

## ABSTRACT

Contemporary wireless communications are based on digital communications technologies. The recent commercial success of mobile cellular communications has been enabled in part by successful designs of digital signal processors with appropriate on-chip memories and specialized accelerators for digital transceiver operations. This article provides an overview of fixed point digital signal processors and ways in which they are used in cellular communications. Directions for future wireless-focused DSP technology developments are discussed.

# Digital Signal Processors in Cellular Radio Communications

Zoran Kostic, AT&T Laboratories-Research

Selvaraj Seetharaman, GlobeSpan Technologies Inc.

**C**urrent trends suggest that the wireless communications industry will continue to grow at an impressive rate. Today, the dominant application of wireless technologies is speech communications. A number of cordless, cellular, and personal communications services (PCS) standards have been developed and at least half a dozen of them are successful such as Global System for Mobile Communications (GSM), IS-136, IS-95, PDC, Digital European Cellular Telecommunications (DECT), and Personal Handy Phone System (PHS). The work on other commercial radio applications, such as wireless local loops, wireless local area networks (WLANs), wireless asynchronous transfer mode (W-ATM), and satellite communications, is being pursued at full speed in laboratories at many companies and universities. Tables 1 and 2 show various wireless applications and standards. Although different wireless applications have many commonalities, each is sufficiently special to require different architectures and implementations. This article takes the example of second-generation mobile wireless systems and discusses the use of digital signal processors (DSPs) in their realization. Also, the article presents a view on the future directions in the design and application of DSPs for third-generation digital wireless systems.

Worldwide deployment of cellular mobile systems and provision of services to large numbers of consumers requires that cost, size, and power consumption of end-user terminals and base stations meet the aggressive expectations of service providers and users. Advances in integrated circuit (IC) technology are among the crucial contributions which enabled the widespread commercialization of wireless communications. Key aspects of very large-scale integration (VLSI) technological development that concern wireless system designers are fundamental advances in semiconductor processes, digital signal processing, and radio frequency circuits.

Second-generation cellular mobile systems are based on digital communications concepts. The number of functions which have to be implemented in digital transceivers and their complexity require that a programmable VLSI device be avail-

| Type of application  | Wide area<br>(high mobility)                                                                        | Local area<br>(low mobility)                                        |
|----------------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| Voice communications | Cellular phones<br>Satellite systems                                                                | Cordless telephones<br>Wireless local loop systems<br>Wireless PBXs |
| Data communications  | Cellular telephones<br>Pagers<br>Specialized packet data<br>(RAM, ARDIS, CDPD)<br>Satellite systems | Cordless telephones<br>Wireless LANs                                |

■ Table 1. Wireless applications.

able. This role has been filled by digital signal processors (DSPs), a special variant of microcomputers. The use of DSPs in wireless has grown many-fold with the start of deployment of GSM, IS-136 and other cellular systems in the early 1990s. Today, tens of millions of DSPs are in use in mobile phones and base stations worldwide.

The development of digital wireless technologies has taken a multiversion/multistandard approach. Opportunities to realize business and intellectual property issues motivate companies to work on proprietary and/or different air interfaces. Although the European-developed GSM is the most successful cellular system, it has recently seen strong competition in emerging markets. The specification of third-generation wireless systems is being done under the umbrella of the International Telecommunication Union (ITU) and is named IMT-2000 (for International Mobile Telecommunication by the year 2000). Although the initial goal of the ITU work was to come up with a single worldwide standard for as many components of the system as possible, it seems that commercial forces are going to modify this goal, and several "compatible" air interface wireless standards will be adopted. The Japanese standardization body Association of Radio Industries and Businesses (ARIB), European Telecommunications Standards Institute (ETSI), and the U.S. Telecommunications Industry Association (TIA) are soon to receive numerous for-

mal contributions for future technologies from participating members. Regardless of these developments, the role of digital signal processors in the future of wireless systems is not in doubt.

This article is divided into seven sections. The second section is devoted to DSPs. The discussion is focused on fixed-point devices. DSP architectures and development tools are reviewed. A table of contemporary DSPs most suitable for wireless applications is presented. The third section is an overview of general issues in wireless systems. The fourth section treats the subject of wireless transceiver design. We focus on physical-layer functionality, that is, DSP implementations of transmit and receive algorithms as well as software/hardware partitioning issues. Digital speech coding is at the heart of second generation wireless systems and we devote the fifth section to this topic. The sixth section discusses future trends in DSP design and applications in wireless communications. The article ends with a summary.

## DIGITAL SIGNAL PROCESSORS

**P**rogrammable DSPs are specialized microcomputers, VLSI devices designed for implementation of extensive arithmetic computation and digital signal processing functions through downloadable or resident software/firmware [3]. Their hardware and instruction sets usually support real-time application constraints. Classical examples of signal processing functions are finite impulse response (FIR) filters and fast Fourier transforms (FFTs). DSPs are meant for performance, not for extensive functionality or programmer convenience.

Hardware and instruction sets of conventional microcontrollers and microprocessors do not include structures specially designed to execute the previously mentioned functions efficiently. Contemporary high-end microprocessors are expensive devices (hundreds of dollars) with significant power consumption (watts), most of which are computer designs targeted at multimedia computing. Microcontrollers target applications such as automobile electronics, and industrial and home appli-

ance control. On the other hand, DSPs suitable for wireless applications distinguish themselves particularly in low power consumption, low cost, and high number-crunching capabilities.

The following subsections summarize DSP architectures, memory structures, DSP software and development tools, and interfaces to DSPs. Detailed presentations on DSPs can be found in [3, 4, 6, 15, 20, 21]. A list of fixed-point DSPs suitable for wireless applications is given in Table 3.

### DSP ARCHITECTURES

The most basic architectural component of a DSP is the fast multiplier/accumulator (MAC) integrated into the data path, rather than placed on a coprocessor of a microcontroller. This makes possible an instruction cycle time equal to the cycle time of the arithmetic hardware. The multiply-accumulate instruction can be expressed as

$$a = a + b * c.$$

The instruction cycle time of contemporary DSPs is on the order of tens of nanoseconds. One of the benchmarks expressing the power of a particular DSP is the MAC time, which also represents the time to execute a single tap instruction in a FIR filter. The optimization of the MAC unit in DSPs leads to the fact that a typical DSP performs nonoptimally for functions such as conditional branching or a sequence of control instructions.

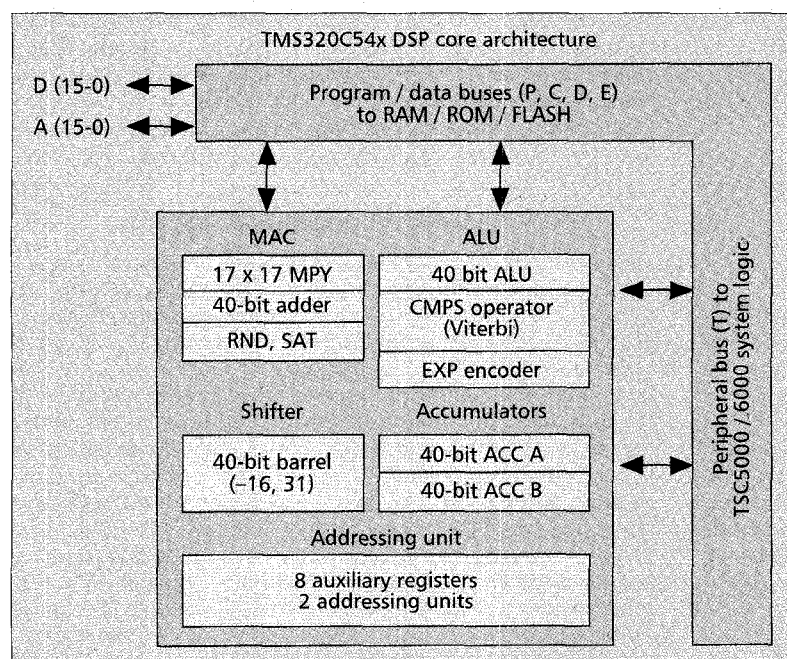
DSPs use extensive pipelining,<sup>1</sup> several independent on-chip memories, parallel function units, and a hardwired (rather than microprogrammed) design. DSPs utilize instructions such as MAC, bit reverse addressing for FFTs, and those that support circular buffering. Hardware-wise, DSPs contain phase-locked loop (PLL) oscillators, several 8- or 16-bit busses, clock interrupt circuits, onboard ROM and RAM, and sometimes an onboard CPU. DSPs typically process buffers of data, and often these buffers are reused on successive frames of data.

Programmable DSPs are either floating-point

<sup>1</sup> Pipelining refers to a special form of parallel execution of several operations, where downstream operations depend on results of upstream operations.

|               | Digital cellular                                     | Digital cordless                  | Paging                                 |
|---------------|------------------------------------------------------|-----------------------------------|----------------------------------------|
| North America | IS-95 (800, 1900)<br>IS-54/136(800, 1900)<br>PCS1900 | Unlicensed<br>900 MHz ISM<br>PACS | FLEX™<br>ReFLEX™<br>InFLEXion™<br>PACT |
| Europe        | GSM<br>DCS1800                                       | DECT                              | FLEX<br>ReFLEX<br>InFLEXion<br>Ermes   |
| Japan         | PDC<br>PDC1500                                       | PHS                               | FLEX<br>ReFLEX<br>InFLEXion            |
| Asia-Pacific  | GSM<br>IS-95<br>IS-54/136                            | DECT<br>PHS                       | FLEX<br>ReFLEX<br>InFLEXion            |

■ **Table 2.** Worldwide digital wireless standards for cellular, cordless, and paging. Courtesy of Texas Instruments; reprinted with permission; <http://www.ti.com/sc/docs/wireless/97/standards.html>



■ **Figure 1.** An example of a fixed-point DSP core. Courtesy Texas Instruments, reprinted with permission; <http://www.ti.com/sc/docs/wireless/97/core.htm>.

or fixed-point devices. The overwhelming virtue of fixed-point DSPs is that they can be very fast and inexpensive. Wireless applications primarily use fixed-point processors, and this article will focus on their applications. The most often used word size in fixed-point DSPs is 16 bits. Fixed-point arithmetic requires programmers to pay attention to precision, scaling, and overflows which come from quantization errors of analog-to-digital converters and finite-length multiplication. Multiplying two 16-bit numbers yields a 32-bit number, which in 16-bit DSPs needs to be truncated down to a 16-bit number again. Larger word sizes for accumulators, shifters, scaling and saturation hardware are used to simplify the programmer's task.

The dominating architecture for digital signal processor designs is the so-called Harvard architecture and its variants [3]. The major feature of this architecture is that the program and data buses are separate. In all Harvard architectures an instruction is fetched at the same time that operands for a previously fetched instruction are. This suggests that a three-stage pipeline consisting of the instruction fetch, operand fetch, and instruction execution is at the heart of this architecture. An example of a DSP core based on the Harvard architecture is given in Fig. 1. Whereas traditional microprocessor architectures achieve high performance with register-to-register instructions, DSPs have sufficiently high memory bandwidth and use more memory-to-memory instructions. This is done by accessing several memory allocations in each instruction cycle, which is possible due to rich addressing modes and parallel memory banks. The basic Harvard architecture contains two memories and two buses, and more sophisticated Harvard architectures may have four data memories and two program memories. DSPs are microcomputers rather than microcontrollers (since they have memory on-chip). Numerous memory banks are hard to implement off-chip due to pin number limitations and the speed penalty of off-chip busing. Often, additional memory is used off the DSP and internal buses are multiplexed to the outside. Register-indirect addressing is the method used to accomplish multiple-address specification in software. Modulo-mode addressing helps with filtering-type operations where circular access to data may be needed.

Storing data without power is a requirement for wireless portable devices. To this end technologies such as flash memory and ferroelectric memory are being included in DSPs, although the cost remains an issue at this time. DSPs with flash memory help designers by:

- Cutting down on ROM code turnaround time
- Facilitating field trial and last-minute debugging and code fixing
- Making external DSP memory boards unnecessary

#### DSP SOFTWARE AND DEVELOPMENT TOOLS

Three major approaches to DSP programming exist: interlocking, time stationary, and data stationary coding. These techniques are related to a particular use of the underlying pipeline-capable DSP hardware [3]. Different DSP hardware manufacturers and product generations follow different approaches to this issue. The use of any of these approaches requires skilled and experienced programmers. Even with DSP speeds seen today, much DSP code often needs to be tuned by hand to accomplish the tight real-time requirements of wireless communications. Optimizing compilers are making strides but are not capable of entirely replacing hand-written assembly code.

Signal processing system designers have at their disposal development tools to simulate, edit, compile, assemble, link, load, and debug DSP software programs [6; 7]. Simulators, hardware emulators, and development boards let developers

work in non-real time or in real time on the development of their code. Debuggers often have limited capabilities in real-time testing due to limited ability to set breakpoints.

Some system-level tools that deal with code development for DSPs are NIPower by Signal Technology Inc., Hypersignal Windows by Hyperception, DADiSP by DSP Development Corp., Signal Processing Worksystem by Alta, COSSAP by Synopsys, SystemView by Elanix, Omnisys by Hewlett Packard, and DSPStation by Mentor Graphics.

A hierarchical block diagram approach to representing both high- and low-level functions, which can be integrated into large models and systems, is more and more popular [9]. Examples of development tools that take this approach are Signal Processing Worksystem and COSSAP. These tools have optional libraries that contain blocks (representing functions) covering a large number of signal processing and digital communications functions. The suite of software by Hewlett Packard has extensive support for mixed analog/digital communications system design. Some specific blocks included in almost all packages are Viterbi trellis search implementations, quadrature amplitude modulation (QAM) modulators and demodulators, wireless channel models, equalizers, and synchronization blocks.

The first generation of simulation tools dealt with "system" level designs emphasizing behavioral (waveform) models unrelated to the actual implementation. Recent generation of communication system simulation tools is integrated with suites of tools that address VLSI/ASIC (application-specific integrated circuit) implementations, most often through an intermediate step of very high-speed integrated circuits hardware description language (VHDL)/Verilog code generation. The underlying methodologies of simulation techniques belong to one of the following classes: time-driven, event-driven, data- or stream-driven, or mixed-mode.

Parameters of blocks in block-diagram-based tools are modifiable, and functions can be either floating- or fixed-point. A user can create his own blocks with arbitrary functionality by writing C or other code and embedding it into a system through wrappers and block diagrams. There exist blocks which model DSPs and their functionality. A developer can write the actual DSP code within such blocks and evaluate its functionality and performance within the system. Some tools can generate DSP code from C code using compilers, which is particularly useful for quick evaluation of candidate algorithms. Hardware accelerators can be attached to software tools to experiment with data acquired in real time.

#### CELLULAR WIRELESS SYSTEMS

The meaning of wireless communications has been changing since the advent and widespread use of analog cellular systems, first defined by AT&T Bell Laboratories several decades ago [1]. At the heart of the concept lies the idea that frequency bandwidth, a limited natural resource, can be reused even in nondistant geographical areas. This can be accomplished if the total available bandwidth is divided into chunks, those chunks are appropriately assigned to neighboring geographical areas, and all components of a system pay attention to transmission power levels. The area is divided in cells of hexagonal shapes; each cell gets a base station, and as mobile users move through the area, the chunks of spectrum they use and base stations with which they communicate are changing. The very same concept remains in effect for future mobile communications systems [1, 2, 8, 10, 24, 25, 27]. Within the last decade, the concept of wireless personal communications has taken off. In many respects, it is a modification of the cellular concept (at the air interface level) with inclusion of low cost and

|                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Analog devices<br>ADSP-21xx family                                                                                | 16-bit fixed point, enhanced Harvard architecture 10.2–33 MIPS range<br>Program RAM: 0, 1, 2, 16 kbyte; data RAM: 512, 1, 2, 4, 16 kbyte, ROM: 0, 4, 8 12 kbyte<br>5V or 3.3V operation, power-down modes, host interface ports, DMA port, 16 bit codec<br><a href="http://www.analog.com/products/selection_trees/dsp/dsp.html">http://www.analog.com/products/selection_trees/dsp/dsp.html</a>                                                                                                                                                                                                                                  |
| ATMEL<br>LODE DSP core                                                                                            | 16-bit fixed point, two single-cycle MACS, 40-bit ALLU, four 40-bit accumulators, 64K program and 64K data memory address space, 16-level hardware stack, nestable loops<br><a href="http://www.atmel.com/atmel/products.tcsi/lcore.html">http://www.atmel.com/atmel/products.tcsi/lcore.html</a>                                                                                                                                                                                                                                                                                                                                 |
| DSP Group: Pine, Oak                                                                                              | 25 ns clycle, 40 MIPS, 3.3 V, DMA mode, power managment, expandable RAM/ROM, core-based<br><a href="http://www.lsil.com/products/unit5_5e.html">http://www.lsil.com/products/unit5_5e.html</a>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Lucent Technologies<br>DSP1610/1611<br>DSP1616<br>DSP1617/1618<br>DSP1620<br>DSP1627/1628/1629<br>SABRE DSP 16000 | Base stations, RAM up to 12K, ROM up to 2K, 26 MIPS at 3.3 V, 50 MIPS at 4.75 V<br>RAM = 2K, ROM = 12K, 26 MIPS at 3.3 V, 50 MIPS at 4.75 V<br>RAM = 4K, ROM = 24K, Viterbi accelarator, 26 MIPS at 3.3 V, 50 MIPS at 4.75 V<br>Base stations, RAM = 32K, ROM = 4K, Viterbi accelerator, 80 MIPS at 3.3 V, 120 MIPS at 4.75 V<br>RAM up to 16K, ROM up to 48K, Viterbi accelerator, 78 MIPS at 2.7 V<br>Two MACs, 8 40-bit accumulators, three-operand adder, 62 K x 16 dual-port RAM, 31-word cache, 200 MIPS<br><a href="http://www.lucent.com/micro/wam/docs/OT97215.pdf">http://www.lucent.com/micro/wam/docs/OT97215.pdf</a> |
| Motorola<br>DSK56000 family<br>DSP56100 family<br>DSP56300 family<br>DSP5600 family                               | 24-bit fixed point<br>16-bit fixed point, 30 MIPS/33 ns/60 MHz, up to 8K RAM and 4K data ROM, 256b RAM and 8K program ROM<br>24-bit fixed point, 80–100 MIPS, 3–3.6V, up to 24K program RAM, 7K data ROM and RAM<br>16/24 bit low power for cellular, 60 MIPS at 2.7V, 24K x 24 program ROM, 4K x 16 X-data RAM, 6K x 16 X-data ROM, 4K x 16 Y-data RAM, 6K x 16 Y-data ROM<br><a href="http://www.mot.com/pub/SPS/DSP/LIBRARY/56100/FM_REV01.PDF">http://www.mot.com/pub/SPS/DSP/LIBRARY/56100/FM_REV01.PDF</a>                                                                                                                  |
| Texas Instruments<br>TM320C1x<br>TM320C2x<br>TM320C5x<br>TM3206x                                                  | 16-bit fixed point, 5 MIPS, 200 ns multiply, up to 256 b RAM and 1.5K ROM, 5 V<br>16-bit fixed point, 10–12 MIPS, 80 ns multiply, up to 544 b RAM, 4K ROM, 5 V<br>16-bit fixed point, 40–50 MIPS, 20–50 ns, up to 10K RAM, up to 32K ROM, 3 and 5 V, \$15–60<br>0.25 $\mu$ technology, 200 MHz, 2000 MIPS, very long instrucion word architecture, RISC-like instructions, eight 32-bit instructions per cycle, 512 b program, 512 b data, 2.5 V CPU with 3.3V peripherals<br><a href="http://www.ti.com/sc/docs/dsp/products.htm">http://www.ti.com/sc/docs/dsp/products.htm</a>                                                 |

■ **Table 3.** List of some fixed-point DSP devices suitable for wireless communications applications.

the anytime/anywhere communications notion [8, 10]. From the point of view of DSP use, the two do not differ much.

In cellular systems, the requirements for base stations and mobile stations are different. At the physical layer, of concern to us, one can say that the functions which base stations and mobiles have to perform are about the same. The difference most often lies in the fact that base stations have to process signals going to and coming from many mobile stations, whereas a mobile deals with a signal of one base station only. There are exceptions to this rule (e.g., soft handoff in IS-95), but it is safe to say that the processing power of base stations has to be significantly higher than that of mobiles. Also, whereas mobile units have hard power consumption and size constraints, these issues are less prominent in the design of traditional base stations. This is changing with the more widespread use of micro- and picocells of increasingly small size.

Wireless systems can be classified according to the power emission levels they are allowed to use. One can consider those systems that operate at 0.5–5 W as high-power systems and those that operate below as low-power systems [11]. Since the most significant amount of power in a radio transceiver is taken by a power amplifier, the way the DSPs are used is not affected. The use of DSPs is going to be determined by the complexity of modulation, coding, specifics of the necessary receiver design, and other similar factors. Examples that prove this point are low-power systems such CT-2 and DECT cordless and versions of cordless in the industrial, scientific, and medical (ISM) band, as well as WLANs and the personal access communications system (PACS) (wireless local loop standard) [11]. Although they require low-power transmission, their physical layers are of significant complexity suitable for DSP applications.

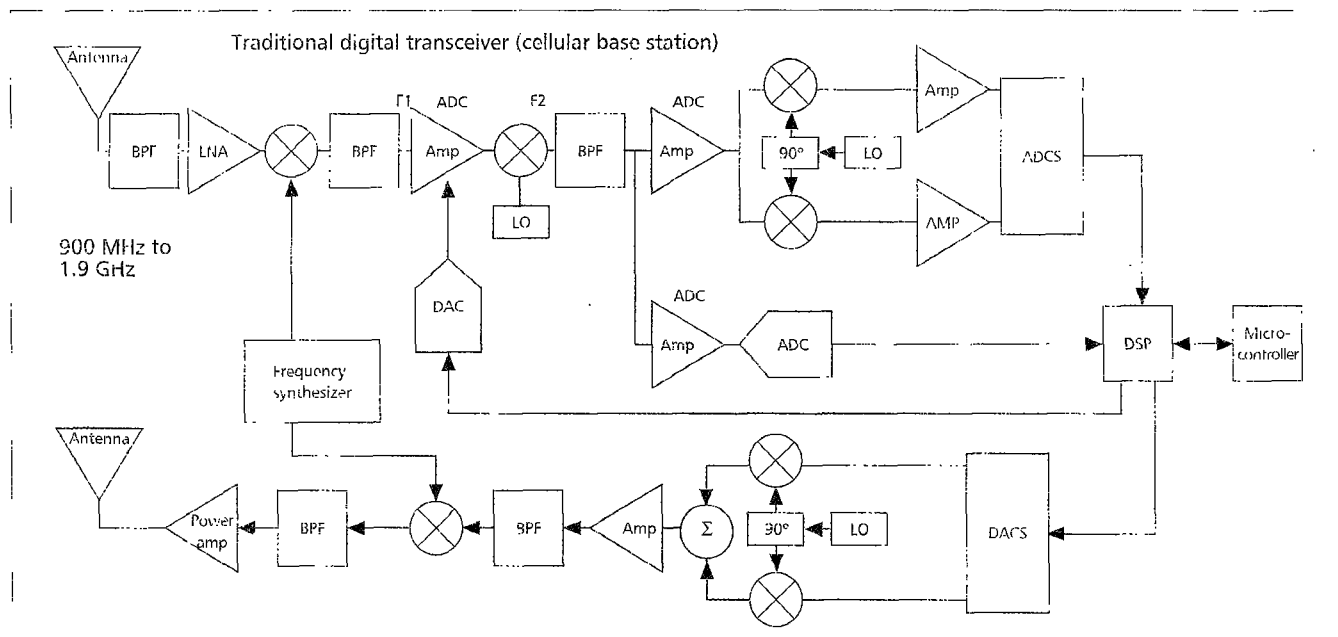
The most flexible development strategy is always the first choice of a seasoned engineer, assuming that all other things are of equal significance. From this point of view, using DSPs is the most desirable option. Algorithms can change conceptually, and the DSP program change is easy to execute. This is not simple when hardwiring is the implementation strategy, as is the case for ASICs. Wireless environments are notorious for undesirable effects on transmitted signals. Simulation of these channels to come up with system parameters is helpful, but only experiments with real-time platforms bring designers to ultimate product solutions. From this point of view DSPs are particularly suitable for wireless. If DSPs are designed making sure that power consumption is low, that they have appropriate amounts of on-chip memory, that they are easy to interface to and very low-cost, it is very compelling to use programmable DSPs rather than other implementation approaches.

## WIRELESS TRANSCEIVERS

**D**ue to the nature of radio communications, physical-layer design problems fall into one of two categories: those concerned with radio frequency/intermediate frequency (RF/IF) subsystems and those with baseband subsystems. We discuss baseband issues. An overview of design problems in the domain of RF/IF is given in [5].

### RADIO ARCHITECTURES

Radio communications require that a desired signal's original bandwidth be up-converted to the range of frequencies suitable for radio transmission, and that the complementary operation be done at the receiver. There are several different approaches to accomplishing the frequency conversion, all of



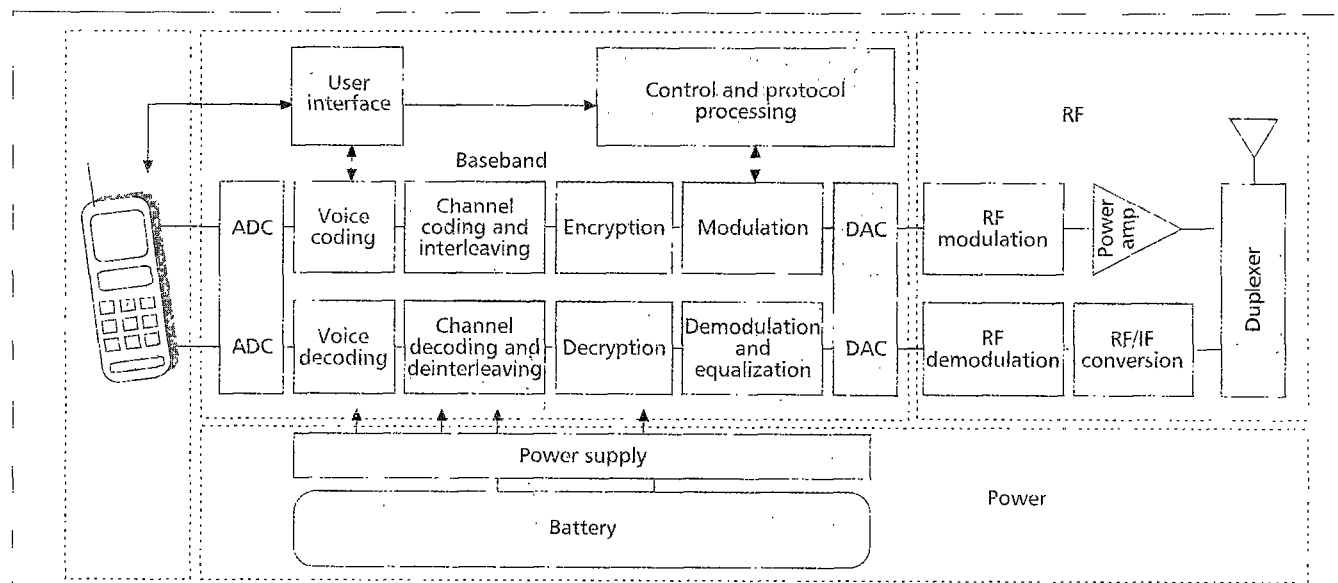
■ **Figure 2.** Superheterodyne radio architecture. Courtesy of Analog Devices, reprinted with permission; [http://www.analog.com/products/signal\\_chains/trad\\_dig\\_trans/trad\\_dig\\_trans.html](http://www.analog.com/products/signal_chains/trad_dig_trans/trad_dig_trans.html).

which have baseband portions amenable to DSP implementations. In [12], four principle radio transceiver architectures are defined: superheterodyne, digital IF receivers, direct conversion receivers, and I/Q-less analog-to-digital (A/D)-less digital receivers. An example of the most often used radio receiver architecture is given in Fig. 2. This figure illustrates the superheterodyne radio architecture where up/down-conversion is done in two mixing steps. For this architecture, all baseband functionality is allocated to a DSP.

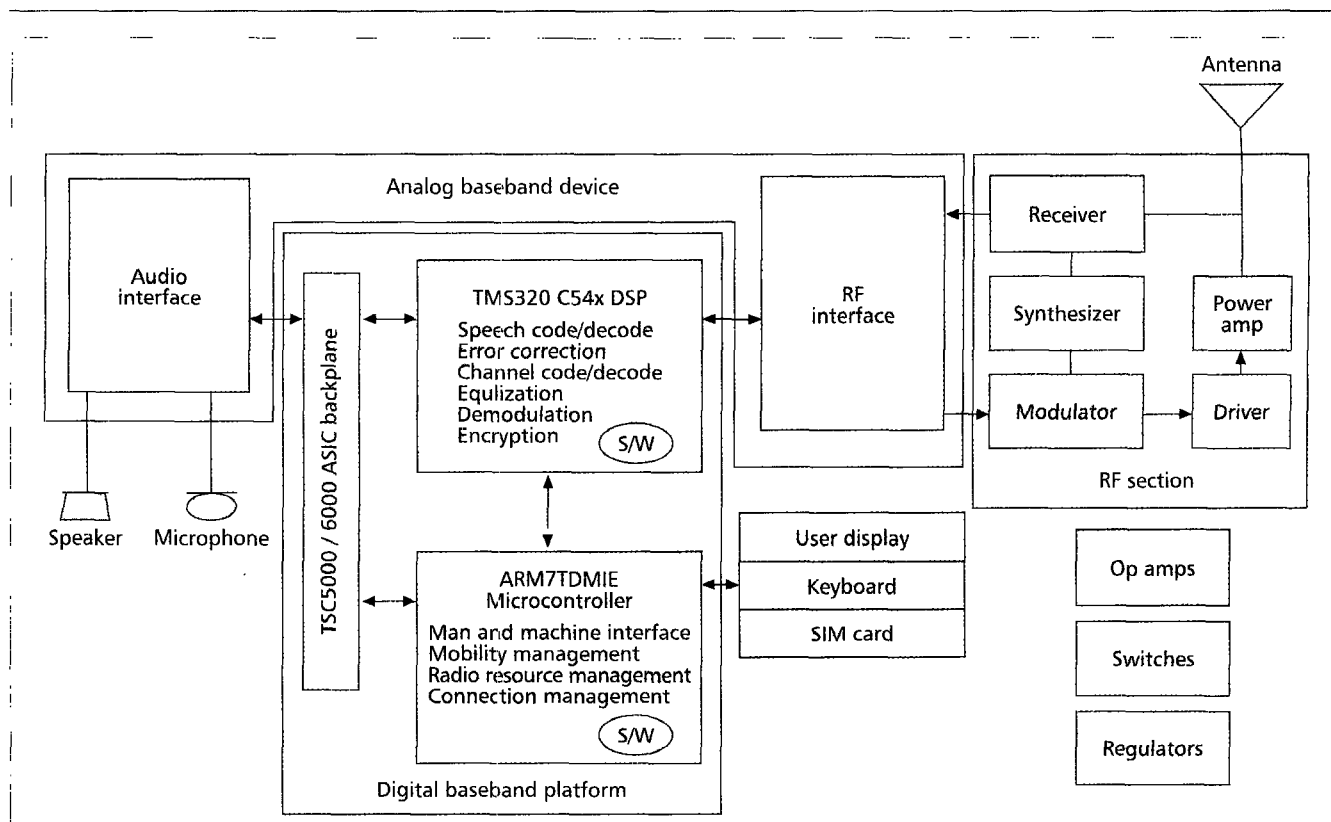
Direct conversion architectures remain vulnerable to problems of high-gain low-noise mixer, high dynamic range, extremely high sensitivity, I and Q phase balancing, rapid large DC offset cancellation, antenna isolation, and high-selectivity filters. Digital IF receivers use superheterodyne principles but move the digitization of the signal higher into the IF frequency band. They do not require analog devices

which individually handle quadrature components of the signal, but leave I and Q separation and processing to digital functions suitable for DSPs. The preeminent problem in using this architecture in cellular communications has to do with power-hungry high-speed A/D converters.

In recent times the concept of software radio has been given significant attention in some communications system development circles. The primary goal of the software radio is to replace as many analog components (those dealing with IF/RF sections) and hardwired digital VLSI devices of a transceiver with programmable devices (such as DSPs). This implicitly involves the use of wideband A/D and digital-to-analog (D/A) converters [13]. Several tutorial articles have been published in this magazine on the topic of software radio [14, 15]. This idea is conceptually very appealing, and very friendly to the use of DSPs. The proliferation of different air interface



■ **Figure 3.** Block diagram of a generic transceiver. Courtesy of Texas Instruments, reprinted with permission; <http://www.ti.com/sc/graphics/wireless/97/enale/rfgraph.gif>.



■ **Figure 4.** An example of partitioning of transceiver functions. Courtesy of Texas Instruments, reprinted with permission; <http://www.ti.com/sc/graphics/wireless/97/baseband/fig3.gif>.

standards may provide a boost to multiple-standard-device designs which would benefit from software radio work. However, the time has not yet come for low-cost low-power communications devices, and we shall not discuss this any further.

In this article we focus on the use of DSPs within traditional superheterodyne architectures.

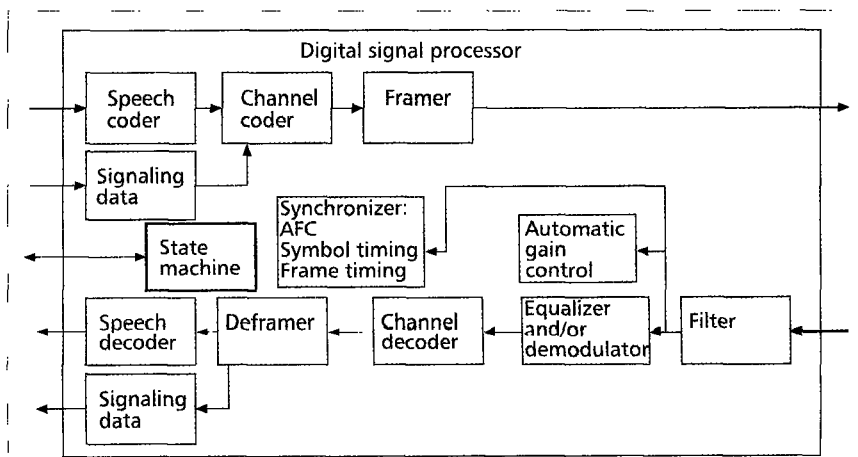
#### BASEBAND HARDWARE ARCHITECTURES

In designing baseband portions of portable digital communications devices, systems engineers can mix and match microcontrollers, DSPs, and ASICs. The choice can be driven by development time constraints, cost, size, level of comfort with functions that need to be implemented, challenges data rates impose on hardware, and other system requirements. Different wireless standards have different optimal hardware designs. A block diagram illustrating a generic wireless transceiver is shown in Fig. 3. In the following we look at examples of two American digital cellular standards: IS-136 and IS-95.

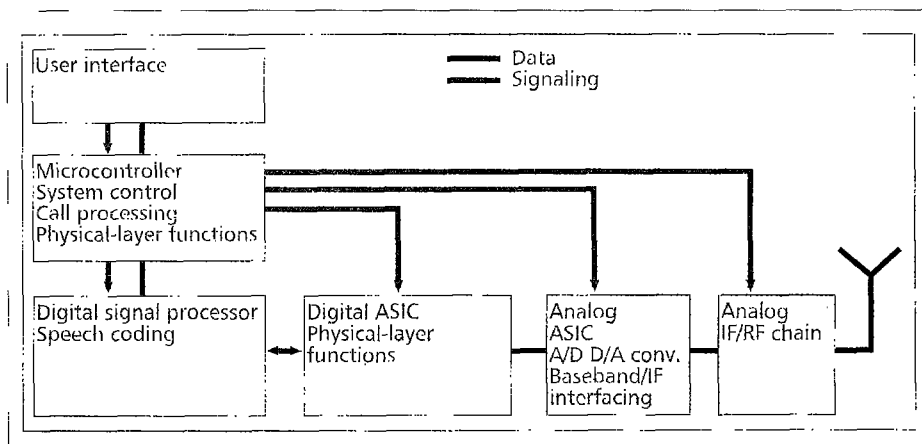
**IS-136** — IS-136 is a time-division multiple access (TDMA)-based standard with channel data rates of 24.3 kbaud with  $\pi/4$ -DQPSK (differential quadrature phase shift keying) modulation. For heterodyne receiver architectures, this rate creates no extravagant challenges to A/D and D/A converters. Physical-layer receiver functions needed to accommodate this standard belong to the class of compute-intensive signal processing operations suitable for DSPs. This is particularly true for digital filtering, equalization, and channel decoding. Speech coding/decoding is based on code excited linear prediction (CELP) principles and requires

vector searches and filtering, both appropriate for DSP implementations. Since cellular standards require numerous call processing operations, the existence of a microcontroller is required. In Figs. 4 and 5 we show examples of a heavily exploited hardware/software architecture for IS-136.

**IS-95** — IS-95 is a code-division multiple access (CDMA)-based standard with channel data rates of 1.2288 Mchip/s and 19.2/28.8 kbaud. The chip rate mentioned above is not the rate at which digital samples can be processed in today's general-purpose programmable DSPs. The implication is that IS-95 mobile hardware architecture cannot follow the same path as IS-136 architectures. A considerable portion of the baseband functionality related to signal despreading has to be implemented in an ASIC. The rest of the receiver physical layer does not have an extensive signal processing component



■ **Figure 5.** Partitioning of IS-136 functions into DSP firmware.



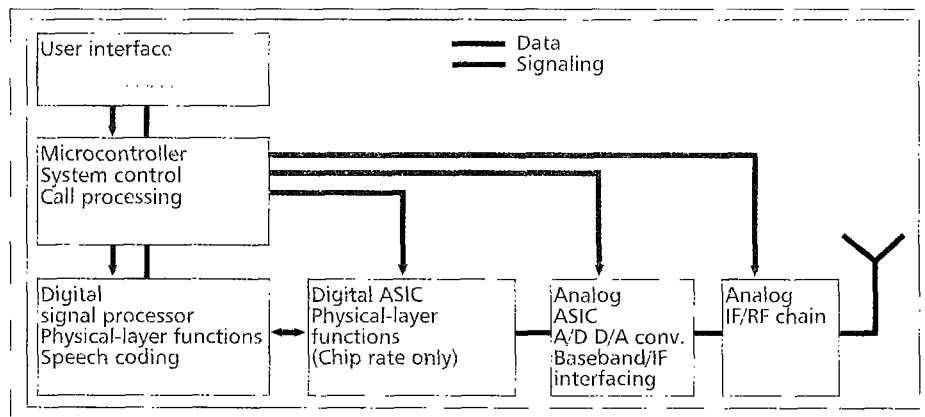
■ Figure 6. ASIC-centric hardware architecture for a CDMA-based IS-95 system.

to it but rather a control-like structure: digital filtering is done at the base transmitter in its entirety, and there is no need for an equalizer. This suggests that one credible hardware architecture puts no physical-layer functions in a DSP, but shares them between an ASIC and a microcontroller, as seen in Fig. 6. Speech coding and decoding are the only functions executed in a DSP. There exists an alternative architecture which is suitable for handling possible frequent modifications of the IS-95 standard, where both ASIC and microcontroller take a lighter processing load and delegate whatever is possible of the physical layer to a DSP. This is illustrated in Fig. 7.

**Base Stations** — In the case of IS-136, for each communications link (mobile user) the functions a base station needs to execute are of the same complexity as those a mobile needs to execute. The difference lies in the fact that base stations need to handle many mobiles. This necessitates that careful thought be given to optimizing DSPs such that they can accommodate this requirement. Indeed, DSP vendors are paying attention to this segment of the wireless market. An illustration of the use of Lucent's DSP 1620 in base station design is shown in Fig. 8. Texas Instruments has recently come up with a family of fixed-point DSPs, TMS320C6x, with 2000 MIPS performance claimed. This DSP uses the VelociTI™ Very Long Instruction Word (VLIW) architecture. It consists of multiple execution units running in parallel to perform multiple instructions during a single clock cycle. In early September 1997, Lucent announced a DSP16000 core family of DSPs using Sabre architecture. This architecture has around 200 MIPS performance, two parallel MACs, a bank of eight 40-bit accumulators, a 31-word instruction cache, and low power consumption. It can handle more demanding algorithms for wireless applications.

In the case of IS-95, the reverse link specification is different from the forward link, which implies that the architecture of the IS-95 base station receiver will differ from the mobile station receiver's. Still, the despreading of the signal has to be executed at speeds unsuitable for DSPs.

Since GSM is the most widely deployed cellular system worldwide, DSP vendors found sufficient motivation to design specialized DSP-based solutions for base station and mobile station design. An example is Lucent's Sceptre chip set [33].



■ Figure 7. DSP-centric hardware architecture for a CDMA-based IS-95 system.

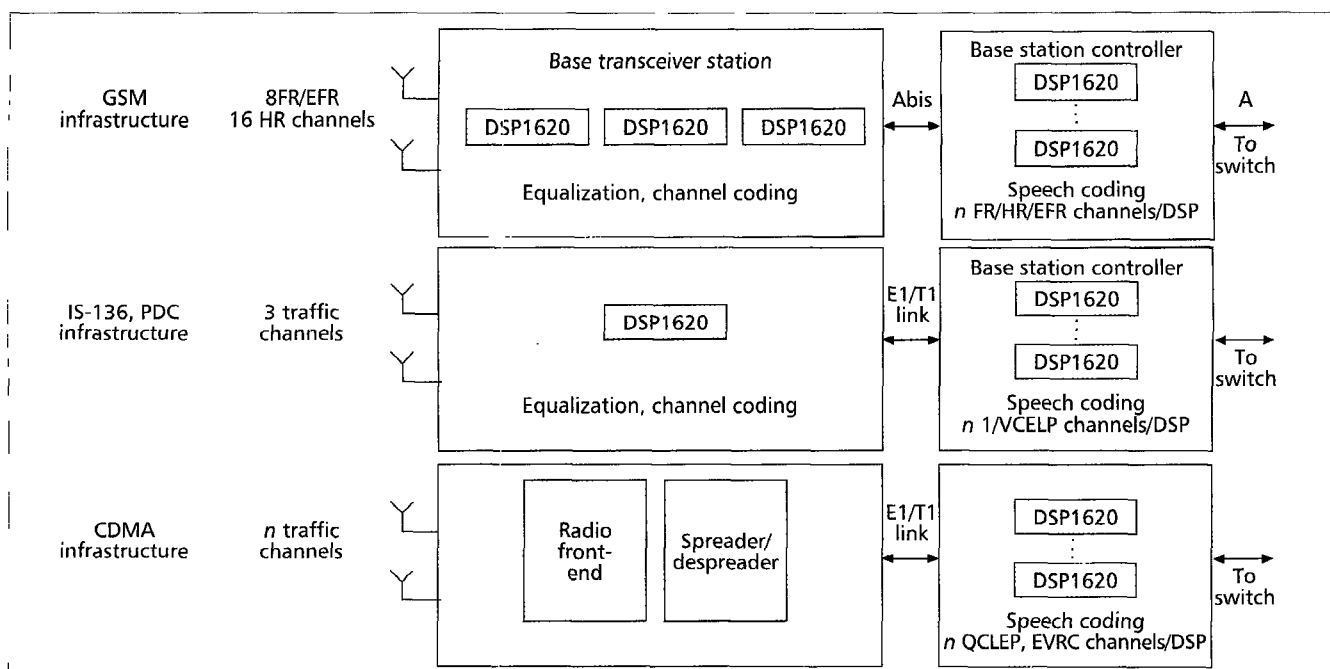
## DSP FIRMWARE ARCHITECTURES

Today's digital cellular communications are used primarily for circuit-switched voice communications. This implies that for the duration of the call, there will be a steady stream of digitally coded speech frames (in IS-95, the data will come at varying rates due to the utilization of voice activity). Thus, at the receiver end there will be a need for continuous collection of received data samples prior to actual processing. On top of speech processing, which is organized in frames, the interleaving function (used for minimizing detrimental

effects of bursty errors) is used. Collected data will thus need to be buffered. These facts suggest the use of firmware architectures that utilize a DSP interrupt for data collection and buffering, and a set of functions which run sequentially in the "background." In IS-136, the designers of the standard have made sure that active transmitter and receiver slots are nonoverlapping. This further simplifies firmware architectures in that no potentially conflicting simultaneous multiple interrupt activity need take place. A timing structure for physical-layer processing for the IS-136 standard is shown in Fig. 9, together with a list of DSP-implemented functions which need to be executed in the background. The same structure makes it possible to eliminate duplexers from the phone design (since IS-136 is a dual-mode standard, the analog mode would still require a duplexer in 800 kHz band cellular frequencies).

In a typical phone design, it is usual to dedicate master and control functionality to a general-purpose microcontroller. Microcontrollers are much more suitable than DSPs for branching functions which are dominant in system control software. The DSP is thus typically controlled by the microcontroller, and it runs physical-layer functions in parallel with the microcontroller running system control functions.

A critical issue in the design of a firmware architecture is the type of interface between the microcontroller and DSP. Several hardware options exist: parallel ports, serial ports, and dedicated hardwired first-in first-out buffers (FIFOs). There is usually a mismatch between fast DSP and slow microcontroller clocks. If the data is communicated over a parallel input-output (PIO) or serial input-output (SIO) port from the DSP to the microcontroller with the intention of generating microcontroller interrupts, the flow of the microcontroller software may be undesirably broken by frequent interrupt service routines.



■ **Figure 8.** The use of multiple DSPs in the base station transceiver design. Courtesy of Lucent Technologies, reprinted with permission; <http://www.lucent.com/micro/wam/docs/OT97199.pdf>.

Special-purpose FIFOs are more desirable from this point of view, although they require some additional hardware.

Prudent programming practices such as writing modular code and breaking software into small, easily readable and modifiable functions are most often not applicable to writing real-time-sensitive wireless applications. Programmers have to aggressively optimize the execution time, size, and memory use of the code. It is a norm that the code for some function has to be rewritten several times before the real-time constraints of the DSP are met and all three factors are harmonized. Sixteen-bit digital signal processors have some number of 32-bit or longer registers. In cases when actual numbers involved in computations do not require more than 16-bit results, a programmer may choose to utilize unused portions of a 32-bit-long register in unexpected ways such as a temporary register/memory location.

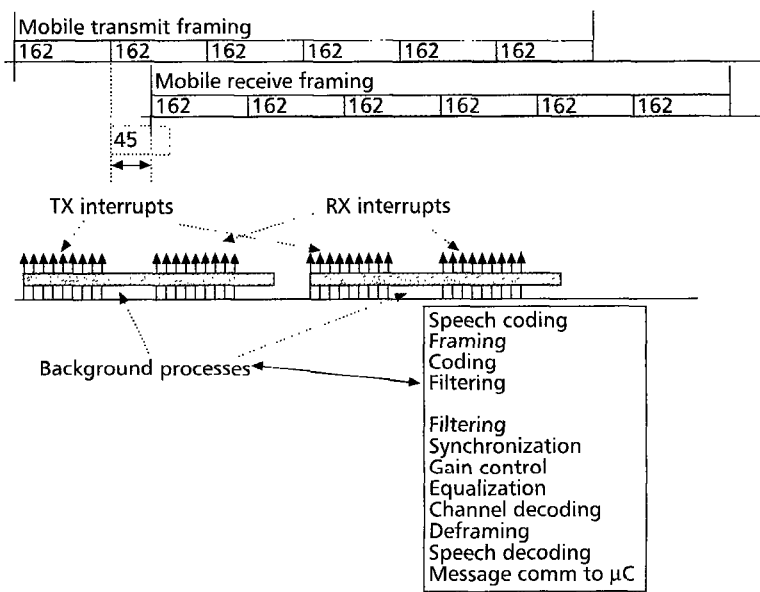
Although one may buy DSP operating system kernels, DSP programmers have not opted to use them. Again with focus on optimization of firmware from every aspect, custom and minimal state machines are used for wireless applications.

It can easily be forgotten that wireless software has to be field tested in devices running in mobile environments. Large amounts of data may need to be collected and analyzed in real time. This requires that a DSP software architecture supports rapid data movement to devices outside the DSP which are dedicated to massive data processing. An excellent example of this would be the so-called mobile station monitor required for testing of IS-95 physical-layer functionality. This monitor needs to continuously analyze and display multipath profiles critical to the operation of rake receiver fingers.

### ALGORITHMS

To illustrate the involvement of DSPs in the wireless transceiver design, we shall briefly describe some issues in the implementation of physical-layer algorithms.

**Filtering** — Filtering is one of the operations cited to be ideally suited for DSP applications. FIR filtering is a repetitive process performed by multiplying the set of input signal samples with a fixed set of known filter coefficients. In the example of IS-136, pulse shaping is done at a transmitter with a square-root raised cosine filter, and appropriate matched filtering has to be done at a receiver. Although straightforward, there are instances when care needs to be exercised to make sure that filtering is executed within the smallest possible number of machine cycles. In Fig. 10 we provide an example which illustrates required modifications to the DSP code and array indexing when there is a mismatch between sampling rates of filter coefficients and the signal to be filtered. The rate of the signal before passing through the pulse shaping filter is



■ **Figure 9.** The timing structure for physical-layer processes in IS-136.

1/T. If one uses a filter with 4 x oversampling, the requirement is that the filter sample rate be 4/T. If one wishes to avoid unnecessary interpolation and multiplication with zeros, a scheme is shown that requires reordering of filter coefficients as well as appropriate loop indexing. Were this modification not done, a penalty in real time would be considerable.

**Receiver Synchronization** — There exist several layers of synchronization:

- Frequency synchronization
- Carrier recovery for coherent demodulation
- Symbol timing recovery
- Slot, frame, and superframe synchronization

Most synchronization methods require that they be accomplished through initial acquisition, tracking, and reacquisition.

Synchronization of frequency is typically accomplished through the use of automatic frequency control (AFC). In digital receivers, the received signal constellation rotation is monitored in a DSP and a phase error based measurement is differentiated and filtered. This error signal is used in a digital VCO to come up with a number which will correct the operation of an analog component (VCXO). From the DSP this control signal can be sent through the dedicated D/A to a digitally controlled device or converted to a pulse-width modulation form

suitable for transmission as an analog signal. Components of wireless products are specified to be inexpensive and thus coarse frequency acquisition algorithms must be applied prior to getting a wireless phone in the mode of fine-resolution frequency tracking. Algorithms for frequency synchronization are often feedback-based and require the operation of the PLL suitable for DSP implementations.

Carrier recovery is associated with coherent receivers, where knowledge of the phase of the received signal is required. It is simple to implement a fully digital phase correction algorithm in DSP firmware, again by monitoring the phase error in a signal constellation. It is the decision of implementers if the phase correction is done in the analog component based on a digital control signal or fully digitally implemented.

In the example of IS-136, frame timing is the first synchronization that can be accomplished using training symbols at the beginning of all data frames. Frame synchronization is accomplished by correlation of the received waveform with the replica of the training waveform(s) known to the receiver. This is the feed-forward type of operation. The receiver repetitively correlates the signal until it identifies a peak in the correlation function, and based on the peak's location in time adjust its timing. It is

```
/*-----
FIR filter coefficients
square root raised cosine alpha = .33
Note permuted indices and 4 segments -
done for filtering efficiency in a DSP
*/
```

```
/*3*/ coeff(0) = -2.57910E-03;
/*7*/ coeff(1) = 8.57456E-03;
/*11*/ coeff(2) = -2.44425E-02;
/*15*/ coeff(3) = 0.254584;
/*19*/ coeff(4) = 3.33408E-02;
/*23*/ coeff(5) = -1.10911E-02;
/*27*/ coeff(6) = 3.63476E-03;
/*31*/ coeff(7) = -5.64713E-04;
```

```
/*2*/ coeff(8) = -4.19223E-03;
/*6*/ coeff(9) = 1.58502E-02;
/*10*/ coeff(10) = -4.47768E-02;
/*14*/ coeff(11) = 0.201956;
/*18*/ coeff(12) = 0.117725;
/*22*/ coeff(13) = -3.45362E-02;
/*26*/ coeff(14) = 1.19383E-02;
/*30*/ coeff(15) = -2.59598E-03;
```

```
/*1*/ coeff(16) = -2.59298E-03;
/*5*/ coeff(17) = 1.19383E-02;
/*9*/ coeff(18) = -3.45362E-02;
/*13*/ coeff(19) = 0.117725;
/*17*/ coeff(20) = 0.201956;
/*21*/ coeff(21) = -4.47768E-02;
/*25*/ coeff(22) = 1.58502E-02;
/*29*/ coeff(23) = -4.19223E-03;
```

```
/*0*/ coeff(24) = -5.64713E-04;
/*4*/ coeff(25) = 3.63476E-03;
/*8*/ coeff(26) = -1.10911E-02;
/*12*/ coeff(27) = 3.33408E-02;
/*16*/ coeff(28) = 0.254584;
/*20*/ coeff(29) = -2.44425E-02;
/*24*/ coeff(30) = 8.57456E-03;
/*28*/ coeff(31) = -2.57910E-03;
```

```
/* *****
*Function pulse_shape_one_half_frameT4
*Description:
*- Does filtering for T/4 samples per symbol
*- Filter coefficients are contained in coeff[32]
* Note that indexing is customized for efficient real-time
* execution
*- Shaping is done at a rate of 4 samples per symbol, this
* requires interpolation with 0 between every pair of symbols
* from I_ymvol_out and Q_symbol_out
* *****
void pulse_shape_one_half_frameT4(void)
{
/*-----
* Local Variable Definitions
*-----*/
int i,j,k;
float sum;
int fir_segment_offset;
staticfloat *curr_I_start_pt;
staticfloat *curr_Q_start_pt;

/*-----
*Pulse Shaping For I channel
*-----*/
/* Loop selecting input symbols */
for (i=0;
i<NUMBER_OF_SLOTS_PER_HALF_FRAME*NUMBER_OF_SYMBOLS_PER_SLOT;
i++)
{
/*Loop selecting one of (4) samples per symbol*/
for (j=0; j<NUMBER_OF_SAMPLES_PER_SYMBOL;j++)
{
/*Computing the output sample index */
/*out sample_no = NUMBER_OF_SAMPLES_PER_SYMBOL*i+j;*/
/*Computing the offset to the correct segment of filter coefficients */
fir_segment_offset = j*8;
/*FIR filtering (only 32/4 multiplications because of interpolation */
sum = 0;
for (k=0; k<8; K++)
{
sum=sum + I_symbol_out[i+k] * coeff [(fir_segment_offset + k)];
}
*I_fir_out_P++=sum;
}
}/*End for (i=0; ... */
```

■ Figure 10. Optimized C code for clock cycle minimization in a rate-mismatched pulse-shaping filter implementation.

|                                 |                 | Function                | Instruction cycles | Number of repetitions/ frame of 20 ms | MIPS    |
|---------------------------------|-----------------|-------------------------|--------------------|---------------------------------------|---------|
| modem_main                      | forward channel | voice_decoding          | 50,000             | 1                                     | 2.5     |
|                                 |                 | analog_codec_isr        | 50                 | 160                                   | 0.4     |
|                                 |                 | deinterleaver           | 8000               | 1                                     | 0.4     |
|                                 |                 | channel_decoder         | 10,000             | 1                                     | 0.5     |
|                                 |                 | AFC                     | 30                 | 160                                   | 0.24    |
|                                 |                 | AGC                     | 80                 | 160                                   | 0.64    |
|                                 |                 | symbol_timing_recovery  | 80                 | 160                                   | 0.64    |
|                                 |                 | frame_timing            | 800                | 14                                    | 0.56    |
|                                 |                 | channel_delay_detection | 30                 | 160                                   | 0.24    |
|                                 |                 | doppler_detection       | 30                 | 160                                   | 1.2     |
|                                 |                 | slicing                 | 200                | 160                                   | 1.6     |
|                                 |                 | differential_detector   | 150                | 160                                   | 1.2     |
|                                 |                 | deframing               | 150                | 1                                     | 0.0075  |
|                                 |                 | messages_to_uc          | 1000               | 1                                     | 0.05    |
|                                 |                 | receive_data_isr        | 200                | 160                                   | 1.6     |
|                                 |                 | equalizer*              | 150,000            | 1                                     | 7.5     |
|                                 | reverse_channel | voice_coding            | 40,000             | 1                                     | 20      |
|                                 |                 | interleaving            | 80,00              | 1                                     | 0.4     |
|                                 |                 | channel_coding          | 2500               | 1                                     | 0.125   |
|                                 |                 | framing                 | 2000               | 1                                     | 0.1     |
| Total MIPS* (no equalizer case) |                 |                         |                    |                                       | 31.3425 |

■ **Table 4.** An example of MIPS requirements for some IS-136 algorithms.

important to note that in cases where frequency offset exists the correlation will fail when it is done only against the ideal original training waveforms. For good performance in realistic environments it is also necessary to correlate the received signal against frequency-shifted original waveforms. A typical DSP is ideal for correlation operations. Tracking of the frame timing can be accomplished by the same operation, except that the span of the received signal which needs to be correlated is significantly smaller since we are close to the actual timing. Here, though, one has the choice of implementing better resolution.

Symbol timing recovery makes sure that a received signal waveform is sampled as close as possible to the optimal sampling point for detection. Since it is desirable to have A/D converters operate at the slowest possible rate, it is required to be able to finely change the sampling position. Indeed, one can choose to adjust the sampling phase of an A/D explicitly. However, more and more often in digital receivers, the preferred choice is to let the converter keep sampling at an arbitrary phase and to use digital interpolation to find the value of the signal at the optimal sampling point from two or more neighboring samples of an arbitrary phase.

**Equalization** — The fact that wireless channels have an associated delay spread which causes intersymbol interference requires in some instances that this be compensated for. For a 24.3 kb/s IS-136 system, the delay spread which causes

trouble and requires an equalizer is on the order of 10  $\mu$ s. Two principal techniques are used for equalization: decision-feedback equalizers (DFEs) and maximum-likelihood sequence estimators (MLSEs). Equalization is one of the most MIPS-intensive functions in cellular phone receivers. Although equalizers are not always needed, since channels often have smaller delay spreads receivers/DSPs have to be designed to be able to handle equalization. DFEs consist of two FIR filters and are amenable to DSP implementations. MLSE equalization requires clever memory addressing approaches which DSPs support.

**Channel Coding/Decoding** — Channel coding is almost always applied in cellular communications systems. Exceptions to the rule are systems where fading rates are expected to be very slow, and thus no amount of coding can recover the signal when in a deep fade for significant amounts of time. The operation of coding is always simpler than decoding. Both block-code and convolutional-code decoding can be demanding in terms of the number of cycles required. DSP vendors are paying particular attention to efficient software implementations and/or building specialized hardware for trellis search techniques which are effective for various decoding schemes. These accelerators are probably the first of a number of accelerators that will deal with speeding up the operation of DSPs. Constraint length 7 rate 1/2 coding used for IS-136 can be dealt with in a DSP, but decoding constraint length 9 rate 1/2 codes for IS-95 are more comfortably done in a specialized

| Speech coder  |                   | kb/s | MIPS | ROM   | RAM   |
|---------------|-------------------|------|------|-------|-------|
| VSELP - IS54  | 1st TDMA IS-54    | 7.95 | 24.5 | 9k    | <2 k  |
| ACELP - IS641 | 2nd TDMA IS-136   | 7.4  | 25.0 | 11.3k | 4.5 k |
| JVSELP - PDC  | Full rate PDC     | 6.7  | 24.7 | 10k   | 2 k   |
| PSI-CELP-PDC  | Half rate PDC     | 3.45 | 38   | 23k   | 4 k   |
| RPE-LTP-GSM   | Full rate GSM     | 13.0 | 8    | 4k    | 1 k   |
| ACELP-GSM     | 2nd full rate GSM | 12.2 | 25   | 15k   | 5 k   |
| VSELP - GSM   | Half rate GSM     | 5.6  | 28   | 20k   | 6 k   |
| QCELP - 8k    | 1st CDMA          | 8.5  | 23   | 7k    | 2.4 k |
| QCELP - 13k   | 2nd CDMA          | 13.2 | 27.5 | 13.5k | 4 k   |
| EVRC- IS127   | 3rd CDMA          | 8.5  | 32   | 15k   | 5.5 k |

■ **Table 5.** Summary of various speech coders and resource requirements.

VLSI accelerator. It is interesting to note that Viterbi MLSE equalization techniques can sometimes share trellis searching structures with channel code decoders. This is most obvious in GSM, where modulation is binary.

**Automatic Gain Control** — While propagating through a wireless channel, a signal can experience dramatic changes in power levels. Standard deviation of a signal due to shadowing is on the order of 8–12 dB, whereas Rayleigh fading can cause as much as 30–40 dB of rapid signal power fluctuations. It is not always desirable to get rid of all Rayleigh fading fluctuations (especially when they occur rapidly), but shadowing needs to be compensated for. Automatic gain control schemes in modern receivers collect and process data in the digital domain, and then send control information to analog components which adjust signal power levels prior to A/D converters. It is not usual to have oversampled A/Ds (in terms of the number of bits), which would let DSPs cover most of the dynamic range of radio signals.

**DSP State Machines** — DSPs used in wireless terminals are usually controlled by microcontrollers. They operate according to microcontroller orders, and, rather than using sophisticated operating systems, follow states driven by state machine flow. Bearing in mind that DSPs are not good in branching operations, it is of some importance to carefully implement them.

To illustrate processing loads with which a DSP might have to deal, we provide an example of approximate MIPS requirements for some IS-136 algorithms in Table 4.

## SPEECH PROCESSING

**S**peech processing for digital wireless communications refers to the process of reducing the bit rate to represent digitized speech while maintaining good quality of the synthesized speech. Many speech-coding standards have been developed in the last decade because of the advances in DSP technology and the increase in demand for new speech coding techniques [34, 35]. Standards bodies such as ITU, TIA, ETSI, and Radio Communications Research (RCR) have been responsible for defining a set of speech coding standards for digital cellular applications.

Speech coders can be characterized by the following attributes: bandwidth, channel-error sensitivity, bit rate, quality, complexity, and delay. Depending on the application, trade-offs can be made between these attributes to come up with a specific coder. A speech signal for digital cellular appli-

cations is band-limited to between 300 Hz and 3400 Hz and is sampled at 8 kHz. Cellular telephone standards use speech coders ranging from 3.45 kb/s to 13.2 kb/s [29, 30, 31, 36, 38–41], as shown in Table 5.

## ALGORITHMS

Most of the speech coders used in cellular standards are based on linear prediction based on analysis by synthesis (LPAS) [35]. Figure 11 shows the basic principles involved in LPAS. A speech signal in a speech coder is represented by a set of parameters such as filter coefficients, gain coefficients, and codebook indices.

The basic principle of LPAS coding is that parameters to generate the excitation signal are determined using a decoding structure similar to that used at the decoding end of the channel. The long-term prediction (LTP) filter models the long-term correlations (fine structures in speech spectrum), while the short-term prediction (STP) filter models the short-term correlations of the speech signal. Almost all the coders used in cellular applications use a 10th-order all-pole filter model for STP, whereas a 1st–3rd-order filter is used for LTP. These coders operate on a block of speech samples, usually of 20–40 ms duration, to extract the various parameters. The decoded speech is produced by passing the output from the excitation generator through the filters. The parameters to generate the excitation are determined by minimizing the error between the original input speech and the synthesized speech.

Once the speech parameters are determined, they have to be quantized. The coder used in first-generation IS-54 TDMA, a vector sum excited linear prediction (VSELP) coder, uses the scalar quantizer to represent the STP filters [29]. This coder has two fixed codebooks and one adaptive codebook. The codebook indices are used for transmission. The gains for these code vectors are vector quantized. While there was a bit-exact specification [35] for the first-generation GSM coder, there was no such requirement for the IS-54 VSELP. While the coders for TDMA and GSM cellular standards operate at a fixed rate, the CDMA standards use variable-rate speech coders [34]. The idea behind variable-rate coding is, that depending on the speech activity in the speech signal, the number of bits used to represent the input speech signal varies. Full rate is used to code active speech, while half and quarter rates can be used for noise-like speech and silent background signal. Qualcomm's QCELP 8.5 kb/s coder is used in the IS-95 standard [34].

The quality of the synthesized speech depends on the bit rate. It can be said that, in general, the higher the bit rate the better the synthesized speech quality. Also, for a given bit rate, better quality can be achieved with more complex speech coders. This means that complexities of the computational requirements increase with lower bit rates and with higher quality. The personal digital cellular (PDC) system used in Japan uses a VSELP coder similar to the one used in IS-54 systems in the United States [29]. This coder uses one fixed codebook twice the size of that used in IS-54. Thus, the quality of speech is slightly lower than that of the 8 kb/s VSELP. The half-rate PDC system uses a more complex coder, the pitch synchronous innovation CELP (PSI-CELP) coder at 3.45kb/s. The quality of this coder is as good as the full-rate PDC coder and even better under error-free conditions [35, 37]. The IS-95 CDMA system uses a 13.2 kb/s QCELP coder in order to provide good speech quality [40].

In order to get better speech quality, different vector quantization methods to represent filter parameters as well as gain parameters have been used. By using fractional lag values to represent the LTP (adaptive codebook), the quality can be improved. Based on the reactions and customer satisfaction,

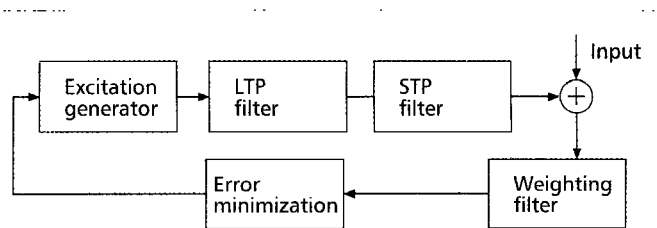
the GSM and North American cellular standards committees have adopted new speech coders for the various cellular standards [31, 32, 39–41]. The IS-136 TDMA systems used in 800 MHz and 1900 MHz in the United States utilize the algebraic code excited linear prediction (ACELP) coder [31]. The coder still uses adaptive codebook to represent the LTP effects, but uses fractional pitch values with 1/3 resolution. The fixed codebook is constructed from a structure based on interleaved single-pulse permutation design. The filter parameters are vector quantized using line spectral frequency (LSF) representation. The gains are vector quantized. The resulting quality of this coder is very good [30]. The GSM full-rate standard has adopted a new coder, at 12.2 kb/s, based on ACELP principles [39]. It uses 1/6 fractional pitch representation, and uses two sets of LSF parameters, vector quantized, for every frame to represent the synthesis filter. Similarly, the IS-95 CDMA standards committee has adopted a new coder, the enhanced variable rate coder (EVRC), for the CDMA system [41]. The EVRC operates at 8.5 kb/s and is also a variable-rate coder like QCELP. The EVRC attempts to match the time-warped version of the original residual signal instead of trying to match the original residual signal like the CELP coders. This reduces the number of bits needed to represent the pitch; as a result, more bits are available to represent the excitation. The EVRC also uses the ACELP codebook for the fixed codebook representation. Different pulse structures are used for different rates depending on speech activity. The quality of the EVRC is very good, and it maintains good quality under error conditions [41]. Even though the half-rate GSM coder was defined in 1995 [35], not many half-rate GSM systems have been installed.

### IMPLEMENTATIONS

All described speech coders are computationally very demanding. They require powerful DSPs to do the coding of speech in real time. The advances in VLSI technology made it possible to have DSPs which can run at 120 MIPS with 16 kbytes of on-chip RAM and 48 kb of on-chip ROM to store the program. During implementation of these speech coders one can use various software and hardware development tools available from leading DSP suppliers. All these coders are first implemented in floating-point C code. The VSELP, JVSLEP, and QCELP coders were only specified at the algorithm level. The standards committee provided the reference coder implemented in floating point using the C language. There were no fixed-point C code specifications, which made the task of evaluating and comparing various implementations of the same coder by different people using different DSPs difficult.

The recent coders have been defined to be bit-exact standards. The basic operations involved in implementing the speech coder are defined using 16-bit and 32-bit operations. These operations are defined very precisely using C code. The standard also provides a set of test vectors and the corresponding test results so that one can test and compare his/her own DSP code to make sure it meets the bit-exact specification. For example, the IS-686 standard specifies that handsets meet these bit-exact requirements, whereas the base stations can deviate from this requirement as long as they meet other requirements [32]. Most currently available DSPs do not have single assembly instructions to directly implement some of these basic operations specified in the standards. However, there are efforts to incorporate newer instruction sets in future DSPs. Table 5 gives a summary of various speech coders used in cellular standards.

It is often assumed that once you have a floating-point C code, it is easier to use a compiler to generate fixed-point DSP assembly code. There have been efforts to come up with methods for doing this [42], but the resulting DSP assembly code is



■ **Figure 11.** Linear prediction-based analysis by synthesis (LPAS) speech coding.

very inefficient. Under these conditions, it is left up to the individual to come up with an efficient DSP code on a given DSP for a specific coder. One of the most important aspects of implementing speech coders is to keep RAM usage to a minimum. There is nothing one can do about the RAM needed for storing the permanent variables. However, one can use various techniques to reduce the temporary RAM required. The first step is to identify the temporary variables and partition them into various sections on frame and subframe levels. Then, by carefully overlaying different sections, one can reduce the RAM requirements. Sometimes one can precalculate some of the intermediate variables and store them in ROM to save RAM. Some DSPs do not save the register contents while calling other subroutines. In those circumstances extra care must be taken in the beginning, and a systematic way of handling the call and return must be defined and strictly followed during implementation. One should structure the modules such that other modules can reuse some parts of the code. Extra care needs to be taken while writing interrupt service routines to handle various interrupts in the system. Even though the speech samples are coming in at 125  $\mu$ s intervals, there are other real-time requirements in any given cellular system. Some DSPs (e.g., Lucent's 1629) have hardware cache to speed up the execution and at the same time reduce power consumption. In that case one has to pay attention to the interrupt latencies involved. With the availability of flash DSPs from Lucent, one can speed up development of DSP code.

### ARCHITECTURES

With the definition of new and better speech coders, the equipment makers are faced with the challenges of supporting all the speech coders on a handset or base station in order to be compatible with other vendors. This has posed a difficult problem because the ROM space to store the coders and RAM usage are such that some combinations of coders cannot be supported easily on a single programmable DSP. One approach in that situation is to integrate multiple DSP cores and memory banks on a single chip. Another is to design a special ASIC hardware implementation of different speech coders. One can add hardware accelerators to current DSP cores, thereby increasing performance and reducing code size. There are some efforts to define DSPs with multiple MAC units in order to meet the ever-increasing demand for more MIPS. By increasing the number of registers used to access the RAM, DSPs can provide more MIPS with less code. Designing more complex addressing techniques, and increasing the on-chip RAM and ROM, one can accommodate increasing requirements for future applications.

## FUTURE TRENDS

### IC TECHNOLOGIES

The electronics industry has an appetite for integrating more and larger functions in ever smaller chips and devices [16]. No end appears to be in sight to transistor downsizing for the growth in IC functionality [4]. Technologies are being developed to shrink the transistor to 0.1  $\mu$ . Major challenges in

putting a billion transistors on a single chip are related to practical questions of manufacturing tolerances and yields. Microprocessors which run at 500 MHz are being designed now. It is desirable to create full systems on a chip; this requires advances in silicon technologies as well as support tools and methods. The complexity of large designs calls for the paradigm based on reusable high-level building blocks, or macroblocks. This paradigm requires a careful approach to dealing with block integration and testing [17]. The growth in the number of logic gates put on a chip is far outpacing the growth of device I/O, which bodes ill for controllability and observability. Chip-embedded control and testing (e.g., built-in self-test, BIST) is one of the strategies for dealing with these problems.

Low power consumption is a critical factor determining how widespread cellular communications are and will be. Future opportunities for low-power high-level integration will be governed by fundamental limits, materials, devices, circuits, and systems. There is no consensus on numbers achievable in practical settings. Some projections are shown in [19]. Pessimistically, in practical environments one cannot go below 0.25  $\mu$  technologies; optimistically, we shall reach 0.0625  $\mu$  sizes. Maximum chip dimensions may be anywhere in the area between 25 mm and 50 mm square.

### WIRELESS APPLICATIONS OF DSPS

There exists sufficient difference between analog RF/IF and baseband processing that the integration of the two is not easily achievable. Sound technical solutions to related problems do not exist [22]. Most integration problems are related to the noise generated by high-speed digital circuitry and self-induced codec noise due to PLL operation. The boundary between analog and digital is moving, and low-power fast A/Ds and D/As and high-speed DSPs are shifting it toward higher and higher frequencies.

Whereas current generation of cellular products mostly use off-the-shelf DSPs which are standalone, the trend toward integrating DSPs with microcontrollers, special-function hardware, and RF/IF interfaces is picking up steam. Some vendors are offering DSPs as cores through licensing agreements. The first mass market chip for the IS-95 standard is an ASIC which contains an embedded DSP for speech processing. Special-purpose DSPs based on DSP cores and built for large product runs are the likely direction in which the evolution of the DSP will occur. An example of a wireless-specialized device design based on a RISC microcontroller and a DSP can be found in [23].

Functions which are often used in wireless applications and are not supported by traditional DSPs are: complex arithmetic, division (such as needed for automatic gain control), coordinate rotation (useful for constellation-based algorithms in a receiver), and square root. Done on a traditional DSP architecture, a combined division and square root operation requires  $5N + 12$  cycles for  $N$ -bit operands [20]. It has been shown that coordinate rotation digital computer (CORDIC)-based approaches to these functions may provide significant speedup of execution.

The example of noninteger-related clocks for physical-layer and speech coding algorithms is an excellent motivator for pursuing single-chip DSP designs with multiple and independent arithmetic units [20, 21]. Due to close functional coupling between speech coding/decoding and physical layer, optimized memory-sharing approaches have to be designed. It is desirable to investigate options for independent interruptability of operation sequences executed on individual arithmetic logic units (ALUs). Another direction for improvement of DSPs is the functional improvement of ALUs.

Recent announcements of DSP vendors indicate that DSP architectures with multiple MACs will appear on the market in 1998. For instance, Lucent's DSP16000 core with two MACs can support the computation of the squares of differences between two sets of numbers with accumulation. A special three-operand adder has been included in the architecture for this purpose. This is a function useful for Viterbi decoding, one of the most widely used algorithms on the wireless receiver side. Atmel has announced a general-purpose 16-bit DSP core named LODE, with two 16 x 16 MACs, particularly targeted at wireless applications.

Measuring and comparing the performance of DSPs is not easy, particularly because different vendors have different approaches to the measurement. Lucent has recently suggested an "applications cube" approach to assess any DSP chip with respect to an application. The new unit presents power, performance, and cost as a measure of a function that is implemented in the processor. Consequently, power will be represented as milliamps per function at a particular voltage, performance as MIPS/function and cost by code-size/function.

An issue of hardware-software codesign for embedded systems is being actively pursued [18]. An example of how complex it is to design hardware and software that cooperate well is the problem of timing and synchronization for the IS-95 CDMA-based digital cellular standard. From setting parameters to control the length of correlation to updating the system time for a mobile receiver, every task requires coordinated hardware/software architectures.

Methodologies and tools for dealing with codesign are not widely known and used. Embedded design can be divided into four major tasks: partitioning the function into smaller pieces, allocating partitions to hardware, scheduling the times at which the functions are executed, and mapping a generic functional description into an implementation on a particular set of components as either software or the logic. In future, it is of considerable interest to automate these processes with the motivation of simplifying the designer's job as well as reducing the design error probability.

Reducing voltage supply levels is a desirable goal, and the trend is to go below 2 VDC. At these voltages, analog circuits start failing, and the number of available MIPS is reduced.

Future wide-area wireless systems will require higher data rates and thus impose tougher requirements on signal processing components [24,25]. Data rate evolution envisioned by system developers targets 144 kb/s, 384 kb/s, and 2 Mb/s for third-generation systems and as much as 32 Mb/s for the mobile broadband communications (MBS) project in Europe. Both low-tier and broadband wireless local loop systems [26,27] share many of the problems with cellular systems discussed in this article.

In the realm of speech coding, new speech coders will emerge as more progress is made in i) understanding the structure of speech signal, ii) understanding of how human hearing and perception work, iii) better quantization techniques to represent speech parameters, and iv) much faster DSPs with more advanced addressing and instruction sets.

### CONCLUSIONS

Digital signal processors have a prominent role in the design of cellular communications systems. They offer development flexibility and are used primarily for number-crunching operations in signal processing algorithms. DSPs are evolving by being integrated with microcontrollers, getting new functions and specialized accelerators. Their future in the world of advanced wireless communications is guaranteed.

## ACKNOWLEDGMENTS

The authors are grateful to Analog Devices, Lucent Technologies, and Texas Instruments for allowing the use of figures from their Web pages.

## REFERENCES

- [1] D. Goodman, "Trends in Cellular and Cordless Communications," *IEEE Commun. Mag.*, June 1991, pp. 31–40.
- [2] P. W. Baier, "Taking the Challenge of Multiple Access for Third Generation Cellular Mobile Radio Systems – A European View," *IEEE Commun. Mag.*, Feb. 1996, pp. 83–89.
- [3] E. A. Lee, "Programmable DSP Architectures," Parts 1 and 2, *IEEE ASSP*, Oct. 1988 and Jan. 1989.
- [4] P. M. Grant, "Signal Processing Hardware and Software," *IEEE Signal Processing Mag.*, Jan. 1996, pp. 86–88.
- [4] L. Geppert, "Solid State," Annual Technology Overview, *IEEE Spectrum*, Jan. 1997, pp. 55–59.
- [5] B. Rezavi, "Challenges in Portable Transceiver Design," *IEEE Circuits and Devices*, Sept. 1996, pp. 12–25.
- [6] B. C. Mather, "Embedding DSP," *IEEE Spectrum*, Nov. 1991, pp. 52–55.
- [7] J. Snyder et al., "Tools for Real-Time Signal-Processing Research," *IEEE Commun. Mag.*, Nov. 1993, pp. 64–74.
- [8] D. Cox, "Wireless Personal Communications, What Is It," *IEEE Pers. Commun.*, Apr. 1995, pp. 20–35.
- [9] K. S. Shanmugan, "Simulation and Implementation Tools for Signal Processing and Communication Systems," *IEEE Commun. Mag.*, July 1994, pp. 36–40.
- [10] Special Issue, "PCS—The Second Generation," *IEEE Commun. Mag.*, Dec. 1992.
- [11] D. Cox et al., "Low-Power Digital Radio as a Ubiquitous Subscriber Loop," *IEEE Commun. Mag.*, Mar. 1991, pp. 92–95.
- [12] H. Meyr and R. Subramanian, "Advanced Digital Receiver Principles and Technologies for PCS," *IEEE Commun. Mag.*, Jan. 1995, pp. 68–78.
- [13] J. Wepman, "Analog to Digital Converters and Their Applications in Radio Receivers," *IEEE Commun. Mag.*, May 1995, pp. 39–45.
- [14] J. Mitola, "The Software Radio Architecture," *IEEE Commun. Mag.*, May 1995, pp. 26–38.
- [15] R. Baines, "The DSP Bottleneck," *IEEE Commun. Mag.*, May 1995, pp. 46–54.
- [16] M. Hunt and J. Rowson, "Blocking in a System on a Chip," *IEEE Spectrum*, Nov. 1996, pp. 35–41.
- [17] R. Chandramouli and S. Pateras, "Testing Systems on a Chip," *IEEE Spectrum*, Nov. 1996, pp. 42–47.
- [18] W. Wolf, "Hardware-Software Co-Design of Embedded Systems," *IEEE Proc.*, vol. 82, no. 7, July 1994, pp. 967–89.
- [19] J. Meindl, "Low Power Microelectronics: Retrospect and Prospect," *IEEE Proc.*, vol. 83, no. 4, April 1995, pp. 619–34.
- [20] H. M. Ahmed, "Directions in DSP Processors," *IEEE JSAC*, vol. 8, no. 8, Oct. 1990.
- [21] H. Ahmed and R. Kline, "Recent Advances in DSP Systems," *IEEE Commun. Mag.*, May 1991, pp. 32–45.
- [22] R. Franzo and M. Diamondstein, "System Focus is Key to Designing Optimized Wireless DSPs," *Wireless Sys. Design*, Aug. 1997, pp. 35–38.
- [23] D. Walsh, "Integrated DSP/RISC Devices Target Cellular/PCS Phones," *Wireless Sys. Design*, July 1997, pp. 50–54.
- [24] M. Callender, "Future Public Land Mobile Telecommunications Systems," *IEEE Pers. Commun.*, 4th qtr. 1994, pp. 18–22.
- [25] Special Issue, "The European Path toward UMTS," *IEEE Pers. Commun.*, Feb. 1995.
- [26] C. Yu et al., "Low-Tier Wireless Local Loop Systems – Part 1 and 2: Comparison of Systems," *IEEE Commun. Mag.*, Mar. 1997.
- [27] W. Honcharenko et al., "Broadband Wireless Access," *IEEE Commun. Mag.*, Jan. 1997, pp. 20–26.
- [28] I. A. Gerson and M. A. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8kbs," *Proc. ICASSP*, 1990, pp. 461–64.
- [29] Gerson, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding for Japan Digital Cellular," *Proc. IEICE*, Nov. 15, 1990, pp. 35–40.
- [30] R. Salami et al., "A Toll Quality 8kb/s Speech Codec for Personal Commun. Systems," *IEEE Trans. Vehic. Tech.*, vol. 43, no. 3, Aug. 1994, pp. 808–16.
- [31] TIA/EIA/IS-641, "TDMA Cellular/PCS—Radio Interface — Enhanced Full-Rate (EFR) Speech Codec," June 1996.
- [32] TIA/EIA/IS-686, "TDMA Cellular/PCS — Radio Interface — Minimum Performance Standards for IS-641 Full-Rate Voice Codec," Dec. 1996.
- [33] AT&T Wireless Products, *Data Book*, AT&T Microelectronics, Sept. 1995.
- [34] *Speech and Audio Coding for Wireless and Network Applications*, B. S. Atal, V. Cuperman, and A. Gersho, eds., Kluwer, 1993.
- [35] *Speech Coding and Synthesis*, W. B. Kleijn, and K. K. Paliwal, Eds., Elsevier, 1995.
- [36] GSM 06.20 (ETS 300 581): "European Digital Cellular Telecommunications Systems: Half Rate Speech Transcoding," Jan. 1995.
- [37] K. Mano et al., "Design of a Pitch Synchronous Innovation CELP Codec for Mobile Communications," *IEEE JSAC*, vol. 13, no. 1, Jan 1995, pp. 31–41.
- [38] R. Cox, and P. Kroon, "Low Bit-Rate Speech Coders for Multimedia Communication," *IEEE Commun. Mag.*, Dec. 1996, pp. 34–41.
- [39] GSM 06.60 (ETS 300 726), "Digital Cellular Telecommunications Systems: Enhanced Full Rate (EFR) Speech Transcoding," ETSI, Nov. 1996.
- [40] EIA/TIA, "High Rate Speech Service Options for Wideband Spread Spectrum Communications Systems," IS96B, Feb. 1995.
- [41] EIA/TIA/IS-127, "Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems," Sept. 1996.
- [42] S. Kim, and W. Sung, "A Floating Point to Fixed Point Assembly Program Translator for the TMS 320C25," *IEEE Trans. CAS-II*, vol. 41, no. 11, Nov. 1994, pp. 730–39.

## BIOGRAPHIES

ZORAN I. KOSTIC [M] (kostic@research.att.com) received a Dipl. Ing. degree in electrical engineering from the University of Novi Sad, Yugoslavia, in 1987. In 1988 and 1991 he received M.S. and Ph.D. degrees in electrical engineering from the University of Rochester. He worked for AT&T Bell Laboratories from 1991 to 1996, and since then has been with AT&T Laboratories Research in the wireless communications systems research department. His research interests are in the areas of communications and digital signal processing. He currently works on wireless communications with emphasis on multiple access techniques, modulation and coding, high-resolution channel estimation, and low-complexity signal processing for wireless modems. His work spans theoretical, simulation, and real-time DSP/VLSI implementation aspects of communication systems. He is an editor of *IEEE Communications Letters*.

SELVARAJ SEETHARAMAN [M] (s.seetharaman@worldnet.att.net) received a B.E. (Honors) in electronics and communications engineering from the University of Madras, India, in 1978, an M.Tech in electrical engineering from the Indian Institute of Technology, Madras, in 1980, and a Ph.D. in systems design engineering from the University of Waterloo, Canada, in 1988. He has worked at IBM, Bell Northern Research, Teknekron Communications Systems Inc., AT&T, and Lucent Technologies. His work is focused on the implementation of speech coders and modems for various digital cellular standards. Presently he is working on xDSL products at GlobeSpan Technologies, Inc. His interests include speech coding, and DSP architectures and their applications in digital communications.