Verification Plan for a Memory Block

- Table of Contents

- **Introduction**

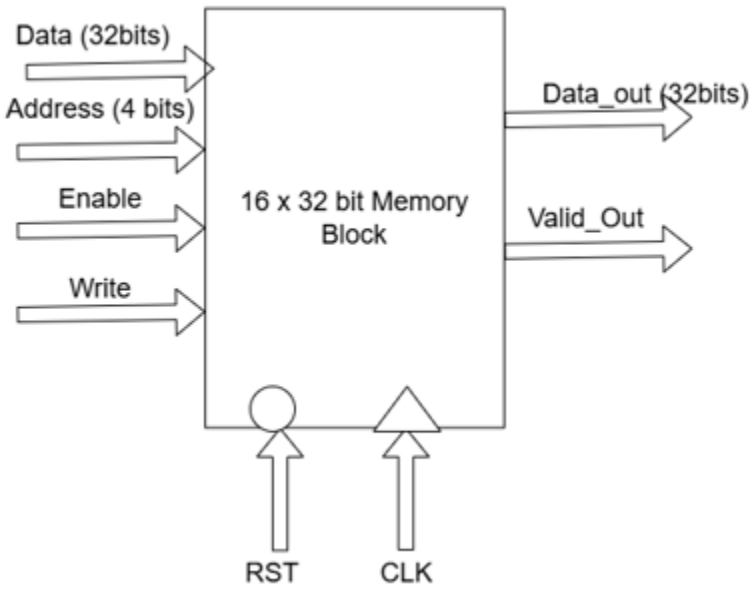
  - In this document we are building verification plan for a Memory Block that consists of 16 locations 32 bit each, the block also has the following:
    - Inputs
      - Input Data (32 bits)
      - Address (4 bits)
      - Enable signal (1 bits)
      - Clock signal (1 bit)
      - Reset signal (1 bit)
      - Write signal (1 bit)
    - outputs
      - Output Data (32 bits)
      - Valid signal (1 bit)

  - Read and Write operations done with the positive edge of the clock.
  - the block uses asynchronous Reset.
  - Valid signal is high for only one clock cycle when we produce new output.
  - Write signal is high when performing write operation and low otherwise.

- I/O table

Signal name	No. of bits	Description
Data_In	32	Write Data
Data_out	32	Read Data
rst	1	Active low reset
clk	1	Clock signal
en	1	Block enable
addr	4	Address to write or read from
valid_out	1	Indicate that the block producing an output
write	1	specifies the operation, low means read

- Block Diagram



- **Strategy**

  - Many verification techniques could be used such as , Emulation/ Prototyping, Formal Verification, Semi-Formal, HW/SW Co-verification and **Simulation Based Verification**

  - We will consider Simulation Based Verification in this Plan

- Coverage

1. Functional Coverage

- **Read Operation Coverage:**
  - **Ensure that all memory locations (16 locations) are read at least once.**
  - **Verify correct output data for various input combinations after a read.**
- **Write Operation Coverage:**
  - **Ensure that all memory locations (16 locations) are written to at least once.**
  - **Verify that the correct data is written for various input data combinations.**
- **Valid Signal Coverage:**
  - **Ensure that the valid signal is asserted (set high) for one clock cycle after a new output is produced.**
- **Reset Signal Coverage:**
  - **Cover both the reset assertion and de-assertion.**
  - **Verify that the memory block clears its contents when the asynchronous reset signal is active.**
  - **Ensure that the block behaves correctly immediately after the reset is de-asserted (e.g., subsequent reads or writes work as expected).**
- **Clock Signal Coverage:**
  - **Ensure that read and write operations occur on the positive edge of the clock.**

2. Code Coverage

- **Statement Coverage:**
  - **Ensure that all RTL code lines related to read, write, valid signal generation, and reset functionality are executed at least once.**
- **Branch Coverage:**
  - **Ensure that all conditional branches are covered:**
- **Toggle Coverage:**

- Ensure that all signals toggle between 1 and 0,

3. Scenario Coverage

- ## Read scenarios:
  - Read operation is performed when the enable signal is high.
  - Read operation is performed when the enable signal is low (output should not change).
  - Read operation is performed when Reset is asserted and when it is not asserted
  - Address changes between consecutive read operations (read from all the 16 locations).
  - Check the valid signal with each read.

- ## Write scenarios:
  - Write operation is performed when the enable signal is high and write signal is high.
  - Write operation is attempted when the write signal is low (no data should be written).
  - Write operation is performed with edge cases such as consecutive writes to the same address and writes to boundary addresses (0 and 15).
  - Write operation is performed when Reset is asserted and when it is not asserted
  - Ensure that Write operation done on all of the 16 locations correctly

- **Test Items**

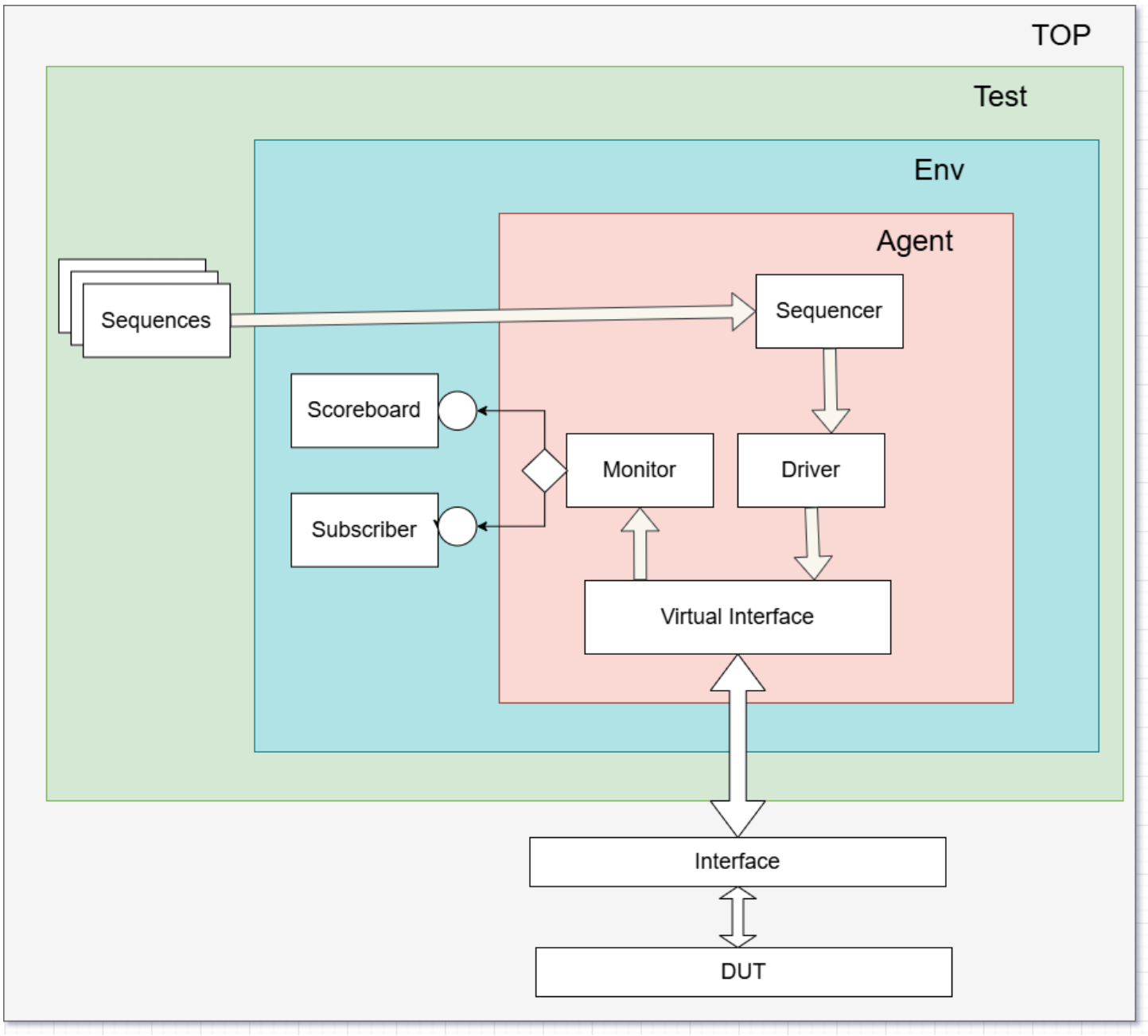
	Signals			
Test name	**Reset**	**Enable**	**Write**	**Address**
Reset-asserted write	low	high	high	A
Reset-asserted Read	low	high	Low	A
Enable-Low Write	high	low	high	A
Enable-Low Read	high	low	low	A
Write	high	high	high	0:15
Read	high	high	low	0:15
Out of Bound Write	high	high	high	16
Out of Bound Read	high	high	low	16

- # Exit Criteria

Coverage Goals and Metrics

- **Functional Coverage Goal:**
    - **Achieve 100% coverage of all defined read, write, reset, valid signal, and clock conditions.**
- **Code Coverage Goal:**
    - **Target 100% statement and branch coverage.**
- **Scenario Coverage Goal:**
    - **Ensure all edge cases, corner cases, and boundary conditions are covered, particularly with the enable, reset, and address signals.**

- **Environment**



- **Do file**

```
# Create a working library called 'work'
vlib work

# Compile the design and testbench files
vlog -f sourcefile.txt +cover=cells

# Simulate the testbench with coverage enabled
vsim -voptargs=+acc work.Top -coverage

# Add all signals to the waveform
add wave *

# Save the coverage database on exit
coverage save Top.ucdb -onexit

# Run the simulation
run -all
```