Single-Port RAM Verification

Aurhor: **Ammar Wahidi**

Undergraduate Student, Electronics and Communication Engineering

Ain Shams University

Email: [aammarahmed123@gmail.com](mailto:aammarahmed123@gmail.com)

*Date: July 10, 2025*

*Revision number: V.2*

# Introduction

This document presents the RTL verification of a single-port RAM IP. It provides a detailed overview of the IP design, the verification strategy employed, the test items and corresponding test case table, as well as coverage results. Additionally, the document includes a list of bugs that were discovered during the verification process.

# IP Design details

A computer program with a memory card

AI-generated content may be incorrect.

## Input

|  |  |
| --- | --- |
| Signal | Description |
| clk | System clock |
| rst | Active-High reset, When asserted, it resets Data\_out to first element in the memory. |
| en | High: Write enable, Low: Read enable |
| Data\_in | 32-bit data input to be written into memory. |
| address | 4-bit address used for both read and write operations. |

## Output

|  |  |
| --- | --- |
| Signal | Description |
| Data\_out | 32-bit data output from memory during a read operation. |
| valid\_out | High: Data\_out is valid,Low: Data\_out is garbage. |

* This memory module exhibits **1-cycle read latency** and **0-cycle write latency**.

# Verification Strategy

# Exit Criteria

# Test Items

### Items shall be:

* Memory Reset
* “reset” Set to 1: Memory access defaults to the first element (address 0).
* “reset” Set to 0: Memory access should follow the input address (address).
* Memory enable
* “en” Set to 1: Perform Write operation — write “Data\_in” to “memory[addr]”.
* “en” Set to 0: Perform Read operation — read from “memory[addr]” to “data\_out”.
* Memory Address Access
* “address” between 1 to 15.
* “address” is 0.
* “address” is undefined (x) or high-impedance (z) values.
* The “address” input should correctly access the intended memory location.
* Memory Data
* “Data\_in” is a valid (Random) value.
* “Data\_in” Maximum Value.
* “Data\_in” Minimum Value (0) .
* “Data\_in” is undefined (x) or high-impedance (z) values.

# Test Cases

|  |  |
| --- | --- |
| Test | Description |
| Test01 | * Memory Reset/enable/Data input is random. |
| Test02 | * Memory Reset/disable. |
| Test03 | * Memory Reset/enable/Data input is Minimum. * Memory Reset /disable. |
| Test04 | * Memory Reset/enable/ address indicates to Random/Data input is Minimum. * Memory Set /disable/ address indicates to 0. |
| Test05 | * Memory Reset/enable/Data input is random value. * Memory Set /disable/ address indicates to 0. |
| Test06 | * Memory Reset/enable/Data input is undefined or high impedance. * Memory Set /disable/ address indicates to 0. |
| Test07 | * Memory Set/enable/Adress indicates to random value between 0 to 15/Data input is random. * Memory Set /disable/ Adress indicates to the same random value between 0 to 15. |
| Test08 | * Memory Set/enable/Adress indicates to random value between 0 to 15/Data input is Maximum. * Memory Set /disable/ Adress indicates to the same random value between 0 to 15. |
| Test09 | * Memory Set/enable/Adress indicates to random value between 0 to 15/Data input is Minimum. * Memory Set /disable/ Adress indicates to the same random value between 0 to 15. |
| Test10 | * Memory Set/disable/Adress indicates to random value between 0 to 15/Data input is random. * Memory Set/disable/Adress indicates to random value between 0 to 15/Data input is random. |
| Test11 | * Memory Set/enable/Adress indicates to random value between 0 to 15/Data input is random. * Memory Set/enable/Adress indicates to random value between 0 to 15/Data input is random. |
| Test12 | * Memory Set/enable/Adress indicates to random value between 0 to 15/ Data input is undefined or high impedance. |
| Test13 | * Memory Set/disable/Adress undefined (x) or high-impedance (z). |
| Test14 | * Memory Set/enable/Adress undefined (x) or high-impedance (z)/ Data input is undefined or high impedance. |

# Coverage Results

# Opened Issues

# Feature Assessment