**FIFO Verification**

Using SV and UVM

[Mahmoud Ghamry](http://www.linkedin.com/in/mahmoud-ghamry)

OCT 2024

Table of Contents

Introduction 1

Architecture 2

Verification Flow 4

Verification plan 5

Design 6

Bugs 7

Verification Files 8

Design After Edit 8

FIFO Interface 10

FIFO Assertion 10

FIFO Top 11

FIFO Sequencer 11

FIFO Environment 12

FIFO Sequences 13

FIFO Sequence item 19

FIFO Agent 20

FIFO Monitor 21

FIFO Driver 22

FIFO Test 23

FIFO Configuration 24

FIFO Coverage 25

FIFO Scoreboard 26

DO FILE – Source file 30

Result 31

**1.Introduction**

In modern digital systems, the First-In-First-Out (FIFO) buffer plays a crucial role in managing data flow between components operating at different speeds. Verifying the functionality, reliability, and performance of a FIFO design is essential to ensure data integrity and seamless communication within these systems. To achieve a robust verification process, the Universal Verification Methodology (UVM) is employed due to its modular and reusable testbench structure.

This report focuses on the verification of a FIFO design using UVM. The primary objective of this project was to create a scalable and reusable test environment to validate the FIFO's functionality under various conditions, including boundary cases and stress scenarios. The verification environment was built with components such as UVM agents, sequences, scoreboards, and coverage analysis to ensure comprehensive functional verification.

This project aims to achieve the following objectives:

* Comprehensive Functional Testing: Validate the FIFO's correct behavior, including handling of read and write operations, data storage, and retrieval, as well as its response to full and empty conditions.
* Coverage-Driven Verification: Measure and analyze the functional coverage to identify gaps in the test scenarios and to ensure that all critical cases are addressed.
* Scalability and Reusability: Design the testbench components to be modular and reusable for other projects or variations of the FIFO design, adhering to best practices in UVM-based verification.

1

**2. Architecture**

The architecture of the FIFO UVM verification environment is designed to ensure maximum test coverage and modularity. It consists of the following key components:

Testbench Hierarchy

* Environment (env): The UVM environment serves as the top-level component that integrates the UVM agents, scoreboard, and coverage monitors. It orchestrates the communication between different components to ensure proper data flow and verification checks.
* UVM Agent: The UVM agent encapsulates the driver, sequencer, and monitor. It acts as an interface between the testbench and the Device Under Test (DUT), generating stimuli and collecting responses.
  + Driver: The driver converts the transaction-level sequences into pin-level signal activities, driving the input signals to the FIFO.
  + Sequencer: The sequencer controls the sequence of transactions that are sent to the driver. It plays a vital role in defining the order of data sent to the FIFO, ensuring the proper stimulus for each test scenario.
  + Monitor: The monitor observes the signal activities and extracts transaction-level data from the DUT's outputs, which are then sent to the scoreboard for comparison.

Scoreboard

The scoreboard is used to verify the functional correctness of the FIFO design by comparing the actual output data from the DUT with the expected data. It leverages a reference model to generate the expected values and highlights any discrepancies between the actual and predicted behavior. This ensures that the FIFO design operates as intended under all test conditions.

Coverage Collection

Functional coverage is a critical component in the verification process to measure the extent to which the FIFO's functionalities have been exercised by the test cases. Covergroups are used to track different scenarios such as:

* FIFO Full: Verifies that the FIFO correctly indicates when it reaches its maximum capacity.
* FIFO Empty: Checks that the FIFO accurately signals when all data has been read out.

2

* Almost Full: Tests the FIFO's behavior when it is nearly full, ensuring that control logic is triggered before overflow occurs.
* Almost Empty: Assesses the FIFO's state when it is almost empty, verifying proper signals before it reaches an empty condition.
* Write Acknowledge (wr\_Ack): Confirms that the FIFO acknowledges write operations appropriately, even under high data input rates.

Test Scenarios

The verification environment includes a variety of directed and random test scenarios designed to validate the FIFO's behavior under normal and edge cases. Some of the specific test sequences include:

* Reset Sequence: Verifies that the FIFO properly initializes and resets all internal states upon receiving a reset signal.
* Write-Only Sequence: Tests the FIFO by performing multiple write operations without reading, observing how the FIFO handles data overflow conditions.
* Read-Only Sequence: Evaluates the FIFO’s behavior when only read operations are performed, ensuring correct data output and proper handling of underflow conditions.
* Write and Read Sequence: Simulates simultaneous read and write operations to test the FIFO's capability to handle continuous data flow in both directions.
* Random Sequence: Uses randomization to generate unpredictable read and write .

صورة تحتوي على نص, لقطة شاشة, خط, رسم بياني

تم إنشاء الوصف تلقائياً

3

**3.** **Verification Flow**

**صورة تحتوي على نص, لقطة شاشة, رسم بياني, خطة

تم إنشاء الوصف تلقائياً**

4





**4. Verification Plan**

صورة تحتوي على نص, الخط, رقم, خط

تم إنشاء الوصف تلقائياً

5

**5.Design**

صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياًصورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً

6

**6.Bugs**

* The underflow flag should be registered.
* Reset the flags (empty – almostempty – wr\_ack – full – alomstfull ).
* Add cases.
  + - If the wr\_en , rd\_en and empty is high so the write is the higher priority and counter decreases .
    - If the wr\_en , rd\_en and full is high so the Read is the higher priority and counter increases .
    - If the wr\_en and rd\_en is high and not full nor empty so the two cases run together but the counter not changed.
    - If the wr\_en and rd\_en is low so everything not changed.

* The almostfull must be high if the FIFO need one input to be high

So I edit it

* Edit underflow, overflow and count

7

**7.** **Verification Files**

**7.1 Design After Edit**

**صورة تحتوي على لقطة شاشة, نص, برمجيات, برامج الوسائط المتعددة

تم إنشاء الوصف تلقائياًصورة تحتوي على نص, لقطة شاشة, برمجيات, برامج الوسائط المتعددة

تم إنشاء الوصف تلقائياًصورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً**8

**صورة تحتوي على نص, لقطة شاشة, برمجيات, برامج الوسائط المتعددة

تم إنشاء الوصف تلقائياً**

صورة تحتوي على لقطة شاشة, نص, الخط

تم إنشاء الوصف تلقائياً

9

**7.2 FIFO Interface**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

**7.3 FIFO Assertion**

صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً

10

**7.4 FIFO TOP**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

**7.5 FIFO Sequencer**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

11

**7.6 FIFO Environment**

صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً

12

**7.7 FIFO Sequences**

**7.7.1 RESET**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

13

**7.7.2 Write Only**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

14

**7.7.3 Read Only**

**صورة تحتوي على نص, لقطة شاشة, برمجيات, نظام التشغيل

تم إنشاء الوصف تلقائياً**

15

**7.7.4 Read and Write Enable**

**صورة تحتوي على نص, لقطة شاشة, برمجيات, نظام التشغيل

تم إنشاء الوصف تلقائياً**

16

**7.7.5 Write and Read Disable**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

17

**7.7.6 Random**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

18

**7.8 FIFO Sequence item**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً** **صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

19

**7.9 FIFO Agent**

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً** **صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

20

**7.10 FIFO Monitor**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياًصورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**21

**7.11 FIFO Driver**

**صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

22

**7.12 FIFO Test**

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً**

**23**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

**7.13 FIFO Configuration**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

24

**7.14 FIFO Coverage**

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياًصورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**25

**7.15 FIFO Scoreboard**

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً** **صورة تحتوي على نص, لقطة شاشة, برمجيات, برامج الوسائط المتعددة

تم إنشاء الوصف تلقائياً**

26

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً** **صورة تحتوي على نص, لقطة شاشة, برمجيات

تم إنشاء الوصف تلقائياً**

**27**

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً** **صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً**

28

**صورة تحتوي على نص, لقطة شاشة

تم إنشاء الوصف تلقائياً**

29

**7.16 DO File**

**صورة تحتوي على نص, الخط, لقطة شاشة

تم إنشاء الوصف تلقائياً**

**7.17 Source file**

**صورة تحتوي على نص, لقطة شاشة, الخط, التصميم

تم إنشاء الوصف تلقائياً**

30

**8.RESULT**

**8.1 UVM REPORT**

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

**8.2 Code Coverage**

****

**صورة تحتوي على نص, لقطة شاشة, الخط

تم إنشاء الوصف تلقائياً**

**8.3 Assertion Coverage**

****

31

**8.4 Function Coverage**

**صورة تحتوي على نص, لقطة شاشة, الخط, التصميم

تم إنشاء الوصف تلقائياً**

**صورة تحتوي على نص, لقطة شاشة, الخط, خط

تم إنشاء الوصف تلقائياً**

****

**8.5 WaveForm**

**Write Sequence**

**صورة تحتوي على لقطة شاشة, برامج الوسائط المتعددة, الإلكترونيات, برمجيات

تم إنشاء الوصف تلقائياً**

32

**Read Sequence**

**صورة تحتوي على لقطة شاشة, نص, برامج الوسائط المتعددة, برمجيات

تم إنشاء الوصف تلقائياً**

**Read and Write Sequence**

**صورة تحتوي على لقطة شاشة, نص, برامج الوسائط المتعددة, برمجيات

تم إنشاء الوصف تلقائياً**

33

**Read and Write Disable Sequence**

**صورة تحتوي على لقطة شاشة, نص, عرض, برمجيات

تم إنشاء الوصف تلقائياً**

**Random Sequence**

**صورة تحتوي على لقطة شاشة, الإلكترونيات, دائرة كهربائية

تم إنشاء الوصف تلقائياً**

**صورة تحتوي على لقطة شاشة, الإلكترونيات, برامج الوسائط المتعددة, خط

تم إنشاء الوصف تلقائياً** **صورة تحتوي على لقطة شاشة, الإلكترونيات, دائرة كهربائية

تم إنشاء الوصف تلقائياً**

**If you want to simulate my UVM Project please visit my** [**repo**](https://github.com/moghamry25/UVM_FIFO)

34