

Digital Signal Processors: Toward Vision 2001

Ronald E. Crochiere
James R. Boddie

The AT&T concept for the 21st century—*Vision 2001*—anticipates an environment in which communication is both easy and convenient, and in which every type of related service is ubiquitous and readily accessible. Today, the increasing sophistication of telephone equipment, modems, and computer multimedia is driving the need for high-volume implementation of application-specific digital signal processors (DSPs) having particular emphasis on low-voltage battery operation, moderate cost, faster time to market, software flexibility, and improved performance. This paper discusses these factors, and it also examines the engineering trends of modern DSPs as they relate to *Vision 2001*. Several basic hardware features are briefly reviewed, and the impact of advances in silicon-process technology on DSP architectures is considered. Evolving trends that affect the design of DSP software and algorithms are presented. The paper also provides some specific examples of DSP-based products within the *Vision 2001* environment.

Introduction

In 1979, AT&T Bell Laboratories developed the first full-featured, single-chip, programmable digital signal processor (DSP) for sampled data signals.¹ This device—initially designed for programmable digital filtering—was quickly adopted for a broad range of speech and communications processing applications. Through the 1980s, developers recognized the DSP's utility for implementing algorithms in single-chip form—algorithms that previously required the use of extra hardware in the form of several special-purpose boards.

Early applications of general-purpose DSPs include:

- Speech coding for AUDIX® and CONVERSANT® systems;
- Signaling tone-generation and detection;
- Speech synthesis for PBXs, switches, and trans-multiplexers;
- Filtering for transmission equipment; and
- Voice recognition and coding for PC and voice-response equipment.

By the early 1990s, the increasing

sophistication of telephone equipment, modems, and computer multimedia was driving the need for high-volume implementation of application-specific DSPs having particular emphasis on low-voltage battery operation, moderate cost, faster time to market, software flexibility, and improved performance. About this same time, AT&T created a concept for communications in the 21st century, which is generally referred to as *Vision 2001*. This concept envisions a communications environment in which access to people, machines, and information is both easy and convenient, and in which every conceivable type of communication and message service is both ubiquitous and readily accessible.

This paper discusses some of the basic trends in the design of today's DSPs as they relate to *Vision 2001*, as well as the key applications such DSPs serve. The first two sections briefly review both the essential hardware features of modern DSPs and the impact of advances of silicon process technology on DSP architectures. Some evolving

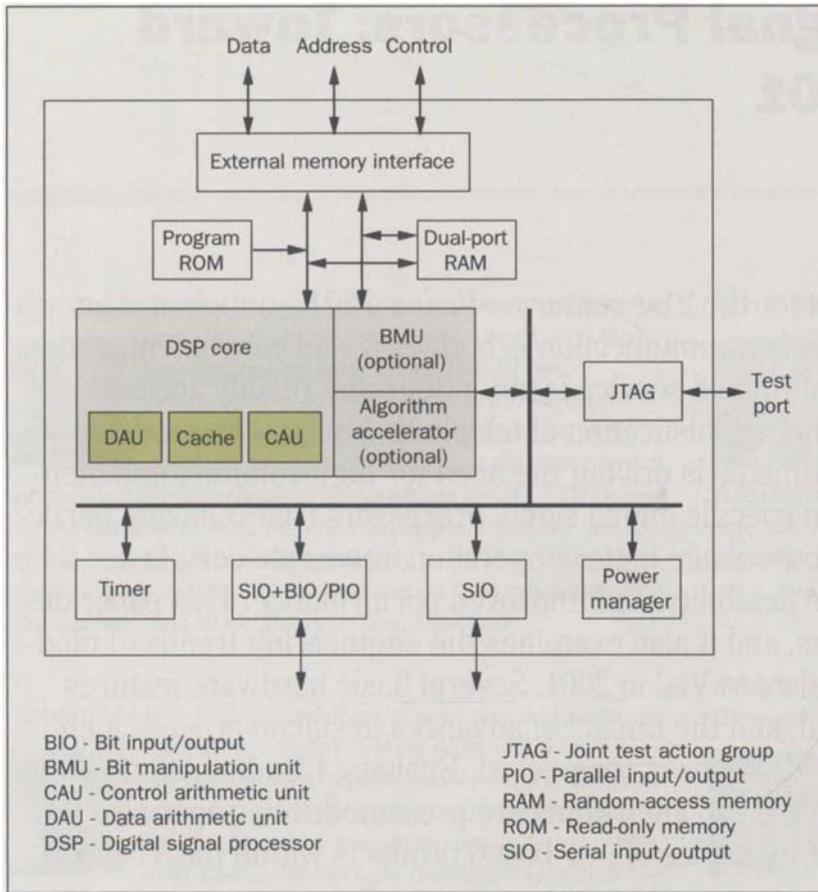


Figure 1. This illustration shows some of the basic features found in DSP architectures. DSPs are designed for high-speed, real-time numerical computation. They differ from conventional microprocessors or microcontrollers in that they are designed specifically for signal processing applications. They can process signals far more cost effectively—and with lower overhead and power consumption—than general-purpose processors. They also feature input/output (I/O) interfaces configured for such sampled data-processing systems as coders/decoders (codecs) and high-speed memory.

trends that affect the design of DSP software and algorithms are also discussed. In later sections, several basic hardware features are reviewed, and the impact of advances in silicon process technology on DSP architectures is considered. Evolving trends that affect the design of DSP software and algorithms are also presented. A discussion of some specific examples of DSP-based products and equipment concludes the paper.

DSP Hardware Architecture

DSPs are designed for high-speed, real-time numerical computation. They differ from conventional microprocessors or microcontrollers in that they are designed specifically for signal processing applications. They can process signals far more cost effectively—and with lower

overhead and power consumption—than general-purpose processors. They also feature input/output (I/O) interfaces configured for such sampled data-processing systems as coders/decoders (codecs) and high-speed memory devices or equipment.

Figure 1 illustrates some of the basic features found in DSP architectures. DSPs streamlined for low-cost applications tend to have less memory than high-performance devices, and they often lack certain features—for example, algorithm accelerators. High-performance DSPs, however, can have all the features of the streamlined devices and others, such as built-in analog-to-digital (A/D) and digital-to-analog (D/A) converters, customized utilization, and interfaces that reduce total system cost and improve the performance of a targeted application.

DSP Core. The central part of a DSP is its core, which contains two high-performance arithmetic units. The data arithmetic unit (DAU) is the main computational element. It is designed for efficient numeric processing, and it is usually pipelined to perform a repetitive sequence—for example, sums of products—in the form $A = A + X * Y$. With this capability, a DSP can perform—in a single instruction cycle—one tap of a digital filter or one multiply-and-accumulate sequence in a vector or matrix calculation. This operation typically forms the kernel of a large class of DSP functions necessary for voice and audio applications.

The computational speed of DSPs is usually measured in millions of instructions per second (MIPS). Typical speeds range from 20 MIPS for low-voltage, low-power devices to more than 60 MIPS for high-performance DSPs. A high-speed cache memory can be used in conjunction with the DAU to accelerate the processing of tight, repetitive, inner-loop calculations. The arithmetic precision of a DSP is determined both by the number of bits in the DAU's word length and by the choice of either fixed- or floating-point arithmetic formats. AT&T currently supports DSP families for the DAU having both 16- and 20-bit fixed-point formats and 32-bit floating-point formats.

The control arithmetic unit (CAU) performs the tasks of both generating and post-modifying addresses, either for accessing *operands* in memory or storing DAU

Panel 1. Abbreviations, Acronyms, and Terms

A/D—analogue to digital
API—application programming interface
ARAM—audio RAM
AT—Hayes-compatible command set
BIO—bit input/output
BMU—bit manipulation unit
CAD—computer-aided design
CAU—control arithmetic unit
CELP—code-excited linear prediction
codec—coder/decoder
D/A—digital to analogue
DAU—data arithmetic unit
DSP—digital signal processor
DTMF—dual-tone multifrequency
EECP—error-correction co-processor
EMI—external memory interface
flash memory—electrically erasable ROM
GSM—Global System for Mobile Communications
(previously Groupe Speciale Mobile)

HIF—host interface
I/O—input/output
JPEG—Joint Photographic Experts Group
JTAG—Joint Test Action Group
LPC—linear predictive coder
MIDI—musical instrument digital interface
MIPS—millions of instructions per second
MML—multimedia module library
MPEG—Moving Picture Experts Group
PBX—private branch exchange
PIO—parallel input/output
RAM—random-access memory
RF—radio frequency
ROM—read-only memory
RPE-LPC—regular pulse-excitation/LPC
SIO—serial input/output
VCOS—Visible Caching Operating System®

results. The CAU can also be used to perform microprocessor-type operations for logic and control operations.

Memory. Two types of memory are typically incorporated into the design of DSPs: *read-only memory* (ROM) for non-volatile storage of program instructions, and *random-access memory* (RAM) for storage of algorithm-state variables, computed data, or control information. RAM is also used for applications in which programs are downloaded or cache loaded into the DSP.

For high-volume, cost-sensitive applications, DSPs are often customized. They are designed having only the minimum required RAM and ROM, as well as the necessary I/O interfaces for a given application. Such a design reduces both the DSP's silicon area and device cost. Memory sizes range from 512 to 8,192 words of RAM and from 4,096 to 32,767 words of ROM.

I/O Interfaces and Test Ports. Four key types of I/O or external communication ports, as well as a variety of bit-I/O and control flags, are typically found in DSPs. Serial input/output (SIO) is normally used for direct interface to such devices as codecs, time-division multiplexed lines, or other DSPs in which a serial data-flow of

sampled data signals is required. Parallel I/O (PIO) is used where bidirectional communication is required between a DSP and other processors, such as a host microprocessor or bus. An external memory interface (EMI) is used where communication with external memory is required, such as dynamic RAM (DRAM) or static RAM (SRAM). The Joint Test Action Group (JTAG) test port supports an industry-standard bus for boundary-scan testing and uploading or downloading of internal RAM.

Accelerators. In addition to multiply-accumulate and addressing computations, signal processing algorithms sometimes require other types of high-intensity kernel operations not directly suited to the DAU or CAU functions. An important example is that of bit manipulations for such operations as scaling, bit packing, and barrel shifting. A bit manipulation unit (BMU) extends the basic DSP instruction set to perform these types of operations for signal processing algorithms.

Power Management. For power-sensitive applications, a power-control manager is added to a DSP to deactivate unneeded on-chip peripherals, thereby reducing power consumption. The power manager can also provide

several levels of power-down or sleep modes when a system is on standby. Such a provision—of course—saves power, which is particularly important for battery-operated equipment. DSPs designed for low-voltage operation and low power consumption currently require fewer than 3 mW per MIP at 2.7 V.

Hardware Trends and Directions

This section discusses some evolving trends that affect DSP technology as it applies to silicon processing, design modularization and core-based design, and flash technology for rapid prototyping.

Silicon Process Technology. The rate of change in silicon process technology has dramatically affected the capability of DSPs. The minimum circuit-feature size that can be imprinted on a silicon chip has been dropping exponentially for the past three decades.² This trend, shown in Figure 2, illustrates that feature size has dropped by a factor of five from 1980 to 1995. It is expected to drop again by a factor of two by the year 2001. Transistor density—the number of transistors that can be imprinted on a chip—increases as the square of this feature size.

In 1979, transistor density became high enough to enable designers to develop the first DSP on a single chip, with the largest chip area being dominated by the DAU and CAU. By 2001, the equivalent transistor area will be reduced by a factor of 100, and the core will occupy only a small fraction of the entire chip.

This size reduction will enable the integration of many more combinations of RAM, ROM, and interfaces. In effect, the chip is becoming a platform for a whole subsystem in which the DSP is only a small part of the chip's total function.

The reduction in feature size not only has facilitated the integration of more transistors in the same silicon area. For many applications, it has also increased transistor speed and reduced the voltage supply requirements from a standard of 5 V to 3 V, and from 5 V to 2.7 V for battery-operated equipment. Today's high-performance DSPs can deliver 50 MIPS of processing power with a 5-V supply, 30 MIPS at 3 V, and 26 MIPS at 2.7 V.³

Design Modularization and Core-Based Design. Market demand for DSPs is increasing, and the need to develop specialized devices quickly for different applications has led to a modular design approach. Using modular design,

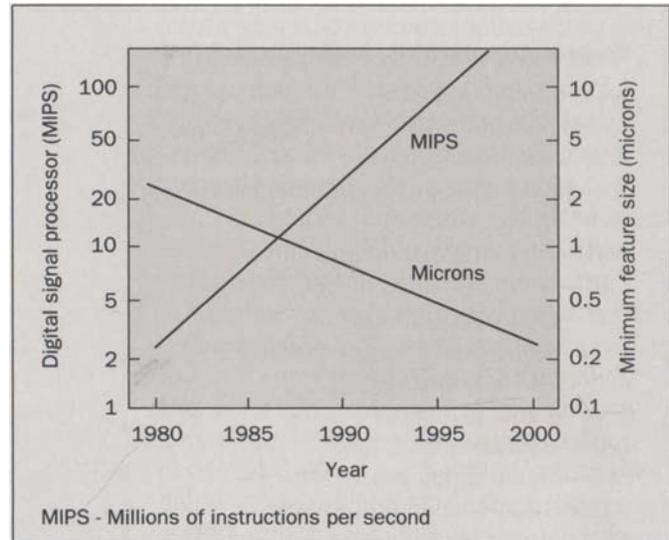


Figure 2. The rate of change in silicon process technology has dramatically affected the capability of DSPs. The minimum circuit-feature size that can be imprinted on a silicon chip has been dropping exponentially for the past three decades. This trend, shown in the graph, illustrates that feature size has dropped by a factor of five from 1980 to 1995. It is expected to drop again by a factor of two by the year 2001. Transistor density—the number of transistors that can be imprinted on a chip—increases as the square of this factor.

new applications can employ the same DSP core elements. Different RAM and ROM capacities are used, as well as different types of peripheral interfaces in the architecture. A modular design approach allows reusing core hardware modules and necessitates custom changes only in the memory configuration or peripherals. Several years are required to develop and build a new core. The customization approach—based on modules—significantly reduces the time to market for application-specific DSPs once the primary core modules are available.

Flash Technology for Rapid Prototyping. As mentioned earlier, modular design reduces hardware development time. Still, the time-to-market factor for updating new ROM codes is about eight weeks from new program creation and verification until the program is processed into a chip. A new technology advancement, based on electrically erasable ROM (flash memory), facilitates DSP field programming. This advancement eliminates the

long wait cycles for field prototypes. With flash technology, new DSP instructions can be downloaded into the FlashDSP™ chip in a matter of seconds. Thus, in developing initial field prototypes, the long lead time usually required for silicon processing is greatly reduced. The FlashDSP chip is pin compatible with the targeted, production DSP (ROM coded), and it can be tested in the same hardware as a production chip.

Software Architecture and Development

Control of a DSP is defined through its basic instruction set. Furthermore, DSPs are highly concurrent devices. Thus, they tend to have instructions with multiple fields that simultaneously control the operation of the different units in a DSP's architecture during the same instruction cycle. As a result, DSP assembly code is typically more difficult to read and develop than conventional microprocessor code or high-level language code, such as C or Fortran.

In spite of these difficulties, DSP programmers historically have opted to develop real-time DSP algorithms directly in DSP assembly code for two primary reasons:

- Efficiency in matching algorithm architecture directly to the underlying DSP architecture; and
- Maintenance of direct control over the hardware's architecture for real-time operations.

As DSP applications become more complex and programs grow larger, higher-level techniques for both software development and hardware design will become more important.

Instruction Set. A DSP instruction set defines the lowest level of software that directly controls the basic functions of the device. A typical DSP instruction contains a field of multiple sub-instructions that operate in parallel. For example, a typical instruction for the AT&T DSP1600 family is:

$a0 = a0 + p$ $p = x * y$ $y = *r0++$ $x = *pt++$

In this instruction, the first field defines an *accumulate* of $a0$ and p . The second field defines a *multiply*. The third and fourth fields define *memory fetches*, plus the auto-increment of *register values*.

The foregoing operation consumes two instruction cycles if fetched from memory and only one cycle if fetched from the high-speed cache. Other DSP instructions include:

- Control operations to direct program flow;
- Directions to load the cache;
- Special functions to perform such operations as round-

- ing, negation, and arithmetic shifts; and
- BMU or accelerator control.

Software Modularization and Control. Software, like hardware, is often developed in a modularized and reusable fashion to shorten its time to market. Such high-complexity software modules as speech coders, speakerphone programs, and voice recognizers are typically developed as core modules. Then, they are customized or integrated into total applications through the control of an "executive" layer of code. This code layer integrates and personalizes some combination of modules for a particular application.

If only a single software module or task is active at one time, then control of the data flow and real-time operation of a DSP can be passed to the module. If more than one module or task must be processed simultaneously, then the executive software must allocate the time for each task, and some means of intercommunication between tasks must occur. For many tasks, some type of structure must be imposed on the software architecture, and this structure generally takes the form of an operating system. Structuring the software architecture is discussed later in the Multimedia subsection.

Software and Hardware Development Tools. A programmer can use a variety of tools to develop DSP code. Assemblers translate the user's assembly-language program into relocatable object code. Linkers then combine multiple sets of separately assembled code into single, executable files. Software simulators allow a user to simulate the operation of a DSP in a computer environment. The simulation can be "frozen" by setting certain conditions for program development and debugging. Hardware development systems and evaluation boards provide platforms on which to download and run DSP code in real time. They also interface to such devices as A/D and D/A converters for program development and real-time evaluation of an algorithm.

High-Level Language Support. Compilers translate high-level programs, such as C language code, into DSP assembly code. Historically, experienced DSP programmers have avoided using C compilers, which are inefficient for creating optimized DSP assembly code—particularly for fixed-point DSPs. This is especially true for production code that usually must be highly optimized and compressed in RAM, ROM, and MIPS.

As both code size and complexity continue to

increase, the use of compilers is expected to become more widespread for non-critical functions and control operations. In addition, system-level computer-aided design (CAD) and behavioral simulators can be more widely used for system modeling and board-level simulation. Thus, the integration of DSP tools and simulators with industry CAD tools is likely to become a trend.⁴ This approach supports a top-down methodology that separates algorithm, accuracy, and real-time issues, as well as hardware and software tradeoffs in the development cycle. Thus, a developer can maintain maximum control, observability, and repeatability in the design process.

DSP Applications and Market Segments

Several major applications are driving DSP architectures toward customized subsystems. In effect, the notion of a general-purpose DSP is giving way to families of application-specific DSPs that explicitly address the needs of each market segment. This section examines the trend by briefly reviewing these applications.

Consumer Telephony Applications. Today, DSPs are finding their way into both consumer and business telephones. DSPs can add significant new product features not possible with conventional analog technology. Figure 3 shows an example of either a digital answering machine or feature telephone using the AT&T DSP1605. A microcontroller manages the systems and communicates with the DSP executive software through a host interface (HIF) port. The DSP's SIO port is connected to an analog phone line or a speaker/microphone (acoustic side) through a codec. The DSP connects to audio random-access memory (ARAM) through its external memory interface (EMI) port for storage of digital messages. It also connects to a ROM for voice prompts, such as time and date stamps. For cost-reduced applications, a DSP can also assimilate the microcontroller's functions and eliminate the need for a separate microcontroller and HIF port. The AT&T DSP1604 offers this option.

The DSP shown in Figure 3 manages all communications to the ARAM, and it also provides the framework for message management, including memory allocation and status, adding and deleting messages, and playback. The DSP performs signal-processing functions for speech compression, voice-activity detection, variable-speed playback, dual-tone multifrequency (DTMF) generation and detection,

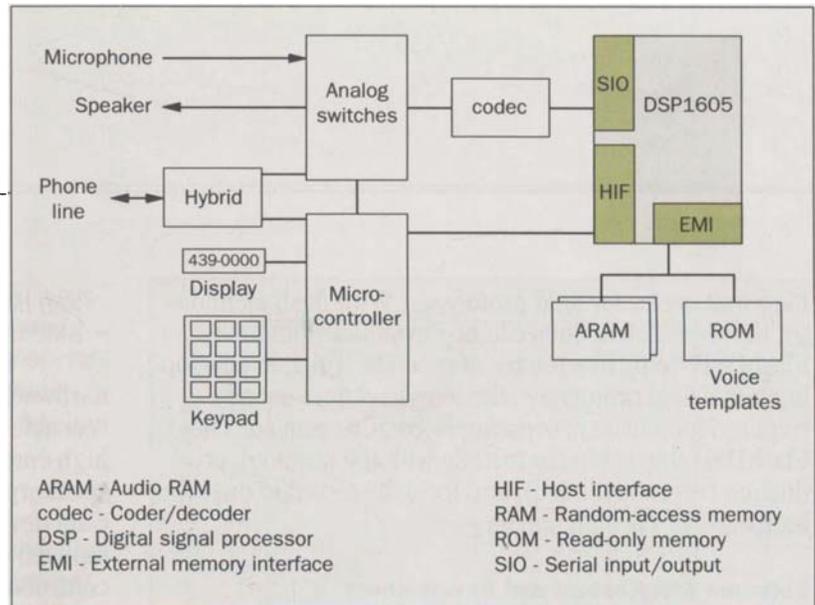


Figure 3. An example of either a digital answering machine or feature telephone using the AT&T DSP1605 is shown in this block diagram. A microcontroller manages the system and communicates with the DSP executive software through a host interface (HIF) port. The DSP's serial input/output (SIO) port is connected to an analog phone line or a speaker/microphone (acoustic side) through a codec. The DSP connects to audio random-access memory (ARAM) through its external memory interface (EMI) port for storage of digital messages. It also connects to ROM for voice prompts, such as time and date stamps. For cost-reduced applications, a DSP can also assimilate the microcontroller's functions and eliminate the need for a separate microcontroller and HIF port.

and signal routing, such as room monitoring. This architecture can support an average of 16 to 20 minutes of message storage in a single 4-Mbit ARAM by using a high-performance, code-excited linear-prediction (CELP) speech coder⁵ in combination with voice-activity detection. The architecture of Figure 3 can support an echo-canceling speakerphone with the addition of a second codec, thus providing more telephone functionality. Furthermore, the architecture—with the addition of an SRAM on the EMI port in a telephone—can support both voice recognition for hands-free control and dialing functions.

Wireless Telecommunications Applications. Wireless telecommunications equipment, such as hand-held terminals and base stations, represent yet another major application of DSPs. Figure 4 shows an example of a highly customized AT&T DSP1618 architecture for the European Global System for Mobile Communications (GSM) cellular terminal standard. The DSP communicates with a microcontroller that handles system-management functions. Additionally, the DSP communicates with a conversion processor chip that provides both the voice-band A/D and

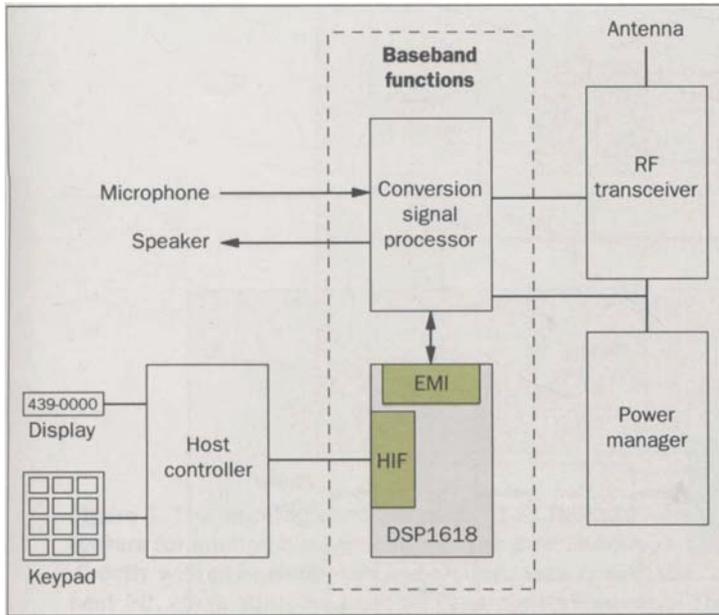


Figure 4. This drawing shows an example of a highly customized AT&T DSP1618 architecture for the European Global System for Mobile Communications (GSM) cellular terminal standard. The DSP communicates with a microcontroller that handles system-management functions. Additionally, the DSP communicates with a conversion processor chip that provides both the voice-band analog-to-digital (A/D) and digital-to-analog (D/A) conversions and analog connections to the speaker and microphone. The conversion processor also performs base-band A/D and D/A conversions, modulation functions, buffering, and communications to the radio interface. A transceiver provides the RF modulation and processing functions for the terminal.

D/A conversions and analog connections to the speaker and microphone. The conversion processor also performs base-band A/D and D/A conversions, modulation functions, buffering, and communications to the radio interface. Lastly, a transceiver provides the radio-frequency (RF) modulation and processing functions for the terminal.

In this configuration, the DSP performs the digital base-band processing functions for both the 13-kbits/s GSM speech-coding and channel processing. The DSP is customized for maximum processing performance and minimum power consumption, 3-V battery operation, low cost, and small size for hand-held operation. To achieve such design objectives, the DSP is configured with two on-chip algorithm accelerators—the error-correction coprocessor (EECP) and bit manipulation unit (BMU)—to expedite speech-coding operations and channel-processing functions. The speech coder is based on the regular pulse-excitation/linear predictive coder (RPE-LPC) algorithm⁶, and the channel processing functions include such operations as Viterbi processing, convolutional decoding, and burst interleaving.

Modem Applications. High-speed facsimile and data communications represent still another important application driving DSP technology.⁷ The target platforms include desktop and laptop computers. Figure 5 shows an example of a traditional DSP-based architecture that supports facsimile and modem data rates up to 19.2 kbits/s. This configuration uses a ROM-coded AT&T DSP16A to perform basic data-pump functions to convert signal formats between analog line signals and digital bit streams. These functions include echo cancellation, feedback equalization, and decision making. In addition, the interface chip provides the following four utilities:

- Microcontroller busing;
- Eye patterning;
- Line interfacing; and
- Clock generating.

The microcontroller handles communications to the host computer and performs data-processing functions, such as error correction and data compression. Communication with the host computer is accomplished by means of an industry-standard AT (Hayes-compatible) command set.

The architecture shown in Figure 5 is continuing to evolve in several ways. For computer applications having high-performance host CPUs (for example, a 486 host), a cost-reduced version of this architecture eliminates the need for a microcontroller, SRAM, and ROM. Such architecture integrates both the error-correction and data-compression functions into the host system. By adding MNP-10 control functions or MOBITEX protocol functions to the controller's software, the architecture can be adopted for either cellular-data or packet-radio applications. (MNP is a trademark of Microcom, Inc. and MOBITEX is a trademark of Televerket Corp.). A downloadable or "soft-modem" architecture results from replacing the DSP16A with an AT&T DSP1633, which can be field upgraded by downloading new software through the host computer. This architecture supports the new V.34 modem standards with data-pump rates up to 28.8 kbits/s.

In addition to data applications, DSPs can provide computer voice functions. A voice-through option in the DSP allows a computer to receive digitized speech and store it on disk for answering-machine functions. In addition, a special program in the DSP enables a computer to have a built-in speakerphone. This combination of computer control over telephone functions makes available

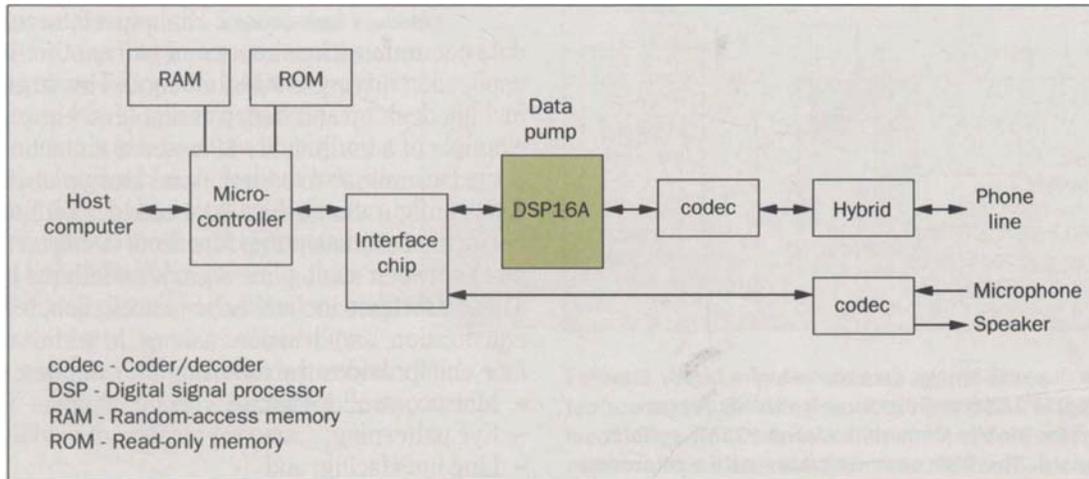


Figure 5. This diagram shows an example of a traditional DSP-based architecture that supports facsimile and modem data rates up to 19.2 kbits/s. The configuration uses a ROM-coded AT&T DSP16A to perform basic data-pump functions to convert signal formats between analog line signals and digital bit streams. These functions include echo cancel-

lation, feedback equalization, and decision making. The interface chip also provides four utilities. The microcontroller handles communications to the host computer and performs data-processing functions. Communication with the host computer is accomplished by means of an industry-standard AT (Hayes-compatible) command set.

powerful new options. For example, a user can simply point and click on a computer's telephone directory to initiate a phone call or even a teleconference. The next step beyond traditional voice and data functions is to combine both utilities on the same call. Such applications as alternate voice and facsimile, as well as concurrent "white board" applications, allow users to exchange graphics simultaneously while speaking.

Multimedia. To extend the "soft modem" concept one step further, multimedia architectures integrate the DSP function directly into that of a computer. Such architectures employ a DSP operating system to manage the interface between the computer and DSP.⁸ Figure 6 shows an example of an AT&T DSP3207 architecture for multimedia applications. The DSP interfaces directly with the system bus and shares memory with the host PC, eliminating the need for separate DSP memory. The DSP is controlled through the Visible Caching Operating System[®] (VCOS), which resides jointly within both the DSP and host, providing the communications infrastructure between them. Software libraries are stored in host application programs and cache loaded onto the DSP through the VCOS operat-

ing system as needed. The VCOS system manages the DSP loading for real-time operation and provides a programmable signal-routing infrastructure between tasks. It furnishes a flexible method by which to manage concurrency and multitasking of multiple processes with the communications between DSP and host. The VCOS system also provides the means for either the addition or deletion of new DSP processes while others are continuously running.

The lower levels of the VCOS system communicate with the DSP and the upper levels communicate with the host application through the application programming interface (API). A compendium of DSP software modules—the VCOS system multimedia module library (MML)—provides the functions a DSP can perform. New modules can be developed and added to this library as needed. The MML provides the following rich set of signal processing capabilities for a computer:

- Standard modem functions for data communications;
- Speech coding and DTMF functions for telecommunications applications;
- Answering-machine and voice-messaging functions;
- Low-delay CELP coding (G.728) and wideband coding

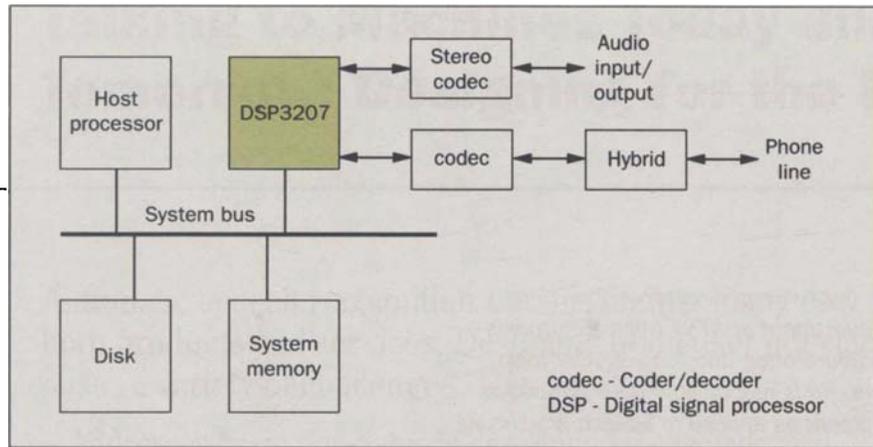


Figure 6. This drawing exemplifies an AT&T DSP3207 architecture for multimedia applications. The DSP interfaces directly with the system bus and shares memory with the host PC, eliminating the need for separate DSP memory. The DSP is controlled through the Visible Caching Operating System[®] (VCOS), which resides jointly within both the DSP and host, providing the communications infrastructure between them. Software libraries are stored in host application programs and cache loaded onto the DSP through the VCOS operating system as needed. The VCOS system manages the DSP loading for real-time operation and provides a programmable signal-routing infrastructure between tasks. It furnishes a flexible method by which to manage concurrency and multitasking of multiple processes with the communications between DSP and host. The VCOS system also provides the means for either the addition or deletion of new DSP processes while others are continuously running.

- (G.722) for audio processing in video conferencing;
- Joint Photographic Experts Group (JPEG) image compression and Moving Picture Experts Group (MPEG) I and II audio coding for multimedia presentations; and
 - Musical instrument digital interface (MIDI) music synthesis and sound-blaster audio for presentations and games.

By combining these functions, product developers can create new applications (without the need to be expert DSP programmers) through calls to the VCOS system's MML.

Conclusion

Trends in the use of DSPs can be applied to the broad goal of Vision 2001, which is to provide seamless, easy-to-use, high quality, and affordable communications between people and machines—anywhere and any time. The basic architectures, as well as issues and trends in silicon technology and software, form the basis of how DSPs are being used in today's communications environment.

The combination of maturing, current markets and the emergence of new opportunities are driving voice, audio, and data architectures toward application-specific

DSPs. As a result, highly integrated subsystems are being developed in which DSPs form the modular hardware core. Working together with the software modules that are either ROM coded or downloaded, DSPs can be personalized for many communications or messaging-service applications.

Such modular-based hardware and software—when combined with rapid delivery to market—form a platform on which key voice, audio, and data technologies are integrated. Application-specific DSPs will continue to play a crucial role in the development of this platform, as well as in the realization of the Vision 2001 concept.

References

1. *Bell System Technical Journal*, Vol. 60, No. 7, Part 2, September 1981, (special issue on DSPs).
2. G. Moore, "VLSI: Some Fundamental Challenges," *IEEE Spectrum*, Vol. 16, No. 4, April 1979, pp. 30-37.
3. R. B. DeRobertis and R. Fahey, "Digital Cells Involved in Power Struggle," *Electronic Engineering Times*, Issue 785, February 21, 1994, pp. 69, 72.
4. P. Wilhelmson, E. Hass, R. Subramanian, T. Gentles, R. Fahey, B. Schlieder, and J. Kunkel, "Simulating a Programmable DSP at the System Level," *DSP Applications Magazine*, November 19, 1993, pp. 6-22.
5. P. Kroon and K. Swaminathan, "A High-Quality Multirate Real-Time CELP Coder," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 5, June 1992, pp. 850-857.
6. P. Kroon, E. F. Deprettere, and R. J. Sluyter, "Regular-Pulse Excitation: A Novel Approach to Effective and Efficient Multipulse Coding of Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 5, October 1986, pp. 1054-1063.
7. *DSP Review*, Vol. 1, No. 1, AT&T Microelectronics, Allentown, Pennsylvania, Spring 1991 (special issue on modems).
8. J. F. Lynch and N. Mera, "Signal Processing for Multimedia" (a discussion of the AT&T Visible Caching Operating System; includes related article on a real-world application), *Byte Magazine*, Vol. 17, No. 2, February 1992, pp. 105-108.

(Manuscript approved February 1995)

Ronald E. Crochiere is engineering director in the DSP



Systems Department at AT&T Microelectronics—AT&T Bell Laboratories division—in Allentown, Pennsylvania. He is responsible for DSP subsystem development as applied to speech processing and consumer telephony. Mr. Crochiere—an IEEE

Fellow and recipient of the AT&T Harold D. Arnold Award—holds a B.S. degree from the Milwaukee School of Engineering in Wisconsin, as well as M.S. and Ph.D. degrees from the Massachusetts Institute of Technology in Cambridge, all in electrical engineering. He joined AT&T in 1974.

James R. Boddie is research and development director in the



Wireless and Messaging Business Unit at AT&T Microelectronics—AT&T Bell Laboratories division—in Allentown, Pennsylvania. He is responsible for the development of DSP-based products, including analog and digital integrated circuits, firmware, and development tools. Mr. Boddie holds B.S. and

Ph.D. degrees from Auburn University in Alabama, as well as M.S. and E.E. degrees from the Massachusetts Institute of Technology in Cambridge, all in electrical engineering. He joined AT&T in 1977.
