A CDMA Satellite ISDN Modem ASIC

Eric Aardoom[1], Bert Gyselinckx[1], Marc Engels[1], Ivo Bolsens[1] and Koen Verdijck[2]

[1] IMEC vzw, VSDM division,
Kapeldreef 75, B-3001 Leuven, Belgium
tel. +32 (0)16 281 406 fax. +32 (0)16 281 515
aardoom@imec.be

[2] SAIT Devlonics n.v.,
Steenweg op Ruisbroek 66,
B-1180 Brussels, Belgium

Abstract— **Code Division Multiple Access (CDMA) promises to become an enabling technology for the future generation of satellite communication services. Spread-spectrum modems for satellite communication currently on the market are targeted towards voice and facsimile communications at rates up to 10 kb/s using transmission bandwidths up to 5 MHz. However, emerging multimedia services such as videoconferencing, videophones and internet access in remote areas require ISDN-compatible data rates and more simultaneous users than can be supported with the current generation of satellite modems. The Satellite ISDN-rate Modem Base-station ASIC (SIMBA) is a high-speed CDMA modem ASIC currently being investigated at IMEC's VSDM division. The aim of SIMBA is to simplify the design of CDMA satellite base-stations and user terminals with optimal cost/performance, by integrating a complete baseband spread-spectrum transceiver on a single die together with a general-purpose microprocessor core. SIMBA also serves as a case study for a system design methodology that facilitates the design of heterogeneous hardware/software systems-on-a-chip, linking the integration of highly optimised application specific DSP blocks with state-of-the-art data converters, as well as commercially available macrocells such as the ARM7 embedded RISC controller and embedded memory blocks.**

Keywords— **Satellite communications, embedded systems, CDMA, spread-spectrum.**

I. Introduction

Telecommunication services in remote areas with low population density are still quite sparse and of low quality. With the advent of satellite communication technology, this situation is rapidly changing. Advanced telecommunication services are soon becoming available at affordable cost, as a result of the deployment of constellations of satellites in low earth orbit, but also through efficient utilisation of the currently installed infrastructure of geostationary satellite transponders. For example, through a network of satellite transponders remote PABXs and cellular base-stations (GSM) can be connected transparently to the public switched telephone network (PSTN). In the near future, such a network will require sufficient bandwidth to support multimedia services such as videophone, videoconferencing, and internet services.

The successful introduction of these new services depends heavily on the satellite network complexity and efficiency, which in turn is largely determined by the selected multiple access scheme. CDMA offers an efficient alternative to more conventional TDMA and FDMA access schemes. It requires minimum coordination between users, is resistant against fading and multipath propagation, is robust against hostile interference and enables transmission at low power spectral density. The main disadvantage of CDMA is its relative bandwidth inefficiency and its susceptibility to multi-user interference. Band-limited Quasi-Synchronous CDMA (BLQS-CDMA) provides a simple solution to these weaknesses through the introduction of spectral shaping and synchronisation of the user terminal transmissions [1].

The user terminals use digital signal processing techniques to meet performance requirements such as spectral purity, bit-error rate and acquisition time. Furthermore, weight, volume and power consumption requirements drive the demand for higher integration of the terminal electronics. The baseband modem is a key subsystem which determines many of the aforementioned performance characteristics.

After the performance of currently available baseband ASICs [2], [3] was evaluated, it became clear that these could not satisfy the needs of new multimedia services. The Satellite ISDN-rate Modem Base-station ASIC (SIMBA) is a CDMA modem ASIC for satellite communication, which aims to extend the usable range of applications beyond voice and facsimile communications to enable basic-rate ISDN connections of 144 kbit/s. This enables a SIMBA based design to support up to eight simultaneous toll quality calls with a single connection, and to satisfy the bandwidth requirements for multimedia services such as videophone, videoconferencing and internet access.

A similar development initiative by Roke Manor Research [4] focusses primarily on effective spread-spectrum demodulation techniques for VSAT (very small aperture terminal) and mobile applications at chip rates up to 12.7 Mchips/s [4], whereas SIMBA serves as a technology driver rather than a device optimised for any particular application. The objective of SIMBA is to extend performance to chip rates approaching 40 Mchips/s. Perhaps even more challenging is the desire to integrate as many functions as possible, including a complete transceiver chain, forward error correction (FEC) functions, an ARM7[1] embedded RISC controller core [5], embedded memory and high-speed serial interfaces on a single die.

This paper presents the design aspects of SIMBA. SIMBA runs at a clock frequency of over 100 MHz, necessitating the development of optimised high-speed DSP blocks, such as multi-rate FIR filters, numerically controlled oscillators (NCOs) and digital mixers. Additionally, synchronisation loops are controlled by dedicated hardware blocks. A substantial portion of the modem functionality is provided by software, which runs on the embedded microcontroller and is tightly coupled with the application-specific DSP functions and digital interfaces. As such, SIMBA is representative of the heterogeneous nature of telecommunication ASICs of the near future, and serves as a case study for design methodology for deeply embedded systems.

This paper is organised as follows. Section II describes the principles of a satellite BLQS-CDMA network. Section III gives an overview of a user terminal for a BLQS-CDMA satellite network employing a highly integrated modem ASIC. Section IV presents a detailed view on the transceiver section of the modem ASIC, which is a key element in providing the required performance for the selected application. In Section V we discuss ASIC design flow aspects for the defined spread-spectrum modem ASIC and provide a simulation result. Section VI contains some concluding remarks.

II. BLQS-CDMA Satellite Communication Networks

BLQS-CDMA is a flexible technique that can efficiently support point-to-point networks. This section discuss three important topics, that make BLQS-CDMA a competitive technique for point-to-point satellite networks: Network synchronisation, bandwidth-efficient modulation and power control. These topics have serious impact on the user terminal architecture, and consequently the architecture of the modem ASIC.

The basic concept of a BLQS-CDMA network is illustrated in Figure 1. There is a point-to-point link between a pair of user terminals A and B, which are being synchronised with the aid of the network coordination station (NCS). When a connection is being set up, the NCS assigns unique spreading codes to the terminals. The NCS also transmits a direct-sequence spread-spectrum pilot signal to which the terminals can synchronise their transmission.

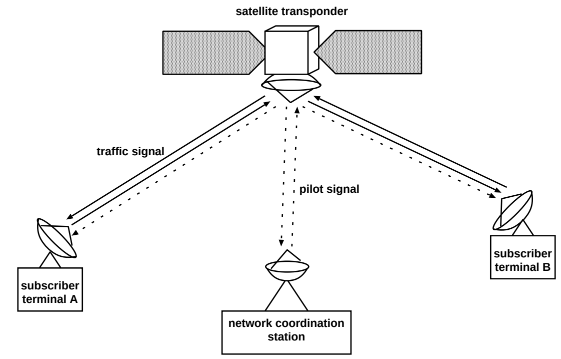

Fig. 1. Network concept.

A. Network Synchronisation

Network synchronisation is performed in roughly two steps: Frequency synchronisation and code phase synchronisation. The pilot signal transmitted by the NCS plays a crucial role in both steps.

Frequency synchronisation is based on Doppler shift measurements, and does not require coordination between the NCS and user terminals, or among user terminals. The NCS receives its own echo through the transponder, measures the frequency of the received pilot signal and corrects the frequency of the transmitted pilot signal to compensate for the Doppler shift on the forward link. This process establishes a zero Doppler pilot signal at the transponder input. A user terminal receives the pilot with a carrier frequency which includes only the return link Doppler

[1] ARM is a trademark of Advanced RISC Machines, Ltd.

shift, and uses the measured Doppler shift to compensate the frequency of its transmitted traffic signals. Consequently, the user terminal transmissions also have zero Doppler at the transponder inputs, and all forward link signals are synchronised in frequency. Code phase synchronisation aims to further minimise mutual interference between terminal transmissions. It requires phase difference measurements between the pilot and traffic signals at the transmitting or receiving terminal. Similar to the Doppler measurement, the terminal can demodulate its own traffic signal and measure the phase difference between the pilot and traffic signal. However, since connections are full-duplex, this would require an additional demodulator channel. Alternatively, the phase measurement can be performed by the receiving terminal as a side-effect of the demodulation process. The receiving terminal then transmits phase correction commands to the transmitting terminal as signalling information. This approach requires no additional demodulator channels in the receiver chain, but the loop delay is fairly long, and the loop time constant must be sufficiently large (on the order of several seconds) to avoid stability problems.

Since network synchronisation is a relatively slow and control-dominated process, it is most efficiently implemented in software, which has the additional benefit of algorithmic flexibility.

B. Bandwidth-Efficient Modulation

The available bandwidth is utilised efficiently by employing a combination of Nyquist pulse shaping of the transmitted signal, QPSK modulation and coherent demodulation. Nyquist pulse shaping minimises inter-symbol interference (ISI) and eliminates the spectral side-lobes characteristic of modulation with rectangular pulses. QPSK modulation doubles the data rate as compared to BPSK while occupying the same bandwidth. Coherent demodulation requires an E_b/N_0 of approximately 3 dB less than differential demodulation techniques for identical bit-error rate performance.

C. Power Control

Power control is essential for optimising CDMA network performance. The purpose of power control is to minimise mutual interference by ensuring that power levels from all transmitting user terminals are roughly the same at the input of the receiving user terminals. The user terminals can monitor the received pilot signal power to implement open-loop power control. Furthermore, remote receiving terminals can control the transmission power of the transmitting terminal by sending the appropriate signalling information on the return link. Therefore, the user terminal must be able to measure received power of the pilot and traffic signals, and to control the power of its own transmitted signal.

III. User Terminal architecture

The user terminal consists of a baseband section, an RF section and data converters connecting the baseband and RF sections, as shown in Figure 2. The modem ASIC includes all functionality of the baseband section, and performs most of the functions related to the data-flow oriented physical layer (layer 1 of the ISO/OSI reference model). The control-oriented data link layer (OSI layer 2) can be handled by a dedicated external protocol processor or the firmware of the embedded ARM7 microprocessor, while the optional network layer (OSI layer 3) would also be implemented in the firmware. The ARM7 also implements the interface to the external world through communication interfaces connected to the external ARM bus. To minimise overhead in high speed applications, dedicated hardware modules exchange data directly without any involvement from the microprocessor. In applications requiring less performance but more flexibility, the microprocessor can handle the data streams between the various modules through the internal ARM bus, enabling both continuous and burst transmission.

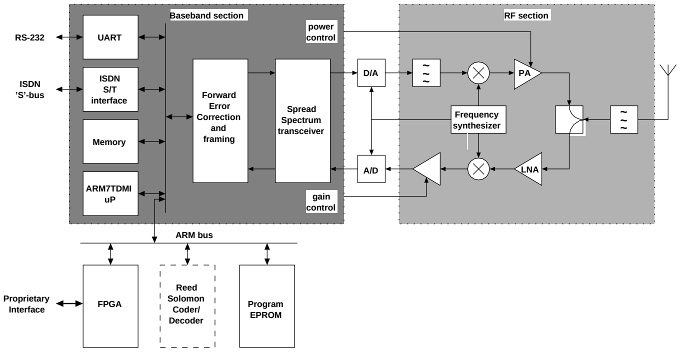

Fig. 2. User terminal.

Target functional and performance characteristics of the user terminal are shown in Table I.

TABLE I
Target Modem ASIC parameters

Multiple access scheme	BLQS-CDMA
Modulation formats	BPSK, DPSK, QPSK, QPN, OQPN
Spreading sequences	Gold, Walsh-Hadamard, M-sequences
Forward error correction	constraint-length 7 convolutional encoding
Coding rate	rate 1/2, 3/4, 5/6, 7/8
Max. symbol rate	200 ksym/s
Max. sample rate	110 MHz
Max. chip rate	36.67 MHz
Max. digital IF	27.5 MHz
Max. bandwidth (-3 dB)	50 MHz
Max. Doppler shift	20 kHz (Ku-band carrier)
Max. Doppler rate	100 Hz/s
Carrier tuning resolution	0.1 Hz
Max. implementation loss	less than 0.5 dB
Max. processing gain	30 dB (code length 1023)
Analogue interface	I and Q, 8 bit DAC/ 6 bit ADC
Synchronisation loops	fully digital
Nyquist pulse shaping	SRRC (roll-off = 0.35)

A. RF Section

The received signal enters the RF section through a diplexer which isolates the transmit and receive sections of the terminal. After the Low Noise Amplifier (LNA) this signal is down-converted to an IF suitable for digitisation. The signal to be transmitted is up-converted from the first IF to C-band or Ku-band carrier, amplified by the RF Power Amplifier (PA) and transmitted to the antenna through the antenna diplexer.

Transmitter and receiver power level are controlled by the power control and gain control outputs of the baseband section. These control outputs are programmed by the firmware of the ARM7.

B. Data Converters

High-speed A/D and D/A converters interface between the analogue and digital sections of the user terminal. Although it is desirable to integrate high-speed data converters on the same die as the digital baseband functions, the performance of state-of-the-art CMOS data converters can not satisfy the requirements for our application. Therefore, in the current design data converters are still external devices. However, eventually the data converters will be integrated on the same die as all digital baseband functionality.

C. Spread-Spectrum Transceiver

The baseband ASIC performs all modulation and demodulation operations. For this purpose it contains a dedicated spread-spectrum transceiver unit, which can simultaneously demodulate a pilot and a traffic signal. Several transceivers can be connected in master-slave mode to extend the communication capacity of a terminal.

All synchronisation and control loops (delay-lock loop, AFC loop, phase-lock loop and AGC loop) are closed in hardware and operate autonomously, unless the microprocessor takes over these control loops. The spread-spectrum transceiver is treated in more detail in Section IV.

D. Forward Error Correction & Framing

FEC and framing are performed either by dedicated modules, by firmware running on the embedded ARM7 processor or by a combination of both. Hardware FEC support consists of a constraint-length 7 convolutional encoder and a Viterbi decoder. Convolutional coding provides a coding gain of approximately 5 dB at the expense of bandwidth expansion by a factor of two. This bandwidth expansion is recovered using a bandwidth-efficient modulation scheme such as QPSK.

The framing functionality collects a packet of data and attaches a synchronisation marker to the head of this packet from the communication interface prior to modulation, while the de-framing function detects the synchronisation marker and signals frame synchronisation to the microprocessor after the demodulator.

Data communication applications may require better error-rate performance than convolutional coding can deliver alone. In this case, an external dedicated Reed-Solomon (RS) codec may be added. Alternatively, depending on throughput requirements, RS coding/decoding may be performed by the firmware. Additional block interleaving/de-interleaving functions can also be provided by firmware, as these operations can be implemented effectively by means of a special memory-addressing scheme for the buffer which stores the RS-encoded block of code symbols.

E. Microprocessor

The microprocessor runs the control-dominated part of the modem functionality such as the transceiver control finite state machine (FSM). Furthermore, depending on performance requirements, the firmware can implement an outer code to improve the end-to-end bit-error performance of the satellite link. Finally, the microprocessor handles communication with the external world through the on-chip interfaces (ISDN and RS-232) and proprietary external interfaces such as an ISDN S/T interface.

IV. Transceiver Architecture

The transceiver is divided into a transmitter and a receiver chain. The transmitter chain modulates the data coming from the FEC/framing unit, and converts the modulated waveforms to IF. The receiver chain performs the opposite operations, converting the received IF signal to baseband and demodulating this signal to obtain a data stream that is fed to the FEC/framing unit.

A. Transmitter Chain

A simplified block diagram of the transmitter chain is shown in Figure 3. It consists of a symbol mapper, a spread-spectrum modulator with programmable spreading sequence generator, a pulse shaping filter, a programmable-delay interpolator, a programmable up-converter, a programmable up-sampler, a fixed IF up-converter and a x/sin(x) pre-compensation filter. The ordering of blocks within the transmitter chain ensures that all blocks run at a minimum clock frequency.

Symbol Mapper: The symbol mapper receives a serial data stream from the FEC/framing module and maps this stream onto a signal constellation. The mapper supports BPSK, DPSK, QPSK, Offset QPSK and their differential counterparts.

Programmable sequence generator: This block generates a variety of spreading sequences such as Gold sequences with length up to 1023, and Walsh-Hadamard sequences with length up to 512.

Spread-Spectrum Modulator: The spread-spectrum modulator multiplies the symbols from the symbol mapper with spreading sequences. Optionally, QPSK symbols are spread with different sequences (Quadrature PN or QPN), which is equivalent to two orthogonal BPSK spread-spectrum signals. Furthermore, the modulation in the Q branch can be delayed by half a chip with relative to the I branch (Offset QPN).

Pulse Shaping Filter: The pulse shaping filter is an 18-th order linear-phase FIR filter with interpolation factor of three. It has a square-root raised-cosine frequency response with roll-off factor r = 0.35.

Programmable-delay Interpolator: This interpolator controls the code phase of the transmitted waveform, as required in a synchronous CDMA network to correct for code Doppler shifts on the forward link. Code phase and phase rate control (Doppler) is performed by a timing NCO (TNCO), which generates the desired phase shift as its output. This functionality replaces the programmable clock generators of [2], [3].

Programmable Up-Converter: The tunable up-converter enables fine-tuning of the transmitted carrier, as required in a synchronous CDMA network to correct for carrier Doppler shifts on the forward link. It is composed of a carrier numerically controlled oscillator (CNCO) and a phase rotator. The CNCO generates the rotation angle, while the phase rotator acts as a digital single-sideband mixer.

Programmable up-sampler: The up-sampler is a second-order Cascaded Integrator Comb (CIC) interpolation filter [6], which supports sample rate conversion factors of powers of two, ranging from 1 to 256.

Fixed Up-Converter: This up-converter translates the baseband spread-spectrum signal to a fixed IF of one fourth of the DAC sampling frequency. This up-converter can be disabled.

Pre-Compensation Filter: This filter compensates for the frequency attenuation as a result of the DAC zero-order hold response. Without this filter the upper and lower sidebands of the modulated IF signal would be

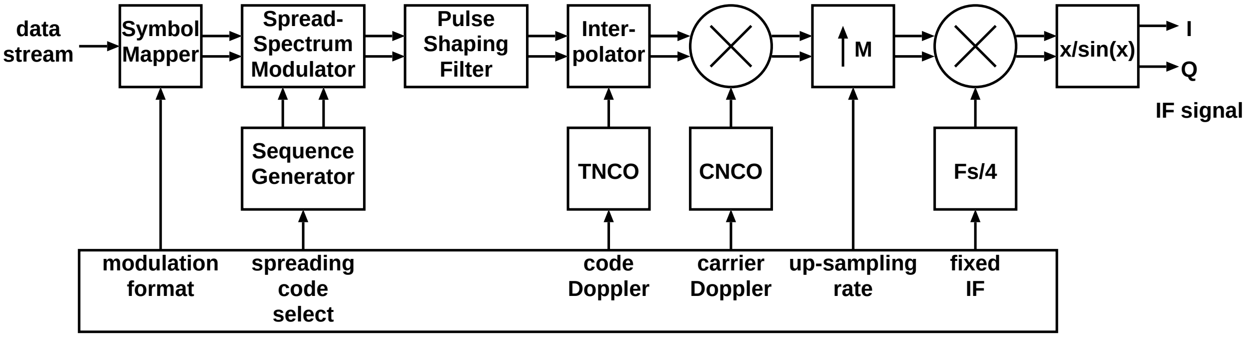

Fig. 3. Transmitter chain.

attenuated differently, resulting in a non-symmetric IF spectrum.

B. Receiver Chain

A simplified block diagram of the receiver chain is shown in Figure 4. The receiver chain consists of a received signal strength indicator (RSSI), a fixed IF down-converter, a down-sampler, a programmable down-converter, a receiver chip matched filter, and one or more coherent spread-spectrum demodulators (shaded area).

*RSSI:*The RSSI module derives an averaged signal level from the quadrature ADC inputs, and generates a control signal for an external AGC. The desired signal level is programmed by the firmware.

*Fixed down-converter:*This down-converter translates the quadrature input signals down by one fourth of the sampling rate. This block can be disabled.

*Programmable down-sampler:*The down-sampler is a second order CIC decimation filter, which performs the inverse operation of the up-sampler. It decimates the sampling rate by powers of two, ranging from 1 to 256.

*Programmable down-converter:*This down-converter is adjusted to eliminate the carrier Doppler shift on the received signal. It can be either controlled by dedicated PLL control logic or by the firmware.

*Chip matched filter:*The chip matched filter is an 18-order FIR filter, derived from the same prototype filter as the pulse shaping filter in the transmitter chain. This matched filter maximises the signal-to-noise ratio, and minimises ISI under imperfect synchronisation conditions. Due to its complexity, it is one of the more expensive blocks in terms of silicon area, speed requirements and power consumption. Consequently, its design requires careful analysis and optimisation.

*Spread-spectrum demodulators:*The demodulators acquire and track the code and carrier of the received signals and extract the data streams. As their performance has a large impact on the total system performance, the demodulator is a key subsystem of the transceiver section. This block is discussed in more detail in the following.

C. Spread-Spectrum Demodulator Architecture

Depending on the application requirements, the multiple spread-spectrum demodulator units can be instantiated. It is designed to minimise implementation imperfections, which cause signal distortion and SNR degradations. Specifically, the synchronisation algorithms are carefully selected to make optimal use of digital technology. In the following we will briefly describe the methods for chip timing and carrier recovery, and explain the reasons for the most important design decisions.

C.1 Chip Timing Recovery

Perhaps the most important design decision was to use a fixed sampling clock instead of a digitally-generated sampling clock as was used in previous designs [2], [3]. Rather, chip timing recovery is performed fully digitally by means of variable-delay interpolation of the chip matched filter output signal [7], [8]. The principal reason for this decision is that a digitally-generated sampling clock requires a master clock rate at least twice the fixed-rate sampling clock. Consequently, the projected bandwidth of 50 MHz requires a master clock frequency in excess of 200 MHz. Another important advantage of a fixed sampling clock is that multiple independent channels can process samples from a single ADC. These channels can track signals with different codes and different Doppler shift

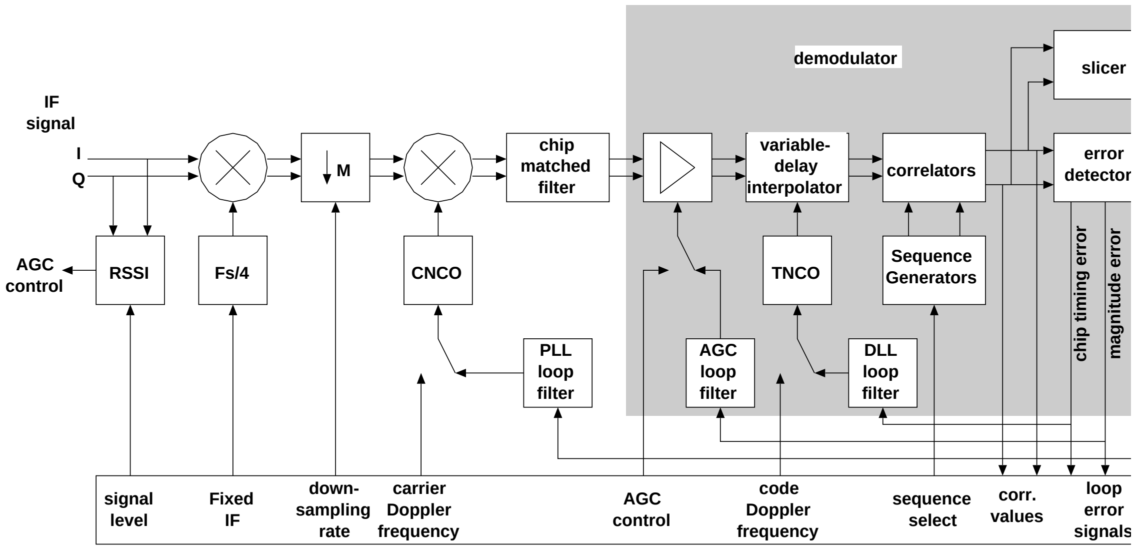

Fig. 4. Receiver chain.

without influencing each other. Finally, careful optimisation of the variable-delay interpolation filter can virtually eliminate implementation losses [9].

Chip timing correction is achieved by dynamically adjusting the delay of the input signal through the signal processing path. The input signal is generally sampled at a rate incommensurate to the chip rate of the received signal. The interpolator "reconstructs" signal samples as they would lie on a sampling grid commensurate to the chip rate. Consequently, interpolation is actually a sampling-rate conversion, which is controlled by the feedback mechanism of the chip timing recovery loop. This conversion is expressed mathematically as:

$$\begin{aligned}\hat{x}(kT) &= y(m_k + \mu_k)T_s \\ &= \sum_{i=I_1}^{I_2} x\left[(m_k - i)T_s\right] h_I\left[(i+\mu_k)T_s\right],\end{aligned}$$

where T denotes the output sampling period, T_s denotes the input sampling period, and μ_k denotes the time delay between both sampling instants kT and m_kT_s. $\hat{x}(.)$ is the interpolant, which results from filtering the input stream $x(.)$ with the interpolation filter h_I for the particular delay μ_k. This principle is illustrated in Figure 5, where interpolator output samples are synthesised from two adjacent samples of the input stream.

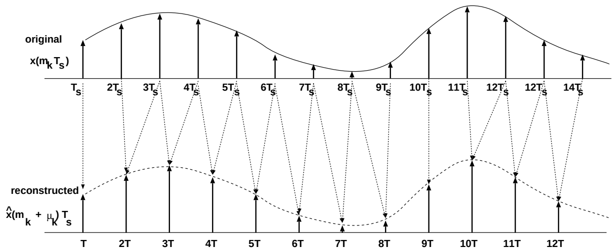

Fig. 5. Variable delay interpolation for chip timing recovery.

C.2 Carrier Recovery

Carrier recovery is based on well-known decision-directed carrier synchronisation techniques for suppressed-carrier signals [10, ch.6]. The carrier Digital PLL (DPLL) controls the down-converter. The carrier NCO (CNCO) generates a down-conversion carrier which is fed to the down-conversion mixer. When the loop is in lock, the CNCO frequency is equal the the carrier frequency of the incoming signal, and the signal spectrum at the output of the demodulator will be concentrated around baseband. This baseband signal is fed to the slicer, which outputs soft-decision data symbols.

V. ASIC Design Flow

In this section we outline the design flow of the modem ASIC. We opted to specify and simulate the architecture of our system using CoWare[2], a design environment for heterogeneous hardware/software sys-

[2] CoWare is a trademark of CoWare, Inc.

tems developed at IMEC [11]. CoWare supports integral specification, simulation, and synthesis of heterogeneous embedded systems, including application specific units, standard macrocells, and software components.

A. CoWare Design Flow

Specification in CoWare starts with an untimed behavioural C model of all functional modules, which are interconnected in a block diagram to form an untimed system model. At this stage the designer selects system parameters such as data word lengths, clock frequency, filter bandwidths, and tracking loop characteristics. The untimed system model is translated automatically into an executable specification, e.g., a C++ program that is compiled and executed directly on the development platform. The result is a very fast system simulation model which can be quickly modified and debugged, enabling extensive exploration of the relative merits of different algorithmic options. This model can be regarded as a prototype of the system, which can be reused and refined into an implementation, rather than being discarded after the initial analysis is performed.

Once the algorithms are frozen, the designer partitions the system into an architecture of hardware and software components. Hardware components can be dedicated blocks or third-party macrocells (IP blocks). At this stage the dedicated hardware blocks are refined to include timing aspects. The designer selects a microprocessor to run the software, and compiles the C software modules to machine code. This machine code then executes on an instruction-set simulator, and communicates with the hardware blocks through standard inter-process communication (IPC) mechanisms of the development platform operating system. Eventually, the timed and partitioned system model is simulated again to verify its correctness in the presence of timing constraints.

Interface synthesis is an important aspect of the methodology. Every function block is encapsulated in an interface, which enables seamless communication between blocks running on independent clocks by means of an asynchronous handshaking protocol. Function blocks are only enabled if there is valid data on the input or a subsequent block requests new data. This facility ensures that clocks are enabled only when data can be processed, a desirable feature for low-power and multi-rate system design. The encapsulation of function blocks also promotes the development and reuse of generic blocks, since there is a clear separation between the functional and communication behaviour.

During the last stage, hardware blocks are gradually refined to a final gate-level circuit. This circuit is then co-simulated with the embedded software running on a bus-cycle-accurate instruction-set simulator. As a final step, a structural VHDL description of all interfaces is automatically generated. Together with a VHDL model of the microprocessor and the gate-level function blocks, a final simulation of the complete system is performed to verify correctness of the implementation.

B. SIMBA Case Study

CoWare is currently used primarily for algorithmic exploration of the transceiver section. Transceiver performance has a large impact on the performance of the complete system, and therefore most of the design effort is being spent on optimising its performance. In the present design stage, untimed behavioural C models of individual modem blocks are developed, and these blocks are used to explore the design space in terms of sampling rate, word length, and ordering of operations in the transmitter and receiver chains.

The combination of fast C based simulation with the data analysis and visualisation capabilities of Matlab[3] provides a powerful tool for performing algorithmic trade-offs. As an example, Figure 6 shows the output frequency spectrum of a simulation data trace produced by cascading the spread-spectrum modulator, the pulse shaping filter, the CIC up-sampling filter, and the fixed up-converter, and feeding this setup with a random QPSK data stream. The plot shows that the spectral images of the baseband signal are attenuated by approximately 40 dB, due to the second order CIC filter. The up-sampling rate equals four.

VI. Conclusions

We have presented the requirements and some design aspects for the next generation modem ASICs for synchronous CDMA satellite networks. This development is driven by the demand for more advanced telecommunication services in remote areas. By virtue of the BLQS-CDMA technique, spread-spectrum promises to become an enabling technology for the cost-effective distribution of these services us-

[3] Matlab is a trademark of The MathWorks, Inc.

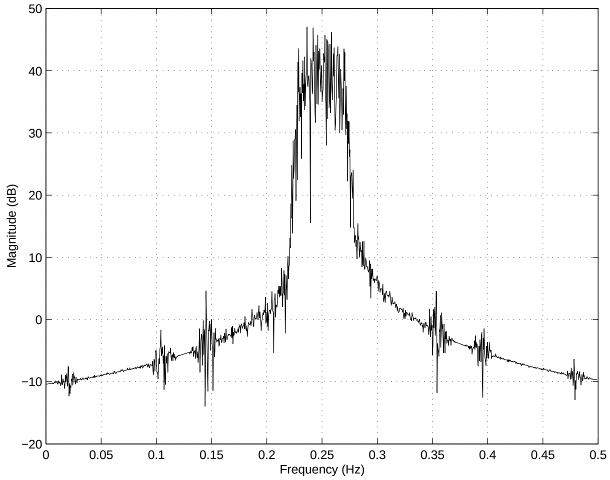

Fig. 6. CIC up-sampler output frequency spectrum (random data, M = 4).

ing the existing constellation of geostationary satellite transponders. The modem ASIC is a key component of base-stations and user terminals for BLQS-CDMA networks, justifying the effort to improve modem performance by increasing the level of functional integration. The high performance requirements for the modem algorithms and the high level of functional integration is representative for telecommunications systems of the near future.

VII. Acknowledgments

This work is supported by the European Space Agency, Noordwijk, The Netherlands, as part of the SCADES-3 project, contract 10208/92/NL/FM(SC). Marc Engels is a senior research assistant of the Belgium National Fund for Scientific Research.

References

[1] Riccardo De Gaudenzi, Carlo Elia, and Roberto Viola, "Bandlimited Quasi-Synchronous CDMA: A Novel Satellite Access Technique for Mobile and Personal Communciation Systems," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 2, pp. 328–343, Feb. 1992.

[2] Lieven Philips, Ivo Bolsens, and Hugo J. De Man, "A Programmable CDMA IF Transceiver ASIC for Wireless communications," in *Proc. Custom Integrated Circuit Conference, CICC'95*, 1995.

[3] Bert Gyselinckx, Luc Rijnders, Marc Engels, and Ivo Bolsens, "A 4*2.5 MChip/s Direct Sequence Spread-Spectrum Receiver ASIC with Digital IF and Integrated ARM6 Core," in *Proc. Custom Integrated Circuits Conference, CICC'97*, 1997.

[4] A.P. Hulbert, "A General-Purpose Digital Demodulator for VSAT and Mobile Spread Spectrum CDMA Signal Reception," in *Proc. 5th ESA International Workshop on DSP Techniques Applied to Space Communications*, 1996, pp. 11.45–11.59.

[5] Advanced RISC Machines, Ltd., "ARM7TDMI Data Sheet, ARM DDI 0029E," Aug. 1995.

[6] Eugene B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 2, pp. 155–162, Apr. 1981.

[7] Floyd M. Gardner, "Interpolation in digital modems–part i: Fundamentals," *IEEE Transactions on Communications*, vol. 41, no. 3, pp. 501–507, Mar. 1993.

[8] Lars Erup, Floyd M. Gardner, and Robert A. Harris, "Interpolation in digital modems–part ii: Implementation and performance," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 998–1008, June 1993.

[9] Vojin Živojnović and Heinrich Meyr, "Design of Optimum Interpolation Filters for Digital Demodulators," in *Proc. ISCAS-93*, 1993, pp. 140–143.

[10] John G. Proakis, *Digital Communications*, McGraw-Hill, third edition, 1995.

[11] Diederik Verkest, Karl Van Rompaey, Ivo Bolsens, and Hugo De Man, "CoWare–A Design Environment for Heterogeneous Hardware/Software Systems," *Design Automation for Embedded Systems*, vol. 1, pp. 357–368, 1996.