Gigabit Ethernet VIP Development using URM

Jagvinder Yadav
Gaurav Singh

Agenda

TATA
TATA ELXSI LIMITED

- Introduction
  - Objective, GbE VIP, URM, Verification Environment.
- Testbench Architecture
- Components of UVC
  - Sequence Driver
  - BFM
  - Monitor
  - Assertions
  - Coverage
  - Scoreboard

Objective

- To develop an automated test-environment which incorporates following features of URM

  - Constrained random stimulus generation

  - Coverage driven verification

  - Reusability

  - Modularity

  - Scalability

GbE VIP Features

TATA
TATA ELXSI LIMITED

- Configurable 10Mbps/100Mbps/1Gbps Ethernet MAC and compliant with MII/GMII Standards
- Half/Full Duplex mode data transfer
- Full CSMA/CD support - jamming, backoff and automatic retransmission
- Supports generation of
  - Data Frames, Control Frames (Pause Frames), Jumbo Frames, VLAN Tagged Frames (IEEE P802.1Q)

GbE VIP Features contd..

- Programmable error injection
- Error detection
- Introducing collisions randomly
- Built-in functional coverage

Universal Reuse Methodology (URM)

- URM is a complete methodology for developing high quality reusable verification components.

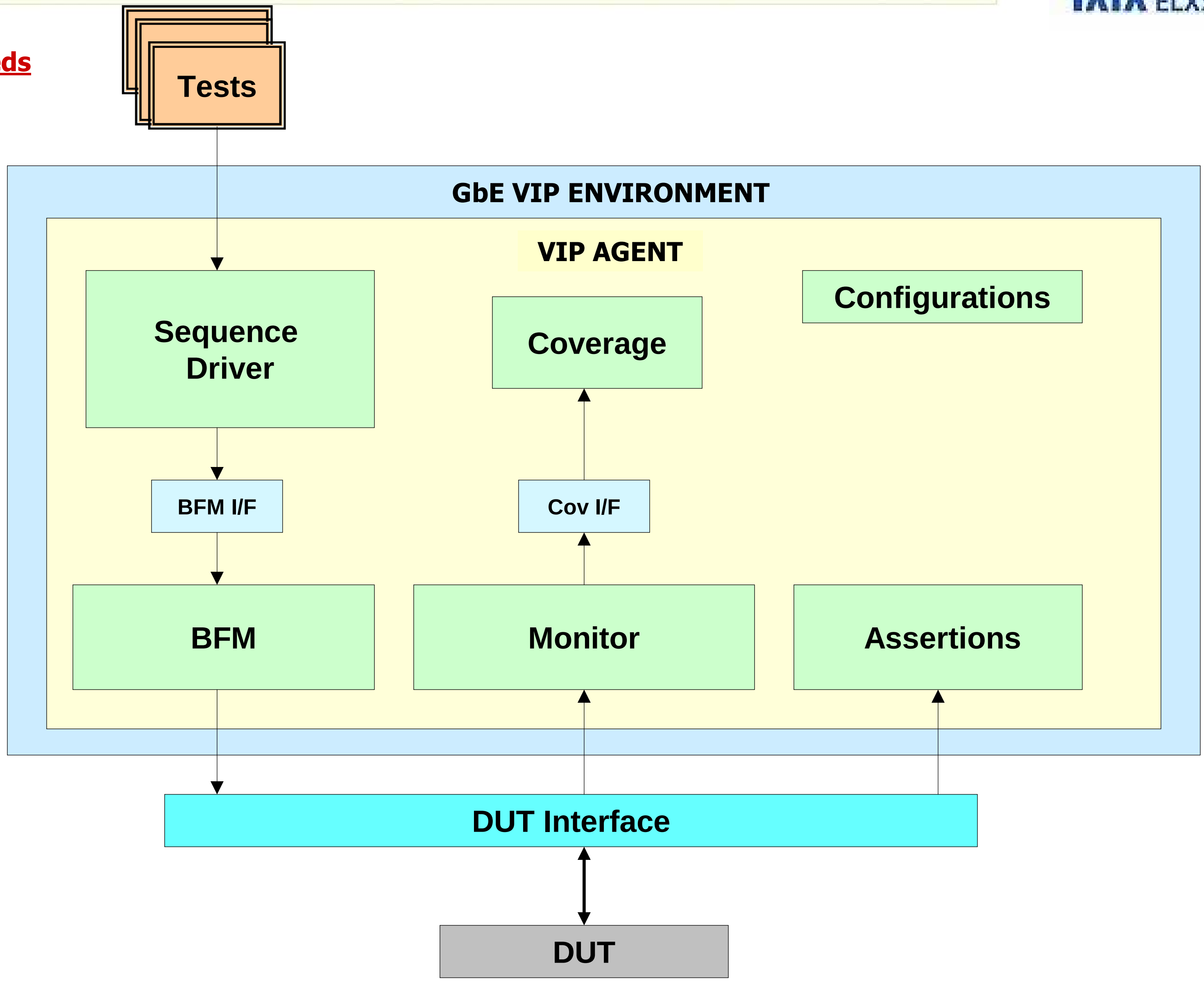
Features:-

- Language Independent (System verilog, Specman ‘e’, VHDL, System C).
- Fully constrained-random, automated test-stimulus generation and functional coverage driven approach.
- Module based and class based architectures
  - Generated transactions are classes but TB infrastructure is module based.
  - Easier to adopt.
  - Object-oriented approach.
  - Testbench automation using base classes.

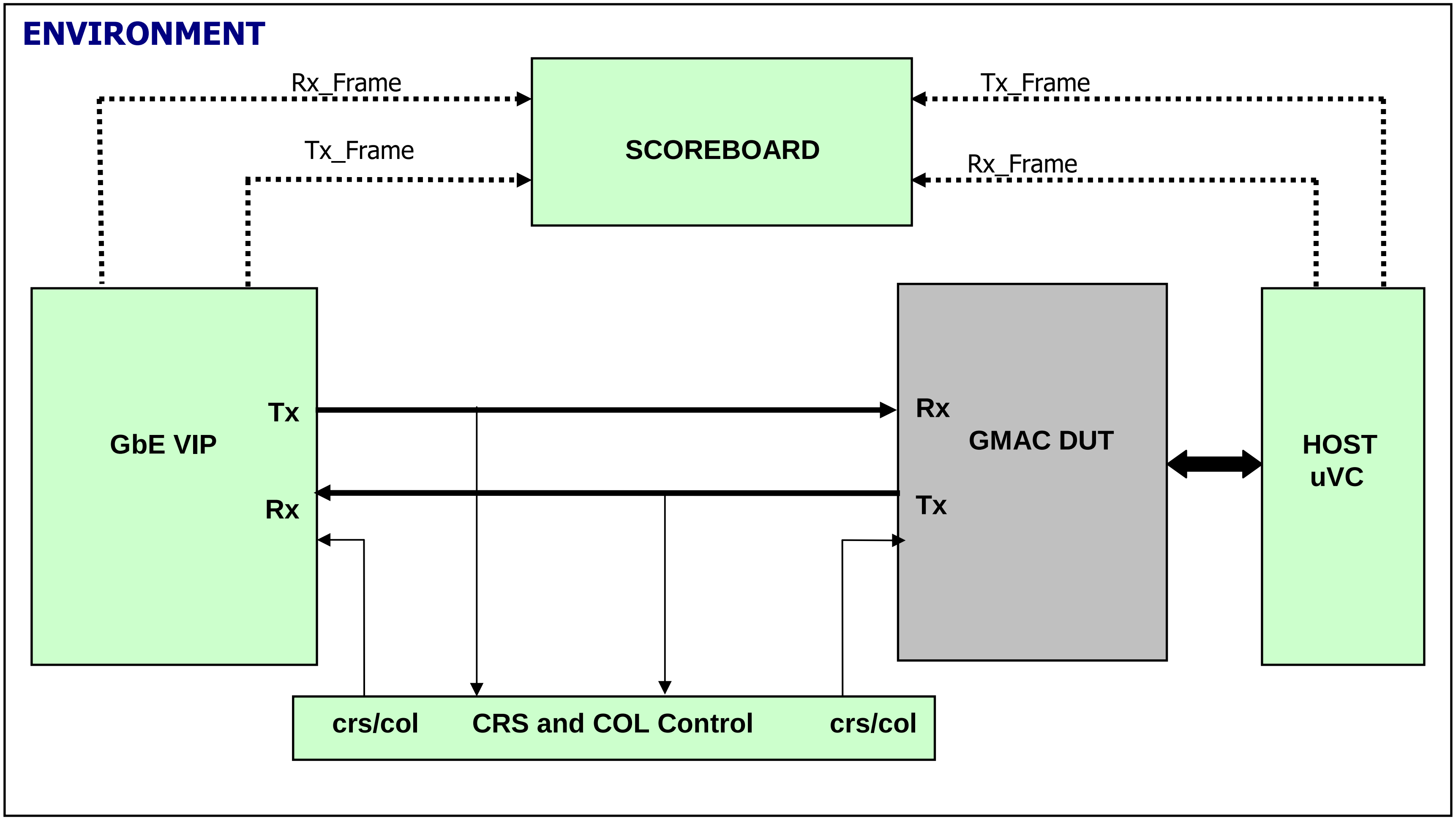
GbE uVC Architecture

Random Seeds

12345671
98765431
23902904
45843298
23432432
24324322
55252255
09273822
13814791
4098e092
23432424
24242355
25262622
26452454
24524522
25262622
26452454
24524522
25262622
26452454
24524522
25262622
26452454
24524522
13814791
4098e092
23432424
24242355
25262622
26452454
24524522
25262622
26452454
24524522
25262622
26452454
25262622
26452454
24524522
25262622
26452454
24524522
25262622
26452454



Verification Environment



Transaction Class

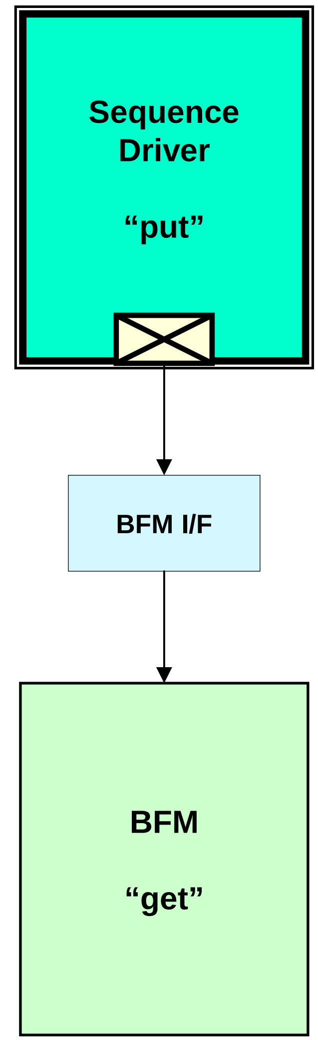
- Base class containing different packet fields of GbE frame.
- Defined inside a package and can be imported across modules of TB to access transactions.
- User-defined constraints, basic properties and methods.
- Post-randomize functions (CRC calculation, Padding and Extension field for GbE).
- Can be extended to add/override constraints, properties, methods.

Transaction Class Example

```
package trans_P;
   typedef struct packed
   {
      bit [56:0] preamble;
      bit [47:0] sfd;
      bit [1:0] length;
      bit [100:0] payload;
   } struct_packet_S;

   class tel_data_C;
      rand  struct_packet_S packet_obj;   //object of struct defined as rand
      //Adding basic constraints
      constraint  basic_const
      {
         packet_obj.preamble == {28{2'b10}};
         packet_obj.sfd == 8'b10101011;
      }

      //add properties to the class
      //add various methods to the class (functions and tasks)
      function void post_randomize();
        for (int i=0; i<length; i++)
        begin
           payload[i] = $urandom;
        end
        fpadding();
        fcrc();
        fextension();
     endfunction
   endclass

endpackage
```

Driver

- Generates packet instance containing random GbE packets required by DUT.
- Generates sequences of random transactions with user controllability.
- Connected to BFM via channel (BFM Interface).
- Passes generated transactions to BFM by putting into BFM interface.
- Directed and random testcase compatible.



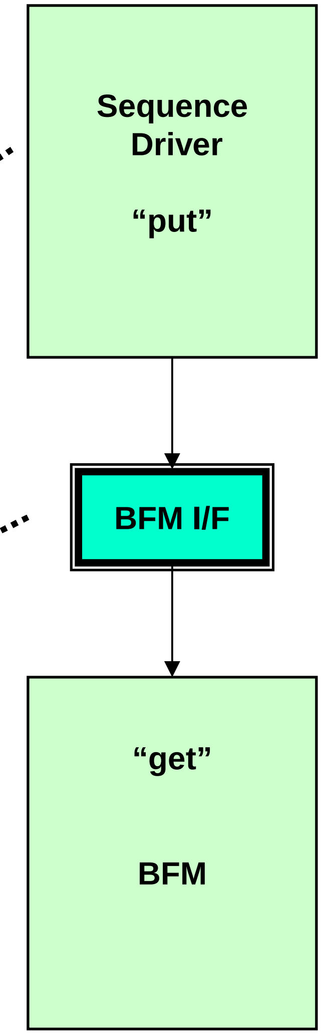
BFM Interface

- A channel to pass transactions from driver to BFM.
- Blocks the next transaction until current transaction is executed by the BFM.

```
for (int i=1; i<= count; i++)
begin
  success = packet_obj.randomize();
  // blocking put
  bfm_if.put(packet_obj);
end
```

```
task automatic put(
      input struct_packet packet1);
     @bfm_if.done;
endtask
task automatic get(
      output struct_packet packet1);
endtask
```

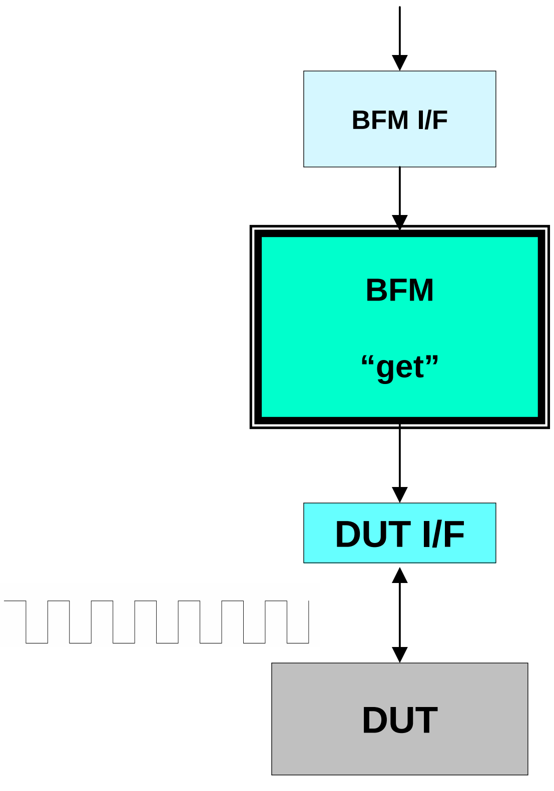
```
forever begin
   // blocking get
   bfm_if.get(packet_obj);
   ........
   ........
 -> bfm_if.done;
end
```



BFM

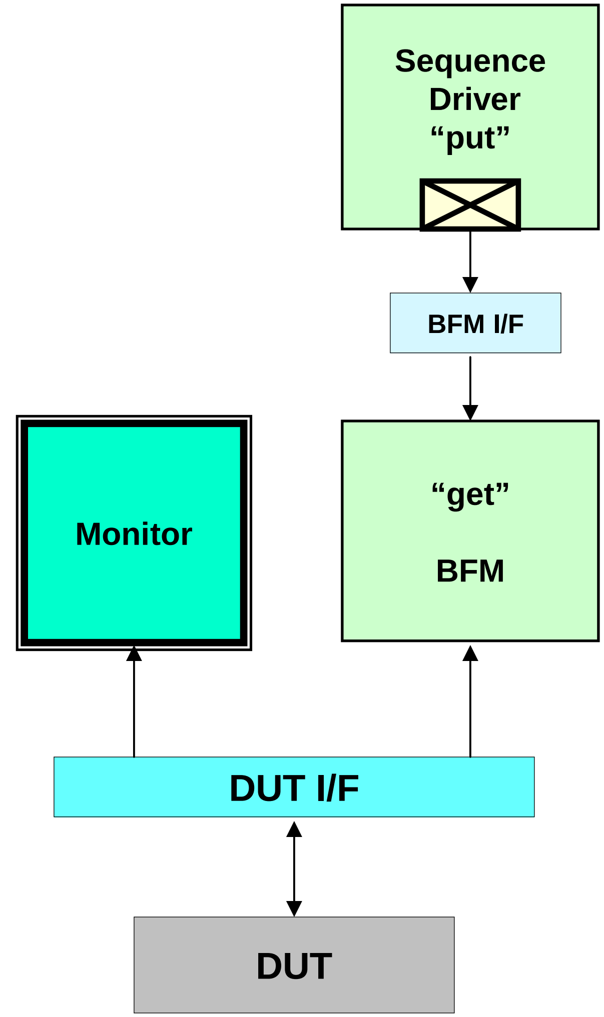
TATA
TATA ELXSI LIMITED

- Protocol-specific module implementing low-level protocol.
- Fetches random packet generated by driver from BFM Interface.
- Designed to work in Pull mode, pulls next transaction from BFM interface.
- Blocked if there are no pending transactions.
- Converts transactions into bit-stream and drives it to DUT via DUT Interface following GbE Protocol.



Monitor

- Connected to DUT Interface to receive transactions from DUT and monitoring bus-level activities.
- Collects data (bit stream) from bus and and reassembles it to form GbE packet.
- Protocol checkers, bug reporting mechanism are a part of monitor.
- Sends the packet to scoreboard for data integrity checking.
- Passes data to coverage module for functional coverage calculation and analysis.



Assertions

TATA
TATA ELXSI LIMITED

- Placed at interface side.
- Report any protocol violations at I/F level.
- Monitor Tx/Rx signals ensuring that GbE protocol violation doesn't occur.
- ABV methodology is used for assertions development (implemented using SVA).

Coverage

Calculates and analyzes functional coverage of GbE VIP.

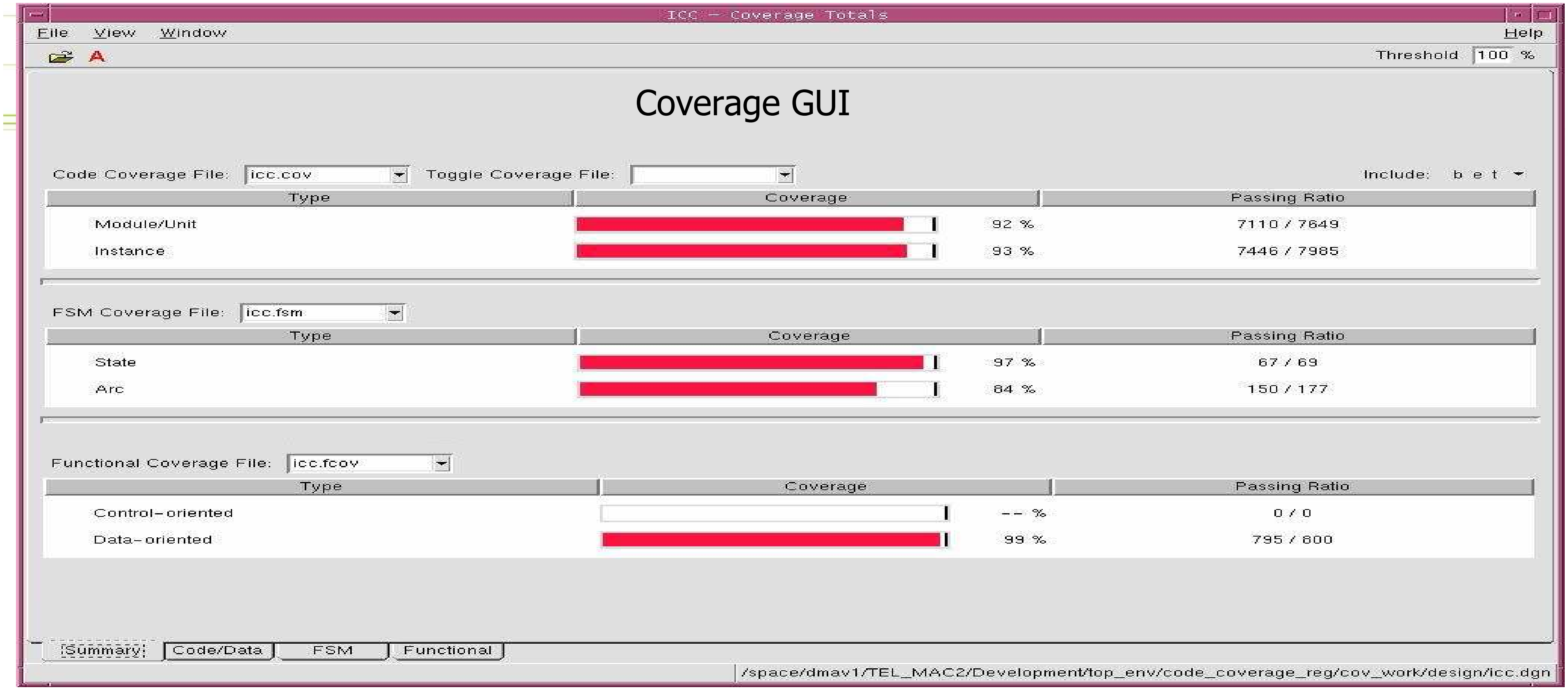
- Stimulus coverage
  - Generated data packet coverage

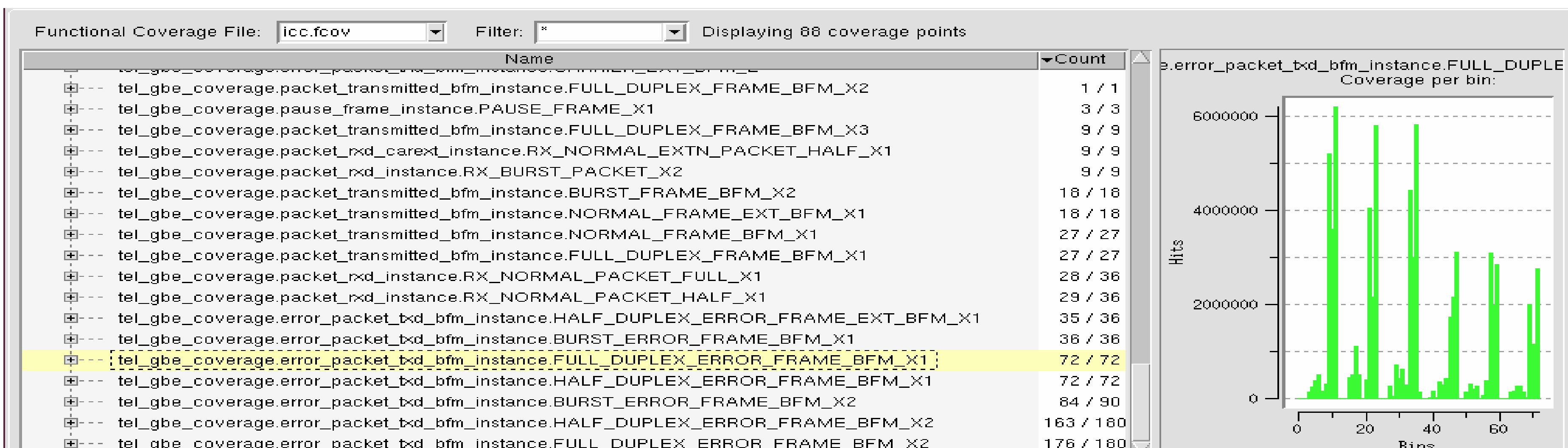
- Checker coverage
  - Protocol checkers coverage

- Scenario coverage
  - Corner-case scenario coverage

- Cross coverage
  - Coverage between two or more coverpoints within a covergroup
    e.g. frame type with length and payload type.

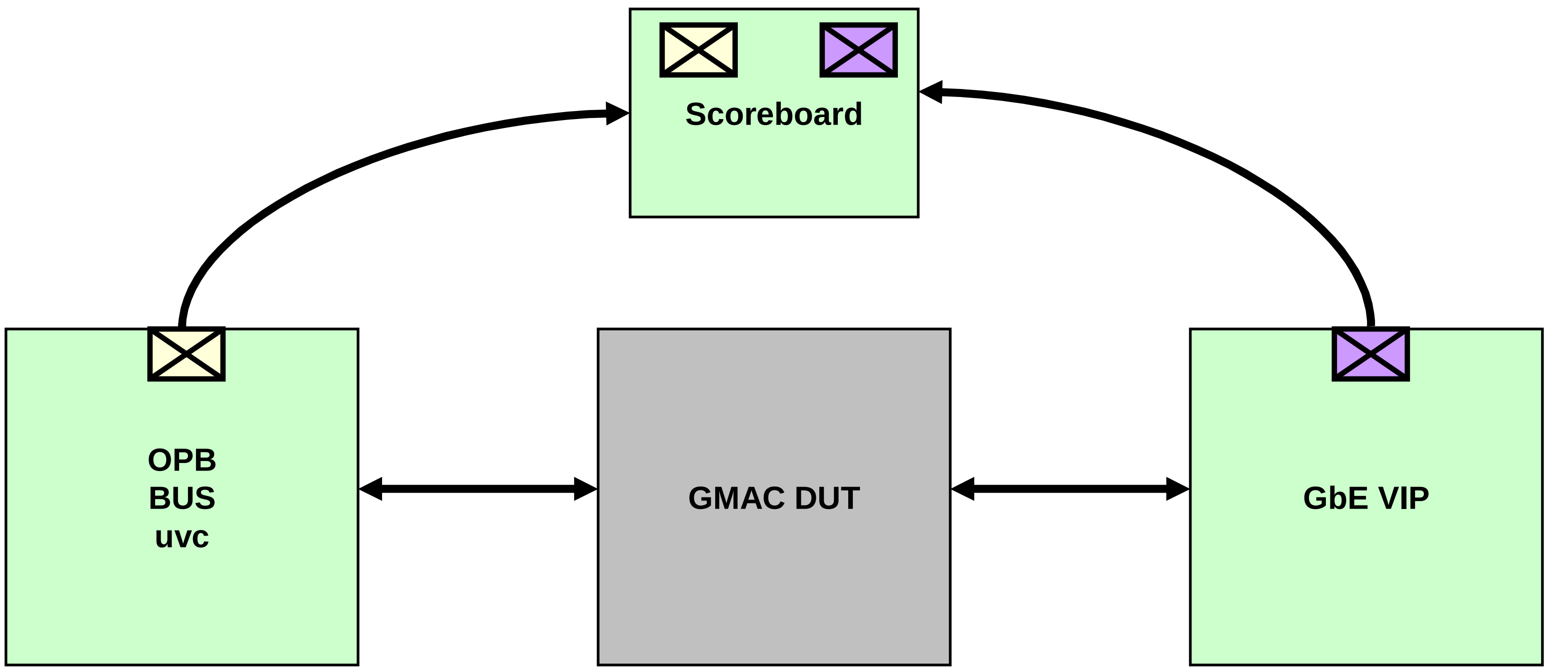
Coverage GUI





Scoreboard

- Abstract and un-timed reference models.
- Perform data-integrity check on data received.
- Contains
  - Packet driven to DUT.
  - Data driven by DUT in response to it.



Conclusion

TATA
TATA ELXSI LIMITED

- URM helped us to create an efficient, scalable and reusable test environment.

- Constrained based random stimulus generation helped us to reduce the time involved in creating directed test scenarios.

- Strong functional coverage analysis model which helped us to cover corner case scenarios also.

Thank You !!!