Video IO using the APIX2 HSMC card and a Cyclone V SoC Development Kit

Date: 7/11/2017

Revision: 1.2

Table of Contents

Introduction

This example design demonstrates using the APIX2 HSMC card from Inova Semiconductors in conjunction with an Altera Cyclone V SoC development kit. The APIX2 HSMC card and cameras incorporate Inova APIX2 transmitter and receiver ICs which utilize high speed serial links to transport audio, video and control and status information. The design demonstrates generation of video output using an APIX2 INAP375T transmitter IC in conjunction with an APIX2 RX HDMI card. It also demonstrates the capture and display of two live video streams from APIX2 equipped cameras. Information about APIX2 devices and boards can be found at

Devices:

- APIX2 transmitter: http://www.inova-semiconductors.de/en/products/apix/inap375t.html
- APIX2 receiver: http://www.inova-semiconductors.de/en/products/apix/inap375r.html

Boards:

- APIX2 HSMC card: http://www.inova-semiconductors.de/en/products/tools/hsmc.html
- APIX2 RX HDMI card: http://www.inova-semiconductors.de/en/products/tools/inap375r-apix2-development-kit.html

Cameras

- https://www.first-sensor.com/cms/upload/datasheets/DS_Standard-Blue_eagle_DC3K-1-LVD_E_5024-310.pdf

Theory of Operation

A block diagram of the example design system is shown in Figure 1 and the FPGA block diagram is shown in Figure 2.

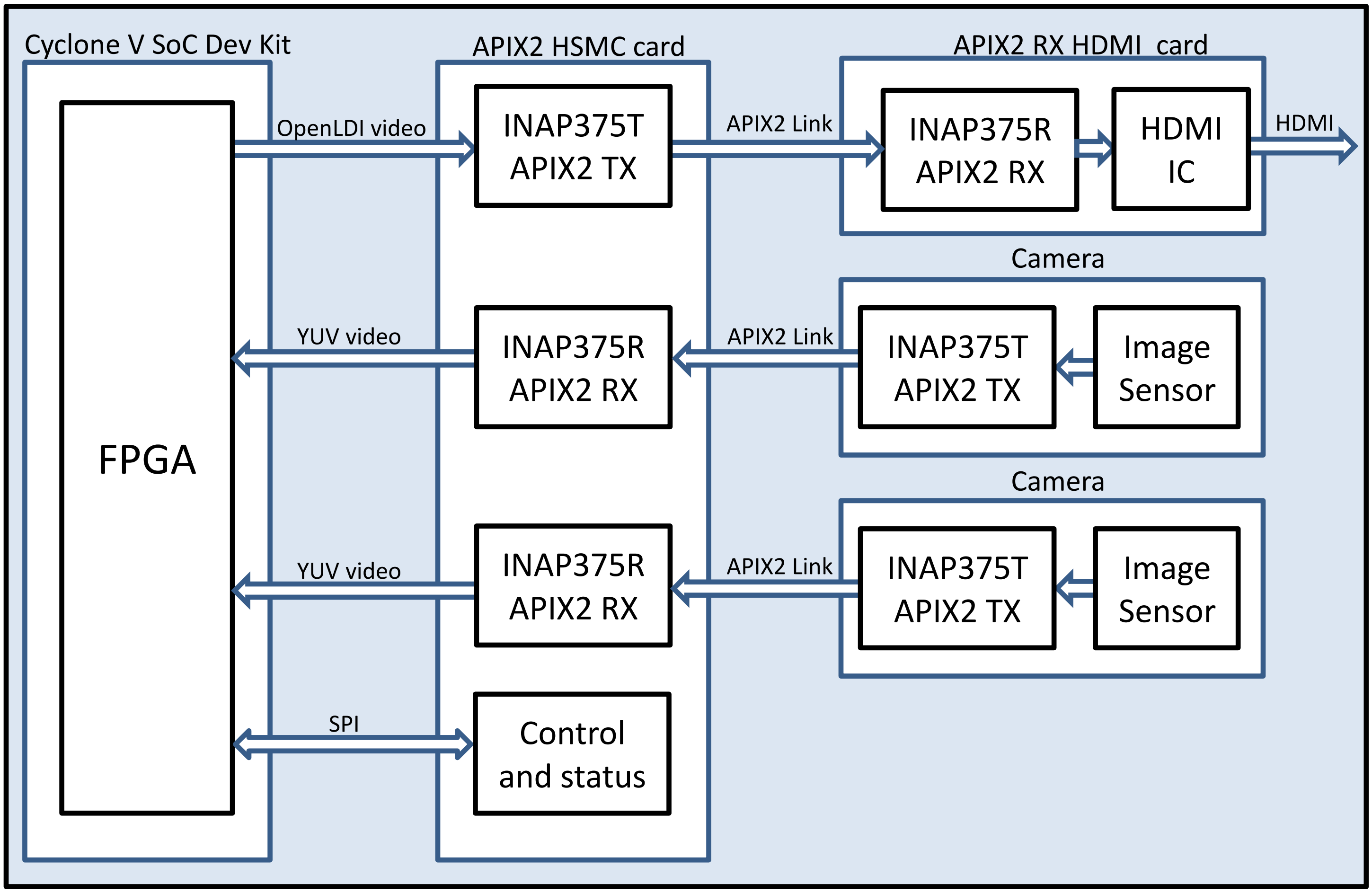

Figure 1. System Block Diagram

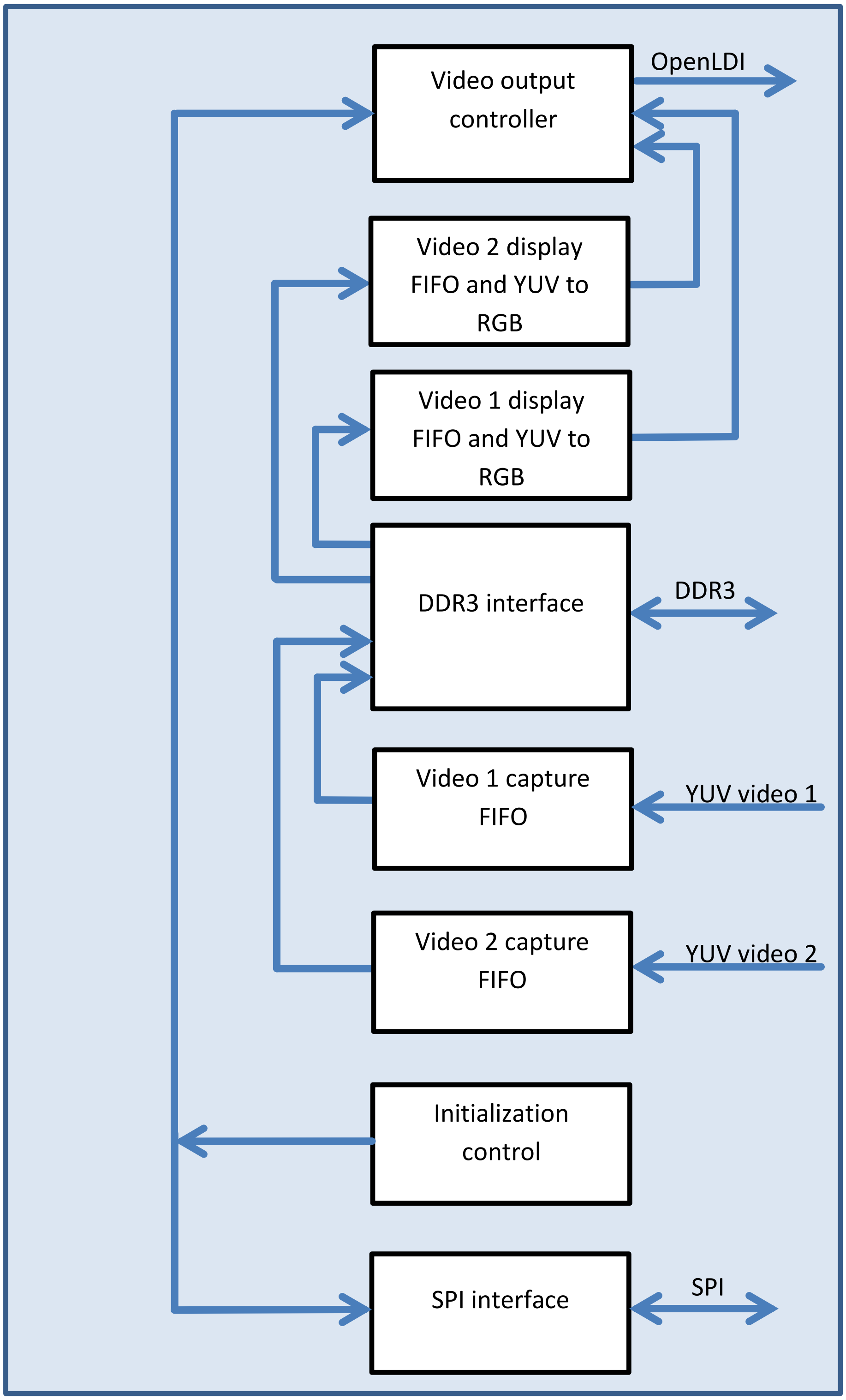

Figure 2. FPGA Block Diagram

This example design does not use the ARM subsystem of the Cyclone V SoC FPGA and only uses the FPGA fabric. On power up, the FPGA is configured from the on board configuration flash. After configuration the FPGA drives the SPI ports on the APIX2 HSMC card to initialize the APIX2 TX devices, APIX2 RX devices, and the image sensors. APIX2 links between the APIX2 HSMC card and the HDMI card and cameras provide access to remote APIX2 device registers. The image sensors on the cameras are configured via an I2C master on the APIX2 TX IC. During initialization each APIX2 device has a few tens of registers programmed and each image sensor has about 2000 registers programmed.

The image sensors are programmed to output YUV 422 10-bit video at 1280x800 resolution at 30 fps. Only the 8 most significant bits of the video are captured. Each video input stream is captured into DDR3 memory connected to the FPGA. Two frame buffers are utilized to prevent image tearing. The most recently completely captured frame is the one read by the video display controller. A stable clock detector is used to keep the video capture logic in reset until a stable clock of expected frequency is found.

The APIX2 TX device on the APIX2 HSMC card is programmed to receive OpenLDI RGB video from the FPGA which is generated at a 1280x800 resolution at 60 fps. The OpenLDI specification can be found at www.inova-semiconductors.de/files/inova/images/standards/openldi.pdf. The video output controller can output a fixed RGB test pattern, video 1, video 2, and combinations of video 1 and video 2. It can automatically sequence through the various output options or be controlled manually. The video output controller operation is controlled by pushbuttons on the Cyclone V SOC development board.

Installing the example design

The example design requires the following hardware and software.

- Quartus Prime version 16.1.
- Cyclone V SoC development kit.
- Inova APIX2 HSMC card v2.0.
- Inova APIX Receiver HDMI card.
- Inova APIX2 link cable.
- Two Inova APIX2 cameras with link cables.
- HDMI display and cable.
- USB cable for JTAG.

Figure 3 shows the assembled system (sans power supplies) and switch settings.

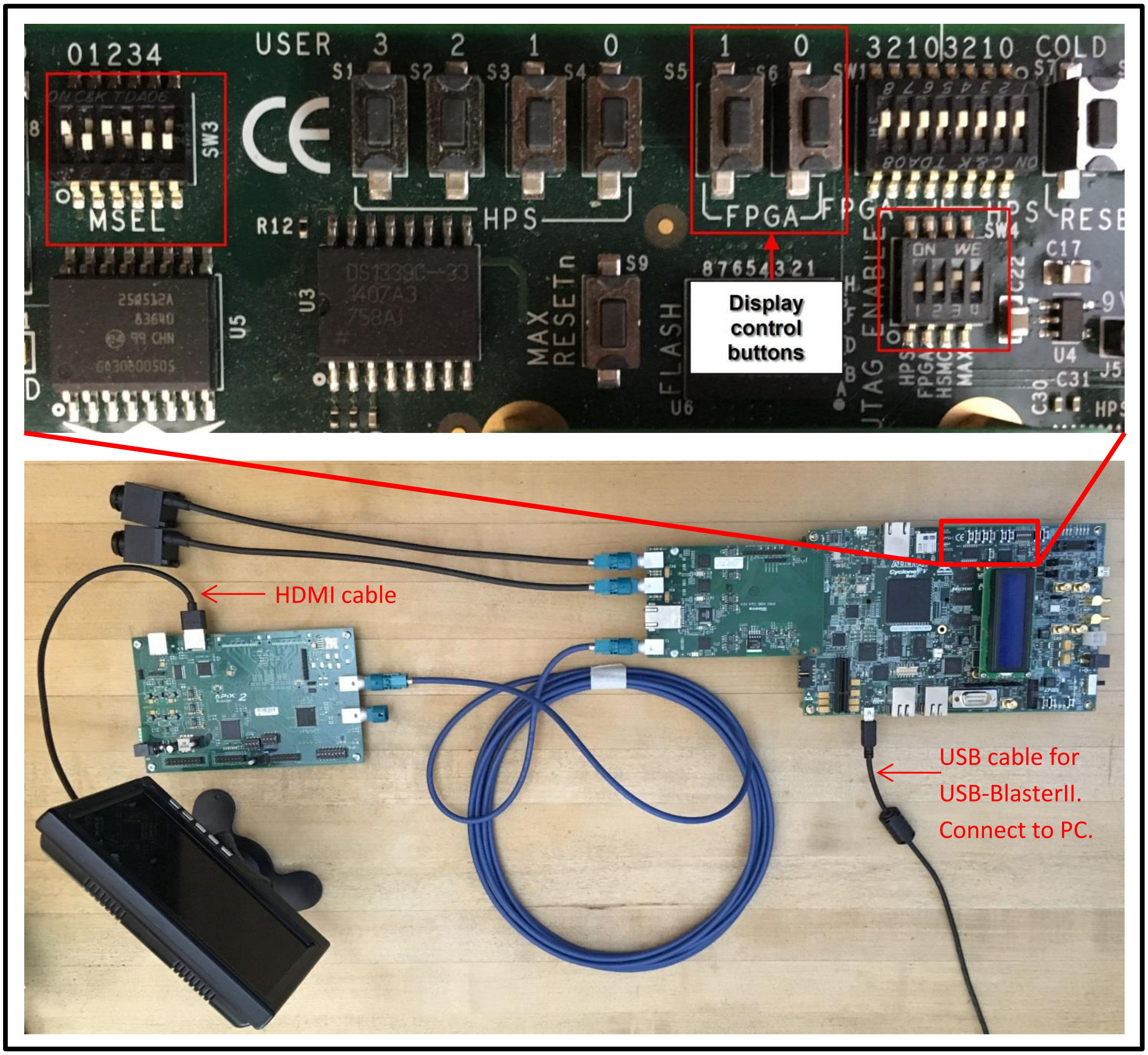

Figure 3. System assembly and switch settings.

The steps required to install the example design are

1. Assemble the system. Refer to Figure 3 for correct assembly and switch settings.
2. Download the design from https://cloud.altera.com/devstore/. Click on “Design Examples” and enter “APIX2” in the search box. Click on “Video IO Using the APIX2 HSMC Card”. Locate the Installation Package section. Click on the Download button. This will download the video_io_using_the_hsmc_card.par file to your computer.
3. For Quartus version 16.0 or later, double click on the par file. This will launch Quartus and unpack the project.
4. Power up the APIX2 Receiver HDMI board and the Cyclone V SoC development kit.

5. Program the FPGA configuration flash. The configuration flash file inova_demo.jic has been prebuilt and is part of the Quartus project. In Quartus, from the File menu, open inova_demo_spi_flash.cdf. This will launch the Programmer in a separate window. In the Programmer window, make sure the Hardware Setup is USB-BlasterII. Click on the "Hardware Setup…" button if not and configure for USB-BlasterII. Click on ./inova_demo.jic and then Start to initiate the programming process. An FPGA design that will interface with the configuration flash is first downloaded and takes a few seconds. The configuration flash is then programmed which takes a few minutes. Close the Programmer window when the process is complete. Quartus can also be exited at this point.

Using the example design

To use the example design first cycle power on the Cyclone V SoC development board which will initiate FPGA configuration from the configuration flash. After configuration the LEDs on the board will blink in a counting pattern. After a few seconds the initialization process will complete and the HDMI display will cycle through the following display modes:

- Fixed RGB test pattern.
- Live stream from camera 1.
- Live stream from camera 2.
- Live stream from both cameras with an animated wipe between the two streams.

Pressing Button 1 (refer to Figure 3) will toggle between automatic and manual cycling between display modes. When manual cycling is selected pressing Button 0 will advance to the next display mode.

Rebuilding the example design

To rebuild the example design, open the project in Quartus and select "Start Compilation" from the Processing menu or clock on the "Start Compilation" button in the menu ribbon. This will take a few minutes to complete. The result of this process is a configuration file output_files/inova_demo.sof. This file can be downloaded to the FPGA using the Quartus Programmer. With the system powered up, open the file output_files/inova_demo.cdf in Quartus. This will launch the Programmer in a separate window. Click the Start button which will download the sof file into the FPGA and restart the FPGA using the new configuration. If the Cyclone V SoC board is power cycled, the FPGA will configure from the configuration flash and the previous configuration is lost. Once the design is satisfactory, the sof file can be converted to a format that can be programmed into the configuration flash. To do this, select "Convert Programming Files…" from the File menu in Quartus which will launch a separate window. Click the "Open Conversion Setup Data" button and select the "create_config_flash_file.cof" file (it may be up or down the file hierarchy) and click the Generate button. This will generate the inova_demo.jic file. Follow the directions in step 5 in the "Installing the example design" section to program the configuration flash.

Installing the loopback diagnostic design

An additional example in the design store uses a loopback configuration to test the hardware. In this diagnostic design, a video stream of pseudo random 24-bit RGB pixels is generated and transmitted by the TX IC and looped back to the APIX2 RX1 IC by connecting a cable from the TX connector to the RX1 connector. To configure the hardware for the loopback diagnostic design:

1. Modify the hardware assembly as shown in Figure 3 by disconnecting the link cable from the HDMI board and the link cable from the RX1 connector on the HSMC card. Connect the link cable from the TX1 connector on the HSMC card to the RX1 connector on the HSMC card. The camera can remain connected to the RX2 connector on the HSMC card but is not required for the diagnostic design.
2. Download the design from https://cloud.altera.com/devstore/. Click on "Design Examples" and enter "APIX2" in the search box. Click on "Loopback test for the APIX2 HSMC Card". Locate the Installation Package section. Click on the Download button. This will download the loopback_test_for_the_hsmc_card.par file to your computer.
3. For Quartus version 16.0 or later, double click on the par file. This will launch Quartus and unpack the project.
4. Power up the Cyclone V SoC development kit.
5. Program the FPGA. The configuration file inova_demo_loopback.sof has been prebuilt and is part of the Quartus project. In Quartus, from the File menu, open inova_demo_loopback.cdf. This will launch the Programmer in a separate window. In the Programmer window, make sure the Hardware Setup is USB-BlasterII. Click on the "Hardware Setup…" button if not and configure for USB-BlasterII. Click on inova_demo_loopback.sof and then Start to initiate the programming process. The FPGA should be programmed and the design should begin running. Close the Programmer window when the process is complete. Quartus can also be exited at this point.

Using the loopback diagnostic design

In the loopback diagnostic design the generated video stream is composed of 1280x800 active pixels at 60 frames per second. The received video stream is monitored for activity and errors which is indicated by the FPGA LEDs on the device kit as follows:

- LED 0: Blinks at about 1Hz when the transmitter is sending video frames.
- LED 1: Blinks at about 1 Hz when video frames are received, i.e., the incoming clock is stable at the expected frequency and there are transitions on the vertical sync signal. Several video frames are required for the receiver to synchronize to incoming video.
- LED 2: Illuminates when there are no errors detected by the receiver. The LED will turn off for 1 second if an error is detected.
- LED 3: Illuminates when there have been no errors detected since configuration. The LED will turn off and stay off if an error is ever detected.

Two pushbuttons (labelled Display control buttons in Figure 3) are used to inject and clear errors as follows:

- Button 0: Injects a single bit error in the transmitted stream when pressed. LED 2 and LED 3 should turn off. LED 2 should illuminate after 1 second.
- Button 1: Clears error indicators. LED 3 should illuminate when pressed.

The first 24-bit pixel in the transmitted stream is a count that increments every frame. This count is used as the initial seed to generate a sequence of pseudo random 24-bit numbers for the remaining pixels. Error checking logic monitors received video frames to ensure that the count is incrementing, that the pseudo random data is correct and that the total number of pixels per frame is correct.