wishbone_classic_1553.v

AUTHORS

JAY CONVERTINO

DATES

2024/10/17

INFORMATION

Brief

wishbone classic to uP core for 1553 comms.

License MIT

wishbone_standard_1553

```
module wishbone_standard_1553 #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
CLOCK_SPEED
 =
100000000,
parameter
```

```
SAMPLE_RATE
 =
2000000,
parameter
BIT_SLICE_OFFSET
 =
0,
parameter
INVERT_DATA
 =
0,
parameter
SAMPLE_SELECT
 =
0
) ( input clk, input rst, input s_wb_cyc, input s_wb_stb, input s_wb_we, inp
```

Wishbone Stanard based 1553 communications device.

Parameters

ADDRESS_WIDTH parameter
Width of the address bus in bits, max 32 bit.

BUS_WIDTH parameter
Width of the data bus in bytes.

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

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
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_wishbone_standard

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

inst_up_1553

```
 up_1553 #(
                                                                          .
 ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                          .
 BUS_WIDTH(BUS_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(clk), .rstn(~rst), .up_rreq(up_rreq), .up_rack(up_rack
```

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