Media processing architecture scales from MP3 to HD video

By Marco Jacobs
Director of Multimedia Solutions
Multimedia Solutions for Europe and Asia
ARC International
E-mail: marco.jacobs@arc.com

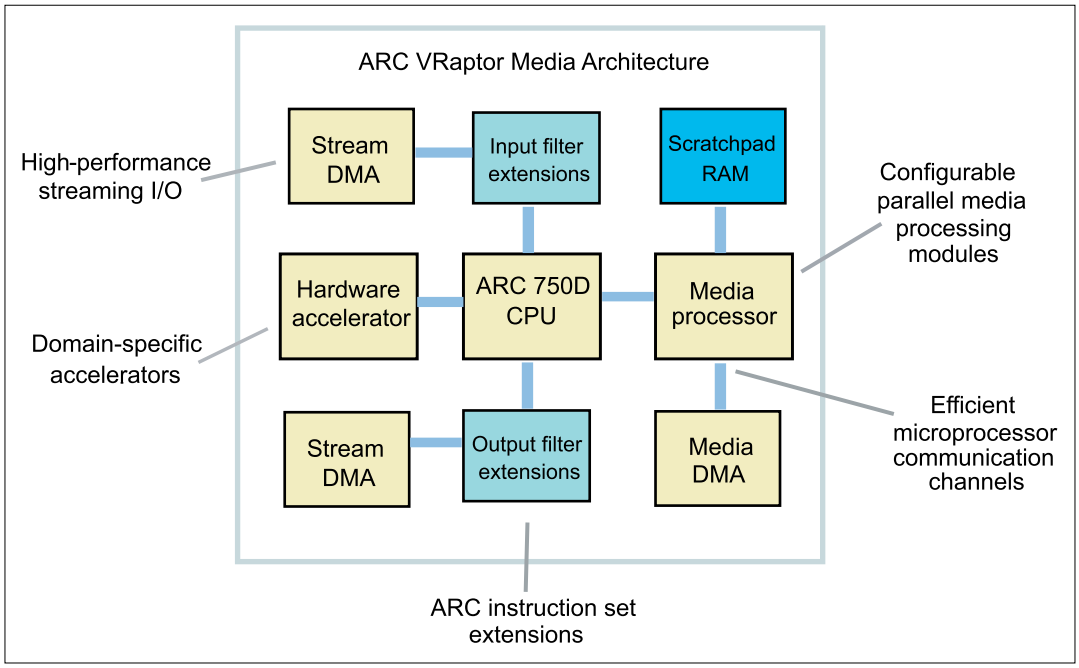

Figure 1: The ARC VRaptor Architecture takes a heterogeneous, multicore approach to complex media processing.

The landscape for multimedia devices varies widely: from sub-$20 MP3 players for kids to almost theatre-sized flat-panel displays for the home. Displaying images on a modern TV means decoding an HD-resolution H.264 bit stream, which demands three orders of magnitude more computational horsepower than decoding an MP3 file. Another example of this processing disparity is a camera phone versus a camcorder. The phone typically captures video in a QCIF (176x144) format, whereas camcorders on the market today capture video in the HD MPEG-2 format. These formats demand widely different amounts of processing performance from the devices' video subsystems. For SoC designers, it's a big benefit that their underlying media processing architecture stays the same. If not, redesigns are required to address each design point in this wide range of applications, resulting in higher SoC design costs and longer times to market.

Consumers want to play and share content across a variety of devices, thus the ability to stream and/or move content will be an important feature in future devices. Very different devices need to be interoperable. Since many A/V coding standards exist, interoperability requires the underlying media architecture to be programmable. Coding standards are then programmed in software, yielding a flexible and upgradeable end device.

Programmability also means that key algorithms affecting audio and image quality can be upgraded—a desirable feature, since engineers constantly develop new methods for improving sound and image quality, the primary factor that differentiates products from their competition. Programmability also allows device manufacturers to further differentiate their products, use the SoC in different ways, reduce the risk of silicon respins and lengthen the product's market life.

As a result, media processing architectures must be able to address high complexity levels and be software-programmable. The ARC VRaptor Architecture (**Figure 1**) takes a heterogeneous, multicore approach: multiple high-performance processors are connected to multiple SIMD processors and multiple DMA engines, and are complemented with domain-specific accelerators—all interconnected with low-overhead, low-latency active communications channels and local wide data buses.

Parallel media processing

When higher performance is required than can be delivered by a single processor running at realistic clock frequencies, performing computations in parallel is the only way to speed things up. However, Amdahl's law tells us that it's the sequential parts of the algorithm—the parts that cannot be parallelized—that ultimately determine the speed-up factor, not the number of processors operating in parallel. Few applications are easy to fully parallelize, where adding processors results in a linear speed-up of the system's performance. Real-world media applications run the gamut: They're not trivial to parallelize, but parallelization done right can certainly increase efficiency and performance. Before designing a subsystem that is a good target for applications that are to run on it, designers should take a careful look at the nature of these applications and see how the application can be split across processing elements.

Video and imaging algorithms are often easily parallelizable. The top-left pixel of an image has very little to do with the bottom-right pixel, so processing can happen independently and therefore in parallel. In a simplified image capture and display pipeline, processing goes from very parallel operations during the image capture stage, to completely serial processing during the entropy encoding/decoding stages of video compression/decompression, and back to completely parallel processing of the pixels on the display. Clearly, a hybrid architecture is required—one that's good at both serial and parallel processing.

Video coding standards such as MPEG-2 typically operate on 8x8 blocks of data, whereas more recent standards like H.264 and VC-1 operate on 4x4 blocks, exhibiting a trend toward finer granularity. Another example is H.264's many prediction modes, which exploit coherency between adjacent 4x4 blocks of pixel data and introduce dependencies. The processing closer to the lens and the display also shows a trend toward finer granularity: adaptive algorithms are widely used, which select different filters depending on surrounding pixel values, again introducing dependencies.

This fine granularity means relatively small amounts of computational work are done between communication events, making efficient communication mechanisms crucial. ARC's VRaptor Architecture's SIMD engines are designed to make the very fine-grain decisions autonomously, while the VRaptor Architecture's active communication channels provide an efficient means for synchronization and communication among processing engines.

Besides parallelization, there's another axis along which to optimize and speed up operations. Along with distributing the application over a number of processing engines, each processing or data-moving element can also be configured and optimized to best suit a particular task. ARC's VRaptor Architecture does not take a "one processor fits all" approach. Instead, its architecture consists of a collection of processing and data-moving engines, each especially suited to and configured for a particular task. They operate together in a data flow pipeline.

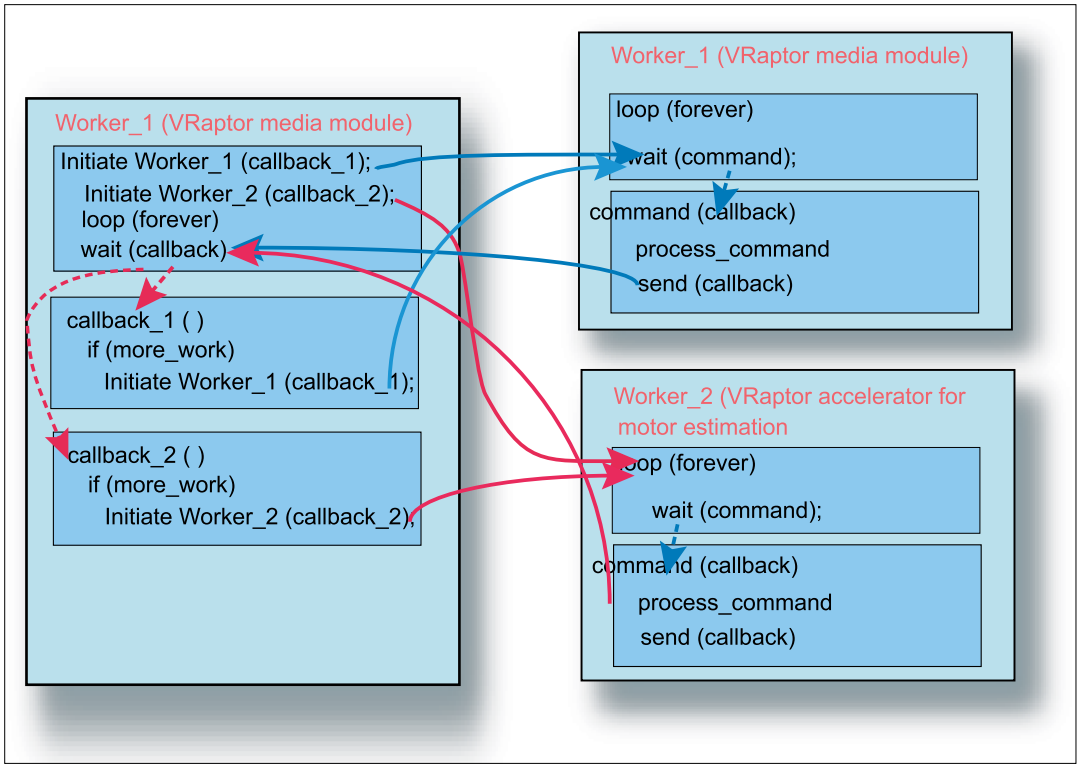

Figure 2: Active communications channels enable low-overhead, low-latency communication among processors, which is essential for implementing A/V processing applications requiring fine-grained communications.

Configurable RISC processor

The VRaptor Architecture is based on the ARC 700 configurable processor core family. The cores are silicon-efficient, and can be tailored and extended to suit a particular application need. The VRaptor Architecture can scale from a single ARC 710D CPU up to multiple configurable ARC 750D CPUs, each with multiple media processors, multiple accelerators, and multiple I/O devices.

The smallest configuration is less than 0.5 mm^2 in a 90nm process, excluding memories. Each processor is extended from a base configuration of an ARC 700 CPU architecture, which ensures instruction-level compatibility with the ARCompact ISA and allows the use of the same tool sets that support other ARC subsystems and processors. Included are MetaWare, Green Hills and GNU tools, all of which provide extensive profiling, debugging and assembly-level support.

The VRaptor Architecture extends the ARC 700 CPU family with a single-instruction, multiple-data (SIMD) media processor that operates on 128bit data vectors. Operands come from a special vector register file and can be organized as four 32bit elements, eight 16bit elements or 16 8bit elements. The SIMD processor typically operates at the same core clock frequency as an ARC 700 CPU and operates in two different ways: a closely coupled mode, which simply extends the ARC 700 CPU family's pipeline, and a loosely coupled mode, in which the SIMD processor effectively operates autonomously in parallel with the ARC 700 CPU architecture.

The SIMD processor also has full control capabilities using jump and branch instructions. Fine-grain parallel decision-making code is enhanced with predicated execution. The pipeline efficiency is enhanced with an aggressive data-forwarding network, which reduces the critical path for computations. The SIMD instruction set is fully orthogonal and includes an ability to broadcast values from scalar registers for all instructions and additional shift instructions.

The advanced media processor functionality allows the ARC VRaptor Architecture's SIMD engines to free the ARC 700 family processor for system and data flow management by offloading low-level control code from the ARC 700 family CPU. The VRaptor Architecture allows many media processors to be controlled by a single ARC 700 family control processor.

Media DMA processors

The VRaptor Architecture uses a media-enhanced 2D DMA controller to move data into and out of the system. They move data autonomously, freeing up the processors to concentrate on compute tasks. ARC's video-centric DMA has been optimized to move 2D blocks of data common in video-coding applications. The closely coupled 2D DMA engine transfers can be set up and kicked off in a very efficient manner, often requiring only a single instruction. Since video data moves consist of relatively small bursts, this is very important to reduce overhead. For example, the H.264 standard requires many 4x4 blocks of pixel data to be fetched, since motion is compensated for at this granularity. The Media DMA has multiple channels to allow both reference frame incoming data blocks and decoded pixel data blocks to be set up and run in parallel. The processing engines can share data directly also via shared wide memories. DMA processors are especially designed for moving data among the processors and system memory.

Domain-specific extensions

The VRaptor Architecture can be seamlessly extended with domain-specific accelerators. Such extensions typically implement a chunk of processing to run on an ARC 700 family CPU or SIMD processor, but these processing chunks are implemented much more efficiently in a fixed hardware block. The VRaptor Architecture includes multistandard, variable-length decoders and encoders. These entropy-coding blocks cannot be parallelized owing to data dependencies, and thus, because of the high number of look-ups in small tables, the encoding or decoding of symbols is more efficiently implemented as a hard-wired extension, though the decoders and encoders remain flexible enough to address many video-coding standards.

Motion estimation is another area where a repetitive simple sum of absolute difference operation, requiring heavy data shuffling and moving, is executed on huge amounts of data. Although such a motion estimation algorithm can be implemented very efficiently on a VRaptor Architecture SIMD engine, a careful design exploration and trade-off demonstrated a clear advantage in implementing the motion estimation for video encoding as a hard-wired extension. While exploring the design space of hard-wired extensions vs. fully programmable blocks, the tool automatically configures the software and compilation tools such that they match the selected system configuration. This allows system designers to quickly measure performance and implementation results without having to adapt or change any software by hand.

Active channels

The 32bit active communications channels are the means through which processing engines communicate. Active communications channels enable low-overhead, low-latency communication among processors, which is essential for implementing video and audio processing applications requiring fine-grained communications. Shared data memories between the processing engines enable local, fast and wide data communications.

The patent-pending active communication channels technology in the VRaptor Architecture is an active channel protocol that is based on remote invocation. It provides point-to-point links in hardware between the VRaptor cores, VRaptor accelerators, and VRaptor I/O elements, and carries commands as well as arguments. Channels are buffered and, for increased performance, have input queues. Thus, sender and receiver do not need to "rendezvous in time" to send or receive messages. The active communications channels enable the sender to explicitly update registers in the receiver. This allows very efficient remote method invocation (i.e. a remote procedure call) by virtue of the fact that the sender can write directly to function argument registers and then write the entry point of a remote function to the program counter. Seen from the receiver, this operation takes the same number of clock cycles as a local function call, which is very efficient.

The active communication channels technology is directly supported by instruction extensions in ARC's configurable ISA and eliminates message interpretation overhead. It provides for a unified programming model that simplifies the programming overhead often associated with multiprocessor architectures.