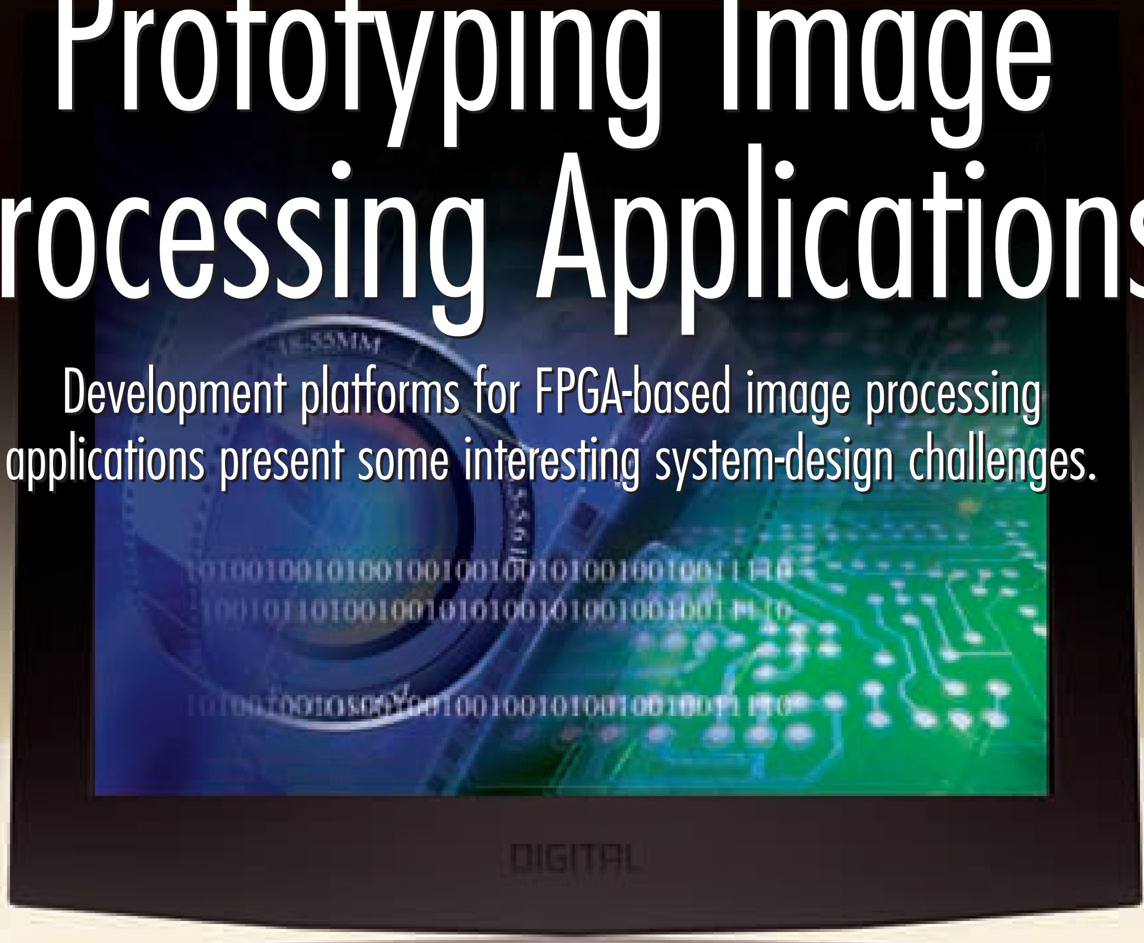
Prototyping Image Processing Applications



Development platforms for FPGA-based image processing applications present some interesting system-design challenges.

by Mike Hodson
Managing Director
Image Processing Techniques Ltd.
mike.hodson@imageproc.com

FPGA devices have been used for many years in the design of professional video systems. The consumer electronics marketplace, however, has traditionally been an area where FPGAs have not made much of an impact, primarily because of device cost.

Recently, goods such as high-resolution flat-panel displays, MPEG-2/MPEG-4 encoders and decoders, portable video players, and digital cameras have grown in popularity. Such items require significant image processing functionality. A few years ago, custom silicon would have been the only sensible choice, but with the introduction of next-generation devices such as the Xilinx® Spartan™-3A DSP and Virtex™-5 families, the price/performance balance of FPGAs versus ASIC or ASSP devices has shifted, leading to significant growth in the use of FPGAs for such applications.

Using FPGAs for image processing applications presents system designers with some interesting problems. The analysis and manipulation of video images is inherently a high-bandwidth process, while software simulations do not always provide engineers with enough information to establish the performance of their algorithms because they are not in real time. For this reason, there is a growing demand for FPGA hardware prototyping systems targeted specifically at the requirements of image processing engineers – development platforms that provide the real-time I/O and memory interface capabilities necessary to prototype today's most demanding signal processing applications, as illustrated in Figure 1.

In this article, I'll look at the requirements and availability of FPGA development platforms for image processing applications.

Required Elements

When considering the design of an FPGA platform for image processing algorithm development, it is useful to review the principal elements that are common to the most popular image analysis techniques, and what hardware resources are needed to support these elements.

High-Speed I/O

Getting real-time video signals into and out of an image processor is obviously critical to the performance of the system. Typically, image data is input and output synchronously through a proprietary or industry-standard interface such as DVI, HDMI, SDI, analog component, S-video, or composite. The images may alternatively be transferred asynchronously through a processor or backplane bus (PCI, VME) directly into frame stores.

The latest generation of serial bus interfaces, such as PCI Express, have sufficient

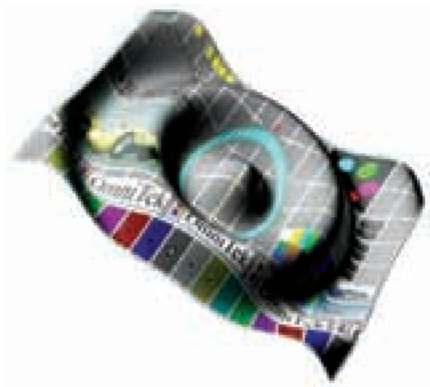


Figure 1 – Real-time 3D image manipulation using FPGA technology

bandwidth for multiple streams of uncompressed high-definition video to be transferred in real-time, while Virtex-5 devices, with their on-chip PCI Express interface and multiple MGTs, provide an excellent range of I/O capabilities for video interfacing.

Frame Stores

The frame stores used for temporary storage of image data are at the core of many image processing functions. They serve many purposes, such as synchronizing multiple image sources, image re-sizing and manipulation, motion or object detection and tracking, noise reduction, and de-interlacing.

The storage of multiple video frames typically requires a large quantity of memory that is accessed sequentially. For this reason, single- or double-data-rate SDRAM is normally employed. However, in applications where image transformations include perspective, rotation, or non-linear "warps" of the source image, the frame store architecture is typically based on very high speed synchronous static RAMs to give low-latency random access operation (rather than linear data bursts from SDRAM).

Filtering and Interpolation

Applications that involve re-sampling of the source image data also need to interpolate or filter the image to avoid aliasing problems. Efficient implementation of filters is therefore important. Spartan-3A DSP and Virtex-5 devices are useful here, as the architecture of their DSP blocks is ideally suited to the efficient implementation of all kinds of filters.

Delay Elements

Image processing algorithms typically require a wide range of delay elements. For delays of a few pixels – to equalize pipeline processing latency or for polyphase filter taps – the SRL16 blocks in Xilinx FPGAs are highly efficient. Block RAMs, with their dual-port architecture, are ideal for line delays, which are needed where the vertical columns of data within 2D raster-scan images are processed.

Look-Up Tables

Adjustments to the dynamic range of image data and non-linear transfer functions can be efficiently implemented using look-up tables. How such tables are implemented – and how they are referred to – depends on the number of inputs they have. Look-up tables (LUTs) are said to be "1D" where the table output value depends on only one input value, "2D" where the output depends on two inputs, and "3D" where it depends on three inputs.

The 1D and 2D cases can usually be implemented in block RAM (depending on the size of the input vectors). However, to obtain sufficient precision in 3D LUTs, it is often necessary to use external memories such as high-speed synchronous static RAMs. DRAMs are not usually suitable because LUT operations are rarely burst-oriented. Instead, look-up address values tend to be random in nature, which makes DRAM usage inefficient.

Color-Space Conversion

A very common image processing operation is the conversion of image data between different color spaces. RGB, YCbCr, XYZ, and xvYCC are all in widespread use at the moment, so there is frequently a need to convert between these different formats. Conversion typically uses a 3 x 3 matrix multiplier array. Such arrays can be implemented very easily using Xilinx DSP blocks.

Video Mixing and Keying

Many image processing applications require multiple image sources to be mixed or cross-faded, or one image to be overlaid on top of another using a "key" or alpha channel as a guide. It is quite possible to perform these operations using a simple switch between sources; however, this does not allow for variable transparency and may also result in out-of-band components in the output image. For this reason, professional mixers and keyers use proper cross-fader logic built from multiplier pairs and key signals with "soft" low-pass filtered edges. Ideally, FPGA-based systems should implement similar logic.

Software Control and Analysis

Virtually all image processing systems require some form of software-programmed control system, either to configure the system correctly or to analyze the data and present the results in an efficient manner. These control systems may range from a simple soft- or hard-IP core processor embedded in the FPGA (such as the PowerPC embedded in Virtex FPGAs) all the way up to the latest Intel Core 2 Quad systems running Microsoft Windows Vista, which have tremendous processing capability. With such high-performance processors, system designers now have the choice to split the image processing tasks between hardware and software; hence the importance of implementing a high-bandwidth data pipe between the two.

Resulting Platform Architecture

The ideal platform for image processing algorithm development will provide most (if not all) of these processing elements.

Although many of the required elements are already supported by the Virtex-5 family of FPGAs, some items (such as large frame stores and dedicated video I/O) require external components. Hosting the whole system in a PC-based environment is also an ideal way to integrate Xilinx ISE™ software with any proprietary user application.

Figure 2 shows the block diagram of a new FPGA prototyping system that has been developed by Image Processing Techniques (IPT) in conjunction with Xilinx. Called the Advanced Video

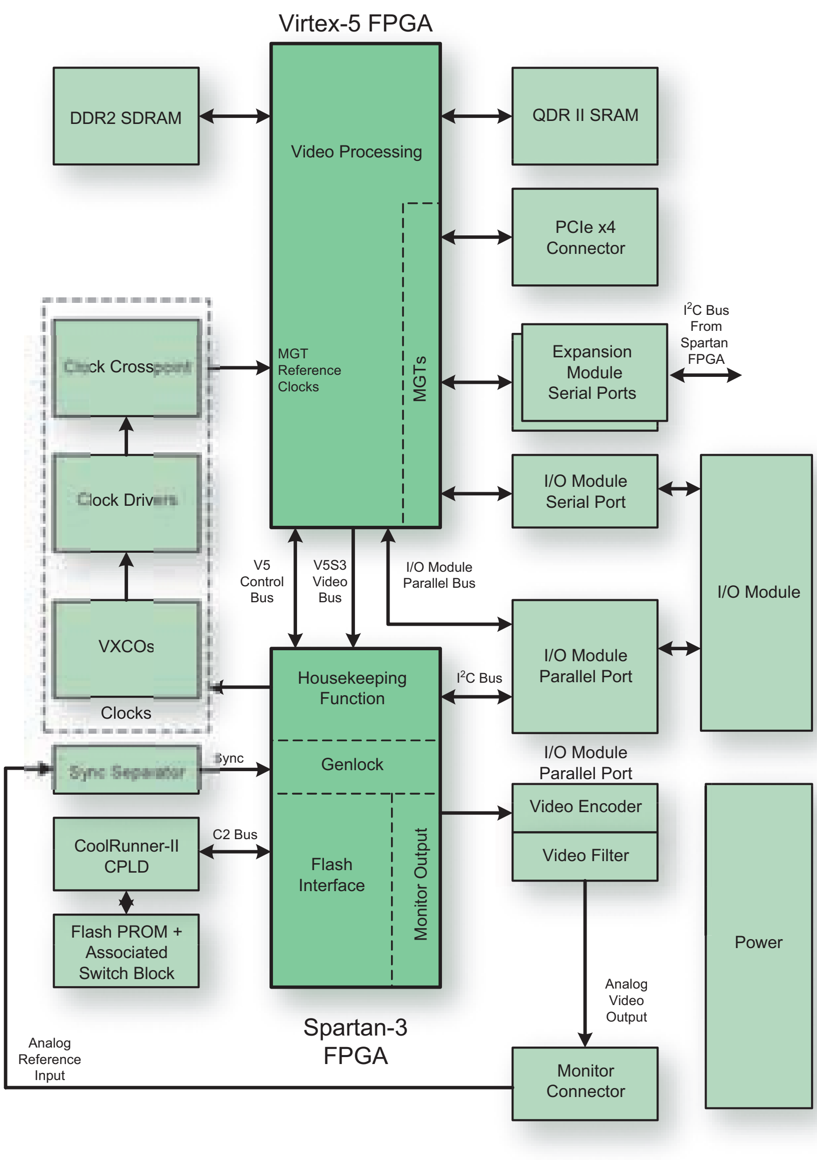


Figure 2 – AVDP block diagram

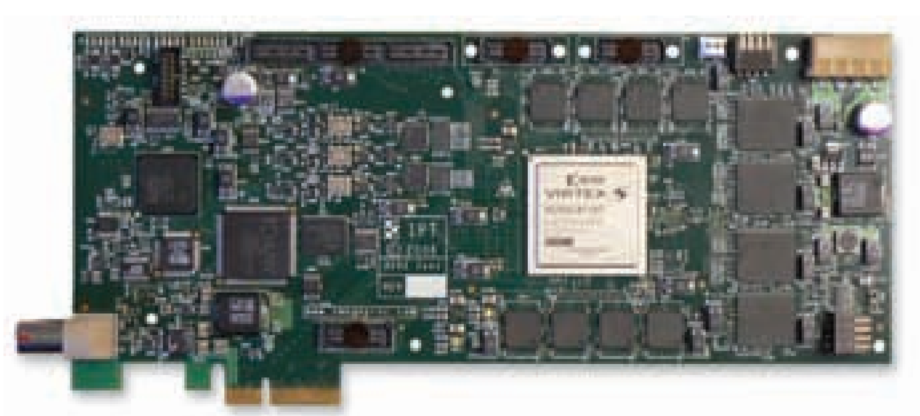


Figure 3 – The Advanced Video Development Platform

Development Platform (AVDP), the system has been specifically designed to assist engineers with the development of a wide range of image processing algorithms.

The top of the AVDP is shown in Figure 3. It is designed around a Xilinx Virtex-5 FPGA, leveraging its advantages of multiple MGTs for high-speed I/O and an architecture that is well suited to the implementation of filters and color-space conversion arrays. The board also features a PCI Express x4 connector, providing an efficient interface to a PC for software control and analysis; up to 1 GB of high-speed DDR2 SDRAM for bulk video storage; and up to 4 x 36 MB of high-speed QDR II SRAM.

The AVDP system is available with a range of Virtex-5 devices installed, from the LX50T all the way up to the DSP-intensive SX95T, and is provided together with a range of specialist video IP in VHDL source code, software API, device drivers, and a debugging application to help designers develop their image processing applications.

Conclusion

The AVDP board, developed by IPT in conjunction with Xilinx, represents the state-of-the-art in FPGA development systems and addresses the requirements of platforms for image processing algorithm development: flexible video I/O, high-speed memory, and high-performance data acquisition and control.

The key applications that the system is designed to accelerate include:

- Codec design and development: MPEG-2, MPEG-4, JPEG-2000
- Video standards conversion
- Video de-interlacing and resizing algorithms for flat-panel displays
- Broadcast graphics: character generators, still-stores, keyers, and mixers
- Real-time image manipulation for video special effects
- Robotic vision processing and algorithm development

For more information, please visit *www.imageproc.com.*