AXI – UART COMMUNICATION SYSTEM

NOVEMBER 2, 2025

MIRAFRA TECHNOLOGY

Contents

CHAPTER – 1

1. Introduction

- The AXI Stream to UART system transfers parallel data from an AXI-Stream interface to a UART serial channel.
- Data send by the AXI-Stream master is pushed into a synchronous FIFO buffer. The FIFO output is then serialized and transmitted by the UART transmitter.
- On the receiver side, the UART receiver deserializes the data, checks parity, and outputs parallel data.

1.1. Functionality

Module	Function
AXIS Master Input	sends valid data packets for transmission using AXI Stream handshaking signals (valid, ready, last).
Synchronous FIFO	Buffers AXI Stream data to handle speed differences between producer (AXIS Master) and consumer (UART TX). Supports parallel read/write operations and “full/empty” detection.
UART Transmitter	Converts parallel FIFO data into serial format as per configured baud rate. Supports even parity generation.
UART Receiver	Deserializes received UART stream, reconstructs parallel data, verifies parity, and asserts rx_valid when data is ready and parity is verified.

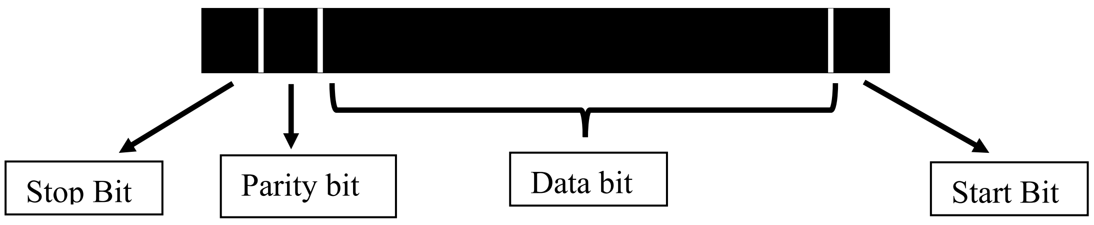
1.2. Operating Sequence

- **AXI Input Stage:** AXI master asserts axis_valid with data on axis_data. When m_axis_ready is high, data is accepted and written to FIFO.
- **FIFO Buffering:** FIFO stores incoming bytes until the UART transmitter is ready to send. Read and write can occur simultaneously.
- **UART Transmission:**
  - Each byte is serialized at 115200 baud.
  - Start bit (0), data bits (8), parity bit, and stop bit (1) are sent.
  - tx_data_ready indicates when UART can accept new data.
- **UART Reception:**
  - Receiver detects start bit, samples bits at mid-baud intervals.
  - Extracts data, verifies parity, and asserts rx_valid.

1.3. Design Parameters

Parameter	Value
Data Bits	8 bits per UART frame
Baud Rate	115200
Clock Frequency	50 Mhz
FIFO Depth	8 bytes

1.4. UART Frame Format



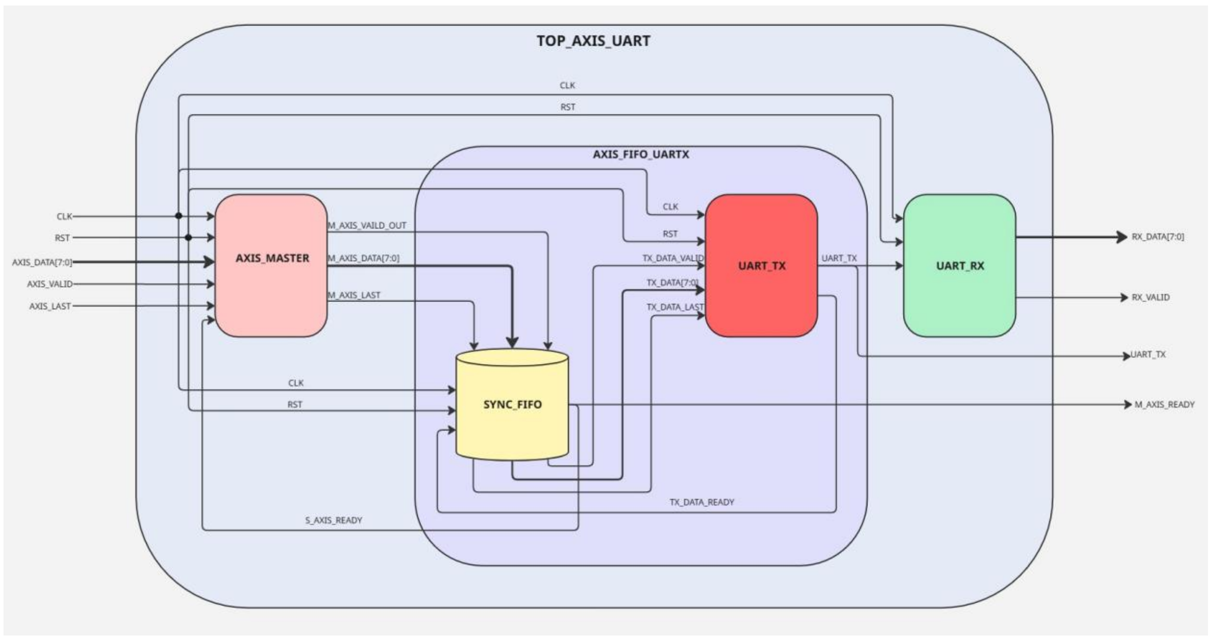
1.5. Working

- The project verifies an AXI to UART interface using a UVM-based testbench.
- The DUT converts AXI-stream input data into serial UART transmission and reconstructs it back into parallel output.
- The input signal axis_data carries the 8-bit data from the AXI-stream interface.
- The signal axis_valid indicates when the data on axis_data is valid.
- The signal axis_last goes high to mark the end of a packet once the last data byte has been sent.
- The UART transmitter sends data serially bit by bit through uart_tx.
- Each UART frame consists of 1 start bit, 8 data bits, 1 parity bit, and 1 stop bit.
- The total frame length is 11 bits per byte of data.
- The baud rate used is 115200 bps, which corresponds to 434 clock cycles per bit.
- One full byte transmission requires 4774 clock cycles (11 × 434).
- Every 434 clock cycles, one bit of data is transmitted on the uart_tx line.
- The UART receiver samples the uart_tx line and reconstructs the original 8-bit parallel data.
- The receiver removes the start, parity, and stop bits, retaining only the data bits.
- After 4774 clock cycles, one complete byte of data appears at rx_data.
- The signal rx_valid goes high when valid parallel data is available.
- The signal m_axis_ready indicates the readiness of the receiver FIFO to accept data.
- When m_axis_ready is 1, the FIFO has space and can accept more data.
- When m_axis_ready is 0, the FIFO is full and cannot accept new data.
- A comparison is made between rx_data (parallel output) and the reconstructed serial data from uart_tx.
- If both values match, the data transmission is verified as correct.
- When data matches, rx_valid is asserted high to indicate successful reception.
- The testbench monitors and verifies data integrity, timing, and control signal behavior throughout the communication process.

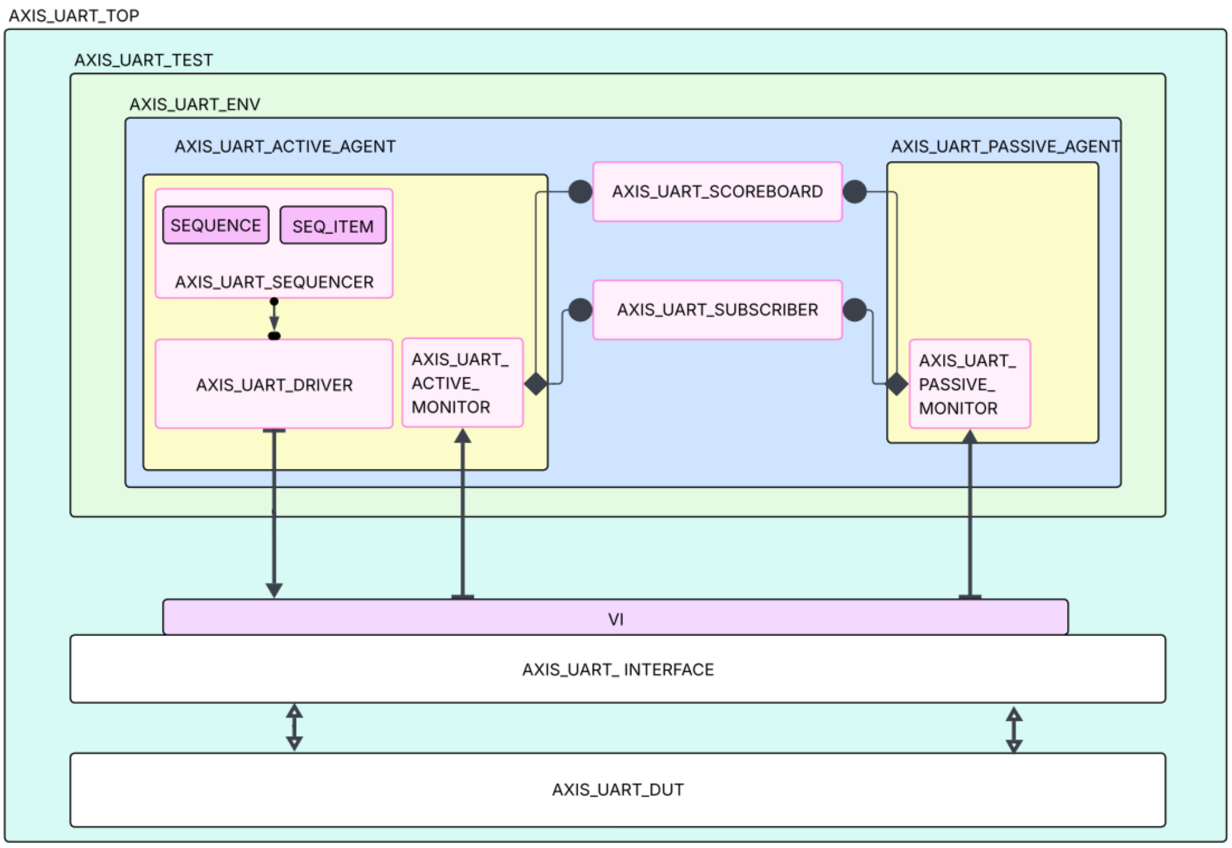
CHAPTER -2

2. Architecture

2.1. Design Architecture



2.2. Testbench Architecture



2.3. Interface Signals

Pin Name	Direction	Width	Function
clk	Input	1	Clock Signal
rst	Input	1	Active High Synchronous Reset
axis_data	Input	8	Parallel input data from AXI Stream master
axis_valid	Input	1	AXI Stream valid signal – indicates valid data for axis_data
axis_last	Input	1	Indicates last data in a frame
m_axis_ready	Output	1	Indicates that system is ready to accept new AXI data i.e. the FIFO is not full.
uart_tx	Output	1	UART serial transmit line
rx_valid	Output	1	High for one clock cycle when new data is received successfully
rx_data	Output	8	Parallel data after UART reception

CHAPTER – 3

3. Testbench Components

- **axi_uart_interface.sv**
  - Defines the connection between the DUT and the UVM testbench.
  - Includes signals like axis_data, axis_valid, axis_last, uart_tx, rx_valid, rx_data, and m_axis_ready.
  - Contains clocking blocks for driver and monitor synchronization.
  - Provides modports for active and passive agents to interact with the DUT.
- **axi_uart_seq_item.sv**
  - Defines the transaction class that represents a single unit of data transfer.
  - Includes fields such as axis_data, axis_valid, axis_last, rx_data, rx_valid, and uart_tx.
  - Used as a communication object between the driver, sequencer, and monitors.
- **axi_uart_sequence.sv**
  - Generates a sequence of transaction items (axi_uart_seq_item).
  - Randomizes data to simulate multiple data packets being sent to the DUT.
  - Controls the number of packets to send using a repeat loop or global variable.
- **axi_uart_sequencer.sv**
  - Acts as a bridge between the sequence and the driver.
  - Sends transactions from the sequence to the driver for execution.
  - Synchronizes the flow of data items during simulation.
- **axi_uart_driver.sv**
  - Drives the AXI input signals (axis_data, axis_valid, axis_last) to the DUT using the virtual interface.
  - Receives sequence items from the sequencer and applies them to the DUT.
  - Ensures timing correctness and data validity on the AXI interface.
- **axi_uart_active_monitor.sv**
  - Monitors the DUT's input (AXI side) interface.
  - Samples axis_data, axis_valid, axis_last, and m_axis_ready signals.
  - Sends observed transactions to the scoreboard and subscriber through an analysis port.
  - Verifies that valid data is sent when expected.
- **axi_uart_passive_monitor.sv**
  - Observes the DUT's output (UART side).
  - Samples uart_tx every 434 clock cycles to reconstruct the serial data.
  - After 4774 clock cycles, produces one byte of received data (rx_data).
  - Sends the monitored data to the scoreboard for comparison with expected results.
- **axi_uart_active_agent.sv**
  - Contains the active components: driver, sequencer, and active monitor.
  - Responsible for generating and applying stimulus to the DUT.
  - Operates in *UVM_ACTIVE* mode to drive signals.
- **axi_uart_passive_agent.sv**
  - Contains only the passive monitor (no driver or sequencer).
  - Operates in *UVM_PASSIVE* mode to observe DUT outputs.
  - Used to monitor UART response behavior.

- **axi_uart_scoreboard.sv**
    - Performs data checking and result verification.
    - Compares transmitted (axis_data) and received (rx_data) values.
    - Uses queues to store expected data and remove entries when matched.
    - Checks parity correctness and FIFO full/empty conditions.
- **axi_uart_subscriber.sv**
    - Collects functional coverage information from analysis ports.
    - Ensures all scenarios such as valid/invalid data, FIFO full/empty, and parity conditions are covered.
- **axi_uart_environment.sv**
    - Instantiates and connects all agents, scoreboard, and subscriber.
    - Defines the structure of the entire testbench.
    - Connects analysis ports of monitors to the scoreboard and subscriber.
- **axi_uart_test.sv**
    - Top-level UVM test class.
    - Creates the environment and runs the main sequence.
    - Raises and drops simulation objections to control phase execution.
    - Optionally applies drain time to allow pending transactions to complete.
- **axi_uart_pkg.sv**
    - Central package file that includes all other class files.
    - Simplifies compilation by bundling all UVM components together.
    - Allows easy import into the testbench.
- **global_pkg.sv**
    - Contains global static variables and counters shared across components.
    - Stores variables like count, baud_count, and randomized pkt (packet count).
    - Helps maintain global simulation states accessible throughout the environment

CHAPTER – 4

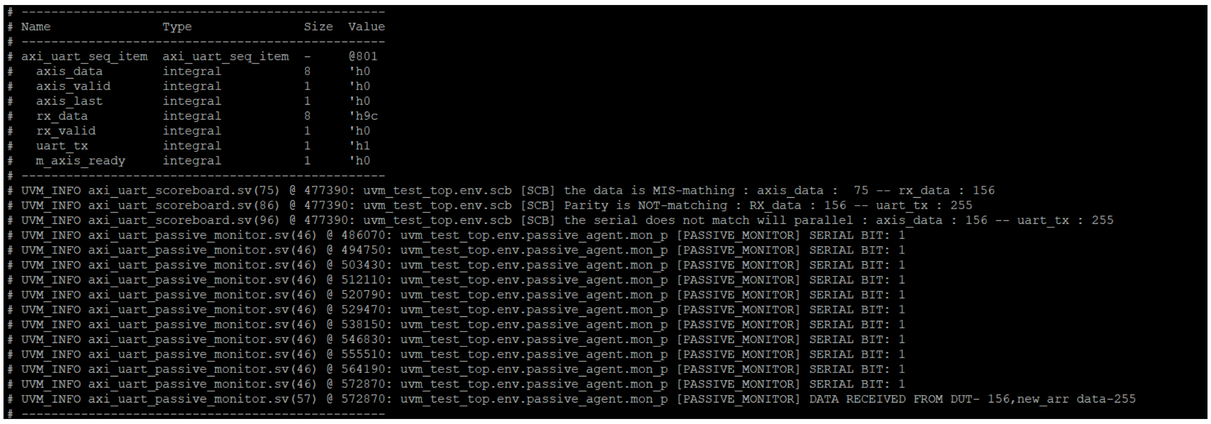
4. Bugs Found

4.1. Reset condition – BUG1

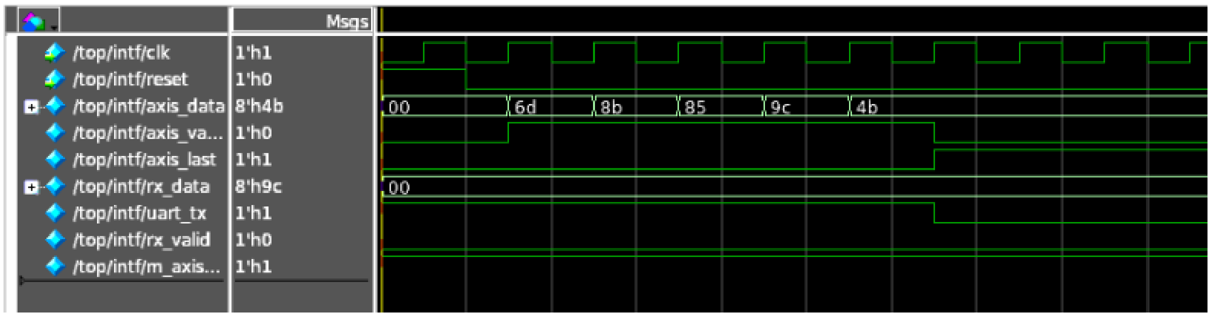
- While asserting the reset high after few clock cycles, the last 2 bytes of data were only coming and all other data were lost.
- When reset given in the initial clock cycles the DUT was working fine but giving in between caused unexpected behaviour.

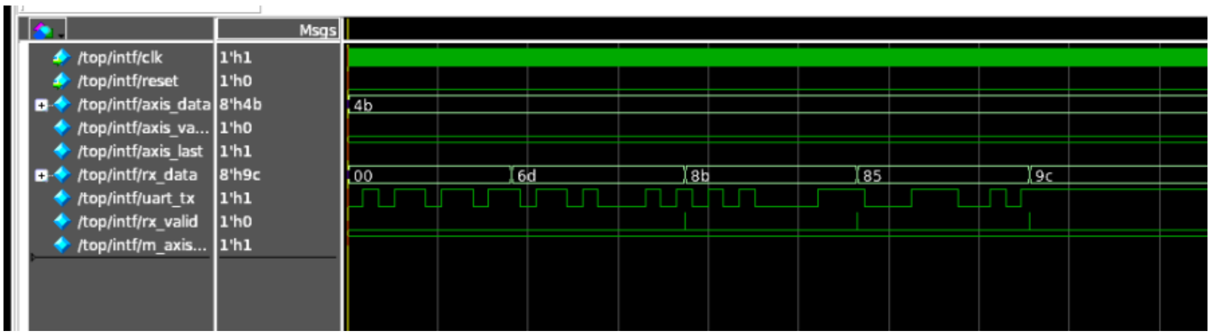
4.2. Sending Stream of Data – BUG2

- While sending the stream of data the last byte of data was always erased.



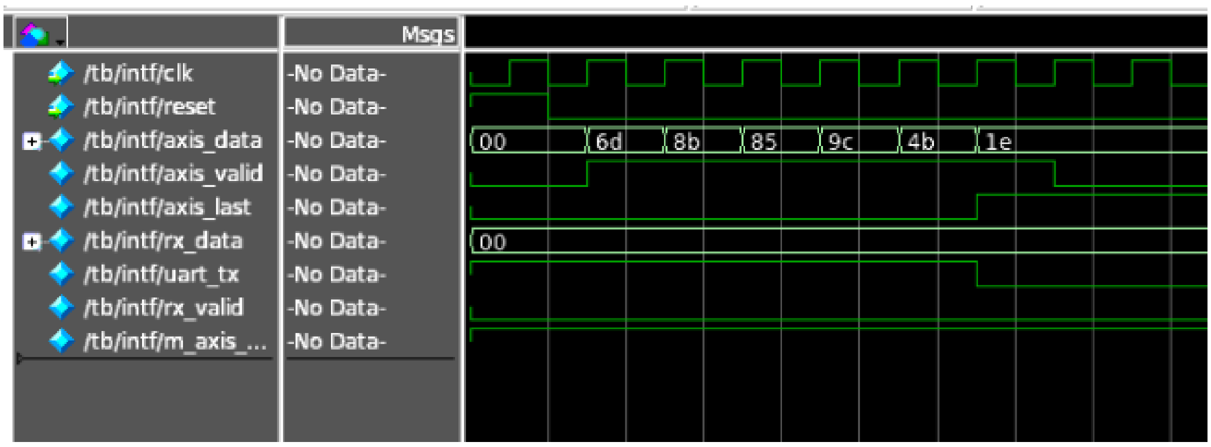
```
# -------------------------------------------------
# Name                Type                Size  Value
# -------------------------------------------------
# axi_uart_seq_item   axi_uart_seq_item   -     @801
#   axis_data         integral            8     'h0
#   axis_valid        integral            1     'h0
#   axis_last         integral            1     'h0
#   rx_data           integral            8     'h9c
#   rx_valid          integral            1     'h0
#   uart_tx           integral            1     'h1
#   m_axis_ready      integral            1     'h0
# -------------------------------------------------
# UVM_INFO axi_uart_scoreboard.sv(75) @ 477390: uvm_test_top.env.scb [SCB] the data is MIS-mathing : axis_data :  75 -- rx_data : 156
# UVM_INFO axi_uart_scoreboard.sv(86) @ 477390: uvm_test_top.env.scb [SCB] Parity is NOT-matching : RX_data : 156 -- uart_tx : 255
# UVM_INFO axi_uart_scoreboard.sv(96) @ 477390: uvm_test_top.env.scb [SCB] the serial does not match will parallel : axis_data : 156 -- uart_tx : 255
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 486070: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 494750: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 503430: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 512110: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 520790: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 529470: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 538150: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 546830: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 555510: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 564190: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(46) @ 572870: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(57) @ 572870: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] DATA RECEIVED FROM DUT- 156,new_arr data-255
# -------------------------------------------------
```



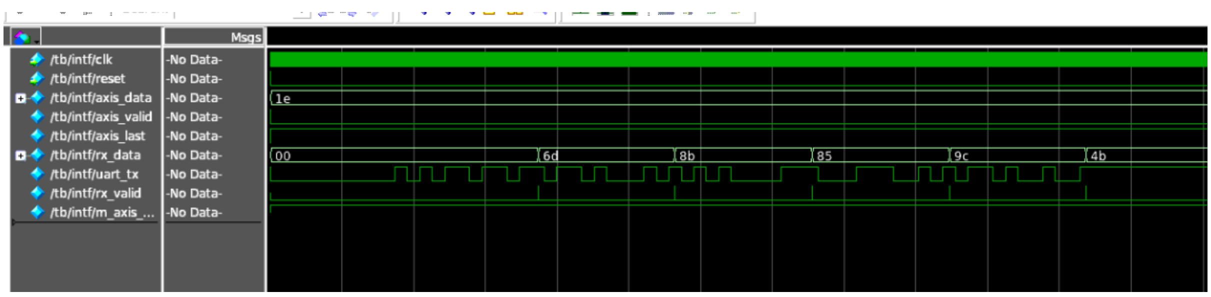


4.3. Axis_last behaviour – BUG3

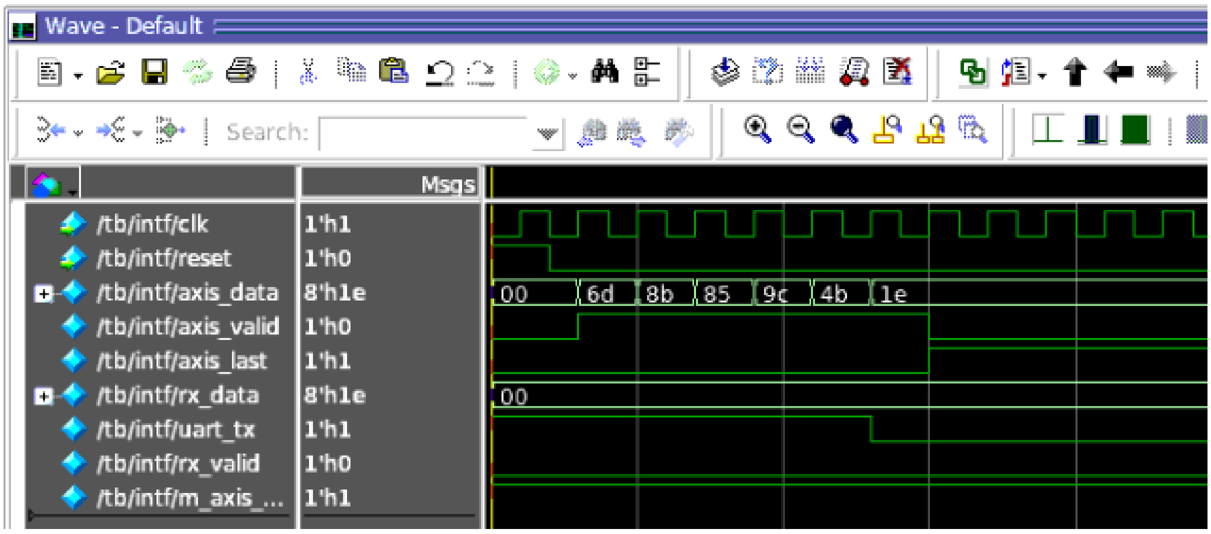
- According to the initial specification the axis_last signal should be triggered with the last byte of data, indicating the DUT that no more data will be sent.
- While asserting the axis_last with the last byte of data ,was making us loose the last byte.
- This was reported and the specification was changed mentioning that the asxis_last should be asserted 1 clock cycle after the last byte of data only.



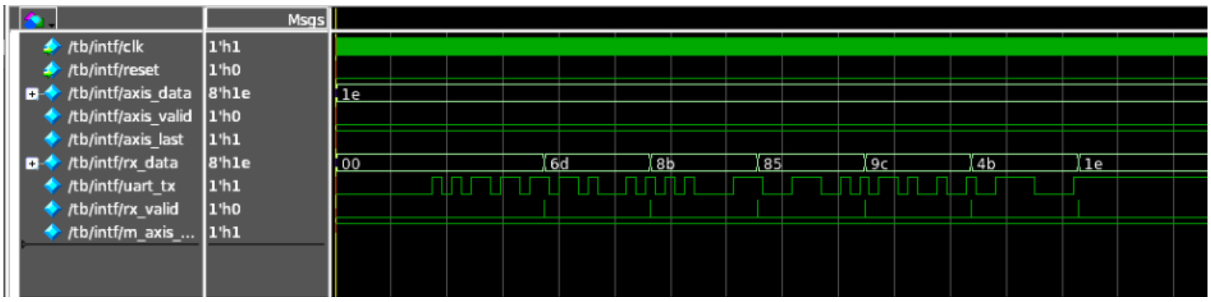
```
/tb/intf/clk         -No Data-
/tb/intf/reset       -No Data-
/tb/intf/axis_data   -No Data-
/tb/intf/axis_valid  -No Data-
/tb/intf/axis_last   -No Data-
/tb/intf/rx_data     -No Data-
/tb/intf/uart_tx     -No Data-
/tb/intf/rx_valid    -No Data-
/tb/intf/m_axis_...  -No Data-
```



```
/tb/intf/clk         -No Data-
/tb/intf/reset       -No Data-
/tb/intf/axis_data   -No Data-
/tb/intf/axis_valid  -No Data-
/tb/intf/axis_last   -No Data-
/tb/intf/rx_data     -No Data-
/tb/intf/uart_tx     -No Data-
/tb/intf/rx_valid    -No Data-
/tb/intf/m_axis_...  -No Data-
```

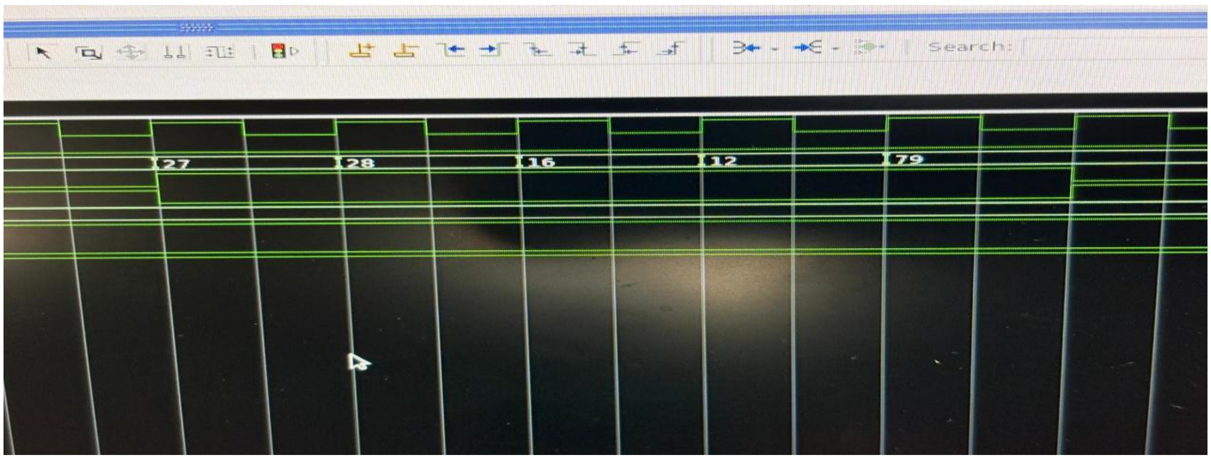


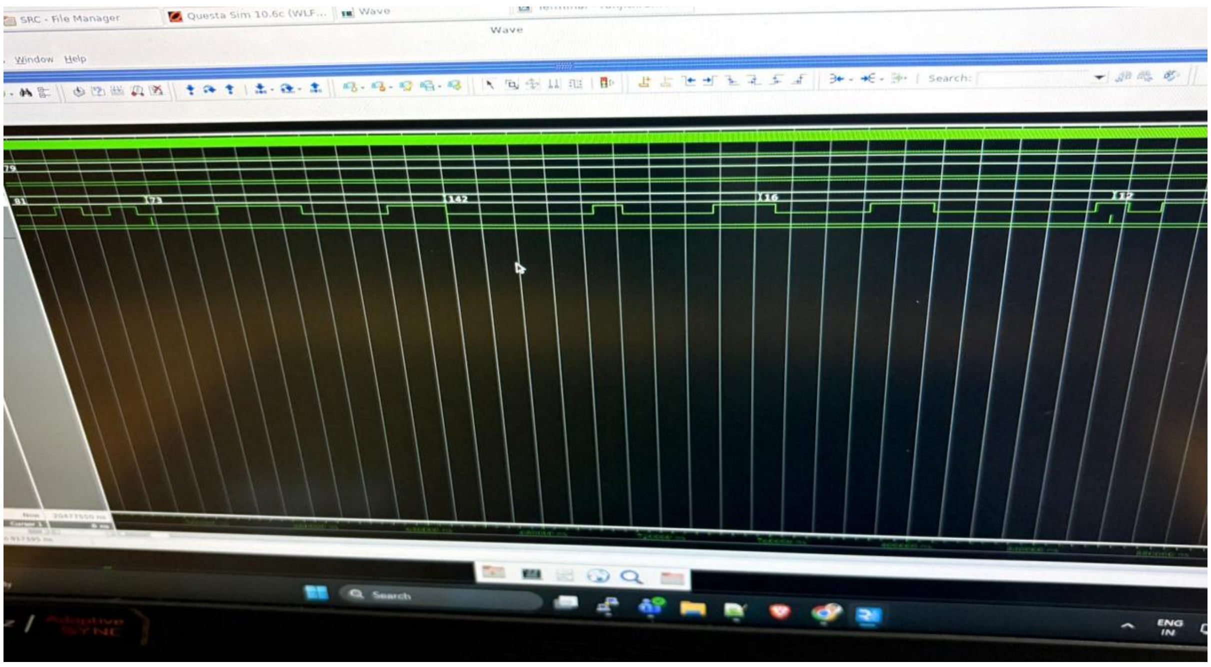
```
/tb/intf/clk         1'h1
/tb/intf/reset       1'h0
/tb/intf/axis_data   8'h1e
/tb/intf/axis_valid  1'h0
/tb/intf/axis_last   1'h1
/tb/intf/rx_data     8'h1e
/tb/intf/uart_tx     1'h1
/tb/intf/rx_valid    1'h0
/tb/intf/m_axis_...  1'h1
```



4.4. Multi Stream of data – BUG4

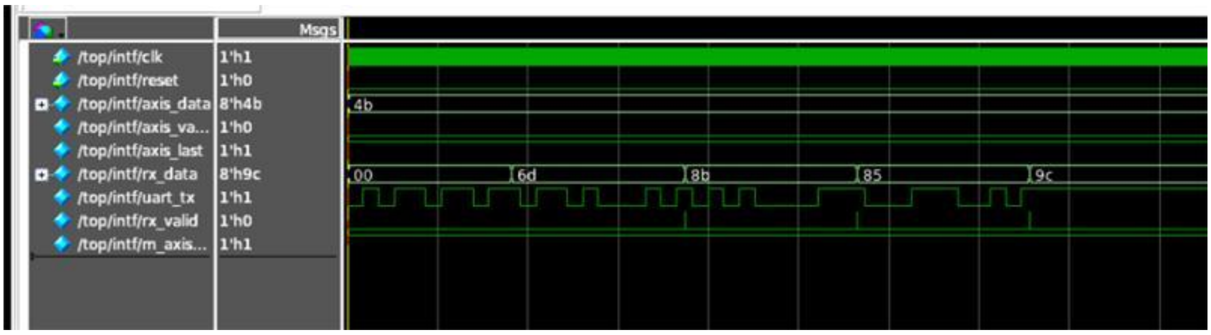
- After receiving the first stream of data successfully we immediately sent the next stream of data in that case the first 2 byte if next stream were missing.





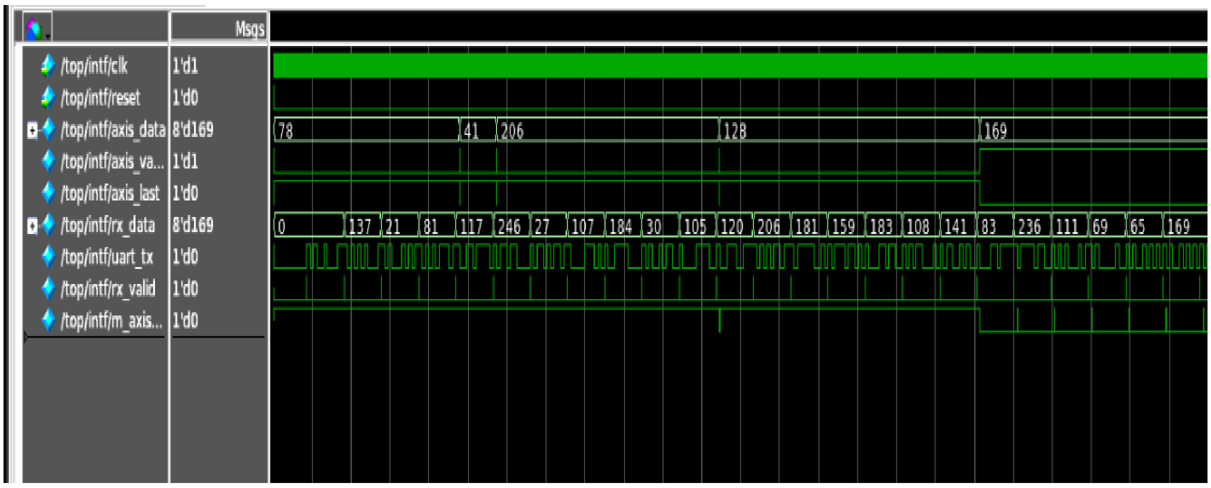
4.5. RX_VALID triggering missed – BUG5

- While sending the stream of data we have observed that even when the parallel data and serial data are matching the RX valid signal is not triggered high for every first byte of data.



4.6. RX_VALID triggering for the data we didn't send – BUG6 (Specification bug)

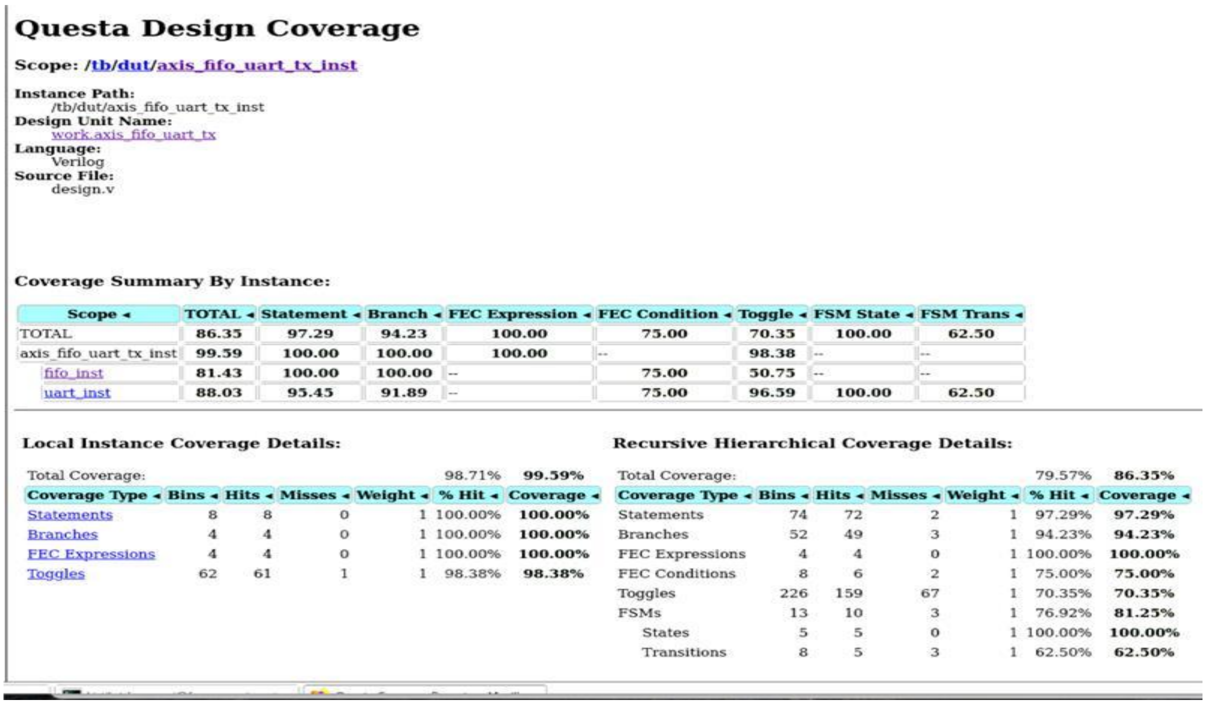
- In the process of verification we have found out that, only for the output of the first data stream we are getting "0" which was not expected as we have not sent 0 from the driver.
- This bug was issued to the design team they have made a change in the specification specifying that the defaut value for the first transaction will be "0".



Sl.no	Bug	Description	Status	Type of BUG
1	Missing of last packet	When N packets are sent,we receive only N-1 packets from DUT,where last data is missed.	Solved	DESIGN
2	axis_last is not working	According to the specification,axis_last signal which needs to be asserted when the last packet of the stream is sent,but the last packet is not coming out from DUT.	Specification changed	SPECIFICATION
3	rx_valid is not asserted for first packet	rx_valid which is asserted when the parity of parallel data and serial data are matched,is not getting asserted for the first packet of the stream even when parity is matching	Solved	DESIGN
4	Junk value from DUT	When multiple stream of packets are sent, if number of packets are greater than fifo size ,everything works fine,but when number of packets are less than fifo size the first stream comes out as expected , but in second stream first 2 packets are lost and instead of that a junk value comes from DUT and then rest 3 packets come out as expected.	Solved	DESIGN
5	Rx_valid asserting without data	Even when we are sending our first data as 137 with axis_valid , the rx-valid is asserting high for data – 0 which is default in the start this condition has not been mentioned in the spec	Solved	SPECIFICATION
6	Reset	When asserting the reset after some time the data which is present in the last and first was only coming	Solved	DESIGN

CHAPTER – 5

5. DUT coverage



Questa Design Coverage

Scope: /tb/dut/axis_fifo_uart_tx_inst

Instance Path:
/tb/dut/axis_fifo_uart_tx_inst
Design Unit Name:
work.axis_fifo_uart_tx
Language:
Verilog
Source File:
design.v

Coverage Summary By Instance:

Scope	TOTAL	Statement	Branch	FEC Expression	FEC Condition	Toggle	FSM State	FSM Trans
TOTAL	86.35	97.29	94.23	100.00	75.00	70.35	100.00	62.50
axis_fifo_uart_tx_inst	99.59	100.00	100.00	100.00	--	98.38	--	--
fifo_inst	81.43	100.00	100.00	--	75.00	50.75	--	--
uart_inst	88.03	95.45	91.89	--	75.00	96.59	100.00	62.50

Local Instance Coverage Details:

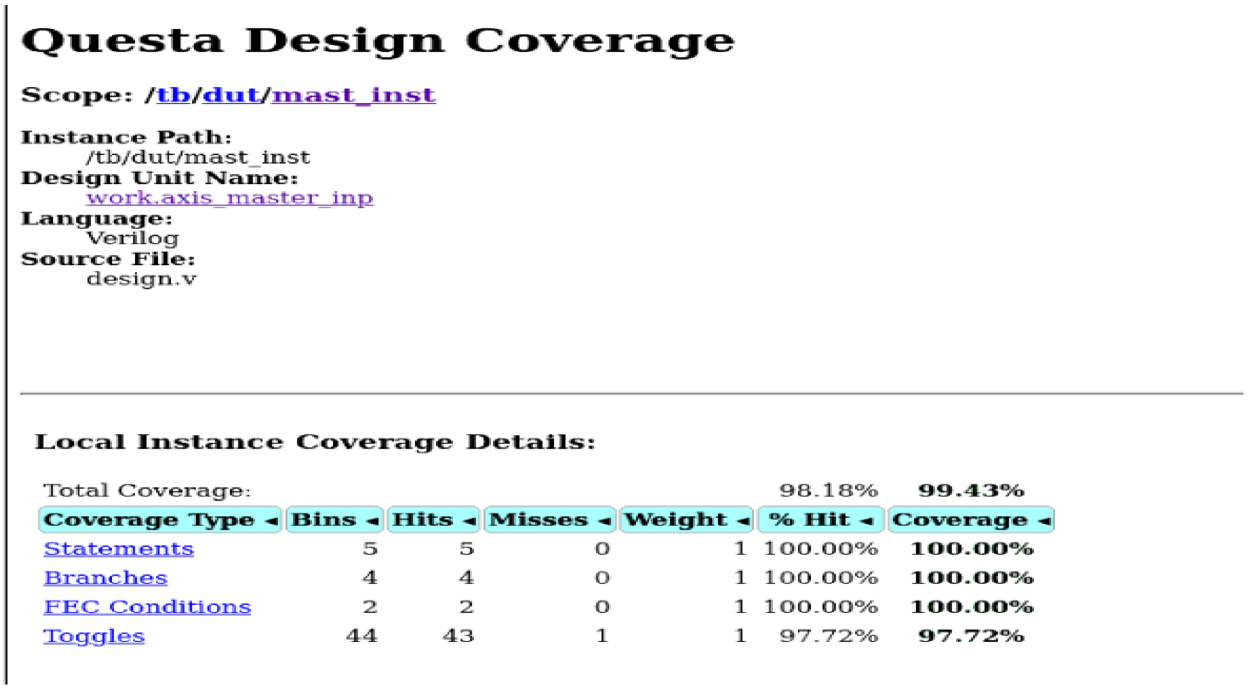
Total Coverage: 98.71% **99.59%**

Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Statements	8	8	0	1	100.00%	100.00%
Branches	4	4	0	1	100.00%	100.00%
FEC Expressions	4	4	0	1	100.00%	100.00%
Toggles	62	61	1	1	98.38%	98.38%

Recursive Hierarchical Coverage Details:

Total Coverage: 79.57% **86.35%**

Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Statements	74	72	2	1	97.29%	97.29%
Branches	52	49	3	1	94.23%	94.23%
FEC Expressions	4	4	0	1	100.00%	100.00%
FEC Conditions	8	6	2	1	75.00%	75.00%
Toggles	226	159	67	1	70.35%	70.35%
FSMs	13	10	3	1	76.92%	81.25%
States	5	5	0	1	100.00%	100.00%
Transitions	8	5	3	1	62.50%	62.50%



Questa Design Coverage

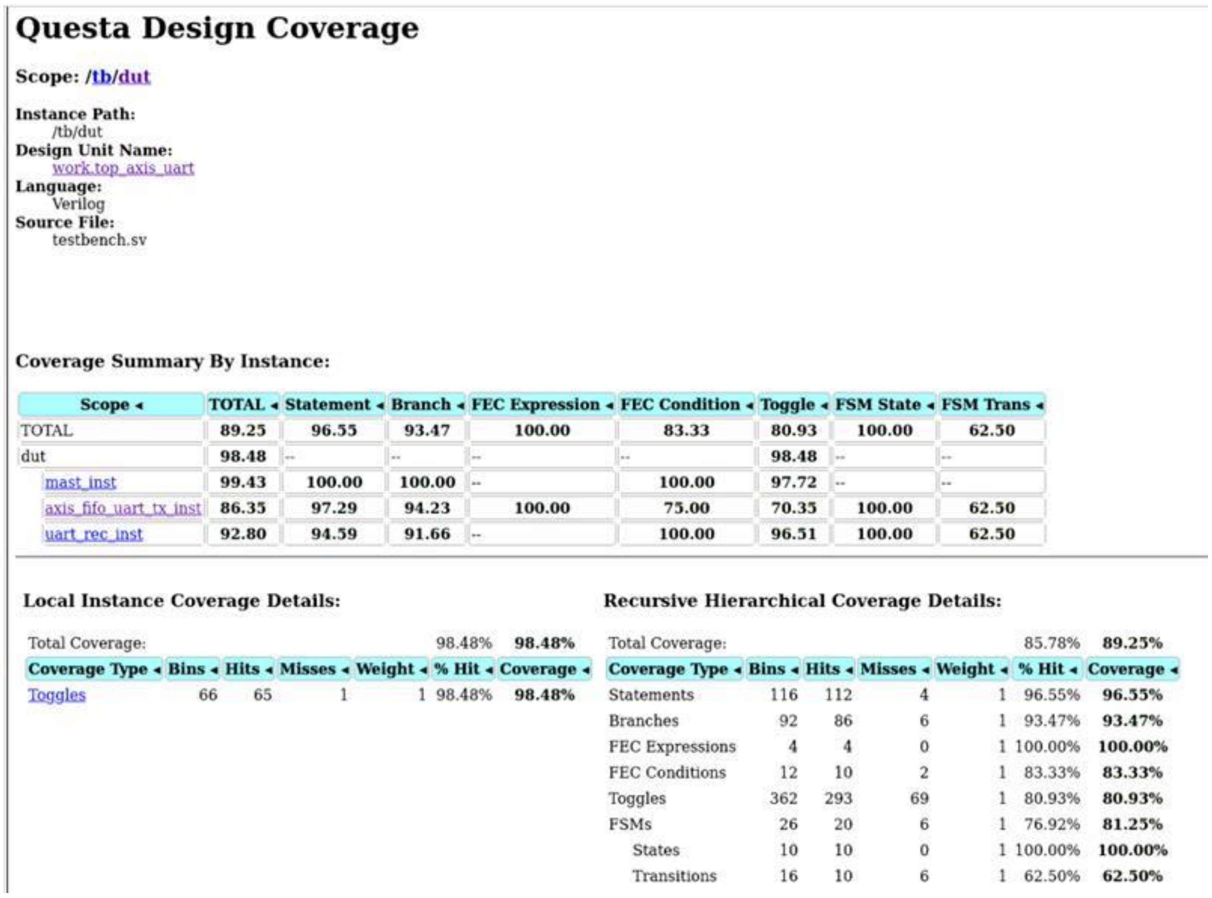
Scope: /tb/dut/mast_inst

Instance Path:
/tb/dut/mast_inst
Design Unit Name:
work.axis_master_inp
Language:
Verilog
Source File:
design.v

Local Instance Coverage Details:

Total Coverage: 98.18% **99.43%**

Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Statements	5	5	0	1	100.00%	100.00%
Branches	4	4	0	1	100.00%	100.00%
FEC Conditions	2	2	0	1	100.00%	100.00%
Toggles	44	43	1	1	97.72%	97.72%



Questa Design Coverage

Scope: /tb/dut

Instance Path:
/tb/dut
Design Unit Name:
work.top_axis_uart
Language:
Verilog
Source File:
testbench.sv

Coverage Summary By Instance:

Scope	TOTAL	Statement	Branch	FEC Expression	FEC Condition	Toggle	FSM State	FSM Trans
TOTAL	89.25	96.55	93.47	100.00	83.33	80.93	100.00	62.50
dut	98.48	--	--	--	--	98.48	--	--
mast_inst	99.43	100.00	100.00	--	100.00	97.72	--	--
axis_fifo_uart_tx_inst	86.35	97.29	94.23	100.00	75.00	70.35	100.00	62.50
uart_rec_inst	92.80	94.59	91.66	--	100.00	96.51	100.00	62.50

Local Instance Coverage Details:

Total Coverage: 98.48% **98.48%**

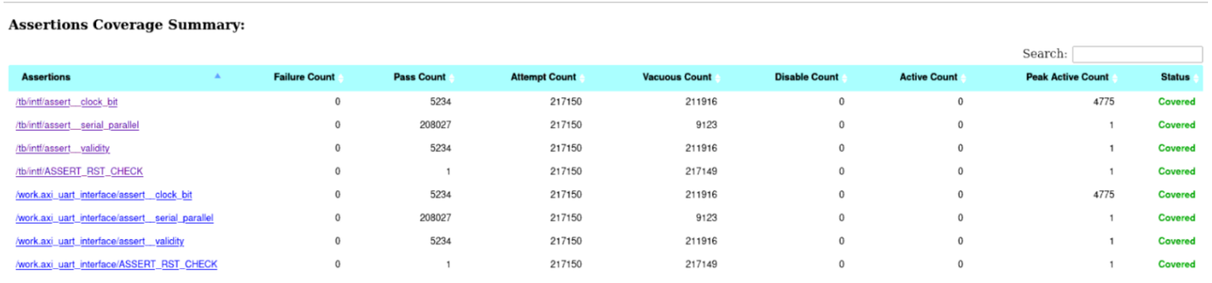
Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Toggles	66	65	1	1	98.48%	**98.48%**

Recursive Hierarchical Coverage Details:

Total Coverage: 85.78% **89.25%**

Coverage Type	Bins	Hits	Misses	Weight	% Hit	Coverage
Statements	116	112	4	1	96.55%	**96.55%**
Branches	92	86	6	1	93.47%	**93.47%**
FEC Expressions	4	4	0	1	100.00%	**100.00%**
FEC Conditions	12	10	2	1	83.33%	**83.33%**
Toggles	362	293	69	1	80.93%	**80.93%**
FSMs	26	20	6	1	76.92%	**81.25%**
States	10	10	0	1	100.00%	**100.00%**
Transitions	16	10	6	1	62.50%	**62.50%**

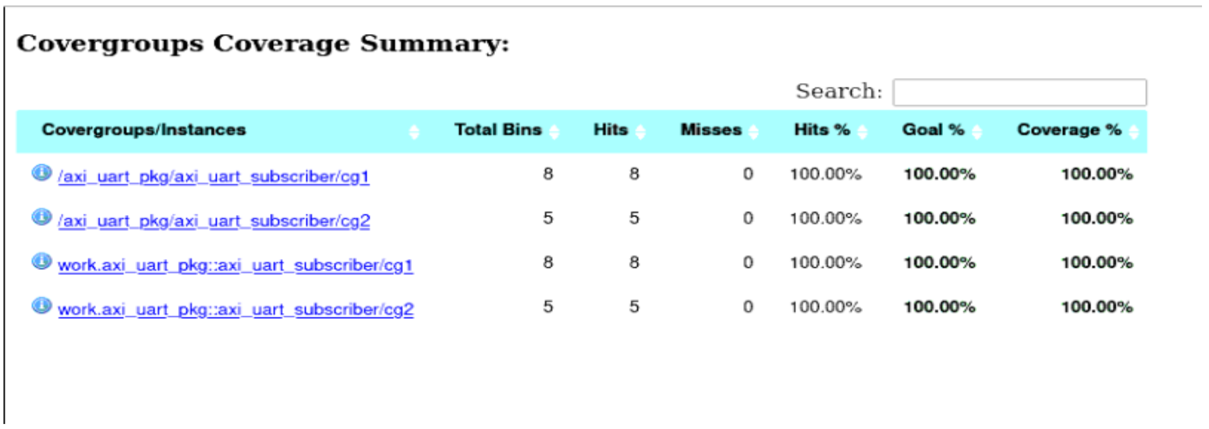
5.1. Functional Coverage



Assertions Coverage Summary:

Search:

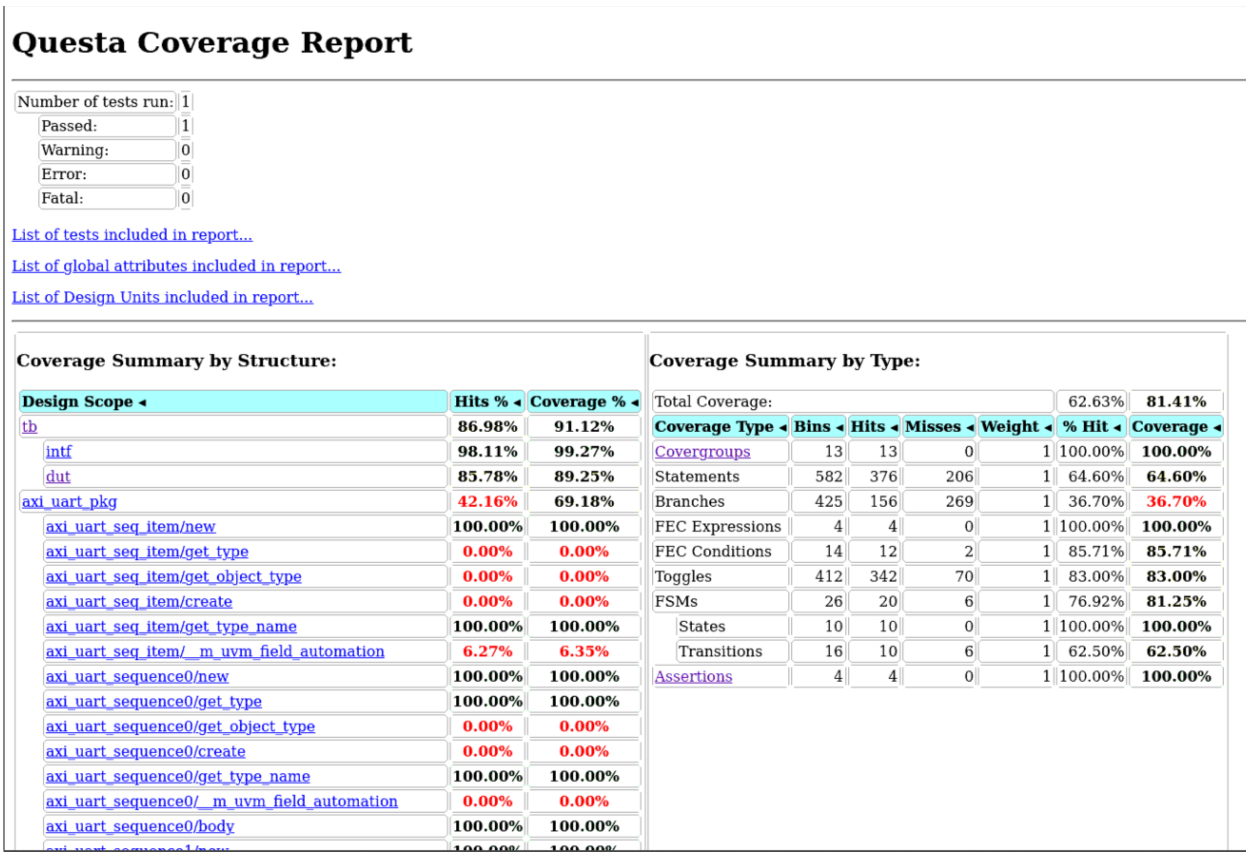
Assertions	Failure Count	Pass Count	Attempt Count	Vacuous Count	Disable Count	Active Count	Peak Active Count	Status
/tb/intf/assert__clock_bit	0	5234	217150	211916	0	0	4775	Covered
/tb/intf/assert__serial_parallel	0	208027	217150	9123	0	0	1	Covered
/tb/intf/assert__validity	0	5234	217150	211916	0	0	1	Covered
/tb/intf/ASSERT_RST_CHECK	0	1	217150	217149	0	0	1	Covered
/work.axi_uart_interface/assert__clock_bit	0	5234	217150	211916	0	0	4775	Covered
/work.axi_uart_interface/assert__serial_parallel	0	208027	217150	9123	0	0	1	Covered
/work.axi_uart_interface/assert__validity	0	5234	217150	211916	0	0	1	Covered
/work.axi_uart_interface/ASSERT_RST_CHECK	0	1	217150	217149	0	0	1	Covered



Covergroups Coverage Summary:

Search:

Covergroups/Instances	Total Bins	Hits	Misses	Hits %	Goal %	Coverage %
/axi_uart_pkg/axi_uart_subscriber/cg1	8	8	0	100.00%	**100.00%**	**100.00%**
/axi_uart_pkg/axi_uart_subscriber/cg2	5	5	0	100.00%	**100.00%**	**100.00%**
work.axi_uart_pkg::axi_uart_subscriber/cg1	8	8	0	100.00%	**100.00%**	**100.00%**
work.axi_uart_pkg::axi_uart_subscriber/cg2	5	5	0	100.00%	**100.00%**	**100.00%**



Questa Coverage Report

Number of tests run:	1
Passed:	1
Warning:	0
Error:	0
Fatal:	0

List of tests included in report...

List of global attributes included in report...

List of Design Units included in report...

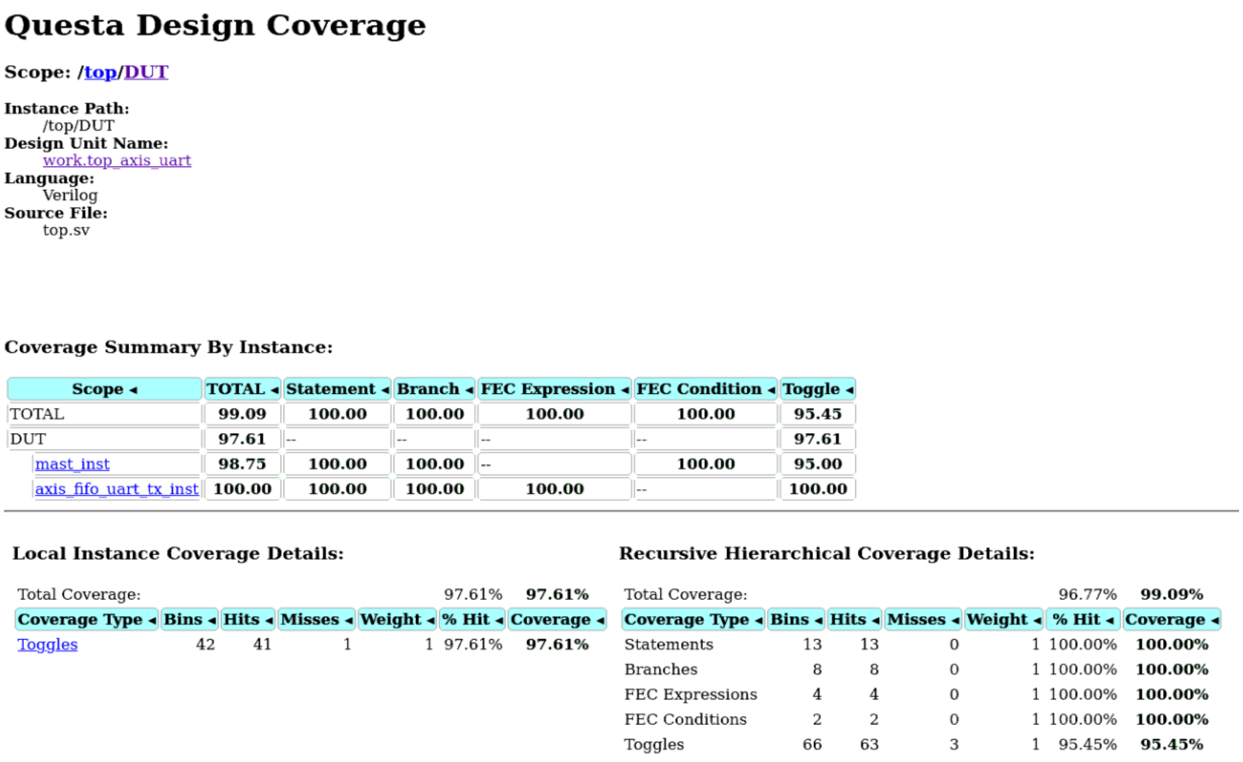
Coverage Summary by Structure:

Design Scope ◂	Hits % ◂	Coverage % ◂
tb	86.98%	91.12%
intf	98.11%	99.27%
dut	85.78%	89.25%
axi_uart_pkg	42.16%	69.18%
axi_uart_seq_item/new	100.00%	100.00%
axi_uart_seq_item/get_type	0.00%	0.00%
axi_uart_seq_item/get_object_type	0.00%	0.00%
axi_uart_seq_item/create	0.00%	0.00%
axi_uart_seq_item/get_type_name	100.00%	100.00%
axi_uart_seq_item/__m_uvm_field_automation	6.27%	6.35%
axi_uart_sequence0/new	100.00%	100.00%
axi_uart_sequence0/get_type	100.00%	100.00%
axi_uart_sequence0/get_object_type	0.00%	0.00%
axi_uart_sequence0/create	0.00%	0.00%
axi_uart_sequence0/get_type_name	100.00%	100.00%
axi_uart_sequence0/__m_uvm_field_automation	0.00%	0.00%
axi_uart_sequence0/body	100.00%	100.00%
axi_uart_sequence1/new	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					62.63%	81.41%
Coverage Type ◂	**Bins ◂**	**Hits ◂**	**Misses ◂**	**Weight ◂**	**% Hit ◂**	**Coverage ◂**
Covergroups	13	13	0	1	100.00%	100.00%
Statements	582	376	206	1	64.60%	64.60%
Branches	425	156	269	1	36.70%	36.70%
FEC Expressions	4	4	0	1	100.00%	100.00%
FEC Conditions	14	12	2	1	85.71%	85.71%
Toggles	412	342	70	1	83.00%	83.00%
FSMs	26	20	6	1	76.92%	81.25%
States	10	10	0	1	100.00%	100.00%
Transitions	16	10	6	1	62.50%	62.50%
Assertions	4	4	0	1	100.00%	100.00%

5.2. Code Coverage After Exclusion



Questa Design Coverage

Scope: /top/DUT

Instance Path:
/top/DUT
Design Unit Name:
work.top_axis_uart
Language:
Verilog
Source File:
top.sv

Coverage Summary By Instance:

Scope ◂	TOTAL ◂	Statement ◂	Branch ◂	FEC Expression ◂	FEC Condition ◂	Toggle ◂
TOTAL	99.09	100.00	100.00	100.00	100.00	95.45
DUT	97.61	--	--	--	--	97.61
mast_inst	98.75	100.00	100.00	--	100.00	95.00
axis_fifo_uart_tx_inst	100.00	100.00	100.00	100.00	--	100.00

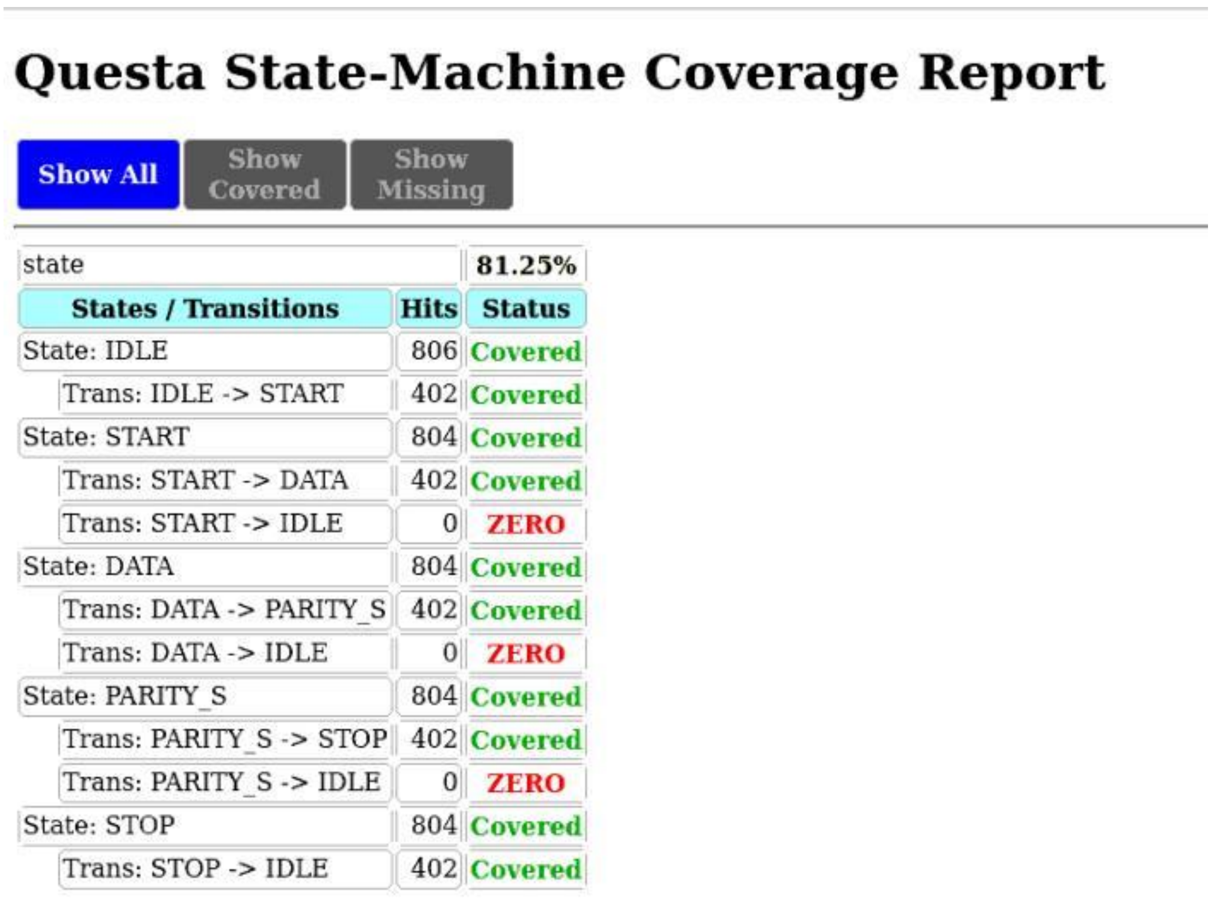
Local Instance Coverage Details:

Total Coverage:					97.61%	97.61%
Coverage Type ◂	**Bins ◂**	**Hits ◂**	**Misses ◂**	**Weight ◂**	**% Hit ◂**	**Coverage ◂**
Toggles	42	41	1	1	97.61%	97.61%

Recursive Hierarchical Coverage Details:

Total Coverage:					96.77%	99.09%
Coverage Type ◂	**Bins ◂**	**Hits ◂**	**Misses ◂**	**Weight ◂**	**% Hit ◂**	**Coverage ◂**
Statements	13	13	0	1	100.00%	100.00%
Branches	8	8	0	1	100.00%	100.00%
FEC Expressions	4	4	0	1	100.00%	100.00%
FEC Conditions	2	2	0	1	100.00%	100.00%
Toggles	66	63	3	1	95.45%	95.45%

5.3. What did we exclude and why ?



Questa State-Machine Coverage Report

Show All | Show Covered | Show Missing

state		81.25%
States / Transitions	**Hits**	**Status**
State: IDLE	806	Covered
Trans: IDLE -> START	402	Covered
State: START	804	Covered
Trans: START -> DATA	402	Covered
Trans: START -> IDLE	0	ZERO
State: DATA	804	Covered
Trans: DATA -> PARITY_S	402	Covered
Trans: DATA -> IDLE	0	ZERO
State: PARITY_S	804	Covered
Trans: PARITY_S -> STOP	402	Covered
Trans: PARITY_S -> IDLE	0	ZERO
State: STOP	804	Covered
Trans: STOP -> IDLE	402	Covered

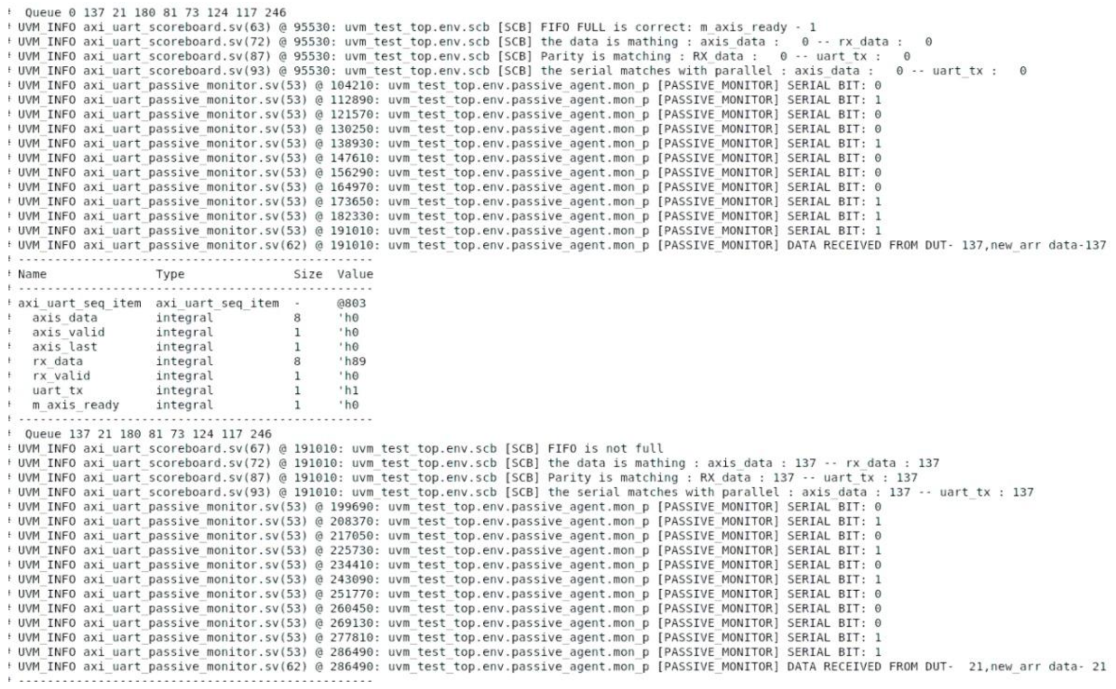
- Here we have excluded the **uart_rec_inst** module because this module was having the state transitions i.e. DATA->IDLE, PARITY->IDLE, START->IDLE.
- This was not hit because it was not feasible to calculate the time of each bit change of the serial uart_rx and pass the reset at that point of time, if we try to do this it will hinder the transaction and remove all the previous stream of data.
- So we excluded the module to achieve this coverage.

- **Assertions**

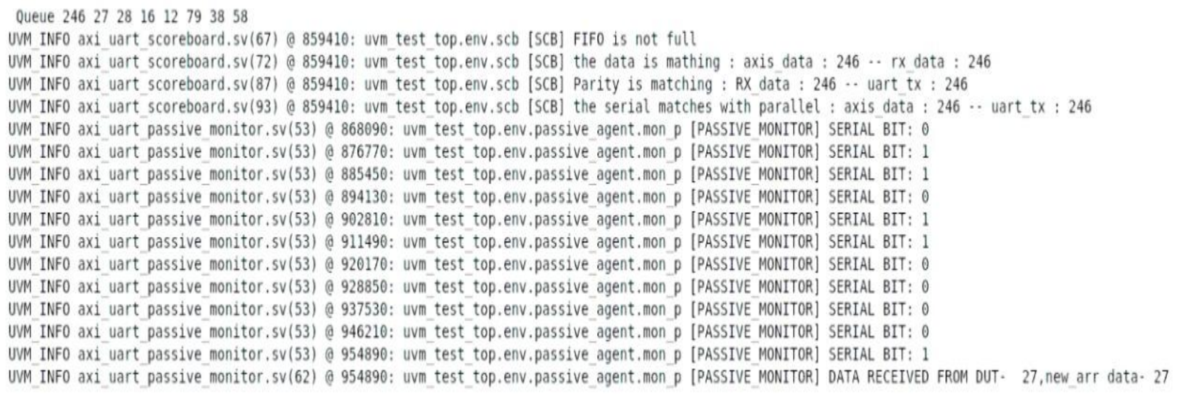
Assertion	Description	Status
RESET	While reset is asserted all the ouputs should be 0 except m_axis_ready	PASS
VALIDITY	When valid is asserted high then all data and valid signal should be known	PASS
OUTPUT _AFTER_4774	When axis_data is present the rx_data should be present after 4774 clock cycles	PASS
Parallel_serial	When RX_DATA is high, the serial data should be present and RX_VALID SHOULD BE HIGH	PASS

CHAPTER – 6

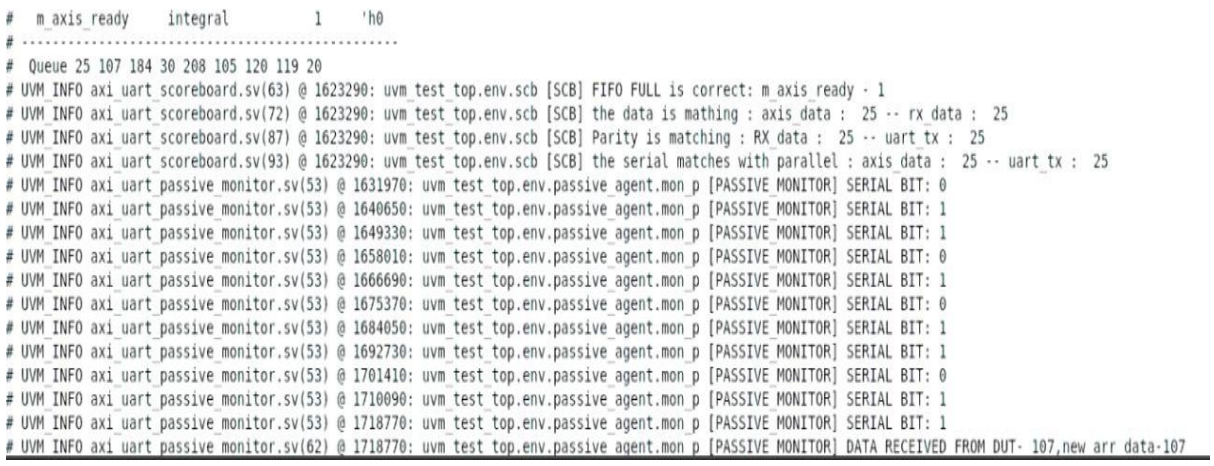
6. OUTPUT



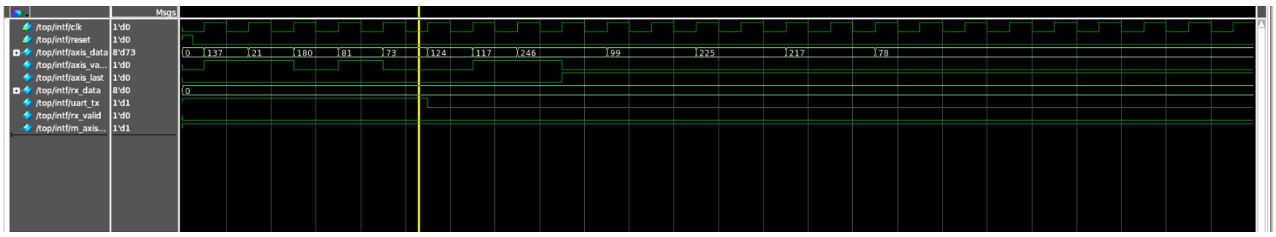
```
'  Queue 0 137 21 180 81 73 124 117 246
' UVM_INFO axi_uart_scoreboard.sv(63) @ 95530: uvm_test_top.env.scb [SCB] FIFO FULL is correct: m_axis_ready - 1
' UVM_INFO axi_uart_scoreboard.sv(72) @ 95530: uvm_test_top.env.scb [SCB] the data is mathing : axis_data :   0 -- rx_data :   0
' UVM_INFO axi_uart_scoreboard.sv(87) @ 95530: uvm_test_top.env.scb [SCB] Parity is matching : RX_data :   0 -- uart_tx :   0
' UVM_INFO axi_uart_scoreboard.sv(93) @ 95530: uvm_test_top.env.scb [SCB] the serial matches with parallel : axis_data :   0 -- uart_tx :   0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 104210: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 112890: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 121570: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 130250: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 138930: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 147610: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 156290: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 164970: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 173650: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 182330: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 191010: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(62) @ 191010: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] DATA RECEIVED FROM DUT- 137,new_arr data-137
' ------------------------------------------------
' Name              Type              Size  Value
' ------------------------------------------------
' axi_uart_seq_item axi_uart_seq_item -     @803
'   axis_data       integral          8     'h0
'   axis_valid      integral          1     'h0
'   axis_last       integral          1     'h0
'   rx_data         integral          8     'h89
'   rx_valid        integral          1     'h0
'   uart_tx         integral          1     'h1
'   m_axis_ready    integral          1     'h0
' ------------------------------------------------
'  Queue 137 21 180 81 73 124 117 246
' UVM_INFO axi_uart_scoreboard.sv(67) @ 191010: uvm_test_top.env.scb [SCB] FIFO is not full
' UVM_INFO axi_uart_scoreboard.sv(72) @ 191010: uvm_test_top.env.scb [SCB] the data is mathing : axis_data : 137 -- rx_data : 137
' UVM_INFO axi_uart_scoreboard.sv(87) @ 191010: uvm_test_top.env.scb [SCB] Parity is matching : RX_data : 137 -- uart_tx : 137
' UVM_INFO axi_uart_scoreboard.sv(93) @ 191010: uvm_test_top.env.scb [SCB] the serial matches with parallel : axis_data : 137 -- uart_tx : 137
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 199690: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 208370: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 217050: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 225730: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 234410: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 243090: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 251770: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 260450: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 269130: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 277810: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(53) @ 286490: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
' UVM_INFO axi_uart_passive_monitor.sv(62) @ 286490: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] DATA RECEIVED FROM DUT-  21,new_arr data- 21
' ------------------------------------------------
```



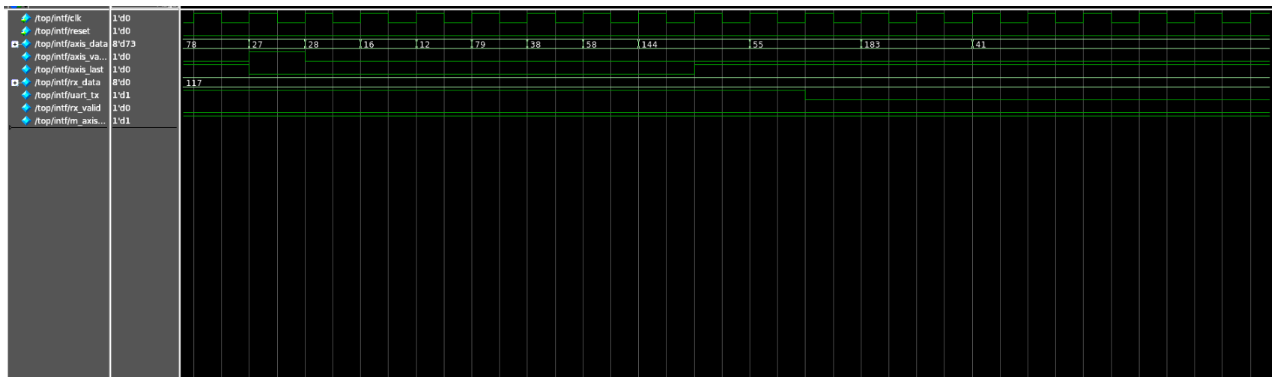
```
 Queue 246 27 28 16 12 79 38 58
UVM_INFO axi_uart_scoreboard.sv(67) @ 859410: uvm_test_top.env.scb [SCB] FIFO is not full
UVM_INFO axi_uart_scoreboard.sv(72) @ 859410: uvm_test_top.env.scb [SCB] the data is mathing : axis_data : 246 -- rx_data : 246
UVM_INFO axi_uart_scoreboard.sv(87) @ 859410: uvm_test_top.env.scb [SCB] Parity is matching : RX_data : 246 -- uart_tx : 246
UVM_INFO axi_uart_scoreboard.sv(93) @ 859410: uvm_test_top.env.scb [SCB] the serial matches with parallel : axis_data : 246 -- uart_tx : 246
UVM_INFO axi_uart_passive_monitor.sv(53) @ 868090: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 876770: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
UVM_INFO axi_uart_passive_monitor.sv(53) @ 885450: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
UVM_INFO axi_uart_passive_monitor.sv(53) @ 894130: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 902810: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
UVM_INFO axi_uart_passive_monitor.sv(53) @ 911490: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
UVM_INFO axi_uart_passive_monitor.sv(53) @ 920170: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 928850: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 937530: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 946210: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
UVM_INFO axi_uart_passive_monitor.sv(53) @ 954890: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
UVM_INFO axi_uart_passive_monitor.sv(62) @ 954890: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] DATA RECEIVED FROM DUT-  27,new_arr data- 27
```



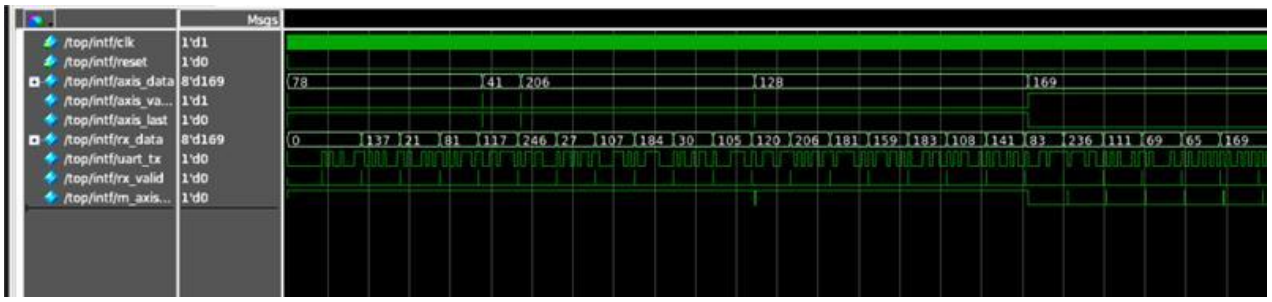
```
#   m_axis_ready     integral          1     'h0
# ------------------------------------------------
#  Queue 25 107 184 30 208 105 120 119 20
# UVM_INFO axi_uart_scoreboard.sv(63) @ 1623290: uvm_test_top.env.scb [SCB] FIFO FULL is correct: m_axis_ready - 1
# UVM_INFO axi_uart_scoreboard.sv(72) @ 1623290: uvm_test_top.env.scb [SCB] the data is mathing : axis_data :  25 -- rx_data :  25
# UVM_INFO axi_uart_scoreboard.sv(87) @ 1623290: uvm_test_top.env.scb [SCB] Parity is matching : RX_data :  25 -- uart_tx :  25
# UVM_INFO axi_uart_scoreboard.sv(93) @ 1623290: uvm_test_top.env.scb [SCB] the serial matches with parallel : axis_data :  25 -- uart_tx :  25
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1631970: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1640650: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1649330: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1658010: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1666690: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1675370: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1684050: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1692730: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1701410: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 0
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1710090: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(53) @ 1718770: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] SERIAL BIT: 1
# UVM_INFO axi_uart_passive_monitor.sv(62) @ 1718770: uvm_test_top.env.passive_agent.mon_p [PASSIVE_MONITOR] DATA RECEIVED FROM DUT- 107,new_arr data-107
```



```
/top/intf/clk          1'd0
/top/intf/reset        1'd0
/top/intf/axis_data    8'd73
/top/intf/axis_va...   1'd0
/top/intf/axis_last    1'd0
/top/intf/rx_data      8'd0
/top/intf/uart_tx      1'd1
/top/intf/rx_valid     1'd0
/top/intf/m_axis...    1'd1
```



```
/top/intf/clk          1'd0
/top/intf/reset        1'd0
/top/intf/axis_data    8'd73
/top/intf/axis_va...   1'd0
/top/intf/axis_last    1'd0
/top/intf/rx_data      8'd0
/top/intf/uart_tx      1'd1
/top/intf/rx_valid     1'd0
/top/intf/m_axis...    1'd1
```



```
/top/intf/clk          1'd1
/top/intf/reset        1'd0
/top/intf/axis_data    8'd169
/top/intf/axis_va...   1'd1
/top/intf/axis_last    1'd0
/top/intf/rx_data      8'd169
/top/intf/uart_tx      1'd0
/top/intf/rx_valid     1'd0
/top/intf/m_axis...    1'd0
```

CHAPTER – 7

7. Conclusion and Learning

- The AXI-UART verification environment was successfully designed using the Universal Verification Methodology (UVM), ensuring modularity, scalability, and reusability across components.
- The verification flow demonstrated proper interaction between AXI (parallel) and UART (serial) protocols, validating data integrity across transmission and reception paths.
- Active and passive agents were used to monitor and drive signals efficiently, while the scoreboard compared UART TX and RX data to ensure correctness.
- Global variables and synchronization techniques were implemented to maintain timing accuracy during parallel-to-serial conversions.
- Functional coverage and assertions were used to validate corner cases, ensuring robust verification.
- Through this project, key learnings included:
    - Understanding and implementing AXI and UART protocol-level verification.
    - Developing reusable UVM components such as agents, monitors, drivers, and scoreboards.
    - Using configuration databases and analysis ports for data sharing between components.
    - Handling timing-sensitive verification involving baud rate calculations and clock synchronization.
    - Improving debugging and reporting skills using UVM messaging and topology tracing.

Overall, this project provided a strong practical understanding of building a complete UVM-based verification environment, enhancing both technical and conceptual verification skills.