StrongArm-1500 Grapples With MPEG-2

Digital's Attached Media Processor Works As Programmable Accelerator

by Jim Turley

Digital Semiconductor has teamed its muscular StrongArm processor with an equally overachieving media coprocessor to produce an integrated multimedia processor for network computers and high-end consumer "information appliances." The SA-1500 combines an integer CPU, FPU, DSP, and media processor in one. The chip easily handles MPEG-2 decompression, making it suitable for media-rich systems for entertainment.

At October's Microprocessor Forum, chip architect Robert Stepanian described Digital's design approach for the SA-1500. Rather than press the ARM instruction set into service as a media engine, Digital designed its own programmable media processor from scratch and paired it with the RISC CPU. As fast as StrongArm is, the ARM instruction set remains one of the few modern architectures without media-processing extensions, making it unsuitable for standalone media processing.

Shortly after Stepanian's presentation at the Forum, a cloud fell over the SA-1500, as well as other StrongArm chips, as Intel moved to purchase Digital Semiconductor's assets, including the entire StrongArm product line and the people who work on it (see MPR 11/17/97, p. 1). The prospective new owners appear deeply divided over the future of StrongArm within Intel. Regardless of whether StrongArm ultimately stays, is sold, or is spun off, customers will remain wary until a definitive statement is made one way or the other. Unfortunately, no such statement appears likely before the chip enters production next spring.

Digital architect Robert Stepanian describes the StrongArm-1500 at the Microprocessor Forum.

MICHAEL MUSTACCHI

Chip Sports Dual Buses, Caches, and Processors

As the block diagram in Figure 1 shows, the SA-1500 includes a StrongArm core with a pair of 16K caches, two bus interfaces, and a package of additional logic collectively known as the attached media processor. The chip maintains separate buses for memory and for I/O. The 64-bit memory interface includes control for 2–4 banks of 100-MHz synchronous DRAMs. The 32-bit I/O bus is used for everything else: ROMs, SRAMs, and peripherals.

The 64-bit, 100-MHz SDRAM interface is a vital element of the chip because the SA-1500 (code-named Bigfoot) is designed to be a voracious data monster. According to Stepanian, the chip can do full MPEG-2 decompression in real time (video and audio, main profile, main level) without breathing hard. With a network or cable interface, keyboard, and mouse, the SA-1500 would make a fine controller for a television set-top box, network computer, or videoconferencing system. Digital also foresees applications in modem banks that take advantage of the chip's DSP capabilities.

The StrongArm core was modified slightly from the version in the SA-110, adding four 32-byte read buffers and a 1K "minicache" in addition to the usual data cache. The read buffers are loaded explicitly through software and can potentially alias the regular data cache. Their purpose is to allow programmers to preload expected data without depending on the vagaries of normal caching protocol.

AMP Provides Media Muscle

The heart of the SA-1500 is the attached media processor, or AMP. The AMP is a fabulously elaborate processor in its own right, with a separate user manual that is larger than the SA-1500's. As the name implies, the AMP handles media processing, including SIMD (single instruction, multiple data) integer and floating-point calculations, multiply-accumulates, and signal-processing work.

The AMP is completely autonomous, able to operate independently of the new chip's StrongArm core. As Figure 2 shows, the AMP has its own register set, writable control store (WCS), prefetch buffer, and bus interface. The media processor has its own instruction set, which it executes in parallel with ARM code. From the CPU's point of view, the AMP is an ARM coprocessor, much like a floating-point unit or the Piccolo DSP coprocessor (see MPR 11/18/96, p. 17).

The AMP is a two-operation long-instruction-word (LIW) processor. Each 64-bit AMP instruction includes two 32-bit opcodes, one for the AMP's main execution unit (EXU) and one for its memory/branch unit (MBU). Operations (as distinct from the long instructions) are always dispatched in pairs and execute in order, although they don't necessarily complete at the same time. Both units are pipelined and will accept a new instruction every clock cycle.

AMP Instructions Work in Parallel

The AMP fetches instructions from its writable control store (WCS), which is filled explicitly by ARM code. The WCS

holds 4K bytes, or 512 AMP instructions. Digital believes 512 instructions are enough for common media-processing subroutines. If the function is too large to fit in the WCS, more code must be manually loaded; the WCS is not a cache, and programmers are responsible for swapping sections of AMP code in and out.

After the WCS is loaded, the CPU can issue a function call to the AMP. AMP processing then proceeds until it hits a HALT instruction. The CPU will trap if another AMP call is issued while the first is still executing. The CPU and the AMP can operate either in parallel or in a tightly-coupled manner in which the CPU "spoon feeds" the AMP one instruction at a time. In the parallel method, not only does the AMP work at the same time as the CPU, but the AMP itself executes two operations at once.

As Figure 2 shows, AMP's two major pipelines are subdivided into six execution units, three on each side. Individual units handle shifts and ALU operations, multiplication (with accumulation), IEEE-754 single-precision FP operations, branches, memory accesses, and prefetching.

Most arithmetic and logical operations handle 8-, 9-, 16-, 18-, 32-, and 36-bit data types. (The extended data types are commonly used for additional precision in Dolby AC-3 and MPEG-2 audio processing.) Nearly all 8- and 9-bit operations handle four operands simultaneously, as in Intel's MMX. Similarly, most 16- and 18-bit operations calculate two results in parallel.

All AMP instructions are pipelined. EXU operations have a three-cycle latency and therefore a three-cycle delay before data is available to the next instruction; MBU operations finish in two cycles. The exceptions to the rule are accumulating operations, which are specially bypassed to allow consecutive MAC operations. There is a two-cycle load-use penalty. The hardware includes interlocks that will stall on register dependencies.
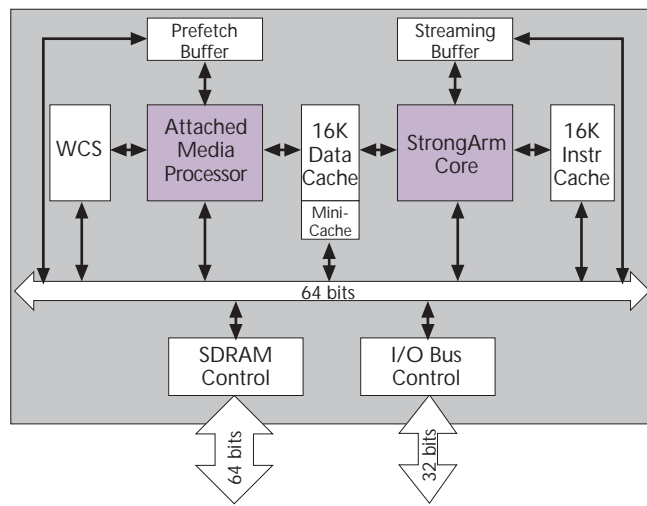
AMP Code Handles Its Own Data Requirements

Media processing is typically very data-intensive work, so starving the execution units is a constant worry. The SA-1500 already has two processors to work on the task, but the AMP has one execution unit devoted solely to keeping the other five fed. A special prefetch unit (one of the six execution units) can be programmed to pre-emptively fetch data from memory and move it onto the chip. Like an intelligent DMA controller, the prefetch unit can have up to five outstanding prefetches in progress, and prefetch descriptors can be chained together.

The prefetch unit accumulates data and stores it in a dedicated 256-byte prefetch buffer. Completed prefetch operations can optionally interrupt the AMP, signaling the arrival of requested data. Such interrupts then redirect the execution of the AMP. Thus, the AMP can be programmed to hop about among tasks as the data for them becomes available on-chip.

Knowing when to schedule and dispatch prefetch threads can be a precarious programming exercise that calls for balancing bandwidth, interprocess communication, and memory latency into one well-optimized function. Fortunately, Digital will provide pretested routines for MPEG-2 audio/video decoding, as well as Dolby AC-3 audio decoding.
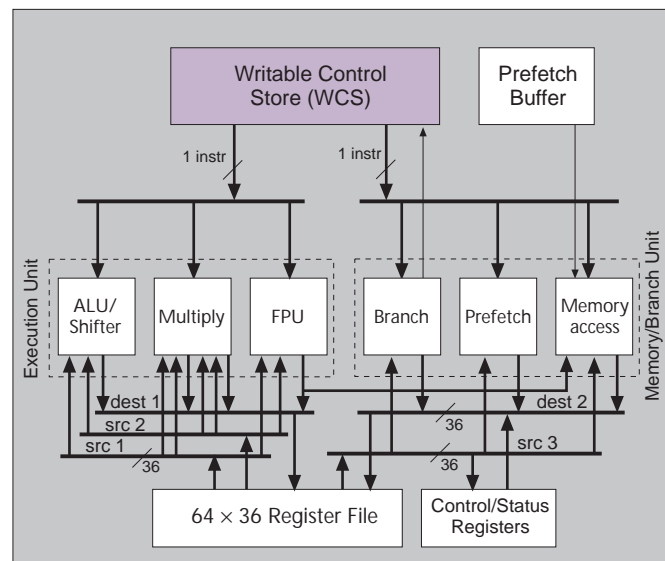

Figure 1. Digital's SA-1500 pairs the StrongArm core with a 64-bit attached media processor (AMP), dual bus interfaces, and caches. (WCS = writable control store.)

Figure 2. The SA-1500's AMP includes six execution units, two of which can run concurrently to perform audio and video convolutions in parallel with ARM code.

Programming Is a Real Task

Like ARM's other major coprocessor, Piccolo, the AMP requires significant effort to yield the best results. Also like Piccolo, the AMP executes its own private instruction set at the same time that the host processor (ARM) is executing its code. In the case of the AMP, that instruction set includes an assortment of SIMD, audio, video, and floating-point operations, including dozens of variations on multiply-accumulate, the lifeblood of most media-processing jobs.

The AMP does nothing automatically. Its code space, in the form of the writable control store, must be loaded manually. Routines that overflow the WCS must be manually paged in or out. Although the AMP can load and store its own data (sharing the data cache with the CPU), it works best when data is explicitly prefetched into the private prefetch buffer. On the plus side, all this manual scheduling makes AMP execution predictable and deterministic. Neither code nor data caches will stand between the AMP and its appointed task.

Aggressive Business Calls for Aggressive Pricing

With no price announced, it's impossible to judge the new chip's merits in a realistic way. Digital has a history of pricing its StrongArm chips aggressively. At, say, $30, the SA-1500 would be a bargain for a processor with as much added value as this one carries. At $75, it would probably fail. The chip is almost unique among processors that can legitimately claim to massage MPEG-2 data in real time and produce respectable results (Hitachi's SH-4 is the other). But that is a false comparison. The SA-1500's real competition is not from other microprocessors but from hardwired MPEG-2 decoders and media processors coming at it from all sides.

Both dedicated and programmable MPEG-2 decoders, such as those from C-Cube and Philips Trimedia, perform basically the same task as the SA-1500. All have wide buses and provisions for shuttling lots of data on and off the chip. Multiple execution units (with emphasis on multiplication and accumulation) keep DCTs and IDCTs running in real-time.

Most MPEG-2 decoders require a separate microprocessor to oversee the decoding process and control the rest of the system. By including its own StrongArm core, the SA-1500 provides a single-chip solution. If Digital prices its chip appropriately, it should be able to offer a cost advantage over these two-chip solutions. The Philips Trimedia chip is one exception, as its programmable engine can handle control tasks as well as video decoding. The StrongArm core is much easier to program than Trimedia's VLIW engine, however.

In the market for MPEG-2 decoders that can do it all, with ease of use and horsepower to spare, the SA-1500 stands alone. But the nascent market for set-top boxes must continue to grow for the SA-1500 to achieve its potential. Ⓜ