

DP159 as DisplayPort Retimer

Mike Campbell

HPA- Consumer and Computing Interface Solutions

ABSTRACT

The SNx5DP159 compensates for ISI loss across a transmission line to provide the optimum DP electrical performance from source to sink. Even though DisplayPort®++ to TMDS applications is the intended target application for the DP159, the DP159 can also be used in a DisplayPort application (like a DisplayPort Monitor). This document is intended to be used to describe how to use the DP159 in a DisplayPort application.

Contents

1	Introduction	2
2	Local I ² C Interface	2
3	Software Implementation	8
4	Initial Power-up Configuration	9
5	DisplayPort Sink Link Training Suggestions	12

List of Figures

1	Local I ² C Example Implementation	2
2	HPD Implementation	3
3	DP159 48-pin QFN AUX Channel Implementation	3
4	DisplayPort Main Link Implementation	4
5	VOD vs VSADJ Resistance	4
6	DP RX Connector	5
7	DP159 RGZ (48-Pin QFN)	6
8	Xilinx FPGA	7
9	DisplayPort 1.2a Link Training CR Phase State Diagram	13
10	DisplayPort 1.2a Link Training Channel Equalization Link Phase State Diagram	14

List of Tables

1	DP159 I ² C Slave Address Options	2
2	REFCLK_OUT Frequencies	12
3	Sink's DPCD TRAINING_AUX_RD_INTERVAL Register	12

1 Introduction

The SNx5DP159 is a Dual Mode DisplayPort® (DP) to Transition Minimized Differential Signal (TMDS) retimer supporting Digital Video Interface (DVI) and High Definition Multimedia Interface (HDMI). The device supports datarates as high as 6 Gbps (HDMI 2.0).

The DP159 is typically used in DisplayPort++ to TMDS applications but can also be used in DisplayPort applications. With its large amount of equalization gain to clean up inter-symbol interference (ISI), retimer functionality to clean up input high frequency (SJ) and random jitter (RJ), the DP159 can also be used in a DisplayPort application where microcontroller or FPGA can configure and manage the DP159. This document details how to implement a DP159 in a DisplayPort application by putting DP159 in a special mode called X-Mode. In X-Mode, the DP159 can support DisplayPort datarates up to 5.4Gbps (HBR2).

1.1 References

1. SNx5DP159 6 Gbps DP++ to HDMI Retimer ([SLLSEJ2](#))
2. DP159 X-Mode Registers (SLLA359)
3. [Total Phase Aardvark I2C/SPI Host Adapter](#)
4. VESA DisplayPort Standard Version 1, Revision 2a. May 23, 2012
5. Xilinx DisplayPort v6.1 LogicCORE IP Product Guide, PG064 September 30, 2015

2 Local I²C Interface

In a DisplayPort application, the DP159 local I²C interface must be used to configure the DP159 PLL, receivers (IN[3:0]P/N), and transmitters (OUT[3:0]P/N). The DP159 internal registers are accessed through the SCL_CTL pin and SDA_CTL pin. The 7-bit I²C slave address of the DP159 is determined by the HDMI_SEL/A1 and EQ_SEL/A0 pins when the I2C_EN pin is high.

Table 1. DP159 I²C Slave Address Options

HDMI_SEL/A1 pin	EQ_SEL/A0 pin	7-bit I ² C Slave Address	Read Slave Address	Write Slave Address
Low (VIL)	Low (VIL)	7'b1011110 or 0x5E	0xBD	0xBC
Low (VIL)	High (VIH)	7'b1011101 or 0x5D	0xBB	0xBA
High (VIH)	Low (VIL)	7'b1011100 or 0x5C	0xB9	0xB8
High (VIH)	High (VIH)	7'b1011011 or 0x5B	0xB7	0xB6

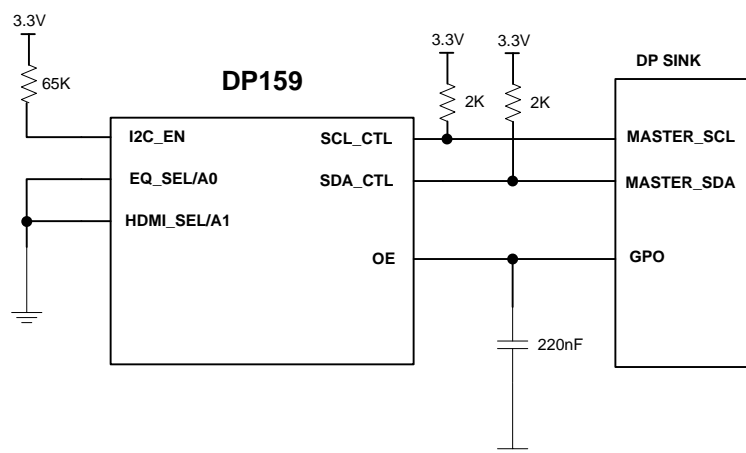


Figure 1. Local I²C Example Implementation

The example implementation selects the slave address of 7'b1011110.

2.1 HPD Implementation

There are two possible implementations for HPD: HPD routed around DP159 or HPD routed thru DP159. Both options are illustrated [Figure 2](#).

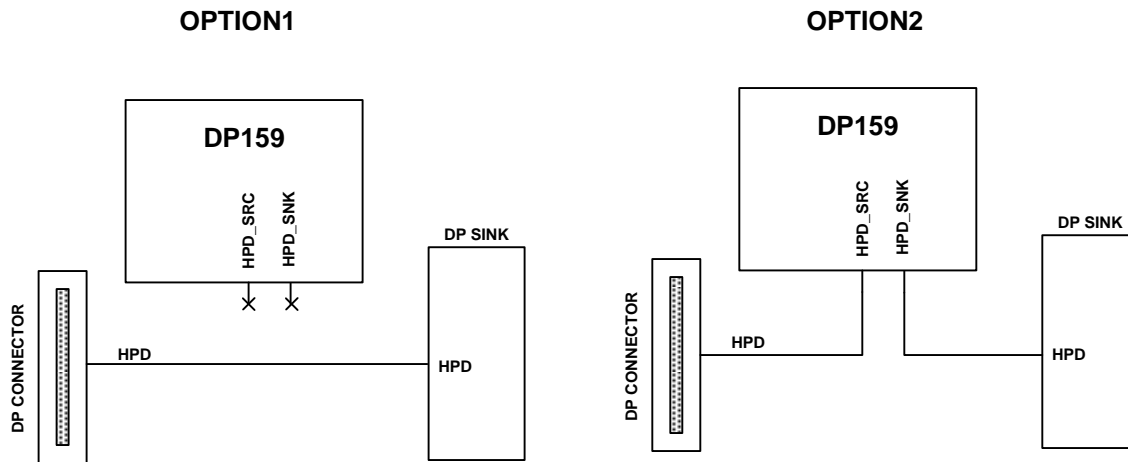


Figure 2. HPD Implementation

2.2 AUX Implementation

The DP159 AUX interface can be used for one of two purposes: (1) Converting I2C-Over-Aux to DDC. (2) Clock output. The conversion of I2C-Over-AUX is not needed for a displayPort sink. The clock output, available only in 48-pin version of DP159, is a divided down (datarate/20 or datarate/40) version of the clock feeding the OUT[3:0]P/N. For example for a 5.4 Gbps (HBR2) datarate, the clock output can be either 135 MHz (divide by 40) or 270 MHz (divide by 20). The clock output maybe useful for some DisplayPort RX that have difficulty tracking the jitter on the OUT[3:0], and therefore, require a reference clock that contains the same high frequency jitter as the OUT[3:0].

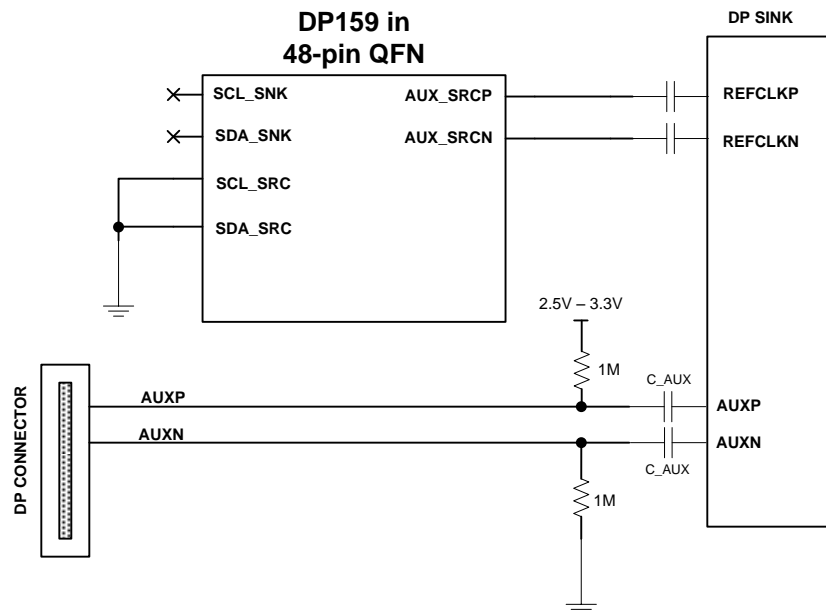


Figure 3. DP159 48-pin QFN AUX Channel Implementation

Do not connect to the DP159's AUX interface to the sinks AUX interface.

2.3 DisplayPort Input and Output Implementation

The DP159 has DisplayPort++ receivers (IN[3:0]) and TMDS transmitters (OUT[3:0]). Because the DP159 receivers are DisplayPort++, there is nothing special required to connect the DP159's to a DisplayPort source. The DP159 transmitters are TMDS, and therefore, must be terminated to 3.3 V through a 50-Ω resistor. The 50-Ω resistors must be placed as close to the DP159 pins as possible. Because DisplayPort is a AC-coupled interface, 100-nF capacitors must be inserted after the 50-Ω termination but before the DisplayPort sinks input. To confine the impedance discontinuities caused by pads of the resistor and capacitor, it is recommended to place the AC capacitors closer to the resistors.

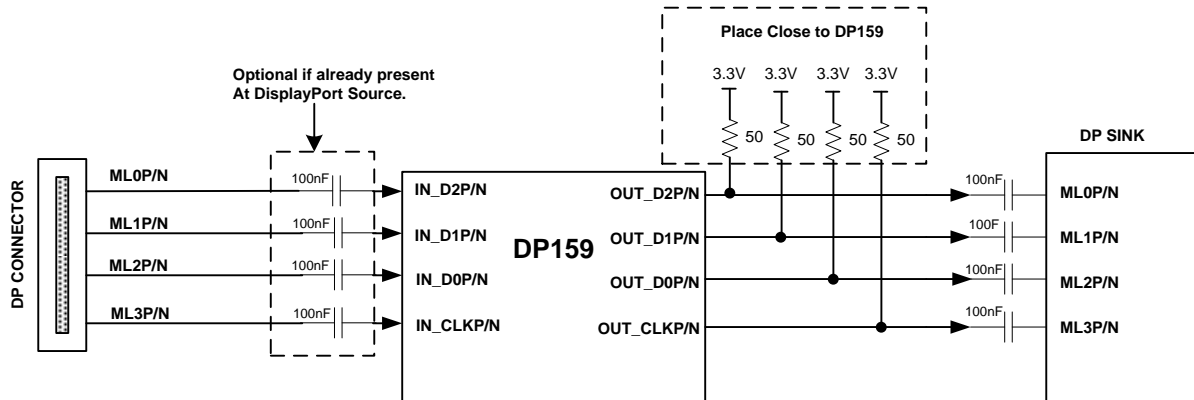


Figure 4. DisplayPort Main Link Implementation

2.4 Output VOD Control

As previously mentioned, the DP159 outputs are TMDS transmitters, and therefore, not DisplayPort compliant. The DP159 output voltage swing level is determined directly by value of the resistor used on the VSADJ pin and by the VSWING register. The recommend value of 4.62 KΩ produces a typically peak-to-peak voltage swing of around 1400 mV. The value of the resistor can be reduced to produce a larger voltage swing or increased to produce a smaller voltage swing.

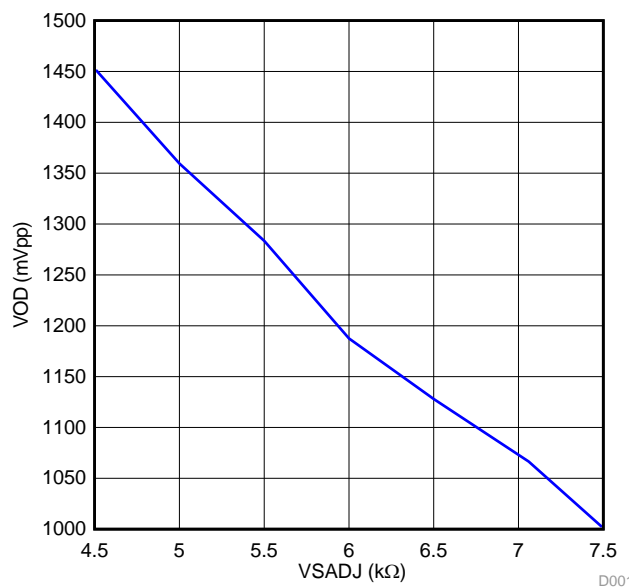


Figure 5. VOD vs VSADJ Resistance

NOTE: Note this chart assumes HDMI termination (single 50-Ω to 3.3 V). The external 50-Ω plus the internal FPGA termination reduces the VOD by as much as 40%.

2.5 DP159 48-pin Example Implementation

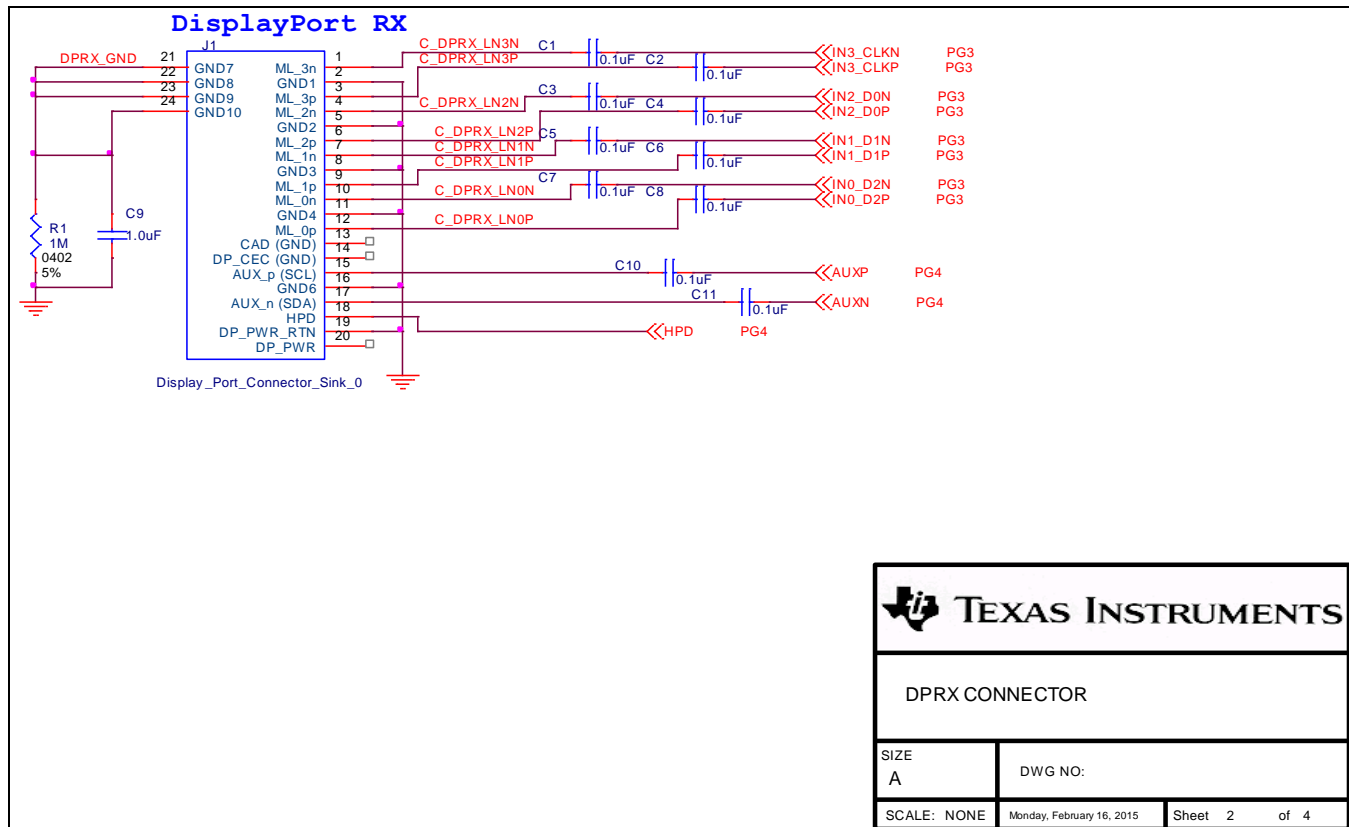


Figure 6. DP RX Connector

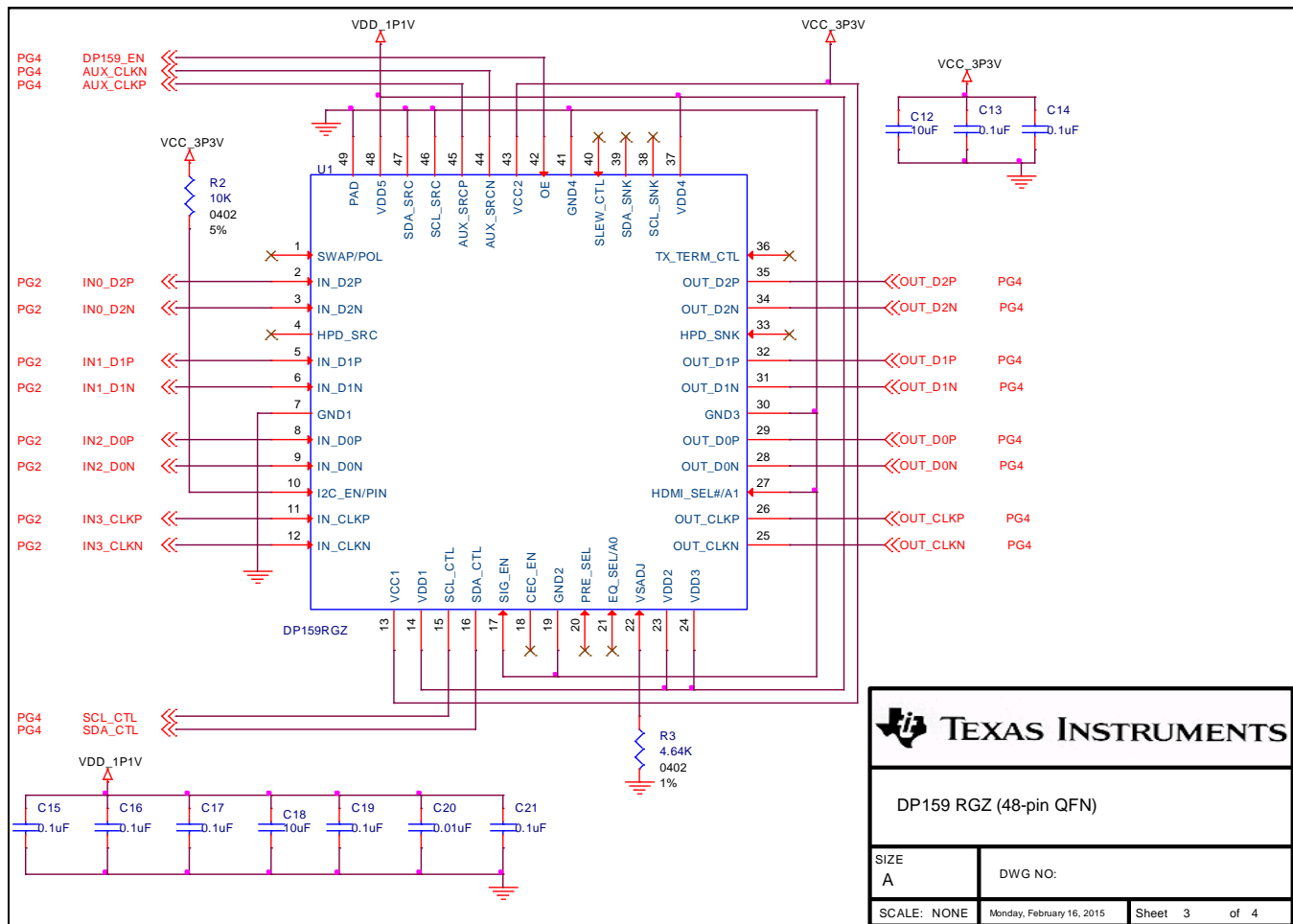


Figure 7. DP159 RGZ (48-Pin QFN)

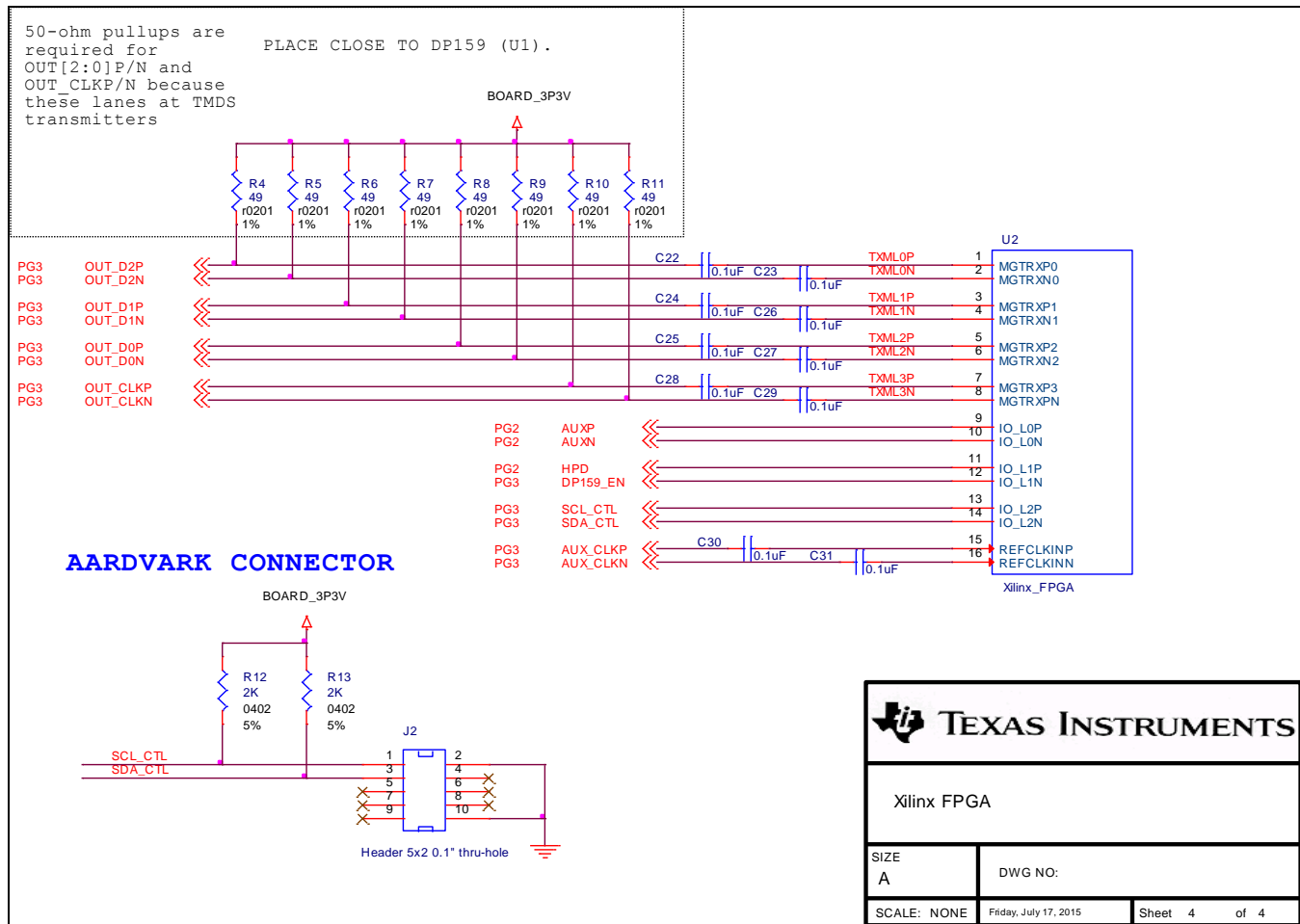


Figure 8. Xilinx FPGA

3 Software Implementation

During link training, the DP159 does not monitor the AUX transaction between a DisplayPort source and a DisplayPort sink as a typical DisplayPort retimer would. Because of this, an external microprocessor (within DisplayPort Source or Sink) must configure the DP159 during all aspects of the link training process.

Software Requirements:

1. Configure DP159 after power up or exiting the reset state (OE pin from low to high).
2. Monitor de-assertion of HPD due to loss of DisplayPort Sink (cable removal)
 - (a) Reinitialize DP159 PLL settings when sink is no longer detected.
 - (b) Disable Rx and Tx lanes.
 - (c) Disable Adaptive Equalizer and enable Fixed EQ.
3. Monitor loss of DisplayPort Source when AUXP transitions from low to high
 - (a) Reinitialize DP159 PLL settings when source is no longer detected.
 - (b) Disable Rx and TX lanes
 - (c) Disable Adaptive Equalizer and enable Fixed EQ.
4. Monitor change in Datarate and number of lanes.
 - (a) Must reinitialize DP159's PLL settings when change in datarate occurs.
 - (b) Must enable/disable Rx and Tx paths based on number of lanes.
 - (c) Disable Adaptive Equalizer and enable Fixed EQ.
5. During Clock Recovery Phase of Link Training
 - (a) Set equalizer to Fixed EQ. The Adaptive Equalizer will not adaptive with a clock pattern
 - (b) Monitor when DP159's PLL is locked.
 - (c) After PLL is locked, transition PLL operating mode to PD mode
6. During Channel Equalization Phase of Link Training
 - (a) Switch from Fixed EQ to adaptive EQ.

Note vast majority of registers mentioned in this section are not found in the SNx5DP159 datasheet, but can be found at: DP159 X-Mode Registers (SLLA359).

4 Initial Power-up Configuration

The following registers must be programmed immediately after power-up or whenever OE transitions from de-asserted (low) to asserted (high).

```

struct DP159InitT
{
    unsigned short regAddr; //DP159's register address
    unsigned short regValue; //Data written
};

struct DP159InitT RBR_HBR_HBR2_Common_Init[] = {
    //Page 0
    {0xFF, 0x00} , //Select Page 0
    {0x09, 0x36} , //Enable X-Mode
    {0x0A, 0x7B} , //Disable HPD_SNK pass thru to HPD_SRC.
    {0x0D, 0x80} , //Enable clock on AUX. Select 1/20 mode.
    {0x0C, 0x6D} , //Set TX Swing to Max
    {0x10, 0x00} , //Turn off pattern verifier
    {0x16, 0xF1} , //Disable char-alignment on all lanes.

    {0xFF, 0x01} , // Select Page 1

    //CONFIGURE PLL BLOCK
    {0x00, 0x02} , //Enable Bandgap.
    {0x04, 0x80} , //PLL_FBDIV[7:0]
    {0x05, 0x00} , //PLL_FBDIV[10:8]
    {0x08, 0x00} ,
    {0x0D, 0x02} , //Select LN0 for clock.
    {0x0E, 0x03} , //CDR_CONFIG[4:0]. FIXED, LN0.
    {0x01, 0x01} , //CP_EN is PLL mode
    {0x02, 0x3F} , //CP_CURRENT is high.
    {0x0B, 0x33} , //Loop Filter to 8K.
    {0xA1, 0x02} , //Allows for Override of PLL settings.
    {0xA4, 0x02} , //Allows for Override of PLL settings.

    //CONFIGURE TX BLOCK
    {0x10, 0xF0} , //ENTX for all four lanes (disable)
    {0x11, 0x30} , //TX_RATE is Full Rate, TX_TERM = 75 to 150 , TX_INVPAIR = None
    {0x14, 0x00} , //HDMI_TWPST1 is 0dB pre-emphasis
    {0x12, 0x03} , //SLEW_CTRL is Normal, SWING is 600mV.
    {0x13, 0xFF} , //FIR_UPD. Load TX settings
    {0x13, 0x00} ,

    //CONFIGURE RX BLOCK
    {0x30, 0xE0} , //Disable Receivers except lane 0
    {0x32, 0x00} , //PD_RXINT
    {0x31, 0x00} , //RX_RATE is Full
    {0x4D, 0x08} , //EQFTC = 0 and EQLEV = 8
    {0x4C, 0x01} , //Enable Fixed EQ
    {0x34, 0x01} , //Enable Offset correction
    {0x32, 0xF0} , //Load RX settings.
    {0x32, 0x00} ,
    {0x33, 0xF0} , //Load EQ settings.

    {0xFF, 0x00} , //Select Page 0

    {0x0A, 0x3B} , //Enable HPD_SNK pass thru to HPD_SRC. Retimer
    {0xFF, 0x01} , //Select Page 1
};

```

4.1 Reinitialize PLL, RX, and TX Settings

The DP159 PLL, RX, and TX settings must be reinitialized anytime there is a change in datarate, lane count, HPD de-asserted (sink removal), or DisplayPort Source no longer detected.

All registers below are located in DP159 Page 1.

```
struct DP159InitT
{
    unsigned short regAddr; //DP159's register address
    unsigned short regValue; //Data written
};

struct DP159InitT Disconnect_Event[] = {
    {0x00, 0x02} , //Disable PLL and clear A_LOCK_OVR
    {0x34, 0x01} , //Enable Offset Correction
    {0x02, 0x3F} , //CP_Current is High BW.
    {0x01, 0x01} , //CP_EN is PLL mode
    {0x0B, 0x33} , //PLL Loop filter 8K
    {0x4D, 0x08} , //EQFTC = 0 and EQLEV = 8
    {0x4C, 0x01} , //Set to Fixed EQ
    {0x33, 0xF0} , //Load EQ settings
    {0x10, 0xF0} , //Disable all Tx Lanes
    {0x30, 0xE0} , //Enable Rx Lane 0 only
};
```

4.2 Bandwidth and Number of Lanes

Before the beginning of link training, the GPU must inform the sink of the link bandwidth (LINK_BW_SET) and number of lanes (LANE_COUNT_SET). It is at this stage software must to program the DP159's PLL enable and number of Rx lanes. The TX lanes will be disabled in this step.

```
#define LBW_HBR2          0x14 // 0x14 = 20 -> 20 * 270 Mbps = 5.40 Gbps
#define LBW_HBR          0x0A // 0x0A = 10 -> 10 * 270 Mbps = 2.70 Gbps
#define LBW_RBR          0x06 // 0x06 = 6 -> 6 * 270 Mbps = 1.62 Gbps

void BW_handler (void) {

    u8  rtxen, eqreg, tsten;
    u16 eq_lev = 8;

    rtxen = (link_laneCnt == 1) ? 0xE1 : (link_laneCnt == 2) ? 0xC3 : 0x0F;
    eqreg = ((link_bw == LBW_HBR2) ? 0x0 : (link_bw == LBW_HBR ) ? 0x1 : 0x2 ) << 4 | (eq_lev &
0x0F);
    tsten = (link_laneCnt == 1) ? 0x11 : (link_laneCnt == 2) ? 0x31 : 0xF1;

    write_csr (0x10, 0xF0); // Disable TX lanes
    write_csr (0x00, 0x02); // Enable Bandgap, Disable PLL, clear A_LOCK_OVR
    write_csr (0x01, 0x01); // CP_EN = PLL (reference) mode
    write_csr (0x0B, 0x33); // Set PLL loop filter
    write_csr (0x02, 0x3F); // Set CP_CURRENT
    write_csr (0x30, rtxen); // Enable RX lanes
    write_csr (0x00, 0x03); // Enable Bandgap, Enable PLL, clear A_LOCK_OVR

    write_csr (0x4C, 0x01); // Enable fixed EQ (use fixed when RX disabled)
    write_csr (0x4D, eqreg); // Set EQFTC and EQLEV (fixed EQ)

}
```

4.3 Link Training Clock Recovery phase

While in the clock recovery phase of link training phase, the DisplayPort source is required to transmit a TPS1 pattern. The TPS1 pattern consists of a repetition of D10.2 symbols (without scrambling). This pattern results in a clock at ½ the datarate (2.7 GHz for HBR2, 1.35 GHz for HBR, and 810 MHz for RBR).

4.3.1 Determine PLL Locked

While the DP159 is receiving the TPS1 pattern, software must monitor the DP159's PLL LOCK_COMPLETE status. The PLL is considered locked when address 0x00 bit 6 in Page 1 is set. Note that depending on the quality of signal being received, the DP159 may not acquire PLL lock until VOD reaches level 1 or Level 2. Because of this, the lock_wait interval must be set to a value large enough to allow the DP159 to acquire lock.

```

u16          lock_wait          = 256; //Approx 300us with 400KHz I2C.

ReadBuffer[0] = 0;
while (ReadBuffer[0] == 0 && lock_cnt < lock_wait) {
    read_csr (0x00);
    ReadBuffer[0] = ReadBuffer[0] & 0x40; // 0x80;
    lock_cnt++;
}

```

4.3.2 Change PLL Mode

Once PLL lock has been achieved, software must immediately transition the PLL mode of operation from PLL_MODE to PD_MODE. The actual PLL setting will be different based on the datarate. It is important to note that A_LOCK_OVR (Page 1 address 0x00 bit 6) must be set to a 1'b1 before transitioning the PLL mode. The following code shows what values based on datarate need to be programmed and the order to program them. It is also important to enable the TX lanes in this stage so that DP sink can start performing clock recovery.

```

struct DP159InitT
{
    unsigned short regAddr; //DP159's register address
    unsigned short regValue; //Data written
};

rtxen = (link_laneCnt == 1) ? 0xE1 : (link_laneCnt == 2) ? 0xC3 : 0x0F;

struct DP159InitT RBR_CRDONE[] = {
    {0x10, rtxen} , //Enable TX lanes
    {0x00, 0x23} , //Enable PLL and Bandgap and A_LOCK_OVR
    {0x02, 0x1F} , //CP_CURRENT for RBR.
    {0x0B, 0x30} , //PLL Loop filter 1K.
    {0x01, 0x02} , //CP_EN is PD mode
};

struct DP159InitT HBR_CRDONE[] = {
    {0x10, rtxen} , //Enable TX lanes
    {0x00, 0x23} , //Enable PLL and Bandgap and A_LOCK_OVR
    {0x02, 0x27} , //CP_CURRENT for HBR
    {0x0B, 0x30} , //PLL Loop filter 1K.
    {0x01, 0x02} , //CP_EN is PD mode
};

struct DP159InitT HBR2_CRDONE[] = {
    {0x10, rtxen} , //Enable TX lanes
    {0x00, 0x23} , //Enable PLL and Bandgap and A_LOCK_OVR
    {0x02, 0x5F} , //CP_CURRENT for HBR2.
    {0x0B, 0x30} , //PLL Loop filter 1K.
    {0x01, 0x02} , //CP_EN is PD mode
};

```

4.4 Link Training Channel Equalization Phase

After completing the clock recovery phase of link training, the source will transition to the channel equalization phase. Once in the channel equalization phase, software should enable DP159 adaptive equalizer.

```

write_csr (0x4C, 0x03); // Enable Adaptive EQ

```

4.5 REFCLK_OUT

As mentioned in [Section 2.2](#), the AUX_P and AUX_N can function as a clock output. In order to enable the clock on these pins the following conditions must be true:

- The AUX_CFG bit (Page 0 offset 0Dh bit 7) must be set to a 1'b1.
- The AUX_BRG_EN bit (Page 0 offset 0Ah bit 3) must be set to a 1'b1.
- LOCK_COMPLETE bit (Page 1 offset 00h bit 6) must be set to a 1'b1.
- TX Lane 0 must be enabled (Page 1 offset 10h bit 0 is 1'b1 and Page 1 offset 10h bit 4 is 1'b0).

Table 2. REFCLK_OUT Frequencies

DisplayPort RATE	TX_CLK_TST REGISTER 0 = 1/20 1 = 1/40	REFCLK_OUT FREQUENCY (MHz)
RBR (1.62 Gbps)	0	81
	1	40.5
HBR (2.7 Gbps)	0	135
	1	67.5
HBR2 (5.4 Gbps)	0	270
	1	135

5 DisplayPort Sink Link Training Suggestions

This section provides suggestions for sink manufacturer on how to communicate with a DisplayPort 1.2 source with a DP159 in the middle.

5.1 Adjust AUX Read Interval

The DisplayPort 1.2a standard defines a DPCD register, called TRAINING_AUX_RD_INTERVAL, within the sink that adjusts the timing of the aux read of DPCD link status registers (DPCD address 00202h thru 00204h) and adjust request registers (DPCD address 00206h and 00207h). Typically, a sink defaults this register to 00h. Changing the default to 01h or 02h allows more time for the I²C writes to the DP159's DCPD registers to complete.

Table 3. Sink's DPCD TRAINING_AUX_RD_INTERVAL Register

DPCD Address	Definition	Read/Write over AUX Ch
0000Eh	TRAINING_AUX_RD_INTERVAL Link status/Adjust request read interval during main link training 00h: 100 μ s for main link clock recovery phase and 400 μ s for main link channel equalization phase 01h: 4 ms all 02h: 8 ms all 03h: 12 ms all 04h: 16 ms all Other values are reserved	Read-Only

5.2 Link Training Clock Recovery (CR) Phase

During link training, a DisplayPort 1.2 compliant source will follow the CR phase of link training detailed in Figure 9. It is important the DisplayPort sink take full advantage of the source behavior. The state diagram shows that CR phase will continue as long as the same VOD level is not used 5 times or a max VOD level is never used. Because of this, the sink CR phase algorithm should never use the same voltage swing level five times or the max voltage swing. Depending on the source, the max voltage swing level can be either Level 2 or Level 3. Typically, a sink algorithm starts at level 0 and progress to level 3. It is suggested to try level 0 two times, level 1 four times, level 2 four times, and level 3 once. Keep in mind source support for level 3 is not required. For the case in which level 3 is not support by the source, level 2 will happen once.

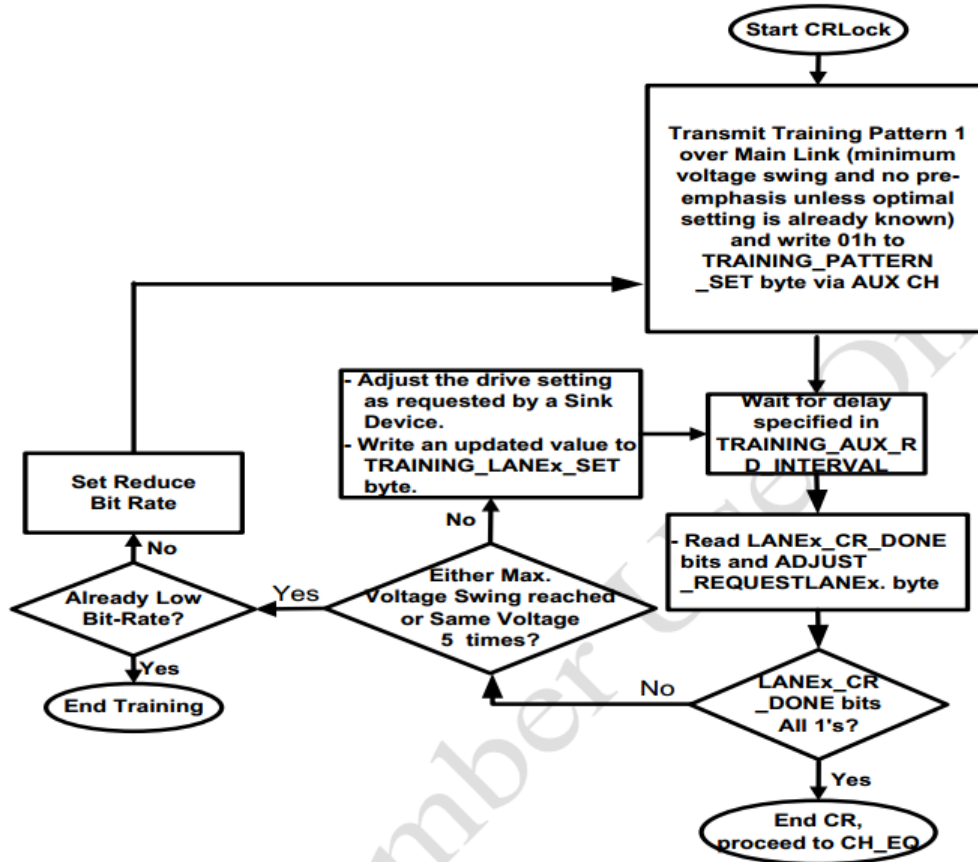


Figure 9. DisplayPort 1.2a Link Training CR Phase State Diagram

5.3 Link Training Channel Equalization (EQ) Phase

During link training, a DisplayPort 1.2 compliant source will follow the channel EQ phase of link training detailed in Figure 10. It is important the DisplayPort sink take full advantage of the source behavior. In the EQ phase of link training, the sink typically requests an increase in pre-emphasis level starting at level 0 and ending at level 3. The channel between DP159 and sink is fixed, and therefore, should be programmed to a fixed voltage swing and pre-emphasis level through the DP159 local I²C interface. The EQ increase requested by the sink not cause the DP159 to increase its output pre-emphasis level. As Figure 10 indicates, a loop count greater than 5 causes EQ phase to end. In order to help ensure a successful EQ phase, the sink channel EQ algorithm should request pre-emphasis level 0 two times, level 1 two times, and level 2 once. The sink is welcome to try other combinations but needs to keep in mind that once is loop count requirement is met the source ends channel equalization phase of link training.

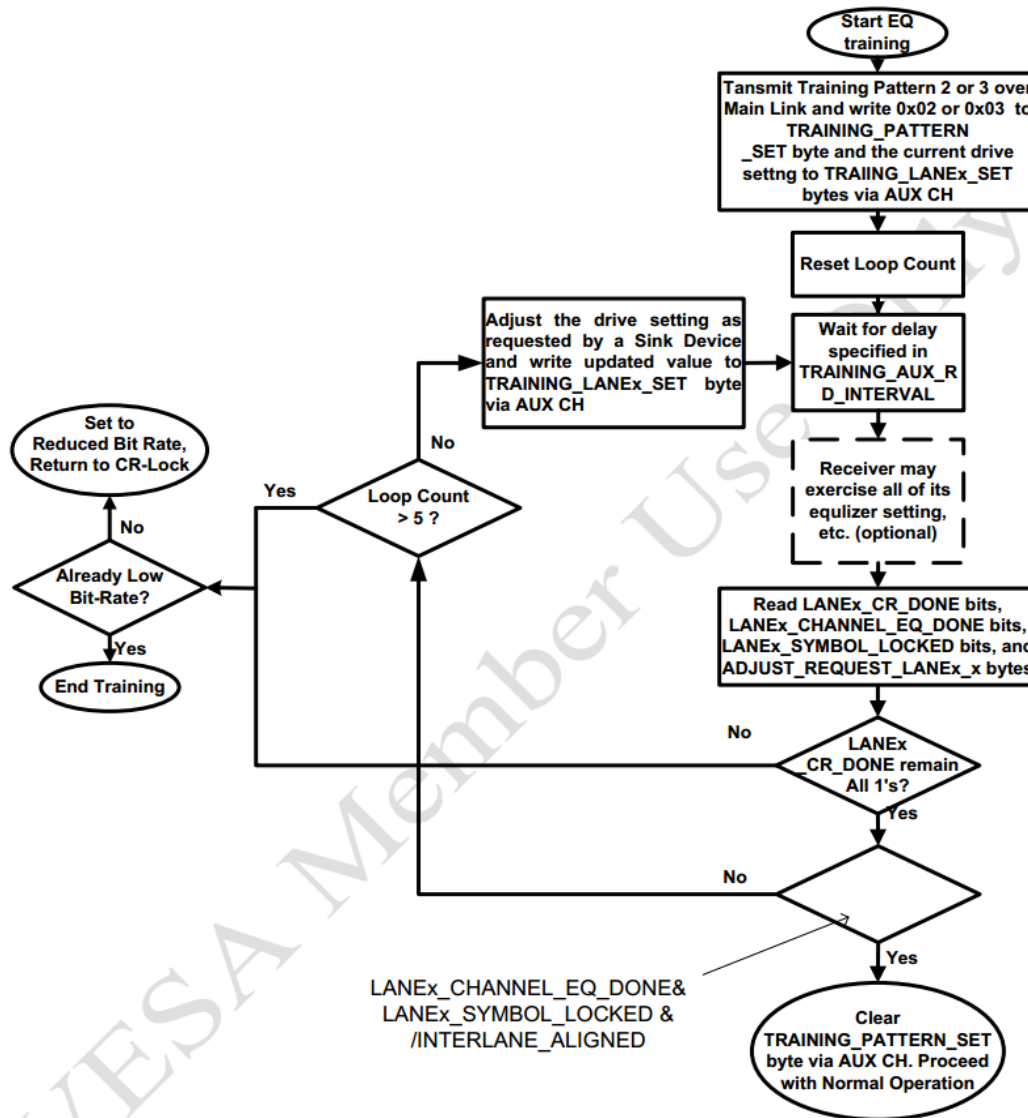


Figure 10. DisplayPort 1.2a Link Training Channel Equalization Link Phase State Diagram

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com