*Project: RMC75E FPGA TEST BENCH*

*Module: module.vhd*

*Author: Satchel Hamilton*

*Company: Delta Motion*

*Date: 7/13/2023*

*Last updated: July 13, 2023*

Contents

[High Level 1](#_Toc141278917)

[Low level 1](#_Toc141278918)

[Simulation 2](#_Toc141278919)

# High Level

**Description:** The RAM128x16 module creates a 128x16-bit random access memory (RAM) capable of storing and retrieving data based on provided address and control signals. It serves as a reliable and efficient means of storing and retrieving data within the RMC75E modular motion controller, making it an integral component for various motion control operations.

# Low level

**Design Details:** The RAM128x16 module consists of the following components:

**Inputs:**

* clk: The clock signal used for synchronous operations.
* we: The write enable signal that controls write operations.
* a: The address input signal used to specify the memory location to access.
* d: The data input signal containing the 16-bit data to be written into the specified memory location.

**Outputs:**

* o: The data output signal that provides the 16-bit data read from the specified memory location.

**Architecture:** The architecture of the RAM128x16 module, named RAM128x16\_arch, utilizes an array type named ram\_type. The ram\_type represents an array of 128 elements, where each element is a 16-bit std\_logic\_vector, serving as the main storage for the RAM module. The RAM128x16 module includes the following components:

* RAM: A signal of type ram\_type, representing the actual memory storage. It is an array with 128 elements, each capable of storing a 16-bit value. The initial value of the RAM signal is set to all '1's.
* read\_a: A signal of type std\_logic\_vector(6 downto 0), used to hold the current address input for read operations.

The behavior of the RAM128x16 module is defined within a process block sensitive to the clk signal. The process handles both write and read operations based on the rising edge of the clock signal.

During a rising edge of the clock, the module checks if we (write enable) is asserted. If so, the module writes the 16-bit data d into the memory location specified by the address a. The address a is converted to an integer index for RAM access.

The read\_a signal is continuously updated with the current address a to ensure the correct data is read from the RAM during subsequent clock cycles.

Finally, the output signal o is assigned the value stored in the RAM at the memory location specified by read\_a, providing the requested 16-bit data output.

**Initialization Note:** The RAM signal should have a value initialization to avoid undefined behavior in the analog module. The RAM128x16 module sets the initial value of the RAM signal to all '1's. This initialization ensures that the RAM module is not left undefined when initialized, allowing it to operate correctly within the modular motion controller.

## Simulation

The RAM128x16 module will be extensively tested using a ModelSim testbench. The testbench will provide various addresses and data inputs to verify the correct functionality of the RAM module. It will test read and write operations, as well as check that the output o correctly reflects the data stored at the specified memory location. The simulation results will be analyzed to ensure that the RAM128x16 module operates correctly, meeting the requirements of a reliable and efficient RAM component.