Application Note

Optimizing FPD-Link ADAS Designs for System Level ESD Immunity

Casey McCrea, Taylor Gage

ABSTRACT

FPD-Link camera designs often undergo rigorous automotive EMC testing during OEM qualification. As camera systems in the vehicle have taken on greater responsibility within the vehicle's safety architecture, robustness requirements for those qualifications have become far more stringent as well. Many OEM customers demand error free performance under various high-stress electrical conditions including conducted interference, radiated interference, and even during ESD strikes to the modules, connectors, or cables. This application note can outline key design guidelines that an FPD-Link system designer can implement in both hardware and software design to maximize system level ESD performance.

Table of Contents

Trademarks

1 Introduction

System level ESD testing is a common requirement across a wide range of global automotive OEM. The goals of system level ESD testing are typically two-fold: to make sure that the electrical components within the module do not sustain damage, and to evaluate application level performance during the ESD events. A common misconception regarding system level ESD testing is that the application level performance can relate to the specified ESD ratings in a component data sheet (example. IEC 6100-4-2, HBM, CDM, etc.) The data sheet specified ESD ratings only describe the IC capability to withstand energy discharge to the pins of the device without sustaining damage. This rating is primarily useful for understanding the chip handling and assembly requirements, but critically the rating does not take into account any aspect of application level performance such as data loss. Understand that the application level performance of high speed copper links under EMI/ESD stress is heavily related to the system design. This guide includes TI's recommendations for maximizing immunity performance through strong hardware design practice and software settings optimization.

2 Typical Test Standards Overview

2.1 ISO 10605 Standard

The ISO 10605 standard is one of the most commonly referenced test setups for evaluating system level ESD performance in the automotive environment. The standard is tailored specifically to simulate ESD into a vehicle electrical system and evaluate the system performance. In the context of an FPD-Link system, the test setup typically includes an end to end video link from ECU to camera module, and a means to evaluate the system performance in real time such as a live screen to display camera data.

The ISO 10605 setup can be conducted with either direct (to the DUT) or indirect (to a coupling plane near the DUT) discharge methods which are typically defined by the OEM requirement. ESD energy is discharged to the test setup via an ESD generator gun with a specified RC network and tip shape. Two different discharge types are specified for testing: air discharge and contact discharge. Of the two, contact discharge is typically more stressful to the system in terms of coupled energy, since less energy is dissipated through ionization of air during the strike. The example setup below is configured according to ISO 10605 Annex F where a coupling plate is used under the FPD cable.

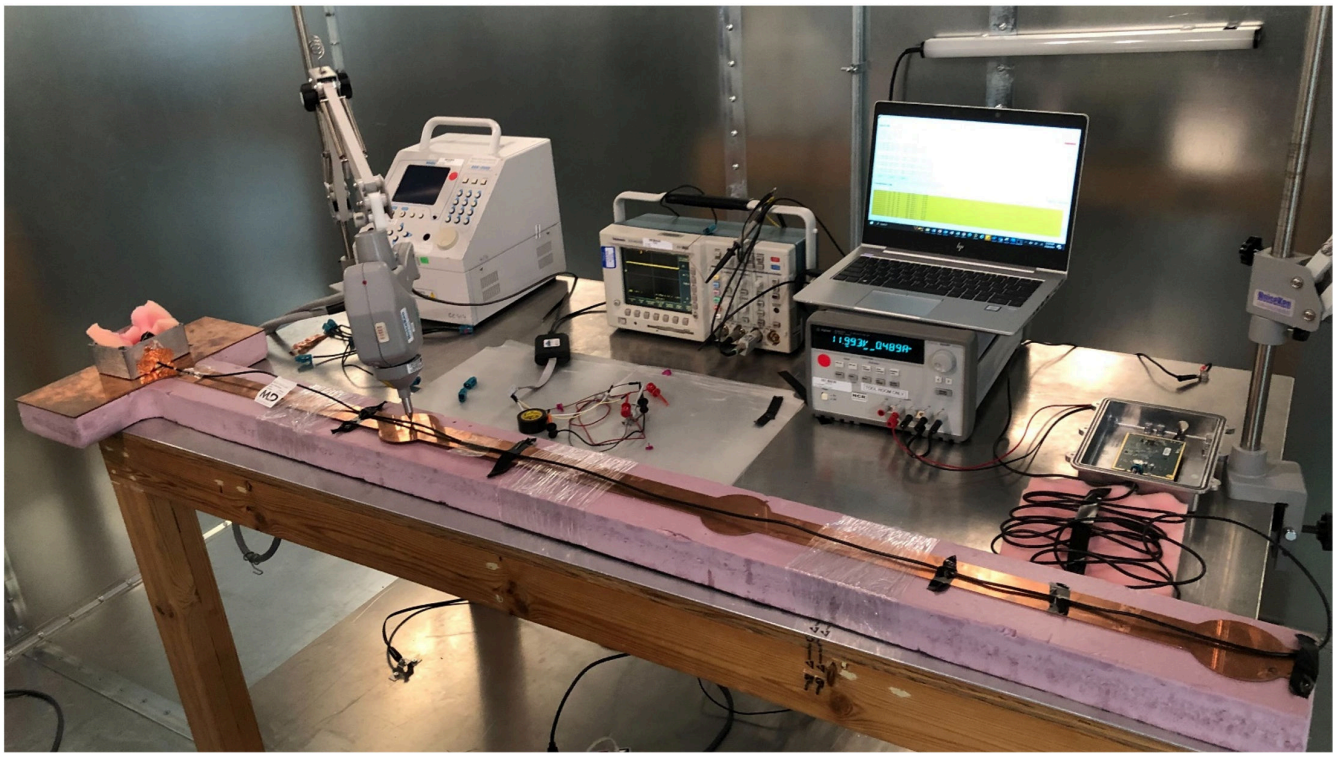

Figure 2-1. ISO 10605 Annex F Example Setup

The second figure outlines the setup used by TI for internal evaluation of camera module performance using a processor platform test board, a full camera module, and a live video display via PC monitor. This setup was used for gathering example data in the Section 5 section. Note that the deserializer system does not include a enclosure like a typical ECU platform. As a result, grounding and shielding of the deserializer system is weaker than expected and average performance is degraded.

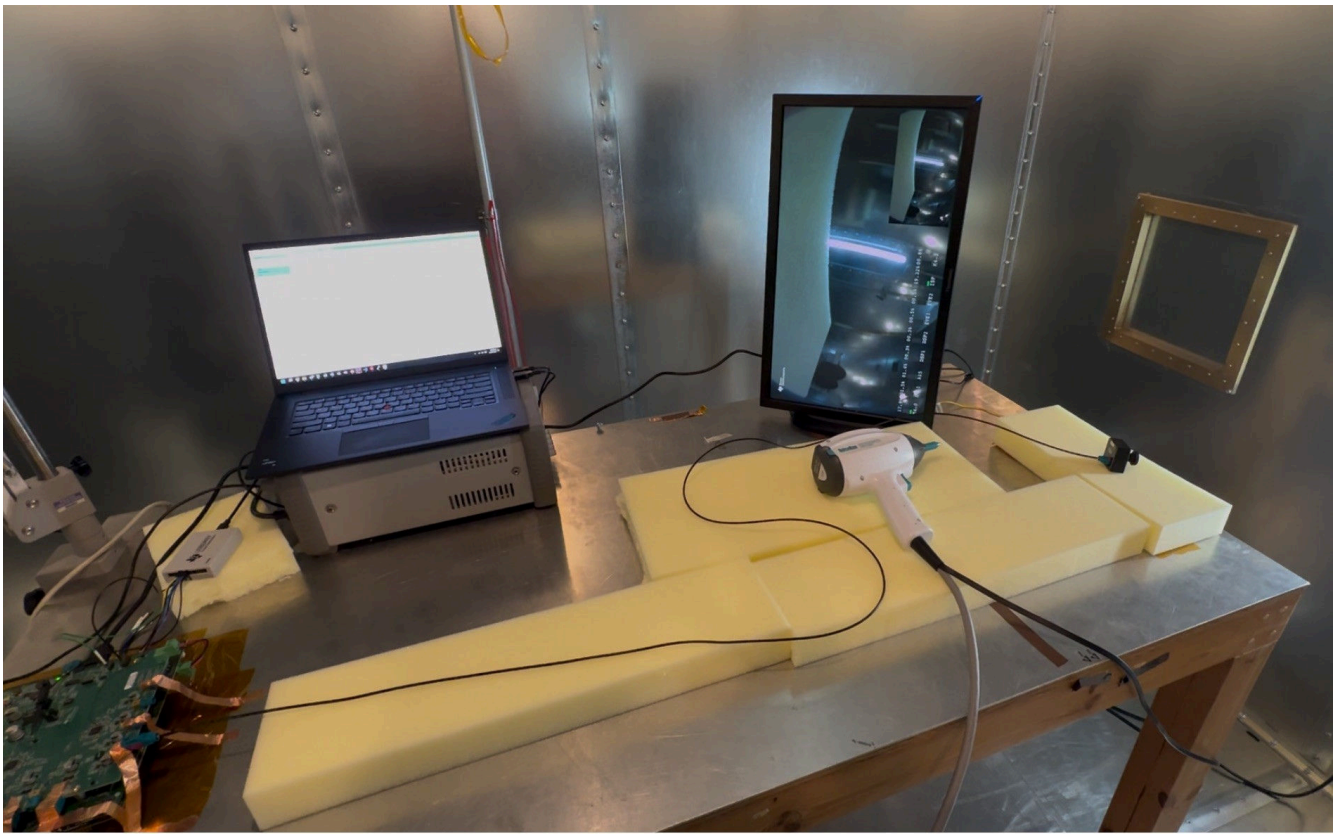

Figure 2-2. ISO 10605 Example Setup

2.2 Performance Status Categorization

System level ESD performance is classified during testing at each test level using a lettering system. The OEM typically defines requirements for system performance status at the various levels.

Table 2-1. Performance Status Table

Status	Description
A	No performance deviation (typically means no perceivable interruption of video for camera systems)
B	Temporary disruption, automatic recovery (typically means momentary visual glitch which is perceivable)
C	Performance disruption which does not automatically recover (typically means a black screen is encountered where power cycle or re-initialization of the system is required to recover video)
D	Permanent damage to the DUT (typically means physical damage to the system including either ICs or passives)

3 FPD-Link Hardware Optimizations

3.1 Connector Grounding

In ADAS applications using PoC, the cable shield acts as the return path for DC power supplied to the remote camera, and acts as the ground reference between the two systems. High energy interference events including ESD strikes and radiated antenna emissions can cause transient disruptions in the system ground reference between the serializer and deserializer device which result in bit errors. Make sure the cable shield is tied into the system ground on both sides of the FPD cable with as little inductance as possible to mitigate this effect. TI recommends to make sure that the connector ground pins are connected to the PCB ground plane on multiple layers with a solid connection rather than thermal relief splines.

Note how the recommended layout not only provides a direct connection for the connector GND pins to a large solid GND plane, but also how the GND plane is stitched to GND planes on other board layers with many vias. In this design, all 8 layers of the board contain a GND plane under the connector.

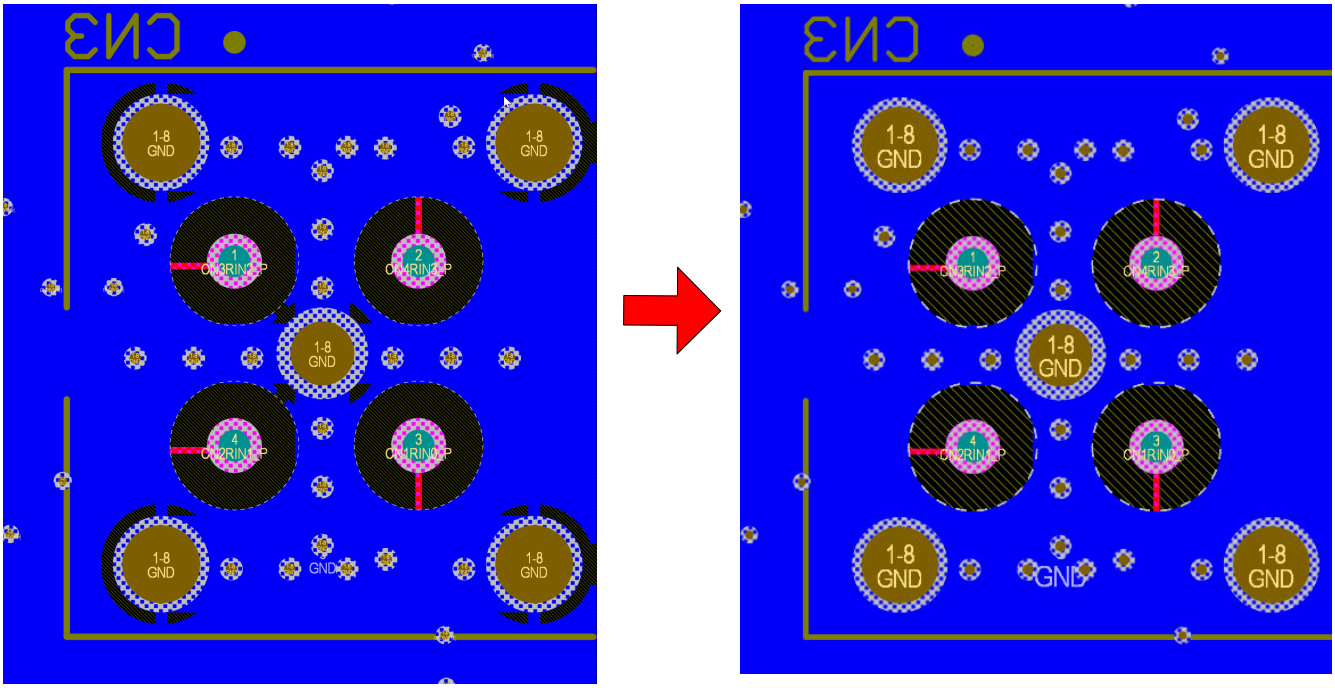

Figure 3-1. Thermal Relief Layout (Left), Recommended Solid GND Layout (Right)

3.2 PCB to Enclosure Grounding

One of the most important aspects of the system design to mitigate the effects of ESD on performance is to provide low inductance discharge paths to direct energy away from the high speed signals. One of the most effective methods to achieve this is to provide strong contact between the PCB ground and the enclosure which is connected to chassis ground. The implementation of an exposed ground ring around the PCB perimeter can give the enclosure multiple points of contact to the board ground via spring connections, or conductive screws and fasteners. Both sides of the PCB need to be in contact with the enclosure ground through a low inductance connection. This helps prevent ground bounce within local sections of the board.

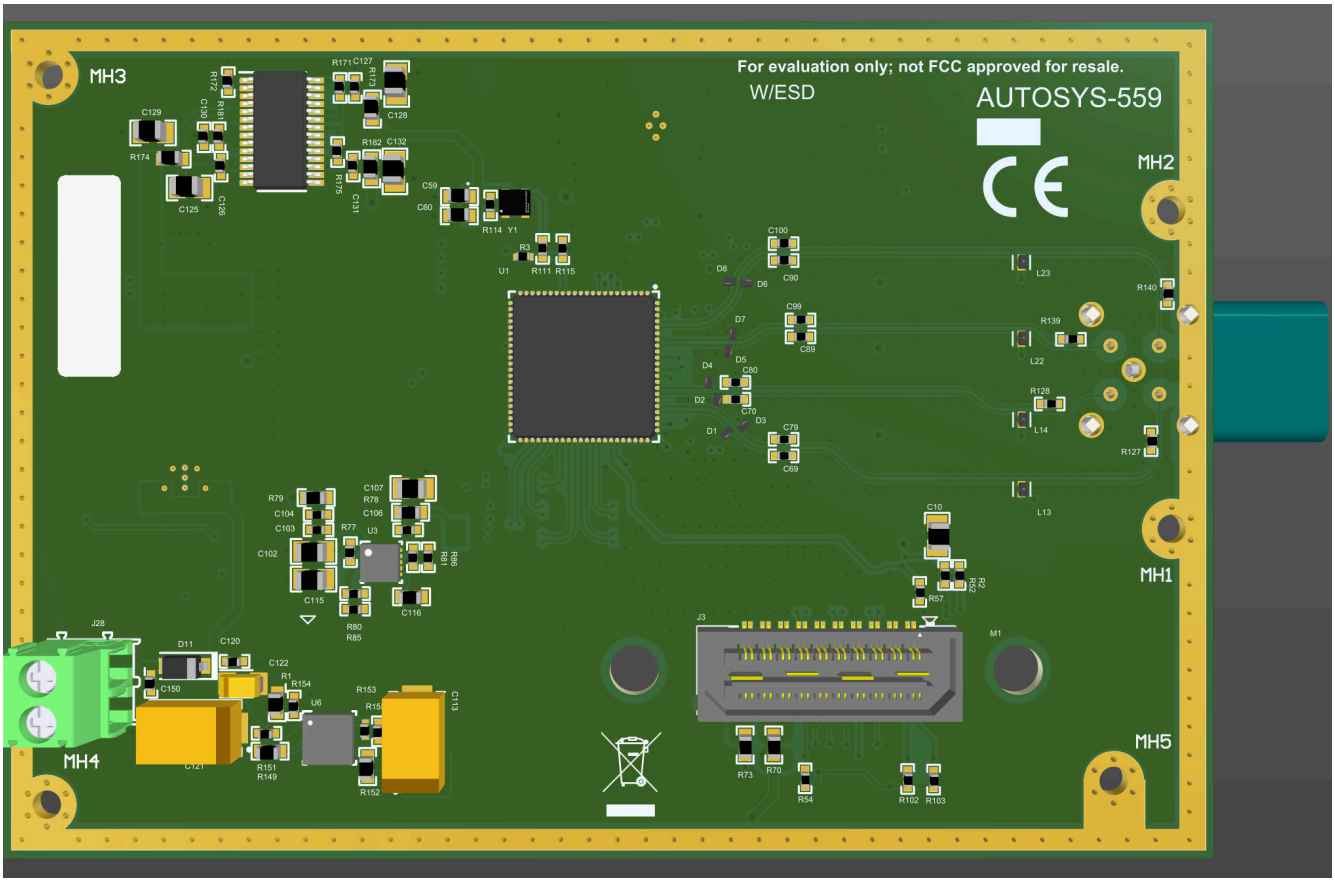

Figure 3-2. Enclosure Ground Ring Example

3.3 MODE Selection

FPD-Link III and IV CSI-2 ADAS serializer support multiple operational modes including synchronous mode and non-synchronous mode. In synchronous mode, the serializer utilizes the back channel signal from the deserializer as a clock reference to generate the forward channel signal, as well as an optional clock output signal (CLKOUT) to a sensor. Due to the architecture of synchronous mode, transient disruption to the back channel signal between the deserializer and the serializer can cause the serializer to temporarily lose the clock reference. When this occurs, there can be a corresponding disruption of the forward channel signal to the deserializer. Conversely, when using non-synchronous mode there is no relationship between the forward channel signal and the back channel signal. Errors in the back channel signal do not impact the serializer ability to transmit a valid forward channel signal which contains the mission critical video data.

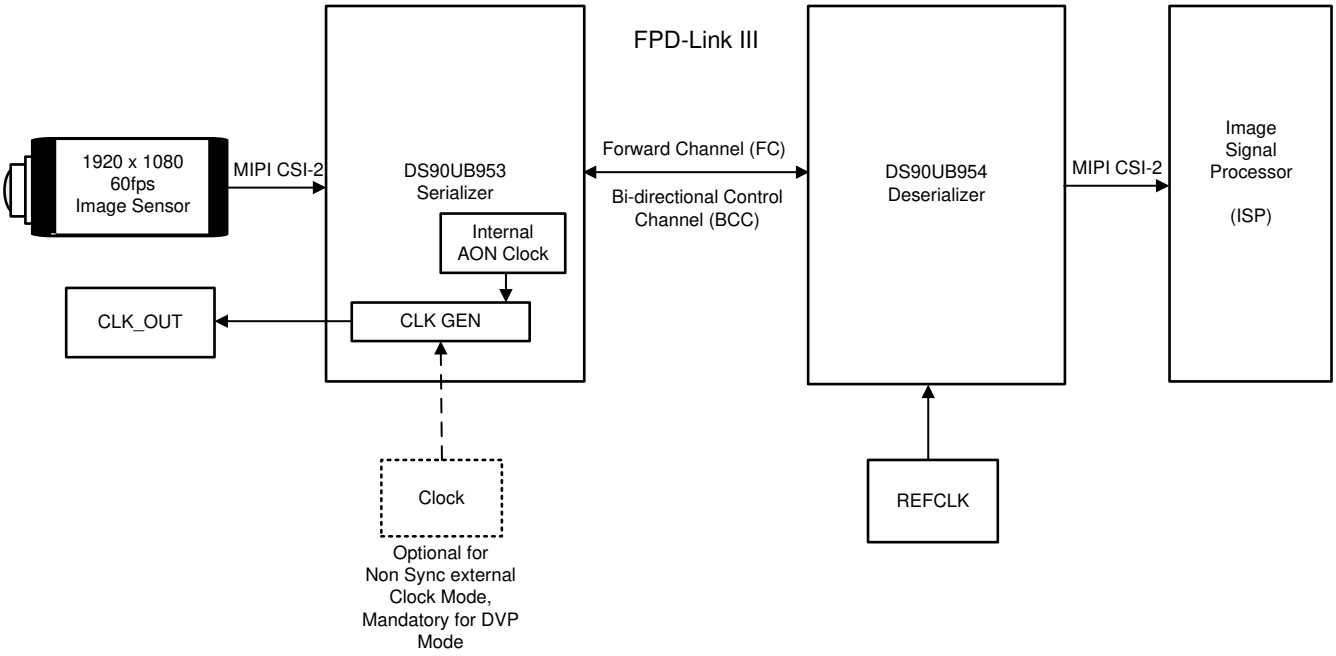

Figure 3-3. FPD-Link ADAS Clocking Architecture

There are two different non-synchronous modes available across FPD-Link III/IV ADAS serializer devices:

- Non-synchronous mode with external CLKIN
  - Available on DS90UB953-Q1, DS90UB935-Q1, DS90UB635-Q1, and DS90UB971-Q1 at all supported rates
  - CLKOUT feature is available
- Non-synchronous mode with internal AON clock
  - Available on DS90UB953-Q1, DS90UB935-Q1, and DS90UB635-Q1
  - Available on DS90UB971-Q1 only in FPD-Link III mode (4Gbps)
  - CLKOUT feature is disabled

TI recommends provisioning camera module designs to support either mode in hardware so that there is flexibility to evaluate both modes during ESD qualification testing. Many system designers prefer synchronous mode to reduce component counts, to enable higher speed back channel GPIO, and to allow for spread spectrum clocking. Depending on the system needs for data rate and CLKOUT capability, the system designer can also provision to add an external oscillator input to the serializer to use non-synchronous external CLKIN mode as an option. From there, the system performance can be evaluated in multiple modes during ESD qualification testing. If the system preference is to utilize synchronous mode and the ESD performance meets the system designer's needs in that mode, then the oscillator can be removed from the BOM population in the final build.

4 FPD-Link Software Optimizations

4.1 LOCK Detection Tuning

The FPD-Link ADAS forward channel protocol packs video payload data as well as other information including GPIO, I2C, status, clocking, and more into serial *frames* which are sent to a downstream deserializer. To properly decode the incoming data frames, the FPD-Link receiver must both lock to the incoming high speed signal by determining the correct transition rate of the data, and the receiver must properly determine the alignment of the incoming data frames (for example, the beginning and end of each serial frame). To accomplish the task of alignment, the FPD-Link ADAS serializer inserts a multi-frame synchronization data pattern called the Decode Cycle Array (DCA) sequence of 130 characters within the forward channel data stream, as well as two dedicated clocking bits per frame (CLK0/CLK1).

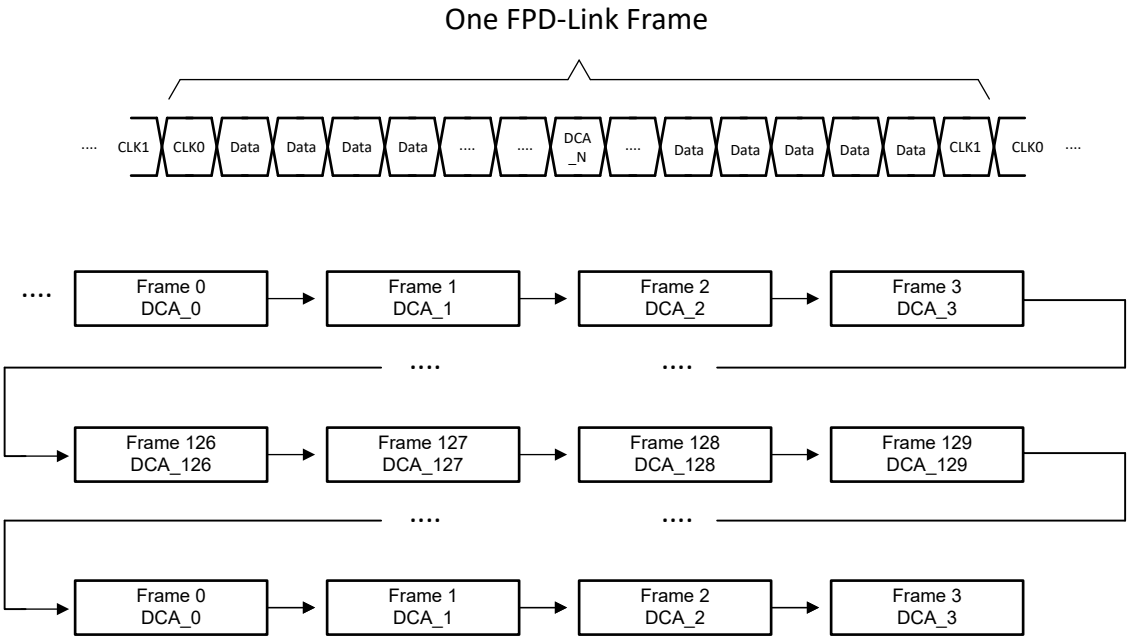

Figure 4-1. FPD-Link ADAS Frame Structure and Sequence

After initial LOCK has been established, the deserializer continuously monitors the incoming serial frames to detect the expected position of the two clocking bits within each frame, and the 130 frame DCA encoding bit pattern. By default, the FPD-Link LOCK signal drops upon the detection of three errors in the clocking and encoding bits over the span of 130 incoming frames. While this default behavior does provide a good leading indicator of an increased BER in the link, the default behavior also causes the link to be highly sensitive to transient error events like ESD strikes. Additionally, note that transient errors can occur in the clocking or encoding bits of the FPD frame without disrupting any application level functionality. So long as errors are transient, the FPD-Link receiver maintains operation and alignment automatically.

To improve robustness of the link under ESD stress, TI recommends to increase the LINK_ERR_THRESH setting within the deserializer which increases the number of errors that must be detected within each 130 frame period before the deserializer drops LOCK and begins re-acquisition. This threshold configuration can be found in register 0xB9 (LINK_ERROR_COUNT). For the configured threshold to take effect, LINK_ERR_COUNT_EN must also be set to 1.

Another optimization which can be applied to decrease sensitivity of the LOCK drop algorithm is to disable the clock bit error detection as part of the link error counter. By configuring register 0xB6 = 0x1C, detection of the clocking bits is no longer used as a marker to drop LOCK. Only the repeating 130 frame DCA encoding pattern is used for LOCK drop detection with this setting.

4.2 Parity Error Handling

In addition to multi-frame encoding and clock error checks in the FPD-Link forward channel frames, each FPD-Link frame also includes known parity. This means that each incoming frame has an even number of 1s which is verified by the deserializer in real time. If the deserializer recognizes an incoming FPD frame has an odd number of 1s, then the deserializer can flag an error and increment a parity error counter for diagnostic purposes.

By default, ADAS deserializer are configured to discard FPD-Link forward channel frames which contain a parity error, meaning that no information contained within that frame is forwarded to downstream data paths. While this default behavior can prevent error propagation, the default behavior also increases sensitivity to transient errors which can have no impact on the application. Consider an example where an incoming FPD-Link frame has a single bit error causing a parity error to be flagged.

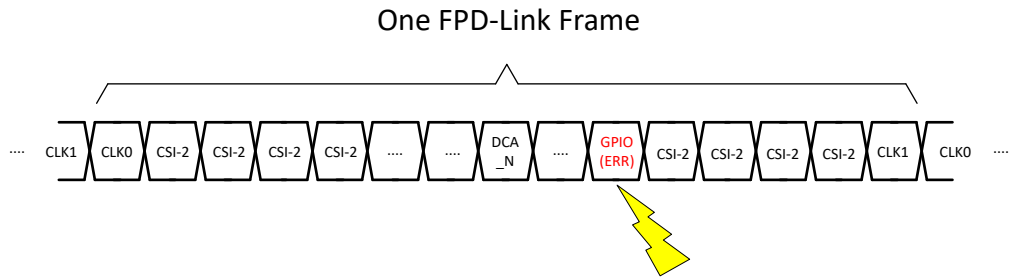

Figure 4-2. Parity Error Example

In this example, a bit error impacted one of the forward channel GPIO slots within the FPD-Link frame. As a result, the number of 1s in the frame changes to an odd value and the deserializer discards the entire packet. However this kind of transient error in one of the frame clocking bits can be tolerated without breaking LOCK or losing frame boundaries. Additionally, this particular error did not affect any of the CSI-2 video data within the frame. Depending on the application use case, this error can have no impact on any aspect of the system whatsoever (for example if no forward channel GPIO are utilized by the application).

FPD-Link III and IV CSI-2 ADAS deserializer products can be configured to not discard packets with parity errors by setting DISCARD_ON_PAR_ERR to 0 in the PORT_CONFIG2 register. When this setting is disabled, sensitivity to transient error events is reduced which can help significantly in achieving class A performance during immunity testing.

Allowing packets with parity errors to be processed rather than discarded does not imply that corrupted video data can be passed through the deserializer unnoticed. This is because FPD-Link ADAS devices operate by tunneling CSI-2 packets from end to end. CSI-2 payload and checksum are both passed unaltered through the serializer and deserializer before reaching the downstream application processor. No re-calculation is done for the CSI-2 payload checksum (CRC). As a result, bit errors in the FPD-Link channel that do corrupt video pixel data cause the corresponding CSI-2 packet to fail CRC checking at the final destination. The application processor can then decide how best to handle the error depending on the application needs.

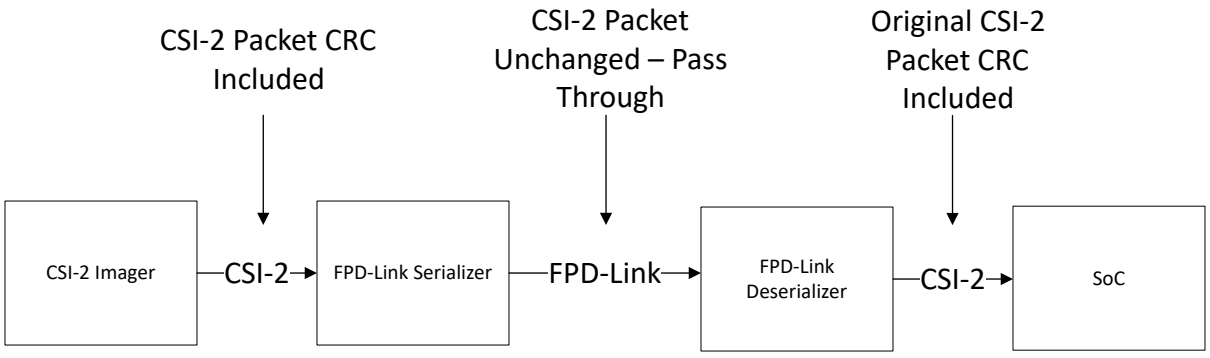

Figure 4-3. CSI-2 Packet Tunneling

4.3 Forward Error Correction

Select FPD-Link devices within the FPD-Link III and FPD-Link IV product families include forward error correction (FEC) functionality within the FPD-Link forward channel path. Forward error correction utilizes unoccupied channel bandwidth to send error correcting code words along with the forward channel data payload which allows the downstream deserializer to correct transient errors. FEC functionality is available on the following devices:

- DS90UB971-Q1
- DS90UB953-Q1
- DS90UB953A-Q1
- DS90UB935-Q1
- DS90UB635-Q1
- DS90UB960-Q1
- DS90UB962-Q1
- DS90UB662-Q1
- DS90UB9702-Q1
- DS90UB9722-Q1
- DS90UB9724-Q1
- DS90UB9742-Q1

FEC functionality is not available on the following devices:

- DS90UB933-Q1
- DS90UB913A-Q1
- DS90UB913Q-Q1
- DS90UB934-Q1
- DS90UB954-Q1
- DS90UB936-Q1
- DS90UB638-Q1
- DS90UB964-Q1

To utilize FEC functionality, both the serializer and deserializer device within the link pair must support FEC.

There are three selectable FEC modes available with different code sizes for the inserted error correcting codes (ECC):

- 6-bit FEC
- 7-bit FEC
- 8-bit FEC

When FEC is enabled, the serializer sends multiple (X) data frames, followed by multiple (Y) frames containing error correcting codes (ECC) in sequence. The error correcting code chunks (Y) protect CSI-2 data, GPIO data, and I2C data by allowing the deserializer to detect 2-bit errors, and correct 1-bit errors within the data chunk (X) much like the data protection within the packet header for CSI-2 data.

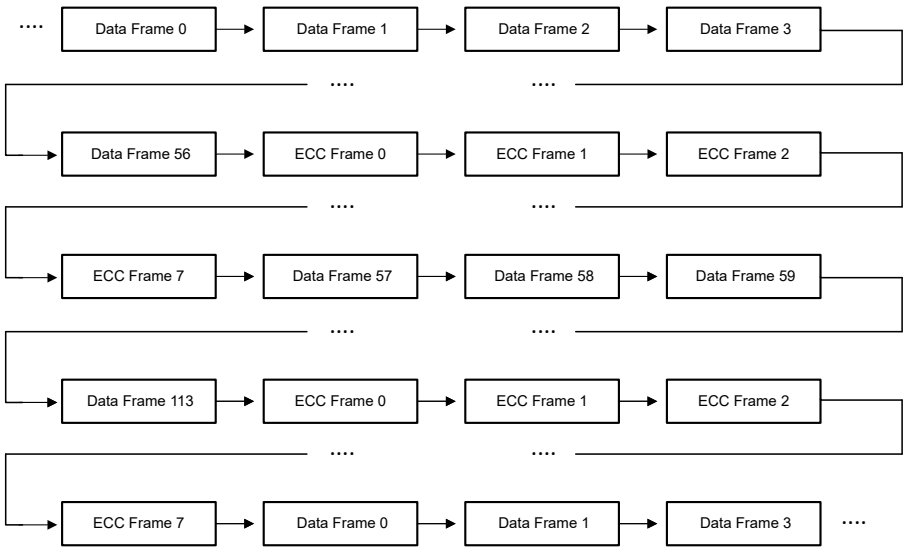

Figure 4-4. 8-Bit FEC Sequence Example

Each FEC mode defines a ratio of (X) data frames and (Y) ECC frames sent within a 130 frame DCA sequence. The 6-bit mode protects smaller chunks of data, but correction codes are sent more frequently. The 8-bit mode protects larger chunks of data, with correction codes sent less frequently. The ratios are defined below:

Table 4-1. ECC Code Transmission

FEC Mode	Number of Data Frames (X)	Number of ECC Frames (Y)	ECC Frames Per DCA Sequence	Data Frames Per DCA Sequence	FEC Overhead
6-bit	20	6	30	100	23%
7-bit	Alternating 36, 37	7	21	109	16%
8-bit	57	8	16	114	12%

FEC overhead reduces the allowable payload capability for video in the FPD-Link forward channel pipeline. Care must be taken to verify that the link video bandwidth after FEC is enabled can meet the application's needs.

Table 4-2. FEC Overhead Impact Examples

Device	FPD-Link Rate (Gbps)	Max CSI-2 Bandwidth (Gbps)	FEC Mode	Max CSI-2 Bandwidth FEC Enabled (Gbps)
DS90UB971-Q1	7.55	6	6-bit	4.8
DS90UB971-Q1	7.55	6	7-bit	5.1
DS90UB971-Q1	7.55	6	8-bit	5.3
DS90UB953-Q1	4	3.2	6-bit	2.6
DS90UB953-Q1	4	3.2	7-bit	2.7
DS90UB953-Q1	4	3.2	8-bit	2.8
DS90UB935-Q1	4	2.528	6-bit	2.528 (Lesser of 2.528Gbps and 3.2Gbps/1.23)
DS90UB935-Q1	4	2.528	7-bit	2.528 (Lesser of 2.528Gbps and 3.2Gbps/1.16)
DS90UB935-Q1	4	2.528	8-bit	2.528 (Lesser of 2.528Gbps and 3.2Gbps/1.12)

To enable FEC functionality, configure deserializer register 0x4A (port specific):

- 6-bit FEC: 0x4A = 0x01
- 7-bit FEC: 0x4A = 0x02
- 8-bit FEC: 0x4A = 0x03

The deserializer notifies the serializer to enter FEC mode via the back channel automatically.

4.3.1 FEC Test Capabilities

Supporting serializer and deserializer devices contain verification features to check FEC functionality is operational. The serializer can inject FEC errors via register command, and the deserializer provides diagnostics for the number of ECC1/ECC2 FEC errors detected in the FPD-Link channel. Note that these are not the same as ECC1/ECC2 CSI-2 errors reported by the CSI_RX_STS register.

```
board.WriteI2C(serAddr,0x76,0x01) # Force 1x FPD ECC1 error (Correctable)
board.WriteI2C(serAddr,0x76,0x02) # Force 1x FPD ECC2 error (Detectable)
```

Deserializer diagnostic counters for FEC errors are port-specific. Select the desired RX port with register 0x4C before reading diagnostics.

```
board.WriteI2C(desAddr,0x4C,0x01) # Select RX port 0
board.ReadI2C(desAddr,0x48) # Read back count of FPD ECC1 Errors (Clear on Read)
board.ReadI2C(desAddr,0x49) # Read back count of FPD ECC2 Errors (Clear on Read)
```

5 Optimization Test Data

To illustrate the impact of hardware and software optimizations on system level ESD performance, TI conducted ISO 10605 testing under multiple deserializer configurations using the FPD-Link IV DS90UB971-Q1 and DS90UB9702-Q1 devices:

- Basic hardware design/No software optimizations
- Optimized hardware design/No software optimizations
- Optimized hardware design + Software optimizations

The tests were conducted using a DS90UB971-Q1 camera module operating at 7.55Gbps, connected to the deserializer hardware (DS90UB9702-Q1) via a 1.5m Dacar 302 cable. The deserializer system is housed inside a metal enclosure to emulate a typical ECU subsystem design. The deserializer enclosure is connected to the table GND plane, and the cable as well as the camera module are suspended above the ground plane using an insulating block.

The goal of this ISO 10605 testing is to emulate a typical OEM EMC approval setup, where performance of the system is categorized into Class A, B, C, or D during ESD strike events. By testing three systems with incremental hardware and software optimizations in place, a typical improvement trend is realized. This testing does not reflect the precise results that are expected in every system design; rather the testing is meant to demonstrate *relative* performance improvement that is possible through optimized hardware and software design.

5.1 Baseline Hardware - No Software Optimization

The first round of testing was performed with a baseline DS90UB9702-Q1 hardware design that does not include the optimizations described earlier in the guide. The system is configured with a basic link setup that does not include any of the software optimizations for transient immunity. The goal of this testing is to establish a baseline for ESD performance. The system was tested with both contact and air discharge methods at various strike points around the camera module and deserializer box. After each stress level, diagnostic registers were recorded to identify if there was any loss of FPD-Link LOCK, or uncorrectable CSI-2 errors during the test which can result in flicker, black screen, or distortion in the video feed.

Table 5-1. Baseline Hardware - No Software Optimization ESD Results

Strike Type	Level	Typical OEM Requirement	Result
Air	±4kV	Class A	B
Contact	±4kV	Class A	B
Air	±6kV	Class A	B
Contact	±6kV	Class A	B
Air	±8kV	Class A	C
Contact	±8kV	Class B	C
Air	±15kV	Class B	C
Contact	±15kV	Class B	C

- Class A = No loss of LOCK, no uncorrectable CSI-2 errors.
- Class B = Temporary loss of LOCK or CSI-2 errors which automatically recover
- Class C = Soft reset of the SoC required to restart video

5.2 Optimized Hardware - No Software Optimization

The second round of testing was performed with an optimized revision of the deserializer hardware with the goal of isolating the high speed signal path from the ESD strike noise. The same initialization software was used as in the previous test to demonstrate the relative improvement from hardware alone. The addition of good layout practice and physical design considerations allows the system to achieve Class A performance at slightly higher levels than the baseline example.

Table 5-2. Optimized Hardware - No Software Optimization ESD Results

Strike Type	Level	Typical OEM Requirement	Result
Air	±4kV	Class A	A
Contact	±4kV	Class A	A
Air	±6kV	Class A	B
Contact	±6kV	Class A	B
Air	±8kV	Class A	B
Contact	±8kV	Class B	B
Air	±15kV	Class B	B
Contact	±15kV	Class B	B

- Class A = No loss of LOCK, no uncorrectable CSI-2 errors.
- Class B = Temporary loss of LOCK or CSI-2 errors which automatically recover

5.3 Optimized Hardware and Software

The final round of testing incorporated software optimizations to the deserializer initialization to further improve performance. The addition of tweaks to the LOCK sensitivity as well as the addition of FEC to the FPD-Link channel significantly improves the overall performance under transient ESD stress, and allows for achieving Class A performance under higher levels of ESD noise.

Table 5-3. Optimized Hardware and Software ESD Results

Strike Type	Level	Typical OEM Requirement	Result
Air	±4kV	Class A	A
Contact	±4kV	Class A	A
Air	±6kV	Class A	A
Contact	±6kV	Class A	A
Air	±8kV	Class A	A
Contact	±8kV	Class B	B
Air	±15kV	Class B	B
Contact	±15kV	Class B	B

- Class A = No loss of LOCK, no uncorrectable CSI-2 errors.
- Class B = Temporary loss of LOCK or CSI-2 errors which automatically recover

6 Example Scripts for Software Optimization

Level 1 Optimizations

The following script example combines the recommended software optimizations for system level ESD testing performance. These settings are generally applicable to all systems using FPD-Link III/IV ADAS devices with FEC capability. This example uses 7-bit FEC, but the FEC setting can be adjusted based on system needs and testing performance.

```
board.WriteI2C(desAddr,0x4C,0x0F) # Select all RX Ports
board.WriteI2C(desAddr,0xB9,0x1F) # Increase LINK_ERR_THRESH
board.WriteI2C(desAddr,0x4A,0x02) # Enable FPD FEC (7-bit)
board.WriteI2C(desAddr,0x7C,0x00) # Disable FPD frame discard on parity error
board.WriteI2C(desAddr,0xB6,0x1C) # Disable CLK0/CLK1 check for LOCK drop
```

Level 2 Optimizations

In some cases, performance can be improved further by adjusting FPD-Link AEQ gain, or by locking the FPD-Link AEQ range to a smaller value. Adjustments to AEQ needs to only be used as a last resort, when all other hardware and level 1 software optimizations have been exhausted. Adjusting AEQ to extreme values can also have unintended negative side effects to link performance, the recommendation is to contact TI for assistance when manually adjusting AEQ.

For FPD-Link devices including DS90UB960-Q1, DS90UB962-Q1, DS90UB662-Q1, DS90UB954-Q1, DS90UB936-Q1, DS90UB934-Q1, DS90UB964-Q1, and DS90UB638-Q1, performance can be improved by forcing low AEQ values. In most cases, for cable lengths <2-3m, forcing an AEQ value of 0 can improve system level ESD performance.

```
# 960, 962, 662 AEQ Force for short cable length
board.WriteI2C(desAddr,0x4C,0x0F) # Select all RX Ports
board.WriteI2C(desAddr,0xD4,0x01) # Force AEQ = 0
```

For FPD-Link IV deserializer operating with FPD-Link IV CDR mode, including DS90UB9702-Q1, DS90UB9722-Q1, DS90UB9724-Q1, and DS90UB9742-Q1, performance can be improved by increasing AEQ values. An example starting value for system designers to experiment with is to add +5 to the automatically selected AEQ. To increase the AEQ, make the following underlined adjustments to TI's initialization script within the en_AEQ_LMS() function.

```
def en_AEQ_LMS(first_time_power_up=1):
    if(first_time_power_up == 1):
    board.WriteI2C(devAddr,0xB1,0x2C)
    read_aeq_init = board.ReadI2C(devAddr,0xB2)
    board.WriteI2C(devAddr,0xB1,0x27)
    board.WriteI2C(devAddr,0xB2,read_aeq_init + 5)
    board.WriteI2C(devAddr,0xB1,0x28)
    board.WriteI2C(devAddr,0xB2,read_aeq_init + 6)
    ...
    ...
```

7 Additional System Level Software Options

For Class A performance at the system level above 6-8kV levels, additional options can be implemented within the SoC/Processor which receives the video data. A common method to achieve Class A performance across higher ESD stress levels is to implement a frame buffering scheme which can discard frames with errors or partially received frames to prevent visual disruption. This method is especially effective for systems where human vision is used to judge the video quality owing to the fact that repeats of single frames are difficult to detect at frame rates of 30Hz or higher. This method does not need be used as a substitute for strong system design practices because low baseline system performance coupled with this method can result in visually perceptible delay.

Step 1

Configure the deserializer device to output an interrupt once the device detects errors in one of the RX ports. The following example assumes a quad channel deserializer is used with all 4 RX ports active.

```
board.WriteI2C(desAddr,0x23,0x8F) # Enable interrupts for all RX ports
```

Step 2

Monitor the interrupt pin through the SoC. When the interrupt is triggered, check the interrupt source to identify the cause and generate a port-specific error flag for conditions which can cause corruption of the video path. Note that some errors like parity or ECC1 are excluded from monitoring since the errors do not directly affect the video.

```
ERR = [0, 0, 0, 0] # RX Port-specific error flags
INT_STS = board.ReadI2C(desAddr,0x24) # Check which RX port triggered the interrupt
if INT_STS & 0x01 != 0: # RX0 interrupt
    board.WriteI2C(desAddr,0x4C,0x01) # Select RX0
    RX_PORT_STS1 = board.ReadI2C(desAddr,0x4D)
    RX_PORT_STS2 = board.ReadI2C(desAddr,0x4E)
    CSI_RX_STS = board.ReadI2C(desAddr,0x7A)
    if RX_PORT_STS1 & 0x10 != 0: # LOCK_STS_CHG
        ERR[0] = 1
    if RX_PORT_STS2 & 0xC1 != 0: # LINE_LEN_CHG, LINE_CNT_CHG
        ERR[0] = 1
    IF CSI_RX_STS & 0X0E != 0: # LENGTH_ERR, CKSUM_ERR, ECC2_ERR
        ERR[0] = 1
if INT_STS & 0x02 != 0: # RX1 interrupt
    board.WriteI2C(desAddr,0x4C,0x12) # Select RX1
    RX_PORT_STS1 = board.ReadI2C(desAddr,0x4D)
    RX_PORT_STS2 = board.ReadI2C(desAddr,0x4E)
    CSI_RX_STS = board.ReadI2C(desAddr,0x7A)
    if RX_PORT_STS1 & 0x10 != 0: # LOCK_STS_CHG
        ERR[1] = 1
    if RX_PORT_STS2 & 0xC1 != 0: # LINE_LEN_CHG, LINE_CNT_CHG
        ERR[1] = 1
    IF CSI_RX_STS & 0X0E != 0: # LENGTH_ERR, CKSUM_ERR, ECC2_ERR
        ERR[1] = 1
if INT_STS & 0x04 != 0: # RX2 interrupt
    board.WriteI2C(desAddr,0x4C,0x24) # Select RX2
    RX_PORT_STS1 = board.ReadI2C(desAddr,0x4D)
    RX_PORT_STS2 = board.ReadI2C(desAddr,0x4E)
    CSI_RX_STS = board.ReadI2C(desAddr,0x7A)
    if RX_PORT_STS1 & 0x10 != 0: # LOCK_STS_CHG
        ERR[2] = 1
    if RX_PORT_STS2 & 0xC1 != 0: # LINE_LEN_CHG, LINE_CNT_CHG
        ERR[2] = 1
    IF CSI_RX_STS & 0X0E != 0: # LENGTH_ERR, CKSUM_ERR, ECC2_ERR
        ERR[2] = 1
if INT_STS & 0x08 != 0: # RX3 interrupt
    board.WriteI2C(desAddr,0x4C,0x38) # Select RX3
    RX_PORT_STS1 = board.ReadI2C(desAddr,0x4D)
    RX_PORT_STS2 = board.ReadI2C(desAddr,0x4E)
    CSI_RX_STS = board.ReadI2C(desAddr,0x7A)
    if RX_PORT_STS1 & 0x10 != 0: # LOCK_STS_CHG
        ERR[3] = 1
    if RX_PORT_STS2 & 0xC1 != 0: # LINE_LEN_CHG, LINE_CNT_CHG
        ERR[3] = 1
    IF CSI_RX_STS & 0X0E != 0: # LENGTH_ERR, CKSUM_ERR, ECC2_ERR
        ERR[3] = 1
```

Step 3

Based on the port-specific error flag, trigger the SoC to discard the current video frame for ports with errors and display the previous frame twice.

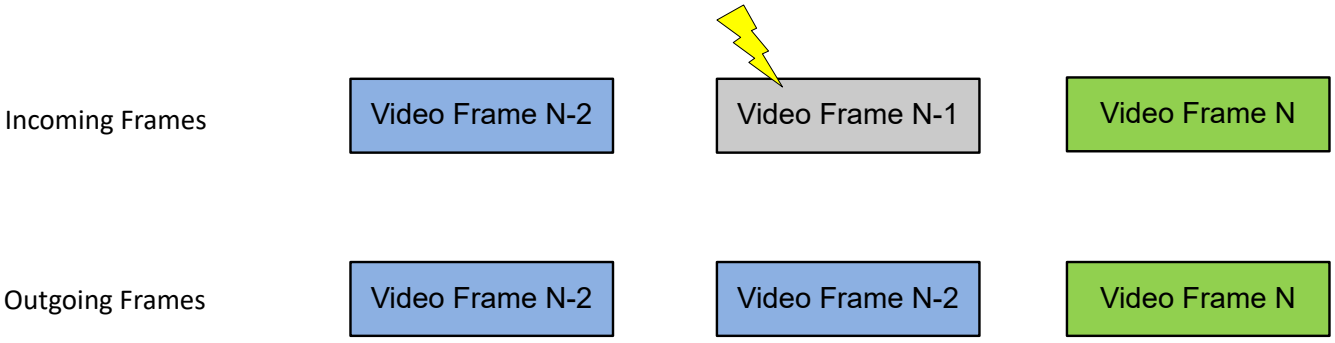

Figure 7-1. Frame Discard and Repeat Example

8 Summary

System level ESD performance with FPD-Link is a function of both hardware and software design practice. At the hardware level, the design must incorporate shielding of the high speed signals, and strong grounding between the system enclosure and the PCB. From the software side, tools including FEC and LOCK detection tuning can be used to boost effective BER for the video path. The techniques described in this application note are targeted towards system level ESD testing, but the same practices can be applied to other noise injection EMC tests including BCI or radiated immunity to improve performance.

9 References

- ISO, *10605:2023 Road Vehicles - Test Methods for Electrical Disturbances From Electrostatic Discharge*

10 Revision History

Changes from Revision * (October 2024) to Revision A (October 2024)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1
• Added additional author	1
• Updated text in parity error example (even vs. odd)	8

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265