VERIFICATION PLAN

FUNDAMENTALS OF PRE- SILICON VALIDATION - ECE-593

SPRING – 2024

ASYNCHRONOUS FIFO

SUBMITTED BY:

Swaroop Chandrashekar

Shashikirana Chinchandlahalli Ravanappa

Smit Patel

Gautam Savaliya

Introduction:

Objective of the verification plan:

The primary objective of this verification plan for the asynchronous FIFO(first in first out) design in System Verilog is to ensure the proper functionality, reliability, and performance of the FIFO under wide coverage of operational conditions. The goal of the verification plan is to confirm that potential corner cases and error scenarios are handled correctly, as well as to fully validate the asynchronous data transfer, storage, and retrieval procedures within the FIFO. This covers extensive testing of asynchronous read and write operations, data integrity checks, write and read clock synchronization, full and empty conditions including half empty and half full conditions and adherence to predetermined interface protocols. The plan includes simulations to verify the asynchronous FIFO for its intended use in a wider system by covering both functional correctness and performance metrics.

Top Level block diagram:

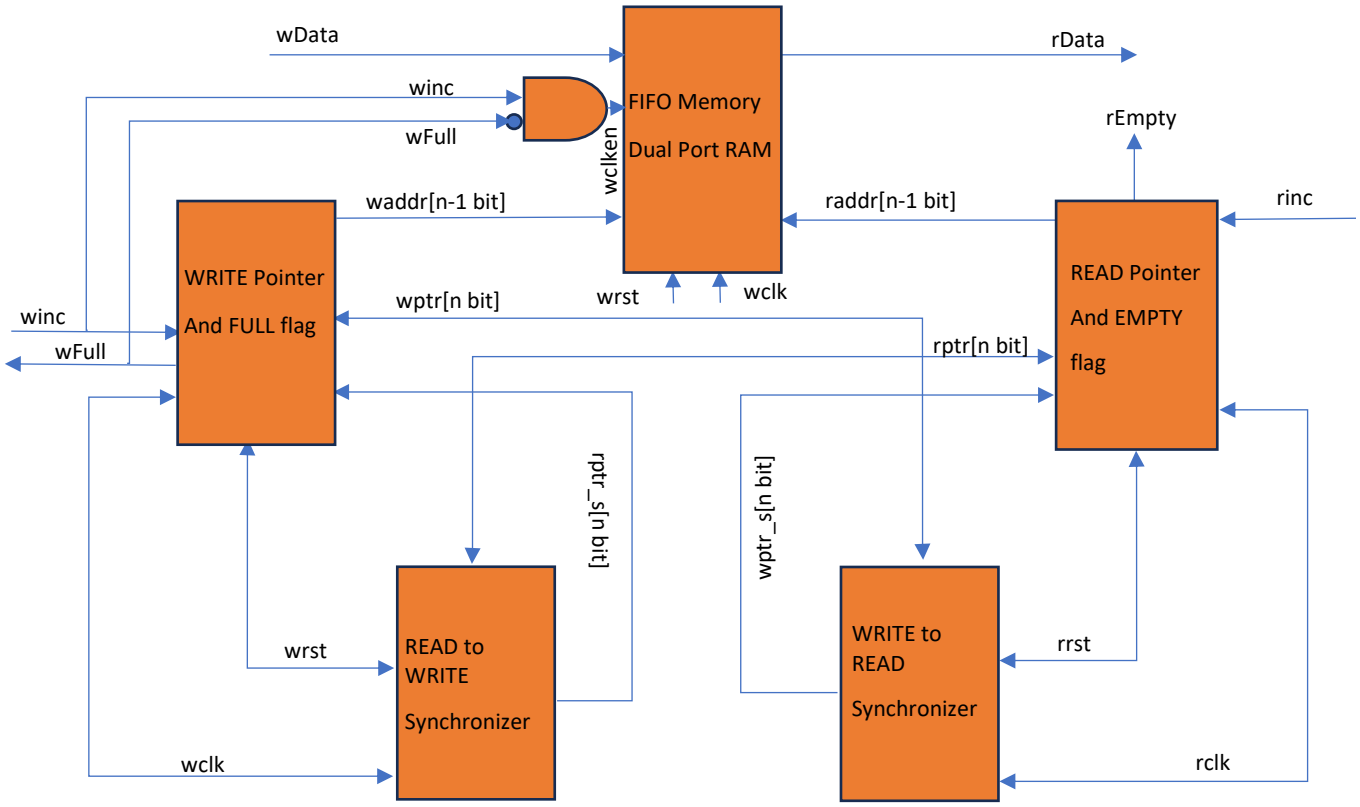


Figure 1: Top Level block diagram for the design

Verification Requirements:

Verification Levels: Six files, each containing a single module, make up the asynchronous FIFO design, hence verification will take place at the module level, one level below the top. This modular design makes it possible to evaluate each component's operation in a effective manner, utilizing the in and out settings to precisely manage the stimulus.

Functions to be Verified:

1) FIFO Memory array Full Flag detection
2) Read to Write Synchronizer pipeline
3) Write to Read Synchronizer pipeline
4) Write and Read Pointer Incrementation
5) Full Flag Generation logic
6) Empty Flag Generation logic

Verification Plan:

Environment Setup:

1) Developed a System Verilog testbench environment to verify the functionality of the FIFO module.
2) Instantiated the SV FIFO module within the conventional testbench environment.
3) Implemented the necessary interfaces to drive inputs and monitor outputs of the FIFO module.

TestPlan Setup:

1) Developed a set of test cases to verify different aspects of the FIFO module, including:
2) Write and read operations under various conditions (e.g., empty FIFO, full FIFO, alternating write and read operations).
3) Boundary conditions (e.g., maximum FIFO depth, single-element FIFO).
4) Data integrity checks (e.g., verifying that data read from the FIFO matches data written to it).

Randomization:
1)Implemented randomization techniques to generate diverse stimuli for the FIFO module inputs.
2) Randomize data values, write/read addresses, write/read enable signals, and other relevant signals to thoroughly exercise the FIFO module.

Error Handling:

1) Implement error detection and reporting mechanisms to handle unexpected conditions during simulation.

Performance Verification:

1) Evaluate the performance of the FIFO module under different scenarios, including stress testing with high data rates and varying clock frequencies.

2) Measure metrics such as throughput, latency, and resource utilization to ensure that the FIFO meets performance requirements.

Simulation Setup:

1) Configure the simulation environment with appropriate test stimuli, simulation duration, and other parameters.

2) Define the simulation runtime settings to enable detailed debugging and analysis of the FIFO module behavior.

Debugging Plan:

1) Perform extensive debugging to identify and resolve any issues encountered during verification.

2) Use waveform viewers, logging, and debug messages to analyze the behavior of the FIFO module and verify correct operation.

Documentation and Reporting:

1) Document the verification plan, test cases, results, and any identified issues or limitations.
2) Generate reports summarizing the verification activities, coverage metrics, and verification results for stakeholders.

Test Case Scenarios

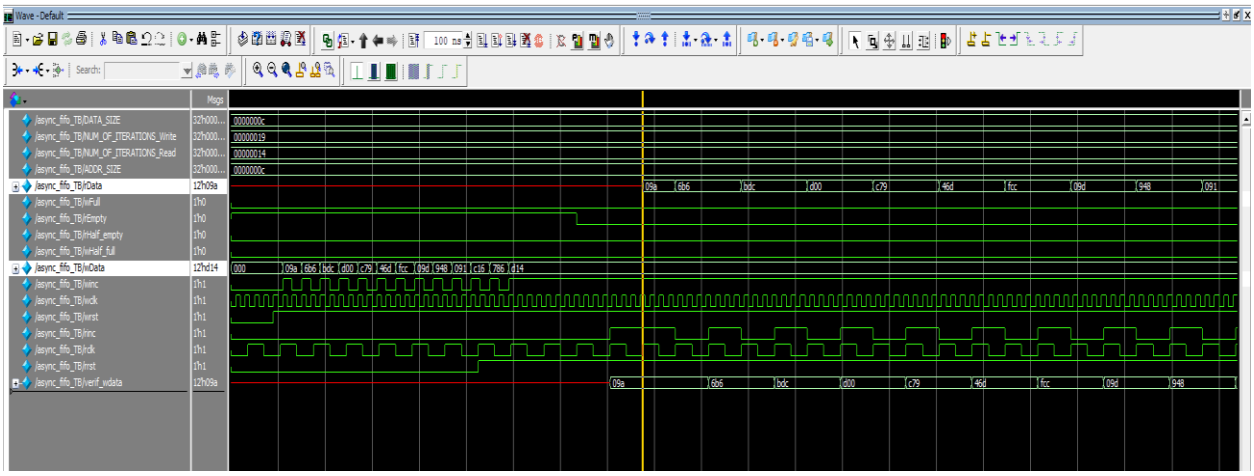
Test Name	**Test Description**
Top Module: Write and Read	Basic Testing
FIFO Memory Array Check	Check Read and Write into Array at module level
FIFO Memory Full Flag Detection	Checks to see if an asserted full flag
Write Pointer increment	Write Pointer must increment on winc
Read Pointer increment	Read Pointer must increment on rinc
Reset Operation	Reset operation from top module must propogate through the entire design
Read to Write Synchronizer pipeline	Ensures a Pointer passed is synchronized during clock domain crossings
Write to Read Synchronizer pipeline	Ensures a Pointer passed is synchronized during clock domain crossings
Full Flag generation	Ensures Full flag generation logic functions properly on write pointer wrap around

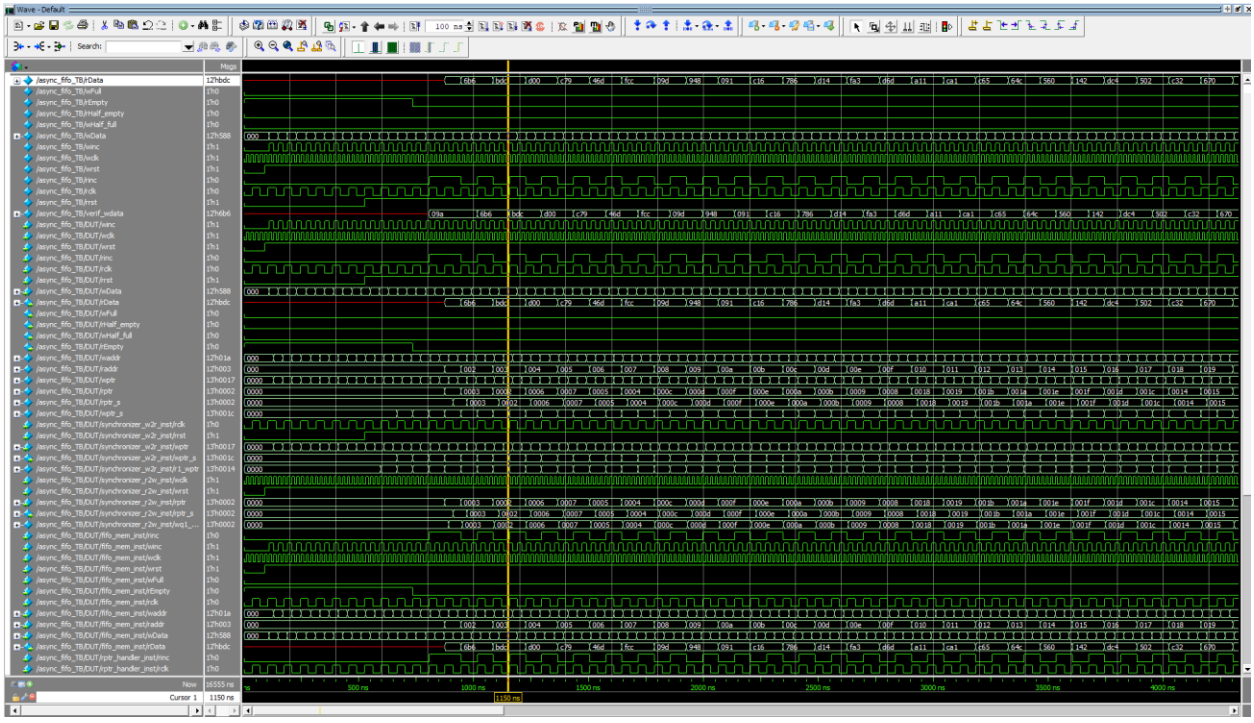
Empty Flag generation	Ensures Empty flag generation logic functions properly on read pointer equal to write pointer
Read Pointer Grey code conversion	Pointer must be converted to Grey Code before being sent to synchronizer
Write Pointer Grey code conversion	Pointer must be converted to Grey Code before being sent to synchronizer

Schedule:

Tasks	Done by	Planned Date	Status
Design specification document	Gautam	4/23	Completed
Verification Plan	Swaroop	4/23	Completed
Design implementation 1) fifo_mem	Swaroop	4/23	Completed
2) synchronizer_r2w	Swaroop	4/23	Completed
3) synchronizer_w2r	Gautam	4/23	Completed
4) rptr_handler	Smit	4/23	Completed
5) wptr_handler	Smit	4/23	Completed
6) Top Module	Shashikirana	4/23	Completed
Conventional Testbench	Shashikirana	4/23	Completed

Waveform:





Github Link:

https://github.com/swaroop-chandrashekar/UVM_Project_Group9