Flash Memory Summit

Accelerating Verification of CXL based Designs

Prashant Dixit

Siemens EDA

Agenda

Functional Verification Solution

Verification IP

VIP Use Models

Host/Device BFM

Verification Plan and Stimuli

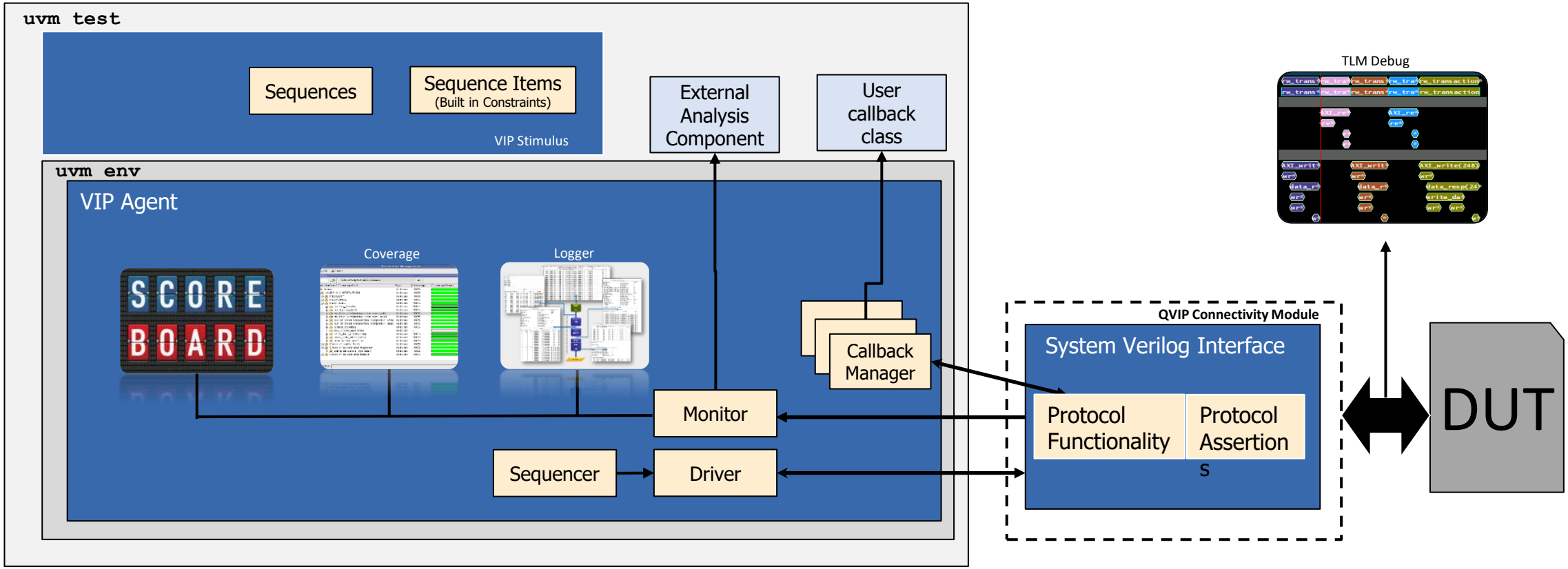
Protocol Director

Debugging Methods

Protocol Checks and Coverage

Summary

Functional Verification Solution



Verification IP

- Host, Device and Monitor BFM
- Verification Plan with Stimulus
  - Compliance Test Suite
  - SIG compliance
- Protocol Director
- Debug : Text and GUI Based Loggers
- Protocol Checks
- Scoreboard
- Coverage

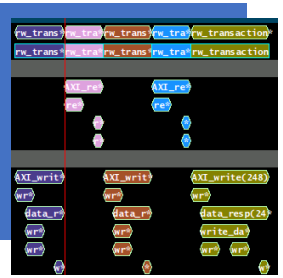
Verification IP

Architected for ease of use

- Standards based SV UVM support
- Configuration & TB generation GUI

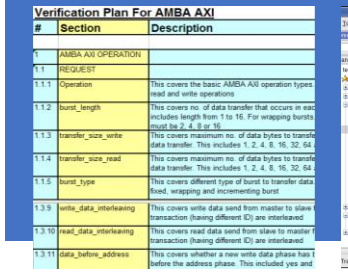
Intuitive Debug

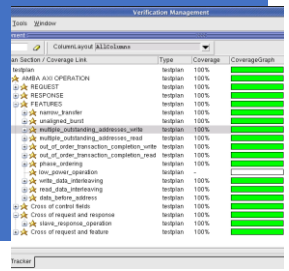
- Transaction viewing
- Scoreboard comparison
- Tracker files

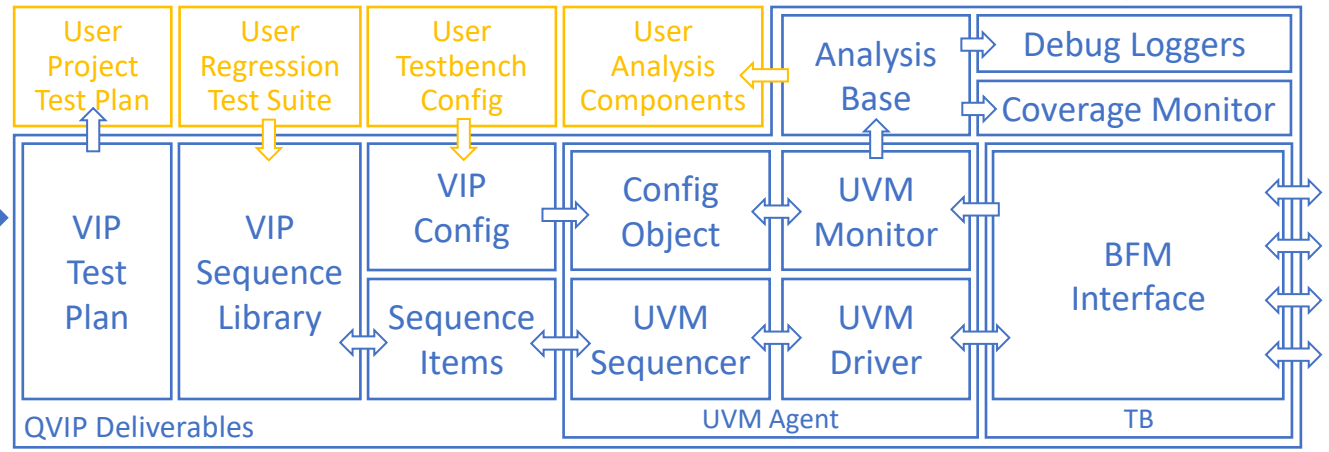


Comprehensive validation

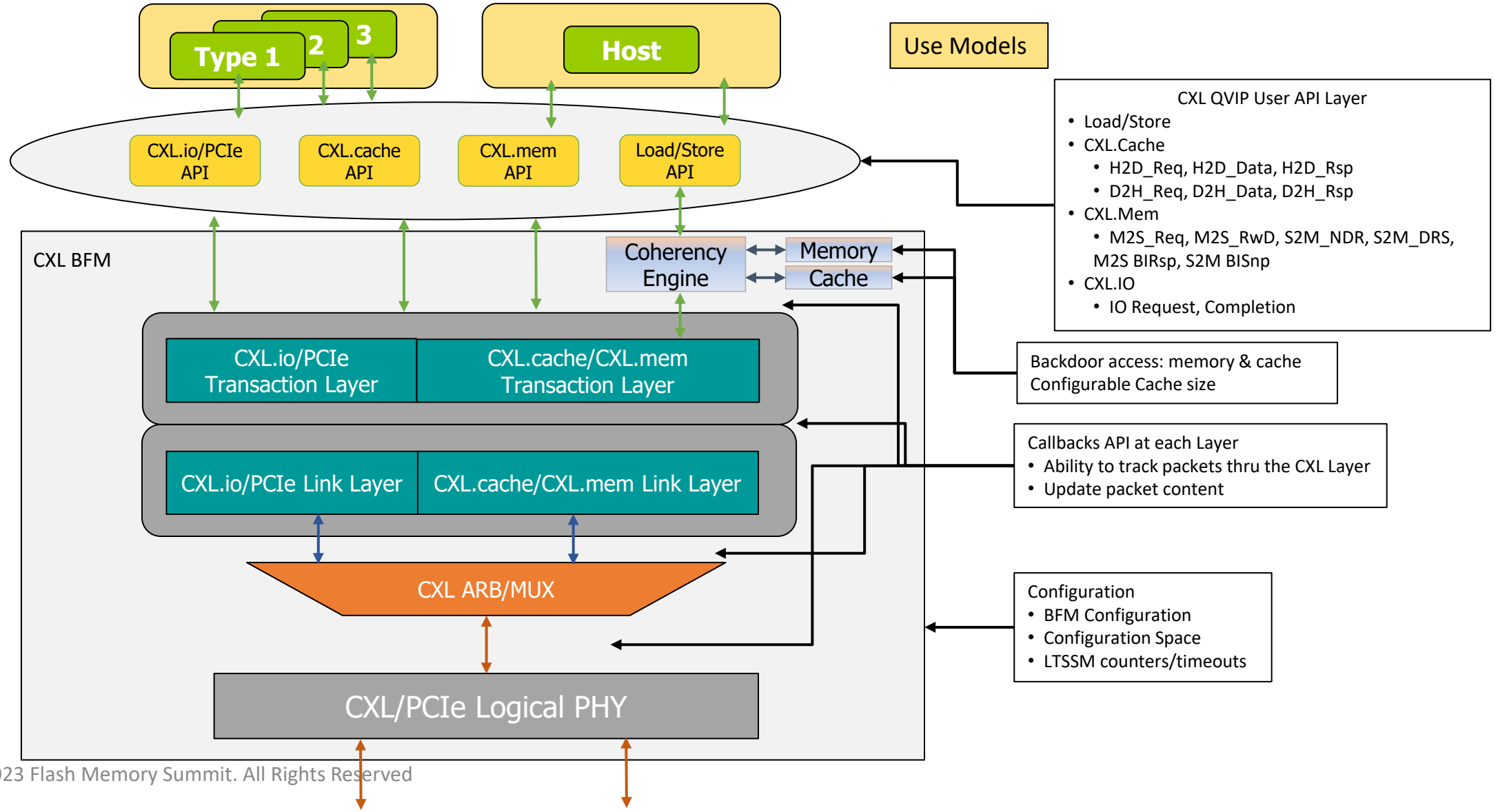
- Test suites
- Protocol checks
- Functional coverage







QVIP Block Diagram



QVIP Use Models

- Intelligent modeling allows the QVIP to mimic either host or device (Type 1/2/3) behavior based on DUT type at the other end of the CXL bus

- Monitor BFM can be plugged into existing env as passive component
  - monitors bus traffic and provides various verification capabilities like protocol checking, coverage and logging

Host/Device BFM

- Load/Store APIs
  - Abstraction for providing high-level API
  - breaks down into lower-level traffic of D2H requests depending upon the cache line states
  - Follows coherency protocol and biasing rules
- Automatic responder
  - DCOH engine/CA responds to the CXL.cache/mem requests automatically
  - Provides appropriate responses based on protocol
- Built-in Cache Model and Snoop Filter
  - Highly configurable cache model with options to set line replacement algorithms, sets, ways, line-width etc.
  - Automated checks at the cache level to verify protocol compliance

Abstractions to run stimuli

- UVM-compliant
- Abstractions
  - Load/Store APIs
  - Sequences
    - Exhaustive set of ready to use sequences
  - Sequence Items
    - For Cache/mem req, data, response
    - For IO request/completion
    - Complete control of every field, can be randomized also.

Verification Plan

CXL CTS - Advanced	TL	B-2 Table 313	Type 3 Memory Request	Host illegaly sends Metafield as Metastate f	mem_rd_data_illegal_metafield	icvip_cxl_mem_rd_data_illegal_metafield_sequence
CXL CTS - Advanced	TL	B-4 Table 315	Type 3 Memory RwD	The Host wants to update memory. Host se	mem_wr_with_metavalue	icvip_cxl_mem_wr_with_metavalue_sequence
CXL CTS - Advanced	TL	B-4 Table 315	Type 3 Memory RwD	The Host wants to update memory. Host se	mem_wr_ptl_with_metavalue	icvip_cxl_mem_wr_ptl_with_metavalue_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to update memory and kee	mem_wr_with_exclusive_cacheli	icvip_cxl_mem_wr_with_exclusive_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to update memory and keep	mem_wr_with_shared_cacheline	icvip_cxl_mem_wr_with_shared_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The host wants to update memory and will	mem_wr_with_invalid_cacheline	icvip_cxl_mem_wr_with_invalid_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to write the line back to me	mem_wr_with_line_write_back	icvip_cxl_mem_wr_with_line_write_back_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to update memory and kee	mem_wr_ptl_with_exclusive_ca	icvip_cxl_mem_wr_ptl_with_exclusive_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to update memory and keep	mem_wr_ptl_with_shared_cache	icvip_cxl_mem_wr_ptl_with_shared_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The host wants to update memory and will	mem_wr_ptl_with_invalid_cache	icvip_cxl_mem_wr_ptl_with_invalid_cacheline_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host wants to write the line back to me	mem_wr_ptl_with_line_write_ba	icvip_cxl_mem_wr_ptl_with_line_write_back_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host Send illegal Snoop Type for S-state	mem_rwd_illegal_snp_type_with	icvip_cxl_mem_rwd_illegal_snp_type_with_meta_shared_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host Send illegal Snoop Type as Snoop I	mem_rwd_illegal_snp_data_with	icvip_cxl_mem_rwd_illegal_snp_data_with_meta_invalid_sequence
CXL CTS - Advanced	TL	B-3 Table 314	Type 2 Memory RwD	The Host Send illegal Snoop Type as Snoop (	mem_rwd_illegal_snp_current_w	icvip_cxl_mem_rwd_illegal_snp_current_with_meta_invalid_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants an exclusive copy of the line	mem_rd_with_exclusive_cacheli	icvip_cxl_mem_rd_with_exclusive_cacheline_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host requesting a shared copy of the lin	mem_rd_with_shared_cacheline	icvip_cxl_mem_rd_with_shared_cacheline_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host requesting a non-cacheable but cu	mem_rd_with_current_cacheline	icvip_cxl_mem_rd_with_current_cacheline_and_flush_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host requesting a non-cacheable but cu	mem_rd_with_current_cacheline	icvip_cxl_mem_rd_with_current_cacheline_no_flush_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants to read line without changir	mem_rd_with_invalid_cacheline	icvip_cxl_mem_rd_with_invalid_cacheline_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants a current value of the line w	mem_rd_with_current_cacheline	icvip_cxl_mem_rd_with_current_cacheline_no_state_change_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants a cacheable copy in either e	mem_rd_data_with_shared_or_	icvip_cxl_mem_rd_data_with_shared_or_exclusive_cacheline_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants ownership of the line witho	mem_inv_ownership_with_no_d	icvip_cxl_mem_inv_ownership_with_no_data_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to degrade to S i	mem_inv_shared_cacheline_with	icvip_cxl_mem_inv_shared_cacheline_with_no_data_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to invalidate the	mem_inv_invalid_cacheline_no_	icvip_cxl_mem_inv_invalid_cacheline_no_metastate_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to invalidate the	mem_inv_invalid_cacheline_with	icvip_cxl_mem_inv_invalid_cacheline_with_metastate_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants ownership of the line witho	mem_inv_nt_ownership_with_n	icvip_cxl_mem_inv_nt_ownership_with_no_data_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to degrade to S i	mem_inv_nt_shared_cacheline_	icvip_cxl_mem_inv_nt_shared_cacheline_with_no_data_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to invalidate the	mem_inv_nt_invalid_cacheline_	icvip_cxl_mem_inv_nt_invalid_cacheline_no_metastate_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	The Host wants the device to invalidate the	mem_inv_nt_invalid_cacheline_	icvip_cxl_mem_inv_nt_invalid_cacheline_with_metastate_sequence
CXL CTS - Advanced	TL	B-1 Table 311	Type 2 Memory Request	Host illegaly sends Snoop_type(SnpInv, SnpC	mem_rd_data_illegal_snp_type	icvip_cxl_mem_rd_data_illegal_snp_type_sequence
CXL CTS - Advanced	TL	B-4 Table 315 and	Type 3 And Type2 Memory R	The Host Send illegal Metafield	mem_rwd_illegal_metafield	icvip_cxl_mem_rwd_illegal_metafield_sequence
CXL CTS - Advanced	TL	B-4 Table 315 and	Type 3 And Type2 Memory R	The Host Send illegal Snoop Type for A-state	mem_rwd_illegal_snp_type	icvip_cxl_mem_rwd_illegal_snp_type_sequence
CXL CTS - Advanced	TL	8.1.3.8.4	DVSEC CXL Range registers	The Host initiates Memory read command c	mem_rd_non_hdm	icvip_cxl_mem_rd_non_hdm_sequence

Stimuli

- Compliance Test Suite (~1000)
  - Exhaustive and ready to use test suite
  - For CXL.io, CXL.cache and CXL.mem
  - Covers all layers : TL, LL, Arbiter and Flex bus
  - Power Management
  - CXL.cache/mem request and response combinations for HDM and main memory
  - Error injection scenarios for various errors with poison and viral
  - Cachebility restrictions/Buried cache scenarios
  - Resets – CXL, FLR, Hot, Cold
  - Security – IDE, TDISP, SPDM/CMA, DOE, KM
  - GUI based tool (Protocol Director) to enable feature/layer wise selection and run stimuli

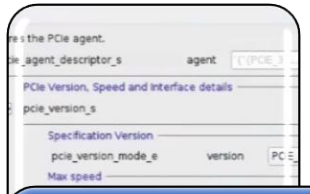
Stimuli

- CXL SIG compliance suite (~200)
  - As per CXL specs
  - Various layers, arbiter, Power management
  - Configuration Registers, RAS
  - Security
  - Memory Mapped registers
  - Reset and Initialization

Workflow

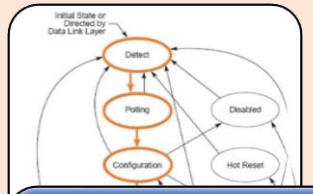
DUT – TB Connections

- Connect DUT pins to VIP signals
- Connect Clock & Reset
- Configure parameters



VIP Configuration

- VIP configured according to design's characteristics

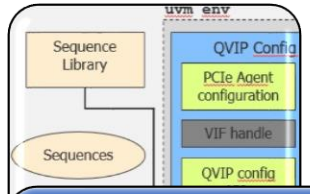


Link Training

- VIP runs link training based on configuration
- Link trained to desired speed

CXL Enumeration

- Run bus enumeration sequence for Device discovery and setting Device capabilities

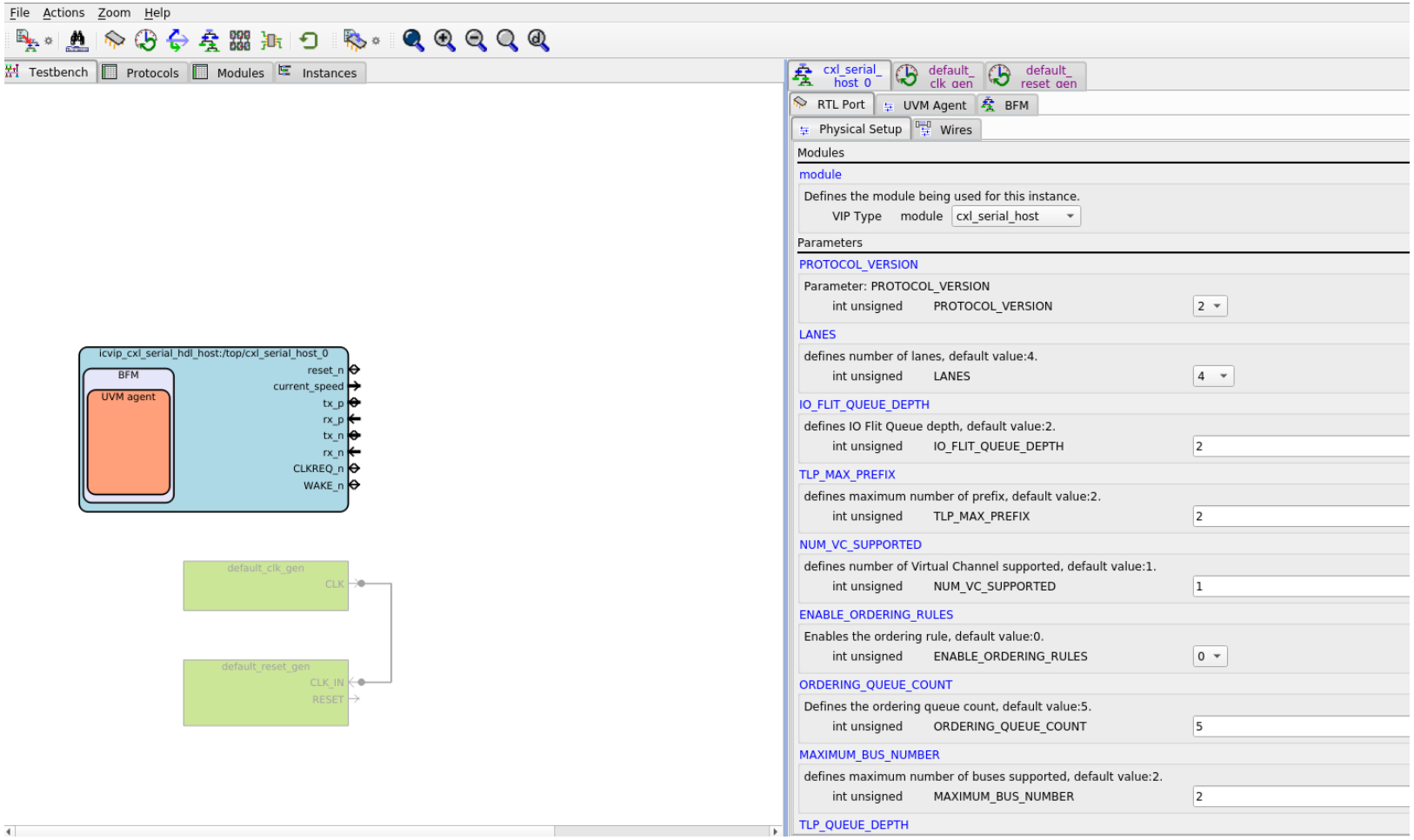


Application Verification

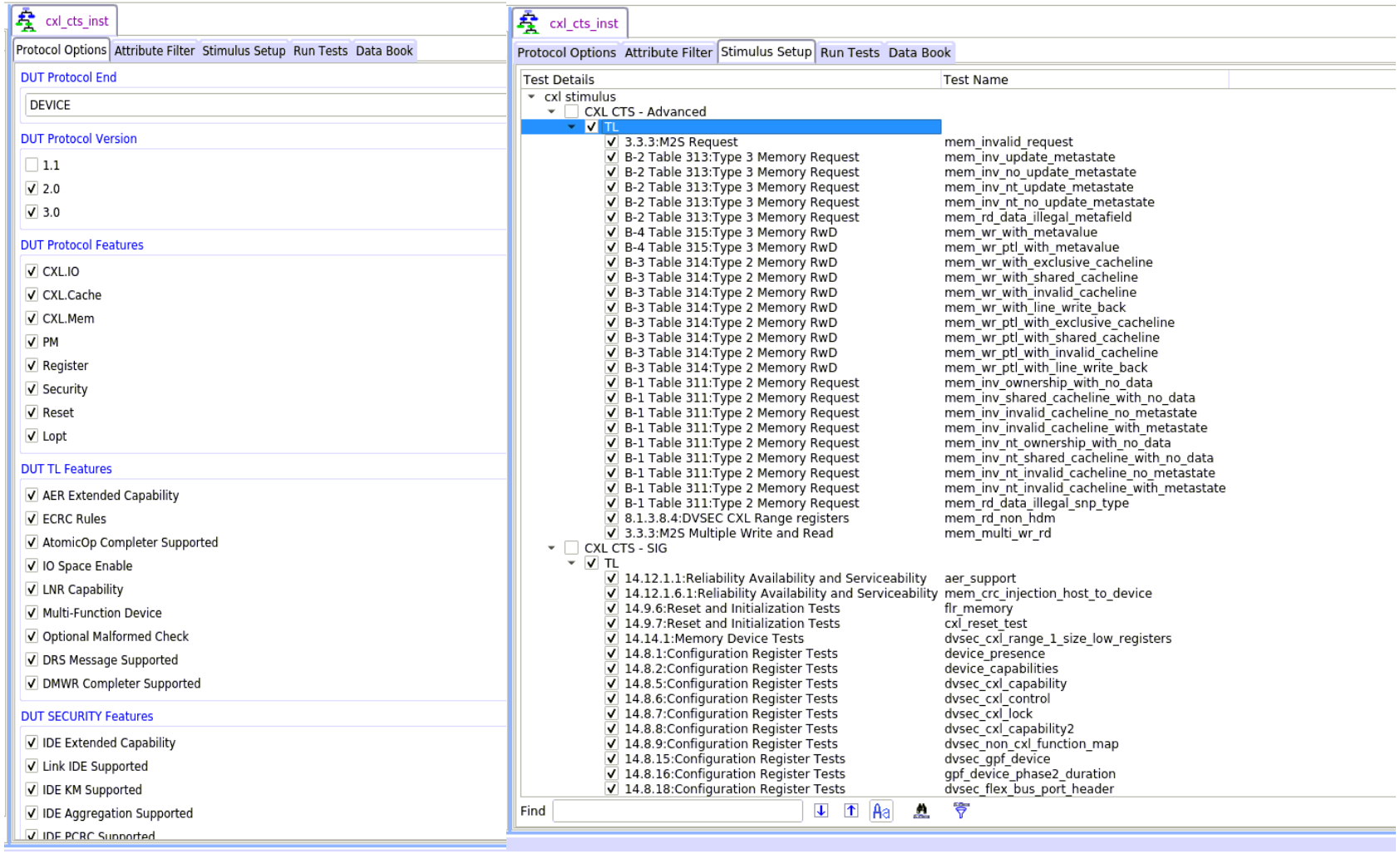
- Users can built-in ready to use stimulus or write device specific stimulus

Getting Started possible in hours

Automatic TB generation (Configurator)



Protocol Director



Protocol Checks

- Exhaustive Set of checks (~1200)
  - CXL.io, CXL.cache and CXL.mem
  - TL, LL, Arbiter, PL and PM
  - Config Space, Memory Mapped registers
  - Coherency checks

ERROR @ 24329536 ps: **ICVIP_CXL_INVALID_MAX_M2S_BIRSP_COUNT_128B_GROUP** - ICVIP_CXL[DEVICE]:

ICVIP_CXL[DEVICE]: Maximum message count for BIResponse message exceeded in rolling 128B group

ICVIP_CXL[DEVICE]: - Rolling group: ICVIP_CXL_ROLLING_GROUP_DA

ICVIP_CXL[DEVICE]: - Message count within 128B Group: 11

ICVIP_CXL[DEVICE]: The maximum message rules are applicable on a rolling 128B group in which the groups are A=Slot 0 to 3, B=Slot 4 to 7, C=Slot 8 to 11, D=Slot 12 to 14. Refer Table 4-17 : For M2S BIRsp - Maximum message count per 128B group is 3. CXL Version 3.0(4.3.4)

ERROR @ 16848875 ps: **ICVIP_CXL_H2D_DATA_POISONED** - ICVIP_CXL[DEVICE]:

ICVIP_CXL[DEVICE]: H2D data is detected with poison bit = 1'b1

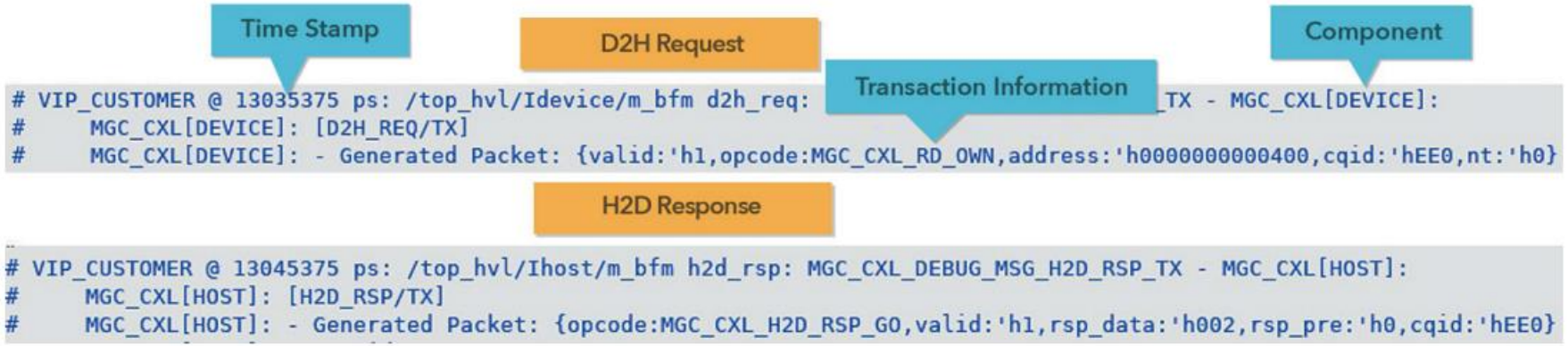
```
ICVIP_CXL_IDE_MAC_NOT_RX_WHEN_EXPECTED, \
ICVIP_CXL_IDE_TMAC_RX_WHEN_NOT_EXPECTED, \
ICVIP_CXL_IDE_KEY_CHANGE_PROTOCOL_FLIT_EARLY_RX, \
ICVIP_CXL_IDE_TX_KEY_REFRESH_TIME_LESS_THAN_RX_MIN, \
ICVIP_CXL_IDE_PROTOCOL_FLIT_RX_EARLY_AFTER_TMAC, \
ICVIP_CXL_IDE_MAC_RX_LINK_INSECURE_STATE, \
ICVIP_CXL_IDE_TMAC_RX_LINK_INSECURE_STATE, \
ICVIP_CXL_IDE_MAC_RX_WHEN_NOT_EXPECTED, \
ICVIP_CXL_IDE_CAPABLE_NOT_SET, \
ICVIP_CXL_IDE_MAC_CHECK_FAILURE, \
ICVIP_CXL_IDE_SKID_MODE_SET_WHEN_NOT_SUPP, \
ICVIP_CXL_CACHEMEM_IDE_KM_QUERY_INVALID_LENGTH, \
ICVIP_CXL_CACHEMEM_IDE_KM_KEY_PROG_INVALID_LENGTH, \
ICVIP_CXL_INVALID_CAPABILITY_ID, \
ICVIP_CXL_PM_TARGET_AGENT_ID_RSVD_PM2IP_MSG, \
ICVIP_CXL_PM_TARGET_AGENT_ID_RSVD_IP2PM_MSG, \
ICVIP_CXL_PM_CAPABILITY_VECTOR_IP2PM_MSG_MISMATCH, \
ICVIP_CXL_PM_ZERO_CREDIT_IP2PM_MSG, \
ICVIP_CXL_PM_MSG_VDM_CODE_INVALID, \
ICVIP_CXL_PM_MSG_LOGICAL_PM_OPCODE_INVALID, \
ICVIP_CXL_PM_EXCESS_CREDIT_RECEIVED, \
ICVIP_CXL_ILLEGAL_G1_SLOT_RECEIVED, \
ICVIP_CXL_ILLEGAL_G4_SLOT_RECEIVED, \
ICVIP_CXL_ILLEGAL_H12_SLOT_RECEIVED, \
ICVIP_CXL_ILLEGAL_H15_SLOT_RECEIVED, \
ICVIP_CXL_TYPE1_G14_RECEIVED_WITH_INVALID_MEM_RWD, \
ICVIP_CXL_TYPE1_G15_RECEIVED_WITH_INVALID_MEM_DRS, \
ICVIP_CXL_H2D_NOT_TIGHTLY_PACKED_WITHIN_G1_SLOT, \
ICVIP_CXL_H2D_NOT_TIGHTLY_PACKED_WITHIN_G12_SLOT, \
ICVIP_CXL_S2M_NOT_TIGHTLY_PACKED_WITHIN_G15_SLOT, \
ICVIP_CXL_H12_RECEIVED_WITH_FIRST_INVALID_DATA_HDR, \
ICVIP_CXL_H14_RECEIVED_WITH_INVALID_RWD, \
ICVIP_CXL_H15_RECEIVED_WITH_FIRST_INVALID_DRS, \
ICVIP_CXL_G12_RECEIVED_WITH_FIRST_INVALID_DATA_HDR, \
ICVIP_CXL_G14_RECEIVED_WITH_INVALID_RWD, \
ICVIP_CXL_G15_RECEIVED_WITH_FIRST_INVALID_DRS,
ICVIP_CXL_BI_RSP_NOT_TIGHTLY_PACKED_WITHIN_H5_SLOT, \
ICVIP_CXL_S2M_NDR_NOT_TIGHTLY_PACKED_IN_GROUP, \
ICVIP_CXL_S2M_BISNP_NOT_TIGHTLY_PACKED_IN_GROUP, \
ICVIP_CXL_D2H_DATA_HDR_NOT_TIGHTLY_PACKED_IN_GROUP, \
ICVIP_CXL_D2H_REQ_NOT_TIGHTLY_PACKED_IN_GROUP, \
ICVIP_CXL_D2H_RSP_NOT_TIGHTLY_PACKED_IN_GROUP, \
ICVIP_CXL_H13_RECEIVED_AND_DATA_SLOT_EXCEEDS_16, \
ICVIP_CXL_H15_RECEIVED_AND_DATA_SLOT_EXCEEDS_16, \
ICVIP_CXL_INVALID_MAX_S2M_BISNP_REQ_COUNT_128B_GROUP, \
```

Debugging methods

- ❑ Debug Messages
- ❑ Loggers and transcript
- ❑ Protocol Aware Debug GUI

Debug Messages

- print the traffic and transactions information in a transcript or on a shell as first step of debugging without looking into the loggers
- Text loggers – layer wise
- Debug GUI



Loggers

- ❑ layer wise loggers
- ❑ Cache and mem loggers

ICVIP_CXL_LOGGER_DIR
ICVIP_CXL_PL_LOGGER
ICVIP_CXL_RAW_DATA_LOGGER
ICVIP_CXL_TL_DLL_LOGGER
ICVIP_CXL_DEBUG_LOGGER
ICVIP_CXL_MEM_LOGGER
ICVIP_CXL_FLIT_LOGGER
ICVIP_CXL_CACHE_LOGGER

Mem Logger Snippet

Requset/Response type

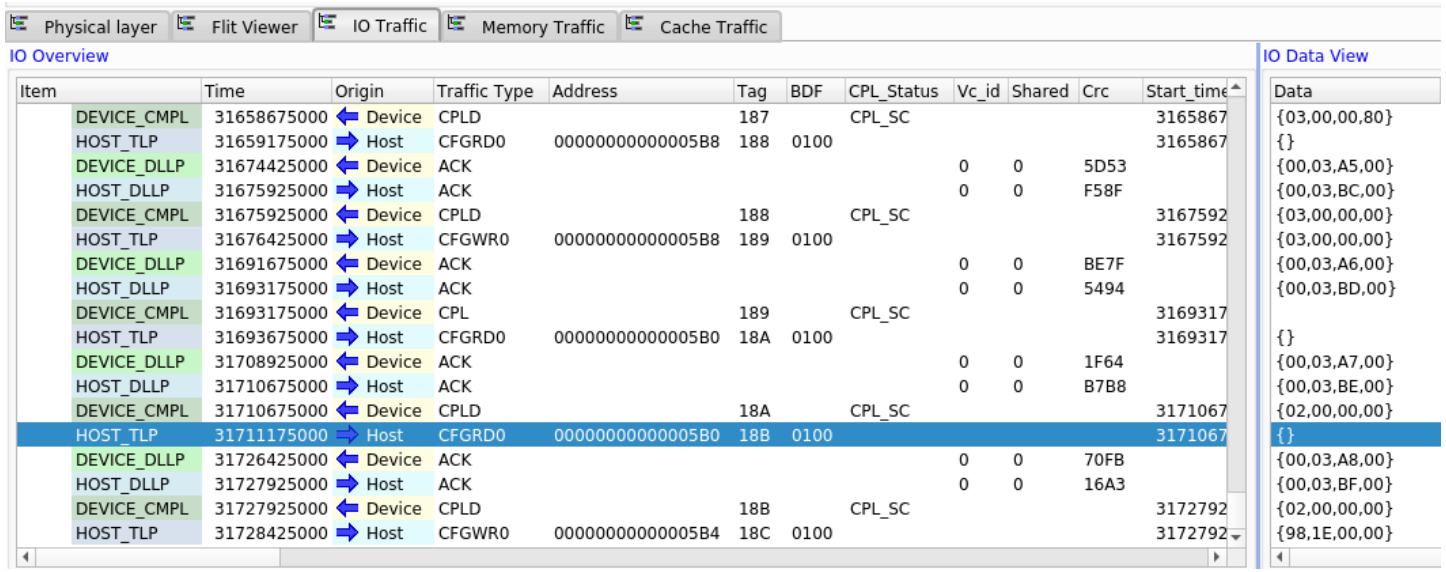
Packet fields: Value

```
31841875  |M2S_RWD      |rx_completed:0|rwd:{ld_id:0,tc:0,poison:0,address:00000076EA30,tag:0000,meta_value:MEM_META_INVALID,meta_field:MEM_META_STATE,snp_type:M2S_SNP_NOP,mem_opcode:RWD_WR,valid:1}|data:{6E,72,B8,77,
31901125  |S2M_RSP_NDR  |dev_load:1|ld_id:0|tag:0000|meta_value:MEM_META_INVALID|meta_field:MEM_META_NOP|mem_opcode:S2M_NDR_CMP|valid:1|
31901375  |M2S_REQ      |ld_id:0|tc:0|addr:00000076EA30|tag:0001|meta_value:MEM_META_INVALID|meta_field:MEM_META_NOP|snp_typ:M2S_SNP_NOP|mem_opcode:REQ_MEM_RD|valid:1|
31957125  |S2M_RSP_DRS  |rx_completed:0|drs:{dev_load:1,ld_id:0,poison:0,tag:0001,meta_value:MEM_META_INVALID,meta_field:MEM_META_NOP,mem_opcode:S2M_DRS_DATA,valid:1}|size:64B|data:{00,00,00,00,00,00,00,00,00,00,00,00
31957375  |S2M_RSP_DRS  |rx_completed:0|drs:{dev_load:1,ld_id:0,poison:0,tag:0001,meta_value:MEM_META_INVALID,meta_field:MEM_META_NOP,mem_opcode:S2M_DRS_DATA,valid:1}|size:64B|data:{6E,72,B8,77,C0,4B,F3,0F,31,12,97,FE
31957625  |S2M_RSP_DRS  |rx_completed:0|drs:{dev_load:1,ld_id:0,poison:0,tag:0001,meta_value:MEM_META_INVALID,meta_field:MEM_META_NOP,mem_opcode:S2M_DRS_DATA,valid:1}|size:64B|data:{6E,72,B8,77,C0,4B,F3,0F,31,12,97,FE
31958125  |S2M_RSP_DRS  |rx_completed:0|drs:{dev_load:1,ld_id:0,poison:0,tag:0001,meta_value:MEM_META_INVALID,meta_field:MEM_META_NOP,mem_opcode:S2M_DRS_DATA,valid:1}|size:64B|data:{6E,72,B8,77,C0,4B,F3,0F,31,12,97,FE
31958375  |S2M_RSP_DRS  |rx_completed:1|drs:{dev_load:1,ld_id:0,poison:0,tag:0001,meta_value:MEM_META_INVALID,meta_field:MEM_META_NOP,mem_opcode:S2M_DRS_DATA,valid:1}|size:64B|data:{6E,72,B8,77,C0,4B,F3,0F,31,12,97,FE
```

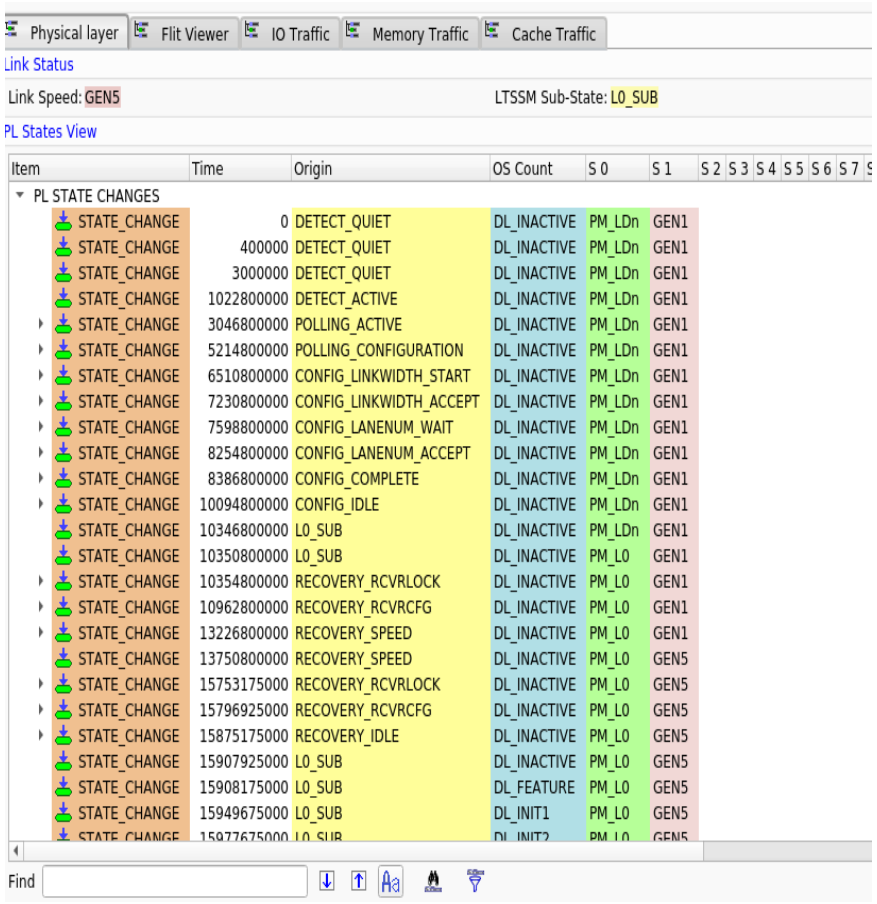
Mem traffic from Master and subordinate

Protocol Aware Debug GUI

- ❑ Physical Layer
- ❑ IO traffic



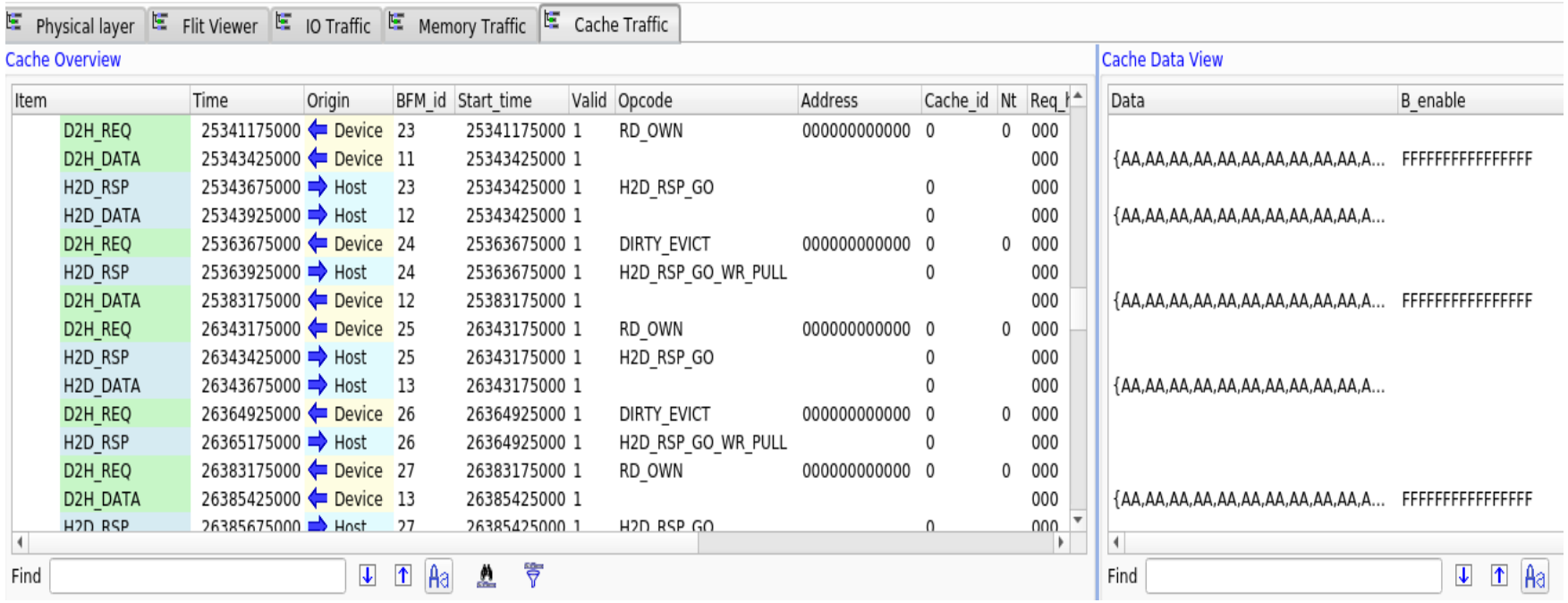
Item	Time	Origin	Traffic Type	Address	Tag	BDF	CPL_Status	Vc_id	Shared	Crc	Start_time	Data
DEVICE_CMPL	31658675000	Device	CPLD		187		CPL_SC				3165867	{03,00,00,80}
HOST_TLP	31659175000	Host	CFGRD0	00000000000005B8	188	0100					3165867	{}
DEVICE_DLLP	31674425000	Device	ACK					0	0	5D53		{00,03,A5,00}
HOST_DLLP	31675925000	Host	ACK					0	0	F58F		{00,03,BC,00}
DEVICE_CMPL	31675925000	Device	CPLD		188		CPL_SC				3167592	{03,00,00,00}
HOST_TLP	31676425000	Host	CFGWR0	00000000000005B8	189	0100					3167592	{03,00,00,00}
DEVICE_DLLP	31691675000	Device	ACK					0	0	BE7F		{00,03,A6,00}
HOST_DLLP	31693175000	Host	ACK					0	0	5494		{00,03,BD,00}
DEVICE_CMPL	31693175000	Device	CPL		189		CPL_SC				3169317	
HOST_TLP	31693675000	Host	CFGRD0	00000000000005B0	18A	0100					3169317	{}
DEVICE_DLLP	31708925000	Device	ACK					0	0	1F64		{00,03,A7,00}
HOST_DLLP	31710675000	Host	ACK					0	0	B7B8		{00,03,BE,00}
DEVICE_CMPL	31710675000	Device	CPLD		18A		CPL_SC				3171067	{02,00,00,00}
HOST_TLP	31711175000	Host	CFGRD0	00000000000005B0	18B	0100					3171067	{}
DEVICE_DLLP	31726425000	Device	ACK					0	0	70FB		{00,03,A8,00}
HOST_DLLP	31727925000	Host	ACK					0	0	16A3		{00,03,BF,00}
DEVICE_CMPL	31727925000	Device	CPLD		18B		CPL_SC				3172792	{02,00,00,00}
HOST_TLP	31728425000	Host	CFGWR0	00000000000005B4	18C	0100					3172792	{98,1E,00,00}



Item	Time	Origin	OS Count	S 0	S 1
PL STATE CHANGES					
STATE_CHANGE	0	DETECT_QUIET	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	400000	DETECT_QUIET	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	3000000	DETECT_QUIET	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	1022800000	DETECT_ACTIVE	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	3046800000	POLLING_ACTIVE	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	5214800000	POLLING_CONFIGURATION	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	6510800000	CONFIG_LINKWIDTH_START	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	7230800000	CONFIG_LINKWIDTH_ACCEPT	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	7598800000	CONFIG_LANENUM_WAIT	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	8254800000	CONFIG_LANENUM_ACCEPT	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	8386800000	CONFIG_COMPLETE	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	10094800000	CONFIG_IDLE	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	10346800000	L0_SUB	DL_INACTIVE	PM_LDn	GEN1
STATE_CHANGE	10350800000	L0_SUB	DL_INACTIVE	PM_L0	GEN1
STATE_CHANGE	10354800000	RECOVERY_RCVRLOCK	DL_INACTIVE	PM_L0	GEN1
STATE_CHANGE	10962800000	RECOVERY_RCVRCFG	DL_INACTIVE	PM_L0	GEN1
STATE_CHANGE	13226800000	RECOVERY_SPEED	DL_INACTIVE	PM_L0	GEN1
STATE_CHANGE	13750800000	RECOVERY_SPEED	DL_INACTIVE	PM_L0	GEN5
STATE_CHANGE	15753175000	RECOVERY_RCVRLOCK	DL_INACTIVE	PM_L0	GEN5
STATE_CHANGE	15796925000	RECOVERY_RCVRCFG	DL_INACTIVE	PM_L0	GEN5
STATE_CHANGE	15875175000	RECOVERY_IDLE	DL_INACTIVE	PM_L0	GEN5
STATE_CHANGE	15907925000	L0_SUB	DL_INACTIVE	PM_L0	GEN5
STATE_CHANGE	15908175000	L0_SUB	DL_FEATURE	PM_L0	GEN5
STATE_CHANGE	15949675000	L0_SUB	DL_INIT1	PM_L0	GEN5

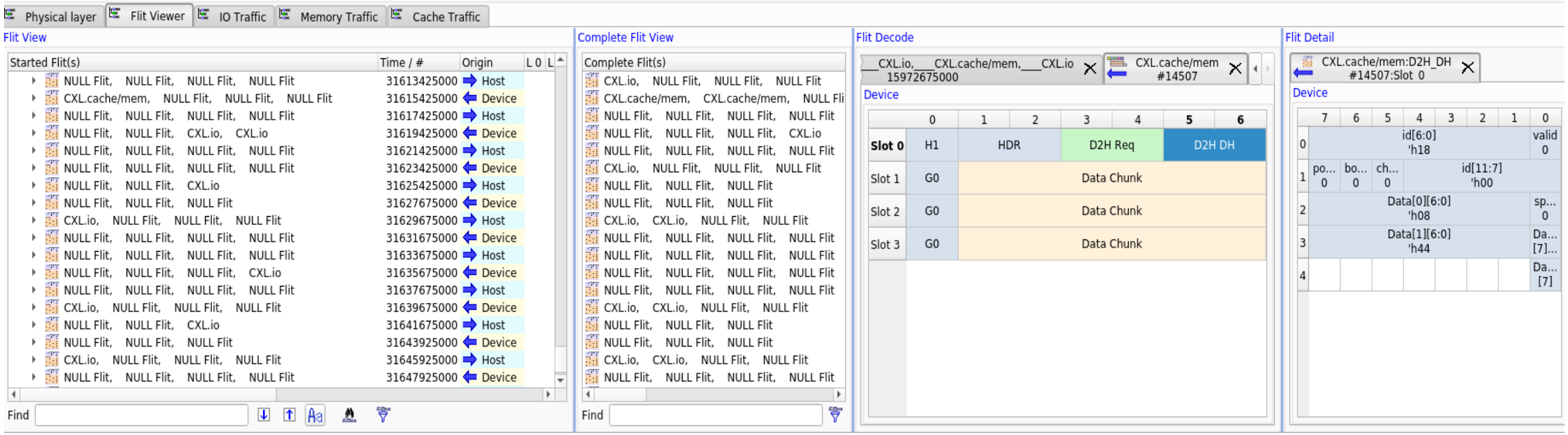
Protocol Aware Debug GUI

- Cache/Mem traffic



Protocol Aware Debug GUI

- ❑ Flit Viewer
- ❑ IO/Cache/Mem traffic



Scoreboard

- Passive component that implements the reference memory to perform read data checking
- Updates the reference memory with the write data reported by monitor when write occurs
- Compares the data from read transaction with the data from reference memory when read occurs
- Logs UVM INFO or ERROR in logs

Coverage

- ❑ Executable verification plan that hierarchically correlates to the CXL specification sections
- ❑ Each cover point has a corresponding test in compliance test suite

Section	Title	Description	Link	Type	Weight	Goal
1	Introduction	This section provides the overview of CXL and its related nomenclature.			0	0
2	Compute Express Link System Architecture	This section describes the performance advantages and key features of CXL.			0	0
3	Compute Express Link Transaction Layer	This section provides transaction layer constructs.				
3.1	CXL.io	This sections specifies the CXL.io transaction layer packet formatting, credit-based flow control etc.				
3.2	CXL.cache	This section specifies the CXL.cache transaction layer constructs.				
3.2.1	Overview	This section defines the interactions between the device and Host as a number of requests that each have at least one associated response message and sometimes a data transfer.				
3.2.2	CXL.cache Channel Description	This section defines the cache channels and provide their credit-based flow information			0	0
3.2.3	CXL.cache Wire Description	This Section defines fields for each CXL.cache channel			1	100
3.2.3.1	D2H Request opcode	This covers the D2H request opcode field	cache_cov::d2h_req_opcode	CoverPoint	1	100
3.2.3.1.1	D2H Request cqid	This covers the D2H request CQID field	cache_cov::d2h_req_cqid	CoverPoint	1	100
3.2.3.1.2	D2H Request NT	This covers the D2H request Non Temoral Field	cache_cov::d2h_req_nt	CoverPoint	1	100

Spec tagging for every section

Detailed coverage of every field

Full flexibility for handling cover points

Summary

- ❑ Automatic UVM-TB generation
- ❑ Compliance test suite (ready to use exhaustive stimulus)
- ❑ Debug capabilities
- ❑ Discussed features enabled our customers to verify their designs faster and achieve lesser time to market

Flash Memory Summit

Thank You !!

Please visit booth # for more details and demo