

# up\_spi\_master.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2024/04/29

---

## INFORMATION

---

### Brief

---

uP Core for interfacing with axis spi that emulates the ALTERA SPI IP.

### License MIT

---

Copyright 2025 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.

## up\_spi\_master

---

```
module up_spi_master #(
  parameter
  ADDRESS_WIDTH
  =
  32,
  parameter
  BUS_WIDTH
  =
  4,
  parameter
  WORD_WIDTH
  =
  4,
  parameter
```

```

CLOCK_SPEED
=
100000000,
parameter
SELECT_WIDTH
=
16,
parameter
DEFAULT_RATE_DIV
=
0,
parameter
DEFAULT_CPOL
=
0,
parameter
DEFAULT_CPHA
=
0
) ( input wire clk, input wire rstn, input wire up_rreq, output wire up_rack

```

SPI Master core with axis input/output data. Read/Write is size of BUS\_WIDTH bytes. Write activates core for read.

## Parameters

<b>ADDRESS_WIDTH</b> parameter	Width of the uP address port, max 32 bit.
<b>BUS_WIDTH</b> parameter	Width of the uP bus data port, only valid values are 2 or 4.
<b>WORD_WIDTH</b> parameter	Width of each SPI Master word. This will also set the bits used in the TX/RX data registers. Must be less than or equal to BUS_WIDTH, VALID: 1 to 4.
<b>CLOCK_SPEED</b> parameter	This is the aclk frequency in Hz, this is the the frequency used for the bus and is divided by the rate.
<b>SELECT_WIDTH</b> parameter	Bit width of the slave select, defaults to 16 to match altera spi ip.
<b>DEFAULT_RATE_DIV</b> 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)
<b>DEFAULT_CPOL</b> parameter	Default clock polarity for the core (0 or 1).
<b>DEFAULT_CPHA</b> parameter	Default clock phase for the core (0 or 1).

## Ports

<b>clk</b>	Clock for all devices in the core
<b>rstn</b>	Negative reset
<b>up_rreq</b>	uP bus read request
<b>up_rack</b>	uP bus read ack
<b>up_raddr</b>	uP bus read address
<b>up_rdata</b>	uP bus read data
<b>up_wreq</b>	uP bus write request
<b>up_wack</b>	uP bus write ack
<b>up_waddr</b>	uP bus write address
<b>up_wdata</b>	uP bus write data
<b>irq</b>	Interrupt when data is received
<b>sclk</b>	spi clock, should only drive output pins to devices.

**mosi**            transmit for master output  
**miso**            receive for master input  
**ss\_n**            slave select output

## DIVISOR

```
localparam DIVISOR = BUS_WIDTH/2
```

Divide the address register default location for 1 byte access to multi byte access. (register offsets are byte offsets).

## REG\_SIZE

```
localparam REG_SIZE = 8
```

Number of bits for the register address

## REGISTER INFORMATION

Core has 7 registers at the offsets that follow when at a full 32 bit bus width, Internal address is OFFSET >> BUS\_WIDTH/2 (32bit would be h4 >> 2 = 1 for internal address).

<b>RX_DATA_REG</b>	h00
<b>TX_DATA_REG</b>	h04
<b>STATUS_REG</b>	h08
<b>CONTROL_REG</b>	h0C
<b>RESERVED</b>	h10
<b>SLAVE_SELECT_REG</b>	h14
<b>EOP_VALUE_REG</b>	h18
<b>CONTROL_EXT_REG</b>	h1C

## RX\_DATA\_REG

```
localparam RX_DATA_REG = 8'h0 >> DIVISOR
```

Defines the address offset for RX DATA OUTPUT

RX DATA REGISTER	
31:N	N:0
UNUSED	RECEIVED DATA

Valid bits are from WORD\_WIDTH\*8-1:0, which are data.

## TX\_DATA\_REG

```
localparam TX_DATA_REG = 8'h4 >> DIVISOR
```

Defines the address offset to write the TX DATA INPUT.

TX DATA REGISTER	
31:N	N:0
UNUSED	TRANSMIT DATA

Valid bits are from WORD\_WIDTH\*8-1:0, which are data.

## STATUS\_REG

```
localparam STATUS_REG = 8'h8 >> DIVISOR
```

Defines the address offset to read the status bits.

STATUS REGISTER								
31:10	9	8	7	6	5	4	3	2:0
UNUSED	EOP	E	RRDY	TRDY	TMT	TOE	ROE	UNUSED

### Status Register, 1 is considered active.

<b>EOP</b>	9, This bit is active(1) when the EOP_VALUE_REG is equal to RX_DATA_REG or TX_DATA_REG.
<b>E</b>	8, Logical or of TOE and ROE (Clear by writing status).
<b>RRDY</b>	7, Receive is ready (full) when the bit is 1, empty when the bit is 0.
<b>TRDY</b>	6, Transmit is ready (empty) when the bit is 1, full when the bit is 0.
<b>TMT</b>	5, Transmit shift register empty is set to 1 when all bits have been output.
<b>TOE</b>	4, Transmit overrun is set to 1 when a TX_DATA_REG write happens whne TRDY is 1 (Clear by writing status reg).
<b>ROE</b>	3, Receive overrun is set to 1 when RRDY is 1 and a new received word is going to be written to RX_DATA_REG (Clear by writing status reg)

## CONTROL\_REG

```
localparam CONTROL_REG = 8'hC >> DIVISOR
```

Defines the address offset to set the control bits.

CONTROL REGISTER									
31:11	10	9	8	7	6	5	4	3	2:0
UNUSED	SSO	IEOP	IE	IRRDY	ITRDY	UNUSED	ITOE	IROE	UNUSED

Control Register, 1 is considered active. **All zeros on reset.**

SSO	10, Setting this to 1 will force all ss_n lines to 0 (selected).
IEOP	9, Generate a interrupt on EOP status bit going active if set to 1.
IE	8, Generate a interrupt on ANY error, active if set to 1.
IRRDY	7, Generate a interrupt on RRDY status bit going active if set to 1.
ITRDY	6, Generate a interrupt on TRDY status bit going active if set to 1.
ITOE	4, Generate a interrupt on TOE status bit going active if set to 1.
IROE	3, Generate a interrupt on ROE status bit going active if set to 1.

## RESERVED

```
localparam RESERVED = 8'h10 >> DIVISOR
```

Defines the address offset that is not used.

## SLAVE\_SELECT\_REG

```
localparam SLAVE_SELECT_REG = 8'h14 >> DIVISOR
```

Defines the address offset to set the slave select value

SLAVE SELECT REGISTER	
31:N	N:0
UNUSED	SLAVE SELECT

Valid bits are from SELECT\_WIDTH-1:0, which are the slave select output lines to drive low during data transmission.

## EOP\_VALUE\_REG

```
localparam EOP_VALUE_REG = 8'h18 >> DIVISOR
```

Defines the address offset to set the end of packet match value

EOP REGISTER	
31:N	N:0
UNUSED	EOP

Valid bits are from BUS\_WIDTH\*8:0, which are used to check for a word match between rx and/or tx and

update status.

### CONTROL\_EXT\_REG

```
localparam CONTROL_EXT_REG = 8'h1C >> DIVISOR
```

Defines the address offset for control register extensions

CONTROL REGISTER EXTENDED			
31:6	5	4	3:0
UNUSED	CPHA	CPOL	RATE DIV

### Control Extension to add capabilities to Altera IP core.

CPHA	5, Clock Phase Bit, 0 or 1 per SPI specs (default value set by IP parameter).
CPOL	4, Clock Polarity bit, 0 or 1 per SPI specs (default value set by IP parameter).
RATE_TOP	3, Top bit for rate control. Divider values are 0 to 15 ( $2^X+1$ where X is the divider value).
RATE_BOT	0, Bottom bit for rate control.

### INSTANTIATED MODULES

#### inst\_axis\_spi

```
axis_spi_master #(
    CLOCK_SPEED(CLOCK_SPEED),
    BUS_WIDTH(WORD_WIDTH),
    SELECT_WIDTH(SELECT_WIDTH)
) inst_axis_spi_master ( .aclk(clk), .arstn(rstn), .s_axis_tdata(r_tx_wdata
```

SPI Master instance with AXIS interface