FPGAs — Enabling DSP in Real-Time Video Processing

With Xilinx FPGAs you can create DSPs that are much faster than any off-the-shelf stand-alone DSP device.

by David Nicklin, Senior Manager
Strategic Solutions, Wireless
Xilinx, Inc.
dave.nicklin@xilinx.com

Real-time video processing makes enormous demands on system-level performance requirements and, except for the simplest of functions, is well beyond the capabilities of any general purpose DSP device. Therefore, DSP-based video designs often require several DSP devices to get the necessary throughput, along with the overhead of multiple program and data memory resources. However, designing with programmable logic allows you to implement video signal processing algorithms using parallel processing techniques, giving you the performance you need within a single, highly flexible programmable logic device. By performing video processing functions in real time, you can minimize the need for frame stores and data buffering.

With the industry pushing for higher levels of video quality, and the development of improved compression schemes, system processing rates have increased dramatically. At the same time, we have introduced new programmable logic devices, such as the Xilinx Spartan™-IIE FPGAs, which draw on the architectural heritage of the FPGA devices that are commonly used in professional broadcast equipment today. As FPGA process development follows Moore's Law, our new products can perform the same functionality as their predecessors, with even higher performance and a much lower cost.

By developing systems using our latest cost-effective FPGAs, you can bring unprecedented levels of professional, broadcast quality, video processing into areas of digital video technology such as high-end consumer products, security systems, industrial and machine vision applications, and frame-grabbers.

One driver of this trend, is the combination of networking, broadcast, processing and display technology, in what the industry has termed "digital convergence." The need to send high bandwidth video data over extremely difficult transmission channels, such as wireless, while still maintaining an acceptable quality-of-service (QoS), is extremely difficult. This has lead to wide ranging research in how to improve error correction, compression and image processing technology, much of it based on advanced FPGA technology.

Image Compression/Decompression – DCT/IDCT

The main video compression scheme used in digital video systems today is MPEG2. It can be found at the heart of digital television, set-top boxes, digital satellite systems, high-definition television (HDTV) decoders, DVD players, video conferencing equipment, and Web phones, to name just a few. Raw digital video information invariably has to be compressed so it can be either sent over a suitable transmission channel, or stored onto a suitable medium such as a disc.

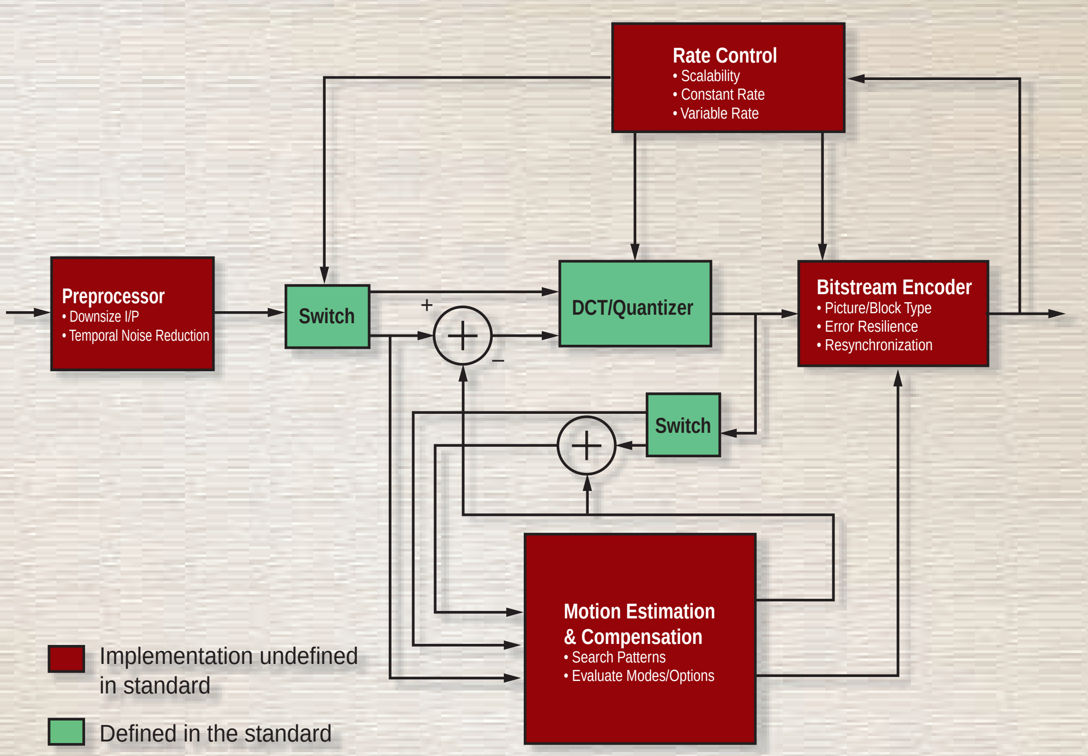

Figure 1 - Simplified video encoder diagram

There are also a number of emerging standards, including most notably MPEG4. Most products based around this technology are still in development, although ramp-up to production is expected shortly. At the heart of the MPEG2 and MPEG4 algorithms is a function called the discrete cosine transform (DCT). The aim of the DCT is to take a square of pixel blocks and remove redundant information that is imperceptible to the viewer. To decompress the data, the inverse discrete cosine transform (IDCT) function is required.

While the DCT section of the MPEG algorithms is standardized and can be implemented very efficiently within FPGAs, MPEG encoding has a lot of blocks that are left undefined. It is within these undefined sections of the standard that a company can truly differentiate its product from its competition and develop its own proprietary algorithms. Many professional MPEG encoders use FPGAs in these sections, such as the motion estimation block, as shown in Figure 1. Because FPGAs are reconfigurable, the equipment can be easily updated to incorporate new algorithms at all stages of development, including in the field after deployment. Companies that rely totally on standard ASSP solutions are limited in their ability to produce products that can make them stand out from the competition, and they therefore run the risk of being seen as just one of a number of similarly specified solutions in the market.

Color Space Conversion

Another important part of a video system is the requirement for color space conversion, a process that defines how an image specified in one color format can be converted into a different color format.

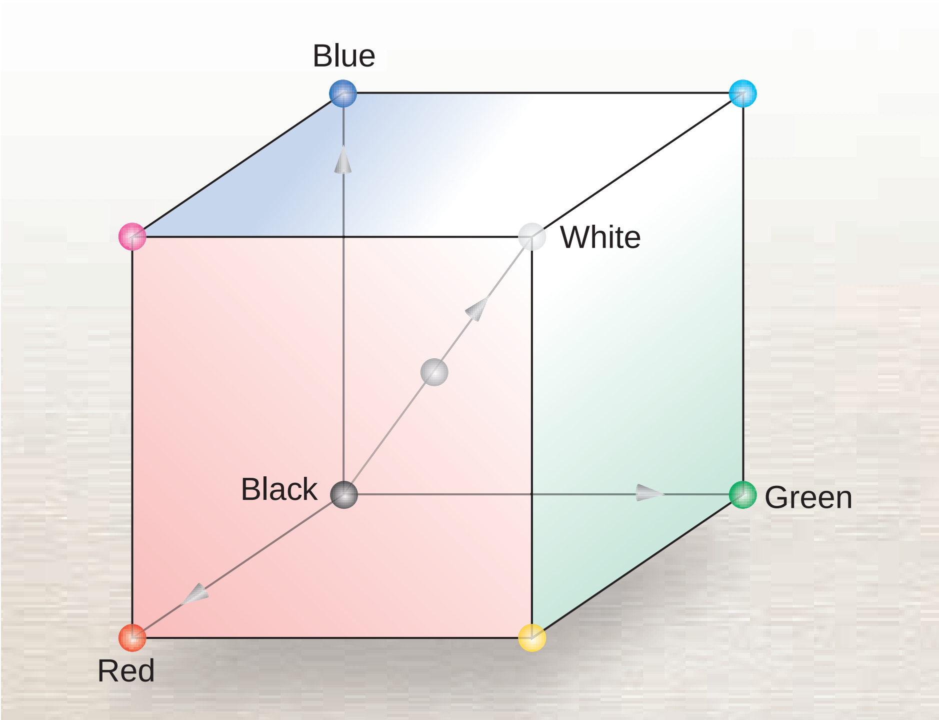

Figure 2 - Color space cube

Receptors in the human eye are only capable of detecting light wavelengths from 400 nm to 700 nm. These receptors are called cones and there are three different types, one for red light, one for green, and one for blue. If a single wavelength of light is observed, the relative responses of these three sensors allow us to discern what we call the color of the light. This phenomenon is extremely useful because it means we can generate a range of colors by simply adding together various proportions of light from just three different wavelengths. The process is known as additive color matching, and is used in color television systems.

It's possible to represent colors of light by plotting the red, green and blue (RGB) components proportions on a 3-dimensional cube, with black at the origin and while at the diagonally opposite corner. The resulting cube is known as the "RGB color space," as shown in Figure 2.

Whether the final display medium is paper, LED, CRT, or plasma displays, the image is always broken down into an array of picture elements or pixels (HDTV, for example, can have 1920 x 1080 pixels). While the mechanics change slightly for each medium, the basic concept is that each pixel displays a proportion of red, green, or blue depending on the voltage signals driven to the display.

Processing an image in RGB format, where each pixel is defined by three 8-bit or 10-bit words corresponding to each primary color, is certainly not the most efficient method. With such a format, every action on a pixel has to be performed on all the red, green, and blue channels. This invariably requires more storage and data bandwidth than other alternative color formats.

To address such issues many broadcast standards, such as the European PAL and North American NTSC television systems, use luminance and color difference video signals. A requirement therefore exists for a mechanism to convert between the different formats, and this is called "color space conversion."

Realizing these circuits in hardware is a relatively simple task, once the coefficients to map from one plane to another are known. One common format conversion is RGB to YCbCr (and conversely YCbCr to RGB). See Figure 3. It has been found that between 60% and 70% of the luminance information (Y) a human eye can detect comes from the green color. The red and blue channels in effect duplicate much of the luminance information, and hence, this duplicate information can be safely removed. The end result is that the image can be represented as signals representing chrominance and luminance. In this format, the luminance is defined as having a range of 16 to 235 in an 8-bit system, and the Cb and Cr signals have a range of 16 to 240, with 128 being equal to 0.

A color in the YCrCb space is converted to the RGB color space using the equations:

$$R' = 1.164\,(Y - 16) + 1.596\,(Cr - 128)$$

$$G' = 1.164\,(Y - 16) - 0.813\,(Cr - 128) - 0.392\,(Cb - 128)$$

$$B' = 1.164\,(Y - 16) + 1.596\,(Cr - 128)$$

R'G'B' are gamma corrected RGB values. A CRT display has a non-linear relationship between signal amplitude and output intensity. By gamma correcting signals before the display, the relationship between received signal amplitude and output intensity can be made linear. The output gain is also limited below certain thresholds to reduce transmission-induced noise in the darker parts of an image.

There are a number of possible implementations – memory, logic, or embedded multipliers – to perform the necessary multiplication functions. It is certainly possible to meet and exceed the 74.25 MHz data rate required for HDTV systems. It is also possible to try different design trade-offs, such as that between system accuracy and design area. For example, for a 3% error in luminance, the design size of a YCrCb-to-RGB color-space-converter can be more than halved. This may be unacceptable in most display products but could be acceptable in other applications such as machine vision or security systems. By using an FPGA, you can tailor the algorithm for the application, thereby maximizing performance, efficiency, or both.

Real-Time Image and Video Processing Functions

Limitations in the performance obtainable with standard DSPs has led to the development of specially designed chips, such as media processors. However, these devices have often proved to be too inflexible in all but a narrow range of applications, and can suffer from performance bottlenecks. The limitations of a processor-based approach become especially apparent in high-resolution systems, such as HDTV and medical imaging. Fundamentally, a processor solution is restricted in how many cycles can be allocated to each tap of a filter, or each stage of a transform. Once the performance limits have been reached there is often no other way around the problem than to add extra DSP parts.

An FPGA, however, can be custom tailored to provide the maximum efficiency of utilization and performance. It's possible for you to trade off area against speed, and invariably perform a given function at a much lower clock rate than a DSP would require.

For example, Visicom Inc. found that for a median filter application a DSP processor would require 67 cycles to perform the algorithm. An FPGA needed to run at only 25 MHz, because it could realize the function in parallel. For the DSP to match the same performance, it would have to run at over 1.5 GHz. In this particular application, the FPGA solution is some 17 times more powerful than a 100 MHz DSP processor.

There are a wide number of real-time image and video processing functions that are well fitted for implementation in FPGA devices – these include real-time functions such as:

- Image rotation
- Scaling
- Color and hue correction
- Shadow enhancement
- Edge detection
- Histogram functions
- Sharpening
- Median filters
- Blob analysis.

Many of these functions are both application specific and system specific, and are based around core structures such as 2D-FIR filters. Such functions can be implemented quickly using HDL language design or by exploiting the DSP building blocks found in high-level core design tools such as the Xilinx CORE Generator™ software. It's also possible to reduce both design and simulation time by employing a system-level design approach, using products such as The MathWorks' MATLAB™ and Simulink™ software, as well as Xilinx System Generator.

Conclusion

With digital convergence, you need to integrate various standards and requirements into one homogeneous product – this demands flexibility in design and implementation.

FPGA technology is addressing the system requirements of new and emerging video applications, bringing with it the features and signal processing performance that have made it the preferred solution for video and image processing in the professional broadcast equipment market for a number of years. The latest generation of FPGAs can provide the same level of performance and functionality, at a fraction of the cost of the FPGAs that were designed into professional equipment just a few years before.

In comparison to ASSP and chipset solutions, FPGAs offer levels of flexibility that designers demand in today's products, while at the same time maintaining a distinct performance advantage over conventional DSPs.

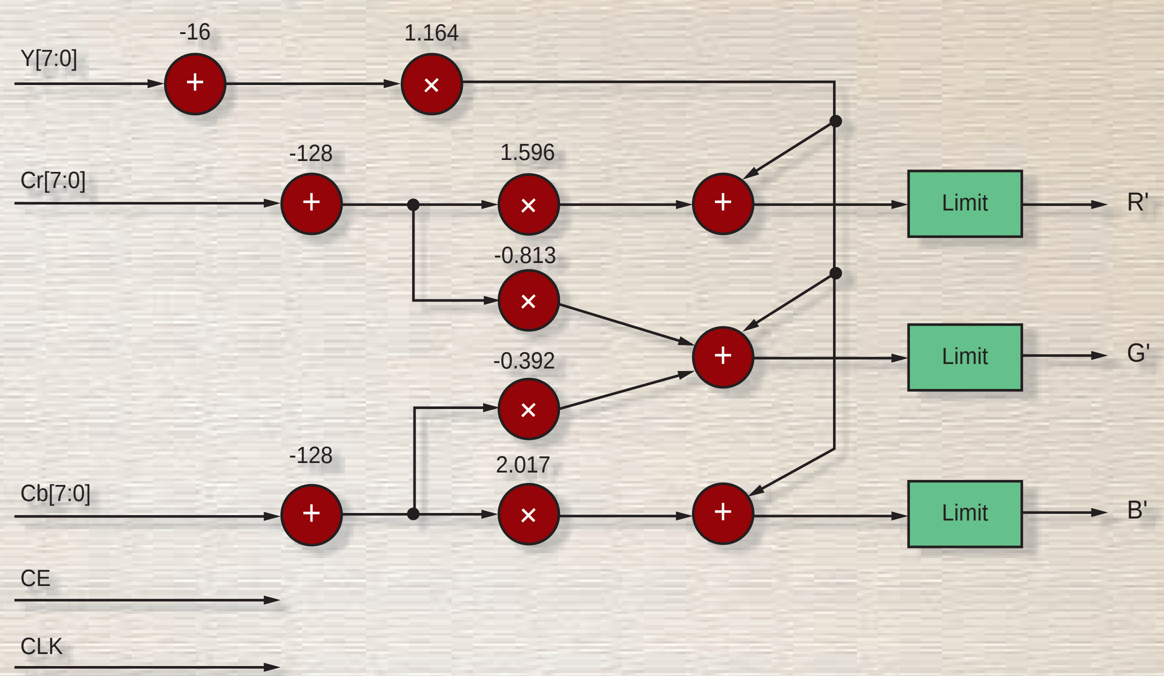

Figure 3 - YCrCb-to-RGB color space converter