axi_lite_block_ram.v

AUTHORS

JAY CONVERTINO

DATES

2024/03/07

INFORMATION

Brief

axi lite block ram

License MIT

axi_lite_block_ram

```
module axi_lite_block_ram #(
parameter
ADDRESS_WIDTH
 =
32,
parameter
BUS_WIDTH
 =
4,
parameter
DEPTH
```

```
  =
 512,
 parameter
 RAM_TYPE
  =
 "block",
 parameter
 HEX_FILE
  =
 ""
) ( input aclk, input arstn, input s_axi_awvalid, input [ADDRESS_WIDTH-1:0]
```

axi lite block ram

Parameters

ADDRESS_WIDTH	parameter	Width of the axi address bus in bits.
BUS_WIDTH	parameter	Bus width for data paths in bytes.
DEPTH	parameter	Depth of the RAM in terms of data width words.
RAM_TYPE	parameter	Used to set the ram_style atribute.
HEX_FILE	parameter	Hex file to write to RAM.

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

c_PWR_RAM

```
localparam c_PWR_RAM = clogb2(
DEPTH
)
```

power of 2 conversion of DEPTH

c_RAM_DEPTH

```
localparam c_RAM_DEPTH = 2 ** c_PWR_RAM
```

create RAM depth based on power of two depth size.

up_rreq

```
wire up_rreq
```

uP read bus request

up_rack

```
reg up_rack
```

uP read bus acknowledge

up_raddr

```
wire [ADDRESS_WIDTH-(
ADDRESS_WIDTH
                                                                    /
16
)-1:0] up_raddr
```

uP read bus address

up_rdata

```
wire [(
BUS_WIDTH*8
)-1:0] up_rdata
```

uP read bus request

up_wreq

```
wire up_wreq
```

uP write bus request

up_wack

```
reg up_wack
```

uP write bus acknowledge

up_waddr

```
wire [ADDRESS_WIDTH-(
ADDRESS_WIDTH
                                                                    /
16
)-1:0] up_waddr
```

uP write bus address

up_wdata

```
wire [(
BUS_WIDTH*8
)-1:0] 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_awv
```

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

inst_dc_block_ram

```
dc_block_ram #(
                                                                    .
RAM_DEPTH(c_RAM_DEPTH),
                                                                    .
BYTE_WIDTH(BUS_WIDTH),
                                                                    .
ADDR_WIDTH(c_PWR_RAM),
                                                                    .
HEX_FILE(HEX_FILE),
                                                                    .
RAM_TYPE(RAM_TYPE)
```

```
) inst_dc_block_ram ( .rd_clk(aclk), .rd_rstn(arstn), .rd_en(up_rreq), .rd_
```

Module instance of dc_block_ram that connects to the uP BUS directly.