
adc_m_dest_axi_wstrb	fmcomms5 ADC DMA
adc_m_dest_axi_wready	fmcomms5 ADC DMA
adc_m_dest_axi_wvalid	fmcomms5 ADC DMA
adc_m_dest_axi_wlast	fmcomms5 ADC DMA
adc_m_dest_axi_bvalid	fmcomms5 ADC DMA
adc_m_dest_axi_bresp	fmcomms5 ADC DMA
adc_m_dest_axi_bready	fmcomms5 ADC DMA
dac_m_src_axi_arready	fmcomms5 DAC DMA
dac_m_src_axi_arvalid	fmcomms5 DAC DMA
dac_m_src_axi_araddr	fmcomms5 DAC DMA
dac_m_src_axi_arlen	fmcomms5 DAC DMA
dac_m_src_axi_arsz	fmcomms5 DAC DMA
dac_m_src_axi_arburst	fmcomms5 DAC DMA
dac_m_src_axi_arprot	fmcomms5 DAC DMA
dac_m_src_axi_arcache	fmcomms5 DAC DMA
dac_m_src_axi_rdata	fmcomms5 DAC DMA
dac_m_src_axi_rready	fmcomms5 DAC DMA
dac_m_src_axi_rvalid	fmcomms5 DAC DMA
dac_m_src_axi_rresp	fmcomms5 DAC DMA
dac_m_src_axi_rlast	fmcomms5 DAC DMA
iic_sda_fmc	i2c for fmc
iic_scl_fmc	i2c for fmc
iic2intc_irpt	i2c for fmc

INSTANTIATED MODULES

iic_sda_iobuf

```
ad_iobuf #(
    .DATA_WIDTH(1)
) iic_sda_iobuf ( .dio_t (sda_t), .dio_i (sda_o), .dio_o (sda_i), .dio_p (sda_p))
```

Tristate i2c sda

iic_scl_iobuf

```
ad_iobuf #(
    .DATA_WIDTH(1)
) iic_scl_iobuf ( .dio_t (scl_t), .dio_i (scl_o), .dio_o (scl_i), .dio_p (scl_p))
```

Tristate i2c scl

inst_dma_rstgen

```
dma_rstgen inst_dma_rstgen (
    .slowest_sync_clk(delay_clk),
    .ext_reset_in(axi_aresetn),
    .aux_reset_in(1'b1),
    .mb_debug_sys_rst(1'b0),
    .dcm_locked(1'b1),
    .mb_reset(),
    .bus_struct_reset(),
    .peripheral_reset(),
    .interconnect_aresetn(),
    .peripheral_aresetn(m_axi_aresetn)
)
```

Generate a new DMA reset based on delay clock.

inst_ad9361x2_pl_wrapper

```
ad9361x2_pl_wrapper #(
    .FPGA_TECHNOLOGY(FPGA_TECHNOLOGY),
    .FPGA_FAMILY(FPGA_FAMILY),
    .FPGA_DEVICE(FPGA_DEVICE)
)
```

```

SPEED_GRADE(SPEED_GRADE),
DEV_PACKAGE(DEV_PACKAGE),
ADC_INIT_DELAY(ADC_INIT_DELAY),
DAC_INIT_DELAY(DAC_INIT_DELAY),
DELAY_REFCLK_FREQUENCY(DELAY_REFCLK_FREQUENCY)
) inst_ad9361x2_pl_wrapper ( .axi_aclk(axi_aclk), .axi_aresetn(axi_aresetn)

```

Module instance of inst_ad9361x2_pl_wrapper for the fmcomms5 device.

inst_axi_crossbar_pl

```

axi_crossbar_pl inst_axi_crossbar_pl (
aclk(axi_aclk),
aresetn(axi_aresetn),
s_axi_awaddr(s_axi_awaddr),
s_axi_awprot(s_axi_awprot),
s_axi_awvalid(s_axi_awvalid),
s_axi_awready(s_axi_awready),
s_axi_wdata(s_axi_wdata),
s_axi_wstrb(s_axi_wstrb),
s_axi_wvalid(s_axi_wvalid),
s_axi_wready(s_axi_wready),
s_axi_bresp(s_axi_bresp),
s_axi_bvalid(s_axi_bvalid),
s_axi_bready(s_axi_bready),
s_axi_araddr(s_axi_araddr),
s_axi_arprot(s_axi_arprot),
s_axi_arvalid(s_axi_arvalid),
s_axi_arready(s_axi_arready),
s_axi_rdata(s_axi_rdata),
s_axi_rresp(s_axi_rresp),
s_axi_rvalid(s_axi_rvalid),
s_axi_rready(s_axi_rready),
m_axi_awaddr({iic_fmc_axi_awaddr, connect_axi_awaddr}),
m_axi_awprot({iic_fmc_axi_awprot, connect_axi_awprot}),

```

```

m_axi_awvalid({iic_fmc_axi_awvalid, connect_axi_awvalid}),
m_axi_awready({iic_fmc_axi_awready, connect_axi_awready}),
m_axi_wdata({iic_fmc_axi_wdata, connect_axi_wdata}),
m_axi_wstrb({iic_fmc_axi_wstrb, connect_axi_wstrb}),
m_axi_wvalid({iic_fmc_axi_wvalid, connect_axi_wvalid}),
m_axi_wready({iic_fmc_axi_wready, connect_axi_wready}),
m_axi_bresp({iic_fmc_axi_bresp, connect_axi_bresp}),
m_axi_bvalid({iic_fmc_axi_bvalid, connect_axi_bvalid}),
m_axi_bready({iic_fmc_axi_bready, connect_axi_bready}),
m_axi_araddr({iic_fmc_axi_araddr, connect_axi_araddr}),
m_axi_arprot({iic_fmc_axi_arprot, connect_axi_arprot}),
m_axi_arvalid({iic_fmc_axi_arvalid, connect_axi_arvalid}),
m_axi_arready({iic_fmc_axi_arready, connect_axi_arready}),
m_axi_rdata({iic_fmc_axi_rdata, connect_axi_rdata}),
m_axi_rresp({iic_fmc_axi_rresp, connect_axi_rresp}),
m_axi_rvalid({iic_fmc_axi_rvalid, connect_axi_rvalid}),
m_axi_rready({iic_fmc_axi_rready, connect_axi_rready})
)

```

Module instance of axi_crossbar_pl for the fmcomms5 device.

inst_axi_iic_fmc

```

axi_iic_fmc inst_axi_iic_fmc (
s_axi_aclk(axi_aclk),
s_axi_aresetn(axi_aresetn),
iic2intc_irpt(iic2intc_irpt),
s_axi_awaddr(iic_fmc_axi_awaddr[8:0]),
s_axi_awvalid(iic_fmc_axi_awvalid),
s_axi_awready(iic_fmc_axi_awready),
s_axi_wdata(iic_fmc_axi_wdata),
s_axi_wstrb(iic_fmc_axi_wstrb),
s_axi_wvalid(iic_fmc_axi_wvalid),
s_axi_wready(iic_fmc_axi_wready),
s_axi_bresp(iic_fmc_axi_bresp),

```

s_axi_bvalid(iic_fmc_axi_bvalid),	.
s_axi_bready(iic_fmc_axi_bready),	.
s_axi_araddr(iic_fmc_axi_araddr[8:0]),	.
s_axi_arvalid(iic_fmc_axi_arvalid),	.
s_axi_arready(iic_fmc_axi_arready),	.
s_axi_rdata(iic_fmc_axi_rdata),	.
s_axi_rresp(iic_fmc_axi_rresp),	.
s_axi_rvalid(iic_fmc_axi_rvalid),	.
s_axi_rready(iic_fmc_axi_rready),	.
sda_i(sda_i),	.
sda_o(sda_o),	.
sda_t(sda_t),	.
scl_i(scl_i),	.
scl_o(scl_o),	.
scl_t(scl_t),	.
gpo()	.
)	

Module instance of axi_iic_fmc for the fmcomms5 device.