

# dc\_block\_ram.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2024/03/07

---

## INFORMATION

---

### Brief

---

Generic Dual Port RAM

### 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.

## dc\_block\_ram

---

```
module dc_block_ram #(
  parameter
  RAM_DEPTH
  =
  1,
  parameter
  BYTE_WIDTH
  =
  1,
  parameter
  ADDR_WIDTH
```

```

    =
    1,
    parameter
    HEX_FILE
    =
    " "
    /
    parameter
    RAM_TYPE
    =
    "block"
) ( input rd_clk, input rd_rstn, input rd_en, output [(BYTE_WIDTH*8)-1:0]

```

Generic Dual Port RAM

## Parameters

<b>RAM_DEPTH</b> <small>parameter</small>	Number of words using the size of BYTE_WIDTH.
<b>BYTE_WIDTH</b> <small>parameter</small>	Width of the data bus in bytes.
<b>ADDR_WIDTH</b> <small>parameter</small>	Width of the address bus in bits.
<b>HEX_FILE</b> <small>parameter</small>	Read a hex value text file as the initial state of the RAM.
<b>RAM_TYPE</b> <small>parameter</small>	Used to set the ram_style attribute.

## Ports

<b>rd_clk</b>	Read clock positive edge
<b>rd_rstn</b>	Read reset active low
<b>rd_en</b>	Read enable active high
<b>rd_data</b>	Read data output
<b>rd_addr</b>	Read data address select
<b>wr_clk</b>	Write clock positive edge
<b>wr_rstn</b>	Write reset active low
<b>wr_en</b>	Write enable active high
<b>wr_ben</b>	Write byte enable, each bit represents one byte of write data.
<b>wr_data</b>	Write data input
<b>wr_addr</b>	Write data address select