

# High Definition, Low Bandwidth

Implementing a high-definition H.264 codec solution with a single Xilinx FPGA.



by Ronnie Smart  
Software Engineer  
Alpha Data Ltd.  
[rsjs@alpha-data.com](mailto:rsjs@alpha-data.com)

Rob Beattie, Ph.D.  
Design Engineer  
4i2i Communications Limited  
[rb@4i2i.com](mailto:rb@4i2i.com)

Ryan Dalzell, Ph.D.  
Design Engineer  
4i2i Communications Limited  
[rd@4i2i.com](mailto:rd@4i2i.com)

Andy Ray  
Design Engineer  
4i2i Communications Limited  
[ar@4i2i.com](mailto:ar@4i2i.com)

Alpha Data, in conjunction with our partners 4i2i Communications, has implemented one of the world's first high-definition H.264 codec solutions in a single FPGA platform on a single PCI board. This solution gives you the means to develop appli-

cations for high-resolution cameras in applications such as broadcasting, video conferencing, video surveillance, and aerospace and defense. The solution uses a Xilinx® Virtex™-4 FPGA board (Alpha Data part number ADM-XRC-4) with an adapter for communication with the camera.

## Camera Link Standard

Historically, the image processing and digital video market has suffered from a lack of interconnect standards for cameras and frame grabbers. Various manufacturers developed different connectors and protocols, each requiring different cables and interface logic. However, the Camera Link standard is becoming increasingly widespread, taking away much of the pain of connecting together digital video hardware. Its greatest strengths are:

- It is camera- and frame grabber-independent
- It has high data throughput, using at most two Mini D Ribbon (MDR) connectors

Therefore, we chose this standard when we developed our adapter, the XRM-CAMERALINK. The Automated Imaging Association (AIA) controls the Camera Link standard.

## Channel Link

The key component of the Camera Link standard is the Channel Link chip set, a parallel-to-serial transmitter and serial-to-parallel receiver developed by National Semiconductor. Channel Link transmits data bits using the latest high-speed LVDS (low-voltage differential signaling) technology.

A conventional RS-644 LVDS requires at least 56 conductors; a single Channel Link transmits signals using 11 conductors, reducing cable forms and shielding requirements.

The Channel Link transmitter achieves this high transmission rate by serializing – for each clock cycle – 28 bits into 4 LVDS data streams. A fifth LVDS stream transmits the clock signal. The Channel Link receiver de-serializes the data streams back into 28 bits.

## Configurations

The standard defines three configurations: base, medium, and full (Figure 1) using one, two, and three Channel Links, respectively. Base configuration is limited to transmitting 28 bits of video data per clock cycle, as it uses only one Channel Link over one MDR cable. You can realize higher transmission rates with the medium configuration (56 bits per clock cycle) and full configuration (84 bits per clock cycle), but they require two MDR cables.

The Camera Link standard states that 24 of the 28 bits transmitted by a Channel Link are image data bits. In base configuration, these bits are split across three 8-bit ports. The standard defines how the image data bits are assigned to the ports (the number and size of pixels transmitted on each clock cycle). Medium and full configurations have six and eight ports, respectively.

The other four bits transmitted by a Channel Link are pixel qualifier signals: frame valid (FVAL), line valid (LVAL), data valid (DVAL), and spare (SPARE). The last is reserved for future use.

## Camera Control

All of the configurations include four RS-644 LVDS pairs for camera control (CC1-CC4). The Camera Link standard does not describe the signals used by camera control; therefore, camera manufacturers define their own signals.

## Asynchronous Communication

Two RS-644 LVDS pairs (Tx and Rx) are used for asynchronous serial communication between the camera and frame grabber. Frame-grabber manufacturers supply an API implementation for serial communication mandated by the standard.

## XRM-CAMERALINK

We designed the XRM-CAMERALINK frame grabber to work with the Virtex-II, Virtex-II Pro, and Virtex-4 series of reconfigurable computing cards. In keeping with the Camera Link standard, it is camera-independent.

The XRM-CAMERALINK has two subsystems: Channel Link receiver and port-to-pixel mappings.

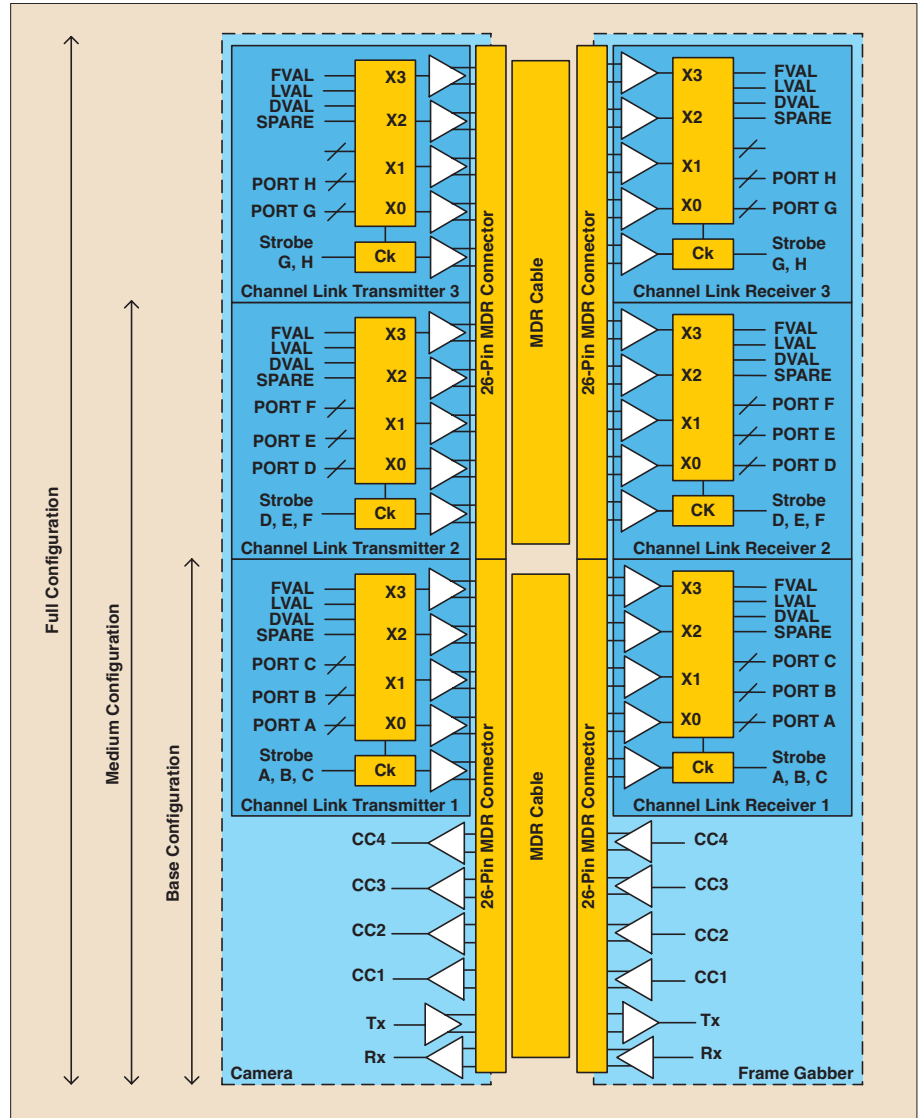


Figure 1 – Camera Link block diagram with the three Channel Links, camera control, and asynchronous serial communication

## Channel Link Receiver

The first subsystem of the frame grabber, the Channel Link receiver (Figure 2), deserializes the four incoming LVDS data streams back into the three 8-bit ports and four pixel qualifier signals.

The clock signal is phase-adjusted using the digital clock manager (DCM) so that the incoming LVDS signals are sampled correctly within a data window. The DCM multiplies the clock by 3.5 to produce the pixel clock.

It is not possible to multiply the clock rate by 7 (7 bits need to be extracted from each LVDS stream on each clock cycle), as the clock frequencies would become too great for FPGA implementation. Instead,

an independent clock signal is split in the I/O block and fed through a double-data-rate register (DDR), pipeline registers, and decoder to produce even and odd syncs.

Each LVDS bit is also fed through a DDR register and a set of pipeline registers to produce even and odd streams. The streams from all of the LVDS bits are fed through decoders to produce two data streams. The two streams are interleaved through a multiplexer (MUX) with even and odd syncs to form the image data bits and pixel qualifier signals.

## Port-to-Pixel Mappings

The second subsystem of the XRM-CAMERALINK translates the image data

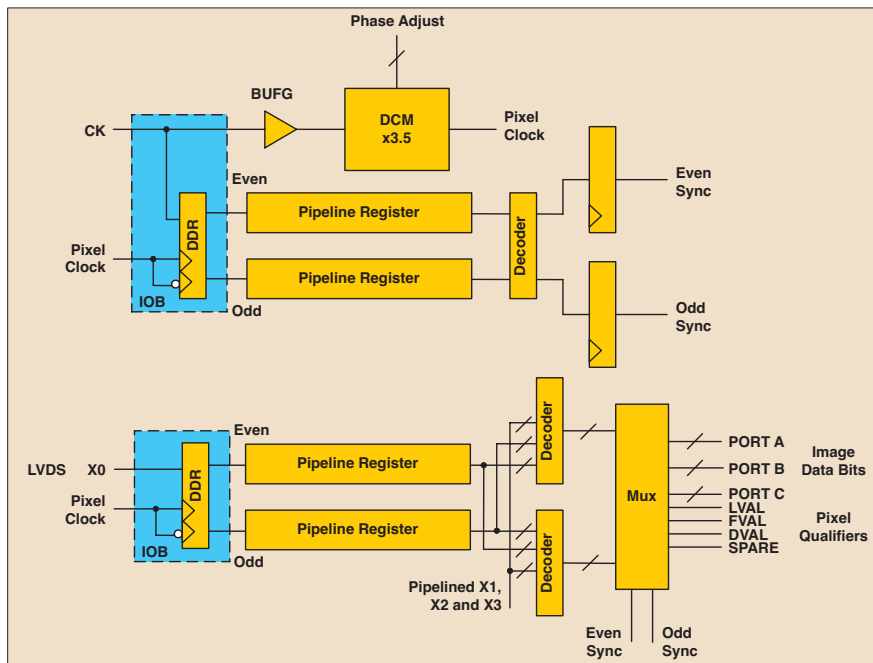


Figure 2 – XRM-CAMERALINK Channel Link receiver (base configuration) block diagram

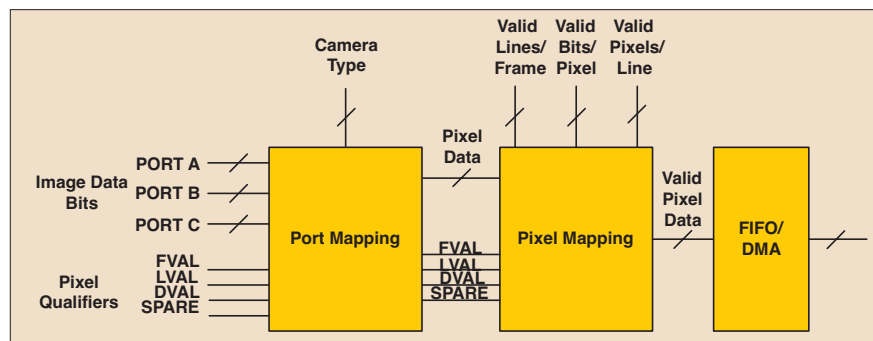


Figure 3 – XRM-CAMERALINK port-and-pixel mappings block diagram

bits and pixel qualifiers into valid pixel data (Figure 3).

The port mappings block allows you to specify (for a particular camera) the number and size of pixels transmitted by the ports for each clock cycle. The image data bits are translated into a stream of pixel data.

The timing references FVAL and LVAL indicate when frames and lines start, respectively. Invalid image bits are indicated by DVAL and are removed in the pixel mappings block.

An additional function of the pixel mappings block is to further clip the frame by defining a region of interest (ROI). Any pixel data outside the ROI is discarded. The ROI is determined by a set of FPGA

registers consisting of valid lines/frame, valid bits/pixel, and valid pixels/line. By altering the ROI on successive frames, for example, you can track an object as it moves across the camera's field of view. The output from the pixel mapping block is a 32-bit word that contains valid pixel data and frame and line-count indicators. Your FPGA design will sink this data.

### Camera Control and Serial Communication

Control of the camera, which is camera-specific, is achieved through the serial interface on the XRM-CAMERALINK.

Two FIFOs within the FPGA buffer data for serial communication with the

camera. On the host, this communication occurs using a dedicated thread, as it is independent of the frame-grabbing logic.

### H.264 Video Codec IP Core

Standard-definition (SD) 30 fps video captured using the XRM-CAMERALINK interface comprises raw video data being streamed across the interface. Cameras support a number of different formats; 4i2i Communications has interfaced cameras that provide RGB video at 8 bits per pixel and Bayer 8 bits per pixel. For the RGB video, this corresponds to a raw digital video data rate of about 248 Mbps.

Within the Virtex-4 device, this raw digital video stream is converted to YUV 4:2:0 video at a data rate of about 124 Mbps and is used as the input to the H.264 video encoder. By eliminating redundancy, this configuration reduces the bit rate to a much more manageable data rate of about 64K to 64 Mbps. The compressed video may then be efficiently stored on disk or transferred across a communication network such as the Internet.

We have also successfully used the camera link in connection with 720p high-definition (HD) cameras. Using slice-based encoding, it is possible to compress 720p 60 fps using multiple instantiations of 4i2i's H.264 IP core. About 24,000 slices are required to implement a three-slice 720p 60 fps H.264 encoder in a Virtex-4 device.

For the H.264 baseline codec IP core, 4i2i uses an architecture based on a dedicated macroblock processing pipeline. All of the video standards in common use today work on small 16 x 16 pixel blocks of video data known as macroblocks. (These are 16 x 16 luma samples, together with their corresponding chroma samples.) The algorithms generally require you to perform a sequence of operations in turn on each macroblock. Typically, these operations are prediction, transformation, quantization, and variable length encoding.

The 4i2i approach to codec implementation is to implement these operations as discrete components or processing modules, each of which process one macroblock at a time, separated by paged memory

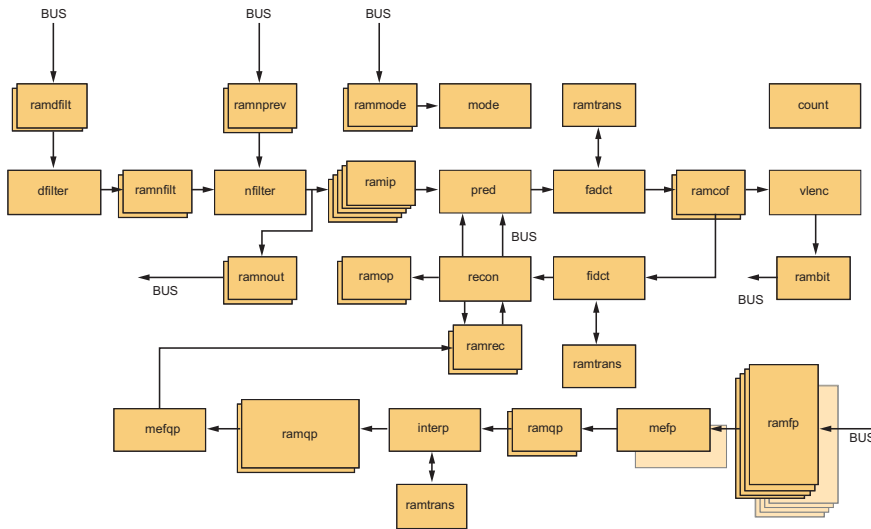


Figure 4 – H.264 encoder IP core architecture

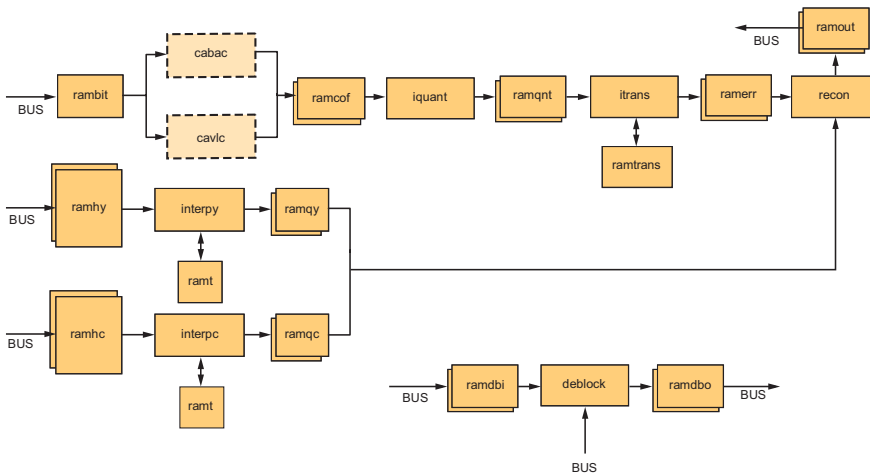


Figure 5 – H.264 decoder IP core architecture

buffers. By altering the number of pages, data scheduling may be affected. Xilinx FPGA devices are well suited to this, as they have an abundance of on-chip small block memories.

This approach has many advantages. First, it does not require any software intervention. Second, it allows all components to operate continuously at maximum throughput. Third, it results in a minimum latency implementation, because all processing operations to produce an encoded bitstream are performed on each macroblock in turn. Fourth, it allows several people to work on the design of the core independently. Finally,

combining several such cores for higher throughput is a straightforward process.

An example of using this technique to implement an H.264 encoder and decoder is shown in Figures 4 and 5, respectively. 4i2i has successfully used exactly the same methodology to develop a range of codecs (SDTV and HDTV) from the ITU, ISO, and SMPTE standards at HDTV, all suitable for FPGA implementation.

Using buffer memories in this manner also has several advantages when it comes to system design. The design is independent of the type of external memory used. Several cores can share the same memory and you can use the full memory bus speed.

The compressed video data is then transferred over the PCI interface to the host PC using DMA, where it may be written to disk or streamed over the Internet.

### Implementation Considerations

The complexity of modern compression algorithms makes it necessary to take care with the structures used in the RTL code to ensure an efficient use of FPGA resources. For example, writing the RTL to ensure that parametric information is stored in distributed RAM can bring considerable area savings. You can save even more resources by implementing concurrent access to more than one line of data from a macroblock buffer, using the multiple memory ports provided to on-chip RAM blocks rather than storing the data in separate register files.

### Conclusion

Camera Link is fast becoming the de facto standard for high-resolution digital frame grabbers, with its high transfer rates and substantial cable reduction. A new connector called the Mini CL is one-third smaller than MDR and further reduces cabling, resulting in smaller cameras.

The XRM-CAMERALINK module provides an all-in-one interface, allowing you to develop applications for any digital camera that conforms to the Camera Link standard.

Camera Link, in connection with a high-performance state-of-the-art video codec such as 4i2i Communications's H.264 IP core, enables huge raw video bandwidth to be reduced to a much more manageable data rate that you can then store on disk or stream across the Internet.

The ADM-XRC-4 features the Virtex-4 FPGA (SX55, LX100, or LX160). Including Camera Link, it supports a range of front-panel adapters for video I/O, such as CCIR, S-Video, and HD-SDI.

For more information about the FPGA development platform or the Camera Link interface, contact Alpha Data at [info@alpha-data.com](mailto:info@alpha-data.com) or [www.alpha-data.com](http://www.alpha-data.com). For more information about the H.264 IP core, contact 4i2i Communications at [contact@4i2i.com](mailto:contact@4i2i.com) or [www.4i2i.com](http://www.4i2i.com).