wishbone_classic_gpio.v

AUTHORS

JAY CONVERTINO

DATES

2024/07/25

INFORMATION

Brief

Wishbone classic UART core.

License MIT

wishbone_classic_gpio

```
module wishbone_classic_gpio #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
GPIO_WIDTH
```

```
  =
 32,
 parameter
 IRQ_ENABLE
  =
 0
) ( input clk, input rst, input s_wb_cyc, input s_wb_stb, input s_wb_we, inp
```

AXI Lite based uart device.

Parameters

ADDRESS_WIDTH Width of the address bus in bits.
parameter

BUS_WIDTH Width of the data bus in bytes.
parameter

GPIO_WIDTH Width of the GPIO for inputs and outputs
parameter

IRQ_ENABLE Enable interrupt
parameter

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_bte Burst Type Extension

s_wb_cti Cycle Type

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

gpio_io_i Input for GPIO

gpio_io_o Output for GPIO

gpio_io_t Tristate for GPIO

up_rreq

```
wire up_rreq
```

uP read bus request

up_rack

```
wire up_rack
```

uP read bus acknowledge

up_raddr

```
wire [ADDRESS_WIDTH-3:0] 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-3:0] up_waddr
```

uP write bus address

up_wdata

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

uP write bus data

INSTANTIANTED MODULES

inst_up_wishbone_classic

```
up_wishbone_classic #(
                                                                                .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                                .
BUS_WIDTH(BUS_WIDTH)
) inst_up_wishbone_classic ( .clk(clk), .rst(rst), .s_wb_cyc(s_wb_cyc), .s_w
```

Module instance of up_wishbone_classic for the Wishbone Classic bus to the uP bus.

inst_up_gpio

```
up_gpio #(
                                                                                .
ADDRESS_WIDTH(ADDRESS_WIDTH),
                                                                                .
BUS_WIDTH(BUS_WIDTH),
                                                                                .
GPIO_WIDTH(GPIO_WIDTH),
                                                                                .
IRQ_ENABLE(IRQ_ENABLE)
) inst_up_gpio ( .clk(aclk), .rstn(arstn), .up_rreq(up_rreq), .up_rack(up_ra
```

Module instance of up_gpio.