DSP-BASED MOBILE AND SATELLITE RECEIVERS, FROM ALGORITHM TO IMPLEMENTATION: A DESIGN COURSE AT AACHEN UNIVERSITY OF TECHNOLOGY

Oliver C. Mauss *Matthias Pankert** *Ferdinand Classen* *Heinrich Meyr*

Aachen University of Technology (RWTH)
Integrated Systems for Signal Processing, ISS - 611810
Templergraben 55, 52056 Aachen, Germany
mauss@ert.rwth-aachen.de

ABSTRACT

Profound knowledge of the interaction between algorithms and digital signal processor (DSP) architectures is required to be able to efficiently design complex communications equipment. Whereas both algorithms and architecture find treatment in many courses individually, education focusing on design methodology for DSP implementation is found to be rare. This contribution describes a concept and its implementation of a design course for DSP-based mobile and satellite communications systems attempting to fill the described gap. To illustrate the proposed concept in more detail, examples of the fall 1994 course are given.

1. INTRODUCTION AND MOTIVATION

Currently, we are witnessing an exploding demand for low-priced, but high-functionality devices for mobile communications. At the same time, the computing power of low-priced digital signal processor (DSP) chips has improved massively, making these well-suited as platform for the design of such systems. Along with major advances in hardware capabilities, computer aided design (CAD) tools have become extremely mature as well and have helped cut down design cycle time by enhancing both simulation and implementation phases of system design.

Obviously, for the development of complex communication equipment, system designers are required, possessing experience with advanced design tools and a sound understanding of the interaction between algorithm and architecture. For some years, this interaction has been in the central focus of interest of the Lab. for Integrated Systems for Signal Processing (ISS) at Aachen University of Technology along with the design of CAD tools to enable the exploration of this interaction. Next to pure research in the described fields, it has always been the objective to include current research results into the education process.

ISS offers to electric engineering students a palette of courses from basic to very specialized:

- Control and System Theory
- Algorithms for Digital Signal Processing
- Computer Aided Design of Digital Systems for Mobile Communications
- VLSI Design Course
- DSP-Based Receiver Design Course

The first two courses cover introductory and in-depth treatment of theoretical topics, respectively. The third is devoted to the concepts of combining communication theory, architectural knowledge and CAD tools to gain efficient system designs. It is felt at ISS, however, that no lecture is capable of replacing practical experience. Thus, two design courses are offered: one covering VLSI design, the second covering the design of programmable DSP-based receivers. It is the latter which is addressed in this paper, although many of the principal concepts hold for both design courses.

At most universities, courses on signal processing and communication theory are often augmented by laboratories where computer experiments help visualize certain effects. Additionally, there are courses covering hardware architectures in detail. Very little focus, however, is usually given to the design methodology itself, i.e. the question of which steps have proven to be successful during the design of a particular system when beginning from scratch. This gap has been identified and has lead to the development and implementation of a design course where DSP design methodology is one of the central educational goals.

The participants of the course are provided with methodological skills related to those two different design stages which have proven to be crucial for accomplishing given design tasks:

1. The techniques of using simulation tools to systematically construct a system fulfilling certain algorithm specifications.
2. The techniques of using implementation tools to transfer a simulation model efficiently to a DSP im-

*Matthias Pankert is now with Philips Research, P.O. Box 1980, 52021 Aachen, Germany.

0-7803-2431-5/95 $4.00

plementation optimized to a given set of implementation criteria.

In general, these two stages are not free of interaction. Until a design has reached completion, a multitude of choices have to be made along the way and by most of the choices algorithmic and implementational parameters are traded off. To an inexperienced designer these decision points and the potential options which may be chosen are not obvious. It is the objective of the design course to give an overview of the design space of the system together with a structured procedure for its exploration. Standard techniques for optimization in the different levels of the design are given. In this way, it is attempted to equip the course participants with a knowledge database of a more experienced user.

Within a realistic case study, the course participants are able to apply the newly acquired techniques to accomplish a given task. In contrast to textbook problems, there is no single solution to the problem. Many different approaches may lead to a valid solution. Additionally, optimization w.r.t. one performance parameter may lead to degradation w.r.t. another. This may be experienced best within a complex example where system parameters are required to be optimized while simultaneously monitoring the effects of the changes on system performance and implementation costs.

2. THE DESIGN COURSE APPROACH

In opposition to experimental laboratories with a strict list of topics to follow, the ISS course does not aim at visualization of classical DSP problems, e.g. FIR filter design, spectral estimation, FFT, etc. By focusing on design methodology, it augments these courses by providing the skills for systematic and efficient design of DSP based communication systems. In the course, several small competing teams are required to design a system according to given specifications and cost criteria, thereby creating a situation as close as possible to a 'real-world' product development. In contrast to textbook problems where a solution may either be correct or false, a multitude of solutions may exist to the problem. A change in a parameter in one component of the system may require altering the structure of another. The change may require the repetition of a design step already performed or it may even prevent finding a valid solution.

It is only in this manner, that the participants can receive an impression of the interdependence of the different sub-problems in a complex system and of the manifold obstacles that must be surpassed for successful system implementation. The system to be developed is taken from the area of digital radio receivers, where a strong interaction between algorithm and architecture exists.

The course consists of several phases:

1) Theoretical background: The requirements for the course are only basic communication and signal processing theory. For the design of the system, however, more detailed knowledge on specific problems e.g. in estimation and detection theory are required. The first part of the course therefore is dedicated to establishing and improving the participants' theoretical background required for the design tasks. Part of this material is to be prepared as prerequirement for the participation in this course.

2) Tool introduction: The use of graphical block-diagram oriented design tools has become industry standard. Within the course, the COSSAP[1] [1] design environment is employed. Next to system-level simulation, COSSAP offers a link to DSP realization through an interface with the DSP code synthesis tool DESCARTES[1] [2]. DESCARTES is capable of creating optimized DSP code from a netlist specification of a data flow block diagram for a wide variety of standard DSPs. In the second course phase, brief guided tours of the employed design software and the workstation environment are performed. Within one session, the participants take a very simple example from simulation to execution on a DSP.

3) Design Methodology: Up to this point, the course participants have been equipped with the additionally required theory and are acquainted with the usage of the CAD tools. The next course phase has the task of showing how the tools are efficiently used for simulation and implementation. E.g., there are many ways to model a receiver system within a graphical simulation environment, some of which have proven to be superior to others. The course participants are also equipped with some common strategies to follow for structured algorithm analysis and stepwise model refinement. Since it is common experience that the initial attempt of setting up a simulation model is unlikely to yield the expected results, some general hints for debugging of a simulation setup round off this part. In most cases, the mere down-loading of the simulation code to a DSP will not suffice to meet the run time or memory requirements. However, first-time DSP programmers are not acquainted with the techniques for writing efficient code for a DSP engine. Therefore, this course section additionally treats methods for improving the implementational performance of the system. This includes common optimization steps on blockdiagram level e.g. to achieve rate reductions, strategies such as vectorization and loop unfolding, code formulation to improve utilization of hardware parallelism, etc. Goal is to create a sensitivity for detecting suboptimum blockdiagram structures and inefficient code. Note that some optimization strategies may exclude one another and totally different strategies may lead to comparably good solutions. However it is an explicit desire to achieve different results among the groups

[1] COSSAP is a system design environment from Synopsys, Inc. (formerly CADIS GmbH), DESCARTES is a DSP code synthesis tool by ISS, Aachen Univ. of Technology.

4) Algorithm design: After having passed the initial course phases, the course participants have received a large amount of information on different design strategies. The students are thus in a situation comparable to a system engineer having to transfer previously gained knowledge to a new design project. It is impossible to exhaustively explore the design space within the limited time of the course. Additionally, hard bounds are given which must be met for certain performance parameters along with an overall cost criterion for the system which is to be optimized. Each of the competing teams for itself is required to design and optimize receiver components considering these design specifications. In this way, a realistic situation in a real-world project is simulated.

5) Implementation and optimization: After having designed the system and having evaluated and optimized its algorithmic performance, it is the task of transferring it to a real-time efficient implementation. Steps 4) and 5) are not free of interdependence, the probability is very high that the parameters that achieve adequate algorithm performance fail to fulfill one of the hard bound criteria. Thus it may be necessary to perform optimization steps and then reevaluate algorithmic performance.

6) Presentation & final discussion: In 'real world' projects, technical quality is only one part of a successful project, a convincing technical presentation is another. For this reason, the teams are required to present the achieved results and solutions to the entire group. Next to training technical presentations, the goal is to arise a discussion, highlighting drawbacks and benefits of specific design approaches.

The course concept supports flexibility. Both application and target platform are not fixed. The course contents are intended to be continuously updated and altered taking new and interesting tendencies in communications systems design into account. The tools and methodology of the course are attempted to be kept very close (if not identical) to those used in research at ISS.

3. CASE STUDY: THE DESIGN OF A DIGITAL QPSK SATELLITE RECEIVER

To give more insight, this section presents the contents of the fall 1994 course as an example. Here, a prototype digital quaternary phase shift keying (QPSK) receiver for a low-rate satellite communications link was to be designed. Next to the components required for demodulation and detection of the QPSK signal transmitted over a satellite channel, the receiver was required to possess phase error (θ) [3] and timing error (ε) [4] estimation and correction units. In trade-off for more in-depth coverage, it was decided to limit the receiver functionality. In particular, the coding and frequency synchronization schemes were omitted, which would be necessary in the real system. This limitation, however, does not influence the goal of the course. That design was considered superior

SNR operation range	$\geq$ 8 dB
max. tolerable implementation loss	$\leq$ 0.2 dB
min. data rate	$\geq$25 kbit/s
coherence time of θ, ε	$\leq$ 8 ms

Table 1. Receiver design specifications

to the others which minimized a simplified system cost metric

$$C_{tot} = C_P + (M_{D_e} + M_{P_e})C_{RAM} + C_{RAM,FX} + M_{P_e}C_{ROM} \quad (1)$$

under fulfillment of the hard boundary constraints of table 1. C_P are the processor costs depending on what clocking speed is required, M_{D_e} and M_{P_e} are the required amount of external data and external program memory, respectively, C_{RAM} and C_{ROM} are the associated variable memory costs per unit, and $C_{RAM,FX}$ accounts for a cost portion of the external memory which is independent of the required amount.

For first-generation prototyping, floating-point DSPs are better suited than their fixed-point counterparts, since they avoid having to perform time-consuming wordlength analysis. Additionally, the use of a floating point DSP for the receiver prototype allows to restrict the course to C-level programming, since C compilers for floating point architectures have proven to yield quite efficient code [5]. The Ariel Hydra V-C40 board has been used extensively at ISS as a prototyping platform in various designs. It provides two Texas Instruments TMS320C40 processors and is coupled to a workstation over a VME bus. In this manner, the hardware in the simulation loop feature of DESCARTES is able to download and execute code on the Hydra board as indicated in fig. 2. The code may be benchmarked in real-time while being fed with simulation data from the COSSAP simulation environment where transmitter and channel have been modeled. Note that all necessary interfacing code as well as automatic profiling

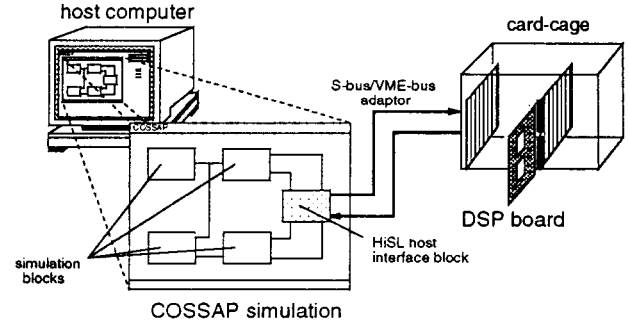

Figure 2. Hardware in the Simulation Loop

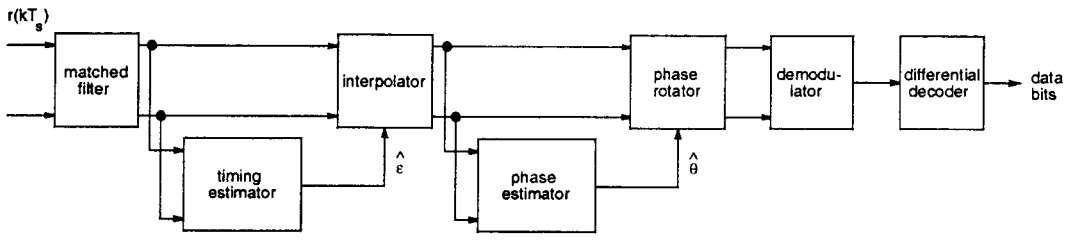

Figure 1. QPSK Satellite Receiver

is provided by the code generator.

For the above reasons, the C40 processor has been chosen as a platform for prototyping of the receiver within the course. Next to serving as a prototype, the results from this implementation are to be understood as an outlook for the processor requirements for a more suitable low-cost fixed point realization of the satellite receiver. Since the processor costs of the C40 processor would dominate the cost criterion (1), these have to be replaced by the costs of a potential candidate for a target processor in order to achieve a useful measure.

Most difficulties encountered by the participants during the course were associated with tool usage. As is usually the experience for most designs, the initial simulation setups failed to achieve the desired results. Due to their lacking of experience, the teams had problems to identify the proper debugging strategies to follow and signals to monitor. This especially became true in the final system design phases where, after having optimized the different components individually, these had to be integrated into the receiver system. In many cases, a bug in the system setup did not cause the system to entirely fail, but introduced a small but irreducible error into the statistical performance. Finding this bug within the complex block diagram often required extensive searching, causing frustration among the students. However, it was one of the intentions of the course to present system design as it really is and not as a clean and straightforward process where all of the difficulties had been eliminated a priori. Some other difficulties were associated with understanding and handling the implications of the stream-driven concept of the simulation tool where there is no explicit time reference.

Interesting enough, those students that attempted to solve the task with a pragmatic approach were the most successful. Others that had tried to perform DSP-specific code and block-diagram tuning already in the system design and simulation phase, got caught in the complexity of handling simultaneously the system design as well as their optimizations. Yet others had lost valuable time by investigating in too much detail the effects of certain parameters on performance. The most successful students had concentrated on rapidly achieving a first version of the receiver, leaving fine-tuning to the later phases. They had optimized the parameters one at a time, reworked a parameters only if necessary and had immediately proceeded to the next tasks after having found a useful solution. They had not come up with the most innovative solution, but had succeeded within the shortest time.

4. CONCLUSIONS

Hands-on experience with the interaction between algorithm and DSP architecture is a necessity for mastering complex system designs. The DSP-based digital receiver design course at ISS provides its participants with familiarity with state-of-the-art tools and relevant methodological concepts for communications systems design and implementation. It seeks to create a situation as closely as possible to the one found in real-world projects. Thereby, it attempts to close the gap in the educational process between communications theory, design software and DSP architectures. Additionally, the open concept provides the flexibility required to keep up with new developments.

REFERENCES

[1] Synopsys Inc., Herzogenrath, Germany, *COSSAP Reference Manual.*

[2] S. Ritz, M. Pankert, V. Živojnović, and H. Meyr, "High level software synthesis for the design of communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 348–358, April 1993.

[3] A. J. Viterbi and A. M. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission," *IEEE Trans. on Inf. Theory*, vol. 29, pp. 543–551, July 1983.

[4] M. Oerder and H. Meyr, "Digital filter and square timing recovery," *IEEE Transactions on Communications*, vol. 36, pp. 605–608, May 1988.

[5] O. Mauss, M. Pankert, and H. Meyr, "Rapid prototyping of a wireless modem for low rate data communications," in *ITG Fachbericht 127, Mikroelektronik für die Informationstechnik*, (Berlin), pp. 297–300, VDE Verlag, March 1994.