Accelerating Verification of Computational Storage Designs

Presenter: Ujjwal Negi

Siemens EDA

Agenda

❑ **The Next Era of storage technology**

- Computational Storage
- Traditional vs Computational storage model
- What's New

❑ **Functional Verification**

- Challenges
- Solution

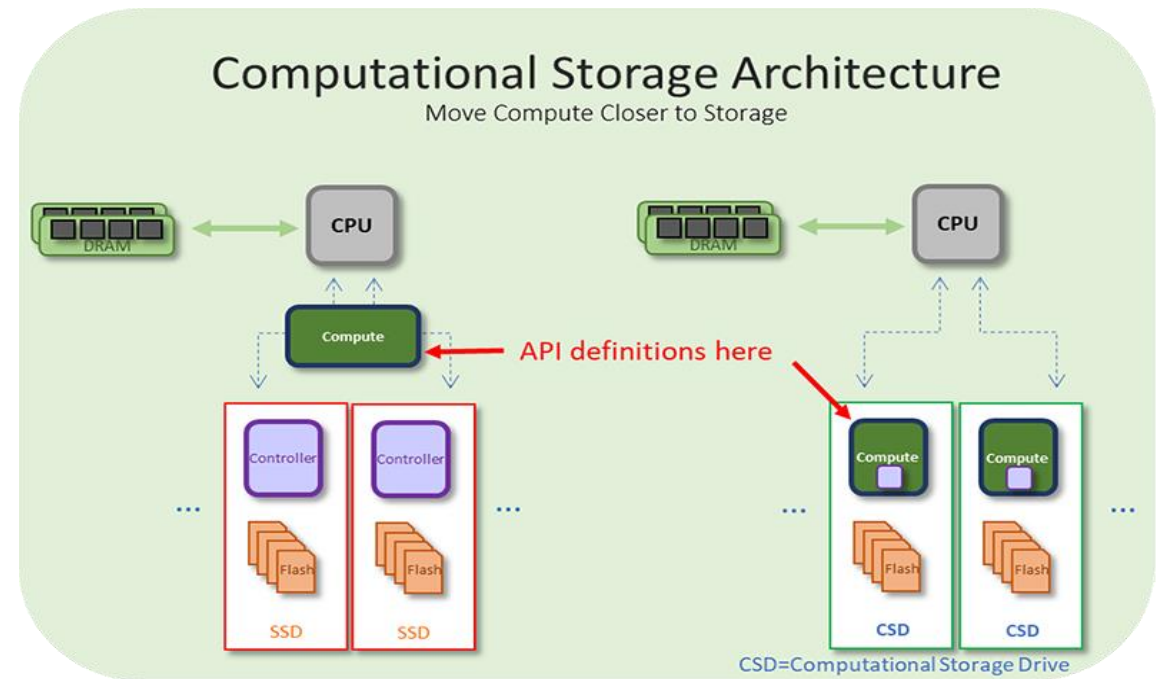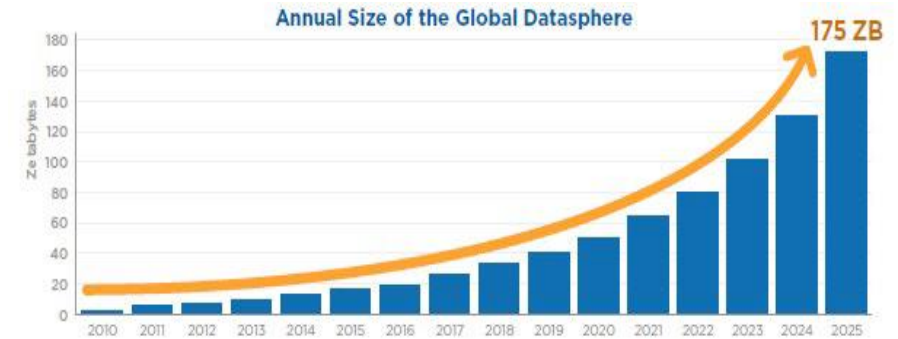# The Next Era of Storage Technology

Computational Storage

- Exponential growth in data

- Need for faster, efficient data processing solution

❑**Solution**

- Replacing traditional NVMe SSDs with Computational Storage drive (CSDs)

- Integrates processing power directly into the storage device

Annual Size of the Global Datasphere

175 ZB

Computational Storage Architecture
Move Compute Closer to Storage

API definitions here

CSD=Computational Storage Drive

Traditional vs Computational Storage Model

Traditional model

Request data from storage

Move data to compute

Move results to storage

SSD

Computational storage model

Request operation from storage

Operation result

CSD

Benefits of Computational Storage

- Offloading host processing
- Reduces data movement
- Improved Performance
- Lower latency
- Bandwidth efficiency

What's New

❑**Command Sets**

- Computational Program
- Subsystem Local Memory

❑**Namespaces**

- Compute - resource on which program are stored and executed
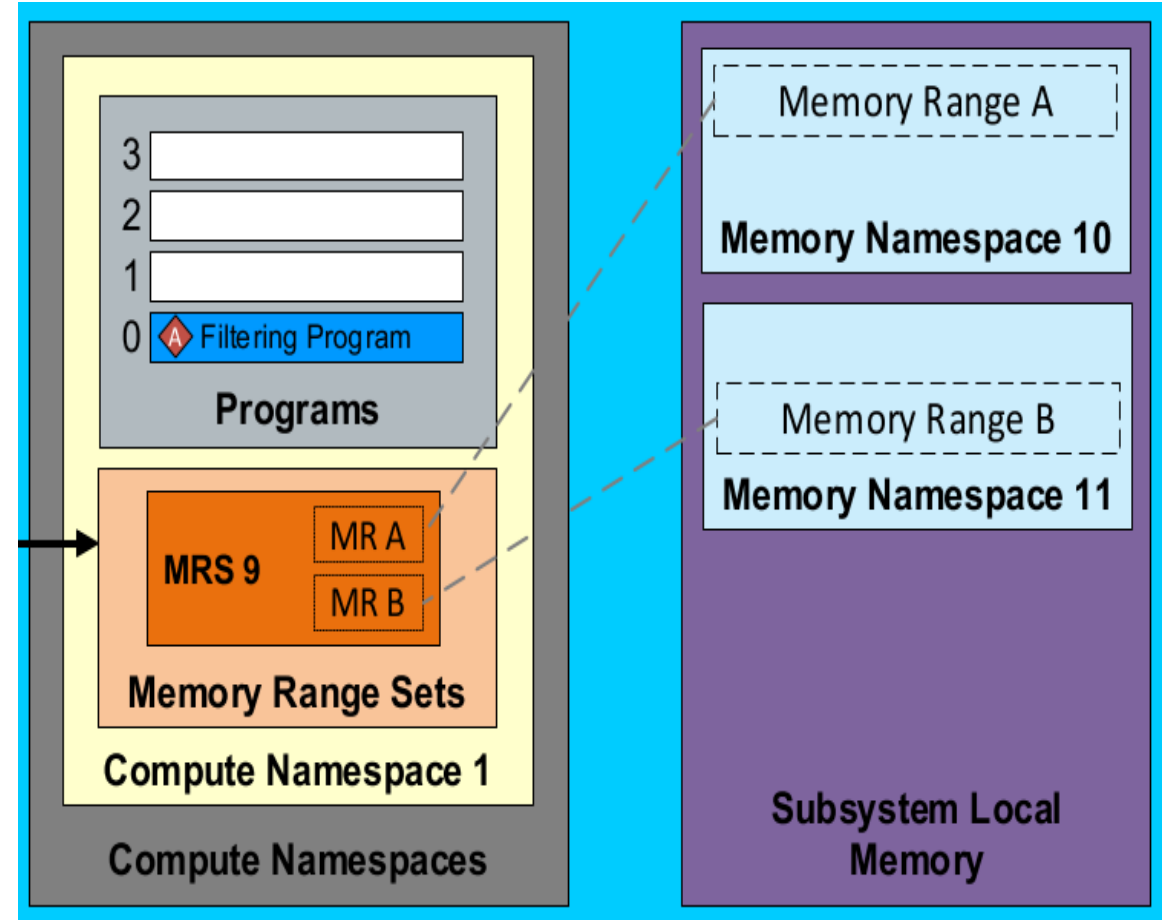- Memory - resource on which I/O data for programs is stored

What's New

☐ **Programs**

- Functional pieces of code designed to accomplish a well-defined purpose
  1. Downloadable
  2. Device-defined

☐ **Memory Range Sets**

- Collection of memory namespace ranges
- Specific to a compute namespace

Functional Verification

Challenges & Solution

Protocol Compliance

➤ verifying that Computational Programs and SLM commands do not introduce protocol violations or unexpected behaviour
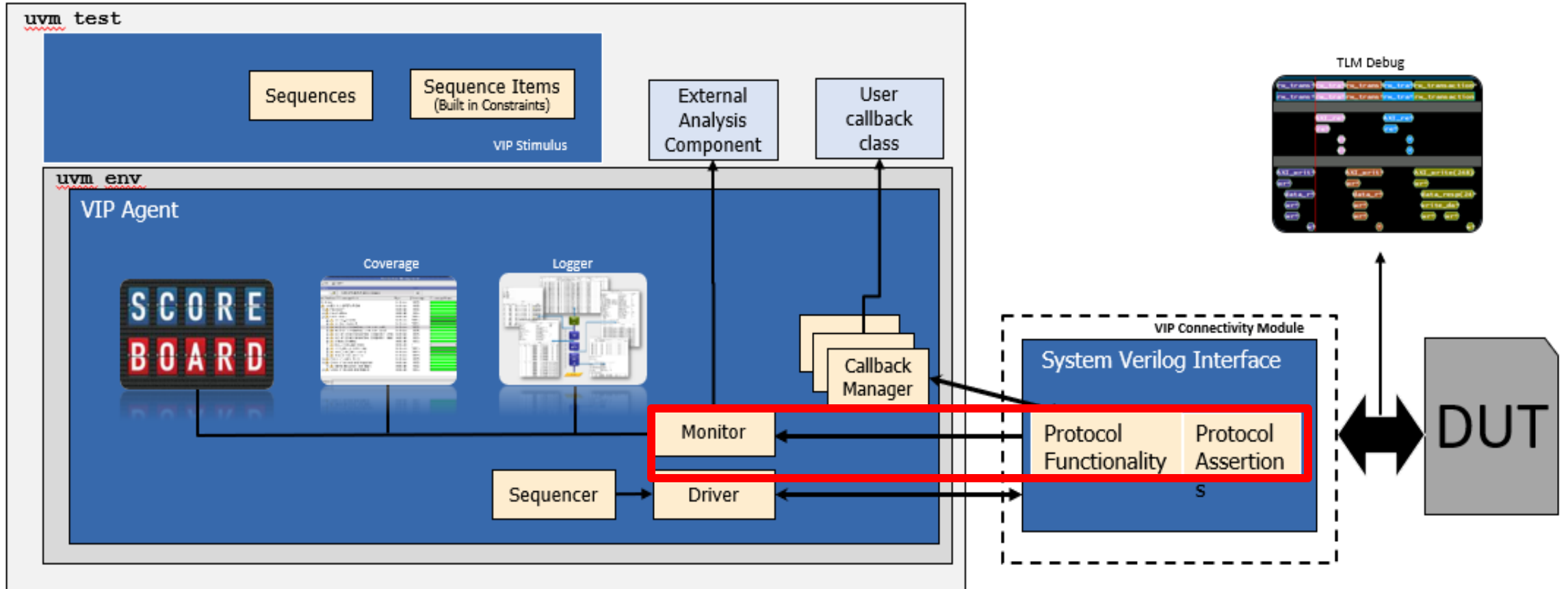
❑ Independent Monitor

- Decodes all transport packets

- Watches complete address space

- Checks for any unnecessary/ unrelated transport packets

- Can be plugged into existing test environment
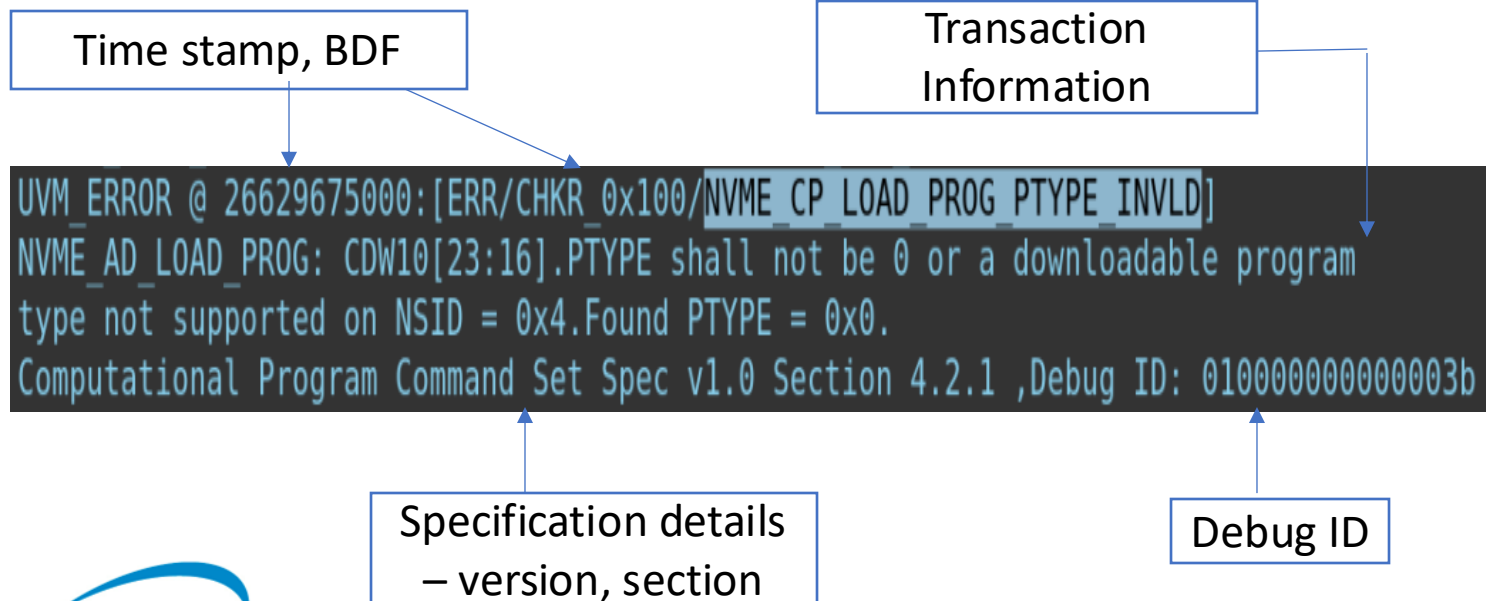
Protocol Compliance

❑ **Independent Monitor**

Protocol Compliance

❑ Exhaustive Protocol suite

- Around 1100+ protocol checks

- Around 100+ assertions based on Computational Program and SLM command set

Time stamp, BDF

Transaction Information

```
UVM_ERROR @ 26629675000:[ERR/CHKR_0x100/NVME_CP_LOAD_PROG_PTYPE_INVLD]
NVME_AD_LOAD_PROG: CDW10[23:16].PTYPE shall not be 0 or a downloadable program
type not supported on NSID = 0x4.Found PTYPE = 0x0.
Computational Program Command Set Spec v1.0 Section 4.2.1 ,Debug ID: 010000000000003b
```

Specification details – version, section

Debug ID

```
// CP Checks
NVME_CP_NSDS_RSVD,
NVME_CP_CDS_RSVD,
NVME_CP_CDS_VER_INVLD,
NVME_CP_PIND_INVLD,
NVME_CP_LOAD_PROG_PIT_RSVD,
NVME_CP_LOAD_PROG_PTYPE_RSVD,
NVME_CP_LOAD_PROG_PTYPE_INVLD,
NVME_CP_LOAD_PROG_PSIZE_EXCEED,
NVME_CP_LOAD_PROG_NUMB_INVLD,
NVME_CP_LOAD_PROG_LOFF_INVLD,
NVME_CP_SEL_RSVD,
NVME_CP_PROG_ACT_MNG_MAXACT_EXCEED,
NVME_CP_RSID_INVLD,
NVME_CP_MEM_RNG_SET_MNG_NUMR_INVLD,
NVME_CP_MEM_RNG_SET_MNG_MAXMEMRS_EXCEED,
NVME_CP_EXEC_PROG_RSID_NUMR_INVLD,
NVME_CP_EXEC_PROG_DLEN_INVLD,
NVME_CP_MEM_RNG_SET_MNG_MNSID_INVLD,
NVME_MEM_RNG_SET_MNG_MEM_RNG_LEN_INVLD,
NVME_CP_MEM_RNG_SET_MNG_MEM_RNG_OVRLP,
NVME_CP_XPCTD_STS_PIND_NOT_DOWNLDBL,
NVME_CP_XPCTD_STS_INVLD_FLD,
NVME_CP_XPCTD_STS_INVLD_PTYPE,
NVME_CP_XPCTD_STS_INVLD_PIND,
NVME_CP_XPCTD_STS_MAXPB_EXCEED,
NVME_CP_XPCTD_STS_NO_PROG,
NVME_CP_XPCTD_STS_MAXACT_EXCEED,
NVME_CP_XPCTD_STS_INVLD_RSID,
NVME_CP_XPCTD_STS_MAXMEMR_EXCEED,
NVME_CP_XPCTD_STS_MAXMEMRS_EXCEED,
NVME_CP_XPCTD_STS_PROG_NOT_ACT,
NVME_CP_XPCTD_STS_INVLD_MEM_NS,
NVME_CP_XPCTD_STS_INVLD_MEM_RNG_SET,
NVME_CP_XPCTD_STS_OVRLP_MEM_RNG,
NVME_GETLP_PTYPE_INVLD,
NVME_GETLP_PTYPE_RSVD,
NVME_DPROG_LIST_LP_DESC_VER_INVLD,
NVME_DPROG_LIST_LP_DESC_RSVD,
NVME_PROG_LIST_LP_PEOCC_RSVD,
NVME_PROG_LIST_LP_PIT_RSVD,
NVME_PROG_LIST_LP_PIT_INVLD,
NVME_PROG_LIST_LP_PID_INVLD,
NVME_PROG_LIST_LP_DESC_RSVD,
NVME_MEMRS_LIST_LP_MAXMEMRS_EXCEED,
NVME_MEMRS_LIST_LP_RSID_INVLD,
NVME_MEMRS_LIST_LP_MAXMEMR_EXCEED,
NVME_MEMRS_LIST_LP_MNSID_INVLD,
NVME_MEMRS_LIST_LP_MEM_RNG_LEN_INVLD,
NVME_MEMRS_LIST_LP_OVRLP_MEM_RNG,
```

Stimuli / Testing

➢ directed testing – creating exhaustive test plans

➢ stress testing – assessing system behaviour under high-load conditions

➢ validating error detection and reporting across the computational storage, subsystem local memory

➢ Handling concurrent operations between computational tasks and standard Hard NVMe operations

Stimuli / Testing

❑ Transport Independent Stimulus Library
- Around 800+ built-in sequences

❑ Highly configurable command structure

❑ Wide pool APIs to set/get command field

❑ Randomization of Stimulus
- corner cases and unexpected scenarios

❑ Automating command creation
- Constraints, APIs
- minimized user input for stress-testing

❑ Error Injection
- Built-in error scenario sequences, Error structure ,Callbacks

```
▼ sequence_lib/
  ► admin_cmd/
  ► base/
  ► boot_partition/
  ► compliance/
  ► cp_cmd/
  ► erroneous/
  ► fabric_cmd/
  ► initialization/
  ► io_cmd/
  ► kv_cmd/
  ► metadata/
  ► namespace/
  ► persistent/
  ► pmr/
  ► prp/
  ► queue/
  ► reset/
    shutdown/
  ► slm_cmd/
  ► zns_cmd/
```

Stimuli / Testing

❑ Transaction mode

- blocking and non-blocking
- Simultaneous or sequential simulation of computational/SLM commands along with NVM, ZNS and KV commands

DEBUG ID	BDF	SRC	RW	TYPE	SQID	CQID	REG_NAME / QENTRY	CMD / REG_DATA / STS	MISC	MISC1	CID	PS DT	PRP2 / SGL1[39:32]	PRP1 / SGL1[31:24]	NSID
--------	0100	H	W	REG	----	----	SQ1TDBL	00000005	------	----------	----	---	---------------	---------------	--------
--------	0100	H	W	MSIX	0	----	MASK_C	00000000	------	----------	----	---	---------------	---------------	--------
0100000100000023	0100	D	R	IOSQ	0001	0001	0	MEM_WR	100	--------	0000	PRP	0000000000000000	0000C14338900000	00000002
0100000100C90024	0100	D	R	IOSQ	0001	0001	1	WR	0	0000	0001	PRP	0000000000000000	000009ED71C92000	00000001
0100070310CA0025	0100	D	R	IOSQ	0001	0001	2	EXEC_PRG	0	--------	0002	PRP	0000000000000000	0000D7322CA87000	00000004
0100800D9100CFD0026	0100	D	R	IOSQ	0001	0001	3	EXEC_PRG	96	--------	0003	PRP	0000000000000000	0000832D9ACFD000	00000004
0100F0100CED0027	0100	D	R	IOSQ	0001	0001	4	EXEC_PRG	100	--------	0004	PRP	0000000000000000	000079DF83CED000	00000004
0100000100000023	0100	D	R	PRPDW	----	----	--------	--------	------	--------	----	---	---------------	---------------	--------
0100000100000023	0100	D	W	IOCQ	0001	0001	0	GEN00	00005	--------	0000	---	---------------	---------------	--------
--------	0100	D	W	INTR	----	----	--------	--------	------	--------	----	---	---------------	---------------	--------
--------	0100	H	W	MSIX	0	----	MASK_S	00000001	------	--------	----	---	---------------	---------------	--------
--------	0100	H	W	REG	----	----	CQ1HDBL	00000001	------	--------	----	---	---------------	---------------	--------
--------	0100	H	W	MSIX	0	----	MASK_C	00000000	------	--------	----	---	---------------	---------------	--------
0100000100000024	0100	D	R	PRPDW	----	----	--------	--------	------	--------	----	---	---------------	---------------	--------
0100000100000024	0100	D	W	IOCQ	0001	0001	1	GEN00	00005	--------	0001	---	---------------	---------------	--------
0100000100000025	0100	D	W	IOCQ	0001	0001	2	GEN00	00005	--------	0002	---	---------------	---------------	--------
--------	0100	D	W	INTR	----	----	--------	--------	------	--------	----	---	---------------	---------------	--------
--------	0100	H	W	MSIX	0	----	MASK_S	00000001	------	--------	----	---	---------------	---------------	--------

Coverage

➢ validate that DUT functions correctly under all possible scenarios

- functional coverage, code coverage and assertion coverage

❑ Comprehensive Coverage Plan

- All fields of Computational and SLM Admin, I/O commands
- Crosses with possible status code types
- Each cover point has a corresponding test in compliance test suite

❑ VIQ

- Reduced coverage closure time – hole analysis, heatmaps, bin distribution
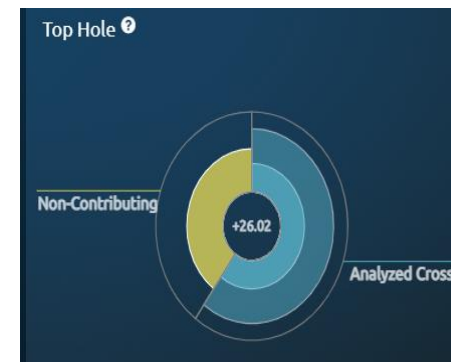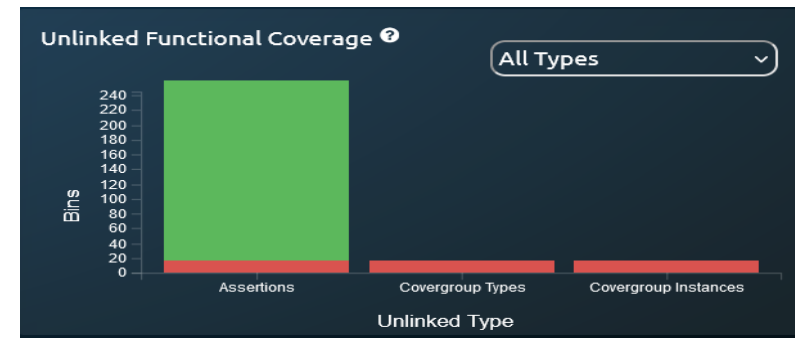- Debugging tool – failure signature detection

Coverage
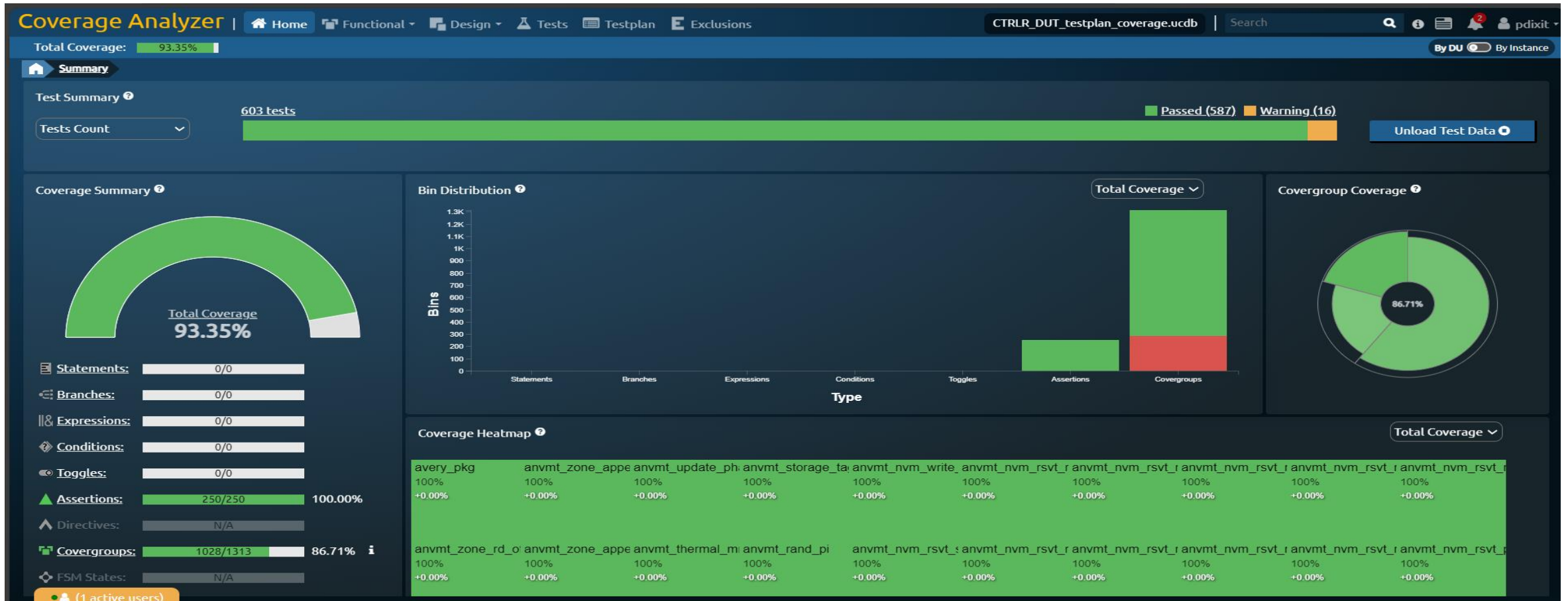
Coverage

Name	At least	Hits ↓	Bin type
auto[NVME_4_6_2]	1	126813	NORMAL_BIN
auto[NVME_4_6_1]	1	84542	NORMAL_BIN
auto[NVME_1_4_4]	1	44665	NORMAL_BIN
auto[NVME_5_3]	1	44653	NORMAL_BIN
auto[NVME_4_6_1_2_1]	1	42275	NORMAL_BIN
auto[NVME_4_6_1_2_21]	1	42271	NORMAL_BIN
auto[NVME_4_6_1_2_16]	1	42271	NORMAL_BIN
auto[NVM20_3_5_3_fig1...	1	42271	NORMAL_BIN

GAP ANALYSIS

Top Uncovered Sections ❓

#	Name	Coverage %
11	anvmt_14_cmb_CQMMS_err	0.00%
63	anvmt_UNH_admin_cmd_1_1_11	0.00%
64	anvmt_UNH_admin_cmd_1_1_12	0.00%
66	anvmt_UNH_admin_cmd_1_1_14	0.00%
445	anvmt_boot_partition_full	0.00%

Unlinked Functional Coverage ❓

All Types

Top Hole ❓

Coverage

Debugging

➢ identifying, isolating, and fixing bugs in large design

➢ hard to trace errors back to their source

❑ Transaction Logger
❑ Beat Logger

Debugging

☐ Transaction Logger

- Debug ID associated with each command
- Print frequently required characteristics of commands

Various decoded fields
Command Specific : -
- PID, PIT, PTYPE, PSIZE … ,etc
Generic :-
-    NSID, CID, PRP …,etc

Name	Type	Size	Value
admin_txn	nvme_txn	-	@22376
Txn Type	string	12	NVME_TXN_SQE
Debug ID	integral	64	'h10000000000001f
Opcode	string	17	NVME_AD_LOAD_PROG
Time	time	64	0
BDF	integral	16	'h100
Command ID	integral	16	'h5
NSID	integral	32	'h4
Program Index	integral	16	'd1
Program Type	integral	8	'hc0
Select	integral	4	'h0
Program Identifier Type	integral	3	'd0
Program Size	integral	32	'd1000
Program Identifier	integral	64	'h0
Number of Bytes	integral	32	'd0
Load Offset	integral	32	'd0
PRP Count	integral	32	'd0
PRP1	integral	64	'h58ce4ce79000

Name	Type	Size	Value
admin_txn	nvme_txn	-	@22376
Txn Type	string	12	NVME_TXN_CQE
Debug ID	integral	64	'h10000000000001f
Opcode	string	17	NVME_AD_LOAD_PROG
Time	time	64	25327925000
BDF	integral	16	'h100
Command ID	integral	16	'h5
NSID	integral	32	'h4
Program Index	integral	16	'd1
Program Type	integral	8	'hc0
Select	integral	4	'h0
Program Identifier Type	integral	3	'd0
Program Size	integral	32	'd1000
Program Identifier	integral	64	'h0
Number of Bytes	integral	32	'd0
Load Offset	integral	32	'd0
PRP1	integral	64	'h58ce4ce79000
Completion Status (DNR)	integral	1	'h0
Completion Status (More)	integral	1	'h0
Completion Status (SCT) :	string	23	Generic Command Status
Completion Status (SC)	string	14	NVME_SUCC_CMPL

Debugging

❑ Beat Logger

- Correlate all transport transactions under single nvme_txn
- Highlights any unwanted address access

Performance Assessment

➢ assessing the performance of NVMe operations and Computational/SLM operations when handled concurrently

❑ **Performance Logger**

- Latency
- Throughput
- IOPS

Performance Assessment

Thank You !

Visit us at booth #(Siemens)
for more information