SIEMENS

LOGICAL LAYER

Verification Plan

Name: Ali Ahmed

Name: Karim Samy

Name: Walid Salah

Name: Mohamed Salah

Table of Contents

Introduction: 1
Verification Environment: 2
Test Environment: 2
System Verilog Mailboxes: 3
Stimulus Generator: 3
Driver: 4
Monitor: 4
Scoreboard: 4
Reference Model: 5
Virtual Sequencer: 5
Transaction Flow: 8
Design Limitations: 11
SYSTEM Level VERIFICATION: 12
Interfaces: 12
Interface 1 ports (Upper Layer Interface): 13
Interface 2 Ports (Electrical Layer Interface): 14
Interface 3 Ports(Configuration Space Interface): 15
ENVIRONMENTS TRANSACTIONS: 15
Transport Layer Transaction 15
Configuration Spaces Transaction 16
Electrical Layer transactions: 16
USB4 Logical Layer Flow 18
Test Scenarios 22
Traceability Matrix: 23
Task Breakdown: 24
Duration of each task: 25

Introduction:

This Document contains the full verification plan proposal for the Logical Layer Architecture. For this Design, we will make a Class-based Verification Environment to be able to test all design features in an efficient and organizable way. The Document flow will be as follows:

First, The Verification IP(VIP) Architecture will be presented along with a thorough description of the role of each component.

Next, full description of the transaction flow through the environment will be represented to visualize the operation of the Environment.

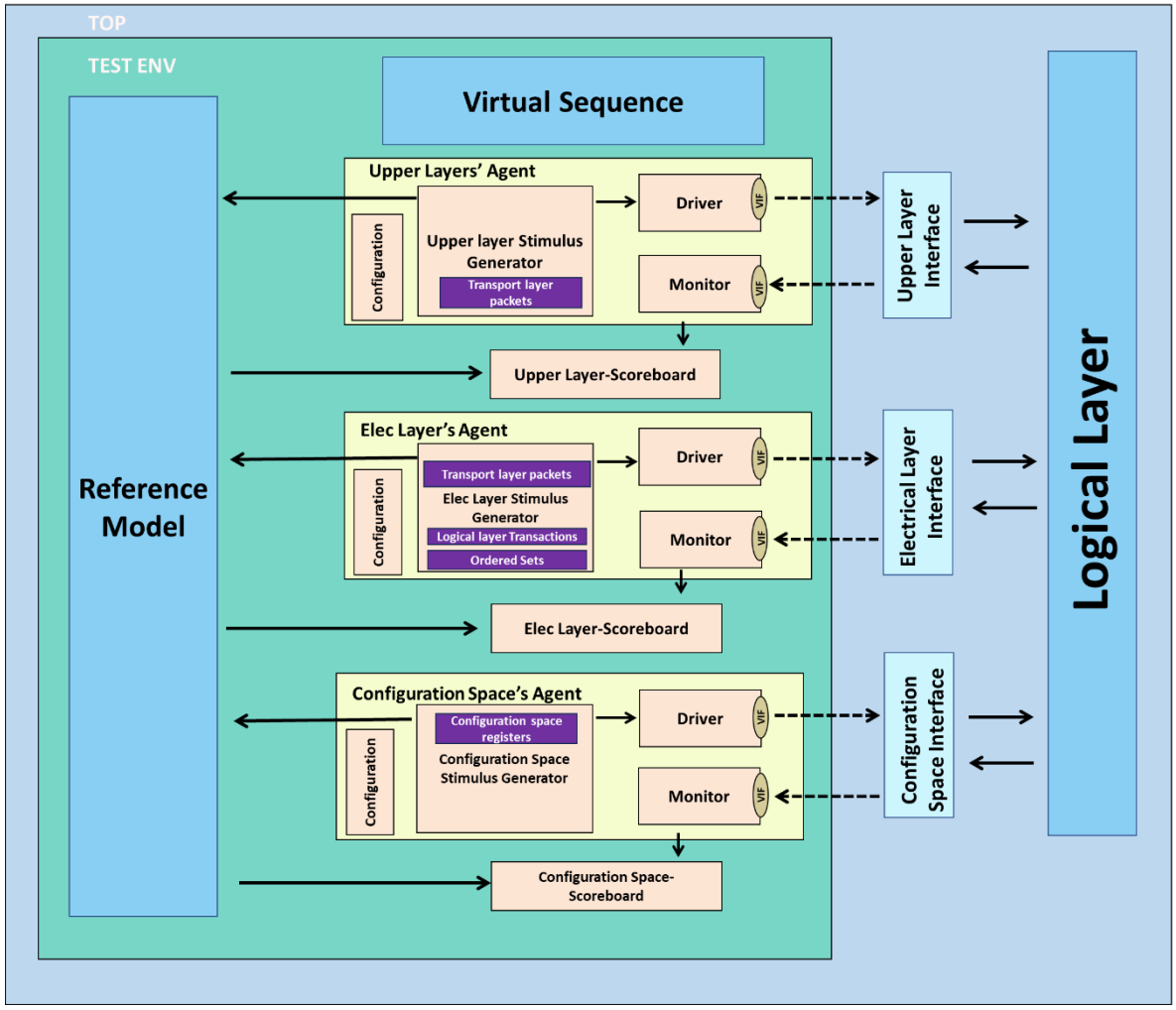
Moreover, Interfaces definition that connects the Environment to the Logical layer will be fully analyzed.

Furthermore, The Design Operational Flow along with the test Scenarios will be represented.

Finally, A traceability matrix will be constructed which is used to make sure that the created test scenarios actually test all the Design features.

Last Page will contain the Complete Task breakdown with the assigned tasks to each team member along with the expected time of completion.

Verification Environment:



Test Environment:

This class contains all the VIP components:

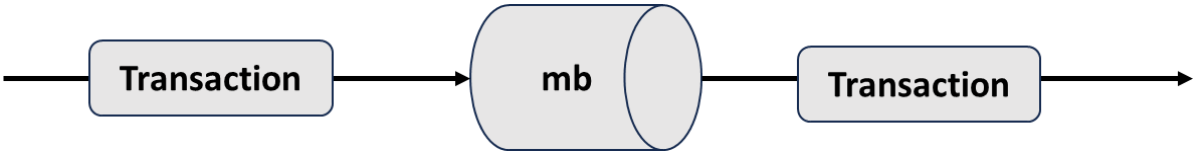
- Reference Model
- Virtual Sequencer
- 3 interfaces:
  - Upper Layer Interface
  - Electrical Layer Interface
  - Configuration Spaces Interface

- 3 Agents for each Interface which includes:
  - Stimulus Generator
  - Driver
  - Monitor

- 3 Scoreboard for the 3 Agents

- Mailboxes for communication
  - S2d_mb: connects the Stimulus Generator to the Driver
  - Scr_mod: Connects the Stimulus Generator to the Reference Model
  - Log_mon: Connects the Monitor to the Scoreboard
  - Mod_scr: Connects the Reference Model to the Scoreboard
  - Sqcr_agnt: Connects the Virtual Sequencer to the Agent

System Verilog Mailboxes:

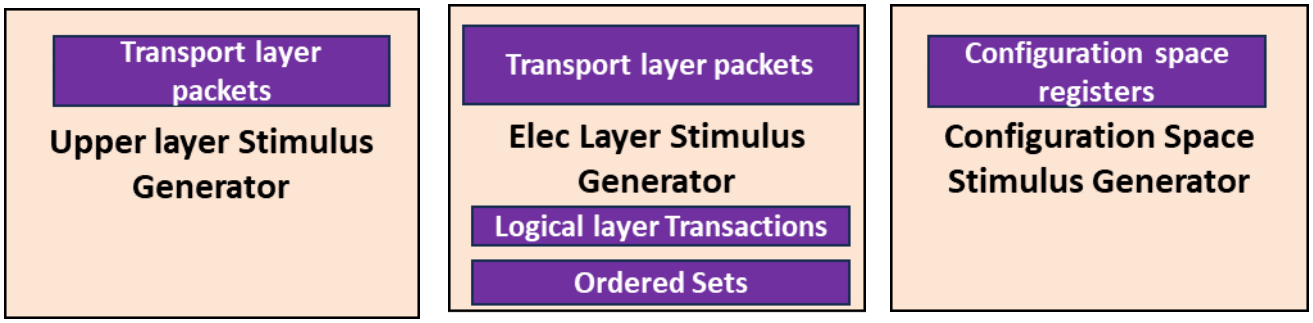


System Verilog Mailboxes enables the communication between the classes using 2 simple functions:

- .put(transaction): used to send transactions to the mailbox.
- .get(transaction): used to receive transactions from the mailbox.

The Mailboxes will be used to connect the operation of all mentioned components together to be able to perform the verification tasks in the correct order seamlessly to test the Logical layer in a fast and efficient way.

Stimulus Generator:

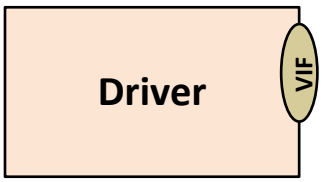


It is the responsibility of the stimulus generator to create different test scenarios to test the required features of the design.

It combines all inputs into a packet/ transaction and sends the data to the driver via a mailbox when the driver is ready to send the next transaction through the Virtual interface.

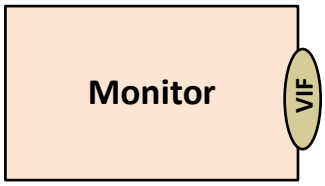
Simulation ends when no more transactions are remaining to be sent by the stimulus generator.

Driver:



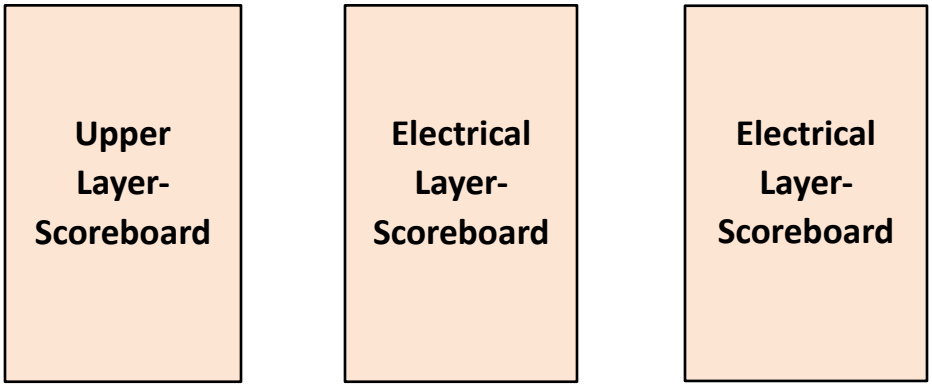
A driver has the responsibility to generate the Pin level signals of the logical layer based on the transaction contents taken received from the Stimulus generator via a mailbox.

Monitor:



The main task for monitor is to capture any activity on the DUT signals (either inputs or outputs). The monitor assigns the pin level signal (from the interface) to the transaction/packet then send the transaction to the scoreboard via a mailbox.

Scoreboard:



This is where the comparison between the DUT outputs and the golden reference is made to verify that the design works as expected.

We will need 3 Scoreboards where each scoreboard will receive a type of transaction based on the interface assigned to it.

Also, Functional Coverage takes place inside the Scoreboard in order to keep track of the progress of testing after each test scenario being applied. Based on the exit coverage criteria for each DUT, the Functional Coverage reports determine whether the design verification is done or not.

Reference Model:

Reference Model

This is a golden reference that mimics the operation of the logical layer using a simpler, not necessarily synthesizable, model. This model takes its inputs from the 3 agents' Stimulus generator.

The reference Model then delivers the Expected output to a Scoreboard via a mailbox connecting them together.

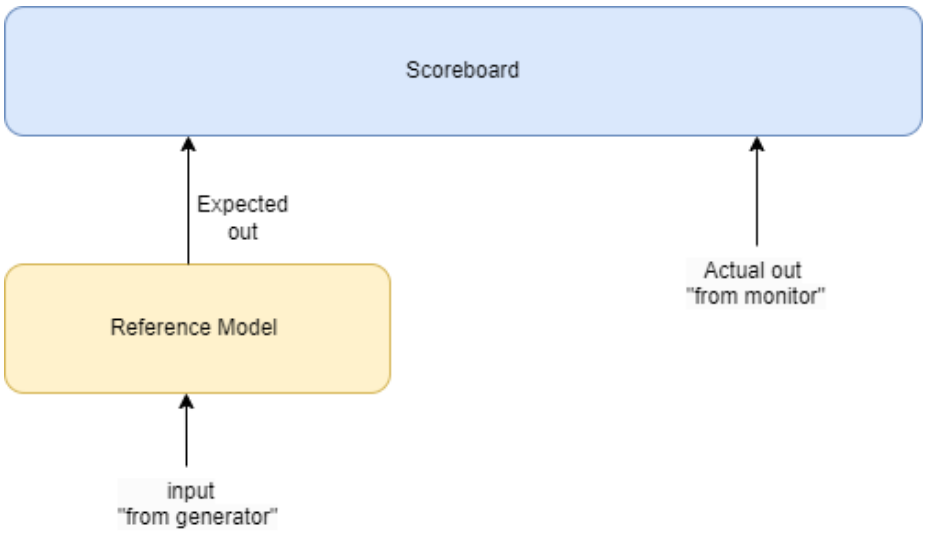
Virtual Sequencer:

Virtual Sequencer

The Virtual Sequencer Synchronizes the operation between all 3 Stimulus generators in order to drive signals in the correct order from and to the Logical Layer.

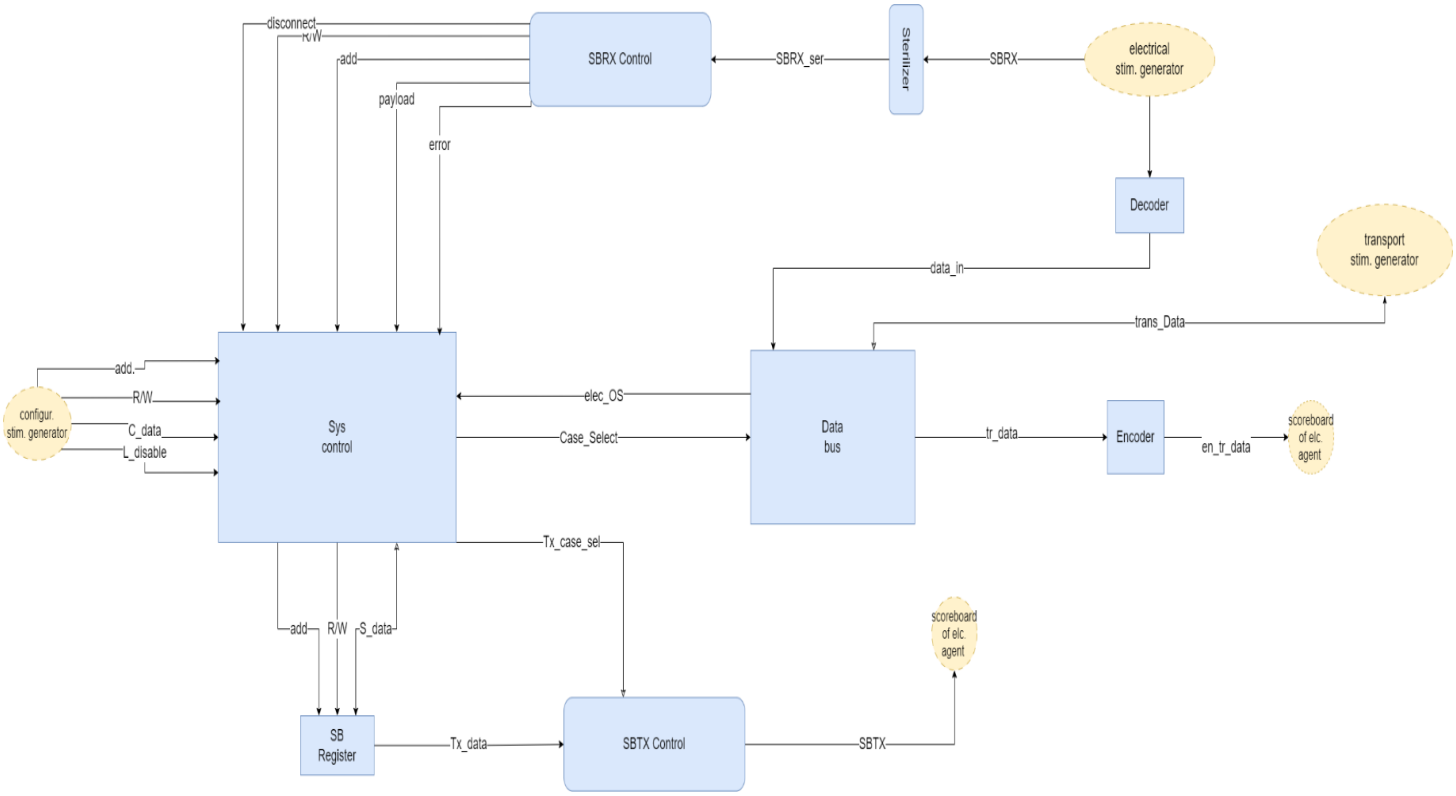
Reference Model:

The primary objective of the reference model is to receive input from the generator and produce the expected output. This output is then compared with the actual output in the scoreboard to determine the presence of any errors.

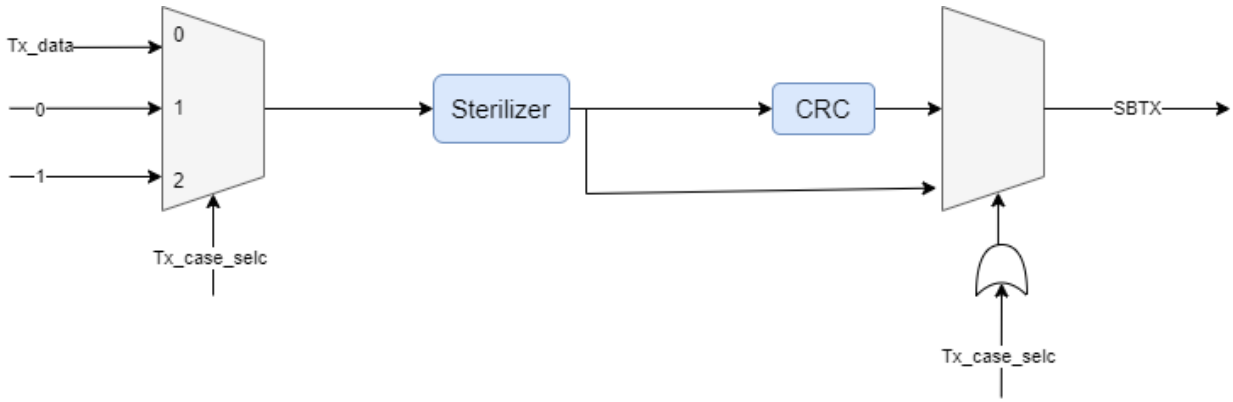


Architecture:

• We want to check the input for each phase by getting the expected output of this phase to push this value into scoreboard.



SBTX Controller:



Step1: read from configuration stimulus generator data and write it on side band memory and check SB_TX output from controller be high.

Step2: wait transaction from electrical stimulus generator for AT transaction and check the correction then out from controller transaction response.

Step3: detect from electrical stimulus generator 16 order sets then out 2 order sets with respect to the type from controller selector and out it on the score board.

Step4: wait data from the transport stimulus generator electrical stimulus generator then forward it on score board to check it with output from DUT.

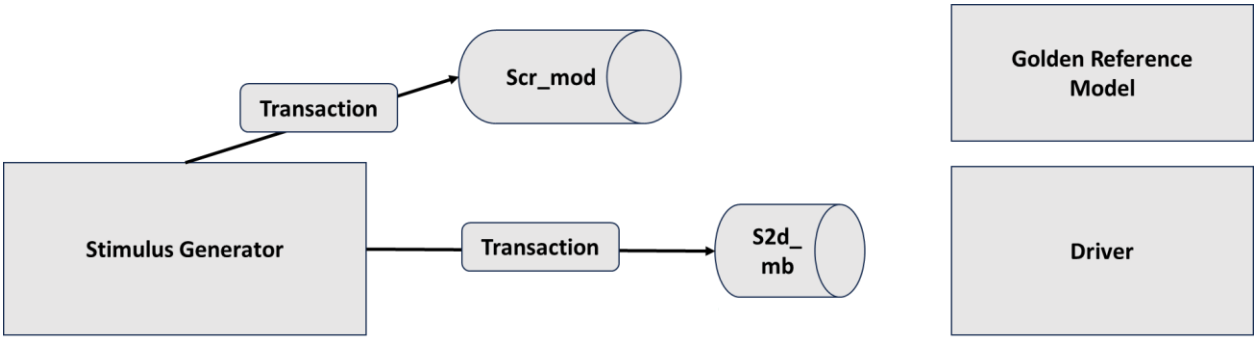
Notes:

- L_disable bin acts as enable for data out on electrical score board throw mailbox.

-we can read or write from configuration stimulus generator data during transfer data form lane adapter.

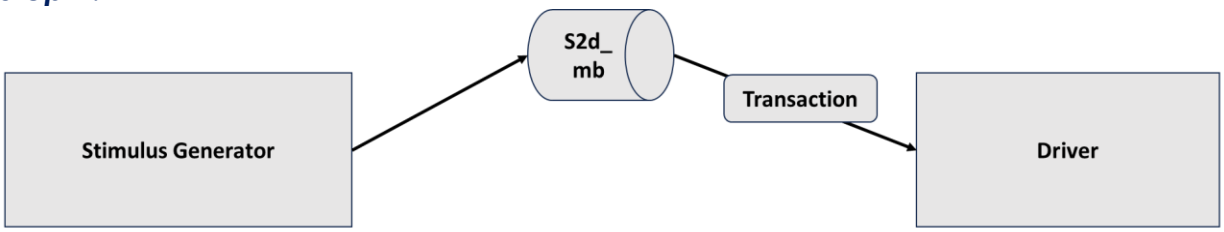
Transaction Flow:

Step 1:



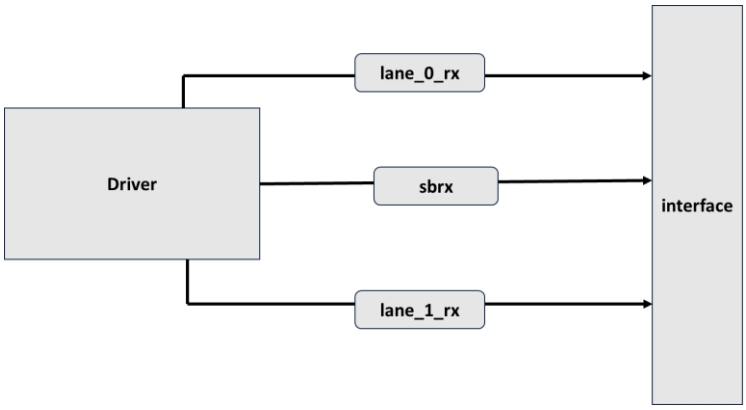
- The stimulus generator sends a scenario transaction to the **driver and Golden Reference's mailbox**.
- The **.put method** is used to push data into the mailboxes.

Step 2:



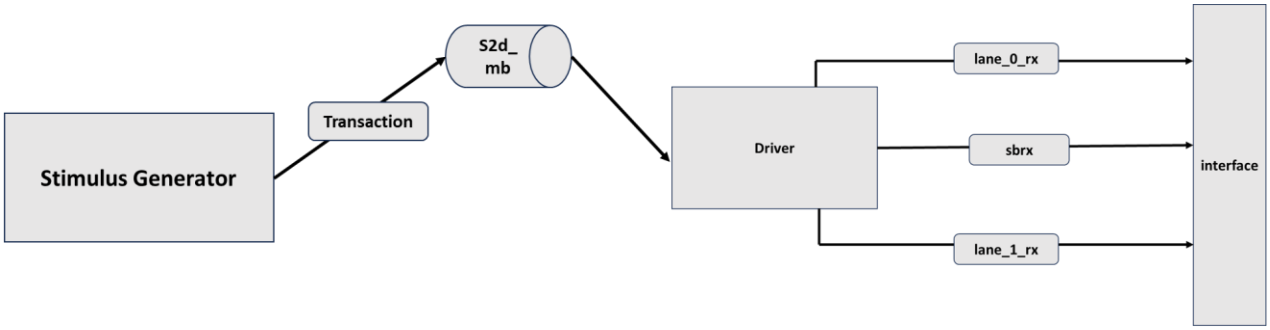
- The driver obtains the stimulus from the mailbox using **.get()** function.
- The generator is **blocked** from sending any other data until the **event (done)** is triggered by the driver.

Step 3:



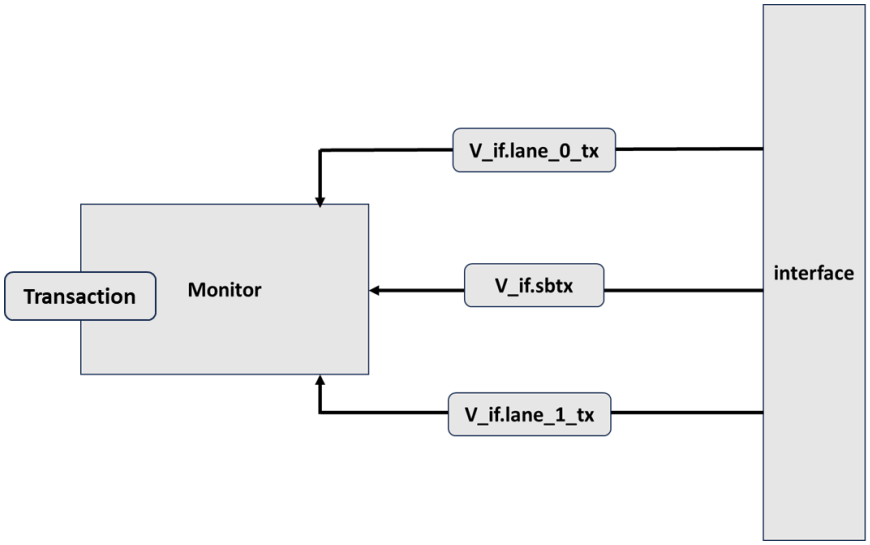
- ❑ According to the obtained transaction, the driver assigns data to the virtual interface every clock cycle.

Step 4:



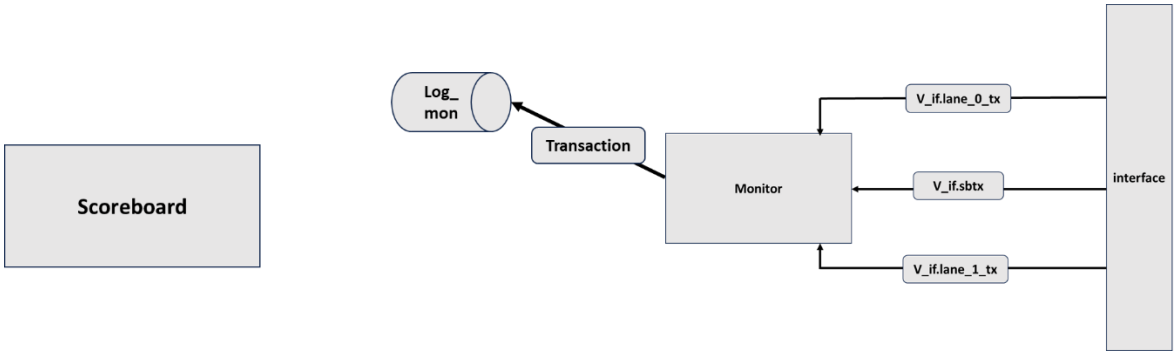
- ❑ Driver triggers the **event (done).**
- ❑ The stimulus generator is now **unblocked** and is free to send the **next** transaction.

Step 5:



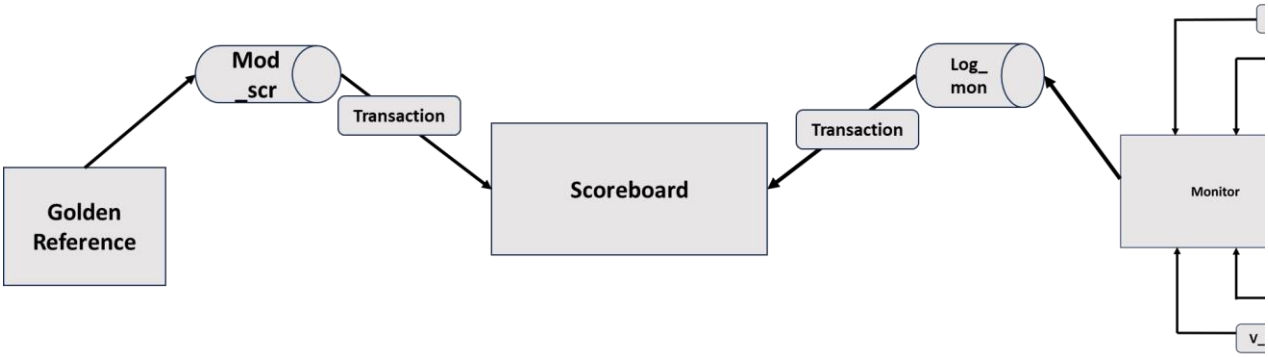
- ❑ Monitor obtains the interface contents (inputs/ outputs).
- ❑ Creates the transaction.

Step 6:



- ❑ Monitor sends data to the scoreboard's mailbox connected via the **.put function**

Step 7:



- ❑ The scoreboard receives both transactions from the golden reference and the monitor.
- ❑ Performs direct comparison between the inputs.
- ❑ Proceeds to check the DUT outputs to verify its functionality.

Design Limitations:

1. No Logical Layer Encoding:
   - ➢ The logical layer should provide encoding mechanisms; however, it won't be part of this implementation and will rely on an external encoding/decoding block for this functionality.

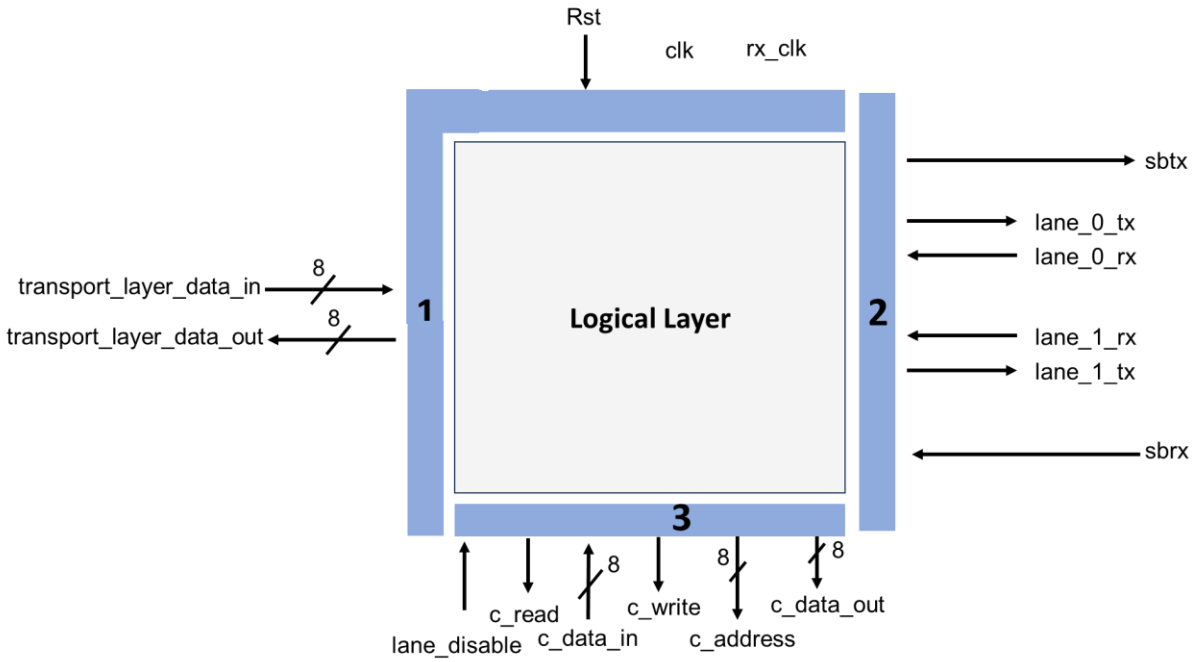
2. No Compliance Testing:
   - ➢ There are no built-in compliance testing features within the logical layer.

3. No Low Power States:
   - ➢ The logical layer does not support low power states, meaning it operates at full power without power-saving features.

4. No Re-timers
   - ➢ The connecting cable is assumed to be always passive (have no re-timers)

SYSTEM Level VERIFICATION:

Interfaces:



For maximum Verification Flexibility, the Interface will be divided into 3 interfaces representing 3 different Entities:

1) Upper layer Protocols
2) Electrical layer (Opposite Router)
3) Configuration Spaces

Interface 1 ports (Upper Layer Interface):

Name	size	description
Operating Clocks		
clk	1	Local clock
gen2_fsm_clk	1	1.212 GHz Clock frequency
gen3_fsm_clk	1	2.424 GHz Clock frequency
gen4_fsm_clk	1	4.95 GHz Clock frequency
Interface Signals		
rst	1	Reset
tansport_layer_data_in	[7:0]	Packet transmitter/ receiver that connects to the transport layer
tansport_layer_data_out	[7:0]	Packet transmitter/ receiver that connects to the transport layer
Internal Signals		
generation_speed	Enum (GEN)	Indicates the Generation speed to determine the operating clock frequency. - Gen2 - Gen3 - Gen4
phase	3	Indicates the current operating phase. - 1 --> phase 1 - 2 --> phase 2 - 3 --> phase 3 - 4 --> phase 4 - 5 --> phase 5

Interface 2 Ports (Electrical Layer Interface):

Name	size	Description
Operating Clocks		
clk	1	Local clock
SB_clock	1	Sideband Clock: 1 MHz
gen2_lane_clk	1	2nd generation lane Clock: 10 GHz
gen3_lane_clk	1	3rd generation lane Clock: 20 GHz
gen4_lane_clk	1	4th generation lane Clock: 40 GHz
Interface Signals		
sbrx	1	Sideband receiver
lane_0_rx	1	Packet/ ordered set receiver
lane_1_rx	1	Packet/ ordered set receiver
sbtx	1	Sideband transmitter
lane_0_tx	1	Packet/ ordered set transmitter
lane_1_tx	1	Packet/ ordered set transmitter
Internal Signals		
generation_speed	Enum (GEN)	Indicates the Generation speed to determine the operating clock frequency. - Gen2 - Gen3 Gen4
phase	3	Indicates the current operating phase. - 1 --> phase 1 - 2 --> phase 2 - 3 --> phase 3 - 4 --> phase 4 - 5 --> phase 5

Interface 3 Ports(Configuration Space Interface):

Name	size	description
Operating Clocks		
clk	1	Local clock
gen4_fsm_clk	1	4^{th} generation lane Clock: 40 GHz
Interface Signals		
lane_disable	1	Disable the FSM Control Unit block
c_data_in	[7:0]	Data read from the configuration spaces
c_read	1	Read flag for the configuration spaces
c_write	1	Write flag for the configuration spaces
c_address	[7:0]	Address to read/write to the configuration space
c_data_out	[7:0]	Data to be written in the configuration spaces

---

ENVIRONMENTS TRANSACTIONS:

Transport Layer Transaction

Bin name	Size(bits)	Description
T_DATA	8	Refer to data between transport layer and electrical layer.
phase	3	
Gen_speed	Enum (GEN)	Indicates the generation speed to be used. - 00’b --> gen2 - 01’b --> gen3 - 10’b --> gen4

Configuration Spaces Transaction

Bin name	Size(bits)	Description
lane_disable	1	Disable the FSM Control Unit block
c_data_in	[7:0]	Data read from the configuration spaces -40'h: In case of USB4 -??'h: In case of Gen4
c_data_out	[7:0]	Data to be written in the configuration spaces
c_read	1	Read flag for the configuration spaces
c_write	1	Write flag for the configuration spaces
c_address	[7:0]	Address to read/write to the configuration space

Electrical Layer transactions:

Bin name	Size(bits)	Description
sbrx	1	SBRX (for sending data to the DUT) Always "1" starting from the training phase except when sending LT_Fall command.
transaction_type	Enum (tr_type)	-000'b: none -001'b: LT_Fall -010'b: AT_cmd -011'b: AT_rsp -100'b: LT_Fall_wrong -101'b: AT_cmd_wrong -110'b: AT_rsp_wrong
read_write	1	Indicates whether the AT operation is read or write. -0'b: read -1'b: write
address	8	Address of the register being read from or written to.
len	7	Number of bytes to read/write. Shall not be greater than 64.
cmd_rsp_data	24	Cmd: Data to be written Rsp: Data read **Note:**

		-Bit 5 in byte 1 in register 12 = 1 in case of GEN3 support. -Bit 2 in byte 2 in register 12 = 1 in case of GEN4 support. - If both bits = 0, therefore GEN2 support. **Length in standard: Not more than 64 bytes (we assumed 3 bytes only for simplicity).
o_sets	Enum (OS_type)	Refer to type of Ordered set. -1000'b: refer to SLOS 1. -1001'b: refer to SLOS 2. -1010'b: refer to TS1 for (Gen 2,3). -1011'b: refer to TS2 for (Gen 2,3). -1100'b: refer to TS1 for gen 4. -1101'b: refer to TS2 for gen 4. -1110'b: refer to TS3 for gen 4. -1111'b: refer to TS4 for gen 4. -MSB refers to whether there are O_sets or no. (* 0xxx'b: refer to no O_sets. * 1xxx'b: refer to there are O_sets.)
gen_speed	Enum (GEN)	Indicates the generation. -00'b --> gen2 -01'b --> gen3 -10'b --> gen4
electrical_to_transport	8	Data sent after training between electrical layer and transport layer.
phase	3	-001'b -->we are in phase 1 -010'b --> we are in phase 2 -011'b --> we are in phase 3 -100'b --> we are in phase 4 -101'b --> we are in phase 5 default --> we aren't in any phase "sbtx = 0"
tr_os	2	Indicates whether the driver will send transaction or ordered set.
sbtx	1	SBTX (for receiving data from the DUT)
transport_to_electrical	8	Data sent after training between electrical layer and transport layer.
crc_received	16	Crc field received in the AT cmd and AT rsp
order	4	Indicates the order of the TS ordered set
lane	Enum (LANE)	Indicates lane that the data is associated with. -00 --> None -01 --> lane_0 -10 --> lane_1 -11 --> both

USB4 Logical Layer Flow

No.	Flow	Description
1	Reset	SBTX should be “0” once the reset is asserted.
Phase 1: initial condition acquisition		
2	Detection of USB4 capability	The logical layer should send signal via c_read/write, c_address to acquire the required data about the connection
3	Detection of USB4 generation	If USB4 connection is detected, the layer then proceeds with the same manner to obtain information about the USB4 version (gen2 or gen3 or gen4?)
Phase 2: Router Detection		
4	Router Detection	• Once SBRX is detected high, the SBTX should turn high as well. • If the USB4 Port receives a Transaction while it is still in Phase 2, it may ignore or drop the Transaction.
Phase 3: Exchange of lane parameters		
5	Lane parameter request	The router is required to send AT transaction via the SBTX which have the address of the register 12 to be read from the sideband of the other router
6	Saving requested lane parameters	The router should detect the parameters that were requested via an AT Response on its SBRX and save their values internally.
7	Response to an AT command	The router should then respond to the AT command received on its SBRX by sending the reg 12 SB register via its SBTX
Phase 4: Link Training		
	Gen4	
8	TS1 state behavior	• Layer should send at least 16 TS1s with indication field = 2h on lane_0_tx and lane_1_tx • Remain in this state until it receives (on lane_x_rx): ➢ Received TS1 with indication field = 2h ➢ Or Received TS2

9	TS2 state behavior	• Layer should send at least 16 TS2s with indication field = 4h on lane_0_tx and lane_1_tx • Remain in this state until it receives (on lane_x_rx): ➢ Received TS2 with indication field = 4h ➢ Or Received TS3
10	TS3 state behavior	• Layer should send at least 16 TS3s with indication field = 6h on lane_0_tx and lane_1_tx • Remain in this state until it receives (on lane_x_rx): ➢ Received TS3 with indication field = 6h ➢ Or Received TS4
11	TS4 state behavior	Layer should send TS4 in increasing order of counter field until it reaches value = Fh
	Gen2 and Gen3	
12	SLOS1 state behavior	• Layer should send 2 SLOS1 • Remain in this state until it receives 2 SLOS1 (on lane_x_rx).
13	SLOS2 state behavior	• Layer should send 2 SLOS2 • Remain in this state until it receives 2 SLOS2 (on lane_x_rx).
14	TS1 state behavior	• Layer should send 32 TS1 for gen 2 and 16 TS1 for gen 3 • Remain in this state until it receives 2 TS1 (on lane_x_rx).
15	TS2 state behavior	• Layer should send 16 TS2 for gen 2 and 8 TS2 for gen 3 • Remain in this state until it receives 2 TS2 (on lane_x_rx).
16	Training Disable	If lane_disable bit inside the configuration space changed to 1, the layer should be disabled and no ordered sets should be sent or processed and SBTX should be zeros.

17	Timeout	If the layer remains for tTrainingAbort time in the same state, the layer should repeat the process of initialization again from phase 3.
Phase 5: Transport layer transmission		
18	Packet transmission from its own router transport layer to another router	Any payload received via the transport_layer_data port should be forwarded to the lane_x_tx in parallel
19	Packet transmission from another router to its own router transport layer	Any payload received via the lane_x_rx should be forwarded to the transport_layer_data port.
20	CL0 Disable	If lane_disable bit inside the configuration space changed to 1, the layer should be disabled and no packets should be forwarded. And SBTX should be zeros.
Disconnections		
21	LT_fall transaction behavior	• Should begin phase 1 again. • Load the configuration spaces and SB registers with the default values as required by the specifications • And SBTX should be zeros
22	SBRX low behavior	• Should begin phase 1 again. • SBTX should be zeros
Outputs default behavior		
23	SBTX	• Should be = 0 when the router just turned on or no connection is still established. • Should be = 1 when there is a link connection with other routers but no transactions to be sent.
Dropping Transactions		

24	Dropping LT Transactions	The LT transactions will be dropped if the received CLSE symbol (byte 2) is not the bitwise complement of the LSE symbol (byte 1).
25	Dropping AT Transactions	• The AT transactions will be dropped if the received CRC transaction is invalid. • The AT transactions will be dropped if the transaction has no data and no CRC field.
Managing wrong inputs		
26	Unsupported Generation	If the connection type stored in the configuration unit is not supported (not gen2, gen3 or gen4), the logical layer shall not proceed the CLd substates.
27	Ignoring unexpected ordered sets	• If the negotiated speed is gen2 or gen3, and the received ordered sets are incorrect or related to gen4. • If the negotiated speed is gen 4, and the received ordered sets are incorrect or related to gen2 or gen3.
28	Ignoring incorrect transactions	If the received transaction on the SBRX is incorrect, the transactions' fsm will act according to its current state.
29	Ignoring write commands in read-only SB register locations	If the link partner sends AT command to write in a RO SB register location, the logical layer shall ignore this write command.

Test Scenarios

Testing Scenario	**Tested Feature**	**Notes**	**Time Estimate**
Basic Logical Layer flow (1)	Different Generations: • Gen2 • Gen3 • Gen4	No other generations are possible	4 days From **20/2/2024** to **24/2/2024**
Basic Logical Layer Flow (2)	C_DISABLE during Operation: • Training (phase 4) • CL0 (phase 5)	Should be applied during phase 4 or 5 only	2 days From **24/2/2024** to **26/2/2024**
Basic Logical Layer Flow (3)	Disconnection due to: • LT_FALL transaction • SBRX low for tDisconnectRx time	• LT_Fall should be applied during the CL0 state. • SBRX may go low during any phase	4 days From **20/2/2024** to **24/2/2024**
Fault injected scenarios			
Mixing Ordered Sets	Restarting Training	Gen2 or gen3 ordered sets: • SLOS1 • TS1.gen2/3 • SLOS2 • TS2.gen2/3 Gen4 ordered sets: • TS1.gen4 • TS3 • TS2.gen4 • TS4	2 days From **24/2/2024** to **26/2/2024**
Wrong AT response	Retransmission of AT commands for wrong responses	Wrong response can be due to • Wrong CRC • Non-existing Gen	2 days From **26/2/2024** to **28/2/2024**
Wrong AT Read command	Ignoring non-existing SB addresses read		2 days

			From **26/2/2024** to **28/2/2024**
Wrong AT Write command	Refusal of writing in Read Only SB registers		2 days From **28/2/2024** to **30/2/2024**
Wrong LT_Fall command	Ignoring Wrong disconnection		2 days From **28/2/2024** to **30/2/2024**

Traceability Matrix:

Test Scenario	**Tested Features**						
	Gen2 operation	Gen3 operation	Gen4 operation	Configuration Space Disable	LT_fall Disconn	SBRX low Disconn	Fault handling
	Basic Testing						
Basic Logical Layer flow (1)							
Basic Logical Layer Flow (2)							
Basic Logical Layer Flow (3)							
	Advanced Testing						
Mixing ordered sets							
Wrong AT response							
Wrong AT Read command							
Wrong AT Write command							
Wrong LT_Fall							

Task Breakdown:

Task		Team Member Responsible	Expected Completion TIME
Build Verification Environment (Implementation)	Electrical layer Agent	Ali Ahmed & Karim Samy	9/2/2024
	Configuration space Agent	Karim Samy	
	Upper layer Agent	Ali Ahmed	
Build Reference Model		Mohamed Salah & Walid Salah	9/2/2024
Build Verification Environment (Integration)		Ali Ahmed & Karim Samy	10/2/2024
Build Verification Environment (Integration with DUT)		All	14/2/2024
Basic Logical Layer Flow (1) Scenario	Development	Ali Ahmed & Karim Samy	24/2/2024
	Running		
	Debugging		
Basic Logical Layer Flow (2) Scenario	Development	Ali Ahmed & Karim Samy	26/2/2024
	Running		
	Debugging		
Basic Logical Layer Flow (3) Scenario	Development	Mohamed Salah & Walid Salah	24/2/2024
	Running		
	Debugging		
Mixing Ordered Sets	Development	Mohamed Salah & Walid Salah	26/2/2024
	Running		
	Debugging		
Wrong AT response	Development	Ali Ahmed & Karim Samy	28/2/2024
	Running		
	Debugging		
Wrong AT Read command	Development	Mohamed Salah & Walid Salah	28/2/2024
	Running		
	Debugging		
Wrong AT Write command	Development	Ali Ahmed & Karim Samy	30/2/2024
	Running		
	Debugging		
Wrong LT_Fall	Development	Mohamed Salah & Walid Salah	30/2/2024
	Running		
	Debugging		

AI		
Build Verification Environment (Implementation)	To be decided	To be decided
Build Reference Model	To be decided	To be decided
Build Verification Environment (Integration with DUT)	To be decide	To be decided
Teset Scenarios	To be decided	To be decided

Duration of each task:

Task		Time	
Build Verification Environment (Implementation)			
Configuration Space's Agent		4 hrs	
Configuration Space's Sequence		2 hrs	
Upper Layer's Agent		4 hrs	
Upper Layer's Sequence		1 hr	
Electrical Layer's Agent	Driver	26 hrs	
	Monitor	37 hrs	
Electrical Layer's Sequence		15 hrs	
All the Scoreboards		1.5 hr	
All Transactions and Interfaces		2 hrs	
Virtual Sequence		3 hrs	
Build Reference Model	Modelling diagram	15 hrs	Avg time per day (4.13 hrs/day)
	Codding 1 (CRC, Serializer, SB registers,)	20 hrs	
	Build phases	38 hrs	
	Debugging and connect with environment	20 hrs	

Build Verification Environment (Integration)		3 hrs
Build Verification Environment (Integration with DUT)		
Basic Logical Layer Flow (1) Scenario	Development	
	Running	
	Debugging	
Basic Logical Layer Flow (2) Scenario	Development	
	Running	
	Debugging	
Basic Logical Layer Flow (3) Scenario	Development	
	Running	
	Debugging	
Mixing Ordered Sets	Development	
	Running	
	Debugging	
Wrong AT response	Development	
	Running	
	Debugging	
Wrong AT Read command	Development	
	Running	
	Debugging	
Wrong AT Write command	Development	
	Running	
	Debugging	
Wrong LT_Fall	Development	
	Running	
	Debugging	