

# Fujitsu Aims Media Processor at DVD

## MMA Combines Long-Instruction-Word Core, Integrated Peripherals



by Peter N. Glaskowsky

Hoping to claim a share of the growing consumer multimedia market, Fujitsu's Shunsuke Kamijo described the company's new multimedia assist (MMA) processor at last month's Microprocessor Forum. The new architecture features a two-pipeline long-instruction-word (LIW) core capable of executing up to six 16-bit integer operations simultaneously for a peak rate of 1.08 GOPS at 180 MHz. This is a higher clock rate than most other announced media processors, the result of a design that favors simplicity over sophistication.

The first chip in the family adds two 8K SRAMs; graphics, DMA, and SDRAM controllers; and several integrated peripherals to the MMA core. Fujitsu did not announce pricing or availability; we expect to see this product in limited sampling by the end of the year, with production volumes available in the first half of 1997.

Unlike other recently announced media processors, MMA is intended for consumer electronics, such as intelligent televisions and DVD players. MMA can perform DVD decoding, modem functions, and videoconferencing in software (but not all at once). MMA provides no floating-point support, 3D acceleration features, or PCI interface, making it unattractive for the PC market.

### MMA Core Includes Five Execution Units

The heart of MMA is its LIW core with two pipelines and five execution units, as Figure 1 shows. Each 64-bit instruction word contains a pair of 32-bit instructions. One instruction is always dispatched to the primary pipeline and the other to the secondary pipeline. This simple LIW is much less sophisticated than the VLIW architecture of Philips' TriMedia (see [091506.PDF](#)). In TriMedia, each instruction word contains up to five operation slots, and there are 27 execution units. Each unit is fully pipelined, allowing the TriMedia core to sustain five operations per clock in long code segments.

In MMA, one pipeline contains a 32-bit ALU and the load/store unit. The other contains a second ALU along with multiply-accumulate and divide-shift units. When the primary ALU is used in combination with the multiply-accumulate unit in the secondary pipeline, the MMA core can achieve a peak execution rate of three 32-bit or six 16-bit operations per cycle.

MMA's instruction-pairing rules are relatively simple. All five execution units are fully pipelined, yielding single-cycle throughput, so one instruction can be dispatched to each pipeline in each clock cycle. Division and modulo operations are an exception: these take 33 to 36 cycles to complete and stall both pipelines while they execute. Taken branches

require three cycles; one branch-delay slot is provided, which may be used for any single-cycle operation.

Fujitsu also provides 27 instructions for multimedia operations such as MPEG decoding and surround-sound processing. These include variations of ADD, SUB, and MAC instructions with signed and unsigned saturating arithmetic as well as 16-bit SIMD versions of most instructions.

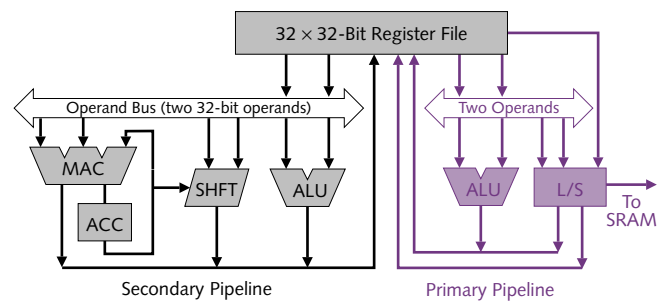
Although the load/store unit supports byte-size data types, the ALUs support only 16- and 32-bit data. Other media processors support byte operations, allowing twice the peak performance when dealing with byte-oriented data types like RGB pixels or 8-bit sound.

MMA provides an unusual saturation mode. A source register may specify the bit at which saturation occurs. This mode allows 32-bit operations to perform 24-bit saturating arithmetic, for example. This should prove useful, since few multimedia data types are exactly 16 or 32 bits in length. For example, MPEG decoding uses 9-bit values, and AC-3 audio requires 24-bit precision. Saturation to an arbitrary (non-power-of-two) value, however, is not supported.

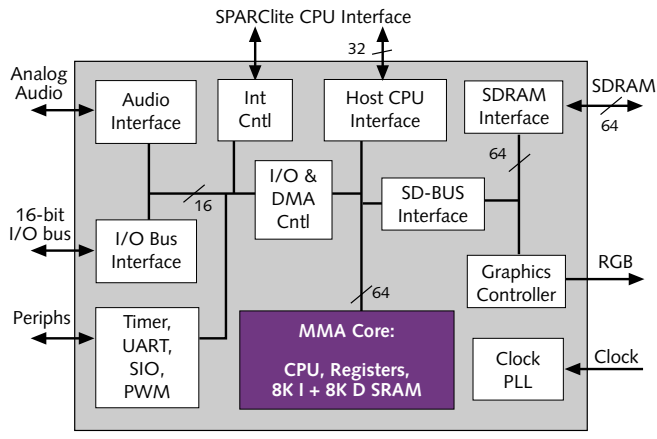
### Register File Matched to Pipeline Requirements

The MMA core includes a  $32 \times 32$ -bit multiported register file. It supports five simultaneous reads and three writes, enabling sustained single-cycle throughput on the inner loops of common multimedia functions.

This register set is smaller than in most competing media processors. For example, TriMedia has 128 32-bit registers. These larger register sets are especially useful for 3D acceleration. MMA's register set was designed for algorithms like the inverse discrete cosine transform (IDCT) in MPEG decoding. These algorithms typically have working sets that will not fit in a register file; instead, registers are used to store control values and coefficients, and data storage depends on fast access to memory.



**Figure 1.** The MMA core includes two pipelines that operate in parallel. The primary pipeline supports ALU and load/store operations, while the secondary pipeline can perform multiply-accumulate, shift, and ALU operations.

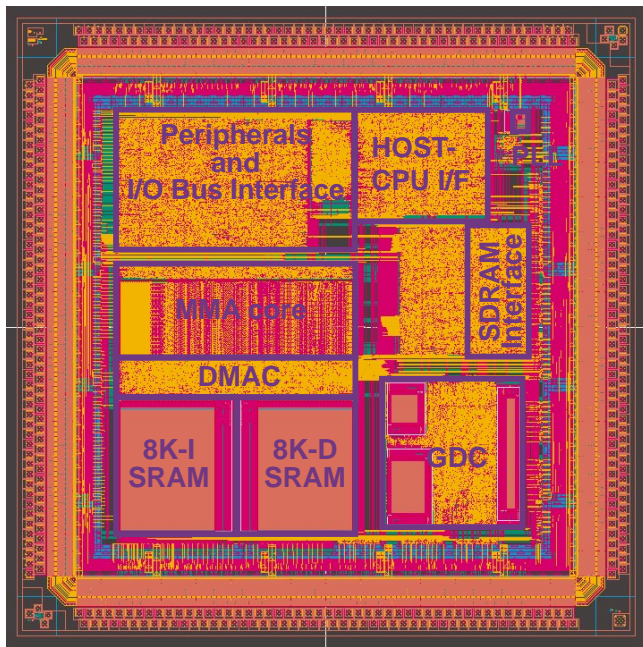


**Figure 2.** To support consumer multimedia applications like intelligent television, MMA includes many functional blocks in addition to the programmable MMA core.

### SRAMs, Not Cache, Meet Core Bandwidth Needs

Rather than relying on caches to reduce average memory access penalties, MMA provides an 8K SRAM instruction store plus an 8K SRAM data store. Each SRAM is organized as 1K 64-bit words, and both operate at the full 180-MHz rate of the core. The instruction SRAM can provide a 64-bit dual instruction word to the core on each clock, while the data SRAM can transfer one, two, or four bytes to or from the register file in each cycle.

Unlike caches, the SRAMs must be managed by software. Transfers between the SRAMs and the local SDRAM can be performed only by the on-chip DMA controller, which is interlocked with the core. During a DMA transfer,



**Figure 3.** This is a die plot of MMA, which is fabricated in a 0.35-micron three-layer-metal CMOS process. The die size is 77 mm<sup>2</sup>, with 1.3 million transistors. The MMA core is only 4.3 mm<sup>2</sup> in size.

the core is stalled. The DMA controller supports normal block transfers at the maximum rate of the DRAM interface, as well as rectangle transfers, a way to realign raster-oriented video data. DMA transfers are controlled to byte boundaries, a necessary feature given the unpredictable block lengths of digital audio and video data.

In effect, MMA trades the complexity of a cache controller for added software complexity. This tradeoff would be unacceptable in a general-purpose processor, but MMA is designed to execute simpler multimedia code that can be optimized for this architecture. A similar scheme is used by Chromatic's Mpact, although Mpact's software "caches" are multiported and offer much higher net throughput.

### SDRAM Interface Sustains 1-Gbyte/s Bandwidth

Figure 2 shows the internal organization of the first MMA product. The MMA core includes the LIW engine, SRAMs, and DMA controller. The integrated SDRAM controller manages up to 32M of SDRAM operating at up to the 180-MHz pipeline rate. While current SDRAMs do not support this high speed, future parts will. At 180 MHz, the peak data rate is 1.4 Gbytes/s, but sustained rates will be lower due to bus turnaround delays and page misses.

In more realistic implementations, the SDRAM interface will run at half the core speed, or 90 MHz, yielding 720 Mbytes/s peak and about 500 Mbytes/s sustained throughput. The SDRAM interface supports four-word burst transfers and is fully pipelined, so subsequent reads to the same DRAM page do not cause wait states.

### Integrated Peripherals Adapt MMA to TV

To support intelligent televisions, MMA includes a graphics display controller (GDC) module. In addition to interlaced and noninterlaced NTSC and VGA resolutions, the GDC also supports a "wide-VGA" mode of 860 × 480 pixels that can be used to enhance quality on high-end televisions. Such televisions offer greater horizontal than vertical resolution, so typical "square pixel" display modes like 640 × 480 do not achieve the best possible visual quality. MMA's wide-VGA support allows processor-generated content like the graphical user interface to be displayed directly in the higher resolution, while some digital video content, such as wide-screen-mode DVD playback, can be scaled to fill the wider effective screen size.

The GDC manages three display windows, each with a separate frame buffer in the local SDRAM. The windows can be positioned and overlapped arbitrarily. Frame buffers can contain pixels in the YCrCb color space for MPEG decoding, RGB for user-interface displays, or a 15-color-plus-transparency mode for processor-generated captions and simple graphics. As the screen is drawn, these pixel types are all converted to standard digital RGB using a color-space conversion engine in the GDC.

The GDC shares access to the SDRAM with the DMA controller and host processor. SDRAM refresh activity can

be synchronized to the display controller, taking place during the horizontal refresh interval. This eliminates the need for deep FIFOs in the display refresh path, since the display controller can always depend on uninterrupted access to the SDRAM during the display period of each scan line.

Other peripherals on the die include a timer, two serial I/O controllers, and a pulse-width-modulation module for motor control, plus interfaces for an audio chip and other off-chip peripheral devices. MMA does not include a RAM-DAC, however, which would have been fairly easy to add given the relatively low resolution display modes it supports.

MMA is not designed for general-purpose tasks like control and communications. Instead, it will typically be used as a coprocessor for multimedia tasks, coupled with a general-purpose host CPU like Fujitsu's SparcLite.

The first MMA implementation has a 32-bit SparcLite interface built in. In this configuration, MMA acts as a unified memory architecture (UMA) DRAM controller for SparcLite, storing the frame buffer plus code and data for both processors in the local SDRAM. A typical intelligent TV controller would consist of the MMA, a SparcLite processor, ROM, RAM, a RAMDAC, and a few analog interface components.

### Software Development Tools

Fujitsu has made a special effort to enable third-party software development, but it has not announced specific third-party relationships. Fujitsu commissioned a full suite of development tools from Green Hills ([www.ghs.com](http://www.ghs.com)), extending the existing set of SPARC tools to include an assembler and simulator for MMA as well as an MMA-aware version of the Green Hills Multi development environment. Multi gives an MMA programmer a unified environment with separate windows for SPARC and MMA operations.

Fujitsu has also developed a version of Wind River's VxWorks ([www.wrs.com](http://www.wrs.com)) that runs on the SparcLite/MMA target system, supporting remote-control debugging operations through VxServ. VxWorks runs only on the SparcLite processor; at this time, Fujitsu has no real-time OS kernel for MMA itself.

Fujitsu is developing its own set of essential software libraries for MMA. At the Forum, Fujitsu's Kamiyo showed its development schedule for six library functions: MPEG decode and encode, JPEG decode and encode, JBIG (a lossless still-image compression scheme), and the V.34 modem algorithm. All of these functions are projected to be available by 1Q97.

### Benchmarks Demonstrate MMA Performance

Kamiyo showed limited benchmarks based on MPEG-1 performance. The 180-MHz MMA can decode video-only

### For More Information

Pricing and availability for MMA have not been announced. Contact Fujitsu (San Jose, Calif.) at 408.922.9574 or on the Web at [www.fujitsumicro.com](http://www.fujitsumicro.com).

MPEG-1 bitstreams at 118.4 frames per second (fps), suggesting that standard 30-fps MPEG-1 requires only 25.3% of the device. When standard 48-kHz audio is included, the decode rate dropped to 94.1 fps, increasing utilization for 30-fps MPEG-1 to about 32%.

Although Fujitsu has not released benchmarks for DVD applications, the company says MMA will be able to perform MPEG-2 video plus Dolby AC-3 audio decoding at 30 fps, meeting the basic requirement for DVD support. This is a critical capability; without DVD capability, MMA could be relegated to the much smaller market for video karaoke players and other MPEG-1 products.

Fujitsu did not discuss performance on other multimedia tasks like V.34 modem operation, but it is unlikely that MMA will be able to support a V.34 modem connection while simultaneously decoding DVD content.

### Goals Determine Results

The first MMA is shown in Figure 3. The die size is 77 mm<sup>2</sup>, fabricated in a 0.35-micron three-layer-metal process, operating at 3.3 V and packaged in a 352-ball BGA. Based on the MDR Cost Model, the estimated manufacturing cost of this part is \$30. The MMA core, a full-custom design, occupies only 4.3 mm<sup>2</sup> of the die. Even at this small size, it offers more than enough processing power for the target applications. Power consumption for the device is very low, at only 600 mW (typical). This compares very well with Trimedia's TM-1 at about 4 W, offering an advantage in consumer applications.

MMA is a well-balanced part that should compete effectively against non-programmable devices in DVD players while also working well in products that require more intelligence. Fujitsu's primary competition will come from dedicated DVD decoders from C-Cube and Oak (see [1015MSB.PDF](#)). Hardwired devices may be smaller and less expensive than MMA, making them a better choice for cost-sensitive DVD players, but MMA is more flexible, making it more suitable for an intelligent TV.

In addition, MMA is likely to be much less expensive than PC-oriented media processors like TM-1 due to its smaller SRAMs and lack of floating-point support and 3D acceleration, making it a good fit in the embedded multimedia applications for which it was developed. ■



**Shunsuke Kamiyo describes Fujitsu's first media processor at the Microprocessor Forum.**

MICHAEL MUSTACCHI