(WS-VTP-00073)

Simulation & Verification Strategy

Last updated: **2025-05-22**

Altera Confidential

Contents

List of Figures

List of Tables

1.0 Introduction

The objectives of this verification are:

- To verify the Sail River Modules and its sub-modules.
- To read the data from PCAP generated from oRAN Studio and inject it to eCPRI streaming interface and compare the observed data of each module with its expected data computed from Reference Model.

1.1 Scope

The scope of this verification strategy document is to provide an insight on the verification environment to be developed for the Sail River Module. The DUT is tested based on the test cases involving the features mentioned in the Test Plan. To verify the functionality of Sail River using System Verilog UVM environment.

Completion of Verification will be based on the below list of items tested:

- 100 MHz ,60 MHz and Mixed Band width (100 - 60 & 60 - 100).
- QPSK, 16QAM, 64QAM and 256QAM modulation schemes for each AxCs.
- DUT tested with 273 PRBs for 100 MHz and 162 PRBs for 60 MHz of input data.
- DUT tested with 1 Subframe i.e for 1 ms Test vector.
- PCAP data bandwidth is 100 MHz and 60 MHz for 8 AxCs.
- Observed data and Expected Data are compared in Scoreboard.

1.2 Goals

The following goals to support the project objectives:

- To verify the Sink and Source data of every block.

1.3 Tools and Languages

Table 1-1. Tools and Languages Used

Sl.No	Tools/Language	Version
1	Questa sim Full Edition	23.1
2	Quartus	24.1
3	System Verilog	

2.0 Strategy and Structure

The overall strategy of this verification is to verify Sail River using:

- UVM based System Verilog test bench.
- Checkers for individual blocks.

2.1 Verification Progress

The verification progress is quantified based on the test cases pass with different modulation schemes like QPSK, 16QAM, 64QAM, 256QAM and with 8 Number of AxCs. eAxC_ID can be a 16-bit random number, in this example it is considered as sequential number for demonstration. The input data is created using Oran Studio generated PCAP with combination of all modulation schemes.

Table 2-1. Default test vectors modulation schemes

eAxC ID	Modulation Scheme
1	QPSK
2	16QAM
3	64QAM
4	256QAM
5	256QAM
6	64QAM
7	16QAM
8	QPSK

2.2 Features to be verified

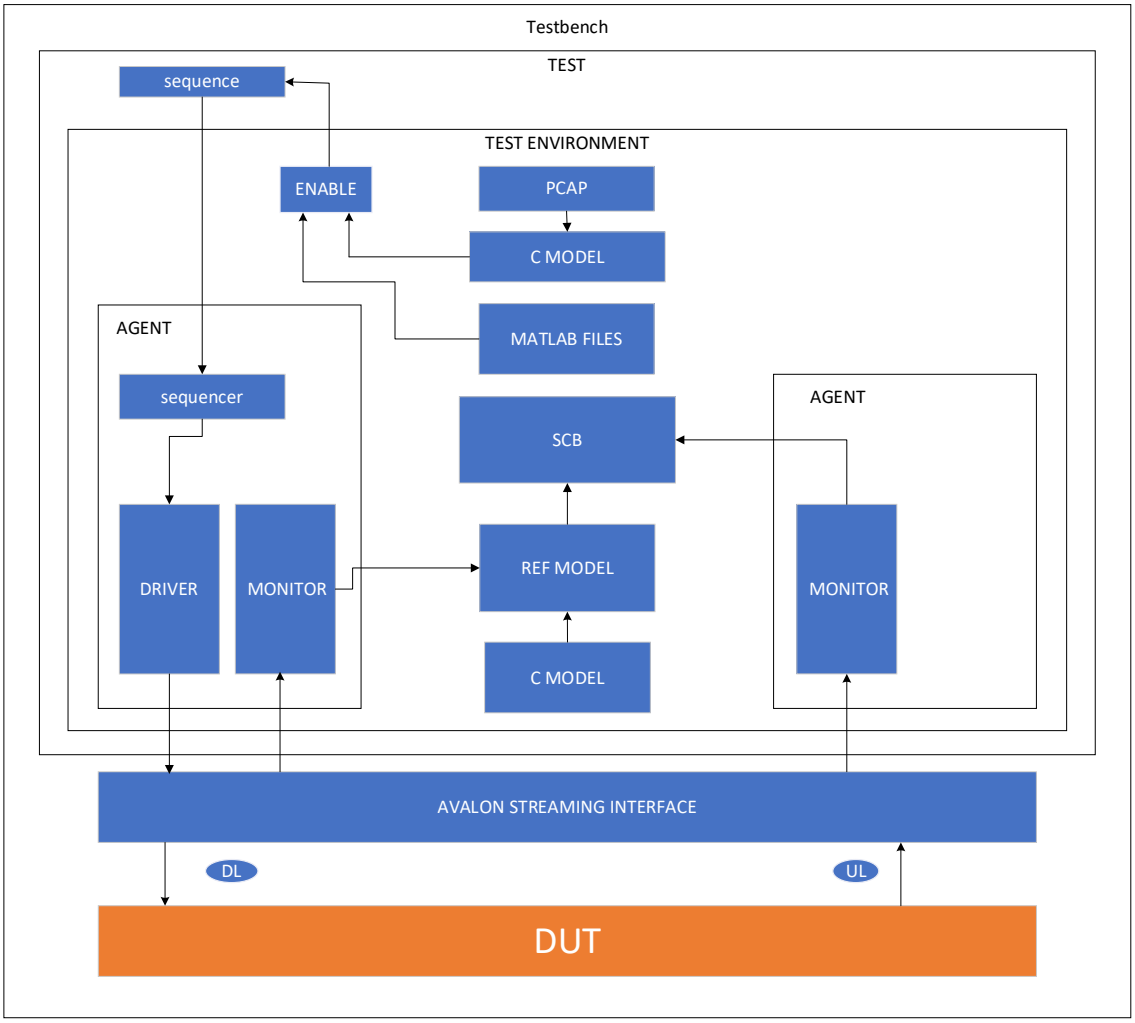
The verification is performed to ensure the functional, interface and performance aspects of the Design. The verification features for both uplink and downlink are described in **WS-VTP-00072 Sail River Gen III.xlsx.**

3.0 Testbench Architecture

3.1 Testbench Architecture

- Below is the overview of Test Bench Architecture

Figure 3-1. Verification Architecture Diagram



3.2 Testbench Environment

3.2.1 UVM Verification Components

The UVM testbench environment consists of UVC. It provides structured components, sequences, and interfaces.

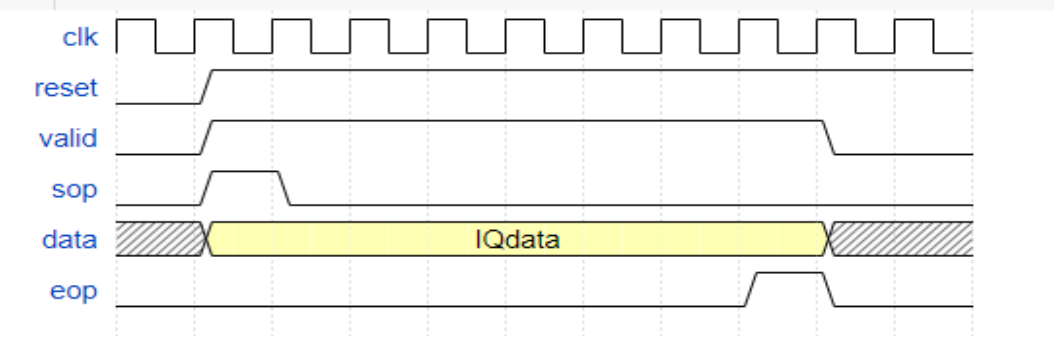
The common components for a UVC are listed below:

- Driver
- Monitor
- Agent
- Env
- Sequencer

3.2.2 Interface

3.2.2.1 Avalon Streaming Interface

Figure 3-2. Timing diagram for Avalon Streaming Interface



Driver

- Driver receives transaction data (eCPRI packet) from the sequence (Read from PCAP).
- The driver schedules AVST streaming data from the Transaction Data (eCPRI packet) to Streaming interface.
- DUT gets the stimulus from the Streaming interface.
- A series of transaction data will be handled by the UVM handshake mechanism.

Monitor

- In Active Mode, the monitor will collect the data which was driven to the DUT via Interface and send it to the Reference Model.
- In Passive Mode, the monitor will collect the data from the DUT via the Interface and send it to the Scoreboard.
- Monitor points using Avalon Streaming Interface are 0,1,3,4,5,28,29,30 mentioned in **Figure 3-4**.

- All the above-mentioned points will be connected to the Scoreboard and compared against the reference data of those points.

Scoreboard

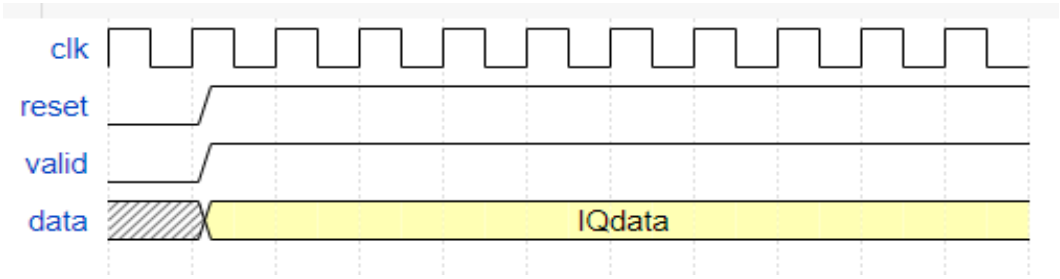
- It is a verification component that contains checkers and verifies the functionality of a design.
- It receives transactions from the Reference Model for each Monitor point. The raw data is then captured and then pushed into reference queue (reference data). The output from DUT is then captured as observed data, the reference data is then popped out and compared with observed data.
- It will report on the Status for each packet PASS/FAIL for each block.

Reference Model

- It is a verification component that emulates the expected behavior of DUT.
- For each Monitor point mentioned above the corresponding reference model computation which will emulate RTL behavior will be done and for some blocks it will integrate the C-Model and sent to the Scoreboard.
- Every Packet will be computed or read from C-Model and will be sent to Scoreboard for comparison and will be compared against that block.

3.2.2.2 Avalon Streaming Interface without SOP and EOP

Figure 3-3. Timing Diagram for Avalon Streaming Interface without SOP and EOP



Driver

- It drives the streaming data to DUT.
- It has a handshake mechanism to get series of transaction data.
- It will send the transaction data received from the sequence (Read from PCAP) to the interface.

Monitor

- In Active Mode, the monitor will collect the data which was driven to the DUT via Interface and send it to the Reference Model.
- In Passive Mode, the monitor will collect the data from the DUT for every 1 Symbol via the Interface and send it to the Scoreboard.
- Monitor points using Avalon Streaming Interface without SOP, EOP are 2, 6-27 mentioned in **Figure 3-4**.

- All the above-mentioned points will be connected to the Scoreboard and compared against the reference data of those points.

Scoreboard

- It is a verification component that contains checkers and verifies the functionality of a design.
- It receives transactions from the Reference Model for each Monitor point. The raw data is then captured and then pushed into reference queue (reference data). The output from DUT is then captured as observed data, the reference data is then popped out and compared with observed data.
- It will report on the Status for each packet PASS/FAIL for each block.

Reference Model

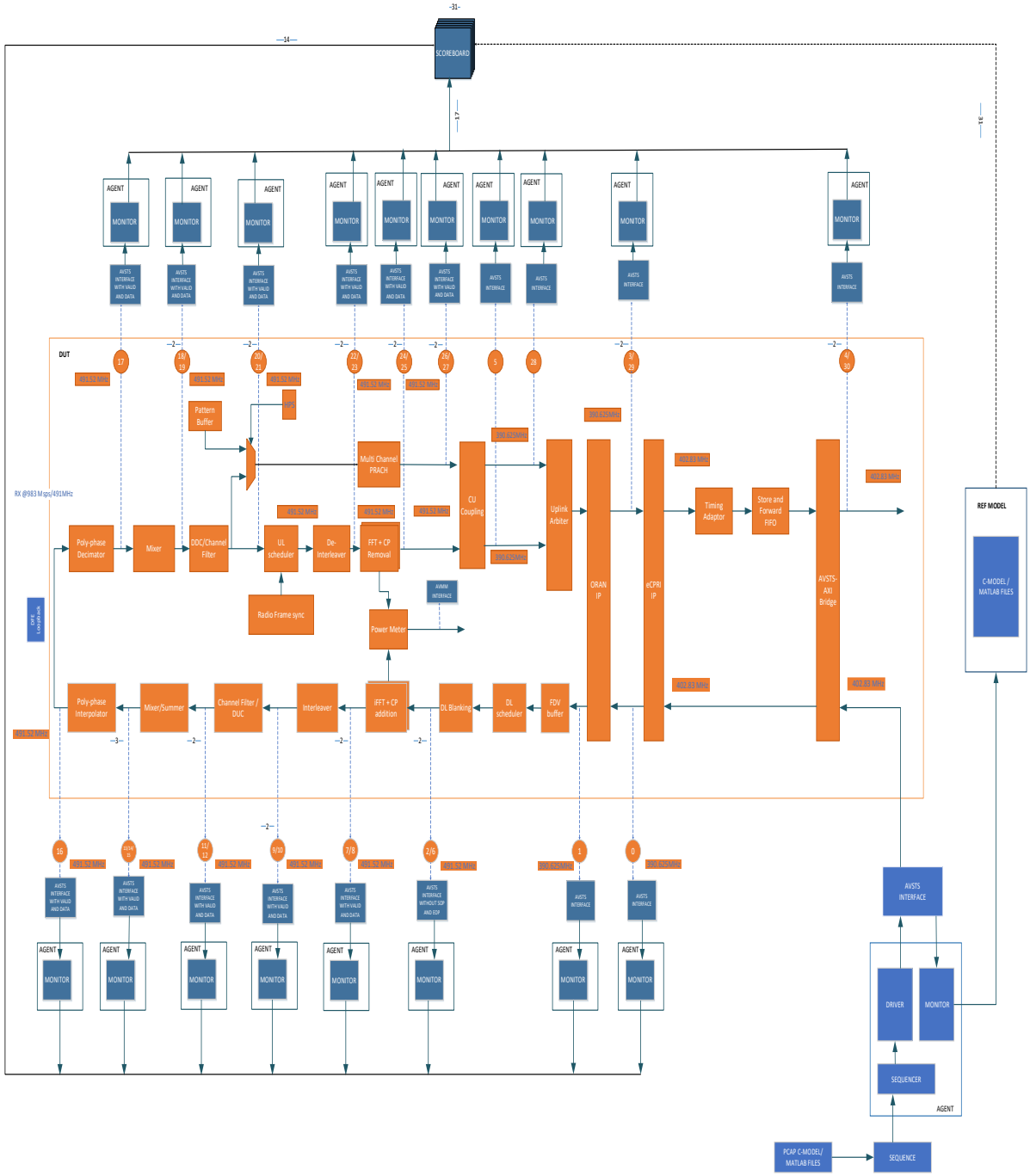
- It is a verification component that emulates the expected behavior of DUT.
- For each Monitor point mentioned above the corresponding reference model computation which will emulate RTL behavior will be done and for some blocks it will integrate the C-Model and sent to the Scoreboard.
- Each Symbol will be computed and sent to Scoreboard for comparison and will be compared against that block symbol by symbol.

3.2.2.3 Configurations

DFE Loopback

- In the case of DFE Loopback, the driving point will be at the AVST-AXI Bridge DL input, and the checkers will be at all the mentioned points in **Figure 3-4**.
- The DFE Loopback will be done at the output of Interpolator in DL path to the input of Decimator at UL path.
- Each Checker points mentioned will have a corresponding Interface (Interface for each block is mentioned in UVC) and Monitors which will be connected to the scoreboard and compared against the expected as shown in **Figure 3-4**.
- The reference data (Expected) will be formed for each block either by C-MODEL or by computing in Reference model or by using files as shown in **Figure 3-4**.

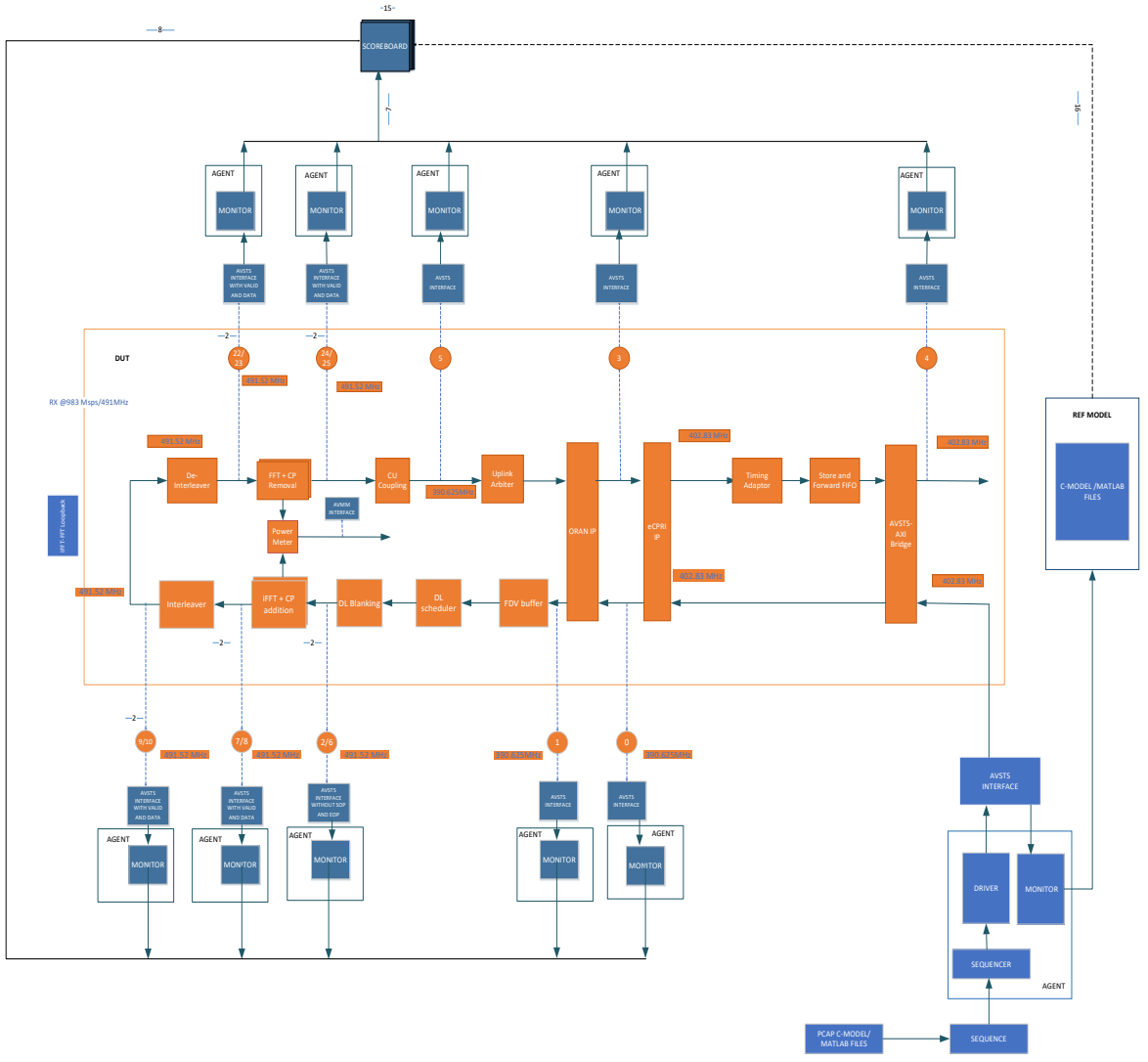
Figure 3-4. Block diagram of DFE Loopback



IFFT-FFT Loopback

- In the case of IFFT-FFT Loopback, the driving point will be at the AVST-AXI Bridge DL input, and the checkers will be at all the mentioned points in **Figure 3-5.**
- The IFFT-FFT Loopback will be done at the output of Interleaver in DL path to the input of De-interleaver at UL path when ifft-fft loopback enable register is 1.
- Each Checker points mentioned will have a corresponding Interface (Interface for each block is mentioned in UVC) and Monitors which will be connected to the scoreboard and compared against the expected as shown in **Figure 3-5**.
- The reference data (Expected) will be formed for each block either by C-MODEL or by computing in Reference model or by using files as shown in **Figure 3-5**.

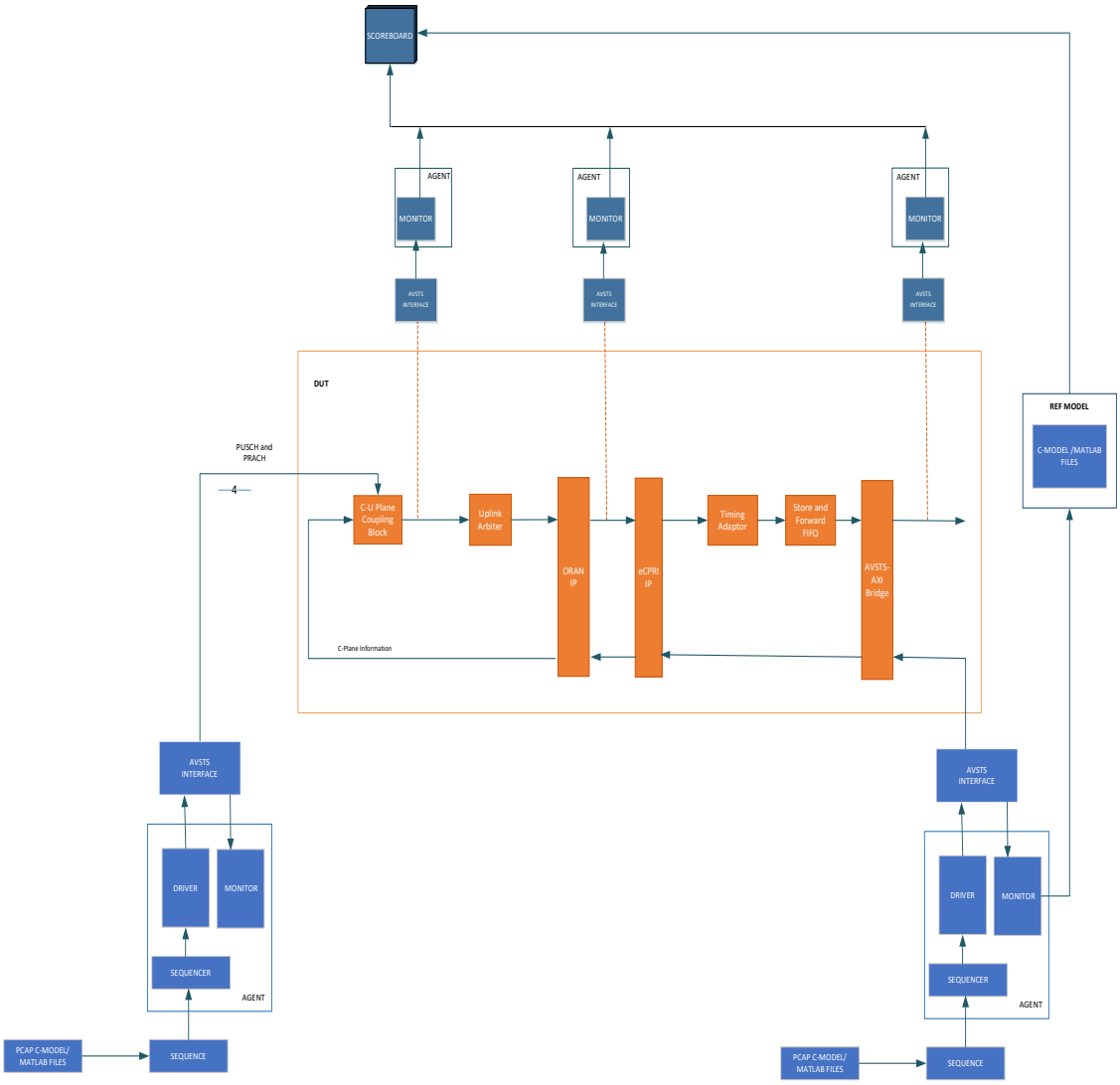
Figure 3-5. Block diagram of IFFT-FFT Loopback



ECPRI-ORAN Subsystem

- In the case of ECPRI-ORAN Subsystem, the driving point will be at the AVST-AXI Bridge DL input, and the checkers will be at all the mentioned points in **Figure 3-6**.
- The UL C-Plane information is sent via DL path. PUSCH and PRACH data will be fed directly to Coupling block.
- Each Checker points mentioned will have a corresponding Interface (Interface for each block is mentioned in UVC) and Monitors which will be connected to the scoreboard and compared against the expected as shown in **Figure 3-6**.
- The reference data (Expected) will be formed for each block either by C-MODEL or by computing in Reference model or by using files as shown in **Figure 3-6**.

Figure 3-6. Block diagram of ECPRI-ORAN Subsystem



3.3 Testing Methods

The TESTING is done with the method below.

- PCAP WITH C Model testing and FILE based testing

3.3.1 PCAP with C-Model testing and File-based testing

In this PCAP with C Model and File based testing, the input is given from PCAP, and it is read using a C Model by the sequence inside testcase, and the output at each block is collected and compared in the scoreboard by using the reference model. The reference for this reference model is computed by using C Models or files.
The C Models used here can be described as follows:

- C Model is used to replicate the expected output.
- We use 5 C Models in the verification mentioned below.
  - PCAP C Model
  - Decompression C Model
  - Compression C Model
  - iFFT C Model
  - FFT C Model

PCAP C Model: This C Model is used to configure the input data. This model is used to read PCAP data and send it to sequencer.

Decompression C Model: This C Model is used to replicate the output of U plane decompression i.e., ORAN De-mapper output.

Compression C Model: This C Model is used to replicate the output of U plane Compression i.e., ORAN Mapper output.

iFFT C Model: This C Model is used to replicate the output of iFFT Block. It is generated from iFFT DSPBA.

FFT C Model: This C Model is used to replicate the output of FFT Block. It is generated from iFFT DSPBA.

PCAP with FILE based reference is done for the following blocks:

- IFFT
- Interleaver
- DUC
- DL Mixer
- Interpolator
- Decimator
- UL Mixer
- Summer
- DDC
- De-Interleaver
- FFT
- PRACH

3.4 UVC

Below is the list of UVC's Used or Needed for Sail River Testbench UVM Environment.

Table 3-1. Sail River UVC

Python script: -

- It will convert PCAP into .txt file
- Decompression
- Blanking

The .txt file will be converted into .MAT File with the help of MATLAB Script.

The .MAT File will be used by the MATLAB Model to create expected files for all modules and the expected data are stored into Files.

Simulation will do the file-based comparison approach.

The input PCAP file is getting injected into eCPRI IP with the help of C model.

Modules	UVC	Model
eCPRI Demapper	Avalon Streaming 64 bits	System Verilog
oRAN Demapper	Avalon Streaming	System Verilog Decompression- C Model
DL Blanking	Avalon streaming but no SOP's and EOP's	System Verilog
iFFT both Lineups	Avalon Streaming Valid and Data 32 bits	File Generated from MATLAB Model
Power Meter	Avalon Memory Mapped and Data 32 bits	System Verilog
Interleaver both Lineups	Avalon Streaming Valid and Data 32 bits	File Generated from MATLAB Model
DUC	Avalon Streaming Valid and Data 128 bits	File Generated from MATLAB Model
Mixer DL		
Summer		
Interpolator	Avalon Streaming Valid and Data 256 bits	
Decimator	Avalon Streaming Valid and Data 128 bits	
Mixer UL		
DDC	Avalon Streaming Valid and Data 32 bits	
Deinterleaver		File Generated from MATLAB Model
Power Meter	Avalon Memory Mapped and Data 32 bits	System Verilog
FFT	Avalon Streaming Valid and Data 32 bits	File Generated from MATLAB Model
PRACH Checker		File Generated from MATLAB Model

CU_Coupling	Avalon Streaming Data Width's Differ for each block	System Verilog
oRAN Mapper		System Verilog Compression- C Model
eCPRI Mapper (AVST-AXI Bridge)		System Verilog

4.0 Document Revision History

Table 4-1. Revision History

Date	Version	Changes
2025-05-22	0.1	Gen III Verification Strategy Document