1. Verification Requirements

A. Verification levels

- In this project we are using design level verification.

B. Functions

- Basic
  - WRITE: write the data when write enable is high.
  - READ: read data when read enable is high.
  - CLEAR: write pointer decrements by 1 when clear is high.
- Advanced
  - BYPASS: When read and write enable's is high on empty read data should be present on write data without getting storedin FIFO.
  - ERROR :
    1. The fifo is empty and we try to read
    2. The fifo is empty and we try to clear
    3. The fifo is empty and we try to read and clear
    4. The fifo is full and we try to write
    5. Commands to write and clear at the same time
    6. Any command to read, write, or clear when RESET is asserted
  - sequence of ops (write the data into FIFO).
  - sequence of ops (read the data out of FIFO).
  - Full, empty dependent corner cases
    1. write cannot be happened when full is high
    2. read cannot happen when empty is high.
    3. ignore data unless read enable or write enable is high.
- Generic
  - no superfluous outputs
  - reset function: empty is asserted and full is de-asserted.

C. Specific tests & methods

- Type of verification
  - Black box testing : Reference model.
  - White box testing: Assertions.
- Verification Strategy

        - Deterministic and simplest strategy.
    - Abstraction Level
        - RTL (Register transfer level) verification.
    - Checking
        - Here we are checking all inputs and outputs and all corner cases.
        - Given stimulus with constrained randomization and directed test cases via tasks.
        - Designed a reference model by using Queue methods.
        - Written a concurrent Assertions to verify the functionality of all signals.

D. Coverage

- In this project we used code coverage and functional coverage.
- We covered every input and output by using cover points and bins.
- We covered back-to-back cases of write and read and clear.
- We covered write when Fifo is full, and write-clear.
- And we covered write, read ,Clear when reset is asserted.
- And we covered clear-read, write-read, clear, read when Fifo is empty.
- And also covered all error cases.
- And we done the cross coverage by using two cover points.
- A bin for write and read when Fifo-full.
- We cross covered the read when Fifo-full.
- We cross covered the write when Fifo-full.
- We cros
- s covered the read when Fifo-empty.
- We cross covered the write when Fifo-empty.
- We cross covered the clear when Fifo-empty.

E. Scenarios

- Read the FIFO when it is empty.
- Write the FIFO when it is Full.

- Clear the FIFO when it is empty.
- Read and Clear Sametime when FIFO is empty.
- Write and Clear Sametime When FIFO is empty.
- Write and Clear Sametime When FIFO is FULL.
- Write and Clear Sametime When FIFO is partial.
- Read and Write Sametime when FIFO is Empty.
- Write the FIFO when RESET is asserted.
- Read the FIFO when RESET is asserted.
- Clear the FIFO when RESET is asserted.
- Back-to-Back Writes, Reads, Clears.
- Write data to FIFO.
- Clear FIFO.
- Read data from FIFO.
- Multiple read and write operations.
- Write and Read when it is Full.
- Corner cases – h00, hFF, hAA,h55.
- Giving Clears until Fifo is Empty.

2. Project Management

A. Tools

- Verilog/System Verilog for RTL design
- Questa Sim for simulation and analysis.
- Coverage tools for coverage analysis

B. Risks/dependencies

- Availability of resources.
- Changes in design specifications.

C. Resources

- Verification engineers.
- Design engineers.

D. Schedule

- Design: 1 week
- Testbench development and bug fixing: 1 week.
- Bypass and error logic and for coverage analysis: 2 weeks.
- For verification and creating classes/Test plan: 2 week.
- Checking – Assertions and Reference model : 2 weeks

Coverage Report:

We have gained an increase in the coverage percentage from the past project(project_2) to project_3 from 88% to 100%.