*By Jamil Chaoui,
OMAP Program Manager,
Texas Instruments.*

OMAP™ : Enabling Multimedia Applications in Third Generation (3G) Wireless Terminals

This paper describes how the Open Multimedia Applications Platform™ software and hardware architecture enables multimedia applications in third-generation (3G) wireless appliances. It provides an overview of OMAP™ strategy, concepts, main milestones, and achievements. It also describes OMAP hardware architecture, explains how multimedia applications can benefit from this advanced architecture, and why a RISC/DSP approach is superior to RISC-only architecture. This paper outlines OMAP software concepts to show how an architecture that combines two heterogeneous processors (RISC and DSP), several operating system (OS) combinations, and applications running on both DSP and RISC can be made accessible seamlessly to third parties. The final parts of the paper outline how key multimedia applications such as speech and video can be integrated in an OMAP solution.

INTRODUCTION

Every wireless handset contains two fundamental components: a modem and an applications processing and delivery system. The modem communicates with a network to send and receive data via an air interface. The applications component provides the functions that the user wants in a multimedia appliance. These may include speech, audio playback, image reproduction and streaming video, fax transmission and reception, e-mail, Internet connections, games, personal organizer functions, and a host of other possibilities. The applications component also handles user interface functions such as speech recognition (voice commands), keyboard, and written character recognition. Except for user interface functions, all of these applications depend on the air interface, which provides only limited bandwidth.

Managing the volume of data required for applications involving speech, audio, image, and video within the available bandwidth requires heavy signal compression before transmission and decoding or expansion within the receiver, which, for purposes of this paper, is a handheld wireless appliance. Wireless appliances need additional signal processing capabilities for the concurrent noise suppression and echo cancellation algorithms essential for even the most basic functionality, i.e., voice telephony.

In many of today's second generation (2G) wireless handsets, the modem component employs a DSP, which provides the air interface and takes care of such essentials as noise suppression and echo cancellation. The applications component, however, relies on a general purpose processor (GPP), usually a RISC processor, which attempts to provide the signal processing needed at the application level, user interfaces, and overall command and control functions.

In 2G appliances, which are essentially, wireless telephones with extremely limited data features, this processing dichotomy has generally proved to be satisfactory. However, for 2.5G and 3G appliances, which offer such capabilities as full-motion video and high-fidelity audio playback, the currently popular division of tasks between the two processors does not provide reliably robust performance and still maintain the useful battery life that is acceptable to consumers.

Next generation appliances, which include numerous applications enabled by high data rates and real-time signal processing, continue to require both a DSP and a GPP. However, the functional wall between the two basic appliance components breaks down as the demand for signal processing within the applications component increases. The DSP, which is optimized for intensive signal processing in real time with extremely low power consumption, becomes the primary processor for both the modem and the applications component. The GPP plays a secondary role, taking care of system management, command, control, and certain user interface activities. The OMAP architecture provides a means to coordinate dual processors across the two basic components of the wireless appliance and to seamlessly take advantage of the unique capabilities of each. Nokia, Ericsson, Sony, Handspring, and others already have selected the OMAP hardware and software architecture as the development platform for their wireless appliances. The first samples of an OMAP-based chipset will be available in 4Q00.

In addition to enabling the numerous, media-rich applications made possible by 2.5G and 3G data rates, the OMAP architecture also provides a new capability-the capability to dynamically download applications and application upgrades from the web in the same way that PC users download applications from the Internet today. This dynamic environment requires the programmability offered by DSPs and the application programming interface (API) features built into the OMAP architecture.

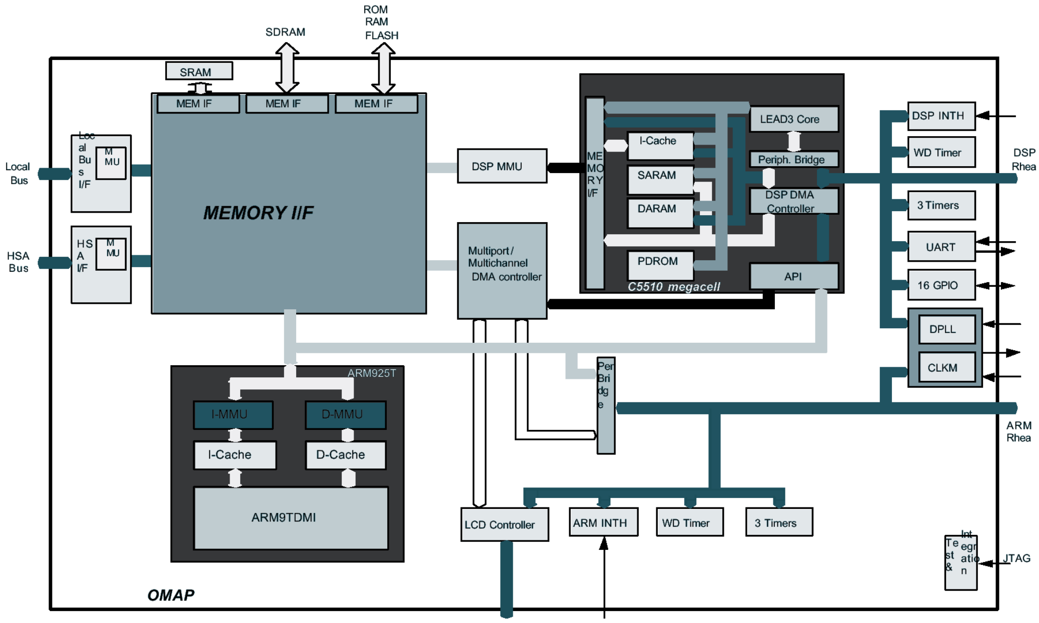


Figure 1. OMAP Hardware Architecture.

OMAP HARDWARE ARCHITECTURE

1. Advantages of a Combined RISC/DSP Architecture

The OMAP architecture is based on a combination of TI's state-of-the-art TMS320C55x™ DSP core and high performance ARM925T CPU. A RISC architecture, like ARM925T, is well suited for control type code (Operating System (OS), User Interface, OS applications). A DSP is best suited for signal processing applications, such as MPEG4 video, speech recognition, and audio playback. The OMAP architecture combines two processors to gain maximum benefits from each.

TI conducted a comparative benchmarking study, based on published data, which shows that a typical signal processing task executed on the latest RISC machine (StrongARM™, ARM9E™) requires three times as many cycles as the same task requires on a C55x™ DSP. See Table 1.

In terms of power consumption, tests show that a given signal-processing task executed on such a RISC engine consumes more than twice the power required to execute the same task on a C55x DSP architecture. Battery life, critical for mobile applications, therefore, is much greater when such tasks are executed on a DSP. The OMAP architecture's use of two processors provides this kind of power consumption benefits. At the same time, it allows the DSP to gain support from the RISC processor.

For instance, a single C55x DSP can process, in real-time, a full videoconferencing application (audio and video at 15 images/sec.), using only 40 percent of the available computational capability. Therefore, 60 percent of the capacity can be employed to run other applications concurrently. At the same time, in the OMAP dual-core architecture, the ARM processor stands ready to handle any other application requirements or can be suspended, thus saving battery life. As a result, the mobile user can enjoy access to popular OS applications (Word™, Excel™, etc.) while also engaging a videoconferencing application.

	ARM9E	StrongARM 1100	TMS320C5510	Units
Echo cancellation 16 bits (32 ms - 8 kHz)	24	39	4	Mcycles/s
Echo cancellation 32 bits (32 ms - 8 kHz)	37	41	15	Mcycles/s
MPEG4/H263 decoding QCIF @ 15 fps	33	34	17	Mcycles/s
MPEG4/H263 encoding QCIF @ 15 fps	179	153	41	Mcycles/s
JPEG decoding (QCIF)	2.1	2.06	1.2	Mcycles/s
MP3 decoding	19	20	17	Mcycles/s
Average cycle ratio with C5510™	3.1	3	1	

Table 1. Comparative Algorithm Execution.

Function of ” Copr ” Opcodes	
copr ()	Qualify instruction
Copr(k6)	Qualify instruction, pass k6 constant to MME control interface
Smem=ACx, copr()	Qualify instruction, write accumulator to memory (16-bit)
Lmem=ACy, copr()	Qualify instruction, write accumulator to memory (32-bit)

Table 2. Description of the ” copr ” Opcodes.

With a RISC processor alone, all of the available computing capacity would be needed to execute only the videoconferencing application, and power consumption would be twice as great as would be the case when the C55x DSP handles that application. The mobile user would not be able to execute other applications simultaneously. And battery life would suffer dramatically.

2. How the Architecture Works

The OMAP hardware architecture, shown in Figure 1, is designed to maximize the overall system performance of a 3G terminal while minimizing power consumption. This is achieved through the use of a C55x DSP core and an ARM925T CPU. Both processors utilize an instruction cache to reduce the average access time to instruction memory and eliminate power-hungry external accesses. In addition, both cores have a memory management unit (MMU) for virtual-to-physical memory translation and task-to-task memory protection.

The OMAP core contains two external memory interfaces and one internal memory port. The external memory interfaces support direct connection to synchronous DRAMs at up to 100 MHz and to standard asynchronous memories, such as SRAM, FLASH, or burst FLASH devices. The latter interface is typically used for program storage. It can be configured as 16 or 32 bits wide. The internal memory port allows direct connection to on-chip memory, such as SRAM, and can be used for frequently-accessed data, such as critical OS routines or the liquid crystal display (LCD) frame buffer. This reduces the access time and eliminates costly external accesses. All three interfaces are completely independent and allow concurrent access from either processor or direct memory access (DMA) unit.

The OMAP core also contains numerous interfaces to connect to peripherals or external devices from either the DSP or GPP. To improve system efficiency, these interfaces also support DMA from each respective processor's DMA unit. The local bus interface is a high-speed, bidirectional, multimaster bus that can be used to connect to external peripherals or additional OMAP-based devices in a multicore product. Additionally, a high-speed access bus is available to allow an external device to share the main OMAP system memory (SDRAM, FLASH, internal memory). This interface provides an efficient mechanism for data communication and also allows the designer to reduce system cost by reducing the number of external memories required in the system.

To support common operating system requirements, the OMAP architecture includes several peripherals-timers, general purpose input/output interfaces (I/Os), a UART, and watchdog timers. These are the minimum peripherals required in the system; other peripherals can be added on the TI peripheral bus (TIPB) interfaces. A color LCD controller is also included to support a direct connection to the LCD panel. The ARM DMA engine contains a dedicated channel that is used to transfer data from the frame buffer to the LCD controller, where the frame buffer can be allocated in the SDRAM or internal SRAM.

3. C55x DSP and Multimedia Extensions

The C55x DSP offers a highly optimized architecture for wireless modem and vocoding applications execution. Corresponding code size and power consumption are also optimized at system level. These features also provide advantages to a wider range of applications, though there may be some trade-offs in performance or power consumption.

The flexible architecture of the TI DSP hardware core allows extension of the core functions for multimedia-specific operations. The C55x DSP family are the first DSPs with such core level multimedia-specific extensions, which facilitate the demands of the multimedia market for real-time, low-power processing of streaming video and audio. The software developer has access to the multimedia extensions using the copr() instruction as described in Table 2, using combinations of C55x arithmetic instructions to create the

DSP Multimedia Extension Data Flow Modes Available		
ACy	=	copr(k8, ACx, ACy)
ACy	=	copr(k8, ACx, ACy), Smem=ACz
ACy	=	copr(k8, ACx, ACy), dbl(Lmem)=ACz
ACy	=	copr(k8, ACx, Smem)
ACy	=	copr(k8, ACx, Xmem), Ymem=ACz
ACy	=	copr(k8, ACx, dbl(Lmem))
ACy	=	copr(k8, ACx, dbl(Xmem)), dbl(Ymem)=ACz
ACy	=	copr(k8, ACx, Xmem, Ymem)
ACx,ACy	=	copr(k8, ACx, ACy, Xmem, Ymem)
ACx	=	copr(k8, Ymem, Coef), mar(Xmem)
ACx	=	copr(k8, ACx, Ymem, Coef), mar(Xmem)
ACx,ACy	=	copr(k8, Xmem, Ymem, Coef)
ACx,ACy	=	copr(k8, ACy, Xmem, Ymem, Coef)
ACx,ACy	=	copr(k8, ACx, ACy, Xmem, Ymem, Coef)

Table 3. Data Flow Modes.

Multimedia (MM) Extension Type	Speed-up Factor
Motion estimation	x5.2
DCT/IDCT	x4.1
Pixel interpolation	x7.3

Table 4. Characteristics of Video Hardware Accelerators.

desired data flow.

Table 3 indicates all data flow modes that can be built using the combination of arithmetic instructions and the opcodes in Table 2.

One of the first application domains that will extend the functionality of wireless terminals is video processing. Motion estimation, discrete cosine transform (DCT) and its inverse function (IDCT), and pixel interpolation require the greatest number of cycles for a pure software implementation using the C55x DSP processor. Consequently, these three application domains are the first multimedia programming extensions that the C55x DSP supports. Table 4 summarizes the characteristics of the extensions.

Formats and Rates	mA @1.5 V (15C035)	mA @ 0.9 V (15C035)
QCIF, 10 fps	12	7
QCIF, 15 fps	19	11
QCIF, 30 fps	37	22
CIF, 10 fps	49	29
CIF, 15 fps	74	44

Table 5. MPEG4 Video Codec Power.

Using the extensions, the overall video-codec application mentioned earlier is twice as fast as a classic software implementation. By reducing cycle count, the DSP real-time operating frequency and, thus, the power consumption are also reduced.

Table 5 summarizes current consumption (at maximum and lowest possible supply voltage) of a C55x DSP video MPEG4 Coder/Decoder using multimedia extensions, for various image rates and formats.

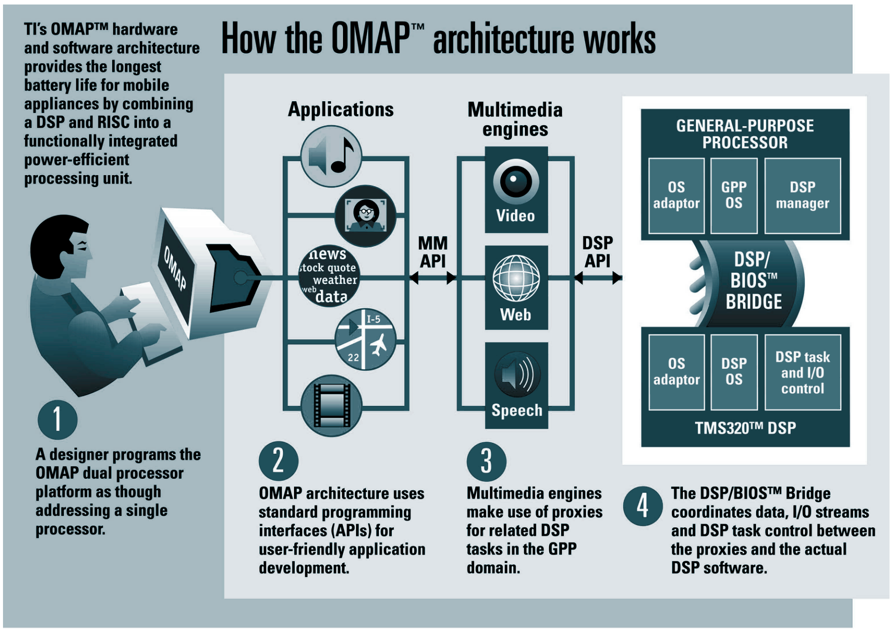
OMAP SOFTWARE ARCHITECTURE

The OMAP architecture includes an open software infrastructure that supports application development and provides a dynamic upgrade capability for heterogeneous multiprocessor system design. This infrastructure includes a framework for developing software that targets the system design and APIs for executing software on the target system.

The 2.5G and 3G wireless systems merge the classic voice-centric telephone phone model with the data functionality of the personal digital assistant (PDA). Non-voice multimedia applications (MPEG video, MP-3 audio, etc.) also will be downloaded to future phone platforms. These systems will also have to accommodate a variety of popular operating systems, such as WinCE™, EPOC™, and others. The dynamic, multitasking nature of these applications will require the use of operating systems on the DSP as well.

As a result, the OMAP architecture requires software that is sufficiently generic to allow easy adaptation and expansion for future technology. At the same time it must provide I/O and processing performance that approach the performance of a specific targeted architecture.

It is important to be able to abstract the implementation of the DSP software architecture from the GPP environment. The OMAP architecture accomplishes this by defining an interface scheme that allows the GPP to be the system master. This interface scheme is called the DSP/BIOS™ Bridge and consists of a set of


How the OMAP™ architecture works

TI's OMAP™ hardware and software architecture provides the longest battery life for mobile appliances by combining a DSP and RISC into a functionally integrated power-efficient processing unit.

1 A designer programs the OMAP dual processor platform as though addressing a single processor.

2 OMAP architecture uses standard programming interfaces (APIs) for user-friendly application development.

3 Multimedia engines make use of proxies for related DSP tasks in the GPP domain.

4 The DSP/BIOS™ Bridge coordinates data, I/O streams and DSP task control between the proxies and the actual DSP software.

Figure 2. How the OMAP Architecture Works.

APIs that includes device driver interfaces. See DSP API in Figure 2.

The most important function of the DSP/BIOS Bridge is providing communications between GPP applications and DSP tasks. The DSP/BIOS Bridge API is abstracted from the high-level application developers by a set of DLL and drivers that is provided in the development toolkit for the platform. This allows application developers to develop on the OMAP platform in the same manner as if they were developing on a single RISC processor. The environment provided for development allows the application developer to call the localized functions for video, audio,

speech, etc. and to develop in the traditional manner on platforms such as the PC. The high-level application developer does not require any awareness of the DSP or DSP/BIOS Bridge API.

The DLL and driver developers actively use the DSP/BIOS Bridge API to:

- Initiate and control tasks on the DSP
- Exchange messages with the DSP
- Stream data to and from the DSP
- Perform status queries

Although the design initially targets a limited set of OSs, the underlying architecture facilitates expansion of this list in the future. Standardization and reuse of existing API and application software are the main goals for the open platform architecture, thus allowing extensive reuse of previously developed software and a faster time to market of new software products.

OMAP MULTIMEDIA APPLICATIONS

1. Video

Video applications include two-way videophone communication and one-way decoding or encoding, which may be used for entertainment, surveillance, or video messaging. While second-generation communicators support speech only, coded at 8 to 13 kbps, even low-motion video on a small display requires at least 20 kbps. 3G wireless standards will make possible the higher bit rates required for more sophisticated video-related applications.

Compressed video is particularly sensitive to errors that can occur with wireless transmission. To achieve high compression ratios, variable-length code words are used, and motion is modeled by copying blocks from one frame to the next. When errors occur, the decoder loses synchronization, and errors propagate from frame to frame. The new MPEG-4 standard supports wireless video with special error resilience features, such as added resynchronization markers and redundant header information. The MPEG-4 data-partitioning tool, originally proposed by TI, puts the most important data in the first partition of a video packet, which makes partial reconstruction possible for better error concealment.

TI's MPEG-4 video software for OMAP architecture is based on reference C software, which is converted to use ETSI C libraries and then ported to C55x DSP assembly code. The ETSI C libraries consist of routines representing all common DSP instructions. The ETSI routines perform the desired function and also evaluate processing cycles and check for saturation, etc. Thus, the ETSI C, commonly used for testing speech codecs, provides a tool for benchmarking and facilitates porting the C code to assembly.

As shown in Section 2.2, the video software runs efficiently on the OMAP architecture. The architecture is able to encode and decode in the same time QCIF (176 ´144 pixels) images at 15 frames per second. The CPU loading for simultaneous encoding and decoding represents only 15 percent of the total DSP capability. Therefore, 85 percent of the CPU is still available to other tasks, such as graphic enhancements, audio playback (MP3), or speech recognition.

The assembly encoder is under development and typically requires about three times as much processing as the decoder. The main processing bottlenecks are motion estimation, DCT, and IDCT. However, through tight coupling of hardware and software, the OMAP architecture improves the video encoding execution by a factor of two.

The OMAP architecture provides not only the computational resources, but also the data-transfer capability needed for video applications. One QCIF frame requires 38016 bytes, for chrominance components downsampled in 4:2:0 format, when transferring uncompressed data from a camera or to a display. The video decoder and encoder must access both the current frame and the previously decoded frame in order to perform the motion compensation and estimation, respectively. Frame rates of 10- to 15-frames per second must be supported for wireless applications.

Third-generation standards for wireless communication, along with the new MPEG-4 video standard, and new low-power platforms like the OMAP architecture will make possible many new video applications. It is quite probable that video applications will differentiate 2G and 3G devices, creating new markets and higher demand for wireless communicators.

2. Speech Applications

A typical embedded system has constraints of low power, small memory size, and little or no disk storage. Therefore, speech recognition algorithms designed for embedded systems, such as wireless phones, must minimize resource usage while providing acceptable recognition performance.

TI proposes a dynamic vocabulary speech recognizer that is split between the C55x DSP and the ARM processor. The computationally intensive, small-footprint speech recognition engine runs on the DSP, while the computationally non-intensive, larger footprint grammar, dictionary, and acoustic model generation components reside on the ARM processor. The interaction between the model generation and recognition modules is minimized and conducted via a hierarchy of APIs, as shown in Figure 3. The advantage of this architecture is that the application can handle new vocabularies in several (potentially unlimited) recognition contexts without pre-compilation or storage of the grammars and models.

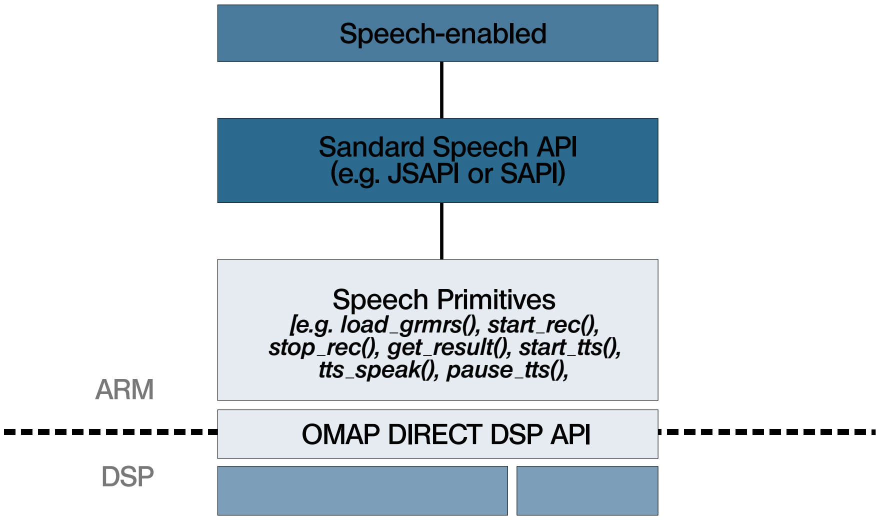

Speech-enabled
Sandard Speech API
(e.g. JSAPI or SAPI)
Speech Primitives
[e.g. load_grmrs(), start_rec(),
stop_rec(), get_result(), start_tts(),
tts_speak(), pause_tts(),
ARM
OMAP DIRECT DSP API
DSP

Figure 3. Hierarchical Organization of Speech APIs.

Note that only the unit selection and waveform generation modules of the Text-to-Speech (TTS) are on the DSP.

For each new recognition context, the grammars and acoustic models are generated

dynamically on the ARM processor and transferred to the recognizer on the DSP. This dynamic vocabulary capability of the speech recognizer is crucial on resource-constrained wireless devices. For example, a voice-enabled web browser on the phone can now handle several different web sites, each with its own different vocabulary, without compiling or storing vocabularies and speech grammars beforehand. Similarly, for a voice-enabled stock quote retrieval application, company names can be dynamically added to or removed from the stock portfolio. At any given time, the size of the active vocabulary is limited by the resource constraints on the DSP (e.g., size of the RAM available for the recognition search). However,

given that different vocabularies can be swapped in and out depending on the recognition context, the application can be designed to give the perception of an unlimited speech recognition vocabulary.

Similarly, the text-to-speech (TTS) system on the wireless device can be split between the ARM processor and the DSP. The text analysis and linguistic processing modules of the TTS reside on the ARM processor along with the phonetic databases. The unit selection and waveform generation modules reside on the DSP. As with the speech recognizer, the interaction between the ARM processor and DSP modules is minimized and conducted via a hierarchy of APIs.

CONCLUSION

This paper has described how the OMAP hardware and software architecture will enable multimedia applications in 3G wireless terminals. The OMAP multiprocessor architecture has been optimized to support heavy multimedia applications, such as video and speech in 3G terminals. Such a complex architecture, combining two heterogeneous processors (RISC and DSP), several OS combinations, and applications running on both the DSP and ARM can be made accessible seamlessly to application developers because of the DSP/BIOS Bridge feature. Moreover, this dual processor architecture is more cost efficient and power efficient than a single processor solution ■

Jamil Chaoui, Texas Instruments OMAP (Open Multimedia Application Platform) Program Manager, Multimedia Application Team Leader. Jamil Chaoui is a member of the technical staff in the European Wireless Application Group. He joined Texas Instruments in 1995. He has been initiating and leading the Waves project, implementing advanced speech processing applications for wireless customers. He is now program manager of OMAP (Open Multimedia Application Platform). He is also leading the OMAP multimedia application team.
Before joining TI, he was working for Alcatel Mobile Phones as a DSP software and system engineer.

- Open Multimedia Applications Platform, OMAP, TMS320C5510, TMS320C55x, C55x, DSP/BIOS are trademarks of Texas Instruments.
- ARM, StrongARM, ARM9E, ARM10 are trademarks of Advanced RISC Machines, Ltd.
- Word, Excel and WinCE are trademarks of Microsoft.
- EPOC is a trademark of Symbian, Ltd.
- All other trademarks are the property of their respective companies.