

# tb\_dc\_block\_ram.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/01/17

---

## INFORMATION

---

### Brief

---

Test bench for Generic Dual Port RAM

### License MIT

---

Copyright 2025 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 tb_dc_block_ram #(
  parameter
  RAM_DEPTH
  =
  256,
  parameter
  BYTE_WIDTH
  =
  4,
  parameter
  ADDR_WIDTH
  =
  32,
  parameter
```

```

    HEX_FILE
    =
    ""
    parameter
    RAM_TYPE
    =
    "block"
    )()

```

Test bench for Generic Dual Port RAM

## Parameters

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

## INSTANTIATED MODULES

---

### clk\_stim

---

```

clk_stimulus #(
    CLOCKS(1),
    CLOCK_BASE(1000000),
    RESETS(1),
    RESET_BASE(2000)
) clk_stim ( .clkv(tb_dut_clk), .rstnv(tb_dut_rstn), .rstv() )

```

Generate a 50/50 duty cycle set of clocks and reset.

### inst\_dc\_block\_ram

---

```

dc_block_ram #(
    RAM_DEPTH(RAM_DEPTH),
    BYTE_WIDTH(BYTE_WIDTH),
    ADDR_WIDTH(ADDR_WIDTH),
    HEX_FILE(HEX_FILE),
    RAM_TYPE(RAM_TYPE)
) inst_dc_block_ram ( .rd_clk(tb_dut_clk), .rd_rstn(tb_dut_rstn), .rd_en(tb_

```

Module instance of dc\_block\_ram