VERONICA_AXI_BAREMETAL

November 25, 2024

Jay Convertino

Contents

1 Usage

1.1 Introduction

The Veronica AXI Baremetal is a base Vexriscv system. This version of the Vexriscv core emulates the E31 core in its mapping of features. Difference is that this core can generate a MMU and non-PMP enabled versions. The JTAG configuration can also be changed between the Xilinx BSCANE, or JTAG IO for external devices, or none. Each target tries to use all of the resources of the board using free IP's from the vendor, or open source IP cores intergrated into fusesoc.

1.2 Dependencies

The following are the dependencies of the cores.

- fusesoc 2.X
- iverilog (simulation)
- cocotb (simulation)

1.2.1 fusesoc_info Depenecies

- nexys-a7-100t
  - AFRL:utility:digilent_nexys-a7-100t_board_base_constr:1.0.0
  - AFRL:utility:digilent_nexys-a7-100t_board_base_ddr_cfg:1.0.0
  - AFRL:utility:vivado_board_support_packages
  - AD:ethernet:util_mii_to_rmii:1.0.0
- crosslink-nx_eval
  - AFRL:utility:lattice_crosslink-nx_eval_board_base:1.0.0
- dep
  - AFRL:utility:helper:1.0.0
  - AFRL:utility:tcl_helper_check:1.0.0
  - zipcpu:axi_lite:crossbar:1.0.0
  - zipcpu:axi:crossbar:1.0.0
  - zipcpu:axi:sdio:1.0.0
  - zipcpu:axi:axixclk:1.0.0
- dep_uc_jtag_io

    - spinalhdl:cpu:veronica_axi_jtag_io:1.0.0
- dep_uc_secure_jtag_io
    - spinalhdl:cpu:veronica_axi_secure_jtag_io:1.0.0
- dep_uc_jtag_bscane
    - spinalhdl:cpu:veronica_axi_jtag_xilinx_bscane:1.0.0
- dep_uc_secure_bscane
    - spinalhdl:cpu:veronica_axi_secure_jtag_xilinx_bscane:1.0.0

2 Architecture

The project contains four wrappers

- **system_wrapper** Contains the top level project module and contains system_ps_wrapper.
- **system_ps_wrapper** Contains the processor system IP wrappers.

Please see 5 for more information per target.

3 Building

The all Veronica AXI Baremetal project source files are written in Verilog 2001. They should synthesize in any modern FPGA software. The core comes as a fusesoc packaged core and can be included in any other core. Be sure to make sure you have meet the dependencies listed in the previous section.

3.1 fusesoc

Fusesoc is a system for building FPGA software without relying on the internal project management of the tool. Avoiding vendor lock in to Vivado or Quartus. These cores, when included in a project, can be easily integrated and targets created based upon the end developer needs. The core by itself is not a part of a system and should be integrated into a fusesoc based system. Simulations are setup to use fusesoc and are a part of its targets.

3.2 Source Files

3.2.1 fusesoc_info File List

- nexys-a7-100t
  - 'nexys-a7-100t/system_wrapper.v': 'file_type': 'verilogSource'
  - 'nexys-a7-100t/system_constr.tcl': 'file_type': 'SDC'
  - 'nexys-a7-100t/system_gen_ps.tcl': 'file_type': 'tclSource'
  - 'nexys-a7-100t/system_ps_wrapper.v': 'file_type': 'verilog-Source'
  - 'nexys-a7-100t/system_gen.tcl': 'file_type': 'tclSource'
- crosslink-nx_eval
  - 'crosslink-nx_eval/system_constr.pdc': 'file_type': 'PDC'
  - 'crosslink-nx_eval/system_wrapper.v': 'file_type': 'verilogSource'
- dep_uc_jtag_io
  - 'nexys-a7-100t/system_define.tcl': 'file_type': 'tclSource'
- dep_uc_secure_jtag_io
  - 'nexys-a7-100t/system_define.tcl': 'file_type': 'tclSource'

3.3 Targets

3.3.1 fusesoc_info Targets

- nexys-a7-100t

  Info: Base for nexys-a7-100t digilent development board builds, do not use.

- nexys-a7-100t_uc_secure_jtag_io

  Info: Build for nexys-a7-100t digilent development board with PMP enabled Veronica RISCV.

- nexys-a7-100t_uc_jtag_io

  Info: Build for nexys-a7-100t digilent development board with standard Veronica RISCV.

- nexys-a7-100t_uc_secure_jtag_bscane

  Info: Build for nexys-a7-100t digilent development board with PMP enabled Veronica RISCV.

- nexys-a7-100t_uc_jtag_bscane

  Info: Build for nexys-a7-100t digilent development board with standard Veronica RISCV.

3.4 Directory Guide

Below highlights important folders from the root of the directory.

1. **docs** Contains all documentation related to this project.
   - **manual** Contains user manual and github page that are generated from the latex sources.
2. **nexys-a7-100t** Contains source files for nexys-a7-100t
3. **crosslink-nx_eval** Contains source file for crosslink-nx_eval, a future target.

4 Simulation

There is no simulation at the moment. Maybe a future addition?

5 Module Documentation

There project has multiple modules. The targets are the top system wrappers.

- **nexys-a7-100t**
- **crosslink-nx_eval**

The next sections document the module in great detail.

system_ps_wrapper.v

AUTHORS

JAY CONVERTINO

DATES

2024/11/25

INFORMATION

Brief

System wrapper for ps.

License MIT

system_ps_wrapper

```
module system_ps_wrapper (
                                                          `ifdef _JTAG_IO

input
tck,
input
tms,
input
tdi,
output
tdo,
                                                          `endif output [
```

```
12:0]
DDR_addr,
                                                output [
2:0]
DDR_ba,
output
DDR_cas_n,
                                                output [
0:0]
DDR_ck_n,
                                                output [
0:0]
DDR_ck_p,
                                                output [
0:0]
DDR_cke,
                                                output [
0:0]
DDR_cs_n,
                                                output [
1:0]
DDR_dm,
                                                 inout [
15:0]
DDR_dq,
                                                 inout [
1:0]
DDR_dqs_n,
                                                 inout [
1:0]
DDR_dqs_p,
                                                output [
0:0]
DDR_odt,
output
DDR_ras_n,
output
DDR_we_n,
                                                 input [
2:0]
IRQ,
output
MDIO_mdc,
inout
MDIO_mdio_io,
input
MII_col,
input
MII_crs,
output
MII_rst_n,
input
MII_rx_clk,
input
MII_rx_dv,
input
MII_rx_er,
                                                 input [
3:0]
MII_rxd,
input
MII_tx_clk,
output
MII_tx_en,
                                                output [
3:0]
```

```
MII_txd,
                                                                output [
31:0]
M_AXI_araddr,
                                                               output [
2:0]
M_AXI_arprot,
input
M_AXI_arready,
output
M_AXI_arvalid,
                                                                output [
31:0]
M_AXI_awaddr,
                                                               output [
2:0]
M_AXI_awprot,
input
M_AXI_awready,
output
M_AXI_awvalid,
output
M_AXI_bready,
                                                                input [
1:0]
M_AXI_bresp,
input
M_AXI_bvalid,
                                                                 input [
31:0]
M_AXI_rdata,
output
M_AXI_rready,
                                                                input [
1:0]
M_AXI_rresp,
input
M_AXI_rvalid,
                                                                output [
31:0]
M_AXI_wdata,
input
M_AXI_wready,
                                                               output [
3:0]
M_AXI_wstrb,
output
M_AXI_wvalid,
inout
QSPI_0_io0_io,
inout
QSPI_0_io1_io,
inout
QSPI_0_io2_io,
inout
QSPI_0_io3_io,
                                                                inout [
0:0]
QSPI_0_ss_io,
input
UART_rxd,
output
UART_txd,
                                                                 input [
31:0]
gpio_io_i,
```

```
                                                                output [
31:0]
gpio_io_o,
                                                                output [
31:0]
gpio_io_t,
output
s_axi_clk,
input
spi_io0_i,
output
spi_io0_o,
output
spi_io0_t,
input
spi_io1_i,
output
spi_io1_o,
output
spi_io1_t,
input
spi_sck_i,
output
spi_sck_o,
output
spi_sck_t,
                                                                 input [
0:0]
spi_ss_i,
                                                               output [
0:0]
spi_ss_o,
output
spi_ss_t,
input
sys_clk,
input
sys_rstn,
output
vga_hsync,
output
vga_vsync,
                                                               output [
5:0]
vga_r,
                                                               output [
5:0]
vga_g,
                                                               output [
5:0]
vga_b,
output
sd_resetn,
input
sd_cd,
output
sd_sck,
inout
sd_cmd,
                                                                inout [
3:0]
sd_dat
)
```

System wrapper for ps.

Ports

tck `ifdef _JTAG_IO input`	JTAG
tms `input`	JTAG
tdi `input`	JTAG
tdo `output`	JTAG
DDR_addr `endif output[ 12: 0]`	DDR interface
DDR_ba `output[ 2: 0]`	DDR interface
DDR_cas_n `output`	DDR interface
DDR_ck_n `output[ 0: 0]`	DDR interface
DDR_ck_p `output[ 0: 0]`	DDR interface
DDR_cke `output[ 0: 0]`	DDR interface
DDR_cs_n `output[ 0: 0]`	DDR interface
DDR_dm `output[ 1: 0]`	DDR interface
DDR_dq `inout[ 15: 0]`	DDR interface
DDR_dqs_n `inout[ 1: 0]`	DDR interface
DDR_dqs_p `inout[ 1: 0]`	DDR interface
DDR_odt `output[ 0: 0]`	DDR interface
DDR_ras_n `output`	DDR interface
DDR_we_n `output`	DDR interface
IRQ `input[ 2: 0]`	External Interrupts
MDIO_mdc `output`	Ethernet Interface MII
MDIO_mdio_io `inout`	Ethernet Interface MII
MII_col `input`	Ethernet Interface MII
MII_crs `input`	Ethernet Interface MII
MII_rst_n `output`	Ethernet Interface MII
MII_rx_clk	Ethernet Interface MII

input

MII_rx_dv input	Ethernet Interface MII
MII_rx_er input	Ethernet Interface MII
MII_rxd input[3: 0]	Ethernet Interface MII
MII_tx_clk input	Ethernet Interface MII
MII_tx_en output	Ethernet Interface MII
MII_txd output[3: 0]	Ethernet Interface MII
M_AXI_araddr output[31: 0]	External AXI Lite Master Interface
M_AXI_arprot output[2: 0]	External AXI Lite Master Interface
M_AXI_arready input	External AXI Lite Master Interface
M_AXI_arvalid output	External AXI Lite Master Interface
M_AXI_awaddr output[31: 0]	External AXI Lite Master Interface
M_AXI_awprot output[2: 0]	External AXI Lite Master Interface
M_AXI_awready input	External AXI Lite Master Interface
M_AXI_awvalid output	External AXI Lite Master Interface
M_AXI_bready output	External AXI Lite Master Interface
M_AXI_bresp input[1: 0]	External AXI Lite Master Interface
M_AXI_bvalid input	External AXI Lite Master Interface
M_AXI_rdata input[31: 0]	External AXI Lite Master Interface
M_AXI_rready output	External AXI Lite Master Interface
M_AXI_rresp input[1: 0]	External AXI Lite Master Interface
M_AXI_rvalid input	External AXI Lite Master Interface
M_AXI_wdata output[31: 0]	External AXI Lite Master Interface
M_AXI_wready input	External AXI Lite Master Interface
M_AXI_wstrb output[3: 0]	External AXI Lite Master Interface
M_AXI_wvalid output	External AXI Lite Master Interface

QSPI_0_io0_io inout	Quad SPI
QSPI_0_io1_io inout	Quad SPI
QSPI_0_io2_io inout	Quad SPI
QSPI_0_io3_io inout	Quad SPI
QSPI_0_ss_io inout[0: 0]	Quad SPI
UART_rxd input	UART RX
UART_txd output	UART TX
gpio_io_i input[31: 0]	GPIO input
gpio_io_o output[31: 0]	GPIO output
gpio_io_t output[31: 0]	GPIO tristate select
s_axi_clk output	AXI Clock
spi_io0_i input	SPI IO
spi_io0_o output	SPI IO
spi_io0_t output	SPI IO
spi_io1_i input	SPI IO
spi_io1_o output	SPI IO
spi_io1_t output	SPI IO
spi_sck_i input	SPI IO
spi_sck_o output	SPI IO
spi_sck_t output	SPI IO
spi_ss_i input[0: 0]	SPI IO
spi_ss_o output[0: 0]	SPI IO
spi_ss_t output	SPI IO
sys_clk input	SYSTEM clock for pll
sys_rstn input	SYSTEM reset

vga_hsync output	VGA
vga_vsync output	VGA
vga_r output[5: 0]	VGA
vga_g output[5: 0]	VGA
vga_b output[5: 0]	VGA
sd_resetn output	sd card
sd_cd input	sd card
sd_sck output	sd card
sd_cmd inout	sd card
sd_dat inout[3: 0]	sd card

INSTANTIANTED MODULES

MDIO_mdio_iobuf

```
IOBUF MDIO_mdio_iobuf (
                                                                    .
I(MDIO_mdio_o),
                                                                    .
IO(MDIO_mdio_io),
                                                                    .
O(MDIO_mdio_i),
                                                                    .
T(MDIO_mdio_t)
)
```

TRISTATE IO

QSPI_0_io0_iobuf

```
IOBUF QSPI_0_io0_iobuf (
                                                                    .
I(QSPI_0_io0_o),
                                                                    .
IO(QSPI_0_io0_io),
                                                                    .
O(QSPI_0_io0_i),
                                                                    .
T(QSPI_0_io0_t)
)
```

TRISTATE IO

QSPI_0_io1_iobuf

```
 IOBUF QSPI_0_io1_iobuf (
                                                                                .
 I(QSPI_0_io1_o),
                                                                                .
 IO(QSPI_0_io1_io),
                                                                                .
 O(QSPI_0_io1_i),
                                                                                .
 T(QSPI_0_io1_t)
)
```

TRISTATE IO

QSPI_0_io2_iobuf

```
 IOBUF QSPI_0_io2_iobuf (
                                                                                .
 I(QSPI_0_io2_o),
                                                                                .
 IO(QSPI_0_io2_io),
                                                                                .
 O(QSPI_0_io2_i),
                                                                                .
 T(QSPI_0_io2_t)
)
```

TRISTATE IO

QSPI_0_io3_iobuf

```
 IOBUF QSPI_0_io3_iobuf (
                                                                                .
 I(QSPI_0_io3_o),
                                                                                .
 IO(QSPI_0_io3_io),
                                                                                .
 O(QSPI_0_io3_i),
                                                                                .
 T(QSPI_0_io3_t)
)
```

TRISTATE IO

QSPI_0_ss_iobuf_0

```
 IOBUF QSPI_0_ss_iobuf_0 (
                                                                                .
 I(QSPI_0_ss_o_0),
                                                                                .
 IO(QSPI_0_ss_io[0]),
                                                                                .
 O(QSPI_0_ss_i_0),
                                                                                .
 T(QSPI_0_ss_t)
)
```

TRISTATE IO

inst_axi_ddr_ctrl

```
axi_ddr_ctrl inst_axi_ddr_ctrl (
.aresetn(ddr_rstgen_peripheral_aresetn),
.ddr2_addr(DDR_addr),
.ddr2_ba(DDR_ba),
.ddr2_cas_n(DDR_cas_n),
.ddr2_ck_n(DDR_ck_n),
.ddr2_ck_p(DDR_ck_p),
.ddr2_cke(DDR_cke),
.ddr2_cs_n(DDR_cs_n),
.ddr2_dm(DDR_dm),
.ddr2_dq(DDR_dq[15:0]),
.ddr2_dqs_n(DDR_dqs_n[1:0]),
.ddr2_dqs_p(DDR_dqs_p[1:0]),
.ddr2_odt(DDR_odt),
.ddr2_ras_n(DDR_ras_n),
.ddr2_we_n(DDR_we_n),
.mmcm_locked(axi_ddr_ctrl_mmcm_locked),
.s_axi_araddr(m_axi_ddr_ARADDR & 32'h0FFFFFFF),
.s_axi_arburst(m_axi_ddr_ARBURST),
.s_axi_arcache(m_axi_ddr_ARCACHE),
.s_axi_arid(m_axi_ddr_ARID),
.s_axi_arlen(m_axi_ddr_ARLEN),
.s_axi_arlock(m_axi_ddr_ARLOCK),
.s_axi_arprot(m_axi_ddr_ARPROT),
.s_axi_arqos(m_axi_ddr_ARQOS),
.s_axi_arready(m_axi_ddr_ARREADY),
.s_axi_arsize(m_axi_ddr_ARSIZE),
.s_axi_arvalid(m_axi_ddr_ARVALID),
.s_axi_awaddr(m_axi_ddr_AWADDR & 32'h0FFFFFFF),
.s_axi_awburst(m_axi_ddr_AWBURST),
```

```
                                                                        .
 s_axi_awcache(m_axi_ddr_AWCACHE),                                      .
 s_axi_awid(m_axi_ddr_AWID),                                            .
 s_axi_awlen(m_axi_ddr_AWLEN),                                          .
 s_axi_awlock(m_axi_ddr_AWLOCK),                                        .
 s_axi_awprot(m_axi_ddr_AWPROT),                                        .
 s_axi_awqos(m_axi_ddr_AWQOS),                                          .
 s_axi_awready(m_axi_ddr_AWREADY),                                      .
 s_axi_awsize(m_axi_ddr_AWSIZE),                                        .
 s_axi_awvalid(m_axi_ddr_AWVALID),                                      .
 s_axi_bid(m_axi_ddr_BID),                                              .
 s_axi_bready(m_axi_ddr_BREADY),                                        .
 s_axi_bvalid(m_axi_ddr_BVALID),                                        .
 s_axi_bresp(m_axi_ddr_BRESP),                                          .
 s_axi_rdata(m_axi_ddr_RDATA),                                          .
 s_axi_rid(m_axi_ddr_RID),                                              .
 s_axi_rlast(m_axi_ddr_RLAST),                                          .
 s_axi_rready(m_axi_ddr_RREADY),                                        .
 s_axi_rvalid(m_axi_ddr_RVALID),                                        .
 s_axi_rresp(m_axi_ddr_RRESP),                                          .
 s_axi_wdata(m_axi_ddr_WDATA),                                          .
 s_axi_wlast(m_axi_ddr_WLAST),                                          .
 s_axi_wready(m_axi_ddr_WREADY),                                        .
 s_axi_wstrb(m_axi_ddr_WSTRB),                                          .
 s_axi_wvalid(m_axi_ddr_WVALID),                                        .
 sys_clk_i(ddr_clk),                                                    .
 sys_rst(sys_rstn),                                                     .
 ui_clk(axi_ddr_ctrl_ui_clk),                                           .
 ui_clk_sync_rst(axi_ddr_ctrl_ui_clk_sync_rst)
)
```

AXI DDR Controller

inst_axi_ethernet

```
axi_ethernet inst_axi_ethernet (
```

```
ip2intc_irpt(axi_ethernet_irq),
phy_col(MII_col),
phy_crs(MII_crs),
phy_dv(MII_rx_dv),
phy_mdc(MDIO_mdc),
phy_mdio_i(MDIO_mdio_i),
phy_mdio_o(MDIO_mdio_o),
phy_mdio_t(MDIO_mdio_t),
phy_rst_n(MII_rst_n),
phy_rx_clk(MII_rx_clk),
phy_rx_data(MII_rxd),
phy_rx_er(MII_rx_er),
phy_tx_clk(MII_tx_clk),
phy_tx_data(MII_txd),
phy_tx_en(MII_tx_en),
s_axi_aclk(axi_cpu_clk),
s_axi_araddr(m_axi_eth_ARADDR[12:0]),
s_axi_aresetn(sys_rstgen_peripheral_aresetn),
s_axi_arready(m_axi_eth_ARREADY),
s_axi_arvalid(m_axi_eth_ARVALID),
s_axi_awaddr(m_axi_eth_AWADDR[12:0]),
s_axi_awready(m_axi_eth_AWREADY),
s_axi_awvalid(m_axi_eth_AWVALID),
s_axi_bready(m_axi_eth_BREADY),
s_axi_bresp(m_axi_eth_BRESP),
s_axi_bvalid(m_axi_eth_BVALID),
s_axi_rdata(m_axi_eth_RDATA),
s_axi_rready(m_axi_eth_RREADY),
s_axi_rresp(m_axi_eth_RRESP),
s_axi_rvalid(m_axi_eth_RVALID),
s_axi_wdata(m_axi_eth_WDATA),
s_axi_wready(m_axi_eth_WREADY),
s_axi_wstrb(m_axi_eth_WSTRB),
```

```
                                                                                .
 s_axi_wvalid(m_axi_eth_WVALID)
)
```

AXI Ethernet MAC

inst_axi_gpio32

```
 axi_gpio32 inst_axi_gpio32 (
                                                                                .
 gpio_io_i(gpio_io_i),
                                                                                .
 gpio_io_o(gpio_io_o),
                                                                                .
 gpio_io_t(gpio_io_t),
                                                                                .
 s_axi_aclk(axi_cpu_clk),
                                                                                .
 s_axi_araddr(m_axi_gpio_ARADDR[8:0]),
                                                                                .
 s_axi_aresetn(sys_rstgen_peripheral_aresetn),
                                                                                .
 s_axi_arready(m_axi_gpio_ARREADY),
                                                                                .
 s_axi_arvalid(m_axi_gpio_ARVALID),
                                                                                .
 s_axi_awaddr(m_axi_gpio_AWADDR[8:0]),
                                                                                .
 s_axi_awready(m_axi_gpio_AWREADY),
                                                                                .
 s_axi_awvalid(m_axi_gpio_AWVALID),
                                                                                .
 s_axi_bready(m_axi_gpio_BREADY),
                                                                                .
 s_axi_bresp(m_axi_gpio_BRESP),
                                                                                .
 s_axi_bvalid(m_axi_gpio_BVALID),
                                                                                .
 s_axi_rdata(m_axi_gpio_RDATA),
                                                                                .
 s_axi_rready(m_axi_gpio_RREADY),
                                                                                .
 s_axi_rresp(m_axi_gpio_RRESP),
                                                                                .
 s_axi_rvalid(m_axi_gpio_RVALID),
                                                                                .
 s_axi_wdata(m_axi_gpio_WDATA),
                                                                                .
 s_axi_wready(m_axi_gpio_WREADY),
                                                                                .
 s_axi_wstrb(m_axi_gpio_WSTRB),
                                                                                .
 s_axi_wvalid(m_axi_gpio_WVALID)
)
```

AXI GPIO

inst_axi_spix4

```
 axi_spix4 inst_axi_spix4 (
                                                                                .
```

```
ext_spi_clk(axi_cpu_clk),
io0_i(QSPI_0_io0_i),
io0_o(QSPI_0_io0_o),
io0_t(QSPI_0_io0_t),
io1_i(QSPI_0_io1_i),
io1_o(QSPI_0_io1_o),
io1_t(QSPI_0_io1_t),
io2_i(QSPI_0_io2_i),
io2_o(QSPI_0_io2_o),
io2_t(QSPI_0_io2_t),
io3_i(QSPI_0_io3_i),
io3_o(QSPI_0_io3_o),
io3_t(QSPI_0_io3_t),
ip2intc_irpt(axi_quad_spi_irq),
s_axi_aclk(axi_cpu_clk),
s_axi_araddr(m_axi_qspi_ARADDR[6:0]),
s_axi_aresetn(sys_rstgen_interconnect_aresetn),
s_axi_arready(m_axi_qspi_ARREADY),
s_axi_arvalid(m_axi_qspi_ARVALID),
s_axi_awaddr(m_axi_qspi_AWADDR[6:0]),
s_axi_awready(m_axi_qspi_AWREADY),
s_axi_awvalid(m_axi_qspi_AWVALID),
s_axi_bready(m_axi_qspi_BREADY),
s_axi_bresp(m_axi_qspi_BRESP),
s_axi_bvalid(m_axi_qspi_BVALID),
s_axi_rdata(m_axi_qspi_RDATA),
s_axi_rready(m_axi_qspi_RREADY),
s_axi_rresp(m_axi_qspi_RRESP),
s_axi_rvalid(m_axi_qspi_RVALID),
s_axi_wdata(m_axi_qspi_WDATA),
s_axi_wready(m_axi_qspi_WREADY),
s_axi_wstrb(m_axi_qspi_WSTRB),
s_axi_wvalid(m_axi_qspi_WVALID),
```

```
 ss_i(QSPI_0_ss_i_0),                                                    .
 ss_o(QSPI_0_ss_o_0),                                                    .
 ss_t(QSPI_0_ss_t_0)
)
```

AXI Quad SPI

inst_axi_spix1

```
axi_spix1 inst_axi_spix1 (                                               .
ext_spi_clk(axi_cpu_clk),                                                .
io0_i(spi_io0_i),                                                        .
io0_o(spi_io0_o),                                                        .
io0_t(spi_io0_t),                                                        .
io1_i(spi_io1_i),                                                        .
io1_o(spi_io1_o),                                                        .
io1_t(spi_io1_t),                                                        .
ip2intc_irpt(axi_spi_irq),                                               .
s_axi_aclk(axi_cpu_clk),                                                 .
s_axi_araddr(m_axi_spi_ARADDR[6:0]),                                     .
s_axi_aresetn(sys_rstgen_peripheral_aresetn),                            .
s_axi_arready(m_axi_spi_ARREADY),                                        .
s_axi_arvalid(m_axi_spi_ARVALID),                                        .
s_axi_awaddr(m_axi_spi_AWADDR[6:0]),                                     .
s_axi_awready(m_axi_spi_AWREADY),                                        .
s_axi_awvalid(m_axi_spi_AWVALID),                                        .
s_axi_bready(m_axi_spi_BREADY),                                          .
s_axi_bresp(m_axi_spi_BRESP),                                            .
s_axi_bvalid(m_axi_spi_BVALID),                                          .
s_axi_rdata(m_axi_spi_RDATA),                                            .
s_axi_rready(m_axi_spi_RREADY),                                          .
s_axi_rresp(m_axi_spi_RRESP),                                            .
s_axi_rvalid(m_axi_spi_RVALID),                                          .
s_axi_wdata(m_axi_spi_WDATA),                                            .
s_axi_wready(m_axi_spi_WREADY),                                          .
```

```
 s_axi_wstrb(m_axi_spi_WSTRB),                                                .
 s_axi_wvalid(m_axi_spi_WVALID),                                              .
 sck_i(spi_sck_i),                                                            .
 sck_o(spi_sck_o),                                                            .
 sck_t(spi_sck_t),                                                            .
 ss_i(spi_ss_i),                                                              .
 ss_o(spi_ss_o),                                                              .
 ss_t(spi_ss_t)
)
```

AXI Standard SPI

inst_axi_uart

```
 axi_uart inst_axi_uart (                                                     .
 interrupt(axi_uartlite_irq),                                                 .
 rx(UART_rxd),                                                                .
 s_axi_aclk(axi_cpu_clk),                                                     .
 s_axi_araddr(m_axi_uart_ARADDR[3:0]),                                        .
 s_axi_aresetn(sys_rstgen_peripheral_aresetn),                                .
 s_axi_arready(m_axi_uart_ARREADY),                                           .
 s_axi_arvalid(m_axi_uart_ARVALID),                                           .
 s_axi_awaddr(m_axi_uart_AWADDR[3:0]),                                        .
 s_axi_awready(m_axi_uart_AWREADY),                                           .
 s_axi_awvalid(m_axi_uart_AWVALID),                                           .
 s_axi_bready(m_axi_uart_BREADY),                                             .
 s_axi_bresp(m_axi_uart_BRESP),                                               .
 s_axi_bvalid(m_axi_uart_BVALID),                                             .
 s_axi_rdata(m_axi_uart_RDATA),                                               .
 s_axi_rready(m_axi_uart_RREADY),                                             .
 s_axi_rresp(m_axi_uart_RRESP),                                               .
 s_axi_rvalid(m_axi_uart_RVALID),                                             .
 s_axi_wdata(m_axi_uart_WDATA),                                               .
 s_axi_wready(m_axi_uart_WREADY),                                             .
 s_axi_wstrb(m_axi_uart_WSTRB),                                               .
```

```
 s_axi_wvalid(m_axi_uart_WVALID),                                                .
 tx(UART_txd)
)
```

AXI UART LITE

inst_axi_double_timer

```
 axi_double_timer inst_axi_double_timer (
                                                                                 .
 capturetrig0(1'b0),                                                             .
 capturetrig1(1'b0),                                                             .
 generateout0(),                                                                 .
 generateout1(),                                                                 .
 pwm0(pwm0),                                                                     .
 interrupt(axi_timer_irq),                                                       .
 freeze(1'b0),                                                                   .
 s_axi_aclk(axi_cpu_clk),                                                        .
 s_axi_araddr(m_axi_timer_ARADDR[4:0]),                                          .
 s_axi_aresetn(sys_rstgen_peripheral_aresetn),                                   .
 s_axi_arready(m_axi_timer_ARREADY),                                             .
 s_axi_arvalid(m_axi_timer_ARVALID),                                             .
 s_axi_awaddr(m_axi_timer_AWADDR[4:0]),                                          .
 s_axi_awready(m_axi_timer_AWREADY),                                             .
 s_axi_awvalid(m_axi_timer_AWVALID),                                             .
 s_axi_bready(m_axi_timer_BREADY),                                               .
 s_axi_bresp(m_axi_timer_BRESP),                                                 .
 s_axi_bvalid(m_axi_timer_BVALID),                                               .
 s_axi_rdata(m_axi_timer_RDATA),                                                 .
 s_axi_rready(m_axi_timer_RREADY),                                               .
 s_axi_rresp(m_axi_timer_RRESP),                                                 .
 s_axi_rvalid(m_axi_timer_RVALID),                                               .
 s_axi_wdata(m_axi_timer_WDATA),                                                 .
 s_axi_wready(m_axi_timer_WREADY),                                               .
 s_axi_wstrb(m_axi_timer_WSTRB),                                                 .
 s_axi_wvalid(m_axi_timer_WVALID)
)
```

inst_axi_tft_vga

```
axi_tft_vga inst_axi_tft_vga (
                                                                .
s_axi_aclk(axi_cpu_clk),                                        .
s_axi_aresetn(sys_rstgen_peripheral_aresetn),                   .
m_axi_aclk(axi_ddr_ctrl_ui_clk),                                .
m_axi_aresetn(ddr_rstgen_peripheral_aresetn),                   .
md_error(),                                                     .
ip2intc_irpt(axi_tft_irq),                                      .
m_axi_arready(s_axi_dma_vga_arready),                           .
m_axi_arvalid(s_axi_dma_vga_arvalid),                           .
m_axi_araddr(s_axi_dma_vga_araddr),                             .
m_axi_arlen(s_axi_dma_vga_arlen),                               .
m_axi_arsize(s_axi_dma_vga_arsize),                             .
m_axi_arburst(s_axi_dma_vga_arburst),                           .
m_axi_arprot(s_axi_dma_vga_arprot),                             .
m_axi_arcache(s_axi_dma_vga_arcache),                           .
m_axi_rready(s_axi_dma_vga_rready),                             .
m_axi_rvalid(s_axi_dma_vga_rvalid),                             .
m_axi_rdata(s_axi_dma_vga_rdata),                               .
m_axi_rresp(s_axi_dma_vga_rresp),                               .
m_axi_rlast(s_axi_dma_vga_rlast),                               .
m_axi_awready(s_axi_dma_vga_awready),                           .
m_axi_awvalid(s_axi_dma_vga_awvalid),                           .
m_axi_awaddr(s_axi_dma_vga_awaddr),                             .
m_axi_awlen(s_axi_dma_vga_awlen),                               .
m_axi_awsize(s_axi_dma_vga_awsize),                             .
m_axi_awburst(s_axi_dma_vga_awburst),                           .
m_axi_awprot(s_axi_dma_vga_awprot),                             .
m_axi_awcache(s_axi_dma_vga_awcache),                           .
m_axi_wready(s_axi_dma_vga_wready),                             .
m_axi_wvalid(s_axi_dma_vga_wvalid),                             .
```

```
 m_axi_wdata(s_axi_dma_vga_wdata),
 m_axi_wstrb(s_axi_dma_vga_wstrb),
 m_axi_wlast(s_axi_dma_vga_wlast),
 m_axi_bready(s_axi_dma_vga_bready),
 m_axi_bvalid(s_axi_dma_vga_bvalid),
 m_axi_bresp(s_axi_dma_vga_bresp),
 s_axi_awaddr(m_axi_vga_AWADDR),
 s_axi_awvalid(m_axi_vga_AWVALID),
 s_axi_awready(m_axi_vga_AWREADY),
 s_axi_wdata(m_axi_vga_WDATA),
 s_axi_wstrb(m_axi_vga_WSTRB),
 s_axi_wvalid(m_axi_vga_WVALID),
 s_axi_wready(m_axi_vga_WREADY),
 s_axi_bresp(m_axi_vga_BRESP),
 s_axi_bvalid(m_axi_vga_BVALID),
 s_axi_bready(m_axi_vga_BREADY),
 s_axi_araddr(m_axi_vga_ARADDR),
 s_axi_arvalid(m_axi_vga_ARVALID),
 s_axi_arready(m_axi_vga_ARREADY),
 s_axi_rdata(m_axi_vga_RDATA),
 s_axi_rresp(m_axi_vga_RRESP),
 s_axi_rvalid(m_axi_vga_RVALID),
 s_axi_rready(m_axi_vga_RREADY),
 sys_tft_clk(tft_clk),
 tft_hsync(vga_hsync),
 tft_vsync(vga_vsync),
 tft_de(),
 tft_dps(),
 tft_vga_clk(),
 tft_vga_r(vga_r),
 tft_vga_g(vga_g),
 tft_vga_b(vga_b)
)
```

AXI TFT VGA 640x480

inst_sdio_top

```
 sdio_top #(
                                                                                .
 ADDRESS_WIDTH(32),
                                                                                .
 OPT_DMA(1),
                                                                                .
 AXI_IW(4)
) inst_sdio_top ( .i_clk(axi_cpu_clk), .i_reset(sys_rstgen_peripheral_reset)
```

AXI SD CARD Module (CRAP TO BE REMOVED AND CHANGED)

inst_clk_wiz_1

```
 clk_wiz_1 inst_clk_wiz_1 (
                                                                                .
 clk_in1(sys_clk),
                                                                                .
 clk_out1(axi_cpu_clk),
                                                                                .
 clk_out2(ddr_clk),
                                                                                .
 clk_out3(tft_clk)
)
```

Generate system clocks

inst_ddr_rstgen

```
 ddr_rstgen inst_ddr_rstgen (
                                                                                .
 aux_reset_in(axi_ddr_ctrl_ui_clk_sync_rst),
                                                                                .
 dcm_locked(axi_ddr_ctrl_mmcm_locked),
                                                                                .
 ext_reset_in(sys_rstn & s_axi_dma_axixclk_aresetn),
                                                                                .
 mb_debug_sys_rst(debug_rst),
                                                                                .
 peripheral_reset(ddr_rstgen_peripheral_reset),
                                                                                .
 peripheral_aresetn(ddr_rstgen_peripheral_aresetn),
                                                                                .
 slowest_sync_clk(axi_ddr_ctrl_ui_clk)
)
```

Generate DDR Reset

inst_axixclk

```
 axixclk #(
                                                                                .
```

```
 C_S_AXI_ID_WIDTH(4),                                                                .
 C_S_AXI_DATA_WIDTH(32),                                                             .
 C_S_AXI_ADDR_WIDTH(32),                                                             .
 XCLOCK_FFS(2),                                                                      .
 LGFIFO(5)
) inst_axixclk ( .S_AXI_ACLK(axi_cpu_clk), .S_AXI_ARESETN(sys_rstgen_periphe
```

AXI Cross clock for DMA to DDR Memory

inst_axilite_perf_xbar

```
 axilxbar #(                                                                         .
 C_AXI_DATA_WIDTH(32),                                                               .
 C_AXI_ADDR_WIDTH(32),                                                               .
 NM(1),                                                                              .
 NS(7),                                                                              .
 SLAVE_ADDR({{32'h44A70000},{32'h44A60000},{32'h44A50000},{32'h44A40000},{32
 SLAVE_MASK({{32'hFFFF0000},{32'hFFFF0000},{32'hFFFF0000},{32'hFFFF0000},{32
) inst_axilite_perf_xbar ( .S_AXI_ACLK (axi_cpu_clk), .S_AXI_ARESETN (sys_rs
```

AXI Lite Crossbar for slow speed devices .. sdio, tft vga, double timer, ethernet, spi, qspi, uart, gpio

inst_axi_mem_xbar

```
 axixbar #(                                                                          .
 C_AXI_DATA_WIDTH(32),                                                               .
 C_AXI_ADDR_WIDTH(32),                                                               .
 C_AXI_ID_WIDTH(4),                                                                  .
 NM(3),                                                                              .
 NS(1),                                                                              .
 SLAVE_ADDR({{32'h90000000}}),                                                       .
 SLAVE_MASK({{32'hC0000000}})
) inst_axi_mem_xbar ( .S_AXI_ACLK(axi_ddr_ctrl_ui_clk), .S_AXI_ARESETN(ddr_r
```

AXI Crossbar .. IBUS/DBUS @ 0x90000000 plus 2 dma cores. single Large RAM slave, 0xC0000000 is 1 GB mask (0x40000000 twos compliment).

inst_veronica

```
Veronica inst_veronica (
                                                                                .
io_aclk(axi_cpu_clk),
                                                                                .
io_arst(sys_rstgen_peripheral_reset),
                                                                                .
io_debug_rst(debug_rst),
                                                                                .
io_ddr_clk(axi_ddr_ctrl_ui_clk),
                                                                                .
io_ddr_rst(ddr_rstgen_peripheral_reset),
                                                                                .
io_irq({{128-9{1'b0}}, spio_irq, axi_timer_irq, axi_tft_irq, axi_quad_spi_i
                                                                        `ifdef
_JTAG_IO
                                                                                .
io_jtag_tms(tms),
                                                                                .
io_jtag_tdi(tdi),
                                                                                .
io_jtag_tdo(tdo),
                                                                                .
io_jtag_tck(tck),
endif
                                                                                .
m_axi_acc_araddr(M_AXI_araddr),
                                                                                .
m_axi_acc_arprot(M_AXI_arprot),
                                                                                .
m_axi_acc_arready(M_AXI_arready),
                                                                                .
m_axi_acc_arvalid(M_AXI_arvalid),
                                                                                .
m_axi_acc_awaddr(M_AXI_awaddr),
                                                                                .
m_axi_acc_awprot(M_AXI_awprot),
                                                                                .
m_axi_acc_awready(M_AXI_awready),
                                                                                .
m_axi_acc_awvalid(M_AXI_awvalid),
                                                                                .
m_axi_acc_bready(M_AXI_bready),
                                                                                .
m_axi_acc_bresp(M_AXI_bresp),
                                                                                .
m_axi_acc_bvalid(M_AXI_bvalid),
                                                                                .
m_axi_acc_rdata(M_AXI_rdata),
                                                                                .
m_axi_acc_rready(M_AXI_rready),
                                                                                .
m_axi_acc_rresp(M_AXI_rresp),
                                                                                .
m_axi_acc_rvalid(M_AXI_rvalid),
                                                                                .
m_axi_acc_wdata(M_AXI_wdata),
                                                                                .
m_axi_acc_wready(M_AXI_wready),
                                                                                .
m_axi_acc_wstrb(M_AXI_wstrb),
                                                                                .
m_axi_acc_wvalid(M_AXI_wvalid),
                                                                                .
m_axi_perf_araddr(s_axi_perf_ARADDR),
                                                                                .
```

```
m_axi_perf_arready(s_axi_perf_ARREADY),
m_axi_perf_arvalid(s_axi_perf_ARVALID),
m_axi_perf_arprot(s_axi_perf_ARPROT),
m_axi_perf_awaddr(s_axi_perf_AWADDR),
m_axi_perf_awprot(s_axi_perf_AWPROT),
m_axi_perf_awready(s_axi_perf_AWREADY),
m_axi_perf_awvalid(s_axi_perf_AWVALID),
m_axi_perf_bready(s_axi_perf_BREADY),
m_axi_perf_bresp(s_axi_perf_BRESP),
m_axi_perf_bvalid(s_axi_perf_BVALID),
m_axi_perf_rdata(s_axi_perf_RDATA),
m_axi_perf_rready(s_axi_perf_RREADY),
m_axi_perf_rresp(s_axi_perf_RRESP),
m_axi_perf_rvalid(s_axi_perf_RVALID),
m_axi_perf_wdata(s_axi_perf_WDATA),
m_axi_perf_wready(s_axi_perf_WREADY),
m_axi_perf_wstrb(s_axi_perf_WSTRB),
m_axi_perf_wvalid(s_axi_perf_WVALID),
m_axi_mbus_araddr(s_axi_mbus_ARADDR),
m_axi_mbus_arburst(s_axi_mbus_ARBURST),
m_axi_mbus_arcache(s_axi_mbus_ARCACHE),
m_axi_mbus_arid(s_axi_mbus_ARID),
m_axi_mbus_arlen(s_axi_mbus_ARLEN),
m_axi_mbus_arprot(s_axi_mbus_ARPROT),
m_axi_mbus_arready(s_axi_mbus_ARREADY),
m_axi_mbus_arsize(s_axi_mbus_ARSIZE),
m_axi_mbus_arvalid(s_axi_mbus_ARVALID),
m_axi_mbus_awaddr(s_axi_mbus_AWADDR),
m_axi_mbus_awburst(s_axi_mbus_AWBURST),
m_axi_mbus_awcache(s_axi_mbus_AWCACHE),
m_axi_mbus_awid(s_axi_mbus_AWID),
m_axi_mbus_awlen(s_axi_mbus_AWLEN),
m_axi_mbus_awprot(s_axi_mbus_AWPROT),
```

```
 m_axi_mbus_awready(s_axi_mbus_AWREADY),                                 .
 m_axi_mbus_awsize(s_axi_mbus_AWSIZE),                                   .
 m_axi_mbus_awvalid(s_axi_mbus_AWVALID),                                 .
 m_axi_mbus_bid(s_axi_mbus_BID),                                         .
 m_axi_mbus_bready(s_axi_mbus_BREADY),                                   .
 m_axi_mbus_bvalid(s_axi_mbus_BVALID),                                   .
 m_axi_mbus_rdata(s_axi_mbus_RDATA),                                     .
 m_axi_mbus_rid(s_axi_mbus_RID),                                         .
 m_axi_mbus_rlast(s_axi_mbus_RLAST),                                     .
 m_axi_mbus_rready(s_axi_mbus_RREADY),                                   .
 m_axi_mbus_rvalid(s_axi_mbus_RVALID),                                   .
 m_axi_mbus_wdata(s_axi_mbus_WDATA),                                     .
 m_axi_mbus_wlast(s_axi_mbus_WLAST),                                     .
 m_axi_mbus_wready(s_axi_mbus_WREADY),                                   .
 m_axi_mbus_wstrb(s_axi_mbus_WSTRB),                                     .
 m_axi_mbus_wvalid(s_axi_mbus_WVALID),                                   .
 m_axi_mbus_arqos(s_axi_mbus_ARQOS),                                     .
 m_axi_mbus_arlock(s_axi_mbus_ARLOCK),                                   .
 m_axi_mbus_awqos(s_axi_mbus_AWQOS),                                     .
 m_axi_mbus_awlock(s_axi_mbus_AWLOCK),                                   .
 m_axi_mbus_rresp(s_axi_mbus_RRESP),                                     .
 m_axi_mbus_bresp(s_axi_mbus_BRESP)
)
```

Veronica AXI Vexriscv CPU

inst_sys_rstgen

```
 sys_rstgen inst_sys_rstgen (                                            .
 aux_reset_in(axi_ddr_ctrl_ui_clk_sync_rst),                             .
 dcm_locked(axi_ddr_ctrl_mmcm_locked),                                   .
 ext_reset_in(sys_rstn),                                                 .
 interconnect_aresetn(sys_rstgen_interconnect_aresetn),                  .
 mb_debug_sys_rst(debug_rst),                                            .
 peripheral_reset(sys_rstgen_peripheral_reset),                          .
```

```
 peripheral_aresetn(sys_rstgen_peripheral_aresetn),
 slowest_sync_clk(axi_cpu_clk)
)
```

Generate general system resets

system_wrapper.v

AUTHORS

JAY CONVERTINO

DATES

2024/11/25

INFORMATION

Brief

System wrapper for ps.

License MIT

system_wrapper

```
module system_wrapper (
                                                          `ifdef _JTAG_IO

input
tck,
input
tms,
input
tdi,
output
tdo,
                                                          `endif
```

```
input
clk,
input
resetn,
                                                    inout [
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_cs_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,
                                                    input [
1:0]
eth_rxd,
output
eth_txen,
                                                   output [
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

tck JTAG
`ifdef _JTAG_IO input`

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

vga_b output[3: 0]	vga interface
vga_hs output	vga interface
vga_vs output	vga interface
leds output[15: 0]	board leds
slide_switches input[15: 0]	board slide switches
ftdi_tx input	FTDI UART TX
ftdi_rx output	FTDI UART RX
ftdi_rts input	FTDI UART RTS
ftdi_cts output	FTDI UART CTS
qspi_dq inout[3: 0]	QUAD SPI
qspi_csn output	QUAD SPI
sd_reset output	SD CARD Reset
sd_cd input	SD CARD CD
sd_sck output	SD CARD sck
sd_cmd inout	SD CARD cmd
sd_dat inout[3: 0]	SD CARD dat

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_t
```

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_io0_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.

system_wrapper.v

AUTHORS

JAY CONVERTINO

DATES

2024/11/25

INFORMATION

Brief

System wrapper for ps (INCOMPLETE, CURRENTLY HAS OLD MURAX CODE.. DO NOT USE).

License MIT

system_wrapper

System wrapper for (INCOMPLETE, CURRENTLY HAS OLD MURAX CODE).