Application Report
SNLA287–December 2018

# Using FPD-Link III SerDes in Applications Requiring High-Bandwidth Digital Content Protection (HDCP)

*FPD-Link Applications*

**ABSTRACT**

FPD-Link III physical layer is the preferred high-speed digital interface between high-definition video sources and displays in automotive applications. The FPD-Link III interface supports various features necessary in an array of automotive video display applications, including features necessary for transport of high-definition audiovisual content that requires copy protection.

This application report provides guidelines for setup and configuration of FPD-Link III SerDes (Serializers and Deserializers) for use in applications requiring High-Bandwidth Digital Content Protection (HDCP). In this report, the DS90UH947-Q1 serializer and DS90UH948-Q1 deserializer are used as examples throughout the document, however the provided information is applicable to all HDCP capable FPD-Link III for IVI (In-Vehicle Infotainment) devices designated by part numbers starting with DS90UH.



**Contents**



**List of Figures**

**List of Tables**



# 1 Introduction

FPD-Link III for IVI Serializers and Deserializers (SerDes) support High-speed Digital Content Protection (HDCP) systems as described in HDCP Rev1.4 with minor differences highlighted later in this document.

In a HDCP system with FPD-Link III devices, the serializer provides HDCP encryption of audiovisual content when connected to an HDCP capable source. The content decryption is performed by the deserializer. HDCP authentication and shared key generation is performed using the HDCP Control Channel, which is embedded in the forward and backward channels of the FPD-Link III interface. On-chip Non-Volatile Memory (NVM) is used to store the HDCP keys. The confidential HDCP keys are loaded into on-chip NVM by TI during the manufacturing process and are not accessible external to the device.

# 2 HDCP Overview

The HDCP functionality in FPD-Link III devices is implemented per the HDCP Rev1.4 specification. This section describes the basics of the implementation and differences from the specification.

## *2.1 Authentication Protocol*

An HDCP enabled FPD-Link III transmitter communicates with an HDCP enabled FPD-Link III receiver over bidirectional I2C link. In an FPD-Link III system, the I2C link is embedded in the FPD-Link III forward channel and back channel links between the transmitter and the receiver.

### 2.1.1 First Part: Encrypted Key Exchange

The first part of the authentication protocol consists of an exchange of information between an HDCP transmitter and receiver to establish shared values for the encryption. All of the information transfer between the HDCP transmitter and receiver is handled completely by hardware. To complete the first part of authentication exchange, an external controller may be required to validate that the HDCP receiver Key Selection Vector (KSV) has not been revoked. This step is not required if the HDCP transmitter is located in a repeater.

The HDCP Rev1.4 specification requires that the HDCP receiver computes the shared secret value, Km, within 100 ms. HDCP FPD-Link III requires that the HDCP receiver computes the Km within 1500 clock cycles of the audiovisual data interface on the link. At 50 MHz, this would correspond to 30 µs allowing for much faster completion of the authentication protocol.

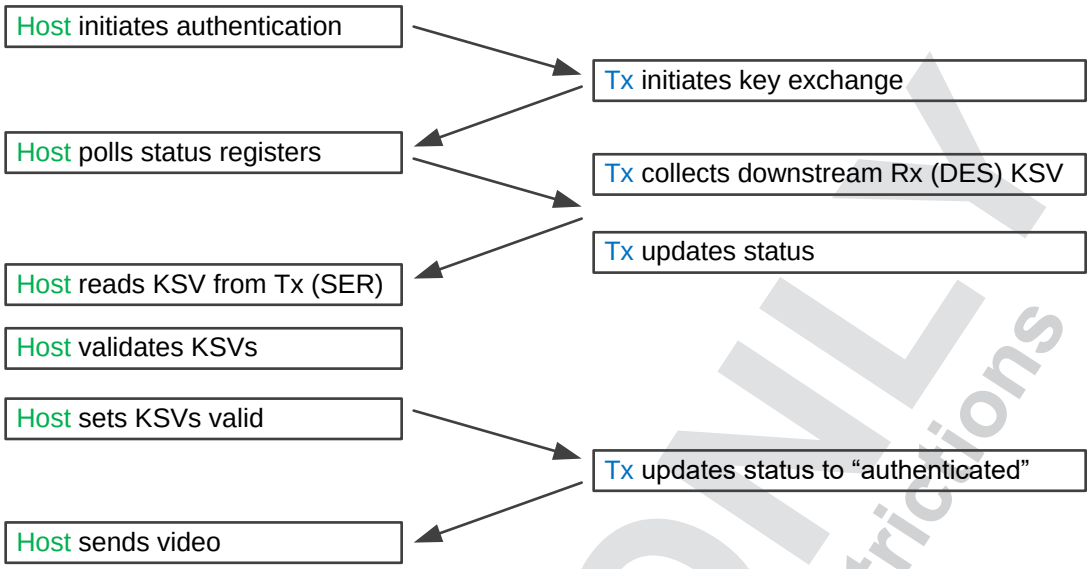



**Figure 1. HDCP Authentication With HDCP FPD-Link III Devices**

### 2.1.2 Second Part: HDCP Repeater Authentication

The second part of the authentication flow is executed if the attached HDCP receiver is also an HDCP repeater as indicated by the REPEATER bit read in the first part of authentication. If so, the HDCP receiver must compile a list of the downstream KSVs. The HDCP transmitter polls the HDCP receiver to determine when the KSV list is ready, reads the KSV list, and indicates to an external controller that the KSV list is ready. The external controller validates the KSV list against the revocation list and indicates to the HDCP transmitter that the list is valid. Errors are also reported to the controller to allow the controller to abort or restart authentication.

Default timing requirements for HDCP repeater authentication are as detailed in Table 2-1 of the HDCP Rev1.4 specification. HDCP enabled FPD-Link III typically completes in tens of milliseconds or less rather than up to the 5 seconds allowed by the specification. Authentication timing is critical in automotive applications where display video must be available in much less than a second.

### 2.1.3 Third Part: Encryption Synchronization

The third part of the authentication flow involves periodic checking of the encryption synchronization between the HDCP transmitter and HDCP receiver. Two methods are defined in the specification and both are supported in the HDCP transmit digital core. The basic Link Verification verifies the Ri parameter generated by the HDCP receiver matches that generated by the HDCP transmitter. The Ri value is generated and verified on every 128th frame using a synchronous method. Optionally, to provide better confirmation of synchronization, the Ri value may be checked prior to frame 128. The implementation also supports Enhanced Link Verification which checks the Pj value for every 16th frame. The Pj value includes checking that decrypted content matches for the transmitter and receiver and therefore provides a better check of the synchronization.

HDCP enabled FPD-Link III also supports an option for faster detection of loss of synchronization. Fast Link Verification (FAST_LV) mode allows speed-up of either the normal or Enhanced Link Verification options. In Fast Link Verification, Ri is checked every 16th frame rather than every 128th frame. In Fast Link Verification if using Enhanced Link Verification, Pj is checked every 2nd frame rather than every 16th frame. The HDCP transmitter enables Fast Link Verification by setting the FAST_LV bit in the HDCP_DBG register at address 0xC0.

## 2.2 HDCP Transmitter with FPD-Link III State Diagram

The HDCP enabled FPD-Link III transmitters implement the HDCP Link State Diagram and HDCP Transmitter Authentication Protocol State Diagram. Since an HDCP enabled FPD-Link III transmitters only operate in HDCP-FPD-Link III mode, and not in HDMI or DVI modes, the transmitters implement a simplified Transmit Link State Diagram as shown in Figure 2. The HDCP Transmitter Authentication Protocol State Diagram is implemented per the HDCP Rev1.4 specification. EDID ROM reading is not required to determine an operating mode. Although EDID ROM reading is not required, EDID ROM information may be read by a host controller via the FPD-Link III bidirectional control channel.

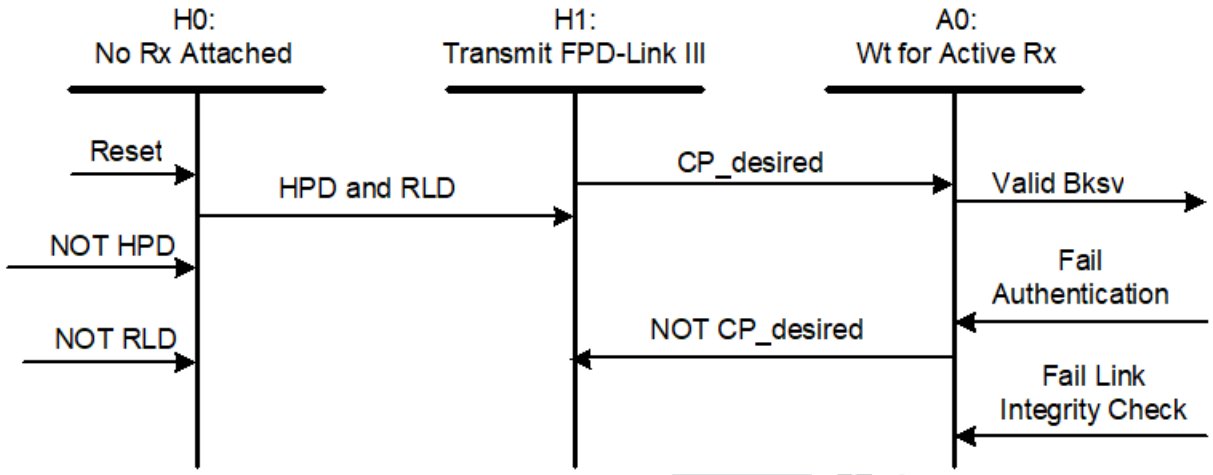



**Figure 2. HDCP Transmitter with FPD-Link III State Diagram**

**Transition Any State : H0.** Reset conditions at the HDCP Transmitter or loss of Hot Plug Detect (HPD) or loss of Receiver Lock Detect (RLD) case the HDCP transmitter to enter the No Receiver Attached state

**Transition H0:H1.** The detection of Hot Plug Detect and Receiver Lock Detect (RLD) indicate that a sink device is attached.

**Transition H1:A0.** If content protection is desired by the Upstream Content Control function, then the HDCP transmitter waits for the availability of an active HDCP receiver.

### 2.2.1 Hot Plug Detect

Since the HDCP with FPD-Link III authentication exchange occurs on the same channel as the video display data, there is no exchange until both FPD-Link III transmitter and receiver are connected and synchronized. This synchronization happens automatically whenever an HDCP enabled FPD-Link III transmitter and receiver are connected and enabled. This simple and robust automatic synchronization is important in automotive infotainment applications.

### 2.2.2 Receiver Lock Detect

Receiver Lock Detect (RLD) is asserted to indicate that the FPD Link III receiver is synchronized and locked to the upstream FPD Link III transmitter.

HDCP enabled FPD-Link III transmitters includes all necessary functionality for supporting Receiver Lock Detect. After asserting the RLD, the HDCP enabled FPD-Link III transmitter is ready to start the authentication protocol upon the request of the upstream content control system.

## 2.3 HDCP Receiver with FPD-Link III State Diagram

HDCP enabled FPD-Link III receivers implement the HDCP Receiver Authentication State Diagram per HDCP Rev1.4 specification. The receiver is capable of Fast Re-authentication.

## 2.4 HDCP Repeater with FPD-Link III State Diagram

HDCP-FPD-Link III repeaters implement the HDCP Rev1.4 Repeater State Diagrams with only one slight change as shown in a simplified HDCP Repeater Downstream Link State Diagram of Figure 3. The HDCP Repeater Downstream Authentication Protocol State Diagram is implemented per HDCP Rev1.4 specification.

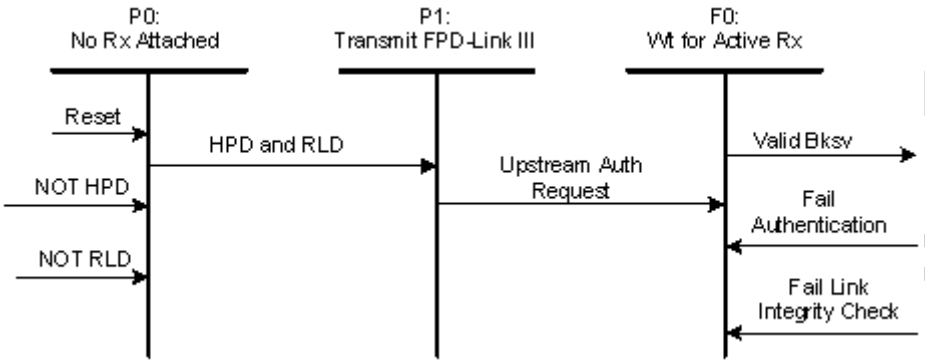



**Figure 3. HDCP Repeater with FPD-Link III State Diagram**

**Transition Any State: P0.** Reset conditions at the HDCP repeater or loss of Hot Plug Detect (HPD) or loss of Receiver Lock Detect (RLD) cause the HDCP repeater to enter the No Receiver Attached state for this port.

**Transition P0:P1.** The detection of Hot Plug Detect and Receiver Lock Detect (RLD) indicate that a sink device is attached.

**Transition P1:F0.** Upon an upstream authentication request, the HDCP repeater waits for the availability of an active HDCP receiver on this port.

## 2.5 HDCP FPD-Link III Port

The HDCP FPD-Link III authentication protocol occurs on a bidirectional control channel embedded in the FPD-Link III forward and back channel streams from the source to the sink. This digital control channel operates equivalently to a standard I2C bus although transfers occur at a much higher rate. While external I2C transactions may also be transferred in either direction on the bidirectional control channel, the HDCP transactions are maintained internal to the link between the HDCP transmitter and HDCP receiver and are not visible at external interfaces. Authentication read and write operations over the HDCP-FPD-Link III control channel must complete within 250 µs per byte transferred.

The HDCP FPD-Link III supports the HDCP Port addresses 0x74 and 0x76 because the embedded control channel appears as an I2C bus to the HDCP system.

All of the HDCP Rev1.4 defined registers are supported with the HDCP FPD-Link III Port implementation. The following lists differences in bit definitions for the HDCP FPD-Link III devices.

- Ainfo ENABLE_1.1_FEATURES enable the Advanced Cipher and EESS modes of operation. Unlike HDMI, EESS is not automatically enabled.
- Bcaps HDMI_RESERVED is always set to a 0.
- Bcaps FAST is set to a 0 by the HDCP receiver and ignored by the HDCP transmitter.
- Bcaps 1.1_FEATURES bit is always set to 1.
- Bcaps FAST_REAUTHENTICATION is always set to a 1.
- Bstatus HDMI_MODE is set to a 0 by the HDCP receiver and ignored by the HDCP transmitter

In addition to the default registers as specified in the HDCP Rev1.4, HDCP FPD-Link III supports additional control bits in an implementation specific register at address 0xC0 (HDCP_DBG). This register includes FAST_LV (Fast Link Verification) bit. When set, the link verification is accelerated by a factor of 8 to provide faster detection of loss of synchronization

## 2.6 HDCP FPD-Link III Encryption Status Signaling

HDCP FPD-Link III supports both the Original Encryption Status Signaling (OESS) and Enhanced Encryption Status Signaling (EESS) defined in the HDCP Rev1.4 specification. The signaling method is different from the definition in the specification as the CTLx signals are not supported in HDCP FPD-Link III devices. Instead the ENC_EN and ENC_DIS controls are embedded directly in the video data during the vertical blanking interval (VBLANK).

Upon detection of the start of VBLANK, the HDCP FPD-Link III transmitter may send two control codes on the video data signals. In the first control slot, the HDCP FPD-Link III transmitter can enable or disable the HDCP AVMUTE state using the Set_AVMUTE / Clear_AVMUTE control codes. Setting the AVMUTE state allows maintaining HDCP authentication during periods when the pixel clock may be changing or unstable. The AVMUTE controls is only sent when the device is in EESS mode. During the second control slot, the HDCP transmitter embeds a code on the video signals to indicate the next frame should be encrypted. During OESS, if encryption is not desired for a frame, no code is sent. During EESS, if encryption is not desired for a frame, the ENC_DIS code is sent. The HDCP FPD-Link III transmitter does not signal ENC_EN or ENC_DIS during the AVMUTE state.

**Table 1. Encryption and AVMUTE Signal Codes**

| CODE | VALUE | DESCRIPTION |
|---|---|---|
| ENC_EN | 0x999999 | Encryption is enabled for this frame. |
| ENC_DIS | 0x111111 | Encryption is disabled for this frame |
| SET_AVMUTE | 0x666666 | Begin AVMUTE state |
| CLEAR_AVMUTE | 0x555555 | Exit AVMUTE state |

Detection of the vertical blanking interval normally occurs at the active edge of the VSYNC signal. In the absence of signaling on the VSYNC signal, the HDCP enabled FPD-Link III transmitter detects a VBLANK interval when a blank interval occurs that is longer than twice the previous Video Data Period (pixel line length).

## 2.7 Data Encryption

The HDCP FPD-Link III transmitter applies encryption to the video data stream prior to sending the data to the FPD-Link III encoder. The encryption is handled by a bit wise exclusive-or function with a 24-bit pseudo-random data stream provided by the HDCP Cipher.

For an HDCP enabled FPD-Link III system, the FPD-Link III encoder and decoder replace the T.M.D.S. encoder and decoder.

By default, HDCP enabled FPD-Link III devices transport encrypted audio data during data island periods. Optionally, HDCP-enabled FPD-Link III devices may send non-encrypted audio data in an embedded channel over the serial link.

### 2.7.1 Encryption / Decryption State Diagrams

HDCP enabled FPD-Link III devices implement the OESS State Diagram per the HDCP Rev1.4 specification.

HDCP enabled FPD-Link III also implement the EESS State Diagram per the HDCP Rev1.4 specification. When the application requires audio encryption, HDCP enabled FPD-Link III devices transport the encrypted audio during the Data Island period. HDCP enabled FPD-Link III devices do not use Guard Bands. AVMUTE operation is supported.

## 2.8 HDCP Cipher

The HDCP Cipher function is implemented per Section 4 of the HDCP Rev1.4 specification. All operating modes of the HDCP Cipher are supported and implemented per the specification.

### 2.9 Renewability

HDCP enabled FPD-Link III devices implement renewability per HDCP Rev1.4 specification.

The HDCP specification calls for verification of Device Private Keys. Digital Content Protection LLC provides system renewability messages (SRMs) which are delivered with content. The SRMs carry a KSV revocation list that indicates KSVs that have been revoked. The HDCP transmitters do not directly check KSVs against the revocation list. Instead, the HDCP transmitters provide the receiver KSV and, in the case of a downstream repeater, a KSV list. Software on an attached host controller must validate the KSV list against the revocation list prior to sending protected content to the HDCP transmitter.

## 3 Typical HDCP with FPD-Link III Applications

HDCP with FPD-Link III applications may range from basic non-repeater application involving a single audio/video source of content and a single display device to more complex HDCP systems involving multiple HDCP repeaters and display, internally or externally controlled.

### 3.1 HDCP Non-repeater Application with FPD-Link III

A basic HDCP with FPD-Link III application consists of an audio/video source of content and a single display device as shown in Figure 4.


**Figure 4. HDCP Non-repeater Application**

HDCP enabled FPD-Link III devices are listed in Table 2.

**Table 2. HDCP Enabled FPD-Link III Devices**

| HDCP Enabled FPD-Link III Transmitters | HDCP Enabled FPD-Link III Receivers |
|---|---|
| DS90UH925Q-Q1<br>DS90UH927Q-Q1<br>DS90UH929-Q1<br>DS90UH941AS-Q1<br>DS90UH947-Q1<br>DS90UH949-Q1<br>DS90UH949A-Q1 | DS90UH926Q-Q1<br>DS90UH928Q-Q1<br>DS90UH948-Q1<br>DS90UH940-Q1<br>DS90UH940N-Q1<br>DS90UH949A-Q1 |

HDCP control and authentication of HDCP non-repeater systems is detailed in Section 8.

### 3.2 Internally Controlled HDCP Repeater Application with FPD-Link III

A typical HDCP repeater application consists of an audio/video source of content, one or more levels of HDCP repeaters, and multiple display devices. The HDCP repeater requires, at a minimum, an HDCP receiver and at least one HDCP transmitter. The HDCP repeater could also include a local display device. Figure 5 and Figure 6 show typical internally controlled HDCP repeater applications.

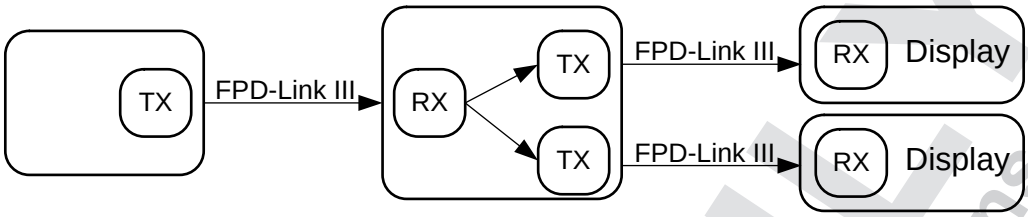



**Figure 5. Internally Controlled 1:2 HDCP Repeater**

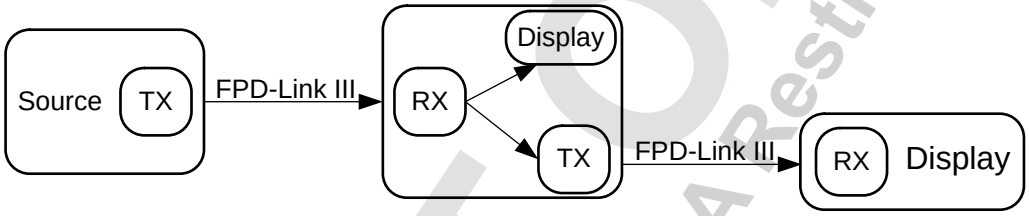



**Figure 6. Internally Controlled 1:2 HDCP Repeater with Local Display**

HDCP enabled FPD-Link III devices supporting repeater implementations are listed in Table 3.

**Table 3. FPD-Link III Devices Supporting HDCP Repeater Implementations**

| Video Interface | HDCP Enabled FPD-Link III Transmitters | HDCP Enabled FPD-Link III Receivers |
|---|---|---|
| RGB | DS90UH925Q-Q1 | DS90UH926Q-Q1 |
| OpenLDI | DS90UH927Q-Q1<br>DS90UH947-Q1 | DS90UH928Q-Q1<br>DS90UH948-Q1 |

HDCP control and authentication of HDCP repeater systems is detailed in Section 8.

### 3.3 Externally Controlled HDCP Repeater Application with FPD-Link III

An HDCP repeater provides a mechanism to extend HDCP transmission over multiple links to multiple display devices. The HDCP repeater provides a mechanism to authenticate all HDCP receivers in the system and distribute protected content to the HDCP receivers using the encryption mechanisms provided in the HDCP specification.

When bridging directly between the FPD-Link III devices that can support HDCP repeaters, an internal repeater controller in the HDCP enabled FPD-Link III receivers (Table 3) can be used to handle the HDCP repeater operation. In systems without a direct connection of the full video stream, an external controller may be required to handle the HDCP repeater functions. An example of a system without the direct connection may include systems that process the video in an FPGA or ASIC before sending downstream as shown in Figure 7.

The authentication procedure requires the HDCP repeater to gather the Key Selection Vector (KSV) for each downstream HDCP receiver and compile this into a list. The KSV list is then made available to the upstream HDCP transmitter to provide the KSV list to the content source for a check against the HDCP revocation list provided with the protected content. All KSVs must be valid for the source to send protected content over the link. In External repeater mode, a local controller provides the interface between the DS90UH948-Q1 HDCP receiver and the FPD-Link III transmitter to control downstream authentication and provide authentication results to the HDCP receiver.

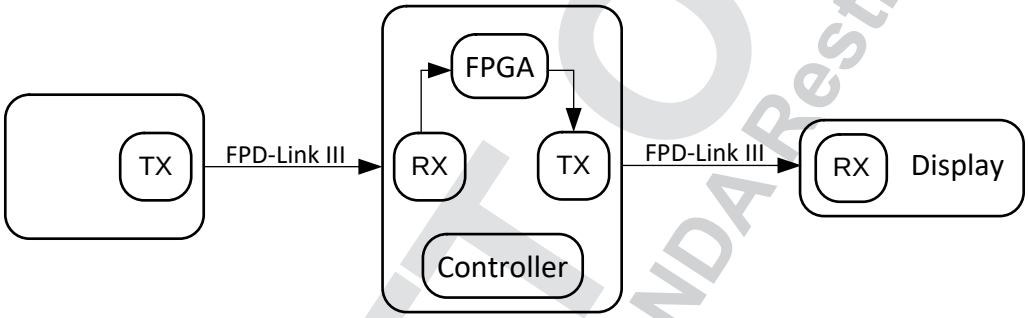



**Figure 7. Externally Controlled 1:1 HDCP Repeater with FPGA**

# 4 Internally Controlled HDCP Repeater with FPD-Link III Implementation

HDCP enabled FPD-Link III devices supporting repeater implementations are listed in Table 3.

For supporting HDCP repeater operation, the HDCP enabled FPD-Link III deserializer includes the ability to control the downstream authentication process, assemble the KSV list for downstream HDCP receivers, and pass the KSV list to the upstream HDCP transmitter. An I2C master proxy within the HDCP deserializer communicates with the I2C slave within the HDCP enabled serializer. The serializer handles authentication with downstream HDCP receivers and makes status available through the I2C interface. The HDCP enabled deserializer monitors the transmit port status for each serializer and reads downstream KSV and KSV list values. In this document, the HDCP enabled deserializer is often referred to as the HDCP transmitter or transmit port. The HDCP enabled deserializer is often referred to as the HDCP receiver.

In addition to the I2C interface used for controlling the authentication process, the HDCP repeater implementation includes two other interfaces. A video interface provides the unencrypted video data and includes the DE/VS/HS control signals. In addition to providing the video data, the video interface communicates control information and packetized audio data during video blanking intervals. A separate I2S audio interface may optionally be used to send I2S audio data between the HDCP receiver and HDCP transmitter in place of using the packetized audio over the video interface. All audio and video data is decrypted at the output of the HDCP receiver and is re-encrypted by the HDCP transmitter.

Figure 8 shows a block diagram of a 1:2 HDCP repeater implementation with FPD-Link III devices.

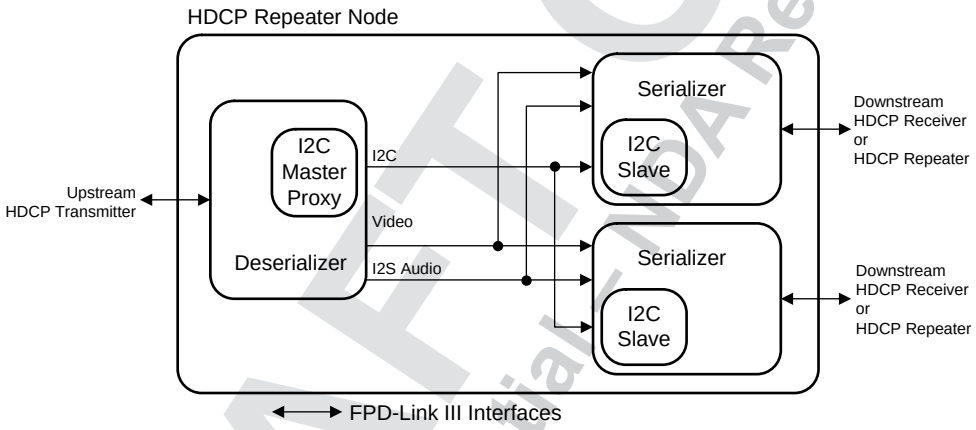



**Figure 8. Internally Controlled 1:2 HDCP Repeater with FPD-Link III Implementation**

### 4.1 HDCP Repeater with FPD-Link III Connections

The repeater requires the following connections between the HDCP enabled deserializer and each HDCP enabled serializer as shown in Figure 9.

1. Video data – Connect all video data and clock signals.
2. I2C – Connect SCL and SDA signals.
3. Audio (optional) – Connect I2S_CLK, I2S_WC, and I2S_Dx signals.
4. IDX pin – Each transmitter and receiver must have an unique I2C address.
5. REPEAT & MODE_SEL pins — All transmitters and receivers must be set into repeater mode.
6. Interrupt pin – Connect deserializer INTB_IN pin to the serializer INTB pin

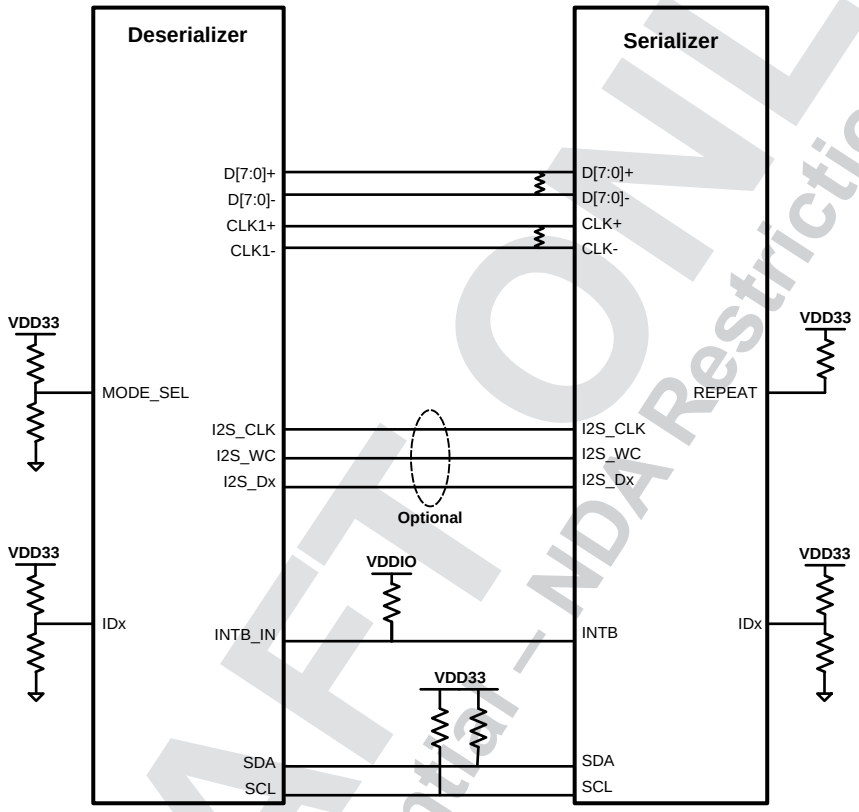



**Figure 9. HDCP Repeater with FPD-Link III Connection Diagram**

### 4.1.1 Maximum Repeater Application

Figure 10 shows the maximum configuration supported for repeater implementations. Two levels of repeaters are supported with a maximum of three transmitters per receiver. In the repeater implementation, the HDCP enabled serializer is the transmitter (TX), and the HDCP enabled deserializer is the receiver (RX). The source transmitter (TX) may be any HDCP capable FPD-Link III for IVI serializer while the display receiver (RX) may be any HDCP capable FPD-Link for IVI deserializer.

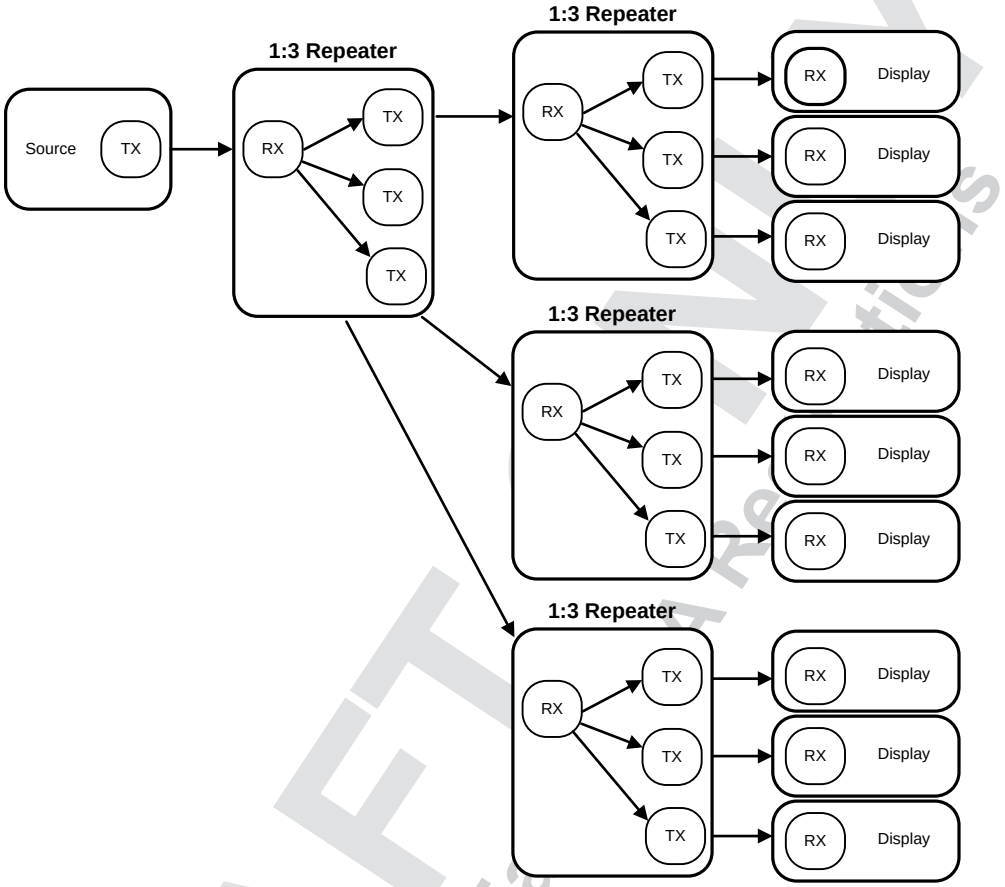



**Figure 10. Maximum Repeater Application**

## 4.2 HDCP Repeater Authentication Control

An HDCP repeater consists of one HDCP receiver and some number of HDCP transmitters. The local controller would interface between the HDCP receiver and the HDCP transmitters in the repeater.

The HDCP repeater functions are divided between the HDCP receiver and HDCP transmitter. The HDCP transmitter handles authentication with a downstream HDCP receiver. The HDCP receiver receives upstream authentication requests from the upstream HDCP transmitter and returns the repeater authentication status and KSV List.

### 4.2.1 HDCP Receiver Operation

The HDCP receiver is responsible for detecting, controlling, and monitoring the HDCP transmitters for proper HDCP authentication. In addition, the HDCP receiver gathers KSV and KSV Lists from the HDCP transmitters, compiles them into a single KSV List, and makes the list available to the upstream HDCP transmitter.

In addition to providing the KSV List, the HDCP receiver provides status that indicates authentication status to the upstream transmitter, the number of attached transmitters, and an indication of downstream Hot Plug Detection (HPD). The downstream HPD is indicated by sending a special bit in the control channel. The HPD indication is latched high and cleared on start of Authentication (write of the transmitter KSV value).

The HDCP receiver supports up to three HDCP transmitters, referred to as ports. When strapped for repeater operation, the HDCP receiver waits for an authentication request from upstream. Upon receiving the authentication request, the HDCP receiver performs the following tasks:

- Poll the local I2C bus for attached HDCP transmitters (ports)
- Configure each transmit port
- Check each port status for Receive Lock Detect to determine if a downstream receiver is present
- Enable HDCP Authentication for ports reporting Receive Lock Detect
- Continue to check status of each port until all ports are either unattached (no Receive Lock) or Authenticated.
- Read receiver KSV values for each authenticated Transmit port
- Read KSV List from any downstream repeaters
- Compile KSV and KSV List values into a single KSV list for passing upstream through the KSV FIFO register
- Generate SHA-1 integrity check value for KSV List
- Indicate Ready in BCAPS register and topology information in BSTATUS registers for reading by upstream HDCP transmitter
- The BSTATUS, KSF FIFO, and Vprime register values are made available for reading by the upstream HDCP transmitter

#### 4.2.1.1 HDCP Receiver – Poll for Transmit Ports

After HDCP authentication is enabled, the HDCP receiver polls over the I2C interface to attempt to detect HDCP capable transmitters. The HDCP capable transmitters include a signature at register offset 0xF0 of the form “_UH9xx” where xx is typically “47” for the DS90UH947-Q1 HDCP Serializer. The HDCP receiver supports up to three attached HDCP transmit ports.

The HDCP receiver steps through each applicable I2C slave address and searches for devices. Applicable I2C addresses are limited to the addresses to which the HDCP transmitters may be strapped by the IDX pin.

#### 4.2.1.2 HDCP Receiver – Downstream Authentication

After polling for HDCP Transmit ports, the HDCP receiver reads each port status to determine if the port is attached to a downstream receiver. If so, the HDCP receiver enables authentication for the transmit port. The HDCP receiver then monitors status for all transmit ports until all ports are either unattached (no RX Lock Detect) or authenticated.

During this time, a five second timer is enabled. If all ports do not reach the unattached or authenticated states within five seconds, the HDCP receiver returns to an unauthenticated state and waits for a new request from upstream.

#### 4.2.1.3 HDCP Receiver – Assemble KSV List

Once all downstream ports are either authenticated or unattached, the local controller would compile the KSV List for all downstream devices. For each authenticated port, the local controller would do the following.

- Read the Bksv for the downstream receiver using the RX_BKSV registers in the HDCP transmitter. The Bksv value is appended to the KSV List.
- Read the Bcaps value for the downstream receiver using the RX_BCAPS register in the HDCP transmitter

If the Bcaps register indicates the downstream device is an HDCP repeater, the HDCP receiver also performs the following.

- Read the Bstatus registers for the downstream repeater using the TX_BSTATUS registers in the HDCP transmitter
- Update local DEPTH and DEVICE_COUNT values based on values in the ports Bstatus registers
- Read downstream repeater KSV List from the KSV FIFO register in the HDCP transmitter. Downstream KSV List is appended to the local KSV list.
- Set KSV_LIST_VALID bit in the HDCP_CTL register of the FPD-Link III transmitter to allow downstream authentication to complete.

Once the KSV list is completed, the HDCP receiver computes a SHA-1 checksum. The SHA-1 is computed over the KSV List, local Bstatus, and the secret value M0 from the HDCP cipher. The HDCP receiver then asserts the READY flag in its Bcaps register to indicate completion status to the upstream HDCP transmitter.

During assembly of the KSV List, the HDCP receiver also determines the total DEPTH and DEVICE_COUNT values. If the DEPTH because greater than 7 or the DEVICE_COUNT becomes greater than 60, the HDCP receiver stops assembling the KSV List, assert the appropriate error flag, MAX_CASCADE_EXCEEDED or MAX_DEPTH_EXCEEDED, in its Bstatus register, and set the Bcaps:Ready flag. This allows the upstream HDCP transmitter to detect and report the topology error. Note that the logical maximum topology values are significantly larger than the maximum physical limits for the DS90UH94x devices.

#### 4.2.1.4 HDCP Receiver – Authenticated State

Once authenticated, the HDCP receiver periodically polls the known transmit ports to monitor their current status. Status polling is typically done every 128 ms. Per the HDCP specification, detection of an unauthenticated transmit port transitions the HDCP receiver to an unauthenticated state. Detection of a downstream hot-plug event also causes loss of authentication.

The HDCP receiver also periodically polls for new HDCP transmitters, typically at a rate of once every second. It performs this by polling for additional HDCP transmitters as described previously. If any new ports are detected, their status is checked. If the status indicates the ports have active downstream receivers, authentication is lost.

#### 4.2.1.5 HDCP Receiver – Restart Authentication

Upon receipt of a new KSV value from the upstream HDCP transmitter, the HDCP receiver restarts authentication. The restart occurs regardless of current authenticated state.

#### 4.2.1.6 HDCP Receiver – Hot Plug Detect Propagation

The HDCP receiver is capable of propagating a downstream Hot Plug detection signal to the upstream HDCP transmitter. The Hot Plug detection is asserted for the following two conditions:

- Assertion of RX Lock Detect by a transmit port in the HDCP repeater
- Assertion of a downstream Hot Plug detect by a transmit port in the HDCP repeater

Both of these conditions are indicated on the DOWN_HPD bit in the HDCP_STS register of the FPD-Link III transmitter. When this occurs, the HDCP transmitter automatically transitions to an unauthenticated state to resume authentication.

When a Hot Plug detection occurs, the indication is passed to the upstream HDCP transmitter via an embedded signal in the reverse communications channel over the serial link. When the Hot Plug indication reaches the HDCP transmitter at the source, the controller should detect the event and restart HDCP authentication.

### 4.2.2 HDCP Transmitter Operation

The HDCP transmitter in an HDCP repeater operates essentially identically to an HDCP transmitter located at the source of the video data. There are two main differences related to starting authentication and validation of the downstream HDCP receiver. The following sections describe those differences as well as some of the transmitter options.

The HDCP transmitter operation is implemented based on the HDCP Downstream Authentication Protocol State Diagram as defined in Figure 2-9 of the HDCP Specification.

#### *4.2.2.1 HDCP Transmitter – Start of Authentication*

To start authentication, an HDCP transmitter at the source waits for an indication that content protection is desired. In the DS90UH947-Q1, that corresponds to setting the HDCP_EN bit in the HDCP_CTL register. In an HDCP repeater, the downstream authentication starts when a request comes from the upstream HDCP transmitter. While the indication is different for the HDCP repeater, the actual implementation is the same for the HDCP transmitter. In an HDCP repeater, the downstream process begins when the local controller sets the HDCP_EN bit in the HDCP transmitter's HDCP_CTL register.

#### *4.2.2.2 HDCP Transmitter – Validation of HDCP receiver*

To validate that HDCP receiver, the HDCP transmitter at the source should validate that the Bksv (downstream HDCP receiver KSV) has twenty ones and twenty zeros. In addition, it must confirm that the KSV is not on the revocation list provided with the protected content. Since a HDCP repeater does not have access to the revocation list, it bypasses that step and immediately proceeds to test for downstream repeater. To bypass the KSV validation, the local controller must set the KSV_VALID bit in the HDCP_CTL register. Similarly, the HDCP repeater does not validate a downstream KSV List against the revocation list, but proceeds to the Authenticated state. After the KSV List has been read by the local controller, it should set the KSV_LIST_VALID bit in the HDCP_CTL register.

Note that the source should validate all KSVs in the system before allowing the transmission of protected content.

#### *4.2.2.3 HDCP Transmitter – Fast Link Verification*

The FPD-Link III transmitter periodically checks the synchronization between the HDCP transmitter and HDCP receiver. If synchronization is lost, the HDCP transmitter must re-initiate authentication to restore synchronization. The FPD-Link III transmitters support an optional mode to check the HDCP synchronization more often than the standard rate. For normal link verification, the HDCP specification calls for checking the Ri value every 128 frames. Setting the FAST_LV control speeds that up to every 16 frames. In Enhanced Link Verification, the Pj value is checked every 16 frames. Setting the FAST_LV control speeds that up to every 2 frames.

By more quickly detecting loss of synchronization, FPD-Link III chipset allows for much shorter loss of video signal to the display upon the loss of synchronization.

Configuration of Fast Link Verification mode is done through setting the FAST_LV bit in the HDCP_DBG register at the FPD-Link III transmitter. This option is automatically passed to the FPD-Link III transmitters in downstream FPD-Link III repeaters.

#### 4.2.2.4 FPD-Link III Transmitter Encryption Modes

The HDCP transmitter includes four modes for enabling encryption. The modes are enabled by setting the ENC_MODE bits in the HDCP_CFG register.

**Table 4. HDCP Encryption Enable Modes**

| MODE | ENC_MODE | DESCRIPTION |
|---|---|---|
| Enc_Authenticated | 00 | Prior to authentication, encryption will be based on register control. When authenticated, the HDCP_ENC_EN bit will be automatically set, and video data will be encrypted. The register control may be overridden by software while authenticated or if authentication is lost. |
| Enc_Reg_Control | 01 | Register control enables or disables encryption. Prior to sending protected content, an attached controller must enable encryption. |
| Enc_Always | 10 | All video data will be encrypted when authenticated. If not authenticated, a blue screen will be transmitted. |
| Enc_InBand_Control | 11 | Encryption controls are passed to the transmitter on the video data interface during the vertical blanking interval as described in the Encryption Status Signaling section. This mode may be used in a repeater configuration. |

Register control of encryption is through the HDCP_ENC_EN and HDCP_ENC_DIS bits in the HDCP_CTL register.

Only the Enc_Inband_Control should not be used for the Externally controlled repeater. If using Enc_Reg_Control, the controller needs to monitor the RX_ENCRYPTED status in the XRPTR_STS register to determine when to enable and disable downstream encryption.

# 5 Externally Controlled Repeater Implementation

HDCP enabled FPD-Link III devices supporting repeater implementations are listed in Table 3.

For implementing an HDCP repeater, a micro-controller may be used to control the downstream authentication process, assemble the KSV list for downstream HDCP receivers, and pass the KSV list to the upstream HDCP transmitter. An I2C master within the micro-controller communicates with the I2C slave within the FPD-Link III transmitter (i.e. DS90UH947-Q1). The FPD-Link III transmitter handles authentication with a downstream HDCP receiver and makes status available through the I2C interface. The micro-controller monitors the transmit port status for each FPD-Link III transmitter and reads downstream KSV and KSV list values from the FPD-Link III transmitter.

In addition to the I2C interface used to control the authentication process, the bridge or repeater implementation includes two other interfaces. A video interface provides the unencrypted video data and includes the DE/VS/HS control signals. A separate I2S audio interface may be used to send I2S audio data between the FPD-Link III receiver and FPD-Link III transmitter. All audio and video data is decrypted at the output of the FPD-Link III receiver and is re-encrypted by the FPD-Link III transmitter.

In a direct FPD-Link III repeater, control information and packetized audio data are passed using the video data signals during video blanking intervals. In the external repeater implementation, that control is not available, so audio must be passed through I2S and control information is all handled through the I2C control.

Figure 11 provides a more detailed block diagram of a 1:1 HDCP repeater with FPGA.

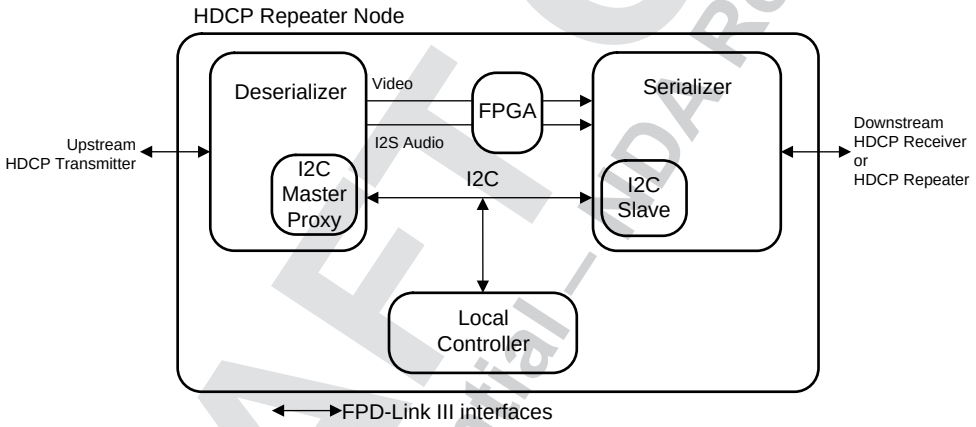



**Figure 11. Externally Controlled 1:1 Repeater with FPGA Implementation**

## 5.1 Configuration of Repeater Operation

Configuration of the FPD-Link III transmitter handled is in two ways. Some functions are set by strap options on the device pins, while other functions are controlled by the local controller.

### 5.1.1 Strap Options

See the device data sheet for details of strap options. Note that FPD-Link III transmitter should not be strapped for repeater mode operation as that should only be done for the internally controlled repeater application.

### 5.1.2 Configuration Through Local Controller

The local controller is responsible for initializing the HDCP options for the FPD-Link III transmitter.

#### 5.1.2.1 HDCP Transmit Debug Register (HDCP_DBG), Address 0xC0

The HDCP transmit debug register allows configuration of several options for the HDCP authentication. These register values are passed to the downstream HDCP receiver at the start of HDCP authentication.

**Table 5. HDCP Transmit Debug Register (HDCP_DBG), Address 0xC0**

| BIT | BIT NAME | DEFAULT | DESCRIPTION |
|---|---|---|---|
| 7 | Reserved | 0, RW | Reserved |
| 6 | HDCP_I2C_TO_DIS | 0, RW | HDCP I2C Timeout Disable:<br>Setting this bit to a 1 disables the bus timeout function in the HDCP I2C master. When enabled, the bus timeout function allows the I2C master to assume the bus is free if no signaling occurs for more than 1 second. |
| 5 | FORCE_RI_ERR | 0, RW/SC | Force Ri Synchronization Error:<br>Forces an Ri synchronization error by causing the HDCP transmitter to not count a frame. Allows checking of the Ri synchronization process. This bit is self-clearing. |
| 4 | DIS_RI_SYNC | 0, RW | Disable Ri Synchronization check:<br>Ri is normally checked both before and after the start of frame 128. The check at frame 127 ensures synchronization between the two. Setting this bit to a 1 disables the check at frame 127. |
| 3 | RGB_CKSUM_EN | 0, RW | Enable RBG video line checksum:<br>Enables sending of ones-complement checksum for each 8-bit RBG data channel following the end of each video data line. |
| 2 | FAST_LV | 0, RW | Fast Link Verification:<br>HDCP periodically verifies that the HDCP receiver is correctly synchronized. Setting this bit increases the rate at which synchronization is verified. When set to a 1, Pj is computed every 2 frames and Ri is computed every 16 frames. When set to a 0, Pj is computed every 16 frames and Ri is computed every 128 frames. |
| 1 | TMR_SPEEDUP | 0, RW | Timer Speedup:<br>Speeds up HDCP authentication timers. |
| 0 | HDCP_I2C_FAST | 0, RW | HDCP I2C Fast mode Enable:<br>Setting this bit to a 1 enables the HDCP I2C master in the HDCP receiver to operate with the Fast mode timing. If set to a 0, the I2C master operates with Standard mode timing. This bit is mirrored in the IND_STS register. |

#### 5.1.2.2 HDCP Transmit Configuration Register (HDCP_CFG), Address 0xC2

The HDCP Transmit Configuration Register allows configuration of several options for the HDCP authentication. Only the HDCP_EESS bit is propagated from the HDCP transmitter to downstream HDCP repeaters. All other assigned configuration bits are listed in Table 6.

**Table 6. HDCP Transmit Configuration Register (HDCP_CFG), Address 0xC2**

| BIT | BIT NAME | DEFAULT | DESCRIPTION |
|---|---|---|---|
| 7 | ENH_LV | 1, RW | Enable Enhanced Link Verification:<br>Enables enhanced link verification. Allows checking of the encryption Pj value on every 16th frame.<br>1 = Enhanced Link Verification enabled<br>0 = Enhanced Link Verification disabled |
| 6 | HDCP_EESS | 0, RW | Enable Enhanced Encryption Status Signaling:<br>Enables Enhanced Encryption Status Signaling (EESS) instead of the Original Encryption Status Signaling (OESS).<br>1 = EESS mode enabled<br>0 = OESS mode enabled |
| 5 | TX_RPTR | 0, RW | Transmit repeater Enable:<br>Transmit repeater does not need to be enabled for the External Controlled repeater. |

**Table 6. HDCP Transmit Configuration Register (HDCP_CFG), Address 0xC2 (continued)**

| BIT | BIT NAME | DEFAULT | DESCRIPTION |
|---|---|---|---|
| 4:3 | ENC_MODE | 00, RW | Encryption Control Mode:<br>Determines mode for controlling whether encryption is required for video frames.<br>00 = Enc_Authenticated<br>01 = Enc_Reg_Control<br>10 = Enc_Always<br>11 = Enc_InBand_Control (per frame)<br>Enc_Inband_Control should not be used for the External Controlled repeater. |
| 2 | WAIT_100MS | 0, RO | Enable 100MS Wait:<br>The HDCP 1.3 specification allows for a 100-ms wait to allow the HDCP receiver to compute the initial encryption values. The FPD-Link III implementation ensures that the receiver will complete the computations before the HDCP transmitter. Thus the timer is unnecessary. To enable the 100-ms timer, set this bit to 1. |
| 1 | RX_DET_SEL | 0, RW | RX Detect Select:<br>Controls assertion of the receiver Detect Interrupt. If set to 0, the receiver Detect Interrupt will be asserted on detection of an FPD-Link III receiver. If set to 1, the receiver Detect Interrupt will also require a receive lock indication from the receiver. |
| 0 | HDCP_AVMUTE | 0, RW | Enable AVMUTE:<br>Setting this bit to a 1 will initiate AVMUTE operation. The transmitter will ignore encryption status controls while in this state. If this bit is set to 0, normal operation will resume. This bit may only be set if the HDCP_EESS bit is also set. |

## 5.2 HDCP Repeater Authentication Control

An HDCP repeater consists of one HDCP receiver and some number of HDCP transmitters. The local controller would interface between the FPD-Link III receiver and the FPD-Link III transmitters in the repeater.

The HDCP repeater functions are divided between the FPD-Link III receiver, local controller, and FPD-Link III transmitter. The FPD-Link III transmitter handles authentication with a downstream HDCP receiver. The FPD-Link III receiver receives upstream authentication requests from the upstream HDCP transmitters and returns the repeater authentication status and KSV List. The local controller interfaces to each of the FPD-Link III transmit ports and provides status and the KSV List information to the FPD-Link III receiver.

### 5.2.1 HDCP Receiver External repeater Control and Status

The HDCP receiver in the DS90UH948-Q1 provides control and status through the I2C interface. Upstream status is provided by the XRPTR_STS register. Control is provided by the XRPTR_CTL register. Additional registers provide the mechanism for returning status (XRPTR_BSTS0/1 registers) as well as the list of KSVs for downstream receivers (KSV_FIFO_DATA and KSV_FIFO_ADDR0/1 registers).

When the upstream HDCP transmitter requests authentication, it sends the transmitter KSV value to the HDCP receiver. When this occurs, the KSV_WRITTEN bit in the XRPTR_STS register is set to a 1. This indicates to the external repeater controller that authentication process is requested.

A second bit in the XRPTR_STS register indicates that encryption is enabled on the upstream link. This flag should be used to control downstream encryption, unless downstream links are set to always encrypt data.

External-controlled repeater operation is enabled by setting the XRPTR_ENABLE bit in the XRPTR_CTL register.

**NOTE:** The DS90UH948-Q1 can be strapped to either normal or repeater mode of operation. Normal mode is preferred as it prevents the internal repeater controller from turning on at power-up.

In addition the XRPTR_ENABLE bit, TI also recommends setting the XRPTR_NO_INBAND bit in the XRPTR_CTL register. This prevents the HDCP receiver from sending the control signals and audio on the video data interface.

#### 5.2.1.1 Upstream Control of HDCP Receiver

The upstream control of the HDCP receiver consists of the following:

1. Monitor the DS90UH948-Q1 XRPTR_STS register to determine when authentication is requested through the KSV_WRITTEN bit.
2. When KSV_WRITTEN is detected, the external controller should start downstream authentication, if not previously started.
3. Once downstream authentication is complete, the external controller should do the following:
   - Update the XRPTR_BSTS0 and XRPTR_BSTS1 register with proper information for the downstream connection. If topology errors have occurred they should be reported in these registers.
   - Compile the downstream KSV List and write the KSVs into the KSV FIFO.
   - Set the XRPTR_LIST_RDY bit in the XRPTR_CTL register. This enable the DS90UH948-Q1 to compute the V` checksum value and indicate Ready to the upstream HDCP transmitter.

In addition to the KSV_WRITTEN status, the XRPTR_STS register also provides an indication, through the RX_ENCRYPTED status bit, that the HDCP receiver is recovering encrypted data.

#### 5.2.1.2 Externally Controlled Repeater KSV FIFO Interface

As part of the Externally controlled repeater operation, the external repeater controller should write the KSV values for all downstream devices into the KSV FIFO. The KSVs may be written to the KSV FIFO using the KSV_FIFO_DATA, and KSV_FIFO_ADDRx registers.

Prior to writing the KSV List to the KSV FIFO, the external controller should first set the KSV_FIFO_ADDR to 0, to point to the first location in the KSV_FIFO. The controller should then write each KSV to the KSV_FIFO_DATA register starting with the least significant byte. Additional KSVs should be written to the KSV_FIFO_DATA register in the same manner. The KSV_FIFO_ADDR automatically increments for each write, so there is no need to set the KSV_FIFO_ADDR other than for the first write to the KSV_FIFO.

For example, the following process should be used to write a single KSV to the KSV FIFO:

1. Write 8’h00 to register 8’hCA (KSV_FIFO_ADDR0)
2. Write 8’h00 to register 8’hCB (KSV_FIFO_ADDR1)
3. Write KSV0 Byte 0 to 8’hC9 (KSV_FIFO_DATA)
4. Write KSV0 Byte 1 to 8’hC9 (KSV_FIFO_DATA)
5. Write KSV0 Byte 2 to 8’hC9 (KSV_FIFO_DATA)
6. Write KSV0 Byte 3 to 8’hC9 (KSV_FIFO_DATA)
7. Write KSV0 Byte 4 to 8’hC9 (KSV_FIFO_DATA)

### 5.2.2 Local Controller Operation

To support HDCP repeater operation, the local controller is responsible for detecting, controlling, and monitoring the FPD-Link III transmitters for proper HDCP Authentication. In addition, the local controller would gather KSV and KSV Lists from the FPD-Link III transmitters, compile them into a single KSV List, and pass the values to the FPD-Link III receiver.

In addition to providing the KSV List, the local controller provides authentication status to the FPD-Link III receiver including the number of attached transmitters, and an indication of downstream Hot Plug Detection (HPD).

Depending on the repeater implementation, the controller may support more than a single downstream transmitter (or port) and would need to compile the status and KSVs for each downstream port into a single repeater status and KSV List.

The local controller should monitor the FPD-Link III receiver for an authentication request. Upon receiving the authentication request, the local controller would perform the following tasks.

- Poll the local I2C bus for attached FPD-Link III transmitters. Polling is not required if transmitter addresses are known.
- Configure each Transmit port

- Check each port status for Receive Lock Detect to determine if a downstream receiver is present
- Enable HDCP Authentication for ports reporting Receive Lock Detect
- Continue to check status of each port until all ports are either unattached (no Receive Lock) or Authenticated.
- When receiver KSV is ready, read receiver KSV value for each authenticated Transmit port and set the port's KSV Valid control.
- Read KSV List from any downstream repeaters. Following KSV List read, set the KSV_LIST_VALID bit in the HDCP_CTL register.
- Compile KSV and KSV List values into a single KSV list for passing upstream through the FPD-Link III receiver.
- Indicate topology information in XRPR_BSTS0/1 registers of the FPD-Link III receiver.

#### 5.2.2.1 Local Controller – Poll for Transmit Ports

After HDCP authentication is enabled, the local controller may poll over the I2C interface to attempt to detect HDCP capable transmitters. The HDCP capable FPD-Link III transmitter includes a signature at the register offset 0xF0 of the form "_UH9xx" where xx is typically "47" for the DS90UH947-Q1 HDCP Serializer. Applicable I2C addresses are limited to the addresses to which the FPD-Link III transmitters may be strapped by the ID[x] pin (using 8-bit addressing, these are even addresses between 0x10 and 0x3e).

Upon detection of an HDCP transmitter, the local controller should write configuration information to the FPD-Link III transmitter to ensure proper repeater operation.

#### 5.2.2.2 Local Controller – Downstream Authentication

After polling for HDCP Transmit ports, the local controller should read each port status to determine if the port is attached to a downstream receiver. If so, the local controller should enable authentication for the transmit port by setting only the HDCP_EN bit in the HDCP_CTL register (other bits should be 0).

When the FPD-Link III transmitter has read the downstream HDCP receiver's KSV, it sets the KSV_RDY flag in the HDCP_STS register. At this point, the local controller should read the downstream KSV through the RX_BKSV registers and then set only the KSV_VALID bit in the FPD-Link III transmitters HDCP_CTL register (other bits should be 0). This allows the downstream authentication to complete. Note that the KSV is eventually validated at the source, but that is not required at this point in a repeater implementation.

The local controller should continue to monitor status for all transmit ports until all ports are either unattached (no RX Lock Detect) or authenticated (HDCP_STS register AUTHED bit).

#### 5.2.2.3 Local Controller – Assemble KSV List

Once all downstream ports are either authenticated or unattached, the local controller would compile the KSV List for all downstream devices. For each authenticated port, the local controller would do the following.

- Copy the receiver Bksv value (read previously from the ports RX_BKSV registers) to the KSV List.
- Read the Bcaps value for the downstream receiver using the RX_BCAPS register in the FPD-Link III transmitter

If the Bcaps register indicates the downstream device is an HDCP repeater, the HDCP receiver also performs the following:

- Read the Bstatus registers for the downstream repeater using the TX_BSTATUS registers in the FPD-Link III transmitter
- Update local DEPTH and DEVICE_COUNT values based on values in the ports Bstatus registers
- Read downstream repeater KSV List from the KSV FIFO register in the HDCP transmitter. Downstream KSV List is appended to the local KSV list.
- Set KSV_LIST_VALID bit in the HDCP_CTL register of the FPD-Link III transmitter to allow downstream authentication to complete.

Once the KSV list is completed, the Bstatus register values for DEPTH and DEVICE count as well as the KSV List should be copied to the FPD-Link III receiver. The local controller should then indicate to the FPD-Link III receiver that the KSV List is available for delivery to the upstream FPD-Link III transmitter.

#### 5.2.2.4 Local Controller – Authenticated State

Once authenticated, the local controller should periodically poll the known transmit ports to monitor their current status. Per the HDCP specification, detection of an unauthenticated transmit port would transition the HDCP receiver to an unauthenticated state. To accomplish this, the local controller would need to indicate the loss of authentication to the FPD-Link III receiver. In external repeater control mode, that requires setting the XRPTR_HPD bit in the XRPTR_CTL register.

Optionally, if a downstream port loses connection to the downstream receiver, the HDCP specification does not require the HDCP receiver to go to the unauthenticated state. Instead, the HDCP receiver may remain authenticated. If the previously connected device is reconnected, the downstream link must re-authenticate, but upstream may remain authenticated. If a different KSV is detected for the link, loss of authentication must be reported through the XRPTR_HPD control.

#### 5.2.2.5 Local Controller – Restart Authentication

Upon receipt of a new KSV value from the upstream HDCP transmitter, the FPD-Link III receiver should indicate a new KSV has been written through status available through its register interface. Upon detection of this condition, the local controller should start downstream authentication if not already started. If downstream authentication is already complete, it is possible to report completion status immediately.

#### 5.2.2.6 Local Controller – Hot Plug Detect Propagation

The local controller must be capable of propagating a downstream hot plug detection signal to the FPD-Link III receiver. The hot plug detection is asserted for the following two conditions:

- Assertion of RX Lock Detect by a transmit port in the HDCP repeater
- Assertion of a downstream Hot Plug detect by a transmit port in the HDCP repeater

Both of these conditions are indicated on the DOWN_HPD bit in the HDCP_STS register of the FPD-Link III transmitter. When this occurs, the HDCP transmitter automatically transitions to an unauthenticated state to resume authentication.

When a Hot Plug detection occurs, the local controller should indicate this condition to the FPD-Link III receiver. As described previously, the repeater is not required to propagate the HPD if downstream is able to re-authenticate to a previously authenticated device (that is, same KSV).

### 5.2.3 HDCP Transmitter Operation

The HDCP transmitter in an HDCP repeater operates essentially identically to an HDCP transmitter located at the source of the video data. There are two main differences related to starting authentication and validation of the downstream HDCP receiver. The following sections describe those differences as well as some of the transmitter options.

#### 5.2.3.1 HDCP Transmitter – Start of Authentication

To start authentication, an HDCP transmitter at the source waits for an indication that content protection is desired. In the DS90UH947-Q1, that corresponds to setting the HDCP_EN bit in the HDCP_CTL register. In an HDCP repeater, the downstream authentication starts when a request comes from the upstream HDCP transmitter. While the indication is different for the HDCP repeater, the actual implementation is the same for the FPD-Link III transmitter. In an HDCP repeater, the downstream process begins when the local controller sets the HDCP_EN bit in the FPD-Link III transmitter's HDCP_CTL register.

#### 5.2.3.2 HDCP Transmitter – Validation of HDCP Receiver

To validate that HDCP receiver, the HDCP transmitter at the source should validate that the Bksv (downstream HDCP receiver KSV) has twenty ones and twenty zeros. In addition, it must confirm that the KSV is not on the revocation list provided with the protected content. Since a HDCP repeater does not have access to the revocation list, it bypasses that step and immediately proceeds to test for downstream repeater. To bypass the KSV validation, the local controller must set the KSV_VALID bit in the HDCP_CTL register. Similarly, the HDCP repeater does not validate a downstream KSV List against the revocation list, but proceeds to the Authenticated state. After the KSV List has been read by the local controller, it should set the KSV_LIST_VALID bit in the HDCP_CTL register.

Note that the source should validate all KSVs in the system before allowing the transmission of protected content.

#### 5.2.3.3 HDCP Transmitter – Fast Link Verification

The HDCP transmitter periodically checks the synchronization between the HDCP transmitter and HDCP receiver. If synchronization is lost, the HDCP transmitter must re-initiate authentication to restore synchronization. The FPD-Link III transmitters support an optional mode to check the HDCP synchronization more often than the standard rate. For normal link verification, the HDCP specification calls for checking the Ri value every 128 frames. Setting the FAST_LV control speeds that up to every 16 frames. In Enhanced Link Verification, the Pj value is checked every 16 frames. Setting the FAST_LV control speeds that up to every 2 frames.

By more quickly detecting loss of synchronization, FPD-Link III chipset allows for much shorter loss of video signal to the display upon the loss of synchronization.

Configuration of Fast Link Verification mode is done through setting the FAST_LV bit in the HDCP_DBG register at the FPD-Link III transmitter. This option is automatically passed to the FPD-Link III transmitters in downstream FPD-Link III repeaters.

#### 5.2.3.4 HDCP Transmitter Encryption Modes

The HDCP transmitter includes four modes for enabling encryption. The modes are enabled by setting the ENC_MODE bits in the HDCP_CFG register.

**Table 7. HDCP Encryption Enable Modes**

| MODE | ENC_MODE | DESCRIPTION |
|---|---|---|
| Enc_Authenticated | 00 | Prior to authentication, encryption will be based on register control. When authenticated, the HDCP_ENC_EN bit will be automatically set, and video data will be encrypted. The register control may be overridden by software while authenticated or if authentication is lost. |
| Enc_Reg_Control | 01 | Register control enables or disables encryption. Prior to sending protected content, an attached controller must enable encryption. |
| Enc_Always | 10 | All video data will be encrypted when authenticated. If not authenticated, a blue screen is transmitted. |
| Enc_InBand_Control | 11 | Encryption controls are passed to the transmitter on the video data interface during the vertical blanking interval as described in the Encryption Status Signaling section. This mode may be used in a repeater configuration. |

Register control of encryption is through the HDCP_ENC_EN and HDCP_ENC_DIS bits in the HDCP_CTL register.

Only the Enc_Inband_Control should not be used for the External Controlled repeater. If using Enc_Reg_Control, the controller needs to monitor the RX_ENCRYPTED status in the XRPTR_STS register to determine when to enable and disable downstream encryption.

# 6 System Considerations

## 6.1 Serializer Startup

Check the respective serializer datasheet for specific power-up and initialization instructions. Typical serializer startup sequence is as follows:

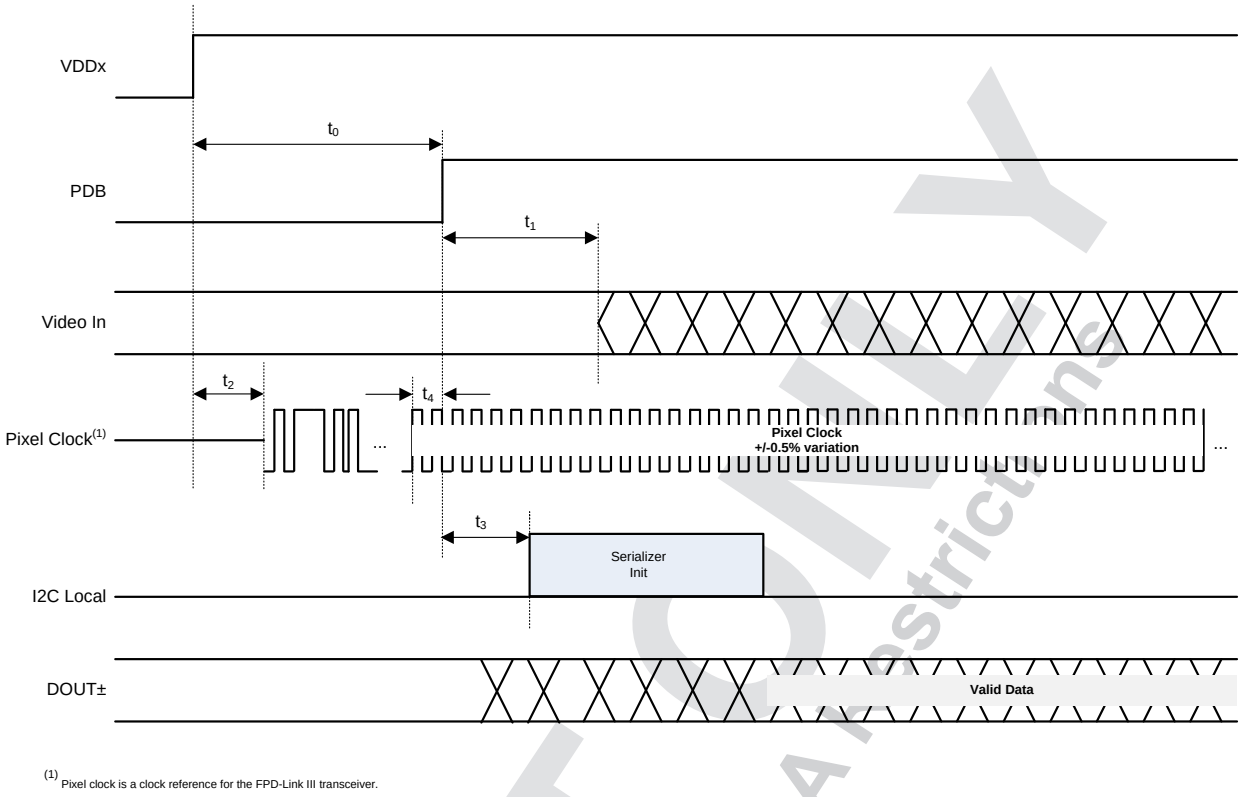
1. Apply all supples in a sequence recommended in the serializer datasheet
2. Wait until all supplies have settled to within recommended limits
3. Apply pixel clock
4. Assert PDB after the pixel clock has stabilized to within 0.5% of the target frequency as shown in Figure 12
5. Apply video inputs
6. Initialize the Serializer
7. Alternatively, PDB may be asserted before the pixel clock as shown in Figure 13. In that case, the initialization sequence has to be concluded with a digital reset or a reset of PLLs, and applied after the pixel clock has stabilized to within 0.5% of the target frequency

The Serializer initialization sequence shown in Figure 12 and Figure 13 consists of any user-defined device configurations and the following instructions:

1. For DS90UH947-Q1, DS90UH949-Q1, or DS90UH949A-Q1, include options for forcing FPD-Link III single-link or dual-link operation.
   - For forcing Single-link mode, use the following configuration:
     - If the compatible Deserializer has High-Speed Control Channel (HSCC) enabled (HSCC_MODE field in the Deserializer HSCC_CONTROL register 0x43 is not 0) prior to forcing Single-link mode, force the back channel capabilities for Port 1.
       - Set Reg0x1E = 0x02. This selects Port 1 in Port Select register.
       - Set Reg0x20 = 0x8F. This makes Port 1 Dual-link capable in Deserializer Capabilities register.
       - Set Reg0x1E = 0x01. This selects Port 0 in Port Select register to restore the register default value.
     - Set Reg0x5B[2:0] = 100b. This enables Auto-detect and disables Dual-link mode in DUAL_CTL1 register)
   - For forcing Dual-link mode, use the following configuration:
     - If the compatible Deserializer has High-Speed Control Channel (HSCC) enabled (HSCC_MODE field in the Deserializer HSCC_CONTROL register 0x43 is not 0) prior to forcing Dual-link mode, force the back channel capabilities for Port 1.
       - Set Reg0x1E = 0x02. This selects Port 1 in Port Select register.
       - Set Reg0x20 = 0x8F. This makes Port 1 Dual-link capable in Deserializer Capabilities register.
       - Set Reg0x1E = 0x01. This selects Port 0 in Port Select register to restore the register default value.
     - Set Reg0x5B[2:0] = 011b. This disables Auto-detect and forces Dual-link mode in DUAL_CTL1 register)
2. For DS90UH947-Q1, DS90UH929-Q1, DS90UH949-Q1, or DS90UH949A-Q1, set Register 0x5B bit 5 to 0. This disables the FPD-Link III PLL from resetting when a frequency change is detected.
3. For DS90UH947-Q1, DS90UH929-Q1, DS90UH949-Q1, or DS90UH949A-Q1, set Register 0x16 to

0x02. This sets the back channel watchdog timer to 2 ms.

4. If implementing the Initialization Sequence B, apply a digital reset without resetting registers. Alternatively, for DS90UH947-Q1, DS90UH929-Q1, DS90UH949-Q1, or DS90UH949A-Q1, you can reset the OpenLDI or HDMI PLL and FPD-Link III PLL by writing the following sequence:
   - Register 0x40 = 0x10
   - Register 0x41 = 0x49
   - Register 0x42 = 0x10
   - Register 0x42 = 0x00
   - Register 0x40 = 0x14
   - Register 0x41 = 0x49
   - Register 0x42 = 0x10
   - Register 0x42 = 0x00
5. For the DS90UH947-Q1 in Dual OpenLDI mode, reset the OpenLDI PLL divider by writing the following sequence:
   - Register 0x40 = 0x10
   - Register 0x41 = 0x49
   - Register 0x42 = 0x16
   - Register 0x41 = 0x47
   - Register 0x42 = 0x20
   - Register 0x42 = 0xA0
   - Register 0x42 = 0x20
   - Register 0x42 = 0x00
   - Register 0x41 = 0x49
   - Register 0x42 = 0x00
6. Set FC_TESTMODE register 0xC0[2] bit. Setting this bit enables faster detection of loss of authentication and ultimately reduces length of "snow" on authentication failures. This register bit needs to be set prior to start of HDCP authentication.
   - For DS90UH947-Q1, DS90UH929-Q1, DS90UH949-Q1, or DS90UH949A-Q1, note that this register is cleared with DIGITAL_RESET0, so software must also set this bit following a DIGITAL_RESET0.




(1) Pixel clock is a clock reference for the FPD-Link III transceiver.

**Figure 12. Serializer Initialization Sequence A**

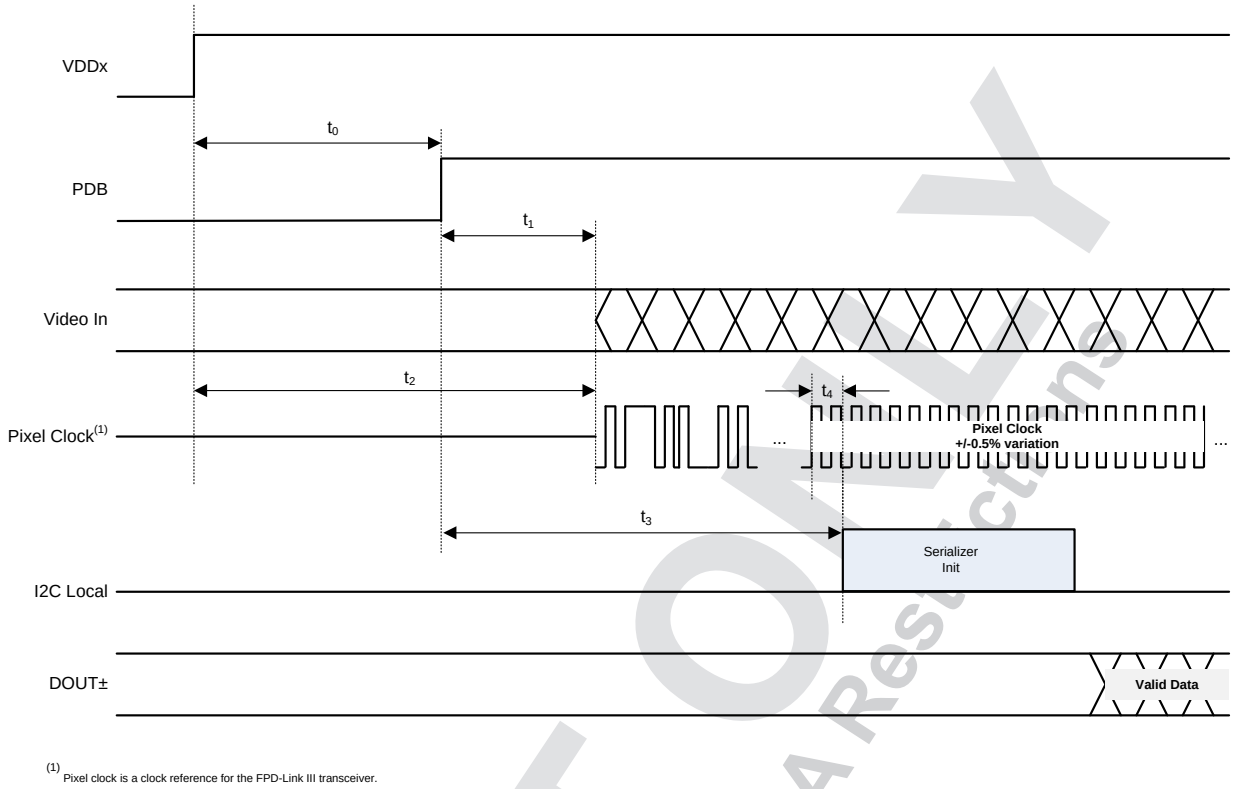



**Figure 13. Serializer Initialization Sequence B**

**Table 8. Serializer Initialization Sequence Timing Parameters**

| PARAMETER | | MIN | TYP | MAX | UNIT | NOTES |
|---|---|---|---|---|---|---|
| $t_0$ | VDDx to PDB delay | 0 | | | ms | Release PDB after all supplies are up and stable. |
| $t_1$ | Video to PDB delay time | 0 | | | ms | Apply video signals after PDB is released. |
| $t_2$ | Pixel Clock to VDDx delay time | 0 | | | ms | Apply pixel clock after all supplies are up. The clock may be applied independent of PDB state. |
| $t_3$ | PDB to I2C Ready (IDX and MODE valid) delay | 2 | | | ms | |
| $t_4$ | Pixel Clock Stable to Reset delay time | 1 | | | µs | Pixel clock frequency must be within 0.5% of the target frequency and stable before releasing PDB (hardware reset).(1) |

(1) If the condition is not met before PDB reset, a software reset is required after the condition has been met.

Note that deviations from the recommended power-up sequencing of the serializer may result in serializer PLL locking to an incorrect, as yet, unsettled clock frequency, that could lead to incorrect data sampling that can create errors and loss of LOCK.

## 6.2 Deserializer Startup and Initialization

Check the respective deserializer datasheet for specific startup instructions. Typical deserializer startup sequence is as follows:

1. Apply all supples in a sequence recommended in the deserializer datasheet
2. Wait until all supplies have settled
3. If possible, wait until valid FPD-Link III data is available at the RIN inputs
4. Assert PDB
5. Initialize the deserializer. See Section 6.2.1

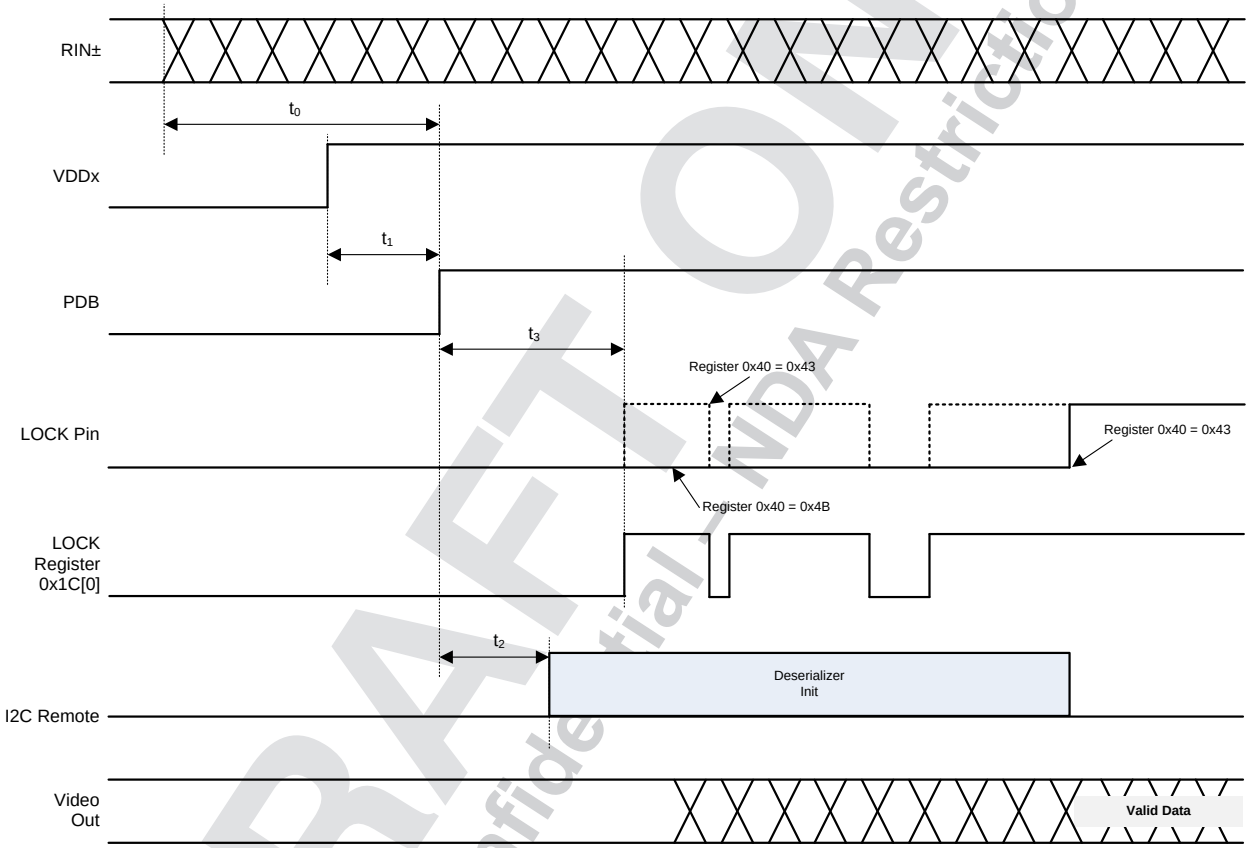



**Figure 14. Deserializer Startup Sequence**

**Table 9. Deserializer Initialization Sequence Timing Parameters**

| PARAMETER | | MIN | TYP | MAX | UNIT | NOTES |
|---|---|---|---|---|---|---|
| $t_0$ | Serializer FPD-Link III forward channel to PDB delay time | 0 | | | ms | Release PDB after all supplies are up and stable, and valid data from serializer is available at the RIN pins.(1) |
| $t_1$ | VDDx to PDB delay | 0 | | | ms | |
| $t_2$ | PDB to I2C ready (IDX and MODE valid) delay | 2 | | | ms | |
| $t_3$ | Deserializer lock time | | | 45 | ms | Assumes default re-lock time of 2.62 ms. |

(1) Deserializer should be enabled after a compatible serializer has started sending valid video data. If this condition is not satisfied, then a digital (software) reset is required after the deserializer locks onto the valid video stream from the serializer. This requirement prevents the deserializer from locking to any random or noise signal, ensures the deserializer has a deterministic startup behavior, specified lock time, and optimal adaptive equalizer setting. See Section 6.2.1.

Deviations from the recommended power-up sequence may cause undesired video artifacts such as flickering and blank screens at the startup as the adaptive equalizer rotates through various values searching for the right setting in the absence of valid data signals in the system context. This effect is more pronounced in noisy systems. Note that the deserializer would eventually lock even with a sequence that deviates from the recommended power on sequence. If deserializer is powered-up before the serializer, the deserializer tries to lock to random noise on the line. In this state, AEQ rotates through different codes trying to lock randomly. When the serializer is enabled, deserializer AEQ may be at an incorrect value. At this value, if the performance is adequate, the device can declare lock and stay in a locked state at a high AEQ value but with very low noise tolerance. With just some extra noise in the system, the part could lose lock creating video artifacts such as blank screen or flicker before the AEQ moves to a different step and eventually locks to the correct value in the presence of valid data.

### 6.2.1 Deserializer Initialization and Monitoring

The deserializer startup behavior can be made very deterministic by following a sequence of register writes described in this section. This sequence, as shown in Figure 15, ensures that the system can power-up without seeing any LOCK drops and video artifacts irrespective of the sequencing between deserializer and serializer. The key for achieving this is forcing the LOCK signal on the deserializer LOCK pin LOW and holding it low until the system can ensure that the internal deserializer LOCK status is valid and stable before releasing the LOCK pin. The following is the recommended sequence:

1. After the deserializer power-up and PDB release, program the deserializer register 0x40 = 0x4B to force LOCK output LOW
   - The setting only affects what comes out on the LOCK pin
   - If LOCK pin output is used to control a display, forcing LOCK pin to LOW blocks out any unwanted display data and avoids flicker or video errors showing on the screen
2. Set the deserializer AEQ range (optional). Typically considered when using long cables.
   - Using the full AEQ range provides the most flexible solution, however, if the channel conditions are known, an improved deserializer lock time can be achieved by narrowing the search window for allowable EQ gain settings.
   - Program register bits 0x35[5:4] = 11
   - Program register bits 0x45[3:0]. The programmed value should be determined based on the channel evaluation.
3. Wait 45 ms.
4. Monitor the deserializer LOCK status register 0x1C[0] bit
   - If 0x1C[0] = 1 go to the next step, else go to the previous step
5. Issue DIGITAL_RESET1 (Reset entire deserializer digital block without registers)
   - For DS90UH940-Q1, prior to DIGITAL_RESET1, set CSIIA register 0x6C to 0xFF
   - Program register 0x01 = 0x01

- For DS90UH940-Q1 or DS90UH940N-Q1, wait 0.5 ms prior to any I2C access to the deserializer

6. Wait 45 ms
7. Check the deserializer LOCK status by reading register 0x1C[0] bit. If 0x1C[0] = 1, go to the next step, else go to the previous step
8. Check the deserializer EQ_STATUS by reading register 0x3B (optional). If EQ level is out of an established threshold range go back to step 6
9. Set AEQ re-lock time to 21 ms per step (Optional)
   - Program register bits 0x45[7:5] = 111
   - Programming this bit ensures AEQ does not quickly increment and rail before any software reads are done
10. Program the deserializer register 0x40 = 0x43 to release LOCK output
    - LOCK pin follows the real Link LOCK signal, which is HIGH at this time
    - The deserializer should output valid display data after this step
    - At this point, the serializer HDCP configuration (See) can be done, followed by the start of authentication control and monitoring process as described in Section 8
11. Wait 21 ms x Number of EQ settings within the AEQ range (optional)
12. Monitor the deserializer AEQ range (optional)
    - Read register 0x3B
    - If the reading is within the defined range, go back to the previous step, else go to the next step
13. Force LOCK pin HIGH (optional)
    - Program the deserializer register 0x40 = 0xCB
14. Reset the deserializer AEQ (optional)
    - Keep the display on while resetting the deserializer AEQ. The display may show errors.
    - Program the register bit 0x35[6] =1
    - Program the register bit 0x35[6] =0
    - The sequence restarts the AEQ loop to the minimum EQ value
15. Release LOCK pin force (optional)
    - Program the deserializer register 0x40 = 0x43
16. Monitor the deserializer LOCK status register 0x1C[0] bit (optional)
    - If 0x1C[0] = 1 go to Step 12, else go to Step 6

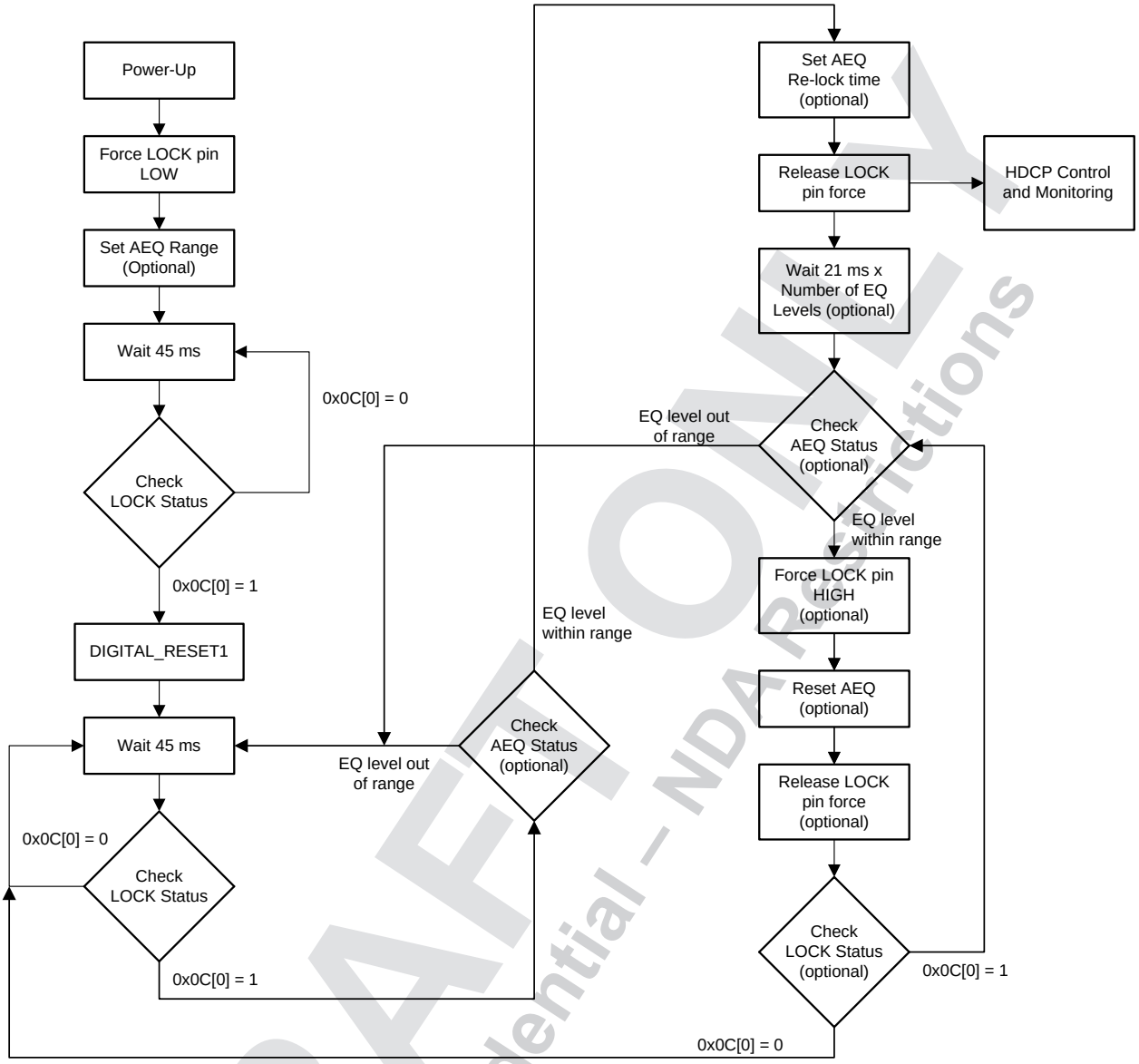



**Figure 15. Deserializer Initialization and Monitoring Flow Diagram**

## 7 Interrupt Support

The HDCP capable FPD-Link III devices support propagation of an interrupt signal from an HDCP receiver, through HDCP repeaters, to the source HDCP transmitter. Each HDCP transmitter may be programmed through the I2C interface to pass an interrupt from the downstream HDCP receiver INTB_IN pin to the transmitter's INTB pin. The interrupt source would typically be a device at the downstream HDCP receiver that needs to notify the host controller of a condition requiring service.

The HDCP transmitter can generate an interrupt signal to the attached controller through the INTB pin. This may be used to indicate need for the controller to process some portion of the authentication flow or to indicate errors in the link status or authentication. The INTB pin is an open-drain, active-low signal that may be shared with other interrupt sources. The HDCP Interrupt Control Register (HDCP_ICR, address 0xC6) enables the various interrupt conditions, while the HDCP Interrupt Status Register (HDCP_ISR, address 0xC7) is used to monitor the interrupt conditions. Bit 0 of the HDCP_ICR is the global interrupt enable that must be set along with at least one of the other interrupt enables to allow the generation of the interrupt on the active-low INTB pin.

**NOTE:** Enabling an interrupt on IE_RXDET_INT ensures software can detect downstream receiver or receiver lock (if HDCP_CFG:RX_DET_SEL =1), while enabling interrupts on IE_KSV_RDY and IE_AUTH_FAIL ensures software can detect loss of authentication.

When the interrupt is detected, the controller should read the HDCP_ISR register to determine the interrupt condition. Bit 0 of the HDCP_ISR indicates an interrupt has occurred, while the individual status bits indicate which conditions have been triggered. The read of the HDCP_ISR also clears the interrupt, releasing the INTB pin. If desired, the controller may then read the HDCP_STS register to determine the current device status. For details on the available interrupt conditions, see the serializer HDCP_ICR and HDCP_ISR register definitions.

The receiver interrupt, bit 5 of HDCP_ICR and HDCP_ISR, is a special case. This interrupt is used to propagate an external interrupt from the HDCP receiver INTB_IN pin to the HDCP transmitter interrupt pin (INTB). The interrupt is active low and is handled similar to other interrupt conditions. Upon detection of a falling edge of the interrupt signal, the HDCP transmitter latches the interrupt condition, set the IS_RX_INT bit in the HDCP_ISR, and assert the INTB pin low. To clear the interrupt signal, the controller must read the HDCP_ISR, releasing the INTB pin, and clearing the HDCP_ISR. It may then check the HDCP_STS:RX_INT bit to determine the current status of the HDCP receiver INTB_IN pin. The INTB pin remains deasserted until the next falling edge of the INTB_IN signal.

Figure 16 shows a typical interrupt propagation block diagram.

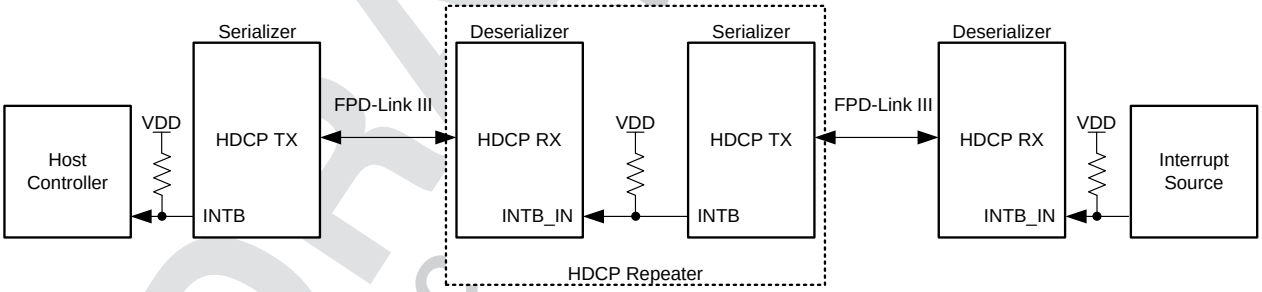



**Figure 16. Interrupt Propagation**

## 7.1 Interrupt Pins

The serializer INTB pin is an active low interrupt output pin that acts as an interrupt for various local and remote interrupt conditions (serializer registers 0xC6 and 0xC7). For the remote interrupt condition, the serializer INTB pin works in conjunction with the INTB_IN pin on the deserializer. This interrupt signal, when configured, propagates from the deserializer to the serializer.

1. On the serializer, set register 0xC6[5] = 1 and 0xC6[0] = 1
2. Deserializer INTB_IN pin is set *LOW* by some downstream device.
3. Serializer pulls INTB pin *LOW*. The signal is active *LOW*, so a *LOW* indicates an interrupt condition.
4. External controller detects INTB = LOW; to determine interrupt source, read HDCP_ISR register.
5. A read to HDCP_ISR clear the interrupt at the serializer, releasing INTB.
6. The external controller typically must then access the remote device to determine downstream interrupt source and clear the interrupt driving the deserializer INTB_IN. This would be when the downstream device releases the INTB_IN pin on the deserializer. The system is now ready to return to step (2) at next falling edge of INTB_IN.

## 8 Control and Monitoring of the Downstream HDCP Authentication Process

Control and monitoring of the downstream HDCP authentication process contains a few operations that are not required if the downstream device is not an HDCP repeater. The following flow should be used by the controller attached to the source HDCP transmitter and covers operation for both repeater and non-repeater cases.

**NOTE:** Wait for serializer RX_LOCK_DET bit rather than RX_DETECT before starting HDCP authentication. RX_LOCK_DET is more restrictive, requiring both back channel link (RX_DETECT) and forward channel lock. Set serializer RX_DET_SEL register 0xC2[1] bit to select RX_LOCK_DET as source for ISR_RX_DET interrupt in HDCP_ISR register 0xC7.

After the serializer has been configured for HDCP operation, the authentication process is controlled over the HDCP transmitter's register interface in the following manner:

1. **Enable HDCP.** Set the HDCP_CTL: EN_HDCP control to a 1 (serializer register bit 0xC3[0]). This allows the HDCP transmitter to begin the authentication process.
2. **Check for KSV Ready.** HDCP transmitter reads the KSV value from the HDCP receiver and makes this available through the register interface. When ready, the HDCP_STS:KSV_RDY flag is set (serializer register bit 0xC4[1]).
   - The controller may poll for KSV Ready or enable an interrupt (set IE_KSV_RDY bit in HDCP_ICR register 0xC6).
   - If the interrupts are enabled, poll the serializer HDCP_ISR 0xC7 register first, and then poll the HDCP_STS 0xC4 and HDCP_CTL 0xC3 registers.
   - Monitoring HDCP_CTL register allows the user to determine if an unexpected device reset has occurred. If a reset has occurred, then the device reconfiguration is required.
   - If an HDCP interrupt is detected, also check the RX_LOCK_DET register 0xC4[5] bit. If that bit is zero, then disable HDCP encryption (HDCP_ENC_DIS = 1).
   -
3. **Read KSV and Repeater.** The controller should read the 40-bit receiver KSV (Bksv) and the RX_BCAPS: repeater status bit.
4. **Check KSV.** Although not required, the KSV may be checked that it contains exactly 20 ones and 20 zeros. The HDCP repeater does not need to check the KSV against a revocation list, as that is only done at a Source transmitter.
5. **Set KSV_Valid.** The controller should set the KSV valid control bit.
   - After setting KSV_VALID (non-repeater mode), check if KSV_RDY = 1 and RX_LOCK_DET = 0 to stop the authentication process. This prevents getting stuck at a blue screen if AUTHED does not occur due to loss of RX_LOCK_DET or failure of authentication.
6. **Check Repeater bit.** If RX_BCAPS: repeater is 0, go to step 11
7. **Repeater: Check for KSV List Ready.** KSV List Ready is asserted when the KSV List is ready to be read (serializer register bit 0xC4[2]).
   - The controller may poll for KSV List Ready or enable an interrupt (set IE_LIST_RDY bit in HDCP_ICR register 0xC6).
   - If an HDCP interrupt is detected, also check the RX_LOCK_DET register 0xC4[5] bit. If that bit is zero, then disable HDCP encryption (HDCP_ENC_DIS = 1).
8. **Repeater: Check KSV List Status.** Check repeater topology by reading RX_BSTATUS registers.
   - If any of the failure bits are set, the device has considered the authentication to fail and restarted authentication. The controller may go to step 1 or exit.
   - The HDCP transmitter does not automatically restart authentication on a topology error.
9. **Repeater: Read KSV List.** The controller should read the KSV List using the register interface using a single I2C auto-incrementing read.
   - The KSV list should be compared against the revocation list.
   - If any KSV matches a KSV on the revocation list, authentication has failed. The controller may restart authentication or exit.

10. **Repeater: Set KSV_List_Valid.** The controller should set the KSV List valid control bit.
11. **Check for Authentication complete.** When authentication is complete, the HDCP_STS:AUTHED flag is set.
    - The controller may poll for authentication complete or enable an interrupt.
    - If the HDCP_STS:KSV_RDY flag is set, indicating authentication has been restarted, the controller should go to step 3.
    - If authentication fails, the HDCP transmitter automatically restarts authentication.
    - After setting KSV_LIST_VALID (repeater mode), check if KSV_RDY = 1 and RX_LOCK_DET = 0 to stop the authentication process. This prevents getting stuck at a blue screen if AUTHED does not occur due to loss of RX_LOCK_DET or failure of authentication.
12. **Authenticated.** Source may set the HDCP_ENC_EN control and begin to send protected content over the video link.
    - If KSV_RDY is detected while previously authenticated and RX Lock is detected (loss of authentication condition), proceed to check KSV and complete authentication. This reduces duration of the blue screen in case of a loss of authentication condition.

    Upon detection of KSV_RDY, go to step 2. Upon loss of Authentication without KSV_RDY, go to step 1.

In the Authenticated state, the controller should monitor the authentication status by polling the status register or via the interrupt pin. Upon loss of authentication, the transmitter immediately begins sending a blue screen to the receiver. The controller may wish to stop sending protected content to allow sending a different screen to the display. Disable content protection by setting the HDCP_ENC_DIS control. The transmitter automatically restarts authentication, so the controller may immediately begin checking for KSV Ready (step 2).

Figure 17 is used to illustrate the authentication control process. In each of the five states—DISABLED, WAIT_KSV, WAIT_LIST, WAIT_AUTHED, and AUTHED—the controller should continue to monitor the HDCP transmitter status by either polling the HDCP_STS register or by waiting for an interrupt from the device.

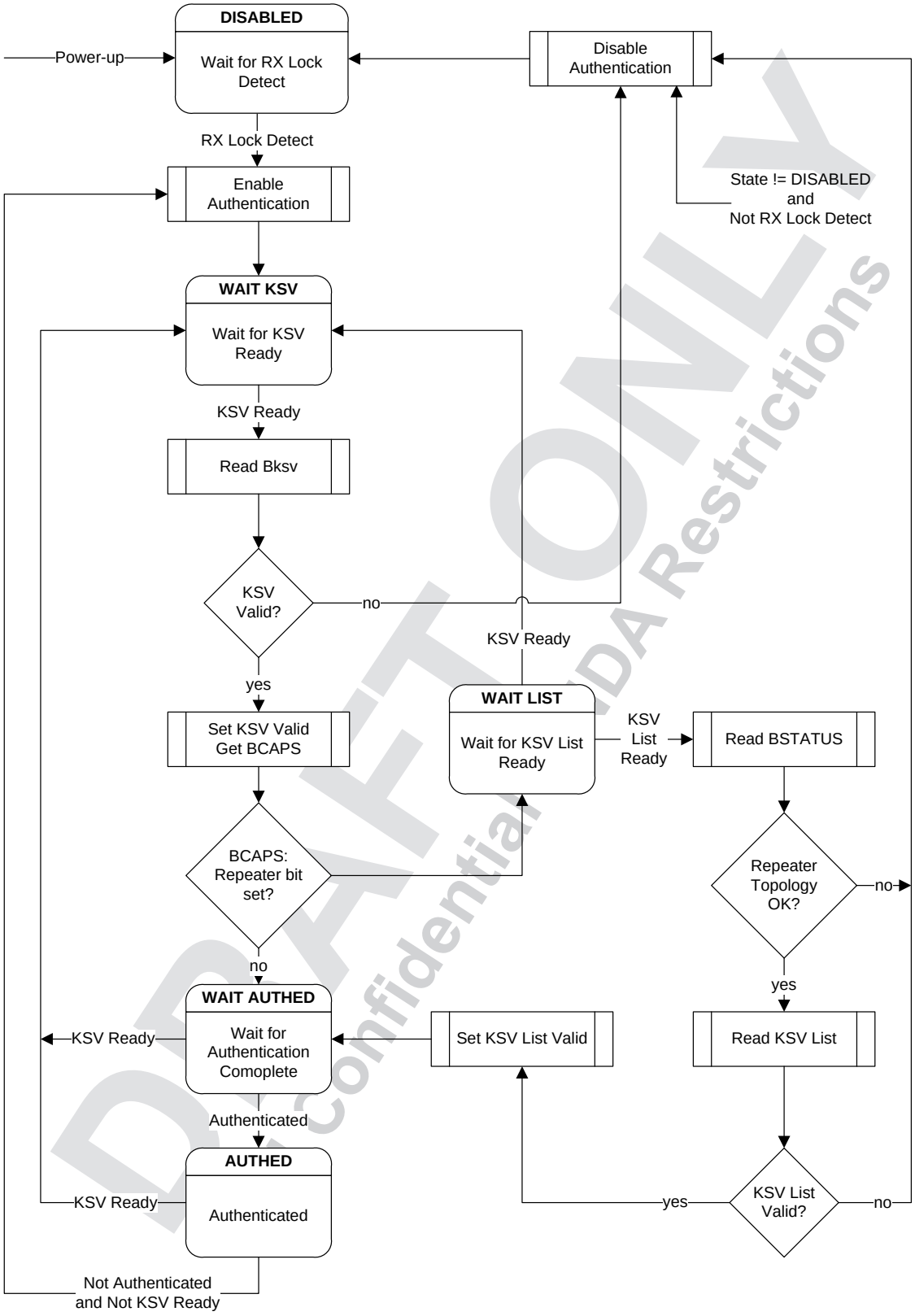



**Figure 17. Downstream HDCP Flow Diagram**

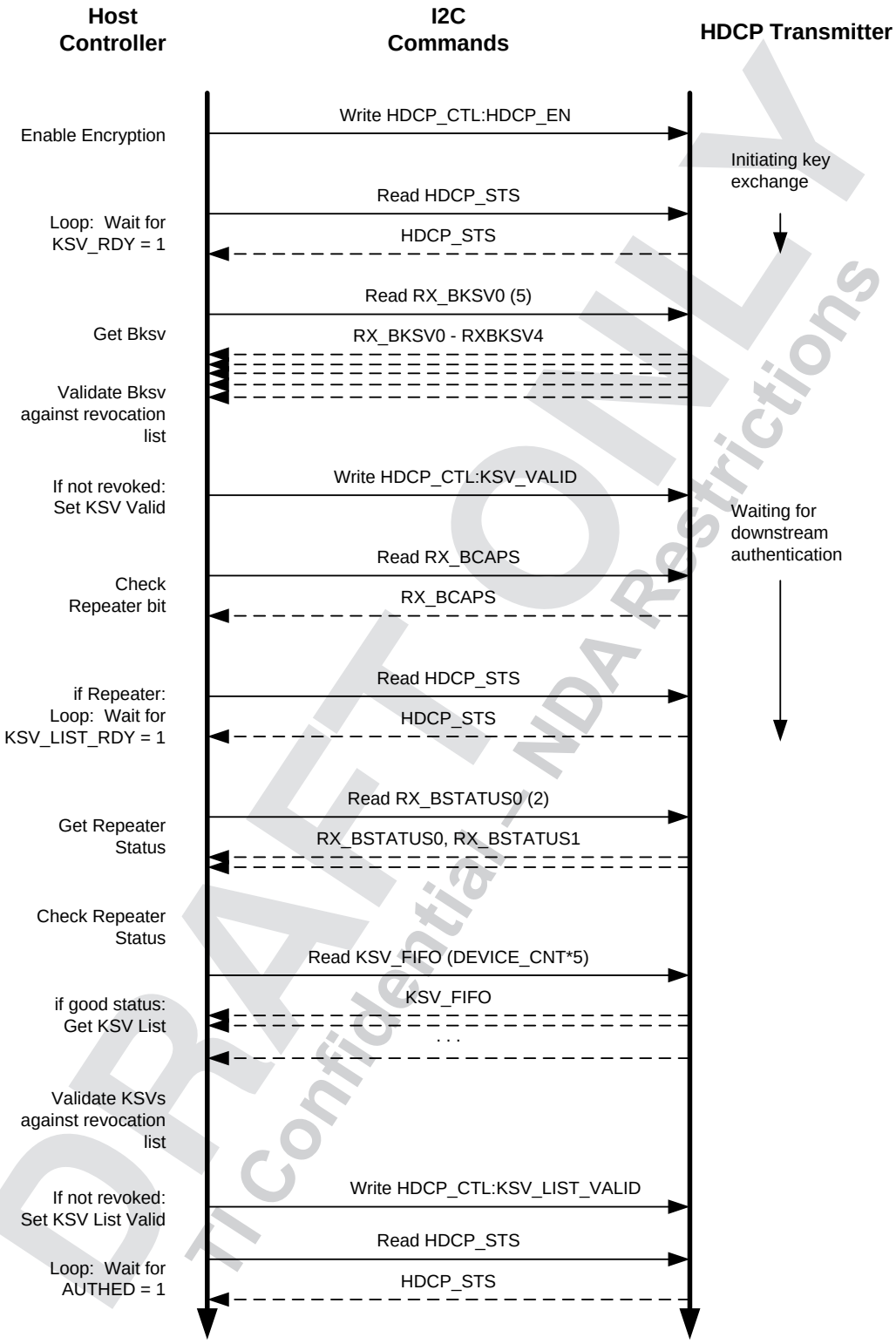



**Figure 18. Authentication Control Sequence Diagram**

## 9 Conclusion

HDCP enabled FPD-Link III devices provide capabilities for implementing various HDCP applications including non-repeater and repeater configurations. They provide seamless support for the delivery of the HDCP Key Selection Vectors (KSVs) for all downstream HDCP receivers to the HDCP transmitter located at the video source.

Following the design guidelines provided in this report is a prerequisite for a successful implementation of HDCP systems.

## 10 References


- Rosselot, David, "HDCP Repeater Design Using the DS90UH925Q and DS90UH926Q", Revision 1.5 (October 17, 2017)
- Rosselot, David, "External Controlled HDCP Repeater Using DS90UH948-Q1", Revision 0.2 (August 16, 2017)
- Ceekala, Vijaya "DS90UH94x Adaptive Equalizer and Startup Description White Paper", Draft Revision 1.2 (February 18, 2018)
- HDCP Rev1.4 Amendment for FPD-Link III, Revision 1.1 (October 4, 2012), Texas Instruments
- High-bandwidth Digital Content Protection System, Revision 1.4. (July 8, 2009), Digital Content Protection LLC
- DS90UH947-Q1 1080p OpenLDI to FPD-Link III Serializer with HDCP (SNLS455)
- DS90UH949A-Q1 2K HDMI-to-FPD-Link III Bridge Serializer With HDCP (SNLS543)
- DS90UH948-Q1 2K FPD-Link III to OpenLDI Deserializer With HDCP (SNLS473)
- DS90UH940N-Q1 1080p FPD-Link III to CSI-2 Deserializer With HDCP (SNLS613)

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

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