axi_lite_1553.v

AUTHORS

JAY CONVERTINO

DATES

2024/10/17

INFORMATION

Brief

AXI Lite 1553 is a core for interfacing with 1553 devices over the AXI lite bus.

License MIT

axi_lite_1553

```
module axi_lite_1553 #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
CLOCK_SPEED
 =
100000000,
parameter
SAMPLE_RATE
 =
2000000,
parameter
```

```
 BIT_SLICE_OFFSET
  =
 0,
 parameter
 INVERT_DATA
  =
 0,
 parameter
 SAMPLE_SELECT
  =
 0
) ( input aclk, input arstn, input s_axi_awvalid, input [ADDRESS_WIDTH-1:0]
```

AXI Lite based 1553 communications device.

Parameters

ADDRESS_WIDTH parameter	Width of the axi address bus, max 32 bit.
CLOCK_SPEED parameter	This is the aclk frequency in Hz
SAMPLE_RATE parameter	Rate of in which to sample the 1553 bus. Must be 2 MHz or more and less than aclk. This is in Hz. BIT_SLICE_OFFSET- Adjust where the sample is taken from the input.
INVERT_DATA parameter	Invert all 1553 bits coming in and out.
SAMPLE_SELECT parameter	Adjust where in the array of samples to select a bit.

Ports

aclk	Clock for all devices in the core
arstn	Negative reset
s_axi_awvalid	Axi Lite aw valid
s_axi_awaddr	Axi Lite aw addr
s_axi_awprot	Axi Lite aw prot
s_axi_awready	Axi Lite aw ready
s_axi_wvalid	Axi Lite w valid
s_axi_wdata	Axi Lite w data
s_axi_wstrb	Axi Lite w strb
s_axi_wready	Axi Lite w ready
s_axi_bvalid	Axi Lite b valid
s_axi_bresp	Axi Lite b resp
s_axi_bready	Axi Lite b ready
s_axi_arvalid	Axi Lite ar valid
s_axi_araddr	Axi Lite ar addr
s_axi_arprot	Axi Lite ar prot
s_axi_arready	Axi Lite ar ready
s_axi_rvalid	Axi Lite r valid
s_axi_rdata	Axi Lite r data
s_axi_rresp	Axi Lite r resp
s_axi_rready	Axi Lite r ready
i_diff	Input differential signal for 1553 bus

o_diff	Output differential signal for 1553 bus
en_o_diff	Enable output of differential signal (for signal switching on 1553 module)
irq	Interrupt when data is received

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_axi

```
up_axi #(
                                                                                .
AXI_ADDRESS_WIDTH(ADDRESS_WIDTH)
) inst_up_axi ( .up_rstn (arstn), .up_clk (aclk), .up_axi_awvalid(s_axi_aw
```

Module instance of up_axi for the AXI Lite bus to the uP bus.

inst_up_1553

```
up_1553 #(
                                                                                .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                                .
CLOCK_SPEED(CLOCK_SPEED),
                                                                                .
SAMPLE_RATE(SAMPLE_RATE),
                                                                                .
BIT_SLICE_OFFSET(BIT_SLICE_OFFSET),
                                                                                .
INVERT_DATA(INVERT_DATA),
                                                                                .
SAMPLE_SELECT(SAMPLE_SELECT)
) inst_up_1553 ( .clk(aclk), .rstn(arstn), .up_rreq(up_rreq), .up_rack(up_ra
```

Module instance of up_1553 creating a Logic wrapper for 1553 bus cores to interface with uP bus.