wishbone_standard_spi_master.v

AUTHORS

JAY CONVERTINO

DATES

2025/04/30

INFORMATION

Brief

Wishbone Standard SPI Master core.

License MIT

wishbone_standard_spi_master

```
module wishbone_standard_spi_master #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
CLOCK_SPEED
 =
100000000,
parameter
```

```
SELECT_WIDTH
 =
16,
parameter
DEFAULT_RATE_DIV
 =
0,
parameter
DEFAULT_CPOL
 =
0,
parameter
DEFAULT_CPHA
 =
0
) ( input clk, input rst, input s_wb_cyc, input s_wb_stb, input s_wb_we, inp
```

Wishbone Standard based SPI Master device.

Parameters

ADDRESS_WIDTH parameter	Width of the uP address port, max 32 bit.
BUS_WIDTH parameter	Width of the uP bus data port(can not be less than 2 bytes, max tested is 4).
CLOCK_SPEED parameter	This is the aclk frequency in Hz, this is the the frequency used for the bus and is divided by the rate.
SELECT_WIDTH parameter	Bit width of the slave select, defaults to 16 to match altera spi ip.
DEFAULT_RATE_DIV parameter	Default divider value of the main clock to use for the spi data output clock rate. 0 is 2 (2^(X+1) X is the DEFAULT_RATE_DIV)
DEFAULT_CPOL parameter	Default clock polarity for the core (0 or 1).
DEFAULT_CPHA parameter	Default clock phase for the core (0 or 1).

Ports

clk	Clock for all devices in the core
rst	Positive reset
s_wb_cyc	Bus Cycle in process
s_wb_stb	Valid data transfer cycle
s_wb_we	Active High write, low read
s_wb_addr	Bus address
s_wb_data_i	Input data
s_wb_sel	Device Select
s_wb_ack	Bus transaction terminated
s_wb_data_o	Output data
s_wb_err	Active high when a bus error is present
irq	Interrupt when data is received
sclk	spi clock, should only drive output pins to devices.
mosi	transmit for master output
miso	receive for master input
ss_n	slave select output

up_rreq

```
wire up_rreq
```

uP read bus request

up_rack

```
wire up_rack
```

uP read bus acknowledge

up_raddr

```
wire [ADDRESS_WIDTH-(
BUS_WIDTH
                                                                         /

2
)-1:0] up_raddr
```

uP read bus address

up_rdata

```
wire [31:0] up_rdata
```

uP read bus request

up_wreq

```
wire up_wreq
```

uP write bus request

up_wack

```
wire up_wack
```

uP write bus acknowledge

up_waddr

```
wire [ADDRESS_WIDTH-(
BUS_WIDTH
                                                                         /

2
)-1:0] up_waddr
```

uP write bus address

up_wdata

```
wire [31:0] up_wdata
```

uP write bus data

INSTANTIANTED MODULES

inst_up_wishbone_standard

```
up_wishbone_standard #(
                                                                          .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                          .
BUS_WIDTH(BUS_WIDTH)
) inst_up_wishbone_standard ( .clk(clk), .rst(rst), .s_wb_cyc(s_wb_cyc), .s_
```

Module instance of up_wishbone_standard for the Wishbone Classic Standard bus to the uP bus.

inst_up_spi_master

```
up_spi_master #(
                                                                          .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                          .
BUS_WIDTH(BUS_WIDTH),
                                                                          .
CLOCK_SPEED(CLOCK_SPEED),
                                                                          .
SELECT_WIDTH(SELECT_WIDTH),
                                                                          .
DEFAULT_RATE_DIV(DEFAULT_RATE_DIV),
                                                                          .
DEFAULT_CPOL(DEFAULT_CPOL),
                                                                          .
DEFAULT_CPHA(DEFAULT_CPHA)
) inst_up_spi_master ( .clk(clk), .rstn(~rst), .up_rreq(up_rreq), .up_rack(u
```

Module instance of up_spi_master creating a Logic wrapper for spi master axis bus cores to interface with uP bus.