System Setup

The system consists of 2 custom designed Zynq FPGA boards of which one is configured as DP Source board, hereafter termed as DPTX and the other as DP Sink board, hereafter termed as DPRX. The DPTX is successfully validated for LANE_COUNT = 4 at LANE_BW = 2.7Gbps with a resolution of 800x600 @60Hz from a custom pattern generator logic. The DPRX validation is currently under progress. For DPRX,

1. DP159 is used as the retimer.
2. A lane_clock of 135MHz is mandatory which is to be provided by the DP159. Since the AUX_P/N pins are not generating this clock, a separate 135MHz clock is supplied using a clock generator from the same DPRX.

<u>DPRX Experimental Procedure:</u>

1. The DPTX & DPRX are connected via DP cable.
2. DPTX & DPRX stacks are not completely developed, hence we are performing all the necessary configuration manually.
3. At power up, initially DPTX is configured and then the DPRX. In DPRX configuration, the OE of the DP159 is toggled for reset and the following registers are configured:

```
InitTable[address, data] = {

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

        {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.

        {0x0D, 0xC0},          // Clock on AUX is 1/40 datarate and enabled.

        {0x0C, 0x6D} , //Set TX Swing to Max

        //{0x0C, 0x00},          // Set TX Swing to NOM

        {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. The DPRX is then forced to wait for Training Pattern 1.

5. When the DPRX waits for TP_1, the DPTX sends the TP_1.
6. When the DPRX receives the TP_1, the following registers are written onto DP159:

```
TP1_handler_val[address, value] = {
        {0x00, 0x02},  // Enable Bandgap, DISABLE PLL, clear A_LOCK_OVR (to reset it)
        {0x01, 0x01},  // CP_EN = PLL (reference) mode
        {0x0B, 0x33},  // Set PLL control
        {0x02, 0x3F},  // Set CP_CURRENT
        {0x30, 0x0F},  // Enable RX lanes
        {0x00, 0x03},  // Enable Bandgap, ENABLE PLL, clear A_LOCK_OVR
        {0x4C, 0x01},  // Enable fixed EQ (to reset adaptive EQ logic)
        {0x4D, 0x18 },  // Set EQFTC and EQLEV (fixed EQ)
        {0x10, 0x0F},  // Enable TX lanes
        {0x00, 0x23},  // Enable PLL and Bandgap, set A_LOCK_OVR, and set expand LPRES
};
```

7. The DP159 is then continuously read for 0x00, monitoring the LOCK_COMPLETE bit, for some time say 500ms, but no LOCK_COMPLETE bit is set. Post 500ms the following registers are written:

```
TP1_handler_val_2[address, value] = {
        {0x02, 0x27},  // CP_CURRENT
        {0x0B, 0x30},  // Set PLL control
        {0x01, 0x02},  // CP_EN is PD mode
        {0xFF, 0x00}, // Select page 0
    {0x16, 0xF1}, // Set DP_TST_EN per #lanes, latch FIFO errors
        {0x10, 0x00},  // Disable PV (allows char-align and 8b10 decode to operate)
        {0xFF, 0x01},   // Select page 1
};
```

8. Post the write of TP1_handler_val_2 values, the DPTX again transmits the TP_1.
9. DPRX receives the TP_1 and again reconfigures DP159 with TP1_handler_val and polls for LOCK_COMPLETE bit. At this point, read from DP159 provides **0x00 = 0x63**, indicating

LOCK_COMPLETE bit as set. However, there is no clock observed on the AUX_P/N pins. Also,

```
ReadBuffer[0] = ReadBuffer[0] & 0x40; // 0x80;
```

Indicates that we may need the bit7 of 0x00 to be set (from "//0x80").

A read on 0x00 = 0x40 or 0x00 = 0x80, either bit6 or bit7 setting would be enough for a valid clock on the AUX_P/N pins?

Observations:

1. No clock is observed on the AUX_P/N pins after DP159 configuration.
2. The DPTX is able to perform AUX transactions. The AUX lines are bypassed on DP159, Figure 3, Section 2.2, DP159 retimer.
3. For Link Training from the DPTX, we are not able to achieve Clock_Recovery for TP_1. We suspect that this is due to the non-synchronized clock with TP_1 as the clock is provided from a clock generator and not from the AUX_P/N pins of DP159.