

up_1553.v

AUTHORS

JAY CONVERTINO

DATES

2024/10/17

INFORMATION

Brief

uP Core for interfacing with simple 1553 communications.

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.

up_1553

```
module up_1553 #(
  parameter
  ADDRESS_WIDTH
  =
  32,
  parameter
  BUS_WIDTH
  =
  4,
  parameter
  CLOCK_SPEED
  =
  100000000
)
```

```

    input
    clk,
    input
    rstn,
    input
    up_rreq,
    output
    up_rack,
    input
    [ADDRESS_WIDTH-(BUS_WIDTH/2)-1:0]
    up_raddr,
    output
    [(BUS_WIDTH*8)-1:0]
    up_rdata,
    input
    up_wreq,
    output
    up_wack,
    input
    [ADDRESS_WIDTH-(BUS_WIDTH/2)-1:0]
    up_waddr,
    input
    [(BUS_WIDTH*8)-1:0]
    up_wdata,
    input
    [1:0]
    rx_diff,
    output
    [1:0]
    tx_diff,
    output
    tx_active,
    output
    irq
)

```

uP based 1553 communications device.

Parameters

ADDRESS_WIDTH parameter	Width of the uP address port, max 32 bit.
BUS_WIDTH parameter	Width of the uP bus data port.
CLOCK_SPEED parameter	This is the ack frequency in Hz

Ports

clk input	Clock for all devices in the core
rstn input	Negative reset
up_rreq input	uP bus read request
up_rack output	uP bus read ack
up_raddr input [ADDRESS_WIDTH-(BUS_WIDTH/ 2)- 1:0]	uP bus read address
up_rdata output [(BUS_WIDTH* 8)- 1:0]	uP bus read data

up_wreq <i>input</i> [(BUS_WIDTH* 8)- 1:0]	uP bus write request
up_wack <i>output</i> [(BUS_WIDTH* 8)- 1:0]	uP bus write ack
up_waddr <i>input</i> [ADDRESS_WIDTH-(BUS_WIDTH/ 2)- 1:0]	uP bus write address
up_wdata <i>input</i> [(BUS_WIDTH* 8)- 1:0]	uP bus write data
rx_diff <i>input</i> [1:0]	Input differential signal for 1553 bus
tx_diff <i>output</i> [1:0]	Output differential signal for 1553 bus
tx_active <i>output</i> [1:0]	Enable output of differential signal as this indicates tx is active (for signal switching on 1553 module)
irq <i>output</i> [1:0]	Interrupt when data is received

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).

FIFO_DEPTH

```
localparam FIFO_DEPTH = 16
```

Depth of the fifo, matches UART LITE (xilinx), so I kept this just cause

DATA_BITS

```
localparam DATA_BITS = 21
```

Number of bits in RX/TX FIFO that are valid.

REGISTER INFORMATION

Core has 4 registers at the offsets that follow.

RX_FIFO_REG	h0
TX_FIFO_REG	h4
STATUS_REG	h8
CONTROL_REG	hC

RX_FIFO_REG

```
localparam RX_FIFO_REG = 4'h0 >> DIVISOR
```

Defines the address offset for RX FIFO

RX FIFO REGISTER		
31:20	20:16	15:0
UNUSED	STATUS DATA	RECEIVED DATA

Valid bits are from 20:0. Bits 20:16 are status bits information about the data. Bit 15:0 are data.

Status Bits

{S:1,D:1,TY:3}

TY Type is 3 bits, 000 NA, 001 = REG, 010 = DATA, 100 = CMD/STATUS

D Delay Enabled is 1 bit, 1 is there was be a delay of 4 us or more, or 0 no delay.

S Sync only when 1, normal is 0

TX_FIFO_REG

```
localparam TX_FIFO_REG = 4'h4 >> DIVISOR
```

Defines the address offset to write the TX FIFO.

TX FIFO REGISTER		
31:20	20:16	15:0
UNUSED	STATUS DATA	TRANSMIT DATA

Valid bits are from 20:0. Bits 20:16 are status bits information about the data. Bit 15:0 are data.

Status Bits

{S:1,D:1,TY:3}

TY Type is 3 bits, 000 NA, 001 = REG, 010 = DATA, 100 = CMD/STATUS

D Delay Enabled is 1 bit, 1 is there must be a delay of 4 us or more, or 0 no delay.

S Sync only when 1, normal is 0

STATUS_REG

```
localparam STATUS_REG = 4'h8 >> DIVISOR
```

Defines the address offset to read the status bits.

STATUS REGISTER							
31:7	6	5	4	3	2	1	0
UNUSED	Parity_err	frame_err	irq_en	tx_full	tx_empty	rx_full	rx_valid

Status Register Bits

parity_err 6, When 1 an error in the RX parity check has occurred

frame_err 5, When 1 an error in the RX frame has occurred (manchester data 2'b11 or 2'b00).
irq_en 4, When 1 the IRQ is enabled by **CONTROL_REG**
tx_full 3, When 1 the tx fifo is full.
tx_empty 2, When 1 the tx fifo is empty.
rx_full 1, When 1 the rx fifo is full.
rx_valid 0, When 1 the rx fifo contains valid data.

CONTROL_REG

localparam CONTROL_REG = 4'hc

Defines the address offset to set the control bits.

CONTROL REGISTER				
31:5	4	3:2	1	0
UNUSED	ENA_INTR_BIT	UNUSED	RST_RX_BIT	RST_TX_BIT

See Also: **ENABLE_INTR_BIT**, **RESET_RX_BIT**, **RESET_TX_BIT**

Control Register Bits

ENABLE_INTR_BIT 4, Control Register offset bit for enabling the interrupt.
RESET_RX_BIT 1, Control Register offset bit for resetting the RX FIFO.
RESET_TX_BIT 0, Control Register offset bit for resetting the TX FIFO.

inst_axis_1553

Decode/Encode differential 1553 data stream to AXIS data format.

inst_rx_fifo

Buffer up to 16 items output from the axis_1553_encoder.

inst_tx_fifo

Buffer up to 16 items to input to the axis_1553_decoder.