Verification Continuum™

# VC Verification IP CXL Subsystem Release Notes

Version S-2021.06, June 2021

SYNOPSYS®

# Copyright Notice and Proprietary Information



## Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

## Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

## Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at http://www.synopsys.com/company/legal/trademarks-brands.html.
All other product or company names may be trademarks of their respective owners.

## Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

## Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

# Release Notes

## 1 Introduction

The VC Subsystem Compute Express Link (CXL Subsystem) product offering for UVM and OVM, which provides a framework to verify using UVM and OVM Test Benches on the CXL Host or Device based designs. The designs are based on Compute Link Express Link specification supporting CXL.io, CXL.cache, and CXL.mem protocols with flex bus based physical layer.

## 2 CXL Subsystem Verification IP Notes

### 2.1 Notes for Release S-2021.06

These feature updates are part of this release:

- Enhanced the VIP to support `SVT_DEBUG_OPTS` feature. Refer to *Debug* section in *CXL Subsystem UVM User Guide* for use model.
- Added the Cryptography Library version 5.11a support for CXL IDE feature encryption/decryption.
- Fixed the Byte ordering issue for PCRC feature enabled with IDE.
- Enhanced the CXL Subsystem enumeration API, `"enumerate_cxl_device"` to check for APN negotiation and Flexbus port DVSEC status register for Cache/Mem/IO/2.0 match.
- MLD feature is EA with support for:
  - All verification topology except multi-link with LPIF.
  - Poison: Refer *section 3.23* in *CXL Subsystem UVM User Guide* to see poison usage per LD.
  - Error Injection: Refer *section 3.23.2* in *CXL Subsystem UVM User Guide* to see usage.
  - GPF: All LDs are responding to GPF as VDM does not explain control for per LD.
  - Hot Plug:
    - All the LDs are removed and added when `hot_plug` is performed. Refer *section 3.23.2* in *CXL Subsystem UVM User Guide* to see usage.
    - Refer "Supported Features and Limitations" section for hot plug limitations.
  - Viral Handling:
    - DL currently updates the `viral_ld_id_vector` for all the LDs to be in viral.

- Refer "Supported Features and Limitations" section for Viral handling limitations.

- Added the following protocol checks for PM VDM handshakes in Subsystem Application layer:
  - `pm_vdm_ip2pm_before_credit_rtn`
  - `pm_vdm_agent_id_mismatch`
  - `pm_vdm_zero_num_credits`
  - `pm_vdm_credit_rtn_timeout`

  Refer the "*Protocol Checks*" tab in *Class Reference* for details.
- Added the support for CXL IO Functional coverage for the following layers:
  - Data Link Layer
  - Transaction Layer

  Refer *CXL Subsystem UVM User Guide* for details.

### 2.1.1 Enhancement STARs

- 3640969: CXL Subsystem VIP - IDE feature - MAC control knob: Use API `'specify_mac_location'`
- 3676725: Insecure/unencrypted traffic before and while IDE is being enabled in the VIP and in the RTL

### 2.1.2 Fixed STARs

None

## 2.2 Backward Incompatibility Changes

- CXL Subsystem VIP uses the `rx_tlp_out_port` TLM port of PCIe TL and Target App. You are recommended to use the `rx_tlp_peek_port` in their testbench instead of this put_port. Otherwise the following error is displayed:

  ```
  port0.tl0.rx_tlp_out_port [Connection Error] connection count of 2 exceeds maximum of 1
  ```
- As the parameter `meta_value` is redundant for `svt_cxl_transaction` class, it is moved to `svt_cxl_channel_transaction` from `svt_cxl_common_transaction` class. Therefore, it is not available for `svt_cxl_transaction`. The `svt_cxl_transaction` class continues to use `req_meta_value` and `rsp_meta_value` for request and response meta values as before.
- **ResetPrep**:

  reset_prep_vdm_exchange (**svt_cxl_subsystem_app_vdm_transaction::reset_type_enum reset_type**, int link_num = 0);
  - Arguments are changed. Earlier highlighted input was an integer variable. It is now changed to Enum.
  - Functionality remains same and it's blocking till handshake is complete.
  - Device VIP automatically responds to ResetPrep VDM request.
- **PMReq:**
  - initiate_pm_rsp(**bit wait_for_req = 1, int pm_rsp_delay = 0**, bit [15:0] snoop_latency = 0, bit [15:0] no_snoop_latency = 0, int link_num = 0);

- Highlighted variables are now removed from argument as Subsystem App VIP now automatically responds to the incoming PM request unless svt_cxl_subsysten_app_configuration attribute "is_pm_rsp_required" is set to 0.
- You do not need to invoke this API unless PM.Rsp needs to be initiated unilaterally.

- initiate_pm_go(**bit initiate_after_rsp = 1, int pm_go_delay = 0**, bit [15:0] snoop_latency = 0, bit [15:0] no_snoop_latency = 0, int link_num = 0);
  - Highlighted variables are now removed from argument as Subsystem App VIP now automatically responds to the incoming PM request unless svt_cxl_subsystem_app_configuration attribute "is_pm_go_required" is set to 0.
  - You do not need to invoke this API unless PM.Go needs to be initiated unilaterally.
- perform_pm_req_entry(**int pm_rsp_delay = 0, int pm_go_delay = 0, bit is_pm_rsp_low_ltr = 1, bit without_pm_go = 0,** int link_num = 0)
  - Highlighted variables are now removed from argument as Subsystem App VIP now automatically responds to the incoming PM request.
  - This API invokes/waits till PMReq entry handshake is successful.

- **Multiple DOE Capability Enumeration**
  - CSR status attribute `"DOE_capability_base"` has been removed and it is advised to use PCIe EP Enumeration status attribute `"ext_cap_base_addr"`.

# 3 Supported Features and Limitations

- Limited unexpected and Error ALMP handling tested.
- The `cxl_io_warm_reset` test might fail on some seeds where `HOT_RESET` is issued and in parallel IO packet scheduled to PL layer. The resolution will be part of the subsequent release.
- Viral state exit in CXL Subsystem Cache-mem TL can happen through specific sequence API instead of mid-sim reset.
- Memory interleaving is not supported for multi-host or multi-device topology.
- The default value of Synopsys controller for CXL Cache and Mem Component registers filed `array_size` is invalid. Therefore, CXL Subsystem VIP `enumerate_cxl_device` API is tweaked such that it cannot access capabilities with value greater than 8.
- The `expected_retry_req_received` check might fail for few seeds when recovery re-initiated during ALMP exchange after recovery.
- CXL IDE feature is supported for VCS simulator only.
- Euclide lint check errors are expected in UVM 1.2 with `UVM_NO_DEPRECATED` macro.
- Euclide lint check is not supported with IEEE 1800.2 UVM.
- For CXL 2.0, if flex Bus enable bits(IO/Cache.Mem) reflected in the received '`modified TS2 Ordered Sets`' during configuration.complete LTSSM state have different values compared to '`modified TS1 Ordered Sets`' received before, then the VIP does not enter into Detect state.
- For CXL 2.0, the CXL.IO and PCIe enable bits are not mutually exclusive when Host VIP communicates results in Phase 2.
- When using Hot Plug API (`perform_hot_plug`), the following issues might be observed:

  a. Failure of following protocol checks

- dl_received_ack_protocol_viol
- dl_initfc_credit_timeout
- arb_mux_unexpected_almp_received_error

b. This fatal message:

```
UVM_FATAL ../../../../pcie_svc/PCIE/Verilog/TransactionLayer/
pciesvc_transaction.sv(3109) @ 59239793.00 ps:
uvm_test_top.pcie_tests_0.env.rc_env.rc_agent.port0.tl0 [report_message]
ReceiveCredits:
```

- Currently, CXL VIP does not support data before response scenarios for H2D Snoop transactions.
- Currently, CXL Cache Mem VIP does not support `addr[5:0] !=0`. So, `addr[5:0]` is defined as '0' inside the VIP for further processing of the transactions.
- ARB/Mux uses previous snapshot on unsuccessful ALMP exchange after recovery as mentioned in spec for `STATUS_EXCHANGE_PENDING` flag. Subsequent release will be updated to resolve VLSM as per resolution table.
- MLD feature limitations:
  - CXL IO side is not supported in this release.
  - `TL_DL_ARB_MUX` back-to-back is not supported when running hot plug.
  - Viral handling is only supported for TL-TL verification topology.
  - HVP and spec linking is not updated with MLD portion of the spec.
- Tests exercising recovery transition in parallel to CXL.IO traffic in progress may result in a hang due to a VIP behavior that the team is seeking clarity from the CXL Consortium.

  This would involve an unexpected state transition to Recovery.Rcvr_Lock LTSSM state. Please contact Synopsys for further assistance.
- Additional debug messages will be observed from PCIe VIP TL Configuration. It is suggested to demote those messages.

```
UVM_INFO svt_pcie_tl_configuration.sv(2748) @ 0.00 ps: reporter [copy_dynamic_fields]
new_inst.tx_ide_selective_stream_cfg.num()=0,

UVM_INFO svt_pcie_tl_configuration.sv(2749) @ 0.00 ps: reporter [copy_dynamic_fields]
new_inst.rx_ide_selective_stream_cfg.num()=0,

UVM_INFO svt_pcie_tl_configuration.sv(2368) @ 0.00 ps: reporter [copy_static_fields]
this.tx_ide_link_stream_cfg.num()=0,
```

The complete list of features and limitations are captured in Section 1.2 of CXL Subsystem User Guide.

# 4 Simulator and Methodology Support

- Simulators Supported:
  - VCS (version Q-2020.03 onwards)
  - Xcelium (UVM, version 18.xx onwards).
    Note - For Xcelium Version 19.xx and above add "-VLOGCONTROLRELAX UMINQT" in build options to demote compile time errors.
  - MTI (UVM, version 2019.4 only)

- Methodology Supported:
  - Accellera UVM,
  - Accellera OVM and
  - IEEE UVM (Supported with VCS simulator only)

The simulator matrix table is available at the following location:

VC VIP Library page
https://spdocs.synopsys.com/dow_retrieve/latest/vg/snps_vip_lib/PDFs/simulator_matrix.pdf

**Note**
CXL SUBSYSTEM SVT supports only MTI simulator version 2019.04.

# 5 Notes for Previous Releases

## 5.1 Notes for Release R-2021.03-2

These feature updates are part of this release:

- Added the support for CacheFlushed transaction.
- Updated the API for HOT UNPLUG/PLUG for selectively unplugging particular LPIF instance using `lpif_id_I` argument.
- Updated the use model for HOT PLUG `@LINK LPIF` VIP such that after seeing `pl_inband_pres` LINK LPIF, the VIP internally unplugs the stack above it.
- Added the support for 32B and Mix-mode (Mix of 32 and 64 byte traffic) for CXL.cache/mem for both Host and Device VIP.
- Supported the controlling of CXL Cache-mem traffic suspending/ resuming using sequence APIs.
- Supported CXL.io Flit mode in ARB/Mux to enable set flag `svt_pcie_dl_packet::enable_cxl_io_flit_mode =1`. By default, CXL.io is configured in TLP/DLLP mode.
- Added the following protocol checks for PM VDM handshakes in Subsystem Application layer.
  - `pm_vdm_response_timeout`
  - `pm_vdm_go_timeout`
  - `pm_vdm_unexpected_response`

  Refer the "*Protocol Checks*" tab in *Class Reference* for details.
- Added the support for MLD feature Phase-1.
  - Host can send the transaction to MLD device having any LDID lesser than the configuration class variable `"num_of_ld_id_supported"`, if MLD is enabled.
  - MLD Device will have multiple CXL agents depending on the number of LDID configured. If the `num_of_ld_id_supported` is configured as 8 in API then there will be 8 CXL agents.
  - Device will receive the transaction and depending on LDID, CXL agent will process the transaction and send back the response.

  For detailed description of API and its usage, refer the *MLD Feature (3.23)* section in *CXL Subsystem UVM User Guide.*
- Enhanced the Multiple DOE Capability Enumeration feature as suggested:

- DOE capabilities base addr is stored in `ep_enumeration_status.ext_cap_base_addr[index]` where index is => `{24'h0,8'h DOE Cap Instance number,16'h Cap ID, 16'h Target BDF}`
- DOE Types enumeration is supported via all DOE capabilities.

  The `"enumerate_cxl_doe_object_types"` API has been enhanced to visit each DOE Capability to discover object protocol types.

- Enhanced the CXL IDE Feature as suggested:
  - Default Endianness is updated to "BIG Endian".
  - AES-GCM Debug control is passed via Cache-Mem configuration class.
    - `enable_aes_gcm_dpic_messages`
    - `cxl_aes_gcm_endianness`
  - Key: Default input is treated as Little Endian.
- Added the debug information error message for the error, *"CXL DVSEC not found"*.
- Updated the default value of the following configurations from 500ns to 300ns:
  - `svt_cxl_subsystem_app_configuration::max_pm_rsp_delay`
  - `svt_cxl_subsystem_app_configuration::max_pm_go_delay`

### 5.1.1 Enhancement STARs

- 3627217: Multiple DOE Capability Enumeration
- 3652560, 3636742: CXL IDE Feature
- 3472256: Add debug tip for error *CXL DVSEC not found*
- 3628774: `svt_cxl_subsystem_virtual_io_enumeration_sequence::configure_bars()` throws error (For details, refer to *PCIe SVT Release Notes*.)

### 5.1.2 Fixed STARs

3644348: UVM FATAL for ep_enumeration_status for Device Full stack VIP

## 5.2 Notes for Release R-2021.03-1

These feature updates are part of this release:

- TB Level Updates:
  - Moved the macros, "`SVT_CXL_SUBSYSTEM_TS_GEN_TL_CACHE_MEM_TRAFFIC" and "`SVT_CXL_SUBSYSTEM_TS_GEN_CXL_IO_TRAFFIC" with "svt_cxl_subsystem_ts_api_base_sequence.sv" base sequence API "generate_random_cxl_io_traffic" and "generate_random_cxl_tl_cm_traffic".

    Refer HTML Class reference guide for details on the API.
- EP Enumeration Status Update for DVSEC/VSEC storage:
  - Updated the existing PCIe EP enumeration status class attribute for direct differentiation of DVSECs/VSEC.

    ```
    bit [11:0] ext_cap_base_addr[bit[39:0]]; => bit[11:0]
    ext_cap_base_addr[bit[63:0]];
    ```
  - Access/indexing way for all extended capabilities:

DVSEC: <16'h DVSEC VENDOR ID> <16'h DVSEC ID> <16'h23 (DVSEC Capability ID)> <16'h BDF>

VSEC: <16'h0> <16'h VSEC ID> <16'hB (VSEC Capability ID)> <16'h BDF>

Others: <16'h 0> <16'h 0> <16'h Capability ID> <16'h BDF>

**Note**: You may face a backward compatibility issue while trying to access capability because array indexing is changed from 40bits to 64bits.

**Action**: Change array accessing index attribute to 64bits width.

- DOE Object Type enumeration support via "enumerate_cxl_device" API:
  - Supported the DOE Object type enumeration based on the "disable_cxl_doe_object_type_enumeration" parameter. By default, this parameter is disabled.
- CSR Status class keep supported DOE Object Types capability information captured through associative array:

  DOE_object_type_addr[<Object Type><Vendor ID>]

  For example:
  - DOE_object_type_addr[021e98] - CXL_COMPLIANCE_DOE_MAILBOX
  - DOE_object_type_addr[001e98] - CXL_CDAT_DOE_MAILBOX
- Enhanced the "enumerate_cxl_device" with the removal of first MWr TLP for RCRB area in case of CXL 1.1 device. CXL 2.0 Specification section "*CXL 1.1 Upstream Port RCRB*" states that TLP should be MRd.
- Added the CXL 2.0 Cache/Mem TL Protocol check hvp with spec linking.
- Subsystem Sequence Collection API enable_aspm_l1_entry() Updates:
  - Supported the checks for level of ASPM via read the PCIe link capability register.
  - If ASPM L1 support is found, writes PCIe Link control register to enable L1 entry.

## 5.3 Notes for Release R-2021.03

These feature updates are part of this release:

- The mem_bar_mapped_to_internal_reg attribute is mapped to CSR Status class for use in traffic generation.
- Host DUT Address randomization in macro SVT_CXL_SUBSYSTEM_TS_GEN_CXL_IO_TRAFFIC is updated to exclude memory ranges.
- The CXL 1.1 Specification applicable RCRB next capability offset as '100h' check is fixed and added a control to disable the check disable_cxl_1_1_upstream_port_rcrb_nxt_cap_off_check=0.
- Updated enumeration API to use link_num in the CFG/MEM traffic generation.
- Fixed typos in the enumeration API in Device Register interface.
- Viral handling feature is supported.

- Added support for Hot plug and Mid-sim reset features. CXL VIP provides a sequence to execute Hot Plug flow. Hot plug flow comprises of Hot removal and Hot add. Mid-sim reset feature is supported as one of the Hot Plug flow. This indicates that "Hot removal followed by immediate Hot add". Currently, these are EA features with preliminary validation. Extended validation is in progress.
- Added CXL Subsystem Application component infrastructure with dedicated configuration, status class and sequencers for transaction and service sequences. Protocol checks are also added for PM VDM feature. You can refer the CXL Subsystem User Guide and SVDOC for more details.
- New configuration variables (`svt_cxl_subsystem_app_configuration`) added for PMReq. CXL Subsystem Application layer provides controls for different responses and their values for PMReq handshake. These are the ones which can be configured to control the handshake.
  `min_pm_rsp_delay`, `min_pm_go_delay`, `max_pm_rsp_delay`, `max_pm_go_delay`, `is_pm_rsp_required`, `is_pm_go_required`, `min_pm_req_snoop_latency`, `min_pm_rsp_snoop_latency`, `min_pm_go_snoop_latency`, `max_pm_req_snoop_latency`, `max_pm_rsp_snoop_latency`, `max_pm_go_snoop_latency`.

  Format for latency field is as per PCIe LTR format. (Latency scale + Latency value)
- **Euclide Support**
  - Added support for Euclide (Eclipse based IDE) for lint rule checking. The VIP works seamlessly with Euclide IDE when configured with testbench rule setting and would not result in any fatal errors.
  - For resolving SVT related errors, please use SVT version R-2020.12-T-20210208.
  - UVM 1.2 is supported without `UVM_NO_DEPRECATED` macro.

### 5.3.1 Enhancement STARs

None

### 5.3.2 Fixed STARs

- 3576344: Using CXL.mem API meta_value(req_meta_value) transaction log print issue.

## 5.4 Notes for Release R-2020.12-2-T-20210129

These feature updates are part of this release:

- Backward incompatible changes in Endpoint Enumeration status class handling through configuration db from test/env.

```
Config DB hierarchy is changed, please refer following :

  for(int i=0;i<cust_cfg.host_cfg.num_cxl_io ;i++) begin

    io_root_hierarchy_string =
$sformatf("pcie_root_hierarchy_%0d_ep_enumeration_status",i);


svt_config_object_db#(svt_pcie_ep_enumeration_seq_status)::set(null,"*",io_root_hierarch
y_string, pcie_ep_enumeration_status);

  end
```

  If it is not updated, then the following error message is observed:

```
UVM_FATAL svt_pcie_device_virtual_sequence_collection.sv(252) @ 0.00 ps: reporter
[svt_pcie_device_virtual_base_sequence::body] Could not get handle for
ep_enumeration_status object.
```

- Endpoint enumeration sequence status class passing is updated for OVM, and the object passing from Testbench to VIP is consistent for OVM and UVM.
- The `enumerate_cxl_device` is enhanced to support multiple instances of set and get through config db method. The `cxl_io_enumeration_status_id` attribute is added in `svt_cxl_subsystem_virtual_api_collection_sequence` sequence. You need to program this attribute and it is mapped internally with `pcie_root_hierarchy_%0d_ep_enumeration_status`.
- The `enumerate_cxl_device` API is enhanced to remove the dependency of IO enumeration sequence handle to get any capability, bus number, or device number.
- CXL 1.1 RCRB Memory space enable bit can be programmed.
- CXL.io Memory space backdoor programming APIs are added in CXL Subsystem API collection sequence.
- CXL Subsystem VIP uses the `rx_tlp_out_port` TLM port of PCIe TL and Target App. This restricts to use this port in Testbench. If you need to connect this TLM in your Testbench, then you need to configure `max_num_exports_of_rx_tlp_out_port` of tl_cfg/target_app_cfg accordingly. Otherwise the following error message is displayed:

  ```
  port0.tl0.rx_tlp_out_port [Connection Error] connection count of 2 exceeds maximum of 1
  ```
- For Power management (PM) VDM Messages, Vendor ID '8086 is also supported, if Alternate Protocol Negotiation (APN) is negotiated with '8086 value (based on the value set in `` `SVT_CXL_DEFAULT_APN_VENDOR_ID ``).
- Support for invoking PM VDM APIs from separate sequences has been included.
- Viral handling is supported.
- Memory Interleaving is supported.

## 5.5 Notes for Release R-2020.12-2

These feature updates are part of this release:

- The `enumerate_cxl_device` is enhanced to enumerate all capabilities by default. You can access those through the PCIe EP Enumeration status class associative array attribute `ext_cap_base_addr`.

### 5.5.1 Enhancement STARs

- None.

### 5.5.2 Fixed STARs

- 3526384: NOA error for APP BFM dispatch seqr

  Fixed TL layer connections to guard with dut_model are not equal to RTL.

## 5.6 Notes for Release R-2020.12-1-T20201221

These feature updates are part of this release:

- L2 state - CXL device initiated entry and exit initiated by host.

- Added the top level plan `cxl_svt_checker_coverage_top_level_plan.hvp` and `cxl_svt_checker_coverage_top_level_plan.hvp.xml` for Cache/Mem TL checks
- Enhanced CXL Security Feature for PCRC Support.
- Added support for PMReq.Req, PMReq.Rsp and PMReq.Go.
- These are the CXL Device Enumeration updates:
  - CXL 2.0 Memory mapped Component registers access are enhanced for limited capabilities.
  - The `get_cap_seq` instance is added to get PCIe Capabilities in user sequences.
  - The `enumeration_completed` status attribute is added in CSR Status.

### 5.6.1 Enhancement STARs

- 3413705: Poison constraint on the M2S needs to be allowed to randomize
- 3459280: Capability to test the metavalue setting of 0x1
- 3460176: SVT_CXL_SUBSYSTEM_TS_GEN_CXL_IO_TRAFFIC is targeting bar 0 only
- 3472265: Remove unnecessary enumeration messages if CXL DVSEC not found.

### 5.6.2 Fixed STARs

None.

## 5.7 Notes for Release R-2020.12

The following feature updates are part of this release:

- Qualified IEEE-UVM with version 1800.2-2017-1.1
- Qualified with VCS +lint=LRM_1800_2009 (VCS Version 2020.12)
- Qualified with Xcelium Version-20.09.002 and MTI Version-2020.3
- ARB/Mux updated to handle race during physical layer retrain and queued higher layer traffic.

### 5.7.1 Enhancement STARs

- None.

### 5.7.2 Fixed STARs

- None.

## 5.8 Notes for Release R-2020.09-3

The following feature updates are part of this release:

- CXL2.0 features (EA)
  - CXL Cache/mem IDE, containment mode with no PCRC support
  - Global Persistent Flush
  - QoS Telemetry
  - Data Object Exchange
  - ARB/Mux enhanced to exchange ALMP in CXL.io only mode after APN negotiated for CXL2.0 version.

- CXL VDM API enhanced for Reset Types.
- Basic functional coverage (EA) for ArbMux, Cache/mem TL and DL. Refer the class reference Coverage Tab for details.
- High Level Verification plans using with Verdi coverage analysis framework.

You can refer the CXL User Guide for more details.

### 5.8.1 Enhancement STARs

None

### 5.8.2 Fixed STARs

None

## 5.9 Notes for Release R-2020.09-2

These are the enhancements in this release:

- This is CXL 2.0 EA release. This release incorporates CXL 2.0 features (APN negotiation, Enumeration, MemSpecRd). CXL 2.0 features requires new license for CXL 2.0. You can contact Synopsys support for details.
- These checks are added in ARB/Mux.
  - `arb_mux_unexpected_almp_received_error`
  - `arb_mux_error_almp_received_error`
- Link Reset Supported added in ARB/Mux.
- Time scale related issue fixes. CXL Subsystem VIP has no problem with timescale of TB now.
- CXL PM credit initialization and Warm Reset are supported and APIs are added for the same. Details can be found in 3.3.2 section of UG. Also example test (cxl_io_warm_reset) is added in VIP example area.
- Support for Enumeration of CXL 2.0 Device.
- CXL.Mem transaction fields combinations are updated based on the CXL specification version.
- Added new protocol checks for CXL TL Mem(`m2s_req_memspecrd_valid_meta_field`, `m2s_rwd_memwrptl_valid_rsp` and `m2s_req_memspecrd_valid_meta_field`) and updated the existing CXL Mem checks as per CXL 2.0 Appendix B. Refer HTML class reference page for further details:

  ```
  $DESIGNWARE_HOME/vip/svt/cxl_subsystem_svt/latest/doc/
  cxl_subsystem_svt_uvm_public/html/protocolChecks.html
  ```

### 5.9.1 Enhancement STARs

- 3346379: CXL Mem Protocol checks will need to be updated to match latest CXL spec(2.0).
- 3398631: time unit issue and protocol check issue
- 3391050: CXL 2.0 Enumeration Phase1
- 3407848: CXL 2.0 Enumeration Phase2
- 3391049: CXL VIP functional coverage specification - scope & strategy
- 3391050: Support for Reset state in ARB/MUX when LTSSM goes to Detect

### 5.9.2 Fixed STARs

None

### 5.9.3 Backward Incompatibility Changes

Overall: R-2020.09-2 release of CXL is compatible with R-2020.09-2 of PCIe release only. svt_pcie_pl_configuration::cxl_spec_ver is used inside CXL subsystem API(configure_flex_bus) which is not available in previous releases of PCIe.

## 5.10 Notes for Release R-2020.09-1

These are the enhancements in this release:

- Support for CXL.cache/mem ordering rules.
- Support for Bias mode.
- Data interleaving depth support for Data interleaving (CXL H2D 32 byte data transfer).
- Support in CXL Cache-Mem TL driver to make ID/ TAG unique for new transactions compared to outstanding transactions.
- Backdoor Access to Memory contents or Memcore.
- Bogus field Support for CXL.Cache Write transactions.
- Poison field support for CXL.Cache/Mem transactions.
- Updates for Functional Coverage Phase 2 have been added.
- These debug ports are added in LPIF interface:
  - ascii_lp_state_req
  - ascii_pl_state_sts
  - ascii_pl_protocol
  - ascii_pl_speedmode
  - ascii_pl_clr_lnkeqreq
  - ascii_pl_set_lnkeqreq
  - link_lpif_current_ssm_state
  - phy_lpif_current_ssm_state

  These signals are driven only when LPIF configuration variable `drive_debug_signals` is enabled. Refer the class reference for more details.
- Enumeration
  - Fixed capability pointer capture after CXL DVSEC observations. Payload bits 11:0 was used instead of 31:20.
  - Modified looping of Capability pointer to process all DVSECs or capabilities.
  - Added control in `enumerate_cxl_device` for PCIe enumeration parameters that can give more control in CXL.io enumeration sequence flow.
  - Added collision detection and avoidance mechanism for BAR, RCRB, and MEMBAR0 regions for `enumerate_cxl_device` API.
- PCIE VIP data classes are enabled in CXL Subsystem class reference.

### 5.10.1 Enhancement STARs

- 3358179: CXL LPIF VIP updates to bypass RETRAIN state during initial linkup as PHY DUT not following LPIF spec.

  Updated VIP to bypass RETRAIN state during initial linkup

### 5.10.2 Fixed STARs

None

## 5.11 Notes for Release R-2020.09

These are the enhancements in this release:

- CXL subsystem SVT EA license keys are no longer supported.
- LPIF: HVP and Spec linking added for Protocol Checks and Functional Coverage.

  The base specification used for Spec linking is LPIFv1_1. Refer the CXL Subsystem UVM User Guide for usage details.

## 5.12 Notes for Release Q-2020.06-3

These features are supported in this release:

- Device initiated transactions on device attached memory.
- ARB/Mux Trace updated to print received DL packet from PHY.
- To handle error scenario, ARB/Mux Service request updated to abort L1.x (ABORT_L1_X) and retrain (ABORT_RETRAIN) request.
- Added `io_state_req_l1_almp_delay_ns` and `cm_state_req_l1_almp_delay_ns` configuration to delay the ALMP request during initialization for IO and CM.
- Support for Data interleaving for CXL H2D 32 byte data transfer.
- CXL Subsystem top level interface is changed from `svt_cxl_if` to `svt_cxl_subsystem_if`.
- MTI simulator UVM flow(CXL.cache/mem)
- LPIF: default specification version is updated to 1.1.
- LPIF: Protocol checks specification section updated based on specification version 1.1 to be reflected in class reference document.
- LPIF: Support added for optional handshake `pl_clk_req/lp_clk_ack` supported by Spec version 1.1.
- LPIF: Functional Coverage is supported. Refer the Class reference document for details.

## 5.13 Notes for Release Q-2020.06-2

These features are supported in this release:

- IEEE UVM for VCS simulator
- L1 state - CXL device initiated entry and exit
- 32-byte(32B) transfers for CXL Cache
- MTI simulator UVM flow(CXL.io)
- CXL TL Callbacks

- CXL Device enumeration API (`enumerate_cxl_device`) is added in the API Sequence class for CXL.io, Upstream port RCRB, and MEMBAR0 registers of the device.

Enhanced STARs:

- P10183633-1281 - CXL Subsystem VIP to advertise 'h1E98 instead of 8086.
  - The `configure_flex_bus` API inside `svt_cxl_subsystem_configuration` is now updated to advertise vendor ID as 16'h1E98 instead of 16'h8086 by default based on the specification.
  - To continue with 16'h8086, you must provide vendor ID value input to this API.
- P10183633-1312 - Scalable TB flow for CXL SS VIP and TS
  - You must use `svt_cxl_subsystem_user_base_test` and `svt_cxl_subsystem_user_basic_env` for VIP as well as TestSuite usage

## 5.14 Notes for Release Q-2020.06-1-T-20200630

The following features are supported in this release:

1. CXL.Mem Type 3 transactions:
   a. M2S Req Transactions:
      - MemINV, MemRd, MemRdData, MemInvNT
   b. M2S RwD Transactions:
      - MemWrPtl
2. Partial cache-line transaction under CXL.cache and CXL.mem protocol involves Byte enable

### 5.14.1 Backward incompatible changes

- The top and bottom layer are moved out of `svt_cxl_system_configuration`: The top_layer' and `bottom_layer` attributes have been moved from the `svt_cxl_system_configuration` to `svt_cxl_cache_mem_configuration` class. This means that instead of `cust_cfg.cache_mem_sys_cfg.top_layer`, it needs to be accessed as `cust_cfg.cache_mem_sys_cfg.host_cfg[index].top_layer` or `cust_cfg.cache_mem_sys_cfg.device_cfg[index].top_layer`

  ```
  svt_cxl_cache_mem_configuration::  cxl_layer_enum top_layer = TRANSACTION;
  cxl_layer_enum bottom_layer = LIN
  ```

  Further, this indicates that you need to know about the top or bottom layer that needs to be set for a specific Host or Device. An utility API `set_top_layer()` or `set_bottom_layer()` in `svt_cxl_system_configuration` can also be used for this purpose. The key aspect to be noted is that the layer information needs to be set for each Host and Device agents separately

Enhancement STARs:

- 3265824: Request to relax constraints to support required memory transactions.

## 5.15 Notes for the Q-2020.06-1 Release

The following are the features of this release:

- CXL.mem Type 2 device transactions:
  - M2S Req transactions: MemINV, MemRd, MemRdData, MemInvNT
  - M2S RwD transactions: MemWr

- Memory Model of type Memory Core or Memserver for CXL.mem Type 2 and Type 3 device transactions.
- Transaction logging feature for ARB-MUX component.
- LPIF Feature Support
  - Data transfer support for LPIF 1.0 and LPIF 1.1
    - Shared/non-shared Data Bus
    - Cache.mem mode
    - IO mode
  - LPIF states supported
    - RESET
    - ACTIVE
    - L1 with cache/mem
    - DISABLED (Internal request to move to DISABLED state is not yet supported)
    - RETRAIN

### 5.15.1 Enhanced STARs:

- 3220290: SS configuration support added for topology#2 with cache.mem and io in LPIF mode
- 3251814: ARB/Mux Trace updated to log DLLP/TLP details, IO/CM VLSM transition, and ALMP transactions
- 3260800: LPIF topology support to make customer consumable.