FPGA-Based MPEG-4 Codec

Using FPGAs to implement complex video codecs goes beyond ASIC prototyping.

by Paul Schumacher
Senior Staff Research Engineer
Xilinx, Inc.
paul.schumacher@xilinx.com

Wilson Chung
Senior Staff Video and Imaging Engineer
Xilinx, Inc.
wilson.chung@xilinx.com

Have you ever wanted to include state-of-the-art video compression in your FPGA design but found it too complex an undertaking? You no longer need to be a video expert to include video compression in your system. Newly released MPEG-4 encoder/decoder cores from Xilinx can help solve your video compression needs.

Video and multimedia systems are becoming increasingly complex, and the availability of low-cost, reliable IP cores for your system is crucial to getting your product to market. In particular, video compression algorithms and standards have become extremely complicated circuits that can take a long time to design and are quite often bottlenecks in getting a system tested and shipped. These MPEG-4 simple profile encoder/decoder cores may just do the trick for your next multimedia system.

Applications

MPEG-4 Part 2 is a recent international video coding standard in a series of such standards: H.261, MPEG-1, MPEG-2, and H.263. It was approved by ISO/IEC as International Standard 14 496-2 (MPEG-4 Part 2) in December 1999. The MPEG-4 Part 2 video codec provides an excellent basis for a number of multimedia applications. The standard provides a set of profiles and levels to allow for a plethora of different application requirements, such as frame size and use of error-resilience tools. Examples of these applications include broadcasting, video editing, teleconferencing, security/surveillance, and consumer electronics applications.

The video coding algorithm used in MPEG-4 Part 2 is an evolution from previous coding standards. The frame data is divided into 16 x 16 macroblocks containing six 8 x 8 blocks for YCbCr 4:2:0 formatted data. Motion estimation with half-pixel resolution is used to efficiently code predicted blocks from the previous frame, while the discrete cosine transform (DCT) provides the residual processing to create a more detailed view of the current frame. Simple profile provides 12 bits of resolution for DCT coefficients with 8 bits per sample for the sampled and reconstructed frame data. Coding efficiency of the MPEG-4 simple profile is better than the previous generation in MPEG-2 across a range of coding bit rates.

A typical multimedia system can use MPEG-4 as the video compression component within a larger system. An example of this is an end-to-end video conferencing system delivering compressed bitstreams between two or more participants. Designations for these sources can modify system requirements, where a key speaker or presenter for a conference may require higher resolution video as well as audio. This type of system can be expanded to video surveillance and security applications, where a display station user may decide to keep a mosaic of all video cameras or focus in on a single camera view for detailed real-time analysis. These applications require that the stream selection is performed at the receiver and is capable of handling real-time viewing specifications.

An FPGA provides an excellent programmable concurrent processing platform that allows for support of varying system requirements while meeting the needs of system throughput. The Xilinx® MPEG-4 decoder core can be built with a scalable, multi-stream interface customized for your application and system requirements, while both the MPEG-4 encoder and decoder are also capable of servicing a user-specified maximum frame size.

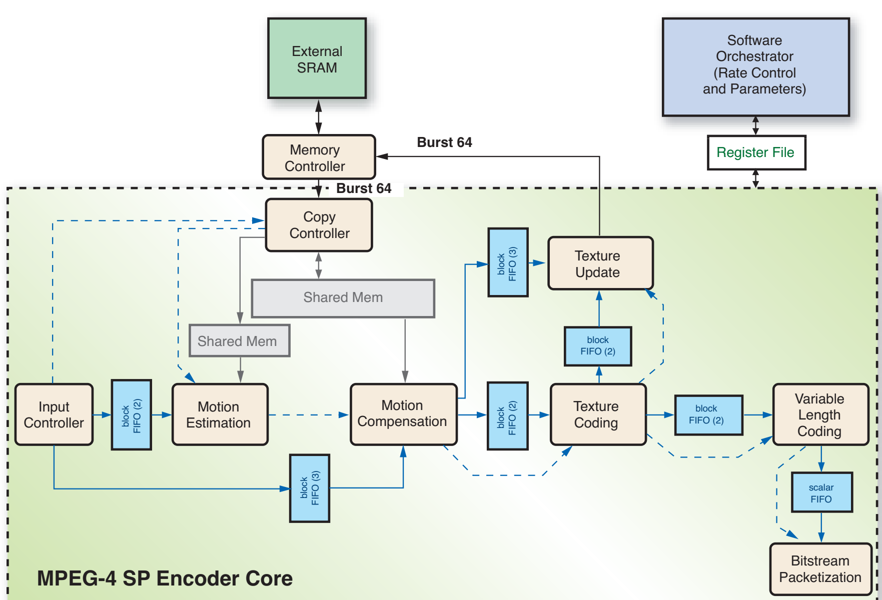

Figure 1 – Block diagram of MPEG-4 Part 2 simple profile encoder core

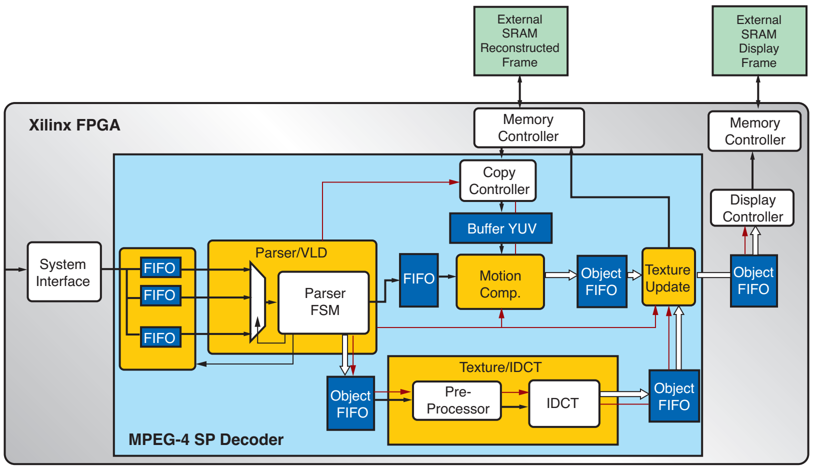

Figure 2 – Block diagram of MPEG-4 Part 2 simple profile decoder core

Architecture

Figures 1 and 2 illustrate the block diagrams for the MPEG-4 simple profile encoder and decoder cores, respectively. Hardware-based, pipelined architectures were used for these implementations, with a host interface provided on the encoder for software-controlled rate control. With an included memory controller, the raw, captured sequence for the encoder and the reconstructed frames for the decoder are stored in an off-chip memory for fast, low-latency access to the pixel data. A simple FIFO interface is provided for communicating the compressed bitstreams, with the decoder custom-built for a user-specified number of bitstreams. A system interface is also included to allow for maximum controllability and observability.

To create scalable multi-stream designs that can meet the needs of different applications, the package provided with the core contains a number of user-specified, compile-time parameters that allow you to customize the encoder and decoder. To create a resource-efficient design, you can also set the maximum supported frame width and height. The compiled design would then include enough memory and registers to support any frame dimensions less than or equal to these two parameters. Other parameters give you complete control over the scalability of the final design and craft a system built exclusively for your application.

Tables 1 and 2 list the FPGA resources for the encoder and decoder cores based on different parameter settings for maximum supported frame size, as well as the number of input bitstreams for the decoder. All of the encoder designs in Table 1 utilize 16 embedded XtremeDSP™ slices, while the decoders in Table 2 utilize 32 embedded XtremeDSP slices. These designs target Virtex™-4 parts, which contain a number of 18 Kb block SelectRAM™ memories as well as embedded XtremeDSP slices. Other compatible FPGA families include Virtex-II, Virtex-II Pro, and Spartan™-3 devices.

Note that the decoder design can automatically instantiate the number of input FIFOs and supporting multiplexing/demultiplexing circuitry based on the number of bitstreams to support. The MPEG-4 encoder is capable of a throughput of approximately 48,000 macroblocks per second, providing enough horsepower to exceed the throughput specifications of simple profile at level 5. Meanwhile, the MPEG-4 decoder design can sustain a throughput of approximately 168,000 macroblocks per second, providing adequate throughput to decode two streams of progressive SDTV (720 x 480 at 60 fps) or 14 streams of CIF resolution. This decoder throughput is more than four times the required throughput for simple profile at level 5.

Conclusion

MPEG-4 simple profile encoder and decoder cores have been designed with unique, scalable, multi-stream capabilities to suit your specific system needs. A number of different applications can take advantage of these cores in a multimedia system, including video conferencing, security, and surveillance, as well as any exciting new consumer application that you have yet to show the world.

High-throughput, pipelined architectures were used for these video designs with enough customizable parameters to create a resource-efficient design exclusive to your application. For more information, visit *www.xilinx.com/dsp*.

The authors would like to acknowledge contributions from Robert Turney, Nick Fedele, Adrian Chirila-Rus, Mark Paluszkiewicz, and Kees Vissers at Xilinx, as well as members at IMEC.

Parameters \ Resources Frame Size	Block RAMs	FPGA Slices	Minimum Clock Rate (MHz)
QCIF @ 15 fps	16	8,051	3.2
CIF @ 30 fps	21	8,309	25.6
4CIF @ 30 fps	30	9,000	100.7

Table 1 – Scalable MPEG-4 Part 2 simple profile encoder core resources

Parameters \ Resources: Frame Size	Streams	Block RAMs	FPGA Slices	Minimum Clock Rate (MHz)
QCIF @ 15 fps	1	10	4,332	0.8
	8	17	5,014	6.6
CIF @ 30 fps	1	16	4,558	6.6
	8	23	5,305	52.8
4CIF @ 30 fps	1	26	5,004	26.4
	8	33	5,764	211.2 *

* Note: Eight streams of 4CIF resolution currently require two instantiations of the decoders.

Table 2 – Scalable, multi-stream MPEG-4 Part 2 simple profile decoder core resources