AC460
Application Note
Building MIPI CSI-2 Applications using SmartFusion2 and IGLOO2 FPGAs

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

51900460. 1.0 9/16

Contents

Figure

Table

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.0

Revision 1.0 is the first publication of this document.

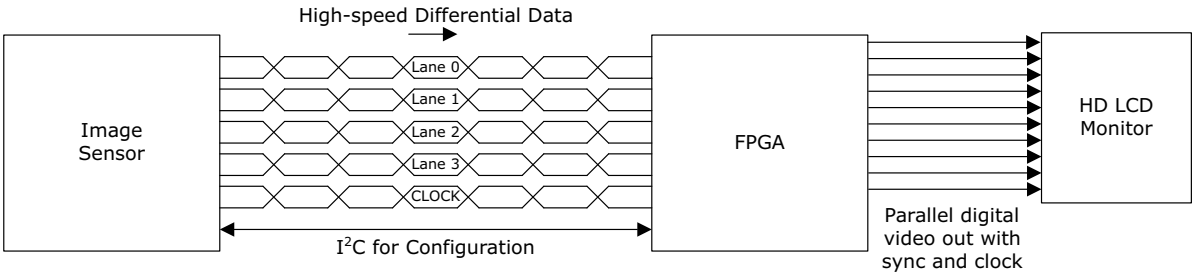
2 Overview

This application note describes how to build Mobile Industry Processor Interface (MIPI) Camera Serial Interface 2 (CSI-2) receive solutions using SmartFusion®2 SoC FPGA and IGLOO®2 FPGA devices. It guides the reader on how to build the design using Libero® SoC software, which I/O interfaces to use. It also lists some of the capabilities and limitations of the device with reference to implementing the MIPI CSI-2 protocol.

2.1 Introduction

The MIPI Alliance Group has defined specifications for CSI-2, which is used by most camera sensor manufacturers to transfer image pixel data from the sensor to the processing elements. Several applications require low-power, secure, and highly-reliable solutions that can leverage SmartFusion2 and IGLOO2 FPGAs to build end products using sensors that support MIPI CSI-2. A typical block diagram of such a system is shown below.

Figure 1 • **MIPI CSI-2 Based System**



In the preceding figure, the MIPI CSI-2 interface consists of one or more high-speed serial unidirectional differential data pairs and a high-speed serial clock from the transmitter (image sensor) to the receiver (FPGA). The MIPI CSI-2 specification defines High Speed (HS) and Low Power (LP) modes of operation. This application note focuses on high-speed operations only, and does not consider low-power operations, as the latter is often not required for interfacing with most applications configured for MIPI CSI-2 receive only. The bi-directional low-speed I^2C interface is used for control and configuration. Image sensors are typically configured by updating registers that are accessible by the FPGA through the 1^2C interface. For more information about the MIPI CSI-2 interface, see the *MIPI CSI-2 Specification* by MIPI Alliance Group.

SmartFusion2 and IGLOO2 SoC FPGAs come with high-speed differential receivers that can be configured to receive MIPI CSI-2 high-speed data. The SmartFusion2 device has a built-in ARM Cortex-M3-based microcontroller subsystem (MSS) that can be used to configure the image sensor over an I2C bus. See the *DS0128: IGLOO2 and SmartFusion2 Datasheet* for more information about SmartFusion2 and IGLOO2 FPGA devices.

The Libero design software can be used to build image processing application designs using easy-to-use SmartHDL schematic-based design entry tools, or standard HDL coding methods. The Libero design software comes with all of the required building blocks to build a MIPI CSI-2 high-speed receiver. Subsequent sections in this document go into the details of building a successful MIPI CSI-2 reference design.

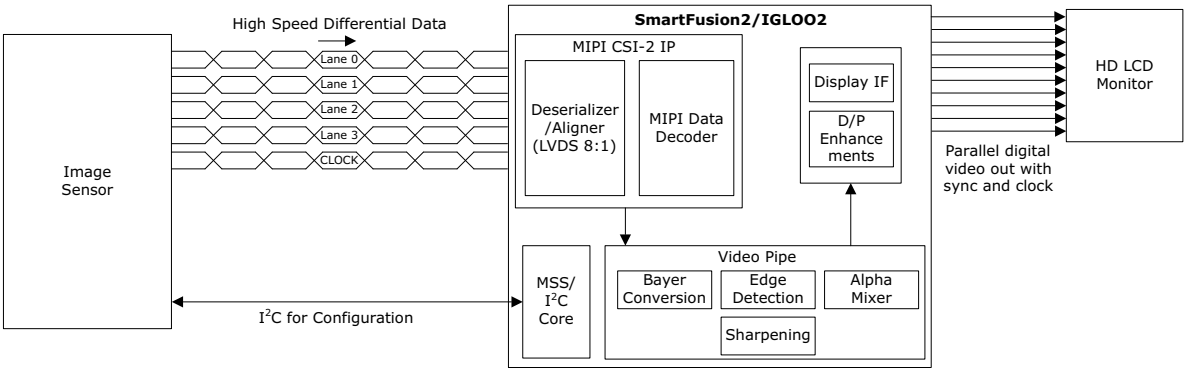
2.2 Reference

- *MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2)*
- *UG0557: SmartFusion2 SoC FPGA Advanced Development Kit User Guide*
- *DS0128: IGLOO2 and SmartFusion2 Datasheet*
- *http://www.microsemi.com/products/fpga-soc/technology-solutions/imaging*

3 Design Overview

The following figure shows a MIPI CSI-2 reference design block diagram for SmartFusion2 and IGLOO2 devices. This reference design includes a MIPI CSI-2 receiver that interfaces with a MIPI image sensor to de-serialize high-speed serial data to raw sensor parallel data. The received RAW Bayer-pattern pixel data is then converted to a 24-bit RGB format. There are several image processing steps before the RGB pixel data is sent to the display controller block. The display controller block generates the required display timing sync signals based on image output configurations.

Figure 2 • **MIPI CSI-2 Reference Design Block Diagram**



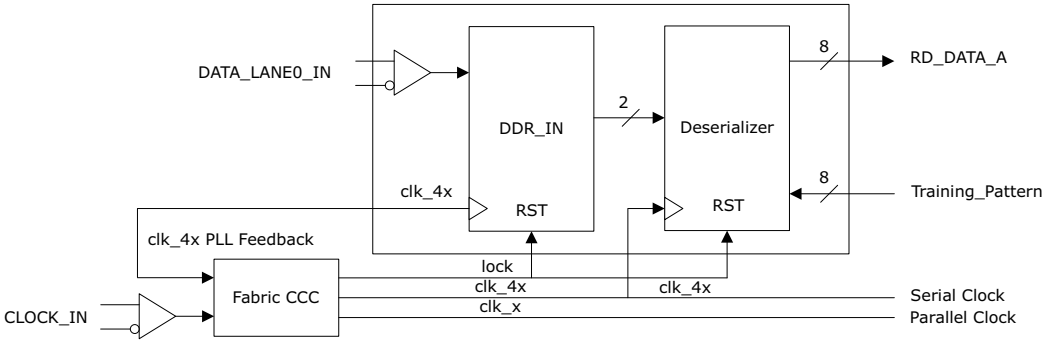
The MIPI image sensor shown in the preceding figure is configured over I^2C lines. In the case of SmartFusion2, the built-in MSS can be used for sensor configuration over I^2C. In the case of IGLOO2 devices, a CoreI2C IP can be instantiated in the FPGA fabric to be used for sensor configuration. The reference design example is implemented using an OnSemi AR-330 sensor and SmartFusion2 M2S150. A sample configuration set for configuring the sensor in 1280×800 resolution mode is provided in the `ar0330.c` file, which is included in the design project's SoftConsole software folder.

The MIPI CSI-2 reference design includes two main HDL blocks for receiving CSI-2 camera data: "LVDS RX 1:8" and "MIPI data decoder". The following sections describe each of these modules in detail.

3.1 De-serializer and Aligner Block (LVDS RX 1:8)

The De-serializer and Aligner block as implemented in SmartFusion2 and IGLOO2 devices for a single lane is shown in the following figure. This block receives high-speed serial MIPI data, de-serializes and converts it to 8-bit parallel data. The aligner block bit aligns the data to the correct 8-bit demarcations in the serial stream. The single lane example design can be extended to four lanes.

Figure 3 • **LVDS RX 1:8 Example Design Diagram**

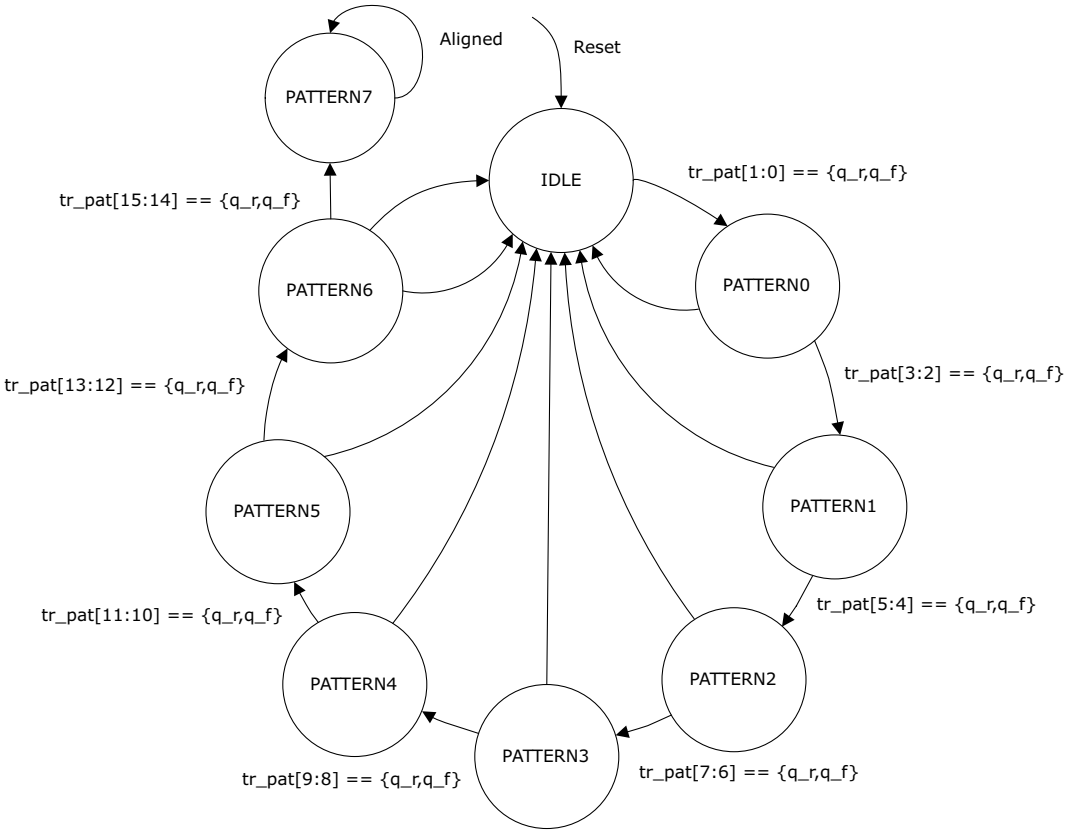


A DDR_IN Libero macro is instantiated in the design to clock data at both, positive and negative edges of the reference clock. A clock control PLL CCC macro is instantiated in the design to derive both, a

de-skewed serial clock and a bit parallel clock (/4 of serial clock) named `clk_x`. The clock conditioning circuit (CCC) is configured for a serial clock (named clk_4x) to be presented on the GL0 pin. This GL0 serial clock output is fed back as a feedback clock to the CCC to cancel out any routing delays that may skew the clock before capturing serial data inside the FPGA fabric.

The Deserializer block hunts within the incoming serial stream for short-packet reception defined by the MIPI protocol. The state machine looks for an identifying pattern, 2 bits at a time, clocked out of the DDR input block and moves to the next state machine as the pattern matching steps are followed. If at any time the pattern matching fails, the state machine moves back to an Idle state and starts all over again until a valid pattern is matched for all 16 bits. This state machine is shown in the following figure. Once a complete pattern is detected, the `align` signal goes up to indicate the next block, the sync decoder, that the 8-bit data from the Deserializer block is known to be aligned with the correct demarcation in the serial stream.

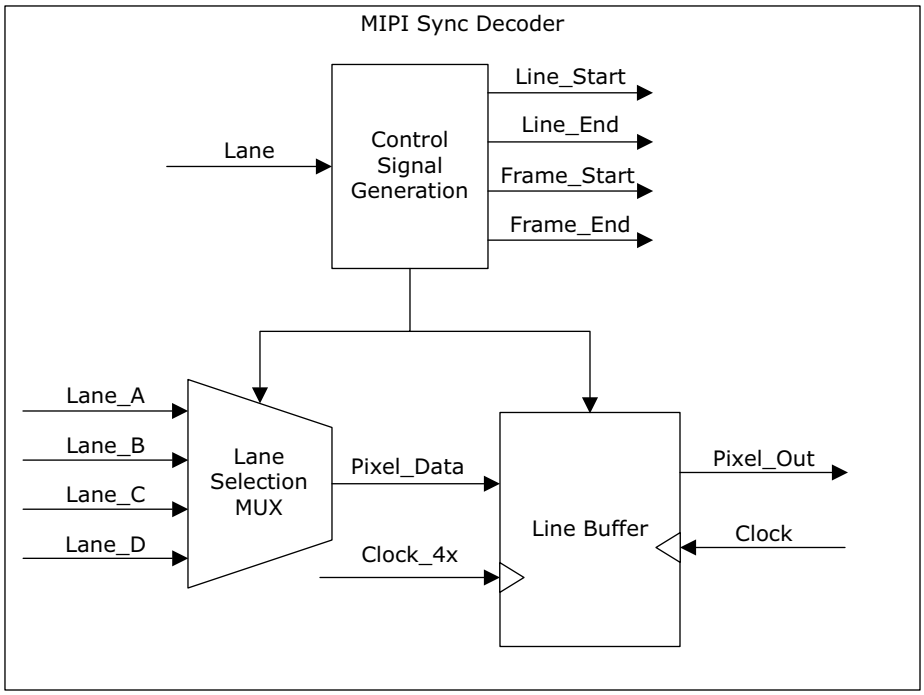
Figure 4 • **Data Aligner FSM Diagram**



3.2 MIPI Sync Decoder

The MIPI Sync decoder module block diagram is shown in the following figure. The sync decoder module decodes the Start of Line, Start of Frame, End of Line, and End of Frame of the image signals along with the RAW pixel data from the Image sensor. The sync decoder block also extracts and aligns data striped across multiple MIPI lanes in mulit-lane MIPI configurations.

Figure 5 • **MIPI Sync Decoder Block**

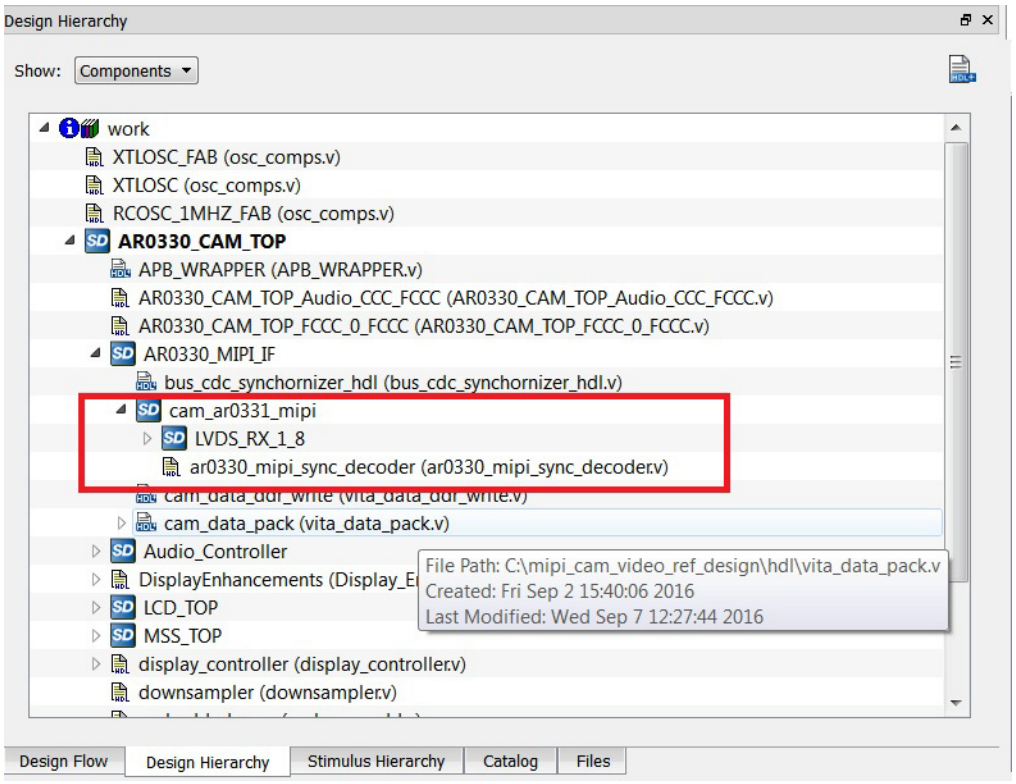


4 Libero Implementation

This section describes Libero implementation of a single-lane MIPI design that is available for download at the following location: http://www.microsemi.com/form/87-design-files-for-mipi-csi-2-camera-sensor-interface.

Once the reference design is opened in the Libero design software, navigating to the **Design Hierarchy** window shows the design hierarchy as shown in the following figure.

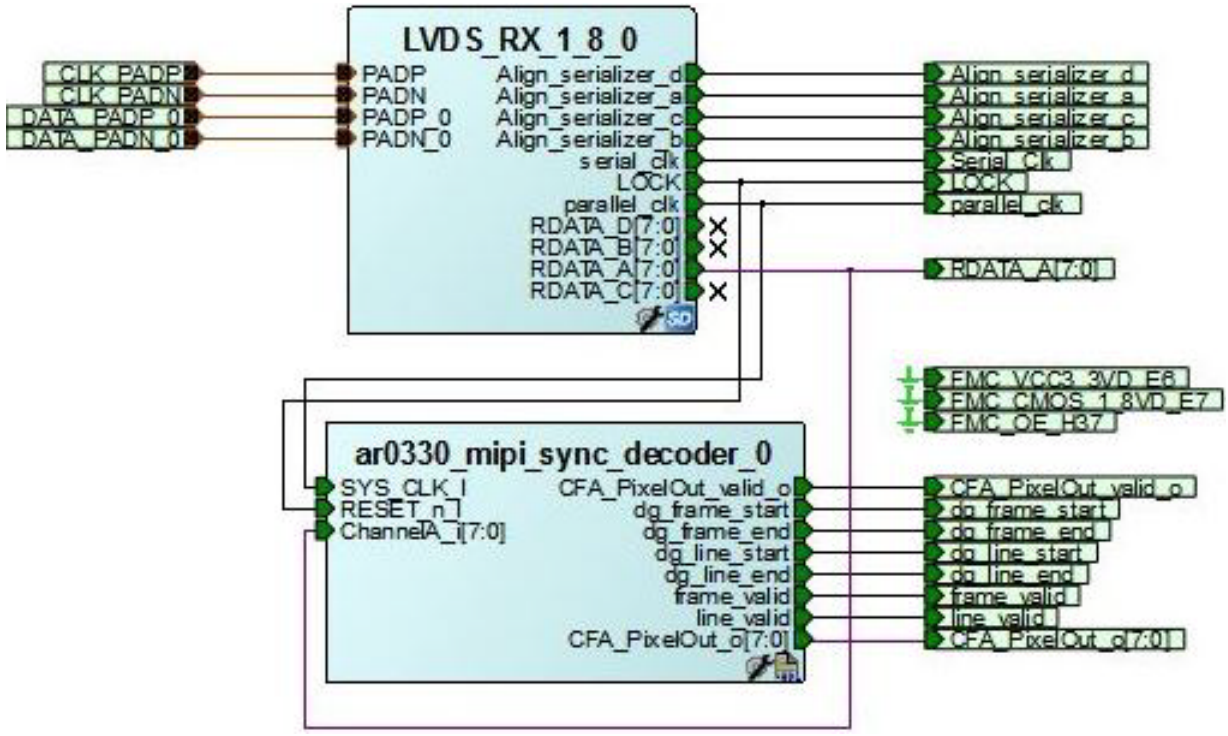
Figure 6 • **Design Hierarchy of MIPI Reference Design**



The highlighted section is the MIPI CSI-2 design module, which consists of two sub-modules: LVDS_RX_1_8 LVDS receiver and ar0330_mipi_sync_decoder MIPI Sync decoder block. As mentioned earlier, the LVDS receiver module performs serial-to-parallel data conversion, and the sync decoder block extracts image data information including start/end of video frame and lines.

The following figure shows a SmartHDL canvas view of a single-lane MIPI receiver and decoder implementation.

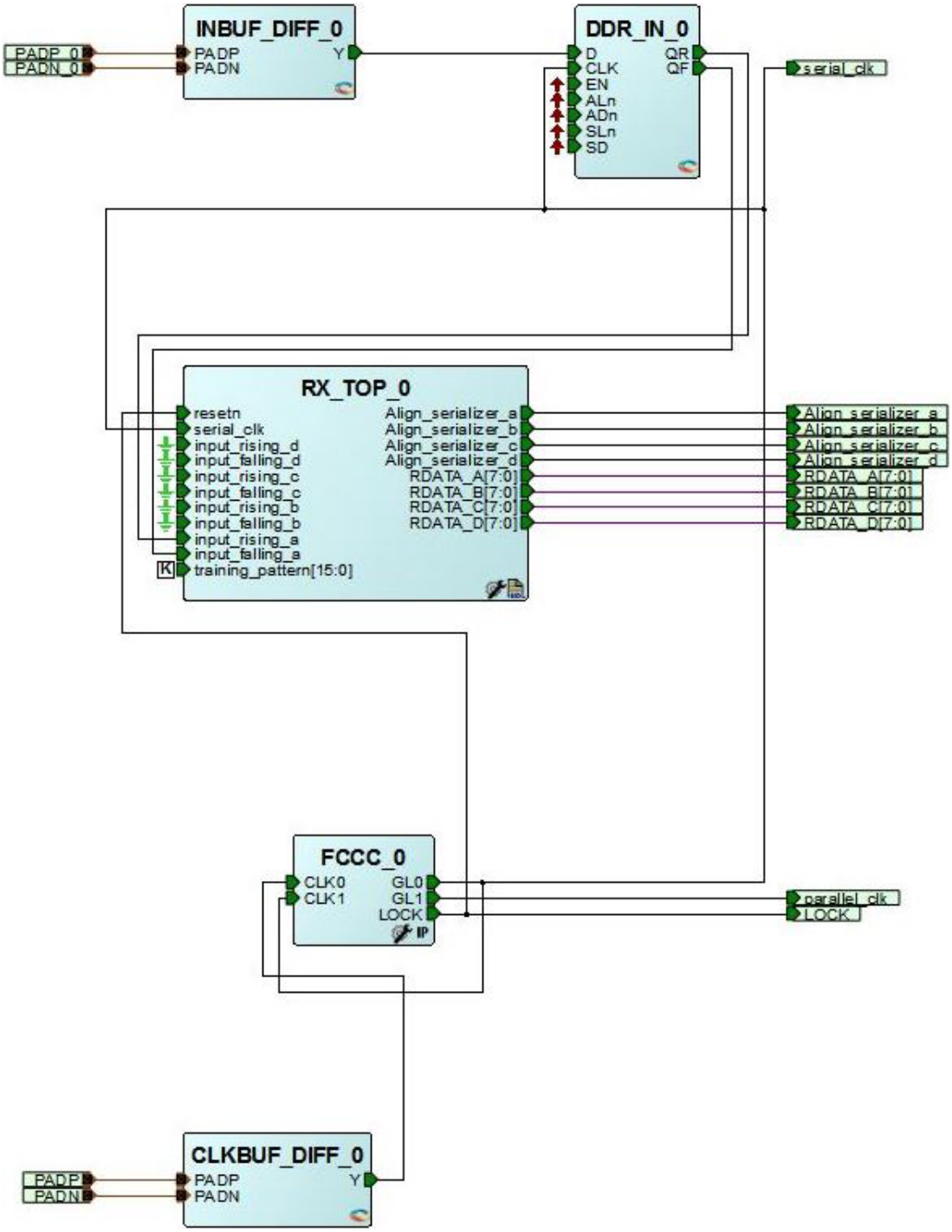
Figure 7 • **Top-level MIPI CSI-2 Libero Design per Single Lane**



In the preceding figure, CLK_PAD_P and CLK_PAD_N constitute a high-speed differential clock. DATA_PADP_0 and DATA_PADN_0 are high-speed differential data ports in DDR format. The LVDS_RX_1_8 module deserializes the high-speed serial data coming from the image sensor down to 8-bit parallel data and generates an align signal for each lane as shown in the preceding figure. The parallel data output and parallel clock outputs from the deserializer are routed to the ar0330_mipi_sync_decoder module, which decodes the parallel data and generates all of the required video timing signals such as Line_start, Line_end, frame_start, frame_end and extracts the RAW pixel data. The RAW8 pixel data is sent to the next block as CFA_PixelOut_o[7:0] along with a valid signal, CFA_PixelOut_Valid_o.

The following figure shows how to use INBUF_DIFF, DDR_IN, CLKBUF_DIFF, and FCCC blocks to interface with high-speed MIPI receive data.

Figure 8 • **LVDS RX 1:8 Libero Design per Single Lane**



MIPI signals use SLVS-200 signaling levels for high-speed transmission. These signals are received by differential I/O buffers of SmartFusion2 and IGLOO2 devices. The differential I/Os must be configured to LVDS 2.5 V. It is recommended to use MSIOD LVDS 2.5 V configuration to achieve best I/O performance. It is also recommended to select the differential MSIOD pairs close to each other to get deterministic and maximum performance.

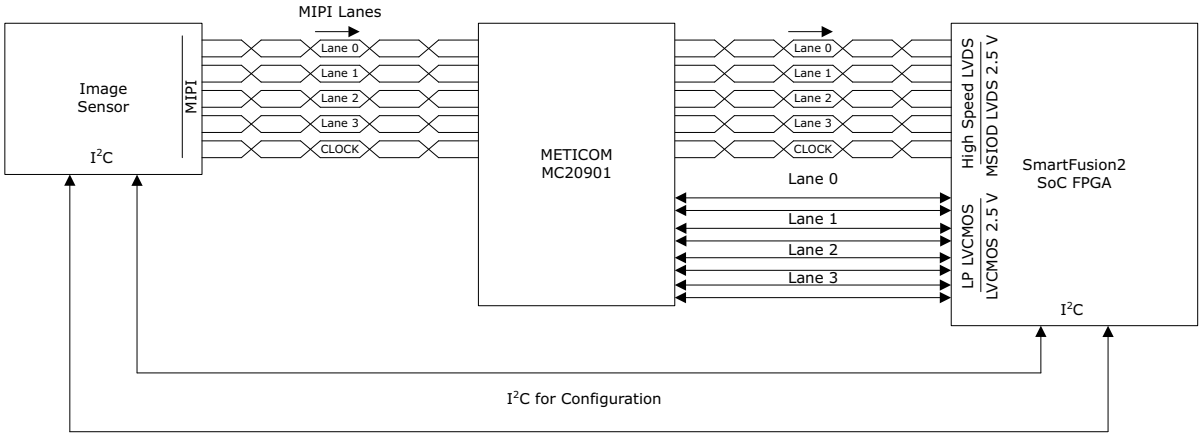
It is recommended to route back the serial clock output of the CCC as feedback to the PLL to cancel out clock skews introduced by global clock buffers and internal clock routing delays that might introduce clock skews.

5 Limitations of MIPI CSI-2 on SmartFusion2 and IGLOO2 Devices

1. MIPI CSI-2 can support speeds of up to 1Gbps or higher in high-speed mode. SmartFusion2 and IGLOO2 devices, however, can support only a maximum speed of 700Mbps on LVDS I/Os (MSIODs). Also, extreme care must be taken to match clock-data skews both external to the FPGA and inside the FPGA, to achieve the best performance.
2. MIPI CSI-2 supports two modes, namely high speed (HS) and low power (LP) modes. In low power modes all differential signals act as single-ended LVCMOS signals and require additional I/O circuitry to switch between LP and HS modes. Only HS mode can be supported by SmartFusion2 and IGLOO2 devices using LVDS I/Os.

When both, HS and LP modes must be supported, it is recommended to use an external device such as Meticom MC20901 DPHY chip to convert SLVS MIPI I/O signals to LVDS for HS data and LVCMOS low power signals. The following example block diagram shows how to use the Meticom device to interface an external Image sensor with SmartFusion2 and IGLOO2. In this example, the high-speed signals are routed to the MSIOD LVDS 2.5 V differential pairs of the FPGA while the low-power signals are routed to LVCMOS 2.5 V signals. Also, care should be taken to route high-speed signals to the same FPGA I/O bank and the I/O selection must minimize clock-data skew.

Figure 9 • **Using an External MIPI Interface Device to Support both HS and LP Modes**



6 Resource Utilization

The following table shows resource utilization for implementing MIPI CSI-2 receive only in the SmartFusion2 and IGLOO2 family of devices. These design utilization numbers do not include image processing IP that may be required to further process received sensor data. This table shows information for only the Deserializer/Aligner and MIPI decoder blocks:

Table 1 • **Resource Utilization for Implementing MIPI CSI-2 Receive only in the SmartFusion2 and IGLOO2 Devices**

Design	LUT	FF	DSP	RAM64x18
CSI-2-RX 1 Lane	499	610	0	4
CSI-2-RX 2 Lane	529	689	0	4
CSI-2-RX 4 Lane	729	889	0	4

7 Conclusion

This application note provides an overview of how to use SmartFusion2 and IGLOO2 devices to design MIPI CSI-2 interfaces. It also discusses a few limitations of SmartFusion2 and IGlOO2 devices in this context, and offers alternatives to overcome them.