MICROPROCESSOR REPORT
THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 6 NUMBER 16 DECEMBER 9, 1992

First Mwave DSP Provides High Integration

Texas Instruments' TMS320M500 Aims to Bring Multimedia to the Masses

By Linley Gwennap

Texas Instruments has announced the first digital signal processor (DSP) to support the Mwave standard announced earlier this year by TI, IBM, and Intermetrics *(see **060602.PDF**)*. Mwave defines a set of hardware and software interfaces that the companies hope will be accepted by the industry as a standard for multimedia PCs. The new M500 DSP (TI uses the prefix "TMS320" for all of its DSPs) allows the alliance to provide sample hardware and software that vendors can begin designing into their products. Although IBM is committed to Mwave for its multimedia PCs, it remains to be seen whether the proposed standard will be widely embraced.

Figure 1 (see below) shows a block diagram of the M500 chip. The core of the chip is a 17-MHz fixed-point CPU that implements the signal-processing algorithms. The CPU has separate instruction and data buses so that it can sustain its peak execution rate of 17 MIPS. These buses connect to external program and data memory. The M500 also connects directly to a PC bus, either ISA or Micro Channel (MCA), allowing the DSP to access the host system memory and communicate with the main CPU. The chip also includes a variety of I/O interfaces to handle serial and audio data and connect to MIDI (Musical Instruments Digital Interface) devices. A powerful DMA engine transfers data from DSP memory to host memory without interrupting the DSP CPU.

CPU and System Interface

The M500 architecture was designed by IBM and TI, and it is different from any previous TMS320 design. It uses 24-bit instructions and 16-bit data words, and it includes 8 general-purpose registers along with various status and control registers. Two of the general registers can hold 32-bit values, while the other six are limited to 16 bits. A special 32-bit register holds the output of the 16×16-bit multiplier. The chip also includes a cycle-count register that software can use to enforce execution time limits.

The multiplier can generate a 32-bit result in a single (59-ns) cycle. The ALU can perform logical operations on 16-bit values, but the adder is extended to 32 bits to allow 32-bit products to be summed without rounding. No additional bits are provided for overflows during long sums of products (as both Motorola and AT&T provide). All 16- and 32-bit ALU functions are completed in a single cycle. The ALU also performs the saturation and rounding functions required by many signal-processing algorithms.

Although the M500 handles some 32-bit operations, the 16-bit multiplier will provide poor low-frequency resolution for manipulating CD-quality audio. Also, the M500 provides no support for floating-point or multiple-precision arithmetic.

The CPU supports a variety of addressing modes, including immediate, direct, indirect, absolute, circular, and indexed (table) addressing. The chip does not provide bit-reversal addressing, used for FFT (Fast Fourier Transform) algorithms.

The on-chip I/O interfaces are accessed through registers mapped into the lower portion of the memory space. The M500 supports up to 64 Kbytes of data memory organized as 32K 16-bit words. The program memory can be up to 96 Kbytes, organized as 32K 24-bit instructions.

When connected to ISA or MCA, the DSP can act as either a bus master or slave device when transferring and receiving data. The DSP can program its on-chip DMA engine to move blocks of data between the host system memory and the DSP memory (either program or data). DMA can also be used for transfers between the on-chip peripherals and the DSP data memory. In either case, the DSP's CPU can continue running during the DMA operation; if both the CPU and the DMA need access to DSP memory, the DMA takes priority. (Since the maximum DMA bandwidth is much smaller than the CPU bandwidth, this priority does not unduly restrict the CPU.) The bus arbiter ensures the smooth flow of data within the DSP.

Peripheral Interfaces Included

The M500 includes a wide variety of interfaces to typical signal-processing peripherals, lowering subsystem cost and distinguishing the M500 from its competition. The chip includes an interface to a CD-quality sound subsystem, two interfaces to voice-quality CODECs, a MIDI port, and a standard UART serial interface.

The CD-quality sound interface allows the chip to be connected directly to an external 16-bit A/D converter (sound input) and a 16-bit D/A converter (sound output). These ports are serial ports and assume that the external converters can accept serial data. A left/right signal allows each port to handle two channels of audio data. The DSP software configures these sound ports using memory-mapped registers, and transfers data using DMA. A third serial port is available for controlling an external device such as a volume control or multiplexer.

Two additional serial ports can be used for voice, audio, or general purposes. The M500 can be configured to connect directly to TI's TLC32046 AIC (analog interface chip), which provides A/D, D/A, and filtering functions for 14-bit data. These ports can also connect to an external 8-bit CODEC, such as TI's TP3054B, to provide logarithmic compression of analog signals. These two options are shown in Figure 1. In general-purpose mode, the ports act as standard synchronous serial ports with 16-bit data words.

The chip can also emulate a standard UART interface for one of these AIC ports. In this mode, the AIC port is accessed directly by the host CPU through I/O registers on the PC bus interface, allowing it to be used as the system's COM port if desired. In the configuration shown in Figure 1, the serial data would be sent across the phone line.

The MIDI interface consists of three signals implementing MIDI in, MIDI out, and MIDI out/through ports. The M500 can configure the MIDI interface in several modes. MIDI data can be passed straight through, written into the DSP memory for processing, or written into the host memory under the control of the DSP. No additional logic is needed for a complete MIDI port.

TI's Office Pro board is the first product to incorporate the M500 DSP. It uses the subsystem configuration shown in Figure 1 and discussed above. Notice that the audio control port drives a switch array through a decoder, allowing software to select between multiple stereo sound ports.

M520 Allows Low-Cost Configuration

For a lower-cost audio-only solution, TI offers the Personal Audio board. This board uses a derivative of the M500 called the M520. This chip integrates 12 Kbytes (4K × 24) of program RAM and 8 Kbytes (4K × 16) of data RAM. To make room, the M500's voice interface, UART, and audio control port are left out, but all of the other M500 peripherals are included on the M520. By eliminating the external memory interface, the package size is reduced from 240 to 144 pins.

The M520 costs about $5 more than the M500, but the overall subsystem cost is much lower due to the integrated memory. One drawback is that the amount of memory on the chip is generally not sufficient for fax, modem, or some telephony functions; TI is working to reduce the size of these DSP tasks to fit into the M520's on-chip memory.

The Personal Audio board consists mainly of the M520 and a CS4215 for stereo sound input and output. No external memory or other CODEC parts are needed. All of the components fit on a half-size ISA card. This board will sell for just $99 in OEM quantities.

Software Layers Provide Flexibility

Mwave includes several layers of software, as depicted in Figure 2. The key interface is the API (Application Programmers Interface), through which an application program can access the functions of the DSP. For Mwave to succeed, a large number of independent software vendors (ISVs) must agree to write their applications using this API. Already, key companies such as Borland, Lotus, WordPerfect, and

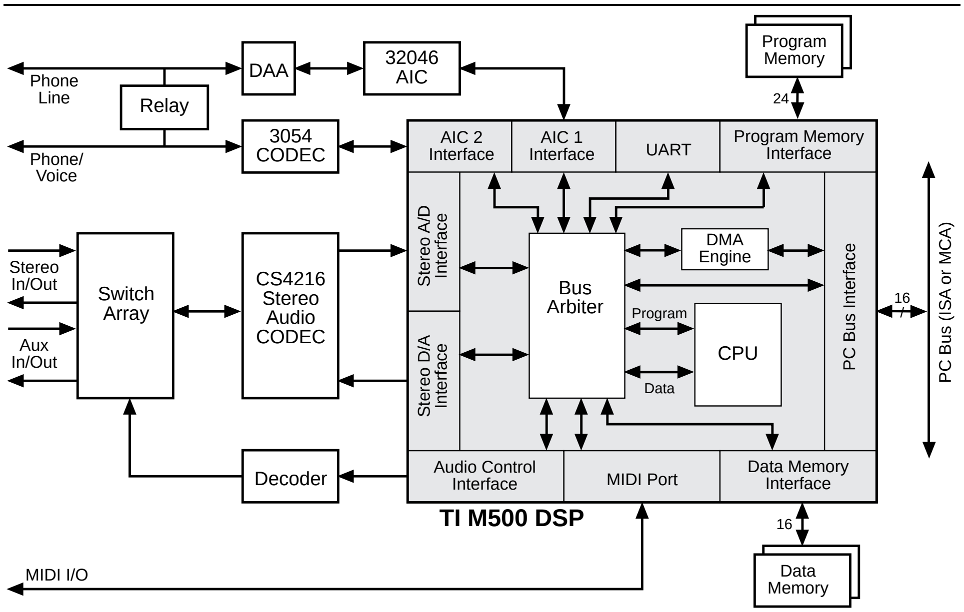

Figure 1. Block diagram of M500 DSP in Office Pro board design.

Harvard Graphics have announced their intention to support Mwave.

The API uses fairly general parameters. For example, to send a fax, a program would call the fax subroutine with a pointer to the file, a phone number, transfer protocol, etc. Because of this generality, other vendors could write device drivers for their own DSPs and still meet the Mwave standard; that is, applications that used the Mwave API could access non-Mwave DSPs without changing their programming. At this time, however, no other DSP vendors are supporting the Mwave API, and TI has not ported Mwave to its other TMS320 DSPs.

Mwave provides a wide range of device drivers. For audio, it offers pulse code modulation (PCM) compression, adaptive PCM compression, text-to-speech conversion, audio playback and mixing, and an eight-voice music synthesizer. Telephone services provide a virtual answering machine, voice/fax discrimination, autodialer, sub-band coder (SBC) for compression, and message speed control. Other available services include JPEG image compression, 9600 bps data modem using V.22bis and V.32 protocols with MNP5 compression, and Group 3 fax modem.

The device drivers take the requested operation from the application and break it down into individual tasks for the DSP, such as moving and compressing data. These tasks are then handed to the Mwave manager. Both the drivers and the Mwave manager execute on the host CPU under Windows or OS/2.

The Mwave manager communicates these tasks to the Mwave OS by writing directly to the DSP. The Mwave OS, running on the DSP itself, is a real-time multitasking operating system that handles task scheduling to optimize DSP resources. "Real-time" means that the OS is able to guarantee a deterministic response once it receives a task. This is needed to avoid unexpected pauses in task processing; these pauses can cause pops and dropouts during audio playback, data loss during fax reception, and make a MIDI instrument difficult to play in rhythm.

The Mwave OS allows multiple tasks to execute simultaneously on the DSP. For example, a fax could be received while the user listens to CD audio. Mwave breaks each task into small parts, or *frames*, and schedules the DSP to alternate between frames from each task. To ensure real-time response, the Mwave OS understands the amount of DSP resources needed by each task and schedules the frames accordingly. In our example, the fax reception requires more DSP performance than the audio playback, so the OS will give larger frames to the fax than to the audio. Any non-real-time tasks, such as image decompression, are scheduled in the bandwidth remaining after handling all necessary real-time frames. The OS uses the M500's cycle-count register to monitor the execution of each frame, terminating the task if it exceeds its assigned cycle limit.

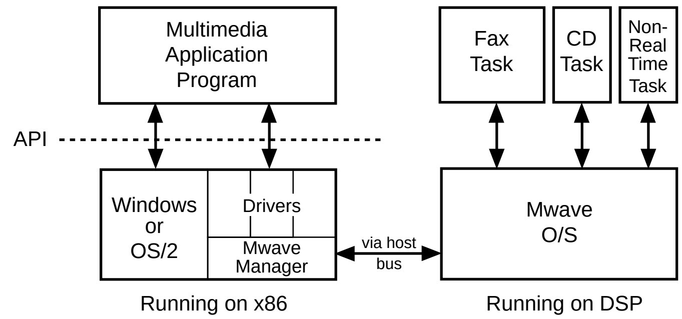

Figure 2. Mwave software components.

To avoid overloading the DSP, the OS does not accept a new task if it requires more than the available DSP resources. If, in the above example, the user attempts to transmit via modem, the OS will detect that there is not enough DSP performance (or other resources) for all three tasks, and inform the application (via the Mwave manager) that the DSP is busy.

The requested tasks are executed on the DSP under the control of the OS. Because the Mwave OS cannot access the host I/O system, the Mwave manager must ensure that all needed data is loaded into host memory before dispatching a task to the DSP. The Mwave manager uses the host OS to perform disk I/O.

Dueling Standards

Despite some weaknesses in the DSP core processor, TI and IBM have done an impressive job in creating the hardware and software required for a multimedia PC. In order for DSPs to become widely useful in a personal computer, there must be a standard way for any application to use the DSP. Without such a standard, a profusion of incompatible software and hardware will eventually frustrate users. A widely accepted API simplifies the programmer's job while allowing for innovation by multiple vendors at the hardware level. One needs only to look at the recent flurry of improvements in PC graphics to see the benefits of a standard API. Use of an API is critical for flexibility; a register-level standard (such as Sound Blaster) makes it harder for hardware vendors to offer new features without violating the standard.

Despite its technological merit, Mwave is not yet a true standard. IBM has declared Mwave as the standard multimedia API for OS/2, but IBM doesn't have the same influence over the more popular Windows, and Microsoft has not yet chosen an API. Both Microsoft and major system vendors are probably concerned about getting such an important technology from their biggest competitor, particularly when there are other alternatives.

AT&T has a similar system built around its VCOS operating system and its 3210 DSP chip. The 3210 archi-

Price and Availability

The TMS320M500 and TMS320M520 DSPs are currently sampling with volume production expected next March. In a 240-pin PQFP, the M500 is priced at \$25–\$30. In a 144-pin PQFP, the M520 is priced at \$30–\$35. Both prices are for "large" quantities.

The Personal Audio board, containing an M520, will be sampling in December with production volumes by March. The board costs \$99 to OEMs in quantities of 100,000. The Office Pro board, using the M500, will sample next January with production volumes by April. The Office Pro is priced at \$235 in quantities of 50,000. In the meantime, a software developers kit (including the Mwave drivers, manager, real-time OS, and a board similar to the Office Pro) is now available for \$995.

Contact Texas Instruments' Semiconductor Group SC-92089, Literature Response Center, PO Box 172228, Denver, CO 80217; 800/477-8924.

tecture has some advantages over the M500: it uses floating-point rather than fixed-point arithmetic; it has a full 32-bit address space, making it easier to share memory with the host CPU; and it has byte addressing, simplifying the use of C-language programs. Because it doesn't offer the same system integration as the M5x0 DSPs, 3210 subsystems are more expensive to build, and VCOS does not currently support any alternate DSPs. VCOS has built a significant base of applications, and AT&T's NCR subsidiary is planning to use the 3210 in all of its future multimedia PCs.

Analog Devices claims to have the most open DSP platform with its "Signal Computing" framework. Analog Devices, taking a different direction than AT&T or TI/IBM, is getting much of the required technology from various third parties and allowing for multiple-sourcing at all levels. For example, Spectron's microSPOX OS provides real-time task scheduling for the DSP and runs on chips from Analog Devices, Motorola, and TI. SynchoMedia, from Audiofile, provides a Windows API for sound applications, while Digianswer A/S (a Danish company) has software that provides an API for telephonic functions.

Although Analog's approach does appear more open, one drawback is that no single entity is putting together a complete package. The Mwave API offers a comprehensive set of DSP functions, but Analog is at the mercy of various third parties to provide equivalent functions. No major system vendors have publicly endorsed Analog's initiative, although both Olivetti and Compaq (who have partnered with Analog on other projects) are rumored to be working on it.

Of the major DSP vendors, Motorola seems to be the furthest behind in establishing a standard API. The company demonstrated several multimedia functions running on systems with a 56002 DSP, but they were not third-party applications. Motorola was not willing to comment on its overall software strategy.

In the meantime, hardware vendors who choose Mwave or VCOS lock themselves into using a single DSP chip from a single vendor. Both IBM and AT&T also happen to be major players in the system business. Analog is less threatening, but its solution has yet to catch on in the market. Software vendors who support any of these standards may have to later rewrite their applications to a new standard in the future. Large application developers may expend the resources to support multiple APIs, but smaller software houses cannot. Many software vendors may forego adding multimedia features to their programs until the standards issue is resolved, or they may try to implement these features using the host CPU instead.

This confusion is likely to slow the penetration of DSPs into personal computers until a single standard is achieved, which could be done in several ways. Microsoft could select a standard, or the major DSP vendors could agree on one, either by working together or through a multi-vendor organization such as the IMA (Interactive Multimedia Association). If none of these routes is taken, one of the dueling standards may eventually prevail, but it could be a long and bloody battle until that happens. ♦

