First in first out (FIFO)

VERSION 1.0

Revision History

| Revision | Description | Date | Modified By |
| --- | --- | --- | --- |
| 1.0 | Initial Document | 17/04/2023 |  |
| 1.1 | -- | -- | -- |

Table of Contents

[Chapter 1: RAM Overview 6](#_Toc406486740)

[Chapter 2: RAM features 8](#_Toc406486741)

[Chapter 3: RAM Verification Plan 9](#_Toc406486744)

[3.1 Feature Extraction 9](#_Toc406486745)

[3.2 Coverage Plan 9](#_Toc406486746)

[3.3 Checker Plan 9](#_Toc406486747)

[3.4 Verification Environment Development 9](#_Toc406486747)

[3.5 Test suite development 9](#_Toc406486750)

[3.5.1 Directed Testcases 9](#_Toc406486751)

[3.5.2 Random Testcases 10](#_Toc406486752)

[Chapter 4: RAM Verification Environment Development 11](#_Toc406486753)

[4.1 Block Diagram 11](#_Toc406486754)

[4.2 Verification Architecture 11](#_Toc406486754)

[4.3 RAM components 12](#_Toc406486755)

[4.3.1 Transcation Class 12](#_Toc406486756)

[4.3.2 Generator Class 12](#_Toc406486757)

[4.3.3 Driver 13](#_Toc406486758)

[4.3.4 Monitor 14](#_Toc406486759)

[4.3.5 Reference Model 16](#_Toc406486760)

[4.3.6 Score Board 16](#_Toc406486761)

[Chapter 5: Running Simulation 24](#_Toc406486772)

[Chapter 6: Closure Reports 24](#_Toc406486772)

List of Figures

List of Tables

**No table of figures entries found.**

# RAM Overview

RAM, or Random Access Memory, is a type of computer memory that temporarily stores data and instructions for the CPU. It is fast, volatile, and upgradeable, and plays a critical role in system performance. It comes in various capacities and speeds, and is managed by the operating system.

**Why we use RAM ?**

* **Purpose**: RAM is used to temporarily store data and instructions that are needed by the CPU for processing. It serves as a high-speed working area where the CPU can quickly access and modify data as needed during the operation of a computer.
* **Capacity**: RAM comes in various capacities, typically measured in gigabytes (GB) or terabytes (TB). The capacity of RAM determines how much data can be stored and accessed by the CPU at any given time. More RAM allows for more data to be stored, which can improve the performance of a computer, especially when running memory-intensive applications or multitasking.
* **Speed**: RAM operates at a much faster speed compared to other types of computer memory, such as storage drives (e.g., hard disk drives or SSDs). The speed of RAM is usually measured in megahertz (MHz) or gigahertz (GHz) and determines how quickly data can be accessed by the CPU. Higher speed RAM can lead to faster data retrieval and processing, resulting in improved system performance.
* **Types**: There are different types of RAM, including DDR4, DDR3, DDR2, and DDR1, which represent different generations of technology. DDR4 is currently the most commonly used type of RAM in modern computers, offering faster speeds and higher capacities compared to older DDR3 and DDR2 RAM.
* **Upgradability**: RAM is typically upgradeable, which means that users can add or replace RAM modules to increase the amount of memory available in their computers. Upgrading RAM can result in improved system performance, especially for computers that are running memory-intensive tasks or have limited RAM capacity.
* **Memory Modules:** RAM is typically installed in the form of memory modules, which are small circuit boards that plug into slots on the computer's motherboard. Common types of memory modules include DIMM (Dual In-Line Memory Module) and SODIMM (Small Outline Dual In-Line Memory Module), which differ in size and are used in different types of computers (e.g., desktops vs. laptops).
* **Operating System Interaction:** The operating system (OS) of a computer manages the allocation of RAM to different applications and processes running on the computer. The OS decides how much RAM is allocated to each process and manages the swapping of data between RAM and storage drives as needed. Sufficient RAM is essential for smooth multitasking and efficient performance of the operating system.

**RAM Block Diagram**

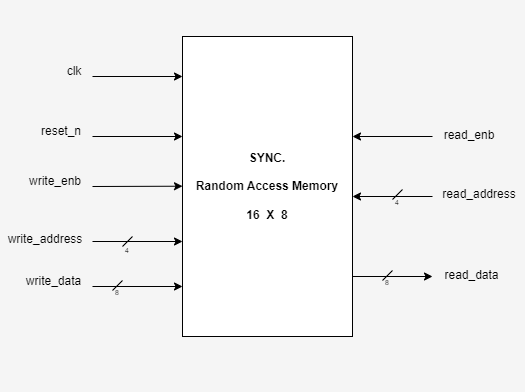


Figure 1.1: RAM Block Diagram

# Asynchronous FIFO features

**The key features of the Dual Port RAM are:**

* + Two independent ports
  + Asynchronous operation
  + Simultaneous read and write
  + High-speed access
  + Large memory capacity
  + Configurable organization
  + Low power consumption

# RAM Verification Plan

Verification involves studying the relevant specifications, extracting features from it that are to be tested, devising a strategy as to how these features are to be tested, developing a verification environment based on the strategy, writing testcases to cover all the scenarios and achieving 100% functional coverage figures.

## Feature Extraction

“Feature Extraction” involves listing out features to be tested from the specification. A feature list (spreadsheets) has been prepared using the ram\_feature\_list document.

ram\_feature\_list.xlsx

The above spreadsheet lists out features to be tested in the corresponding specification, assigns a feature id to them and tells how the feature is to be tested (testcase name or checker task name).

## Coverage Plan

A functional coverage plan needs to be made based on the feature extraction document. This plan lists out the various combinations of stimuli that need to be generated for the proper verification of the UVC. This has been included in the feature extraction spreadsheet itself.

## Checker Plan

A checker plan needs to be made based on the feature extraction document. Implementations for the features marked as “checker” are elaborated here. We have included it in the feature extraction spreadsheet itself.

## Verification Environment Development

Our environment is based on SystemVerilog.

## Test suite development

### Directed Testcases

Testcases written to test specific areas of the UVC or to generate a specific kind or sequence of transactions is known as a directed testcase. These testcases are helpful in the initial and final stages of verification. In the initial stages, when neither the verification environment nor the UVC is matured, these testcases help in checking and correcting specific pieces of code in both the UVC and the verification environment. In the final stages, they are used to hit specific functional or code coverage areas.

### Random Testcases

Random testcases are written to test the UVC extensively. Random scenarios are generated based on constraints provided in the testcase. These testcases are run several times with different seed numbers to generate different scenarios to achieve more functional coverage figure.

# RAM Verification Environment Development

## RAM Block Diagram

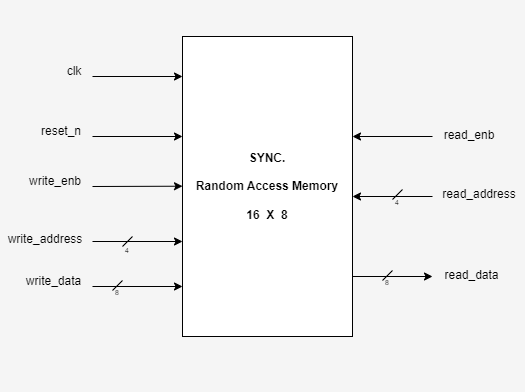


Figure 4.1: RAM Block Diagram

## Verification Architecture

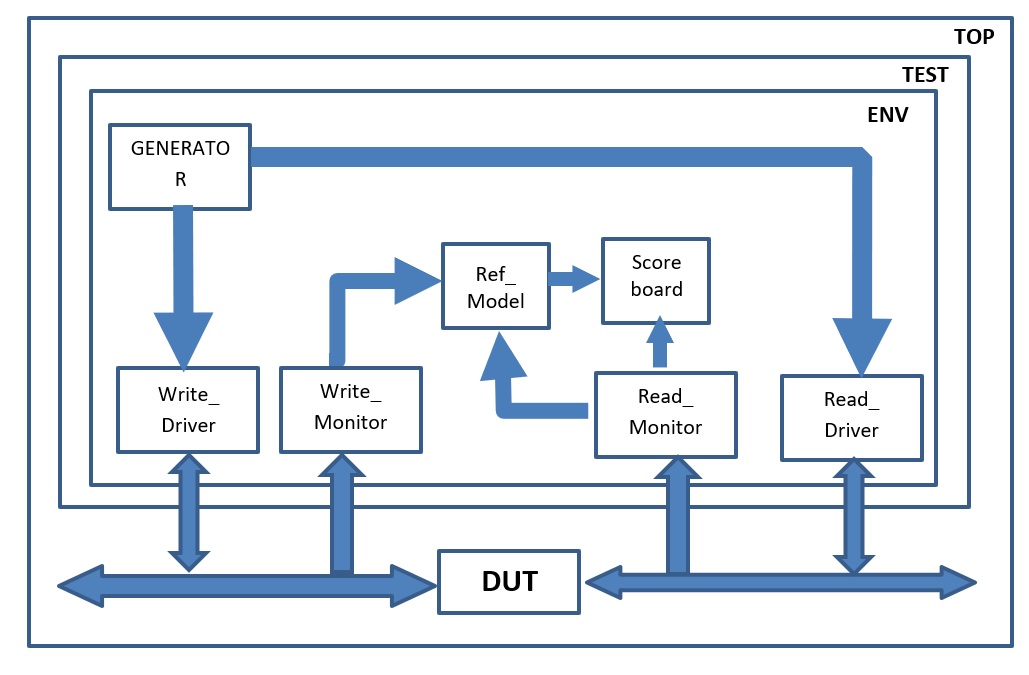


Figure 4.2: RAM Verification Architecture in System Verilog

## RAM Component

### Transaction Class

Write a description of Transaction class

### Generator Class

Write a description of Generator class

### Driver Class

Write a description of Driver class

### Monitor Class

Write a description of Monitor class

### Reference Model

Write a description of Reference class

### Scoreboard

Write a description of Scoreboard class

# Chapter 5: Running Simulation

# Chapter 6: Closure Report