AC469
Application Note
PolarFire FPGA: Video and Imaging Solution

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
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: 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, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

51900469. 1.0 3/18

Contents

Figures

Tables

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 PolarFire FPGA Video and Imaging Solution

This application note describes a reference design that demonstrates interfacing MIPI-based camera, real-time video processing using Video Image Signal Processing (ISP) IPs and displays real time processed video using HDMI display controller. It also demonstrates the usage of MIPI IO functionality and basic video processing capabilities of Microsemi Libero® SoC Video ISP IPs. This reference design is created with Microsemi Libero SoC PolarFire®, which includes Mi-V ecosystem and SoftConsole v5.2.

This reference design supports:

- Interfacing MIPI-based cameras using configurable MIPI_CSI2 RxDecoder IP
- Writing and Reading the Video/Image data into DDR using Arbiter logic IP
- Interfacing with DDR memory to store frame by frame data
- Converting Video/Image Bayer CFA to RGB data
- Processing Video/Image data using basic ISP IPs

The processed Video/Image data is displayed on a HDMI-based monitor using display controller IP.

This reference design is built for PolarFire Evaluation Kit and MIPI-based Imaging and Video daughter card. PolarFire Evaluation Kit offers a full featured 300 K LE FPGA device. This 300K LE device inherently integrates reliable flash-based FPGA fabric, a Mi-V soft processor core, advanced data security features, digital signal processing (DSP) blocks, static random-access memory (SRAM), embedded non-volatile memory (eNVM), and industry-required high-performance communication interfaces-all on a single chip.

2.1 Features

This reference design offers the following features:

- Interfacing MIPI-based camera configuration (1x, 2x, 4x Lane, Raw8 format) and evaluation at various lanes.
- 2 GB DDR memory interface.
- Mi-V soft processor core and its peripheral Interfaces—SPI, GPIO, UART, SPI Flash, and I2C
- HDMI Interface—display the video on HDMI based Monitors.
- Graphical User Interface (GUI).

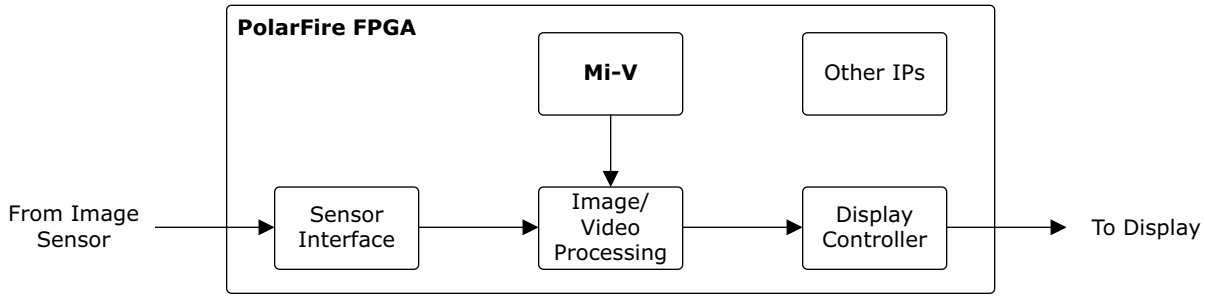
The design uses the following Video-ISP IP cores:

- Bayer to RGB conversion IP
- Color space conversion (RGB2YCbCRr and YCbCr2RGB) IP
- Image sharpening Filter IP
- Image edge detection IP
- Alpha blending IP
- Brightness, contrast, and hue adjustment favor IP

2.2 System Architecture

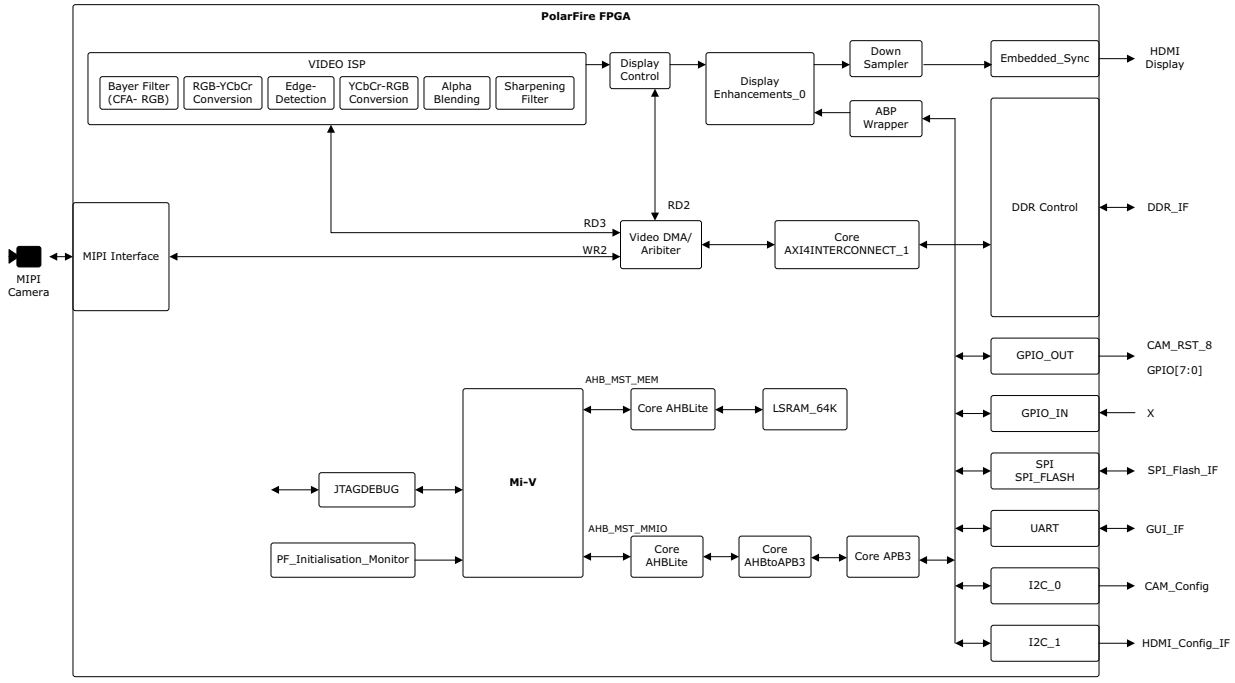
The basic system architecture consists of three main modules—Sensor Interface, Video/Image Signal Processing (ISP) module and Display Controller module. This flexible modular approach in designing the solution supports various targeted applications by changing to the suitable camera specific and display interfaces using suitable ISP IPs. The following figure shows the system architecture of the reference design.

Figure 1 • **High-level System Architecture**



The present solution demonstrates the interfacing of MIPI-based camera and processing a real-time video using basic Video ISP IPs and display on an HDMI-based monitor. The solution can be used by varying the Video ISP IPs as per needs of the targeted application. The following figure shows the detailed system design block diagram.

Figure 2 • **Detailed System Design**



The reference design is mainly built around Mi-V soft processor core, Video DMA/Arbiter logic, Video ISPs, DDR, and Display controller. As shown in Figure 2, page 3, the MIPI-based camera is interfaced to the PolarFire FPGA. The MIPI CSI-2 RX IP core decodes the received serial data and writes the data into CDC FIFO, which is based on the MIPI configuration set to 1x or 2x or 4x Lane and Raw8 format. The MIPI Camera/Senor interface module consists of PLL0 clock, which is set in-line to the lane selection of a specific camera used. See Clocking, page 7 for more information about PLL0 configuration values set with in MIPI Interface for AR330 sensor.

The Arbiter logic IP is used to write the CFA data into the DDR memory using its write channel interface and to read out the data using its one of the read channel interface. The main PLL2 is configured to generate a DDR clock and a processor clock.

The Video ISP module processes the CFA video data received from DDR through Arbiter logic IP and converts to RGB format data using Bayer Interpolation IP. In this design, Video ISP is pipelined and designed with two data path flows. MUX is used to select one of the data flow path to show the Edge detection and normal video function, as shown in Figure 4, page 5. Selection of functions is through GUI, where the processor receives the command message from GUI and sends the select input to high/low using GPIO interfaced to Edge_Sel signal. Alpha blending IP is set to default in the design. The video data is passed through the Image sharpening filter IP, where the K-factor input value can be varied through GUI via APB wrapper Interface.

The Display interface module consists mainly of Display Controller IP and Display enhancement IP. Display controller IP is used to generates the sync signals and synchronizes the input video data. Display Enhancement IP is used to read the input values from GUI and vary the control parameters like brightness, contrast, and hue of the video display via APB wrapper Interface. The PLL3 is configured to generate a HDMI clock based on the required resolution. The output of the video is displayed on an HDMI-based monitor, which is connected to HDMI port of MIPI CSI2 daughter card through HDMI cable, where the daughter card is connected to the FMC connector of the PolarFire Evaluation Kit board.

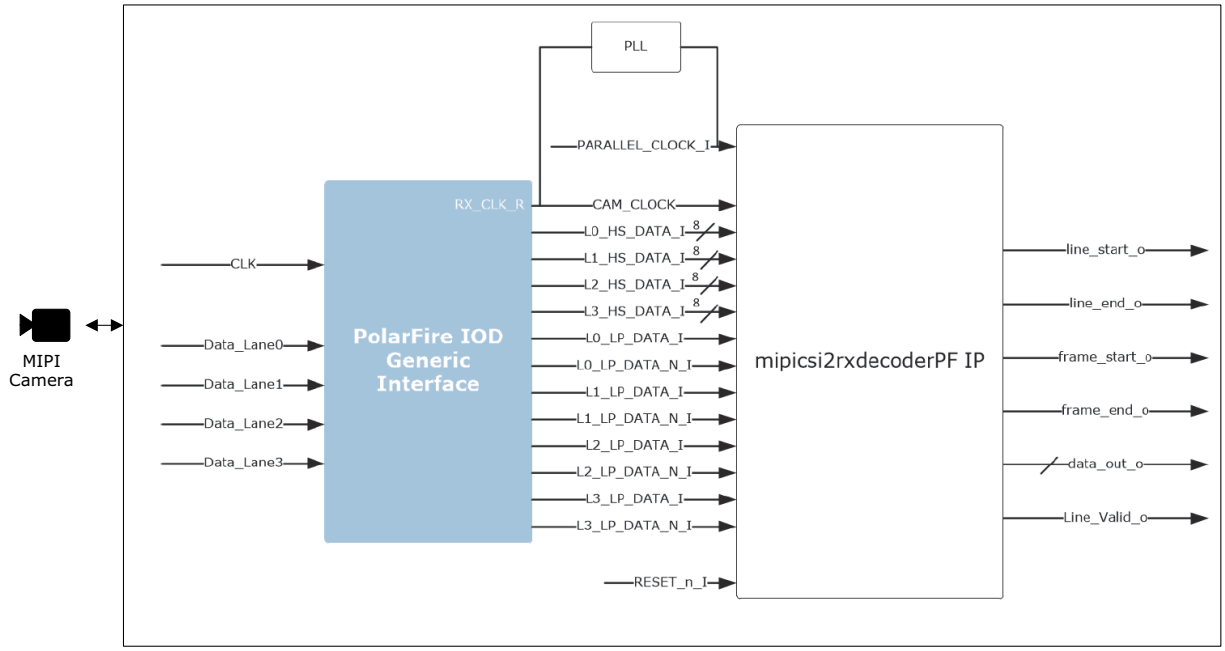
2.3 Subsystem Components

The reference design has three main components—Sensor Interface, Video ISP, and Display Interface.

2.3.1 Sensor Interface

Sensor interface is the front-end component that implements the sensor interfacing and its related configurable data decoding IP. In the reference design, the MIPI CSI-2 based camera sensor is connected with the PolarFire device using the dedicated MIPI IOs. The MIPI CSI-2 RxDecoder IP is used to interface the MIPI sensor for receiving and decoding the serial CFA data. The following figure shows the sensor interface.

Figure 3 • **Sensor Interface**



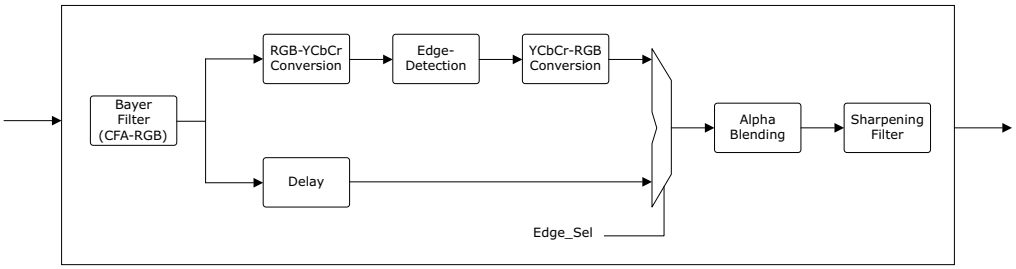
2.3.1.1 MIPI CSI-2 RX IP

MIPI CSI-2 RxDecoder IP is used along with PolarFire IOG blocks to decode the MIPI CSI-2 packets. The IP decodes both HS long and short packets. It takes the serial MIPI CSI-2 data from camera and gives out the parallel pixel data along with valid signals line_valid_o (which validates the pixel data data_out_o), frame_start_o, frame_end_o, line_start_o and line_end_o. For more information about the MIPI CSI2 RxDecoder IP, see *UG0806: MIPI CSI-2 Receiver User Guide*.

2.3.2 Video/Image Signal Processing

The Video Processing Engine (VPE) or Image Signal Processing (ISP) is a sub-system component used to process the CFA data using Video/Image ISP IPs and transmits the processed data to the display controller. Based on the targeted application, you can integrate various IPs to process the Video/Image in real time. The following figure shows the Video ISP.

Figure 4 • **Video ISP**



2.3.2.1 Bayer IP

When the camera sensor sends the data in bayer format, Bayer Interpolation IP is used to convert the bayer data to RGB data, which implements Freeman demosaicing algorithm. It is a two-stage process that combines bilinear interpolation and median filtering. To minimize the zippering artifacts introduced by bilinear demosaicing, Freeman algorithm utilizes color difference such as, red minus green and blue minus green, for median filtering. The filtered differences are then added back to the green plane to obtain the final red and blue planes. In this method, fringes at the edges of different color areas can be eliminated. For more information about Bayer IP, see *UG0640: Bayer Conversion User Guide*.

2.3.2.2 Color Space Conversion IPs

A color space is a mathematical representation of a set of colors. The most popular color models are:

- RGB—used in computer graphics
- YIQ, YUV, and YCbCr—used in video compression

The objective is to convert the video inputs into the desired color space before performing any video processing on it. For more information about color space conversion, see *UG0639: Color Space Conversion User Guide*.

2.3.2.3 Edge Detection IP

The purpose of edge detection is to significantly reduce the amount of data in an image and preserve the structural properties for further image processing. In a gray level image, the edge is a local feature with in a neighborhood separate region. For more information about Edge detection, see *UG0693: Image Edge Detection User Guide*.

2.3.2.4 Alpha Blending IP

Alpha blending is the process of combining an image with a background to create the appearance of partial or full transparency. It is used to render multiple images into a single background image in separate passes and make one final image. For more information about Alpha Blending, see *UG0641: Alpha Blending User Guide*.

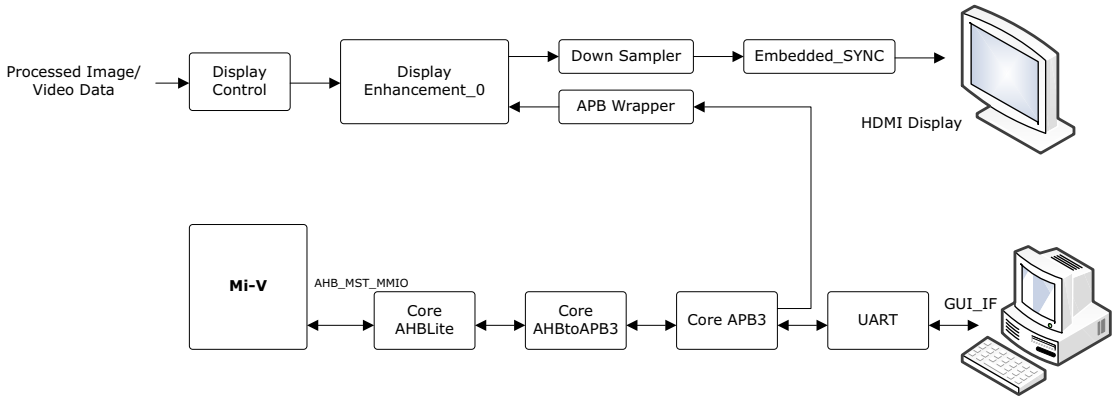
2.3.2.5 Image Sharpening Filter

Image sharpening encompasses any enhancement technique that highlights the edges and fine details of an image. Image sharpening is done by adding to the original image a signal proportional to a high-pass filtered version of the image. This process, referred to as unsharp masking on a one-dimensional signal, involves two steps. The original image is first filtered by a high-pass filter that extracts the high-frequency components. A scaled version of the high-pass filter output is then added to the original image, thereby producing a sharpened image. For more information about Image Sharpening Filter, see *UG0642: Image Sharpening Filter User Guide*.

2.3.3 Display Interface

The Display interface is a sub-system module consists mainly of Display Controller IP and Display enhancement IP. In this design, a Mi-V processor is used to read the input values from GUI and vary the control parameters like the brightness, contrast, and hue of the video display via APB wrapper Interface. The following figure shows the Display interface.

Figure 5 • **Display Interface**



2.3.3.1 Display Enhancement IP

Display Enhancement module enables to adjust the brightness, contrast, saturation, and hue of the final video display to personal preferences. These adjustments are done on YCbCr color space as calculations are simpler compared to RGB domain. For more information about Display Enhancement, see *UG0646: Display Enhancement User Guide*.

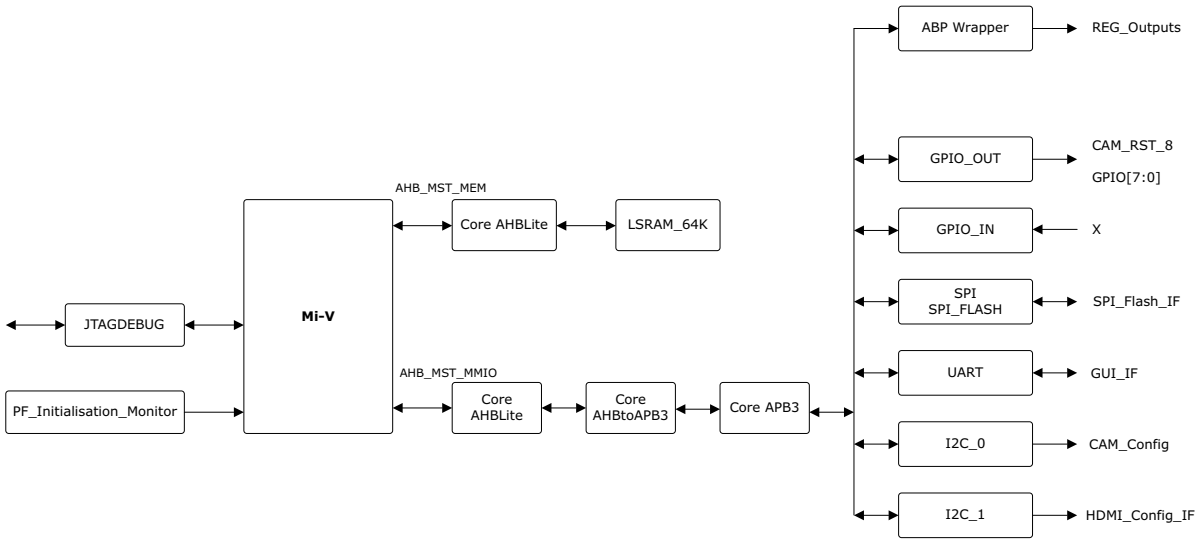
2.3.3.2 Display Controller IP

The display controller generates display synchronization signals based on the set display resolution. The synchronization signals are horizontal sync, vertical sync, and data enable. The input video data is also synchronized with these sync signals. The sync signals along with video data can be fed to a DVI, HDMI, or VGA card that interfaces with the display monitor. For more information about Display Controller, see *UG0649: Display Controller User Guide*.

2.3.4 Processor System

The following figure shows the Mi-V based processor system and its interfaces. In this design, the processor operates at 50 MHz clock and uses the on-chip LSRAM of 64K to execute the application program.

Figure 6 • **Mi-V based Processor System**



2.3.5 Memory

In this design, DDR is used to store up to four image frames, which are written and read by the Arbiter IP. An on-board SPI Flash is utilized to store the Software image. On power-up, the in-build device controller reads the software image from SPI Flash and writes into LSRAM, where the Mi-V processor executes the application.

2.3.6 Clocking

The following table lists the PLLs configuration utilized in the design.

Table 1 • **PLL Input and Output Clock**

PLL	Reference Input Clock	PLL Output Clock
PLL0 (MIPI - 4-Lane)	Input_Frequecy: 19.5 MHz (IOD generated Clock output)	Output_Clock_0: 19.5 MHz (CAM clock) Output_Clock_1: 78 MHz (Parallel Clock)
PLL1	DDR controller utilizes one PLL. Not applicable for configuration.	
PLL2	Input_Frequecy: 50MHz (On-board Clock - Clk_in)	Output_Clock_0: 166.6667 MHz (DDR input clock) Output_Clock_1: 50 MHz (HDMI Clock for display)
PLL3	Input_Frequecy: 166.666 MHz (DDR generated Sys_Clk as input to PLL3)	Output_Clock_0: 74.25 MHz (HDMI Clock for Display)

2.4 Design Requirements

The following table lists the hardware and software requirements.

Table 2 • **Design Requirements**

Design Requirement	Description
Hardware	
Imaging and Video MIPI-CSI2 Daughter Card	VIDEO-DC-MIPI
PolarFire Evaluation Kit[1]	MPF300TS-1FCG1152EES Device
Image sensor module	ONSemi AR0330 Image Sensor from Leopard Imaging (LI-AR0330-MIPI v1.1)
Image sensor ribbon cable	
Mini USB to Type A USB cable[2]	
HDMI cable	HDMI A Male to Male Cable
HDMI monitor[3]	Any 21-inch display with HDMI input
USB micro AB connector[2]	
Power adapter (T1121-P5P-ND)[2]	
Operating system	Windows 7 or later
Software	
Libero SoC PolarFire	v2.0
SoftConsole	v5.2
FlashPro programming software	v12.200.0.20

1. Not shipped with the Imaging and Video MIPI CSI-2 Daughter Card; must be purchased separately.
2. Included with PolarFire Evaluation Kit.
3. If the display does not support HDMI input, use an adapter that converts the HDMI out from the imaging card to a protocol supported by the display.

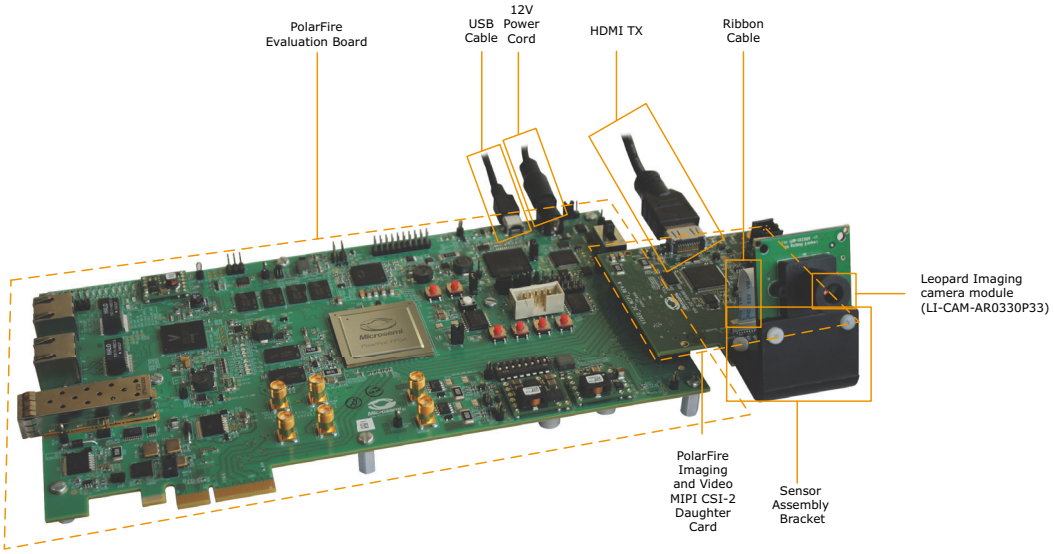
2.5 Using the Reference Design

The following sections describes the required hardware setup.

2.5.1 Setting up the Polar Evaluation Board

The following figure shows the complete setup of the Imaging and Video MIPI CSI-2 Daughter Card and the PolarFire Evaluation board.

Figure 7 • **Board Setup with Imaging and Video MIPI CSI-2 Daughter Card**



2.5.2 Setting Up the Hardware

The hardware setup for the reference design involves establishing appropriate hardware connections between the Imaging and Video MIPI CSI-2 Daughter Card and the PolarFire Evaluation Board.

Figure 7, page 9 shows the complete hardware setup. The following sections show the hardware setup for the Imaging and Video MIPI CSI-2 Daughter Card and the PolarFire Evaluation Board separately.

2.5.2.1 Setting Up the PolarFire Evaluation Board

1. Connect the power supply cable to the **J9** connector.
2. Close pins 3-4 of **J12** to select the 2.5 V core voltage.
3. Connect the USB cable from the host PC to **J5** (FTDI port).
4. Make sure the default jumper settings shown in the following table are retained.

Table 3 • **Jumper and Resistor Settings for MPF300TS Device**

Jumper/Resistor	Setting
J18, J19, J20, J21, J22	Closed
J28	Open
J26	Closed
J4	Closed
J12	Closed
J23	Closed

5. Power-up the board using the SW3 slide switch.

For more information about how to program and run the demo, see *DG0807: PolarFire Imaging and Video Kit Demo Guide (MIPI CSI-2 Camera Sensor)*.

2.6 System Register Map

The following table lists the base addresses for various components in the subsystems.

Table 4 • **System Register Map**

Subsystem/Component	Base Address
Core_AXI4Interconnect - Peripheral-PF_DDR:AXI3 Slave Address	0x00000000
Core AHBLite - Peripheral - LSRAM: AHB Slave Interface	0x80000000
Core AHBLite - to APB3 - Peripheral- AHB Slave	0x70000000
Core APB3 - Peripheral - Core UARTapb	0x70001000
Core APB3 - Peripheral - Core GPIO_INapb	0x70002000
Core APB3 - Peripheral - RESERVED	0x70003000
Core APB3 - Peripheral - RESERVED	0x70004000
Core APB3 - Peripheral - Core GPIO_outapb	0x70005000
Core APB3 - Peripheral - Core SPI_0_outapb	0x70006000
Core APB3 - Peripheral - Core I2C_0_apb	0x70007000
Core APB3 - Peripheral - Core I2C_1_apb	0x70008000
Core APB3 - Peripheral - Core apb_wrapper	0x70009000