

# HD-video encoding with DSP and FPGA partitioning

IMPLEMENTING A SCALABLE VIDEO-ENCODING ARCHITECTURE THAT INCLUDES DSPs AND, WHEN NECESSARY, FPGAs AS COPROCESSORS TO OFFLOAD CERTAIN TASKS SATISFIES EVEN THE MOST DEMANDING VIDEO APPLICATIONS.

As video and imaging applications evolve toward high-definition compression standards, coprocessing architectures that include both DSPs and FPGAs are becoming popular. However, using partitioned systems is not the only option, because DSP architectures now include enhancements in performance, peripheral mix, video-hardware acceleration, and implementation techniques, and these advances have significantly broadened the range of applications in which DSPs can provide a complete approach.

DSPs have an inherent advantage because they are programmable, and their versatility allows designers to execute almost any algorithm. But when the computational load exponentially grows, as is the case with HD (high-definition) video, you can sometimes employ FPGAs to hard-wire certain computationally intensive tasks, thereby offloading the DSP. In video encoding, as in virtually all other engineering designs, no one-size-fits-all approaches exist. Even when you are employing a consistent codec, the end application plays

a critical role in determining what level of computing power and memory bandwidth you require. These requirements, in turn, can play a dominant role in both hardware- and software-implementation strategies.

When dealing with compressed video, a standard compression algorithm is the most likely choice for experienced design teams. Once you select a codec, however, the next critical step is to assess the requirements of ME (motion estimation) and MC (motion compensation), because they can be two of the most demanding video-compression functions. Not surprisingly, the computational and memory bandwidth that the ME and MC engines demand depends on the amount of motion in the scene.

The H.264 AVC (advanced-video-coding) codec, for example, can find use in applications such as video surveillance, in which little action occurs over many hours of surveillance. At the other end of the spectrum, encoding HD video for a broadcast application can require a memory bandwidth of 20 Gbytes/sec or higher. HD videoconferencing, which might

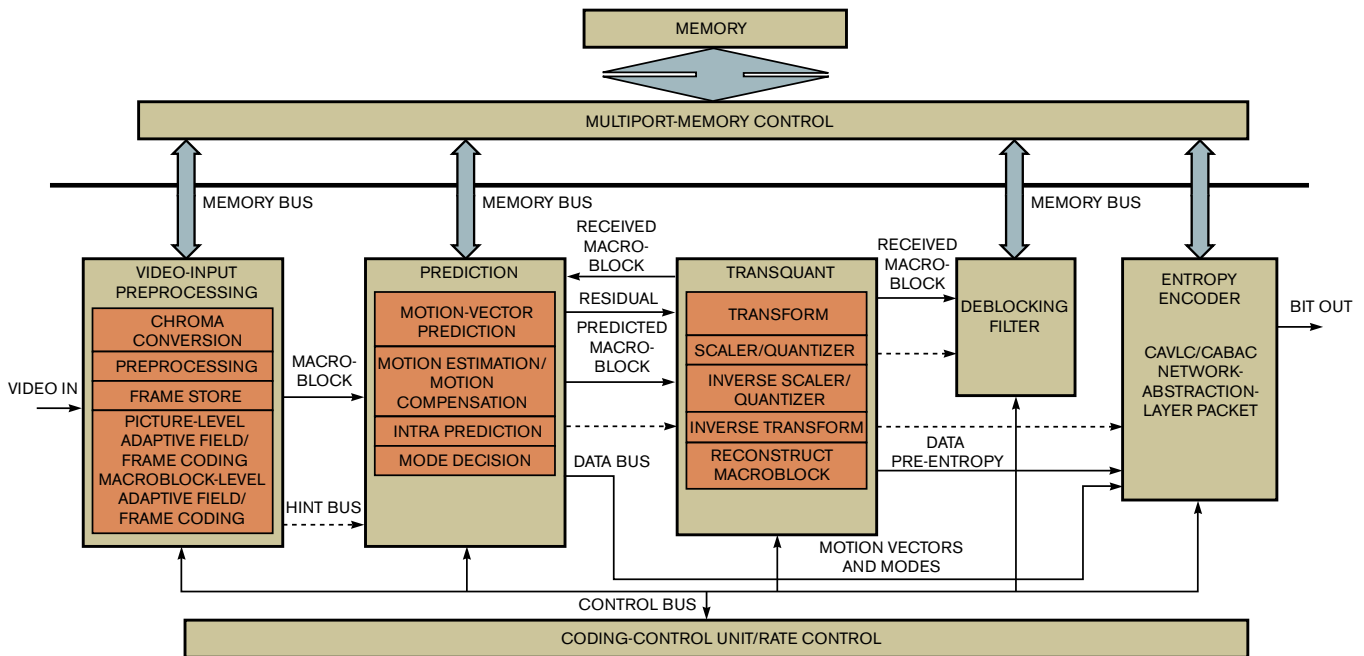


Figure 1 The encoder's motion-prediction block is critical to hardware partitioning.

require a memory bandwidth of 1.5 Gbytes/sec, lies between those extremes.

## ENCODING HIGH-DEFINITION VIDEO

Although codec profiles go a long way toward creating a “packaged” approach for design engineers, the application has an impact equal to or higher than the codec on the implementation’s hardware architecture. You can expect an HD-teleconferencing application, for example, to have relatively little frame-to-frame motion, but a broadcast-TV application must deal with the more intense video of sporting events, action movies, and other content in which you should expect a substantial amount of motion.

The ME and MC engines are key elements in a hardware-partitioning strategy, particularly for encoding video. The design team must consider whether they should implement only the ME engine on the FPGA or whether the computational load is heavy enough to require hardware acceleration of both the ME and the MC engines. The required memory bandwidth, which can be 20 Gbytes/sec or more, is just as important as the computational loading. FPGA-hardware architects may have flexibilities to scale the memory bandwidth as high as necessary and higher than the bandwidth that a DSP alone supports.

The H.264/AVC high profile is the obvious architecture for HD encoding of broadcast transmissions (Figure 1). For ME calculations, the current frame and each of the frames to which it will refer subdivide into macroblocks, which are typically 16×16 pixels in size but can be as small as 4×4 pixels. In a “matching” process, a search attempts to locate the macroblock in the reference frame that satisfies a predetermined minimum-error criterion from the current frame. The ME commonly uses the SAD (sum-of-absolute-differences) error criterion:

$$SAD = \sum_{i=0}^{15} \sum_{j=0}^{15} |x_{ij} - y_{ij}|$$

where  $x$  is the current frame’s macroblock,  $y$  is the reference frame’s macroblock, and  $ij$  denotes the row ( $i$ ) and column ( $j$ ) of the frame. In some applications, the ME engine may have to calculate only 64 SADs per cycle, whereas, in others, it may have to execute thousands. The difference is significant, and, in high-end applications, it can lead to the use of architectures that either feature multiple DSPs or partition some of the calculations in a separate FPGA-based hardware accelerator.

Regardless of whether the FPGA is necessary to speedily calculate SADs or for its memory bandwidth, to be effective, it must have tightly coupled communication with the DSP. A macroblock-based pipeline-processing technique addresses this design challenge (Figure 2). The design should also reserve sufficient internal buffers to comprehend multiple macroblocks. While one macroblock is undergoing processing and a write to an internal buffer, the

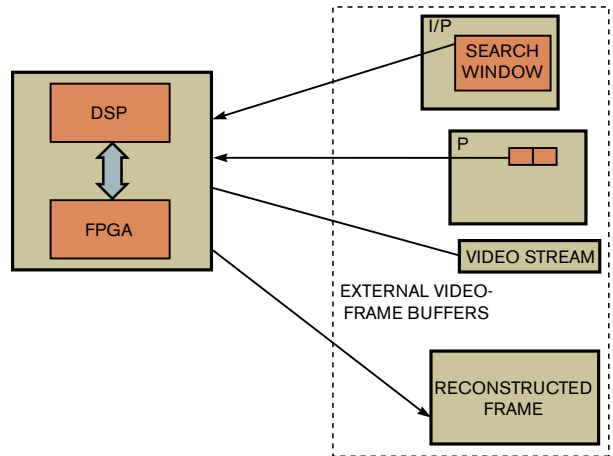


Figure 2 Macroblock-based pipeline processing encodes the P frame (middle) relative to the past reference frame. This reference frame can be either a P frame or an I frame (top). The past reference frame is the closest preceding reference frame.

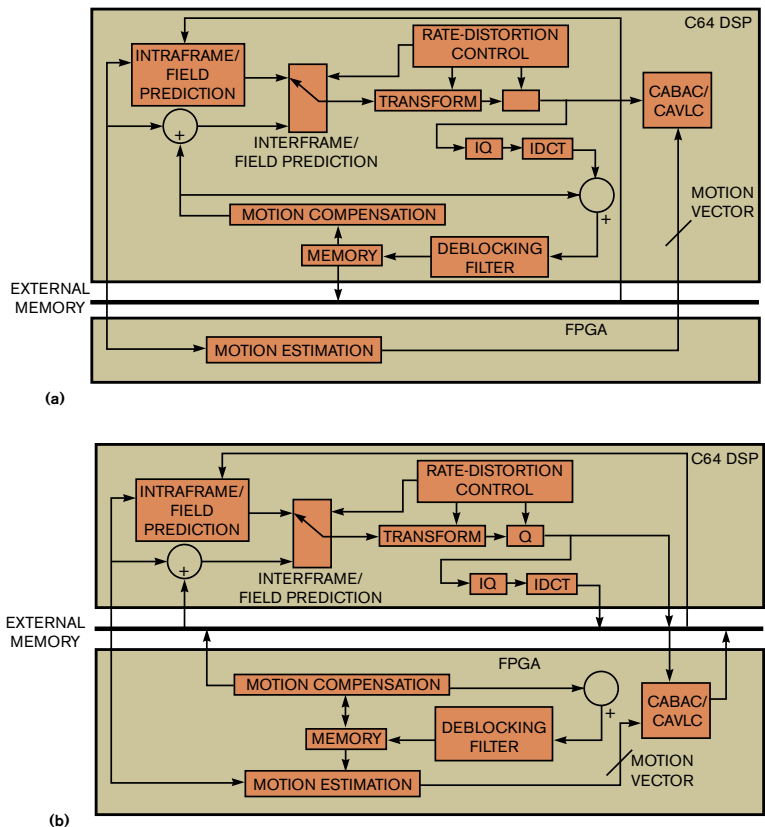


Figure 3 Executing motion compensation in the DSP requires only straightforward communication protocols (a), whereas the more FPGA-centric alternative increases the interactions between the DSP, the FPGA, and the system memory.

already-processed macroblock data in the other buffers can move to a subsequent processing unit.

In a synchronous design, it is important for the DSP and the

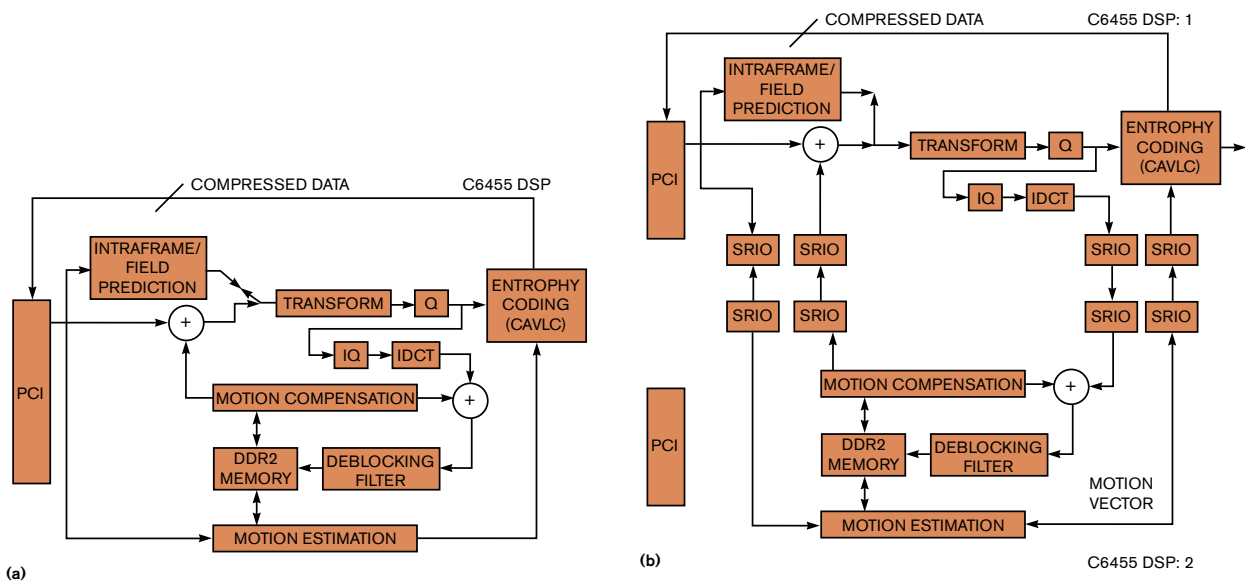


Figure 4 SD (a) and HD (b) encoding require, respectively, one and two 1-GHz DSPs.

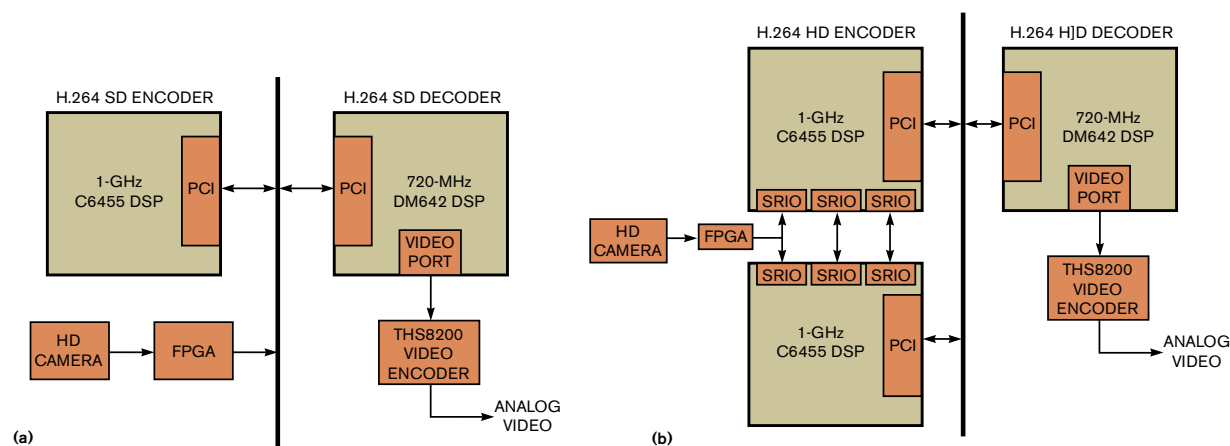


Figure 5 Simultaneous encoding-and-decoding applications at SD (a) and HD (b) resolutions supplement the encoding-only configurations with a 720-MHz media processor and an FPGA.

FPGA to access memory in a particular order and granularity, and it is important to minimize the number of clock cycles due to latency, bus contention, alignment issues, the DMA-transfer rate, and the types of incorporated memory. Interchip communication is equally critical in the implementation model (Figure 3). The architecture in Figure 3a implements only the ME engine on the FPGA, and the architecture in Figure 3b implements both the ME and the MC engines. Additional complexity in the MC case results from the fact that the ME engine and MC engine must continuously interact with each other. The architecture moves more than just the ME engine to the FPGA. The memory buffer, deblocking filter, and CABAC (context-adaptive-binary-arithmetic-coding) or CAVLC (context-adaptive-variable-length-coding) block also migrate from the DSP. CABAC compresses syntax elements in the video stream, and CAVLC, a less complex alternative, codes quantized transform-coefficient values.

The architecture in Figure 3b keeps a balance of functions between the DSP and the FPGA and enables both high per-

formance and improved flexibility for H.264/AVC encoding. However, you should avoid using it whenever possible, because implementing memory-data transfers and communication protocols between the DSP and FPGA can be complex. The architecture in Figure 3a, conversely, often is a better choice because it simplifies memory-data transfers and the communication protocol between the DSP and the FPGA.

### H.264/AVC ENCODING OF BROADCAST VIDEO

Broadcast-video encoding requires a different peripheral mix than with less demanding encoders in consumer devices' videoconferencing applications. High-end encoders must deliver high channel densities and throughput, along with a low cost per channel. The right mix of peripherals and memory go a long way toward reaching these goals. High-bandwidth peripherals are important in DSP- and FPGA-partitioning decisions and can allow designers to create high-performance applications by integrating multiple DSPs on a board. For instance, a 500-MHz DDR2 external-memory interface provides

twice the throughput of its lower speed DDR predecessor, allowing system designers to more quickly transfer data. The approach provides 2 Mbytes of L2 cache memory, enabling extra performance, further reducing the price per channel in infrastructure applications. And a gigabit Ethernet MAC (media-access controller) has 10 times the bandwidth of previous-generation devices.

Using an SRIO (Serial Rapid input/output) bus on the DSP decreases overall system cost by reducing the need for additional devices for switching and processor aggregation. SRIO interconnect also enables high-speed, packet-switched, peer-to-peer connectivity, providing a performance breakthrough for multichannel implementations on multiple processors. A one-lane SRIO link is fast enough to send 1080i raw video between devices, and a four-lane SRIO link can easily shuttle 1080p raw video with bandwidth to spare. The use of SRIO in infrastructure applications with DSP farms can significantly cut system cost by reducing device count, board size, and per-device cost.

Although SRIO is not the only option that facilitates chip-to-chip connections, it delivers several advantages over traditional interchip connections, such as PCI and EMIF (External Memory Interface). SRIO achieves 1250-Mbyte/sec bandwidth versus 133 Mbytes/sec with PCI. In addition, SRIO directly supports message passing and multicast, which PCI does not support. SRIO also uses only 16 pins versus approximately 90 for EMIF and supports seamless connection with the master and slave interface, providing robust protocol and in-band interrupts.

**MORE AT EDN.COM** ▶

+ Go to [www.edn.com/ms4246](http://www.edn.com/ms4246) and click on **Feedback Loop** to post a comment on this article.

The decreased bandwidth potential of traditional, non-SRIO interfaces may create the need for multiple parallel interfaces to achieve the required performance. Additionally, bus sharing by multiple devices can greatly reduce the I/O performance. Some interfaces can act as a master or a slave but not both, thereby requiring additional system support and glue logic. Traditional interfaces may also be physically unsuit-

able because they may consume too much PCB (printed-circuit-board) space due to wide parallel interfaces, or they may simply lack the needed advanced features, such as error detection and correction, status or acknowledgment feedback, or in-band-interrupt and -signaling functions.

## SCALABLE SYSTEMS

Integrating high-bandwidth I/O blocks into a DSP has the expected result of adding another design option. With the availability of DSPs that can satisfy the memory bandwidth of real-time HD encoding of broadcast video, you should consider using multiple DSPs in most scenarios, instead of an inherently complex DSP-plus-FPGA combination. The primary motivation for using two DSPs for HD encoding is that chip designers have solved the interchip-communications problem for you. Scalability provides another DSP-centric motivation. Because the evolution to HD has only just begun, in many instances, designers find it useful to provide an SD (standard-definition) approach that they can scale to HD with little additional effort. Employing DSPs with high-

performance I/O offers an easy migration path.

The starting point in this scalability strategy is encoding SD video. A 1-GHz DSP with a rich peripheral set can encode H.264/AVC's SD baseline profile at 720×480-pixel resolution and 30 frames/sec (**Figure 4**). Motion compensation executes on-chip. When the encoding requirement moves to HD—that is, 1280×720-pixel resolution at 30 frames/sec—you can employ two 1-GHz DSPs, with SRIO for interprocessor communication. ME and MC migrate from the chip that originally handled SD encoding to the second DSP. Note that neither of these designs requires FPGA assistance.

When the design scenario moves to an application requiring simultaneous encoding and decoding, you can still use DSPs to do most of the work (**Figure 5**). SD decoding on a 720-MHz media processor and encoding on a 1-GHz DSP benefit from the use of a separate FPGA to buffer the video from the camera. HD encoding and decoding employs fundamentally the same architecture as that of HD encoding but with the addition of the same low-cost, high-performance media processor and FPGA.

The HD system can perform simultaneous H.264/AVC, baseline profile, HD encoding, and HD decoding at 1280×720

pixels at 30 frames/sec. A DSP with SRIO provides chip-level interconnect and processor-to-processor communication at speeds as high as 10 Gbps with full duplex interconnectivity. In addition, using two or more DSPs with SRIO on the same board eases the implementation of multiprocessing architectures and ensures that no computing bottlenecks arise. A board with 10 of these DSPs, each clocking at 1 GHz and working in parallel, achieves 10-GHz performance. You can design the board to support multiple I/O modules, such as SRIO, HD SDI (serial-digital interface), and CameraLink.**EDN**

#### AUTHORS' BIOGRAPHIES

*Cheng Peng, PhD, is a video-applications engineer at Texas Instruments, where he has worked for five years. He currently develops video-surveillance-over-Internet Protocol products. He focuses on video compression, including MPEG-2, MPEG-4, and H.264; HD-video implementation; and video intelligence, including motion detection, object tracking, and object recognition, using the TMS320C6000 DSP. He received a doctorate in electrical engineering from Texas A & M University (Galveston). He has published a book and several papers in the IEEE Journal.*

*Thanh Tran, PhD, is an embedded-systems manager at Texas Instruments, where he leads the hardware-systems team in developing reference designs and frameworks for high-speed DSPs and SOCs (systems on chips). He has extensive experience in audio-, video-, computer-, and communication-system design. He has a bachelor's degree in electrical engineering from the University of Illinois—Urbana/Champaign and master's and doctorate degrees in electrical engineering from the University of Houston. He has published more than 15 technical papers and holds 20 patents. He is also an adjunct faculty member at Rice University (Houston).*