White Paper

VIAVI Solutions

PCIe 6.0 Flit Mode Error Injection and Recovery Analysis

TABLE OF CONTENTS

Abstract 3
1. Introduction 4
2. PCIe Protocol Stack and Flit Mode Architecture 5
    2.1 Layered Error Detection 5
    2.2 The Flit as the Primary Transmission Unit 5
    2.3 Processing Order 6
3. Error Detection and Recovery in Flit Mode 7
    3.1 Multi-Layer Integrity Model 7
    3.2 Selective NAK vs. Standard NAK 7
    3.3 Explicit vs. Implicit Sequence Numbering 8
4. Exerciser and Analyzer Tooling 8
    4.1 The Case for External Error Injection 8
    4.2 Dual-Function Test Platforms 9
    4.3 Exercise-and-Analyze Workflow 10
    4.4 Error Injection at Various Protocol Layers 10
5. PHY Logical Layer: LTSSM Error Injection 11
    5.1 LTSSM Overview 11
    5.2 LTSSM Test Cases 12
    5.3 Training Sequence Ordered Sets at 64 GT/s 12
    5.4 Test Case 1: Delay TS0/TS1 Transition (Phase 0 Timeout) 13
6. Data Link Layer: DLP Error Injection in Flit Mode 15
    6.1 DLP Field Structure 15
    6.2 DLP Injection Capabilities 15
    6.3 Pre-FEC vs. Post-FEC Injection 16
    6.4 Test Case: Explicit Sequence Number Error (Pre-FEC) 17
    6.5 Test Case: Sequence Number Error (Post-FEC) 19
7. Transaction Layer: TLP Error Injection 19
    7.1 TLP-Level Error Categories 19
    7.2 Data Poisoning in Flit Mode 20
    7.3 Test Case: Poisoned TLP via EP Bit 20
    7.4 Test Case: Poisoned TLP via Flit_status Field 22
8. Conclusion and Key Takeaways 22
    8.1 Error Injection is a Core Validation Discipline 22
    8.2 Flit Mode Demands a New Test Strategy 22
    8.3 Deterministic Injection + Integrated Capture 23
9. About the Author and Tools 23

Abstract

PCIe® 6.0 introduces Flit (Flow Control Unit) mode as the fundamental unit of transfer at 64 GT/s. The fixed size 256-byte Flit replaces the variable-length TLP-centric model of prior generations. The shift to PAM4 signaling, interleaved Forward Error Correction (FEC), Flit-level CRC, Selective NAK, and replay mechanisms create both new validation challenges and new opportunities for systematic fault injection testing.

This white paper provides a comprehensive treatment of error injection and recovery analysis in PCIe 6.0 Flit mode. It covers the theoretical underpinnings of the Flit integrity model, the role of protocol exerciser and analyzer tools in controlled fault injection, and a series of concrete test cases spanning the PHY Logical, Data Link, and Transaction layers. Each test case examines the injection mechanism, the expected DUT behavior per specification, and the actual protocol trace evidence of compliant recovery. The goal is to equip PCIe 6.0 validation engineers with a structured approach to corner-case testing that goes well beyond nominal path verification.

1. Introduction

PCI Express (PCIe) has been the dominant high-speed serial interconnect standard for computing platforms for over two decades. Each generation has roughly doubled bandwidth while maintaining backward compatibility. PCIe 6.0 reaches 64 GT/s per lane through two major architectural shifts: PAM4 (Pulse Amplitude Modulation, 4 levels) signaling and Flit mode transmission.

PAM4 delivers two bits per symbol, enabling the 64 GT/s data rate without requiring a corresponding doubling of the electrical bandwidth. However, PAM4 is inherently more susceptible to noise than the NRZ (Non-Return-to-Zero) signaling used in PCIe 5.0 and earlier. At 64 GT/s, the expected Bit Error Rate (BER) is significantly higher than previous generations, and errors tend to occur in bursts within a lane. This reality drove the introduction of Flit mode.

In Flit mode, all data is transmitted in 256-byte fixed-length units called Flits. Each Flit carries a payload of TLP and DLLP bytes protected by a multi-layer integrity scheme: interleaved FEC (Forward Error Correction) for single-burst error correction, a strong Flit-level CRC for residual error detection, and a Flit-level replay mechanism triggered by NAK signals. These mechanisms replace or complement the TLP-level reliability model of Non-Flit Mode (NFM).

The introduction of Flit mode has significant implications for validation. Error scenarios that did not exist in prior generations — FEC-correctable versus FEC-uncorrectable faults, Selective NAK versus Standard NAK behavior, implicit versus explicit sequence numbering, Flit-level poisoning mechanisms — must all be explicitly verified. A validation program built around PCIe 4.0 or 5.0 assumptions is insufficient for 6.0.

This white paper is organized around the primary error domains in PCIe 6.0 Flit mode: the PHY Logical layer (LTSSM and ordered set handling), the Data Link layer (DLP fields, sequence numbers, FEC, CRC, and NAK), and the Transaction layer (TLP semantics, ECRC, and data poisoning). For each domain, we present the relevant architecture, describe the exerciser-based injection methodology, and walk through specific test cases with observed results.

2. PCIe Protocol Stack and Flit Mode Architecture

2.1 Layered Error Detection

PCIe is a layered architecture – Physical (Electrical and Logical), Data Link, and Transaction — each contributing specific error detection and handling capabilities. Understanding the role of each layer is prerequisite to understanding where and how errors can be injected.

Layer	Primary Function	NFM Packets	Flit Mode
Physical (Logical)	Link training, equalization	Ordered Sets (TS1, TS2, etc.)	TS0 added; PAM4 EQ phases
Data Link	Sequence numbering, flow control, replay	DLLPs + TLP sequence numbers	Flit-level CRC, FEC, Selective NAK
Transaction	End-to-end TLP semantics, ECRC, completion	TLPs (variable length)	TLPs packed into 256-byte Flits

2.2 The Flit as the Primary Transmission Unit

In Flit mode, Fixed size 256-byte Flit is the primary unit of transmission with fixed TLP and DLP Byte placements. Flit is partitioned into four regions:

- TLP region (236 bytes): carries TLPs packed end-to-end. Multiple small TLPs may reside in a single Flit; a large TLP may span multiple Flits.
- DLP region (6 bytes): carries Data Link Packet fields including Flit usage type, sequence numbers, ACK/NAK commands, DLLP payloads, and the Prior Flit bit.
- CRC region (6 bytes): a 48-bit CRC computed over the TLP and DLP bytes appended
- ECC region (8 bytes): Forward Error Correction code computed over all TLP + DLP + CRC bytes

Table 4-10 Flit Layout in a x16 Link

Description	Lane															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TLP Bytes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
TLP, DLP	224	225	226	227	228	229	230	231	232	233	234	235	DLP 0	DLP 1	DLP 2	DLP 3
DLP, CRC, ECC	DLP 4	DLP 5	CRC 0	CRC 1	CRC 2	CRC 3	CRC 4	CRC 5	CRC 6	CRC 7	ECC1 [0]	ECC2 [0]	ECC0 [0]	ECC1 [1]	ECC2 [1]	ECC0 [1]

256-Byte Flit Layout - Fixed TLP and DLP placements

The FEC code is 3-way interleaved across the Flit bytes, such that a burst error of length ≤16 bits in a single lane impacts at most one byte in each FEC group. Each FEC group can correct a single byte error. This design provides robustness against the localized burst errors characteristic of PAM4 signaling.

2.3 Processing Order

On the transmit side, the protocol generates CRC over TLP+DLP bytes and then applies FEC encoding. On the receive side, the order is reversed:

- If FEC correction succeeds and CRC passes: the Flit is processed normally.
- If FEC correction succeeds but CRC fails: the error was beyond FEC correction capability. A NAK is issued and Flit replay is initiated.
- If FEC correction cannot be determined: the CRC check is the arbiter. A NAK triggers replay.

Error Handling Philosophy: Correct what you can (FEC), detect what you cannot correct (CRC), and replay what you detect (NAK). That hierarchy governs all the test cases examined in this white paper.

3. Error Detection and Recovery in Flit Mode

3.1 Multi-Layer Integrity Model

PCIe 6.0 Flit mode provides multiple overlapping layers of error protection and detection:

- Flit-level CRC (48 bits): protects the TLP and DLP payload bytes. CRC failure after FEC decoding triggers Flit-level replay.
- Forward Error Correction (FEC / ECC): 8 bytes of ECC per 256-byte Flit
- Sequence Numbers and ACK/NAK: the Flit-level replay mechanism. Explicit sequence numbers synchronize transmitter and receiver at link up, recovery, and replay events. Implicit sequence numbers are inferred during normal operation.
- ECRC (End-to-End CRC): optional TLP-level CRC for true end-to-end data integrity, unchanged from prior generations.
- Data Poisoning: forward-and-flag mechanism for corrupted TLP payloads — the EP bit in the TLP header, and/or the Flit_status field in the Flit header.
- Flow Control Protocol Errors: violations in FC credit signaling are flagged and reported

3.2 Selective NAK vs. Standard NAK

One of the most significant enhancements in PCIe 6.0 Flit mode is Selective NAK (Selective Negative Acknowledgment). Understanding the distinction from Standard NAK is critical for both testing and debugging.

Standard NAK (inherited from prior generations): when a corrupted Flit is detected, the receiver issues a NAK requesting retransmission starting from the first bad Flit and including all subsequently transmitted Flits. Even Flits that were transmitted after the bad one — and were themselves error-free — must be retransmitted. Standard NAK is mandatory for severe error scenarios and when the transmitter does not support selective retransmission.

Selective NAK (new in PCIe 6.0): if a corrupted Flit is detected and the following Flit's Prior Flit Payload bit indicates that the failed Flit was a Payload Flit, the receiver can request retransmission of only that specific Flit. Subsequent Flits are held in the receiver's retry buffer and processed once the replayed Flit is successfully received. This reduces unnecessary retransmission overhead and latency.

The Prior Flit Payload bit in the DLP region of each Flit is the enabling mechanism. If the bad Flit was a NOP (no payload), the NAK is withdrawn entirely — there is no data to replay. This NAK withdrawal behavior is a subtle but important protocol optimization that must be explicitly verified.

3.3 Explicit vs. Implicit Sequence Numbering

In Non-Flit mode, every TLP carries an explicit sequence number in its Data Link Layer header. Flit mode takes a different approach to minimize per-Flit overhead.

Explicit Flit Sequence Numbering is used after specific protocol events: link entry to L0, recovery from an error, and replay initiation. The sequence number is carried explicitly in the DLP bytes of the Flit to resynchronize transmitter and receiver state. Explicit sequence numbers provide absolute reference points in the trace that make recovery events clearly visible.

Implicit Flit Sequence Numbering is used during normal operation. The receiver maintains an internal counter and infers sequence numbers from the protocol rules governing Flit ordering. The sequence number is not carried on the wire for every Flit. This maximizes efficiency but makes raw trace analysis more complex — the analyst cannot simply read sequence numbers from each Flit. VIAVI's Xgig protocol analyzers address this by reconstructing implicit sequence numbers from the captured trace and presenting them as a derived column in the trace viewer.

4. Exerciser and Analyzer Tooling

4.1 The Case for External Error Injection

PCIe 6.0 introduced optional built-in error injection features in the specification. These native injection capabilities are valuable when a device under test (DUT) implements them, because they provide in-situ control over the type, location, and frequency of injected errors.

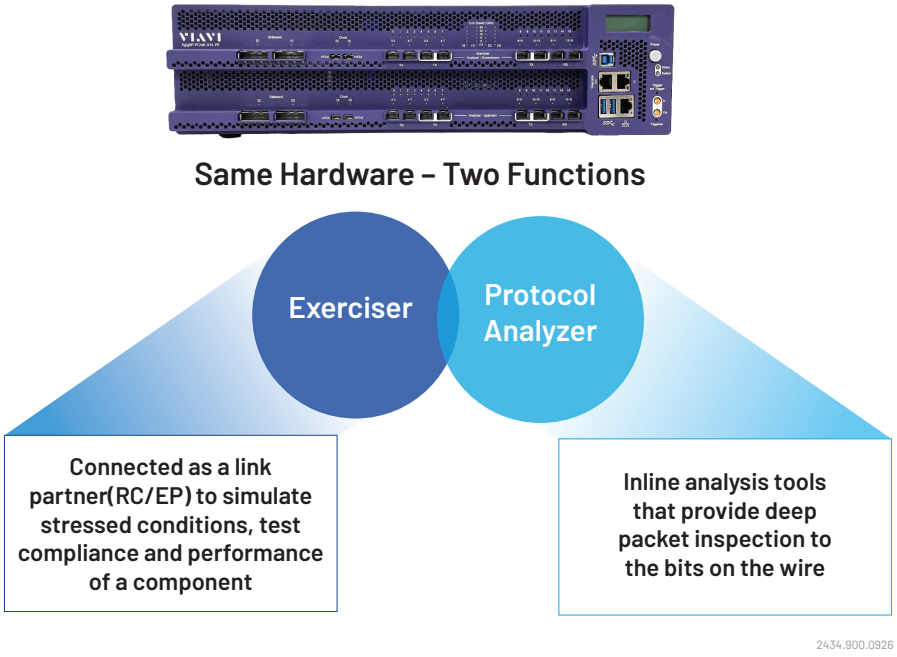
However, external protocol exerciser tools offer complementary — and in many cases more powerful — capabilities. The key advantages of exerciser-based injection are:

- Independence from DUT capabilities
- Deterministic and repeatable injection
- Integrated trace capture
- Coverage of corner cases not reachable through nominal traffic

4.2 Dual-Function Test Platforms

VIAVI's PCIe 6.0 Xgig platforms are capable of operating as both a Protocol Exerciser and a Protocol Analyzer simultaneously on the same hardware.

The tight synchronization between the exerciser and analyzer functions is the key enabler for the injection-and-observation workflow described in this white paper. The trace captures shown in subsequent sections were all produced by this type of integrated platform.



Dual-function test platform — exerciser and analyzer on a single PCIe 6.0 card

4.3 Exercise-and-Analyze Workflow

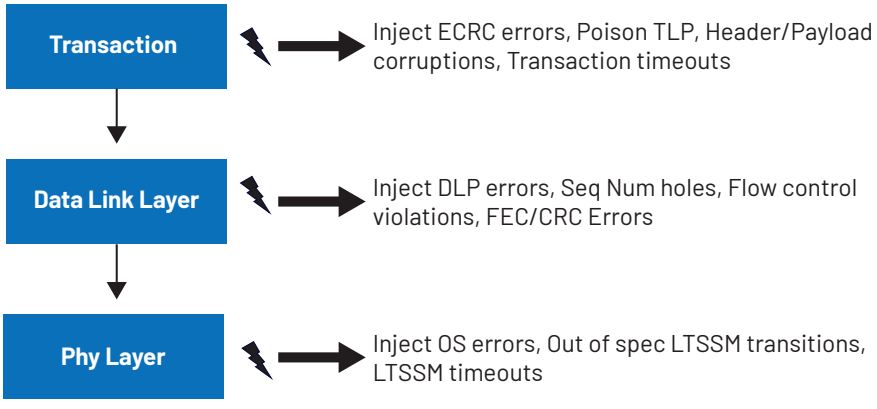
The standard workflow for exerciser-based error injection testing follows four phases:

- **Pre-test configuration:** define trigger criteria (what condition causes the injection to fire) and program the error modification parameters (what exactly to change when the trigger fires). Optionally capture a baseline trace of normal behavior for comparison.
- **Traffic generation:** configure the exerciser to bring up the link via LTSSM, establish protocol communication with the DUT, and generate the traffic pattern required to reach the trigger condition.
- **Synchronized capture:** the analyzer captures the full traffic stream including the injection event and the DUT's response. The trigger event is time-stamped and cross-referenced in the trace.
- **Post-test analysis:** verify that the error was injected as configured, that the DUT detected the error through the expected mechanism, and that the DUT recovery behavior matches the PCIe 6.0 specification requirements.

4.4 Error Injection at Various Protocol Layers

Every Error detection mechanism and every shall in the spec creates an Error Injection test case. Many areas of failure are listed in the figure below. The goal of exerciser-based testing is not to produce failures — it is to produce controlled, verifiable, specification-compliant recoveries. A DUT that hangs, crashes, or recovers through an incorrect mechanism is a test failure even if the link eventually returns to L0. The next three chapters will review test cases by the Protocol layer.

Error Injection at PCIe Protocol Layers



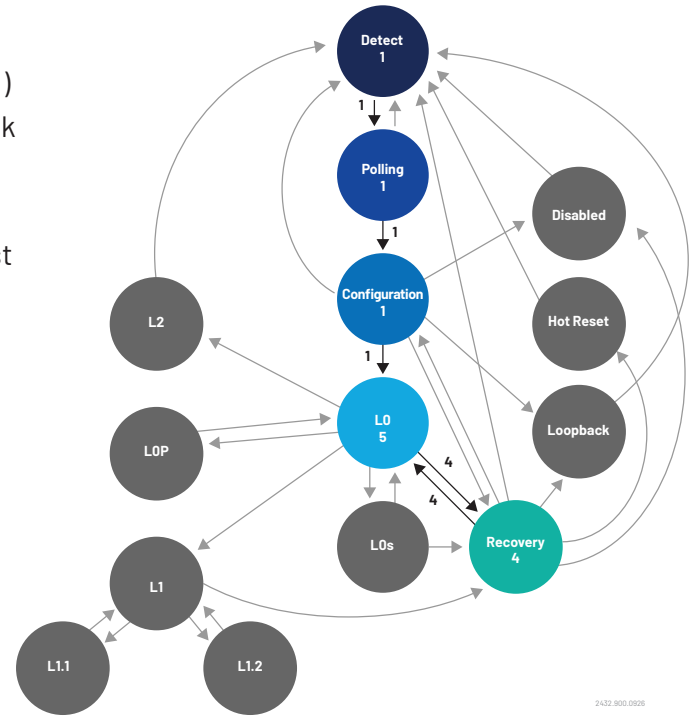
** PCIe 6.0 spec introduced optional features, Error Injection features, for intentional injection of errors. The idea is to provide full control over the type, location, and frequency of errors injected into the PCIe events. Exercisers provide an alternate way to test this and other advanced test cases.

2433.900.0926

5. PHY Logical Layer: LTSSM Error Injection

5.1 LTSSM Overview

The Link Training and Status State Machine (LTSSM) governs the entire PCIe link bring-up sequence. Link training begins when power and clock are available to both components; the PHYs step through multiple states, and the link enters L0 at the highest negotiated speed only after all required transitions are successfully completed.



LTSSM – Link up to 64GT/s

For PCIe 6.0, the highest negotiated speed is 64 GT/s. Reaching 64 GT/s requires completing all equalization phases at the 32 GT/s precursor speed and then completing the full PAM4-specific equalization process at 64 GT/s. This equalization process is more complex and failure-prone than anything in prior PCIe generations.

Early LTSSM states — Detect, Polling, and Configuration — establish the basic link properties and then the link moves to Link equalization which is one of the most complex and failure-prone aspect of PCIe 6.0 bring-up. During EQ phase the link proceeds through four phases (Phase 0 through Phase 3). Both sides of the link iteratively tune their transmitter coefficients and receiver settings through an exchange of Training Sequence Ordered Sets (TS0 and TS1 at 64 GT/s). The EQ process at 64 GT/s with PAM4 signaling is significantly more demanding than at 32 GT/s NRZ. EQ failures can be caused by timing violations, failures in the dynamic EQ tuning algorithm, or insufficient signal quality.

LTSSM is a critical validation target for two reasons. First, errors in LTSSM prevent the link from reaching L0 at the negotiated speed, or width and cause it to fall back to a lower speed — fundamental operational failures. Second, the DUT must respond correctly to non-compliant LTSSM sequences from a misbehaving link partner. Exerciser-based injection allows systematic testing of both dimensions,

5.2 LTSSM Test Cases

An exerciser enables testing of non-standard lane configurations, injection of delayed or malformed Training Sequences to present the DUT with edge-case negotiation scenarios. These scenarios surface boundary test cases that nominal bring-up testing misses. Some LTSSM test cases are listed below.

- LTSSM Protocol Violations
  - TS0 / TS1 / TS2 Ordered Set modification
  - SDS corruption
  - EI entry/exit — EIOS/EIEOS test
  - Link width negotiation tests
  - OS receiver criteria tests
- Timing Tests
  - Force LTSSM timeouts
  - Force speed change failures
  - Selective lane failures
  - Dynamic EQ failures
- Transition Tests
  - Create out-of-spec transitions
  - TS0/TS1 transition failure
  - L0 transition
  - Low-power state entry/exit failures
  - Reset scenarios

5.3 Training Sequence Ordered Sets at 64 GT/s

PCIe 6.0 introduces the TS0 Ordered Set as a new 16-byte OS used at 64 GT/s. TS0 enables the receiver to transition from NRZ-like to PAM4 waveform recognition and performs the initial handshake when a port enters 64 GT/s operation. The transition from TS0 to TS1 is governed by the EC (Equalization Complete) bit in the OS:

- Phase 0 entry: both Downstream and Upstream ports transmit TS0 with EC = 00b.
- DS port transition: when the DS port's receiver achieves target BER (1E-4), it transitions to EC = 01b and waits for the US port to do the same.
- Phase 2 transition: when the US port sends EC = 01b, DS port transitions to Phase 2.
- Phase 3 transition: when US port sends EC = 11b, DS port expects the NRZ-to-PAM4 transition.

All OS are protected through replication at 64 GT/s and above. Understanding the exact byte layout of TS0 and TS1 is critical for injection testing: the exerciser must modify the correct bytes while maintaining parity consistency across both 8-byte chunks of the OS.

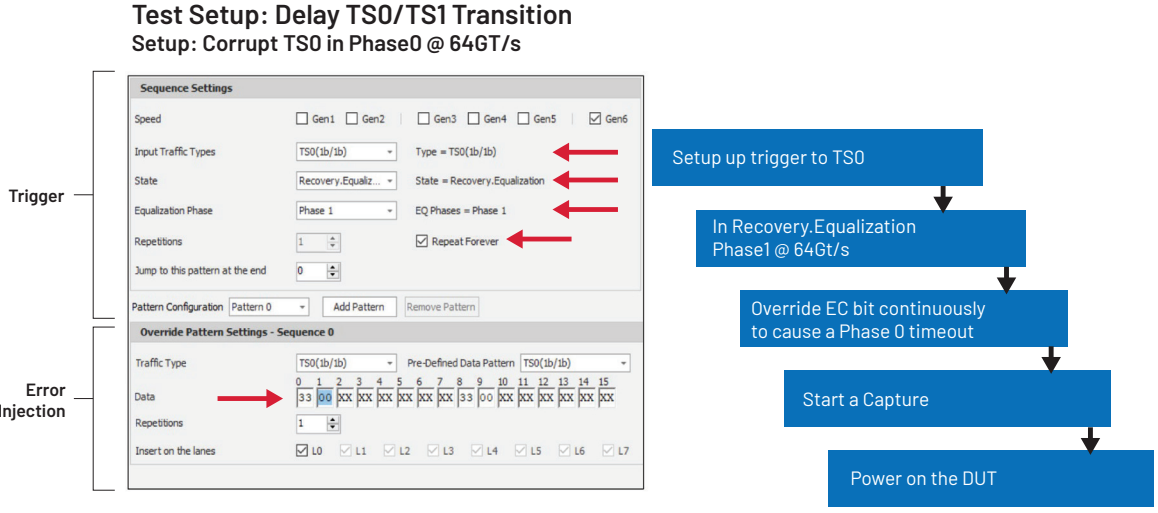
5.4 Test Case 1: Delay TS0/TS1 Transition (Phase 0 Timeout)

Objective

Verify that when the EP is prevented from seeing the Phase 0-to-Phase 1 EC bit transition, it correctly times out after the specification-defined interval and falls back from 64 GT/s to 32 GT/s.

Injection Setup

The exerciser is configured to act as the Downstream port. The trigger condition fires on TS0 Ordered Sets during Phase 1 at 64 GT/s. The error modification clears the EC bit in the TS0 OS (maintaining it at EC = 00b rather than transitioning to EC = 01b). The modification is applied consistently to bytes in both 8-byte halves of the TS0 OS to maintain parity correctness.



Injection setup — TS0 EC bit held at 00b to delay Phase 0 → Phase 1 transition

Parity must be maintained across both 8-byte chunks. Breaking parity triggers a different error path (parity error handler) and invalidates the timeout test.

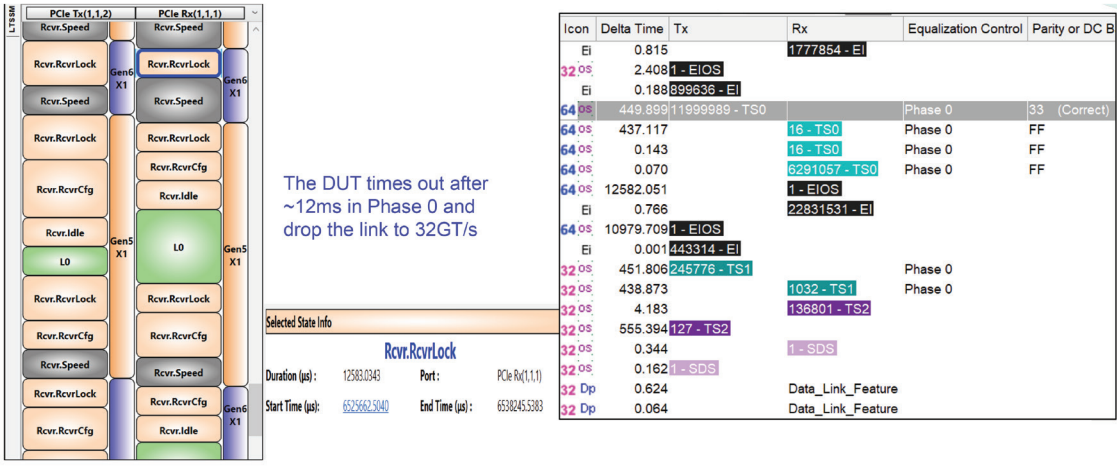
Expected Behavior

The EP (acting as US port) continues to receive valid-looking Phase 0 TS0 OS and does not see the Phase 1 transition it is waiting for. After approximately 12 milliseconds — the timeout interval specified in PCIe 6.0 — the EP drops the link from 64 GT/s back to 32 GT/s to recover.

Observed Result

The captured trace shows the modified TS0 OS with EC bits held at 00b throughout Phase 0. The EP remains waiting for EC = 01b from the DS port. At approximately 12 milliseconds, the EP initiates a speed change back to 32 GT/s. No hang, no crash, no indefinite stall at 64 GT/s. The DUT's LTSSM timeout behavior is compliant.

Test Result: Delay TS0/TS1 Transition
Result: EP waiting in TS0, Phase0 (EC=0) @ 64GT/s



Observed result — EP falls back from 64 GT/s to 32 GT/s after LTSSM timeout

6. Data Link Layer: DLP Error Injection in Flit Mode

6.1 DLP Field Structure

The six DLP bytes in each 256-byte Flit carry the Data Link layer control information for that Flit. The fields include:

- **Flit Usage:** indicates whether the Flit is a NOP Flit (no TLP payload) or a Payload Flit.
- **Prior Flit Payload bit:** indicates whether the immediately preceding Flit carried a TLP payload. This bit is the key enabler of Selective NAK and NAK withdrawal logic.
- **DLLP Type:** distinguishes DLLP, Optimized Flow Control, and Flit Marker types.
- **Replay Command:** specifies the replay action associated with this Flit — Flit Ack, Selective NAK, Standard NAK, or Explicit Sequence Number.
- **Sequence Number:** carries either an Ack Number (acknowledging received Flits) or a Sequence Number (for explicit sequence synchronization).
- **DLLP Payload:** carries the content of the DLLP or Flit Marker.

These fields collectively define the Flit-level reliability protocol. A DLP injection test targets one or more of these fields to test specific aspects of the receiver's DLP processing logic.

6.2 DLP Injection Capabilities

The exerciser supports flexible trigger and injection configuration at the DLP level. Trigger conditions can be specified on any combination of Flit type (NOP or Payload), DLLP Type, Replay Command type, and sequence number value. This allows injection to be targeted at very specific protocol exchanges.

Once triggered, the exerciser can inject the following categories of DLP errors:

- **Sequence number errors:** skipped sequence numbers, duplicate sequence numbers, or explicit sequence number mismatches.
- **ACK/NAK corruption:** invalid acknowledgment or negative acknowledgment fields.
- Prior Flit bit manipulation: flipping the Prior Flit Payload bit to test NAK withdrawal behavior.
- **Flow control violations:** corrupted FC credit updates to test FC error handling.
- **CRC errors:** direct CRC field corruption to force replay without FEC involvement.
- **Pre-FEC injection:** error introduced before FEC encoding, so the error is present in the FEC-encoded stream. Tests the replay path.
- **Post-FEC injection:** error introduced after FEC encoding, directly into the encoded bits. Tests the FEC correction path.

6.3 Pre-FEC vs. Post-FEC Injection

The distinction between pre-FEC and post-FEC injection is one of the most important concepts in Flit mode error testing. Using the wrong injection mode can lead an engineer to test the wrong behavior entirely.

Pre-FEC injection: the error is introduced into the Flit payload before FEC encoding is applied. FEC encoding is then computed over the (already corrupted) data. On the receive side, the FEC decoder will not encounter a ECC error, instead the logic that needs to be tested (Such as ACK/NAK handling, Seq Num logic etc.) will trigger an error and may result in a CRC failure and a NAK. This tests the replay path.

Post-FEC injection: the error is introduced directly into the FEC-encoded bit stream, after encoding. The receiver's FEC decoder attempts to correct the introduced errors. If the error pattern is within FEC's correction capability (e.g., a single byte error within one FEC group), FEC corrects it, the CRC passes, and the Flit is processed normally — this tests the FEC correction path. If the error pattern exceeds FEC's capability, the CRC fails and replay is triggered — this tests the FEC-uncorrectable path.

Design principle: use pre-FEC injection to test NAK and replay handling. Use post-FEC injection to test FEC correction capability. These are distinct tests for distinct parts of the error handling architecture.

6.4 Test Case: Explicit Sequence Number Error (Pre-FEC)

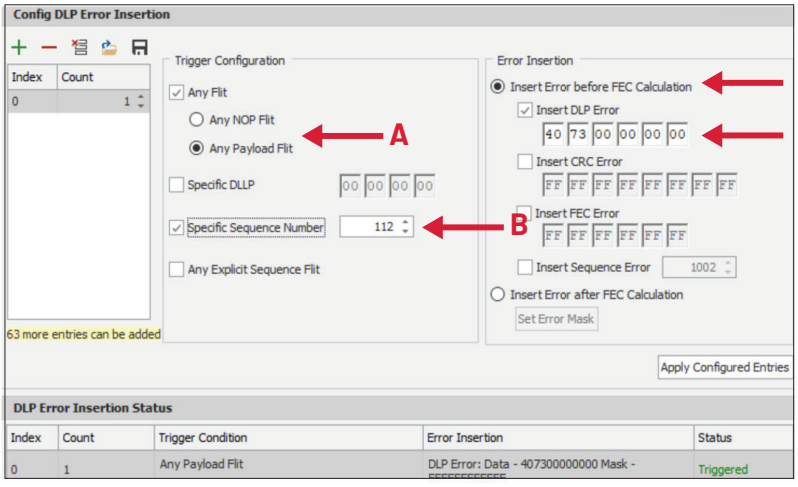
Pre-Test Baseline

Before injection, the exerciser status is checked to confirm the current transmit sequence number (in this case is 0x6F, or 111 decimal). The next Payload Flit to be transmitted will carry sequence number 0x70 (112 decimal).

Injection Setup

Trigger condition: any Payload Flit with sequence number 0x70. Error injection: pre-FEC modification of the Explicit bit and one bit in the Sequence Number field in the DLP bytes. The corruption creates an Sequence number hole by modifying the sequence number to 0x73. The receiver is expected to detect a mismatch, triggering a NAK and replay.

Test Case - Inject Explicit SeqNum Error
Test Setup: Trigger on a Payload Flit and Modify the SeqNum Pre-FEC



Trigger Criteria
A and B

Any Payload Flit with SeqNum 0x70 = 112

Error Insertion
C and D

Insert a DLP error Pre-FEC, that modifies the explicit bit and one bit in the SeqNum field

Pre-FEC explicit sequence number injection setup — corrupting Explicit bit and SeqNum field

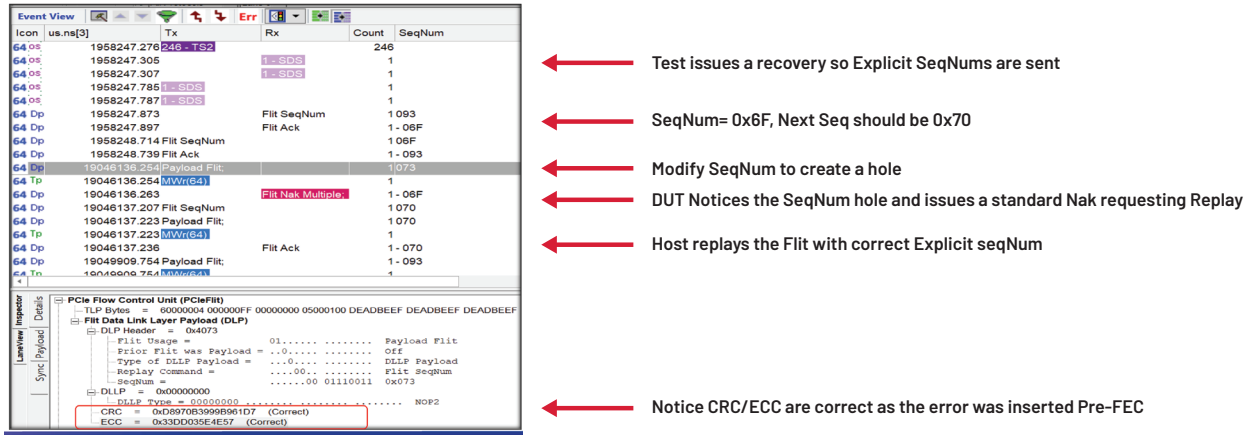
Traffic Generation

The exerciser transmits a Payload Flit containing a Memory Write TLP addressed to the DUT. This Flit carries sequence number 0x70 and thus satisfies the trigger condition. The injected DLP errors are applied pre-FEC and the analyzer captures the full exchange. Note: Issue a recovery before injecting the error so explicit sequence numbers show in the trace.

Observed Result

The trace shows the corrupted Flit arriving at the host. The host detects the invalid explicit sequence number — the combination of the corrupted Explicit bit and the modified sequence number value does not match the expected sequence state — and issues a NAK. The host then replays the Flit with the correct explicit sequence number. The replay carries the original Memory Write TLP payload with corrected DLP fields and the transaction completes successfully. DUT sequence number validation logic is confirmed working.

Recovery: Capture and Analyze the Trace
Result – Inject Explicit SeqNum Error



Observed result — host issues NAK and replays Flit with corrected explicit sequence number

6.5 Test Case: Sequence Number Error (Post-FEC)

Injection Setup

Trigger condition: any Payload Flit. Error injection: post-FEC modification of the Explicit bit and one bit in the Sequence Number field. The same fields as the pre-FEC test case, but now the modification is applied after FEC encoding.

Observed Result

The FEC decoder encounters the post-FEC errors and is unable to correct them (the modified bit pattern exceeds the single-byte per FEC group correction capacity). The CRC check fails after the FEC attempt. The DUT issues a NAK and replay, matching the pre-FEC test case in outcome but through the FEC-uncorrectable path.

Together, the pre-FEC and post-FEC test cases provide coverage of both the FEC correction path and the FEC-uncorrectable replay path.

7. Transaction Layer: TLP Error Injection

7.1 TLP-Level Error Categories

At the Transaction Layer, the exerciser can inject a broad range of TLP-level errors. The key categories and their validation significance are:

- **Malformed TLPs:** incorrect type, format, or length fields. The receiver's TLP parser must detect the malformation and respond with appropriate error signaling rather than processing the TLP.
- **Misaligned address fields:** for Memory TLPs, an address that does not meet alignment requirements. Expected response: Malformed TLP error.
- **ECRC errors:** corruption of the End-to-End CRC to test end-to-end data integrity checking.
- **Poisoned TLPs:** Poison TLPs with the EP (Error Pointer) or via the Flit-level Flit_status field.
- **Completion timeouts:** missing or delayed completion responses to Non-Posted TLPs. Tests the DUT's timeout and retry logic.
- **Unsupported Request (UR) and Completer Abort (CA):** injection of transactions that should produce UR or CA completions.
- **Incorrect transaction tag usage:** duplicate tags or out-of-order tag management.
- **TLP ordering rule violations:** transactions that violate the PCIe ordering rules, verifying the DUT's ordering enforcement.

For each category, the PCIe specification defines the expected DUT response.
The analyzer trace provides ground truth of what the DUT actually did, enabling direct compliance verification.

7.2 Data Poisoning in Flit Mode

Data poisoning is a protocol mechanism that deserves particular attention in PCIe 6.0 because Flit mode introduces a second poisoning signaling path alongside the traditional TLP header EP bit.

The purpose of data poisoning is error containment with explicit flagging. Rather than silently forwarding corrupt data (which is the most dangerous failure mode) or dropping the corrupted packet (which may cause timeouts without locating the error source), PCIe allows a component to forward a TLP while explicitly marking the payload as corrupted. The receiving component sees the poison indication and knows not to use the data. The hardware can pinpoint exactly where the original error occurred without halting the link.

PCIe 6.0 Flit mode supports two poisoning mechanisms:

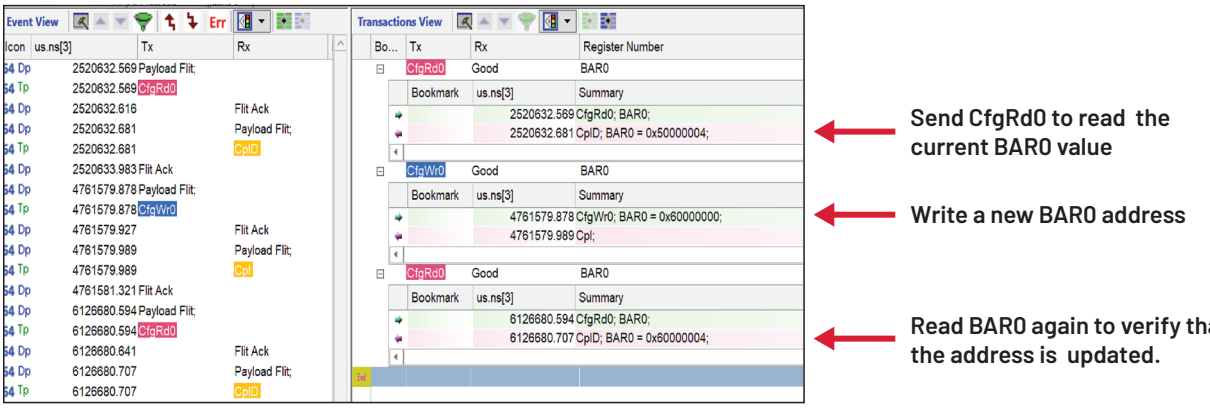
- **EP bit in the TLP header:** the traditional mechanism. Directly marks the TLP payload as poisoned.
- **Flit_status field in the Flit header:** a Flit mode-specific mechanism. The Physical Layer Logical Block sets the Flit_status field to indicate that the last TLP ending in the current Flit is poisoned.

Support for TLP poisoning is optional in PCIe 6.0. A component may implement the EP bit mechanism only, the Flit_status mechanism only, both, or neither. The validation test plan must target whichever mechanisms the DUT is specified to support.

7.3 Test Case: Poisoned TLP via EP Bit

Baseline Verification

Before injecting a poisoned TLP, a baseline is established to confirm that the DUT correctly processes normal Configuration Write transactions. A CfgRd0 reads the current BAR0 value. A normal (non-poisoned) CfgWr0 writes a new BAR0 address. A second CfgRd0 confirms the write was accepted. This establishes that the DUT's Config Write path is operational.



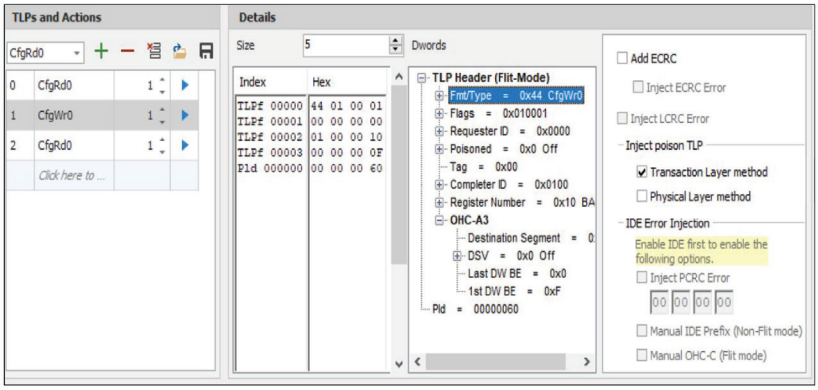
Baseline, DUT allows and accepts writing to the BAR0 register

Injection

The exerciser sends a CfgWr0 attempting to write a different BAR0 address, but with the EP bit set in the TLP header. Everything else about the TLP is well-formed. This is followed by a CfgRd0 to verify whether BAR0 was actually updated.

Inject Poison TLP – EP Bit

Test Criteria: Send CfgRd0, poisoned CfgWr0 TLP, then CfgRd0 again



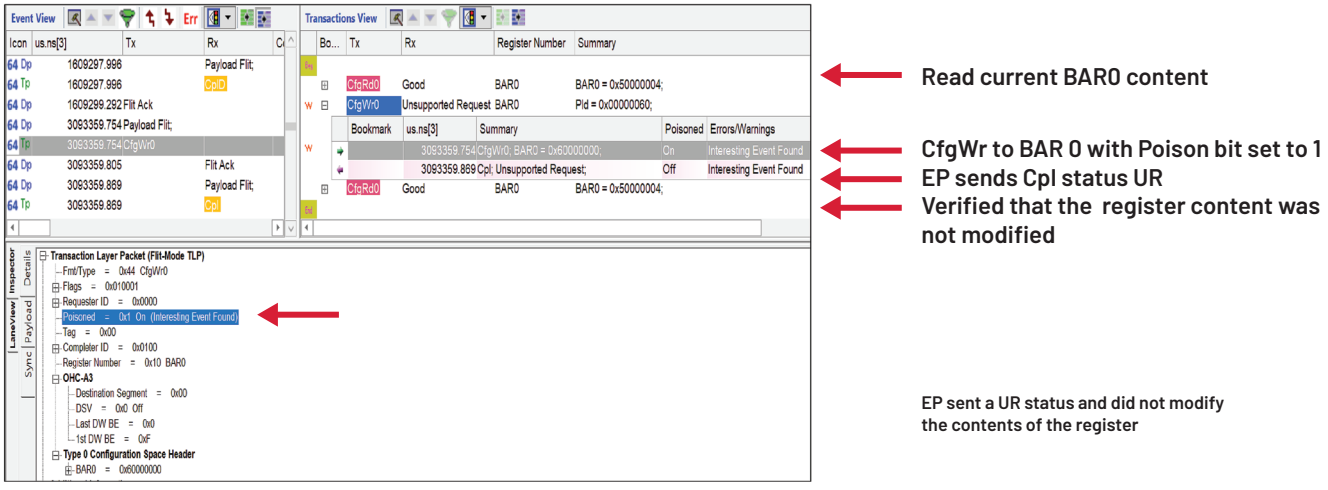
Poisoned TLP injection setup — EP bit set in TLP header on CfgWr0

Expected and Observed Result

The EP correctly detects the poisoned TLP — indicated by the EP bit in the TLP header — and responds with UR (Unsupported Request) completion status. The BAR0 register is not modified. The subsequent CfgRd0 confirms that BAR0 still holds the value from before the poisoned write. The link remains fully operational and subsequent transactions complete normally. This confirms correct EP bit poisoning detection.

Inject Poison TLP – Using EP Bit

Test Result: Verify that the Receiver did not modify BAR0 and replied with UR



Observed result — UR completion, BAR0 unchanged, confirming EP bit detection

7.4 Test Case: Poisoned TLP via Flit_status Field

Injection

The same test sequence as the EP bit case (CfgRd0 / CfgWr0 / CfgRd0), but the poisoning mechanism is changed: the exerciser sets the Flit_status field in the Flit header to indicate poisoning, rather than setting the EP bit in the TLP header. The TLP header itself has the EP bit clear.

Expected and Observed Result

The EP again responds with UR completion status and does not modify BAR0. The DUT correctly interprets the Flit_status poison indication from the Physical Layer Logical Block and treats the TLP payload as invalid despite the EP bit being clear in the TLP header. The CfgRd0 confirms BAR0 is unchanged. This confirms correct Flit_status poisoning detection — the Flit mode-specific mechanism.

Having both injection tests in the validation plan is essential for any DUT that claims to support both poisoning mechanisms. Each mechanism exercises a distinct code path in the receive-side TLP processing logic and must be independently verified.

8. Conclusion and Key Takeaways

8.1 Error Injection is a Core Validation Discipline

Nominal path testing — verifying that a PCIe device correctly handles well-formed traffic under ideal conditions — is necessary but not sufficient for high-confidence PCIe 6.0 validation. Error injection testing is the complementary discipline that validates the other half of the specification: how the device behaves when things go wrong.

PCIe 6.0 Flit mode introduces new error scenarios that simply do not exist in prior generations. A validation plan that does not explicitly address FEC correction versus uncorrectable faults, Selective NAK versus Standard NAK behavior, NAK withdrawal, implicit sequence number reconstruction, and dual poisoning mechanisms has significant gaps in its coverage.

8.2 Flit Mode Demands a New Test Strategy

The specific areas requiring new or updated test cases for PCIe 6.0 Flit mode include:

- **PHY Logical layer:** LTSSM at 64 GT/s with PAM4 equalization, including TS0 OS handling and EC bit transitions across all four EQ phases.
- **FEC:** correctable vs. uncorrectable error injection (pre-FEC and post-FEC), FEC decoder performance under burst error patterns.
- **CRC:** Flit-level CRC corruption to trigger replay, combined with FEC to test the layered detection hierarchy.

- **Selective NAK:** correct upgrade from Standard NAK when Prior Flit Payload bit indicates a Payload Flit.
- **NAK withdrawal:** correct suppression of replay when Prior Flit Payload bit indicates a NOP Flit.
- **Implicit sequence numbering:** analyzer-assisted reconstruction to verify sequence tracking during normal and recovery operation.
- **Dual poisoning mechanisms:** both EP bit and Flit_status field, tested independently.

8.3 Deterministic Injection + Integrated Capture

The approach presented in this white paper — deterministic fault injection via the Xgig protocol exerciser operating as an external link partner, with tightly synchronized protocol capture via the integrated analyzer — provides the combination of stimulus control and observability required for thorough Flit mode validation.

The test cases presented in this white paper — from LTSSM equalization timeout at 64 GT/s to dual-mechanism TLP poisoning — represent a starting framework for PCIe 6.0 Flit mode error injection validation. Engineers building out their own test plans should extend these cases with device-specific scenarios, scaling factors (multiple lanes, multiple TLP sizes), and stress conditions (sustained injection under load, injection during power state transitions).

9. About the Author and Tools

Yamini Shastry leads the Application Engineering team at VIAVI Solutions. Her expertise is in PCIe protocol analysis and validation tooling.

VIAVI Solutions develops protocol test and measurement tools for high-speed interconnects including PCIe, Ethernet, and optical transport networks. The exerciser and analyzer capabilities described in this white paper are implemented in VIAVI's PCIe 6.0 protocol test platform. Interested readers are encouraged to contact VIAVI Solutions for further information on tool capabilities and demonstration opportunities.

Contact Us: +1 844 GO VIAVI | (+1 844 468 4284). To reach the VIAVI office nearest you, visit viavisolutions.com/contact

© 2026 VIAVI Solutions Inc. Product specifications and descriptions in this document are subject to change without notice. Patented as described at viavisolutions.com/patents

pcie6-flitmode-wp-snt-nse-ae
30195216 901 1126