

A Software Solution for Chip Rate Processing in CDMA Wireless Infrastructure

Keld Lange, Alcatel Mobile Networks Division, Germany

Gero Blanke, Alcatel Research & Innovation, Germany

Rasekh Rifaat, Analog Devices Inc.

ABSTRACT

Third-generation cellular infrastructure requires extremely high-performance signal processing in the baseband receiver. Currently, chip rate processing is implemented using FPGA and ASIC technology. The use of a digital signal processor is explored for UTRA FDD systems with the goal of reducing cost and increasing flexibility. By combining chip rate and symbol rate processing within a single platform and taking advantage of the natural capacity of the air interface, load balancing can be performed, which reduces the amount of processing power needed, thereby reducing the cost of the receiver.

INTRODUCTION

Mobile radio and cellular technology has undergone significant changes over its history. There are currently four generations of mobile radio systems:

- Analog cellular systems
- Digital systems currently in use (e.g., GSM, IS-95, or PDC)
- Third-generation (3G) systems currently being developed primarily for multimedia and high-bit-rate data applications, as well as to increase capacity
- Fourth-generation systems that are still in their definition phase

Over this history, there has been a desire to reduce the number of different standards and protocols, and to improve interoperability between the remaining ones. Second-generation systems are incompatible mainly due to differing air interfaces and frequency bands. Third-generation systems [1], while originally being developed by separate standardization bodies, have (in general) joined forces and agreed to a com-

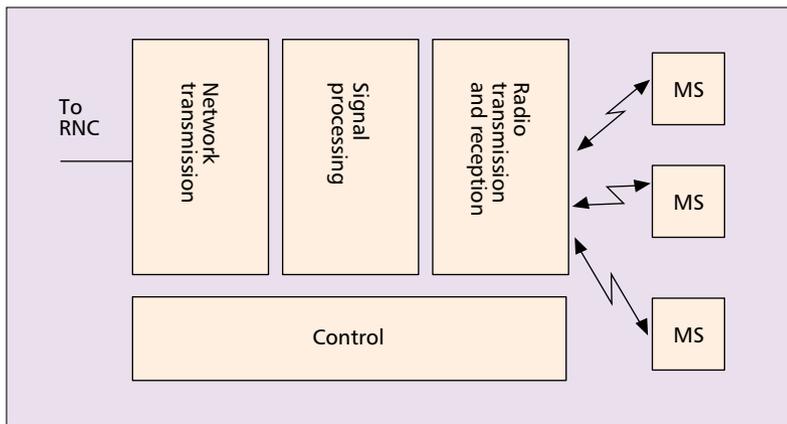
mon third-generation core network and an air interface consisting of three modes:

- Time-division duplex (TDD) based on Universal Terrestrial Radio Access (UTRA) TDD
- Multicarrier (MC) based on the cdma2000 multicarrier option
- Direct spread (DS) based on wideband code-division multiple access (WCDMA) or UTRA frequency-division duplex (FDD)

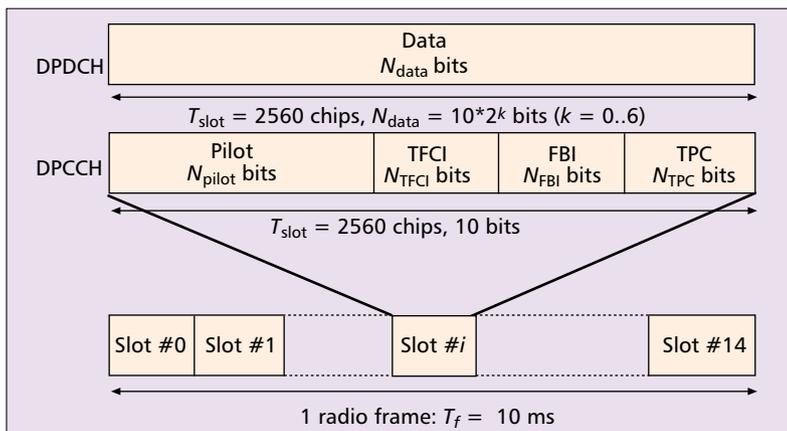
The flexibility of the UTRA FDD mode puts high requirements on its implementation, which we concentrate on in this article. First, we provide some perspective on the difficulty of providing a software solution for chip rate processing.

SPREADING AND DESPREADING

CDMA systems translate narrowband signals (a sequence of symbols) into signals with wider bandwidth (a sequence of chips) via a technique known as spreading. In simplified terms, a symbol to be transmitted is converted into a longer sequence of chips. This sequence is constructed by multiplying each complex single bit symbol with the time-varying complex single bit spreading code sequence, which is different for each channel. The length of the spreading code sequence used per symbol is also known as the spreading factor and is dependent on the channel type. For example, a 384 kb/s channel in UTRA FDD uses a spreading factor of 4 [2, 3]. The chip sequences are constructed in such a way that at the receiver, the original symbols may be estimated by applying a channel equalization and despreading operation on the multiple-bit IQ samples of the received signal. In order to determine the number of operations needed to perform this chip rate processing, we take a look at the UTRA FDD standard.



■ Figure 1. Functional architecture.



■ Figure 2. Frame structure for uplink DPDCH/DPCCH [2].

ARCHITECTURE AND FUNCTIONS OF UMTS BASE STATIONS

In UMTS, a system is set up through an air interface between the base station (node B) [4] and the mobile station (user equipment). Figure 1 shows a simplified functional architecture of a base station. The network transmission section connects the base station to a Radio Network Controller (RNC) utilizing an ATM or TCP/IP connection. The signal processing section performs both the chip rate and symbol rate processing needed for direct spread systems. Normally, the signal processing section has the largest computational power in the base station. Finally, the radio transmission and reception section perform the conversion between baseband signals and the air interface. A common controller commands each independent section.

UTRA FDD CHANNELS

UMTS characterizes different channels by the properties of the radio bearer (e.g., delay, guaranteed quality of service) without regard to the application using the channel (voice, text, video, fax, etc.). UMTS supports fixed-rate services, but compared to second-generation systems allows a great deal of flexibility in the channel utilization. Within the limits of spectrum availability, this leads to variable user data rates or effective band-

width-on-demand services. Packet data services can change the required radio resources within milliseconds, allowing for optimum throughput.

UTRA FDD is also able to collect several data streams into one coded composite transport channel (CCTrCH). For example, the output from an AMR voice codec is three separate bitstreams requiring different coding and quality of service levels. The output bits are mapped into three different transport channels that are processed according to their service requirements, and then multiplexed and transmitted together. Flexible bit rates also efficiently support multimedia applications, which could include highly variable voice, video, and file transfer components.

THE PHYSICAL LAYER STRUCTURE OF UTRA FDD

UTRA FDD channels can be segmented into radio frames and slots (Fig. 2). Each radio frame has a length of 10 ms, which at a chip rate of 3.84 Mc/s corresponds to 38,400 chips. In contrast to time-division multiple access (TDMA) systems, there is no frame or slot synchronization between the physical channels of different subscribers. The radio frames are again divided into 15 slots of 666.67 μ s or 2560 chips. One slot is the smallest unit available for issuing power control commands.

A layered approach exists for transmissions in UMTS where data is generated at higher layers and then transferred to transport channels. Transport channels define the characteristics and coding of the data over the air interface. For example, a dedicated channel (DCH) is the transport channel used to transmit all the higher-layer data of one user, consisting of both the application's data and control information.

For physical transmission, the transport channels are mapped to different physical channels. Physical channels are characterized by a specific carrier frequency, as well as by certain scrambling and channelization codes. For uplink, the physical channels are the uplink dedicated physical data channel (DPDCH) and the uplink dedicated physical control channel (DPCCH). The uplink DPDCH carries the DCH transport channel, while the uplink DPCCH carries control information for the physical layer. For uplink, the DPDCH and DPCCH are I/Q code multiplexed within each radio frame by each mobile.

In the transmitting mobile station, between 10 and 640 user bits are spread using spreading factors between 256 and 4, respectively, to create the 2560 chips transmitted in the uplink DPDCH. This corresponds to a DCH bit rate between 15 and 960 kb/s.

The DPCCH provides the information needed to determine the spreading factor used in the DPDCH along with other physical layer control information. The DPCCH always has a spreading factor of 256 and consists of 10 information bits split among:

- Known pilot bits to support channel estimation
- Transmit power control (TPC) commands
- Feedback information (FBI)
- An optional transport format combination indicator (TFCI)

The TFCI informs the base station which transport channels are active in the simultaneously transmitted uplink DPDCH radio frame, and which spreading factor is in use. The TFCI allows data transmission with a variable spreading factor on the DPDCH with changes allowed on a frame-by-frame basis. A 10-bit TFCI is protected by channel coding and thus extended to 32 bits. They are then split over the 15 slots of one frame with two bits in each slot, and two bits thrown away. The TFCI is transmitted in the same radio frame as the corresponding DPDCH bits in order to guarantee that if one frame is severely disturbed on the air interface, only that frame will be lost. However, it also means that the receiver is unable to start the final despreading until it has received a complete frame including all TFCI bits and decoded them. **This fact is at the heart of the UTRA FDD base station design.**

DESPREADER PRINCIPLES AND APPROACHES

There are two approaches to having to wait until a full frame is received before knowing the spreading factor of the frame. The first approach involves pre-despreading with the minimum possible spreading factor and storing the interim results before final despreading when the spreading factor is known. The second approach is to store all the received chips until the TFCI is decoded, and then to despread the data in a single pass.

With pre-despreading, the amount of memory needed is dependent on the minimum spreading factor, frame size, number of bits for the symbol representation, number of subscribers the base station needs to support, and whether or not channel compensation has already been performed. Assuming a 128-user system with a minimum spreading factor of 4, an efficient implementation of the channel compensation (at the final despreading stage) and a rake receiver with 8 fingers/subscriber, the memory storage required could easily exceed 19.7 Mbytes¹ with a bandwidth of 3.8 Gbytes/s² depending on the number of bits used for the representation. With channel compensation and finger combining at the despread stage the memory requirements decrease in the above example by a factor of 8 (2.5 Mbytes with a bandwidth of 0.5 Gbyte/s), but additional multiply and accumulate operations (MAC) on the order of 3.9 GigaMACs/s³ are needed.

For a single pass system, the amount of storage required depends on the frame size, oversampling rate, number of bits needed for a single chip representation, and the number of antennas. As well, an additional margin to avoid overwriting chips that have not yet been processed should be added. Per antenna there is a memory need of about 0.2 Mbytes⁴ regardless of the number of users. This memory has a bandwidth of 7.9 Gbytes/s⁵ for processing 128 users and 8 fingers each.

A common approach for chip rate processing is to use an ASIC or FPGA implementation with external memory. However, having either the required memory or the 3.9 GigaMAC/s of processing power pushes the edge of today's technologies. Even for the second case to incorporate such

a large buffer onto several ASICs requires sophistication only a few manufacturers currently offer.

The alternative approach presented here is to use the large internal memory of an array of commercial off-the-shelf digital signal processors (DSPs) as a buffer for a single-pass despreading system. There are several advantages to such a solution, which can be supported with today's DSPs.

IMPLEMENTATION OF A DATA DESPREADER USING DSPS

Despreading in UTRA FDD is the process of complex multiplying and accumulating the received IQ samples (chips) with the complex single bit channelization code used for spreading and the scrambling code used to separate different mobile stations. The scrambling code has a length of 38,400 chips (one radio frame) that are generated from a seed value provided by the higher layers. Because the scrambling code has been chosen to be as noiselike as possible the input signals have an extremely spiky auto-correlation function. Multipath transmissions are then resolved by aligning the despreading function to the various multipath delays and then correcting the phase of the received chips (channel equalization). These operations are implemented in a so-called finger of a rake receiver. The outputs of multiple fingers with different delays and operating on several antennas are combined to form a stronger estimate of the received symbol. This structure is known as a rake receiver and provides similar functionality to the equalizer in GSM systems.

The scrambling code can be reused for each finger of the rake receiver since the despreading procedure is identical to all time-delayed fingers. The symbols the despreaders generate are then multiplied with a channel compensation coefficient for each finger to remove the irregularities of the radio channel. Finally, each of the fingers is combined in a maximum ratio sense.

In order to implement a rake receiver, a large number of complex MAC operations is required. In particular, for a single UTRA FDD channel, the number of complex multiplies required is equal to the chip rate multiplied by the number of fingers in the antenna. For an 8-finger rake receiver and 128 users, the number of complex multiplies grows to 3.9 complex GigaMAC/s.⁶

A key property of these complex MACs is that the scrambling code is always $\pm 1 \pm j$. Because of this, true complex multiplies are not required, merely addition and negation units. For FPGA and ASIC implementations this has been exploited in many systems. Software solutions can also take advantage of this fact through dedicated co-processors, despreading instructions, or standard add and subtract operations.

PRACTICAL ISSUES FOR IMPLEMENTATION

In order to build a practical and cost-effective base station solution that can make use of the load balancing and software chip rate solution described, there are several properties that are

Despreading in UTRA FDD is the process of complex multiplying and accumulating the received IQ samples (chips) with the complex single bit channelization code used for spreading and the scrambling code used to separate different mobile stations.

¹ $3.84 \text{ MHz}[\text{chip rate}]/4[\text{SF4}] * 128[\text{users}] * 2 \text{ bytes}[\text{per symbol}] * 8[\text{fingers}]/100 \text{ Hz}[\text{frame rate}] = 19.6608 \text{ Mbytes}$

² $3.84 \text{ MHz}[\text{chip rate}]/4[\text{SF4}] * 128[\text{users}] * 2 \text{ bytes}[\text{IQ sample}] * 8[\text{fingers}] * 2[\text{read/write operations}] = 3.84 \text{ Gbytes/s}$

³ $3.84 \text{ MHz}[\text{chip rate}]/4[\text{SF4}] * 128[\text{users}] * 8[\text{fingers}] * 4\text{MAC}[\text{per complex multiply}] = 3.93216 \text{ GigaMAC}$

⁴ $3.84 \text{ MHz}[\text{chip rate}]/100 \text{ Hz}[\text{frame rate}] * 2[\text{oversampling factor}] * 2 \text{ bytes}[\text{IQ sample}] * 1.3[\text{margin}] = 0.19968 \text{ Mbyte}$

⁵ $3.84 * (128[\text{users}] * 8[\text{fingers}]) * 2 = 7864.32 \text{ Gbytes/s}$

⁶ $3.84 \text{ MHz}[\text{chip rate}] * 128[\text{users}] * 8[\text{fingers}] = 3.93216 \text{ GigaMAC}$

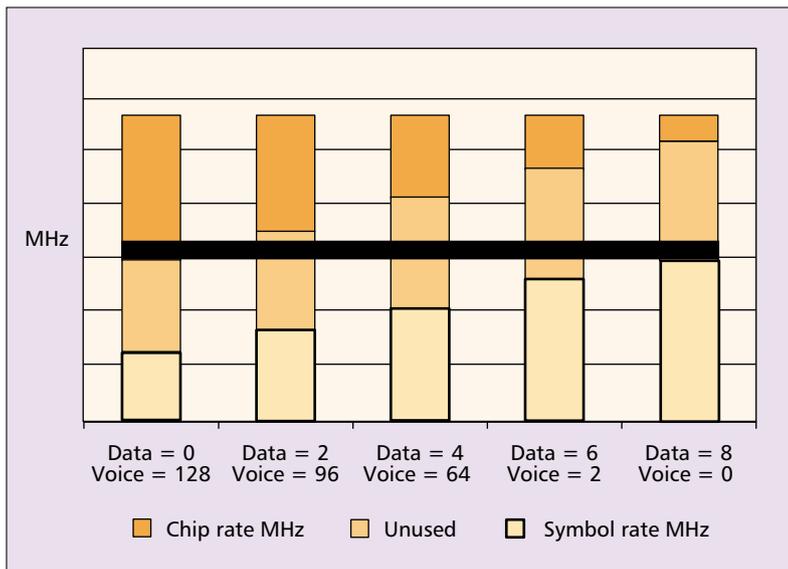


Figure 3. Resource requirements of a system that splits up chip rate and symbol rate processing.

needed in the DSP. This includes the processing power, memory configuration, and multiprocessor communication features.

Several high-performance DSPs are considered for the base station market including Analog Devices TigerSHARC®, Texas Instruments C6XXX series, and the jointly developed Starcore® product from Motorola and Agere Systems. Some of the processors include dedicated co-processors that assist in channel decoding functions. Another approach being promoted is to add general-purpose instructions that can be used as the core in a software solution and speed up functions such as add-compare-select and despreading [5].

A large memory buffer is required to store the antenna chip buffer for a full frame. One approach would be to use standard off-the-shelf SDRAM or SRAM external to a DSP that has a built-in memory controller. Designs such as this are limited due to the bandwidth of those mem-

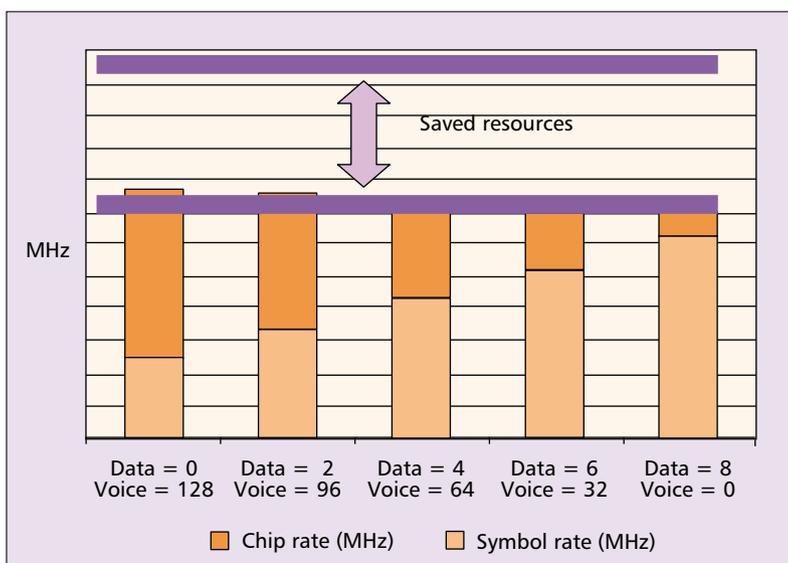


Figure 4. Saved resources of load balancing between voice and data users.

ories. However, off-the-shelf DSPs also have a large amount of onboard SRAM (4–7 Mb) with massive bandwidth capabilities suggesting that another approach is to split the memory load over several DSPs and then split the processing load accordingly as well. In this case, a highly efficient method for connecting and communicating within the cluster of DSPs is necessary. The ability to pass data from one DSP to another without additional external logic becomes an important factor in the design of the system.

Finally, while clock speed is an important factor in judging the processing power of a DSP, other factors such as internal bus bandwidth, ease of programming, ability to process interrupts, and number of parallel operations can play an important role in comparing DSPs regardless of clock speed.

LOAD BALANCING BETWEEN SYMBOL AND CHIP RATE PROCESSING

So far, a DSP alternative to common despreader implementations using ASICs or FPGAs has been described along with some of the features needed for a DSP in this solution. Implementing both symbol rate and chip rate processing on the same device permits a high degree of flexibility in resource allocation, depending on the actual channel mix. This is not possible with the conventional approach that takes separate devices for the different tasks. A pure DSP solution allows for load balancing between symbol rate and chip rate processing, and thus can save resources and ultimately cut down on the cost of the base station.

UTRA FDD offers all kinds of channels, from low-rate voice channels to high-rate data channels for multimedia applications or high-speed internet access. However, there is a limit to the total available transmission bandwidth or, equivalently, the capacity of the air interface reached by various channel mixtures. For instance, the air interface capacity used for transmission of about 128 AMR voice channels or 8 data channels with 384 kb/s is the same. Other cases involving various mixes of channels can also be considered, but it is easier to first analyze the system from these two extremes of air interface capacity.

The different channel types have specific requirements for the baseband processing parts of a base station. In particular, there are three groups of functions that have different processing requirements:

- **Proportional:** Some functions such as interleaving are more or less proportional to the number of bits in a channel. For a completely saturated air interface, the processing load caused by these functions is nearly independent of the channel types used.
- **Dependent:** Channel decoding functions such as Viterbi and Turbo decoding are extremely dependent on the type of channels being processed. For instance, Turbo decoders are more than an order of magnitude more complex than Viterbi decoders.

- Independent: Chip rate functions such as despreading are independent of channel type, but the amount of processing depends heavily on the number of channels being processed.

Because of the large number of operations needed for chip rate processing functions, the conventional design with ASICs/FPGAs for chip rate and DSPs for symbol rate had to be dimensioned with the worst-case demand for processing power. If we consider the voice channel case, the chip rate processing must be able to handle 128 simultaneous rake receivers along with 128 simultaneous Viterbi decoders. However, if the same system were to handle the eight 384 kb/s channels, only eight of the rake receivers would be utilized while the remaining 120 would lay idle. Additional processing power would be needed in the DSP since the number of operations needed for a single Turbo decoder is roughly that needed for the Viterbi decoding of 64 voice channels. As a result, a base station that has separate chip and symbol rate processing must be designed with the worst case for each. It is extremely useful to be able to visualize concepts such as this one. Figure 3 shows how the resources must be allocated to the maximum chip rate and symbol rate processing if they are separated. It is clear that a significant amount of processing power is wasted as the balance shifts between various data and voice mixes.

Consider a software solution that can combine both chip rate and symbol rate processing designed to handle the same situation; an interesting phenomenon occurs. The cycles that are dedicated to chip rate processing in the 128-voice-channel case can be switched to perform Turbo decoding for the high-data-rate case. Figure 4 shows the resources that can be saved by combining chip rate and symbol rate processing into software processing. The ratios and cycle counts for the various channel combinations are based on the Analog Devices TigerSHARC™ DSP. It is graphically visible how, with a software solution, processing power can be transferred from one task to another depending on the channel configuration. Proprietary systems based on a full software solution running on a group of DSPs are being developed for 3G wireless infrastructure rollout.

The same considerations also apply for any combination of the two cases mentioned, as well as with different channel types, as long as the total air interface capacity is not exceeded. A simple system might require that the number of supported channels of each type be known in advance and not be dynamic. With the software solution described here, a real-time operating system could be used to create a system that could dynamically switch the processing load depending on subscriber requirements without significant hardware design impact. This matches

perfectly the requirements of the UTRA FDD interface. A multiprocessor system similar to the one described here has been implemented using the Analog Devices TigerSHARC DSP.

CONCLUSION

After discussing the details of the UTRA FDD standard, we have seen that applying a software-based solution to data despreading enables several unique advantages including flexibility and cost saving because of the large antenna buffer required. Additionally, load balancing allows processing power that is wasted in traditional base station systems to be reused, thereby reducing the total amount of processing power needed in the system. Many of the ideas and concepts here can be applied to other wireless standards and can hasten the ability to provide a single software platform that can be deployed for multiple wireless standards, thereby reducing the system and production costs of the baseband processing.

REFERENCES

- [1] R. Prasad, W. Mohr, and W. Konhäuser, *Third Generation Mobile Communication Systems*, Artech House, 2000.
- [2] 3GPP Tech. Spec. 25.211 V3.8.0, "Physical Channels and Mapping of Transport Channels onto Physical Channels (FDD)," rel. 1999, Sept. 2001.
- [3] 3GPP, Tech. Spec. 25.212 V3.7.0, "Multiplexing and Channel Coding (FDD)," rel. 1999, Sept. 2001.
- [4] S. Breyer, B. Haberland, and E. Suzuki: "UMTS Node B Architecture in a Multi-standard Environment," *Alcatel Telecommun. Rev.*, 1st qtr. 2001; http://atr.alcatel.de/hefte/01i_1/gb/pdf_gb/10breyer gb.pdf (also available in Spanish, French, and German)
- [5] TigerSHARC DSP Instruction Set Specification Part# ADSP-TS101S, rev. 1.0, July 2001, Analog Devices, Inc.

BIOGRAPHIES

KELD LANGE received a Ph.D. in electrical engineering from the University of Karlsruhe, Germany, in 1997. He joined the Institute for Automation and Robotics at the University of Karlsruhe in 1990. At Alcatel's BTS competence center in 1996, he started developing hardware for GSM base station transceivers. Since then he has worked on architecture and implementation aspects of Alcatel's UMTS node B prototyping systems and now joins the Evolium (an Alcatel-Fujitsu company) node B development.

GERO BLANKE received a Dipl.-Ing. degree in electrical engineering from the Technical University of Brunswick, Germany, in 1990. As a member of Alcatel's mobile radio infrastructure development program, he worked on GSM base stations before joining Alcatel Research and Innovation in 1992. Since then he has participated in various mobile-radio-related research and development projects. His research interests include digital signal processing in third-generation mobile radio systems.

RASEKH RIFAAT (rasekh.rifaat@analog.com) graduated with B.Sc. and M.Sc. degrees in 1994 and 1998, respectively, from the University of Manitoba, Winnipeg, Canada. From 1994 to 1996 he worked on signal processing algorithms and software for seismic data signal processing. In 1999 he joined the DSP Wireless Infrastructure Systems Engineering group at Analog Devices where his research interests include advanced communications systems as well as DSP algorithms and architectures.

While clock speed is an important factor in judging the processing power of a DSP, other factors such as internal bus bandwidth, ease of programming, ability to process interrupts, and number of parallel operations can play an important role in comparing DSPs regardless of clock speed.