system wrapper.v

AUTHORS

JAY CONVERTINO

DATES

2024/11/25

INFORMATION

Brief

System wrapper for ps.

License MIT

Copyright 2024 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_wrapper

```
module system_wrapper (
    input
    tck,
    input
    tms,
    input
    tdi,
    output
    tdo,
```

<pre>input clk, input resetn,</pre>	down f
12:0] ddr2_addr,	inout [
2:0] ddr2_ba, inout ddr2_cas_n, inout ddr2_ck_n, inout ddr2_ck_p, inout ddr2_cke, inout ddr2_ccs_n,	inout [
1:0] ddr2_dm,	inout [
15:0] ddr2_dq,	inout [
1:0] ddr2_dqs_n,	inout [
1:0] ddr2_dqs_p, inout ddr2_odt, inout ddr2_ras_n, inout ddr2_reset_n, inout ddr2_we_n, output eth_mdc, inout eth_mdio, input eth_rstn, input eth_crsdv, input eth_rxerr,	
1:0] eth_rxd, output eth_txen,	input [
1:0] eth_txd, output eth_refclk, input eth_50mclk,	output [
3:0] vga_r,	output [

```
3:0]
vga_g,
                                                                       output [
3:0]
vga_b,
output
vga_hs,
output
vga_vs,
                                                                        output [
15:0]
leds,
                                                                         input [
15:0]
slide_switches,
input
ftdi_tx,
output
ftdi_rx,
input
ftdi_rts,
output
ftdi_cts,
                                                                         inout [
3:0]
qspi_dq,
output
qspi_csn,
output
sd_reset,
input
sd_cd,
output
sd_sck,
inout
sd_cmd,
                                                                        inout [
3:0]
sd_dat
```

System wrapper for Vexriscv Veronica RISCV ps.

Ports

```
JTAG
tck
ifdef _JTAG_IO input
                      JTAG
tms
input
tdi
                      JTAG
input
tdo
                      JTAG
output
clk
                      Master Input Clock
endif input
resetn
                      Master Reset Input
input
ddr2_addr
                      DDR interface
inout[ 12: 0]
ddr2_ba
                      DDR interface
inout[ 2: 0]
```

ddr2_cas_n	DDR interface
ddr2_ck_n	DDR interface
ddr2_ck_p	DDR interface
ddr2_cke	DDR interface
ddr2_cs_n	DDR interface
ddr2_dm	DDR interface
ddr2_dq inout[15: 0]	DDR interface
ddr2_dqs_n inout[1: 0]	DDR interface
ddr2_dqs_p	DDR interface
ddr2_odt	DDR interface
ddr2_ras_n	DDR interface
ddr2_reset_n	DDR interface
ddr2_we_n	DDR interface
eth_mdc	ethernet interface
eth_mdio	ethernet interface
eth_rstn	ethernet interface
eth_crsdv	ethernet interface
eth_rxerr	ethernet interface
eth_rxd input[1: 0]	ethernet interface
eth_txen	ethernet interface
eth_txd output[1: 0]	ethernet interface
eth_refclk	ethernet interface
eth_50mclk	ethernet interface
vga_r output[3: 0]	vga interface
vga_g output[3: θ]	vga interface

vga interface

leds board leds

output[15: 0]

slide_switches board slide switches

input[15: 0]

ftdi_tx FTDI UART TX

input

vga_b

ftdi_rx FTDI UART RX output

ftdi_rts FTDI UART RTS

input

ftdi_cts FTDI UART CTS

output

qspi_dq QUAD SPI

inout[3: 8]

qspi_csn QUAD SPI

output

sd reset SD CARD Reset

out out

sd_cd SD CARD CD

input

sd_sck SD CARD sck

output

sd_cmd SD CARD cmd

inout

sd_dat SD CARD dat

inout[3: 0]

INSTANTIANTED MODULES

inst_util_mii_to_rmii

```
util_mii_to_rmii #(
    ...
INTF_CFG(0),
    ...
RATE_10_100(0)
) inst_util_mii_to_rmii ( .mac_tx_en(MII_tx_en), .mac_txd(MII_txd), .mac_tx
```

convert from mii to rmii (UNTESTED)

inst_system_ps_wrapper

```
system_ps_wrapper inst_system_ps_wrapper (
```

```
`ifdef
_JTAG_IO
tck(tck),
tms(tms),
tdi(tdi),
tdo(tdo),
endif
DDR_addr(ddr2_addr),
DDR_ba(ddr2_ba),
DDR_cas_n(ddr2_cas_n),
DDR_ck_n(ddr2_ck_n),
DDR_ck_p(ddr2_ck_p),
DDR_cke(ddr2_cke),
DDR_cs_n(ddr2_cs_n),
DDR_dm(ddr2_dm),
DDR_dq(ddr2_dq),
DDR_dqs_n(ddr2_dqs_n),
DDR_dqs_p(ddr2_dqs_p),
DDR_odt(ddr2_odt),
DDR_ras_n(ddr2_ras_n),
DDR_we_n(ddr2_we_n),
IRQ(3'b000),
MDIO_mdc(eth_mdc),
MDIO_mdio_io(eth_mdio),
MII_col(MII_col),
MII_crs(MII_crs),
MII_rst_n(MII_rst_n),
MII_rx_clk(MII_rx_clk),
MII_rx_dv(MII_rx_dv),
MII_rx_er(MII_rx_er),
MII_rxd(MII_rxd),
MII_tx_clk(MII_tx_clk),
MII_tx_en(MII_tx_en),
MII_txd(MII_txd),
```

```
M_AXI_araddr(),
M_AXI_arprot(),
M_AXI_arready(1'b1),
M_AXI_arvalid(),
M_AXI_awaddr(),
M_AXI_awprot(),
M_AXI_awready(1'b1),
M_AXI_awvalid(),
M_AXI_bready(),
M_AXI_bresp(3'b000),
M_AXI_bvalid(2'b00),
M_AXI_rdata(32'h00000000),
M_AXI_rready(),
M_AXI_rresp(3'b000),
M_AXI_rvalid(1'b00),
M_AXI_wdata(),
M_AXI_wready(1'b1),
M_AXI_wstrb(),
M_AXI_wvalid(),
QSPI_0_io0_io(qspi_dq[0]),
QSPI_0_io1_io(qspi_dq[1]),
QSPI_0_io2_io(qspi_dq[2]),
QSPI_0_io3_io(qspi_dq[3]),
QSPI_0_ss_io(qspi_csn),
UART_rxd(ftdi_tx),
UART_txd(ftdi_rx),
gpio_io_i(slide_switches),
gpio_io_o(leds),
gpio_io_t(),
s_axi_clk(),
spi_io0_i(1'b0),
spi_ioO_o(),
spi_io0_t(),
```

```
spi_io1_i(1'b0),
spi_io1_o(),
spi_io1_t(),
spi_sck_i(1'b0),
spi_sck_o(),
spi_sck_t(),
spi_ss_i(1'b0),
spi_ss_o(),
spi_ss_t(),
sys_clk(clk),
sys_rstn(resetn),
vga_hsync(vga_hs),
vga_vsync(vga_vs),
vga_r(s_vga_r),
vga_g(s_vga_g),
vga_b(s_vga_b),
sd_resetn(s_sd_resetn),
sd_cd(sd_cd),
sd_sck(sd_sck),
sd_cmd(sd_cmd),
sd_dat(sd_dat)
```

Wraps all of the RISCV CPU core and its devices.